

Decision Management Community Challenge – July 2024

Seth Meldon

Progress Corticon

For this challenge (rule files available to download [here](#)), we segment out the rule logic into discrete rulesheets (decision tables), *Baseline* and *Buy*, sequenced to execute in a defined order as specified in a Ruleflow, shown in figure 1.



Figure 1

The logic is defined as business rules made up of conditions and actions, expressed as the values that data fields may have (i.e. the field *Stock.name* could have a value of *ABC*).

The data fields are represented in a Rule Vocabulary, shown in figure 2, which contains two related entities—Stock and Portfolio. A Portfolio has a 1->many relationship with Stock. This is important to capture in the model, as don't care about a Stock in and of themselves, but a Stock within the context of a portfolio's other stocks.

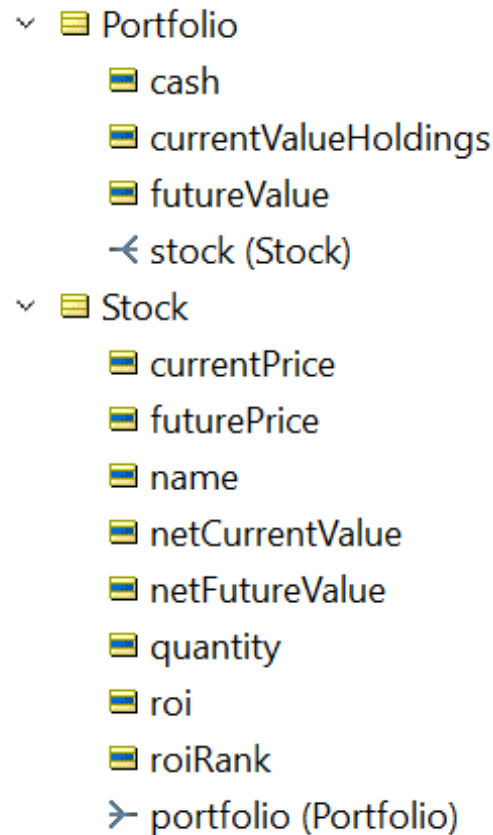


Figure 2

The first rulesheet sets preliminary values and calculates + ranks the stocks' ROI in comparison to the others. The conditions and actions are defined in a rulesheet by dragging and dropping from the Rule Vocabulary onto the Rulesheet depending on the role (a field's value may represent a part of the 'if' condition for the rule, the 'then' action taken when that condition is met, or the value of a field may be used to delineate, sort, or filter entities of the same type with different values for a given attribute field). Any rules defined in the zero column of a rulesheet need no condition to fire, they will always execute—this is useful for initial calculations like calculating an individual stock's ROI.

Anywhere this logic is defined, we can likewise write a plain language description of what the rule syntax is doing. Figure 3 below shows the first rulesheet with the natural language view toggled on, and figure 4 shows the same rulesheet with it toggled off (showing the actual rule syntax used):

Scope		Conditions	0
<div> <div>Portfolio</div> <div> <div>cash</div> <div>currentValueHoldings</div> <div>futureValue</div> <div> <div>stock (Stock) [holdings]</div> <div> <div>currentPrice</div> <div>futurePrice</div> <div>netCurrentValue</div> <div>netFutureValue</div> <div>quantity</div> <div>roi</div> <div>roiRank</div> </div> </div> <div>Stock</div> </div> </div>	a		
	b		
	c		
	d		
		Actions	
		Post Message(s)	
	A	Set ROI % for each stock to be future price / current price *100	<input checked="" type="checkbox"/>
	B	Set initial quantity for each stock to the current net value in the portfolio of the stock / current price of that stock	<input checked="" type="checkbox"/>
	C	Set net current value for all stocks to 1000 (the minimum to be invested in all stocks)	<input checked="" type="checkbox"/>
	D	Set portfolio current value to sum of each stocks' net current value in the portfolio	<input checked="" type="checkbox"/>
	E	Set portfolio future value to sum of each stocks' net future value in the portfolio	<input checked="" type="checkbox"/>
	F	Set net future value for each stock to the quantity of the stock held in the portfolio * its future value	<input checked="" type="checkbox"/>
	G	Subtract the value of the portfolio's current holdings' net value from the initial amount of cash on hand	<input checked="" type="checkbox"/>
	H	Set the stock with the best ROI to have an roiRank of 1	<input checked="" type="checkbox"/>
	I	Set the stock with the second best ROI to have an roiRank of 2	<input checked="" type="checkbox"/>
	J	Set the stock with the third best ROI to have an roiRank of 3	<input checked="" type="checkbox"/>
	K	Set the stock with the fourth best ROI to have an roiRank of 4	<input checked="" type="checkbox"/>
Filters			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Figure 3

