rebuild-from-update-hook

Subject: [HOWTO] Using post-update hook Message-ID: <7vy86o6usx.fsf@assigned-by-dhcp.cox.net> From: Junio C Hamano <gitster@pobox.com> Date: Fri, 26 Aug 2005 18:19:10 -0700 Abstract: In this how-to article, JC talks about how he uses the post-update hook to automate Git documentation page shown at https://www.kernel.org/pub/software/scm/git/docs/. Content-type: text/asciidoc

How to rebuild from update hook ==========

The pages under https://www.kernel.org/pub/software/scm/git/docs/ are built from Documentation/ directory of the git.git project and needed to be kept up-to-date. The www.kernel.org/ servers are mirrored and I was told that the origin of the mirror is on the machine $some.kernel.org, on which I was given an account when I took over Git maintainership from Linus.

The directories relevant to this how-to are these two:

```
/pub/scm/git/git.git/   The public Git repository.
/pub/software/scm/git/docs/ The HTML documentation page.
```

So I made a repository to generate the documentation under my home directory over there.

```
$ cd
$ mkdir doc-git && cd doc-git
$ git clone /pub/scm/git/git.git/ docgen
```

What needs to happen is to update the $HOME/doc-git/docgen/ working tree, build HTML docs there and install the result in /pub/software/scm/git/docs/ directory. So I wrote a little script:

```
$ cat >dododoc.sh <<\EOF
#!/bin/sh
cd $HOME/doc-git/docgen || exit

unset GIT_DIR

git pull /pub/scm/git/git.git/ master &&
cd Documentation &&
make install-webdoc
EOF
```

Initially I used to run this by hand whenever I push into the public Git repository. Then I did a cron job that ran twice a day. The current round uses the post-update hook mechanism, like this:

```
$ cat >/pub/scm/git/git.git/hooks/post-update <<\EOF
#!/bin/sh
#
# An example hook script to prepare a packed repository for use over
# dumb transports.
#
# To enable this hook, make this file executable by "chmod +x post-update".

case " $* " in
*' refs/heads/master '*)
        echo $HOME/doc-git/dododoc.sh | at now
        ;;
esac
exec git-update-server-info
EOF
$ chmod +x /pub/scm/git/git.git/hooks/post-update
```

There are four things worth mentioning:

- The update-hook is run after the repository accepts a "git push", under my user privilege. It is given the full names of refs that have been updated as arguments. My post-update runs the dododoc.sh script only when the master head is updated.

- When update-hook is run, GIT_DIR is set to ’.’ by the calling receive-pack. This is inherited by the dododoc.sh run via the "at" command, and needs to be unset; otherwise, "git pull" it does into $HOME/doc-git/docgen/ repository would not work correctly.

- The stdout of update hook script is not connected to git push; I run the heavy part of the command inside "at", to receive the execution report via e-mail.

- This is still crude and does not protect against simultaneous make invocations stomping on each other. I would need to add some locking mechanism for this.

□□□ #2
□ □□□ □□□ 2026-10-31 20:14:56 UTC
□ □□□ □□□ 2026-10-31 20:14:57 UTC