

---

# Rebase Helper Documentation

*Release 0.10.1*

**Petr Hracek**

**Tomas Hozza**

Sep 11, 2017



---

## Contents

---

<b>1</b>	<b>General workflow</b>	<b>3</b>
<b>2</b>	<b>Patch rebasing workflow</b>	<b>5</b>
<b>3</b>	<b>How to run rebase-helper</b>	<b>7</b>
3.1	User Guide . . . . .	7
3.2	API . . . . .	11
3.3	Change Log . . . . .	30
	<b>Python Module Index</b>	<b>37</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.



---

## General workflow

---

- *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
- old and new source tarballs are downloaded and extracted to *rebase-helper-workspace* directory
- downstream patches are rebased on top of new sources using `git-rebase`, resulting modified patches are saved to *rebase-helper-results/rebased-sources* directory
- old and new source RPMs are created and built with selected build tool
- multiple checker tools are run against both sets of packages and their output is stored in *rebase-helper-results/checkers* directory
- *rebase-helper-workspace* directory is removed





---

### Patch rebasing workflow

---

- 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*



---

## How to run rebase-helper

---

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

There are two ways how to specify the new version. You can pass it to **rebase-helper** directly, e.g.:

```
rebase-helper 3.1.10
```

or you can let *rebase-helper* determine it from the new version tarball, e.g.:

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

Starting with **rebase-helper 0.10.0** you don't have to specify the new version at all, and **rebase-helper** will attempt to determine it automatically using one of available *versioneers*.

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

## User Guide

### 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 *install\_requires* argument of *setup* 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

## Usage

### SYNOPSIS

```
rebase-helper [-h] [--version] [-v]
  [-b | --comparepkgs-only COMPAREPKGS_DIR | -p]
  [--buildtool {copr, rpmbuild, mock}]
  [--pkgcomparetool {rpmdiff, abipkgdiff, pkgdiff, csmock}]
  [--outputtool {text, json}] [--versioneer {pypi, anitya}]
  [--not-download-sources] [-w] [-c] [--color {always, never, auto}]
  [--disable-inapplicable-patches] [--non-interactive]
  [--build-tasks OLD_TASK, NEW_TASK] [--builds-nowait]
  [--builder-options=UILDER_OPTIONS] [--build-retries BUILD_RETRIES]
  [--results-dir RESULTS_DIR] [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

new upstream sources

#### Optional arguments

##### -h, --help

show help message and exit

##### --version

show rebase-helper version and exit

##### -v, --verbose

be more verbose (recommended)

##### -b, --build-only

only build SRPM and RPMs

##### --comparepkgs-only COMPAREPKGS\_DIR

compare already built packages, COMPAREPKGS\_DIR must be a directory with the following structure:  
<dir\_name>/{old,new}/RPM

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

only apply patches

**--buildtool** {copr, rpmbuild, mock}  
build tool to use, defaults to mock

**--pkgcomparetool** {rpmdiff, abipkgdiff, pkgdiff, csmock}  
set of tools to use for package comparison, defaults to rpmdiff, abipkgdiff, pkgdiff

**--outputtool** {text, json}  
tool to use for formatting rebase output, defaults to text

**--versioneer** {pypi, anitya}  
tool to use for determining latest upstream version, defaults to anitya

**--not-download-sources**  
do not download sources

**-w, --keep-workspace**  
do not remove workspace directory after finishing

**-c, --continue**  
continue previously interrupted rebase

**--color** {always, never, auto}  
colorize the output, defaults to auto

**--disable-inapplicable-patches**  
disable inapplicable patches in rebased SPEC file

**--non-interactive**  
do not interact with user

**--build-tasks** OLD\_TASK, NEW\_TASK  
comma-separated remote build task ids

**--builds-nowait**  
do not wait for remote builds to finish

**--builder-options=**BUILDER\_OPTIONS  
enable arbitrary local builder option(s), enclose BUILDER\_OPTIONS in quotes and note that = before it is mandatory

**--build-retries** BUILD\_RETRIES  
number of retries of a failed build, defaults to 2

**--results-dir** RESULTS\_DIR  
directory where rebase-helper output will be stored

## Writing plugins

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

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

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

### Example

Listing 3.1: my\_spec\_hook/\_\_init\_\_.py

```
from rebasehelper.specfile import BaseSpecHook

class MySpecHook(BaseSpecHook):

    NAME = 'MySpecHook'

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

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

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

Listing 3.2: 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']
    }
)

```

## API

### Application module

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

**build\_packages** ()

Function calls build class for building packages

**debug\_log\_file** = None

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

Extracts given archive into the destination and handle all exceptions.

#### Parameters

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

#### Returns

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

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

**generate\_patch** ()

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

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

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

#### Returns

**get\_checker\_outputs** ()

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

**get\_rebased\_patches** ()

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

**get\_rebasehelper\_data** ()

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

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

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

**Returns**

**kwargs** = {}

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

**new\_sources** = ''

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

**old\_sources** = ''

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

**prepare\_sources** ()

Function prepares a sources.

**Returns**

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

Save rebase-helper result and print the summary using `output_tools_runner` :param *exception*: Error message from rebase-helper :return:

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

**rebase\_spec\_file** = None

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

**rebased\_patches** = {}

**rebased\_repo** = None

**report\_log\_file** = None

**result\_file** = ''

**run** ()

**run\_download\_compare** (*tasks\_dict, dir\_name*)

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

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

**Parameters** **results\_dir** (*str*) – Path to directory in which to store the results.

**Returns** None

**set\_upstream\_monitoring** ()

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

**spec\_file** = None

**spec\_file\_path** = None

**temp\_dir** = ''

**upstream\_monitoring** = False



## Archive module

```

class rebasehelper.archive.Archive (filename=None)
    Class representing an archive with sources

    extract_archive (path=None)
        Extracts the archive into the given path

        Parameters path – Path where to extract the archive to.

        Returns

    classmethod get_supported_archives ()
        Return list of supported archive types

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

    EXTENSION = ''

    classmethod extract (filename=None, *args, **kwargs)
        Extracts the archive into the given path

        Parameters path – Path where to extract the archive to.