Scope		Conditions	0
<div> <div>Portfolio</div> <div> <div>cash</div> <div>currentValueHoldings</div> <div>futureValue</div> <div>stock (Stock) [holdings] <div> <div>currentPrice</div> <div>futurePrice</div> <div>netCurrentValue</div> <div>netFutureValue</div> <div>quantity</div> <div>roi</div> <div>roiRank</div> </div> </div> <div>Stock</div> </div> </div>	a		
	b		
	c		
	d		
	Actions		
	Post Message(s)		
	A	holdings.roi=holdings.futurePrice/holdings.currentPrice*100	<input checked="" type="checkbox"/>
	B	holdings.quantity=(holdings.netCurrentValue/holdings.currentPrice)	<input checked="" type="checkbox"/>
	C	holdings.netCurrentValue=1000	<input checked="" type="checkbox"/>
	D	Portfolio.currentValueHoldings = holdings.netCurrentValue->sum	<input checked="" type="checkbox"/>
	E	Portfolio.futureValue = holdings.netFutureValue->sum	<input checked="" type="checkbox"/>
	F	holdings.netFutureValue = holdings.quantity * holdings.futurePrice	<input checked="" type="checkbox"/>
	G	Portfolio.cash-=Portfolio.currentValueHoldings	<input checked="" type="checkbox"/>
	H	holdings->sortedByDesc(roi)->at(1).roiRank = 1	<input checked="" type="checkbox"/>
	I	holdings->sortedByDesc(roi)->at(2).roiRank = 2	<input checked="" type="checkbox"/>
	J	holdings->sortedByDesc(roi)->at(3).roiRank = 3	<input checked="" type="checkbox"/>
	K	holdings->sortedByDesc(roi)->at(4).roiRank = 4	<input checked="" type="checkbox"/>

Filters	
1	
2	
3	
4	
5	
6	
7	
8	
9	

Figure 4

The scope section in the top left quadrant of figures 3 and 4 demonstrate how we are making use of the association between Portfolio and Stock in order to define rules which involve all of the Stocks within a given Portfolio. The 'parent' entity here is Portfolio, and the 'child' entity is Stock. Notice in the scope section that we have given an 'alias', named "holdings", to the collection of all instances of Stock that are a child to a given Portfolio. We can then use this alias to do things like in row D, where we assign the Portfolio's current holdings value to be the sum of its child entities, Stocks, net current values.

In rows H through K, we are ranking the ROI of each stock 1 through 4, and assigning the value to the field roiRank on the applicable entity within the collection.

Now, we can define a test case using the stocks values specified in the challenge prompt, to see how the first rulesheet will evaluate the data. Each pane of the ruletest for this rulesheet is labeled in figure 5 and described below the image.

