Adventures in Groovy – Part 29: Troubleshooting Data Movement With GridBuilder

One of the challenges working with grids is validating the results. As with an Essbase calculation, Smart View reports are developed to validate results. The same happens when calculations, or data movement, is executed in Groovy via grids. When the results in Smart View aren't accurate, where do you go?

Make It Simple

By now you have probably used the grid iterator to iterate through cells to validate data, write values to the log, and check to see if the cells have been edited. The same can be done with GridBuilders. All aspects of the grid can be logged. If done correctly, this can be copied directly from the log and pasted into excel to accomplish 2 things. One, you have a report in Excel. Two, you have a Smart View ready ad-hoc report that can be refreshed.

Send The Grid To The Log

Grids can be different so this may be a start for you to construct this validation. This example has 2 column headers. The rest should be very close and likely completely reusable. To break this down, we have a source grid we are pulling data from to create a grid to send to another plan type. Basically, this loops through the members in the POV and replicates rows for the number of columns in the grid. This is repatative, but it will provide a retrievable Smart View.

```
// Loop through the POV to create column headers for each
povmbrs*.essbaseMbrName.each{ POV ->
```

```
  // Add a blank column for the row members
  print ','
  // Loop through the columns and repeat the POV member for each of the columns
  colmbrs[0]*.essbaseMbrName.size().times{
    print POV + ','
  }
  // Print a line return for the next POV member
  println ''
}
//Print a blank column and then each of the column headers for both headers
println ',' + colmbrs[0]*.essbaseMbrName.join(',')
println ',' + colmbrs[1]*.essbaseMbrName.join(',')
```

At this point, the log will show the column headers. The following is created while looking through the source grid and produces the row header and the respective data for each of the column headers.

```
...{
sValues.add(it.crossDimCell(cMonth.toString(),cCurrency.toString()).data)
                                    addcells                <<
it.crossDimCell(cMonth.toString(),cCurrency.toString()).data
}
// After the variables are created with the numeric data to be used when creating the rows, the row is created
finGrid.addRow([acctMap.get(it.getMemberName('Account'))],addcells)
// Print to the log exactly what is being used to create the grid
println  "${it.getMemberName('Account')}"  +  ","  +  sValues.join(",")
```

At this point, the entire Smart View is created in the log and can be copied and pasted to Excel. The log will look something like this.

```
,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,
,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_
Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,
,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_
Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,
,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,
,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel
,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,
,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_M
aterial_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_
Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,
,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_
Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,
,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Ve
ndor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,
,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,
,Jan,Jan,Feb,Feb,Mar,Mar,Apr,Apr,May,May,Jun,Jun,Jul,Jul,Aug,Aug,Sep,Sep,Oct,Oct,Nov,Nov,Dec,Dec
,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD
Regular_Cases,29285.461814637485,29259.494863633503,33368.95540139619,33408.64281552195,28298.29488097554,28291.101796096253,25615.573257959983,25615.5732580
35547,28035.660265185812,28035.66026502542,29636.513952507612,29636.51395261425,22393.9555118016,22393.955511454235,25103.29666961775,25103.29667000104,2643
1.73291933657,26431.732919391456,28848.81784568152,28848.81784568891,27616.877481297608,27616.87748098876,36958.82821814836,36958.82821812173
42001,2936008.606423491,2925202.2292920765,4080740.4618713213,4096141.298239027,4689344.985135378,4686701.1528344,4167923.3517984436,4167923.7510451903,43612
74.343641649,4361275.342330161,4911901.237603318,4911901.636334188,3326686.227839543,3326686.2283058898,4032492.861675521,4032492.86188431,4261800.877750758,
4261800.877360412,4915612.549497408,4915612.549622078,4426064.761494611,4426064.762909748,6310893.665620634,6310893.665684896
50001,2404534.4273910057,2396133.63779072,3277555.8035608097,3290201.454582131,3859438.6810221463,3857234.524439308,3457334.091902443,3457334.4928352744,3675
190.0079336124,3675191.0080927587,4102659.6340585584,4102660.0349632725,2704268.066182739,2704268.0665583606,3324911.393864256,3324911.393865494,3531968.0909
88797,3531968.090982434,4126231.8168021776,4126231.8165512322,3622030.8157288805,3622030.8140955963,5380144.519972735,5380144.520088039
50015,20.0,20.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10
000.0,10000.0,10000.0
56010,-117916.70752999028,-117753.5148925302,-51471.05673797781,-51940.00521266759,-177804.4014007446,-177536.0797889963,-171191.98845515805,-171191.58839171205
,-210721.84894863682,-210720.8497169004,-232756.95726883595,-232756.55714092613,-115522.11854760228,-115522.11861207416,-155392.93497319758,-155392.93506424737,-
180238.2747354005,-180238.2747977321,-237371.91341150456,-237371.91341691514,-129544.6791353713,-129544.67921221143,-362391.97182807245,-362391.971846169
50010,0.0,0.0,50.0,50.0,0.0,0.0,0.0,0.0,0.0,40.0,40.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56055,0.0,0.0,0.0,0.0,5.0,5.0,5.0,5.0,2.0,2.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56300,1.0,1.0,1.0,1.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56092,0.0,0.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0
56230,112465.0,112465.0,100486.0,100486.0,100026.0,100026.0,26.0,26.0,4.0,4.0,3.0,3.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56200,60106.0,60106.0,40135.0,40135.0,30005.0,30005.0,9.0,9.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56205,58.0,58.0,39.0,39.0,9.0,9.0,8.0,8.0,5.0,5.0,5.0,5.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56090,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
```

