Performance 사용법

MMC2의 Performance 노드에서는 대상 Server에 등록된 Trigger들의 전반적인 구동 성능을 모니터링 하는 영역입니다. Trigger Performance를 통해서 성능을 분석하여 엔진 구동 시 Task의 어떤 단계에서 Delay가 발생하는지 파악이 가능합니다. 특히 개발자나 관리자들은 해당 데이터를 통해서 Vmodel 또는 DLL이 변경 전과 후의 성능 비교가 가능하여 개선 여부를 파악하는데 용이합니다. 다음은 대상 Server에 등록된 Trigger의 Performance를 보는 방법과 Performance 창의 기능에 대해서 기술합니다.

대상 Trigger Performance 기록 방법

대상 Server에 등록된 Trigger의 성능을 Performance에서 확인하기 위해서는 Performance에 기록하기 위한 Trigger Argument를 설정해야 합니다. Trigger의 Argument는 다음과 같이 설정이 가능합니다.

1. Server Explorer에서 Trigger Argument를 설정 할 대상 Server를 선택합니다.
2. 대상 Server의 Trigger 노드를 더블 클릭하여 Trigger 창을 활성화 합니다.
3. Performance에 기록할 대상 Trigger를 선택하여 더블 클릭 혹은 상단 메뉴 바의 **[Edit]** 버튼을 클릭합니다.
4. Edit Trigger Dialog에서 Target Job 탭으로 이동합니다.
5. Target Job에 보이는 리스트 하단으로 내려가서performance-profiling 체크 박스에 체크 후 **[OK]** 버튼을 클릭하여 설정을 저장합니다.

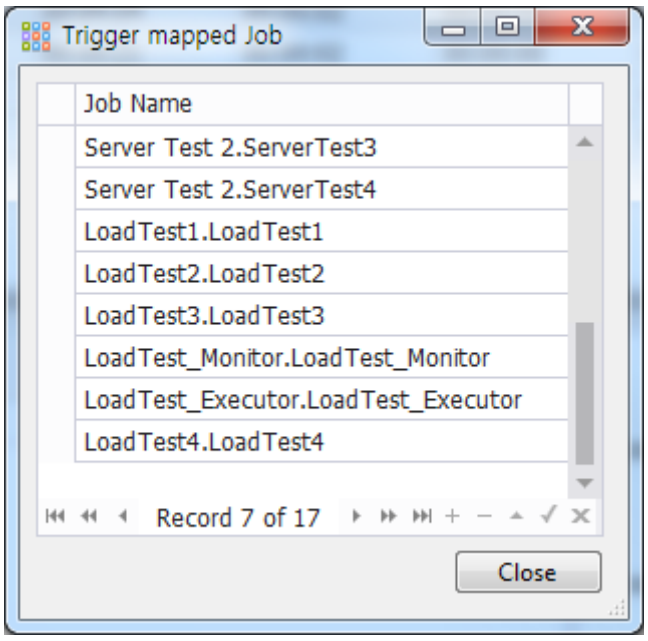
Performance UI 설명

- **Job Summary** Job Summary에서는 대상 Server에 등록된 총 Job의 개수와 Job과 매핑된 Trigger의 개수를 확인 할 수 있습니다.

Job Summary

Item	Count
Total Job	19
Trigger mapped Job	17
Trigger un-mapped Job	2

  - **Total Job** : 대상 Server에 등록된 Job의 총 개수입니다.
  - **Trigger mapped Job** : 대상 Server에 등록된 Job 중 Trigger가 등록된 Job의 개수입니다. Trigger mapped Job Row를 더블 클릭 시 Trigger가 등록된 Job의 목록을 확인 할 수 있습니다. 하나의 Job에 Trigger가 여러개 매핑이 되어 있어도 Job의 Trigger 매핑 여부를 집계하기 때문에 Trigger mapped Job에서의 카운트는 1개만 증가합니다. (EX: Job A {Trigger A, Trigger B, Trigger C} -> Trigger mapped Job = 1)