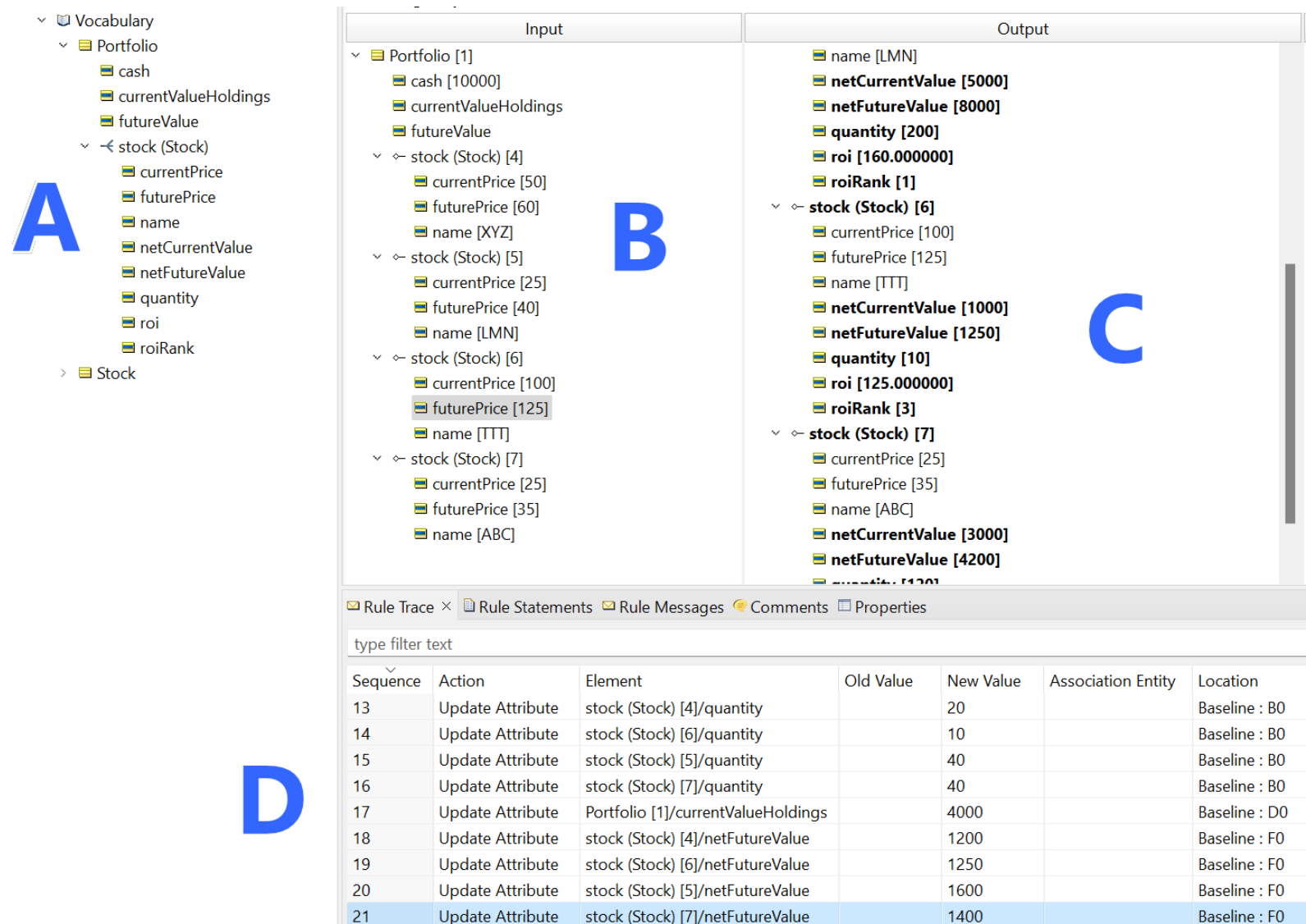


Figure 5

A. Rule vocabulary tree. The entity Portfolio has been dragged into the input pane, along with 4 instances of the child entity Stock

