

Groovy: How to define a manipulated string as a variable?

I'm writing a Jenkins pipeline as a code for an Android application.

I have the following strings:

```
BUILDFLAV = "Staging"
BUILDTYPE = "Debug"
```

I want to define another variable called artifact_name which will look like:

```
product_name-BUILDFLAV(in lowercase)-BUILDTYPE(in lowercase).apk
```

for example:

```
App01-staging-debug.apk
```

Since I'm new to groovy i'm not sure that I'm doing it correctly.

I've defined 2 variables:

```
aa = BUILDFLAV[0].toLowerCase()+BUILDFLAV.substring(1)
bb = BUILDTYPE[0].toLowerCase()+BUILDTYPE.substring(1)
```

This allows me to change the first char of each string from uppercase to lowercase.

Then, I want the artifact_name variable to look like so:

```
App01-${aa}-${bb}.apk
```

So I've tried something like that:

```
artifact_name =
"App01-${BUILDFLAV[0]}.toLowerCase()+${BUILDFLAV}.substring(1)+${BUILDTYPE[0]}.toLowerCase()
```

But it throws me an error.

Doing this, however, worked but seems like there's a more elegant way to do that:

If you ask yourself why I'm not setting the variables in lowercase from the beginning is because the gradle task which is run in the build is using the first letter uppercase convention and what I'm trying to achieve is related to a more advanced step of the build.

Is it possible to achieve the same result by doing what I did in vars "aa" and "bb" in just one line? like the above example with artifact_name...

If you want to use templates in strings in groovy, you can put it inside `${..}` like you already did. However you need to put everything that needs to be executed inside it.

So use

```
ARTIFACT_NAME =
"App01-${BUILDFLAV[0].toLowerCase()}${BUILDFLAV.substring(1).toLowerCase()}-${BUILDTYPE[0].to
```

But please note that it is unnecessary to cast the first char to lowercase and then appending the rest. The next will give the same result

```
`ARTIFACT_NAME = "App01-${BUILDFLAV.toLowerCase()}-${BUILDTYPE.toLowerCase()}.apk"`
```

If you for whatever reason want to do it like that anyway, you can use `BUILDTYPE[1..-1]` instead of `BUILDTYPE.substring(1)`

Dumbed down/different way of explaining Rik's answer for anyone that wants to know how to do this on a general-level:

1. Acquire a plugin that allows you to inject variables via Groovy scripts
2. In the `Evaluated Groovy script` field, replace variables wrapped with `%` like `%Var%` with their actual value with this following block of text:

```
def map = [%New_Var%: %Old_Var%.toUpperCase()]
return map
```

Then you can verify it's working for Linux as: `echo $New_Var`
And Windows as: `echo %New_Var%`

As an example,

Let's say you have a parameter named `Label` , that you've set to `Hello`

```
def map = [ucase_Label: Label.toUpperCase()]
return map
```

```
echo $ucase_Label
```

```
HELLO
```

And if you need the value appended/prepended, you can do stuff like,

```
def map = [newString: "some_String"+Label.toUpperCase()]
```

edited Dec 14 '17 at 0:13

answered Dec 14 '17 at 0:06



kayleeFrye_onDeck

2,587 ● 2 ● 19 ● 36

