

---

# **rebase-helper**

***Release 0.22.1.dev16+g1c19c48***

**Jul 17, 2020**



---

## 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>
<b>5</b>	<b>Videos</b>	<b>11</b>
5.1	User Guide . . . . .	12
5.2	API . . . . .	18
5.3	Change Log . . . . .	46
	<b>Python Module Index</b>	<b>63</b>
	<b>Index</b>	<b>65</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 quay.io/rebasehelper/rebase-helper:latest
```

See [container 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

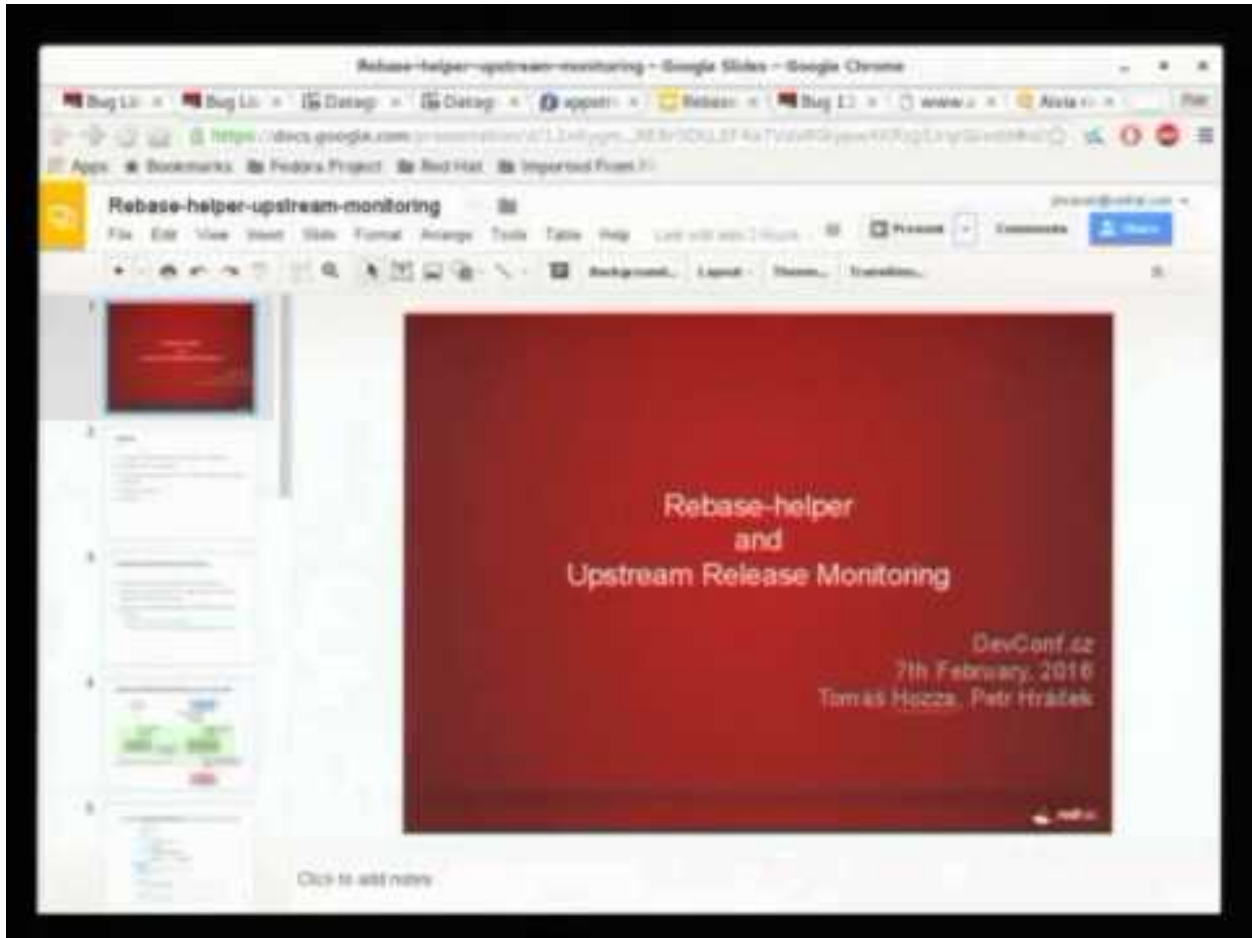
- **Comparison**

- 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



Another presentation, this time from DevConf.CZ 2016, by Petr Hráček and Tomáš Hozza:



## 5.1 User Guide

### 5.1.1 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



**rebase-helper** is also published on [PyPI](#).

## 5.1.2 Logging

Since version 0.18.0 **rebase-helper** uses standard module-level loggers.

There are also 3 loggers with a special purpose:

Logger name	Purpose
<code>rebasehelper.traceback</code>	traceback generated by unhandled exceptions
<code>rebasehelper.summary</code>	messages forming the rebase summary
<code>rebasehelper.report</code>	messages forming the textual rebase report (used by <code>text</code> output tool)

**rebase-helper** uses `rebasehelper.logger.CustomLogger` logger class which provides extra logging levels.

`rebasehelper.logger.LoggerHelper` class provides 3 utility methods to manage default handlers:

- `rebasehelper.logger.LoggerHelper.create_stream_handlers()` sets up the default console handlers for `rebasehelper` and `rebasehelper.summary` loggers.
- `rebasehelper.logger.LoggerHelper.create_file_handlers()` sets up the default file handlers for `rebasehelper` logger, with INFO, VERBOSE and DEBUG levels. The respective log files are located in `rebase-helper-results/logs/`.
- `rebasehelper.logger.LoggerHelper.remove_file_handlers()` removes specified file handlers from `rebasehelper` logger.

## Examples

```
# use a single module and get logs from it

import logging

from rebasehelper.specfile import SpecFile

logger = logging.getLogger('rebasehelper.specfile')
logger.addHandler(logging.StreamHandler())

spec = SpecFile('test.spec')
spec.set_release_number(2)
```

```
# run a complete rebase with default log handlers

from rebasehelper.logger import LoggerHelper
from rebasehelper.config import Config
from rebasehelper.cli import CLI
from rebasehelper.application import Application

LoggerHelper.create_stream_handlers()
config = Config()
cli = CLI()
config.merge(cli)
exec_dir, res_dir = Application.setup(config)
# default file handlers are automatically created and removed by Application instance,
```

(continues on next page)

(continued from previous page)

```
# unless disabled by passing create_logs=False
app = Application(config, exec_dir, res_dir)
app.run()
```

### 5.1.3 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.

### 5.1.4 Usage

#### SYNOPSIS

```
rebase-helper [-h] [--version] [-v] [--color {always,never,auto}]
  [--background {dark,light,auto}] [--results-dir RESULTS_DIR]
  [--workspace-dir WORKSPACE_DIR]
  [--buildtool {copr,koji,mock,rpmbuild}]
  [--srpm-buildtool {mock,rpmbuild}]
  [--pkgcomparetool [{abipkgdiff,csmock,licensecheck,pkgdiff,rpmdiff,
  rpminspect-rpm,rpminspect-srpm,sonamecheck}]]
  [--outputtool {json,text}]
  [--versioneer {anitya,cpan,hackage,npmjs,pypi,rubygems}]
  [--versioneer-blacklist [{anitya,cpan,hackage,npmjs,pypi,rubygems}]]
  [--spec-hook-blacklist
  [{commit-hash-updater,escape-macros,paths-to-rpm-macros,pypi-url-fix,
  replace-old-version,ruby-helper,typo-fix}]]
  [--build-log-hook-blacklist [{files}]] [--bugzilla-id BUG_ID]
  [--non-interactive] [--favor-on-conflict {downstream,upstream,off}]
  [--not-download-sources] [-w] [--apply-changes]
  [--disable-inapplicable-patches] [--skip-version-check]
  [--update-sources] [--skip-upload] [--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]
  [-D 'MACRO EXPR'] [--copr-project-permanent]
  [--copr-project-frontpage] [--copr-chroots COPR_CHROOTS]
  [--replace-old-version-with-macro] [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

location where the rebase-helper-results directory will be created

#### --workspace-dir WORKSPACE\_DIR

location where the rebase-helper-workspace directory will be created

#### --buildtool {copr, koji, mock, rpmbuild}

build tool to use, defaults to mock

#### --srpm-buildtool {mock, rpmbuild}

SRPM build tool to use, defaults to rpmbuild

#### --pkgcomparetool [{abipkgdiff, csmock, licensecheck, pkgdiff, rpmdiff, rpminspect-rpm, rpminspect}

set of tools to use for package comparison, defaults to abipkgdiff, licensecheck, pkgdiff, rpmdiff, rpminspect-rpm, rpminspect-srpm, sonamecheck if available

#### --outputtool {json, text}

tool to use for formatting rebase output, defaults to text

#### --versioneer {anitya, cpan, hackage, npmjs, pypi, rubygems}

tool to use for determining latest upstream version

#### --versioneer-blacklist [{anitya, cpan, hackage, npmjs, pypi, rubygems}]

prevent specified versioneers from being run

**--spec-hook-blacklist** [{commit-hash-updater,escape-macros,paths-to-rpm-macros,pypi-url-fix,  
prevent specified spec hooks from being run

**--build-log-hook-blacklist** [{files}]  
prevent specified build log hooks from being run

**--bugzilla-id** BUG\_ID  
do a rebase based on Upstream Release Monitoring bugzilla

**--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

**--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

**--skip-upload**  
skip uploading 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

**-D 'MACRO EXPR', --define 'MACRO EXPR'**  
define an rpm macro, can be used multiple times

**--copr-project-permanent**  
make the created copr project permanent

- copr-project-frontpage**  
make the created copr project visible on the frontpage
- copr-chroots** COPR\_CHROOTS  
comma-separated list of chroots to create copr project with
- replace-old-version-with-macro**  
replace old version string with %{version} instead of new version string

## 5.1.5 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, build log 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.plugins.build_tools.rpm. BuildToolBase
SRPM build tool	rebasehelper. srpm_build_tools	rebasehelper.plugins.build_tools.srpm. SRPMBuildToolBase
checker	rebasehelper.checkers	rebasehelper.plugins.checkers.BaseChecker
output tool	rebasehelper. output_tools	rebasehelper.plugins.output_tools. BaseOutputTool
SPEC hook	rebasehelper. spec_hooks	rebasehelper.plugins.spec_hooks. BaseSpecHook
build log hook	rebasehelper. build_log_hooks	rebasehelper.plugins.build_log_hooks. BaseBuildLogHook
versioneer	rebasehelper. versioneers	rebasehelper.plugins.versioneers. BaseVersioneer

### Example

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

```

from rebasehelper.plugins.spec_hooks 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

        :param spec_file: SpecFile object representing original SPEC file
        :param rebase_spec_file: SpecFile object representing rebased SPEC file

```

(continues on next page)

(continued from previous page)

```

"""
    rebase_spec_file.spec_content.section('%package').insert(0, '# processed by {}
↪\n'.format(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']
    }
)

```

## 5.2 API

### 5.2.1 Helpers

#### Bugzilla helper module

```

class rebasehelper.helpers.bugzilla_helper.BugzillaHelper
    Class for working with Upstream Release Monitoring on bugzilla.

    BUGZILLA_REST_API_URL = 'https://bugzilla.redhat.com/rest'
    DIST_GIT_REPO_URL = 'https://src.fedoraproject.org/rpms'
    UPSTREAM_RELEASE_MONITORING_USERNAME = 'upstream-release-monitoring'

    classmethod clone_repository (component: str, bugzilla_id: str) → str
        Clones remote dist-git repository of a component.

        Parameters
            • component – Package to clone.
            • bugzilla_id – ID of the bugzilla.

        Returns Path to the cloned repository.

        Raises
            • RebaseHelperError – If the directory, that the repository
            • is supposed to be cloned into, exists.

    classmethod get_bugzilla_component (bugzilla_id: str) → str
        Gets a component of the bugzilla.

        Parameters bugzilla_id – ID of the bugzilla.