- B. The initial values are set upon the attributes of the Portfolio/Stock entities. Note that this test has been set to test against '/Challenge July 2024/Baseline.ers'. This is the path to the individual rulesheet we're testing—we can run tests against rulesheets or entire ruleflows.
- C. Output shows in bold all values that are new/changed as a result of the rules in the tested rulesheet
- D. When we run the test, the Rule Trace view shows us the sequence of events from start to finish of Corticon evaluating the rules in the rulesheet. Column 1 is integer number of when the action was taken, column 2 describes the type of action takes, column 3 shows which attribute on which instance of an entity the action was taken, column 4 shows the value of this field prior to the action, column 5 shows the value of this field after the action was taken, column 6 specifies any parent/child associations created by the rules, and column 7 documents the specific rulesheet name and rule number which had executed at this stage in the ruletest.

The second rulesheet is shown in natural language view in figure 6, and in the rule language in figure 7:

Scope		Conditions	0	1	2	3	4
<div> <div>Portfolio</div> <div> <div>Filters</div> <div> <div>cash</div> <div>futureValue</div> <div>stock (Stock) [allStocks]</div> <div>stock (Stock) [ranked1] <div>Filters <div>currentPrice</div> <div>futurePrice</div> <div>netCurrentValue</div> <div>quantity</div> <div>roi</div> <div>roiRank</div> </div> </div> <div>stock (Stock) [ranked2]</div> <div>stock (Stock) [ranked3]</div> <div>stock (Stock) [ranked4]</div> </div> </div> </div>		a netCurrentValue of 'ranked1' alias < 5000 (T/F)		T	-	-	-
		b Cash on hand is >= current price of 'ranked1'		T	-	-	-
		c					
		d netCurrentValue of 'ranked2' alias < 5000 (T/F)		-	T	-	-
		e Cash on hand is >= current price of 'ranked2'		-	T	-	-
		f					
		g netCurrentValue of 'ranked3' alias < 5000 (T/F)		-	-	T	-
		h Cash on hand is >= current price of 'ranked3'		-	-	T	-
		i					
		j netCurrentValue of 'ranked4' alias < 5000 (T/F)		-	-	-	T
		k Cash on hand is >= current price of 'ranked4'		-	-	-	T
		l					
		m					
		n					
		Actions					
		Post Message(s)					
		A Add 1 to ranked1.quantity	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		B Add ranked1.currentPrice to ranked1.netCurrent Value	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		C Subtract ranked1.currentPrice from Portfolio.cash	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		D					
		E Add 1 to ranked2.quantity	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		F Add ranked2.currentPrice to ranked2.netCurrent Value	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		G Subtract ranked1.currentPrice from Portfolio.cash	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		H					
		I Add 1 to ranked3.quantity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		J Add ranked3.currentPrice to ranked3.netCurrent Value	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		K Subtract ranked1.currentPrice from Portfolio.cash	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
		L					
		M Add 1 to ranked4.quantity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
		N Add ranked4.currentPrice to ranked4.netCurrent Value	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
		O Subtract ranked1.currentPrice from Portfolio.cash	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
		P					
		Q Set portfolio future value to be the sum of all holdings' net future v...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		R Set the sum of all holdings' net future values to be their future price * the quantity held	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
		Overrides		{2, 3, 4}	{3, 4}	4	
Filters							
1	Alias 'ranked1' for the instance of Portfolio.stock where Portfolio.stock.roiRank = 1						
2	Alias 'ranked2' for the instance of Portfolio.stock where Portfolio.stock.roiRank = 2						
3	Alias 'ranked3' for the instance of Portfolio.stock where Portfolio.stock.roiRank = 3						
4	Alias 'ranked4' for the instance of Portfolio.stock where Portfolio.stock.roiRank = 4						

Figure 6

Scope

Portfolio

Filters

cash

futureValue

stock (Stock) [allStocks]

stock (Stock) [ranked1]

