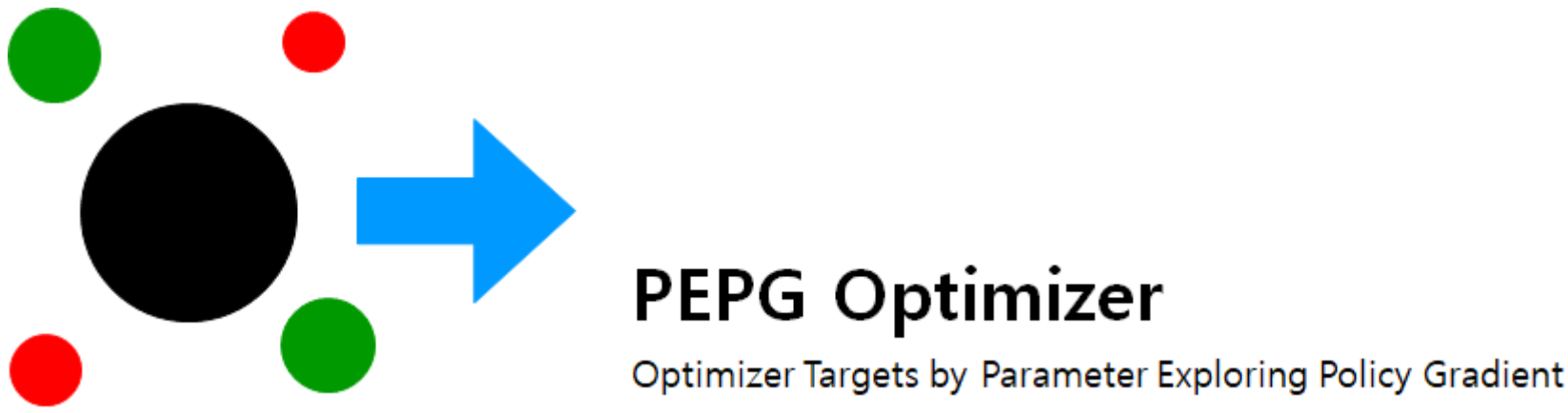
PEPG Optimizer



PEPG(Parameter Exploring Policy Gradient)란? 의사결정문제에서 모델이 없는 강화학습 방식을 적용한 방법.

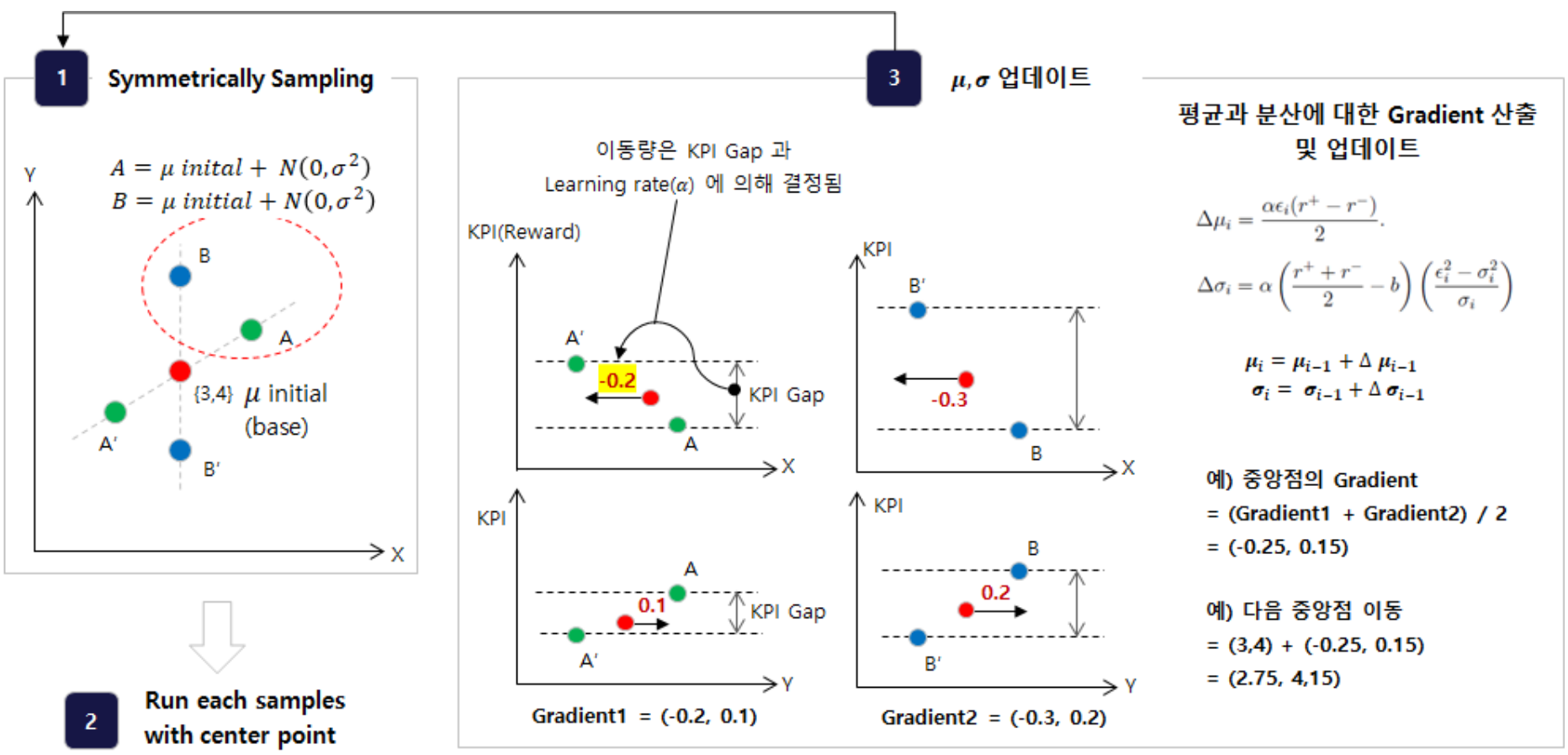
Weight Optimizer와의 차이점?
Mozart Model의 내부에서 학습할 Factor들을 설정하지 않고, WISE 시나리오에서 직접 Model의 Factor들을 설정해서 실험을 구현할 수 있습니다. Weight Optimizer는 Weight 만을 조정하는 특수한 Plug-Ins입니다. PEPG는 좀 더 다양한 종류의 Factor들의 범위를 설정하여 KPI의 결과(증가 혹은 감소)를 확인할 수 있습니다.

Sample Scenario : 아래의 메뉴얼에서 소개하고 있는 PEPG Optimizer의 Sample 시나리오입니다.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/37bd5a82-c4b0-4d0a-aff0-1962ee907600/PEPG_LocalOptimum_(2).zip

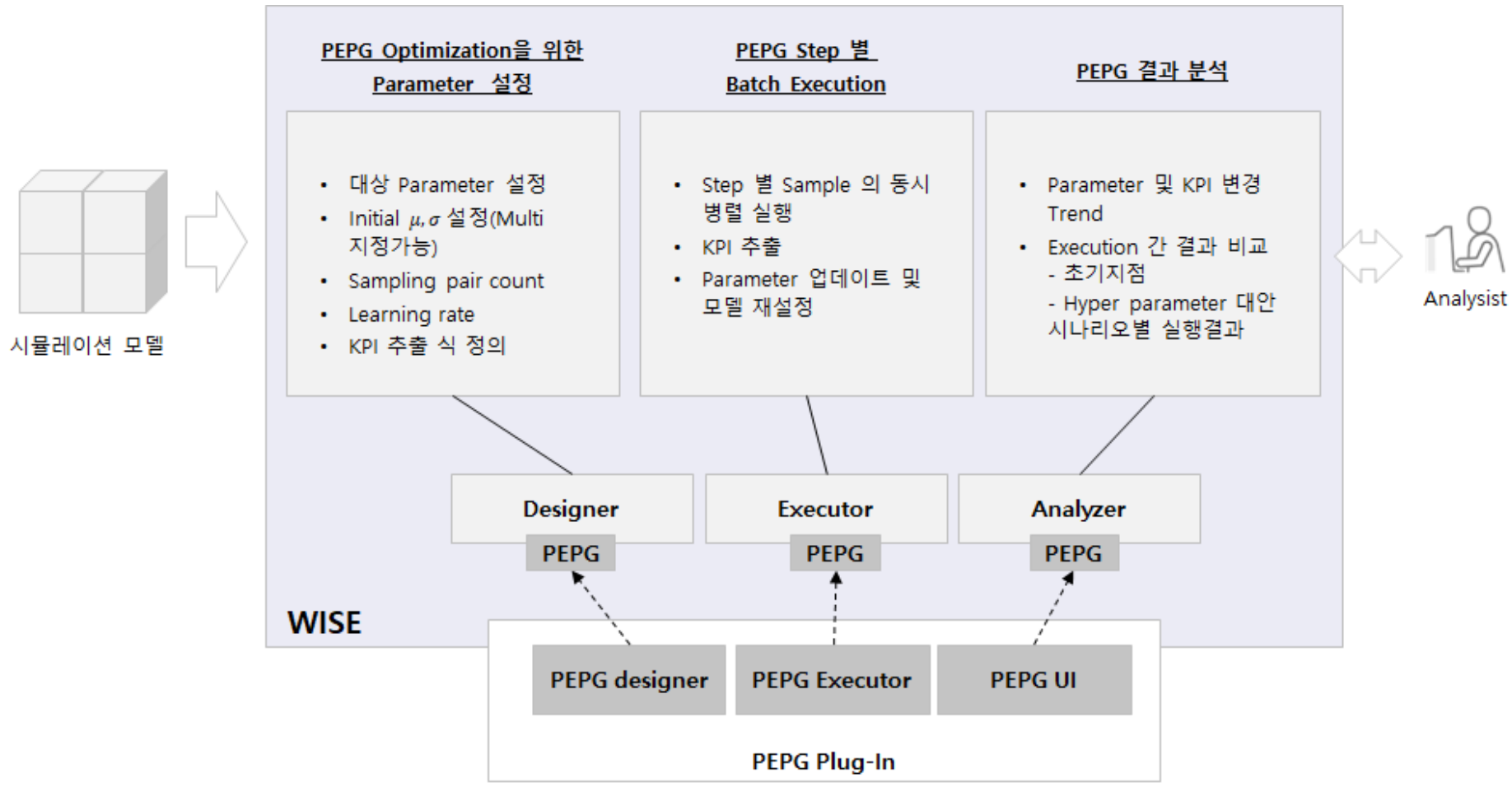
개요

PEPG(Parameter Exploring Policy Gradient)는 의사결정문제에서 모델이 없는 강화학습 방식을 적용하는 방법으로, Parameter 공간에서 직접 샘플링하여 분산을 낮추는 Gradient 추정치를 얻을 수 있는 방법입니다.



*** Hyper Parameter : Number of sampling pair, Learning Rate, KPI, initial μ,σ**

WISE 의 PEPG Plugin 을 통해 시뮬레이션모델의 관심 Parameter, KPI 를 정의하고 Execution 설정한 후 대상 Parameter 를 최적화 할 수 있습니다. 또한 PEPG 실행 결과분석 및 Log 분석이 가능한 UI 를 제공합니다.



플러그인 설치

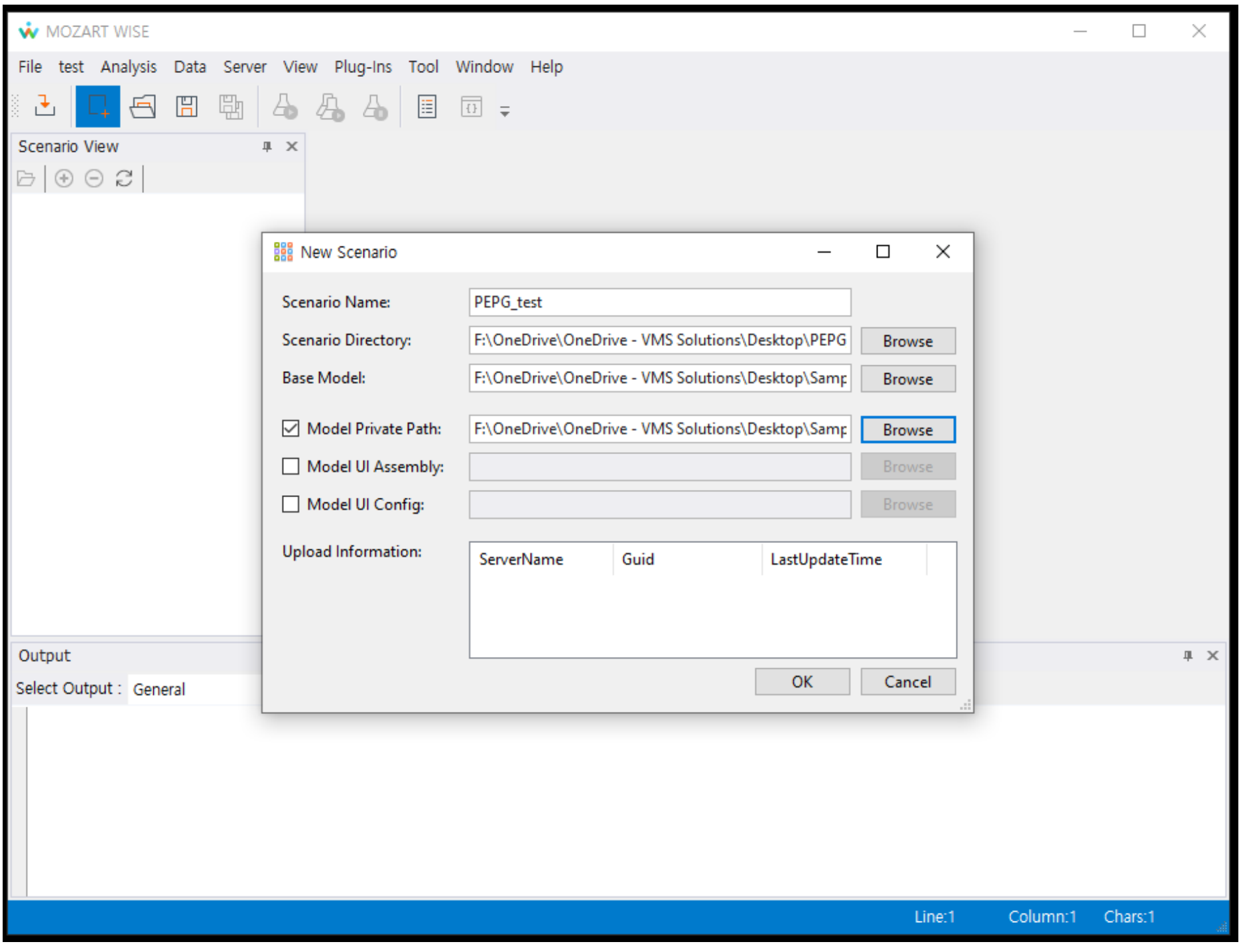
PEPG Optimizer 플러그인의 설치는 WISE 의 다른 플러그인과 마찬가지로 [Plug-Ins]>[Plug-In Manager] 을 사용하여 설치합니다. 여기를 참조합니다.

