(/linux/) linux (https://www.baeldung.com/linux/)

Find and Convert Files Ending With CRLF

Last modified: March 6, 2021

by baeldung (https://www.baeldung.com/linux/author/baeldung)

Files (https://www.baeldung.com/linux/category/files)

awk (https://www.baeldung.com/linux/tag/awk)

cat (https://www.baeldung.com/linux/tag/cat)

grep (https://www.baeldung.com/linux/tag/grep)

sed (https://www.baeldung.com/linux/tag/sed)

you're interested in sharing that experience with the community, have a look at our **Contribution Guidelines** (/contribution-guidelines).

1. Overview

In Linux, files that end with a carriage return and line feed (CRLF) can lead to some troubles when processing.

In this tutorial, we'll learn how to find those files and convert the line endings to LF.

2. Creating an Example File

First, let's create a set of example files to test our strategies:

```
$ mkdir -p /tmp/test_folder
```

With that, we've created our test folder. Let's fill it with two files ending with CRLF and LF, respectively:

```
$ printf "Hi \r\n" | tee /tmp/test_folder/crlf_ending{1,2}
Hi
 $ printf "Hi \n" | tee /tmp/test_folder/lf_ending{1,2}
Hi
```

In the first one-liner, we've created two files called *crlf_ending1* and *crlf_ending2*. **Both files are filled with the message "Hi " and followed by a CRLF line ending**.

In the second, **we've made it the same but with LF line endings and their respective filenames**.

3. Searching for Files With CRLF Endings

There are a few ways to find files ending with CRLF using Linux commands.

3.1. *cat*

Let's start with the *cat* command (https://www.gnu.org/software/coreutils/manual/html_node/cat-invocation.html#cat-invocation):

```
$ cat -A /tmp/test_folder/{crlf_ending1,lf_ending1}
Hi ^M$
Hi $
```

Here, **we can see the difference in both files by using the *-A* parameter that makes *cat* display non-printing characters**.

3.2. *grep*

Now, **let's try with the *grep* command (/linux/common-text-search#the-grep-command)**:

```
$ grep -rIl -m 1 $'\r' /tmp/test_folder/
/tmp/test_folder/crlf_ending2
/tmp/test_folder/crlf_ending1
```

Let's take a look at the arguments:

- *-r*, to read the entire folder recursively
- *-I*, to ignore binary files
- *-l*, to print only the name of the matching file
- *-m 1*, to stop reading after the first match

3.3. *file*

Additionally, **we can use the *file* command (https://linux.die.net/man/1/file) to extract information**:

```
$ file /tmp/test_folder/*
/tmp/test_folder/crlf_ending1: ASCII text, with CRLF line
terminators
/tmp/test_folder/crlf_ending2: ASCII text, with CRLF line
terminators
/tmp/test_folder/lf_ending1:   ASCII text
/tmp/test_folder/lf_ending2:   ASCII text
```

3.4. *dos2unix*

The *dos2unix* command (https://linux.die.net/man/1/dos2unix) is a great tool to achieve this task **but isn't always installed in all Linux distributions**.

To install it on Debian-based systems, we can type:

```
$ sudo apt-get install dos2unix
```

Next, **for getting information about the line endings, let's use a nice feature included with the *-i* parameter**:

```
$ dos2unix -i /tmp/test_folder/*
       1       0       0  no_bom    text    /tmp/test_folder
/crlf_ending1
       1       0       0  no_bom    text    /tmp/test_folder
/crlf_ending2
       0       1       0  no_bom    text    /tmp/test_folder
/lf_ending1
       0       1       0  no_bom    text    /tmp/test_folder
/lf_ending2
```

Here, in the first and second columns, we can see the number of DOS and Unix line breaks, respectively.

4. Convert CRLF to LF

Now that we've learned how to identify files with CRLF line endings, let's use some tools to convert CRLF to LF.

4.1. *sed* Command

The *sed* command (/linux/sed-editor) is a great tool for text processing. Let's use it to find and replace the line endings in the *crlf_ending1* file:

```
$ sed 's/\r//' /tmp/test_folder/crlf_ending1 | cat -A -
Hi $
```

In this example, we're using the *sed* command *'s/\r//'* to find and replace the *\r* character with an empty value.

Additionally, **if we want to edit the file inline, we can use the *-i* parameter**.

Finally, by using the *cat* command, we can see the final output without the *^M* character.

4.2. *tr* Command

The *tr* command (/linux/tr-manipulate-strings) is a simple and powerful tool that can remove or translate characters.

Let's use the parameter *-d* to remove the *\r* character:

```
$ tr -d '\r' < /tmp/test_folder/crlf_ending1 | cat -A -
Hi $
```

4.3. *awk* Tool

Also, **we can use the *awk* tool (/linux/awk-guide) to remove the *\r* character**:

```
$ awk 'gsub(/\r/,"")' /tmp/test_folder/crlf_ending1 | cat -A -
Hi $
```

Here, we're using the *gsub* function (/linux/grep-sed-awk-differences#3-substituting-the-matching-string) to make the replacement. Then, **by omitting the action, *awk* prints the entire record with the substitution**.

4.4. Perl

We can also use the Perl interpreter (https://perldoc.perl.org/perlrun) as in our scenario with *sed*:

```
$ perl -pe 's/\r//' /tmp/test_folder/crlf_ending1 | cat -A -
Hi $
```

Let’s take a close look at the parameters:

- *-p*, for reading each line
- *-e ‘s/\r//’*, to enter the script that will delete the *\r* character

4.5. *dos2unix*

Again, **we can use the *dos2unix* tool to keep things simple**.

