

---

# Rebase Helper Documentation

*Release 0.14.0*

**Petr Hracek**

**Tomas Hozza**

**Dec 21, 2018**



---

## Contents

---

<b>1</b>	<b>How to get it running?</b>	<b>3</b>
<b>2</b>	<b>How to use it?</b>	<b>5</b>
<b>3</b>	<b>What do I get from it?</b>	<b>7</b>
<b>4</b>	<b>How does it work?</b>	<b>9</b>
4.1	User Guide . . . . .	10
4.2	API . . . . .	14
4.3	Change Log . . . . .	38
	<b>Python Module Index</b>	<b>47</b>



There are several steps that need to be done when rebasing a package. The goal of **rebase-helper** is to automate most of these steps.



# CHAPTER 1

---

## How to get it running?

---

**rebase-helper** is packaged in Fedora, so you can just install it with **dnf**.

If you wish to use the latest codebase, consult [installation instructions](#).





## CHAPTER 2

---

### How to use it?

---

After installation, execute **rebase-helper** from a directory containing SPEC file, sources and patches (usually cloned dist-git repository).

Without any arguments or configuration **rebase-helper** will attempt to determine the latest upstream version automatically. If that fails, or if you wish to rebase to some different version, you can specify it explicitly as an argument:

```
$ rebase-helper 3.1.10
```

or you can pass source tarball filename instead:

```
$ rebase-helper foo-4.2.tar.gz
```

For complete CLI reference see [usage](#).

Alternatively, you can run **rebase-helper** in a container:

```
$ docker run -it -e PACKAGE=foo rebasehelper/rebase-helper:latest
```

See [docker reference](#) for more information.



---

What do I get from it?

---

**rebase-helper** always creates *rebase-helper-results* directory containing the following items:

Path	Description
<i>report.txt</i>	summary report with all important information
<i>changes.patch</i>	diff against original files, directly applicable to dist-git repo
<i>logs/</i>	log files of various verbosity levels
<i>rebased-sources/</i>	git repository with all modified files
<i>checkers/</i>	reports from individual checkers that were run
<i>old-build/</i>	logs and results of old (original) version build
<i>new-build/</i>	logs and results of new (rebased) version build



---

## How does it work?

---

The following steps describe a rebase process:

- **Preparation**

- *rebase-helper-workspace* and *rebase-helper-results* directories are created
- original SPEC file is copied to *rebase-helper-results/rebased-sources* directory and its Version tag is modified

- **Getting sources**

- old and new source tarballs are downloaded and extracted to *rebase-helper-workspace* directory
- old sources are downloaded from lookaside cache if possible

- **Downstream patches**

- new git repository is initialized and the old sources are extracted and committed
- each downstream patch is applied and changes introduced by it are committed
- new sources are extracted and added as a remote repository
- *git-rebase* is used to rebase the commits on top of new sources
- original patches are modified/deleted accordingly
- resulting files are stored in *rebase-helper-results/rebased-sources*
- diff against original files is saved to *rebase-helper-results/changes.patch*

- **Build**

- old and new source RPMs are created and built with selected build tool
- old SRPM and RPMs can also be downloaded from Koji to speed up the rebase

- **Comparison**

- multiple checker tools are run against both sets of packages and their output is stored in *rebase-helper-results/checkers* directory

- Cleanup
  - *rebase-helper-workspace* directory is removed

## 4.1 User Guide

### 4.1.1 Installation

**rebase-helper** is packaged in Fedora, so you can just install it with **dnf**.

If you can't or don't want to use **rebase-helper** package, you have to install, apart from Python requirements listed in *get\_requirements* function in *setup.py*, the following dependencies:

Dependency	Package name (in Fedora)	Notes
git	git	for rebasing of downstream patches
rpmbuild	rpm-build	for building SRPM and for <b>rpmbuild</b> build tool
mock	mock	for <b>mock</b> build tool, optional
koji	koji	for <b>koji</b> build tool, optional
rpmdiff	rpmlint	for <b>rpmdiff</b> checker, optional
abipkgdiff	libabigail	for <b>abipkgdiff</b> checker, optional
pkgdiff	pkgdiff	for <b>pkgdiff</b> checker, optional
csmock	csmock	for <b>csmock</b> checker, optional
licensecheck	licensecheck	for <b>licensecheck</b> checker, optional

### 4.1.2 Rebasing in container

**rebase-helper** can be run in Docker container. The package to be rebased has to be specified in `PACKAGE` environment variable. Alternatively, you can set `REPOSITORY` environment variable and point it to URL of any dist-git repository. In both cases, you can reference a specific branch, tag or commit by appending it to the package name or the repository URL:

```
$ docker run -it -e PACKAGE=foo#branch=f26 rebasehelper/rebase-helper:latest
--outputtool json
```

Results of the rebase will be stored in an exported volume.

### 4.1.3 Usage

#### SYNOPSIS

```
rebase-helper [-h] [--version] [-v] [--color {always,never,auto}]
  [--background {dark,light,auto}] [--results-dir RESULTS_DIR]
  [-p|-b|--comparepkgs-only COMPAREPKGS_DIR] [-c]
  [--buildtool {copr,rpmbuild,koji,mock}]
  [--srpm-buildtool {rpmbuild,mock}]
  [--pkgcomparetool {abipkgdiff,rpmdiff,licensecheck,pkgdiff,csmock}]
  [--outputtool {text,json}]
  [--versioneer {rubygems,pypi,anitya,hackage,cpan,npmjs}]
  [--versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmjs}]
```

```
[--spec-hook-blacklist {pypi-url-fix,paths-to-rpm-macros,typo-fix,
ruby-helper,commit-hash-updater}]
[--build-log-hook-blacklist {files}][--non-interactive]
[--favor-on-conflict {downstream,upstream,off}]
[--not-download-sources] [-w][--apply-changes]
[--disable-inapplicable-patches][--get-old-build-from-koji]
[--skip-version-check][--update-sources][--force-build-log-hooks]
[--builds-nowait][--build-tasks OLD_TASK,NEW_TASK]
[--builder-options BUILDER_OPTIONS]
[--srpm-builder-options SRPM_BUILDER_OPTIONS]
[--changelog-entry CHANGELOG_ENTRY][--config-file CONFIG_FILE]
[SOURCES]
```

## DESCRIPTION

**rebase-helper** is a tool which helps package maintainers to rebase their packages to latest upstream versions.

It should be executed from a directory containing spec file, sources and patches (usually cloned dist-git repository).

The new version is specified by *SOURCES* argument, which can be either version number or filename of the new source archive. Starting with version 0.10.0, this argument can be omitted and the new version determined automatically using one of available *versioneers*.

## OPTIONS

### Positional arguments

#### SOURCES

version number or filename of the new source archive

### Optional arguments

#### -h, --help

show help message and exit

#### --version

show rebase-helper version and exit

#### -v, --verbose

be more verbose

#### --color {always,never,auto}

colorize the output, defaults to auto

#### --background {dark,light,auto}

use color scheme for the given background, defaults to auto

#### --results-dir RESULTS\_DIR

directory where rebase-helper output will be stored

#### -p, --patch-only

only apply patches

- b, --build-only**  
only build SRPMs and RPMs
- comparepkgs-only** COMPAREPKGS\_DIR  
compare already built packages, COMPAREPKGS\_DIR must be a directory with the following structure:  
<dir\_name>/{old,new}/RPM
- c, --continue**  
continue previously interrupted rebase
- buildtool** {copr, rpmbuild, koji, mock}  
build tool to use, defaults to mock
- srpm-buildtool** {rpmbuild, mock}  
SRPM build tool to use, defaults to rpmbuild
- pkgcomparetool** {abipkgdiff, rpmdiff, licensecheck, pkgdiff, csmock}  
set of tools to use for package comparison, defaults to abipkgdiff, rpmdiff, licensecheck, pkgdiff if available
- outputtool** {text, json}  
tool to use for formatting rebase output, defaults to text
- versioneer** {rubygems, pypi, anitya, hackage, cpan, npmjs}  
tool to use for determining latest upstream version
- versioneer-blacklist** {rubygems, pypi, anitya, hackage, cpan, npmjs}  
prevent specified versioneers from being run
- spec-hook-blacklist** {pypi-url-fix, paths-to-rpm-macros, typo-fix, ruby-helper, commit-hash-updater}  
prevent specified spec hooks from being run
- build-log-hook-blacklist** {files}  
prevent specified build log hooks from being run
- non-interactive**  
do not interact with user
- favor-on-conflict** {downstream, upstream, off}  
favor downstream or upstream changes when conflicts appear
- not-download-sources**  
do not download sources
- w, --keep-workspace**  
do not remove workspace directory after finishing
- apply-changes**  
apply changes.patch after a successful rebase
- disable-inapplicable-patches**  
disable inapplicable patches in rebased SPEC file
- get-old-build-from-koji**  
do not build old sources, download latest build from Koji instead
- skip-version-check**  
force rebase even if current version is newer than requested version
- update-sources**  
update “sources” file and upload new sources to lookaside cache
- force-build-log-hooks**  
enforce running of build log hooks (even in non-interactive mode)



- builds-nowait**  
do not wait for remote builds to finish
- build-tasks** OLD\_TASK, NEW\_TASK  
comma-separated remote build task ids
- builder-options** BUILDER\_OPTIONS  
enable arbitrary local builder option(s), enclose BUILDER\_OPTIONS in quotes to pass more than one
- srpm-builder-options** SRPM\_BUILDER\_OPTIONS  
enable arbitrary local srpm builder option(s), enclose SRPM\_BUILDER\_OPTIONS in quotes to pass more than one
- changelog-entry** CHANGELOG\_ENTRY  
text to use as changelog entry, can contain RPM macros, which will be expanded
- config-file** CONFIG\_FILE  
path to a configuration file, defaults to \$XDG\_CONFIG\_HOME/rebase-helper.cfg

#### 4.1.4 Writing plugins

Starting with version 0.10.0, **rebase-helper** is extensible through plugins.

You can implement your own build tool, checker, output tool, SPEC hook or versioneer. All you have to do is to derive your plugin from corresponding base class, implement all necessary methods and register it using one of the following entry points:

Plugin type	Entry point	Base class
build tool	rebasehelper.build_tools	rebasehelper.build_helper. BuildToolBase
checker	rebasehelper.checkers	rebasehelper.checker.BaseChecker
output tool	rebasehelper. output_tools	rebasehelper.output_tool. BaseOutputTool
SPEC hook	rebasehelper.spec_hooks	rebasehelper.specfile.BaseSpecHook
versioneer	rebasehelper.versioners	rebasehelper.versioneer.BaseVersioneer

#### Example

Listing 1: my\_spec\_hook/\_\_init\_\_.py

```

from rebasehelper.specfile import BaseSpecHook

class MySpecHook(BaseSpecHook):

    NAME = 'MySpecHook'

    @classmethod
    def get_name(cls):
        return cls.NAME

    @classmethod
    def run(cls, spec_file, rebase_spec_file):
        """
        This method is called after original SPEC file is processed

```

(continues on next page)

(continued from previous page)

```

:param spec_file: SpecFile object representing original SPEC file
:param rebase_spec_file: SpecFile object representing rebased SPEC file
"""
rebase_spec_file.spec_content.insert(0, '# processed by %s\n' % cls.NAME)
rebase_spec_file.save()

```

Listing 2: setup.py

```

from setuptools import setup

setup(
    name='MySpecHook',
    version='0.1',
    description='Custom SPEC hook for rebase-helper',
    author='John Doe',
    install_requires=['rebasehelper>=0.10.0'],
    packages=['my_spec_hook'],
    entry_points={
        'rebasehelper.spec_hooks': ['my_spec_hook = my_spec_hook:MySpecHook']
    }
)

```

## 4.2 API

### 4.2.1 Helpers

#### Console helper module

**class** `rebasehelper.helpers.console_helper.ConsoleHelper`

Class for interacting with the command line.