```

**Returns** Component of the bugzilla.

**Raises**

- `RebaseHelperError` – If no such bugzilla exists or if the bugzilla
- was not created by Upstream Release Monitoring.

**classmethod `get_version_from_comments`** (*bugzilla\_id: str*) → Optional[str]  
Gets version from bugzilla comments.

**Parameters** `bugzilla_id` – ID of the bugzilla.

**Returns** Version specified by Upstream Release Monitoring in comments or None, if no version could be found.

**Raises** `RebaseHelperError` – If no such bugzilla exists.

**classmethod `prepare_rebase_repository`** (*bugzilla\_id: str*) → Tuple[str, str]  
Clones a repository based on Upstream Release Monitoring bugzilla.

**Parameters** `bugzilla_id` – ID of the bugzilla.

**Returns** Path of the cloned repository and version to rebase to.

**Raises**

- `RebaseHelperError` – If there was an error while obtaining
- data from bugzilla or if there was a problem while cloning
- the repository.

## 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
```

```
DELETE_PROJECT_AFTER = 60
```

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

```
classmethod create_project (client, project, chroots, description, instructions, permanent=False, hide=True)
```

```
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` (*login=False*, *profile='koji'*)

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

**Parameters**

- **login** (*bool*) – Whether to perform a login.
- **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_build` (*session, package, version*)

Looks up Koji build of a specific version of a package.

**Parameters**

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

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

**Return type** tuple

**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 old build info from Koji.

**Parameters**

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

**Returns** Koji build id, package 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, target\_dir=None*)

**rpkg\_config\_dir** = '/etc/rpkg'

**classmethod** **update\_sources** (*tool, basepath, package, old\_sources, new\_sources, upload=True, source\_dir=""*)

**Macro helper module**

**class** rebasehelper.helpers.macro\_helper.**MacroHelper**

Class for working with RPM macros

**MACROS\_WHITELIST** = ['\_bindir', '\_datadir', '\_includedir', '\_infodir', '\_initdir', '\_libdir']

**static** **dump** ()

Gets list of all defined macros.

**Returns** All defined macros.

**Return type** list

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

**classmethod** **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

**classmethod** **purge\_macro** (*macro: str*) → None

**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, ignore_stderr=False)`

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.
- **ignore\_stderr** (*bool*) – Whether to ignore stderr output.

**Returns** Exit code of the subprocess.

**Return type** int

**static** `run_subprocess_cwd(cmd, cwd=None, input_file=None, output_file=None, ignore_stderr=False, 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.
- **ignore\_stderr** (*bool*) – Whether to ignore stderr output.
- **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, out-
                             put_file=None, ignore_stderr=False, 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.
- **ignore\_stderr** (*bool*) – Whether to ignore stderr output.
- **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, ig-
                           nore_stderr=False, 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.
- **ignore\_stderr** (*bool*) – Whether to ignore stderr output.
- **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.RpmHeader (hdr: <sphinx.ext.autodoc.importer._MockObject
                                                  object at 0x7f5f968b5048>)
```

```
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** *RpmHeader*

**classmethod** `get_rpm_spec(path: str, sourcedir: str, predefined_macros: Dict[str, str])` → `<sphinx.ext.autodoc.importer._MockObject object at 0x7f5f968b55f8>`

**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

## 5.2.2 Plugins

### Build Log Hooks package

**class** `rebasehelper.plugins.build_log_hooks.BaseBuildLogHook`  
Base class for a build log hook.

**CATEGORIES** = []

**classmethod** `format(data)`  
Formats build log hook output to a readable text form.

**classmethod** `merge_two_results(old, new)`

**classmethod** `run(spec_file, rebase_spec_file, results_dir, **kwargs)`  
Runs the build log hook.

**Parameters**

- `spec_file` (`rebasehelper.specfile.SpecFile`) – Original SpecFile object.



- **rebase\_spec\_file** (`rebasehelper.specfile.SpecFile`) – Rebased Spec-File object.
- **kwargs** (`dict`) – Keyword arguments from instance of Application.

**Returns** The first element is a dict containing changes, the second is a bool whether the build process should be restarted.

**Return type** tuple

```
class rebasehelper.plugins.build_log_hooks.BuildLogHookCollection (entrypoint:
                                                                    str, man-
                                                                    ager:
                                                                    PluginMan-
                                                                    ager)
```

```
run (spec_file, rebase_spec_file, non_interactive, force_build_log_hooks, **kwargs)
```

Runs all non-blacklisted build log hooks.

**Parameters**

- **spec\_file** (`rebasehelper.specfile.SpecFile`) – Original SpecFile object.
- **rebase\_spec\_file** (`rebasehelper.specfile.SpecFile`) – Rebased Spec-File object.
- **non\_interactive** (`bool`) – Whether rebase-helper is in non-interactive mode.
- **force\_build\_log\_hooks** (`bool`) – Whether to run the hooks even in non-interactive mode.
- **kwargs** (`dict`) – Keyword arguments from instance of Application.

**Returns** Whether build log hooks made some changes to the SPEC file.

**Return type** bool

## Build Tools package

```
class rebasehelper.plugins.build_tools.BuildTemporaryEnvironment (sources,
                                                                    patches,
                                                                    spec, re-
                                                                    sults_dir)
```

Class representing temporary environment.

```
TEMPDIR_RESULTS = 'TEMPDIR_RESULTS'
```

```
TEMPDIR_SOURCES = 'TEMPDIR_SOURCES'
```

```
TEMPDIR_SPEC = 'TEMPDIR_SPEC'
```

```
TEMPDIR_SPECS = 'TEMPDIR_SPECS'
```

```
class rebasehelper.plugins.build_tools.MockTemporaryEnvironment (sources,
                                                                    patches, spec,
                                                                    results_dir)
```

Class representing temporary environment for MockBuildTool.

```
class rebasehelper.plugins.build_tools.RpmbuildTemporaryEnvironment (sources,
                                                                    patches,
                                                                    spec, re-
                                                                    sults_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'
```

`rebasehelper.plugins.build_tools.check_mock_privileges()` → bool

`rebasehelper.plugins.build_tools.get_mock_logfile_path(ret, results_dir, tmp_path=None)`

Get path to logfile containing the error message

#### Parameters

- **ret** – return code from mock
- **results\_dir** – directory where logs will be stored
- **tmp\_path** – temporary directory where logs are during build

#### Returns

## Checkers package

**class** `rebasehelper.plugins.checkers.BaseChecker`

Base class of package checkers.

#### DEFAULT

If True, the checker is run by default.

**Type** bool

#### CATEGORY

Category which determines when the checker is run.

**Type** *CheckerCategory*

#### results\_dir

Path where the results are stored.

**Type** str

**CATEGORY = None**

**DEFAULT = False**

**classmethod** `format(data)`

Formats checker output to a readable text form.

**classmethod** `get_category()`

**classmethod** `get_checker_output_dir_short()`

Return short version of checker output directory

**classmethod** `get_important_changes(checker_output)`

Each checker has an opportunity to highlight some of its output. This function is optional, as not all checkers provide output with critical information.

**Parameters** `checker_output` (*dict*) – Dictionary with output from the given checker.

**Returns** List of strings to be output to CLI as warning messages.

**Return type** list

**classmethod** `get_underlined_title` (*text*, *separator*='=')

**classmethod** `is_available` ()

**classmethod** `prepare_results_dir` ()

Creates checker's results dir.

Removes the existing results dir if it exists to avoid collisions.

**results\_dir** = None

**classmethod** `run_check` (*results\_dir*, *\*\*kwargs*)

Perform the check itself and return results.

**class** `rebasehelper.plugins.checkers.CheckerCategory`

An enumeration.

**RPM** = 3

**SOURCE** = 1

**SRPM** = 2

**class** `rebasehelper.plugins.checkers.CheckerCollection` (*entrypoint*: *str*, *manager*: *PluginManager*)

Class representing the process of running various checkers on final packages.

**get\_default\_plugins** (*return\_one*: *bool* = *False*) → Union[str, List[str], None]

**get\_supported\_plugins** () → List[str]

**run** (*results\_dir*: *str*, *checker\_name*: *str*, *\*\*kwargs*) → Optional[Dict[str, Any]]

Runs a particular checker and returns the results.

**Parameters**

- **results\_dir** – Path to a directory in which the checker should store the results.
- **checker\_name** – Name of the checker to be run.

**Raises** `NotImplementedError` – If a checker with the given name doesn't exist.

**Returns** Results of the checker.

## Output Tools package

**class** `rebasehelper.plugins.output_tools.BaseOutputTool`

Base class for an output tool

**DEFAULT** = **False**

**EXTENSION** = ''

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

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

Prepends a path with path to rebase-helper-results.

Takes directory changes (such as cd into cloned repo with `-bugzilla-id`) into account.

**Parameters**

- **app** – Application instance.
- **path\_members** – Path members to prepend the results dir to.

**Returns** Prepended path.

**classmethod print\_cli\_summary** (*app*)  
Outputs a summary of a rebase-helper run.

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

**classmethod print\_important\_checkers\_output** ()  
Iterates over all checkers output to highlight important checkers warning

**classmethod print\_patches\_cli** ()  
Outputs info about patches

**classmethod print\_patches\_section\_cli** (*logger\_method*, *patch\_type*)  
Outputs information about one of the patch types.

**Parameters**

- **logger\_method** – Method to use for logging
- **patch\_type** – A key to use for filtering patch dictionary obtained from results\_store.

**classmethod print\_report\_file\_path** (*app*)  
Outputs path to the report file

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

**class** rebasehelper.plugins.output\_tools.**OutputToolCollection** (*entrypoint*: *str*,  
*manager*: *Plugin-Manager*)

Class representing the process of running various output tools.

**run** (*tool*, *logs*=None, *app*=None)  
Runs specified output tool.

**Parameters**

- **tool** (*str*) – Tool to run.
- **logs** (*list*) – Build logs containing information about the failed rebase.
- **app** (*rebasehelper.application.Application*) – Application class instance.

