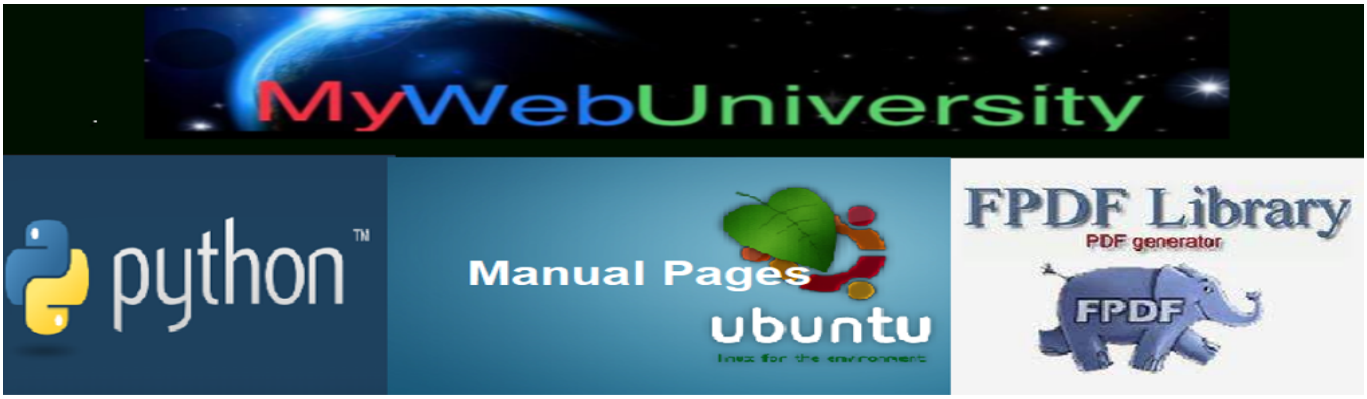

Rocky Enterprise Linux 9.2 Manual Pages on command 'git-mktag.1'

$ man git-mktag.1

GIT-MKTAG(1) Git Manual GIT-MKTAG(1)

NAME

git-mktag - Creates a tag object with extra validation

SYNOPSIS

git mktag

DESCRIPTION

Reads a tag contents on standard input and creates a tag object. The output is the new tag?s <object> identifier.

This command is mostly equivalent to git-hash-object(1) invoked with -t tag -w --stdin.

I.e. both of these will create and write a tag found in my-tag:

```
git mktag <my-tag
git hash-object -t tag -w --stdin <my-tag
```

The difference is that mktag will die before writing the tag if the tag doesn?t pass a git-fsck(1) check.

The "fsck" check done mktag is stricter than what git-fsck(1) would run by default in that all fsck.<msg-id> messages are promoted from warnings to errors (so e.g. a missing "tagger" line is an error).

Extra headers in the object are also an error under mktag, but ignored by git-fsck(1).

This extra check can be turned off by setting the appropriate fsck.<msg-id> varible:

```
git -c fsck.extraHeaderEntry=ignore mktag <my-tag-with-headers
```

OPTIONS

--strict

By default mktag turns on the equivalent of git-fsck(1) --strict mode. Use --no-strict

to disable it.

TAG FORMAT

A tag signature file, to be fed to this command?s standard input, has a very simple fixed format: four lines of

object <hash>

type <typename>

tag <tagname>

tagger <tagger>

followed by some optional free-form message (some tags created by older Git may not have tagger line). The message, when it exists, is separated by a blank line from the header. The message part may contain a signature that Git itself doesn?t care about, but that can be verified with gpg.

GIT

Part of the git(1) suite