
Rebase Helper Documentation

Release 0.13.1

Petr Hracek

Tomas Hozza

Apr 23, 2018

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
4.1	User Guide	10
4.2	API	14
4.3	Change Log	32
	Python Module Index	41

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>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
<i>debug.log</i>	log containing the most verbose output of a rebase

How does it work?

The following steps describe a rebase process:

- **Preparation**

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

- **Getting sources**

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

- **Downstream patches**

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

- **Build**

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

- **Comparison**

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

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

4.1 User Guide

4.1.1 Installation

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

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

Dependency	Package name (in Fedora)	Notes
git	git	for rebasing of downstream patches
rpmbuild	rpm-build	for building SRPM and for 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

4.1.2 Rebasing in container

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

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

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

4.1.3 Usage

SYNOPSIS

```
rebase-helper [-h] [--version] [-v] [--color {always,never,auto}]
  [--background {dark,light,auto}] [--results-dir RESULTS_DIR]
  [-p|-b|--comparepkgs-only COMPAREPKGS_DIR] [-c]
  [--buildtool {copr,rpmbuild,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
  {RubyHelperHook,CommitHashUpdaterHook,PyPIURLFixHook,TypoFixHook}]
```

```

[--non-interactive] [--not-download-sources] [-w] [--apply-changes]
[--disable-inapplicable-patches] [--get-old-build-from-koji]
[--skip-version-check] [--update-sources] [--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, 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
- 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** {RubyHelperHook, CommitHashUpdaterHook, PyPIURLFixHook, TypoFixHook}
prevent specified spec hooks from being run
- non-interactive**
do not interact with user
- 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
- builds-nowait**
do not wait for remote builds to finish
- build-tasks** OLD_TASK, NEW_TASK
comma-separated remote build task ids
- builder-options** BUILDER_OPTIONS
enable arbitrary local builder option(s), enclose BUILDER_OPTIONS in quotes to pass more than one
- srpm-builder-options** SRPM_BUILDER_OPTIONS
enable arbitrary local srpm builder option(s), enclose SRPM_BUILDER_OPTIONS in quotes to pass more than one
- changelog-entry** CHANGELOG_ENTRY
text to use as changelog entry, can contain RPM macros, which will be expanded

--config-file CONFIG_FILE
 path to a configuration file, defaults to \$XDG_CONFIG_HOME/rebase-helper.cfg

4.1.4 Writing plugins

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

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

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

Example

Listing 4.1: my_spec_hook/__init__.py

```
from rebasehelper.specfile import BaseSpecHook

class MySpecHook(BaseSpecHook):

    NAME = 'MySpecHook'

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

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

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

Listing 4.2: setup.py

```
from setuptools import setup

setup(
    name='MySpecHook',
    version='0.1',
```

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

4.2 API

4.2.1 Application module

4.2.2 Archive module

4.2.3 Build helper module

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

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

TEMPDIR_RESULTS = 'TEMPDIR_RESULTS'

TEMPDIR_SOURCES = 'TEMPDIR_SOURCES'

TEMPDIR_SPEC = 'TEMPDIR_SPEC'

TEMPDIR_SPECS = 'TEMPDIR_SPECS'

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

DEFAULT = **False**

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

classmethod `build` (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

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

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

static `get_builder_options` (**kwargs)

classmethod `get_detached_task` (task_id, results_dir)
Gets packages and logs for specified task

Parameters

- **task_id** – detached task id
- **results_dir** – path to DIR where results should be stored

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

classmethod **get_logs** ()

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

classmethod **get_task_info** (*build_dict*)

Gets information about detached remote task

Parameters **build_dict** – build data

Returns task info

classmethod **is_default** ()

Checks if the tool is the default build tool.

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

Checks if tool name matches the desired one.

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

Prepare for building.

Parameters **spec** – spec file object

classmethod **wait_for_task** (*build_dict, 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.Builder` (*tool=None*)

Class representing a process of building binaries from sources.

accepts_options ()

build (**args, **kwargs*)

Build sources.

build_tools = {'copr': <class 'rebasehelper.build_tools.copr_tool.CoprBuildTool'>, 'rpm'

creates_tasks ()

classmethod **get_default_tool** ()

Returns default build tool

get_detached_task (*task_id, results_dir*)

Get detached task

get_logs ()

Get logs.

classmethod **get_supported_tools** ()

Returns a list of supported build tools

Returns list of supported build tools

get_task_info (*build_dict*)
Get task info

classmethod load_build_tools ()

prepare (*spec, conf*)
Prepare for build

wait_for_task (*build_dict, task_id, results_dir*)
Wait for task

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.**SRPMBuildToolBase**
Base class for SRPM builder tools

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_build_tool_name ()
Returns name of the SRPM Build Tool

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*)

classmethod is_default ()
Returns true if the SRPM Build Tool is set to be default

class rebasehelper.build_helper.**SRPMBuilder** (*tool=None*)
Builder class for building SRPMs.

build (**args, **kwargs*)

classmethod get_default_tool ()
Returns default build tool

get_logs ()
Get logs.

```
classmethod get_supported_tools ()
```

```
    Returns list of supported srpm build tools
```

```
classmethod load_srpm_build_tools ()
```

```
srpm_build_tools = {'rpmbuild': <class 'rebasehelper.srpm_build_tools.rpmbuild_tool.Rp
```

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

```
    Error indicating failure to build Source Package.
```

4.2.4 CLI module

```
class rebasehelper.cli.CLI (args=None)
```

```
    Class for processing data from commandline
```

```
    static build_parser ()
```

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

```
    static run ()
```

```
class rebasehelper.cli.CustomAction (option_strings, switch=False, actual_default=None,
                                     dest=None, default=None, nargs=None, required=False,
                                     type=None, metavar=None, help=None, choices=None)
```

```
class rebasehelper.cli.CustomArgumentParser (prog=None, usage=None, descrip-
                                             tion=None, epilog=None, version=None,
                                             parents=[], formatter_class=<class 'arg-
                                             parse.HelpFormatter'>, prefix_chars='-
                                             ', fromfile_prefix_chars=None,
                                             argument_default=None, con-
                                             flict_handler='error', add_help=True)
```

```
    error (message)
```

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

4.2.5 Exception module

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

```
    Error indicating failure unable to find checker binary.
```

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

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

4.2.6 Logger module

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

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

```
    emit (record)
```

```
    set_terminal_background (background)
```

```
terminal_background = 'dark'
```

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

```
HEADING = 26
IMPORTANT = 27
SUCCESS = 25
TRACE = 11
```

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.

