OMRON

데이터 플로우 컨트롤러

DX 시리즈

SpeeDBee Synapse

사용자 매뉴얼

DX100-0010

⚠ 안내 (For Reference Only)

본 문서는 AI 번역으로 작성된 참고용 국문자료 입니다.
일부 표현은 원문과 다소 다를 수 있으며,
중요한 내용은 원문과 함께 참고해 주시기 바랍니다.

CPU 유니트



SBSB-540A

양해 말씀

・본 매뉴얼 내용의 일부 또는 전부를 무단으로 복사, 복제, 전재하는 것을 금지합니다.

・본 매뉴얼의 내용에 관해서는 개량을 위해 예고 없이 사양 등을 변경하는 경우가 있습니다.
미리 양해해 주시기 바랍니다.

・본 매뉴얼의 내용에 관해서는 만전을 기하고 있습니다만, 만일 의심스러운 점이나 오류 등 발견하신 사항이 있으시면 번거로우시겠지만 당사 지점 또는 영업소로 연락해 주십시오.
그때 권말에 기재된 Man.No.(매뉴얼 No.)도 함께 알려 주십시오.

AI 번역본
For Reference Only

상표

・Microsoft, Windows, Excel, Visual Basic, Microsoft Edge는 미국 Microsoft Corporation의 미국 및 그 밖의 국가에서의 등록상표 또는 상표입니다.

・ODVA, CIP, CompoNet, DeviceNet, EtherNet/IP는 ODVA의 상표입니다.

・SpeeDBee Synapse는 주식회사 솔티스터의 상표입니다.

・Grafana는 Grafana Labs의 상표입니다.

그 밖에, 본문 중에 게재하고 있는 회사명 및 제품명은, 각각 각 사의 상표 또는 등록상표입니다.

저작권

・스크린샷은 마이크로소프트의 허락을 얻어 사용하고 있습니다.

・본 제품에는 서드파티 소프트웨어가 내장되어 있습니다. 이 소프트웨어와 관련된 라이선스와 저작권에 대해서는 https://www.fa.omron.co.jp/product/tool/dx-info/index.html 을 참조하십시오.

Table of Contents

1 목차

2 머리말

3 SpeeDBee Synapse의 개요

4 도입 플로우

- 4.1 도입 플로우
- 4.2 설치（Linux）
- 4.3 설치（Windows）
- 4.4 초기 설정
- 4.5 컴포넌트의 설정
- 4.6 수집 서비스의 실행

5 각종 기능 설명

- 5.1 각종 기능 설명
- 5.2 엔진 기능의 전체상
- 5.3 플로우 에디터
- 5.4 컬렉터
- 5.5 시리얼라이저
- 5.6 에미터
- 5.7 로직（이벤트）
- 5.8 로직（실시간 분석/데이터 조작）
- 5.9 액션
- 5.10 개발용 컴포넌트
- 5.11 시스템 컴포넌트
- 5.12 컴포넌트 공통의 설정 항목
- 5.13 스테이터스 타임라인
- 5.14 각종 로그
- 5.15 DB 쿼리（SQL）
- 5.16 연산식・조건식
- 5.17 설정 정보의 관리(프로젝트)

6 응용적인 이용 방법

AI 번역본
-For Reference Only-

6.1 응용적인 이용 방법
6.2 커스텀 컴포넌트의 작성・이용
6.2.1 커스텀 컴포넌트의 작성・이용
6.2.2 커스텀 컴포넌트의 개요
6.2.3 Python 커스텀 컴포넌트 작성
6.2.4 C언어 커스텀 컴포넌트 작성
6.2.5 커스텀 UI 작성
6.2.6 커스텀 컴포넌트 개발 환경(SCCDE)
6.3 제어 프로그램과의 연계（WebAPI）
6.4 Grafana와의 연계
6.5 글로벌 변수
7 트러블슈트
8 보안 이용 가이드
9 후방 호환에 관하여
10 실행 라이선스
11 i-Trigger（확장 패키지）
11.1 i-Trigger（확장 패키지）
11.2 개요
11.3 사용할 수 있는 카메라에 대해
11.4 설치
11.5 동영상 재생용 뷰어
11.6 관련 컴포넌트
11.7 튜토리얼
11.8 i-Trigger의 기록 사양
12 릴리스 노트
13 상표
14 제조원 문의
15 부록
15.1 부록
15.2 에러 코드 목록
15.3 커스텀 컴포넌트 Python-API 레퍼런스
15.4 커스텀 컴포넌트 C-API 레퍼런스
15.5 커스텀 UI JS-API 레퍼런스