사용 방법

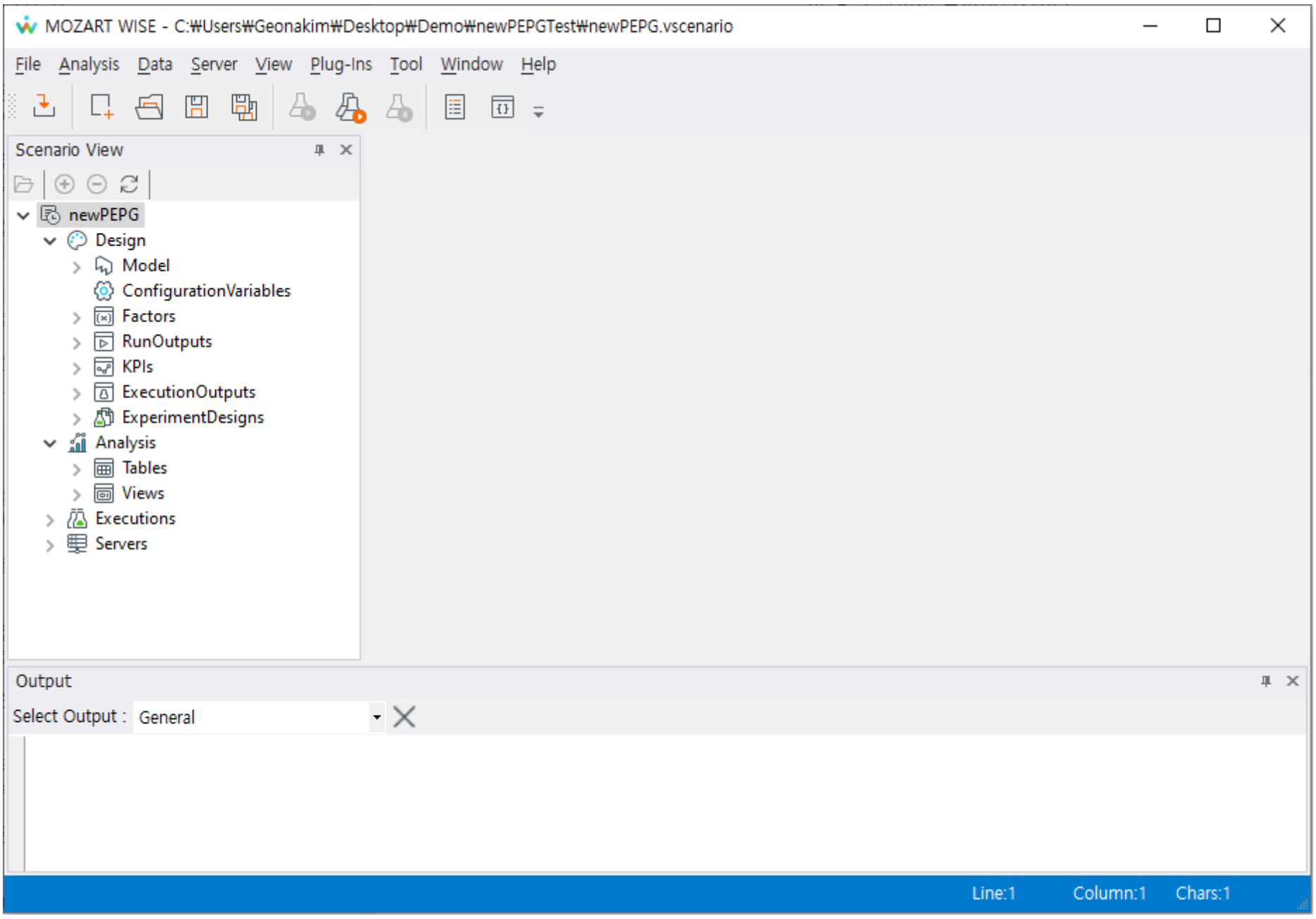
본 장에서 PEPG의 사용방법을 1) 기본 시나리오 생성, 2) PEPG 실험 설계, 3) PEPG Parameter 설정, 4) KPI 설정, 5) PEPG Execution 생성 및 설정, 6) 실행 및 결과 분석의 총 6단계로 설명합니다.

1) 기본 시나리오 생성

PEPG의 기본 시나리오는 일반적인 WISE 시나리오와 동일한 방법으로 생성됩니다. W.O Plugin과 다르게 특별한 학습 모델이 필요하지 않습니다. WISE를 실행한 뒤 New Scenario을 클릭 한 후 새로운 시나리오를 생성합니다.



Create New Scenario



Scenario View

2) PEPG 실험 설계

PEPG는 Mozart Model의 Input Table의 특정값들을 Parameter로 조정하여 사용자가 설정한 Target KPI가 최대가 되는 값을 찾아가는 Plugin입니다. PEPG를 실행하기 전에 어떤 Target Parameter들을 조정해서 어떤 Target KPI를 최대화 할 것인지, 실험 설계를 실시합니다.

- Target Parameter : 실험에서 조정할 Factor들
- Target KPI : Factor들을 조정해서 얻을 목표값

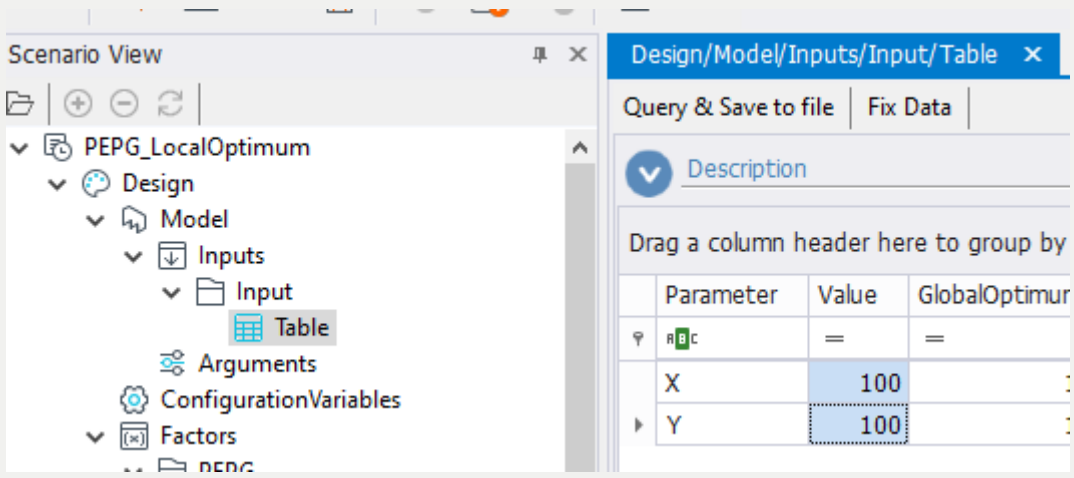
아래의 실험 예시는 X와 Y를 Parameter로 하여 해당값을 조정해 가면서 해당 Parameter가 Global Optimum(최적해)로 가는지 확인하는 실험 설계입니다.

실험 설계(예시)

종류	Aa 이름	Expression	비고
Target_Parameter	X	First(Inputs.Table, a.Parameter = "X", "Value")	
Target_Parameter	Y	First(Inputs.Table, a.Parameter = "Y", "Value")	
Target_KPI	KPI	First(Outputs.KPI, 1=1,"KpiValue")	실험의 성능 지표 KPI
KPI	LocalOptimum1	First(Outputs.KPIInfo, a.TYPE = "LocalOptimum1","KPI")	지역 최적해1
KPI	LocalOptimum2	First(Outputs.KPIInfo, a.TYPE = "LocalOptimum2","KPI")	지역 최적해2
KPI	LocalOptimum3	First(Outputs.KPIInfo, a.TYPE = "LocalOptimum3","KPI")	지역 최적해3

종류	이름	Expression	비고
KPI	GlobalOptimum	First(Outputs.KPIInfo, a.TYPE = "GlobalOptimum","KPI")	실제 최적값

Target Parameter의 Expression은 아래의 Cell의 위치를 가르키고 있습니다.
- X : Parameter Column이 "X"인 Row의 Value Column의 Cell 값
- Y : Parameter Column이 "Y"인 Row의 Value Column의 Cell 값



▼ 다른 실험 예시

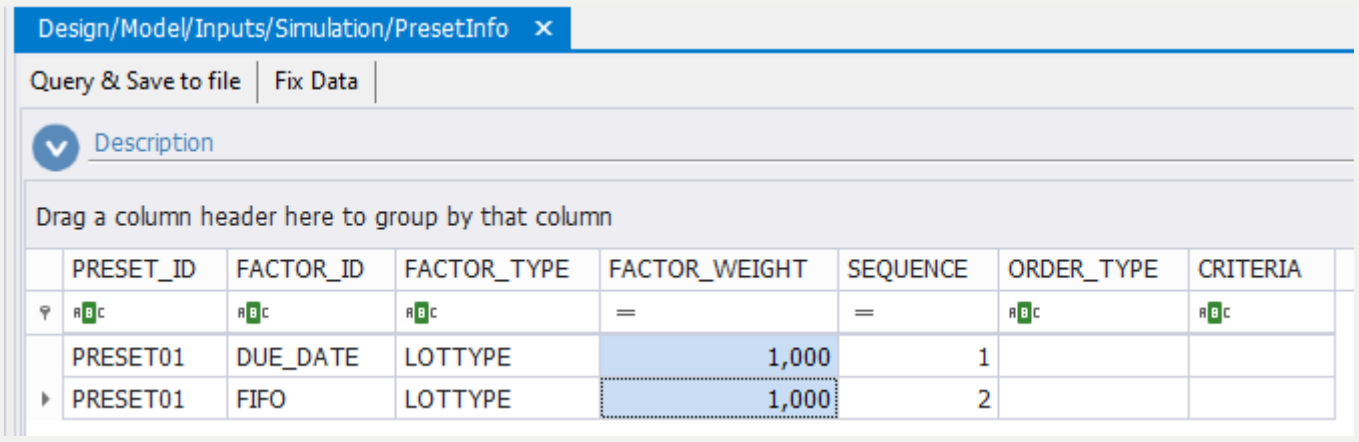
Preset Factor 중 DUE_DATE와 FIFO를 조정해서 DELAY(생산일정 지연)를 줄이고 ON_TIME(생산일정 충족)과 PROD_QTY(제품 생산량)를 늘리려는 실험의 예시입니다.

조정할 Target Parameter인 DUE_DATE와 FIFO의 Weight Factor를 Target Parameter들로 설정하였습니다. 그리고, KPI가 되는 DELAY, ON_TIME, PROD_QTY의 변화에 따라 점수가 변화하는 SCORE라는 Target KPI를 설정하였습니다. 이제 실험 설계에 따라 다음장에서 Parameter들을 설정합니다.

실험 설계(예시)

종류	이름	Expression	비고
Target_Parameter	PRESET01_DUE_DATE_LOTTYPE	First(Inputs.PresetInfo, a.PRESET_ID = "PRESET01" && a.FACTOR_ID = "DUE_DATE" && a.FACTOR_TYPE = "LOTTYPE", "FACTOR_WEIGHT")	Preset Factor 중에 Due Date와 관련된 Weight Factor
Target_Parameter	PRESET01_FIFO_LOTTYPE	First(Inputs.PresetInfo, a.PRESET_ID = "PRESET01" && a.FACTOR_ID = "FIFO" && a.FACTOR_TYPE = "LOTTYPE", "FACTOR_WEIGHT")	Preset Factor 중에 FIFO와 관련된 Weight Factor
KPI	PROD_QTY	Count(Outputs.EqpPlan,a.STEP_ID = "STEP07","PRODUCT_ID")	제품 생산량
KPI	ON_TIME	First(CustomRunOutputs.OnTimeTable,1=1,"ON_TIME")	생산일정 충족회수
KPI	DELAY	First(CustomRunOutputs.OnTimeTable,1=1,"DELAY")	생산일정 지연회수
Target_KPI	SCORE	Kpis.DELAY*-5 + Kpis.ON_TIME*10 + Kpis.PROD_QTY*1	최종 KPI 점수

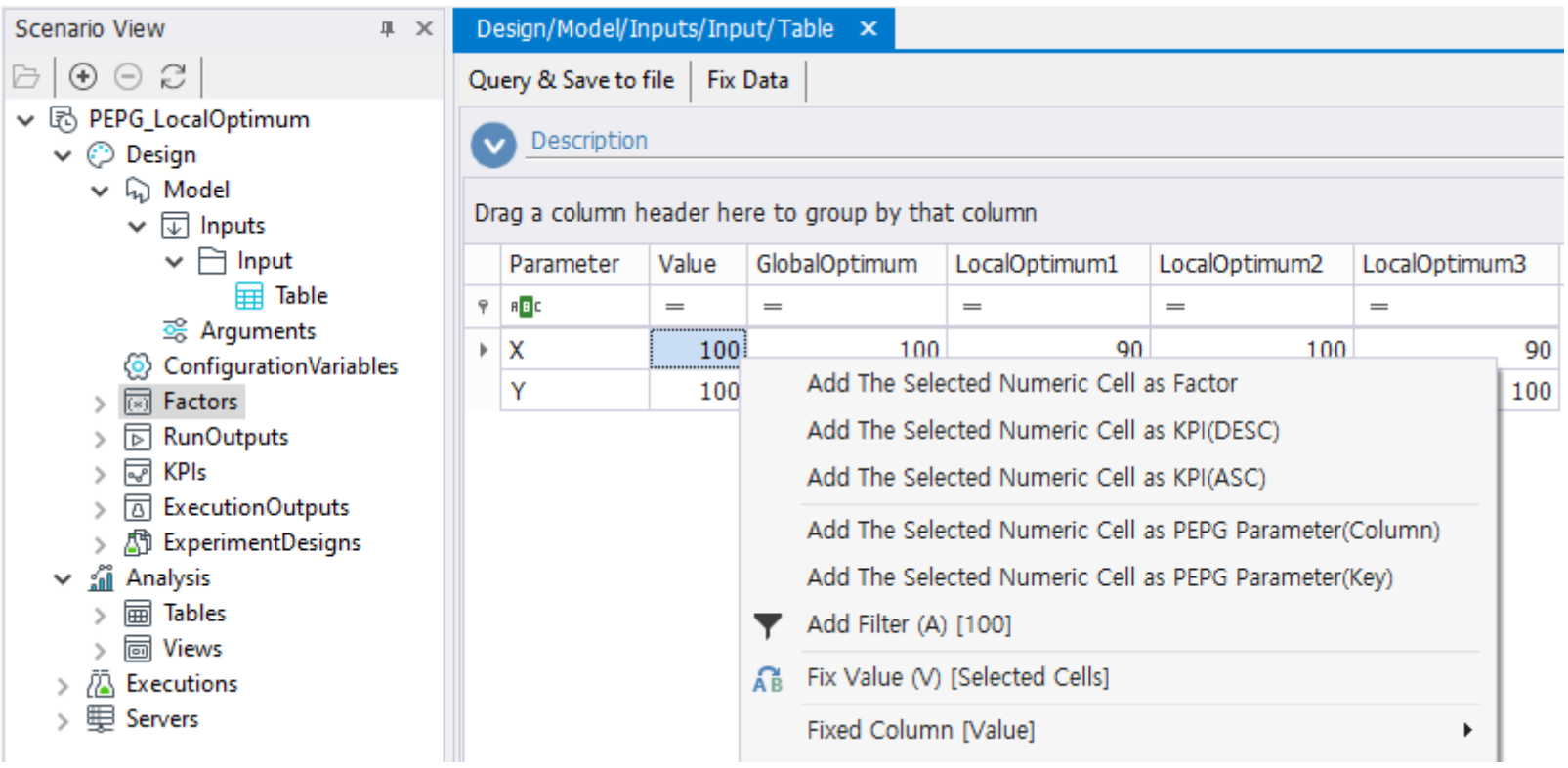
Target Parameter의 Expression은 아래의 Cell의 위치를 가르키고 있습니다.
- PRESET01_DUE_DATE_LOTTYPE : PRESET01, DUE_DATE, LOTTYUPE
- PRESET01_FIFO_LOTTYPE : PRESET01, FIFO, LOTTYUPE



