NAME

repostorm − The Repository Master

SYNOPSIS

repostorm -<−OPTION> [FOLDER|DEBNAME|DISTNAME|USERNAME@URL/DISTNAME|ADD|REMOVE]

DESCRIPTION

repostorm - Software suited for extraction, packaging and maintaining a repository.

FEATURES

Repostorm has many advanced features to automate the building of debs to be Lintian error free. It has advanced scanning of files to set proper permissions and a fix routine to scan for errors and auto correct them. I have ran this tool on a folder with 13 GB of debs; prior to running the tool a total of 136,435 errors and warnings. After running the tool 1,323. repostorm is not perfect yet
but is going in the right direction. The numbers continue to diminish.

COMMAND SWITCH USAGE

Every 'flag' option has a 'no-flag' counterpart with exception of help. Specifying no switch is the same as using the **−−help** switch.

If an option or flag is marked as [XXX], it is optional. Options specified with <XXX> are required. They will only work in combination with the XXX option. Example: repostorm **−−service**
Will not work because the format is <REQUIRED> <on|off> [--autofix] in this case **−−service** is the <REQUIRED> switch as per Synopsis above. No secondary <required> switch was provided. IE repostorm **−−service ON** which is actually case insensitive the | seperator signifies either, but not both. On the other hand repostorm **−−extract** is just fine without providing a secondary switch. Please see OPTIONS section below.

repostorm as of version 1.7.1-8 has bash auto completion. This means you can enter repostorm -se hit the tab key and it will change the output to repostorm **−−service** hit tab twice again and it will display the options of **on** and **off**. Tab again and it will enter **−−autofix** for you.

OPTIONS

Analyze

−a , **−−analyze** [DEB(S)]
Specifying no [DEB] name will analyze all debs in current directory for errors prompting first. All error free debs are moved to a hidden folder called .debs/ in the current folder. If non-existant the .debs/ folder will be created for you. This should be the second switch executed. If it isn't broke don't fix it.

Build

−b , **−−build** [FOLDER]
Builds specified deb. Specifying no folder name will build all debs in current directory; prompting what will be built. All error free debs are moved to a hidden sub-folder called .debs/ in the current folder. Per session logging of individual errors to [FOLDER].deb.errors and globally to all_errors.txt for quick review these logs are removed with the -clearlogs switch see below. Full logging is sent to /log/repostorm_errors.log.

Cleanfolder

−C , **−−cleanfolder** [FOLDER]
Will erase specified [FOLDER] ignoring permissions. *Warning: Specifying no folder name will erase all sub-folders in the current* folder. Please use care. repostorm will prompt you of folders to be effected. It will also erase all previous build and error logs.

Clearlogs

−C , **−−clearlogs** [FOLDER]
Will erase all logs. Specifying an optional [FOLDER] will erase error logs specific to [FOLDER].

Extract

−e , **−−extract** [DEB]
Extracts specified [DEB]. Specifying no deb filename will extract all debs in current directory; prompting which debs will be extracted.

Fix

−f , **−−fix** [FOLDER]
Will attempt to fix all errors in the specified [FOLDER] to be Lintian / Debian error free and complaint to Debian standards. Specifying no folder name will process all folders with a [FOLDER].deb.errors that were generated by the build switch. This will take a long time to be 100% implemented.

Help

−h , **−−help** [COMMAND]
Display help for [COMMAND]
possible commands...

-a	--analyze	pre-scan deb(s) for error(s)
-b	--build	builds deb(s)
-e	--extract	extract debian archive
-c	--cleanlogs	removes log files
-C	--cleanfolder	removes folder(s)
-f	--fix	attempts to fix error(s)
-h	--help	this help message
-l	--log	view or clean logs
-p	--prepare	prepare good .debs/ for publishing
-P	--publish	publishes them to address specified
-R	--relax	relax permissions
-r	--report	examine reported errorlog(s)
-S	--scan	version scans debs
-s	--service	installs or removes repostorm as a service
-V	--verbose	reports errors verbosly and instructs how to fix
-v	--version	dump version info

Log

−l , **−−log** <view/clean/logfilename.errors>
View, clear global or individual logging activity.

Prepare

−p , **−−prepare** <DISTNAME>
Prepares known good .debs/ for publishing. Builds the directory structure in a folder called <DISTNAME>. repostorm then GPG key signs the debs as well as the release key for distribution. This command switch also auto generates packages.gz for repository management. Knowledge of how a repository works is highly recommended when using this command switch. Please man gpg for more information on key signing.

Publish

−P , **−−publish** <USERNAME:[PASSWORD]@URL/DISTNAME> Will publish known good debs to remote address of your specification. repostorm will upload all files processed by the prepare switch associated smartly. It will compare files on the remote server to local host and only update appropriately, furthermore by architecture. A future version will allow multiple distributions to be handled. The password is [optional] as ssh will prompt for it prior to engaging rsync. Please see man ssh for more information on how the command is passed. Knowledge of how a repository works is highly recommended when using this command switch.