AI 번역본
-For Reference Only-

15.6 이용하고 있는 오픈 소스

15.7 색인

15.8 DB 쿼리 서비스 Web-API 사양

AI 번역본
-For Reference Only-

1 SpeeDBee Synapse 도큐먼트

- 머리말
- SpeeDBee Synapse의 개요
- 도입 플로우
- 각종 기능 설명
- 응용적인 이용 방법
- 트러블슈트
- 보안 이용 가이드
- 후방 호환에 관하여
- 실행 라이선스
- i-Trigger（확장 패키지）
- 릴리스 노트
- 상표
- 제조원 문의
- 부록

AI 번역본
-For Reference Only-

2 머리말

본 매뉴얼은,「 SpeeDBee Synapse（ 약칭 : SBS ）」의 조작 방법에 대해 기재한 것입니다. SpeeDBee Synapse를 활용함으로써, 제조 현장 설비의 데이터 수집, 타 시스템 연계가 가능해집니다. 물론, 제조 현장 이외에서도 IoT 시스템을 구축할 수 있습니다.

2.1 대상 독자

본 매뉴얼은 다음 분들을 대상으로 기술하고 있습니다.

- IoT 시스템, FA 시스템을 설계하시는 분
- IoT 시스템, FA 시스템을 관리하시는 분
- IoT 시스템, FA 시스템을 실제로 조작, 운용하시는 분

또한, 프로그래밍 언어를 사용하여, 독자적인 기능을「 커스텀 컴포넌트 」로서 실장할 수 있습니다. 그 경우는, 프로그래밍에 관한 지식이 필요합니다. 상세에 대해서는 매뉴얼 내에서 해설합니다.

2.2 동의 사항

SpeeDBee Synapse가 설치되어 있는 하드웨어 본체의 매뉴얼, 또는 SpeeDBee Synapse의 설치 프로그램 실행 시에 표시되는 내용을 확인하십시오.

2.3 안전상의 주의

아래 기능에 대해서, 대향 기기의 레지스터나 변수, 스테이터스 등을 변경하는 것이 가능하기 때문에, 변경 결과에 따라 설비의 거동에 영향을 줄 가능성이 있습니다.
이들 기능을 사용할 때는 설비 영향 등을 충분히 고려 및 검증한 후에 사용하십시오.