Args: *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: logging.FileHandler: Created file handler instance.

static add_stream_handler (*logger_object, level=None, formatter_object=None*)
Adds stream handler to the given logger.

Args: *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: logging.StreamHandler: Created stream handler instance.

static get_basic_logger (*logger_name, level=10*)
Sets up a basic logger without any handler.

Args: *logger_name* (str): Logger name. *level* (int): Severity threshold.

Returns: logging.Logger: Created logger instance.

4.2.7 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 = 'ext'
NAME = 'name'
```

static get_default_tool ()
Returns default output tool

classmethod get_extension ()

classmethod get_name ()

classmethod get_report_path (*app*)

static get_supported_tools ()
Returns list of supported output tools

classmethod match (*cmd=None*)
Checks if tool name matches the desired one.

```
classmethod 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.

```
run_output_tools (logs=None, app=None)
```

Runs all output tools.

Parameters

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

4.2.8 Patch helper module

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

Class for git command used for patching old and new sources

```
CMD = 'git'
```

```
classmethod apply_old_patches ()
```

Function applies a patch to a old/new sources

```
static apply_patch (repo, patch_object)
```

Function applies patches to old sources It tries apply patch with am command and if it fails then with command –apply

```
diff_cls = None
```

```
classmethod init_git (directory)
```

Function initialize old and new Git repository

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

```
new_repo = None
```

```
new_sources = ''
```

```
non_interactive = False
```

```
old_repo = None
old_sources = ''
output_data = None
patches = None
classmethod run_patch (old_dir, new_dir, rest_sources, patches, **kwargs)
    The function can be used for patching one directory against another
source_dir = ''
```

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

```
classmethod match (cmd=None)
    Method checks whether it is usefull patch method
classmethod run_patch (old_dir, new_dir, rest_sources, patches, **kwargs)
    Method will check all patches in relevant package
```

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

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

Parameters

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

Returns

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

4.2.9 Specfile module

class `rebasehelper.specfile.BaseSpecHook`
Base class for a spec hook

```
classmethod get_categories ()
    Returns list of categories of a spec hook
```

```
classmethod get_name ()
    Returns the name of a spec hook
```

```
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.SpecFile (path, changelog_entry, sources_location="", download=True)
```

Class representing a SPEC file

```
category = None
```

```
static construct_string_with_comment (lines)
```

Wraps the line in a rebase-helper specific comments

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

Returns list with lines

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

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

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

Returns copy of the current object

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

```
download = False
```

```
download_remote_sources ()
```

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

Returns None

```
extra_version = None
```

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

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

Parameters

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

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

`find_archive_target_in_prep (archive)`

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

Parameters `archive` – Path to archive

Returns Target path relative to builddir or None if not determined

`get_applied_patches ()`

Method returns list of all applied patches.

Returns list of PatchObject

`get_archive ()`

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

Returns basename of first Source in SPEC file

Return type str

`get_epoch_number ()`

Method for getting epoch of the package

Returns

`get_extra_version ()`

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

Returns String

`get_extra_version_separator ()`

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

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

Return type str

`get_full_version ()`

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

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

Return type str

`get_new_log ()`

`get_not_used_patches ()`

Method returns list of all unapplied patches.

Returns list of PatchObject

`get_package_name ()`

Function returns a package name

Returns

`get_patches ()`

Method returns list of all applied and not applied patches

Returns list of PatchObject

`get_path ()`

Return only spec file path

Returns

static get_paths_with_rpm_macros (*files*)
 Method modifies paths in passed list to use RPM macros

Parameters **files** – list of absolute paths

Returns modified list of paths with RPM macros

get_prep_section ()
 Function returns whole prep section

get_release ()
 Method for getting full release string of the package

Returns

get_release_number ()
 Method for getting the release of the package

Returns

get_requires ()
 Function returns a package requirements

Returns

get_setup_dirname ()
 Get dirname from %setup or %autosetup macro arguments

Returns dirname

get_sources ()
 Method returns dictionary with local sources list.

Returns list of Sources with absolute path

Return type list of str

get_spec_section (*section_name*)
 Returns the section of selected name

Parameters **section_name** – section name to get

Returns list of lines contained in the selected section

get_version ()
 Method returns the version

Returns

hdr = None

insert_changelog (*new_log*)

is_test_suite_enabled ()
 Returns whether test suite is enabled during the build time

Returns True if enabled or False if not

modify_spec_files_section (*files*)
 Function repairs spec file according to new sources.

Parameters **files** –

Returns

patches = None

path = ''

prep_section = []

redefine_release_with_macro (*macro*)

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

Parameters **macro** –

Returns

removed_patches = []

revert_redefine_release_with_macro (*macro*)

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

Parameters **macro** –

Returns

rpm_sections = {}

save ()

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

set_extra_version (*extra_version*)

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

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

Returns None

set_extra_version_separator (*separator*)

Set the string that separates the version and extra version

Parameters **separator** –

Returns

set_release_number (*release*)

Method to set release number

Parameters **release** –

Returns

set_spec_section (*section_name, new_section*)

Returns the section of selected name

Parameters **section_name** – section name to get

Returns list of lines contained in the selected section

set_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** = []**static split_version_string** (*version_string*="")

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

Parameters **version_string** – version string such as '1.1.1' or '1.2.3b1', ...**Returns** tuple of strings with (extracted version, extra version, separator) or (None, None, None) if extraction failed**update_changelog** (*new_log*)

Function updates changelog with new version

update_paths_to_patches ()**update_setup_dirname** (*dirname*)

Update %setup or %autosetup dirname argument if needed

Parameters **dirname** – new dirname to be used**write_updated_patches** (*patches, disable_inapplicable*)

Function writes the patches to -rebase.spec file

class `rebasehelper.specfile.SpecHooksRunner`

Class representing the process of running various spec file hooks.

get_available_spec_hooks ()

Returns a list of all available spec hooks

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

Runs all spec hooks.