**class** `Capturer` (*stdout=False, stderr=False*)

ContextManager for capturing stdout/stderr

**static** `color_is_light` (*rgb, bit\_width*)

Determines whether a color is light or dark.

#### Parameters

- **rgb** (*tuple*) – RGB tuple.
- **bit\_width** – Number of bits defining the RGB.

**Returns** Whether a color is light or dark.

**Return type** `bool`

**classmethod** `cprint` (*message, fg=None, bg=None, style=None*)

Prints colored output if possible.

#### Parameters

- **message** (*str*) – String to be printed out.
- **fg** (*str*) – Foreground color.

- **bg** (*str*) – Background color.
- **style** (*str*) – Style to be applied to the printed message. Possible styles: bold, faint, italic, underline, blink, blink2, negative, concealed, crossed. Some styles may not be supported by every terminal, e.g. 'blink'. Multiple styles should be connected with a '+', e.g. 'bold+italic'.

**static detect\_background** ()

Detects terminal background color and decides whether it is light or dark.

**Returns** Whether to use dark or light color scheme.

**Return type** str

**static exchange\_control\_sequence** (*query*, *timeout=0.05*)

Captures a response of a control sequence from STDIN.

**Parameters**

- **query** (*str*) – Control sequence.
- **timeout** (*int*, *float*) – Time given to the terminal to react.

**Returns** Response of the terminal.

**Return type** str

**static parse\_rgb\_device\_specification** (*specification*)

Parses RGB device specification.

**Parameters** **specification** (*str*) – RGB device specification.

**Returns** If the specification follows correct format, the first element is RGB tuple and the second is bit width of the RGB. Otherwise, both elements are None.

**Return type** tuple

**classmethod should\_use\_colors** (*conf*)

Determines whether ANSI colors should be used for CLI output.

**Parameters** **conf** (*rebasehelper.config.Config*) – Configuration object with arguments from the command line.

**Returns** Whether colors should be used.

**Return type** bool

**use\_colors = False**

## Copr helper module

**class** rebasehelper.helpers.copr\_helper.CoprHelper

**classmethod** **build** (*client*, *project*, *srpm*)

**classmethod** **create\_project** (*client*, *project*, *chroot*, *description*, *instructions*)

**classmethod** **download\_build** (*client*, *build\_id*, *destination*)

**classmethod** **get\_build\_status** (*client*, *build\_id*)

**classmethod** **get\_build\_url** (*client*, *build\_id*)

**classmethod** **get\_client** ()

```
classmethod watch_build(client, build_id)
```

### Download helper module

```
class rebasehelper.helpers.download_helper.DownloadHelper
```

Class for downloading files and performing HTTP requests.

```
static download_file(url, destination_path, blocksize=8192)
```

Downloads a file from HTTP, HTTPS or FTP URL.

#### Parameters

- **url** (*str*) – URL to be downloaded.
- **destination\_path** (*str*) – Path to where the downloaded file will be stored.
- **blocksize** (*int*) – Block size in bytes.

```
static progress(download_total, downloaded, start_time)
```

Prints current progress and estimated remaining time of a download to the standard output.

#### Parameters

- **download\_total** (*int*) – Total download size in bytes.
- **downloaded** (*int*) – Size of the already downloaded portion of a file in bytes.
- **start\_time** (*float*) – Time when the download started in seconds since epoch.

```
static request(url, **kwargs)
```

Performs an HTTP request or an FTP RETR command.

#### Parameters

- **url** (*str*) – HTTP, HTTPS or FTP URL.
- **\*\*kwargs** – Keyword arguments to be passed to `requests.session.get()`.

**Returns** Response object.

**Return type** `requests.Response`

### Git helper module

```
class rebasehelper.helpers.git_helper.GitHelper
```

Class which operates with git repositories

```
GIT_USER_EMAIL = 'rebase-helper@localhost.local'
```

```
GIT_USER_NAME = 'rebase-helper'
```

```
classmethod get_email()
```

```
classmethod get_user()
```

```
classmethod run_mergetool(repo)
```

### Input helper module

```
class rebasehelper.helpers.input_helper.InputHelper
```

Class for command line interaction with the user.

**static** `get_message` (*message*, *default\_yes=True*, *any\_input=False*)

Prompts a user with yes/no message and gets the response.

**Parameters**

- **message** (*str*) – Prompt string.
- **default\_yes** (*bool*) – If the default value should be YES.
- **any\_input** (*bool*) – Whether to return default value regardless of input.

**Returns** True or False, based on user’s input.

**Return type** bool

## Koji Helper module

**class** `rebasehelper.helpers.koji_helper.KojiHelper`

**classmethod** `create_session` (*profile='koji'*)

Creates new Koji session and immediately logs in to a Koji hub.

**Parameters** **profile** (*str*) – Koji profile to use.

**Returns** Newly created session instance.

**Return type** `koji.ClientSession`

**Raises** `RebaseHelperError` – If login failed.

**classmethod** `display_task_results` (*tasks*)

Prints states of Koji tasks.

**Parameters** **tasks** (*list*) – List of `koji.TaskWatcher` instances.

**classmethod** `download_build` (*session*, *build\_id*, *destination*, *arches*)

Downloads RPMs and logs of a Koji build.

**Parameters**

- **session** (*koji.ClientSession*) – Active Koji session instance.
- **build\_id** (*str*) – Koji build ID.
- **destination** (*str*) – Path where to download files to.
- **arches** (*list*) – List of architectures to be downloaded.

**Returns** List of downloaded RPMs and list of downloaded logs.

**Return type** tuple

**Raises** `DownloadError` – If download failed.

**classmethod** `download_task_results` (*session*, *tasklist*, *destination*)

Downloads packages and logs of finished Koji tasks.

**Parameters**

- **session** (*koji.ClientSession*) – Active Koji session instance.
- **tasklist** (*list*) – List of task IDs.
- **destination** (*str*) – Path where to download files to.

**Returns** List of downloaded RPMs and list of downloaded logs.

**Return type** tuple

**Raises** `DownloadError` – If download failed.

**functional = False**

**classmethod** `get_latest_build` (*session*, *package*)

Looks up latest Koji build of a package.

**Parameters**

- **session** (*koji.ClientSession*) – Active Koji session instance.
- **package** (*str*) – Package name.

**Returns** Found latest package version and Koji build ID.

**Return type** tuple

**classmethod** `get_old_build_info` (*package\_name*, *package\_version*)

Gets latest build info from koji.

**Parameters**

- **package\_name** (*str*) – Package name from specfile.
- **package\_version** (*str*) – Package version from specfile.

**Returns** Koji version, koji build id, package version, package full version.

**Return type** tuple

**classmethod** `get_task_url` (*session*, *task\_id*)

**classmethod** `upload_srpm` (*session*, *srpm*)

Uploads SRPM to a Koji hub.

**Parameters**

- **session** (*koji.ClientSession*) – Active Koji session instance.
- **srpm** (*str*) – Valid path to SRPM.

**Returns** Remote path to the uploaded SRPM.

**Return type** str

**Raises** `RebaseHelperError` – If upload failed.

**classmethod** `watch_koji_tasks` (*session*, *tasklist*)

Waits for Koji tasks to finish and prints their states.

**Parameters**

- **session** (*koji.ClientSession*) – Active Koji session instance.
- **tasklist** (*list*) – List of task IDs.

**Returns** Dictionary mapping task IDs to their states or None if interrupted.

**Return type** dict

## Lookaside cache helper module

**class** `rebasehelper.helpers.lookaside_cache_helper.LookasideCacheHelper`

Class for downloading files from Fedora/RHEL lookaside cache

**classmethod** `download` (*tool*, *basepath*, *package*)

```
rpkg_config_dir = '/etc/rpkg'
classmethod update_sources (tool, basepath, package, old_sources, new_sources)
```

## Macro helper module

```
class rebasehelper.helpers.macro_helper.MacroHelper
```

Class for working with RPM macros

```
MACROS_WHITELIST = ['_sysconfdir', '_bindir', '_libdir', '_libexecdir', '_sbindir', '_
```

```
static dump()
```

Gets list of all defined macros.

**Returns** All defined macros.

**Return type** list

```
static expand (s, default=None)
```

```
static expand_macros (macros)
```

Expands values of multiple macros.

**Parameters** **macros** (*list*) – List of macros to be expanded, macros are represented as dicts.

**Returns** List of macros with expanded values.

**Return type** list

```
static filter (macros, **kwargs)
```

Finds all macros satisfying certain conditions.

**Parameters**

- **macros** (*list*) – Macros to be filtered.
- **\*\*kwargs** – Filters to be used.

**Returns** Macros satisfying the conditions.

**Return type** list

```
static substitute_path_with_macros (path, macros)
```

Substitutes parts of a path with macros.

**Parameters**

- **path** (*str*) – Path to be changed.
- **macros** (*list*) – Macros which can be used as a substitution.

**Returns** Path expressed using macros.

**Return type** str

## Path helper module

```
class rebasehelper.helpers.path_helper.PathHelper
```

Class for performing path related tasks.

```
static file_available (filename)
```

Checks if the given file exists.

**Parameters** **filename** (*str*) – Path to the file.

**Returns** Whether the file exists.

**Return type** bool

**static find\_all\_files** (*top\_path, pattern*)

Recursively searches for all files matching the given pattern.

**Parameters**

- **top\_path** (*str*) – Directory where to start the search.
- **pattern** (*str*) – Filename pattern.

**Returns** List containing absolute paths to all found files.

**Return type** list

**static find\_all\_files\_current\_dir** (*top\_path, pattern*)

Searches for all files that match the given pattern inside a directory.

**Parameters**

- **top\_path** (*str*) – Directory where to start the search.
- **pattern** (*str*) – Filename pattern.

**Returns** List containing absolute paths to all found files.

**Return type** list

**static find\_first\_dir\_with\_file** (*top\_path, pattern*)

Recursively searches for a directory containing a file that matches the given pattern.

**Parameters**

- **top\_path** (*str*) – Directory where to start the search.
- **pattern** (*str*) – Filename pattern.

**Returns** Full path to the directory containing the first occurrence of the searched file. None if there is no file matching the pattern.

**Return type** str

**static find\_first\_file** (*top\_path, pattern, recursion\_level=None*)

Recursively searches for a file that matches the given pattern.

**Parameters**

- **top\_path** (*str*) – Directory where to start the search.
- **pattern** (*str*) – Filename pattern.
- **recursion\_level** (*int*) – How deep in the directory tree the search can go.

**Returns** Path to the file matching the pattern or None if there is no file matching the pattern.

**Return type** str

**static get\_temp\_dir** ()

Creates a new temporary directory.

**Returns** Path to the created directory.

**Return type** str



## Process helper module

**class** `rebasehelper.helpers.process_helper.ProcessHelper`

Class for executing subprocesses.

**DEV\_NULL** = `'/dev/null'`

**static** `run_subprocess(cmd, input_file=None, output_file=None)`

Runs the specified command in a subprocess.