        Returns

    classmethod match (filename=None, *args, **kwargs)
        Checks if the filename matches the archive type. If yes, returns True, otherwise returns False.

    classmethod open (filename=None, *args, **kwargs)
        Opens archive with the given filename and returns the proper archive type object.

class rebasehelper.archive.Bz2ArchiveType
    .bz2 archive type

    EXTENSION = '.bz2'

    classmethod extract (archive=None, filename=None, path=None, *args, **kwargs)

    classmethod open (filename=None)

class rebasehelper.archive.GemPseudoArchiveType
    .gem files are not archives - this is a pseudo type

    EXTENSION = '.gem'

    classmethod extract (archive=None, filename=None, path=None, *args, **kwargs)

    classmethod open (filename=None)

class rebasehelper.archive.TarBz2ArchiveType
    .tar.bz2 archive type

    EXTENSION = '.tar.bz2'

class rebasehelper.archive.TarGzArchiveType
    .tar.gz archive type

    EXTENSION = '.tar.gz'

    classmethod extract (archive=None, filename=None, path=None, *args, **kwargs)

    classmethod open (filename=None)

class rebasehelper.archive.TarXzArchiveType
    .tar.xz archive type

```

**EXTENSION** = `‘.tar.xz’`

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

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

**class** `rebasehelper.archive.TgzArchiveType`  
.tgz archive type

**EXTENSION** = `‘.tgz’`

**class** `rebasehelper.archive.ZipArchiveType`  
.zip archive type

**EXTENSION** = `‘.zip’`

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

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

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

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

## Build helper module

**exception** `rebasehelper.build_helper.BinaryPackageBuildError`  
Error indicating failure to build Binary Package

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

Class representing temporary environment.

**TEMPDIR\_RESULTS** = `‘TEMPDIR_RESULTS’`

**TEMPDIR\_SOURCES** = `‘TEMPDIR_SOURCES’`

**TEMPDIR\_SPEC** = `‘TEMPDIR_SPEC’`

**TEMPDIR\_SPECS** = `‘TEMPDIR_SPECS’`

**class** `rebasehelper.build_helper.BuildToolBase`  
Base class for various build tools

**DEFAULT** = `False`

**classmethod** `accepts_options` ()  
Checks if the tool accepts additional command line options.

**classmethod** `build` (*\*args, \*\*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

**classmethod** `creates_tasks` ()  
Checks if the tool creates build tasks.

**classmethod** `get_build_tool_name` ()  
Returns the name of the build tool.

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

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

Gets packages and logs for specified task

**Parameters**

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

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

**classmethod** `get_logs` ()

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

**classmethod** `get_task_info` (*build\_dict*)

Gets information about detached remote task

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

**Returns** task info

**classmethod** `is_default` ()

Checks if the tool is the default build tool.

**classmethod** `match` (*cmd=None*, *\*args*, *\*\*kwargs*)

Checks if tool name matches the desired one.

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

Prepare for building.

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

**classmethod** `wait_for_task` (*build\_dict*, *results\_dir*)

Waits until specified task is finished

**Parameters**

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

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

**class** `rebasehelper.build_helper.Builder` (*tool=None*)

Class representing a process of building binaries from sources.

**accepts\_options** ()

**build** (*\*args*, *\*\*kwargs*)

Build sources.

**build\_tools** = {'copr': <class 'rebasehelper.build\_tools.copr\_tool.CoprBuildTool'>, 'rpmbuild': <class 'rebasehelper.b

**creates\_tasks** ()

**classmethod** `get_default_tool` ()

Returns default build tool

**get\_detached\_task** (*task\_id*, *results\_dir*)

Get detached task

**get\_logs** ()

Get logs.

**classmethod** `get_supported_tools` ()

Returns a list of supported build tools

**Returns** list of supported build tools

**get\_task\_info** (*build\_dict*)  
Get task info

**classmethod load\_build\_tools** ()

**prepare** (*spec, conf*)  
Prepare for build

**wait\_for\_task** (*build\_dict, results\_dir*)  
Wait for task

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

Class representing temporary environment for RpmbuildBuildTool.

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

**TEMPDIR\_BUILDRoot** = 'TEMPDIR\_BUILDRoot'

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

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

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

**exception** `rebasehelper.build_helper.SourcePackageBuildError`  
Error indicating failure to build Source Package.

## Build log analyzer module

**class** `rebasehelper.build_log_analyzer.BuildLogAnalyzer`  
Class analyze the log provided by build programs

**log\_dirname** = ''

**classmethod parse\_log** (*dir\_name, log\_name*)  
Function analyze the logs for specific section

### Parameters

- **log\_name** – Logfile name which is analyzed
- **start\_string** – Start string

**Returns** list of files

**exception** `rebasehelper.build_log_analyzer.BuildLogAnalyzerMakeError`  
Error indicating failure problem with building

**exception** `rebasehelper.build_log_analyzer.BuildLogAnalyzerMissingError`  
Error indicating build.log is missing

**exception** `rebasehelper.build_log_analyzer.BuildLogAnalyzerPatchError`  
Error indicating failure problem with building

## CLI module

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

**static build\_parser** ()

```
class rebasehelper.cli.CliHelper
```

```
    static run ()
```

```
class rebasehelper.cli.CustomArgumentParser (prog=None, usage=None, description=None,
                                             epilog=None, version=None,
                                             parents=[], formatter_class=<class 'arg-
                                             parse.HelpFormatter'>, prefix_chars='-
                                             ', fromfile_prefix_chars=None, argu-
                                             ment_default=None, conflict_handler='error',
                                             add_help=True)
```

```
    error (message)
```

```
class rebasehelper.cli.CustomHelpFormatter (prog, indent_increment=2,
                                             max_help_position=24, width=None)
```

## Exception module

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

Error indicating failure unable to find checker binary.

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

Class representing Error raised inside rebase-helper after intentionally catching some expected and well known exception/error.

## Logger module

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

Helper class for setting up a logger

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

Adds FileHandler to a given logger

### Parameters

- **logger** – Logger object to which the file handler will be added
- **path** – Path to file where the debug log will be written

**Returns** None

```
    static add_stream_handler (logger, level=None)
```

Adds console handler with given severity.

### Parameters

- **logger** – logger object to add the handler to
- **level** – severity level

**Returns** created handler object

```
    static get_basic_logger (logger_name, level=10)
```

Sets-up a basic logger without any handler

### Parameters

- **logger\_name** – Logger name
- **level** – severity level

**Returns** created logger

### Output tool module

**class** `rebasehelper.output_tool.BaseOutputTool`

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

**DEFAULT** = `False`

**EXTENSION** = `'ext'`

**NAME** = `'name'`

**static** `get_default_tool()`

Returns default output tool

**classmethod** `get_extension()`

**classmethod** `get_name()`

**classmethod** `get_report_path(app)`

**static** `get_supported_tools()`

Returns list of supported output tools

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

Checks if tool name matches the desired one.

**classmethod** `print_cli_summary(app)`

Print report of the rebase

**Parameters** `app` – Application instance

**classmethod** `print_patches_cli()`

Print info about patches

**classmethod** `print_patches_section_cli(color, patch_type)`

Print info about one of the patches key section

**Parameters**

- `color` – color used for the message printing
- `patch_type` – string containing key for the `patch_dict`

**classmethod** `print_report_file_path()`

Print path to the report file

**classmethod** `run()`

**class** `rebasehelper.output_tool.OutputToolRunner`

Class representing the process of running various output tools.

**run\_output\_tools** (`logs=None, app=None`)

Runs all output tools.

**Parameters**

- `log` – Log that probably contains the important message concerning the rebase fail
- `app` – Application class instance

## Patch helper module

```

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

    CMD = 'git'

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

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

    classmethod call_prep_script (prep_script_path)

    classmethod create_prep_script (prep)
        Function abstract special things from prep section and apply them to old sources

    diff_cls = None

    exec_prep_script = False

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

    classmethod match (cmd=None)

    new_repo = None

    new_sources = ''

    non_interactive = False

    old_repo = None

    old_sources = ''

    output_data = None

    patch_sources_by_prep_script = False

    patches = None

    prep_section = False

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

    source_dir = ''

