# Experiment Design

Experiment Designer 는 What-if Simulation 실험을 위해 요인(Factor)들을 설정 및 수정하고 변경된 요인들을 통한 실험 결과(KPI & Analysis)를 확인할 수 있게 해주는 기능입니다.

- WISE 실험을 위해서는 가장 먼저 Base Model 구동시 관심있는 독립변수(Factor) 와 수준(Factor Value) 을 정의하여 실험을 계획하고 결과평가를 위한 종속변수(KPI) 를 정의하는 작업을 합니다.
- Experiment Design 에서는 앞서 정의된 독립변수(Factor)와 종속변수(KPI)를 통해 실험 계획을 구성합니다.

> 📌 123.0 버전 이후에는 Experiment Design 기능이 추가되었습니다.

- Model의 실험 결과를 제외하고 사용자가 원하는 실험 결과 Table을 구성하기 위해 RunOutput과 ExecutionOutput기능을 제공합니다.

1. Factors
2. KPIs
3. ExperimentDesigns
4. RunOutput
5. ExecutionOutput

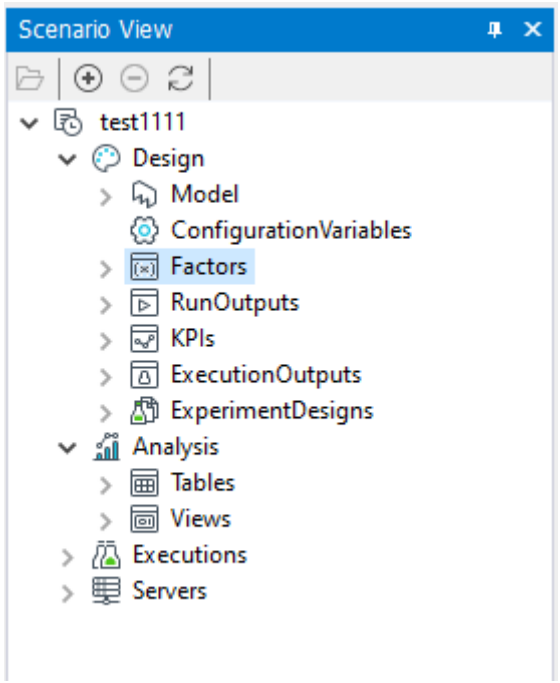
---

123.0 이전 버전의 Factors

> 💡 **123 버전의 WISE에서의 Factor가 이전 버전의 Factor와 달라진 점**
> 1) Factor와 Factor Value(Levels)를 하나의 탭에서 편집이 가능
> 2) 기존의 Factor Value들이 Factor와 통합됨, 새로운 Factor는 Factor Value의 특성을 가짐(예) Data, DataSet, Argument, ArgumentSet, Model, Dlls)
> 3) Factor의 Target 설정시 Expression 대신 선택 버튼으로 쉽게 선택 가능해짐
> 4) Factor내의 Factor Value(Levels)들도 Level Generator로 쉽게 Range, List, Interval등의 기능으로 생성 가능
> 5) 숫자만 선택 가능했던 기존 Factor와 다르게 문자열, 날짜, 논리연산도 Factor로 설정이 가능해지고 해당 Type에 따른 Levels들을 설정할 수 있음

# Factors

Factors 는 시뮬레이션의 결과에 영향을 주는 입력 항목 중 What-If Simulation에 관심 있는 항목을 정의하기 위해 사용되는 요소들입니다. Factors는 WISE 의 Scenario View 의 트리 메뉴에서 **[Design > Factors]** 노드 메뉴를 통해 사용할 수 있으며, Factor 와 실험할 각 수준들(Levels)을 정의할 수 있습니다.

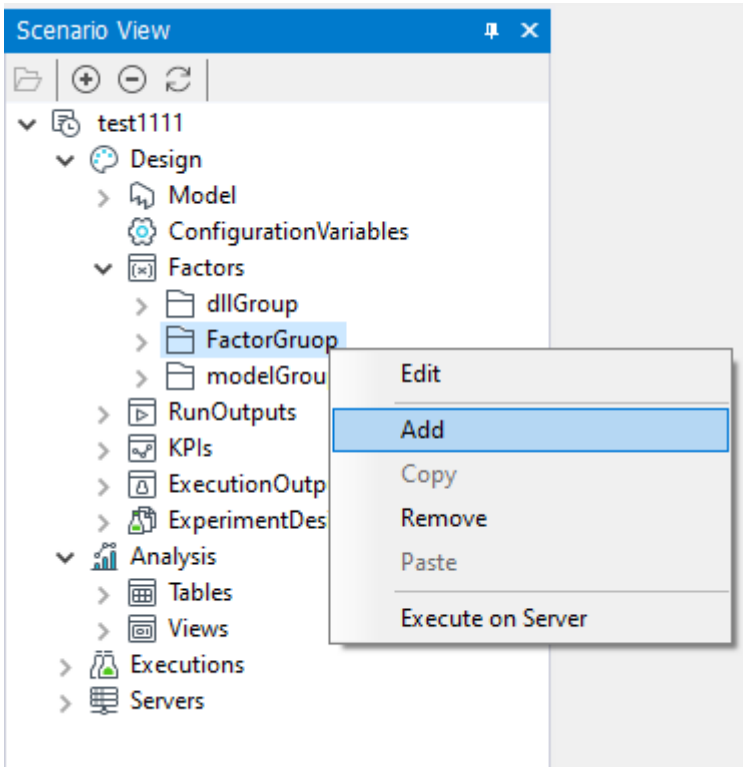



Factors 위치

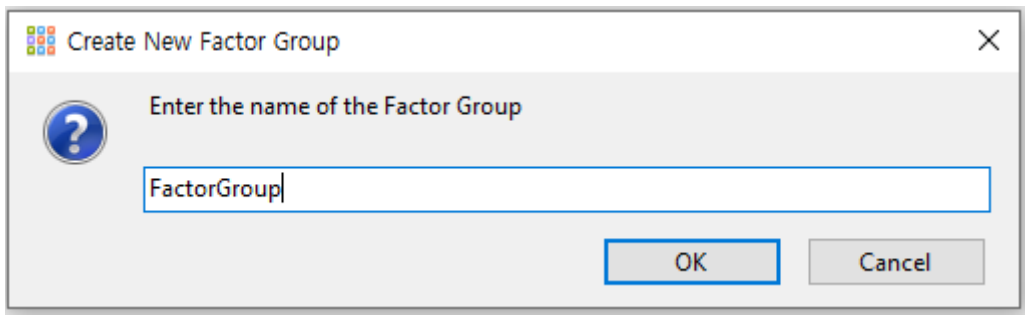
## Factor 등록 절차

Factor 는 **[Design] > [Factors] > [Group 폴더] > [Factor]**에 추가하여 사용할 수 있으며, 사용자의 선택 여하에 따라 Factor를 추가 및 복사, 삭제하여 사용할 수 있습니다.

> 📌 Factor Group, Factor순으로 생성하며, 각 구성요소들은 Edit(이름 변경), Add(구성요소 추가), Copy(복사), Paste(붙여넣기), Remove(삭제) 기능이 동일하게 사용 가능합니다.

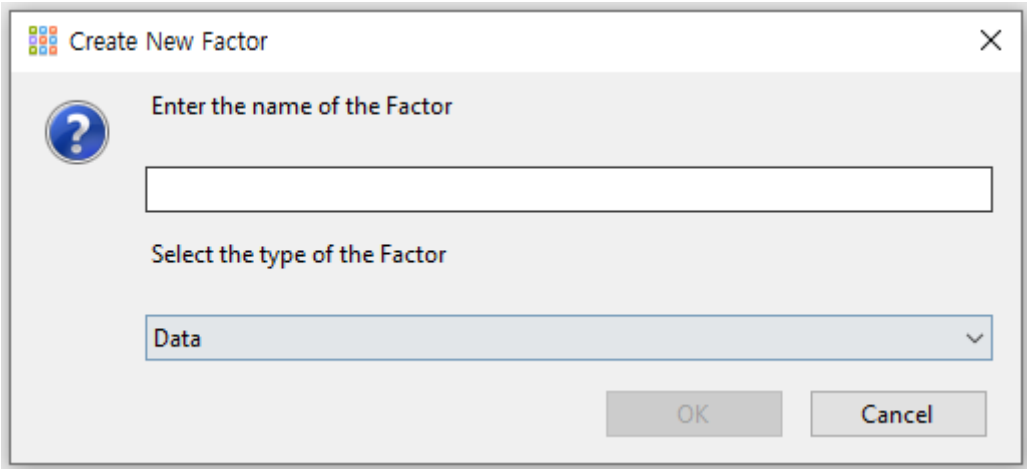



1. **[Design]>[Factors]** 노드에서 **[Add]** 버튼을 눌러 Factor 그룹을 추가합니다.

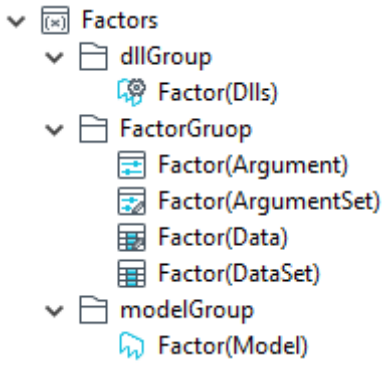



2. 추가된 Factor 그룹 노드에서 **[Add]** 버튼을 눌러 Factor 를 추가합니다.

   1) Factor Group이 등록된 상태에서 Factor Group을 선택한 후 **[Add]** 버튼으로 Factor 를 추가합니다.




2) Factor 의 명칭을 입력하고, 아래의 6가지 유형 중 한가지를 선택합니다.




WISE Scenario 상의 Factor 노드

- `DataSet` : 편집된 InputTable들을 Set들로 저장하고 있으며, 각 Set의 Input Table 을 직접 사용자가 편집하여 정의합니다.
- `Data` : Base Model의 Input Table의 특정 값을 Target으로 정의하고, 변경대상 Input Table의 해당 데이터 값을 편집합니다. 해당 데이터 값은 특정 집합의 범위를 가질 수 있습니다.
- `ArgumentSet` : 편집된 Input Argument들을 Set들로 저장하고 있으며, 각 Set의 Input Argument 들 중 특정 Argument 의 값을 편집합니다.
- `Argument` : Input Argument들 중 특정 Argument 의 값을 직접 편집합니다. 특정 Argument의 값은 특정 집합의 범위를 가질 수 있습니다.
- `Model` : 여러 모델들을 하나의 Factor안에 Set들로 저장하고 있으며, 각 Model의 Input Argument와 Input Data를 그대로 사용합니다. (생성 시 Default Model로 BaseModel이 첫번째 Level로 생성됩니다.)
- `Dlls` : 여러 Engine Dlls를 하나의 Factor안에 Set들로 저장하고 있으며, 각 Dlls를 Engine Dlls로 사용합니다. (생성 시 Default Dlls로 BaseDlls가 첫번째 Level로 생성됩니다.

3. 각 Factor(독립변수)는 Levels를 포함하고 있습니다. Levels는 Factor(독립변수)의 수준들을 의미합니다. 예를 들면 '제품1의 투입간격을 조정하면서 장비의 가동율을 알아보고 싶다' 하면, 투입간격을 Factor(독립변수)로 두고 조정하고 싶은 값들을 Factor(독립변수)의 Levels(수준들)로 설정합니다. 장비의 가동율은 KPI가 됩니다. 이러한 관계를 통해 실험 후 Run(실험) 별 Factor 값과 KPI 의 값의 관계를 분석할 수 있습니다.

> 🚫 동일한 구성요소들은 동일한 이름을 가질 수 없습니다
> Example) Factor Group과 Factor는 동일한 이름을 가질 수 있지만, 동일한 Factor Group들은 동일한 이름을 가질 수 없습니다.

## Levels 정의

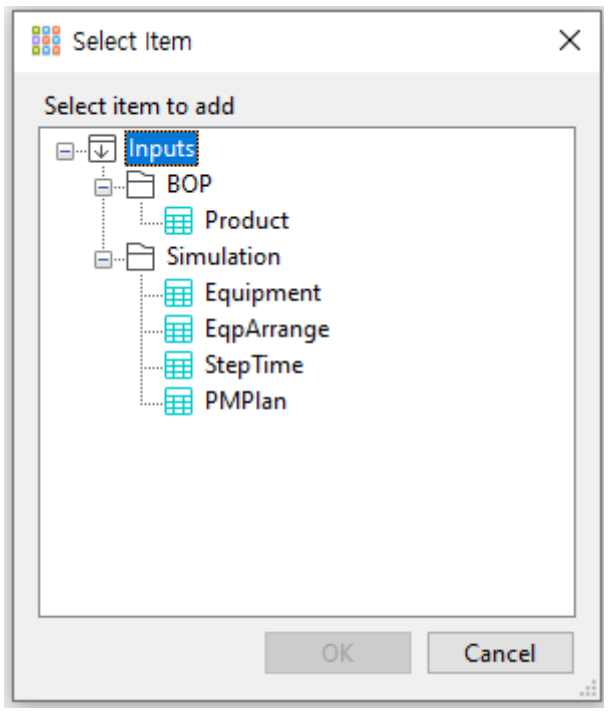
Levels 는 해당 Factor 에 대해 실험할 수준들을 정의합니다. 한 개의 Factor 에 대해 여러 개의 수준(Level)들를 등록할 수 있습니다. Factor 로 정의할 수 있는 대상은 Base Model 의 모든 Input 과 Input Argument 항목입니다. 즉, Factor의 Level은 Base Model 에서 수정할 수 있는 정보 중 Base Model 과 다른 하나의 실행 케이스를 정의합니다.

예를 들어 Input Argument 에 `DefaultSetupTimeMinute` 이라는 변수가 정의되어 있다고 할 때 Factor 를 Setup Time 으로 정의한 후, Levels에 `DefaultSetupTimeMinute` 의 값을 10, 15, 20, 25, 30 로 설정한 5개의 Level을 등록할 수 있습니다. 이경우 등록한 Level 각각이 하나의 실행케이스가 됩니다. Base Model 의 `DefaultSetupTimeMinute` 값이 5로 설정되어 있었다면 상기와 같은 실험계획은 Default Setup Time 의 변화함에 따른 관심 KPI 의 변화정도를 알아보기 위한 실험이 됩니다.

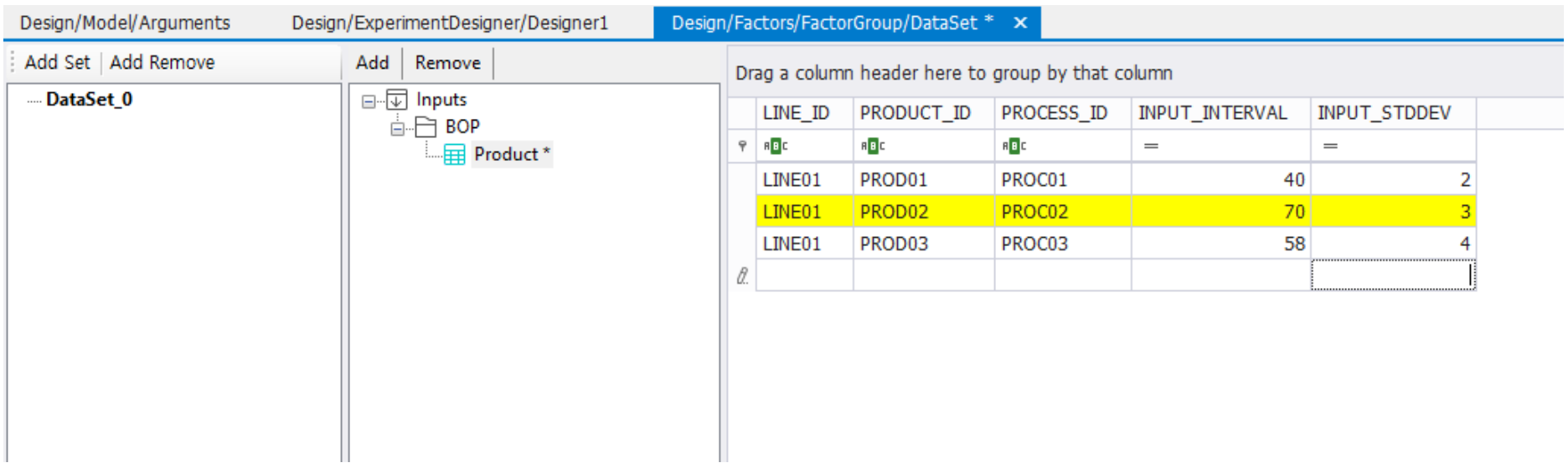
# 유형별 Factor 등록 절차 및 예

## DataSet 기능

1. Factor 편집화면의 좌측 트리상단의 **[Add Set]** 명령을 실행합니다.
2. 생성된 Set의 오른쪽 트리 상단에 Input 노드가 생성됩니다.
3. 변경하고 싶은 테이블이 존재한다면 Input 노드위에 **[Add]**를 누르면 Input Table 선택창이 열립니다.




4. Input Table 선택창에서 변경대상 테이블을 선택합니다. Input Table 중 Primary Key 가 설정된 테이블이 표시됩니다. 만일 선택을 위한 트리에 테이블이 표시되지 않는다면 Base Model 에 변경이 필요한 Input Table 에 Key를 설정해야 합니다.
5. 테이블이 선택되면 오른쪽 그리드에 Base Model 의 입력데이터가 로드됩니다.
6. 데이터 중 변경이 필요한 항목의 데이터를 수정합니다.
7. 수정을 한 항목은 행이 노란색으로 변경되어 수정된 데이터가 표시됩니다.
8. 수정된 값을 저장합니다.