Filters

currentPrice

futurePrice

netCurrentValue

quantity

roi

roiRank

stock (Stock) [ranked2]

stock (Stock) [ranked3]

stock (Stock) [ranked4]

Filters

1

ranked1.roiRank=1

2

ranked2.roiRank=2

3

ranked3.roiRank=3

4

ranked4.roiRank=4

5

Conditions

a

ranked1.netCurrentValue<5000

b

Portfolio.cash>=ranked1.currentPrice

c

d

ranked2.netCurrentValue<5000

e

Portfolio.cash>=ranked2.currentPrice

f

g

ranked3.netCurrentValue<5000

h

Portfolio.cash>=ranked3.currentPrice

i

j

ranked4.netCurrentValue<5000

k

Portfolio.cash>=ranked4.currentPrice

l

m

n

Actions

Post Message(s)

A

ranked1.quantity+=1

B

ranked1.netCurrentValue+=ranked1.currentPrice

C

Portfolio.cash-=ranked1.currentPrice

D

E

ranked2.quantity+=1

F

ranked2.netCurrentValue+=ranked2.currentPrice

G

Portfolio.cash-=ranked2.currentPrice

H

I

ranked3.quantity+=1

J

ranked3.netCurrentValue+=ranked3.currentPrice

K

Portfolio.cash-=ranked3.currentPrice

L

M

ranked4.quantity+=1

N

ranked4.netCurrentValue+=ranked4.currentPrice

O

Portfolio.cash-=ranked4.currentPrice

P

Q

Portfolio.futureValue=allStocks.netFutureValue->sum;

R

allStocks.netFutureValue=allStocks.futurePrice*allStocks.quantity;

Overrides


{2, 3, 4}

{3, 4}

4

0	1	2	3	4
	T	-	-	-
	T	-	-	-
	-	T	-	-
	-	T	-	-
	-	-	T	-
	-	-	T	-
	-	-	-	T
	-	-	-	T

Figure 7

Here, we are using a number of different rule modeling techniques described below. Before we get there though, you may have noticed that on the ruleflow in figure 1, this rulesheet had this symbol:  while the preceding rulesheet did not. This symbol appears when we set a rulesheet to execute iteratively—by this we mean that Corticon will continually restart evaluation of these rules after all applicable rules have fired. Until the data being evaluated by the rules ceases to change after being evaluated by the rules in this rulesheet, then the data will be continually routed back through this rulesheet. Once it ceases to change, the data is passed into the next node of a ruleflow, or in this case, sent back as the final output result.

This functionality is useful here because the rules are evaluating the optimal investment choice over and over until all of the money is invested, subject to the minimum and maximum investments in any one given stock and their respective returns. With that in mind, here are rules being evaluated by each iterative pass through this rulesheet.

Scope Pane:

- 4 different aliases to represent the distinct instances of Portfolio.stock based upon the previously assigned field, roiRank. The best ROI has a value for roiRank of '1', and is given the alias 'ranked1'. This convention applies for the second, third, and fourth best stocks as well.
- A fifth alias represents all of the stocks in the collection Portfolio.stock

Filter Pane:

- Each of the four ranked stocks is assigned a unique alias based upon the value of the attribute Stock.roiRank

Rules:

- For all four of the aliases representing the stocks' ROI rank, a rule has been defined with two conditions:
 - 1) is the total amount invested in the stock greater than 5000?
 - 2) is the amount of cash on hand greater than the price of the stock?
- If yes to both conditions, then:
 - 1) increment the value of the attribute 'quantity' up by 1 for the stock
 - 2) increment the netCurrentValue of that stock in the portfolio up by the currentPrice of that stock
 - 3) decrement Portfolio.cash down by the currentPrice of that stock
- Notice the bottom row labeled 'Overrides' underneath the rulesheet's actions. Here we are specifying which rule should take precedence in the case when the conditions have been met for more than 1 of the rules. Rule 1 specifies that if the net current value of the stock with the best ROI is under 5,000 and the currentPrice of that stock is less than or equal to the value of Portfolio.cash, then this is the rule that will fire regardless of whether those conditions have been met with any of the other 3 stocks. This pattern continues with the second best, which will not take precedence over the stock with the best ROI, but *will* override the third and fourth best stock.
- In the action row 'Q', we are assigning the value of Portfolio.futureValue to be the sum of the netFutureValue for each instance of Portfolio.stock
- In row 'R', we are assigning the value of each stock's future value to be product of the quantity of that stock * the future price of that stock