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

    classmethod match (cmd=None, *args, **kwargs)
        Method checks whether it is usefull patch method

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

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

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

```

### Parameters

- **old\_dir** – path to dir with old patches
- **new\_dir** – path to dir with new patches
- **patches** – old patches
- **rebased\_patches** – rebased patches
- **kwargs** --

### Returns

`rebasehelper.patch_helper.register_patch_tool(patch_tool)`

## Specfile module

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

Base class for a spec hook

**classmethod** `get_name()`

Returns the name of a spec hook

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

Runs a spec hook.

### Parameters

- **spec\_file** – Original spec file object
- **rebase\_spec\_file** – Rebased spec file object

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

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

Class represents set of information about patches

`get_index()`

`get_option()`

`get_patch_name()`

`get_path()`

`git_generated = ''`

`index = ''`

`option = ''`

`path = ''`

`set_path(new_path)`

**class** `rebasehelper.specfile.SpecFile` (*path, sources\_location='', download=True*)

Class representing a SPEC file

**static construct\_string\_with\_comment** (*lines*)

Wraps the line in a rebase-helper specific comments

**Parameters** **lines** – line (or list of lines) to be wrapped

**Returns** list with lines

**copy** (*new\_path=None*)

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



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

**Returns** copy of the current object

`defined_sections = ['%package', '%description', '%prep', '%build', '%install', '%check', '%files', '%changelog']`

`download = False`

`download_remote_sources ()`

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

**Returns** None

`extra_version = None`

`static extract_version_from_archive_name (archive_path, source_string='')`

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

**Parameters**

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

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

`find_archive_target_in_prep (archive)`

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

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

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

`get_applied_patches ()`

Method returns list of all applied patches.

**Returns** list of PatchObject

`get_archive ()`

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

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

**Return type** str

`get_epoch_number ()`

Method for getting epoch of the package

**Returns**

`get_extra_version ()`

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

**Returns** String

`get_extra_version_separator ()`

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

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

**Return type** str

**get\_full\_version ()**

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

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

**Return type** str

**get\_new\_log ()**

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

Method returns list of all unapplied patches.

**Returns** list of PatchObject

**get\_package\_name ()**

Function returns a package name

**Returns**

**get\_patch\_option (line)**

Function returns a patch options

**Parameters line** –

**Returns** patch options like -p1

**get\_patches ()**

Method returns list of all applied and not applied patches

**Returns** list of PatchObject

**get\_path ()**

Return only spec file path

**Returns**

**static get\_paths\_with\_rpm\_macros (files)**

Method modifies paths in passed list to use RPM macros

**Parameters files** – list of absolute paths

**Returns** modified list of paths with RPM macros

**get\_prep\_section (complete=False)**

Function returns whole prep section

**get\_release ()**

Method for getting full release string of the package

**Returns**

**get\_release\_number ()**

Method for getting the release of the package

**Returns**

**get\_requires ()**

Function returns a package requirements

**Returns**

**get\_setup\_dirname ()**

Get dirname from %setup or %autosetup macro arguments

**Returns** dirname

**get\_sources** ()

Method returns dictionary with local sources list.

**Returns** list of Sources with absolute path

**Return type** list of str

**get\_spec\_section** (*section\_name*)

Returns the section of selected name

**Parameters** **section\_name** – section name to get

**Returns** list of lines contained in the selected section

**get\_version** ()

Method returns the version

**Returns**

**hdr** = None

**insert\_changelog** (*new\_log*)

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

Returns whether test suite is enabled during the build time

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

**modify\_spec\_files\_section** (*files*)

Function repairs spec file according to new sources.

**Parameters** **files** –

**Returns**

**patches** = None

**path** = ‘

**prep\_section** = []

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

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

**Parameters** **macro** –

**Returns**

**removed\_patches** = []

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

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

**Parameters** **macro** –

**Returns**

**rpm\_sections** = {}

**save** ()

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

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

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

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

**Returns** None

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

Set the string that separates the version and extra version

**Parameters** **separator** –

**Returns**

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

Method to set release number

**Parameters** **release** –

**Returns**

**set\_spec\_section** (*section\_name, new\_section*)

Returns the section of selected name

**Parameters** **section\_name** – section name to get

**Returns** list of lines contained in the selected section

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

Method to update the version in the SPEC file

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

**Returns** None

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

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

**Parameters** **archive\_path** –

**Returns**

**sources** = None

**spc** = None

**spec\_content** = []

**static split\_version\_string** (*version\_string=''*)

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

**Parameters** **version\_string** – version string such as '1.1.1' or '1.2.3b1', ...

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

**update\_changelog** (*new\_log*)

Function updates changelog with new version

**update\_paths\_to\_patches** ()

**update\_setup\_dirname** (*dirname*)

Update %setup or %autosetup dirname argument if needed

**Parameters** **dirname** – new dirname to be used

**write\_updated\_patches** (*patches, disable\_inapplicable*)

Function writes the patches to -rebase.spec file

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

Class representing the process of running various spec file hooks.

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

Runs all spec hooks.

**Parameters**

- **spec\_file** – Original spec file object
- **rebase\_spec\_file** – Rebased spec file object

`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

## Utils module

**class** `rebasehelper.utils.ConsoleHelper`

Class for command line interaction with the user.

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

ContextManager for capturing stdout/stderr

**classmethod** `ConsoleHelper.cprint` (*message, color=None*)

Print colored output if possible

**Parameters**

- **color** – color to be used in the output
- **message** – string to be printed out

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

Function for command line messages

**Parameters**

- **message** – prompt string
- **default\_yes** – If the default value is YES
- **any\_input** – if True, return input without checking it first

**Returns** True or False, based on user's input

**classmethod** `ConsoleHelper.should_use_colors` (*conf*)

Determine whether ansi colors should be used for CLI output

`ConsoleHelper.use_colors = False`

**class** `rebasehelper.utils.CoprHelper`

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

**classmethod** `create_project` (*client, project, chroot, description, instructions*)

**classmethod** `download_build` (*client, build\_id, destination*)

**classmethod** `get_build_status` (*client, build\_id*)

**classmethod** `get_build_url` (*client, build\_id*)

**classmethod** `get_client()`

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

**exception** `rebasehelper.utils.DownloadError`

Exception indicating that download of a file failed.

**class** `rebasehelper.utils.DownloadHelper`

Class for downloading sources defined in SPEC file

**static** `download_file(url, destination_path, timeout=10, blocksize=8192)`

Method for downloading file from HTTP, HTTPS and FTP URL.

#### Parameters

- **url** – URL from which to download the file
- **destination\_path** – path where to store downloaded file
- **timeout** – timeout in seconds for blocking actions like connecting, etc.
- **blocksize** – size in Bytes of blocks used for downloading the file and reporting progress

**Returns** None

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

The function prints download progress and remaining time of the download directly to the standard output.

#### Parameters

- **download\_total** (*int or float*) – size of the file which is being downloaded
- **downloaded** (*int or float*) – already downloaded size of the file
- **start\_time** (*float*) – time in seconds since the epoch from the point when the download started. This is used to calculate the remaining time of the download.

**Returns** None

**class** `rebasehelper.utils.FileHelper`

**static** `file_available(filename)`

**class** `rebasehelper.utils.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)`

**exception** `rebasehelper.utils.GitRebaseError`

Error indicating problems with Git

**exception** `rebasehelper.utils.GitRuntimeError`

Error indicating problems with Git

**class** `rebasehelper.utils.KojiHelper`

**baseurl** = 'http://kojipkgs.fedoraproject.org/work/'