Parameters

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

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

Function returns a name in results directory

Parameters

- **dir_name** –
- **name** –

Returns full path to results dir with name

4.2.10 Utils module

class `rebasehelper.utils.ConsoleHelper`

Class for command line interaction with the user.

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

ContextManager for capturing stdout/stderr

static `color_is_light` (*rgb, bit_width*)

Determines whether a color is light or dark.

Args: `rgb`(tuple): RGB tuple. `bit_width`: Number of bits defining the RGB.

Returns: bool: Whether a color is light or dark.

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

Prints colored output if possible.

Args: `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: str: Whether to use dark or light color scheme.

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

Captures a response of a control sequence from STDIN.

Args: `query` (str): Control sequence. `timeout` (int, float): Time given to the terminal to react.

Returns: str: Response of the terminal.

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

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

Args: `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: bool: True or False, based on user's input.

static `parse_rgb_device_specification` (*specification*)

Parses RGB device specification.

Args: `specification`(str): RGB device specification.

Returns: tuple: 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.

classmethod `should_use_colors` (*conf*)

Determines whether ANSI colors should be used for CLI output.

Args: `conf` (rebasehelper.config.Config): Configuration object with arguments from the command line.

Returns: bool: Whether colors should be used.

`use_colors = False`

class `rebasehelper.utils.CoprHelper`

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

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

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

classmethod `get_build_status` (*client, build_id*)

classmethod `get_build_url` (*client, build_id*)

```
classmethod get_client ()
```

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

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

Exception indicating that download of a file failed.

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

Class for downloading sources defined in SPEC file

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

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

Parameters

- **url** – URL from which to download the file
- **destination_path** – path where to store downloaded file
- **blocksize** – size in Bytes of blocks used for downloading the file and reporting progress

Returns None

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

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

Parameters

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

Returns None

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

Method to get a Response object from HTTP, HTTPS or FTP URL

Parameters

- **url** – URL from which the Response is obtained
- **kwargs** – optional arguments used to get Response object

Returns Response object

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

```
static file_available (filename)
```

```
class rebasehelper.utils.GitHelper
```

Class which operates with git repositories

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

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

```
classmethod get_email ()
```

```
classmethod get_user ()
```

```
classmethod run_mergetool (repo)
```

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

Error indicating problems with Git

exception `rebasehelper.utils.GitRuntimeError`

Error indicating problems with Git

class `rebasehelper.utils.KojiHelper`

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

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

Args: `profile` (str): Koji profile to use.

Returns: `koji.ClientSession`: Newly created session instance.

Raises: `RebaseHelperError`: If login failed.

classmethod `display_task_results` (*tasks*)

Prints states of Koji tasks.

Args: `tasks` (list): List of `koji.TaskWatcher` instances.

classmethod `download_build` (*session, build_id, destination*)

Downloads RPMs and logs of a Koji build.

Args: `session` (`koji.ClientSession`): Active Koji session instance. `build_id` (str): Koji build ID. `destination` (str): Path where to download files to.

Returns: tuple: List of downloaded RPMs and list of downloaded logs.

Raises: `DownloadError`: If download failed.

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

Downloads packages and logs of finished Koji tasks.

Args: `session` (`koji.ClientSession`): Active Koji session instance. `tasklist` (list): List of task IDs. `destination` (str): Path where to download files to.

Returns: tuple: List of downloaded RPMs and list of downloaded logs.

Raises: `DownloadError`: If download failed.

functional = **False**

classmethod `get_latest_build` (*session, package*)

Looks up latest Koji build of a package.

Args: `session` (`koji.ClientSession`): Active Koji session instance. `package` (str): Package name.

Returns: tuple: Found latest package version and Koji build ID.

classmethod `get_task_url` (*session, task_id*)

classmethod `upload_srpm` (*session, srpm*)

Uploads SRPM to a Koji hub.

Args: `session` (`koji.ClientSession`): Active Koji session instance. `srpm` (str): Valid path to SRPM.

Returns: str: Remote path to the uploaded SRPM.

Raises: `RebaseHelperError`: If upload failed.

classmethod `watch_koji_tasks` (*session, tasklist*)

Waits for Koji tasks to finish and prints their states.

Args: `session` (`koji.ClientSession`): Active Koji session instance. `tasklist` (list): List of task IDs.

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

```

exception rebasehelper.utils.LookasideCacheError
    Exception indicating a problem accessing lookaside cache

class rebasehelper.utils.LookasideCacheHelper
    Class for downloading files from Fedora/RHEL lookaside cache

    classmethod download (tool, basepath, package)

    rpkg_config_dir = '/etc/rpkg'

    classmethod update_sources (tool, basepath, package, old_sources, new_sources)

class rebasehelper.utils.MacroHelper
    Helper class for working with RPM macros

    static dump ()
        Returns list of all defined macros

        Returns list of macros

    static expand (s, default=None)

    static filter (macros, **kwargs)
        Returns all macros satisfying specified filters

        Parameters

        • macros – list of macros to be filtered

        • kwargs – filters

        Returns filtered list of macros

exception rebasehelper.utils.ParseError

class rebasehelper.utils.PathHelper
    Class which finds a file or files in specific directory

    static find_all_files (top_path, pattern)
        Finds a file that matches the given 'pattern' recursively starting in the 'top_path' directory. If found, returns full path to the first occurrence of the file, otherwise returns None.

    static find_all_files_current_dir (top_path, pattern)
        Finds all files that matches the given 'pattern' in the 'top_path' directory. If found, returns fields of all files, otherwise returns None.

    static find_first_dir_with_file (top_path, pattern)
        Finds a file that matches the given 'pattern' recursively starting in the 'top_path' directory. If found, returns full path to the directory with first occurrence of the file, otherwise returns None.

    static find_first_file (top_path, pattern, recursion_level=None)
        Finds a file that matches the given 'pattern' recursively starting in the 'top_path' directory. If found, returns full path to the first occurrence of the file, otherwise returns None.

    static get_temp_dir ()
        Returns a path to new temporary directory.

class rebasehelper.utils.ProcessHelper
    Class for execution subprocess

    DEV_NULL = '/dev/null'

    static run_subprocess (cmd, input_file=None, output_file=None)
        Runs the passed command as a subprocess.

        Parameters