- **Total Job** : 대상 Server에 등록된 Job의 총 개수입니다.
- **Trigger mapped Job** : 대상 Server에 등록된 Job 중 Trigger가 등록된 Job의 개수입니다. Trigger mapped Job Row를 더블 클릭 시 Trigger가 등록된 Job의 목록을 확인 할 수 있습니다. 하나의 Job에 Trigger가 여러개 매핑이 되어 있어도 Job의 Trigger 매핑 여부를 집계하기 때문에 Trigger mapped Job에서의 카운트는 1개만 증가합니다. (EX: Job A {Trigger A, Trigger B, Trigger C} -> Trigger mapped Job = 1)

Trigger Summary

Item	Count
Total Trigger	20
Active Trigger	14
Profiling Trigger	6

- **Total Trigger** : 대상 Server에 등록된 Trigger의 총 개수입니다.
- **Active Trigger** : 대상 Server에 등록된 Trigger 중 **[Enabled]** 가 활성화 된 Trigger의 개수입니다.
- **Profiling Trigger** : 대상 Server에 등록된 Trigger 중performance-profiling이 True인 Trigger들의 개수입니다.

Trigger Performance

Trigger Performance에서는 Profiling Trigger의 대상 Trigger들의 성능집계 정보를 수치 및 그래프로 표현하는 영역입니다. Task가 실행이 되었을 시 각 단계별 소모 시간, Trigger의 총 실행 회수 및 성공/실패 비율 등의 정보를 확인 할 수 있습니다. 다음 은 Trigger Performance 영역의 용어 정의를 기술합니다.

- **Period** : 사용자가 Trigger Performance를 조회하고자 하는 기간을 설정합니다. 기간은 Days/Hours 중 선택이 가능하며 현재를 기준으로 설정한 Days/Hours 전에 집계된 이력들을 호출합니다.
- **Trigger**
  - Name : Triggers 노드 기준 등록된 Trigger의 명칭입니다.
  - Description : 대상 서버의 등록된 Trigger의 설명입니다. Description은 Triggers 노드에서 입력이 가능합니다.

Trigger

Name	Description
Colaboration	
Dependent	Dependent Trigger Test
Fab	
Fab3	
FabCopyTest	

RunTime(sec)

- DOWNLOAD : Task 수행된 총 횟수의 DB로부터 Data를 Download하는데 소모된 평균 시간을 표시합니다.