・PLC 쓰기
・Modbus 쓰기
・EtherNet/IP 쓰기

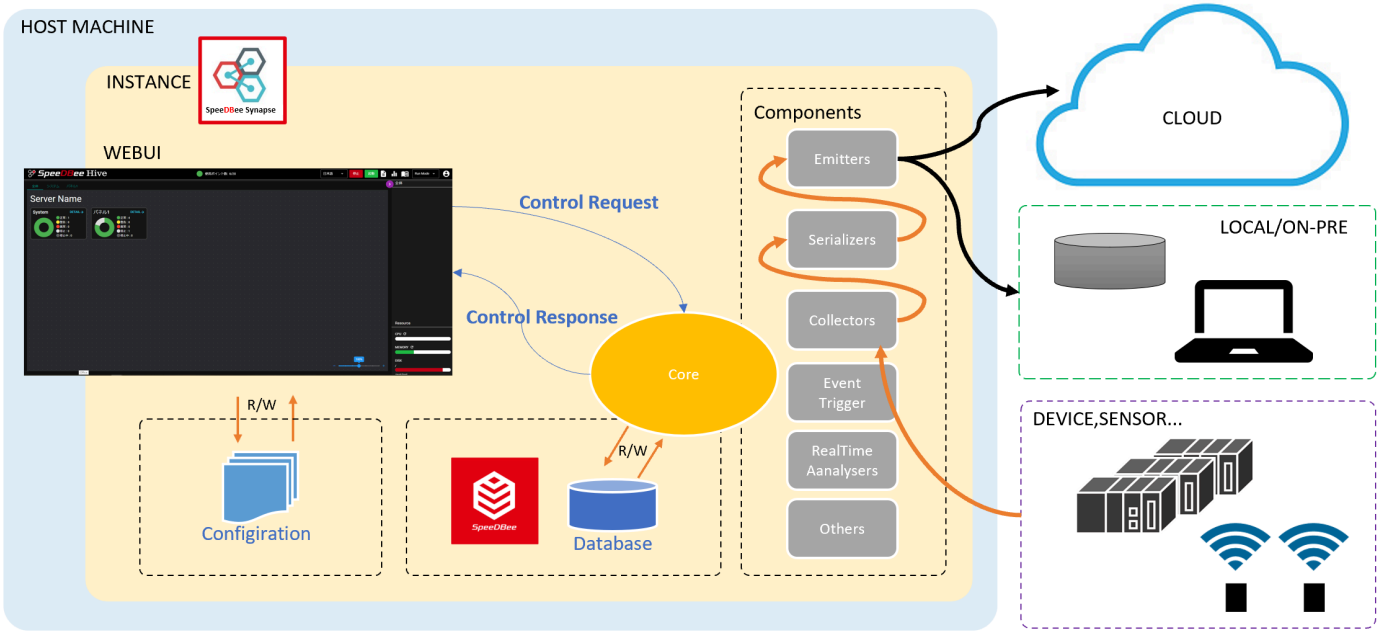
AI 번역본
-For Reference Only-

3 SpeeDBee Synapse의 개요

본 장에서는, SpeeDBee Synapse의 특징, 구성, 운용까지의 흐름에 대해 설명합니다.

3.1 SpeeDBee Synapse의 특징

SpeeDBee Synapse는, FA나 IoT를 구성하는 다양한 기기（PLC 등）로부터 데이터를 수집할 수 있습니다. 수집한 데이터를 SpeeDBee Synapse의 각종 기능을 이용함으로써, 변환이나 분석, 그리고, 타 시스템과의 데이터 연계가 가능합니다. 이로써 데이터 수집부터 데이터 활용까지를 실현하는 시스템을 구축할 수 있습니다.



3.2 SpeeDBee Synapse의 구성

SpeeDBee Synapse는, 크게 4가지 기능으로 구성됩니다.

기능명	설명
코어	수집 엔진의 전체 기능을 제어하는 중심부입니다.
컴포넌트	데이터의 수집, 가공, 분석, 송신 등을 실현하는 모듈군입니다. 컴포넌트 간을 접속함으로써 다양한 기능을 실현할 수 있습니다.
시계열 데이터베이스	수집한 데이터를 시간에 따라 저장하는 데이터베이스입니다.
WEBUI	각종 기능의 설정, 데이터의 가시화를 하는 사용자 인터페이스입니다.

컴포넌트는 7 개의 카테고리로 분류됩니다. 각 카테고리에 속하는 컴포넌트를 아래 표에 나타냅니다.

카테고리	명칭	설명
컬렉터	-	데이터 수집 기능을 제공합니다. 데이터 수집 후의 각종 기능을 이용하기 위한 기점이 됩니다.
	EtherNet/IP	EtherNet/IP 대응 기기에서 데이터 수집합니다.
	PLC	PLC의 레지스터/변수 정보에서 데이터 수집합니다.
	Modbus TCP	Modbus TCP 대응 기기에서 데이터 수집합니다.
	MQTT	MQTT 프로토콜（Subscribe）에 의한 데이터 수집을 수행합니다.
	리소스	가동 머신 내의 시스템 리소스（CPU 사용률, 메모리 소비 등）를 수집합니다.
	연산 컬렉터	수집 데이터를 사용한 연산의 결과를 새로운 별도 데이터로서 등록합니다.
에미터	-	데이터 출력 또는 데이터 송신 기능을 제공합니다. 외부 시스템과의 연계에 활용할 수 있습니다.
	Cloud（Azure용）	Azure IoT Hub용으로 데이터 송신을 수행합니다.
	MQTT	MQTT 프로토콜（Publish）에 의한 데이터 송신을 수행합니다.
	FTP/FTPS/SFTP	FTP, FTPS 또는 SFTP에 의한 데이터 전송을 수행합니다.
	File	수집한 데이터를 파일로서, 스토리지에 저장합니다.
	Cloud（AWS용）	AWS IoT Core용으로 데이터 송신을 수행합니다.
시리얼라이저	-	데이터를 지정 포맷으로 가공합니다.
	CSV	지정 데이터를 CSV 형식으로 가공합니다.
	JSON	지정 데이터를 JSON 형식으로 가공합니다.
액션	-	수집한 데이터를 사용한 임의의 액션을 제공합니다.