```

baseurl_pkg = 'https://kojipkgs.fedoraproject.org/packages/'
ca_cert = '/home/docs/.fedora-server-ca.cert'
cert = '/home/docs/.fedora.cert'
classmethod display_task_results (tasks)
    Taken from from koji_cli.lib
classmethod download_build (build_id, destination)
    Downloads all x86_64 RPMs and logs of a Koji build
    Parameters
        • build_id – Koji build ID
        • destination – target path
    Returns (list of paths to RPMs, list of paths to logs)
classmethod download_scratch_build (session, task_list, dir_name)
functional = False
classmethod get_koji_tasks (task_id, dir_name)
classmethod get_latest_build (package)
    Searches for the latest Koji build of a package
    Parameters package – package name
    Returns (latest version, Koji build ID)
koji_web = 'koji.fedoraproject.org'
scratch_url = 'http://koji.fedoraproject.org/work/'
server = 'https://koji.fedoraproject.org/kojihub'
classmethod session_maker (baseurl=None)
classmethod upload_srpm (session, source)
classmethod watch_koji_tasks (session, tasklist)
    Taken from from koji_cli.lib
exception rebasehelper.utils.LookasideCacheError
    Exception indicating a problem accessing lookaside cache
class rebasehelper.utils.LookasideCacheHelper
    Class for downloading files from Fedora/RHEL lookaside cache
    classmethod download (tool, basepath, package)
    rpkg_config_dir = '/etc/rpkg'
class rebasehelper.utils.MacroHelper
    Helper class for working with RPM macros
    static dump ()
        Returns list of all defined macros
        Returns list of macros
    static filter (macros, **kwargs)
        Returns all macros satisfying specified filters
    Parameters

```

- **macros** – list of macros to be filtered
- **kwargs** – filters

**Returns** filtered list of macros

**class** `rebasehelper.utils.PathHelper`

Class which finds a file or files in specific directory

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

Finds a file that matches the given 'pattern' recursively starting in the 'top\_path' directory. If found, returns full path to the first occurrence of the file, otherwise returns None.

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

Finds all files that matches the given 'pattern' in the 'top\_path' directory. If found, returns fields of all files, otherwise returns None.

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

Finds a file that matches the given 'pattern' recursively starting in the 'top\_path' directory. If found, returns full path to the directory with first occurrence of the file, otherwise returns None.

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

Finds a file that matches the given 'pattern' recursively starting in the 'top\_path' directory. If found, returns full path to the first occurrence of the file, otherwise returns None.

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

Returns a path to new temporary directory.

**class** `rebasehelper.utils.ProcessHelper`

Class for execution subprocess

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

**static run\_subprocess** (*cmd, input=None, output=None*)

Runs the passed command as a subprocess.

**Parameters**

- **cmd** – command with arguments to be run
- **input** – file to read the input from. If None, read from STDIN
- **output** – file to write the output of the command. If None, write to STDOUT

**Returns** exit code of the process

**static run\_subprocess\_cwd** (*cmd, cwd=None, input=None, output=None, shell=False*)

Runs the passed command as a subprocess in different working directory.

**Parameters**

- **cmd** – command with arguments to be run
- **cwd** – the directory to change the working dir to
- **input** – file to read the input from. If None, read from STDIN
- **output** – file to write the output of the command. If None, write to STDOUT
- **shell** – if to run the command as shell command (default: False)

**Returns** exit code of the process

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

Runs the passed command as a subprocess in different working directory with possibly changed ENVIRONMENT VARIABLES.



**Parameters**

- **cmd** – command with arguments to be run
- **cwd** – the directory to change the working dir to
- **env** – dictionary with ENVIRONMENT VARIABLES to define
- **input** – file to read the input from. If None, read from STDIN
- **output** – file to write the output of the command. If None, write to STDOUT
- **shell** – if to run the command as shell command (default: False)

**Returns** exit code of the process

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

Runs the passed command as a subprocess with possibly changed ENVIRONMENT VARIABLES.

**Parameters**

- **cmd** – command with arguments to be run
- **env** – dictionary with ENVIRONMENT VARIABLES to define
- **input** – file to read the input from. If None, read from STDIN
- **output** – file to write the output of the command. If None, write to STDOUT
- **shell** – if to run the command as shell command (default: False)

**Returns** exit code of the process

**class** `rebasehelper.utils.RpmHelper`

Helper class for doing various tasks with RPM database, packages, ...

**ARCHES = None**

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

Check if all packages in passed list are installed.

**Parameters** **pkg\_names** – iterable with package named to check for

**Returns** True if all packages are installed, False if at least one package is not installed.

**static get\_arches** ()

Get list of all known architectures

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

Function returns a rpm header from given rpm package for later on analysis

**Parameters** **rpm\_name** –

**Returns**

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

Method returns a name of the package from RPM file format

**Parameters** **pkg\_name** –

**Returns**

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

Install all build requires for a package using PolicyKits

**Parameters** **spec\_path** – absolute path to SPEC file

**Returns**

**static** `is_package_installed` (*pkg\_name=None*)

Checks whether package with passed name is installed.

**Parameters** `package_name` – package name we want to check for

**Returns** True if installed, False if not installed

**classmethod** `split_nevra` (*string*)

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

**class** `rebasehelper.utils.TemporaryEnvironment` (*exit\_callback=None, \*\*kwargs*)

Class representing a temporary environment (directory) that can be used as a workspace. It can be used with `with` statement.

**TEMPDIR** = 'TEMPDIR'

**env** ()

Returns copy of `_env` dictionary.

