
rebase-helper

Release 0.16.3

Jul 10, 2019

Contents

| | | |
|----------|-------------------------------|-----------|
| 1 | How to get it running? | 3 |
| 2 | How to use it? | 5 |
| 3 | What do I get from it? | 7 |
| 4 | How does it work? | 9 |
| 5 | Video presentation | 11 |
| 5.1 | User Guide | 12 |
| 5.2 | API | 16 |
| 5.3 | Change Log | 41 |
| | Python Module Index | 53 |
| | Index | 55 |

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

CHAPTER 1

How to get it running?

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

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

CHAPTER 2

How to use it?

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

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

```
$ rebase-helper 3.1.10
```

or you can pass source tarball filename instead:

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

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

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

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

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

What do I get from it?

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

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

How does it work?

The following steps describe a rebase process:

- **Preparation**

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

- **Getting sources**

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

- **Downstream patches**

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

- **Build**

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

- **Comparison**

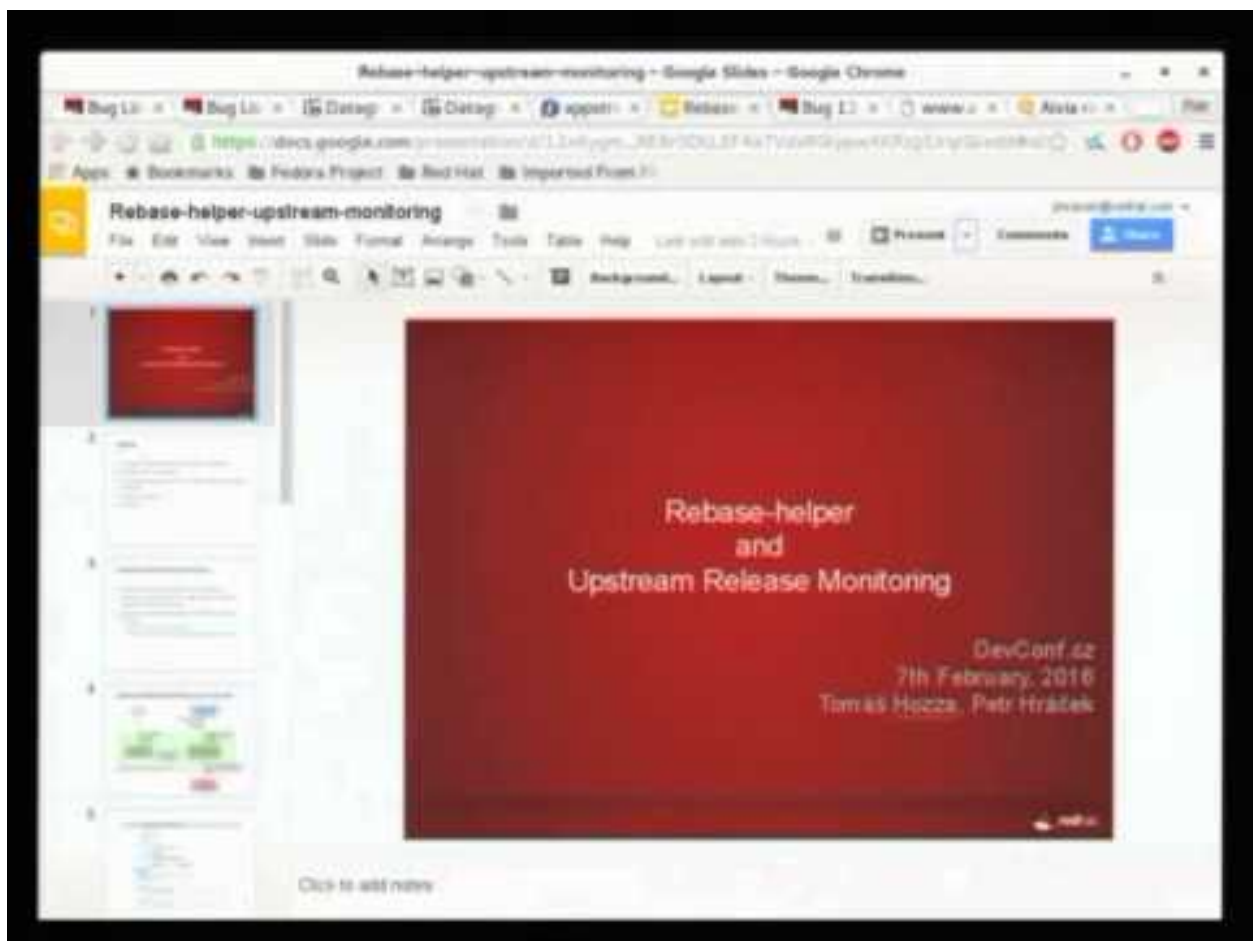
- multiple checker tools are run against both sets of packages and their output is stored in *rebase-helper-results/checkers* directory

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

CHAPTER 5

Video presentation

At DevConf.cz 2016, Tomáš Hozza, Petr Hráček presented on rebase-helper.



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 rpmbuild build tool |
| mock | mock | for mock build tool, optional |
| koji | koji | for koji build tool, optional |
| rpmdiff | rpmlint | for rpmdiff checker, optional |
| abipkgdiff | libabigail | for abipkgdiff checker, optional |
| pkgdiff | pkgdiff | for pkgdiff checker, optional |
| csmock | csmock | for csmock checker, optional |
| licensecheck | licensecheck | for licensecheck checker, optional |

5.1.2 Rebasing in container

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

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

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

5.1.3 Usage

SYNOPSIS

```
rebase-helper [-h] [--version] [-v] [--color {always,never,auto}]
  [--background {dark,light,auto}] [--results-dir RESULTS_DIR]
  [-p|-b|--comparepkgs-only COMPAREPKGS_DIR] [-c]
  [--buildtool {copr,rpmbuild,koji,mock}]
  [--srpm-buildtool {rpmbuild,mock}]
  [--pkgcomparetool {abipkgdiff,rpmdiff,licensecheck,pkgdiff,csmock}]
  [--outputtool {text,json}]
  [--versioneer {rubygems,pypi,anitya,hackage,cpan,npmjs}]
  [--versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmjs}]
  [--spec-hook-blacklist
  {replace-old-version,ruby-helper,commit-hash-updater,escape-macros,
  pypi-url-fix,paths-to-rpm-macros,typo-fix}]
  [--build-log-hook-blacklist {files}] [--non-interactive]
  [--favor-on-conflict {downstream,upstream,off}]
```



```

[--not-download-sources] [-w] [--apply-changes]
[--disable-inapplicable-patches] [--get-old-build-from-koji]
[--skip-version-check] [--update-sources] [--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]
[SOURCES]

```

DESCRIPTION

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

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

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

OPTIONS

Positional arguments

SOURCES

version number or filename of the new source archive

Optional arguments

-h, --help

show help message and exit

--version

show rebase-helper version and exit

-v, --verbose

be more verbose

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

colorize the output, defaults to *auto*

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

use color scheme for the given background, defaults to *auto*

--results-dir *RESULTS_DIR*

directory where rebase-helper output will be stored

-p, --patch-only

only apply patches

-b, --build-only

only build SRPMs and RPMs

--comparepkgs-only *COMPAREPKGS_DIR*

compare already built packages, *COMPAREPKGS_DIR* must be a directory with the following structure:
 <dir_name>/{old,new}/RPM

