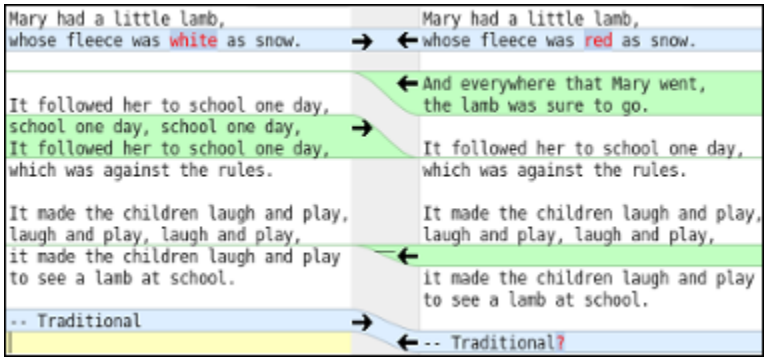
www.spkaa.com
Ph: 888-310-4540

SPK and Associates
900 E Hamilton Ave, Ste.100
Campbell, CA 95008

Integrating Meld with Git under Linux

Have you ever used Git? Git has quickly grown to become one of today's most popular source code management solutions for software engineering projects large, small, professional and personal. Developed by Linus Torvalds, Git is classified as a "distributed version control system (DVCS) with an emphasis on speed."

How about Meld? Meld describes itself as "a visual diff and merge tool targeted at developers," and in this author's opinion, is vastly more intuitive than Git's built-in diff tool. Meld allows versions of documents to be compared side-by-side, with color-coded highlighting and arrows indicating additions, deletions and modifications.



Because Git and Meld are so great at what they do, let's integrate them to work together so that when we need to perform a diff operation Meld is launched automatically with the appropriate files.

If you don't already have them, go ahead and install Git and Meld:

```
sudo apt-get install git-core
sudo apt-get install meld
```

www.spkaa.com
Ph: 888-310-4540

SPK and Associates
900 E Hamilton Ave, Ste.100
Campbell, CA 95008

Git allows us to modify its configuration file in order to specify external tools to use in place of some of its built-in functionality. It's not a difficult thing to modify Git in this way, but there is a slight catch.

First, let's see what happens when we try the straight-forward approach of simply telling Git to use Meld.

```
$ git config --global diff.external meld
```

Now head over to one of your Git repositories and try running "`git diff`" on a modified file.

```
david@Ewok ~ $ cd testclone/
david@Ewok ~/testclone $ git diff README.txt
```

Doesn't work, does it!

```
Usage:
  meld                          Start with an empty window
  meld <file|dir>               Start a version control comparison
  meld <file> <file> [<file>]   Start a 2- or 3-way file comparison
  meld <dir> <dir> [<dir>]      Start a 2- or 3-way directory comparison
  meld <file> <dir>             Start a comparison between file and dir/file

meld: error: too many arguments (wanted 0-4, got 7)
external diff died, stopping at README.txt.
```

Meld should have complained, saying that you gave it too many parameters. By default, Git will actually try to send 7 parameters when you call git diff. Meld only needs 2 parameters -- the names of the files to be compared. Let's take a closer look and see what Git is passing to Meld so that we can fix this situation.

Just for the purposes of seeing what parameters are being passes, create a test script named "`params.sh`" with the following 2 lines:

```
#!/bin/bash
echo $*
```

Make the script executable…

```
$ chmod +x params.sh
```

www.spkaa.com
Ph: 888-310-4540

SPK and Associates
900 E Hamilton Ave, Ste.100
Campbell, CA 95008

If we instruct git diff to call this test script we can see what parameters it is trying to pass:

```
david@Ewok ~/testclone $ git config --global diff.external /home/david/testclone
/params.sh
```

Try running the diff again

```
david@Ewok ~/testclone $ git diff README.txt
```

You should see seven values which correspond to the following:

```
[path] [old-file] [old-hex] [old-mode] [new-file] [new-hex] [new-mode]
```

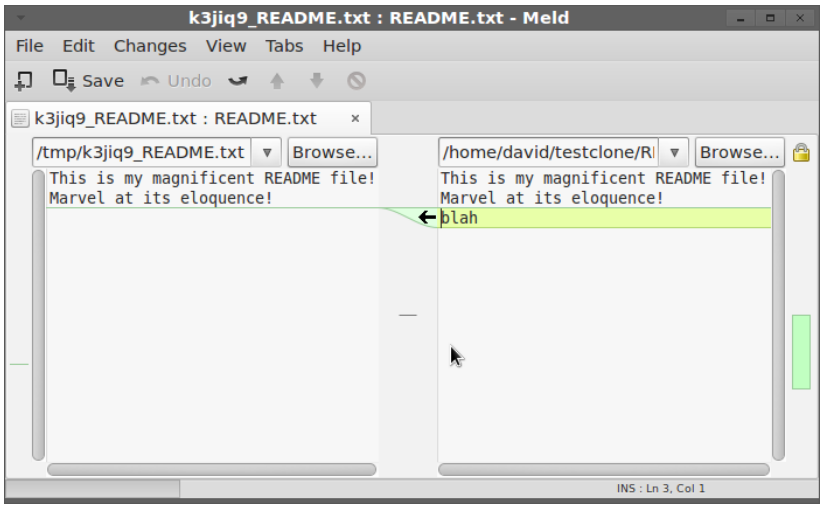
As I mentioned, Meld only needs two parameters to launch correctly -- the names of the two files being compared. In this case, we want to send Meld the **second and fifth parameters** being provided by Git. To do this, we will create a very simple wrapper script, named diff.sh.

```
#!/bin/bash
meld "$2" "$5"
```

Make the script executable and instruct Git to call it when running git diff:

```
$ git config --global diff.external /home/david/Scripts/diff.sh
```

Now, when you run git diff, Meld should open with the files you are seeking to compare.



www.spkaa.com
Ph: 888-310-4540

SPK and Associates
900 E Hamilton Ave, Ste.100
Campbell, CA 95008

If for some reason you would like to run the default git diff program after having made this modification you can add `the --no-ext-diff` flag when running the git diff command. This will execute the default git diff program while preserving Meld as the default setup.

```
david@Ewok ~/testclone $ git diff --no-ext-diff README.txt
diff --git a/README.txt b/README.txt
index 2769429..8265164 100644
--- a/README.txt
+++ b/README.txt
@@ -1,2 +1,3 @@
 This is my magnificent README file!
 Marvel at its eloquence!
+blah
```

If you want to undo your modification completely, type the following:

```
$ git config --global --unset diff.external
```

You could also manually remove the following lines from the `~/.gitconfig` file to produce the same effect:

```
[diff]
      external = <path to your script>
```

David Hubbell
SPK Software Engineer