**Returns** copy of `_env` dictionary

**path** ()

Returns path to the temporary environment.

**Returns** abs path to the environment

`rebasehelper.utils.get_value_from_kwargs` (*kwargs, field, source='old'*)

Function returns a part of `self.kwargs` dictionary

**Parameters**

- **kwargs** –
- **source** – 'old' or 'new'
- **field** – like 'patches', 'source'

**Returns** value from dictionary

## Change Log

### [Unreleased]

### [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

## [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

## [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

## [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

## [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

### [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

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

#### Added

- Two new command line options used by upstream monitoring

#### Fixed

- **fedpkg** reimplementation

### [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

### [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

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

#### Added

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

### [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`

### [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

## [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

## [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

## [0.3.0]

### Added

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

## [0.2.0]

### Added

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

## [0.1.0]

### Added

- Initial classes
- CLI interface



**r**

rebasehelper.application, 11  
rebasehelper.archive, 13  
rebasehelper.build\_helper, 14  
rebasehelper.build\_log\_analyzer, 16  
rebasehelper.cli, 16  
rebasehelper.exceptions, 17  
rebasehelper.logger, 17  
rebasehelper.output\_tool, 18  
rebasehelper.patch\_helper, 19  
rebasehelper.specfile, 20  
rebasehelper.utils, 25



## Symbols

-build-retries BUILD\_RETRIES  
     rebase-helper command line option, 9  
 -build-tasks OLD\_TASK,NEW\_TASK  
     rebase-helper command line option, 9  
 -builder-options=BUILDER\_OPTIONS  
     rebase-helper command line option, 9  
 -builds-nowait  
     rebase-helper command line option, 9  
 -buildtool {copr,rpmbuild,mock}  
     rebase-helper command line option, 9  
 -color {always,never,auto}  
     rebase-helper command line option, 9  
 -comparepkgs-only COMPAREPKGS\_DIR  
     rebase-helper command line option, 8  
 -disable-inapplicable-patches  
     rebase-helper command line option, 9  
 -non-interactive  
     rebase-helper command line option, 9  
 -not-download-sources  
     rebase-helper command line option, 9  
 -outputtool {text,json}  
     rebase-helper command line option, 9  
 -pkgcomparetool {rpmdiff,abipkgdiff,pkgdiff,csmock}  
     rebase-helper command line option, 9  
 -results-dir RESULTS\_DIR  
     rebase-helper command line option, 9  
 -version  
     rebase-helper command line option, 8  
 -versioneer {py,anitya}  
     rebase-helper command line option, 9  
 -b, -build-only  
     rebase-helper command line option, 8  
 -c, -continue  
     rebase-helper command line option, 9  
 -h, -help  
     rebase-helper command line option, 8  
 -p, -patch-only  
     rebase-helper command line option, 8

-v, -verbose  
     rebase-helper command line option, 8  
 -w, -keep-workspace  
     rebase-helper command line option, 9

## A

accepts\_options() (rebasehelper.build\_helper.Builder  
     method), 15  
 accepts\_options() (rebase-  
     helper.build\_helper.BuildToolBase class  
     method), 14  
 add\_file\_handler() (rebasehelper.logger.LoggerHelper  
     static method), 17  
 add\_stream\_handler() (rebasehelper.logger.LoggerHelper  
     static method), 17  
 all\_packages\_installed() (rebasehelper.utils.RpmHelper  
     static method), 29  
 Application (class in rebasehelper.application), 11  
 apply\_old\_patches() (rebase-  
     helper.patch\_helper.GitPatchTool class  
     method), 19  
 apply\_patch() (rebasehelper.patch\_helper.GitPatchTool  
     static method), 19  
 ARCHES (rebasehelper.utils.RpmHelper attribute), 29  
 Archive (class in rebasehelper.archive), 13  
 ArchiveTypeBase (class in rebasehelper.archive), 13

## B

BaseOutputTool (class in rebasehelper.output\_tool), 18  
 BaseSpecHook (class in rebasehelper.specfile), 20  
 baseurl (rebasehelper.utils.KojiHelper attribute), 26  
 baseurl\_pkg (rebasehelper.utils.KojiHelper attribute), 26  
 BinaryPackageBuildError, 14  
 build() (rebasehelper.build\_helper.Builder method), 15  
 build() (rebasehelper.build\_helper.BuildToolBase class  
     method), 14  
 build() (rebasehelper.utils.CoprHelper class method), 25  
 build\_packages() (rebasehelper.application.Application  
     method), 11

- build\_parser() (rebasehelper.cli.CLI static method), 16
- build\_tools (rebasehelper.build\_helper.Builder attribute), 15
- Builder (class in rebasehelper.build\_helper), 15
- BuildLogAnalyzer (class in rebasehelper.build\_log\_analyzer), 16
- BuildLogAnalyzerMakeError, 16
- BuildLogAnalyzerMissingError, 16
- BuildLogAnalyzerPatchError, 16
- BuildTemporaryEnvironment (class in rebasehelper.build\_helper), 14
- BuildToolBase (class in rebasehelper.build\_helper), 14
- Bz2ArchiveType (class in rebasehelper.archive), 13
- C**
- ca\_cert (rebasehelper.utils.KojiHelper attribute), 27
- call\_prep\_script() (rebasehelper.patch\_helper.GitPatchTool class method), 19
- cert (rebasehelper.utils.KojiHelper attribute), 27
- CheckerNotFoundError, 17
- CLI (class in rebasehelper.cli), 16
- CliHelper (class in rebasehelper.cli), 16
- CMD (rebasehelper.patch\_helper.GitPatchTool attribute), 19
- ConsoleHelper (class in rebasehelper.utils), 25
- ConsoleHelper.Capturer (class in rebasehelper.utils), 25
- construct\_string\_with\_comment() (rebasehelper.specfile.SpecFile static method), 20
- CoprHelper (class in rebasehelper.utils), 25
- copy() (rebasehelper.specfile.SpecFile method), 20
- cpriint() (rebasehelper.utils.ConsoleHelper class method), 25
- create\_prep\_script() (rebasehelper.patch\_helper.GitPatchTool class method), 19
- create\_project() (rebasehelper.utils.CoprHelper class method), 25
- creates\_tasks() (rebasehelper.build\_helper.Builder method), 15
- creates\_tasks() (rebasehelper.build\_helper.BuildToolBase class method), 14
- CustomArgumentParser (class in rebasehelper.cli), 17
- CustomHelpFormatter (class in rebasehelper.cli), 17
- D**
- debug\_log\_file (rebasehelper.application.Application attribute), 11
- DEFAULT (rebasehelper.build\_helper.BuildToolBase attribute), 14
- DEFAULT (rebasehelper.output\_tool.BaseOutputTool attribute), 18