- c, --continue**
continue previously interrupted rebase
- buildtool** {copr, rpmbuild, koji, mock}
build tool to use, defaults to mock
- srpm-buildtool** {rpmbuild, mock}
SRPM build tool to use, defaults to rpmbuild
- pkgcomparetool** {abipkgdiff, rpmdiff, licensecheck, pkgdiff, csmock}
set of tools to use for package comparison, defaults to abipkgdiff, rpmdiff, licensecheck, pkgdiff if available
- outputtool** {text, json}
tool to use for formatting rebase output, defaults to text
- versioneer** {rubygems, pypi, anitya, hackage, cpan, npmjs}
tool to use for determining latest upstream version
- versioneer-blacklist** {rubygems, pypi, anitya, hackage, cpan, npmjs}
prevent specified versioneers from being run
- spec-hook-blacklist** {replace-old-version, ruby-helper, commit-hash-updater, escape-macros, py}
prevent specified spec hooks from being run
- build-log-hook-blacklist** {files}
prevent specified build log hooks from being run
- non-interactive**
do not interact with user
- favor-on-conflict** {downstream, upstream, off}
favor downstream or upstream changes when conflicts appear
- not-download-sources**
do not download sources
- w, --keep-workspace**
do not remove workspace directory after finishing
- apply-changes**
apply changes.patch after a successful rebase
- disable-inapplicable-patches**
disable inapplicable patches in rebased SPEC file
- get-old-build-from-koji**
do not build old sources, download latest build from Koji instead
- skip-version-check**
force rebase even if current version is newer than requested version
- update-sources**
update "sources" file and upload new sources to lookaside cache
- 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

5.1.4 Writing plugins

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

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

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

Example

Listing 1: my_spec_hook/__init__.py

```

from rebasehelper.specfile import BaseSpecHook

class MySpecHook(BaseSpecHook):

    NAME = 'MySpecHook'

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

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

        :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.insert(0, '# processed by %s\n' % cls.NAME)
rebase_spec_file.save()

```

Listing 2: setup.py

```

from setuptools import setup

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

```

5.2 API

5.2.1 Helpers

Console helper module

class `rebasehelper.helpers.console_helper.ConsoleHelper`

Class for interacting with the command line.

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

ContextManager for capturing stdout/stderr

static method `color_is_light` (*rgb, bit_width*)

Determines whether a color is light or dark.

Parameters

- **rgb** (*tuple*) – RGB tuple.
- **bit_width** – Number of bits defining the RGB.

Returns Whether a color is light or dark.

Return type `bool`

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

Prints colored output if possible.

Parameters

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

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

static detect_background ()

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

Returns Whether to use dark or light color scheme.

Return type str

static exchange_control_sequence (*query*, *timeout=0.05*)

Captures a response of a control sequence from STDIN.

Parameters

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

Returns Response of the terminal.

Return type str

static parse_rgb_device_specification (*specification*)

Parses RGB device specification.

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

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

Return type tuple

classmethod should_use_colors (*conf*)

Determines whether ANSI colors should be used for CLI output.

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

Returns Whether colors should be used.

Return type bool

use_colors = False

Copr helper module

class rebasehelper.helpers.copr_helper.CoprHelper

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

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

classmethod **download_build** (*client*, *build_id*, *destination*)

classmethod **get_build_status** (*client*, *build_id*)

classmethod **get_build_url** (*client*, *build_id*)

classmethod **get_client** ()

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

Download helper module

class `rebasehelper.helpers.download_helper.DownloadHelper`

Class for downloading files and performing HTTP requests.

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

Downloads a file from HTTP, HTTPS or FTP URL.

Parameters

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

static `progress(download_total, downloaded, start_time)`

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

Parameters

- `download_total` (*int*) – Total download size in bytes.
- `downloaded` (*int*) – Size of the already downloaded portion of a file in bytes.
- `start_time` (*float*) – Time when the download started in seconds since epoch.

static `request(url, **kwargs)`

Performs an HTTP request or an FTP RETR command.

Parameters

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

Returns Response object.

Return type `requests.Response`

Git helper module

class `rebasehelper.helpers.git_helper.GitHelper`

Class which operates with git repositories

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

`GIT_USER_NAME = 'rebase-helper'`

`classmethod get_email()`

`classmethod get_user()`

`classmethod run_mergetool(repo)`

Input helper module

class `rebasehelper.helpers.input_helper.InputHelper`

Class for command line interaction with the user.

static `get_message(message, default_yes=True, any_input=False)`

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

Parameters

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

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

Return type bool

Koji Helper module

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

classmethod **create_session** (*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, package full version.

Return type tuple

classmethod `get_task_url` (*session*, *task_id*)

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

Uploads SRPM to a Koji hub.

Parameters

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

Returns Remote path to the uploaded SRPM.

Return type str

Raises `RebaseHelperError` – If upload failed.

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

Waits for Koji tasks to finish and prints their states.

Parameters

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

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

Return type dict

Lookaside cache helper module

```
class rebasehelper.helpers.lookaside_cache_helper.LookasideCacheHelper
    Class for downloading files from Fedora/RHEL lookaside cache
    classmethod download(tool, basepath, package)
    rpkg_config_dir = '/etc/rpkg'
    classmethod update_sources(tool, basepath, package, old_sources, new_sources, upload=True)
```

Macro helper module

```
class rebasehelper.helpers.macro_helper.MacroHelper
    Class for working with RPM macros
    MACROS_WHITELIST = ['_bindir', '_datadir', '_includedir', '_infodir', '_initdir', '_libdir', '_libexecdir', '_libexecdir', '_libexecdir', '_libexecdir']
    static dump()
        Gets list of all defined macros.
        Returns All defined macros.
        Return type list
    static expand(s, default=None)
    static expand_macros(macros)
        Expands values of multiple macros.
        Parameters macros (list) – List of macros to be expanded, macros are represented as dicts.
        Returns List of macros with expanded values.
        Return type list
    static filter(macros, **kwargs)
        Finds all macros satisfying certain conditions.
        Parameters
            • macros (list) – Macros to be filtered.
            • **kwargs – Filters to be used.
        Returns Macros satisfying the conditions.
        Return type list
    static substitute_path_with_macros(path, macros)
        Substitutes parts of a path with macros.
        Parameters
            • path (str) – Path to be changed.
            • macros (list) – Macros which can be used as a substitution.
        Returns Path expressed using macros.
```

Return type str

Path helper module

class `rebasehelper.helpers.path_helper.PathHelper`

Class for performing path related tasks.

static file_available (*filename*)

Checks if the given file exists.

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

Returns Whether the file exists.

Return type bool

static find_all_files (*top_path*, *pattern*)

Recursively searches for all files matching the given pattern.

Parameters

- `top_path` (*str*) – Directory where to start the search.
- `pattern` (*str*) – Filename pattern.

Returns List containing absolute paths to all found files.

Return type list

static find_all_files_current_dir (*top_path*, *pattern*)

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

Parameters

- `top_path` (*str*) – Directory where to start the search.
- `pattern` (*str*) – Filename pattern.

Returns List containing absolute paths to all found files.

Return type list

static find_first_dir_with_file (*top_path*, *pattern*)

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

Parameters

- `top_path` (*str*) – Directory where to start the search.
- `pattern` (*str*) – Filename pattern.

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

Return type str

static find_first_file (*top_path*, *pattern*, *recursion_level=None*)

Recursively searches for a file that matches the given pattern.

Parameters

- `top_path` (*str*) – Directory where to start the search.
- `pattern` (*str*) – Filename pattern.
- `recursion_level` (*int*) – How deep in the directory tree the search can go.

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

Return type str

static `get_temp_dir()`

Creates a new temporary directory.

Returns Path to the created directory.

Return type str

Process helper module

class `rebasehelper.helpers.process_helper.ProcessHelper`

Class for executing subprocesses.

`DEV_NULL = '/dev/null'`

static `run_subprocess(cmd, input_file=None, output_file=None, 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, output_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*, *ignore_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.**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

classmethod decode (*data*)

static get_arches ()

Gets list of all known architectures

static get_header_from_rpm (*rpm_name*)

Gets an RPM header from the given RPM package.

Parameters **rpm_name** (*str*) – Path to the package.

Returns Header object obtained from the package.

Return type rpm.hdr

Returns

classmethod `get_info_from_rpm` (*rpm_name*, *info*)

Gets package name from an RPM file.

Parameters

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

Returns Package name obtained from the RPM file.

Return type str

Returns

static `install_build_dependencies` (*spec_path=None*, *assume_yes=False*)

Installs build dependencies of a package using dnf.

