

Configure Git

This chapter describes maintainer level git configuration.

Tagged branches used in `ref Documentation/maintainer/pull-requests.rst <pullrequests>` should be signed with the developers public GPG key. Signed tags can be created by passing the `-u` flag to `git tag`. However, since you would *usually* use the same key for the same project, you can set it once with

```
System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\maintainer\ [linux-master] [Documentation] [maintainer] configure-git.rst, line 8); backlink
```

Unknown interpreted text role "ref".

```
git config user.signingkey "keyname"
```

Alternatively, edit your `.git/config` or `~/.gitconfig` file by hand:

```
[user]
    name = Jane Developer
    email = jd@domain.org
    signingkey = jd@domain.org
```

You may need to tell git to use `gpg2`

```
[gpg]
    program = /path/to/gpg2
```

You may also like to tell `gpg` which `tty` to use (add to your shell rc file)

```
export GPG_TTY=$(tty)
```

Creating commit links to lore.kernel.org

The web site <https://lore.kernel.org> is meant as a grand archive of all mail list traffic concerning or influencing the kernel development. Storing archives of patches here is a recommended practice, and when a maintainer applies a patch to a subsystem tree, it is a good idea to provide a `Link:` tag with a reference back to the lore archive so that people that browse the commit history can find related discussions and rationale behind a certain change. The link tag will look like this:

Link: <https://lore.kernel.org/r/<message-id>>

This can be configured to happen automatically any time you issue `git am` by adding the following hook into your git:

```
System Message: WARNING/2 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\maintainer\ [linux-master] [Documentation] [maintainer] configure-git.rst, line 53)
```

Cannot analyze code. No Pygments lexer found for "none".

```
.. code-block:: none

    $ git config am.messageid true
    $ cat >.git/hooks/applypatch-msg <<'EOF'
    #!/bin/sh
    . git-sh-setup
    perl -pi -e 's|^Message-Id:\s*<?([>]+)?\?${Link: https://lore.kernel.org/r/$1|g};' "$1"
    test -x "$GIT_DIR/hooks/commit-msg" &&
        exec "$GIT_DIR/hooks/commit-msg" ${1+"$@"}
    :
    EOF
    $ chmod a+x .git/hooks/applypatch-msg
```