```

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

Returns exit code of the process

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

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

Parameters

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

Returns exit code of the process

```
static run_subprocess_cwd_env (cmd, cwd=None, env=None, input_file=None, out-  
                               put_file=None, shell=False)
```

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

Parameters

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

Returns exit code of the process

```
static run_subprocess_env (cmd, env=None, input_file=None, output_file=None, shell=False)
```

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

Parameters

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

Returns exit code of the process

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

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

```
ARCHES = None
```

```

static all_packages_installed (pkg_names=None)
    Check if all packages in passed list are installed.

    Parameters pkg_names – iterable with package named to check for

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

static get_arches ()
    Get list of all known architectures

static get_header_from_rpm (rpm_name)
    Function returns a rpm header from given rpm package for later on analysis

    Parameters rpm_name –

    Returns

static get_info_from_rpm (rpm_name, info)
    Method returns a name of the package from RPM file format

    Parameters pkg_name –

    Returns

static install_build_dependencies (spec_path=None, assume_yes=False)
    Install all build requires for a package using PolicyKits

    Parameters spec_path – absolute path to SPEC file

    Returns

static is_package_installed (pkg_name=None)
    Checks whether package with passed name is installed.

    Parameters package_name – package name we want to check for

    Returns True if installed, False if not installed

classmethod parse_spec (path, flags=None)

classmethod split_nevra (s)
    Splits string into name, epoch, version, release and arch components

class rebasehelper.utils.SilentArgumentParser (prog=None, usage=None, descrip-
    tion=None, epilog=None, version=None,
    parents=[], formatter_class=<class 'arg-
    parse.HelpFormatter'>, prefix_chars='-
    ', fromfile_prefix_chars=None,
    argument_default=None, con-
    flict_handler='error', add_help=True)

    error (message)

class rebasehelper.utils.TemporaryEnvironment (exit_callback=None)
    Class representing a temporary environment (directory) that can be used as a workspace. It can be used with
    with statement.

    TEMPDIR = 'TEMPDIR'

    env ()
    Returns copy of _env dictionary.

    Returns copy of _env dictionary

    path ()
    Returns path to the temporary environment.

```

Returns abs path to the environment

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

Function returns a part of self.kwargs dictionary

Parameters

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

Returns value from dictionary

4.3 Change Log

4.3.1 [Unreleased]

4.3.2 [0.13.1] - 2018-04-19

Added

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

Changed

- Extended README .md
- Cleaned up constants

Fixed

- Fixed crash after failed rebase when no checkers were run

4.3.3 [0.13.0] - 2018-03-29

Added

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

Changed

- Switched to `requests` library for downloads
- Made error messages from **Koji** builds more useful
- Reworked handling of downstream patches
- Changed package build process to build first SRPMs and then RPMs
- Divided checkers into categories running at different phases of rebase
- **Koji** build tool refactored to be better adjustable and extensible
- Colorized **rebase-helper** output and enhanced log messages
- Significantly improved rebase summary and report

Fixed

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

4.3.4 [0.12.0] - 2017-12-19

Added

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

Changed

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

Fixed

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

4.3.5 [0.11.0] - 2017-10-04

Added

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

Changed

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

Fixed

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

4.3.6 [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

4.3.7 [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

4.3.8 [0.9.0] - 2017-01-05

Added

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

Changed

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

Fixed

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

4.3.9 [0.8.0] - 2016-07-31

Added

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

Changed

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

Fixed

- Many bug fixes and code clean up

4.3.10 [0.7.3] - 2016-04-08

Added

- Added `rpm.addMacro`

Fixed

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

4.3.11 [0.7.2] - 2016-03-15

Added

- Added information about scratch builds

Fixed

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

4.3.12 [0.7.1] - 2016-02-22

Added

- Two new command line options used by upstream monitoring

Fixed

- **fedpkg** reimplementation

4.3.13 [0.7.0] - 2016-01-13

Changed

- Several improvements

Fixed

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

Removed

- Removed a bunch of debug stuff

4.3.14 [0.6.2] - 2015-11-09

Fixed

- Logs are being saved to their own directory
- Prep script is moved into workspace directory
- No more traceback in case `koji` module is not present

- Each checker creates its own log file
- **rebase-helper** informs if it failed or not
- Report on script is smarter

4.3.15 [0.6.1] - 2015-10-30

Added

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

4.3.16 [0.6.0] - 2015-07-31

Added

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

Fixed

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

4.3.17 [0.5.0] - 2015-05-22

Added

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

Changed

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

Fixed

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

Removed

- `DiffHelper` class is not needed

4.3.18 [0.4.0] - 2014-12-05

Added

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

Changed

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

4.3.19 [0.3.1] - 2014-07-25

Added

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

Fixed

- Several fixes

4.3.20 [0.3.0]

Added

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

4.3.21 [0.2.0]

Added

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

4.3.22 [0.1.0]

Added

- Initial classes
- CLI interface

r

rebasehelper.build_helper, 14
rebasehelper.cli, 17
rebasehelper.exceptions, 17
rebasehelper.logger, 17
rebasehelper.output_tool, 18
rebasehelper.patch_helper, 19
rebasehelper.specfile, 20
rebasehelper.utils, 25