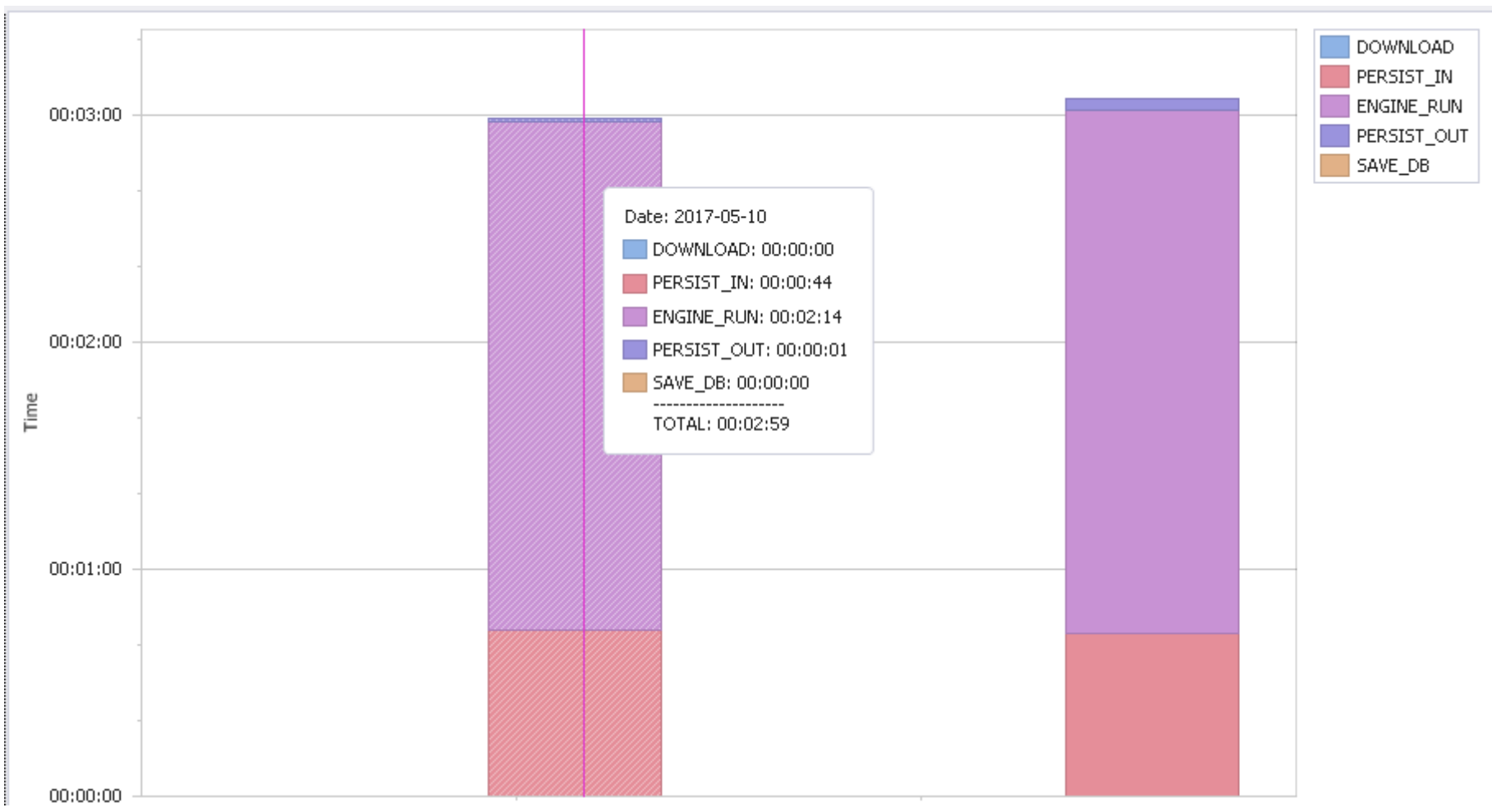
- PERSIST_IN : Task 수행된 총 횟수의 Input Data를 로딩하는데 소모된 평균 시간을 표시합니다.
- ENGINE_RUN : Task 수행된 총 횟수의 모듈(Pegging, Simulation, CBS, etc)를 수행의 평균 소모 시간을 표시합니다.
- PERSIST_OUT : Task 수행된 총 횟수의 엔진 결과를 저장하는데 소모된 평균 시간을 표시합니다.
- SAVE_DB : Task 수행된 총 횟수의 결과 데이터를 대상 DB에 Upload하는데 소모된 평균 시간을 표시합니다.
- TOTAL_RUN : DOWNLOAD - SAVE_DB Action 까지의 누적 시간의 평균 시간입니다.

RunTime(sec)					
DOWNLOAD	PERSIST_IN	ENGINE_RUN	PERSIST_OUT	SAVE_DB	TOTAL_RUN
00:00:00	00:01:18	00:04:04	00:00:02	00:00:00	00:06:07
00:00:00	00:03:26	00:00:21	00:00:02	00:00:00	00:04:09
00:00:00	00:00:49	00:02:25	00:00:02	00:00:00	00:03:37
00:00:00	00:00:41	00:00:20	00:00:00	00:00:00	00:01:04
00:00:00	00:00:00	00:00:00	00:00:00	00:00:00	00:00:02

Trigger Performance의 RunTime(sec)에서는 대상 Trigger의 Action 별 평균 소모 시간을 표시가 되며 Trigger Performance 영역의 좌측 하단 그리드에서는 대상 Trigger의 Action 별 최대/최소 소모시간 조회가 가능합니다.

Action	Min	Average	Max
PERSIST_IN	00:00:49	00:01:18	00:01:32
ENGINE_RUN	00:02:29	00:04:04	00:04:32
PERSIST_OUT	00:00:01	00:00:02	00:00:03

또한, Trigger Performance 영역의 우측 하단에서는 사용자가 설정한 기간에 대상 Trigger의 실행 횟수만큼 각 Action 별 TOTAL_RUN 시간에서 차지하는 비율을 그래프 형태로 확인이 가능합니다. 상세한 정보 조회를 위해서는 조회하고자 하는 바 그래프에 마우스 오버 시 해당 시간에 수행된 Trigger의 Action별 수행 시간을 확인 할 수 있습니다.