## Plugin Collection module

**class** rebasehelper.plugins.plugin\_collection.**PluginCollection** (*entrypoint*: *str*,  
*manager*: *Plugin-Manager*)

**get\_all\_plugins** () → List[str]

**get\_default\_plugins** (*return\_one*: *bool* = False) → Union[str, List[str], None]

**get\_options** () → List[Union[Dict[str, Any], List[Dict[str, Any]]]]  
Gets options of all plugins of one type.

**Returns** List of plugins' options.

**Return type** list

**get\_plugin** (*tool*: *str*) → Optional[Type[rebasehelper.plugins.plugin.Plugin]]

**get\_supported\_plugins** () → List[str]

## Plugin Manager module

```
class rebasehelper.plugins.plugin_manager.PluginManager
```

```
COLLECTIONS = [<class 'rebasehelper.plugins.build_log_hooks.BuildLogHookCollection'>],
```

```
get_options () → List[Union[Dict[str, Any], List[Dict[str, Any]]]]
```

Gets options of all plugins.

**Returns** List of plugins' options.

**Return type** list

## RPM Build Tools package

```
class rebasehelper.plugins.build_tools.rpm.BuildToolBase
```

Build tool base class.

**DEFAULT**

If True, the build tool is default tool.

**Type** bool

**ACCEPTS\_OPTIONS**

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

**Type** bool

**CREATES\_TASKS**

If True, the build tool creates remote tasks.

**Type** bool

**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_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.plugins.build_tools.rpm.BuildToolCollection` (*entrypoint: str, manager: PluginManager*)

Collection of RPM build tools.

## SPEC Hooks package

**class** `rebasehelper.plugins.spec_hooks.BaseSpecHook`

Base class for a spec hook

**CATEGORIES** = []

**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.plugins.spec_hooks.SpecHookCollection` (*entrypoint: str, manager: PluginManager*)

Class representing the process of running various spec file hooks.

**run** (*spec\_file, rebase\_spec\_file, \*\*kwargs*)

Runs all non-blacklisted spec hooks.

**Parameters**

- `spec_file` (`rebasehelper.specfile.SpecFile`) – Original `SpecFile` object.
- `rebase_spec_file` (`rebasehelper.specfile.SpecFile`) – Rebased `SpecFile` object.
- `**kwargs` – Keyword arguments from Application instance.

## SRPM Build Tools package

**class** `rebasehelper.plugins.build_tools.srpm.SRPMBuildToolBase`

SRPM build tool base class.

**DEFAULT**

If True, the build tool is default tool.

**Type** bool

**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

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

```
class rebasehelper.plugins.build_tools.srpm.SRPMBuildToolCollection (entrypoint: str, manager: PluginManager)
```

Collection of SRPM build tools.

**Versioneers package**

```
class rebasehelper.plugins.versioneers.BaseVersioneer
```

Base class for a versioneer

**CATEGORIES = []**

```
classmethod run (package_name)
```

Runs a versioneer.

**Parameters** **package\_name** – Name of a package

**Returns** Latest upstream version of a package

```
class rebasehelper.plugins.versioneers.VersioneerCollection (entrypoint: str, manager: PluginManager)
```

```
run (versioneer, package_name, category, versioneer_blacklist=None)
```

Runs the specified versioneer or all versioneers subsequently until one of them succeeds.

**Parameters**

- **versioneer** (*str*) – Name of a versioneer.
- **package\_name** (*str*) – Name of a package.
- **category** (*str*) – Package category.
- **versioneer\_blacklist** (*list*) – List of versioneers that will be skipped.

**Returns** Latest upstream version of a package.

**Return type** str

### 5.2.3 Application module

```
class rebasehelper.application.Application (cli_conf: rebasehelper.config.Config,  
start_dir: str, execution_dir: str, results_dir:  
str, create_logs: bool = True)
```

```
apply_changes ()
```

```
build_binary_packages ()
```

Function calls build class for building packages

```
build_source_packages ()
```

```
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_new_build_logs ()
```

```
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. :param exception: Error message from rebase-helper :return:

```
print_task_info (builder)
```

```
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)
```



## 5.2.4 Archive module

```

class rebasehelper.archive.Archive(filename)
    Class representing an archive with sources

    extract_archive(path)
        Extracts the archive into the given path

        Parameters path – Path where to extract the archive to.

    classmethod get_supported_archives()

class rebasehelper.archive.ArchiveTypeBase
    Base class for various archive types

    EXTENSION = ''

    classmethod extract(archive, filename, path)
        Extracts the archive into the given path

        Parameters path – Path where to extract the archive to.

        Returns

    classmethod match(filename)
        Checks if the filename matches the archive type. If yes, returns True, otherwise returns False.

    classmethod open(filename)
        Opens archive with the given filename and returns the proper archive type object.

class rebasehelper.archive.Bz2ArchiveType

    EXTENSION = '.bz2'

class rebasehelper.archive.CrateArchiveType

    EXTENSION = '.crate'

class rebasehelper.archive.GemArchiveType

    EXTENSION = '.gem'

    class GemArchive(filename)

        close()

        extract(path)

    classmethod extract(archive, filename, path)
        Extracts the archive into the given path

        Parameters path – Path where to extract the archive to.

        Returns

    classmethod open(filename)
        Opens archive with the given filename and returns the proper archive type object.

class rebasehelper.archive.TarArchiveType

    EXTENSION = '.tar'

```

**class** rebasehelper.archive.TarBz2ArchiveType

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

**classmethod extract** (*archive, filename, path*)

Extracts the archive into the given path

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

**Returns**

**classmethod open** (*filename*)

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

**class** rebasehelper.archive.TarGzArchiveType

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

**classmethod extract** (*archive, filename, path*)

Extracts the archive into the given path

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

**Returns**

**classmethod open** (*filename*)

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

**class** rebasehelper.archive.TarXzArchiveType

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

**classmethod extract** (*archive, filename, path*)

Extracts the archive into the given path

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

**Returns**

**classmethod open** (*filename*)

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

**class** rebasehelper.archive.TgzArchiveType

**EXTENSION** = '.tgz'

**class** rebasehelper.archive.ZipArchiveType

**EXTENSION** = '.zip'

**classmethod extract** (*archive, filename, path*)

Extracts the archive into the given path

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

**Returns**

**classmethod match** (*filename*)

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

**classmethod open** (*filename*)

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

`rebasehelper.archive.register_archive_type (archive)`

## 5.2.5 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`

**classmethod convert\_macros\_to\_dict** (*macros\_list: List[str]*) → Dict[str, str]

Converts macros from CLI to a dictionary.

**Parameters macros\_list** – List of macros in the format ‘MACRO EXPRESSION’.

**Returns** The converted macros, MACRO are keys, EXPRESSION are values.

**Raises** `RebaseHelperError` if the macros don’t follow the correct format.

**classmethod run** ()

## 5.2.6 Exception module

**exception** `rebasehelper.exceptions.BinaryPackageBuildError (*args, **kwargs)`

Error indicating failure during the build of binary package.

**exception** `rebasehelper.exceptions.CheckerNotFoundError`

Error indicating not being able to find checker binary.

**exception** `rebasehelper.exceptions.DownloadError`

Exception indicating that download of a file failed.

**exception** `rebasehelper.exceptions.LookasideCacheError`

Exception indicating a problem in 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.

**exception** `rebasehelper.exceptions.SourcePackageBuildError (*args, **kwargs)`

Error indicating failure during the build of source package

## 5.2.7 Logger module

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

**colors** = {'dark': {10: {'bg': 'default', 'fg': 'brightblack', 'style': None}, 11:

**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: logging.Logger, path: str, formatter: Optional[logging.Formatter] = None, level: Optional[int] = None) → Optional[logging.FileHandler]
```

Adds file handler to the given logger.

#### Parameters

- **logger** – Logger object to add the handler to.
- **path** – Path to a log file.
- **formatter** – Formatter object used to format logged messages.
- **level** – Severity threshold.

**Returns** Created file handler instance or None if creation failed.

```
static add_stream_handler(logger: logging.Logger, level: Optional[int] = None, formatter: Optional[logging.Formatter] = None) → rebasehelper.logger.ColorizingStreamHandler
```

Adds stream handler to the given logger.

#### Parameters

- **logger** – Logger object to add the handler to.
- **level** – Severity threshold.
- **formatter** – Formatter object used to format logged messages.

**Returns** Created stream handler instance.

```
classmethod create_file_handlers(results_dir: str) → List[logging.FileHandler]
```

Creates rebase-helper file handlers.

**Parameters** **results\_dir** – Path to rebase-helper-results directory.

**Returns** List of created file handler instances.

```
classmethod create_stream_handlers() → Tuple[rebasehelper.logger.ColorizingStreamHandler, rebasehelper.logger.ColorizingStreamHandler]
```

```
memory_handler = None
```

**classmethod** `remove_file_handlers` (*handlers: List[logging.FileHandler]*) → None  
Removes rebase-helper file handlers.

**Parameters** `handlers` – List of file handlers to remove.

**classmethod** `remove_memory_handler` () → None

**classmethod** `setup_memory_handler` () → None

**class** `rebasehelper.logger.MemoryHandler`

BufferingHandler with infinite capacity

**replay\_into** (*target: logging.Handler*) → None

**shouldFlush** (*record: logging.LogRecord*) → bool

Should the handler flush its buffer?

Returns true if the buffer is up to capacity. This method can be overridden to implement custom flushing strategies.

## 5.2.8 Patcher module

**class** `rebasehelper.patcher.Patcher`

Class for git command used for patching old and new sources

**classmethod** `apply_old_patches` (*source\_dir*)

Function applies a patch to a old/new sources

**classmethod** `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`

**static** `decorate_patch_name` (*patch\_name*)

**classmethod** `extract_patch_name` (*message*)

**classmethod** `init_git` (*directory*)

Function initialize old and new Git repository

**classmethod** `insert_patch_name` (*message, patch\_name*)

`new_repo = None`

`new_sources = None`

`non_interactive = False`

`old_repo = None`

`old_sources = None`

`output_data = None`

**classmethod** `patch` (*old\_dir, new\_dir, rest\_sources, patches, \*\*kwargs*)

The function can be used for patching one directory against another

`patches = []`

**classmethod** `strip_patch_name` (*diff, patch\_name*)

## 5.2.9 Spec content module

**class** `rebasehelper.spec_content.SpecContent` (*content: str*)

Class representing content of a SPEC file.

**DISALLOW\_INLINE\_COMMENTS** = ['%package', '%patchlist', '%sourcelist', '%description', '%

**SECTION\_HEADERS** = ['%package', '%prep', '%generate\_build\_requires', '%build', '%instal

**classmethod** `get_comment_span` (*line: str, section: str*) → Tuple[int, int]

Gets span of a comment depending on the section.

**Parameters**

- **line** – Line to find the comment in.
- **section** – Section the line is in.