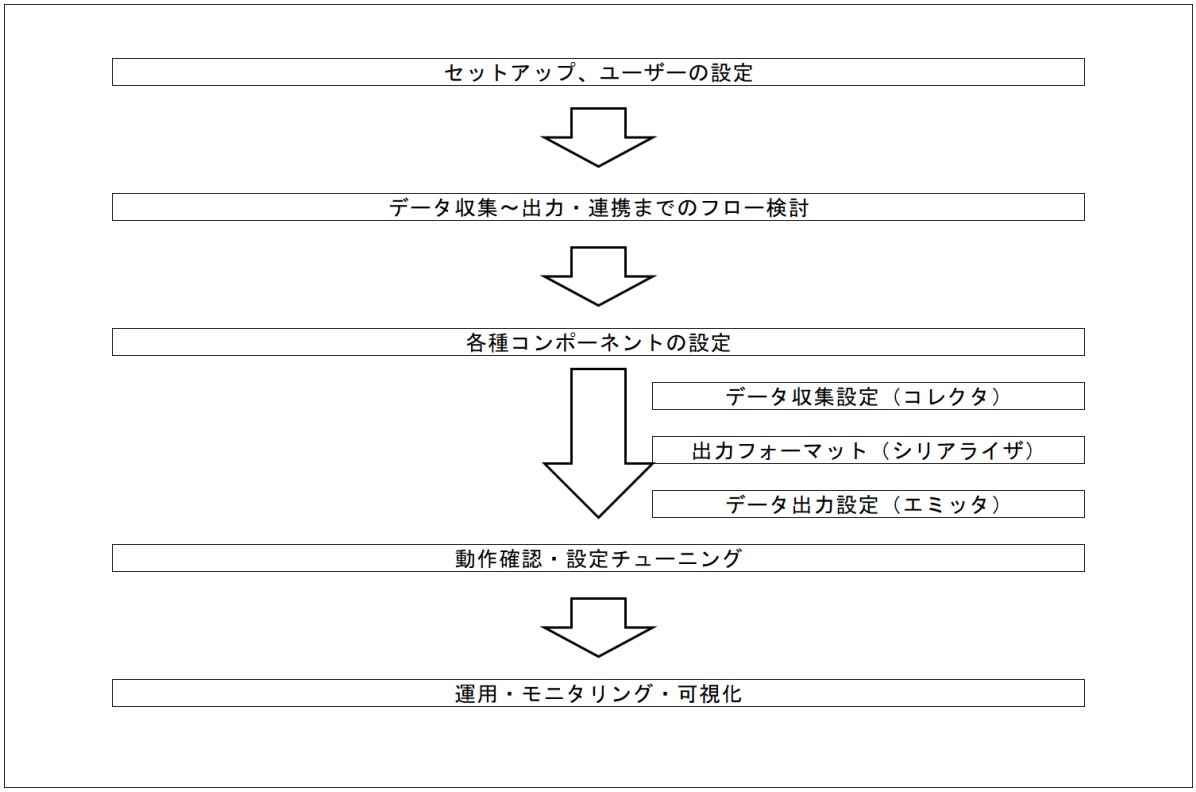
AI 번역본
-For Reference Only-

카테고리	명칭	설명
	셸 커맨드	수집한 데이터를 사용한 임의의 셸 커맨드를 실행합니다.
	메일 송신	수집한 데이터를 사용하여 메일 송신을 수행합니다.
	Modbus 쓰기	Modbus TCP 대응 기기의 지정 주소에 값을 씁니다.
	PLC 쓰기	PLC의 지정 주소에 값을 씁니다.
	EtherNet/IP 쓰기	EtherNet/IP 대응 기기에 값을 씁니다.
로직	-	실시간 분석이나 수집 데이터의 판정 처리를 제공합니다.
	이벤트 데이터	수집한 데이터를 조건 판정하고, 조건 합치한 경우만, 새롭게 별도 데이터로서 등록합니다.
	이벤트 트리거	수집한 데이터를 조건 판정하고, 조건이 일치하는 구간의 정보 제공을 수행합니다.
	기본 통계	수집 데이터를 사용하여 통계 처리한 결과를 별도 데이터로서 등록합니다.
	이동 평균	수집 데이터의 이동 평균값을 데이터로서 등록합니다.
	FFT	수집 데이터를 FFT한 결과를 데이터로서 등록합니다.（주파수, 강도）
시스템	-	시스템에 관련하는 기능을 제공합니다.
	DB 쿼리 서비스	SpeeDBee Synapse 내에서 모은 데이터로의 조회가 가능합니다. Grafana 연계에서도 사용합니다.
	코어 인터널	SpeeDBee Synapse 내의 내부 스테이터스를 관리하는 컴포넌트입니다.
	에러 매니저	SpeeDBee Synapse 내에서 발생한 전체 에러를 검지할 수 있는 컴포넌트입니다.
커스텀	-	사용자가 Python 또는 C언어로 설계한 프로그램으로 구성된 컴포넌트입니다. 표준 컴포넌트로는 할 수 없는 다양한 기능을 추가할 수 있습니다.

AI 번역본
-For Reference Only-

3.3 운용까지의 흐름

SpeeDBee Synapse의 운용까지의 기본적인 흐름은, 다음과 같습니다.



①셋업・사용자의 설정

본 매뉴얼을 참고하여, SpeeDBee Synapse의 셋업을 수행합니다. 또한, 이용자의 권한에 따른 사용자를 설정합니다.

②데이터 수집 ~ 출력・연계까지의 플로우 검토