PRESET_ID	FACTOR_ID	FACTOR_TYPE	FACTOR_WEIGHT	SEQUENCE	ORDER_TYPE	CRITERIA
PRESET01	DUE_DATE	LOTTYPE	1,000	1		
PRESET01	FIFO	LOTTYPE	1,000	2		

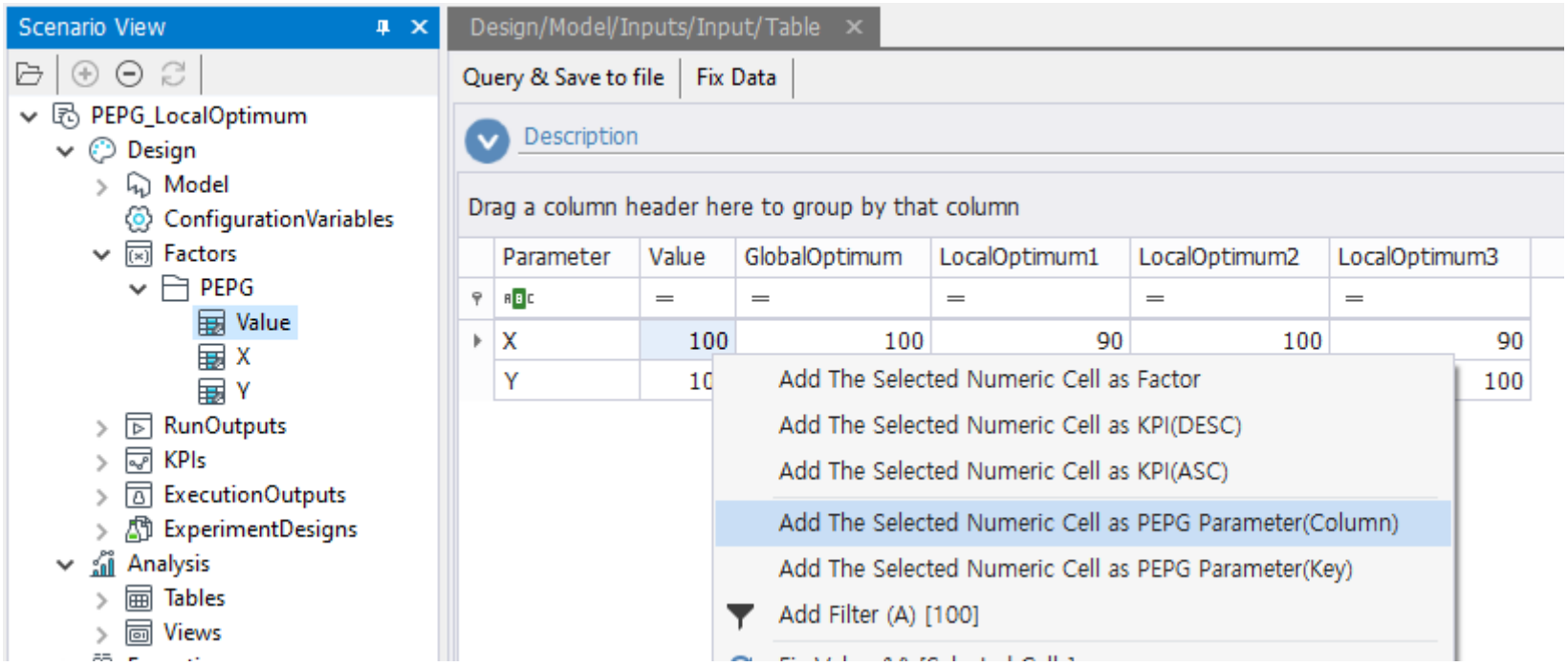
3) PEPG Parameter 설정

PEPG에서 조정할 Parameter들은 해당 WISE 시나리오 Base Model의 Inputs 정보에서 사용자가 쉽게 선택하고 추가할 수 있습니다. 설정할 값이 있는 Input Table을 찾아서 조정할 Cell위의 오른쪽 클릭을 하면 아래와 같이 Parameter를 설정할 수 있는 두 개의 Add 방법이 존재합니다.



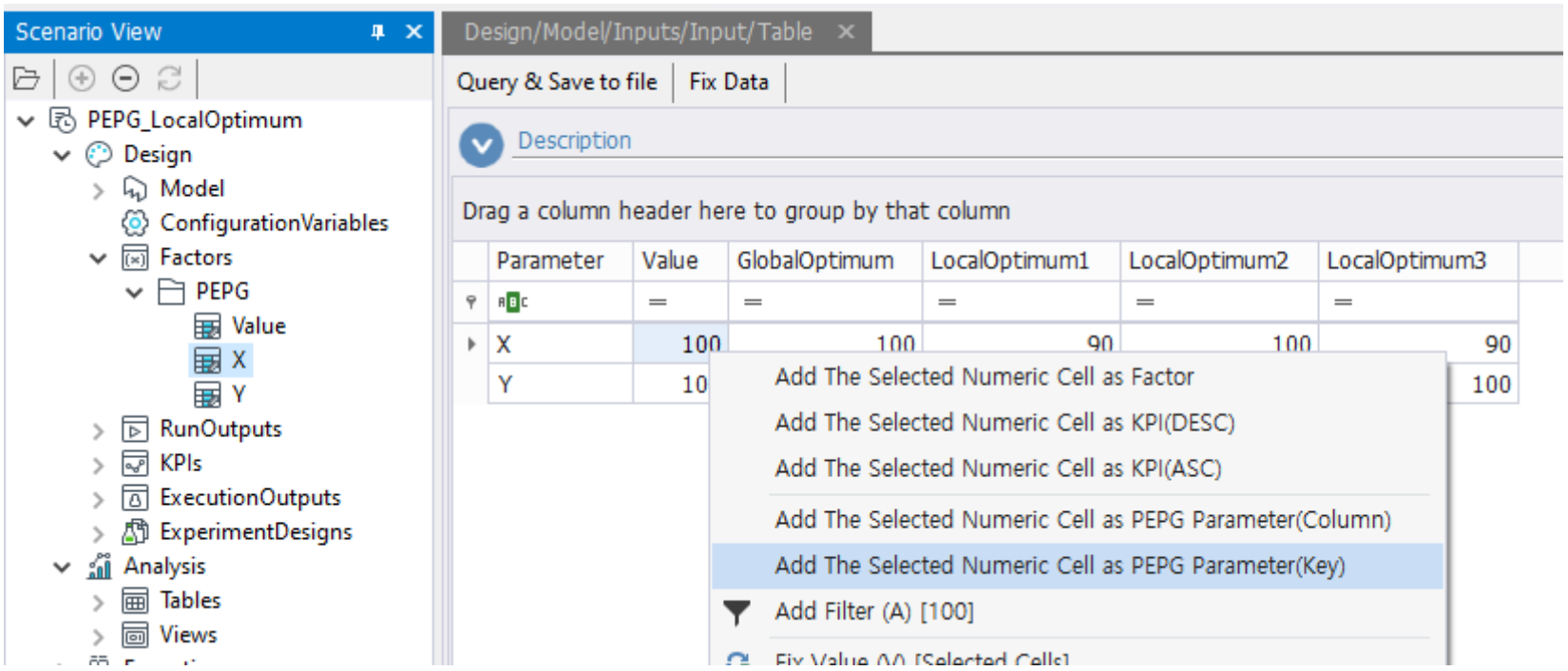
- Add The Selected Numeric Cell as PEPG Parameter(Column)

  ▼ Column 이름으로 Parameter(Factor)를 생성합니다.



- Add The Selected Numeric Cell as PEPG Parameter(Key)

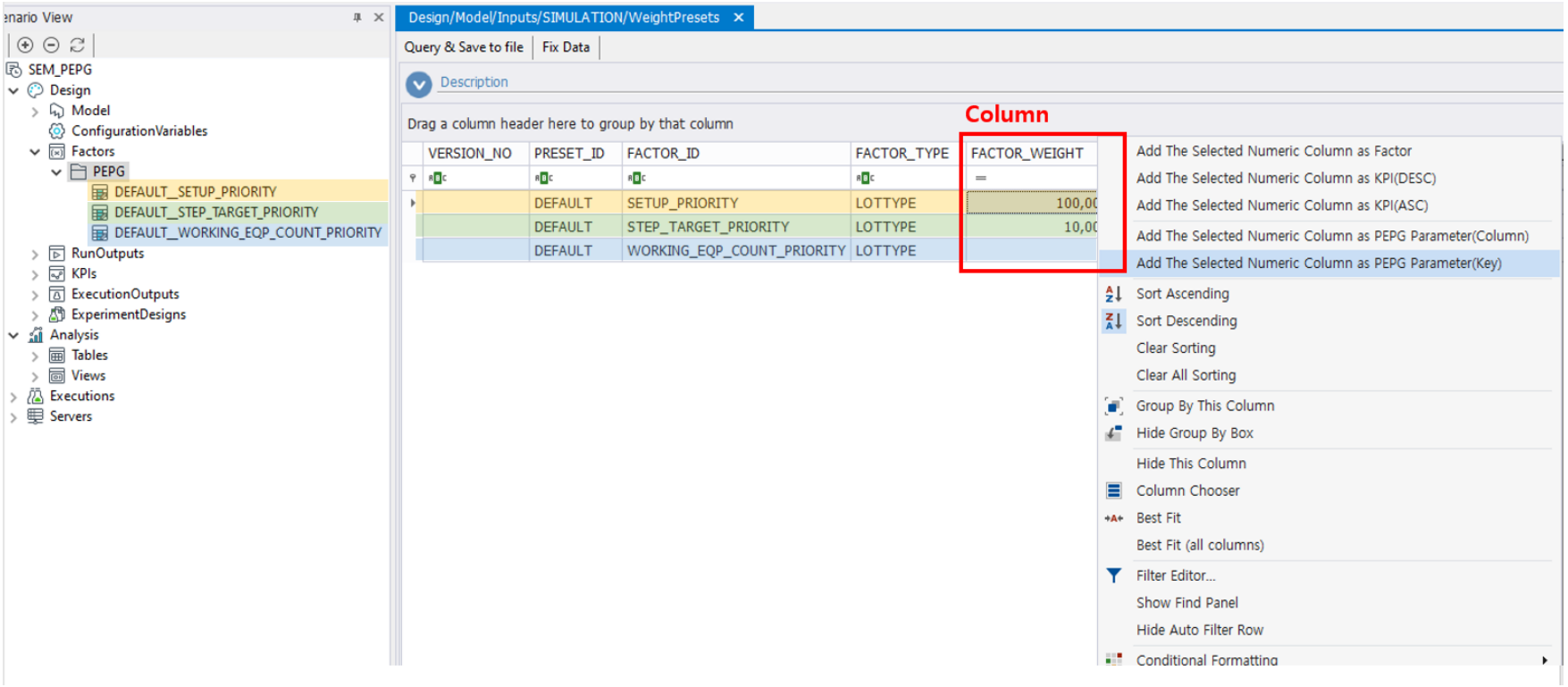
  ▼ Key값들로 Parameter(Factor)를 생성합니다.



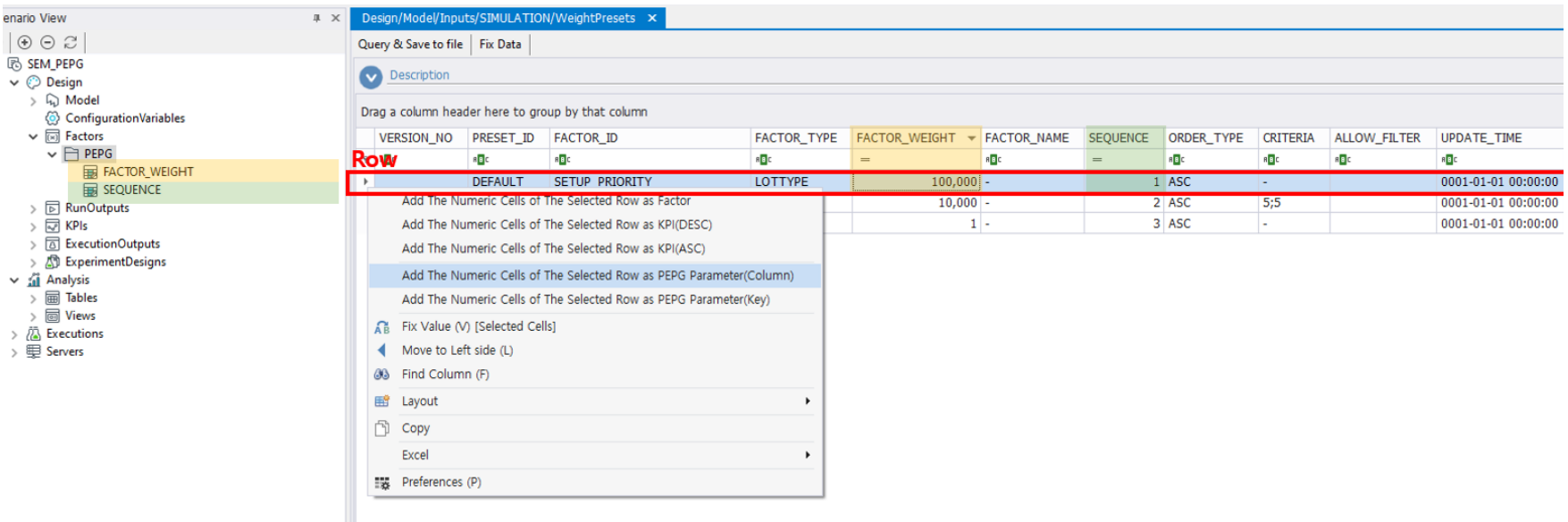
> 💡 Parameter를 생성할 때 Key값으로 생성하기를 추천합니다. Column이름으로 설정할 경우 생성된 Parameter의 이름 만보고 실제 Parameter값을 유추하기 어려운 상황이 발생할 수 있습니다.

