# Adventures in Groovy – Part 5: Accessing Substitution Variables

## Introduction

Accessing Substitution Variables is critical in most calculations, and accessing them in Groovy is a little more complex than it needs to be with not having an API to get them.  Since the SubstitutionVariable is not available, there are a couple ways to get them.  The precursor to this post is three-fold.

1. Read the Bug Report: Groovy SubstitutionVariable Class Not Functioning  post on Jan 8, 2018 regarding the SubstitutionVariable class availability.
2. Thanks to *Abhi* for providing a great alternative.
3. It may be helpful to read Adventures in Groovy Part 4: Run Time Prompts to understand how to access RTPs in a Groovy calculation.

In my bug report above, I suggested grabbing them via a hidden column or row from a form.  A reader suggested a another way to do this, and I think it is a better way to accomplish it.  Rather than grabbing the substitution variable by adding it to the form and hiding the column/row from the user, Abhi provided a much cleaner approach to working around not having access to the SubstitutionVariable class by using hidden RTPs.

## Create Run Time Prompts to Access Substitution Variables

Assume the following 3 variables are required in business rules.  Create a new RTP for each.  The naming convention is

irrelevant, but should be considered and be consistent for easy reference in the business rules.  In this read, I have assumed there isn't an existing RTP with the defaults set to a substitution variable.  Even if there is, it might be beneficial to create ones specifically for this need so future changes don't impact the values.

**Name:** subVar_CurMonth
**Type:** Member
**Dimension:** Period
**Default Value:** &v_CurMonth
**RTP Text:** N/A

**Name:** subVar_CurYear
**Type:** Member
**Dimension:** Period
**Default Value:** &v_CurYear
**RTP Text:** N/A

**Name:** subVar_BudYear
**Type:** Member
**Dimension:** Period
**Default Value:** &v_BudYear
**RTP Text:** N/A

## Business Rule Inclusion

Inside the business rule, the following convention is required to add the variables.

/*RTPS: {subVar_CurMonth subVar_CurYear subVar_BudYear}*/

Set all the RTPs in the Variables tab to *set to hidden* so the user isn't prompted for these.  Now, the substitution variables can be referenced.

```
def varCurMonth = rtps.subVar_CurMonth.toString()
def varCurYear = rtps.subVar_CurYear.toString()
def varBudYear = rtps.subVar_BudYear.toString()
```

## Conclusion

Since these are likely to be used in many rules, it would be beneficial to add these to a script and embed that script into the rules that need to access these.  Any new variable that needs to be included can be added to the script, and all the business rules would then have access to them.  There are a number of ways to do this with Groovy calculations, but the simplest way is to embed it like a non Groovy business rule.  This can be dragged from the left pane, or entered manually.  The syntax is

```
%Script(name:="script    name",application:="application Name",plantype:="plantype name"
```

If and when Oracle releases the class that provides direct access to sub vars, expect it to be documented here.