Parameters

- **spec_path** (*str*) – Absolute path to the SPEC file.
- **assume_yes** (*bool*) – Whether to automatically answer ‘yes’ to all questions.

Returns Exit code of the subprocess run.

Return type int

static `is_package_installed` (*pkg_name=None*)

Checks whether a package is installed.

Parameters **pkg_name** (*str*) – Name of the package.

Returns Whether the package is installed.

Return type bool

classmethod `parse_spec` (*path*, *flags=None*)

classmethod `split_nevra` (*s*)

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

5.2.2 Application module

class `rebasehelper.application.Application` (*cli_conf*, *execution_dir*, *results_dir*, *debug_log_file*)

`apply_changes` ()

`build_binary_packages` ()

Function calls build class for building packages

`build_source_packages` ()

`debug_log_file` = None

static `extract_archive` (*archive_path*, *destination*)

Extracts given archive into the destination and handle all exceptions.

Parameters

- **archive_path** – path to the archive to be extracted

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

Returns

static extract_sources (*archive_path, destination*)

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

generate_patch ()

Generates patch to the results_dir containing all needed changes for the rebased package version

get_all_log_files ()

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

Returns

get_checker_outputs ()

get_new_build_logs ()

get_rebased_patches ()

Function returns a list of patches either ‘’: [list_of_deleted_patches] :return:

get_rebasehelper_data ()

get_rpm_packages (*dirname*)

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

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

Returns

kwargs = {}

new_rest_sources = []

new_sources = ''

old_rest_sources = []

old_sources = ''

patch_sources (*sources*)

prepare_next_run (*results_dir*)

prepare_sources ()

Function prepares a sources.

Returns

print_summary (*exception=None*)

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

print_task_info (*builder*)

rebase_spec_file = None

rebase_spec_file_path = None

rebased_patches = {}

rebased_repo = None

report_log_file = None

result_file = ''

```

run ()
run_package_checkers (results_dir, **kwargs)
    Runs checkers on packages and stores results in a given directory.

    Parameters
        • results_dir (str) – Path to directory in which to store the results.
        • category (str) – checker type(SOURCE/SRPM/RPM)

    Returns None

static setup (cli_conf)
static setup_logging (results_dir)
    Adds file handlers of various verbosity to loggers.

    Parameters results_dir – Path to directory which results are stored in.

    Returns Path to debug log.

    Return type string

spec_file = None
spec_file_path = None
temp_dir = ''

```

5.2.3 Archive module

```

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

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

        Parameters path – Path where to extract the archive to.

        Returns

    classmethod get_supported_archives ()
        Return list of supported archive types

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

    EXTENSION = ''

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

        Parameters path – Path where to extract the archive to.

        Returns

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

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

class rebasehelper.archive.Bz2ArchiveType
    .bz2 archive type

```

EXTENSION = `' .bz2 '`

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

Parameters path – Path where to extract the archive to.

Returns

classmethod open (*filename=None*)

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

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

EXTENSION = `' .gem '`

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

Parameters path – Path where to extract the archive to.

Returns

classmethod open (*filename=None*)

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

class `rebasehelper.archive.TarArchiveType`
.tar archive type

EXTENSION = `' .tar '`

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

EXTENSION = `' .tar.bz2 '`

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

EXTENSION = `' .tar.gz '`

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

Parameters path – Path where to extract the archive to.

Returns

classmethod open (*filename=None*)

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

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

EXTENSION = `' .tar.xz '`

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

Parameters path – Path where to extract the archive to.

Returns

classmethod open (*filename=None*)

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


```

class rebasehelper.archive.TgzArchiveType
    .tgz archive type

    EXTENSION = '.tgz'

class rebasehelper.archive.ZipArchiveType
    .zip archive type

    EXTENSION = '.zip'

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

        Parameters path – Path where to extract the archive to.

        Returns

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

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

rebasehelper.archive.register_archive_type (archive)

```

5.2.4 Build helper module

```

exception rebasehelper.build_helper.BinaryPackageBuildError (*args, **kwargs)
    Error indicating failure to build Binary Package

class rebasehelper.build_helper.BuildHelper

    get_all_tools ()

    get_default_tool ()

    get_supported_tools ()

    get_tool (tool)

class rebasehelper.build_helper.BuildTemporaryEnvironment (sources, patches, spec,
                                                         results_dir)
    Class representing temporary environment.

    TEMPDIR_RESULTS = 'TEMPDIR_RESULTS'

    TEMPDIR_SOURCES = 'TEMPDIR_SOURCES'

    TEMPDIR_SPEC = 'TEMPDIR_SPEC'

    TEMPDIR_SPECS = 'TEMPDIR_SPECS'

class rebasehelper.build_helper.BuildToolBase
    Build tool base class.

    DEFAULT
        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_logs ()

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

classmethod get_task_info (*build_dict*)

Gets information about detached remote task

Parameters **build_dict** – build data

Returns task info

classmethod prepare (*spec, conf*)

Prepare for building.

Parameters **spec** – spec file object

classmethod wait_for_task (*build_dict, task_id, results_dir*)

Waits until specified task is finished

Parameters

- **build_dict** – build data
- **results_dir** – path to DIR where results should be stored

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

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

Class representing temporary environment for MockBuildTool.

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

Class representing temporary environment for RpmbuildBuildTool.

TEMPDIR_BUILD = 'TEMPDIR_BUILD'

```

TEMPDIR_BUILDROOT = 'TEMPDIR_BUILDROOT'
TEMPDIR_RPMBUILD = 'TEMPDIR_RPMBUILD'
TEMPDIR_RPMS = 'TEMPDIR_RPMS'
TEMPDIR_SRPMS = 'TEMPDIR_SRPMS'
class rebasehelper.build_helper.SRPMBuildHelper

    get_all_tools()
    get_default_tool()
    get_supported_tools()
    get_tool(tool)
class rebasehelper.build_helper.SRPMBuildToolBase
    SRPM build tool base class.

    DEFAULT
        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

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

    static get_srpm_builder_options(**kwargs)
exception rebasehelper.build_helper.SourcePackageBuildError(*args, **kwargs)
    Error indicating failure to build Source Package.

```

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

    static run()

```

5.2.6 Exception module

exception `rebasehelper.exceptions.CheckerNotFoundError`

Error indicating failure unable to find checker binary

exception `rebasehelper.exceptions.DownloadError`

Exception indicating that download of a file failed

exception `rebasehelper.exceptions.LookasideCacheError`

Exception indicating a problem accessing lookaside cache

exception `rebasehelper.exceptions.ParseError`

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

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

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_object, path, formatter_object=None, level=None*)

Adds file handler to the given logger.

Parameters

- **logger_object** (*logging.Logger*) – Logger object to add the handler to.
- **path** (*str*) – Path to a log file.
- **formatter_object** (*logging.Formatter*) – Formatter object used to format logged messages.

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

Returns Created file handler instance.

Return type logging.FileHandler

static add_stream_handler (*logger_object, level=None, formatter_object=None*)

Adds stream handler to the given logger.

Parameters

- **logger_object** (*logging.Logger*) – Logger object to add the handler to.
- **level** (*int*) – Severity threshold.
- **formatter_object** (*logging.Formatter*) – Formatter object used to format logged messages.

Returns Created stream handler instance.

Return type logging.StreamHandler

static get_basic_logger (*logger_name, level=10*)

Sets up a basic logger without any handler.

Parameters

- **logger_name** (*str*) – Logger name.
- **level** (*int*) – Severity threshold.

Returns Created logger instance.