DataSet 유형의 Set 입력 예시

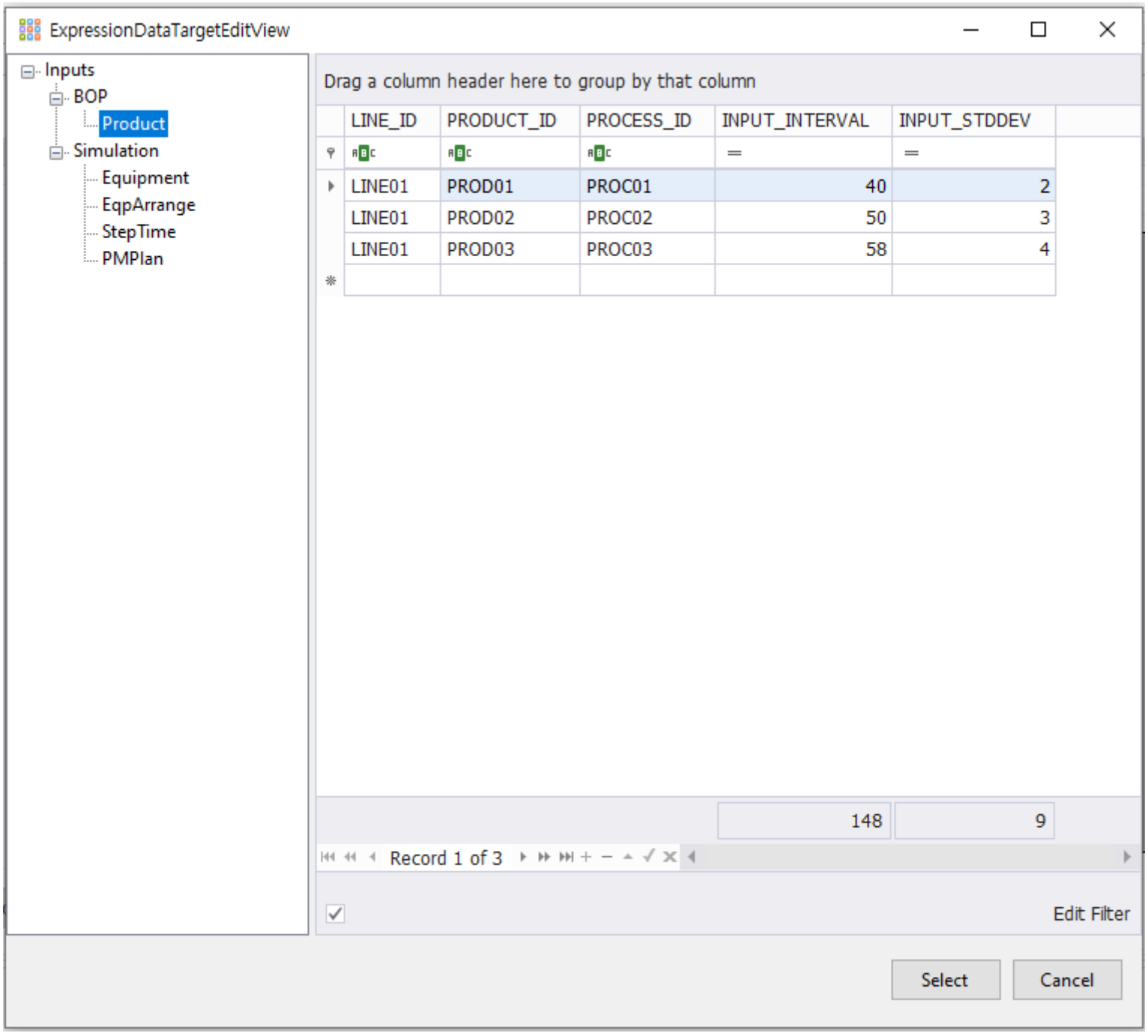
상기 예시에서는 모델의 제품(Product) 데이터 테이블에서 "PROD02" 번 제품의 INPUT_INTERBAL 컬럼 값을 70으로 변경한 테이블을 Set으로 정의한 예시입니다.

동일한 방식으로 Input 노드가 있는 트리에 변경이 필요한 추가적인 Input Table 을 등록하고 값을 변경할 수 있습니다. 이 경우 해당 Set의 변경된 모든 정보로 정의되는 하나의 Set(Level)이 됩니다. Set 유형의 경우 Set으로 설정된 Level들을 포함하고 있으며 각 Level별로 실행 케이스가 생성 가능합니다.

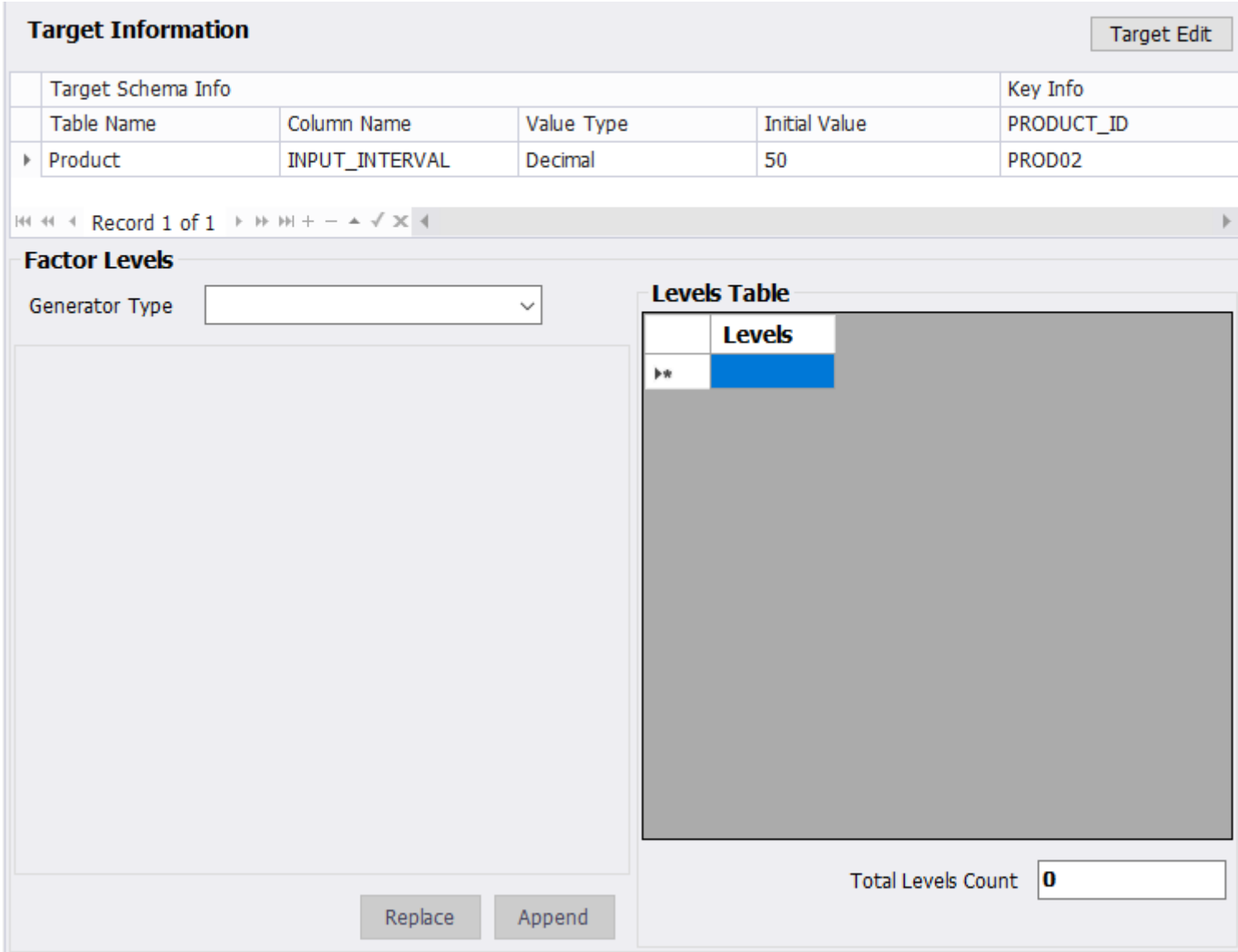
## Data 기능

1. Factor 편집화면의 좌측 상단의 **[Target Edit]** 명령을 실행합니다.

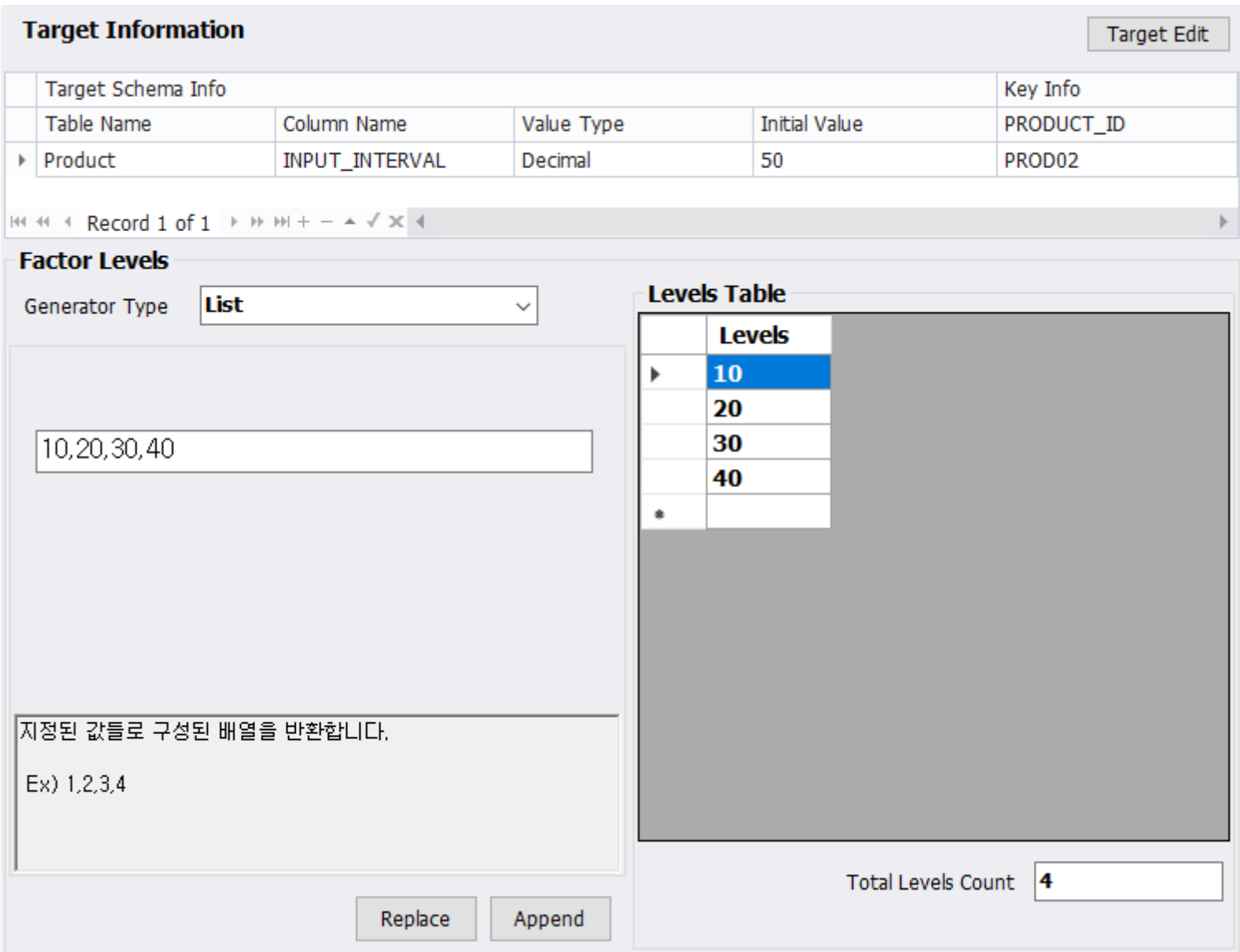
2. Input table 선택창에서 변경대상 테이블을 선택합니다. Input table 중 primary key 가 설정된 table 표시됩니다. 만일 선택을 위한 트리에 값이 노드가 표시되지 않는다면 Base Model 에 변경이 필요한 Input Table 에 Key를 설정해야 합니다.
3. 테이블이 선택되면 오른쪽 화면에 Factor의 Target을 설정할 수 있는 창이 열립니다.




4. Target으로 지정하고 싶은 Table내의 값을 선택한 후에 **[Select]** 명령을 실행합니다.
5. Target된 값들의 정보가 Target Table로 표시가 됩니다.



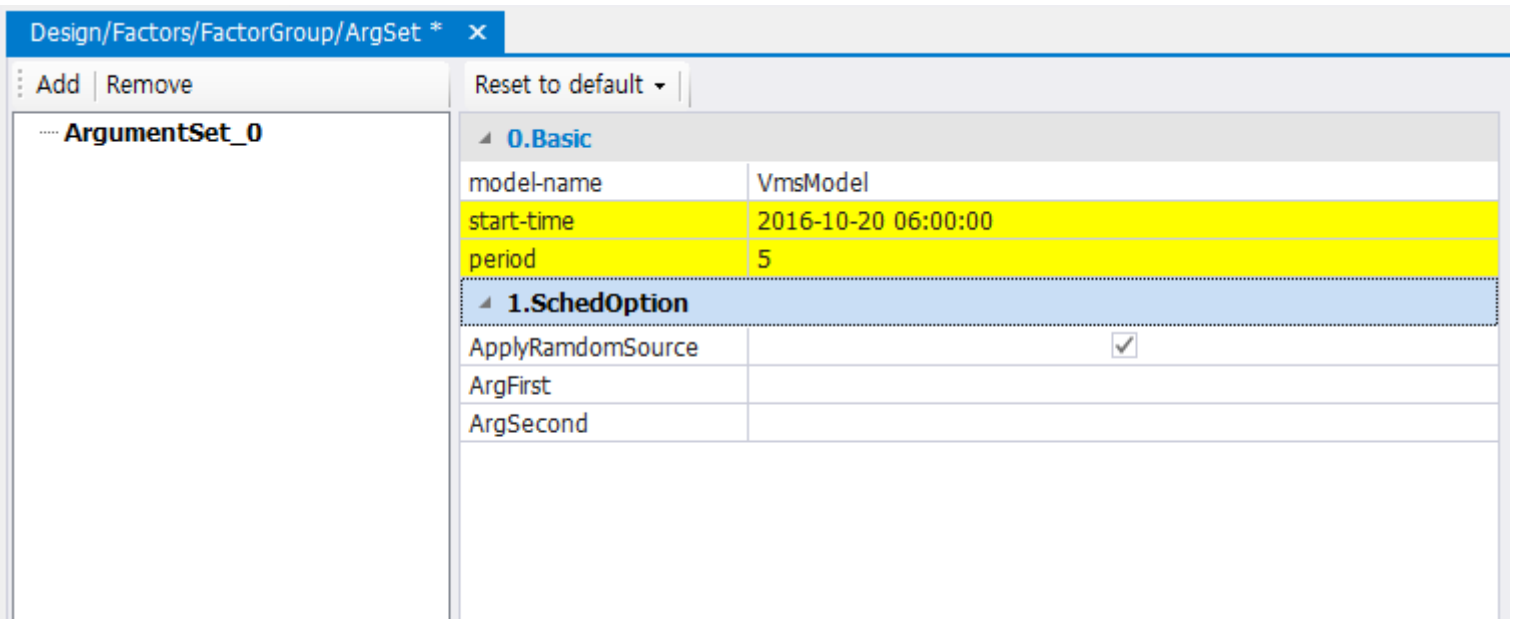

6. 해당 Target의 Type에 따라 Generator의 종류가 달라집니다.
   a. 정수 & 실수 : `List`, `Range`, `DistributionSampling`
   b. 논리연산(True & False만 존재함)
   c. 날짜 : `List`, `IntervalTime`
   d. 문자열 : `List`
7. Generator Type을 선택해서 원하는 Level들을 생성합니다. Levels Genarator의 상세한 사용법은 여기를 참조합니다.



상기 예제에서는 모델의 제품(Product) 데이터 테이블의 값에 실험하고 싶은 값들을 정의하는 예제입니다. `List` Generator Type을 사용하여 Product 테이블의 제품 "PROD01" 의 INPUT_INTERVAL 컬럼 값을 10, 20, 30, 40으로 정의합니다. 이러한 방식으로 정의하는 경우 하나의 Factor를 사용하여 다수의 케이스를 만들 수 있습니다. 예제의 경우 총 4개의 케이스를 생성합니다. Data 유형의 Factor는 이와 같이 하나의 Factor를 통해 다수의 테스트 케이스를 정의할 수 있습니다.