**Returns** Span of the comment. If no comment is found, both tuple elements are equal to the length of the line for convenient use in a slice.

**replace\_section** (*name: str, content: List[str]*) → bool

Replaces content of a section.

In case there are multiple sections with the same name, the first one is replaced.

**Parameters**

- **name** – Section name.
- **content** – Section content as a list of lines.

**Returns** False if section was not found else True.

**section** (*name: str*) → Optional[List[str]]

Gets content of a section.

In case there are multiple sections with the same name, the first one is returned.

**Parameters** **name** – Section name.

**Returns** Section content as a list of lines.

## 5.2.10 Specfile module

**class** `rebasehelper.specfile.PackageCategory`

An enumeration.

**R** = `re.compile('^R-')`

**haskell** = `re.compile('^ghc-')`

**nodejs** = `re.compile('^nodejs-')`

**perl** = `re.compile('^perl-')`

**php** = `re.compile('^php-')`

**python** = `re.compile('^python[23]?-')`

**ruby** = `re.compile('^rubygem-')`

**rust** = `re.compile('^rust-')`

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

**class** `rebasehelper.specfile.PatchObject` (*path, index, strip*)  
 Class represents set of information about patches

**get\_patch\_name** ()

**class** `rebasehelper.specfile.SpecFile` (*path: str, sources\_location: str = "", predefined\_macros: Optional[Dict[str, str]] = None*)  
 Class representing a SPEC file. Be aware that using `SpecFile` modifies RPM macros in global context.

**copy** (*new\_path*)  
 Creates a copy of the current object and copies the SPEC file to a new location.

**Parameters** `new_path` (*str*) – Path to copy the new SPEC file to.

**Returns** The created `SpecFile` instance.

**Return type** *SpecFile*

**download\_remote\_sources** ()  
 Method that iterates over all sources and downloads ones, which contain URL instead of just a file.

**Returns** None

**static extract\_version\_from\_archive\_name** (*archive\_path: str, main\_source: str*) → *str*  
 Extracts version string from source archive name.

**Parameters**

- **archive\_path** – Path to the main sources archive.
- **main\_source** – Value of Source0 tag.

**Returns** Extracted version string.

**Raises** `RebaseHelperError` in case version can't be determined.

**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** () → *str*

**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\_main\_files\_section** ()  
 Finds the exact name of the main %files section.

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

**Return type** *str*

**get\_main\_source** () → *str*

**get\_new\_log** (*changelog\_entry*)

Constructs a new changelog entry.

**Parameters** **changelog\_entry** (*str*) – Message to use in the entry.

**Returns** List of lines of the new entry.

**Return type** list

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

Method returns list of all unplied patches.

**Returns** list of PatchObject

**get\_patches** ()

Method returns list of all applied and not applied patches

**Returns** list of PatchObject

**get\_prep\_section** ()

Function returns whole prep section

**get\_raw\_tag\_value** (*tag\_name: str, section: Union[str, int, None] = None*) → Optional[str]

**get\_release** () → str

Returns release string without %dist

**get\_setup\_dirname** ()

Get dirname from %setup or %autosetup macro arguments

**Returns** dirname

**get\_sources** () → List[str]

Gets a list of local sources.

**static get\_subpackage\_name** (*files\_section*)

Gets subpackage name based on the %files section.

**get\_version** () → str

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

Returns whether test suite is enabled during the build time

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

**parse\_release** () → Tuple[bool, int, Optional[str]]

Parses release string.

**Returns** Tuple of is\_prerelease, release\_number and extra\_version.

**Raises** RebaseHelperError in case release string is not valid.

**reload** ()

Reloads the whole Spec file.

**save** () → None

Saves changes made to SpecContent and updates the internal state.

**set\_extra\_version** (*extra\_version: Optional[str], version\_changed: bool*) → None

Updates SPEC file with the specified extra version.

**Parameters**

- **extra\_version** – Extra version string or None.
- **version\_changed** – Whether version (the value of Version tag) changed.



**set\_raw\_tag\_value** (*tag\_name: str, value: str, section: Union[str, int, None] = None*) → None

**set\_release** (*release: str*) → None

**set\_release\_number** (*release: str*) → None

**set\_tag** (*\*args, \*\*kwargs*)

**set\_version** (*version: str*) → None

**static split\_version\_string** (*version\_string: str, current\_version: str*) → Tuple[str, Optional[str]]  
Splits version string into version and extra version.

#### Parameters

- **version\_string** – Complete version string.
- **current\_version** – Current version (the value of Version tag).

**Returns** Tuple of version and extra\_version.

**Raises** RebaseHelperError in case passed version string is not valid.

**tag** (*name: str, section: Union[str, int, None] = None*) → Optional[rebasehelper.tags.Tag]  
Returns the first non-unique tag.

**update** () → None

**update\_changelog** (*\*args, \*\*kwargs*)

**update\_paths\_to\_sources\_and\_patches** (*\*args, \*\*kwargs*)

**update\_setup\_dirname** (*\*args, \*\*kwargs*)

**write\_updated\_patches** (*\*args, \*\*kwargs*)

`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

`rebasehelper.specfile.saves` (*func*)  
Decorator for saving the SpecFile after a method is run.

## 5.2.11 Tags module

**class** `rebasehelper.tags.Tag` (*section\_index: int, section\_name: str, line: int, name: str, value\_span: Tuple[int, int], valid: bool, index: Optional[int] = None*)

**class** `rebasehelper.tags.Tags` (*raw\_content: rebasehelper.spec\_content.SpecContent, parsed\_content: rebasehelper.spec\_content.SpecContent*)

**filter** (*section\_index: Optional[int] = None, section\_name: Optional[str] = None, name: Optional[str] = None, valid: Optional[bool] = True*) → Iterator[rebasehelper.tags.Tag]  
Filters tags based on section, name or validity. Defaults to all valid tags in all sections.

#### Parameters

- **section\_index** – If specified, includes tags only from section of this index.
- **section\_name** – If specified, includes tags only from sections of this name.
- **name** – If specified, includes tags matching this name. Wildcards are supported.
- **valid** – If specified, includes tags of this validity.

**Returns** Iterator of matching Tag objects.

## 5.2.12 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

## 5.3 Change Log

### 5.3.1 [Unreleased]

### 5.3.2 [0.22.0] - 2020-03-31

#### Added

- Added more type hints, including all public API methods

#### Fixed

- `SpecFile.reload()` no longer pointlessly calls `SpecFile._read_spec_content()`
- Deleted files are now skipped when detecting unresolved conflicts during `git rebase`
- Fixed detection of Koji log file containing build failure
- Adapted to changes in `git 2.26.0`
- Fixed unhandled exception during upload to lookaside cache

## Changed

- Removed no longer necessary workarounds from Fedora base images
- Simplified packit configuration
- Options `--pkgcomparetool`, `--versioneer-blacklist`, `--spec-hook-blacklist` and `--build-log-hook-blacklist` can now be specified without an argument to indicate none of the tools/hooks should be run

### 5.3.3 [0.21.0] - 2020-02-21

#### Added

- Added public API tests for `Tags` class
- Added support for `%patchlist` and `%sourcelist`
- Added support for automatic `Source/Patch` numbering

#### Fixed

- **commit-hash-updater** SPEC hook now handles empty release name
- `sources` is now ignored if it's not a regular file
- Fixed summary and report paths when using `--bugzilla-id` or `--results-dir`
- Fixed and extended detection of ABI changes reported by **abipkgdiff**
- Removed deprecated encoding parameter in `json.load()` for Python 3.9
- Fixed processing of remote patches
- Fixed handling of intermediate macros in `SpecFile.set_tag()`

#### Changed

- All RPM macros are now reset when `SpecFile` object is destroyed
- Renamed `docker` directory to `containers` and `Dockerfiles` to `Containerfiles`
- Switched from Docker Hub to `quay.io` for automatic image building
- Improved and cleaned up `SpecFile` tests
- **replace-old-version** SPEC hook can now replace also extraversion

### 5.3.4 [0.20.0] - 2019-12-06

#### Added

- Introduced `Tags` class unifying and simplifying access to SPEC tags
- Added proper support for `crate` and `gem` archives
- Added `--bugzilla-id` option to perform a rebase based on *Upstream Release Monitoring* bugzilla
- Added `-D/--define` option to define macros

- Added tests for public API

### Fixed

- Fixed `--build-tasks` option
- Fixed detecting unresolved conflicts in non-UTF-8 files
- Prevented loss of messages logged before logging file handlers are created
- **rebase-helper** now skips unparseable lines in `%prep` instead of tracebacking on them
- Fixed parsing SPEC files with `-h` in `%prep`
- Fixed processing SPEC files with zero-padded indexed tags and `%patch` macros

### Changed

- Completely reworked dealing with extraversions
- Improved `SpecFile.set_tag()` to minimize changes made to the SPEC file

## 5.3.5 [0.19.0] - 2019-09-26

### Added

- Added `--workspace-dir` option to allow specifying custom workspace directory
- Added **sonamecheck** checker for detecting *SONAME* changes
- Added `--copr-project-permanent`, `--copr-project-frontpage` and `--copr-chroots` options

### Fixed

- Strings like “1” are no longer replaced with macros in `%prep`
- SPEC files without *Source0* tag are now handled correctly
- Fixed **copr** build tool, switched to V3 API
- Avoided parsing SPEC without properly setting `%{_sourcedir}` macro first

### Changed

- Introduced `RpmHeader` class for more convenient access to package header attributes
- Modification of *Patch* tags now preserves whitespace to minimize differences in SPEC
- Moved Bash completion script from `/etc/bash_completion.d` to `/usr/share/bash-completion/completions`

## Removed

- Removed non-working `--patch-only`, `--build-only` and `--comparepkgs-only` options
- Temporarily removed `--continue` option
- Removed no longer used `python3-six` build dependency
- Removed `copr` workaround in favor of making **copr** build tool unavailable in case it's not working

## 5.3.6 [0.18.0] - 2019-08-21

### Added

- Added workaround for missing `mock` group in Fedora Rawhide

### Fixed

- Moved setup dependencies from `install_requires` to `setup_requires`

### Changed

- Refactored logging, see [logging documentation](#) for details
- Reason of build failure is now always logged

### Removed

- Removed unused `Application` methods and attributes
- Removed unused testing files

## 5.3.7 [0.17.2] - 2019-08-09

### Added

- Added tests for `SpecContent` class
- Enabled and configured [Packit-as-a-Service](#)

### Fixed

- Added exception handling to PyPI release webhook endpoint
- Fixed `TestCLI.test_cli_unit()` test
- Updated `MANIFEST.in` to include all necessary files

## Changed

- Improved tests for `Application` class
- `SpecFile.update_changelog()` now creates `%changelog` section if it doesn't exist
- **rebase-helper** now uses `setuptools-scm` to determine version from git
- `setup.py sdist` now supports overriding distribution base name with `--base-name` option