When pasted into a text editor that doesn’t wrap, it looks a little more palatable!

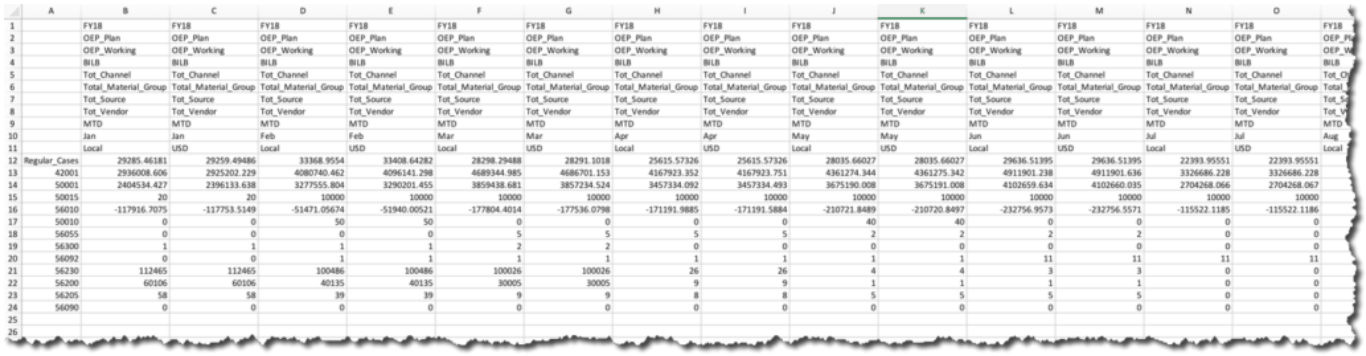
```
,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,FY18,
,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,OEP_Plan,O
,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Working,OEP_Wo
,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,BILB,
,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Channel,Tot_Cha
,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_Group,Total_Material_G
,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Source,Tot_Sourc
,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vendor,Tot_Vend
,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,MTD,
,Jan,Jan,Feb,Feb,Mar,Mar,Apr,Apr,May,May,Jun,Jun,Jul,Jul,Aug,Aug,Sep,Sep,Oct,Oct,Nov,Nov,Dec,Dec
,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD,Local,USD
Regular_Cases,29285.461814637485,29259.494863633503,33368.95540139619,33408.64281552195,28298.29488097554,28291.101796096253,25615.573257959983,25615.573258035547,2
42001,2936008.606423491,2925202.2292920765,4080740.4618713213,4096141.298239027,4689344.985135378,4686701.1528344,4167923.3517984436,4167923.7510451903,4361274.343
50001,2404534.4273910057,2396133.63779072,3277555.8035608097,3290201.454582131,3859438.6810221463,3857234.524439308,3457334.091902443,3457334.4928352744,3675190.00
50015,20.0,20.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,10000.0,1000
56010,-117916.70752999028,-117753.5148925302,-51471.05673797781,-51940.00521266759,-177804.4014007446,-177536.0797889963,-171191.98845515805,-171191.58839171205,-21
50010,0.0,0.0,50.0,50.0,0.0,0.0,0.0,0.0,0.0,40.0,40.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56055,0.0,0.0,0.0,0.0,5.0,5.0,5.0,5.0,2.0,2.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56300,1.0,1.0,1.0,1.0,2.0,2.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56092,0.0,0.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0,11.0
56230,112465.0,112465.0,100486.0,100486.0,100026.0,100026.0,26.0,26.0,4.0,4.0,3.0,3.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56200,60106.0,60106.0,40135.0,40135.0,30005.0,30005.0,9.0,9.0,1.0,1.0,1.0,1.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56205,58.0,58.0,39.0,39.0,9.0,9.0,8.0,8.0,5.0,5.0,5.0,5.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
56090,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
```