## ArgumentSet 기능

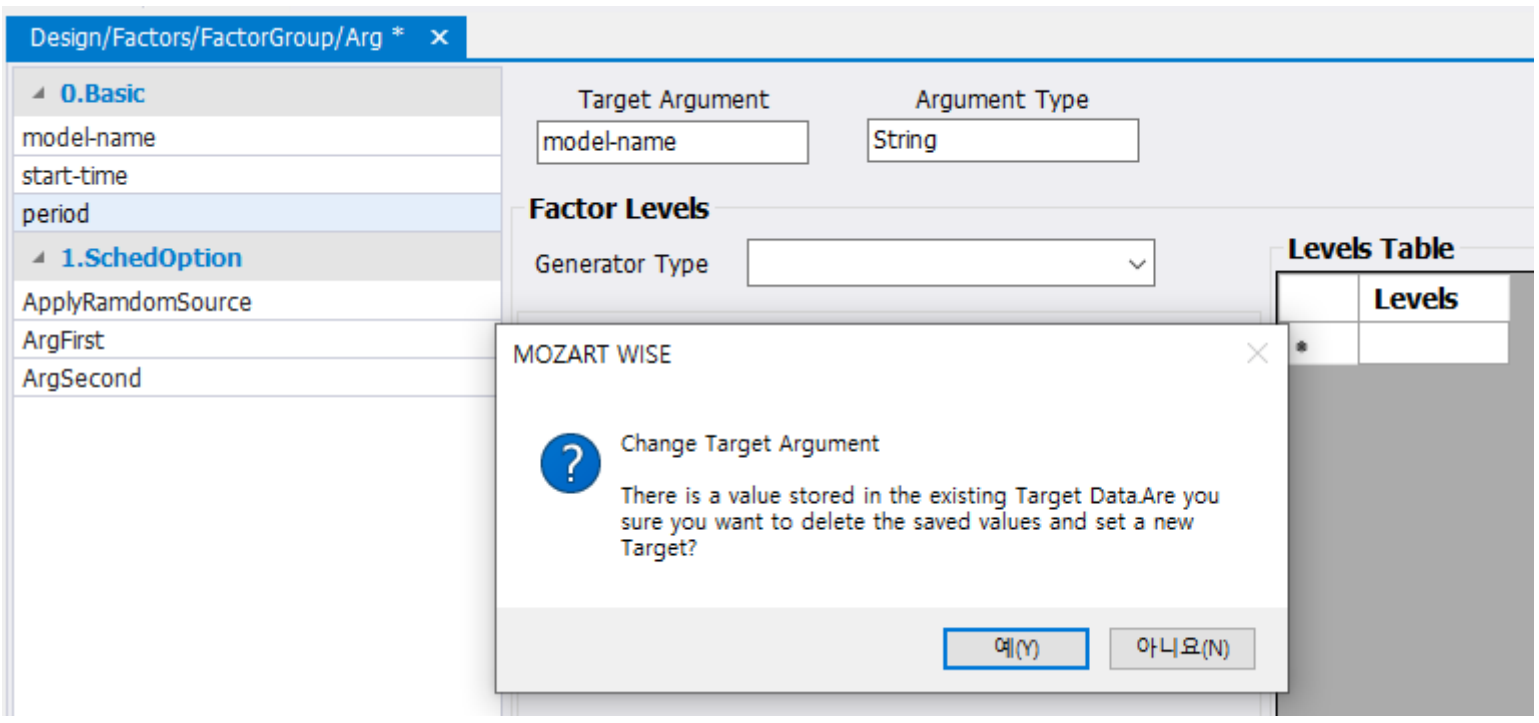
1. Factor 편집화면의 좌측 트리상단의 **[Add Set]** 명령을 실행합니다.
2. 생성된 Set의 오른쪽에 Input Argument 리스트가 표시된 입력 창이 나타납니다
3. Input Argument 리스트가 표시된 입력 창에서 필요한 Argument 를 수정합니다.
4. 수정된 값은 노란색으로 변경여부가 표시됩니다.
5. 수정 후 저장합니다.




상기 예제는 Input Argument 중 두 개의 Argument 의 값을 변경한 것을 하나의 케이스로 등록한 예시입니다. 예제에서는 하나의 케이스를 정의할 때 두 개의 Argument 값을 변경하였습니다.

## Argument 기능

1. Input Argument 리스트가 표시된 왼쪽 창에서 변경대상 Argument 를 선택합니다. 처음 생성하면 가장 맨위 Argument로 설정되어 있으며, 원하는 Argument를 선택하면 아래와 같이 Target을 변경할 것인지에 대한 알림을 확인할 수 있습니다. **[예]**를 누르시면 오른쪽 맨위에 **[Target Argument]**와 **[Argument Type]** 아래의 텍스트박스에 설정된 값을 통해서 해당 Argument가 설정됨을 확인하실 수 있습니다.



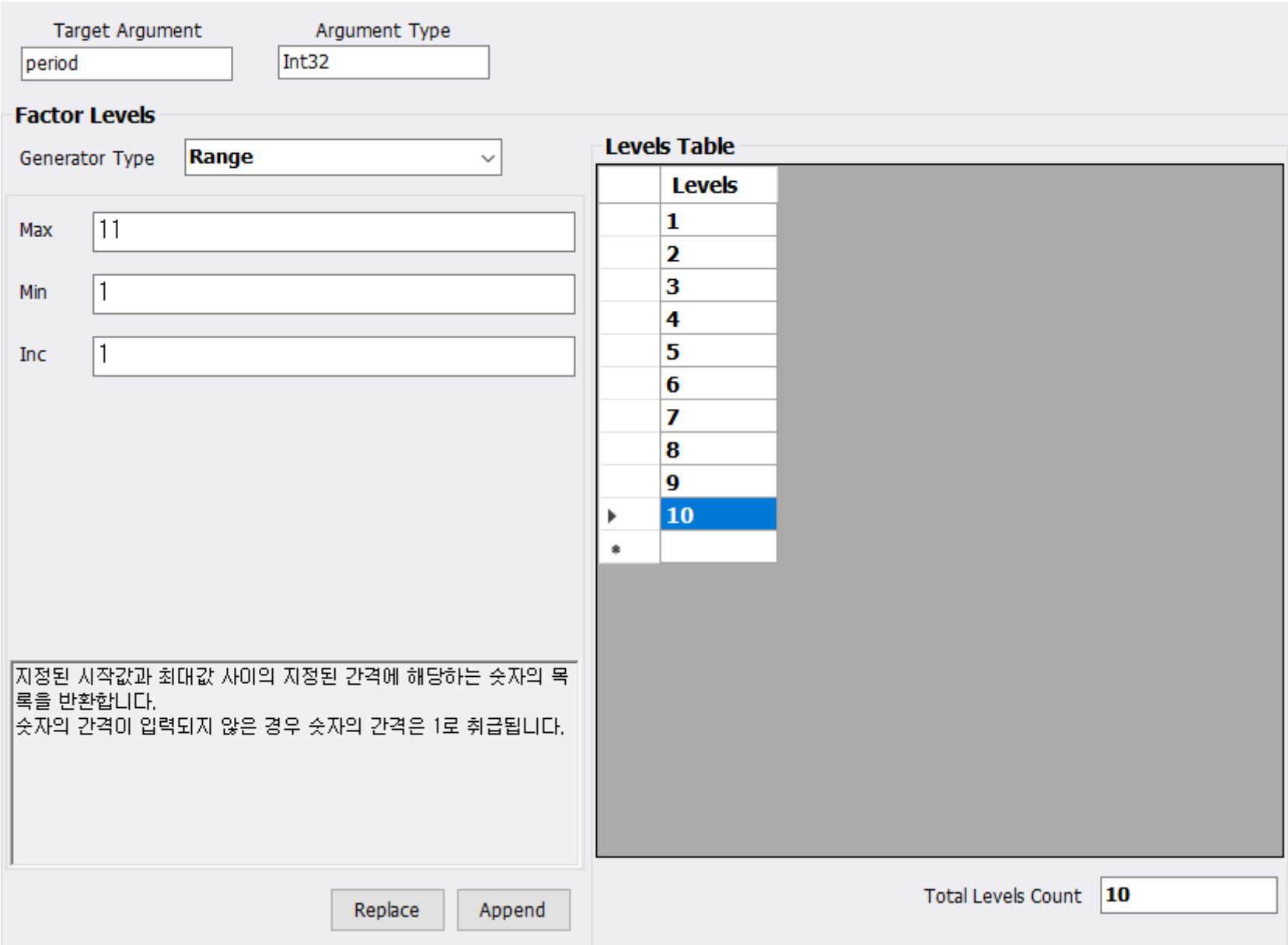

2. 설정된 Argument의 Levels를 설정해줍니다.
   a. 해당 Target의 Type에 따라 Generator의 종류가 달라집니다.
      i. 정수 & 실수 : `List`, `Range`, `DistributionSampling`

ii. 논리연산(True & False만 존재함)

iii. 날짜 : `List`, `IntervalTime`

iv. 문자열 : `List`

b. Generator Type을 선택해서 원하는 Level들을 생성합니다. Levels Genarator의 상세한 사용법은 여기를 참조합니다.



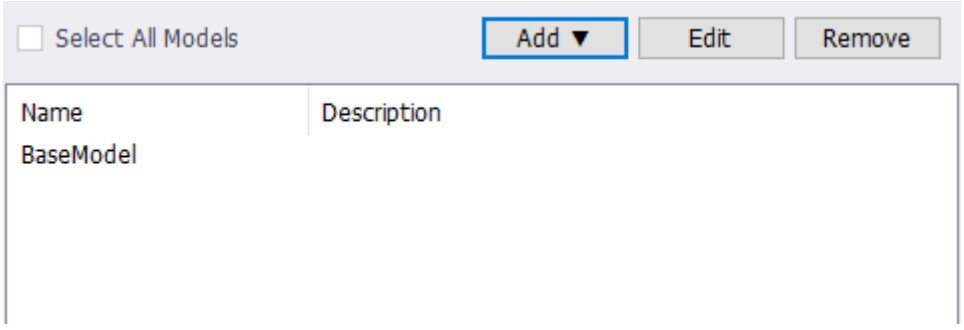

3. 수정 후 저장합니다.

위 예시는 Input Argument 중 Period (계획기간) 를 1일에서 10일까지 하루 씩 증가시키면서 모델을 실행시키도록 Levels를 추가하는 예시입니다. KPI 를 수행시간이나 제품 출하수량 등으로 설정하는 경우 계획기간 증가에 따른 수행시간의 관계 혹은 생산량의 관계 등을 테스트할 수 있습니다.

## Model 기능

📌 Model Factor의 경우 초기 생성시 BaseModel이 기본으로 생성되어 있습니다.

1. Factor의 Set로 사용할 모델을 선택합니다.
   - Add : Factor의 Set을 추가합니다.
     - Add Model : 로컬 컴퓨터에 저장된 Model을 Factor의 Set으로 지정합니다.
     - Add Folders : 로컬 컴퓨터에 저장된 Model들이 들어있는 폴더들을 Factor의 Set들로 지정합니다.
     - Add Zips : 로컬 컴퓨터에 저장된 Model들이 들어있는 압축파일들을 Factor의 Set들로 지정합니다.
     - Add Server Zips : 서버에 저장된 Model들이 들어있는 압축파일들을 Factor의 Set들로 지정합니다.
   - Edit : 선택한 Model의 이름과 Description을 편집합니다. (BaseModel은 편집이 안됩니다.)
   - Remove : 선택한 Model들을 삭제합니다. (BaseModel은 삭제가 안됩니다.)
     - Select All Models 기능을 통해 모든 Model을 한번에 선택해서 삭제 가능합니다.

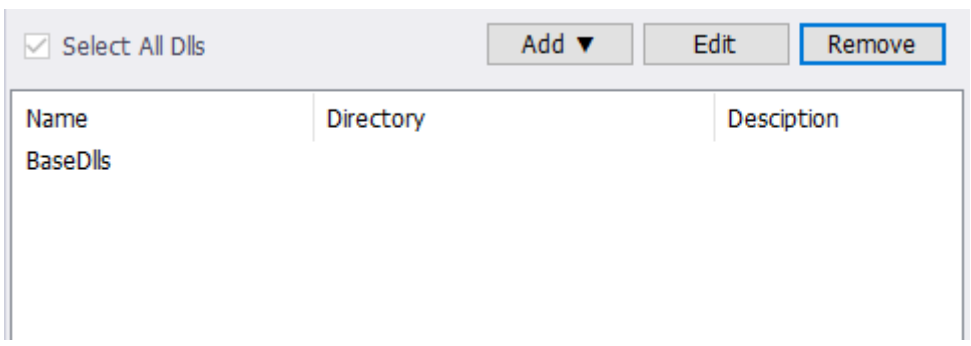



2. 불러온 모델은 Argument와 Input Data를 가지고 있으며, 해당 Data는 직접 수정이 불가능합니다.

## Dlls 기능

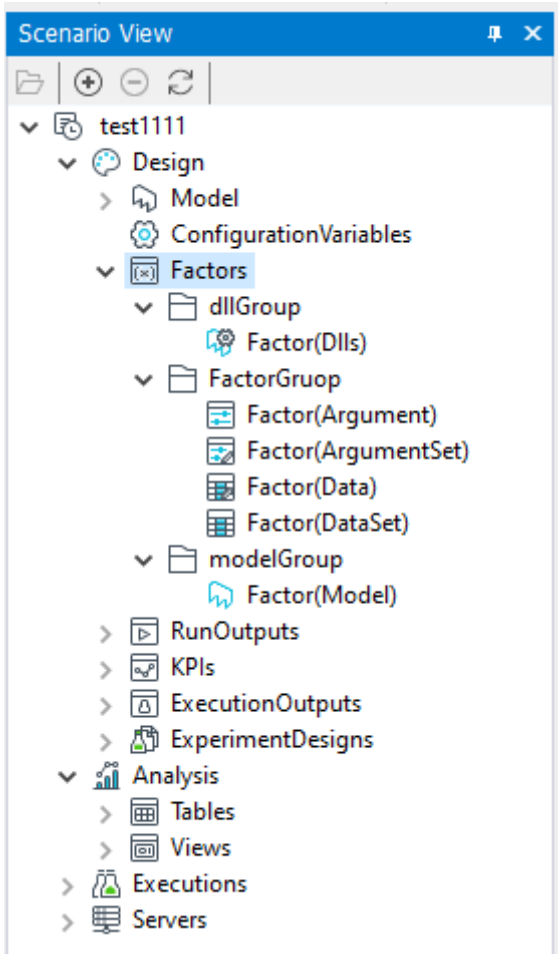
📌 Dlls Factor의 경우 초기 생성시 BaseDlls가 기본으로 생성되어 있습니다.

1. Factor의 Set으로 사용할 Engine Dlls 폴더를 선택합니다.
   - Add : Factor의 Set을 추가합니다.
     - Add Folder: Engine Dlls 폴더를 Factor의 Set으로 추가합니다.
     - Add Folders : Engine Dlls 폴더를 Factor의 Set들로 추가합니다.
   - Edit : 선택한 Set의 이름과 Path, Description을 편집합니다. (BaseDlls은 편집이 안됩니다.)
   - Remove : 선택한 Dlls들을 삭제합니다. (BaseDlls은 삭제가 안됩니다.)
     - Select All Dlls 기능을 통해 모든 Dlls를 한번에 선택해서 삭제 가능합니다.

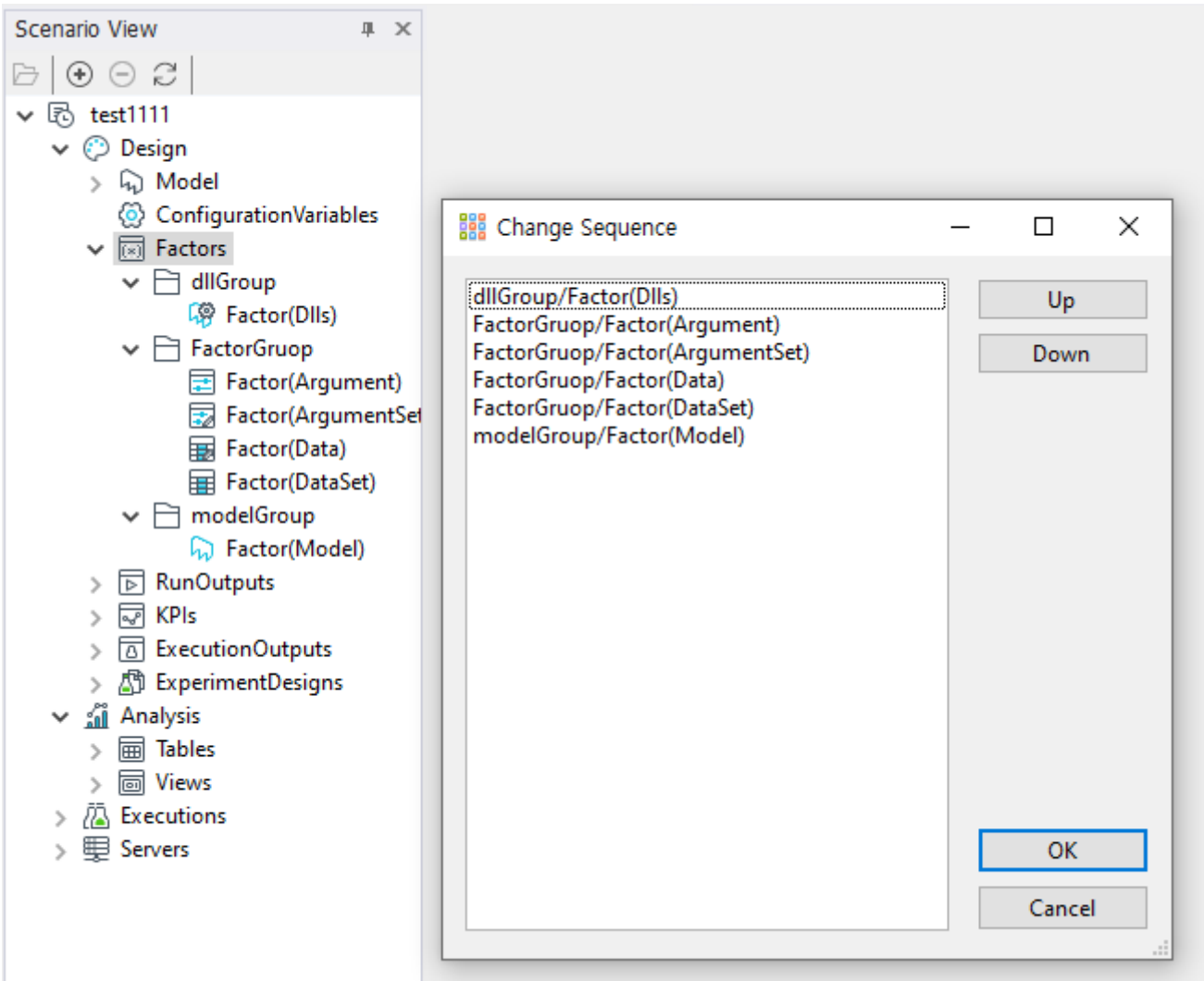



## Change Sequence

각 Factor Group내의 Factor의 계산 순서를 변경하는 기능입니다.