## 5.3.8 [0.17.1] - 2019-08-01

### Fixed

- Fixed PyPI release webhook endpoint

### Changed

- Removed direct dependencies preventing PyPI release

## 5.3.9 [0.17.0] - 2019-07-31

### Added

- Added possibility for plugins to specify their own arguments
- Added basic type hints and enabled mypy linter
- Added `rust` package category
- **replace-old-version** SPEC hook can now replace old version with `%{version}` macro
- **replace-old-version** SPEC hook can now replace also parts of version in Sources and Patches

### Fixed

- Fixed broken `ansicolors` dependency
- Fixed printing of output of unavailable checkers
- Fixed determining unmatched quotation in `%prep` section
- Made **files** build log hook handle conditions and macros in `%files` section
- **files** build log hook now adds man pages in a way that follows [Fedora Packaging Guidelines](#)
- **files** build log hook now ignores `debuginfo` files
- Fixed issues with polluted global macro namespace when parsing multiple different SPEC files
- Fixed handling of remote downstream patches (URLs)
- Fixed parsing of patch strip options in `%prep`
- Fixed wrapping of extra long lines in the usage documentation
- Sources from lookaside cache are now downloaded to a proper location

## Changed

- Restructured the code layout of plugins
- Refactored parts of SpecFile class
- Disabled removing of “unused” patches
- `mock` is now automatically run with superuser privileges if necessary
- Local builder is now used if `--get-old-build-from-koji` is specified and the build can't be downloaded from Koji
- checkers (even the default ones) are now skipped if they are not available
- Sources are now copied to destination if they cannot be extracted
- Reimplemented downloading Koji builds
- Updated documentation

## Removed

- Removed Python 2 support

### 5.3.10 [0.16.3] - 2019-05-03

#### Fixed

- Fixed handling of SPEC files with conditionalized sections
- Fixed **replace-old-version** SPEC hook not to update version in *%changelog* and local sources
- Fixed capturing RPM error output during SPEC parsing
- Fixed handling of absolute *%license* and *%doc* paths in **files** build log hook
- Fixed logging of SRPM and RPM build errors

#### Changed

- Updated packit configuration for packit 0.2.0
- Adapted to upcoming change in RPM python API
- Made `SpecFile` class more suitable for external use

### 5.3.11 [0.16.2] - 2019-03-07

#### Added

- Added support for `packit`
- Added SPEC hook for replacing old version string

### Fixed

- Fixed documentation building by mocking `requests-gssapi`
- Fixed `TestOutputTool` for checkers

### Changed

- Build log hooks are now run only if build of **new** binary packages fails
- It is now possible to use `--get-old-build-from-koji` option without FAS

## 5.3.12 [0.16.1] - 2019-02-28

### Fixed

- Made `GitPatchTool` auto-skip empty commits caused by new rebase implementation in **git 2.20**
- Fixed `TestGitHelper` to work on real systems with existing git configuration

## 5.3.13 [0.16.0] - 2019-02-27

### Added

- Added category for *R* packages
- Added `make test-podman` as an alternative to `make test-docker`
- Added `--skip-upload` option (to be used in conjunction with `--update-sources`)
- Added check that all sources for the new version are present
- Added SPEC hook for escaping macros in comments

### Changed

- `--get-old-build-from-koji` now tries to get specific version build (as opposed to the latest one)
- Implemented parsing of multiline macros and shell expansions in SPEC files
- **rebase-helper** can now handle multiline enquoted strings in *%prep* section
- Refactored `GitPatchTool` to make the rebase process more robust and to preserve as much of the original downstream patches as possible
- `git mergetool` is now run again if there are some unresolved conflicts left
- Associated comments are now removed along with patches

### Fixed

- Fixed populating list of logs on build failures
- Added missing abort after failed `git am`
- Fixed processing SPEC files without *%prep* section



- Fixed several issues in **ruby-helper** SPEC hook
- Fixed unwanted expansion of `%autosetup` macro
- Fixed automatic rebuilding based on build log hooks result
- Fixed removal of `%doc` and `%license` files in subpackages

#### Removed

- Removed `requests-kerberos` support and switched to `requests-gssapi` exclusively

### 5.3.14 [0.15.0] - 2018-12-21

#### Added

- Implemented build log hooks and added **files** hook to detect and fix missing/unpackaged files

#### Changed

- Refactored and simplified all plugins

#### Fixed

- Fixed not listing all argument choices while generating documentation
- Fixed error in parsing `rpmdiff` output
- Fixed insertion of extra blank lines to a SPEC file after removing patches

#### Removed

- Removed unneeded packages from base Docker image

### 5.3.15 [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

### 5.3.16 [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

### 5.3.17 [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

### 5.3.18 [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

### 5.3.19 [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

## 5.3.20 [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

### 5.3.21 [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  $\geq 1.6$
- `%prep` section parser is now able to handle backslash-split lines

### 5.3.22 [0.10.0] - 2017-08-25

#### Added

- Implemented extensible SPEC hooks and versioners
- 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

### 5.3.23 [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

### 5.3.24 [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

### 5.3.25 [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

### 5.3.26 [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

### 5.3.27 [0.7.1] - 2016-02-22

#### Added

- Two new command line options used by upstream monitoring

#### Fixed

- **fedpkg** reimplementation

### 5.3.28 [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

## 5.3.29 [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

## 5.3.30 [0.6.1] - 2015-10-30

### Added

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

## 5.3.31 [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`

## 5.3.32 [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

## 5.3.33 [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

## 5.3.34 [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

### 5.3.35 [0.3.0]

#### Added

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

### 5.3.36 [0.2.0]

#### Added

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

### 5.3.37 [0.1.0]

#### Added

- Initial classes
- CLI interface

### r

- rebasehelper.application, 36
- rebasehelper.archive, 37
- rebasehelper.cli, 39
- rebasehelper.exceptions, 39
- rebasehelper.helpers.bugzilla\_helper, 18
- rebasehelper.helpers.console\_helper, 19
- rebasehelper.helpers.copr\_helper, 20
- rebasehelper.helpers.download\_helper, 21
- rebasehelper.helpers.git\_helper, 21
- rebasehelper.helpers.input\_helper, 21
- rebasehelper.helpers.koji\_helper, 22
- rebasehelper.helpers.lookaside\_cache\_helper, 24
- rebasehelper.helpers.macro\_helper, 24
- rebasehelper.helpers.path\_helper, 25
- rebasehelper.helpers.process\_helper, 26
- rebasehelper.helpers.rpm\_helper, 27
- rebasehelper.logger, 39
- rebasehelper.patcher, 41
- rebasehelper.plugins.build\_log\_hooks, 28
- rebasehelper.plugins.build\_tools, 29
- rebasehelper.plugins.build\_tools.rpm, 33
- rebasehelper.plugins.build\_tools.srpm, 34
- rebasehelper.plugins.checkers, 30
- rebasehelper.plugins.output\_tools, 31
- rebasehelper.plugins.plugin\_collection, 32
- rebasehelper.plugins.plugin\_manager, 33
- rebasehelper.plugins.spec\_hooks, 34
- rebasehelper.plugins.versioners, 35
- rebasehelper.spec\_content, 42
- rebasehelper.specfile, 42
- rebasehelper.tags, 45
- rebasehelper.temporary\_environment, 46



## Symbols

- apply-changes
  - rebase-helper command line option, 16
- background {dark,light,auto}
  - rebase-helper command line option, 15
- bugzilla-id BUG\_ID
  - rebase-helper command line option, 16
- build-log-hook-blacklist [{files}]
  - rebase-helper command line option, 16
- build-tasks OLD\_TASK,NEW\_TASK
  - rebase-helper command line option, 16
- builder-options BUILDER\_OPTIONS
  - rebase-helper command line option, 16
- builds-nowait
  - rebase-helper command line option, 16
- buildtool {copr,koji,mock,rpmbuild}
  - rebase-helper command line option, 15
- changelog-entry CHANGELOG\_ENTRY
  - rebase-helper command line option, 16
- color {always,never,auto}
  - rebase-helper command line option, 15
- config-file CONFIG\_FILE
  - rebase-helper command line option, 16
- copr-chroots COPR\_CHROOTS
  - rebase-helper command line option, 17
- copr-project-frontpage
  - rebase-helper command line option, 17
- copr-project-permanent
  - rebase-helper command line option, 16
- disable-inapplicable-patches
  - rebase-helper command line option, 16
- favor-on-conflict
  - {downstream,upstream,off}
  - rebase-helper command line option, 16
- force-build-log-hooks
  - rebase-helper command line option, 16
- non-interactive
  - rebase-helper command line option, 16
- not-download-sources
  - rebase-helper command line option, 16
- outputtool {json,text}
  - rebase-helper command line option, 15
- pkgcomparetool [{abipkgdiff,csmock,licensecheck,pl}]
  - rebase-helper command line option, 15
- replace-old-version-with-macro
  - rebase-helper command line option, 17
- results-dir RESULTS\_DIR
  - rebase-helper command line option, 15
- skip-upload
  - rebase-helper command line option, 16
- skip-version-check
  - rebase-helper command line option, 16
- spec-hook-blacklist
  - {commit-hash-updater,escape-macros,paths-to}

rebase-helper command line option, 16

-srpm-builder-options  
SRPM\_BUILDER\_OPTIONS  
rebase-helper command line option, 16

-srpm-buildtool {mock,rpmbuild}  
rebase-helper command line option, 15

-update-sources  
rebase-helper command line option, 16

-version  
rebase-helper command line option, 15

-versioneer {anitya,cpan,hackage,npmjs,pypi,subversion}  
rebase-helper command line option, 15

-versioneer-blacklist  
[{anitya,cpan,hackage,npmjs,pypi,subversion}]  
rebase-helper command line option, 15

-workspace-dir WORKSPACE\_DIR  
rebase-helper command line option, 15

-D 'MACRO EXPR', -define 'MACRO EXPR'  
rebase-helper command line option, 16

-h, -help  
rebase-helper command line option, 15

-v, -verbose  
rebase-helper command line option, 15

-w, -keep-workspace  
rebase-helper command line option, 16

## A

ACCEPTS\_OPTIONS (rebase-helper.plugins.build\_tools.rpm.BuildToolBase attribute), 33

add\_file\_handler() (rebase-helper.logger.LoggerHelper static method), 40

add\_stream\_handler() (rebase-helper.logger.LoggerHelper static method), 40

all\_packages\_installed() (rebase-helper.helpers.rpm\_helper.RpmHelper static method), 27