Now, let’s use it in an example file:

```
$ dos2unix /tmp/test_folder/crlf_ending1
dos2unix: converting file /tmp/test_folder/crlf_ending1 to Unix
format...
```

Let’s take a look at the content of the file:

```
$ cat -A /tmp/test_folder/crlf_ending1
Hi $
```

We can see that the CRLF line ending has been converted to LF.

Finally, let's recover our file:

```
$ unix2dos /tmp/test_folder/crlf_ending1
unix2dos: converting file /tmp/test_folder/crlf_ending1 to DOS
format..
```

As a final note, **if we only want to see the content converted but without actually changing the file, we can use a redirection (/linux/pipes-redirection#reading6)**:

```
$ dos2unix < /tmp/test_folder/crlf_ending1 | cat - -A
Hi $
```

4.6. *recode*

recode (https://linux.die.net/man/1/recode) **is an interesting tool that converts files between character sets**.

Let's use it on our file:

```
$ recode CP1252...UTF-8 /tmp/test_folder/crlf_ending1
```

Here, **we've converted our file from CP1252 (or Windows-1252) encoding to UTF-8**.

Now, let's see the content:

```
$ cat -A /tmp/test_folder/crlf_ending1
Hi $
```

Finally, let's convert our file to the previous encoding:

```
$ recode UTF-8...CP1252 /tmp/test_folder/crlf_ending1
```

4.7. Using the Vim Editor

To convert the CRLF line endings to LF with *vim* (/linux/vi-editor), let's open our file:

```
$ vim /tmp/test_folder/crlf_ending1
```

Now, we can type ESC + ":" to enter command mode.

Then, we'll type ***set ff=unix*** and press ENTER.

Finally, let's press ESC + *ZZ* to exit and save the file.

Let's see the content:

```
$ cat -A /tmp/test_folder/crlf_ending1
Hi $
```

To recover our file, we can repeat the previous steps but type *set ff=dos* instead.

4.8. Using the Bash Builtins

Finally, **let's use some bash builtins to convert the line endings**:

```
$ while read line
do
    echo "${line/$'\r'/}"
done < /tmp/test_folder/crlf_ending1 | cat -A
```

As a result, we should see:

```
Hi $
```

In this scenario, **we've fed the *while* loop (/linux/vi-editor) with our test file**. Then, **we've used *parameter expansion* (/linux/vi-editor) to delete the *\r* character**.

5. Find and Convert Files at the Same Time

Now that we know how to find files with CRLF line endings and convert them to LF, we can combine these operations.

First, **we can avoid finding files and, instead, apply commands like *dos2unix*, or *sed*, directly to an entire folder or pattern**:

```
$ dos2unix /tmp/test_folder/crlf_ending*
dos2unix: converting file /tmp/test_folder/crlf_ending1 to Unix
format...
dos2unix: converting file /tmp/test_folder/crlf_ending2 to Unix
format...
```

With *sed:*

```
$ sed -i 's/\r//' /tmp/test_folder/crlf_ending*
```

But, **if we want to convert only the files with CRLF endings, we can combine some tools using the *xargs* command (/linux/xargs)**:

```
$ grep -rIl -m 1 $'\r' /tmp/test_folder/ | xargs -P0 -I {}
dos2unix {}
```

Let's use another combination:

```
$ file /tmp/test_folder/* \
    | awk -F : '/CRLF/ && $0=$1' \
    | xargs -P0 -I {} sed -i 's/\r//' {}
```

Here, we're using *awk* **to list only the name of each file containing just CRLF endings**.

6. Conclusion

In this article, we've learned how to identify files with CRLF line endings.

Then, we've looked at how to convert the line endings from CRLF to LF.

And finally, we combine these strategies for finding and converting the files in a one-liner.

If you have a few years of experience in the Linux ecosystem, and you're interested in sharing that experience with the community, have a look at our **Contribution Guidelines** (/contribution-guidelines).

Comments are closed on this article!

CATEGORIES

ADMINISTRATION (/LINUX/CATEGORY/ADMINISTRATION)

FILES (/LINUX/CATEGORY/FILES)

FILESYSTEMS (/LINUX/CATEGORY/FILESYSTEMS)

INSTALLATION (/LINUX/CATEGORY/INSTALLATION)

NETWORKING (/LINUX/CATEGORY/NETWORKING)

PROCESSES (/LINUX/CATEGORY/PROCESSES)

SCRIPTING (/LINUX/CATEGORY/SCRIPTING)

SEARCH (/LINUX/CATEGORY/SEARCH)

SECURITY (/LINUX/CATEGORY/SECURITY)

WEB (/LINUX/CATEGORY/WEB)

SERIES

LINUX FILES (/LINUX/LINUX-FILES-SERIES)

LINUX SCRIPTING (/LINUX/LINUX-SCRIPTING-SERIES)

ABOUT

ABOUT BAELDUNG (/ABOUT)
THE FULL ARCHIVE (HTTPS://WWW.BAELDUNG.COM/LINUX/FULL_ARCHIVE)
WRITE FOR BAELDUNG (/CONTRIBUTION-GUIDELINES)
EDITORS (HTTPS://WWW.BAELDUNG.COM/EDITORS)

TERMS OF SERVICE (HTTPS://WWW.BAELDUNG.COM/TERMS-OF-SERVICE)
PRIVACY POLICY (HTTPS://WWW.BAELDUNG.COM/PRIVACY-POLICY)
COMPANY INFO (HTTPS://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO)
CONTACT (/CONTACT)