- defined\_sections (rebasehelper.specfile.SpecFile attribute), 21
- DEV\_NULL (rebasehelper.utils.ProcessHelper attribute), 28
- diff\_cls (rebasehelper.patch\_helper.GitPatchTool attribute), 19
- display\_task\_results() (rebasehelper.utils.KojiHelper class method), 27
- download (rebasehelper.specfile.SpecFile attribute), 21
- download() (rebasehelper.utils.LookasideCacheHelper class method), 27
- download\_build() (rebasehelper.utils.CoprHelper class method), 25
- download\_build() (rebasehelper.utils.KojiHelper class method), 27
- download\_file() (rebasehelper.utils.DownloadHelper static method), 26
- download\_remote\_sources() (rebasehelper.specfile.SpecFile method), 21
- download\_scratch\_build() (rebasehelper.utils.KojiHelper class method), 27
- DownloadError, 26
- DownloadHelper (class in rebasehelper.utils), 26
- dump() (rebasehelper.utils.MacroHelper static method), 27
- E**
- env() (rebasehelper.utils.TemporaryEnvironment method), 30
- error() (rebasehelper.cli.CustomArgumentParser method), 17
- exec\_prep\_script (rebasehelper.patch\_helper.GitPatchTool attribute), 19
- EXTENSION (rebasehelper.archive.ArchiveTypeBase attribute), 13
- EXTENSION (rebasehelper.archive.Bz2ArchiveType attribute), 13
- EXTENSION (rebasehelper.archive.GemPseudoArchiveType attribute), 13
- EXTENSION (rebasehelper.archive.TarBz2ArchiveType attribute), 13
- EXTENSION (rebasehelper.archive.TarGzArchiveType attribute), 13
- EXTENSION (rebasehelper.archive.TarXzArchiveType attribute), 13
- EXTENSION (rebasehelper.archive.TgzArchiveType attribute), 14
- EXTENSION (rebasehelper.archive.ZipArchiveType attribute), 14
- EXTENSION (rebasehelper.output\_tool.BaseOutputTool attribute), 18
- extra\_version (rebasehelper.specfile.SpecFile attribute), 21

- extract() (rebasehelper.archive.ArchiveTypeBase class method), 13
- extract() (rebasehelper.archive.Bz2ArchiveType class method), 13
- extract() (rebasehelper.archive.GemPseudoArchiveType class method), 13
- extract() (rebasehelper.archive.TarGzArchiveType class method), 13
- extract() (rebasehelper.archive.TarXzArchiveType class method), 14
- extract() (rebasehelper.archive.ZipArchiveType class method), 14
- extract\_archive() (rebasehelper.application.Application static method), 11
- extract\_archive() (rebasehelper.archive.Archive method), 13
- extract\_sources() (rebasehelper.application.Application static method), 11
- extract\_version\_from\_archive\_name() (rebasehelper.specfile.SpecFile static method), 21
- ## F
- file\_available() (rebasehelper.utils.FileHelper static method), 26
- FileHelper (class in rebasehelper.utils), 26
- filter() (rebasehelper.utils.MacroHelper static method), 27
- find\_all\_files() (rebasehelper.utils.PathHelper static method), 28
- find\_all\_files\_current\_dir() (rebasehelper.utils.PathHelper static method), 28
- find\_archive\_target\_in\_prep() (rebasehelper.specfile.SpecFile method), 21
- find\_first\_dir\_with\_file() (rebasehelper.utils.PathHelper static method), 28
- find\_first\_file() (rebasehelper.utils.PathHelper static method), 28
- functional (rebasehelper.utils.KojiHelper attribute), 27
- ## G
- GemPseudoArchiveType (class in rebasehelper.archive), 13
- generate\_patch() (rebasehelper.application.Application method), 11
- get\_all\_log\_files() (rebasehelper.application.Application method), 11
- get\_applied\_patches() (rebasehelper.specfile.SpecFile method), 21
- get\_arches() (rebasehelper.utils.RpmHelper static method), 29
- get\_archive() (rebasehelper.specfile.SpecFile method), 21
- get\_basic\_logger() (rebasehelper.logger.LoggerHelper static method), 17
- get\_build\_status() (rebasehelper.utils.CoprHelper class method), 25
- get\_build\_tool\_name() (rebasehelper.build\_helper.BuildToolBase class method), 14
- get\_build\_url() (rebasehelper.utils.CoprHelper class method), 25
- get\_builder\_options() (rebasehelper.build\_helper.BuildToolBase static method), 14
- get\_checker\_outputs() (rebasehelper.application.Application method), 11
- get\_client() (rebasehelper.utils.CoprHelper class method), 26
- get\_default\_tool() (rebasehelper.build\_helper.Builder class method), 15
- get\_default\_tool() (rebasehelper.output\_tool.BaseOutputTool static method), 18
- get\_detached\_task() (rebasehelper.build\_helper.Builder method), 15
- get\_detached\_task() (rebasehelper.build\_helper.BuildToolBase class method), 14
- get\_email() (rebasehelper.utils.GitHelper class method), 26
- get\_epoch\_number() (rebasehelper.specfile.SpecFile method), 21
- get\_extension() (rebasehelper.output\_tool.BaseOutputTool class method), 18
- get\_extra\_version() (rebasehelper.specfile.SpecFile method), 21
- get\_extra\_version\_separator() (rebasehelper.specfile.SpecFile method), 21
- get\_full\_version() (rebasehelper.specfile.SpecFile method), 21
- get\_header\_from\_rpm() (rebasehelper.utils.RpmHelper static method), 29
- get\_index() (rebasehelper.specfile.PatchObject method), 20
- get\_info\_from\_rpm() (rebasehelper.utils.RpmHelper static method), 29
- get\_koji\_tasks() (rebasehelper.utils.KojiHelper class method), 27
- get\_latest\_build() (rebasehelper.utils.KojiHelper class method), 27
- get\_logs() (rebasehelper.build\_helper.Builder method), 15
- get\_logs() (rebasehelper.build\_helper.BuildToolBase class method), 15
- get\_message() (rebasehelper.utils.ConsoleHelper static method), 25

[get\\_name\(\)](#) (rebasehelper.output\_tool.BaseOutputTool class method), 18  
[get\\_name\(\)](#) (rebasehelper.specfile.BaseSpecHook class method), 20  
[get\\_new\\_build\\_logs\(\)](#) (rebasehelper.application.Application method), 11  
[get\\_new\\_log\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_not\\_used\\_patches\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_option\(\)](#) (rebasehelper.specfile.PatchObject method), 20  
[get\\_package\\_name\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_patch\\_name\(\)](#) (rebasehelper.specfile.PatchObject method), 20  
[get\\_patch\\_option\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_patches\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_path\(\)](#) (rebasehelper.specfile.PatchObject method), 20  
[get\\_path\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_paths\\_with\\_rpm\\_macros\(\)](#) (rebasehelper.specfile.SpecFile static method), 22  
[get\\_prep\\_section\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_rebase\\_name\(\)](#) (in module rebasehelper.specfile), 25  
[get\\_rebased\\_patches\(\)](#) (rebasehelper.application.Application method), 11  
[get\\_rebasehelper\\_data\(\)](#) (rebasehelper.application.Application method), 11  