Application (class in rebasehelper.application), 36

apply\_changes() (rebase-helper.application.Application method),

36

apply\_old\_patches() (rebase-helper.patcher.Patcher class method), 41

apply\_patch() (rebasehelper.patcher.Patcher class method), 41

ARCHES (rebasehelper.helpers.rpm\_helper.RpmHelper attribute), 27

Archive (class in rebasehelper.archive), 37

ArchiveTypeBase (class in rebasehelper.archive), 37

## B

BaseBuildLogHook (class in rebase-helper.plugins.build\_log\_hooks), 28

BaseChecker (class in rebasehelper.plugins.checkers), 30

BaseOutputTool (class in rebase-helper.plugins.output\_tools), 31

BaseSpecHook (class in rebase-helper.plugins.spec\_hooks), 34

BaseVersioneer (class in rebase-helper.plugins.versioners), 35

BinaryPackageBuildError, 39

BUGZILLA\_REST\_API\_URL (rebase-helper.helpers.bugzilla\_helper.BugzillaHelper attribute), 18

BugzillaHelper (class in rebase-helper.helpers.bugzilla\_helper), 18

build() (rebasehelper.helpers.copr\_helper.CoprHelper class method), 20

build() (rebasehelper.plugins.build\_tools.rpm.BuildToolBase class method), 33

build() (rebasehelper.plugins.build\_tools.srpm.SRPMBuildToolBase class method), 35

build\_binary\_packages() (rebase-helper.application.Application method), 36

build\_parser() (rebasehelper.cli.CLI static method), 39

build\_source\_packages() (rebase-helper.application.Application method), 36

BuildLogHookCollection (class in rebase-helper.plugins.build\_log\_hooks), 29

BuildTemporaryEnvironment (class in rebase-helper.plugins.build\_tools), 29

BuildToolBase (class in rebase-helper.plugins.build\_tools.rpm), 33

BuildToolCollection (class in rebase-helper.plugins.build\_tools.rpm), 34

Bz2ArchiveType (class in rebasehelper.archive), 37

## C

CATEGORIES (rebase-helper.plugins.build\_log\_hooks.BaseBuildLogHook

- attribute), 28
- CATEGORIES (rebase-helper.plugins.spec\_hooks.BaseSpecHook attribute), 34
- CATEGORIES (rebase-helper.plugins.versioners.BaseVersioner attribute), 35
- CATEGORY (rebasehelper.plugins.checkers.BaseChecker attribute), 30
- check\_mock\_privileges() (in module rebase-helper.plugins.build\_tools), 30
- CheckerCategory (class in rebase-helper.plugins.checkers), 31
- CheckerCollection (class in rebase-helper.plugins.checkers), 31
- CheckerNotFoundError, 39
- CLI (class in rebasehelper.cli), 39
- CliHelper (class in rebasehelper.cli), 39
- clone\_repository() (rebase-helper.helpers.bugzilla\_helper.BugzillaHelper class method), 18
- close() (rebasehelper.archive.GemArchiveType.GemArchive method), 37
- COLLECTIONS (rebase-helper.plugins.plugin\_manager.PluginManager attribute), 33
- color\_is\_light() (rebase-helper.helpers.console\_helper.ConsoleHelper static method), 19
- ColorizingStreamHandler (class in rebase-helper.logger), 39
- colors (rebasehelper.logger.ColorizingStreamHandler attribute), 39
- ConsoleHelper (class in rebase-helper.helpers.console\_helper), 19
- ConsoleHelper.Capturer (class in rebase-helper.helpers.console\_helper), 19
- convert\_macros\_to\_dict() (rebase-helper.cli.CliHelper class method), 39
- CoprHelper (class in rebase-helper.helpers.copr\_helper), 20
- copy() (rebasehelper.specfile.SpecFile method), 43
- cprint() (rebasehelper.helpers.console\_helper.ConsoleHelper class method), 19
- CrateArchiveType (class in rebasehelper.archive), 37
- create\_file\_handlers() (rebase-helper.logger.LoggerHelper class method), 40
- create\_project() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 20
- create\_session() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 22
- create\_stream\_handlers() (rebase-helper.logger.LoggerHelper class method), 40
- CREATES\_TASKS (rebase-helper.plugins.build\_tools.rpm.BuildToolBase attribute), 33
- CustomLogger (class in rebasehelper.logger), 40
- ## D
- decorate\_patch\_name() (rebase-helper.patcher.Patcher static method), 41
- DEFAULT (rebasehelper.plugins.build\_tools.rpm.BuildToolBase attribute), 33
- DEFAULT (rebasehelper.plugins.build\_tools.srpm.SRPMBuildToolBase attribute), 34, 35
- DEFAULT (rebasehelper.plugins.checkers.BaseChecker attribute), 30
- DEFAULT (rebasehelper.plugins.output\_tools.BaseOutputTool attribute), 31
- DELETE\_PROJECT\_AFTER (rebase-helper.helpers.copr\_helper.CoprHelper attribute), 20
- detect\_background() (rebase-helper.helpers.console\_helper.ConsoleHelper static method), 20
- DEV\_NULL (rebasehelper.helpers.process\_helper.ProcessHelper attribute), 26
- DISALLOW\_INLINE\_COMMENTS (rebase-helper.spec\_content.SpecContent attribute), 42
- display\_task\_results() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 22
- DIST\_GIT\_REPO\_URL (rebase-helper.helpers.bugzilla\_helper.BugzillaHelper attribute), 18
- download() (rebase-helper.helpers.lookaside\_cache\_helper.LookasideCacheHelper class method), 24
- download\_build() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 20
- download\_build() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 22
- download\_file() (rebase-helper.helpers.download\_helper.DownloadHelper static method), 21
- download\_remote\_sources() (rebase-helper.specfile.SpecFile method), 43
- download\_task\_results() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 22

- DownloadError, 39
- DownloadHelper (class in rebase-helper.helpers.download\_helper), 21
- dump() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 24
- ## E
- emit() (rebasehelper.logger.ColorizingStreamHandler method), 39
- env() (rebasehelper.temporary\_environment.TemporaryEnvironment method), 46
- environment variable  
PACKAGE, 14  
REPOSITORY, 14
- exchange\_control\_sequence() (rebase-helper.helpers.console\_helper.ConsoleHelper static method), 20
- expand() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 24
- expand\_macros() (rebase-helper.helpers.macro\_helper.MacroHelper class method), 24
- EXTENSION (rebasehelper.archive.ArchiveTypeBase attribute), 37
- EXTENSION (rebasehelper.archive.Bz2ArchiveType attribute), 37
- EXTENSION (rebasehelper.archive.CrateArchiveType attribute), 37
- EXTENSION (rebasehelper.archive.GemArchiveType attribute), 37
- EXTENSION (rebasehelper.archive.TarArchiveType attribute), 37
- EXTENSION (rebasehelper.archive.TarBz2ArchiveType attribute), 38
- EXTENSION (rebasehelper.archive.TarGzArchiveType attribute), 38
- EXTENSION (rebasehelper.archive.TarXzArchiveType attribute), 38
- EXTENSION (rebasehelper.archive.TgzArchiveType attribute), 38
- EXTENSION (rebasehelper.archive.ZipArchiveType attribute), 38
- EXTENSION (rebasehelper.plugins.output\_tools.BaseOutputTool attribute), 31
- extract() (rebasehelper.archive.ArchiveTypeBase class method), 37
- extract() (rebasehelper.archive.GemArchiveType class method), 37
- extract() (rebasehelper.archive.GemArchiveType.GemArchive class method), 37
- extract() (rebasehelper.archive.TarBz2ArchiveType class method), 38
- extract() (rebasehelper.archive.TarGzArchiveType class method), 38
- extract() (rebasehelper.archive.TarXzArchiveType class method), 38
- extract() (rebasehelper.archive.ZipArchiveType class method), 38
- extract\_archive() (rebase-helper.application.Application static method), 36
- extract\_archive() (rebasehelper.archive.Archive method), 37
- extract\_patch\_name() (rebase-helper.patcher.Patcher class method), 41
- extract\_sources() (rebase-helper.application.Application static method), 36
- extract\_version\_from\_archive\_name() (rebasehelper.specfile.SpecFile static method), 43
- ## F
- file\_available() (rebase-helper.helpers.path\_helper.PathHelper static method), 25
- filter() (rebasehelper.helpers.macro\_helper.MacroHelper static method), 24
- filter() (rebasehelper.tags.Tags method), 45
- find\_all\_files() (rebase-helper.helpers.path\_helper.PathHelper static method), 25
- find\_all\_files\_current\_dir() (rebase-helper.helpers.path\_helper.PathHelper static method), 25
- find\_archive\_target\_in\_prep() (rebase-helper.specfile.SpecFile method), 43
- find\_first\_dir\_with\_file() (rebase-helper.helpers.path\_helper.PathHelper static method), 25
- find\_first\_file() (rebase-helper.helpers.path\_helper.PathHelper static method), 25
- format() (rebasehelper.plugins.build\_log\_hooks.BaseBuildLogHook class method), 28
- format() (rebasehelper.plugins.checkers.BaseChecker class method), 30
- functional (rebase-helper.helpers.koji\_helper.KojiHelper attribute), 23
- ## G
- GemArchiveType (class in rebasehelper.archive), 37
- GemArchiveType.GemArchive (class in rebase-helper.archive), 37
- generate\_patch() (rebase-helper.application.Application method), 36



get\_all\_plugins() (rebase-helper.plugins.plugin\_collection.PluginCollection method), 30  
 get\_applied\_patches() (rebase-helper.specfile.SpecFile method), 43  
 get\_arches() (rebase-helper.helpers.rpm\_helper.RpmHelper static method), 27  
 get\_archive() (rebasehelper.specfile.SpecFile method), 43  
 get\_bugzilla\_component() (rebase-helper.helpers.bugzilla\_helper.BugzillaHelper class method), 18  
 get\_build() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 23  
 get\_build\_status() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 20  
 get\_build\_url() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 20  
 get\_builder\_options() (rebase-helper.plugins.build\_tools.rpm.BuildToolBase static method), 33  
 get\_category() (rebase-helper.plugins.checkers.BaseChecker class method), 30  
 get\_checker\_output\_dir\_short() (rebase-helper.plugins.checkers.BaseChecker class method), 30  
 get\_client() (rebase-helper.helpers.copr\_helper.CoprHelper class method), 20  
 get\_comment\_span() (rebase-helper.spec\_content.SpecContent class method), 42  
 get\_default\_plugins() (rebase-helper.plugins.checkers.CheckerCollection method), 31  
 get\_default\_plugins() (rebase-helper.plugins.plugin\_collection.PluginCollection method), 32  
 get\_detached\_task() (rebase-helper.plugins.build\_tools.rpm.BuildToolBase class method), 33  
 get\_email() (rebase-helper.helpers.git\_helper.GitHelper class method), 21  
 get\_header\_from\_rpm() (rebase-helper.helpers.rpm\_helper.RpmHelper static method), 28  
 get\_important\_changes() (rebase-helper.plugins.checkers.BaseChecker class method), 30  
 get\_latest\_build() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 23  
 get\_main\_files\_section() (rebase-helper.specfile.SpecFile method), 43  
 get\_main\_source() (rebasehelper.specfile.SpecFile method), 43  
 get\_message() (rebase-helper.helpers.input\_helper.InputHelper static method), 21  
 get\_mock\_logfile\_path() (in module rebase-helper.plugins.build\_tools), 30  
 get\_new\_build\_logs() (rebase-helper.application.Application method), 36  
 get\_new\_log() (rebasehelper.specfile.SpecFile method), 43  
 get\_not\_used\_patches() (rebase-helper.specfile.SpecFile method), 44  
 get\_NVR() (rebasehelper.specfile.SpecFile method), 43  
 get\_old\_build\_info() (rebase-helper.helpers.koji\_helper.KojiHelper class method), 23  
 get\_options() (rebase-helper.plugins.plugin\_collection.PluginCollection method), 32  
 get\_options() (rebase-helper.plugins.plugin\_manager.PluginManager method), 33  
 get\_patch\_name() (rebase-helper.specfile.PatchObject method), 43  
 get\_patches() (rebasehelper.specfile.SpecFile method), 44  
 get\_plugin() (rebase-helper.plugins.plugin\_collection.PluginCollection method), 32  
 get\_prep\_section() (rebase-helper.specfile.SpecFile method), 44  
 get\_raw\_tag\_value() (rebase-helper.specfile.SpecFile method), 44  
 get\_rebase\_name() (in module rebase-helper.specfile), 45  
 get\_release() (rebasehelper.specfile.SpecFile method), 44  
 get\_report\_path() (rebase-helper.plugins.output\_tools.BaseOutputTool class method), 31  
 get\_rpm\_spec() (rebase-helper.helpers.rpm\_helper.RpmHelper class method), 28  
 get\_setup\_dirname() (rebase-helper.specfile.SpecFile method), 44  
 get\_sources() (rebasehelper.specfile.SpecFile