시스템의 목적에 따라, 실제로, 어떤 데이터를 수집할지, 사용할 기기의 선정・검토를 수행합니다. 또한, 모은 데이터를 최종적으로, 어떠한 형식으로 출력하여 활용할지를 설계합니다. 이용할 컬렉터, 시리얼라이저, 에미터 등 대응 기능을 확인합니다.
이 단계에서, 지원하고 있지 않은 프로토콜 등이 필요한 경우는, 커스텀 컴포넌트의 개발도 시야에 넣어 검토합니다.

-For Reference Only-

③각종 컴포넌트의 설정

검토한 플로우에 따라, 각종 컴포넌트의 설정을 수행합니다. 기본적인 흐름으로서는, 데이터 수집, 가공, 출력의 순서로 설정해 갑니다.

- 데이터 수집 설정（컬렉터）

데이터 수집의 설정을 수행합니다. 수집원의 기기나 그 설정이 올바른 것을 확인합니다. 그 후, 실제로 컴포넌트의 설정을 수행합니다.

- 출력 포맷의 설정（시리얼라이저）

데이터의 출력 형식의 설정을 수행합니다. CSV나 JSON 중 하나를 설정합니다. 또한, 출력 파일을 어떠한 단위로 분할할지 등도 여기에서 설정합니다.

- 데이터 출력 설정（에미터）

출력한 데이터를 최종적으로 어떻게 처리할지를 설정합니다. 예를 들어, File 에미터에서는, 출력 파일을 스토리지에 저장하기 위해, 그 저장 위치의 설정 등을 수행합니다. 외부로의 데이터 송신을 수행하는 경우는, 외부 접속에 필요한 정보를 확인하여 설정을 수행합니다.

④동작 확인・설정 튜닝

설정한 각 컴포넌트가 올바르게 동작하고 있는지 확인합니다. 정상적으로 데이터 수집이 되고 있는 것, 지정한 형식으로의 데이터 출력 및 송신이 되고 있는 것을 확인합니다. 의도하지 않은 동작 등이 있으면 설정을 재검토하여, 설정 변경, 재확인을 수행합니다.

⑤운용・모니터링・가시화

실제로 시스템을 운용합니다. 데이터 활용 엔진이 제공하는 기능을 사용하여, 데이터의 모니터링이나 그래프의 가시화를 할 수 있습니다. 운용 중에 문제가 생긴 경우는, 재검토를 수행하여 설정 튜닝을 다시 실시하십시오.

이 밖에, 목적에 따라, 실시간 분석이나 이벤트 연계 등의 설정을 수행함으로써, SpeeDBee Synapse를 활용한 시스템을 구축할 수 있습니다.