Finally, let's test the rules to see how it all comes together. We will repeat the same ruletest, but now instead of only testing the first rulesheet, we'll change the 'test target' to the whole ruleflow that contains the two rulesheets, with the second configured to execute iteratively. The output of this test is shown in figure 8.

Output	
Portfolio [1]	
cash [0]	
currentValueHoldings [4000]	
futureValue [14650]	
stock (Stock) [4]	
currentPrice [50]	
futurePrice [60]	
name [XYZ]	
netCurrentValue [1000]	
netFutureValue [1200]	
quantity [20]	
roi [120.000000]	
roiRank [4]	
stock (Stock) [5]	
currentPrice [25]	
futurePrice [40]	
name [LMN]	
netCurrentValue [5000]	
netFutureValue [8000]	
quantity [200]	
roi [160.000000]	
roiRank [1]	
stock (Stock) [6]	
currentPrice [100]	
futurePrice [125]	
name [TTT]	
netCurrentValue [1000]	
netFutureValue [1250]	
quantity [10]	
roi [125.000000]	
roiRank [3]	
stock (Stock) [7]	
currentPrice [25]	
futurePrice [35]	
name [ABC]	
netCurrentValue [3000]	
netFutureValue [4200]	
quantity [120]	
roi [140.000000]	
roiRank [2]	

Figure 8

As before, the ruletest logs the sequence of rules that execute across the whole ruleflow. Because we're executing the second rulesheet iteratively until we've invested all money, the rule trace data gives us a full picture of how Corticon is reaching the answer. From the rule trace pane, we can right click and export the information to a CSV table. The first 50 and last 50 records from that table are shown below.

Sequence	Action	Element	Old Value	New Value	Association Entity	Location
1	Update Attribute	stock (Stock) [5]/roi		160		Baseline : A0
2	Update Attribute	stock (Stock) [7]/roi		140		Baseline : A0
3	Update Attribute	stock (Stock) [6]/roi		125		Baseline : A0
4	Update Attribute	stock (Stock) [4]/roi		120		Baseline : A0
5	Update Attribute	stock (Stock) [5]/netCurrentValue		1000		Baseline : C0
6	Update Attribute	stock (Stock) [7]/netCurrentValue		1000		Baseline : C0
7	Update Attribute	stock (Stock) [6]/netCurrentValue		1000		Baseline : C0
8	Update Attribute	stock (Stock) [4]/netCurrentValue		1000		Baseline : C0
9	Update Attribute	stock (Stock) [5]/roiRank		1		Baseline : H0
10	Update Attribute	stock (Stock) [7]/roiRank		2		Baseline : I0
11	Update Attribute	stock (Stock) [6]/roiRank		3		Baseline : J0
12	Update Attribute	stock (Stock) [4]/roiRank		4		Baseline : K0
13	Update Attribute	stock (Stock) [5]/quantity		40		Baseline : B0
14	Update Attribute	stock (Stock) [7]/quantity		40		Baseline : B0
15	Update Attribute	stock (Stock) [6]/quantity		10		Baseline : B0
16	Update Attribute	stock (Stock) [4]/quantity		20		Baseline : B0
17	Update Attribute	Portfolio [1]/currentValueHoldings		4000		Baseline : D0
18	Update Attribute	stock (Stock) [5]/netFutureValue		1600		Baseline : F0
19	Update Attribute	stock (Stock) [7]/netFutureValue		1400		Baseline : F0
20	Update Attribute	stock (Stock) [6]/netFutureValue		1250		Baseline : F0
21	Update Attribute	stock (Stock) [4]/netFutureValue		1200		Baseline : F0
22	Update Attribute	Portfolio [1]/cash	10000	6000		Baseline : G0
23	Update Attribute	Portfolio [1]/futureValue		5450		Baseline : E0
24	Update Attribute	stock (Stock) [5]/quantity	40	41		Buy : 1
25	Update Attribute	stock (Stock) [5]/netCurrentValue	1000	1025		Buy : 1
26	Update Attribute	Portfolio [1]/cash	6000	5975		Buy : 1
27	Update Attribute	stock (Stock) [5]/netFutureValue	1600	1640		Buy : R0
28	Update Attribute	Portfolio [1]/futureValue	5450	5490		Buy : Q0
29	Update Attribute	stock (Stock) [5]/quantity	41	42		Buy : 1
30	Update Attribute	stock (Stock) [5]/netCurrentValue	1025	1050		Buy : 1
31	Update Attribute	Portfolio [1]/cash	5975	5950		Buy : 1