- 해당 Row나 Column을 한번에 Parameter로 등록하는 기능

  ▼ Column의 모든 Cell들을 Target Factor로 지정하기

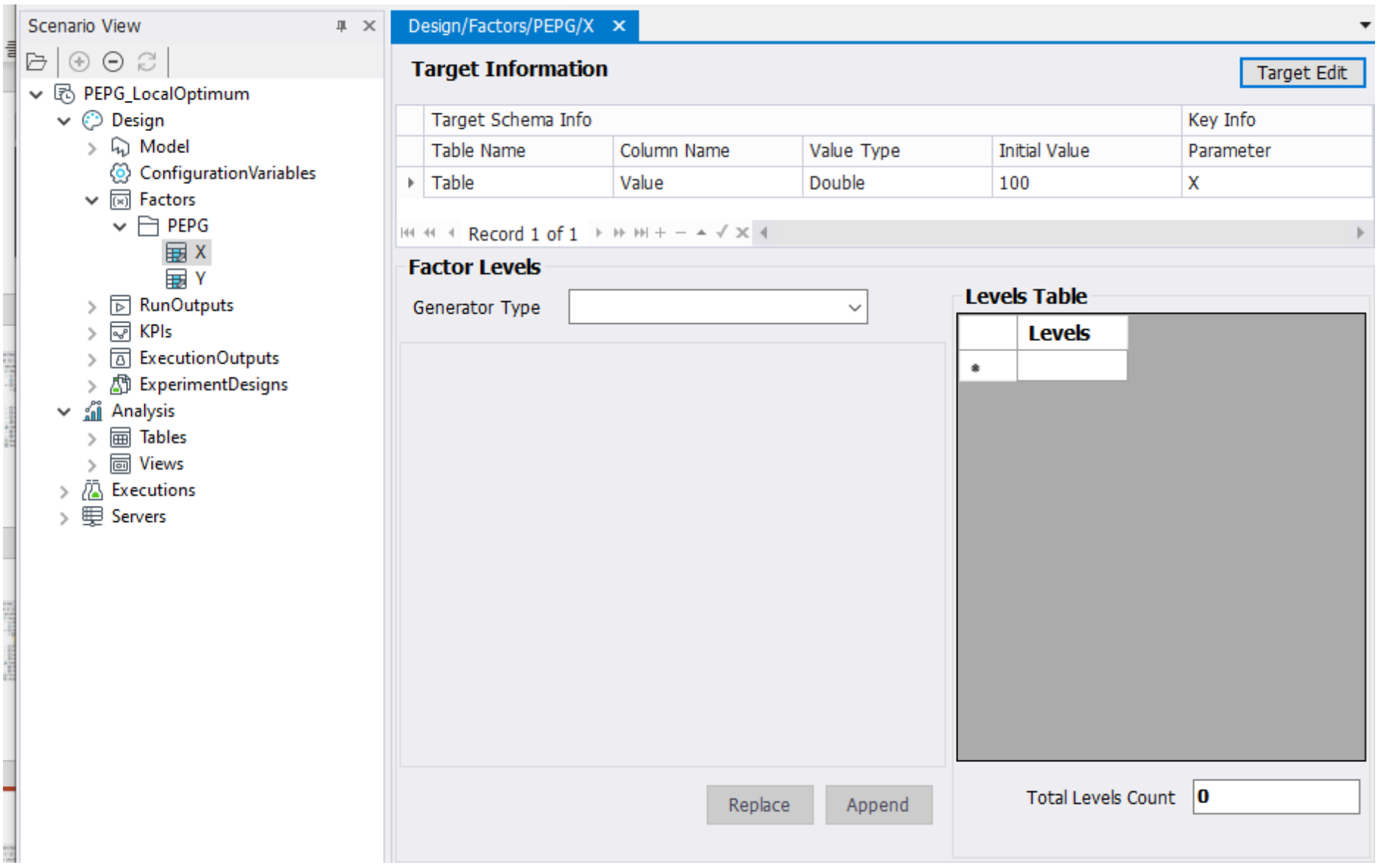


▼ Row의 속한 모든 Cell을 Target Factor로 지정하기

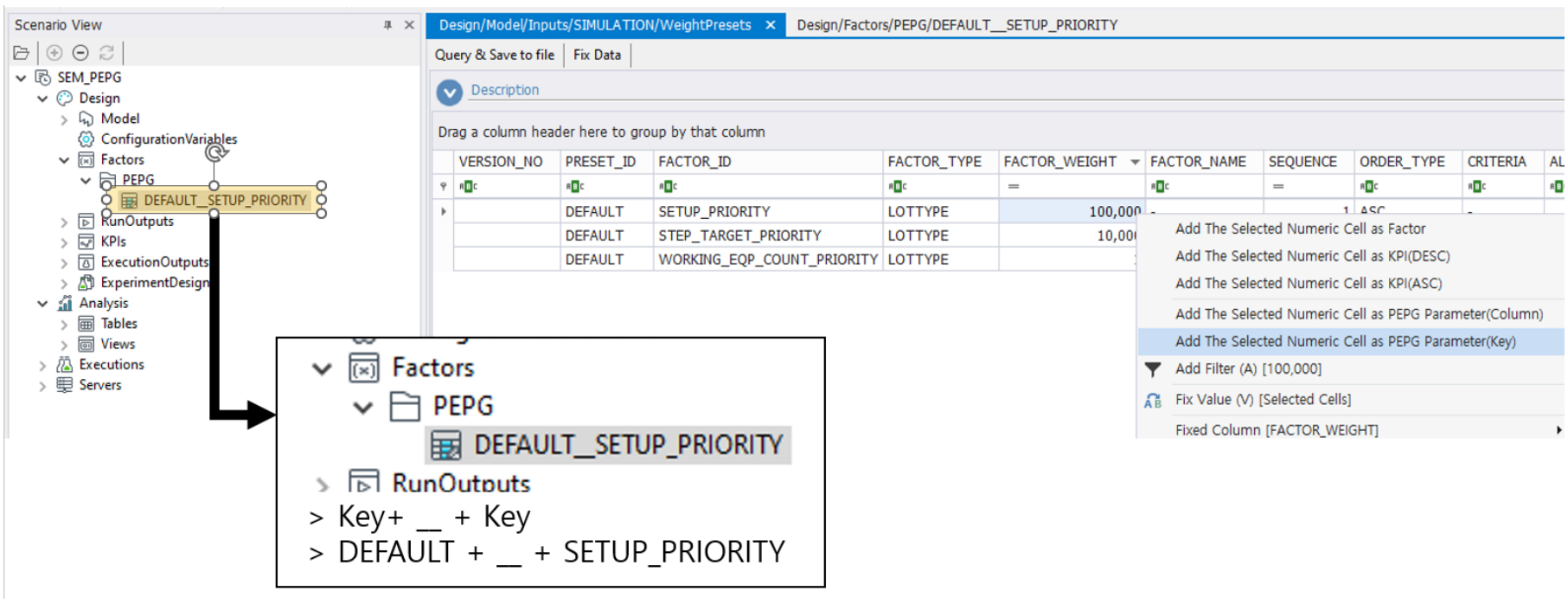


원하는 Parameter를 Input Table의 값에서 Add만 해주면 아래와 같이 자동으로 Factor들로 생성이 됩니다. Target도 따로 설정해줄 필요 없이 자동으로 생성됩니다.

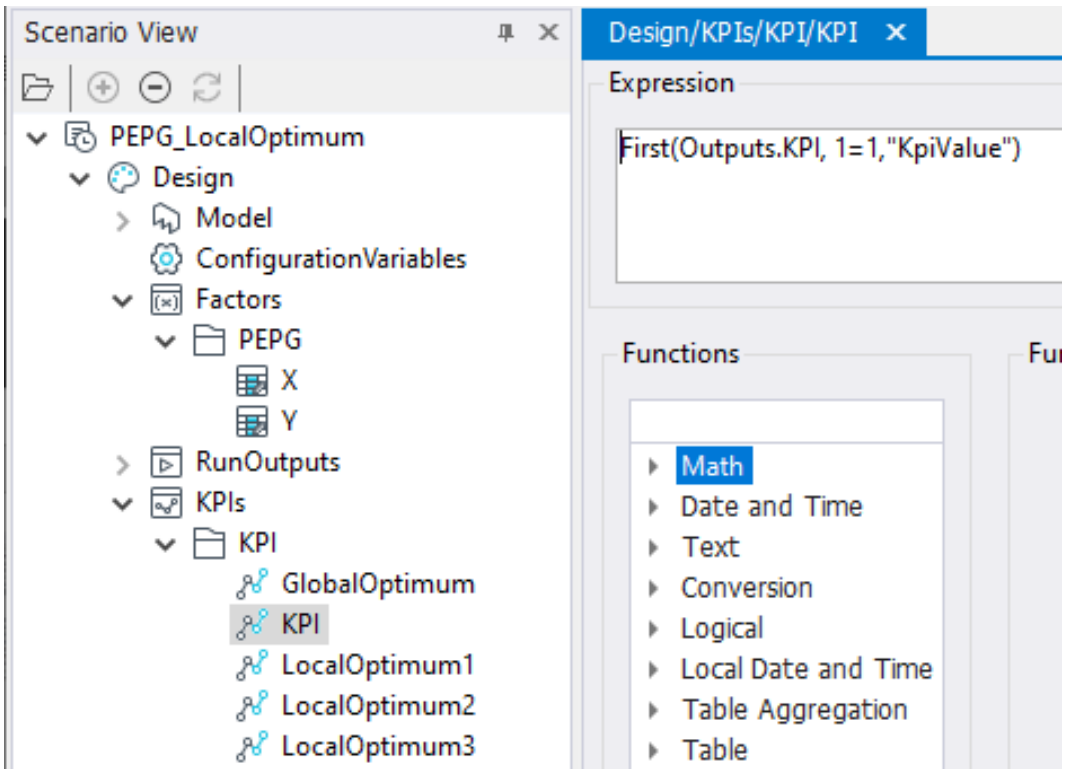
📢 2022.123.0 이전의 WISE에서는 Expression으로 아래가 표시됩니다.



- Target Factor들의 구분자 : Key로 생성된 Factor의 이름은 Key들의 조합으로 구성되어 있습니다. 해당 Key들은 아래와 같이 구분자 **"__"로 구성되어 있습니다. (** "__" 는 '_'를 두 번 입력한 문자입니다)



4) KPI 설정



KPI는 시나리오의 결과나 성능을 측정하는 지표입니다. Parameter(Factor)들을 조정하여 향상시킬 지표를 KPI로 설정합니다. Target KPI가 하나 이거나 독립적이라면 위와같이 그래도 Output값에서 가져올 수 있습니다. 그러나Target KPI가 여러개의 지표를 반영해야 한다면 여러 개의 지표를 KPI로 생성하고 그 지표들의 Target KPI에 반영될 수 있도록 Target KPI의 Expression을 설정해야 합니다. 아래의 간단한 예시를 들어놨습니다.

▼ (예시)

생산량(PROD_QTY)을 늘리고 생산 일정 만족(ON_TIME)을 높이면서 생산 일정 지연(DELAY)은 최소한으로 줄이는 Target KPI를 생성합니다. 가장 먼저 결과 지표인 생산량(PROD_QTY), 생산 일정 만족(ON_TIME), 생산 일정 지연(DELAY)을 KPI로 생성합니니다.

KPI

- PROD_QTY : Count(Outputs.EqpPlan,a.STEP_ID = "STEP07","PRODUCT_ID")
- ON_TIME : First(CustomRunOutputs.OnTimeTable,1=1,"ON_TIME")
- DELAY : First(CustomRunOutputs.OnTimeTable,1=1,"DELAY")

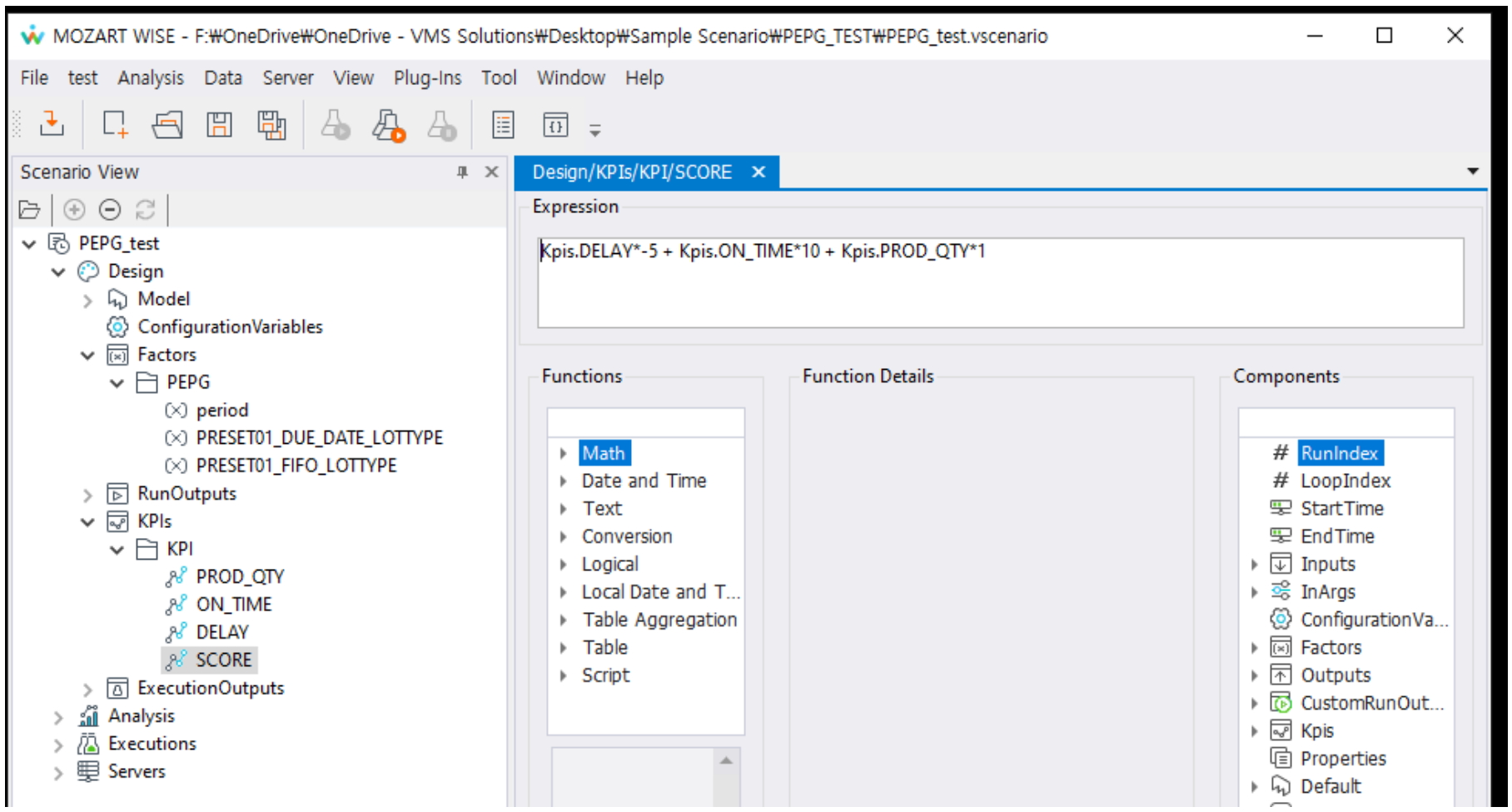
> ❓ CustomRunOutputs의 경우 Model이 제공하는 Output으로 알기 어려운 KPI를 얻기 위해 RunOutputs를 생성하고 해당 Output 결과를 KPI로 가져오는 Expression을 설정하였습니다.