[get\\_release\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_release\\_number\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_report\\_path\(\)](#) (rebasehelper.output\_tool.BaseOutputTool class method), 18  
[get\\_requires\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_rpm\\_packages\(\)](#) (rebasehelper.application.Application method), 12  
[get\\_setup\\_dirname\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_sources\(\)](#) (rebasehelper.specfile.SpecFile method), 22  
[get\\_spec\\_section\(\)](#) (rebasehelper.specfile.SpecFile method), 23  
[get\\_supported\\_archives\(\)](#) (rebasehelper.archive.Archive class method), 13  
[get\\_supported\\_tools\(\)](#) (rebasehelper.build\_helper.Builder class method), 15  
[get\\_supported\\_tools\(\)](#) (rebasehelper.output\_tool.BaseOutputTool static method), 18  
[get\\_task\\_info\(\)](#) (rebasehelper.build\_helper.Builder method), 16  
[get\\_task\\_info\(\)](#) (rebasehelper.build\_helper.BuildToolBase class method), 15  
[get\\_temp\\_dir\(\)](#) (rebasehelper.utils.PathHelper static method), 28  
[get\\_user\(\)](#) (rebasehelper.utils.GitHelper class method), 26  
[get\\_value\\_from\\_kwargs\(\)](#) (in module rebasehelper.utils), 30  
[get\\_version\(\)](#) (rebasehelper.specfile.SpecFile method), 23  
[git\\_generated](#) (rebasehelper.specfile.PatchObject attribute), 20  
[GIT\\_USER\\_EMAIL](#) (rebasehelper.utils.GitHelper attribute), 26  
[GIT\\_USER\\_NAME](#) (rebasehelper.utils.GitHelper attribute), 26  
[GitHelper](#) (class in rebasehelper.utils), 26  
[GitPatchTool](#) (class in rebasehelper.patch\_helper), 19  
[GitRebaseError](#), 26  
[GitRuntimeError](#), 26

## H

[hdr](#) (rebasehelper.specfile.SpecFile attribute), 23

## I

[index](#) (rebasehelper.specfile.PatchObject attribute), 20  
[init\\_git\(\)](#) (rebasehelper.patch\_helper.GitPatchTool class method), 19  
[insert\\_changelog\(\)](#) (rebasehelper.specfile.SpecFile method), 23  
[install\\_build\\_dependencies\(\)](#) (rebasehelper.utils.RpmHelper static method), 29  
[is\\_default\(\)](#) (rebasehelper.build\_helper.BuildToolBase class method), 15  
[is\\_package\\_installed\(\)](#) (rebasehelper.utils.RpmHelper static method), 29  
[is\\_test\\_suite\\_enabled\(\)](#) (rebasehelper.specfile.SpecFile method), 23

## K

[koji\\_web](#) (rebasehelper.utils.KojiHelper attribute), 27  
[KojiHelper](#) (class in rebasehelper.utils), 26  
[kwargs](#) (rebasehelper.application.Application attribute), 12

## L

[load\\_build\\_tools\(\)](#) (rebasehelper.build\_helper.Builder class method), 16

log\_dirname (rebasehelper.build\_log\_analyzer.BuildLogAnalyzer attribute), 16  
 LoggerHelper (class in rebasehelper.logger), 17  
 LookasideCacheError, 27  
 LookasideCacheHelper (class in rebasehelper.utils), 27

## M

MacroHelper (class in rebasehelper.utils), 27  
 match() (rebasehelper.archive.ArchiveTypeBase class method), 13  
 match() (rebasehelper.archive.ZipArchiveType class method), 14  
 match() (rebasehelper.build\_helper.BuildToolBase class method), 15  
 match() (rebasehelper.output\_tool.BaseOutputTool class method), 18  
 match() (rebasehelper.patch\_helper.GitPatchTool class method), 19  
 match() (rebasehelper.patch\_helper.PatchBase class method), 19  
 modify\_spec\_files\_section() (rebasehelper.specfile.SpecFile method), 23

## N

NAME (rebasehelper.output\_tool.BaseOutputTool attribute), 18  
 new\_repo (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 new\_rest\_sources (rebasehelper.application.Application attribute), 12  
 new\_sources (rebasehelper.application.Application attribute), 12  
 new\_sources (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 non\_interactive (rebasehelper.patch\_helper.GitPatchTool attribute), 19

## O

old\_repo (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 old\_rest\_sources (rebasehelper.application.Application attribute), 12  
 old\_sources (rebasehelper.application.Application attribute), 12  
 old\_sources (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 open() (rebasehelper.archive.ArchiveTypeBase class method), 13  
 open() (rebasehelper.archive.Bz2ArchiveType class method), 13  
 open() (rebasehelper.archive.GemPseudoArchiveType class method), 13  
 open() (rebasehelper.archive.TarGzArchiveType class method), 13

open() (rebasehelper.archive.TarXzArchiveType class method), 14  
 open() (rebasehelper.archive.ZipArchiveType class method), 14  
 option (rebasehelper.specfile.PatchObject attribute), 20  
 output\_data (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 OutputToolRunner (class in rebasehelper.output\_tool), 18

## P

parse\_log() (rebasehelper.build\_log\_analyzer.BuildLogAnalyzer class method), 16  
 patch() (rebasehelper.patch\_helper.Patcher method), 19  
 patch\_sources() (rebasehelper.application.Application method), 12  
 patch\_sources\_by\_prep\_script (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 PatchBase (class in rebasehelper.patch\_helper), 19  
 Patcher (class in rebasehelper.patch\_helper), 19  
 patches (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 patches (rebasehelper.specfile.SpecFile attribute), 23  
 PatchList (class in rebasehelper.specfile), 20  
 PatchObject (class in rebasehelper.specfile), 20  
 path (rebasehelper.specfile.PatchObject attribute), 20  
 path (rebasehelper.specfile.SpecFile attribute), 23  
 path() (rebasehelper.utils.TemporaryEnvironment method), 30  
 PathHelper (class in rebasehelper.utils), 28  
 prep\_section (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 prep\_section (rebasehelper.specfile.SpecFile attribute), 23  
 prepare() (rebasehelper.build\_helper.Builder method), 16  
 prepare() (rebasehelper.build\_helper.BuildToolBase class method), 15  
 prepare\_sources() (rebasehelper.application.Application method), 12  
 print\_cli\_summary() (rebasehelper.output\_tool.BaseOutputTool class method), 18  
 print\_patches\_cli() (rebasehelper.output\_tool.BaseOutputTool class method), 18  
 print\_patches\_section\_cli() (rebasehelper.output\_tool.BaseOutputTool class method), 18  
 print\_report\_file\_path() (rebasehelper.output\_tool.BaseOutputTool class method), 18  
 print\_summary() (rebasehelper.application.Application method), 12  
 print\_task\_info() (rebasehelper.application.Application method), 12

ProcessHelper (class in rebasehelper.utils), 28  
 progress() (rebasehelper.utils.DownloadHelper static method), 26

## R

rebase-helper command line option

- build-retries BUILD\_RETRIES, 9
- build-tasks OLD\_TASK,NEW\_TASK, 9
- builder-options=BUILDER\_OPTIONS, 9
- builds-nowait, 9
- buildtool {copr,rpmbuild,mock}, 9
- color {always,never,auto}, 9
- comparepkgs-only COMPAREPKGS\_DIR, 8