### Parameters

- **cmd** (*iterable*) – A sequence of program arguments.
- **input\_file** (*str*, *typing.BytesIO*) – File to read the input from.
- **output\_file** (*str*, *typing.BytesIO*) – File to write the output of the command to.

**Returns** Exit code of the subprocess.

**Return type** `int`

**static** `run_subprocess_cwd(cmd, cwd=None, input_file=None, output_file=None, shell=False)`

Runs the specified command in a subprocess in a different working directory.

### Parameters

- **cmd** (*iterable*) – A sequence of program arguments.
- **cwd** (*str*) – Working directory for the command.
- **input\_file** (*str*, *typing.BytesIO*) – File to read the input from.
- **output\_file** (*str*, *typing.BytesIO*) – File to write the output of the command to.
- **shell** (*bool*) – Whether to run the command in a shell.

**Returns** Exit code of the subprocess.

**Return type** `int`

**static** `run_subprocess_cwd_env(cmd, cwd=None, env=None, input_file=None, output_file=None, shell=False)`

Runs the specified command in a subprocess in a different working directory with a redefined environment.

### Parameters

- **cmd** (*iterable*) – A sequence of program arguments.
- **cwd** (*str*) – Working directory for the command.
- **env** (*dict*) – Environment variables for the new process.
- **input\_file** (*str*, *typing.BytesIO*) – File to read the input from.
- **output\_file** (*str*, *typing.BytesIO*) – File to write the output of the command to.
- **shell** (*bool*) – Whether to run the command in a shell.

**Returns** Exit code of the subprocess.

**Return type** `int`

**static run\_subprocess\_env** (*cmd, env=None, input\_file=None, output\_file=None, shell=False*)

Runs the specified command in a subprocess with a redefined environment.

### Parameters

- **cmd** (*iterable*) – A sequence of program arguments.
- **env** (*dict*) – Environment variables for the new process.
- **input\_file** (*str, typing.BytesIO*) – File to read the input from.
- **output\_file** (*str, typing.BytesIO*) – File to write the output of the command to.
- **shell** (*bool*) – Whether to run the command in a shell.

**Returns** Exit code of the subprocess.

**Return type** int

## RPM helper module

**class** rebasehelper.helpers.rpm\_helper.**RpmHelper**

Class for working with RPM database and packages.

**ARCHES** = []

**static all\_packages\_installed** (*pkg\_names=None*)

Checks if all specified packages are installed.

**Parameters** **pkg\_names** (*list*) – List of package names to check.

**Returns** True if all packages are installed, False otherwise.

**Return type** bool

**static get\_arches** ()

Gets list of all known architectures

**static get\_header\_from\_rpm** (*rpm\_name*)

Gets an RPM header from the given RPM package.

**Parameters** **rpm\_name** (*str*) – Path to the package.

**Returns** Header object obtained from the package.

**Return type** rpm.hdr

**Returns**

**static get\_info\_from\_rpm** (*rpm\_name, info*)

Gets package name from an RPM file.

**Parameters**

- **rpm\_name** (*str*) – Path to the file.
- **info** (*bool*) – Which part of the RPM header to return.

**Returns** Package name obtained from the RPM file.

**Return type** str

**Returns**

**static install\_build\_dependencies** (*spec\_path=None, assume\_yes=False*)

Installs build dependencies of a package using dnf.

**Parameters**

- **spec\_path** (*str*) – Absolute path to the SPEC file.
- **assume\_yes** (*bool*) – Whether to automatically answer ‘yes’ to all questions.

**Returns** Exit code of the subprocess run.

**Return type** int

**static is\_package\_installed** (*pkg\_name=None*)

Checks whether a package is installed.

**Parameters** **pkg\_name** (*str*) – Name of the package.

**Returns** Whether the package is installed.

**Return type** bool

**classmethod parse\_spec** (*path, flags=None*)

**classmethod split\_nevra** (*s*)

Splits string into name, epoch, version, release and arch components

## 4.2.2 Application module

**class** rebasehelper.application.**Application** (*cli\_conf, execution\_dir, results\_dir, debug\_log\_file*)

**apply\_changes** ()

**build\_binary\_packages** ()

Function calls build class for building packages

**build\_source\_packages** ()

**debug\_log\_file** = None

**static extract\_archive** (*archive\_path, destination*)

Extracts given archive into the destination and handle all exceptions.

**Parameters**

- **archive\_path** – path to the archive to be extracted
- **destination** – path to a destination, where the archive should be extracted to

**Returns**

**static extract\_sources** (*archive\_path, destination*)

Function extracts a given Archive and returns a full dirname to sources

**generate\_patch** ()

Generates patch to the results\_dir containing all needed changes for the rebased package version

**get\_all\_log\_files** ()

Function returns all log\_files created by rebase-helper First if debug log file and second is report summary log file

**Returns**

**get\_checker\_outputs** ()

**get\_new\_build\_logs** ()

**get\_rebased\_patches** ()

Function returns a list of patches either “: [list\_of\_deleted\_patches] :return:

**get\_rebasehelper\_data** ()

**get\_rpm\_packages** (*dirname*)

Function returns RPM packages stored in *dirname/old* and *dirname/new* directories

**Parameters** *dirname* – directory where are stored old and new RPMS

**Returns**

**kwargs** = {}

**new\_rest\_sources** = []

**new\_sources** = ''

**old\_rest\_sources** = []

**old\_sources** = ''

**patch\_sources** (*sources*)

**prepare\_next\_run** (*results\_dir*)

**prepare\_sources** ()

Function prepares a sources.

**Returns**

**print\_summary** (*exception=None*)

Save rebase-helper result and print the summary using *output\_tools\_runner* :param *exception*: Error message from rebase-helper :return:

**print\_task\_info** (*builder*)

**rebase\_spec\_file** = None

**rebase\_spec\_file\_path** = None

**rebased\_patches** = {}

**rebased\_repo** = None

**report\_log\_file** = None

**result\_file** = ''

**run** ()

**run\_package\_checkers** (*results\_dir*, *\*\*kwargs*)

Runs checkers on packages and stores results in a given directory.

**Parameters**

- **results\_dir** (*str*) – Path to directory in which to store the results.
- **category** (*str*) – checker type(SOURCE/SRPM/RPM)

**Returns** None

**static setup** (*cli\_conf*)

**static setup\_logging** (*results\_dir*)

Adds file handlers of various verbosity to loggers.

**Parameters** *results\_dir* – Path to directory which results are stored in.

**Returns** Path to debug log.

**Return type** string

`spec_file = None`

`spec_file_path = None`

`temp_dir = ''`

### 4.2.3 Archive module

**class** `rebasehelper.archive.Archive` (*filename=None*)

Class representing an archive with sources

**extract\_archive** (*path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `get_supported_archives` ()

Return list of supported archive types

**class** `rebasehelper.archive.ArchiveTypeBase`

Base class for various archive types

**EXTENSION** = ''

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `match` (*filename=None*)

Checks if the filename matches the archive type. If yes, returns True, otherwise returns False.

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

**class** `rebasehelper.archive.Bz2ArchiveType`

.bz2 archive type

**EXTENSION** = '.bz2'

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

**class** `rebasehelper.archive.GemPseudoArchiveType`

.gem files are not archives - this is a pseudo type

**EXTENSION** = '.gem'

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

**class** `rebasehelper.archive.TarArchiveType`

.tar archive type

**EXTENSION** = `' .tar '`

**class** `rebasehelper.archive.TarBz2ArchiveType`

.tar.bz2 archive type

**EXTENSION** = `' .tar.bz2 '`

**class** `rebasehelper.archive.TarGzArchiveType`

.tar.gz archive type

**EXTENSION** = `' .tar.gz '`

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

**class** `rebasehelper.archive.TarXzArchiveType`

.tar.xz archive type

**EXTENSION** = `' .tar.xz '`

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

**class** `rebasehelper.archive.TgzArchiveType`

.tgz archive type

**EXTENSION** = `' .tgz '`

**class** `rebasehelper.archive.ZipArchiveType`

.zip archive type

**EXTENSION** = `' .zip '`

**classmethod** `extract` (*archive=None, filename=None, path=None*)

Extracts the archive into the given path

**Parameters** `path` – Path where to extract the archive to.

**Returns**

**classmethod** `match` (*filename=None*)

Checks if the filename matches the archive type. If yes, returns True, otherwise returns False.

**classmethod** `open` (*filename=None*)

Opens archive with the given filename and returns the proper archive type object.

`rebasehelper.archive.register_archive_type` (*archive*)

## 4.2.4 Build helper module

**exception** `rebasehelper.build_helper.BinaryPackageBuildError` (*\*args, \*\*kwargs*)

Error indicating failure to build Binary Package

**class** `rebasehelper.build_helper.BuildHelper`

`get_all_tools` ()

`get_default_tool` ()

`get_supported_tools` ()

`get_tool` (*tool*)

**class** `rebasehelper.build_helper.BuildTemporaryEnvironment` (*sources, patches, spec, results\_dir*)

Class representing temporary environment.

`TEMPDIR_RESULTS` = 'TEMPDIR\_RESULTS'

`TEMPDIR_SOURCES` = 'TEMPDIR\_SOURCES'

`TEMPDIR_SPEC` = 'TEMPDIR\_SPEC'

`TEMPDIR_SPECS` = 'TEMPDIR\_SPECS'

**class** `rebasehelper.build_helper.BuildToolBase`

Build tool base class.

**DEFAULT**

*bool* – If True, the build tool is default tool.

**ACCEPTS\_OPTIONS**

*bool* – If True, the build tool accepts additional options passed via `-builder-options`.

**CREATES\_TASKS**

*bool* – If True, the build tool creates remote tasks.

**ACCEPTS\_OPTIONS** = **False**

**CREATES\_TASKS** = **False**

**DEFAULT** = **False**

**classmethod** `build` (*spec, results\_dir, srpm, \*\*kwargs*)

Build binaries from the sources.

Keyword arguments: `spec` – path to a SPEC file `sources` – list with absolute paths to SOURCES `patches` – list with absolute paths to PATCHES `results_dir` – path to DIR where results should be stored

Returns: dict with: `'srpm'` -> absolute path to SRPM `'rpm'` -> list of absolute paths to RPMs `'logs'` -> list of absolute paths to logs

**static** `get_builder_options` (*\*\*kwargs*)

**classmethod** `get_detached_task` (*task\_id, results\_dir*)

Gets packages and logs for specified task

**Parameters**

- **task\_id** – detached task id
- **results\_dir** – path to DIR where results should be stored

**Returns** tuple with: list of absolute paths to RPMs list of absolute paths to logs

**classmethod** `get_logs()`

Get logs from previously failed build Returns: dict with 'logs' -> list of absolute paths to logs

**classmethod** `get_task_info(build_dict)`

Gets information about detached remote task

**Parameters** **build\_dict** – build data

**Returns** task info

**classmethod** `prepare(spec, conf)`

Prepare for building.

**Parameters** **spec** – spec file object

**classmethod** `wait_for_task(build_dict, task_id, results_dir)`

Waits until specified task is finished

**Parameters**

- **build\_dict** – build data
- **results\_dir** – path to DIR where results should be stored

**Returns** tuple with: list of absolute paths to RPMs list of absolute paths to logs

**class** `rebasehelper.build_helper.MockTemporaryEnvironment(sources, patches, spec, results_dir)`

Class representing temporary environment for MockBuildTool.