- `method`), 44
- `get_srpm_builder_options()` (*rebase-helper.plugins.build\_tools.srpm.SRPMBuildToolBase static method*), 35
- `get_subpackage_name()` (*rebase-helper.specfile.SpecFile static method*), 44
- `get_supported_archives()` (*rebase-helper.archive.Archive class method*), 37
- `get_supported_plugins()` (*rebase-helper.plugins.checkers.CheckerCollection method*), 31
- `get_supported_plugins()` (*rebase-helper.plugins.plugin\_collection.PluginCollection method*), 32
- `get_task_info()` (*rebase-helper.plugins.build\_tools.rpm.BuildToolBase class method*), 33
- `get_task_url()` (*rebase-helper.helpers.koji\_helper.KojiHelper class method*), 23
- `get_temp_dir()` (*rebase-helper.helpers.path\_helper.PathHelper static method*), 26
- `get_underlined_title()` (*rebase-helper.plugins.checkers.BaseChecker class method*), 31
- `get_user()` (*rebase-helper.helpers.git\_helper.GitHelper class method*), 21
- `get_version()` (*rebasehelper.specfile.SpecFile method*), 44
- `get_version_from_comments()` (*rebase-helper.helpers.bugzilla\_helper.BugzillaHelper class method*), 19
- `GIT_USER_EMAIL` (*rebase-helper.helpers.git\_helper.GitHelper attribute*), 21
- `GIT_USER_NAME` (*rebase-helper.helpers.git\_helper.GitHelper attribute*), 21
- `GitHelper` (*class in rebasehelper.helpers.git\_helper*), 21
- H**
- `haskell` (*rebasehelper.specfile.PackageCategory attribute*), 42
- `HEADING` (*rebasehelper.logger.CustomLogger attribute*), 40
- I**
- `IMPORTANT` (*rebasehelper.logger.CustomLogger attribute*), 40
- `init_git()` (*rebasehelper.patcher.Patcher class method*), 41
- `InputHelper` (*class in rebase-helper.helpers.input\_helper*), 21
- `insert_patch_name()` (*rebase-helper.patcher.Patcher class method*), 41
- `install_build_dependencies()` (*rebase-helper.helpers.rpm\_helper.RpmHelper static method*), 28
- `is_available()` (*rebase-helper.plugins.checkers.BaseChecker class method*), 31
- `is_package_installed()` (*rebase-helper.helpers.rpm\_helper.RpmHelper static method*), 28
- `is_test_suite_enabled()` (*rebase-helper.specfile.SpecFile method*), 44
- K**
- `KojiHelper` (*class in rebase-helper.helpers.koji\_helper*), 22
- L**
- `LoggerHelper` (*class in rebasehelper.logger*), 40
- `LookasideCacheError`, 39
- `LookasideCacheHelper` (*class in rebase-helper.helpers.lookaside\_cache\_helper*), 24
- M**
- `MacroHelper` (*class in rebase-helper.helpers.macro\_helper*), 24
- `MACROS_WHITELIST` (*rebase-helper.helpers.macro\_helper.MacroHelper attribute*), 24
- `match()` (*rebasehelper.archive.ArchiveTypeBase class method*), 37
- `match()` (*rebasehelper.archive.ZipArchiveType class method*), 38
- `memory_handler` (*rebasehelper.logger.LoggerHelper attribute*), 40
- `MemoryHandler` (*class in rebasehelper.logger*), 41
- `merge_two_results()` (*rebase-helper.plugins.build\_log\_hooks.BaseBuildLogHook class method*), 28
- `MockTemporaryEnvironment` (*class in rebase-helper.plugins.build\_tools*), 29
- N**
- `new_repo` (*rebasehelper.patcher.Patcher attribute*), 41
- `new_sources` (*rebasehelper.patcher.Patcher attribute*), 41
- `nodejs` (*rebasehelper.specfile.PackageCategory attribute*), 42
- `non_interactive` (*rebasehelper.patcher.Patcher attribute*), 41

## O

old\_repo (*rebasehelper.patcher.Patcher* attribute), 41  
 old\_sources (*rebasehelper.patcher.Patcher* attribute), 41  
 open () (*rebasehelper.archive.ArchiveTypeBase* class method), 37  
 open () (*rebasehelper.archive.GemArchiveType* class method), 37  
 open () (*rebasehelper.archive.TarBz2ArchiveType* class method), 38  
 open () (*rebasehelper.archive.TarGzArchiveType* class method), 38  
 open () (*rebasehelper.archive.TarXzArchiveType* class method), 38  
 open () (*rebasehelper.archive.ZipArchiveType* class method), 38  
 output\_data (*rebasehelper.patcher.Patcher* attribute), 41  
 OutputToolCollection (class in *rebase-helper.plugins.output\_tools*), 32

## P

PACKAGE, 14  
 PackageCategory (class in *rebasehelper.specfile*), 42  
 parse\_release () (*rebasehelper.specfile.SpecFile* method), 44  
 parse\_rgb\_device\_specification () (*rebase-helper.helpers.console\_helper.ConsoleHelper* static method), 20  
 parse\_spec () (*rebase-helper.helpers.rpm\_helper.RpmHelper* class method), 28  
 ParseError, 39  
 patch () (*rebasehelper.patcher.Patcher* class method), 41  
 patch\_sources () (*rebase-helper.application.Application* method), 36  
 Patcher (class in *rebasehelper.patcher*), 41  
 patches (*rebasehelper.patcher.Patcher* attribute), 41  
 PatchList (class in *rebasehelper.specfile*), 42  
 PatchObject (class in *rebasehelper.specfile*), 42  
 path () (*rebasehelper.temporary\_environment.TemporaryEnvironment* method), 46  
 PathHelper (class in *rebase-helper.helpers.path\_helper*), 25  
 perl (*rebasehelper.specfile.PackageCategory* attribute), 42  
 php (*rebasehelper.specfile.PackageCategory* attribute), 42  
 PluginCollection (class in *rebase-helper.plugins.plugin\_collection*), 32  
 PluginManager (class in *rebase-helper.plugins.plugin\_manager*), 33  
 prepare () (*rebasehelper.plugins.build\_tools.rpm.BuildToolBase* class method), 34  
 prepare\_next\_run () (*rebase-helper.application.Application* method), 36  
 prepare\_rebase\_repository () (*rebase-helper.helpers.bugzilla\_helper.BugzillaHelper* class method), 19  
 prepare\_results\_dir () (*rebase-helper.plugins.checkers.BaseChecker* class method), 31  
 prepare\_sources () (*rebase-helper.application.Application* method), 36  
 prepend\_results\_dir\_name () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 31  
 print\_cli\_summary () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 32  
 print\_important\_checkers\_output () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 32  
 print\_patches\_cli () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 32  
 print\_patches\_section\_cli () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 32  
 print\_report\_file\_path () (*rebase-helper.plugins.output\_tools.BaseOutputTool* class method), 32  
 print\_summary () (*rebase-helper.application.Application* method), 36  
 print\_task\_info () (*rebase-helper.application.Application* method), 36  
 ProcessHelper (class in *rebase-helper.helpers.process\_helper*), 26  
 progress () (*rebase-helper.helpers.download\_helper.DownloadHelper* static method), 21  
 purge\_macro () (*rebase-helper.helpers.macro\_helper.MacroHelper* class method), 24  
 python (*rebasehelper.specfile.PackageCategory* attribute), 42

## R

R (*rebasehelper.specfile.PackageCategory* attribute), 42  
 rebase-helper command line option  
 -apply-changes, 16  
 -background {dark, light, auto}, 15