- disable-inapplicable-patches, 9
- non-interactive, 9
- not-download-sources, 9
- outputtool {text,json}, 9
- pkgcomparetool {rpmddiff,abipkgdiff,pkgdiff,csmock}, 9
- results-dir RESULTS\_DIR, 9
- version, 8
- versioneer {pypi,anitya}, 9
- b, -build-only, 8
- c, -continue, 9
- h, -help, 8
- p, -patch-only, 8
- v, -verbose, 8
- w, -keep-workspace, 9

SOURCES, 8  
 rebase\_spec\_file (rebasehelper.application.Application attribute), 12

rebase\_spec\_file\_path (rebasehelper.application.Application attribute), 12

rebased\_patches (rebasehelper.application.Application attribute), 12

rebased\_repo (rebasehelper.application.Application attribute), 12

rebasehelper.application (module), 11

rebasehelper.archive (module), 13

rebasehelper.build\_helper (module), 14

rebasehelper.build\_log\_analyzer (module), 16

rebasehelper.cli (module), 16

rebasehelper.exceptions (module), 17

rebasehelper.logger (module), 17

rebasehelper.output\_tool (module), 18

rebasehelper.patch\_helper (module), 19

rebasehelper.specfile (module), 20

rebasehelper.utils (module), 25

RebaseHelperError, 17

redefine\_release\_with\_macro() (rebasehelper.specfile.SpecFile method), 23

register\_archive\_type() (in module rebasehelper.archive), 14

register\_patch\_tool() (in module rebasehelper.patch\_helper), 20

removed\_patches (rebasehelper.specfile.SpecFile attribute), 23

report\_log\_file (rebasehelper.application.Application attribute), 12

result\_file (rebasehelper.application.Application attribute), 12

revert\_redefine\_release\_with\_macro() (rebasehelper.specfile.SpecFile method), 23

rpkg\_config\_dir (rebasehelper.utils.LookasideCacheHelper attribute), 27

rpm\_sections (rebasehelper.specfile.SpecFile attribute), 23

RpmbuildTemporaryEnvironment (class in rebasehelper.build\_helper), 16

{rpmddiff} RpmHelper (class in rebasehelper.utils), 29

run() (rebasehelper.application.Application method), 12

run() (rebasehelper.cli.CliHelper static method), 17

run() (rebasehelper.output\_tool.BaseOutputTool class method), 18

run() (rebasehelper.specfile.BaseSpecHook class method), 20

run\_download\_compare() (rebasehelper.application.Application method), 12

run\_mergetool() (rebasehelper.utils.GitHelper class method), 26

run\_output\_tools() (rebasehelper.output\_tool.OutputToolRunner method), 18

run\_package\_checkers() (rebasehelper.application.Application method), 12

run\_patch() (rebasehelper.patch\_helper.GitPatchTool class method), 19

run\_patch() (rebasehelper.patch\_helper.PatchBase class method), 19

run\_spec\_hooks() (rebasehelper.specfile.SpecHooksRunner method), 24

run\_subprocess() (rebasehelper.utils.ProcessHelper static method), 28

run\_subprocess\_cwd() (rebasehelper.utils.ProcessHelper static method), 28

run\_subprocess\_cwd\_env() (rebasehelper.utils.ProcessHelper static method), 28

run\_subprocess\_env() (rebasehelper.utils.ProcessHelper static method), 29

## S

save() (rebasehelper.specfile.SpecFile method), 23



- scratch\_url (rebasehelper.utils.KojiHelper attribute), 27  
 server (rebasehelper.utils.KojiHelper attribute), 27  
 session\_maker() (rebasehelper.utils.KojiHelper class method), 27  
 set\_extra\_version() (rebasehelper.specfile.SpecFile method), 23  
 set\_extra\_version\_separator() (rebasehelper.specfile.SpecFile method), 24  
 set\_path() (rebasehelper.specfile.PatchObject method), 20  
 set\_release\_number() (rebasehelper.specfile.SpecFile method), 24  
 set\_spec\_section() (rebasehelper.specfile.SpecFile method), 24  
 set\_upstream\_monitoring() (rebasehelper.application.Application method), 12  
 set\_version() (rebasehelper.specfile.SpecFile method), 24  
 set\_version\_using\_archive() (rebasehelper.specfile.SpecFile method), 24  
 setup() (rebasehelper.application.Application static method), 12  
 should\_use\_colors() (rebasehelper.utils.ConsoleHelper class method), 25  
 source\_dir (rebasehelper.patch\_helper.GitPatchTool attribute), 19  
 SourcePackageBuildError, 16  
 SOURCES  
     rebase-helper command line option, 8  
 sources (rebasehelper.specfile.SpecFile attribute), 24  
 spc (rebasehelper.specfile.SpecFile attribute), 24  
 spec\_content (rebasehelper.specfile.SpecFile attribute), 24  
 spec\_file (rebasehelper.application.Application attribute), 12  
 spec\_file\_path (rebasehelper.application.Application attribute), 12  
 SpecFile (class in rebasehelper.specfile), 20  
 SpecHooksRunner (class in rebasehelper.specfile), 24  
 split\_nevra() (rebasehelper.utils.RpmHelper class method), 30  
 split\_version\_string() (rebasehelper.specfile.SpecFile static method), 24
- T**
- TarBz2ArchiveType (class in rebasehelper.archive), 13  
 TarGzArchiveType (class in rebasehelper.archive), 13  
 TarXzArchiveType (class in rebasehelper.archive), 13  
 temp\_dir (rebasehelper.application.Application attribute), 12  
 TEMPDIR (rebasehelper.utils.TemporaryEnvironment attribute), 30  
 TEMPDIR\_BUILD (rebasehelper.build\_helper.RpmbuildTemporaryEnvironment attribute), 16  
 TEMPDIR\_BUILDROOT (rebasehelper.build\_helper.RpmbuildTemporaryEnvironment attribute), 16  
 TEMPDIR\_RESULTS (rebasehelper.build\_helper.BuildTemporaryEnvironment attribute), 14  
 TEMPDIR\_RPMBUILD (rebasehelper.build\_helper.RpmbuildTemporaryEnvironment attribute), 16  
 TEMPDIR\_RPMS (rebasehelper.build\_helper.RpmbuildTemporaryEnvironment attribute), 16  
 TEMPDIR\_SOURCES (rebasehelper.build\_helper.BuildTemporaryEnvironment attribute), 14  
 TEMPDIR\_SPEC (rebasehelper.build\_helper.BuildTemporaryEnvironment attribute), 14  
 TEMPDIR\_SPECS (rebasehelper.build\_helper.BuildTemporaryEnvironment attribute), 14  
 TEMPDIR\_SRPMS (rebasehelper.build\_helper.RpmbuildTemporaryEnvironment attribute), 16  
 TemporaryEnvironment (class in rebasehelper.utils), 30  
 TgzArchiveType (class in rebasehelper.archive), 14
- U**
- update\_changelog() (rebasehelper.specfile.SpecFile method), 24  
 update\_paths\_to\_patches() (rebasehelper.specfile.SpecFile method), 24  
 update\_setup\_dirname() (rebasehelper.specfile.SpecFile method), 24  
 upload\_srpm() (rebasehelper.utils.KojiHelper class method), 27  
 upstream\_monitoring (rebasehelper.application.Application attribute), 12  
 use\_colors (rebasehelper.utils.ConsoleHelper attribute), 25
- W**
- wait\_for\_task() (rebasehelper.build\_helper.Builder method), 16  
 wait\_for\_task() (rebasehelper.build\_helper.BuildToolBase class method), 15  
 watch\_build() (rebasehelper.utils.CoprHelper class method), 26  
 watch\_koji\_tasks() (rebasehelper.utils.KojiHelper class method), 27  
 write\_updated\_patches() (rebasehelper.specfile.SpecFile method), 24

## Z

ZipArchiveType (class in rebasehelper.archive), 14