**class** `rebasehelper.build_helper.RpmbuildTemporaryEnvironment(sources, patches, spec, results_dir)`

Class representing temporary environment for RpmbuildBuildTool.

**TEMPDIR\_BUILD** = 'TEMPDIR\_BUILD'

**TEMPDIR\_BUILDROOT** = 'TEMPDIR\_BUILDROOT'

**TEMPDIR\_RPMBUILD** = 'TEMPDIR\_RPMBUILD'

**TEMPDIR\_RPMS** = 'TEMPDIR\_RPMS'

**TEMPDIR\_SRPMS** = 'TEMPDIR\_SRPMS'

**class** `rebasehelper.build_helper.SRPMBuildHelper`

`get_all_tools()`

`get_default_tool()`

`get_supported_tools()`

`get_tool(tool)`

**class** `rebasehelper.build_helper.SRPMBuildToolBase`

SRPM build tool base class.

**DEFAULT**

*bool* – If True, the build tool is default tool.



**DEFAULT = False**

**classmethod build** (*spec, results\_dir, \*\*kwargs*)  
Build SRPM with chosen SRPM Build Tool

**Parameters**

- **spec** – SpecFile object
- **results\_dir** – absolute path to DIR where results should be stored

**Returns** absolute path to SRPM, list with absolute paths to logs

**classmethod get\_logs** ()

Get logs from previously failed build Returns: dict with 'logs' -> list of absolute paths to logs

**static get\_srpm\_builder\_options** (\*\*kwargs)

**exception rebasehelper.build\_helper.SourcePackageBuildError** (\*args, \*\*kwargs)  
Error indicating failure to build Source Package.

## 4.2.5 CLI module

**class rebasehelper.cli.CLI** (*args=None*)  
Class for processing data from commandline

**static build\_parser** (*available\_choices\_only=False*)

**class rebasehelper.cli.CliHelper**

**static run** ()

## 4.2.6 Exception module

**exception rebasehelper.exceptions.CheckerNotFoundError**  
Error indicating failure unable to find checker binary

**exception rebasehelper.exceptions.DownloadError**  
Exception indicating that download of a file failed

**exception rebasehelper.exceptions.LookasideCacheError**  
Exception indicating a problem accessing lookaside cache

**exception rebasehelper.exceptions.ParseError**

**exception rebasehelper.exceptions.RebaseHelperError** (\*args, \*\*kwargs)  
Class representing Error raised inside rebase-helper after intentionally catching some expected and well known exception/error.

## 4.2.7 Logger module

**class rebasehelper.logger.ColorizingStreamHandler** (*stream=None*)

**colors** = {'dark': {50: {'style': 'bold', 'bg': 'red', 'fg': 'white'}, 27: {'styl

**emit** (*record*)

Emit a record.

If a formatter is specified, it is used to format the record. The record is then written to the stream with a trailing newline. If exception information is present, it is formatted using `traceback.print_exception` and appended to the stream. If the stream has an `'encoding'` attribute, it is used to determine how to do the output to the stream.

**set\_terminal\_background** (*background*)

`terminal_background = 'dark'`

**class** `rebasehelper.logger.CustomLogger` (*name, level=0*)

`HEADING = 26`

`IMPORTANT = 27`

`SUCCESS = 25`

`TRACE = 11`

`VERBOSE = 12`

**class** `rebasehelper.logger.LoggerHelper`

Helper class for setting up a logger

**static** `add_file_handler` (*logger\_object, path, formatter\_object=None, level=None*)

Adds file handler to the given logger.

#### Parameters

- **logger\_object** (*logging.Logger*) – Logger object to add the handler to.
- **path** (*str*) – Path to a log file.
- **formatter\_object** (*logging.Formatter*) – Formatter object used to format logged messages.
- **level** (*int*) – Severity threshold.

**Returns** Created file handler instance.

**Return type** `logging.FileHandler`

**static** `add_stream_handler` (*logger\_object, level=None, formatter\_object=None*)

Adds stream handler to the given logger.

#### Parameters

- **logger\_object** (*logging.Logger*) – Logger object to add the handler to.
- **level** (*int*) – Severity threshold.
- **formatter\_object** (*logging.Formatter*) – Formatter object used to format logged messages.

**Returns** Created stream handler instance.

**Return type** `logging.StreamHandler`

**static** `get_basic_logger` (*logger\_name, level=10*)

Sets up a basic logger without any handler.

#### Parameters

- **logger\_name** (*str*) – Logger name.

- **level** (*int*) – Severity threshold.

**Returns** Created logger instance.

**Return type** logging.Logger

## 4.2.8 Ouptut tool module

**class** rebasehelper.output\_tool.**BaseOutputTool**

Base class for OutputTools. `print_cli_summary` must be overridden in order to produce different CLI output

**DEFAULT** = **False**

**EXTENSION** = ''

**classmethod** `get_report_path` (*app*)

**classmethod** `prepend_results_dir_name` (*\*path\_members*)

**classmethod** `print_cli_summary` (*app*)

Print report of the rebase

**Parameters** **app** – Application instance

**classmethod** `print_important_checkers_output` ()

Iterates over all checkers output to highlight important checkers warning

**classmethod** `print_patches_cli` ()

Print info about patches

**classmethod** `print_patches_section_cli` (*logger\_method, patch\_type*)

Print info about one of the patches key section

**Parameters**

- **logger\_method** – method to be used for logging
- **patch\_type** – string containing key for the patch\_dict

**classmethod** `print_report_file_path` ()

Print path to the report file

**classmethod** `run` (*logs, app*)

**class** rebasehelper.output\_tool.**OutputToolRunner**

Class representing the process of running various output tools.

**get\_all\_tools** ()

**get\_default\_tool** ()

**get\_supported\_tools** ()

**run\_output\_tool** (*tool, logs=None, app=None*)

Runs specified output tool.

**Parameters**

- **tool** – Tool to run
- **log** – Log that probably contains the important message concerning the rebase fail
- **app** – Application class instance

## 4.2.9 Patch helper module

```
class rebasehelper.patch_helper.GitPatchTool
    Class for git command used for patching old and new sources

    CMD = 'git'

    classmethod apply_old_patches ()
        Function applies a patch to a old/new sources

    static apply_patch (repo, patch_object)
        Function applies patches to old sources It tries apply patch with am command and if it fails then with
        command -apply

    diff_cls = None

    classmethod init_git (directory)
        Function initialize old and new Git repository

    classmethod match (cmd=None)
        Method checks whether it is usefull patch method

    new_repo = None

    new_sources = ''

    non_interactive = False

    old_repo = None

    old_sources = ''

    output_data = None

    patches = []

    classmethod run_patch (old_dir, new_dir, rest_sources, patches, **kwargs)
        The function can be used for patching one directory against another

    source_dir = ''

class rebasehelper.patch_helper.PatchBase
    Class used for using several patching command tools, ... Each method should overwrite method like run_check

    classmethod match (cmd=None)
        Method checks whether it is usefull patch method

    classmethod run_patch (old_dir, new_dir, rest_sources, patches, **kwargs)
        Method will check all patches in relevant package

class rebasehelper.patch_helper.Patcher (tool=None)
    Class representing a process of applying and generating rebased patch using specific tool.

    patch (old_dir, new_dir, rest_sources, patches, **kwargs)
        Apply patches and generate rebased patches if needed

        Parameters

        • old_dir – path to dir with old patches
        • new_dir – path to dir with new patches
        • patches – old patches
        • rebased_patches – rebased patches
```

- **kwargs** --

#### Returns

`rebasehelper.patch_helper.register_patch_tool(patch_tool)`

## 4.2.10 Specfile module

**class** `rebasehelper.specfile.BaseSpecHook`

Base class for a spec hook

**CATEGORIES** = None

**classmethod** `run(spec_file, rebase_spec_file, **kwargs)`

Runs a spec hook.

#### Parameters

- **spec\_file** – Original spec file object
- **rebase\_spec\_file** – Rebased spec file object
- **kwargs** – Keyword arguments from Application instance

**class** `rebasehelper.specfile.PatchList`

**class** `rebasehelper.specfile.PatchObject(path, index, strip)`

Class represents set of information about patches

`get_index()`

`get_patch_name()`

`get_path()`

`get_strip()`

`git_generated = ''`

`index = ''`

`path = ''`

`set_path(new_path)`

`strip = ''`

**class** `rebasehelper.specfile.SpecContent(content)`

Class representing content of a SPEC file.

**SECTION\_HEADERS** = ['%package', '%prep', '%build', '%install', '%check', '%clean', '%pr

**class** `rebasehelper.specfile.SpecFile(path, changelog_entry, sources_location="", download=True)`

Class representing a SPEC file

**category** = None

**copy** (`new_path=None`)

Create a copy of the current object and copy the SPEC file the new object represents to a new location.

**Parameters** `new_path` – new path to which to copy the SPEC file

**Returns** copy of the current object

**download** = False

**download\_remote\_sources** ()

Method that iterates over all sources and downloads ones, which contain URL instead of just a file.

**Returns** None

**extra\_version = None**

**static extract\_version\_from\_archive\_name** (*archive\_path*, *source\_string=""*)

Method extracts the version from archive name based on the source string from SPEC file. It extracts also an extra version such as 'b1', 'rc1', ...

**Parameters**

- **archive\_path** – archive name or path with archive name from which to extract the version
- **source\_string** – Source string from SPEC file used to construct version extraction regex

**Returns** tuple of strings with (extracted version, extra version) or (None, None) if extraction failed

**find\_archive\_target\_in\_prep** (*archive*)

Tries to find a command that is used to extract the specified archive and attempts to determine target path from it. 'tar' and 'unzip' commands are supported so far.

**Parameters** **archive** – Path to archive

**Returns** Target path relative to builddir or None if not determined

**get\_NVR** ()

**get\_applied\_patches** ()

Method returns list of all applied patches.

**Returns** list of PatchObject

**get\_archive** ()

Method returns the basename of first Source in SPEC file a.k.a. Source0

**Returns** basename of first Source in SPEC file

**Return type** str

**get\_epoch\_number** ()

Method for getting epoch of the package

**Returns**

**get\_extra\_version** ()

Returns an extra version of the package - like b1, rc2, ...

**Returns** String

**get\_extra\_version\_separator** ()

Returns the separator between version and extra version as used by upstream. If there is not separator or extra version, it returns an empty string.

**Returns** String with the separator between version as extra version as used by upstream.

**Return type** str

**get\_full\_version** ()

Returns the full version string, which is a combination of version, separator and extra version.

**Returns** String with full version, including the extra version part.

**Return type** str

**get\_main\_files\_section()**

Finds the exact name of the main %files section.

**Returns** Name of the main files section.

**Return type** str

**get\_new\_log()**

**get\_not\_used\_patches()**

Method returns list of all unapplied patches.

**Returns** list of PatchObject

**get\_package\_name()**

Function returns a package name

**Returns**

**get\_patches()**

Method returns list of all applied and not applied patches

**Returns** list of PatchObject

**get\_path()**

Return only spec file path

**Returns**

**get\_prep\_section()**

Function returns whole prep section

**get\_release()**

Method for getting full release string of the package

**Returns**

**get\_release\_number()**

Method for getting the release of the package

**Returns**

**get\_requires()**

Function returns a package requirements

**Returns**

**get\_setup\_dirname()**

Get dirname from %setup or %autoseup macro arguments

**Returns** dirname

**get\_sources()**

Method returns dictionary with local sources list.