32	Update Attribute	stock (Stock) [5]/netFutureValue	1640	1680		Buy : R0
33	Update Attribute	Portfolio [1]/futureValue	5490	5530		Buy : Q0
34	Update Attribute	stock (Stock) [5]/quantity	42	43		Buy : 1
35	Update Attribute	stock (Stock) [5]/netCurrentValue	1050	1075		Buy : 1
36	Update Attribute	Portfolio [1]/cash	5950	5925		Buy : 1
37	Update Attribute	stock (Stock) [5]/netFutureValue	1680	1720		Buy : R0
38	Update Attribute	Portfolio [1]/futureValue	5530	5570		Buy : Q0
39	Update Attribute	stock (Stock) [5]/quantity	43	44		Buy : 1
40	Update Attribute	stock (Stock) [5]/netCurrentValue	1075	1100		Buy : 1
41	Update Attribute	Portfolio [1]/cash	5925	5900		Buy : 1
42	Update Attribute	stock (Stock) [5]/netFutureValue	1720	1760		Buy : R0
43	Update Attribute	Portfolio [1]/futureValue	5570	5610		Buy : Q0
44	Update Attribute	stock (Stock) [5]/quantity	44	45		Buy : 1
45	Update Attribute	stock (Stock) [5]/netCurrentValue	1100	1125		Buy : 1
46	Update Attribute	Portfolio [1]/cash	5900	5875		Buy : 1
47	Update Attribute	stock (Stock) [5]/netFutureValue	1760	1800		Buy : R0
48	Update Attribute	Portfolio [1]/futureValue	5610	5650		Buy : Q0
49	Update Attribute	stock (Stock) [5]/quantity	45	46		Buy : 1
50	Update Attribute	stock (Stock) [5]/netCurrentValue	1125	1150		Buy : 1
1174	Update Attribute	stock (Stock) [7]/quantity	110	111		Buy : 2
1175	Update Attribute	stock (Stock) [7]/netCurrentValue	2750	2775		Buy : 2
1176	Update Attribute	Portfolio [1]/cash	250	225		Buy : 2
1177	Update Attribute	stock (Stock) [7]/netFutureValue	3850	3885		Buy : R0
1178	Update Attribute	Portfolio [1]/futureValue	14300	14335		Buy : Q0
1179	Update Attribute	stock (Stock) [7]/quantity	111	112		Buy : 2
1180	Update Attribute	stock (Stock) [7]/netCurrentValue	2775	2800		Buy : 2
1181	Update Attribute	Portfolio [1]/cash	225	200		Buy : 2
1182	Update Attribute	stock (Stock) [7]/netFutureValue	3885	3920		Buy : R0
1183	Update Attribute	Portfolio [1]/futureValue	14335	14370		Buy : Q0
1184	Update Attribute	stock (Stock) [7]/quantity	112	113		Buy : 2
1185	Update Attribute	stock (Stock) [7]/netCurrentValue	2800	2825		Buy : 2
1186	Update Attribute	Portfolio [1]/cash	200	175		Buy : 2
1187	Update Attribute	stock (Stock) [7]/netFutureValue	3920	3955		Buy : R0
1188	Update Attribute	Portfolio [1]/futureValue	14370	14405		Buy : Q0
1189	Update Attribute	stock (Stock) [7]/quantity	113	114		Buy : 2