Symbols

- apply-changes
 - rebase-helper command line option, 12
- background {dark,light,auto}
 - rebase-helper command line option, 11
- build-tasks OLD_TASK,NEW_TASK
 - rebase-helper command line option, 12
- builder-options BUILDER_OPTIONS
 - rebase-helper command line option, 12
- builds-nowait
 - rebase-helper command line option, 12
- buildtool {copr,rpmbuild,mock}
 - rebase-helper command line option, 12
- changelog-entry CHANGELOG_ENTRY
 - rebase-helper command line option, 12
- color {always,never,auto}
 - rebase-helper command line option, 11
- comparepkgs-only COMPAREPKGS_DIR
 - rebase-helper command line option, 11
- config-file CONFIG_FILE
 - rebase-helper command line option, 13
- disable-inapplicable-patches
 - rebase-helper command line option, 12
- get-old-build-from-koji
 - rebase-helper command line option, 12
- non-interactive
 - rebase-helper command line option, 12
- not-download-sources
 - rebase-helper command line option, 12
- outputtool {text,json}
 - rebase-helper command line option, 12
- pkgcomparetool {abip-kgdiff,rpmdiff,licensecheck,pkgdiff,csmock}
 - rebase-helper command line option, 12
- results-dir RESULTS_DIR
 - rebase-helper command line option, 11
- skip-version-check
 - rebase-helper command line option, 12
- spec-hook-blacklist {RubyHelper-Hook,CommitHashUpdaterHook,PyPIURLFixHook,TypoFixHook}
 - rebase-helper command line option, 12
- srpm-builder-options SRPM_BUILDER_OPTIONS
 - rebase-helper command line option, 12
- srpm-buildtool {rpmbuild,mock}
 - rebase-helper command line option, 12
- update-sources
 - rebase-helper command line option, 12
- version
 - rebase-helper command line option, 11
- versioneer {rubygems,pypi,anitya,hackage,cpan,npmjs}
 - rebase-helper command line option, 12
- versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmjs}
 - rebase-helper command line option, 12
- b, -build-only
 - rebase-helper command line option, 11
- c, -continue
 - rebase-helper command line option, 12
- h, -help
 - rebase-helper command line option, 11
- p, -patch-only
 - rebase-helper command line option, 11
- v, -verbose
 - rebase-helper command line option, 11
- w, -keep-workspace
 - rebase-helper command line option, 12

A

- accepts_options() (rebasehelper.build_helper.Builder method), 15
- accepts_options() (rebasehelper.build_helper.BuildToolBase class method), 14
- add_file_handler() (rebasehelper.logger.LoggerHelper static method), 18
- add_stream_handler() (rebasehelper.logger.LoggerHelper static method), 18
- all_packages_installed() (rebasehelper.utils.RpmHelper static method), 30

apply_old_patches() (rebase-helper.patch_helper.GitPatchTool class method), 19
 apply_patch() (rebasehelper.patch_helper.GitPatchTool static method), 19
 ARCHES (rebasehelper.utils.RpmHelper attribute), 30

B

BaseOutputTool (class in rebasehelper.output_tool), 18
 BaseSpecHook (class in rebasehelper.specfile), 20
 BinaryPackageBuildError, 14
 build() (rebasehelper.build_helper.Builder method), 15
 build() (rebasehelper.build_helper.BuildToolBase class method), 14
 build() (rebasehelper.build_helper.SRPMBuildToolBase method), 16
 build() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
 build() (rebasehelper.utils.CoprHelper class method), 26
 build_parser() (rebasehelper.cli.CLI static method), 17
 build_tools (rebasehelper.build_helper.Builder attribute), 15
 Builder (class in rebasehelper.build_helper), 15
 BuildTemporaryEnvironment (class in rebase-helper.build_helper), 14
 BuildToolBase (class in rebasehelper.build_helper), 14

C

category (rebasehelper.specfile.SpecFile attribute), 21
 CheckerNotFound, 17
 CLI (class in rebasehelper.cli), 17
 CliHelper (class in rebasehelper.cli), 17
 CMD (rebasehelper.patch_helper.GitPatchTool attribute), 19
 color_is_light() (rebasehelper.utils.ConsoleHelper static method), 25
 ColorizingStreamHandler (class in rebasehelper.logger), 17
 colors (rebasehelper.logger.ColorizingStreamHandler attribute), 17
 ConsoleHelper (class in rebasehelper.utils), 25
 ConsoleHelper.Capturer (class in rebasehelper.utils), 25
 construct_string_with_comment() (rebase-helper.specfile.SpecFile static method), 21
 CoprHelper (class in rebasehelper.utils), 26
 copy() (rebasehelper.specfile.SpecFile method), 21
 cprint() (rebasehelper.utils.ConsoleHelper class method), 26
 create_project() (rebasehelper.utils.CoprHelper class method), 26
 create_session() (rebasehelper.utils.KojiHelper class method), 28

creates_tasks() (rebasehelper.build_helper.Builder method), 15
 creates_tasks() (rebasehelper.build_helper.BuildToolBase class method), 14
 CustomAction (class in rebasehelper.cli), 17
 CustomArgumentParser (class in rebasehelper.cli), 17
 CustomHelpFormatter (class in rebasehelper.cli), 17
 CustomLogger (class in rebasehelper.logger), 18

D

DEFAULT (rebasehelper.build_helper.BuildToolBase attribute), 14
 DEFAULT (rebasehelper.build_helper.SRPMBuildToolBase attribute), 16
 DEFAULT (rebasehelper.output_tool.BaseOutputTool attribute), 18
 defined_sections (rebasehelper.specfile.SpecFile attribute), 21
 detect_background() (rebasehelper.utils.ConsoleHelper static method), 26
 DEV_NULL (rebasehelper.utils.ProcessHelper attribute), 29
 diff_cls (rebasehelper.patch_helper.GitPatchTool attribute), 19
 display_task_results() (rebasehelper.utils.KojiHelper class method), 28
 download (rebasehelper.specfile.SpecFile attribute), 21
 download() (rebasehelper.utils.LookasideCacheHelper class method), 29
 download_build() (rebasehelper.utils.CoprHelper class method), 26
 download_build() (rebasehelper.utils.KojiHelper class method), 28
 download_file() (rebasehelper.utils.DownloadHelper static method), 27
 download_remote_sources() (rebase-helper.specfile.SpecFile method), 21
 download_task_results() (rebasehelper.utils.KojiHelper class method), 28
 DownloadError, 27
 DownloadHelper (class in rebasehelper.utils), 27
 dump() (rebasehelper.utils.MacroHelper static method), 29

E

emit() (rebasehelper.logger.ColorizingStreamHandler method), 17
 env() (rebasehelper.utils.TemporaryEnvironment method), 31
 environment variable
 PACKAGE, 10
 REPOSITORY, 10
 error() (rebasehelper.cli.CustomArgumentParser method), 17