-bugzilla-id BUG\_ID, 16  
 -build-log-hook-blacklist  
   [*{files}*], 16  
 -build-tasks OLD\_TASK, NEW\_TASK, 16  
 -builder-options BUILDER\_OPTIONS, 16  
 -builds-nowait, 16  
 -buildtool *{copr, koji, mock, rpmbuild}*, 15  
 -changelog-entry CHANGELOG\_ENTRY, 16  
 -color *{always, never, auto}*, 15  
 -config-file CONFIG\_FILE, 16  
 -copr-chroots COPR\_CHROOTS, 17  
 -copr-project-frontpage, 17  
 -copr-project-permanent, 16  
 -disable-inapplicable-patches, 16  
 -favor-on-conflict  
   *{downstream, upstream, off}*, 16  
 -force-build-log-hooks, 16  
 -non-interactive, 16  
 -not-download-sources, 16  
 -outputtool *{json, text}*, 15  
 -pkgcomparetool  
   [*{abipkgdiff, csmock, licensecheck, rpm, rpm-inspect-srpm, sonar}*], 15  
 -replace-old-version-with-macro, 17  
 -results-dir RESULTS\_DIR, 15  
 -skip-upload, 16  
 -skip-version-check, 16  
 -spec-hook-blacklist  
   [*{commit-hash-updater, escape-macros, path-macros, pypi-url-fix, replace-old-version}*], 16  
 -srpm-builder-options  
   SRPM\_BUILDER\_OPTIONS, 16  
 -srpm-buildtool *{mock, rpmbuild}*, 15  
 -update-sources, 16  
 -version, 15  
 -versioneer *{anitya, cpan, hackage, npmjs, pypi, rubygems}*, 15  
 -versioneer-blacklist  
   [*{anitya, cpan, hackage, npmjs, pypi, rubygems}*], 15  
 -workspace-dir WORKSPACE\_DIR, 15  
 -D 'MACRO EXPR', -define 'MACRO EXPR', 16  
 -h, -help, 15  
 -v, -verbose, 15  
 -w, -keep-workspace, 16  
 SOURCES, 15  
 rebasehelper.application (*module*), 36  
 rebasehelper.archive (*module*), 37  
 rebasehelper.cli (*module*), 39  
 rebasehelper.exceptions (*module*), 39  
 rebasehelper.helpers.bugzilla\_helper  
   (*module*), 18  
   rebasehelper.helpers.console\_helper  
     (*module*), 19  
   rebasehelper.helpers.copr\_helper (*module*), 20  
   rebasehelper.helpers.download\_helper  
     (*module*), 21  
   rebasehelper.helpers.git\_helper (*module*), 21  
   rebasehelper.helpers.input\_helper (*module*), 21  
   rebasehelper.helpers.koji\_helper (*module*), 22  
   rebasehelper.helpers.lookaside\_cache\_helper  
     (*module*), 24  
   rebasehelper.helpers.macro\_helper (*module*), 24  
   rebasehelper.helpers.path\_helper (*module*), 25  
   rebasehelper.helpers.process\_helper  
     (*module*), 26  
   rebasehelper.helpers.rpm\_helper (*module*), 27  
   rebasehelper.helpers.srpm\_helper (*module*), 27  
   rebasehelper.helpers.specfile\_helper (*module*), 29  
   rebasehelper.patcher (*module*), 41  
   rebasehelper.plugins.build\_log\_hooks  
     (*module*), 28  
   rebasehelper.plugins.build\_tools (*module*), 29  
     rebasehelper.plugins.build\_tools.rpm  
       (*module*), 33  
     rebasehelper.plugins.build\_tools.srpm  
       (*module*), 34  
   rebasehelper.plugins.checkers (*module*), 30  
   rebasehelper.plugins.output\_tools (*module*), 31  
   rebasehelper.plugins.plugin\_collection  
     (*module*), 33  
   rebasehelper.plugins.plugin\_manager  
     (*module*), 33  
   rebasehelper.plugins.spec\_hooks (*module*), 34  
   rebasehelper.plugins.versioners (*module*), 35  
   rebasehelper.spec\_content (*module*), 42  
   rebasehelper.specfile (*module*), 42  
   rebasehelper.tags (*module*), 45  
   rebasehelper.temporary\_environment (*module*), 46  
   RebaseHelperError, 39  
   register\_archive\_type() (*in module rebase-helper.archive*), 38  
   reload() (*rebasehelper.specfile.SpecFile method*), 44  
   remove\_file\_handlers() (*rebase-helper.logger.LoggerHelper class method*),

40  
remove\_memory\_handler() (rebase-helper.logger.LoggerHelper class method), 41  
replace\_section() (rebase-helper.spec\_content.SpecContent method), 42  
replay\_into() (rebase-helper.logger.MemoryHandler method), 41  
REPOSITORY, 14  
request() (rebasehelper.helpers.download\_helper.DownloadHelper static method), 21  
results\_dir (rebase-helper.plugins.checkers.BaseChecker attribute), 30, 31  
rpkg\_config\_dir (rebase-helper.helpers.lookaside\_cache\_helper.LookasideCacheHelper attribute), 24  
RPM (rebasehelper.plugins.checkers.CheckerCategory attribute), 31  
RpmbuildTemporaryEnvironment (class in rebasehelper.plugins.build\_tools), 29  
RpmHeader (class in rebasehelper.helpers.rpm\_helper), 27  
RpmHelper (class in rebasehelper.helpers.rpm\_helper), 27  
ruby (rebasehelper.specfile.PackageCategory attribute), 42  
run() (rebasehelper.application.Application method), 36  
run() (rebasehelper.cli.CliHelper class method), 39  
run() (rebasehelper.plugins.build\_log\_hooks.BaseBuildLogHook class method), 28  
run() (rebasehelper.plugins.build\_log\_hooks.BuildLogHookCollection method), 29  
run() (rebasehelper.plugins.checkers.CheckerCollection method), 31  
run() (rebasehelper.plugins.output\_tools.BaseOutputTool class method), 32  
run() (rebasehelper.plugins.output\_tools.OutputToolCollection method), 32  
run() (rebasehelper.plugins.spec\_hooks.BaseSpecHook class method), 34  
run() (rebasehelper.plugins.spec\_hooks.SpecHookCollection method), 34  
run() (rebasehelper.plugins.versioners.BaseVersioneer class method), 35  
run() (rebasehelper.plugins.versioners.VersioneerCollection method), 35  
run\_check() (rebase-helper.plugins.checkers.BaseChecker class method), 31  
run\_mergetool() (rebase-helper.helpers.git\_helper.GitHelper class method), 21  
run\_package\_checkers() (rebase-helper.application.Application method), 36  
run\_subprocess() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 26  
run\_subprocess\_cwd() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 26  
run\_subprocess\_cwd\_env() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 26  
run\_subprocess\_env() (rebase-helper.helpers.process\_helper.ProcessHelper static method), 27  
run\_subprocess\_env\_basehelper\_specfile\_package\_category\_attribute() (rebasehelper.specfile.PackageCategory attribute), 42  
**S**  
save() (rebasehelper.specfile.SpecFile method), 44  
saves() (in module rebasehelper.specfile), 45  
section() (rebasehelper.spec\_content.SpecContent method), 42  
SECTION\_HEADERS (rebase-helper.spec\_content.SpecContent attribute), 42  
set\_extra\_version() (rebase-helper.specfile.SpecFile method), 44  
set\_raw\_tag\_value() (rebase-helper.specfile.SpecFile method), 44  
set\_release() (rebasehelper.specfile.SpecFile method), 45  
set\_base\_number() (rebase-helper.specfile.SpecFile method), 45  
set\_tag() (rebasehelper.specfile.SpecFile method), 45  
set\_terminal\_background() (rebase-helper.logger.ColorizingStreamHandler method), 40  
set\_version() (rebasehelper.specfile.SpecFile method), 45  
setup() (rebasehelper.application.Application static method), 36  
setup\_memory\_handler() (rebase-helper.logger.LoggerHelper class method), 41  
should\_use\_colors() (rebase-helper.helpers.console\_helper.ConsoleHelper class method), 20  
shouldFlush() (rebase-helper.logger.MemoryHandler method), 41



SOURCE (*rebasehelper.plugins.checkers.CheckerCategory* attribute), 31

SourcePackageBuildError, 39

SOURCES  
     rebase-helper command line option, 15

SpecContent (*class in rebasehelper.spec\_content*), 42

SpecFile (*class in rebasehelper.specfile*), 43

SpecHookCollection (*class in rebase-helper.plugins.spec\_hooks*), 34

split\_nevra() (*rebase-helper.helpers.rpm\_helper.RpmHelper* class method), 28

split\_version\_string() (*rebase-helper.specfile.SpecFile* static method), 45

SRPM (*rebasehelper.plugins.checkers.CheckerCategory* attribute), 31

SRPMBuildToolBase (*class in rebase-helper.plugins.build\_tools.srpm*), 34

SRPMBuildToolCollection (*class in rebase-helper.plugins.build\_tools.srpm*), 35

strip\_patch\_name() (*rebasehelper.patcher.Patcher* class method), 41

substitute\_path\_with\_macros() (*rebase-helper.helpers.macro\_helper.MacroHelper* static method), 24

SUCCESS (*rebasehelper.logger.CustomLogger* attribute), 40

## T

Tag (*class in rebasehelper.tags*), 45

tag() (*rebasehelper.specfile.SpecFile* method), 45

Tags (*class in rebasehelper.tags*), 45

TarArchiveType (*class in rebasehelper.archive*), 37

TarBz2ArchiveType (*class in rebasehelper.archive*), 37

TarGzArchiveType (*class in rebasehelper.archive*), 38

TarXzArchiveType (*class in rebasehelper.archive*), 38

TEMPDIR (*rebasehelper.temporary\_environment.TemporaryEnvironment* attribute), 46

TEMPDIR\_BUILD (*rebase-helper.plugins.build\_tools.RpmbuildTemporaryEnvironment* attribute), 29

TEMPDIR\_BUILDROOT (*rebase-helper.plugins.build\_tools.RpmbuildTemporaryEnvironment* attribute), 30

TEMPDIR\_RESULTS (*rebase-helper.plugins.build\_tools.BuildTemporaryEnvironment* attribute), 29

TEMPDIR\_RPMBUILD (*rebase-helper.plugins.build\_tools.RpmbuildTemporaryEnvironment* attribute), 30

TEMPDIR\_RPMS (*rebase-helper.plugins.build\_tools.RpmbuildTemporaryEnvironment* attribute), 30

TEMPDIR\_SOURCES (*rebase-helper.plugins.build\_tools.BuildTemporaryEnvironment* attribute), 29

TEMPDIR\_SPEC (*rebase-helper.plugins.build\_tools.BuildTemporaryEnvironment* attribute), 29

TEMPDIR\_SPECS (*rebase-helper.plugins.build\_tools.BuildTemporaryEnvironment* attribute), 29

TEMPDIR\_SRPMS (*rebase-helper.plugins.build\_tools.RpmbuildTemporaryEnvironment* attribute), 30

TemporaryEnvironment (*class in rebase-helper.temporary\_environment*), 46

terminal\_background (*rebase-helper.logger.ColorizingStreamHandler* attribute), 40

TgzArchiveType (*class in rebasehelper.archive*), 38

TRACE (*rebasehelper.logger.CustomLogger* attribute), 40

## U

update() (*rebasehelper.specfile.SpecFile* method), 45

update\_changelog() (*rebase-helper.specfile.SpecFile* method), 45

update\_paths\_to\_sources\_and\_patches() (*rebasehelper.specfile.SpecFile* method), 45

update\_setup\_dirname() (*rebase-helper.specfile.SpecFile* method), 45

update\_sources() (*rebase-helper.helpers.lookaside\_cache\_helper.LookasideCacheHelper* class method), 24

upload\_srpm() (*rebase-helper.helpers.koji\_helper.KojiHelper* class method), 23

UPSTREAM\_RELEASE\_MONITORING\_USERNAME (*rebasehelper.helpers.bugzilla\_helper.BugzillaHelper* attribute), 18

## V

VERBOSE (*rebasehelper.logger.CustomLogger* attribute), 40

VersioneerCollection (*class in rebase-helper.plugins.versioners*), 35

## W

wait\_for\_task() (*rebase-helper.plugins.build\_tools.rpm.BuildToolBase* class method), 34

`watch_build()` (*rebase-helper.helpers.copr\_helper.CoprHelper* class method), 20

`watch_koji_tasks()` (*rebase-helper.helpers.koji\_helper.KojiHelper* class method), 23

`write_updated_patches()` (*rebase-helper.specfile.SpecFile* method), 45

## Z

`ZipArchiveType` (*class in rebasehelper.archive*), 38