Return type logging.Logger

5.2.8 Ouptut tool module

class rebasehelper.output_tool.**BaseOutputTool**

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

DEFAULT = **False**

EXTENSION = ''

classmethod get_report_path (*app*)

classmethod prepend_results_dir_name (**path_members*)

classmethod print_cli_summary (*app*)

Print report of the rebase

Parameters **app** – Application instance

classmethod print_important_checkers_output ()

Iterates over all checkers output to highlight important checkers warning

classmethod print_patches_cli ()

Print info about patches

classmethod print_patches_section_cli (*logger_method, patch_type*)

Print info about one of the patches key section

Parameters

- **logger_method** – method to be used for logging
- **patch_type** – string containing key for the patch_dict

```
classmethod print_report_file_path()
```

```
    Print path to the report file
```

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

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

```
    Class representing the process of running various output tools.
```

```
get_all_tools()
```

```
get_default_tool()
```

```
get_supported_tools()
```

```
run_output_tool(tool, logs=None, app=None)
```

```
    Runs specified output tool.
```

Parameters

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

5.2.9 Patch helper module

```
class rebasehelper.patch_helper.GitPatchTool
```

```
    Class for git command used for patching old and new sources
```

```
CMD = 'git'
```

```
classmethod apply_old_patches()
```

```
    Function applies a patch to a old/new sources
```

```
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)
```

```
diff_cls = None
```

```
classmethod extract_patch_name(message)
```

```
classmethod init_git(directory)
```

```
    Function initialize old and new Git repository
```

```
classmethod insert_patch_name(message, patch_name)
```

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

```
    Method checks whether it is usefull patch method
```

```
new_repo = None
```

```
new_sources = ''
```

```
non_interactive = False
```

```
old_repo = None
```

```
old_sources = ''
```

```
output_data = None
```

```
patches = []
```

classmethod `run_patch` (*old_dir, new_dir, rest_sources, patches, **kwargs*)

The function can be used for patching one directory against another

`source_dir = ''`

classmethod `strip_patch_name` (*diff, patch_name*)

class `rebasehelper.patch_helper.PatchBase`

Class used for using several patching command tools, ... Each method should overwrite method like `run_check`

classmethod `match` (*cmd=None*)

Method checks whether it is usefull patch method

classmethod `run_patch` (*old_dir, new_dir, rest_sources, patches, **kwargs*)

Method will check all patches in relevant package

class `rebasehelper.patch_helper.Patcher` (*tool=None*)

Class representing a process of applying and generating rebased patch using specific tool.

patch (*old_dir, new_dir, rest_sources, patches, **kwargs*)

Apply patches and generate rebased patches if needed

Parameters

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

Returns

`rebasehelper.patch_helper.register_patch_tool` (*patch_tool*)

5.2.10 Specfile module

class `rebasehelper.specfile.BaseSpecHook`

Base class for a spec hook

`CATEGORIES = None`

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

Runs a spec hook.

Parameters

- `spec_file` – Original spec file object
- `rebase_spec_file` – Rebased spec file object
- `kwargs` – Keyword arguments from Application instance

class `rebasehelper.specfile.PatchList`

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

Class represents set of information about patches

`get_index()`

`get_patch_name()`

```
get_path()
get_strip()
git_generated = ''
index = ''
path = ''
set_path(new_path)
strip = ''
```

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

Class representing content of a SPEC file.

```
SECTION_HEADERS = ['%package', '%prep', '%build', '%install', '%check', '%clean', '%pr
```

```
replace_section (name, content)
```

Replaces content of a section.

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

Parameters

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

Returns False if section was not found else True.

Return type bool

```
section (name)
```

Gets content of a section.

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

Parameters **name** (*str*) – Section name.

Returns Section content as a list of lines.

Return type list

```
class rebasehelper.specfile.SpecFile (path, sources_location="")
```

Class representing a SPEC file

```
category = None
```

```
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 = False
```

```
download_remote_sources ()
```

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

Returns None

```
extra_version = None
```


static extract_version_from_archive_name (*archive_path*, *source_string*="")

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

Parameters

- **archive_path** – archive name or path with archive name from which to extract the version
- **source_string** – Source string from SPEC file used to construct version extraction regex

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

find_archive_target_in_prep (*archive*)

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

Parameters **archive** – Path to archive

Returns Target path relative to builddir or None if not determined

get_NVR ()

get_applied_patches ()

Method returns list of all applied patches.

Returns list of PatchObject

get_archive ()

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

Returns basename of first Source in SPEC file

Return type str

get_epoch_number ()

Method for getting epoch of the package

Returns

get_extra_version ()

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

Returns String

get_extra_version_separator ()

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

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

Return type str

get_full_version ()

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

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

Return type str

get_main_files_section ()

Finds the exact name of the main %files section.

Returns Name of the main files section.

Return type str

get_new_log (*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 unapplied patches.

Returns list of PatchObject

get_package_name ()

Function returns a package name

Returns

get_patches ()

Method returns list of all applied and not applied patches

Returns list of PatchObject

get_path ()

Return only spec file path

Returns

get_prep_section ()

Function returns whole prep section

get_release ()

Method for getting full release string of the package

Returns

get_release_number ()

Method for getting the release of the package

Returns

get_requires ()

Function returns a package requirements

Returns

get_setup_dirname ()

Get dirname from %setup or %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

static get_subpackage_name (*files_section*)

Gets subpackage name based on the %files section.

get_version ()

Method returns the version

Returns**hdr = None****is_test_suite_enabled()**

Returns whether test suite is enabled during the build time

Returns True if enabled or False if not**patches = None****path = ''****prep_section = []****redefine_release_with_macro** (*macro*)

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

Parameters **macro** –**Returns****removed_patches = []****revert_redefine_release_with_macro** (*macro*)

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

Parameters **macro** –**Returns****save()**

Save changes made to the spec_content to the disc and update internal variables

set_extra_version (*extra_version*)

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

Parameters **extra_version** – the extra version string, if any (e.g. 'b1', 'rc2', ...)**Returns** None**set_extra_version_separator** (*separator*)

Set the string that separates the version and extra version

Parameters **separator** –**Returns****set_release_number** (*release*)

Method to set release number

Parameters **release** –**Returns****set_tag** (*tag, value, preserve_macros=False*)

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

set_version (*version*)

Method to update the version in the SPEC file

Parameters **version** – string with new version**Returns** None

set_version_using_archive (*archive_path*)

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

Parameters *archive_path* –

Returns

sources = None

spc = None

spec_content = None

static split_version_string (*version_string*="")

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

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

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

update_changelog (*changelog_entry*)

Inserts a new entry into the changelog and saves the SpecFile.

Parameters *changelog_entry* (*str*) – Message to use in the entry.

update_paths_to_patches ()

update_setup_dirname (*dirname*)

Update %setup or %autosetup dirname argument if needed

Parameters *dirname* – new dirname to be used

write_updated_patches (*patches*, *disable_inapplicable*)

Function writes the patches to -rebase.spec file

class rebasehelper.specfile.SpecHooksRunner

Class representing the process of running various spec file hooks.

get_all_spec_hooks ()

get_available_spec_hooks ()

run_spec_hooks (*spec_file*, *rebase_spec_file*, ***kwargs*)

Runs all non-blacklisted spec hooks.

Parameters

- **spec_file** – Original spec file object
- **rebase_spec_file** – Rebased spec file object
- **kwargs** – Keyword arguments from Application instance

rebasehelper.specfile.**get_rebase_name** (*dir_name*, *name*)

Function returns a name in results directory

Parameters

- **dir_name** –
- **name** –

Returns full path to results dir with name

5.2.11 Temporary environment module

class `rebasehelper.temporary_environment.TemporaryEnvironment` (*exit_callback=None*)
 Class representing a temporary environment (directory) that can be used as a workspace.

Works as a context manager.

TEMPDIR = 'TEMPDIR'

env ()

Gets a copy of `_env` dictionary.

Returns Copy of `_env` dictionary.

Return type dict

path ()

Gets the path to the temporary environment.

Returns Absolute path to the temporary environment.