- error() (rebasehelper.utils.SilentArgumentParser method), 31
- exchange_control_sequence() (rebasehelper.utils.ConsoleHelper static method), 26
- expand() (rebasehelper.utils.MacroHelper static method), 29
- EXTENSION (rebasehelper.output_tool.BaseOutputTool attribute), 18
- extra_version (rebasehelper.specfile.SpecFile attribute), 21
- extract_version_from_archive_name() (rebasehelper.specfile.SpecFile static method), 21
- ## F
- file_available() (rebasehelper.utils.FileHelper static method), 27
- FileHelper (class in rebasehelper.utils), 27
- filter() (rebasehelper.utils.MacroHelper static method), 29
- find_all_files() (rebasehelper.utils.PathHelper static method), 29
- find_all_files_current_dir() (rebasehelper.utils.PathHelper static method), 29
- find_archive_target_in_prep() (rebasehelper.specfile.SpecFile method), 21
- find_first_dir_with_file() (rebasehelper.utils.PathHelper static method), 29
- find_first_file() (rebasehelper.utils.PathHelper static method), 29
- functional (rebasehelper.utils.KojiHelper attribute), 28
- ## G
- get_applied_patches() (rebasehelper.specfile.SpecFile method), 22
- get_arches() (rebasehelper.utils.RpmHelper static method), 31
- get_archive() (rebasehelper.specfile.SpecFile method), 22
- get_available_spec_hooks() (rebasehelper.specfile.SpecHooksRunner method), 25
- get_basic_logger() (rebasehelper.logger.LoggerHelper static method), 18
- get_build_status() (rebasehelper.utils.CoprHelper class method), 26
- get_build_tool_name() (rebasehelper.build_helper.BuildToolBase class method), 14
- get_build_tool_name() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
- get_build_url() (rebasehelper.utils.CoprHelper class method), 26
- get_builder_options() (rebasehelper.build_helper.BuildToolBase static method), 14
- get_categories() (rebasehelper.specfile.BaseSpecHook class method), 20
- get_client() (rebasehelper.utils.CoprHelper class method), 26
- get_default_tool() (rebasehelper.build_helper.Builder class method), 15
- get_default_tool() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
- get_default_tool() (rebasehelper.output_tool.BaseOutputTool static method), 18
- get_detached_task() (rebasehelper.build_helper.Builder method), 15
- get_detached_task() (rebasehelper.build_helper.BuildToolBase class method), 14
- get_email() (rebasehelper.utils.GitHelper class method), 27
- get_epoch_number() (rebasehelper.specfile.SpecFile method), 22
- get_extension() (rebasehelper.output_tool.BaseOutputTool class method), 18
- get_extra_version() (rebasehelper.specfile.SpecFile method), 22
- get_extra_version_separator() (rebasehelper.specfile.SpecFile method), 22
- get_full_version() (rebasehelper.specfile.SpecFile method), 22
- get_header_from_rpm() (rebasehelper.utils.RpmHelper static method), 31
- get_index() (rebasehelper.specfile.PatchObject method), 21
- get_info_from_rpm() (rebasehelper.utils.RpmHelper static method), 31
- get_latest_build() (rebasehelper.utils.KojiHelper class method), 28
- get_logs() (rebasehelper.build_helper.Builder method), 15
- get_logs() (rebasehelper.build_helper.BuildToolBase class method), 15
- get_logs() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
- get_logs() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
- get_message() (rebasehelper.utils.ConsoleHelper static method), 26
- get_name() (rebasehelper.output_tool.BaseOutputTool class method), 18
- get_name() (rebasehelper.specfile.BaseSpecHook class

- method), 20
 - get_new_log() (rebasehelper.specfile.SpecFile method), 22
 - get_not_used_patches() (rebasehelper.specfile.SpecFile method), 22
 - get_package_name() (rebasehelper.specfile.SpecFile method), 22
 - get_patch_name() (rebasehelper.specfile.PatchObject method), 21
 - get_patches() (rebasehelper.specfile.SpecFile method), 22
 - get_path() (rebasehelper.specfile.PatchObject method), 21
 - get_path() (rebasehelper.specfile.SpecFile method), 22
 - get_paths_with_rpm_macros() (rebasehelper.specfile.SpecFile static method), 22
 - get_prep_section() (rebasehelper.specfile.SpecFile method), 23
 - get_rebase_name() (in module rebasehelper.specfile), 25
 - get_release() (rebasehelper.specfile.SpecFile method), 23
 - get_release_number() (rebasehelper.specfile.SpecFile method), 23
 - get_report_path() (rebasehelper.output_tool.BaseOutputTool class method), 18
 - get_requires() (rebasehelper.specfile.SpecFile method), 23
 - get_setup_dirname() (rebasehelper.specfile.SpecFile method), 23
 - get_sources() (rebasehelper.specfile.SpecFile method), 23
 - get_spec_section() (rebasehelper.specfile.SpecFile method), 23
 - get_srpm_builder_options() (rebasehelper.build_helper.SRPMBuildToolBase static method), 16
 - get_strip() (rebasehelper.specfile.PatchObject method), 21
 - get_supported_tools() (rebasehelper.build_helper.Builder class method), 15
 - get_supported_tools() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
 - get_supported_tools() (rebasehelper.output_tool.BaseOutputTool static method), 18
 - get_task_info() (rebasehelper.build_helper.Builder method), 15
 - get_task_info() (rebasehelper.build_helper.BuildToolBase class method), 15
 - get_task_url() (rebasehelper.utils.KojiHelper class method), 28
 - get_temp_dir() (rebasehelper.utils.PathHelper static method), 29
 - get_user() (rebasehelper.utils.GitHelper class method), 27
 - get_value_from_kwargs() (in module rebasehelper.utils), 32
 - get_version() (rebasehelper.specfile.SpecFile method), 23
 - git_generated (rebasehelper.specfile.PatchObject attribute), 21
 - GIT_USER_EMAIL (rebasehelper.utils.GitHelper attribute), 27
 - GIT_USER_NAME (rebasehelper.utils.GitHelper attribute), 27
 - GitHelper (class in rebasehelper.utils), 27
 - GitPatchTool (class in rebasehelper.patch_helper), 19
 - GitRebaseError, 27
 - GitRuntimeError, 27
- ## H
- hdr (rebasehelper.specfile.SpecFile attribute), 23
 - HEADING (rebasehelper.logger.CustomLogger attribute), 18
- ## I
- IMPORTANT (rebasehelper.logger.CustomLogger attribute), 18
 - index (rebasehelper.specfile.PatchObject attribute), 21
 - init_git() (rebasehelper.patch_helper.GitPatchTool class method), 19
 - insert_changelog() (rebasehelper.specfile.SpecFile method), 23
 - install_build_dependencies() (rebasehelper.utils.RpmHelper static method), 31
 - is_default() (rebasehelper.build_helper.BuildToolBase class method), 15
 - is_default() (rebasehelper.build_helper.SRPMBuildToolBase class method), 16
 - is_package_installed() (rebasehelper.utils.RpmHelper static method), 31
 - is_test_suite_enabled() (rebasehelper.specfile.SpecFile method), 23
- ## K
- KojiHelper (class in rebasehelper.utils), 28
- ## L
- load_build_tools() (rebasehelper.build_helper.Builder class method), 16
 - load_srpm_build_tools() (rebasehelper.build_helper.SRPMBuildToolBase class method), 17
 - LoggerHelper (class in rebasehelper.logger), 18
 - LookasideCacheError, 28
 - LookasideCacheHelper (class in rebasehelper.utils), 29
- ## M
- MacroHelper (class in rebasehelper.utils), 29