KPI들을 생성된 뒤에 3개의 KPI들이 반영된 하나의 Target KPI를 생성합니다.

Target KPI

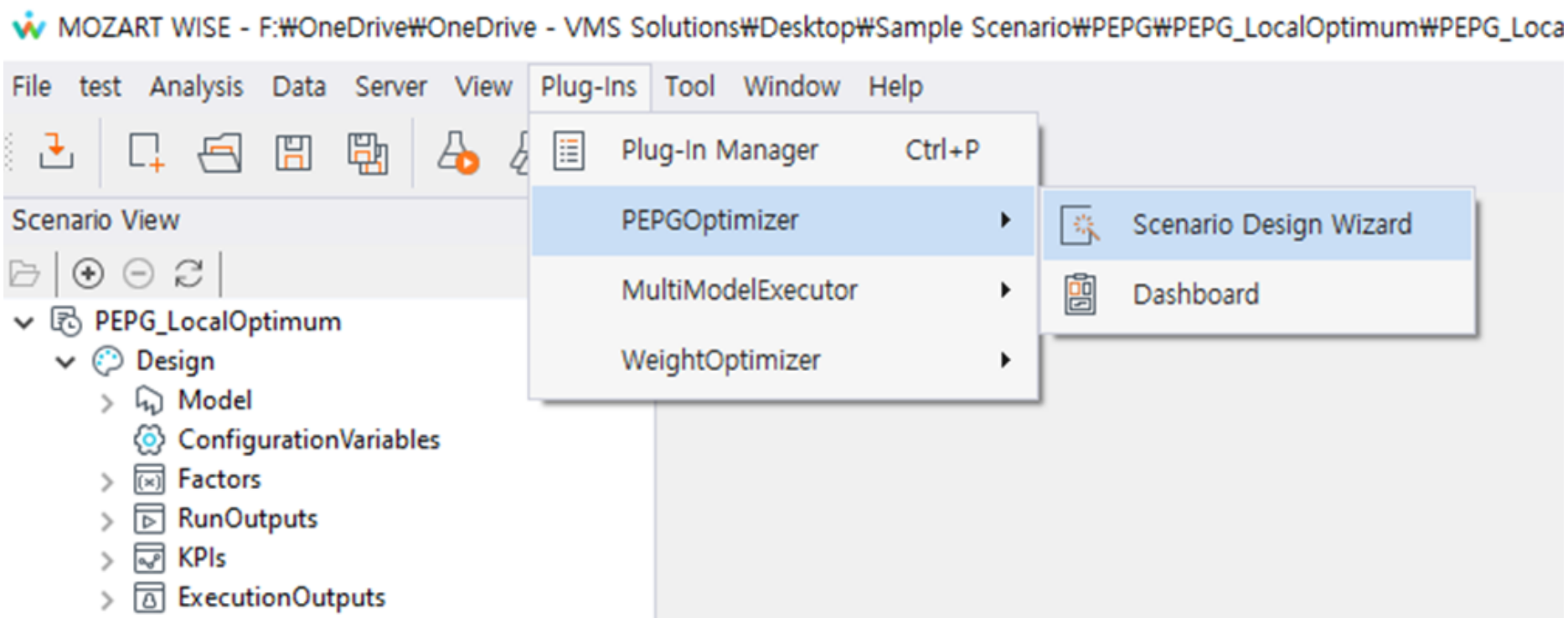
- SCORE : Kpis.DELAY*-5 + Kpis.ON_TIME*10 + Kpis.PROD_QTY*1

> 💡 DELAY를 줄이고(-5), ON_TIME을 늘리고(10), PROD_QTY도 늘립니다(1)



5) PEPG Execution 생성 및 설정 (Scenario Design Wizard)

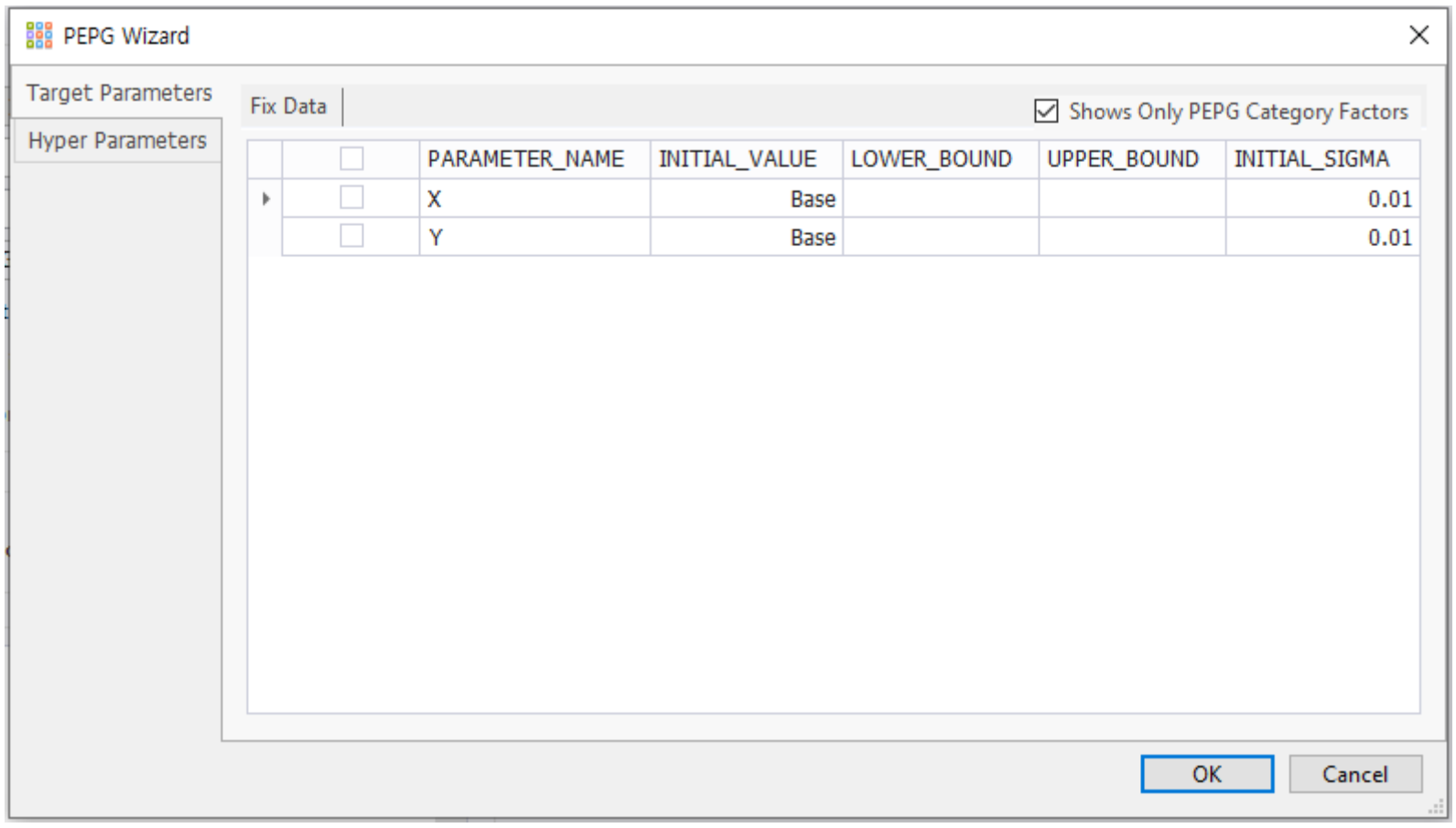
PEPG 실험을 위한 Execution을 생성하기 위해서 Plug-Ins/PEPGOptimizer/Scenario Design Wizard를 선택합니다.



Scenario Design Wizard(강화학습을 수행하기 위한 WISE Setting을 자동으로 해주는 기능)를 열면 아래와 같이 두개의 Tab이 보입니다.

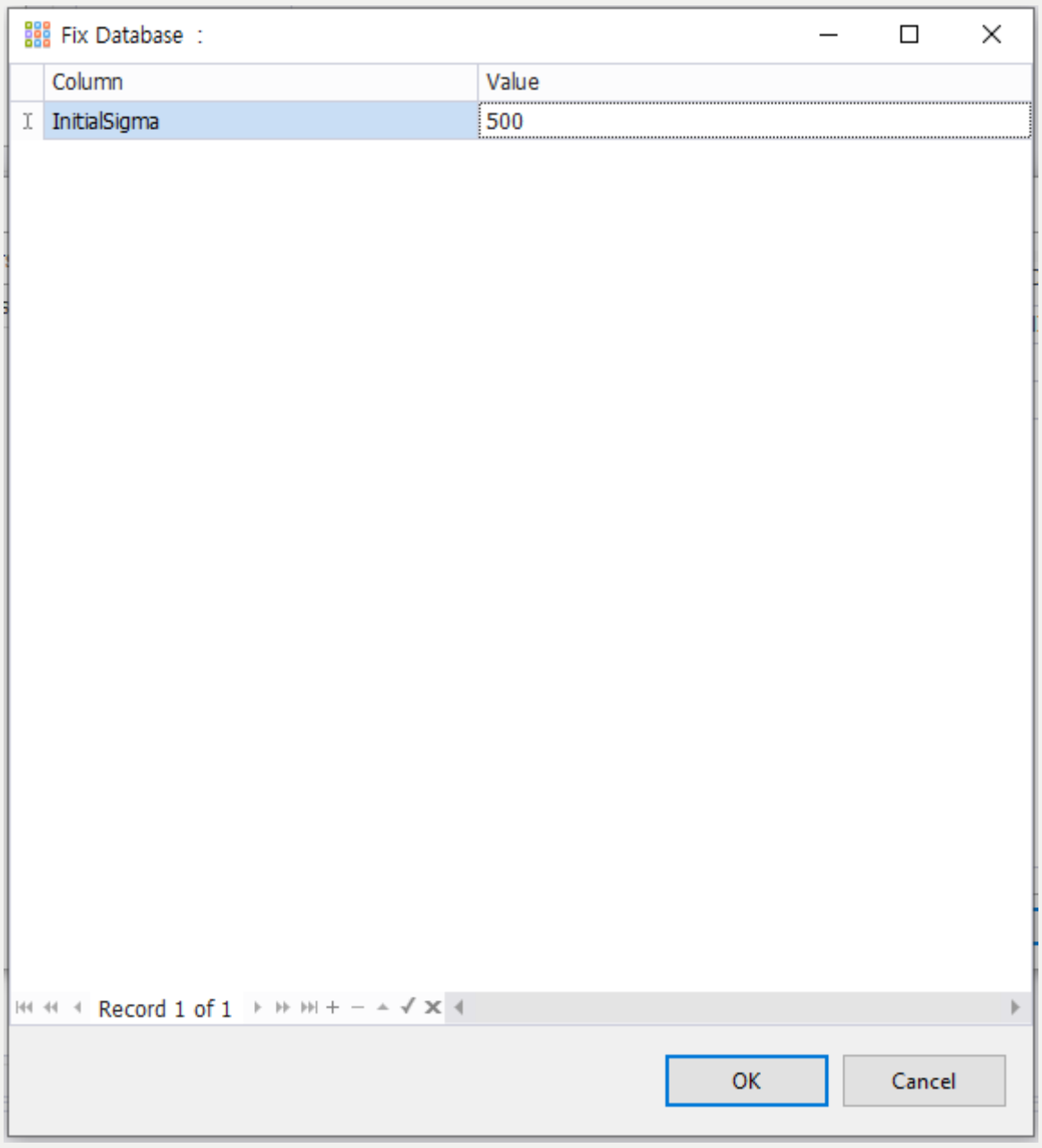
- **Target Parameters** : 사용자가 앞서 Parameter로 설정한 Factor들을 표시 및 조정하는 Tab
- **Hyper Parameters** : PEPG 실험을 위한 학습 설정을 조정하는 Tab