**Returns** list of Sources with absolute path

**Return type** list of str

**get\_version()**

Method returns the version

**Returns**

**hdr = None**

**is\_test\_suite\_enabled()**

Returns whether test suite is enabled during the build time

**Returns** True if enabled or False if not

**patches = None**

**path = ''**

**prep\_section = []**

**redefine\_release\_with\_macro(*macro*)**

Method redefines the Release: line to include passed macro and comments out the old line

**Parameters** *macro* –

**Returns**

**removed\_patches = []**

**revert\_redefine\_release\_with\_macro(*macro*)**

Method removes the redefined the Release: line with given macro and uncomments the old Release line.

**Parameters** *macro* –

**Returns**

**save()**

Save changes made to the spec\_content to the disc and update internal variables

**set\_extra\_version(*extra\_version*)**

Method to update the extra version in the SPEC file. Redefined Source0 if needed and also changes Release accordingly.

**Parameters** *extra\_version* – the extra version string, if any (e.g. 'b1', 'rc2', ...)

**Returns** None

**set\_extra\_version\_separator(*separator*)**

Set the string that separates the version and extra version

**Parameters** *separator* –

**Returns**

**set\_release\_number(*release*)**

Method to set release number

**Parameters** *release* –

**Returns**

**set\_tag(*tag, value, preserve\_macros=False*)**

Sets value of a tag while trying to preserve macros if requested

**set\_version(*version*)**

Method to update the version in the SPEC file

**Parameters** *version* – string with new version

**Returns** None

**set\_version\_using\_archive(*archive\_path*)**

Method to update the version in the SPEC file using a archive path. The version is extracted from the archive name.

**Parameters** *archive\_path* –



**Returns****sources = None****spc = None****spec\_content = None****static split\_version\_string** (*version\_string=""*)

Method splits version string into version and possibly extra string as 'rc1' or 'b1', ...

**Parameters version\_string** – version string such as '1.1.1' or '1.2.3b1', ...**Returns** tuple of strings with (extracted version, extra version, separator) or (None, None, None) if extraction failed**update\_paths\_to\_patches** ()**update\_setup\_dirname** (*dirname*)

Update %setup or %autosetup dirname argument if needed

**Parameters dirname** – new dirname to be used**write\_updated\_patches** (*patches, disable\_inapplicable*)

Function writes the patches to -rebase.spec file

**class** rebasehelper.specfile.**SpecHooksRunner**

Class representing the process of running various spec file hooks.

**get\_all\_spec\_hooks** ()**get\_available\_spec\_hooks** ()**run\_spec\_hooks** (*spec\_file, rebase\_spec\_file, \*\*kwargs*)

Runs all non-blacklisted spec hooks.

**Parameters**

- **spec\_file** – Original spec file object
- **rebase\_spec\_file** – Rebased spec file object
- **kwargs** – Keyword arguments from Application instance

**rebasehelper.specfile.get\_rebase\_name** (*dir\_name, name*)

Function returns a name in results directory

**Parameters**

- **dir\_name** –
- **name** –

**Returns** full path to results dir with name

## 4.2.11 Temporary environment module

**class** rebasehelper.temporary\_environment.**TemporaryEnvironment** (*exit\_callback=None*)

Class representing a temporary environment (directory) that can be used as a workspace.

Works as a context manager.

**TEMPDIR = 'TEMPDIR'****env** ()

Gets a copy of \_env dictionary.

**Returns** Copy of `_env` dictionary.

**Return type** dict

`path()`

Gets the path to the temporary environment.

**Returns** Absolute path to the temporary environment.

**Return type** str

## 4.3 Change Log

### 4.3.1 [Unreleased]

### 4.3.2 [0.14.0] - 2018-10-04

#### Added

- Added **PathsToRPMMacros** SPEC hook for transforming paths in `%files` section
- Added `--favor-on-conflict` option to prefer upstream or downstream changes with conflicting patches

#### Changed

- Extended **PyPIURLFix** SPEC hook to incorporate the new <https://pypi.org> website
- Made processing of patches in a SPEC file more robust
- Rewritten functional test to use an artificial package designed to check most aspects of the rebase process
- `pylint` is now run with Python 3 only, as Python 2 variant is no longer supported
- Code refactoring, simplified `SpecFile` class
- Checkers are no longer required for **rebase-helper** to run, only available checkers are used

#### Fixed

- Fixed bug in **licensecheck** checker when used with **json** output tool
- Fixed SPEC hook tests
- Fixed strangely acting lookaside cache upload progressbar
- Fixed downloading of SRPMs with `--get-old-build-from-koji`
- Fixed building usage documentation

### 4.3.3 [0.13.2] - 2018-05-18

#### Added

- Added **licensecheck** checker for detecting license changes
- Added another *not-so-verbose* verbosity level

### Changed

- Refactored `utils` module

### Fixed

- Fixed **abipkgdiff** detecting changes in only one object file
- Fixed uploads to lookaside cache
- Fixed broken consequent build retries

## 4.3.4 [0.13.1] - 2018-04-19

### Added

- Added `--apply-changes` option to apply *changes.patch* after successful rebase
- Implemented `.gitignore` update with `--update-sources`

### Changed

- Extended `README.md`
- Cleaned up constants

### Fixed

- Fixed crash after failed rebase when no checkers were run

## 4.3.5 [0.13.0] - 2018-03-29

### Added

- Added possibility to make changes to specfile between build retries
- Added **CommitHashUpdater** SPEC hook
- Added **hackage** versioneer
- Added support for uncompressed tar archives
- Created integration environment for test suite to isolate it from the internet
- Added `--update-sources` option to update *sources* file and upload new sources to lookaside cache

### Changed

- Switched to `requests` library for downloads
- Made error messages from **Koji** builds more useful
- Reworked handling of downstream patches
- Changed package build process to build first SRPMs and then RPMs

- Divided checkers into categories running at different phases of rebase
- **Koji** build tool refactored to be better adjustable and extensible
- Colorized **rebase-helper** output and enhanced log messages
- Significantly improved rebase summary and report

### Fixed

- Fixed `TestConsoleHelper.test_get_message()` test
- Fixed bug in **rpmdiff** output analysis
- Fixed some code styling errors and a large number of issues found by static analysis

## 4.3.6 [0.12.0] - 2017-12-19

### Added

- Added **npmjs** and **cpan** versioneers
- Added possibility to specify custom `py.test` arguments
- Added possibility to customize changelog entry
- Added version check to abort rebase if requested version is not newer than current
- Added separate `tox` tasks for linting
- Implemented **rpmbuild** and **mock** SRPM build tools
- Added possibility to configure rebase-helper with configuration file
- Added possibility to blacklist certain SPEC hooks or versioneers
- Created `rebasehelper/rebase-helper` Docker Hub repository

### Changed

- Made several speed optimizations in the test suite
- Tests requiring superuser privileges are now automatically skipped if necessary
- Simplified build analysis and made related log messages more useful

### Fixed

- Fixed documentation builds on `readthedocs.org` broken by `rpm distribution` requirement
- Fixed reading username and e-mail from `git` configuration
- Added missing dependencies to `Dockerfile`
- Fixed processing of custom builder options
- Added workarounds for RPM bugs related to `%sources` and `%patches`
- Fixed several unhandled exceptions
- Fixed parsing tarball filename containing certain characters

### 4.3.7 [0.11.0] - 2017-10-04

#### Added

- Added `rpm-py-installer` to install `rpm-python` from `pip`
- Implemented detection of package category (*python, perl, ruby, nodejs, php*)
- Added **RubyGems** versioneer
- Added **RubyHelper** SPEC hook for getting additional sources based on instructions in SPEC file comments

#### Changed

- Value of *Version* and *Release* tags is now preserved if there are any macros that can be modified instead
- Versioneers and SPEC hooks are now run only for matching package categories
- Bash completion is now generated from source code, so it is always up-to-date

#### Fixed

- Prevented unwanted modifications of *%prep* section
- Fixed unexpected removal of rpms and build logs after last build retry
- Added files are no longer listed as removed in **rpmdiff** report

### 4.3.8 [0.10.1] - 2017-08-30

#### Added

- Added `--version` argument

#### Changed

- **Anitya** versioneer now primarily searches for projects using Fedora mapping
- Python dependencies moved from `requirements.txt` to `setup.py`

#### Fixed

- Made `CustomManPagesBuilder` work with `Sphinx >= 1.6`
- *%prep* section parser is now able to handle backslash-split lines

### 4.3.9 [0.10.0] - 2017-08-25

#### Added

- Implemented extensible SPEC hooks and versioneers
- Added **PyPI** SPEC hook for automatic fixing of Source URL of Python packages

- Added **Anitya** and **PyPI** versioners for determining latest upstream version of a package
- Added possibility to download old version build of a package from Koji
- Added support for test suite to be run in Docker containers
- Implemented functional tests for automatic testing of whole rebase process
- Diff against original source files is now generated as *changes.patch*

### Changed

- Introduced plugin system for extending build tools, checkers and output tools
- Updated for **Koji 1.13** which finally brings Python 3 support
- Improved output information and reports
- Added colorized output
- Improved project documentation

### Fixed

- Pre-configured git username and e-mail address is now used if available
- Fixed several issues in **rpmdiff** and especially **abipkgdiff** checkers
- Fixed several test suite related issues

## 4.3.10 [0.9.0] - 2017-01-05

### Added

- Old sources are now downloaded from Fedora lookaside cache
- Auto-generated and improved CLI documentation and man page
- Added support for downloading files of unknown size

### Changed

- `SpecFile` class preparation for pre-download hooks
- Code cleanup and refactorization

### Fixed

- Fixed regexp for getting release number from SPEC
- Fixed functionality of `--results-dir` option
- Several upstream monitoring fixes
- Fixed issues caused by Fedora Flag Day

### 4.3.11 [0.8.0] - 2016-07-31

#### Added

- Added support for JSON output format
- Added support for **copr** build tool
- Added support for passing arbitrary extra arguments to local builders (**mock**, **rpmbuild**) with `--builder-options`.
- Added new option `--build-retries` allows the user to specify number of build retries (by default 2)
- Added support for **csmock** check tool

#### Changed

- Renamed **fedpkg** build tool to **koji** to make it more clear
- Downloading of files is now done only using standard Python library and not using PyCURL

#### Fixed

- Many bug fixes and code clean up

### 4.3.12 [0.7.3] - 2016-04-08

#### Added

- Added `rpm.addMacro`

#### Fixed

- Handled exceptions raised during parsing of SPEC files
- Fixed unapplied patches mixing with deleted ones

### 4.3.13 [0.7.2] - 2016-03-15

#### Added

- Added information about scratch builds

#### Fixed

- Added check if file exists and is empty for **the-new-hotness**
- Patches are applied in case `--builds-nowait` option is used

#### 4.3.14 [0.7.1] - 2016-02-22

##### Added

- Two new command line options used by upstream monitoring

##### Fixed

- **fedpkg** reimplementation

#### 4.3.15 [0.7.0] - 2016-01-13

##### Changed

- Several improvements

##### Fixed

- **pkgdiff** is now smarter
- Included `tar.bz2` into list of supported formats
- Added support for noarch package in case of **fedpkg** build
- Checker should return `None` if there is no debug package

