

# Changing XML property values with SED and REGEX

August 14, 2013

I'm currently working on a project where we need to be migrating an application from one environment to another. Because of that, some configuration properties will change, like the database connection for example. I wanted to keep the main configuration file intact and only replace some of the values, but then it became a bit painful. First, after looking around for tooling a bit, I settled on sed which appeared to be the right tool for the job. As they describe it themselves:

*"Sed is a stream editor. A stream editor is used to perform basic text transformations on an input stream (a file or input from a pipeline)."*

Next, here is what my property file looked like:

```
<property name="hibernate.connection.password">password</property>
<property name="hibernate.connection.url">url</property>
<property name="hibernate.connection.username">username</property>
```

So I didn't know the values from the configuration part, I only knew the property names (or tag) from those elements. To replace "hibernate.connection.password" with the actual password for example, the following command will do the trick.

```
sed -i '' 's/\(<[^\"]*"hibernate.connection.password">\)\([^\<]*\)\(<[^\>]*\)/\1newpassword\3/g' confluence.cfg.xml
```

To break it down: -i stands for 'edit files in place', by default sed writes it's output to a new file. In my case I want the output to stay in the same file. In order to use -i you have to supply a backup file, If you don't want to submit one, like in my example, leave it out by using two single quotes. Your total sed command needs to be between single quotes.

```
s/
```

This stands for search. A normal find and replace sed command could be:

```
sed -i 's/old/new/g' file.txt
```

The next step in my command is defining regular expression groups. The first part is to capture <property name="hibernate.connection.password">. These groups need to be escaped.

```
\(<[^\"]*"hibernate.connection.password">\)
```

The second part is the actual value, which could be anything...

```
\([^\<]*\)
```

The last part is what's after the value, ending with ">".

```
\(<[^\>]*\)
```

What we want to replace is the second part, we want to keep the first and the last parts as they are.

```
/\1newpassword\3
```

So this will result in:

```
<property name="hibernate.connection.password">newpassword</property>
```

We want to search the entire file so we use the /g global flag. The last argument of the command (e.g. 'confluence.cfg.xml' or 'file.txt') is used to identify the file we want to update.