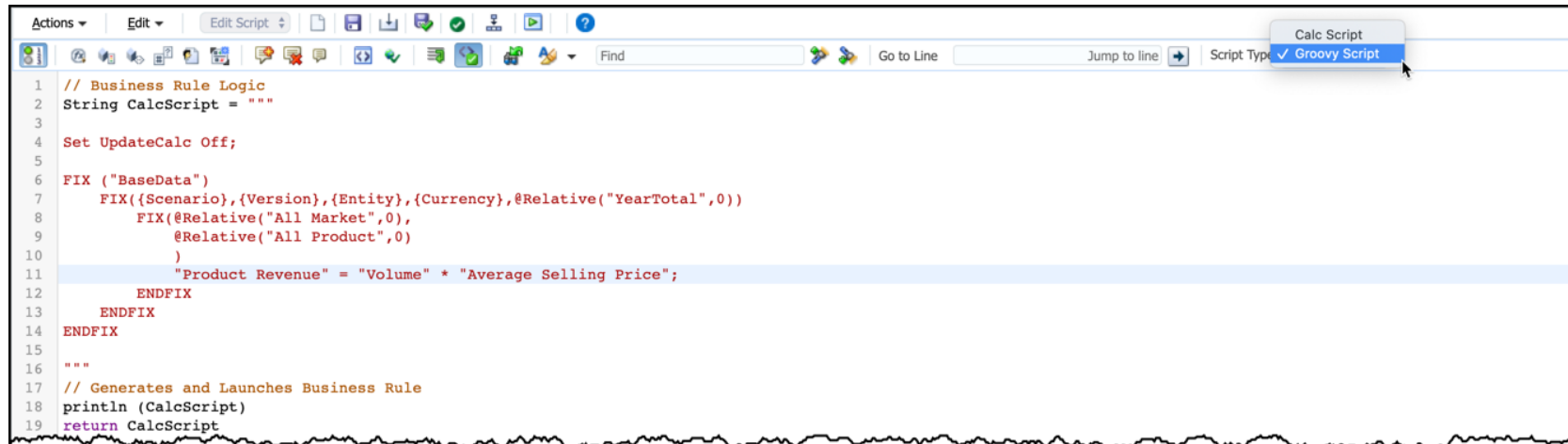


Groovy Rules

- Groovy vs Business Rules
 - Groovy is a framework for Java
 - But in EPM Cloud, it's built-in, EPM aware and easier to run with one caveat
 - Requires one (1) framework to be enabled
 - Faster and better than Business Rules?
- Calculation Manager
 - Manage both Groovy and Business Rules



The screenshot shows the EPM Cloud Calculation Manager interface. The top toolbar includes buttons for 'Actions', 'Edit', 'Edit Script', 'Find', 'Go to Line', 'Jump to line', and 'Script Type'. The 'Script Type' dropdown menu is open, showing 'Calc Script' and '✓ Groovy Script'. The main editor area displays a Groovy script for a business rule logic. The script is as follows:

```
1 // Business Rule Logic
2 String CalcScript = ""
3
4 Set UpdateCalc Off;
5
6 FIX ("BaseData")
7     FIX({Scenario},{Version},{Entity},{Currency},{Relative("YearTotal",0)})
8         FIX(@Relative("All Market",0),
9             @Relative("All Product",0)
10        )
11     "Product Revenue" = "Volume" * "Average Selling Price";
12 ENDFIX
13 ENDFIX
14 ENDFIX
15
16 ""
17 // Generates and Launches Business Rule
18 println (CalcScript)
19 return CalcScript
```

Run Your 1st Groovy rule

- Basic Rule vs Groovy Rule

```
1 Set UpdateCalc Off;
2
3 FIX ("BaseData")
4     FIX({Scenario},{Version},{Entity},{Currency},@Relative("YearTotal",0))
5         FIX(@Relative("All Market",0))
6         FIX(@Relative("All Product",0))
7             "Product Revenue" = "Volume" * "Average Selling Price";
8         ENDFIX
9     ENDFIX
10 ENDFIX
11
12
```

```
1 // Business Rule Logic
2 String CalcScript = ""
3
4 Set UpdateCalc Off;
5
6 FIX ("BaseData")
7     FIX({Scenario},{Version},{Entity},{Currency},@Relative("YearTotal",0))
8         FIX(@Relative("All Market",0))
9         FIX(@Relative("All Product",0))
10             "Product Revenue" = "Volume" * "Average Selling Price";
11         ENDFIX
12     ENDFIX
13 ENDFIX
14
15
16 ""
17 // Generates and Launches Business Rule
18 println (CalcScript)
19 return CalcScript
20
```