1. Factor 계산 순서 변경



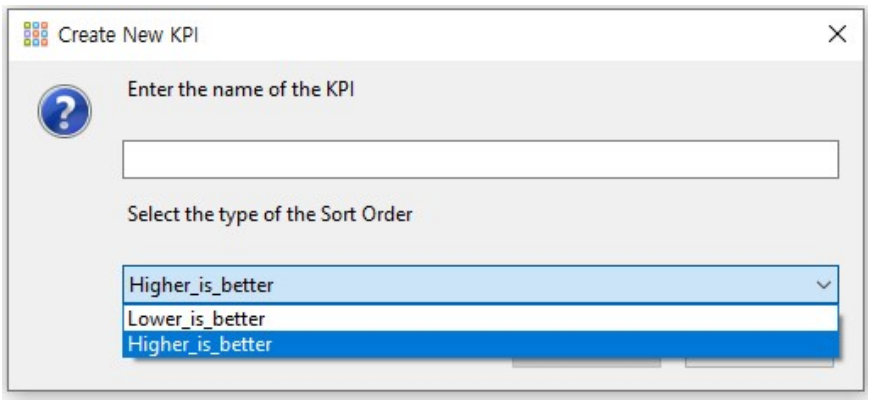

# KPIs

KPIs 는 시뮬레이션 결과를 평가하기 위한 Index들의 집합입니다. Factor의 조합에 따라 모델을 실행하여 시뮬레이션 결과를 평가하는 것을 목적으로 합니다. 이에 좋은 결과를 산출하는 Factor들의 조합을 찾거나, Factor 값과 KPI의 관계를 확인하기 위해 관심 있는 항목을 KPI로 정의하기 위해 사용할 수 있습니다.

## KPIs 등록 절차

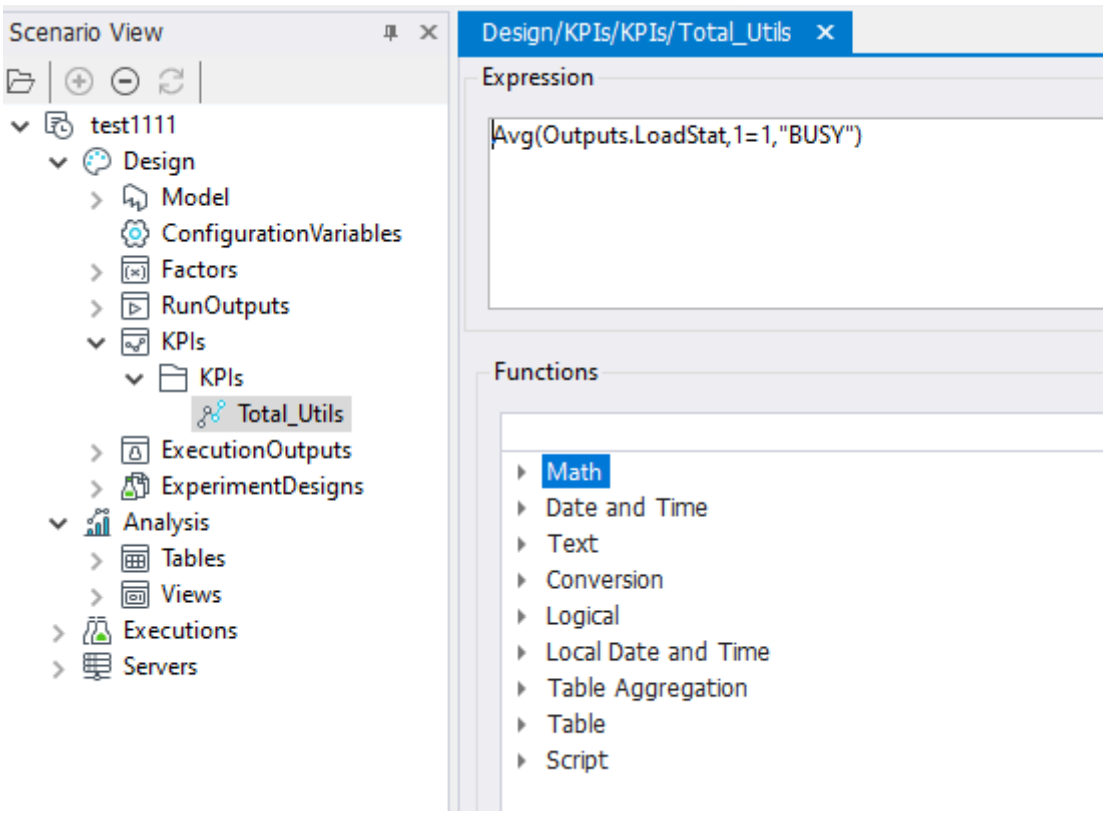
1. **[Design]** > **[KPIs]** 노드에서 [Add] 버튼을 눌러 KPI 그룹을 추가합니다.
2. 추가된 KPI 그룹 노드에서 [Add] 버튼을 눌러 KPI 를 추가합니다.

3. KPI 명칭을 입력합니다.
4. KPI 값이 높을 수록 좋은 값인지, 낮을 수록 좋은 값인지를 선택합니다 (Higher_is_better, Lower_is_better). KPI 관련 결과 조회시점에 KPI  소팅에 활용하기 위해 필요한 속성입니다.



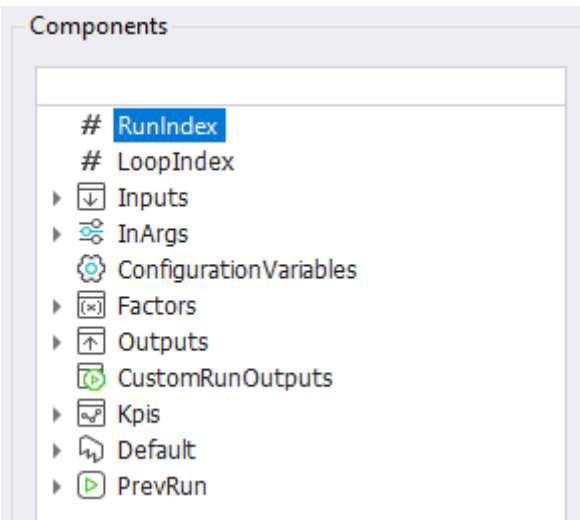

KPI 선정 화면

5. KPI 항목이 추가된 후  KPI 값을 산출하기 위한 Expression 을 정의합니다. Expression 의 결과값은 Scalar Value 가 나오도록 작성해야 합니다.
6. Expression 작성 후 저장합니다.




KPI 작성 예시

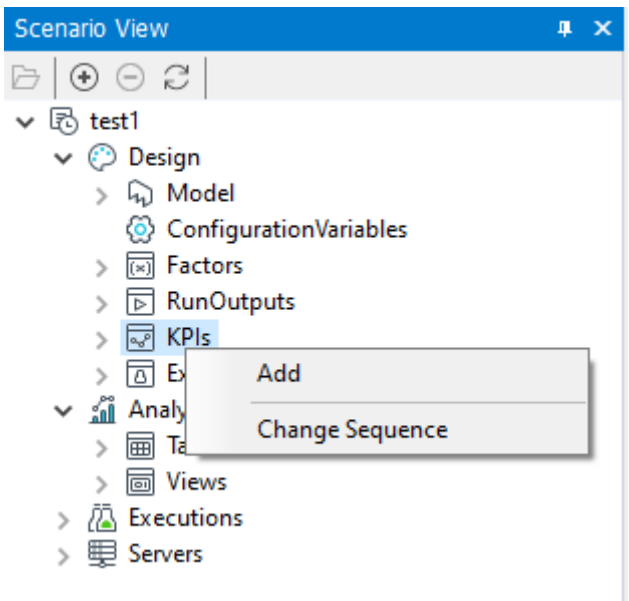
상기 예시는 Model 의 Output 중 LoadStat 이라는 Output 테이블의 `BUSY` 컬럼 값의 평균값으로 KPI 를 정의한 예시입니다. KPI 를 정의하기 위해서는 위와 같이 Expression 을 사용하거나 Linq Script 를 사용할 수 있습니다. 또한 Expression 에서 사용될 수 있는 데이터는 Expression Editor 의 우측 Components 트리에 포함된 모든 데이터를 포함합니다. 각 요소에 대한 설명은 **Expression 편집기능**을 참조합니다.




Expression 편집시 사용할 수 있는 데이터 요소

### Change Sequence

각 KPI Group내의 KPI의 계산 순서를 변경하는 기능입니다.



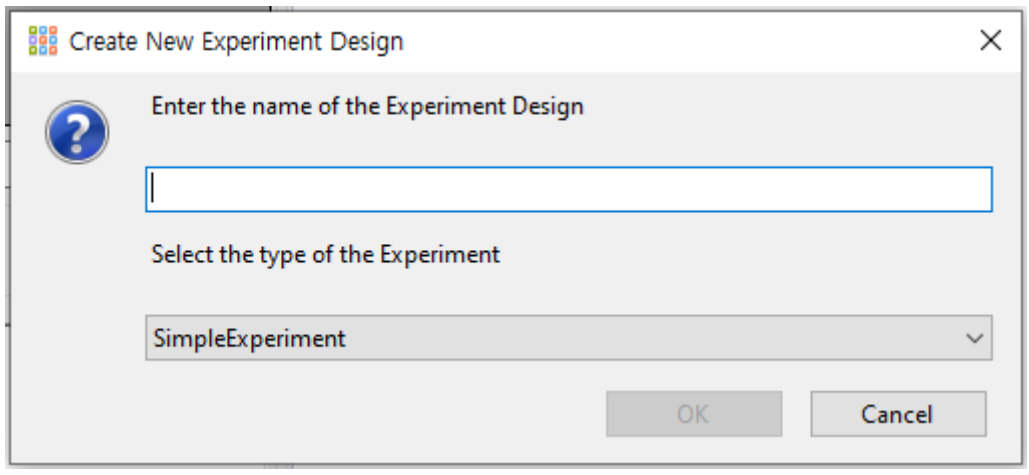

---

# Experiment Designs

Experiment Designs에서는 실험할 Factor와 KPI를 선택해서 실험계획을 구성할 수 있습니다. 실험계획은 기본적인 실험부터 복잡한 실험까지 원하는 실험 유형을 선택해서 앞서 정의한 Factor들로 구성할 수 있습니다.

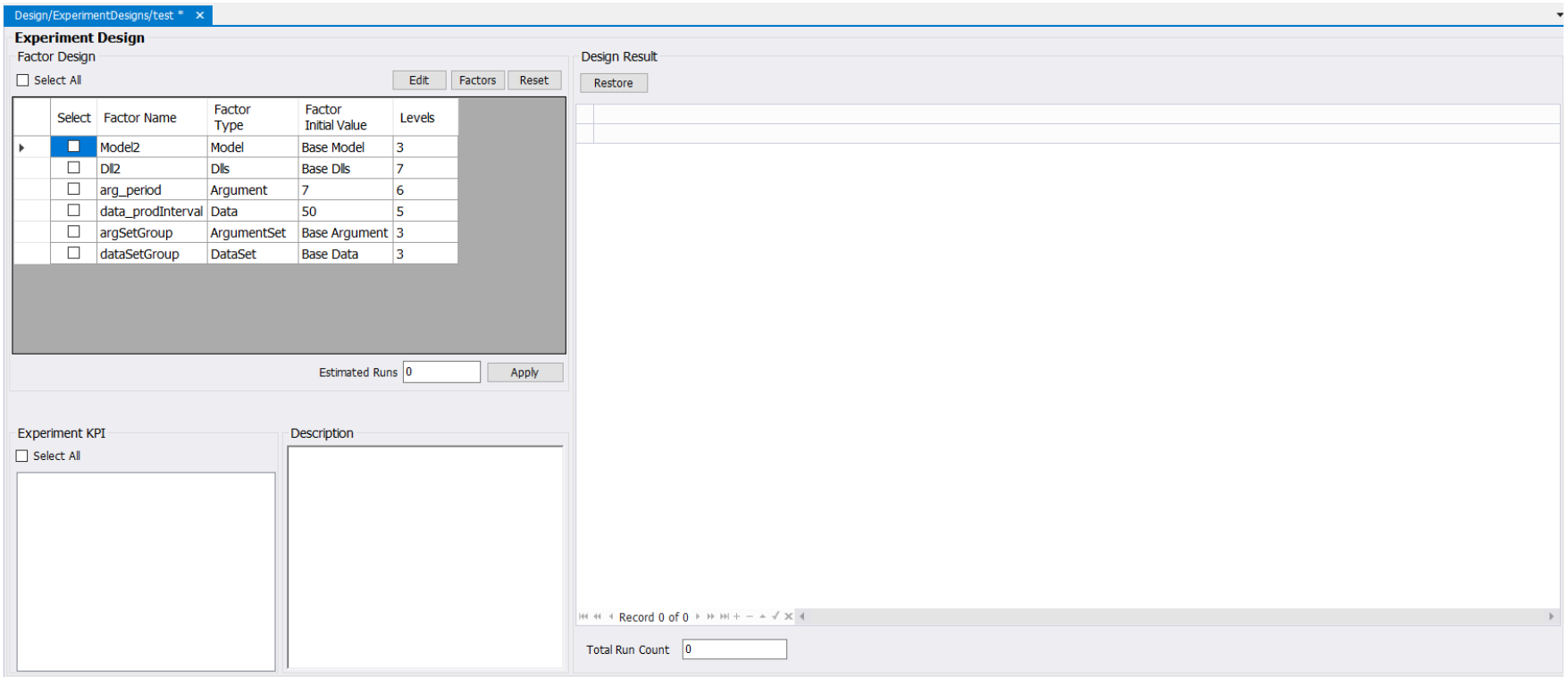
## Experiment Design 등록 절차

1. **[Design]>[Experiment Designs]** 노드에서 **[Add]** 버튼을 눌러 Experiment Design 를 추가합니다.
2. Experiment Design 명칭을 입력하고, 아래의 실험 유형 중 한가지를 선택합니다.



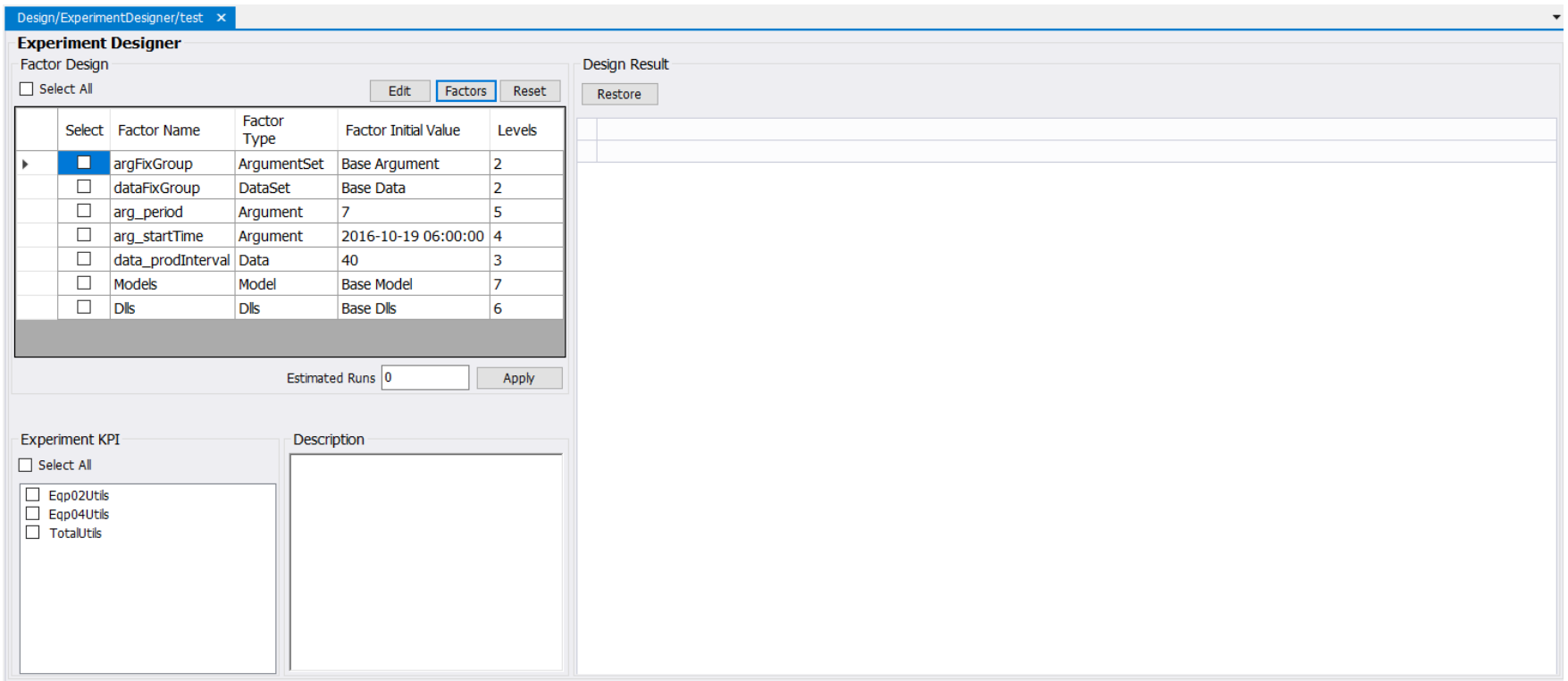

새로운 Experiment Design를 생성하는 화면

- `SimpleExperiment` : 선택한 Factor들로 Full Factorial로 실험을 구성합니다. 간단한 What-If Simulation 실험 설계에 사용됩니다.