AI 번역본
-For Reference Only-

4 도입 플로우

AI 번역본
-For Reference Only-

4.1 도입 플로우

본 장에서는, SpeeDBee Synapse를 도입하기 위한 순서를 설명합니다. 도입

플로우로서, 다음 흐름으로 시스템을 구축합니다.

1. 설치（Linux）
2. 설치（Windows）
3. 초기 설정
4. 컴포넌트의 설정
5. 수집 서비스의 실행

여기에서 설명하는 시스템은, 매우 심플한 것이지만, SpeeDBee Synapse의 기본을 이해하기 위해 한번 읽어 보실 것을 권장합니다.

AI 번역본
-For Reference Only-

4.2 설치（Linux）

여기에서는, SpeeDBee Synapse（Linux）의 설치/제거 방법을 설명합니다.

4.2.1 실행 환경

본 패키지의 동작 환경은, Ubuntu20(AMD64)/Ubuntu22(AMD64, ARM64)입니다.
위 이외의 환경에서는, 해당 환경의 매뉴얼을 확인하십시오.

> Data Flow Controller판은 Synapse가 초기 상태로 설치되어 있기 때문에, 이 장의 내용을 수행할 필요는 없습니다.

4.2.2 설치 및 업데이트

아래 순서로, 디바이스에 SpeeDBee Synapse를 설치할 수 있습니다. 업데이트 시에도 동일한 순서가 됩니다.

파일을 배치

1. SpeeDBee Synapse의 파일을, 디바이스의 /tmp 디렉터리에 배치합니다. 아래는, Windows PowerShell에서 scp 커맨드로 배치하는 예입니다.

AI 번역본
-For Reference Only-

디바이스의 OS가 Ubuntu20(amd64)인 경우

```
PS >scp speedbeesynapse_ubuntu20_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(amd64)인 경우

```
PS >scp speedbeesynapse_ubuntu22_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(arm64)인 경우

```
PS >scp speedbeesynapse_ubuntu22_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Raspberry Pi OS(bookworm 64-bit)인 경우

```
PS >scp speedbeesynapse_raspi_bookworm_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 debian12(bookworm arm64)인 경우

```
PS >scp speedbeesynapse_debian12_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

AI 번역본
-For Reference Only-

설치 및 기동

1. ssh를 사용하여 디바이스에 로그인합니다. 아래는 Windows PowerShell에서 ssh 커맨드로 로그인하는 예입니다.

```
PS >ssh (계정명)@(디바이스의 IP 주소)
```

2. 아래 커맨드를 실행하여, SpeeDBee Synapse를 설치합니다.

디바이스의 OS가 Ubuntu20(amd64)인 경우

```
$ sudo apt install /tmp/speedbeesynapse_ubuntu20_x.x.x-r0_amd64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(amd64)인 경우

```
$ sudo apt install /tmp/speedbeesynapse_ubuntu22_x.x.x-r0_amd64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(arm64)인 경우

```
$ sudo apt install /tmp/speedbeesynapse_ubuntu22_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Raspberry Pi OS(bookworm 64-bit)인 경우

```
$ sudo apt install /tmp/speedbeesynapse_raspi_bookworm_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 debian12(bookworm arm64)인 경우

```
$ sudo apt install /tmp/speedbeesynapse_debian12_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

프록시 경유로의 설치

사내에 프록시 서버（포워드 프록시）를 배치하고 있는 환경에서의 설치는, 프록시로의 접속 정보를 명시적으로 설정할 필요가 있습니다. 아래 커맨드와 같이 환경 변수 http_proxy , https_proxy 를 지정하십시오.

```
$ sudo http_proxy=<프록시 서버의 url> https_proxy=<프록시 서버의 url> apt install ...
```

※Basic 인증의 사용자명・패스워드를 지정하는 경우는, 이하와 같이 url에 포함시키십시오.

```
http://<사용자명>:<패스워드>@<호스트>:<포트>
```

3. 아래 커맨드를 실행하여, SpeeDBee Synapse를 기동합니다.

```
$ sudo systemctl start speedbeesynapse
```

브라우저에서 액세스

1. 브라우저에서 아래 URL에 액세스함으로써, SpeeDBee Synapse의 화면을 열 수 있습니다.
   - `http://{호스트명or디바이스의 IP 주소}:8120`

     예）`http://192.168.1.3:8120`

     > ⚠ SpeeDBee Synapse에 액세스하는 PC에 대해
     >
     > 설치한 디바이스와 동일한 네트워크에 속하는 PC에서, 브라우저（Google Chrome/Microsoft Edge）로 SpeeDBee Synapse의 URL에 액세스해 주십시오.
     > 호스트명으로 액세스하는 경우, PC가 호스트명으로 액세스할 수 있는 환경 설정이 필요합니다.
     > ※일부 iPad 단말（Safari 사용）에서도 동작 확인을 수행하고 있으며, 기본적인 조작이 가능합니다.