Reliability

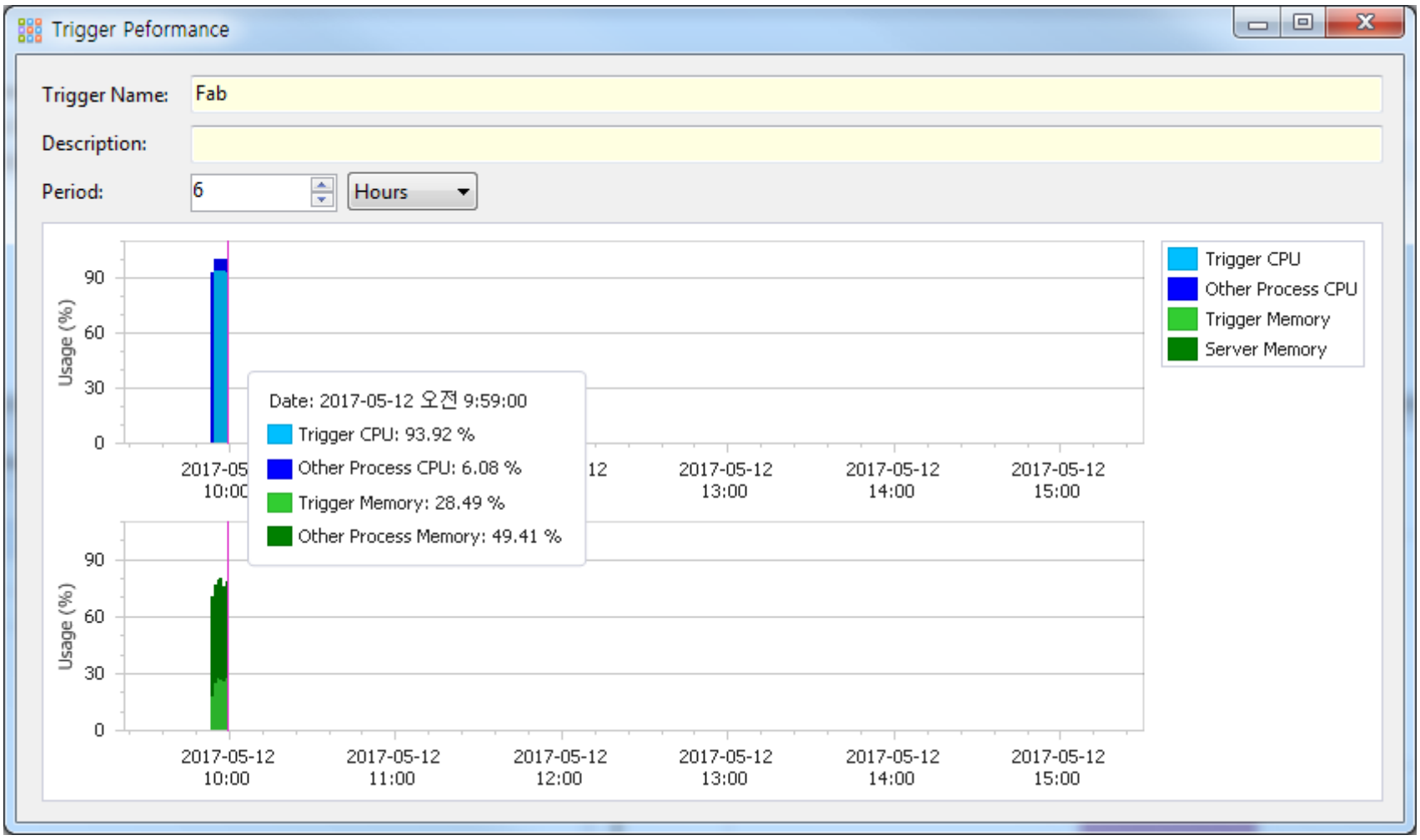
- MIN : 대상 Trigger 총 실행 횟수 중 Task 수행 시간이 가장 짧은 시간입니다.
- MAX : 대상 Trigger 총 실행 횟수 중 Task 수행 시간이 가장 긴 시간입니다.
- LIMIT : 사용자가 설정한 대상 Trigger의 수행 제한 시간입니다. Triggers -> 대상 Trigger -> Schedule Tab -> Stop task if it runs longer than 옵션에 설정된 시간입니다.