match() (rebasehelper.build_helper.BuildToolBase class method), 15
 match() (rebasehelper.output_tool.BaseOutputTool class method), 18
 match() (rebasehelper.patch_helper.GitPatchTool class method), 19
 match() (rebasehelper.patch_helper.PatchBase class method), 20
 MockTemporaryEnvironment (class in rebasehelper.build_helper), 16
 modify_spec_files_section() (rebasehelper.specfile.SpecFile method), 23

N

NAME (rebasehelper.output_tool.BaseOutputTool attribute), 18
 new_repo (rebasehelper.patch_helper.GitPatchTool attribute), 19
 new_sources (rebasehelper.patch_helper.GitPatchTool attribute), 19
 non_interactive (rebasehelper.patch_helper.GitPatchTool attribute), 19

O

old_repo (rebasehelper.patch_helper.GitPatchTool attribute), 19
 old_sources (rebasehelper.patch_helper.GitPatchTool attribute), 20
 output_data (rebasehelper.patch_helper.GitPatchTool attribute), 20
 OutputToolRunner (class in rebasehelper.output_tool), 19

P

PACKAGE, 10
 parse_rgb_device_specification() (rebasehelper.utils.ConsoleHelper static method), 26
 parse_spec() (rebasehelper.utils.RpmHelper class method), 31
 ParseError, 29
 patch() (rebasehelper.patch_helper.Patcher method), 20
 PatchBase (class in rebasehelper.patch_helper), 20
 Patcher (class in rebasehelper.patch_helper), 20
 patches (rebasehelper.patch_helper.GitPatchTool attribute), 20
 patches (rebasehelper.specfile.SpecFile attribute), 23
 PatchList (class in rebasehelper.specfile), 21
 PatchObject (class in rebasehelper.specfile), 21
 path (rebasehelper.specfile.PatchObject attribute), 21
 path (rebasehelper.specfile.SpecFile attribute), 23
 path() (rebasehelper.utils.TemporaryEnvironment method), 31
 PathHelper (class in rebasehelper.utils), 29
 prep_section (rebasehelper.specfile.SpecFile attribute), 23

prepare() (rebasehelper.build_helper.Builder method), 16
 prepare() (rebasehelper.build_helper.BuildToolBase class method), 15
 prepend_results_dir_name() (rebasehelper.output_tool.BaseOutputTool class method), 18
 print_cli_summary() (rebasehelper.output_tool.BaseOutputTool class method), 19
 print_important_checkers_output() (rebasehelper.output_tool.BaseOutputTool class method), 19
 print_patches_cli() (rebasehelper.output_tool.BaseOutputTool class method), 19
 print_patches_section_cli() (rebasehelper.output_tool.BaseOutputTool class method), 19
 print_report_file_path() (rebasehelper.output_tool.BaseOutputTool class method), 19
 ProcessHelper (class in rebasehelper.utils), 29
 progress() (rebasehelper.utils.DownloadHelper static method), 27

R

rebase-helper command line option
 -apply-changes, 12
 -background {dark,light,auto}, 11
 -build-tasks OLD_TASK,NEW_TASK, 12
 -builder-options BUILDER_OPTIONS, 12
 -builds-nowait, 12
 -buildtool {copr,rpmbuild,mock}, 12
 -changelog-entry CHANGELOG_ENTRY, 12
 -color {always,never,auto}, 11
 -comparepkgs-only COMPAREPKGS_DIR, 11
 -config-file CONFIG_FILE, 13
 -disable-inapplicable-patches, 12
 -get-old-build-from-koji, 12
 -non-interactive, 12
 -not-download-sources, 12
 -outputtool {text,json}, 12
 -pkgcomparetool {abipkgdiff,rpmdiff,licensecheck,pkgdiff,csmock}, 12
 -results-dir RESULTS_DIR, 11
 -skip-version-check, 12
 -spec-hook-blacklist {RubyHelperHook,CommitHashUpdaterHook,PyPIURLFixHook,TypoFixHook}, 12
 -srpm-builder-options SRPM_BUILDER_OPTIONS, 12
 -srpm-buildtool {rpmbuild,mock}, 12
 -update-sources, 12