Return type str

5.3 Change Log

5.3.1 [Unreleased]

5.3.2 [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

5.3.3 [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

5.3.4 [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

5.3.5 [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

5.3.6 [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

5.3.7 [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

5.3.8 [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

5.3.9 [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

5.3.10 [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

5.3.11 [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

5.3.12 [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

5.3.13 [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

5.3.14 [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

5.3.15 [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

5.3.16 [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

5.3.17 [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

5.3.18 [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

5.3.19 [0.7.1] - 2016-02-22

Added

- Two new command line options used by upstream monitoring

Fixed

- **fedpkg** reimplementation

5.3.20 [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

5.3.21 [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

5.3.22 [0.6.1] - 2015-10-30

Added

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

5.3.23 [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`

5.3.24 [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

5.3.25 [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

5.3.26 [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

5.3.27 [0.3.0]

Added

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

5.3.28 [0.2.0]

Added

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

5.3.29 [0.1.0]

Added

- Initial classes
- CLI interface

r

rebasehelper.application, 25
rebasehelper.archive, 27
rebasehelper.build_helper, 29
rebasehelper.cli, 31
rebasehelper.exceptions, 32
rebasehelper.helpers.console_helper, 16
rebasehelper.helpers.copr_helper, 17
rebasehelper.helpers.download_helper,
18
rebasehelper.helpers.git_helper, 18
rebasehelper.helpers.input_helper, 18
rebasehelper.helpers.koji_helper, 19
rebasehelper.helpers.lookaside_cache_helper,
21
rebasehelper.helpers.macro_helper, 21
rebasehelper.helpers.path_helper, 22
rebasehelper.helpers.process_helper, 23
rebasehelper.helpers.rpm_helper, 24
rebasehelper.logger, 32
rebasehelper.output_tool, 33
rebasehelper.patch_helper, 34
rebasehelper.specfile, 35
rebasehelper.temporary_environment, 41

Symbols

- apply-changes
 - rebase-helper command line option, 14
- background {dark,light,auto}
 - rebase-helper command line option, 13
- build-log-hook-blacklist {files}
 - rebase-helper command line option, 14
- build-tasks OLD_TASK,NEW_TASK
 - rebase-helper command line option, 15
- builder-options BUILDER_OPTIONS
 - rebase-helper command line option, 15
- builds-nowait
 - rebase-helper command line option, 14
- buildtool {copr,rpmbuild,koji,mock}
 - rebase-helper command line option, 14
- changelog-entry CHANGELOG_ENTRY
 - rebase-helper command line option, 15
- color {always,never,auto}
 - rebase-helper command line option, 13
- comparepkgs-only COMPAREPKGS_DIR
 - rebase-helper command line option, 13
- config-file CONFIG_FILE
 - rebase-helper command line option, 15
- disable-inapplicable-patches
 - rebase-helper command line option, 14
- favor-on-conflict
 - {downstream,upstream,off}
 - rebase-helper command line option, 14
- force-build-log-hooks
 - rebase-helper command line option, 14
- get-old-build-from-koji
 - rebase-helper command line option, 14
- non-interactive
 - rebase-helper command line option, 14
- not-download-sources
 - rebase-helper command line option, 14
- outputtool {text,json}
 - rebase-helper command line option, 14
- pkgcomparetool {abipkgdiff,rpmdiff,licensecheck,pl}
 - rebase-helper command line option, 14
- results-dir RESULTS_DIR
 - rebase-helper command line option, 13
- skip-upload
 - rebase-helper command line option, 14
- skip-version-check
 - rebase-helper command line option, 14
- spec-hook-blacklist
 - {replace-old-version,ruby-helper,commit-hash}
 - rebase-helper command line option, 14
- srpm-builder-options
 - SRPM_BUILDER_OPTIONS
 - rebase-helper command line option, 15
- srpm-buildtool {rpmbuild,mock}
 - rebase-helper command line option, 14

-update-sources
 rebase-helper command line option, 14

-version
 rebase-helper command line option, 13

-versioneer {rubygems, pypi, anitya, hackage, cpan, npmjs}
 rebase-helper command line option, 14

-versioneer-blacklist
 {rubygems, pypi, anitya, hackage, cpan, npmjs}
 rebase-helper command line option, 14

-b, -build-only
 rebase-helper command line option, 13

-c, -continue
 rebase-helper command line option, 14

-h, -help
 rebase-helper command line option, 13

-p, -patch-only
 rebase-helper command line option, 13

-v, -verbose
 rebase-helper command line option, 13

-w, -keep-workspace
 rebase-helper command line option, 14

method), 34

ARCHES (*rebasehelper.helpers.rpm_helper.RpmHelper attribute*), 24

Archive (*class in rebasehelper.archive*), 27

ArchiveTypeBase (*class in rebasehelper.archive*), 27

B

A

ACCEPTS_OPTIONS (*rebase-helper.build_helper.BuildToolBase attribute*), 29, 30

add_file_handler() (*rebase-helper.logger.LoggerHelper static method*), 32

add_stream_handler() (*rebase-helper.logger.LoggerHelper static method*), 33

all_packages_installed() (*rebase-helper.helpers.rpm_helper.RpmHelper static method*), 24

Application (*class in rebasehelper.application*), 25

apply_changes() (*rebase-helper.application.Application method*), 25

apply_old_patches() (*rebase-helper.patch_helper.GitPatchTool class method*), 34

apply_patch() (*rebase-helper.patch_helper.GitPatchTool class*

BaseOutputTool (*class in rebasehelper.output_tool*), 33

BaseSpecHook (*class in rebasehelper.specfile*), 35

BinaryPackageBuildError, 29

build() (*rebasehelper.build_helper.BuildToolBase class method*), 30

build() (*rebasehelper.build_helper.SRPMBuildToolBase class method*), 31

build() (*rebasehelper.helpers.copr_helper.CoprHelper class method*), 17

build_binary_packages() (*rebase-helper.application.Application method*), 25

build_parser() (*rebasehelper.cli.CLI static method*), 31

build_source_packages() (*rebase-helper.application.Application method*), 25

BuildHelper (*class in rebasehelper.build_helper*), 29

BuildTemporaryEnvironment (*class in rebase-helper.build_helper*), 29

BuildToolBase (*class in rebasehelper.build_helper*), 29

Bz2ArchiveType (*class in rebasehelper.archive*), 27

C

CATEGORIES (*rebasehelper.specfile.BaseSpecHook attribute*), 35

category (*rebasehelper.specfile.SpecFile attribute*), 36

CheckerNotFoundError, 32

CLI (*class in rebasehelper.cli*), 31

CliHelper (*class in rebasehelper.cli*), 31

CMD (*rebasehelper.patch_helper.GitPatchTool attribute*), 34

color_is_light() (*rebase-helper.helpers.console_helper.ConsoleHelper static method*), 16