3. Experiment Design 항목이 추가된 후 실험하고자 하는 Factor와 결과로 보고 싶은 KPI를 선택합니다.

▼ SimpleExperiment

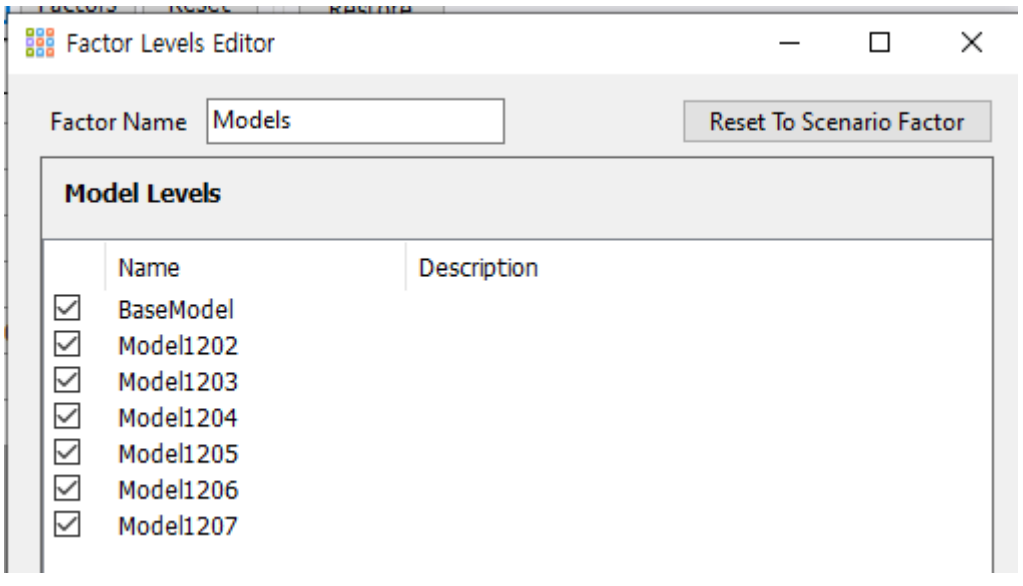



Simple Experiment Type의 Experiment Design 화면

- Factor Design

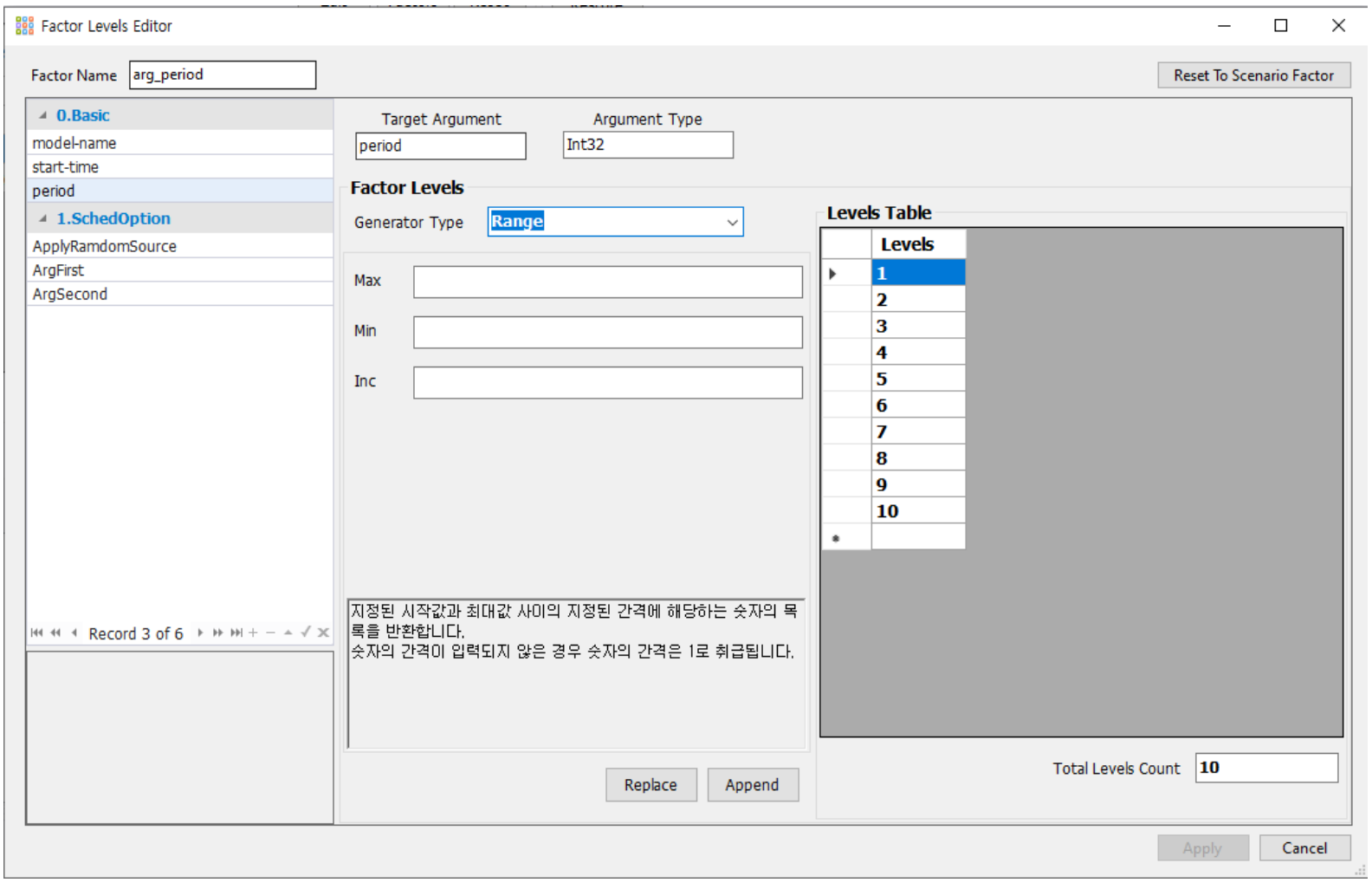
> 💡 Model과 Dll은 각 1개씩만 선택할 수 있습니다.

  - **[Select All]** : Experiment Design에 설정된 모든 Factor를 선택합니다.
  - **[Edit]** : 선택된 Factor의 Level을 변경하는 화면을 엽니다.
    - **[Reset To Scenario Factor]** : 해당 Factor의 Levels를 시나리오에 정의된 Factor의 Levels로 Reset 시킵니다.



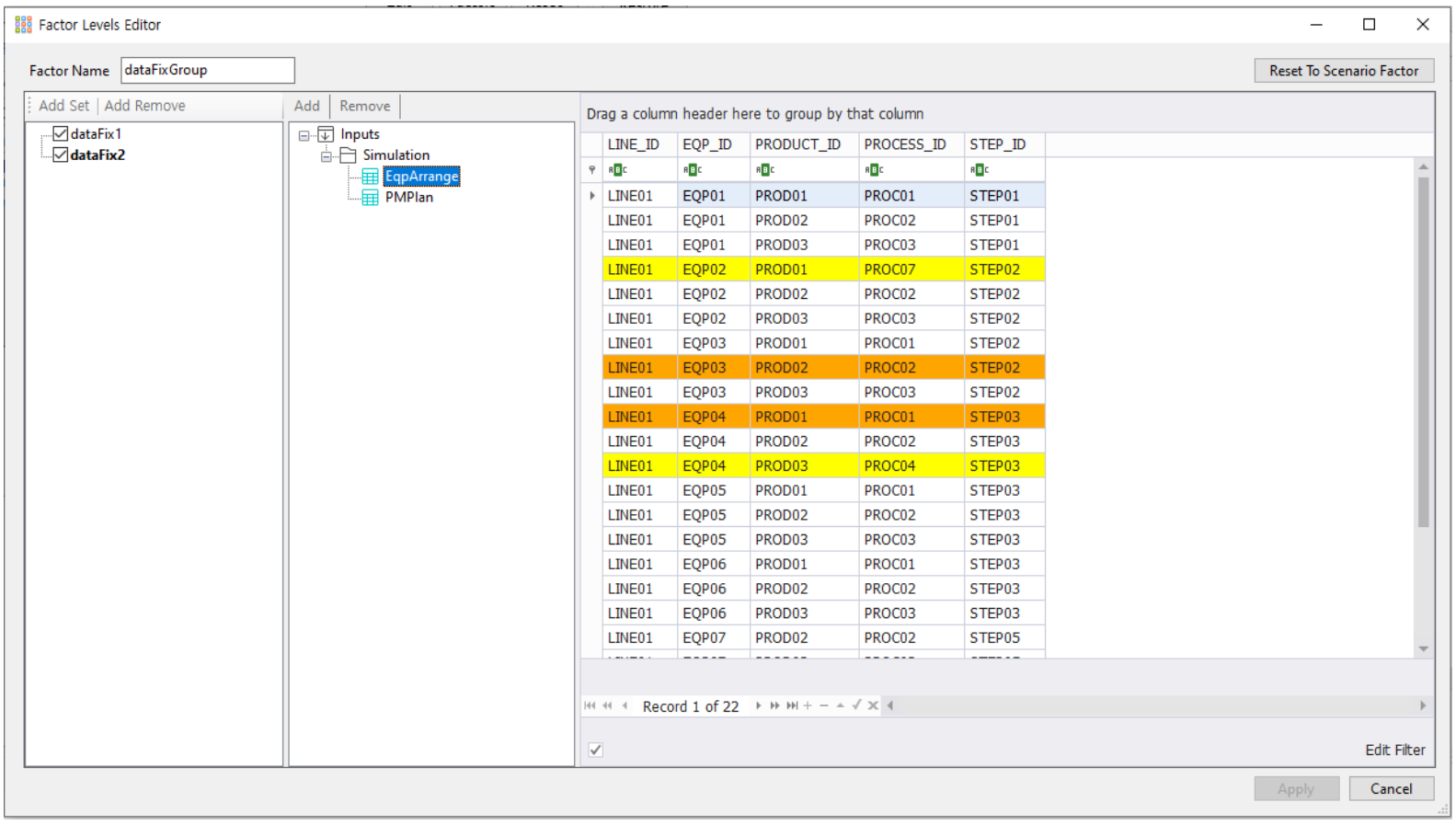

Factor Levels Editor 화면

- Data, Argument Type의 Factor들은 Generator 기능을 사용하여 Level값들을 추가하고 삭제할 수 있습니다. 다만, Target 정보는 변경이 불가능합니다.




Argument Type의 Factor Levels Editor 화면 예시

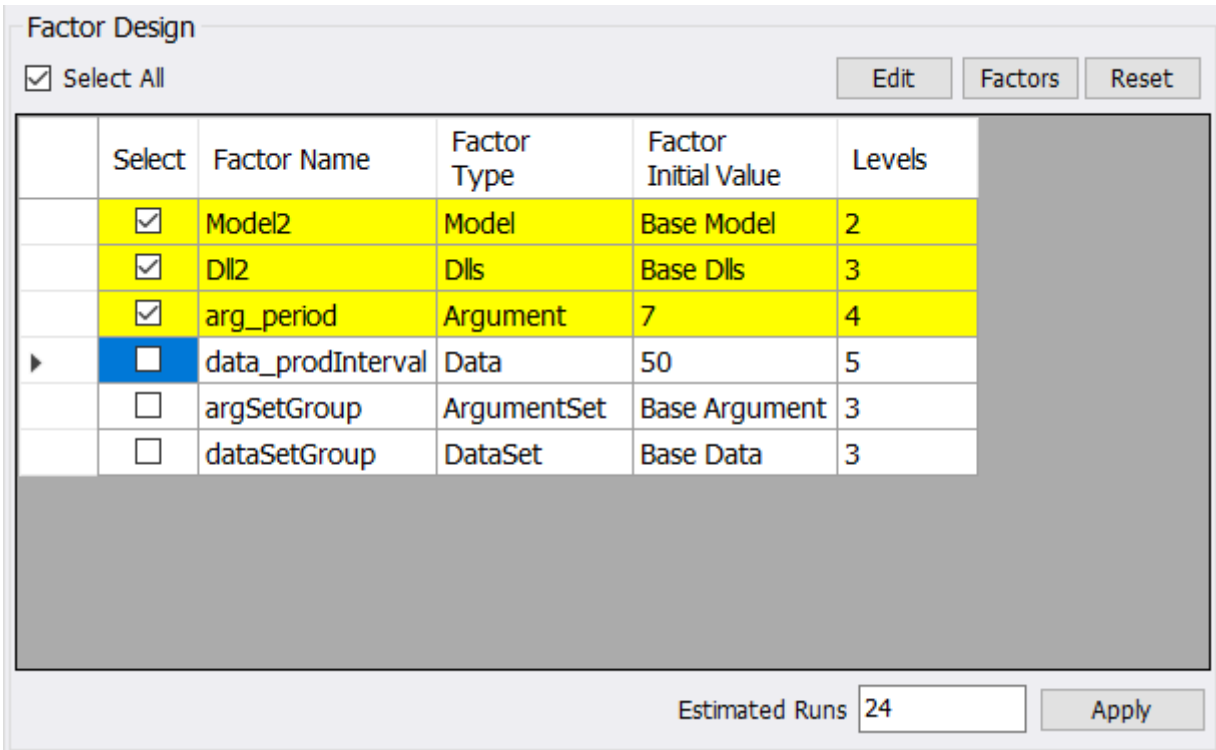
- DataSet, ArgumentSet, Model, Dlls의 경우에는 앞서 정의된 Level들을 선택할 수만 있습니다.



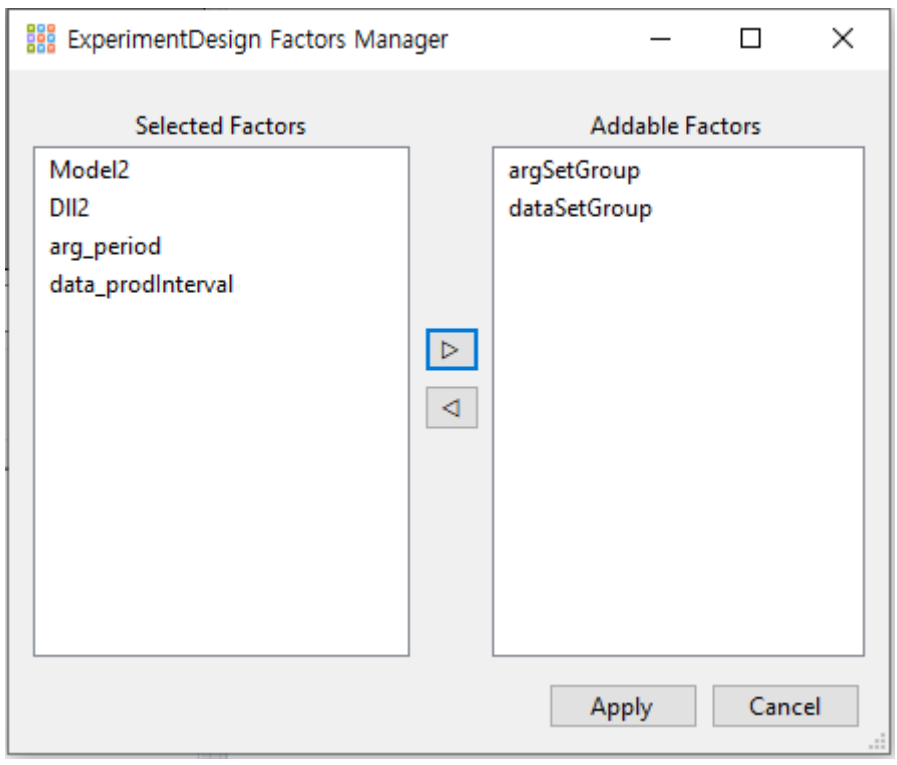

DataSet Type의 Factor Levels Editor 화면 예시

- Levels가 변경된 Factor의 경우 **노란색**으로 표시됩니다

> 💡 Levels가 변경되면 해당 Levels의 개수는 자동으로 Estimated Runs 계산에 반영됩니다.

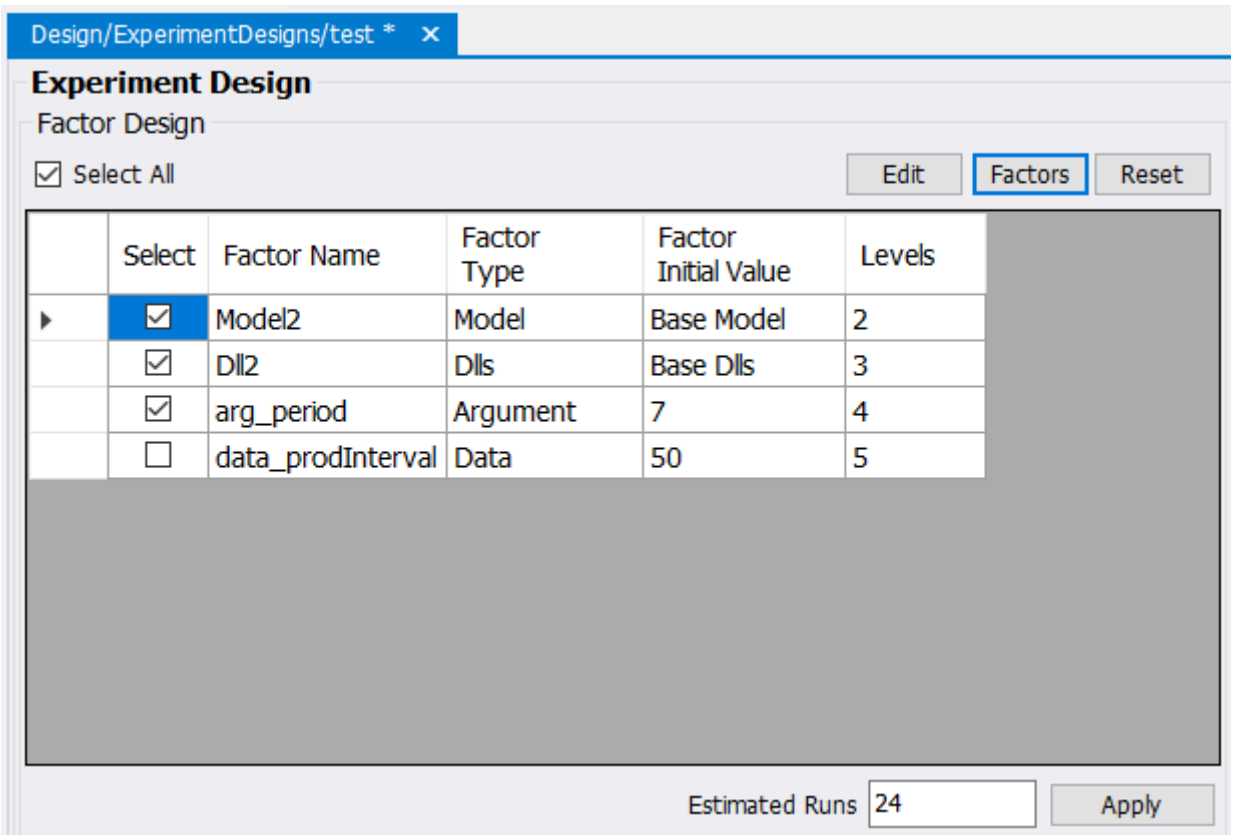



- **[Factors]** : 해당 Experiment Design에서 설계한 Factor를 선택하는 화면을 엽니다.
  - 실험설계에 사용하고자 하는 Factor만 선택할 수 있습니다. **[◁]**, **[▷]** 버튼을 사용해서 원하는 Factor를 추가 및 제외할 수 있습니다.

