Adventures in Groovy – Part 45: Locking Cells On Load

Often times, there is a need to lock cells from a user outside of the traditional dimensional security methods. Maybe you don't want to add dimensional security to accounts so your data push allows users to push data to another cube. Or, maybe there is more complicated logic to identify who can and who can't edit a cell. The logic to do with is very simple.

Locking Cells

If you have read many of my articles, you likely have read about how to iterate though cells with the data grid iterator. If you haven't, starting with Part 3 would be a good start. The grid iterator allows the script to loop through all cells, edited cells, or a filtered list of cells. This is traditionally used to execute logic after a user edits a form, but it can be used to lock cells when the form is loaded. Whether it is simply specific accounts, products, or a more complicated filter like products that have or don't have a specific attribute, it quite easy to do.

Following are two ways to lock the cells. The first is more dynamic. This could do all kinds of things, like lock the cell if it was above a specific value, or if it had a specific attribute, or even something silly, like if it the user's name was Kyle!

```
// identify the rows to be locked in a list collection
List<String> lockCells = ['Net Income']
// lock the cells
operation.grid.dataCellIterator().each {
     // Lock the cells if the measure member of the cell is in
the lockCells list
     if(lockCells.contains(it.getMemberName("Measures"))){
          it.setForceReadOnly(true)
     }
}
```

The next example would do the same thing as above, but would filter the cells the iterator loops on and lock all the cells that meet the filter applied.

```
// identify the rows to be locked in a list collection
List<String> lockCells = ['Net Income']
// lock the cells
operation.grid.dataCellIterator(lockCells.join(',')).each {
     it.setForceReadOnly(true)
}
```

Add To The Form

Now that the rule is built, the only thing left is to add this to the form in which it should run. The key is to add this rule and check the Run After Load checkbox. This will allow the form to load (or render), and then run the rule, which in this case, will change predefined cell properties so that it is not editable.

That’s A Wrap

This is not the longest post I have ever made, but it doesn’t mean it is less useful. Locking cells based on criteria, criterial we never had access to before, is quite interesting. When security is set to lock a cell from a user through dimensional security, it also means the users can’t use data pushes to move that calculated data to other cubes, which is an issue. Think about adding an attribute or UDA to those accounts and then building a common script that runs on all forms that locks the cells with that attribute or UDA. It wouldn’t be any more difficult to build, and it can be automated through metadata builds. Hmmm, wouldn’t that be nice!