ColorizingStreamHandler (*class in rebase-helper.logger*), 32

colors (*rebasehelper.logger.ColorizingStreamHandler attribute*), 32

ConsoleHelper (*class in rebase-helper.helpers.console_helper*), 16

ConsoleHelper.Capturer (*class in rebase-helper.helpers.console_helper*), 16

CoprHelper (*class in rebase-helper.helpers.copr_helper*), 17

- copy () (*rebasehelper.specfile.SpecFile* method), 36
- cprint () (*rebasehelper.helpers.console_helper.ConsoleHelper* class method), 16
- create_project () (*rebasehelper.helpers.copr_helper.CoprHelper* class method), 17
- create_session () (*rebasehelper.helpers.koji_helper.KojiHelper* class method), 19
- CREATES_TASKS (*rebasehelper.build_helper.BuildToolBase* attribute), 29, 30
- CustomLogger (class in *rebasehelper.logger*), 32
- ## D
- debug_log_file (*rebasehelper.application.Application* attribute), 25
- decode () (*rebasehelper.helpers.rpm_helper.RpmHelper* class method), 24
- decorate_patch_name () (*rebasehelper.patch_helper.GitPatchTool* static method), 34
- DEFAULT (*rebasehelper.build_helper.BuildToolBase* attribute), 29, 30
- DEFAULT (*rebasehelper.build_helper.SRPMBuildToolBase* attribute), 31
- DEFAULT (*rebasehelper.output_tool.BaseOutputTool* attribute), 33
- detect_background () (*rebasehelper.helpers.console_helper.ConsoleHelper* static method), 17
- DEV_NULL (*rebasehelper.helpers.process_helper.ProcessHelper* attribute), 23
- diff_cls (*rebasehelper.patch_helper.GitPatchTool* attribute), 34
- display_task_results () (*rebasehelper.helpers.koji_helper.KojiHelper* class method), 19
- download (*rebasehelper.specfile.SpecFile* attribute), 36
- download () (*rebasehelper.helpers.lookaside_cache_helper.LookasideCacheHelper* class method), 21
- download_build () (*rebasehelper.helpers.copr_helper.CoprHelper* class method), 17
- download_build () (*rebasehelper.helpers.koji_helper.KojiHelper* class method), 19
- download_file () (*rebasehelper.helpers.download_helper.DownloadHelper* static method), 18
- download_remote_sources () (*rebasehelper.specfile.SpecFile* method), 36
- download_task_results () (*rebasehelper.helpers.koji_helper.KojiHelper* class method), 19
- DownloadError, 32
- DownloadHelper (class in *rebasehelper.helpers.download_helper*), 18
- dump () (*rebasehelper.helpers.macro_helper.MacroHelper* static method), 21
- ## E
- emit () (*rebasehelper.logger.ColorizingStreamHandler* method), 32
- env () (*rebasehelper.temporary_environment.TemporaryEnvironment* method), 41
- environment variable PACKAGE, 12
REPOSITORY, 12
- exchange_control_sequence () (*rebasehelper.helpers.console_helper.ConsoleHelper* static method), 17
- expand () (*rebasehelper.helpers.macro_helper.MacroHelper* static method), 21
- expand_macros () (*rebasehelper.helpers.macro_helper.MacroHelper* static method), 21
- EXTENSION (*rebasehelper.archive.ArchiveTypeBase* attribute), 27
- EXTENSION (*rebasehelper.archive.Bz2ArchiveType* attribute), 27
- EXTENSION (*rebasehelper.archive.GemPseudoArchiveType* attribute), 28
- EXTENSION (*rebasehelper.archive.TarArchiveType* attribute), 28
- EXTENSION (*rebasehelper.archive.TarBz2ArchiveType* attribute), 28
- EXTENSION (*rebasehelper.archive.TarGzArchiveType* attribute), 28
- EXTENSION (*rebasehelper.archive.TarXzArchiveType* attribute), 28
- EXTENSION (*rebasehelper.archive.TgzArchiveType* attribute), 29
- EXTENSION (*rebasehelper.archive.ZipArchiveType* attribute), 29
- EXTENSION (*rebasehelper.output_tool.BaseOutputTool* attribute), 33
- extra_version (*rebasehelper.specfile.SpecFile* attribute), 36
- extract () (*rebasehelper.archive.ArchiveTypeBase* class method), 27
- extract () (*rebasehelper.archive.Bz2ArchiveType* class method), 28
- extract () (*rebasehelper.archive.GemPseudoArchiveType* class method), 28

| | |
|--|---|
| <code>extract()</code> (<i>rebasehelper.archive.TarGzArchiveType</i> class method), 28 | <code>get_all_spec_hooks()</code> (<i>rebase-helper.specfile.SpecHooksRunner</i> class method), 40 |
| <code>extract()</code> (<i>rebasehelper.archive.TarXzArchiveType</i> class method), 28 | <code>get_all_tools()</code> (<i>rebase-helper.build_helper.BuildHelper</i> class method), 29 |
| <code>extract()</code> (<i>rebasehelper.archive.ZipArchiveType</i> class method), 29 | <code>get_all_tools()</code> (<i>rebase-helper.build_helper.SRPMBuildHelper</i> class method), 31 |
| <code>extract_archive()</code> (<i>rebase-helper.application.Application</i> static method), 25 | <code>get_all_tools()</code> (<i>rebase-helper.output_tool.OutputToolRunner</i> class method), 34 |
| <code>extract_archive()</code> (<i>rebasehelper.archive.Archive</i> class method), 27 | <code>get_applied_patches()</code> (<i>rebase-helper.specfile.SpecFile</i> class method), 37 |
| <code>extract_patch_name()</code> (<i>rebase-helper.patch_helper.GitPatchTool</i> class method), 34 | <code>get_arches()</code> (<i>rebase-helper.helpers.rpm_helper.RpmHelper</i> static method), 24 |
| <code>extract_sources()</code> (<i>rebase-helper.application.Application</i> static method), 26 | <code>get_archive()</code> (<i>rebasehelper.specfile.SpecFile</i> class method), 37 |
| <code>extract_version_from_archive_name()</code> (<i>rebasehelper.specfile.SpecFile</i> static method), 36 | <code>get_available_spec_hooks()</code> (<i>rebase-helper.specfile.SpecHooksRunner</i> class method), 40 |
| F | <code>get_basic_logger()</code> (<i>rebase-helper.logger.LoggerHelper</i> static method), 33 |
| <code>file_available()</code> (<i>rebase-helper.helpers.path_helper.PathHelper</i> static method), 22 | <code>get_build()</code> (<i>rebase-helper.helpers.koji_helper.KojiHelper</i> class method), 20 |
| <code>filter()</code> (<i>rebasehelper.helpers.macro_helper.MacroHelper</i> static method), 21 | <code>get_build_status()</code> (<i>rebase-helper.helpers.copr_helper.CoprHelper</i> class method), 17 |
| <code>find_all_files()</code> (<i>rebase-helper.helpers.path_helper.PathHelper</i> static method), 22 | <code>get_build_url()</code> (<i>rebase-helper.helpers.copr_helper.CoprHelper</i> class method), 17 |
| <code>find_all_files_current_dir()</code> (<i>rebase-helper.helpers.path_helper.PathHelper</i> static method), 22 | <code>get_builder_options()</code> (<i>rebase-helper.build_helper.BuildToolBase</i> static method), 30 |
| <code>find_archive_target_in_prep()</code> (<i>rebase-helper.specfile.SpecFile</i> class method), 37 | <code>get_checker_outputs()</code> (<i>rebase-helper.application.Application</i> class method), 26 |
| <code>find_first_dir_with_file()</code> (<i>rebase-helper.helpers.path_helper.PathHelper</i> static method), 22 | <code>get_client()</code> (<i>rebase-helper.helpers.copr_helper.CoprHelper</i> class method), 17 |
| <code>find_first_file()</code> (<i>rebase-helper.helpers.path_helper.PathHelper</i> static method), 22 | <code>get_default_tool()</code> (<i>rebase-helper.build_helper.BuildHelper</i> class method), 29 |
| <code>functional</code> (<i>rebase-helper.helpers.koji_helper.KojiHelper</i> attribute), 20 | <code>get_default_tool()</code> (<i>rebase-helper.build_helper.SRPMBuildHelper</i> class method), 31 |
| G | <code>get_default_tool()</code> (<i>rebase-helper.output_tool.OutputToolRunner</i> class method), 34 |
| <code>GemPseudoArchiveType</code> (class in <i>rebase-helper.archive</i>), 28 | <code>get_detached_task()</code> (<i>rebase-helper.build_helper.BuildToolBase</i> class method), 30 |
| <code>generate_patch()</code> (<i>rebase-helper.application.Application</i> class method), 26 | |
| <code>get_all_log_files()</code> (<i>rebase-helper.application.Application</i> class method), 26 | |