##### Removed

- Removed a bunch of debug stuff

#### 4.3.16 [0.6.2] - 2015-11-09

##### Fixed

- Logs are being saved to their own directory
- Prep script is moved into workspace directory
- No more traceback in case `koji` module is not present
- Each checker creates its own log file
- **rebase-helper** informs if it failed or not
- Report on script is smarter

#### 4.3.17 [0.6.1] - 2015-10-30

##### Added

- `upstream-monitoring.py` - used by upstream monitoring service
- `rebase-helper-fedmsg.py` - testing Python script



### 4.3.18 [0.6.0] - 2015-07-31

#### Added

- Parts of `%prep` section related to patching are executed
- Support for **abipkgdiff**

#### Fixed

- Several fixes
- Replaced `yum` with `dnf`

### 4.3.19 [0.5.0] - 2015-05-22

#### Added

- Added support for building packages via **fedpkg** (or **koji**)
- Added summary report for better overview
- `continue` option implemented for `git rebase`
- Added several tests
- Added class for operating with Git repositories

#### Changed

- `git rebase` is used instead of `patch` command

#### Fixed

- Fixed several decoding issues
- Several **PEP8** and **W1202** fixes

#### Removed

- `DiffHelper` class is not needed

### 4.3.20 [0.4.0] - 2014-12-05

#### Added

- Handling of extra versions like `b1`, `rc1`, etc.
- Added build log analyzer to detect unpackaged files or other issues
- Added Bash completion

## Changed

- Improved version extraction from archive name
- **rebase-helper** output is logged to `rebase-helper-results` directory
- `SpecFile` class rewritten

### 4.3.21 [0.3.1] - 2014-07-25

#### Added

- New build class
- `--build-only` option
- Installation of build dependencies in case of **rpmbuild** tool
- More tests
- `RebaseHelperError` class for catching exceptions

#### Fixed

- Several fixes

### 4.3.22 [0.3.0]

#### Added

- **pkgdiff** tool for comparing RPM packages
- Tests for `Archive` class and SPEC file

### 4.3.23 [0.2.0]

#### Added

- `diff_helper` for comparing two tarballs
- Applying patches to tarballs
- `patch_helper`

### 4.3.24 [0.1.0]

#### Added

- Initial classes
- CLI interface

**r**

rebasehelper.application, 23  
rebasehelper.archive, 25  
rebasehelper.build\_helper, 27  
rebasehelper.cli, 29  
rebasehelper.exceptions, 29  
rebasehelper.helpers.console\_helper, 14  
rebasehelper.helpers.copr\_helper, 15  
rebasehelper.helpers.download\_helper,  
16  
rebasehelper.helpers.git\_helper, 16  
rebasehelper.helpers.input\_helper, 16  
rebasehelper.helpers.koji\_helper, 17  
rebasehelper.helpers.lookaside\_cache\_helper,  
18  
rebasehelper.helpers.macro\_helper, 19  
rebasehelper.helpers.path\_helper, 19  
rebasehelper.helpers.process\_helper, 21  
rebasehelper.helpers.rpm\_helper, 22  
rebasehelper.logger, 29  
rebasehelper.output\_tool, 31  
rebasehelper.patch\_helper, 32  
rebasehelper.specfile, 33  
rebasehelper.temporary\_environment, 37



## Symbols

- `-apply-changes`  
rebase-helper command line option, 12
  - `-background {dark,light,auto}`  
rebase-helper command line option, 11
  - `-build-log-hook-blacklist {files}`  
rebase-helper command line option, 12
  - `-build-tasks OLD_TASK,NEW_TASK`  
rebase-helper command line option, 13
  - `-builder-options BUILDER_OPTIONS`  
rebase-helper command line option, 13
  - `-builds-nowait`  
rebase-helper command line option, 13
  - `-buildtool {copr,rpmbuild,koji,mock}`  
rebase-helper command line option, 12
  - `-changelog-entry CHANGELOG_ENTRY`  
rebase-helper command line option, 13
  - `-color {always,never,auto}`  
rebase-helper command line option, 11
  - `-comparepkgs-only COMPAREPKGS_DIR`  
rebase-helper command line option, 12
  - `-config-file CONFIG_FILE`  
rebase-helper command line option, 13
  - `-disable-inapplicable-patches`  
rebase-helper command line option, 12
  - `-favor-on-conflict {downstream,upstream,off}`  
rebase-helper command line option, 12
  - `-force-build-log-hooks`  
rebase-helper command line option, 12
  - `-get-old-build-from-koji`  
rebase-helper command line option, 12
  - `-non-interactive`  
rebase-helper command line option, 12
  - `-not-download-sources`  
rebase-helper command line option, 12
  - `-outputtool {text,json}`  
rebase-helper command line option, 12
  - `-pkgcomparetool {abip-kgdiff,rpmdiff,licensecheck,pkgdiff,csmock}`  
rebase-helper command line option, 12
  - `-results-dir RESULTS_DIR`  
rebase-helper command line option, 11
  - `-skip-version-check`  
rebase-helper command line option, 12
  - `-spec-hook-blacklist {pypi-url-fix,paths-to-rpm-macros,typo-fix,ruby-helper,commit-hash-updater}`  
rebase-helper command line option, 12
  - `-srpm-builder-options SRPM_BUILDER_OPTIONS`  
rebase-helper command line option, 13
  - `-srpm-buildtool {rpmbuild,mock}`  
rebase-helper command line option, 12
  - `-update-sources`  
rebase-helper command line option, 12
  - `-version`  
rebase-helper command line option, 11
  - `-versioneer {rubygems,pypi,anitya,hackage,cpan,npmjs}`  
rebase-helper command line option, 12
  - `-versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmjs}`  
rebase-helper command line option, 12
  - `-b, -build-only`  
rebase-helper command line option, 12
  - `-c, -continue`  
rebase-helper command line option, 12
  - `-h, -help`  
rebase-helper command line option, 11
  - `-p, -patch-only`  
rebase-helper command line option, 11
  - `-v, -verbose`  
rebase-helper command line option, 11
  - `-w, -keep-workspace`  
rebase-helper command line option, 12
- A**
- `ACCEPTS_OPTIONS` (rebase-helper.build\_helper.BuildToolBase attribute), 27
  - `add_file_handler()` (rebasehelper.logger.LoggerHelper static method), 30

- add\_stream\_handler() (rebasehelper.logger.LoggerHelper static method), 30
  - all\_packages\_installed() (rebasehelper.helpers.rpm\_helper.RpmHelper static method), 22
  - Application (class in rebasehelper.application), 23
  - apply\_changes() (rebasehelper.application.Application method), 23
  - apply\_old\_patches() (rebasehelper.patch\_helper.GitPatchTool class method), 32
  - apply\_patch() (rebasehelper.patch\_helper.GitPatchTool static method), 32
  - ARCHES (rebasehelper.helpers.rpm\_helper.RpmHelper attribute), 22
  - Archive (class in rebasehelper.archive), 25
  - ArchiveTypeBase (class in rebasehelper.archive), 25
- ## B
- BaseOutputTool (class in rebasehelper.output\_tool), 31
  - BaseSpecHook (class in rebasehelper.specfile), 33
  - BinaryPackageBuildError, 27
  - build() (rebasehelper.build\_helper.BuildToolBase class method), 27
  - build() (rebasehelper.build\_helper.SRPMBuildToolBase class method), 29
  - build() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 15
  - build\_binary\_packages() (rebasehelper.application.Application method), 23
  - build\_parser() (rebasehelper.cli.CLI static method), 29
  - build\_source\_packages() (rebasehelper.application.Application method), 23
  - BuildHelper (class in rebasehelper.build\_helper), 27
  - BuildTemporaryEnvironment (class in rebasehelper.build\_helper), 27
  - BuildToolBase (class in rebasehelper.build\_helper), 27
  - Bz2ArchiveType (class in rebasehelper.archive), 25
- ## C
- CATEGORIES (rebasehelper.specfile.BaseSpecHook attribute), 33
  - category (rebasehelper.specfile.SpecFile attribute), 33
  - CheckerNotFoundError, 29
  - CLI (class in rebasehelper.cli), 29
  - CliHelper (class in rebasehelper.cli), 29
  - CMD (rebasehelper.patch\_helper.GitPatchTool attribute), 32
  - color\_is\_light() (rebasehelper.helpers.console\_helper.ConsoleHelper static method), 14
  - ColorizingStreamHandler (class in rebasehelper.logger), 29
  - colors (rebasehelper.logger.ColorizingStreamHandler attribute), 29
  - ConsoleHelper (class in rebasehelper.helpers.console\_helper), 14
  - ConsoleHelper.Capturer (class in rebasehelper.helpers.console\_helper), 14
  - CoprHelper (class in rebasehelper.helpers.copr\_helper), 15
  - copy() (rebasehelper.specfile.SpecFile method), 33
  - cprint() (rebasehelper.helpers.console\_helper.ConsoleHelper class method), 14
  - create\_project() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 15
  - create\_session() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 17
  - CREATES\_TASKS (rebasehelper.build\_helper.BuildToolBase attribute), 27
  - CustomLogger (class in rebasehelper.logger), 30
- ## D
- debug\_log\_file (rebasehelper.application.Application attribute), 23
  - DEFAULT (rebasehelper.build\_helper.BuildToolBase attribute), 27
  - DEFAULT (rebasehelper.build\_helper.SRPMBuildToolBase attribute), 28
  - DEFAULT (rebasehelper.output\_tool.BaseOutputTool attribute), 31
  - detect\_background() (rebasehelper.helpers.console\_helper.ConsoleHelper static method), 15
  - DEV\_NULL (rebasehelper.helpers.process\_helper.ProcessHelper attribute), 21
  - diff\_cls (rebasehelper.patch\_helper.GitPatchTool attribute), 32
  - display\_task\_results() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 17
  - download (rebasehelper.specfile.SpecFile attribute), 33
  - download() (rebasehelper.helpers.lookaside\_cache\_helper.LookasideCache class method), 18
  - download\_build() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 15
  - download\_build() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 17
  - download\_file() (rebasehelper.helpers.download\_helper.DownloadHelper