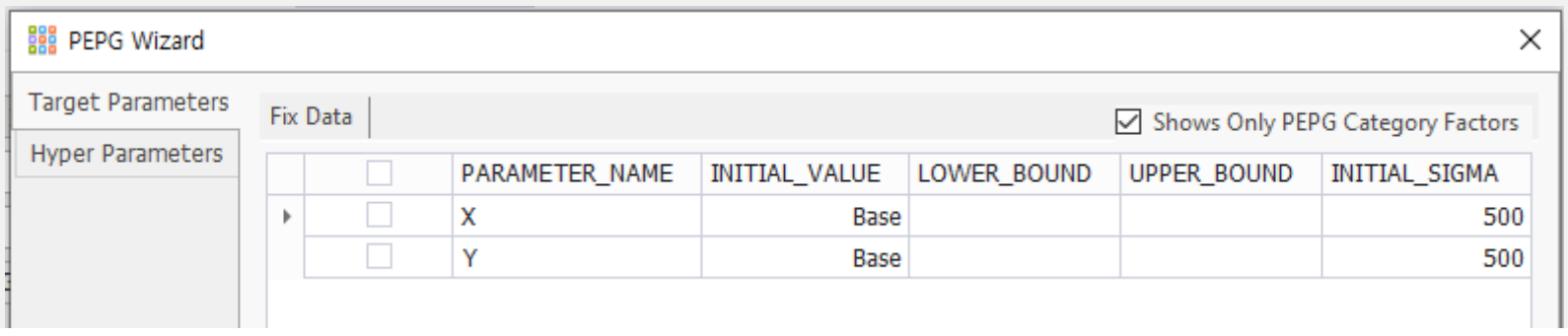
Target Parameter



▼ Fix Data 기능

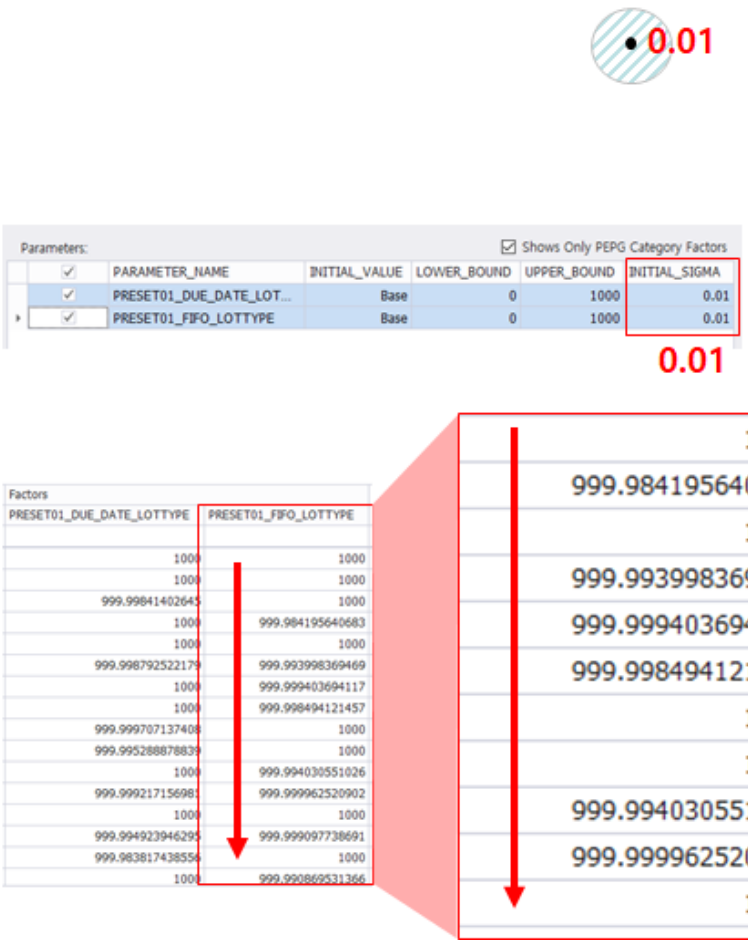
특정 컬럼의 값을 동일하게 바꿉니다.



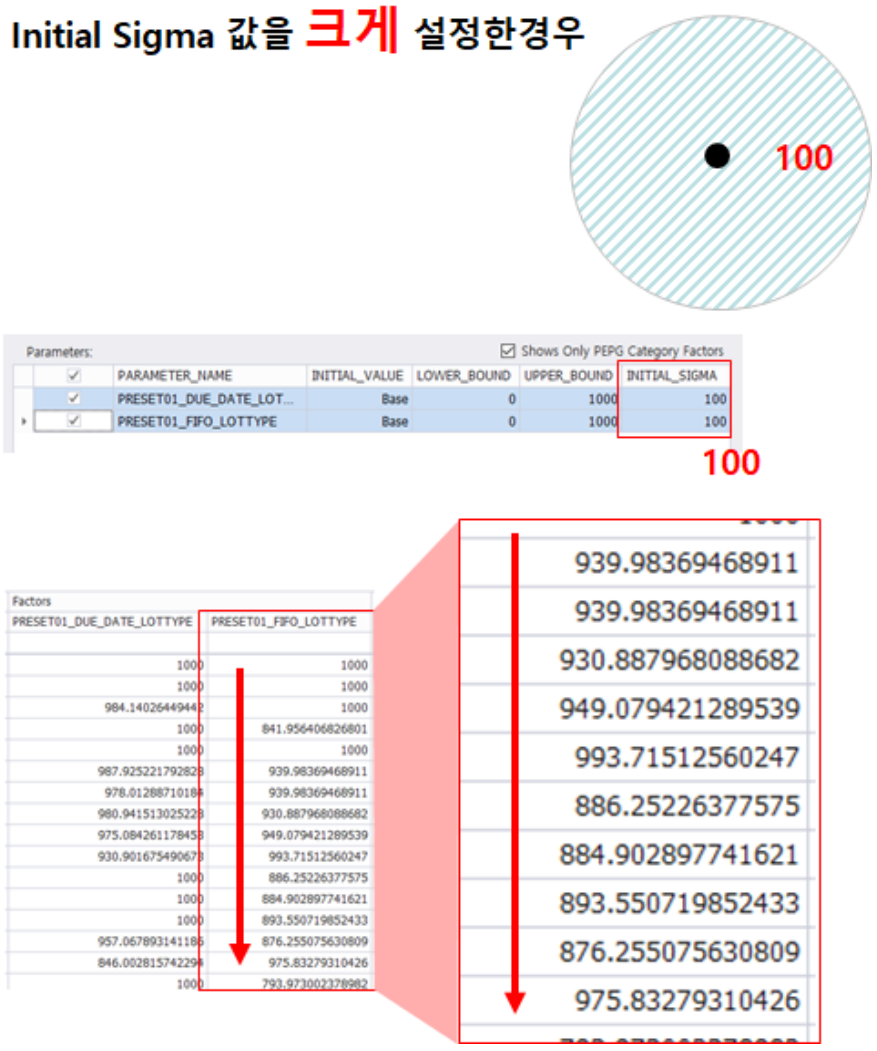


1. INITIAL_VALUE : 초기 파라메터 설정 - Base (Base에 설정된 값을 초기값으로 설정), 다른 값으로 초기값 변경 가능
2. LOWER_BOUND : Parameter의 최소 범위 설정(해당 값 미만으로 파라메터 값이 변경되지 않음)
3. UPPER_BOUND : Parameter의 최대 범위 설정(해당 값 초과로 파라메터 값이 변경되지 않음)
4. INITIAL_SIGMA : 샘플링 범위 설정(설정 값의 약 3배의 범위에서 샘플링 값이 결정됨)

Initial Sigma 값을 작게 설정한경우



Initial Sigma 값을 크게 설정한경우

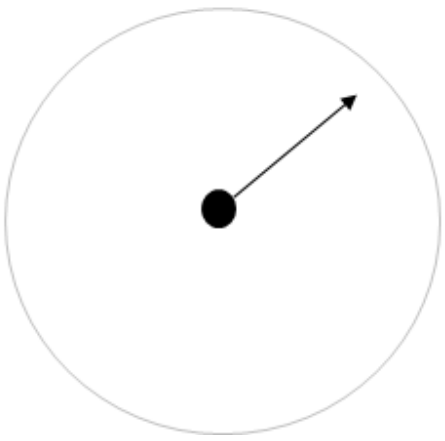


> 💡 PEPG 방법론에 의해서 실험이 반복될 수록 최적값을 찾기 위해 샘플링 범위 좁혀 나가게 됩니다. 이는 즉 실험이 반복될 수록 SIGMA 값이 줄어드는 것을 의미하며, SIGMA값이 0이 되면, INITIAL_SIGMA로 다시 변경해서 주위를 탐색하는 로직이 포함되어 있습니다.

Hyper Parameters

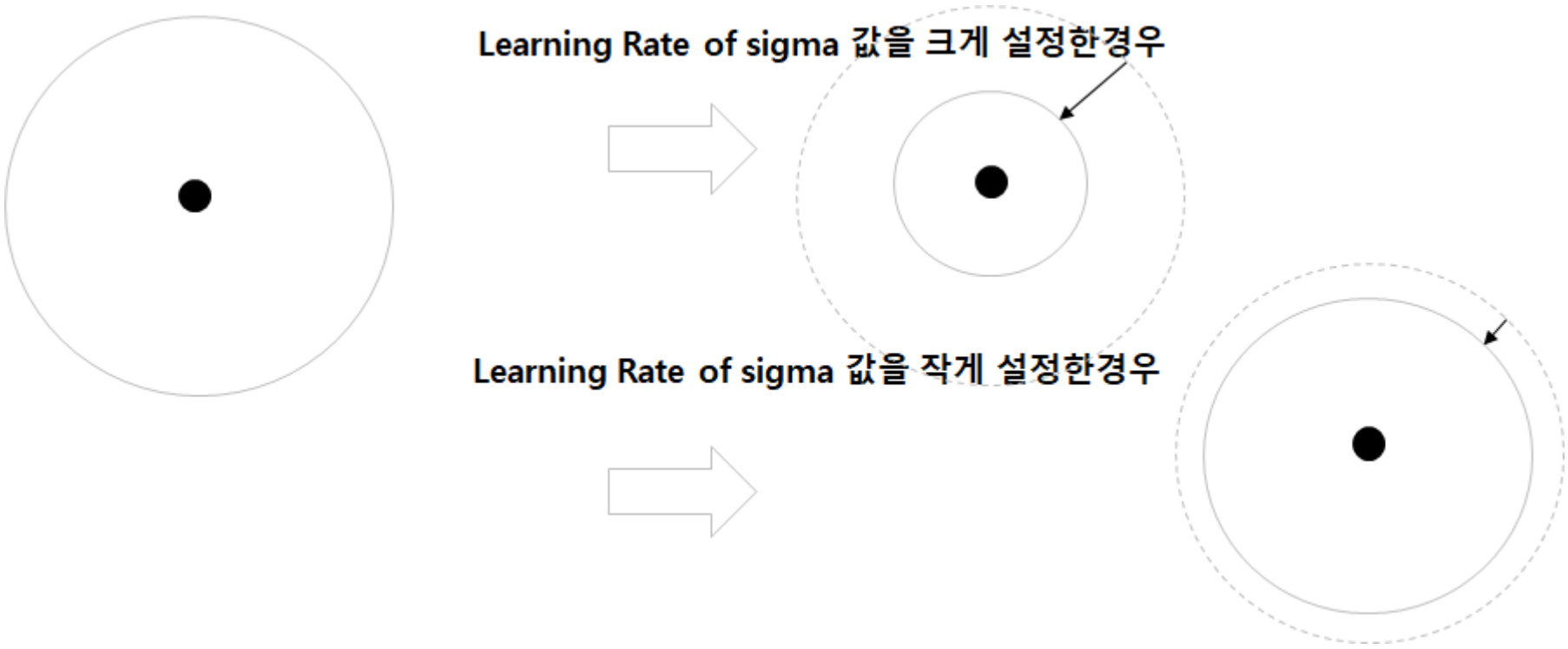
1. Number of Sampling Pair : 샘플링 수
2. Learning Rate of Mu : 샘플이 이동하는 거리를 결정하는 파라메터(1 추천)