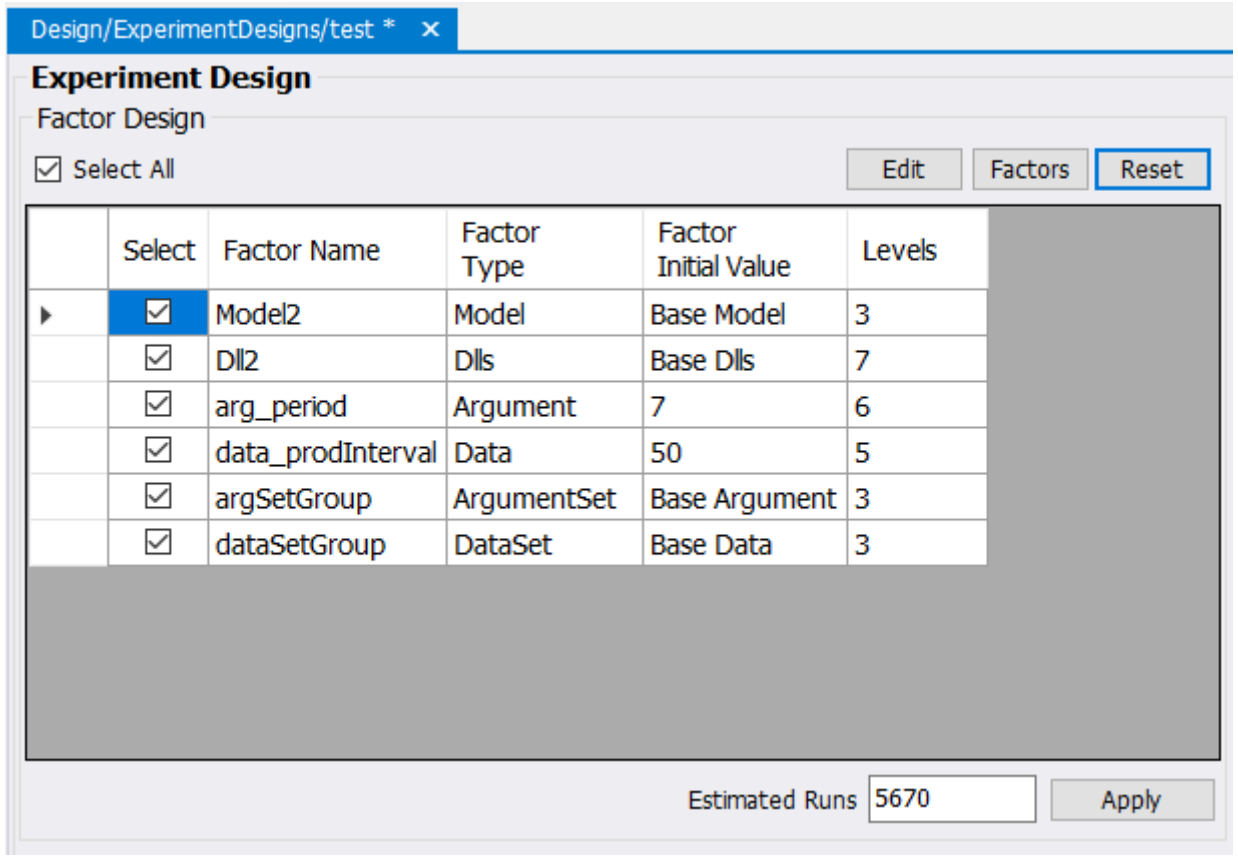



Factors Manager 화면

- 해당 Experiment Design 생성 이후에 시나리오에 생성한 Factor들은 Addable Factors에 존재합니다. 해당 Factor들을 추가해서 새롭게 정의한 Factor들을 해당 Experiment Design에서 설계할 수 있습니다.
- 선택한 Factors를 적용하기 위해서 **[Apply]** 버튼을 눌러 적용합니다.




- **[Reset]** : 시나리오에 정의된 Factor로 모두 Reset 시킵니다.

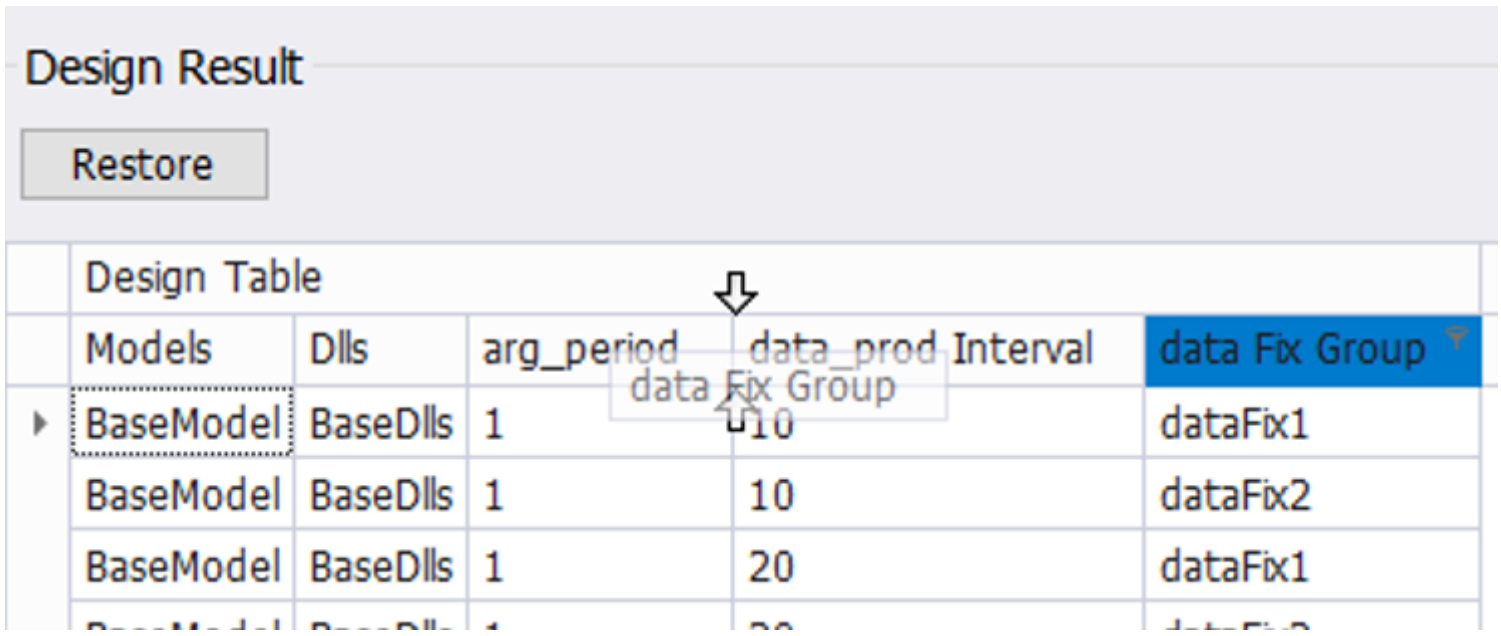



Reset 된 Factor Design 화면

- Estimated Runs : 선택한 Factor들의 Levels를 조합한 FullFactorial 실험설계시 생성되는 실험 개수의 예측치를 표시합니다.
- **[Apply]** : 선택된 Factor들로 FullFactorial 실험설계를 하여 Design Result에 실험을 표시합니다.

- KPI 선택 : 실험 결과로 보고 싶은 KPI를 선택합니다.
  - **[Select All]** : 모든 KPI를 선택합니다.
- Description : Experiment Design에 대한 설명을 적습니다.
- Design Result
  - 실험 순서 : Column의 순서를 이동하면 해당 순서대로 Factor가 적용됩니다.

> 💡 **Model**과 **Dlls** Factor의 경우는 가장 최우선 적용 순서로 순서 변동이 불가능합니다.




실험 순서 변경 예시

  - **[Restore]** : Column을 Grid내에서 Drag&Drop으로 삭제시에 원래대로 복구하는 기능
  - Remove Run : 실험하고 싶지 않는 Run(실험)을 클릭해서 Keyboard의 **[Delete]** 버튼을 눌러 삭제 하는 기능(다중 선택 가능)

Design Result

Restore

Design Table

| Models | Dlls | arg_period | data_prod Interval | data Fix Group |
|---|---|---|---|---|
| BaseModel | BaseDlls | 1 | 10 | dataFix1 |
| BaseModel | BaseDlls | 1 | 10 | dataFix2 |
| BaseModel | BaseDlls | 1 | 20 | dataFix1 |
| BaseModel | BaseDlls | 1 | 20 | dataFix2 |
| BaseModel | BaseDlls | 1 | 30 | dataFix1 |
| BaseModel | BaseDlls | 1 | 30 | dataFix2 |
| BaseModel | BaseDlls | 2 | 10 | dataFix1 |
| BaseModel | BaseDlls | 2 | 10 | dataFix2 |
| BaseModel | BaseDlls | 2 | 20 | dataFix1 |
| BaseModel | BaseDlls | 2 | 20 | dataFix2 |
| BaseModel | BaseDlls | 2 | 30 | dataFix1 |
| BaseModel | BaseDlls | 2 | 30 | dataFix2 |
| BaseModel | BaseDlls | 3 | 10 | dataFix1 |
| BaseModel | BaseDlls | 3 | 10 | dataFix2 |
| BaseModel | BaseDlls | 3 | 20 | dataFix1 |
| BaseModel | BaseDlls | 3 | 20 | dataFix2 |
| BaseModel | BaseDlls | 3 | 30 | dataFix1 |
| BaseModel | BaseDlls | 3 | 30 | dataFix2 |
| BaseModel | 116 | 1 | 10 | dataFix1 |
| BaseModel | 116 | 1 | 10 | dataFix2 |
| BaseModel | 116 | 1 | 20 | dataFix1 |
| BaseModel | 116 | 1 | 20 | dataFix2 |
| BaseModel | 116 | 1 | 30 | dataFix1 |
| BaseModel | 116 | 1 | 30 | dataFix2 |

Record 12 of 216

Total Run Count 216

Design Result

Restore

Design Table

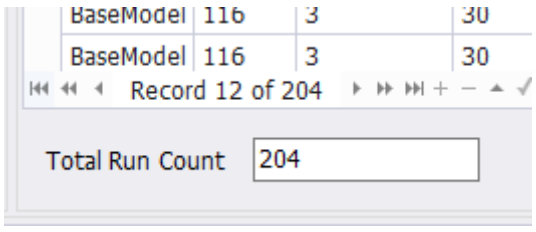
| Models | Dlls | arg_period | data_prod Interval | data Fix Group |
|---|---|---|---|---|
| BaseModel | BaseDlls | 3 | 10 | dataFix1 |
| BaseModel | BaseDlls | 3 | 10 | dataFix2 |
| BaseModel | BaseDlls | 3 | 20 | dataFix1 |
| BaseModel | BaseDlls | 3 | 20 | dataFix2 |
| BaseModel | BaseDlls | 3 | 30 | dataFix1 |
| BaseModel | BaseDlls | 3 | 30 | dataFix2 |
| BaseModel | 116 | 1 | 10 | dataFix1 |
| BaseModel | 116 | 1 | 10 | dataFix2 |
| BaseModel | 116 | 1 | 20 | dataFix1 |
| BaseModel | 116 | 1 | 20 | dataFix2 |
| BaseModel | 116 | 1 | 30 | dataFix1 |
| BaseModel | 116 | 1 | 30 | dataFix2 |
| BaseModel | 116 | 2 | 10 | dataFix1 |
| BaseModel | 116 | 2 | 10 | dataFix2 |
| BaseModel | 116 | 2 | 20 | dataFix1 |
| BaseModel | 116 | 2 | 20 | dataFix2 |
| BaseModel | 116 | 2 | 30 | dataFix1 |
| BaseModel | 116 | 2 | 30 | dataFix2 |
| BaseModel | 116 | 3 | 10 | dataFix1 |
| BaseModel | 116 | 3 | 10 | dataFix2 |
| BaseModel | 116 | 3 | 20 | dataFix1 |
| BaseModel | 116 | 3 | 20 | dataFix2 |
| BaseModel | 116 | 3 | 30 | dataFix1 |
| BaseModel | 116 | 3 | 30 | dataFix2 |

Record 12 of 204

Total Run Count 204

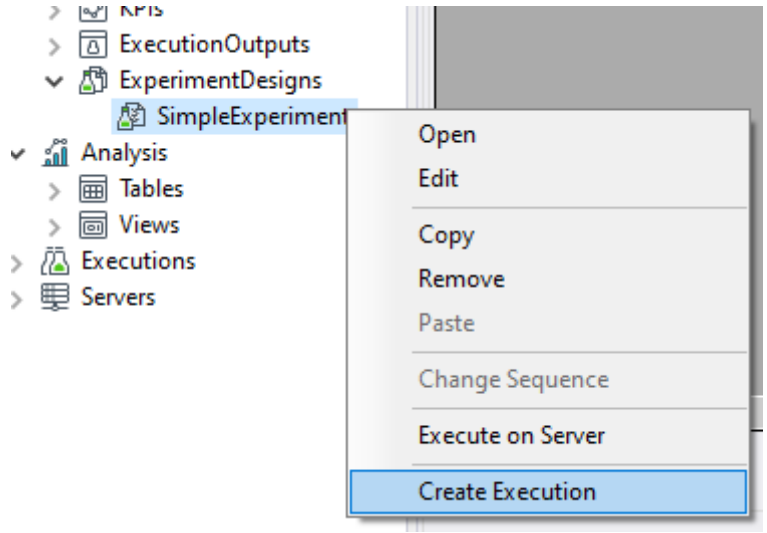
실험하기를 원하지 않는 Run(실험)들을 선택해서 삭제하는 화면 예시

- Total Run Count : Design Result의 총 실험 개수를 표시하는 기능



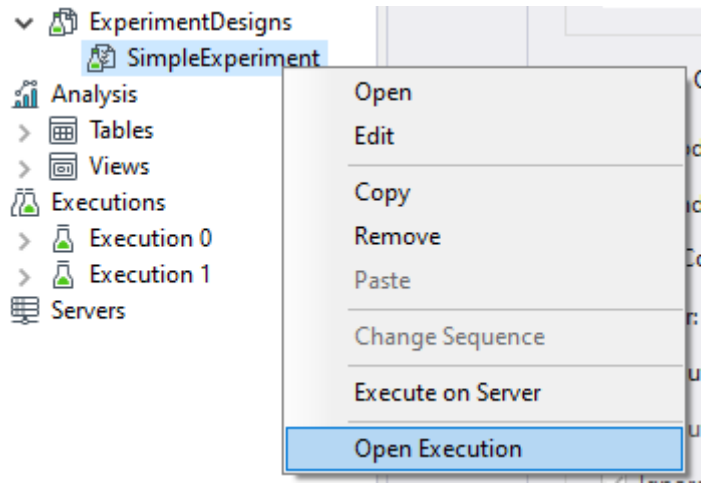

총 실험 개수

4. 설계가 완료된 Experiment Design을 실행시키는 Executor를 생성하기 위해서 해당 **[test]**를 선택하고 오른쪽 마우스 클릭을 합니다. 이를 통해 나오는 팝업창에서 **[Create Execution]**을 클릭하면 해당 Experiment Design을 실행시켜주는 Design Executor를 **[Exeuctions]** 아래에 생성합니다.

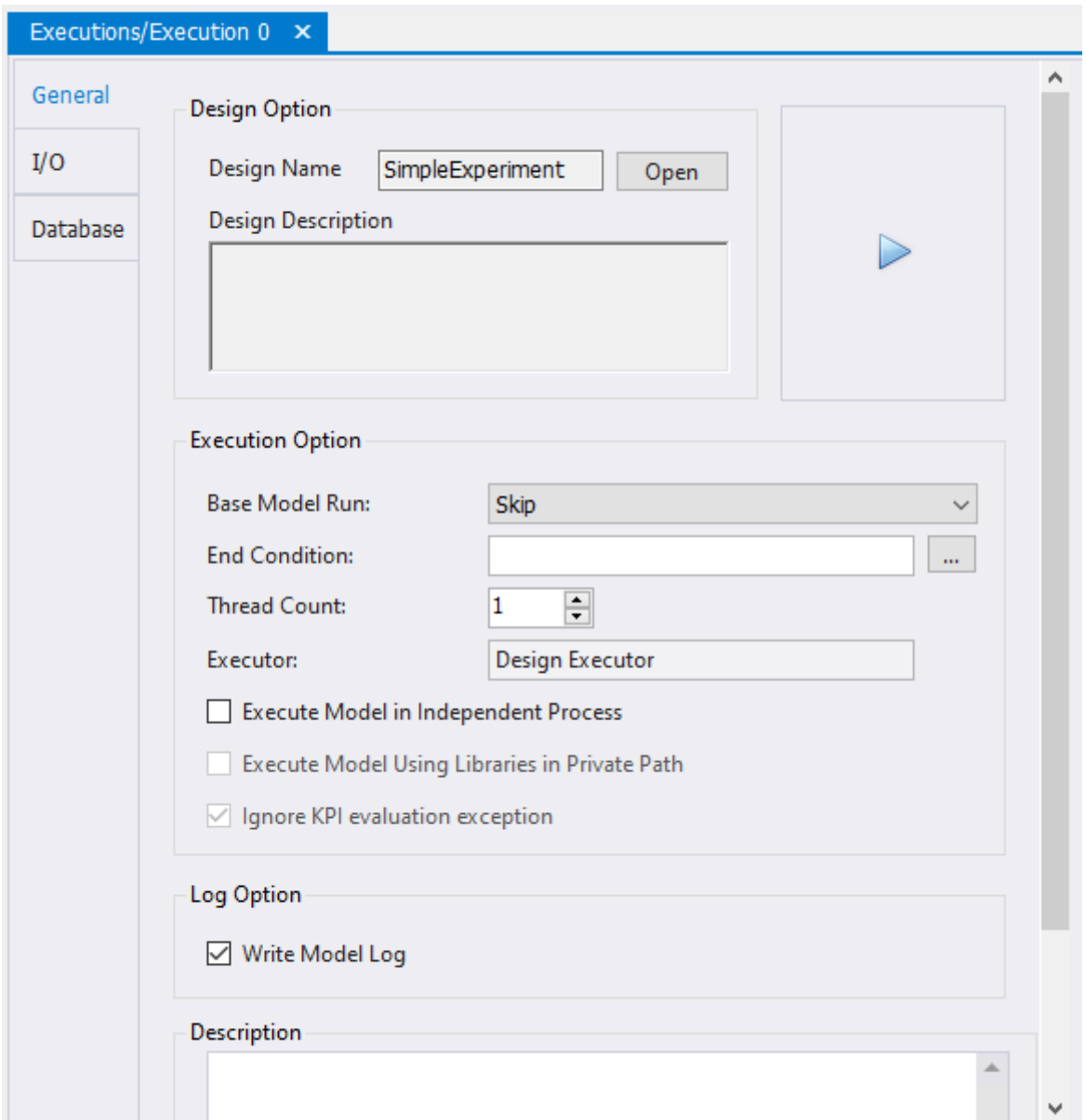



ExperimentDesign에서 Create Execution 메뉴

> 이미 연결된 Executor가 있는 경우에는 **[Open Execution]**으로 표시됩니다.