- RUN_COUNT : 대상 Trigger가 실행된 총 횟수입니다. Stop task if it runs longer than 옵션 또는 Monitoring에서 Stop Trigger 기능을 통해 사용자가 강제로 중지한 경우 집계되지 않습니다.
- FAIL_COUNT : 대상 Trigger가 총 실행된 횟수 중 오류에 의해 비정상적으로 중지된 횟수입니다.
- FAIL_RATE : 대상 Trigger의 RUN_COUNT 중 FAIL_COUNT의 비율입니다.
- SUCCESS_RATE : 대상 Trigger의 RUN_COUNT 중 정상 수행된 비율입니다.

Reliability

MIN	MAX	LIMIT	RUN_COUNT	FAIL_COUNT	FAIL_RATE	SUCCESS_RATE
00:03:35	00:07:06	00:00:00	5	0	0 %	100 %
00:04:09	00:04:09	00:00:00	1	0	0 %	100 %
00:00:01	00:07:37	00:02:00	22	6	27 %	73 %
00:01:04	00:01:04	00:00:00	1	1	100 %	0 %
00:00:02	00:00:02	00:00:00	1	1	100 %	0 %

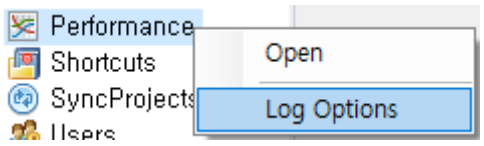
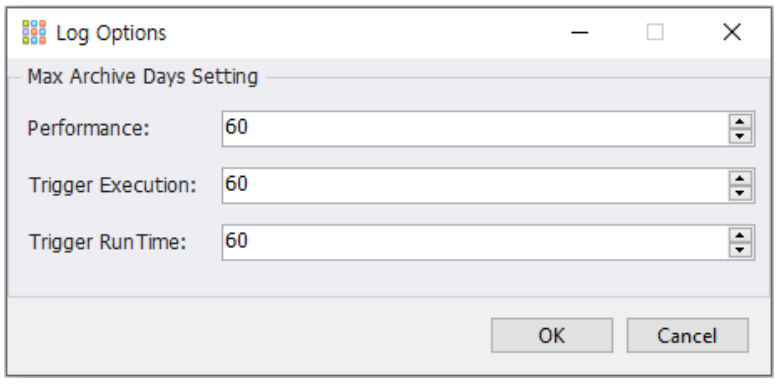
- **Trigger Performance Dialog (Trigger의 대상 Server Machine의 자원 사용량 확인하기)**
  Performance 창에서는 Trigger가 수행된 시간 동안 대상 Server Machine의 CPU/Memory 사용량 및 비율을 그래프 형태로 Trigger Performance Dialog에서 조회가 가능합니다. ***Performance 창의 Trigger Performance 그리드에서 대상 Trigger의 Row를 더블 클릭 시*** Trigger Performance Dialog 창이 열립니다.



  Period는 Trigger Performance에서 사용자가 지정한 Period 기준을 따르고 있으며, Trigger Performance Dialog 창에서 사용자가 임의로 기간을 다시 설정 할 수 있습니다. Trigger Performance Dialog 창에서는 대상 Trigger가 수행된 시점에 수행 시간 만큼 집계가 됩니다. Trigger 수행 시간 동안의 Server Machine의 총 CPU/Memory 사용량을 그래프로 표시되며 상단의 푸른색 그래프는 CPU 사용량, 하단의 녹색 그래프는 Memory 사용량입니다. 그 중 연한 푸른색은 CPU 총 사용량 중 대상 Trigger (MozartAgent 프로세스)가 차지하는 비율이고, 연한 녹색은 Memory 총 사용량 중 대상 Trigger (MozartAgent 프로세스)가 차지하는 비율입니다. 진한 색상의 경우는 Trigger가 차지하는 비율 외에 다른 프로세스들이 차지하는 비율을 나타냅니다. MMC2에서의 대부분의 그래프와 마찬가지로 Trigger Performance Dialog 역시 그래프 바 위에 마우스 오버 시 상세한 정보 확인이 가능합니다.

Log 보존 기간 설정

Trigger 가 수행한 이력 정보와 Performance 기록 정보의 보관 기간을 지정 할 수 있습니다. Performance 우측 마우스를 눌러 Log Options 메뉴를 통해 설정할 수 있습니다.



기본 설정된 보관 기간은 60일로 되어 있으며, 보관 기간을 일자(Day) 기준으로 재설정 할 수 있습니다.