1190	Update Attribute	stock (Stock) [7]/netCurrentValue	2825	2850		Buy : 2
1191	Update Attribute	Portfolio [1]/cash	175	150		Buy : 2
1192	Update Attribute	stock (Stock) [7]/netFutureValue	3955	3990		Buy : R0
1193	Update Attribute	Portfolio [1]/futureValue	14405	14440		Buy : Q0
1194	Update Attribute	stock (Stock) [7]/quantity	114	115		Buy : 2
1195	Update Attribute	stock (Stock) [7]/netCurrentValue	2850	2875		Buy : 2
1196	Update Attribute	Portfolio [1]/cash	150	125		Buy : 2
1197	Update Attribute	stock (Stock) [7]/netFutureValue	3990	4025		Buy : R0
1198	Update Attribute	Portfolio [1]/futureValue	14440	14475		Buy : Q0
1199	Update Attribute	stock (Stock) [7]/quantity	115	116		Buy : 2
1200	Update Attribute	stock (Stock) [7]/netCurrentValue	2875	2900		Buy : 2
1201	Update Attribute	Portfolio [1]/cash	125	100		Buy : 2
1202	Update Attribute	stock (Stock) [7]/netFutureValue	4025	4060		Buy : R0
1203	Update Attribute	Portfolio [1]/futureValue	14475	14510		Buy : Q0
1204	Update Attribute	stock (Stock) [7]/quantity	116	117		Buy : 2
1205	Update Attribute	stock (Stock) [7]/netCurrentValue	2900	2925		Buy : 2
1206	Update Attribute	Portfolio [1]/cash	100	75		Buy : 2
1207	Update Attribute	stock (Stock) [7]/netFutureValue	4060	4095		Buy : R0
1208	Update Attribute	Portfolio [1]/futureValue	14510	14545		Buy : Q0
1209	Update Attribute	stock (Stock) [7]/quantity	117	118		Buy : 2
1210	Update Attribute	stock (Stock) [7]/netCurrentValue	2925	2950		Buy : 2
1211	Update Attribute	Portfolio [1]/cash	75	50		Buy : 2
1212	Update Attribute	stock (Stock) [7]/netFutureValue	4095	4130		Buy : R0
1213	Update Attribute	Portfolio [1]/futureValue	14545	14580		Buy : Q0
1214	Update Attribute	stock (Stock) [7]/quantity	118	119		Buy : 2
1215	Update Attribute	stock (Stock) [7]/netCurrentValue	2950	2975		Buy : 2
1216	Update Attribute	Portfolio [1]/cash	50	25		Buy : 2
1217	Update Attribute	stock (Stock) [7]/netFutureValue	4130	4165		Buy : R0
1218	Update Attribute	Portfolio [1]/futureValue	14580	14615		Buy : Q0
1219	Update Attribute	stock (Stock) [7]/quantity	119	120		Buy : 2
1220	Update Attribute	stock (Stock) [7]/netCurrentValue	2975	3000		Buy : 2
1221	Update Attribute	Portfolio [1]/cash	25	0		Buy : 2
1222	Update Attribute	stock (Stock) [7]/netFutureValue	4165	4200		Buy : R0
1223	Update Attribute	Portfolio [1]/futureValue	14615	14650		Buy : Q0