Learning Rate of mu 값을 크게 설정한경우

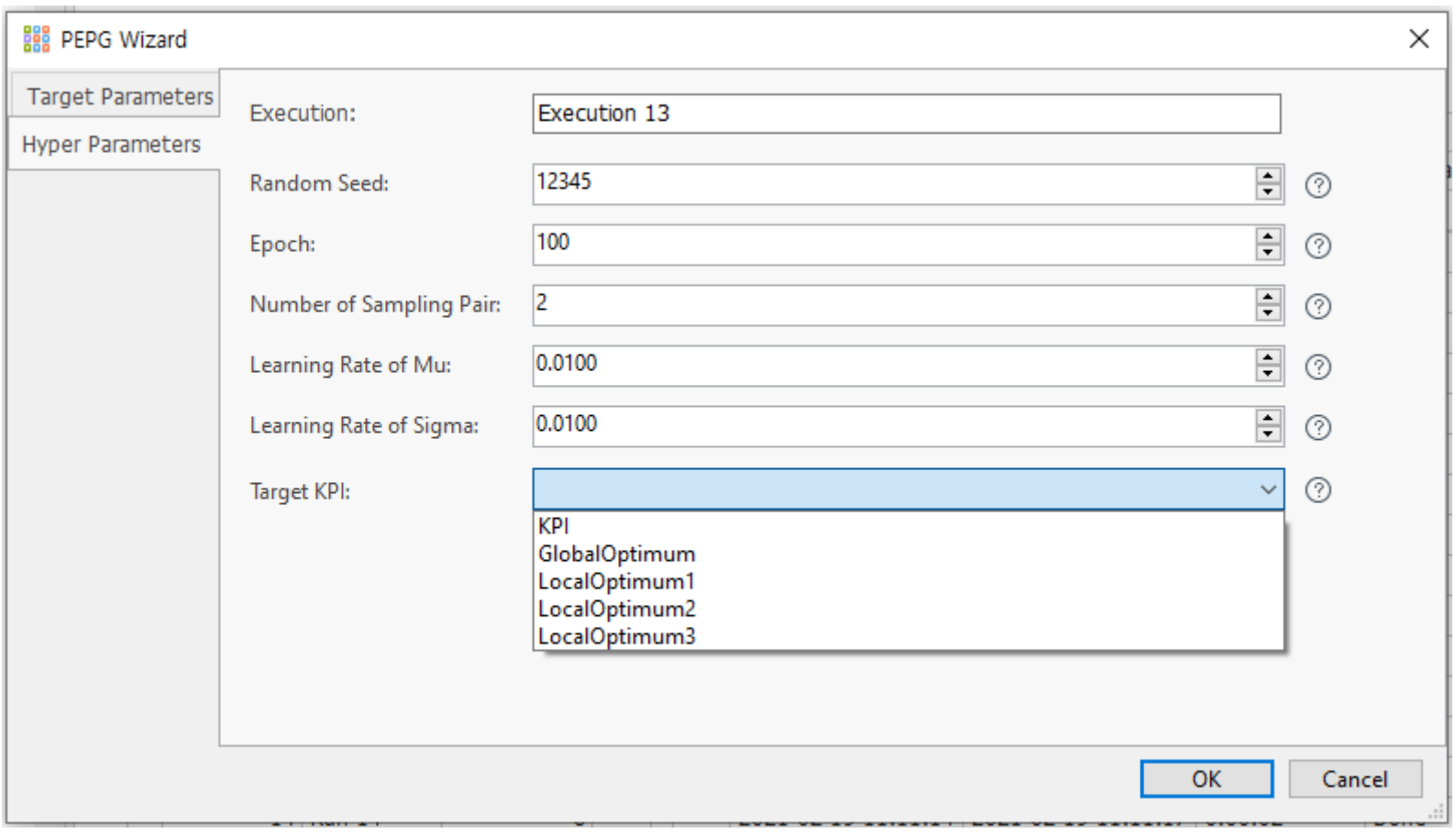


Learning Rate of mu 값을 작게 설정한경우

3. Learning Rate of Sigma : 샘플링의 범위를 결정하는 파라메터 (0.01 추천)



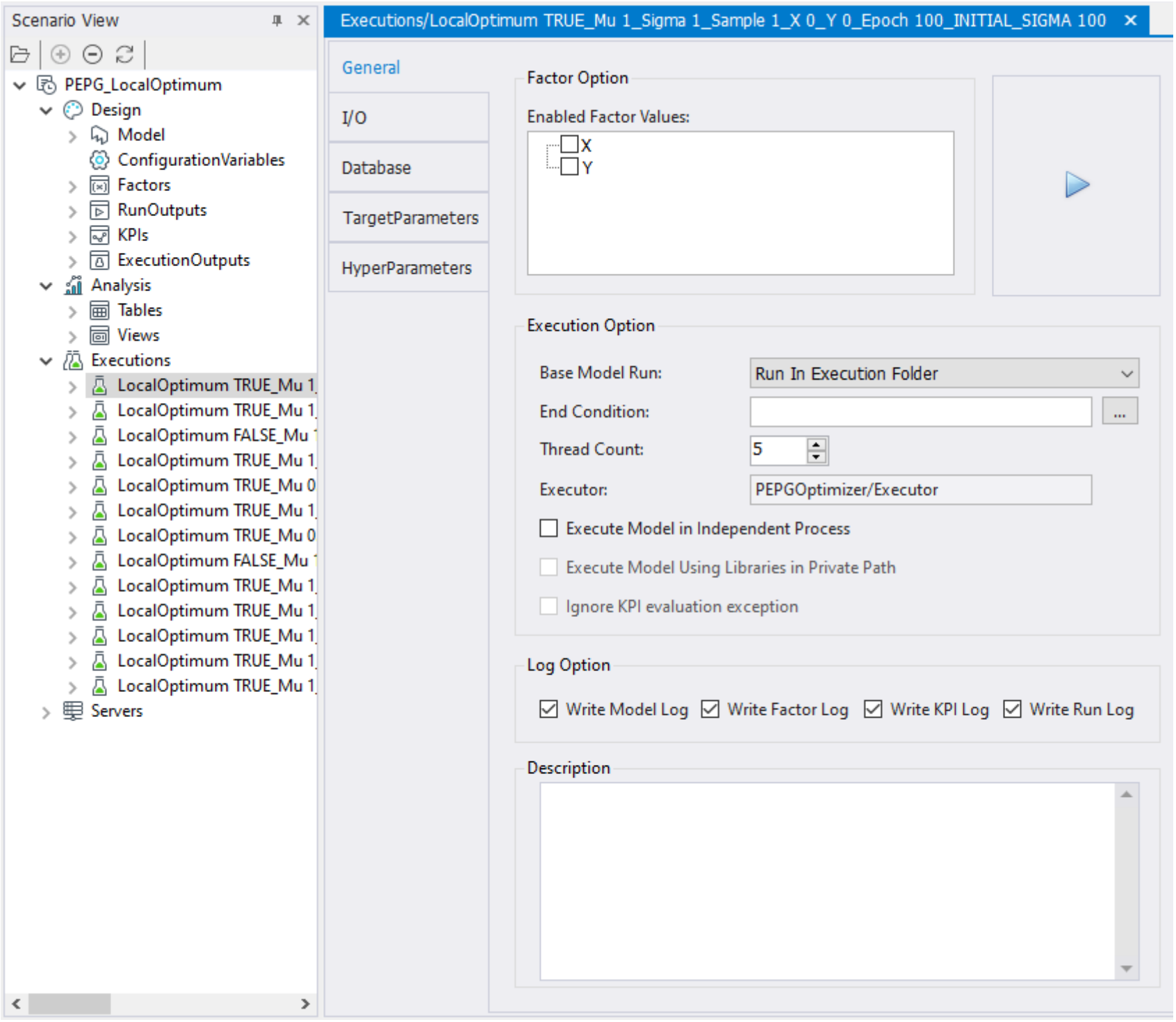
4. Target KPI : 목표 KPI



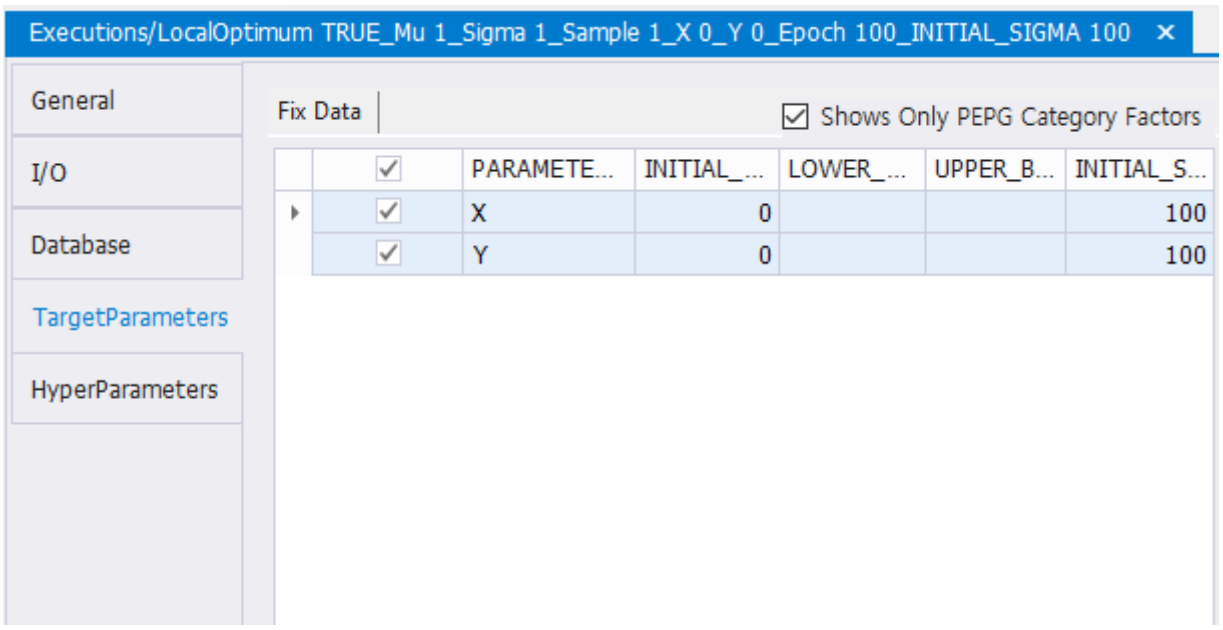
6) 실행 및 결과 분석

PEPG Execution

- PEPG 실험을 위한 Execution

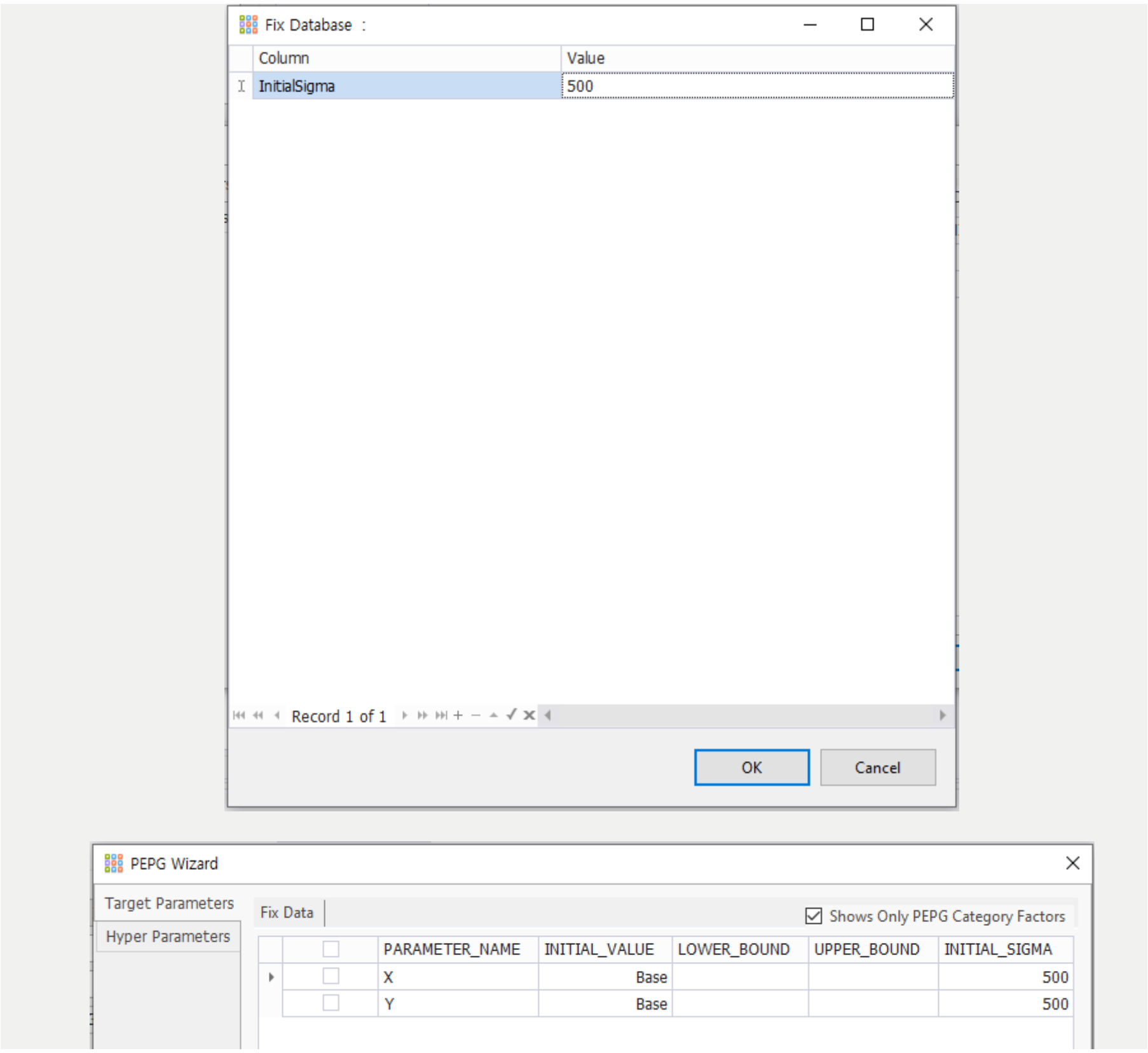


- PEPG 실험의 Factor를 확인 및 설정하기 위해서는 **General Tab**의 Enabled Factor Values가 아닌 **Target Parameters Tab**을 확인해야합니다

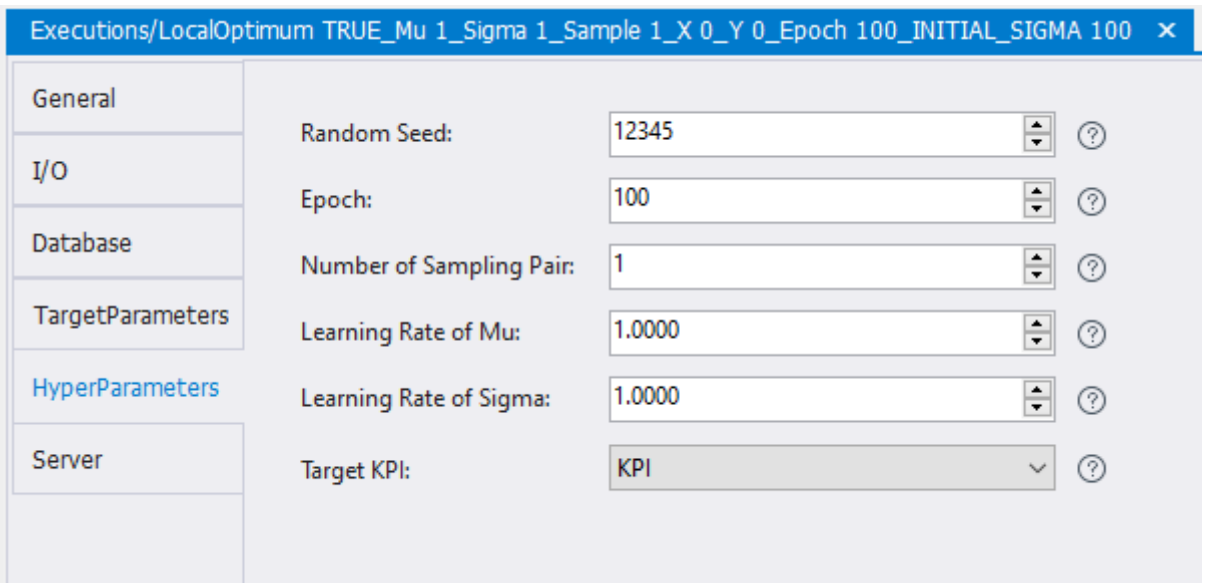


▼ Fix Data 기능

특정 컬럼의 값을 동일하게 바꿉니다.



- 생성된 Execution에서 Hyper Parameter를 수정할 수 있습니다.