method), 30
get_email() (*rebase-helper.helpers.git_helper.GitHelper class method*), 18
get_epoch_number() (*rebase-helper.specfile.SpecFile method*), 37
get_extra_version() (*rebase-helper.specfile.SpecFile method*), 37
get_extra_version_separator() (*rebase-helper.specfile.SpecFile method*), 37
get_full_version() (*rebase-helper.specfile.SpecFile method*), 37
get_header_from_rpm() (*rebase-helper.helpers.rpm_helper.RpmHelper static method*), 24
get_index() (*rebasehelper.specfile.PatchObject method*), 35
get_info_from_rpm() (*rebase-helper.helpers.rpm_helper.RpmHelper class method*), 25
get_latest_build() (*rebase-helper.helpers.koji_helper.KojiHelper class method*), 20
get_logs() (*rebasehelper.build_helper.BuildToolBase class method*), 30
get_logs() (*rebase-helper.build_helper.SRPMBuildToolBase class method*), 31
get_main_files_section() (*rebase-helper.specfile.SpecFile method*), 37
get_message() (*rebase-helper.helpers.input_helper.InputHelper static method*), 18
get_new_build_logs() (*rebase-helper.application.Application method*), 26
get_new_log() (*rebasehelper.specfile.SpecFile method*), 38
get_not_used_patches() (*rebase-helper.specfile.SpecFile method*), 38
get_NVR() (*rebasehelper.specfile.SpecFile method*), 37
get_old_build_info() (*rebase-helper.helpers.koji_helper.KojiHelper class method*), 20
get_package_name() (*rebase-helper.specfile.SpecFile method*), 38
get_patch_name() (*rebase-helper.specfile.PatchObject method*), 35
get_patches() (*rebasehelper.specfile.SpecFile method*), 38
get_path() (*rebasehelper.specfile.PatchObject method*), 35
get_path() (*rebasehelper.specfile.SpecFile method*), 38
get_prep_section() (*rebase-helper.specfile.SpecFile method*), 38
get_rebase_name() (*in module rebase-helper.specfile*), 40
get_rebased_patches() (*rebase-helper.application.Application method*), 26
get_rebasehelper_data() (*rebase-helper.application.Application method*), 26
get_release() (*rebasehelper.specfile.SpecFile method*), 38
get_release_number() (*rebase-helper.specfile.SpecFile method*), 38
get_report_path() (*rebase-helper.output_tool.BaseOutputTool class method*), 33
get_requires() (*rebasehelper.specfile.SpecFile method*), 38
get_rpm_packages() (*rebase-helper.application.Application method*), 26
get_setup_dirname() (*rebase-helper.specfile.SpecFile method*), 38
get_sources() (*rebasehelper.specfile.SpecFile method*), 38
get_srpm_builder_options() (*rebase-helper.build_helper.SRPMBuildToolBase static method*), 31
get_strip() (*rebasehelper.specfile.PatchObject method*), 36
get_subpackage_name() (*rebase-helper.specfile.SpecFile static method*), 38
get_supported_archives() (*rebase-helper.archive.Archive class method*), 27
get_supported_tools() (*rebase-helper.build_helper.BuildHelper method*), 29
get_supported_tools() (*rebase-helper.build_helper.SRPMBuildHelper method*), 31
get_supported_tools() (*rebase-helper.output_tool.OutputToolRunner method*), 34
get_task_info() (*rebase-helper.build_helper.BuildToolBase class method*), 30
get_task_url() (*rebase-helper.helpers.koji_helper.KojiHelper class method*), 20
get_temp_dir() (*rebase-helper.helpers.path_helper.PathHelper static method*), 23
get_tool() (*rebasehelper.build_helper.BuildHelper*

method), 29
 get_tool() (rebase-helper.build_helper.SRPMBuildHelper method), 31
 get_user() (rebase-helper.helpers.git_helper.GitHelper class method), 18
 get_version() (rebasehelper.specfile.SpecFile method), 38
 git_generated (rebasehelper.specfile.PatchObject attribute), 36
 GIT_USER_EMAIL (rebase-helper.helpers.git_helper.GitHelper attribute), 18
 GIT_USER_NAME (rebase-helper.helpers.git_helper.GitHelper attribute), 18
 GitHelper (class in rebasehelper.helpers.git_helper), 18
 GitPatchTool (class in rebasehelper.patch_helper), 34

H

hdr (rebasehelper.specfile.SpecFile attribute), 39
 HEADING (rebasehelper.logger.CustomLogger attribute), 32

I

IMPORTANT (rebasehelper.logger.CustomLogger attribute), 32
 index (rebasehelper.specfile.PatchObject attribute), 36
 init_git() (rebasehelper.patch_helper.GitPatchTool class method), 34
 InputHelper (class in rebase-helper.helpers.input_helper), 18
 insert_patch_name() (rebase-helper.patch_helper.GitPatchTool class method), 34
 install_build_dependencies() (rebase-helper.helpers.rpm_helper.RpmHelper static method), 25
 is_package_installed() (rebase-helper.helpers.rpm_helper.RpmHelper static method), 25
 is_test_suite_enabled() (rebase-helper.specfile.SpecFile method), 39

K

KojiHelper (class in rebase-helper.helpers.koji_helper), 19
 kwargs (rebasehelper.application.Application attribute), 26

L

LoggerHelper (class in rebasehelper.logger), 32
 LookasideCacheError, 32
 LookasideCacheHelper (class in rebase-helper.helpers.lookaside_cache_helper), 21

M

MacroHelper (class in rebase-helper.helpers.macro_helper), 21
 MACROS_WHITELIST (rebase-helper.helpers.macro_helper.MacroHelper attribute), 21
 match() (rebasehelper.archive.ArchiveTypeBase class method), 27
 match() (rebasehelper.archive.ZipArchiveType class method), 29
 match() (rebasehelper.patch_helper.GitPatchTool class method), 34
 match() (rebasehelper.patch_helper.PatchBase class method), 35
 MockTemporaryEnvironment (class in rebase-helper.build_helper), 30

N

new_repo (rebasehelper.patch_helper.GitPatchTool attribute), 34
 new_rest_sources (rebase-helper.application.Application attribute), 26
 new_sources (rebasehelper.application.Application attribute), 26
 new_sources (rebase-helper.patch_helper.GitPatchTool attribute), 34
 non_interactive (rebase-helper.patch_helper.GitPatchTool attribute), 34

O

old_repo (rebasehelper.patch_helper.GitPatchTool attribute), 34
 old_rest_sources (rebase-helper.application.Application attribute), 26
 old_sources (rebasehelper.application.Application attribute), 26
 old_sources (rebase-helper.patch_helper.GitPatchTool attribute), 34
 open() (rebasehelper.archive.ArchiveTypeBase class method), 27
 open() (rebasehelper.archive.Bz2ArchiveType class method), 28

- open() (*rebasehelper.archive.GemPseudoArchiveType* class method), 28
- open() (*rebasehelper.archive.TarGzArchiveType* class method), 28
- open() (*rebasehelper.archive.TarXzArchiveType* class method), 28
- open() (*rebasehelper.archive.ZipArchiveType* class method), 29
- output_data (*rebasehelper.patch_helper.GitPatchTool* attribute), 34
- OutputToolRunner (class in *rebasehelper.output_tool*), 34
- ## P
- PACKAGE, 12
- parse_rgb_device_specification() (*rebasehelper.helpers.console_helper.ConsoleHelper* static method), 17
- parse_spec() (*rebasehelper.helpers.rpm_helper.RpmHelper* class method), 25
- ParseError, 32
- patch() (*rebasehelper.patch_helper.Patcher* method), 35
- patch_sources() (*rebasehelper.application.Application* method), 26
- PatchBase (class in *rebasehelper.patch_helper*), 35
- Patcher (class in *rebasehelper.patch_helper*), 35
- patches (*rebasehelper.patch_helper.GitPatchTool* attribute), 34
- patches (*rebasehelper.specfile.SpecFile* attribute), 39
- PatchList (class in *rebasehelper.specfile*), 35
- PatchObject (class in *rebasehelper.specfile*), 35
- path (*rebasehelper.specfile.PatchObject* attribute), 36
- path (*rebasehelper.specfile.SpecFile* attribute), 39
- path() (*rebasehelper.temporary_environment.TemporaryEnvironment* method), 41
- PathHelper (class in *rebasehelper.helpers.path_helper*), 22
- prep_section (*rebasehelper.specfile.SpecFile* attribute), 39
- prepare() (*rebasehelper.build_helper.BuildToolBase* class method), 30
- prepare_next_run() (*rebasehelper.application.Application* method), 26
- prepare_sources() (*rebasehelper.application.Application* method), 26
- prepend_results_dir_name() (*rebasehelper.output_tool.BaseOutputTool* class method), 33
- print_cli_summary() (*rebasehelper.output_tool.BaseOutputTool* class method), 33
- print_important_checkers_output() (*rebasehelper.output_tool.BaseOutputTool* class method), 33
- print_patches_cli() (*rebasehelper.output_tool.BaseOutputTool* class method), 33
- print_patches_section_cli() (*rebasehelper.output_tool.BaseOutputTool* class method), 33
- print_report_file_path() (*rebasehelper.output_tool.BaseOutputTool* class method), 34
- print_summary() (*rebasehelper.application.Application* method), 26
- print_task_info() (*rebasehelper.application.Application* method), 26
- ProcessHelper (class in *rebasehelper.helpers.process_helper*), 23
- progress() (*rebasehelper.helpers.download_helper.DownloadHelper* static method), 18
- ## R
- rebase-helper command line option
- apply-changes, 14
 - background {dark, light, auto}, 13
 - build-log-hook-blacklist {files}, 14
 - build-tasks OLD_TASK, NEW_TASK, 15
 - builder-options BUILDER_OPTIONS, 15
 - builds-nowait, 14
 - buildtool {copr, rpmbuild, koji, mock}, 14
 - changelog-entry CHANGELOG_ENTRY, 15
 - color {always, never, auto}, 13
 - comparepkgs-only COMPAREPKGS_DIR, 13
 - config-file CONFIG_FILE, 15
 - disable-inapplicable-patches, 14
 - favor-on-conflict {downstream, upstream, off}, 14
 - force-build-log-hooks, 14
 - get-old-build-from-koji, 14
 - non-interactive, 14
 - not-download-sources, 14
 - outputtool {text, json}, 14
 - pkgcomparetool {abipkgdiff, rpmdiff, licensecheck, pkgdiff, csm}, 14