- static method), 16  
 download\_remote\_sources() (rebase-helper.specfile.SpecFile method), 33  
 download\_task\_results() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 17  
 DownloadError, 29  
 DownloadHelper (class in rebase-helper.helpers.download\_helper), 16  
 dump() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 19
- ## E
- emit() (rebasehelper.logger.ColorizingStreamHandler method), 29  
 env() (rebasehelper.temporary\_environment.TemporaryEnvironment helper.specfile.SpecFile static method), 37  
 environment variable  
     PACKAGE, 10  
     REPOSITORY, 10  
 exchange\_control\_sequence() (rebase-helper.helpers.console\_helper.ConsoleHelper static method), 15  
 expand() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 19  
 expand\_macros() (rebase-helper.helpers.macro\_helper.MacroHelper static method), 19  
 EXTENSION (rebasehelper.archive.ArchiveTypeBase attribute), 25  
 EXTENSION (rebasehelper.archive.Bz2ArchiveType attribute), 25  
 EXTENSION (rebasehelper.archive.GemPseudoArchiveType attribute), 25  
 EXTENSION (rebasehelper.archive.TarArchiveType attribute), 26  
 EXTENSION (rebasehelper.archive.TarBz2ArchiveType attribute), 26  
 EXTENSION (rebasehelper.archive.TarGzArchiveType attribute), 26  
 EXTENSION (rebasehelper.archive.TarXzArchiveType attribute), 26  
 EXTENSION (rebasehelper.archive.TgzArchiveType attribute), 26  
 EXTENSION (rebasehelper.archive.ZipArchiveType attribute), 26  
 EXTENSION (rebasehelper.output\_tool.BaseOutputTool attribute), 31  
 extra\_version (rebasehelper.specfile.SpecFile attribute), 34  
 extract() (rebasehelper.archive.ArchiveTypeBase class method), 25  
 extract() (rebasehelper.archive.Bz2ArchiveType class method), 25  
 extract() (rebasehelper.archive.GemPseudoArchiveType class method), 25  
 extract() (rebasehelper.archive.TarGzArchiveType class method), 26  
 extract() (rebasehelper.archive.TarXzArchiveType class method), 26  
 extract() (rebasehelper.archive.ZipArchiveType class method), 26  
 extract\_archive() (rebasehelper.application.Application static method), 23  
 extract\_archive() (rebasehelper.archive.Archive method), 25  
 extract\_sources() (rebasehelper.application.Application static method), 23  
 extract\_version\_from\_archive\_name() (rebase-helper.specfile.SpecFile static method), 34
- ## F
- file\_available() (rebase-helper.helpers.path\_helper.PathHelper static method), 19  
 filter() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 19  
 find\_all\_files() (rebase-helper.helpers.path\_helper.PathHelper static method), 20  
 find\_all\_files\_current\_dir() (rebase-helper.helpers.path\_helper.PathHelper static method), 20  
 find\_archive\_target\_in\_prep() (rebase-helper.specfile.SpecFile method), 34  
 find\_first\_dir\_with\_file() (rebase-helper.helpers.path\_helper.PathHelper static method), 20  
 find\_first\_file() (rebase-helper.helpers.path\_helper.PathHelper static method), 20  
 functional (rebasehelper.helpers.koji\_helper.KojiHelper attribute), 18
- ## G
- GemPseudoArchiveType (class in rebasehelper.archive), 25  
 generate\_patch() (rebasehelper.application.Application method), 23  
 get\_all\_log\_files() (rebasehelper.application.Application method), 23  
 get\_all\_spec\_hooks() (rebase-helper.specfile.SpecHooksRunner method), 37  
 get\_all\_tools() (rebasehelper.build\_helper.BuildHelper method), 27

get\_all\_tools() (rebase-helper.build\_helper.SRPMBuildHelper method), 28  
 get\_all\_tools() (rebase-helper.output\_tool.OutputToolRunner method), 31  
 get\_applied\_patches() (rebasehelper.specfile.SpecFile method), 34  
 get\_arches() (rebasehelper.helpers.rpm\_helper.RpmHelper static method), 22  
 get\_archive() (rebasehelper.specfile.SpecFile method), 34  
 get\_available\_spec\_hooks() (rebase-helper.specfile.SpecHooksRunner method), 37  
 get\_basic\_logger() (rebasehelper.logger.LoggerHelper static method), 30  
 get\_build\_status() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 15  
 get\_build\_url() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 15  
 get\_builder\_options() (rebase-helper.build\_helper.BuildToolBase static method), 27  
 get\_checker\_outputs() (rebase-helper.application.Application method), 23  
 get\_client() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 15  
 get\_default\_tool() (rebase-helper.build\_helper.BuildHelper method), 27  
 get\_default\_tool() (rebase-helper.build\_helper.SRPMBuildHelper method), 28  
 get\_default\_tool() (rebase-helper.output\_tool.OutputToolRunner method), 31  
 get\_detached\_task() (rebase-helper.build\_helper.BuildToolBase class method), 27  
 get\_email() (rebasehelper.helpers.git\_helper.GitHelper class method), 16  
 get\_epoch\_number() (rebasehelper.specfile.SpecFile method), 34  
 get\_extra\_version() (rebasehelper.specfile.SpecFile method), 34  
 get\_extra\_version\_separator() (rebase-helper.specfile.SpecFile method), 34  
 get\_full\_version() (rebasehelper.specfile.SpecFile method), 34  
 get\_header\_from\_rpm() (rebase-helper.helpers.rpm\_helper.RpmHelper static method), 22  
 get\_index() (rebasehelper.specfile.PatchObject method), 33  
 get\_info\_from\_rpm() (rebase-helper.helpers.rpm\_helper.RpmHelper static method), 22  
 get\_latest\_build() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 18  
 get\_logs() (rebasehelper.build\_helper.BuildToolBase class method), 28  
 get\_logs() (rebasehelper.build\_helper.SRPMBuildToolBase class method), 29  
 get\_main\_files\_section() (rebasehelper.specfile.SpecFile method), 35  
 get\_message() (rebasehelper.helpers.input\_helper.InputHelper static method), 16  
 get\_new\_build\_logs() (rebase-helper.application.Application method), 23  
 get\_new\_log() (rebasehelper.specfile.SpecFile method), 35  
 get\_not\_used\_patches() (rebasehelper.specfile.SpecFile method), 35  
 get\_NVR() (rebasehelper.specfile.SpecFile method), 34  
 get\_old\_build\_info() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 18  
 get\_package\_name() (rebasehelper.specfile.SpecFile method), 35  
 get\_patch\_name() (rebasehelper.specfile.PatchObject method), 33  
 get\_patches() (rebasehelper.specfile.SpecFile method), 35  
 get\_path() (rebasehelper.specfile.PatchObject method), 33  
 get\_path() (rebasehelper.specfile.SpecFile method), 35  
 get\_prep\_section() (rebasehelper.specfile.SpecFile method), 35  
 get\_rebase\_name() (in module rebasehelper.specfile), 37  
 get\_rebased\_patches() (rebase-helper.application.Application method), 23  
 get\_rebasehelper\_data() (rebase-helper.application.Application method), 24  
 get\_release() (rebasehelper.specfile.SpecFile method), 35  
 get\_release\_number() (rebasehelper.specfile.SpecFile method), 35  
 get\_report\_path() (rebase-helper.output\_tool.BaseOutputTool class method), 31  
 get\_requires() (rebasehelper.specfile.SpecFile method), 35  
 get\_rpm\_packages() (rebase-



- helper.application.Application method), 24  
 get\_setup\_dirname() (rebasehelper.specfile.SpecFile method), 35  
 get\_sources() (rebasehelper.specfile.SpecFile method), 35  
 get\_srpm\_builder\_options() (rebasehelper.build\_helper.SRPMBuildToolBase static method), 29  
 get\_strip() (rebasehelper.specfile.PatchObject method), 33  
 get\_supported\_archives() (rebasehelper.archive.Archive class method), 25  
 get\_supported\_tools() (rebasehelper.build\_helper.BuildHelper method), 27  
 get\_supported\_tools() (rebasehelper.build\_helper.SRPMBuildHelper method), 28  
 get\_supported\_tools() (rebasehelper.output\_tool.OutputToolRunner method), 31  
 get\_task\_info() (rebasehelper.build\_helper.BuildToolBase class method), 28  
 get\_task\_url() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 18  
 get\_temp\_dir() (rebasehelper.helpers.path\_helper.PathHelper static method), 20  
 get\_tool() (rebasehelper.build\_helper.BuildHelper method), 27  
 get\_tool() (rebasehelper.build\_helper.SRPMBuildHelper method), 28  
 get\_user() (rebasehelper.helpers.git\_helper.GitHelper class method), 16  
 get\_version() (rebasehelper.specfile.SpecFile method), 35  
 git\_generated (rebasehelper.specfile.PatchObject attribute), 33  
 GIT\_USER\_EMAIL (rebasehelper.helpers.git\_helper.GitHelper attribute), 16  
 GIT\_USER\_NAME (rebasehelper.helpers.git\_helper.GitHelper attribute), 16  
 GitHelper (class in rebasehelper.helpers.git\_helper), 16  
 GitPatchTool (class in rebasehelper.patch\_helper), 32
- ## H
- hdr (rebasehelper.specfile.SpecFile attribute), 35  
 HEADING (rebasehelper.logger.CustomLogger attribute), 30
- ## I
- IMPORTANT (rebasehelper.logger.CustomLogger attribute), 30  
 index (rebasehelper.specfile.PatchObject attribute), 33  
 init\_git() (rebasehelper.patch\_helper.GitPatchTool class method), 32  
 InputHelper (class in rebasehelper.helpers.input\_helper), 16  
 install\_build\_dependencies() (rebasehelper.helpers.rpm\_helper.RpmHelper static method), 22  
 is\_package\_installed() (rebasehelper.helpers.rpm\_helper.RpmHelper static method), 23  
 is\_test\_suite\_enabled() (rebasehelper.specfile.SpecFile method), 35
- ## K
- KojiHelper (class in rebasehelper.helpers.koji\_helper), 17  
 kwargs (rebasehelper.application.Application attribute), 24
- ## L
- LoggerHelper (class in rebasehelper.logger), 30  
 LookasideCacheError, 29  
 LookasideCacheHelper (class in rebasehelper.helpers.lookaside\_cache\_helper), 18
- ## M
- MacroHelper (class in rebasehelper.helpers.macro\_helper), 19  
 MACROS\_WHITELIST (rebasehelper.helpers.macro\_helper.MacroHelper attribute), 19  
 match() (rebasehelper.archive.ArchiveTypeBase class method), 25  
 match() (rebasehelper.archive.ZipArchiveType class method), 26  
 match() (rebasehelper.patch\_helper.GitPatchTool class method), 32  
 match() (rebasehelper.patch\_helper.PatchBase class method), 32  
 MockTemporaryEnvironment (class in rebasehelper.build\_helper), 28
- ## N
- new\_repo (rebasehelper.patch\_helper.GitPatchTool attribute), 32  
 new\_rest\_sources (rebasehelper.application.Application attribute), 24  
 new\_sources (rebasehelper.application.Application attribute), 24  
 new\_sources (rebasehelper.patch\_helper.GitPatchTool attribute), 32

non\_interactive (rebasehelper.patch\_helper.GitPatchTool attribute), 32

## O

old\_repo (rebasehelper.patch\_helper.GitPatchTool attribute), 32

old\_rest\_sources (rebasehelper.application.Application attribute), 24

old\_sources (rebasehelper.application.Application attribute), 24

old\_sources (rebasehelper.patch\_helper.GitPatchTool attribute), 32

open() (rebasehelper.archive.ArchiveTypeBase class method), 25

open() (rebasehelper.archive.Bz2ArchiveType class method), 25

open() (rebasehelper.archive.GemPseudoArchiveType class method), 26

open() (rebasehelper.archive.TarGzArchiveType class method), 26

open() (rebasehelper.archive.TarXzArchiveType class method), 26

open() (rebasehelper.archive.ZipArchiveType class method), 26

output\_data (rebasehelper.patch\_helper.GitPatchTool attribute), 32

OutputToolRunner (class in rebasehelper.output\_tool), 31

## P

PACKAGE, 10

parse\_rgb\_device\_specification() (rebasehelper.helpers.console\_helper.ConsoleHelper static method), 15

parse\_spec() (rebasehelper.helpers.rpm\_helper.RpmHelper class method), 23

ParseError, 29

patch() (rebasehelper.patch\_helper.Patcher method), 32

patch\_sources() (rebasehelper.application.Application method), 24

PatchBase (class in rebasehelper.patch\_helper), 32

Patcher (class in rebasehelper.patch\_helper), 32

patches (rebasehelper.patch\_helper.GitPatchTool attribute), 32

patches (rebasehelper.specfile.SpecFile attribute), 36

PatchList (class in rebasehelper.specfile), 33

PatchObject (class in rebasehelper.specfile), 33

path (rebasehelper.specfile.PatchObject attribute), 33

path (rebasehelper.specfile.SpecFile attribute), 36

path() (rebasehelper.temporary\_environment.TemporaryEnvironment method), 38

PathHelper (class in rebasehelper.helpers.path\_helper), 19

prep\_section (rebasehelper.specfile.SpecFile attribute), 36

prepare() (rebasehelper.build\_helper.BuildToolBase class method), 28

prepare\_next\_run() (rebasehelper.application.Application method), 24

prepare\_sources() (rebasehelper.application.Application method), 24

prepend\_results\_dir\_name() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_cli\_summary() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_important\_checkers\_output() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_patches\_cli() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_patches\_section\_cli() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_report\_file\_path() (rebasehelper.output\_tool.BaseOutputTool class method), 31

print\_summary() (rebasehelper.application.Application method), 24

print\_task\_info() (rebasehelper.application.Application method), 24

ProcessHelper (class in rebasehelper.helpers.process\_helper), 21

progress() (rebasehelper.helpers.download\_helper.DownloadHelper static method), 16