Encapsulates Business Rule

Launches Business Rule

Calc Only Updated Data

Plan Volume - Enhanced															Actions		Save	Refresh	Close				
Currency USD		Entity Sales US	Scenario Plan	Version Working 1	Years FY18													Data		Ad hoc		Format	
Working 1																							
		Jan			Feb			Mar			Q1			Apr			May						
		Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price	Product Revenue	Volume	Average Selling Price		
US Market	GC One	5,000	325	975,000	1,000	300	300,000	1,000	300	300,000	7,000	308	1,575,000	1,000	300	300,000	1,000	300	300,000	1,000	300		
	GC One S	1,100	400	440,000	1,100	400	440,000	1,100	400	440,000	3,300	400	1,320,000	1,100	400	440,000	1,100	400	440,000	1,100	400		
	GC One X	2,000	500	600,000	1,200	500	600,000	1,200	500	600,000	4,400	500	1,800,000	1,200	500	600,000	1,200	500	600,000	1,200	500		
	GC One XS	3,000	600	1,800,000	1,300	600	780,000	1,300	600	780,000	5,600	600	3,360,000	1,300	600	780,000	1,300	600	780,000	1,300	600		
	GC One XL	1,400	700	980,000	1,400	700	980,000	1,400	700	980,000	4,200	700	2,940,000	1,400	700	980,000	1,400	700	980,000	1,400	700		

```

1 //Get current Data Form
2 DataGrid CurrGrid = operation.Grid
3
4 // Holds the value for the cells that have changed
5 Set<String> MarketList = []
6 Set<String> ProductList = []
7
8 // Iterates through cells and identifies only edited cells
9 CurrGrid.dataCellIterator({DataCell cell -> cell.edited}).each { DataCell cell ->
10     MarketList << cell.getMemberName("Market")
11     ProductList << cell.getMemberName("Product")
12 }
13 // Create Comma Delimited String for Business rule logic
14 String sMarkets = "" "${MarketList.join(', ')}" ""
15 String sProducts = "" "${ProductList.join(', ')}" ""

```



```

Completed
Log messages :

Set UpdateCalc Off;

FIX ("BaseData")
    FIX("OEP_Plan","Working 1","Sales US","USD",@Relative("YearTotal",0))
    FIX("US Market")
    FIX("GC One","GC One X")
    "Product Revenue" = "Volume" * "Average Selling Price";
    ENDFIX
    ENDFIX
    ENDFIX
    ENDFIX

```

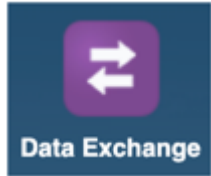
Calc Only Updated Data

```
1 // Business Rule Logic
2 String CalcScript = ""
3
4 Set UpdateCalc Off;
5
6 FIX ("BaseData")
7   FIX({Scenario},{Version},{Entity},{Currency},@Relative("YearTotal",0))
8     FIX(@Relative("All Market",0))
9     FIX(@Relative("All Product",0))
10      "Product Revenue" = "Volume" * "Average Selling Price";
11    ENDFIX
12  ENDFIX
13 ENDFIX
14
15 ""
16
17 // Generates and Launches Business Rule
18 println (CalcScript)
19 return CalcScript
20
```



```
1 //Get current Data Form
2 DataGrid CurrGrid = operation.Grid
3
4 // Holds the value for the cells that have changed
5 Set<String> MarketList = []
6 Set<String> ProductList = []
7 Set<String> VersionList = []
8
9 // Iterates through cells and identifies only edited cells
10 CurrGrid.dataCellIterator({DataCell cell -> cell.edited}).each { DataCell cell ->
11   MarketList << cell.getMemberName("Market")
12   ProductList << cell.getMemberName("Product")
13   VersionList << cell.getMemberName("Version")
14 }
15 // Create Comma Delimited String for Business rule logic
16 String sMarkets = "" "${MarketList.join(',')} " ""
17 String sProducts = "" "${ProductList.join(',')} " ""
18 String sVersions = "" "${VersionList.join(',')} " ""
19
20 // Business Rule Logic
21 String CalcScript = ""
22
23 Set UpdateCalc Off;
24
25 FIX ("BaseData")
26   FIX({Scenario},$sVersions,{Entity},{Currency},@Relative("YearTotal",0))
27     FIX($sMarkets)
28     FIX($sProducts)
29     "Product Revenue" = "Volume" * "Average Selling Price";
30   ENDFIX
31 ENDFIX
32
33 ENDFIX
34
35 ""
36
37 // Generates and Launches Business Rule
38 println (CalcScript)
39 return CalcScript
40
```

Consolidate Data Faster!



Create and Manage

- Action Menus
- Alias Tables
- Dimensions
- Forms
- Rules

Create and Manage

- Action Menus
- Alias Tables
- Dimensions
- Forms
- Rules

Data Integration | Data Maps

Create Refresh

Search | All Sources All Targets

Name	Source	Target	Modified	Actions
Revenue to OEP_FS	Plan2	EPBCS : OEP_FS	4/1/20	...

Simple Form: Plan Volume - Enhanced

Properties Layout Other Options Business Rules Smart Push

☒ Run Smart Push In Background ☐ Use Database Suppression

Action View + Detach

Map Reporting Application	Dimension	Use Form Context	Overwrite Selection	Run After Save	Delete
Plan2 to Revenue				<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

```
//Get current Data Form
DataGrid CurrGrid = operation.Grid

// Holds the value for the cells that have changed
Set<String> MarketList = []
Set<String> ProductList = []

// Iterates through cells and identifies only edited cells
CurrGrid.dataCellIterator({DataCell cell -> cell.edited}).each { DataCell cell ->
    MarketList << cell.getMemberName("Market")
    ProductList << cell.getMemberName("Product")
}

// Create Comma Delimited String for Business rule logic
String sMarkets = "" "${MarketList.join(',')} "
String sProducts = "" "${ProductList.join(',')} "

// Launch SmartPush

boolean ClearData = true
String FinRptg_DataMap = "Revenue to FinRptg"

operation.application.getDataMap(FinRptg_DataMap).createSmartPush().execute(["Market" : sMarkets, "Product" : sProducts ], ClearData)
```