- run_package_checkers() (*rebase-helper.application.Application* method), 27
- run_patch() (*rebase-helper.patch_helper.GitPatchTool* class method), 34
- run_patch() (*rebasehelper.patch_helper.PatchBase* class method), 35
- run_spec_hooks() (*rebase-helper.specfile.SpecHooksRunner* method), 40
- run_subprocess() (*rebase-helper.helpers.process_helper.ProcessHelper* static method), 23
- run_subprocess_cwd() (*rebase-helper.helpers.process_helper.ProcessHelper* static method), 23
- run_subprocess_cwd_env() (*rebase-helper.helpers.process_helper.ProcessHelper* static method), 23
- run_subprocess_env() (*rebase-helper.helpers.process_helper.ProcessHelper* static method), 24
- ## S
- save() (*rebasehelper.specfile.SpecFile* method), 39
- section() (*rebasehelper.specfile.SpecContent* method), 36
- SECTION_HEADERS (*rebase-helper.specfile.SpecContent* attribute), 36
- set_extra_version() (*rebase-helper.specfile.SpecFile* method), 39
- set_extra_version_separator() (*rebase-helper.specfile.SpecFile* method), 39
- set_path() (*rebasehelper.specfile.PatchObject* method), 36
- set_release_number() (*rebase-helper.specfile.SpecFile* method), 39
- set_tag() (*rebasehelper.specfile.SpecFile* method), 39
- set_terminal_background() (*rebase-helper.logger.ColorizingStreamHandler* method), 32
- set_version() (*rebasehelper.specfile.SpecFile* method), 39
- set_version_using_archive() (*rebase-helper.specfile.SpecFile* method), 39
- setup() (*rebasehelper.application.Application* static method), 27
- setup_logging() (*rebase-helper.application.Application* static method), 27
- should_use_colors() (*rebase-helper.helpers.console_helper.ConsoleHelper* class method), 17
- source_dir (*rebasehelper.patch_helper.GitPatchTool* attribute), 35
- SourcePackageBuildError, 31
- SOURCES
rebase-helper command line option, 13
- sources (*rebasehelper.specfile.SpecFile* attribute), 40
- spc (*rebasehelper.specfile.SpecFile* attribute), 40
- spec_content (*rebasehelper.specfile.SpecFile* attribute), 40
- spec_file (*rebasehelper.application.Application* attribute), 27
- spec_file_path (*rebase-helper.application.Application* attribute), 27
- SpecContent (class in *rebasehelper.specfile*), 36
- SpecFile (class in *rebasehelper.specfile*), 36
- SpecHooksRunner (class in *rebasehelper.specfile*), 40
- split_nevra() (*rebase-helper.helpers.rpm_helper.RpmHelper* class method), 25
- split_version_string() (*rebase-helper.specfile.SpecFile* static method), 40
- SRPMBuildHelper (class in *rebase-helper.build_helper*), 31
- SRPMBuildToolBase (class in *rebase-helper.build_helper*), 31
- strip (*rebasehelper.specfile.PatchObject* attribute), 36
- strip_patch_name() (*rebase-helper.patch_helper.GitPatchTool* class method), 35
- substitute_path_with_macros() (*rebase-helper.helpers.macro_helper.MacroHelper* static method), 21
- SUCCESS (*rebasehelper.logger.CustomLogger* attribute), 32
- ## T
- TarArchiveType (class in *rebasehelper.archive*), 28
- TarBz2ArchiveType (class in *rebasehelper.archive*), 28
- TarGzArchiveType (class in *rebasehelper.archive*), 28
- TarXzArchiveType (class in *rebasehelper.archive*), 28
- temp_dir (*rebasehelper.application.Application* attribute), 27
- TEMPDIR (*rebasehelper.temporary_environment.TemporaryEnvironment* attribute), 41
- TEMPDIR_BUILD (*rebase-helper.build_helper.RpmbuildTemporaryEnvironment* attribute), 30
- TEMPDIR_BUILDROOT (*rebase-helper.build_helper.RpmbuildTemporaryEnvironment*

attribute), 30
 TEMPDIR_RESULTS (rebase-
 helper.build_helper.BuildTemporaryEnvironment
 attribute), 29
 TEMPDIR_RPMBUILD (rebase-
 helper.build_helper.RpmbuildTemporaryEnvironment
 attribute), 31
 TEMPDIR_RPMS (rebase-
 helper.build_helper.RpmbuildTemporaryEnvironment
 attribute), 31
 TEMPDIR_SOURCES (rebase-
 helper.build_helper.BuildTemporaryEnvironment
 attribute), 29
 TEMPDIR_SPEC (rebase-
 helper.build_helper.BuildTemporaryEnvironment
 attribute), 29
 TEMPDIR_SPECS (rebase-
 helper.build_helper.BuildTemporaryEnvironment
 attribute), 29
 TEMPDIR_SRPMS (rebase-
 helper.build_helper.RpmbuildTemporaryEnvironment
 attribute), 31
 TemporaryEnvironment (class in rebase-
 helper.temporary_environment), 41
 terminal_background (rebase-
 helper.logger.ColorizingStreamHandler at-
 tribute), 32
 TgzArchiveType (class in rebasehelper.archive), 28
 TRACE (rebasehelper.logger.CustomLogger attribute), 32

U

update_changelog() (rebase-
 helper.specfile.SpecFile method), 40
 update_paths_to_patches() (rebase-
 helper.specfile.SpecFile method), 40
 update_setup_dirname() (rebase-
 helper.specfile.SpecFile method), 40
 update_sources() (rebase-
 helper.helpers.lookaside_cache_helper.LookasideCacheHelper
 class method), 21
 upload_srpm() (rebase-
 helper.helpers.koji_helper.KojiHelper class
 method), 20
 use_colors (rebase-
 helper.helpers.console_helper.ConsoleHelper
 attribute), 17

V

VERBOSE (rebasehelper.logger.CustomLogger attribute),
 32

W

wait_for_task() (rebase-
 helper.build_helper.BuildToolBase class

 method), 30
 watch_build() (rebase-
 helper.helpers.copr_helper.CoprHelper class
 method), 17
 watch_koji_tasks() (rebase-
 helper.helpers.koji_helper.KojiHelper class
 method), 20
 write_updated_patches() (rebase-
 helper.specfile.SpecFile method), 40
Z
 ZipArchiveType (class in rebasehelper.archive), 29