실험 결과 및 분석 화면

1. **Report**

   Report에서 Target Fator의 변화값 과 KPI의 변화값 확인 가능

Execution Analysis Report

Execution Start Time: 2021-02-19 11:11:00
Execution End Time: 2021-02-19 11:16:14
Estimated End Time: 2021-02-19 11:16:14
Execution Elapsed Time: 00:05:14
Execution State: -
Total Run Count: 298/298
Experiment Design : HasNotDesign

Execution		Parallel Run		Performance					Factors		KPIs					Models Info	
Run Index	Run Name	Loop Index	Selected	Start Time	End Time	Elapsed Time	Run State	Engine Status	X	Y	KPI	Global Optimum	Local Optimum1	Local Optimum2	Local Optimum3	Models Info	Message
0	BaseRun	1	☑	2021-02-19 11:11:00	2021-02-19 11:11:01	0:00:01	Done	Success	100	100	100	100	-310	-110	-110	Open in Studio	
1	Run 1	2	☑	2021-02-19 11:11:02	2021-02-19 11:11:04	0:00:02	Done	Success	0	0	-100	-100	-110	-110	-110	Open in Studio	
2	Run 2	2	☐	2021-02-19 11:11:02	2021-02-19 11:11:04	0:00:02	Done	Success	-15.8597355055802	12.0747782071719	-82.065486	-103.784957	-106.215043	-137.934514	-82.065486	Open in Studio	
3	Run 3	2	☐	2021-02-19 11:11:02	2021-02-19 11:11:05	0:00:02	Done	Success	15.8597355055802	-12.0747782071719	-82.065486	-96.215043	-113.784957	-82.065486	-137.934514	Open in Studio	
4	Run 4	3	☑	2021-02-19 11:11:05	2021-02-19 11:11:07	0:00:02	Done	Success	0	0	-100	-100	-110	-110	-110	Open in Studio	
5	Run 5	3	☐	2021-02-19 11:11:06	2021-02-19 11:11:08	0:00:02	Done	Success	158.043593173199	60.0163053108897	1.972712	1.972712	-328.059898	-128.059898	-208.027288	Open in Studio	
6	Run 6	3	☐	2021-02-19 11:11:05	2021-02-19 11:11:08	0:00:02	Done	Success	-158.043593173199	-60.0163053108897	-8.027288	-318.059898	-8.027288	-208.027288	-128.059898	Open in Studio	
7	Run 7	4	☑	2021-02-19 11:11:09	2021-02-19 11:11:11	0:00:02	Done	Success	158.043593173199	60.0163053108897	1.972712	1.972712	-328.059898	-128.059898	-208.027288	Open in Studio	
8	Run 8	4	☐	2021-02-19 11:11:08	2021-02-19 11:11:10	0:00:02	Done	Success	160.972219096586	12.905093699722	-48.067125	-48.067125	-283.877313	-83.877313	-258.067125	Open in Studio	
9	Run 9	4	☐	2021-02-19 11:11:08	2021-02-19 11:11:10	0:00:02	Done	Success	155.114967249811	107.127516922057	37.757516	37.757516	-372.242484	-172.242484	-172.242484	Open in Studio	
10	Run 10	5	☑	2021-02-19 11:11:11	2021-02-19 11:11:13	0:00:02	Done	Success	155.114967249811	107.127516922057	37.757516	37.757516	-372.242484	-172.242484	-172.242484	Open in Studio	
11	Run 11	5	☐	2021-02-19 11:11:12	2021-02-19 11:11:14	0:00:02	Done	Success	146.019240649383	160.858947835417	-6.878188	-6.878188	-416.878188	-216.878188	-216.878188	Open in Studio	
12	Run 12	5	☐	2021-02-19 11:11:11	2021-02-19 11:11:14	0:00:02	Done	Success	164.210693850239	53.3960860086976	-10.814608	-10.814608	-327.60678	-127.60678	-220.814608	Open in Studio	
13	Run 13	6	☑	2021-02-19 11:11:14	2021-02-19 11:11:17	0:00:02	Done	Success	154.730830047727	109.39674096998	35.872429	35.872429	-374.127571	-174.127571	-174.127571	Open in Studio	
14	Run 14	6	☐	2021-02-19 11:11:14	2021-02-19 11:11:17	0:00:02	Done	Success	197.662936906541	-44.6004432877259	-63.062494	-142.26338	-263.062494	-63.062494	-352.26338	Open in Studio	
15	Run 15	6	☐	2021-02-19 11:11:15	2021-02-19 11:11:17	0:00:02	Done	Success	111.798723188913	263.393925227687	-75.192648	-75.192648	-485.192648	-285.192648	-285.192648	Open in Studio	
16	Run 16	7	☑	2021-02-19 11:11:17	2021-02-19 11:11:20	0:00:02	Done	Success	157.16696551508	100.658339109657	42.174695	42.174695	-367.825305	-167.825305	-167.825305	Open in Studio	
17	Run 17	7	☐	2021-02-19 11:11:17	2021-02-19 11:11:20	0:00:02	Done	Success	165.814787625892	191.588234472296	-57.403022	-57.403022	-467.403022	-267.403022	-267.403022	Open in Studio	
44,253		15,148									759.529747	-6,630.405515	-69,174.824221	-32,897.230321	-45,887.999415		

Record 1 of 298

- 위의 Report의 보면 Selected Column이 존재합니다. 해당 Column을 통해 선택되어 있는 Run들이 기준 Run이고 선택되지는 않았지만 같은 Loop Index에서 돈 Run들이 Sampling된 Run임을 확인할 수 있습니다.

Execution Analysis Report

Execution Start Time: 2021-02-19 11:11:00
Execution End Time: 2021-02-19 11:16:14
Estimated End Time: 2021-02-19 11:16:14
Execution Elapsed Time: 00:05:14
Execution State: -
Total Run Count: 298/298
Experiment Design : HasNotDesign

Execution		Parallel Run	
Run Index	Run Name	Loop Index	Selected
0	BaseRun	1	☑
1	Run 1	2	☑
2	Run 2	2	☐
3	Run 3	2	☐
4	Run 4	3	☑
5	Run 5	3	☐
6	Run 6	3	☐
7	Run 7	4	☑
8	Run 8	4	☐
9	Run 9	4	☐

- Report를 통해 PEPG 알고리즘의 **<u>학습진행 로직</u>**을 확인할 수 있습니다. 아래에 예시를 참고해주세요

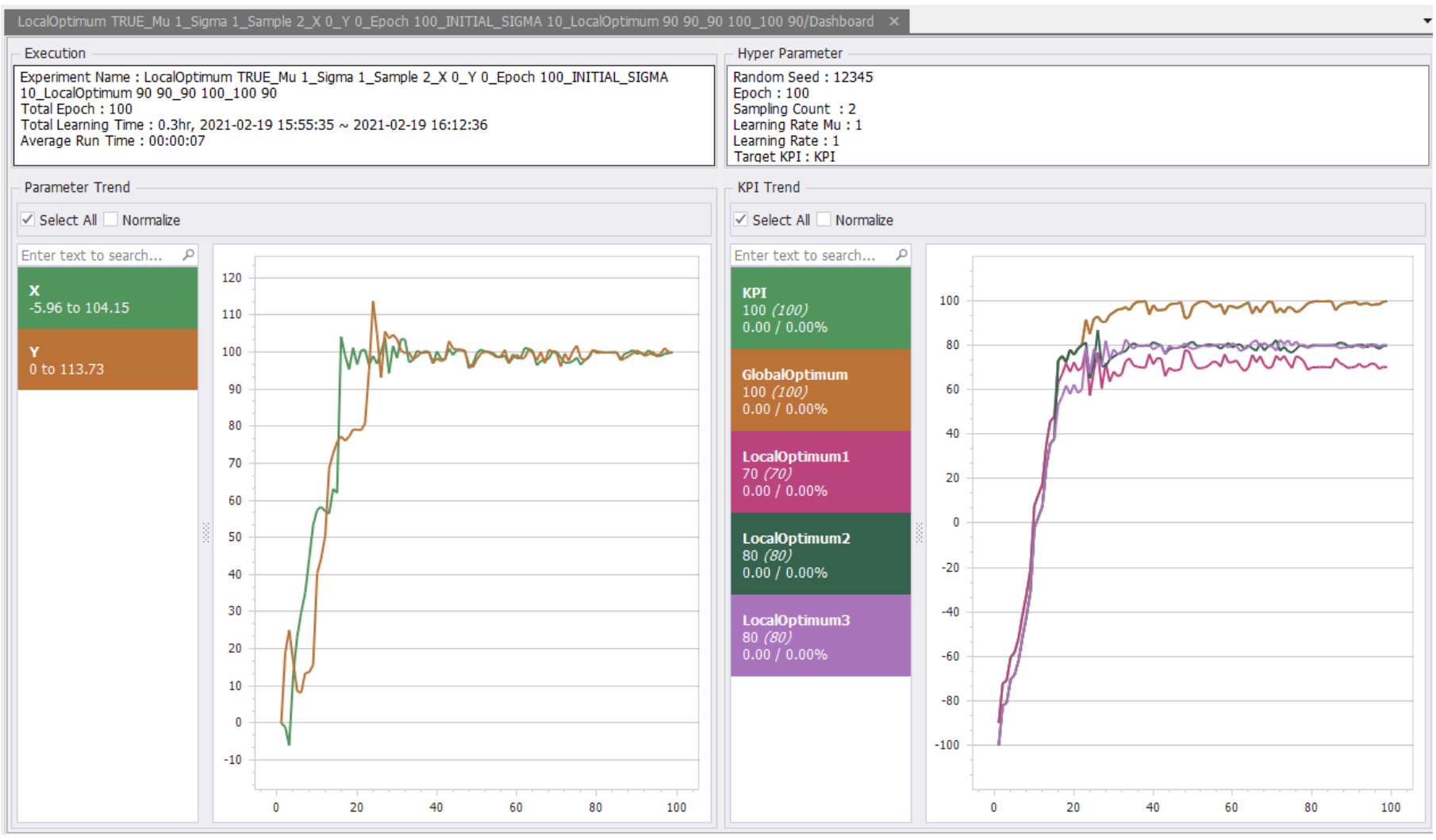
▼ (예시)

1. BaseRun이 실행됩니다.
2. BaseRun이 Loop Index 1의 기준 Run(Run 1)이 됩니다.
3. 기준 Run(Run 1)의 주변 Sampling Run(Run 2,3,4,5)들을 생성합니다.
4. Sampling Run들의 중심 Run을 구하고 다음 Loop Index 2의 기준 Run(Run 6)으로 설정합니다
5. 위의 작업을 반복합니다.

Execution		Parallel Run		Factors
Run Index	Run Name	Loop Index	Selected	X
			■	
0	BaseRun	1	✓	100
1	Run 1	2	✓	0
2	Run 2	2		-1.58597355055802
3	Run 3	2		1.58597355055802
4	Run 4	2		1.20747782071719
5	Run 5	2		-1.20747782071719
6	Run 6	3	✓	-0.973749449682784
7	Run 7	3		-0.680886857344012
8	Run 8	3		-1.26661204202156
9	Run 9	3		-5.68487061079955
10	Run 10	3		3.73737171143398
11	Run 11	4	✓	-5.95768832408803
12	Run 12	4		-1.66447763820666
13	Run 13	4		-10.2508990099694
14	Run 14	4		-21.3574067498587

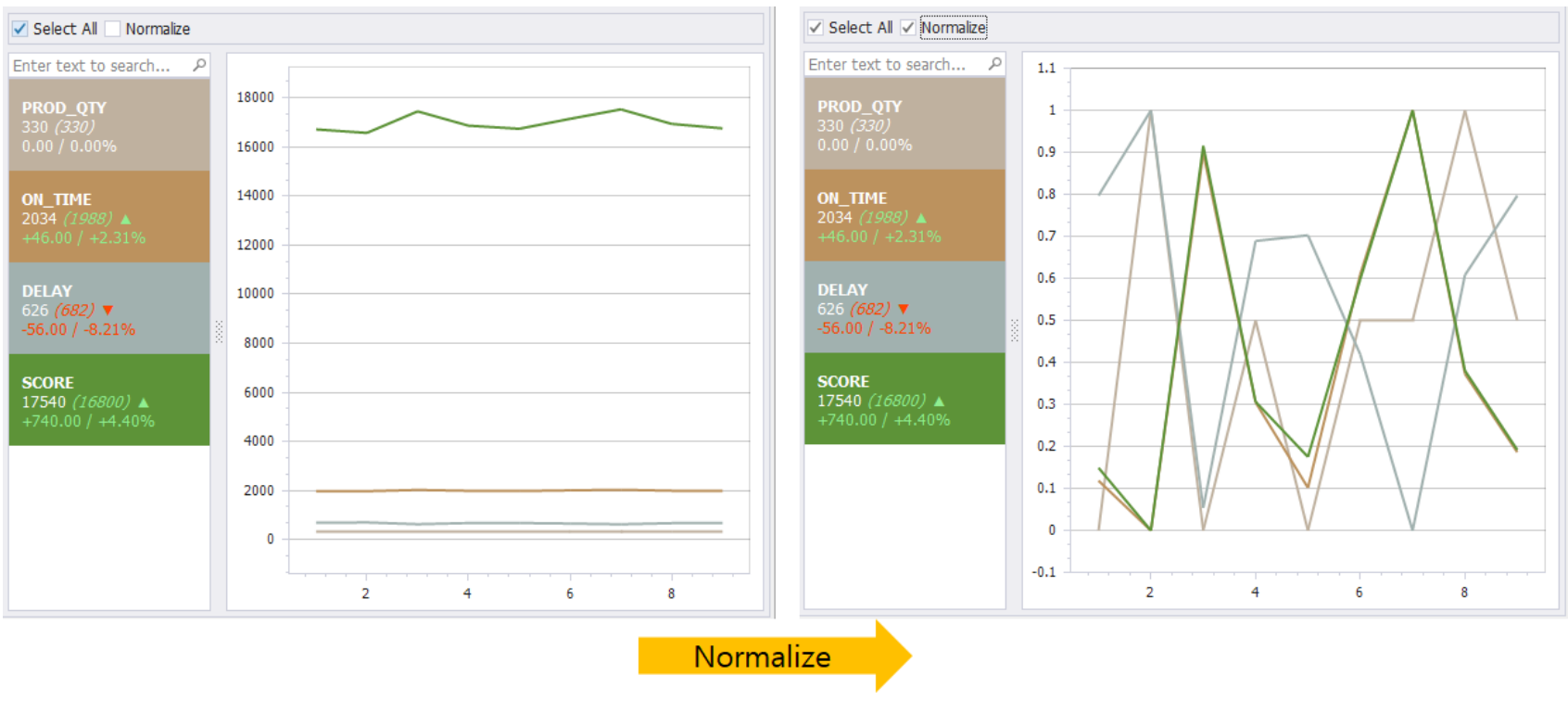
2. **DashBoard**

Plug-Ins/PEPGOptimizer/Dashboard에서 파라메터의 변화값 과 KPI의 변화값 확인 가능



- Execution
  - Experiment Name : Execution 이름
  - Total Epoch : 실험한 횟수
  - Total Learning Time : 총 학습 시간
  - Average Run Time : Run별 평균 시간
- Hyper Parameter : 실험 환경 설정 정보
  - Random Seed
  - Epoch
  - Sampling Count

    - Learning Rate Mu
    - Learning Rate
- Parameter Trend : 조정한 Parameter들의 변화 추이를 Chart로 확인
    - Select All : 모든 Paramter를 선택
    - Normalize : Chart로 표시하고자 하는 모든 값들을 Normalizer
    - Parameter Select View : Chart로 확인하고 싶은 Paramter를 선택
    - Parameter Trend Chart
- KPI Trend : 목표한 KPI 결과 변화를 Chart로 확인
    - Select All : 모든 KPI를 선택
    - Normalize : Chart로 표시하고자 하는 모든 값들을 Normalizer



    - KPI Select View : Chart로 확인하고 싶은 KPI를 선택
    - KPI Trend Chart