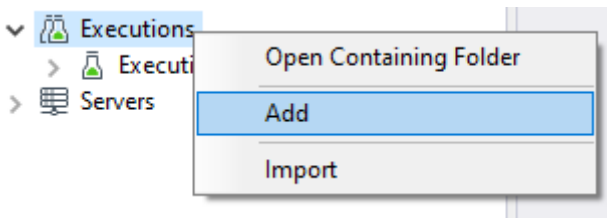


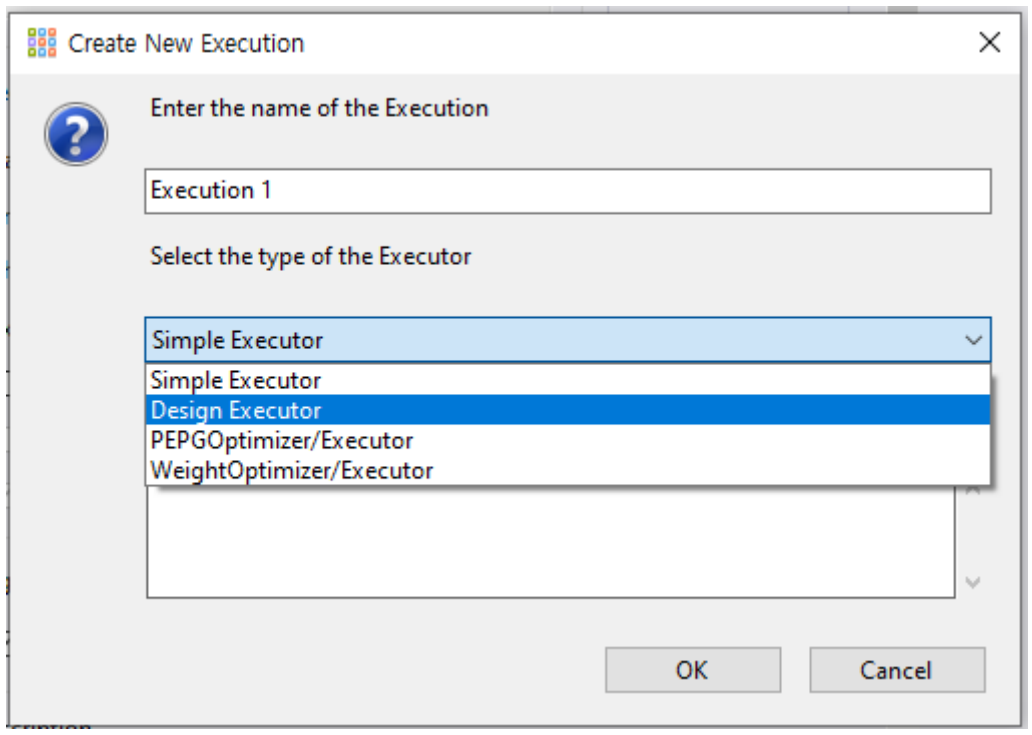

ExperimentDesign에서 Open Execution 메뉴



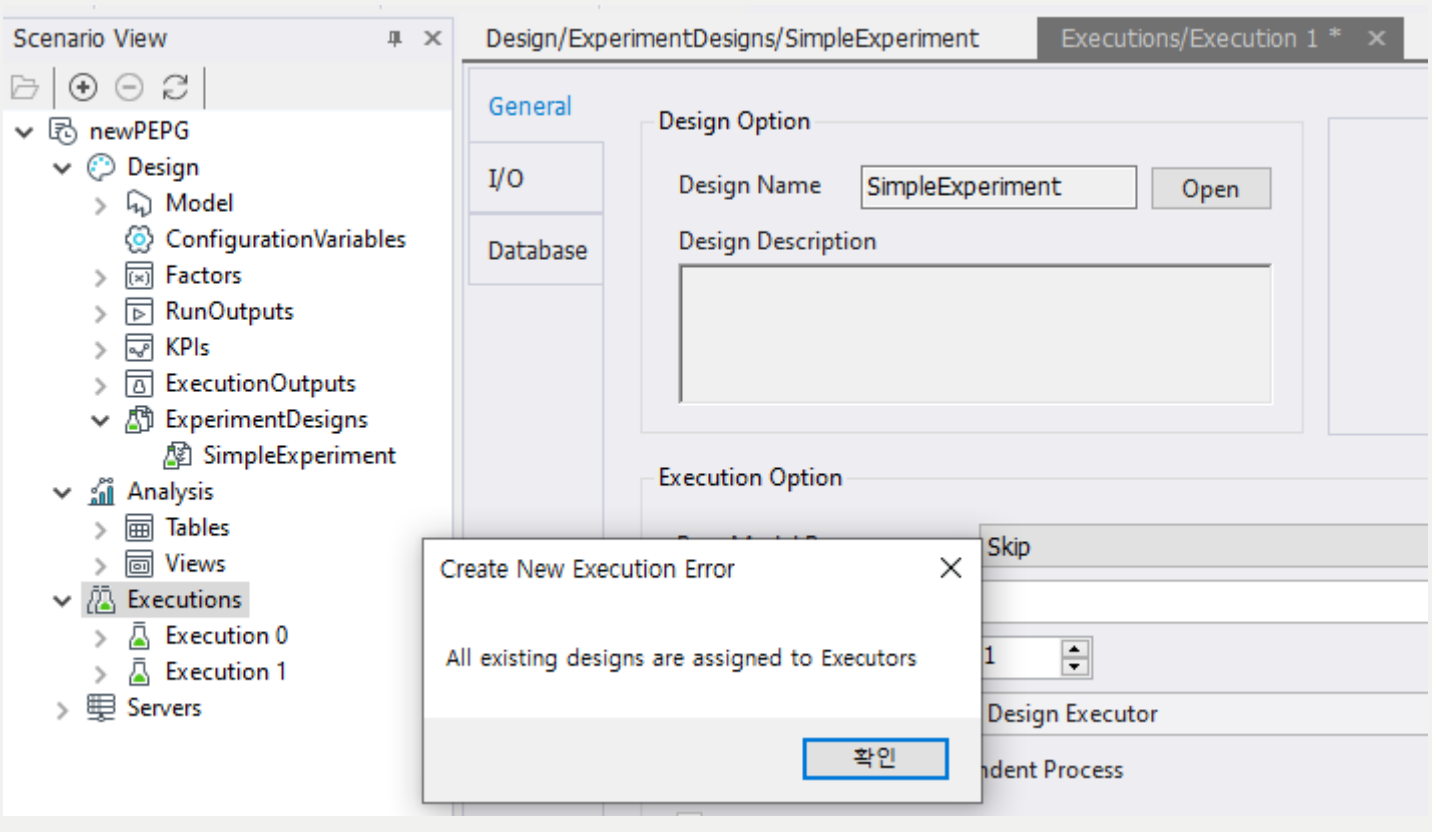

생성된 Design Execution

5. Design Execution
   a. Design Execution은 위와 같이 ExperimentDesign 노드에서 생성할 수도, 직접 Executions 아래에 생성할 수도 있다.
   b. 후자의 방법은 Executions 노드에서 **[Add]**를 클릭하여 생성합니다.

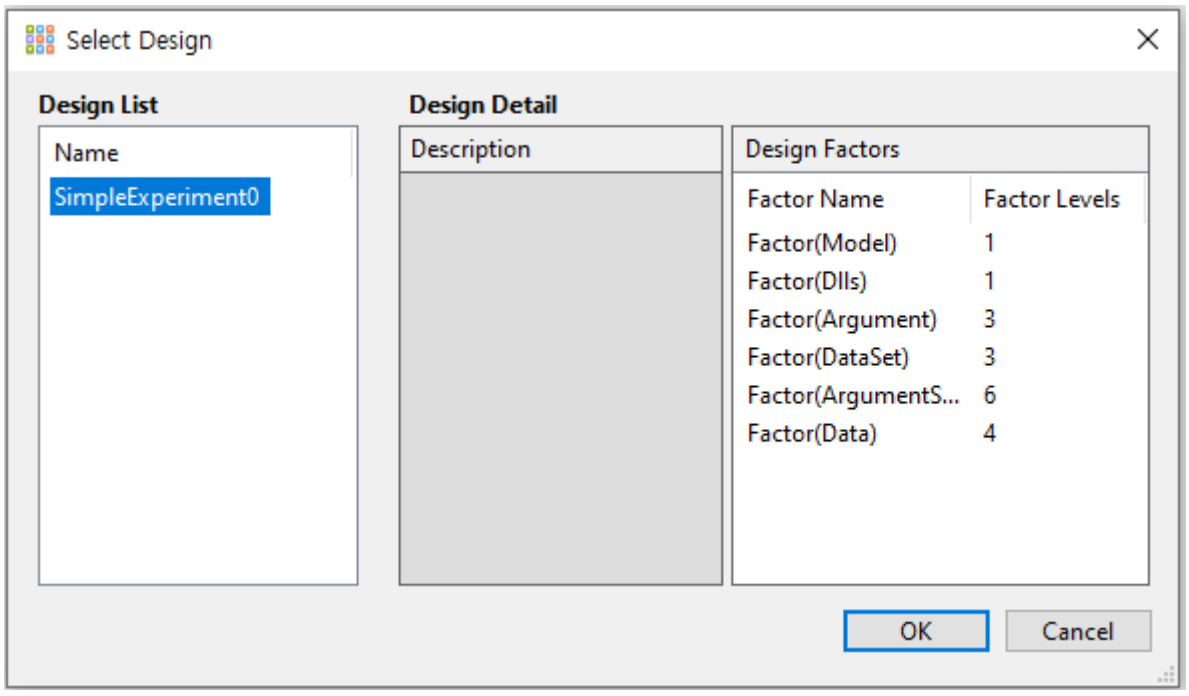





📌 하나의 Experiment Design은 하나의 Design Executor를 가집니다. Design Executor를 생성시에 모든 Experiment Design이 각자의 Design Executor를 가지고 있으면 아래와 같이 'All existing designs are assigned to Executors' 알람이 나타납니다.

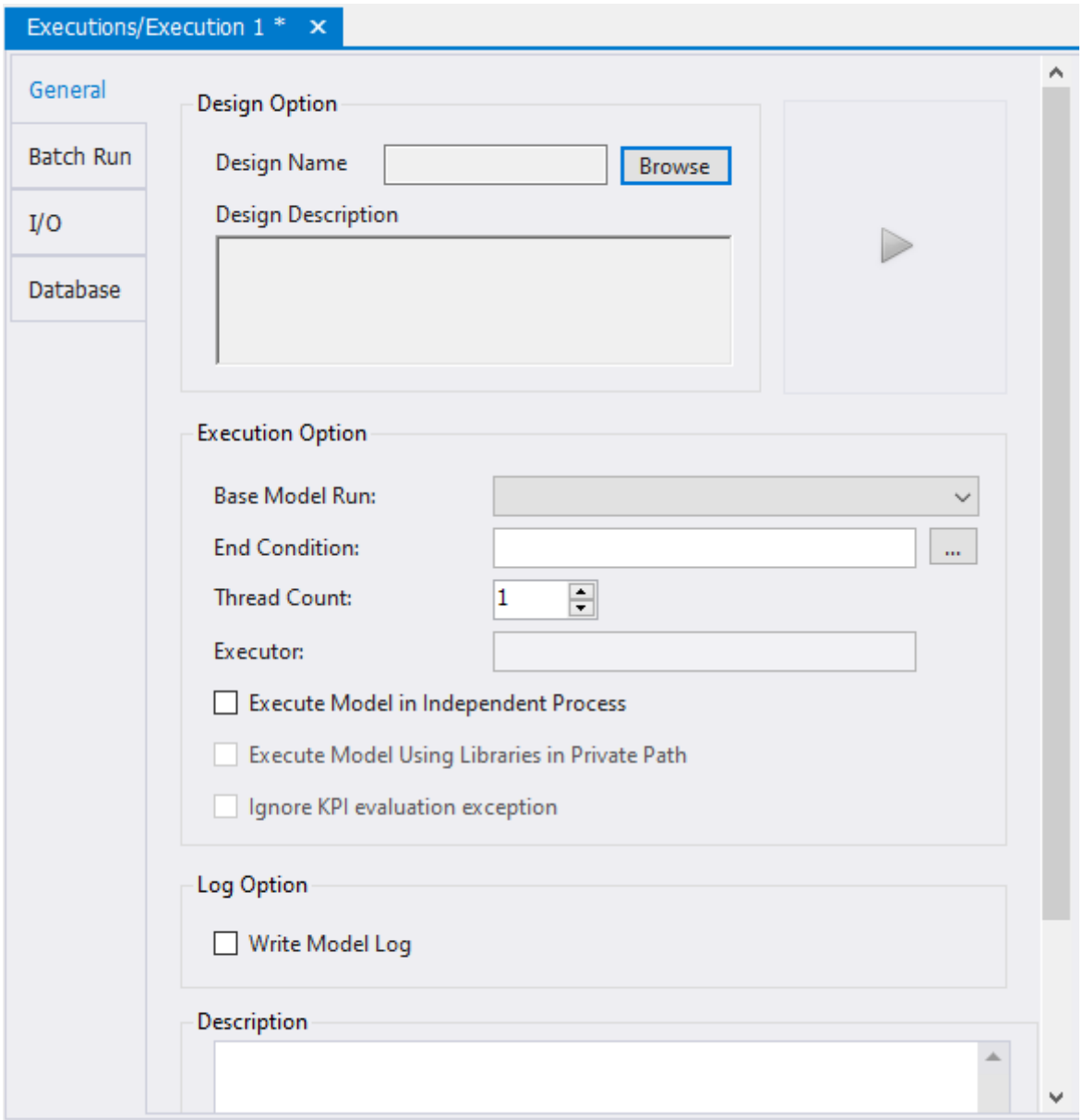



c. Executions 노드에서 직접 생성하게 되면, 어떤 ExperimentDesign을 실험할지 알 수 없기 때문에 어떤 Experiment Design을 실험할지 선택하는 화면이 나타납니다. 실험할 Experiment Design을 선택하면 해당 Experiment Design의 설명과 실험할 Factors들을 확인할 수 있습니다.




Select Design 화면

d. **[OK]** 버튼을 누르면 Design Executor가 생성됩니다.
e. **[Cancel]**을 누르면, 비어있는 Design Executor가 생성됩니다.




f. [Browse]를 누르면 'Select Design'화면이 나타납니다.
g. Experiment Design을 선택하고 오른쪽의 실행버튼을 클릭하면 Experiment Design에서 설계한 실험들이 실행됩니다.

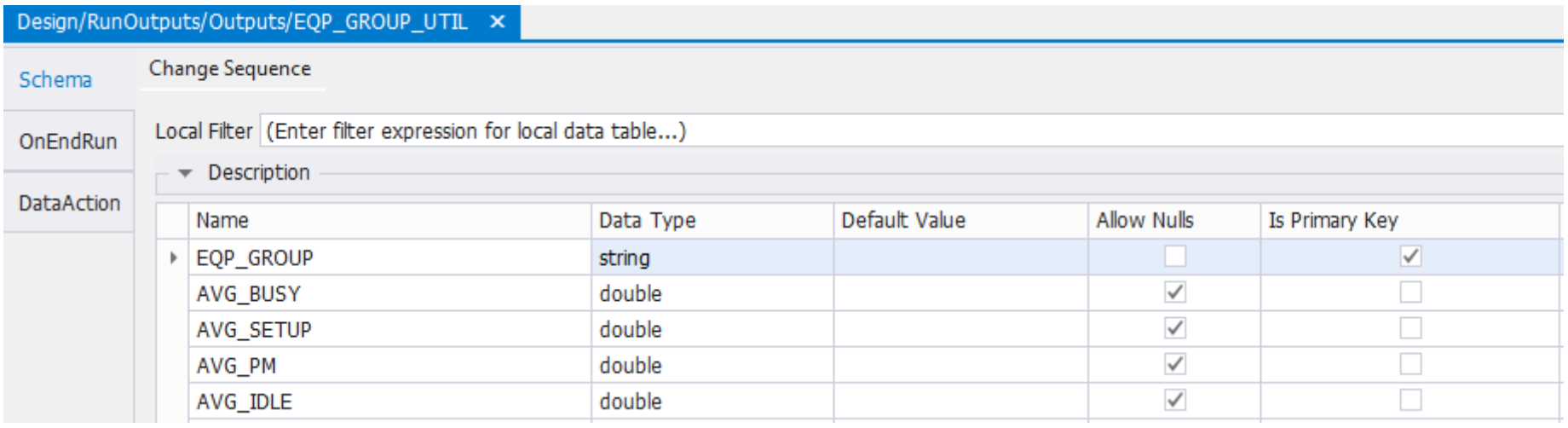
---

# RunOutputs

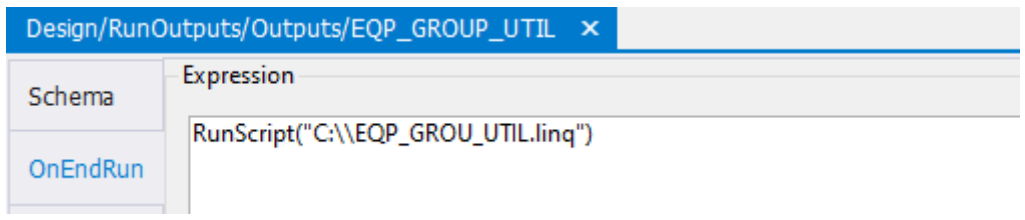
RunOutput은 시뮬레이션 종료 후 결과 분석에 필요한 데이터를 미리 추출 및 가공해 놓는 모델 단위의 사용자 정의 아웃풋입니다. 각 모델의 실행이 완료될 때마다 RunOutput 데이터를 생성합니다. 하나의 RunOutput은 스키마와 스키마를 채우기 위한 Expression으로 구성되어 있습니다. 기록된 RunOutput은 개별 Run의 RunOutputs 노드에서 확인할 수 있습니다.