## R

rebase-helper command line option

–apply-changes, 12

–background {dark,light,auto}, 11

–build-log-hook-blacklist {files}, 12

–build-tasks OLD\_TASK,NEW\_TASK, 13

–builder-options BUILDER\_OPTIONS, 13

–builds-nowait, 13

–buildtool {copr,rpmbuild,koji,mock}, 12

–changelog-entry CHANGELOG\_ENTRY, 13

–color {always,never,auto}, 11

–comparepkgs-only COMPAREPKGS\_DIR, 12

–config-file CONFIG\_FILE, 13

–disable-inapplicable-patches, 12

–favor-on-conflict {downstream,upstream,off}, 12

–force-build-log-hooks, 12

–get-old-build-from-koji, 12

–non-interactive, 12

–not-download-sources, 12

–outputtool {text,json}, 12

- pkgcomparetool {abipkgdiff,rpmdiff,licensecheck,pkgdiff,csmock}, 12
- results-dir RESULTS\_DIR, 11
- skip-version-check, 12
- spec-hook-blacklist {pypi-url-fix,paths-to-rpm-macros,typo-fix,ruby-helper,commit-hash-updater}, 12
- srpm-builder-options SRPM\_BUILDER\_OPTIONS, 13
- srpm-buildtool {rpmbuild,mock}, 12
- update-sources, 12
- version, 11
- versioneer {rubygems,pypi,anitya,hackage,cpan,npmjs}, 12
- versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmjs}, 12
- b, -build-only, 12
- c, -continue, 12
- h, -help, 11
- p, -patch-only, 11
- v, -verbose, 11
- w, -keep-workspace, 12
- SOURCES, 11
- rebase\_spec\_file (rebasehelper.application.Application attribute), 24
- rebase\_spec\_file\_path (rebasehelper.application.Application attribute), 24
- rebased\_patches (rebasehelper.application.Application attribute), 24
- rebased\_repo (rebasehelper.application.Application attribute), 24
- rebasehelper.application (module), 23
- rebasehelper.archive (module), 25
- rebasehelper.build\_helper (module), 27
- rebasehelper.cli (module), 29
- rebasehelper.exceptions (module), 29
- rebasehelper.helpers.console\_helper (module), 14
- rebasehelper.helpers.copr\_helper (module), 15
- rebasehelper.helpers.download\_helper (module), 16
- rebasehelper.helpers.git\_helper (module), 16
- rebasehelper.helpers.input\_helper (module), 16
- rebasehelper.helpers.koji\_helper (module), 17
- rebasehelper.helpers.lookaside\_cache\_helper (module), 18
- rebasehelper.helpers.macro\_helper (module), 19
- rebasehelper.helpers.path\_helper (module), 19
- rebasehelper.helpers.process\_helper (module), 21
- rebasehelper.helpers.rpm\_helper (module), 22
- rebasehelper.logger (module), 29
- rebasehelper.output\_tool (module), 31
- rebasehelper.patch\_helper (module), 32
- rebasehelper.specfile (module), 33
- rebasehelper.temporary\_environment (module), 37
- RebaseHelperError, 29
- redefine\_release\_with\_macro() (rebasehelper.specfile.SpecFile method), 36
- register\_archive\_type() (in module rebasehelper.archive), 27
- register\_patch\_tool() (in module rebasehelper.patch\_helper), 33
- removed\_patches (rebasehelper.specfile.SpecFile attribute), 36
- report\_log\_file (rebasehelper.application.Application attribute), 24
- REPOSITORY, 10
- request() (rebasehelper.helpers.download\_helper.DownloadHelper static method), 16
- result\_file (rebasehelper.application.Application attribute), 24
- revert\_redefine\_release\_with\_macro() (rebasehelper.specfile.SpecFile method), 36
- rpkg\_config\_dir (rebasehelper.helpers.lookaside\_cache\_helper.LookasideCacheHelper attribute), 18
- RpmbuildTemporaryEnvironment (class in rebasehelper.build\_helper), 28
- RpmHelper (class in rebasehelper.helpers.rpm\_helper), 22
- run() (rebasehelper.application.Application method), 24
- run() (rebasehelper.cli.CliHelper static method), 29
- run() (rebasehelper.output\_tool.BaseOutputTool class method), 31
- run() (rebasehelper.specfile.BaseSpecHook class method), 33
- run\_mergestool() (rebasehelper.helpers.git\_helper.GitHelper class method), 16
- run\_output\_tool() (rebasehelper.output\_tool.OutputToolRunner method), 31
- run\_package\_checkers() (rebasehelper.application.Application method), 24
- run\_patch() (rebasehelper.patch\_helper.GitPatchTool class method), 32
- run\_patch() (rebasehelper.patch\_helper.PatchBase class method), 32
- run\_spec\_hooks() (rebasehelper.specfile.SpecHooksRunner method), 37
- run\_subprocess() (rebasehelper.helpers.process\_helper.ProcessHelper static method), 21
- run\_subprocess\_cwd() (rebasehelper.helpers.process\_helper.ProcessHelper

- static method), 21
  - run\_subprocess\_cwd\_env() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 21
  - run\_subprocess\_env() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 21
- S**
- save() (rebasehelper.specfile.SpecFile method), 36
  - SECTION\_HEADERS (rebase-helper.specfile.SpecContent attribute), 33
  - set\_extra\_version() (rebasehelper.specfile.SpecFile method), 36
  - set\_extra\_version\_separator() (rebase-helper.specfile.SpecFile method), 36
  - set\_path() (rebasehelper.specfile.PatchObject method), 33
  - set\_release\_number() (rebasehelper.specfile.SpecFile method), 36
  - set\_tag() (rebasehelper.specfile.SpecFile method), 36
  - set\_terminal\_background() (rebase-helper.logger.ColorizingStreamHandler method), 30
  - set\_version() (rebasehelper.specfile.SpecFile method), 36
  - set\_version\_using\_archive() (rebase-helper.specfile.SpecFile method), 36
  - setup() (rebasehelper.application.Application static method), 24
  - setup\_logging() (rebasehelper.application.Application static method), 24
  - should\_use\_colors() (rebase-helper.helpers.console\_helper.ConsoleHelper class method), 15
  - source\_dir (rebasehelper.patch\_helper.GitPatchTool attribute), 32
  - SourcePackageBuildError, 29
  - SOURCES
    - rebase-helper command line option, 11
  - sources (rebasehelper.specfile.SpecFile attribute), 37
  - spc (rebasehelper.specfile.SpecFile attribute), 37
  - spec\_content (rebasehelper.specfile.SpecFile attribute), 37
  - spec\_file (rebasehelper.application.Application attribute), 25
  - spec\_file\_path (rebasehelper.application.Application attribute), 25
  - SpecContent (class in rebasehelper.specfile), 33
  - SpecFile (class in rebasehelper.specfile), 33
  - SpecHooksRunner (class in rebasehelper.specfile), 37
  - split\_nevra() (rebasehelper.helpers.rpm\_helper.RpmHelper class method), 23
  - split\_version\_string() (rebasehelper.specfile.SpecFile static method), 37
  - SRPMBuildHelper (class in rebasehelper.build\_helper), 28
  - SRPMBuildToolBase (class in rebase-helper.build\_helper), 28
  - strip (rebasehelper.specfile.PatchObject attribute), 33
  - substitute\_path\_with\_macros() (rebase-helper.helpers.macro\_helper.MacroHelper static method), 19
  - SUCCESS (rebasehelper.logger.CustomLogger attribute), 30
- T**
- TarArchiveType (class in rebasehelper.archive), 26
  - TarBz2ArchiveType (class in rebasehelper.archive), 26
  - TarGzArchiveType (class in rebasehelper.archive), 26
  - TarXzArchiveType (class in rebasehelper.archive), 26
  - temp\_dir (rebasehelper.application.Application attribute), 25
  - TEMPDIR (rebasehelper.temporary\_environment.TemporaryEnvironment attribute), 37
  - TEMPDIR\_BUILD (rebase-helper.build\_helper.RpmbuildTemporaryEnvironment attribute), 28
  - TEMPDIR\_BUILDROOT (rebase-helper.build\_helper.RpmbuildTemporaryEnvironment attribute), 28
  - TEMPDIR\_RESULTS (rebase-helper.build\_helper.BuildTemporaryEnvironment attribute), 27
  - TEMPDIR\_RPMBUILD (rebase-helper.build\_helper.RpmbuildTemporaryEnvironment attribute), 28
  - TEMPDIR\_RPMS (rebase-helper.build\_helper.RpmbuildTemporaryEnvironment attribute), 28
  - TEMPDIR\_SOURCES (rebase-helper.build\_helper.BuildTemporaryEnvironment attribute), 27
  - TEMPDIR\_SPEC (rebase-helper.build\_helper.BuildTemporaryEnvironment attribute), 27
  - TEMPDIR\_SPECS (rebase-helper.build\_helper.BuildTemporaryEnvironment attribute), 27
  - TEMPDIR\_SRPMS (rebase-helper.build\_helper.RpmbuildTemporaryEnvironment attribute), 28
  - TemporaryEnvironment (class in rebase-helper.temporary\_environment), 37
  - terminal\_background (rebase-helper.logger.ColorizingStreamHandler attribute), 30
  - TgzArchiveType (class in rebasehelper.archive), 26

TRACE (rebasehelper.logger.CustomLogger attribute), 30

## U

update\_paths\_to\_patches() (rebasehelper.specfile.SpecFile method), 37

update\_setup\_dirname() (rebasehelper.specfile.SpecFile method), 37

update\_sources() (rebasehelper.helpers.lookaside\_cache\_helper.LookasideCacheHelper class method), 19

upload\_srpm() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 18

use\_colors (rebasehelper.helpers.console\_helper.ConsoleHelper attribute), 15

## V

VERBOSE (rebasehelper.logger.CustomLogger attribute), 30

## W

wait\_for\_task() (rebasehelper.build\_helper.BuildToolBase class method), 28

watch\_build() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 15

watch\_koji\_tasks() (rebasehelper.helpers.koji\_helper.KojiHelper class method), 18

write\_updated\_patches() (rebasehelper.specfile.SpecFile method), 37

## Z

ZipArchiveType (class in rebasehelper.archive), 26