Relaxation

−r , **−−relax** [FOLDER]
Will relax the permissions on a given [FOLDER] allowing you to edit or delete files within the folder. The proper permissions will automatically be reset when you use the build switch. *Warning: Not specifying a folder will reset permissions on all sub-folders.* Please use care where you execute this tool. repostorm will display the folders to be effected and prompt you prior to doing anything.

Service

−s , **−−service** <on|off> [--autofix]
Installs or removes based on second option to set repostorm as a service. Will install or remove repostorm as a service. Repostorm will monitor any package being installed and scan for errors and warnings optionally [--autofix] fixing them via end users discretion. This powerful feature allows the end user to make the decision if or not errors and warnings enter their system prior to the package even being installed. Similar to what anti-virus software does.

Scan

−s , **−−scan**
Version scans all debs to be processed. This should be the first step you take when dealing with large amounts of debs. When processing a single deb this switch is useless. There is no sense in building 2 seperate versions of debs. On the large scale this is not a required, but recommended base of first action.

Verbose

−V , **−−verbose** [FOLDER]
Initiates verbose mode. Invokes Lintian to verbosely report errors and gives instructions on how to fix them. If I see this project all the way through this switch may be come unnecessary, time will tell.

Version

−v , **−−version**
Display repostorm's version number and exits.

PUBLISHING NOTES

Specifying the -p prepare or the -P publish switch I felt I should take the time to document. Most people will not be interested in either switch unless they intend to run a repository on-line or locally. Time permitting, I will either write a on-line document and link here or elaborate more in this section of the man page.

EXAMPLES OF REPOSTORM USAGE

TYPICAL USAGE IN ORDER
1. Scanning of deb(s) to ensure newest version (sometimes unnecessary)
2. Analyze, if it isn't broke, don't fix it.
3. Extraction of the deb(s)
4. Re-building of the deb(s) to obtain the error(s) and warning(s)
5. Scanning and fixing of errors(s) and warning(s)
6. Goto #4 until error(s) and warning(s) = 0

Repository maintainers will do the above followed by:
7. Prepare switch followed by repository name
8. Publish switch followed by remote address

EXAMPLES
Extract a deb:
repostorm -e ultamatix-1.9.3_all.deb

Re-build the deb:
repostorm -b ultamatix-1.9.3_all/

Fix errors:
repostorm -f ultamatix-1.9.3_all/

Extract all debs in your current folder into their own sub-folder:
repostorm -e

Prepare known good debs:
repostorm -p themes

Publish known good debs to remote server:
repostorm -P USERNAME:[PASSWORD]@Http://themelinux.com/themes/ themes

Service with autofix option:
repostorm -S on --autofix

FILES

/bin/repostorm
Software for extraction, packaging and maintaining a repository.

/bin/scanpackages
Software for obtaining version information.

/usr/share/man/man1/repostorm.1.gz
The Repository Master manual (this manual)

/usr/share/doc/copywrite
License agreement.

/usr/share/doc/changelog.gz
Repostorm Changelog

/usr/share/man/man1/scanpackages.1.gz
The manual for scanpackages

˜/.gnome2/nautilus-scripts/repostorm
GUI - yad (yet another dialog) based nautilus script front-end command switchable to repostorm.

/etc/bash_completion.d/repostorm
Bash auto completion command interpreter

/usr/share/bash_completion/completions/repostorm
Bash auto completion command interpreter

/usr/share/ultimate_edition/repostorm.pdf
repostorm manual in pdf format.

/usr/share/ultimate_edition/repostorm.ps
repostorm manual in postscript format.

SEE ALSO

gpg(1), **rsync**(1), **ssh**(1)

The full documentation for this tool is maintained as a Tex-info manual. If repostorm and the info program are properly installed at your site, the command

```
info repostorm
```

should give you access to the complete manual including a menu structure and an index.

COPYRIGHT

BUGS

GNU repostorm home page: <http://www.repostorm.com/>. E-mail bug reports to: <theemahn@repostorm.com>. Be sure to include the word repostorm somewhere in the Subject: field."

AUTHOR

Glenn "TheeMahn" Cady <theemahn@repostorm.com>

A MESSAGE FROM THE AUTHOR
Who reads these man pages anyways? The entire purpose for me taking on this project is so I could lose a few of my jobs. Repository management and a few programming tasks in which it currently automates. I would love nothing more then to say my goal is for repostorm to function to the point where 0 errors and 0 warnings get through. Time however, is not and has never been on my side. I have a slew of other projects going on and I am but one Mahn. I hope you enjoy the software. Feel free to send me a beer.

This man page was written and is maintained by TheeMahn, it too has been "repostormed"
Lintian error free you can count on it ;)