Create An Excel Report / Smart View Ad-hoc

At this point, you are ready to get this into Excel. Select the part of the log related to the validation and copy it to Excel. I normally wrap the Groovy code above in a few println statements so it is easier to identify what is related to this validation effort and what isn’t.

```
println '*************** BEGIN VALIDATION ***************'
// CODE ABOVE
println '*************** END VALIDATION ***************'
```

If it doesn’t parse by comma automatically, go to the Data ribbon and select the option to convert the selection to Text to Columns and select comma. This will parse it to what is required. This may not need to be done depending on a few things, which won’t be discussed here. The result of the example above looks like this.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1		FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18	FY18
2		OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Plan	OEP_Pl
3		OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_Working	OEP_W
4		BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB	BILB
5		Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Channel	Tot_Ch
6		Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_Material_Group	Total_
7		Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_Source	Tot_So
8		Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Vendor	Tot_Ve
9		MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD	MTD
10		Jan	Jan	Feb	Feb	Mar	Mar	Apr	Apr	May	May	Jun	Jun	Jul	Jul	Aug	
11		Local	USD	Local	USD	Local	USD	Local	USD	Local	USD	Local	USD	Local	USD	Local	
12	Regular_Cases	29285.46181	29259.49486	33368.9554	33408.64282	28298.29488	28291.1018	25615.57326	25615.57326	28035.66027	28035.66027	29636.51395	29636.51395	22393.95551	22393.95551		
13	42001	2936008.606	2925202.229	4080740.462	4096141.298	4689344.985	4686701.153	4167923.352	4167923.751	4361274.344	4361275.342	4911901.238	4911901.636	3326686.228	3326686.228		
14	50001	2404534.427	2396133.638	3277555.804	3290201.455	3859438.681	3857234.524	3457334.092	3457334.493	3675190.008	3675191.008	4102659.634	4102660.035	2704268.066	2704268.067		
15	50015	20	20	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000	10000		
16	56010	-117916.7075	-117753.5149	-51471.05674	-51940.00521	-177804.4014	-177536.0798	-171191.9885	-171191.5884	-210721.8489	-210720.8497	-232756.9573	-232756.5571	-115522.1185	-115522.1186		
17	50010	0	0	50	50	0	0	0	0	40	40	0	0	0	0		
18	56055	0	0	0	0	5	5	5	5	2	2	2	2	0	0		
19	56300	1	1	1	1	2	2	0	0	0	0	0	0	0	0		
20	56092	0	0	1	1	1	1	1	1	1	1	11	11	11	11		
21	56230	112465	112465	100486	100486	100026	100026	26	26	4	4	3	3	0	0		
22	56200	60106	60106	40135	40135	30005	30005	9	9	1	1	1	1	0	0		
23	56205	58	58	39	39	9	9	8	8	5	5	5	5	0	0		
24	56090	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
25																	
26																	

Finishing

Now there is an easy viewable report of what is being used to create the grid. If this data is incorrect, move backwards in the process to the source grid and fix it. This should provide all the information to do that. Is the POV correct? For me, this is normally the issue – I am pulling the wrong POV. Once the source grid POV is changed, go through the process again and you should see better results.