## RunOutput 등록 절차

1. **[Design]>[RunOutputs]** 노드에서 **[Add]** 버튼을 눌러 RunOutput 그룹을 추가합니다.
2. 추가된 RunOutput 그룹 노드에서 **[Add]** 버튼을 눌러 RunOutput을 추가합니다.
3. RunOutput 명칭을 입력합니다.
4. RunOutput 항목이 추가된 후 **[Schema]** 탭에서 추출 및 가공한 데이터를 저장할 스키마를 정의합니다.
5. **[OnEndRun]** 탭에서 스키마를 채우기 위한 Expression을 정의합니다. Expression의 결과값은 DataTable 형식이어야 합니다.
6. Expression 작성 후 저장합니다.




RunOutput 스키마 작성 예시

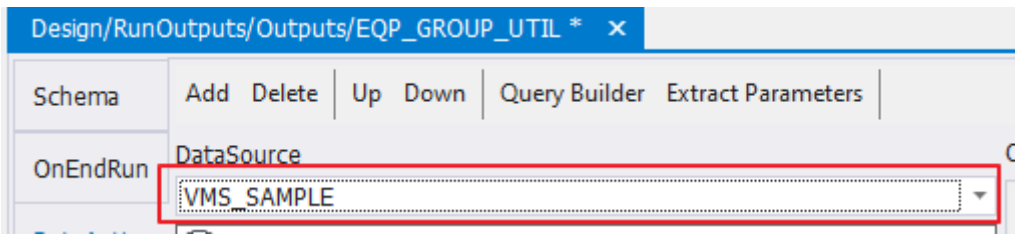



RunOutput OnEndRun Expression 작성 예시

위 예시는 모델 실행이 종료된 후 장비 그룹별 평균 가동률을 계산해서 저장하는 RunOutput을 정의한 예시입니다. 현재 Expression에서는 DataTable을 생성하거나 DataTable에 Row를 추가할 수 있는 함수를 제공하고 있지 않기 때문에, 반드시 LINQPad 스크립트를 실행해서 데이터를 생성해야 합니다.

## RunOutput의 DataAction 편집

1. RunOutput의 DataAction의 편집 절차는 기본적으로 모델 내 DataItem의 DataAction 편집 절차와 유사합니다.
2. Scenario Explorer에서 RunOutput을 더블클릭하여 설정 화면을 표시합니다.
3. [DataAction] 탭에서 모든 DataCommand에 대해 공통으로 사용할 DataSource를 선택합니다.
   a. BaseModel에 등록된 DataSource가 목록에 표시됩니다.




4. 좌측에서 DataCommand를 선택하고, 우측 Query 입력창에 Query를 입력합니다.
   a. Bind Table 옵션은 DataCommand를 Row마다 실행할지 여부입니다. Bind Table 옵션에 체크 해제된 경우 입력된 DataCommand는 한 번만 실행됩니다.

CommandType Text | AltDataSource - | Bind Table

```
INSERT INTO
  OUT_EQP_GROUP_UTIL
VALUES {
  @VERSION_NO,
  @EQP_GROUP,
  @AVG_BUSY,
  @AVG_SETUP,
  @AVG_PM,
  @ABG_IDLE
}
```

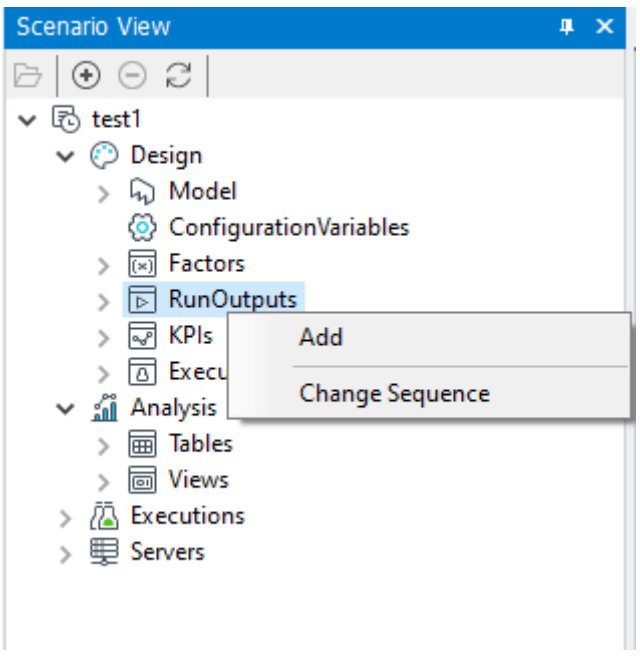
5. 입력한 Query에 Parameter가 존재하는 경우 상단의 [Extract Parameters] 버튼을 눌러 Query 내 Parameter를 추출하거나 직접 Parameter를 추가합니다.

| Name | DbType | Size | Precision | Scale | Direction | SourceVersion | SourceColumn |
|---|---|---|---|---|---|---|---|
| @VERSION_NO | VarChar | 0 | 0 | 0 | Input | Current | |
| @EQP_GROUP | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_BUSY | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_SETUP | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_PM | VarChar | 0 | 0 | 0 | Input | Current | |
| @ABG_IDLE | VarChar | 0 | 0 | 0 | Input | Current | |

6. 저장 버튼을 눌러 변경 사항을 저장합니다.

### Change Sequence

각 RunOutput Group내의 RunOutput의 계산 순서를 변경하는 기능입니다.




---

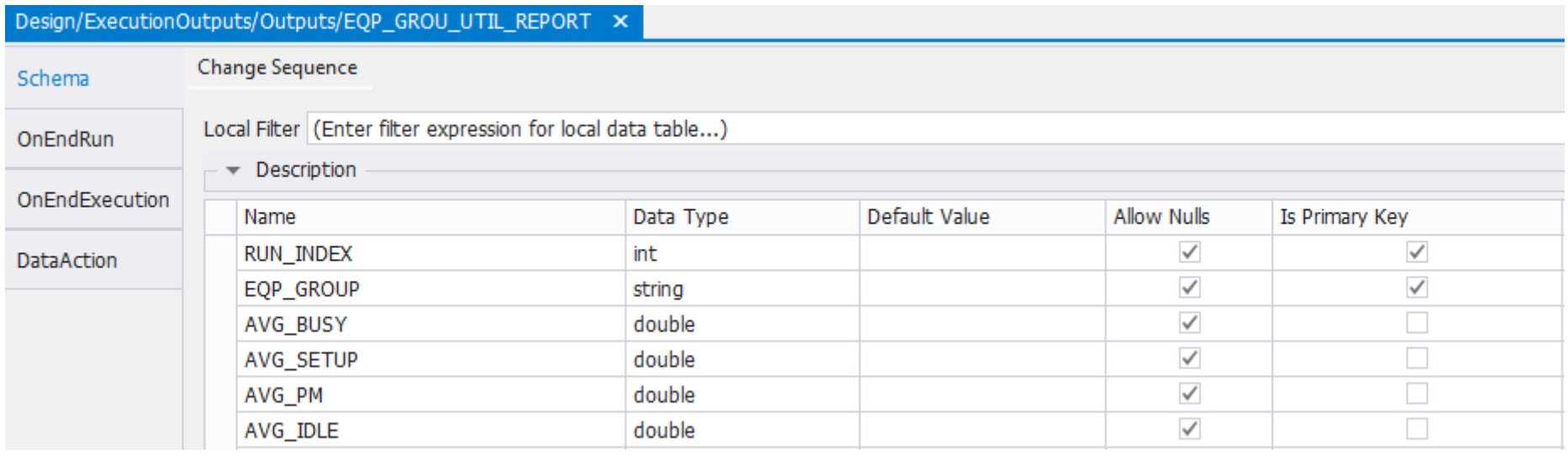
# ExecutionOutputs

ExecutionOutput은 Simulation 종료 후 결과 분석에 필요한 데이터를 미리 추출 및 가공해 놓는 시나리오 단위의 사용자 정의 아웃풋입니다. 각 모델의 실행이 완료될 때마다 또는 시나리오에 정의된 모든 모델의 실행이 완료된 후의 두 가지 시점에 ExecutionOutput 데이터를 생성합니다. RunOutput은 실행된 모델의 수 만큼 테이블이 생성되지만 ExecutionOutput은 하나의 테이블만 생성됩니다. 하나의 ExecutionOutput은 스키마와 스키마를 채우기 위한 OnEndRun Expression 및 OnEndExecution Expression으로 구성되어 있습니다. 기록된 ExecutionOutput은 Execution의 ExecutionOutputs 노드에서 확인할 수 있습니다.
모든 시나리오에는 Run 별 Factor와 KPI 값을 기록하는 DefaultReport ExecutionOutput이 등록되어 있습니다.

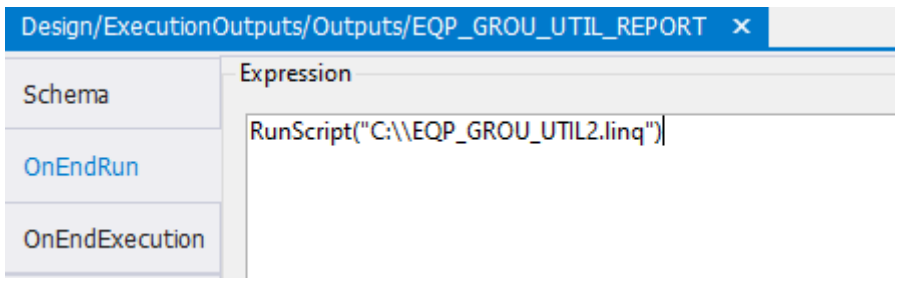
## ExecutionOutput 등록 절차

1. **[Design]>[ExecutionOutputs]** 노드에서 **[Add]** 버튼을 눌러 ExecutionOutput 그룹을 추가합니다.
2. 추가된 ExecutionOutput 그룹 노드에서 **[Add]** 버튼을 눌러 ExecutionOutput을 추가합니다.
3. ExecutionOutout 명칭을 입력합니다.
4. ExecutionOutput 항목이 추가된 후 [Schema] 탭에서 추출 및 가공한 데이터를 저장할 스키마를 정의합니다.

5. [OnEndRun] 탭에서 모델 실행이 완료될 때마다 스키마를 채우기 위한 Expression을 정의합니다. Expression의 결과값은 DataTable 형식이어야 합니다.
6. [OnEndExecution] 탭에서 모든 모델의 실행이 완료된 후 스키마를 채우기 위한 Expression을 정의합니다. Expression의 결과값은 마찬가지로 DataTable 형식이어야 합니다.
7. Expression 작성 후 저장합니다.




ExecutionOutput 스키마 작성 예시

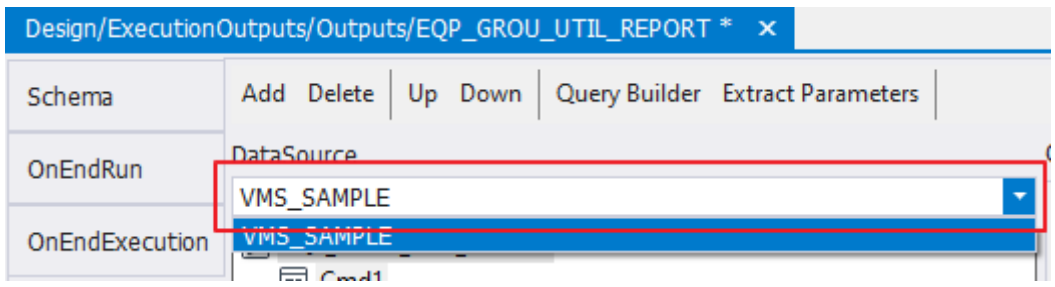



ExecutionOutput OnEndRun Expression 작성 예시

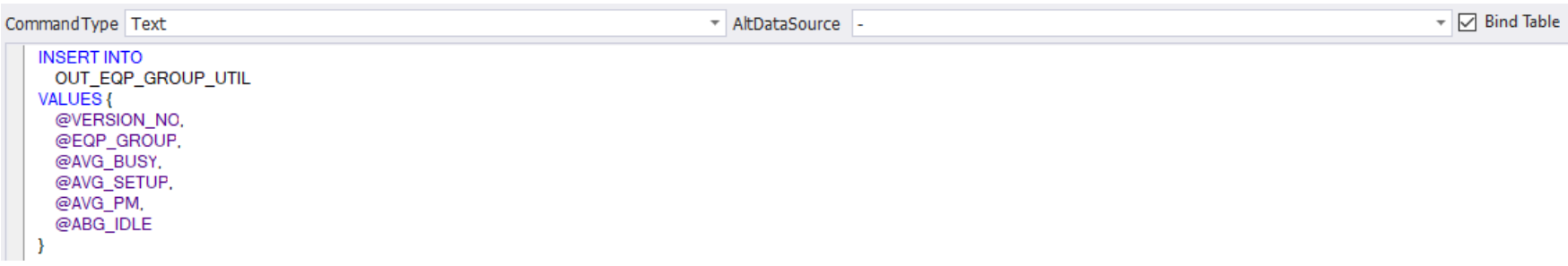
위 예시는 RunOutput과 유사하게 모델 실행이 종료되었을 때 장비 그룹별 평균 가동률을 계산해서 저장하는 ExecutionOutput을 정의한 예시입니다. 현재 Expression에서는 DataTable을 생성하거나 DataTable에 Row를 추가할 수 있는 함수를 제공하고 있지 않기 때문에, 반드시 LINQPad 스크립트를 실행해서 데이터를 생성해야 합니다.

## ExecutionOutput의 DataAction 편집

1. ExecutionOutput의 DataAction의 편집 절차는 기본적으로 모델 내 DataItem의 DataAction 편집 절차와 유사합니다.
2. Scenario Explorer에서 ExecutionOutput을 더블클릭하여 설정 화면을 표시합니다.
3. [DataAction] 탭에서 모든 DataCommand에 대해 공통으로 사용할 DataSource를 선택합니다.
   a. BaseModel에 등록된 DataSource가 목록에 표시됩니다.



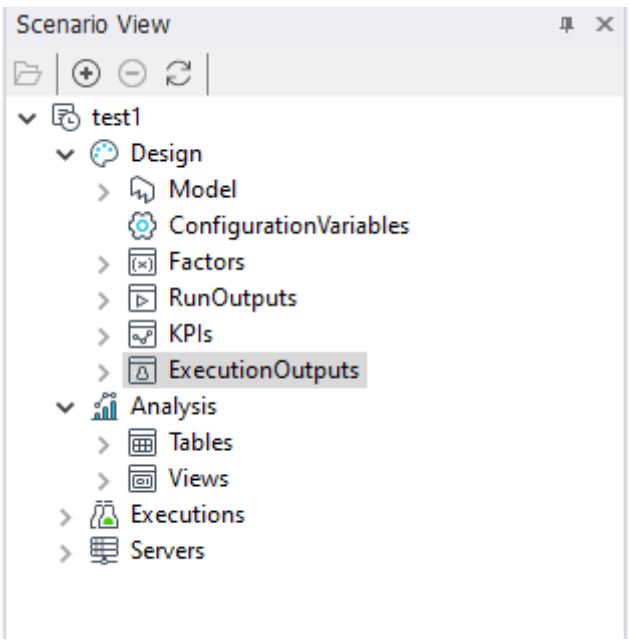

4. 좌측에서 DataCommand를 선택하고, 우측 Query 입력창에 Query를 입력합니다.
   a. Bind Table 옵션은 DataCommand를 Row마다 실행할지 여부입니다. Bind Table 옵션에 체크 해제된 경우 입력된 DataCommand는 한 번만 실행됩니다.



5. 입력한 Query에 Parameter가 존재하는 경우 상단의 [Extract Parameters] 버튼을 눌러 Query 내 Parameter를 추출하거나 직접 Parameter를 추가합니다.

| Name | DbType | Size | Precision | Scale | Direction | SourceVersion | SourceColumn |
|---|---|---|---|---|---|---|---|
| @VERSION_NO | VarChar | 0 | 0 | 0 | Input | Current | |
| @EQP_GROUP | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_BUSY | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_SETUP | VarChar | 0 | 0 | 0 | Input | Current | |
| @AVG_PM | VarChar | 0 | 0 | 0 | Input | Current | |
| @ABG_IDLE | VarChar | 0 | 0 | 0 | Input | Current | |

6. 저장 버튼을 눌러 변경 사항을 저장합니다.

## Change Sequence

각 ExecutionOutput Group내의 ExecutionOutput의 계산 순서를 변경하는 기능입니다.