- version, 11
 - versioneer {rubygems,pypi,anitya,hackage,cpan,npmmjs}, 12
 - versioneer-blacklist {rubygems,pypi,anitya,hackage,cpan,npmmjs}, 12
 - b, --build-only, 11
 - c, --continue, 12
 - h, --help, 11
 - p, --patch-only, 11
 - v, --verbose, 11
 - w, --keep-workspace, 12
 - SOURCES, 11
 - rebasehelper.build_helper (module), 14
 - rebasehelper.cli (module), 17
 - rebasehelper.exceptions (module), 17
 - rebasehelper.logger (module), 17
 - rebasehelper.output_tool (module), 18
 - rebasehelper.patch_helper (module), 19
 - rebasehelper.specfile (module), 20
 - rebasehelper.utils (module), 25
 - RebaseHelperError, 17
 - redefine_release_with_macro() (rebasehelper.specfile.SpecFile method), 24
 - register_patch_tool() (in module rebasehelper.patch_helper), 20
 - removed_patches (rebasehelper.specfile.SpecFile attribute), 24
 - REPOSITORY, 10
 - request() (rebasehelper.utils.DownloadHelper static method), 27
 - revert_redefine_release_with_macro() (rebasehelper.specfile.SpecFile method), 24
 - rpkg_config_dir (rebasehelper.utils.LookasideCacheHelper attribute), 29
 - rpm_sections (rebasehelper.specfile.SpecFile attribute), 24
 - RpmbuildTemporaryEnvironment (class in rebasehelper.build_helper), 16
 - RpmHelper (class in rebasehelper.utils), 30
 - run() (rebasehelper.cli.CliHelper static method), 17
 - run() (rebasehelper.output_tool.BaseOutputTool class method), 19
 - run() (rebasehelper.specfile.BaseSpecHook class method), 20
 - run_mergetool() (rebasehelper.utils.GitHelper class method), 27
 - run_output_tools() (rebasehelper.output_tool.OutputToolRunner method), 19
 - run_patch() (rebasehelper.patch_helper.GitPatchTool class method), 20
 - run_patch() (rebasehelper.patch_helper.PatchBase class method), 20
 - run_spec_hooks() (rebasehelper.specfile.SpecHooksRunner method), 25
 - run_subprocess() (rebasehelper.utils.ProcessHelper static method), 29
 - run_subprocess_cwd() (rebasehelper.utils.ProcessHelper static method), 30
 - run_subprocess_cwd_env() (rebasehelper.utils.ProcessHelper static method), 30
 - run_subprocess_env() (rebasehelper.utils.ProcessHelper static method), 30
- ## S
- save() (rebasehelper.specfile.SpecFile method), 24
 - set_extra_version() (rebasehelper.specfile.SpecFile method), 24
 - set_extra_version_separator() (rebasehelper.specfile.SpecFile method), 24
 - set_path() (rebasehelper.specfile.PatchObject method), 21
 - set_release_number() (rebasehelper.specfile.SpecFile method), 24
 - set_spec_section() (rebasehelper.specfile.SpecFile method), 24
 - set_tag() (rebasehelper.specfile.SpecFile method), 24
 - set_terminal_background() (rebasehelper.logger.ColorizingStreamHandler method), 17
 - set_version() (rebasehelper.specfile.SpecFile method), 24
 - set_version_using_archive() (rebasehelper.specfile.SpecFile method), 24
 - should_use_colors() (rebasehelper.utils.ConsoleHelper class method), 26
 - SilentArgumentParser (class in rebasehelper.utils), 31
 - source_dir (rebasehelper.patch_helper.GitPatchTool attribute), 20
 - SourcePackageBuildError, 17
 - SOURCES
 - rebase-helper command line option, 11
 - sources (rebasehelper.specfile.SpecFile attribute), 25
 - spc (rebasehelper.specfile.SpecFile attribute), 25
 - spec_content (rebasehelper.specfile.SpecFile attribute), 25
 - SpecFile (class in rebasehelper.specfile), 21
 - SpecHooksRunner (class in rebasehelper.specfile), 25
 - split_nevra() (rebasehelper.utils.RpmHelper class method), 31
 - split_version_string() (rebasehelper.specfile.SpecFile static method), 25
 - srpm_build_tools (rebasehelper.build_helper.SRPMBuilder attribute), 17
 - SRPMBuilder (class in rebasehelper.build_helper), 16

SRPMBuildToolBase (class in rebase-helper.build_helper), 16

strip (rebasehelper.specfile.PatchObject attribute), 21

SUCCESS (rebasehelper.logger.CustomLogger attribute), 18

upload_srpm() (rebasehelper.utils.KojiHelper class method), 28

use_colors (rebasehelper.utils.ConsoleHelper attribute), 26

T

TEMPDIR (rebasehelper.utils.TemporaryEnvironment attribute), 31

TEMPDIR_BUILD (rebase-helper.build_helper.RpmbuildTemporaryEnvironment attribute), 16

TEMPDIR_BUILDROOT (rebase-helper.build_helper.RpmbuildTemporaryEnvironment attribute), 16

TEMPDIR_RESULTS (rebase-helper.build_helper.BuildTemporaryEnvironment attribute), 14

TEMPDIR_RPMBUILD (rebase-helper.build_helper.RpmbuildTemporaryEnvironment attribute), 16

TEMPDIR_RPMS (rebase-helper.build_helper.RpmbuildTemporaryEnvironment attribute), 16

TEMPDIR_SOURCES (rebase-helper.build_helper.BuildTemporaryEnvironment attribute), 14

TEMPDIR_SPEC (rebase-helper.build_helper.BuildTemporaryEnvironment attribute), 14

TEMPDIR_SPECS (rebase-helper.build_helper.BuildTemporaryEnvironment attribute), 14

TEMPDIR_SRPMS (rebase-helper.build_helper.RpmbuildTemporaryEnvironment attribute), 16

TemporaryEnvironment (class in rebasehelper.utils), 31

terminal_background (rebase-helper.logger.ColorizingStreamHandler attribute), 17

TRACE (rebasehelper.logger.CustomLogger attribute), 18

U

update_changelog() (rebasehelper.specfile.SpecFile method), 25

update_paths_to_patches() (rebase-helper.specfile.SpecFile method), 25

update_setup_dirname() (rebasehelper.specfile.SpecFile method), 25

update_sources() (rebase-helper.utils.LookasideCacheHelper class method), 29

W

wait_for_task() (rebasehelper.build_helper.Builder method), 16

wait_for_task() (rebase-helper.build_helper.BuildToolBase class method), 15

watch_build() (rebasehelper.utils.CoprHelper class method), 27

watch_koji_tasks() (rebasehelper.utils.KojiHelper class method), 28

write_updated_patches() (rebasehelper.specfile.SpecFile method), 25