     > ⚠ 업데이트 후의 조작에 대해
     >
     > SpeeDBee Synapse의 업데이트를 화면을 표시한 채로 실시하고, 그대로 조작을 계속하면 정상적으로 동작하지 않을 가능성이 있습니다. 업데이트 후에 설정을 수행하는 경우는, 화면의 다시 읽기（리로드）를 수행해 주십시오. 업데이트 후에, 매뉴얼에 기재되어 있는 새로운 화면 항목이 표시되지 않는 등의 증상이 있는 경우는, 브라우저의 캐시를 모두 삭제하십시오.

2. http 경유이기 때문에, 첫 액세스 시에는 화면상에 보안에 관한 경고가 표시되는 경우가 있습니다.

> ✎ https 경유로의 액세스
>
> https 경유로 SpeeDBee Synapse에 액세스하는 것도 가능합니다. 상세는「HTTP/HTTPS 통신」을 참조하십시오.

AI 번역본
For Reference Only

4.2.3 SpeeDBee Synapse의 제어

SpeeDBee Synapse를 설치한 디바이스에 ssh를 사용하여 로그인하고, 아래의 각 커맨드를 실행함으로써, SpeeDBee Synapse를 제어할 수 있습니다.

- 기동

```
$ sudo systemctl start speedbeesynapse
```

- 재기동

```
$ sudo systemctl restart speedbeesynapse
```

- 정지

```
$ sudo systemctl stop speedbeesynapse
```

- 자동 기동의 유효화

  이 설정을 수행함으로써, 디바이스를 재기동한 경우에 자동으로 SpeeDBee Synapse가 기동하게 됩니다.

```
$ sudo systemctl enable speedbeesynapse
```

- 자동 기동의 무효화

  이 설정을 수행하면, 디바이스를 재기동했을 때 SpeeDBee Synapse는 자동으로는 기동하지 않습니다.

```
$ sudo systemctl disable speedbeesynapse
```

4.2.4 제거

아래 순서로, 디바이스에서 SpeeDBee Synapse를 제거할 수 있습니다.

1. ssh를 사용하여 디바이스에 로그인합니다. 아래는 Windows PowerShell에서 ssh 커맨드로 로그인하는 예입니다.

```
PS >ssh (계정명)@(디바이스의 IP 주소)
```

2. 아래 커맨드를 실행하여, SpeeDBee Synapse를 제거합니다.
   - 로그나 설정 파일류를 남기는 경우

```
$ sudo apt remove speedbeesynapse
```

   - 로그나 설정 파일류도 모두 삭제하는 경우

```
$ sudo apt purge speedbeesynapse
```

-For Reference Only-

다시 SpeeDBee Synapse를 설치하는 경우의 순서는,「 설치 」에 기재된 순서와 동일합니다.

제거 시에 로그나 설정 파일류를 남겼던 경우는, 재설치 후에도 그 설정 내용이 인계됩니다.

4.3 설치（Windows）

여기에서는, SpeeDBee Synapse（Windows）의 설치/제거 방법을 설명합니다. 조작 내용은, Windows 11을 사용하여 기재하고 있습니다. OS 버전에 따라 사용하는 Windows 애플리케이션의 기동 방법이 다른 경우가 있습니다. OS 버전에 맞추어, 조작하십시오.

4.3.1 실행 환경

본 패키지의 동작 환경은, 다음과 같습니다.

- OS：Windows 10（64bit） version 22H2 이후 , Windows 11（64bit） version 23H2 이후
- 아키텍처：amd64（x86_64）

위 이외의 환경에서는, 해당 환경의 매뉴얼을 확인하십시오.

4.3.2 설치 및 업데이트

아래 순서로, PC에 SpeeDBee Synapse를 설치할 수 있습니다. 업데이트 시에도 동일한 순서가 됩니다.

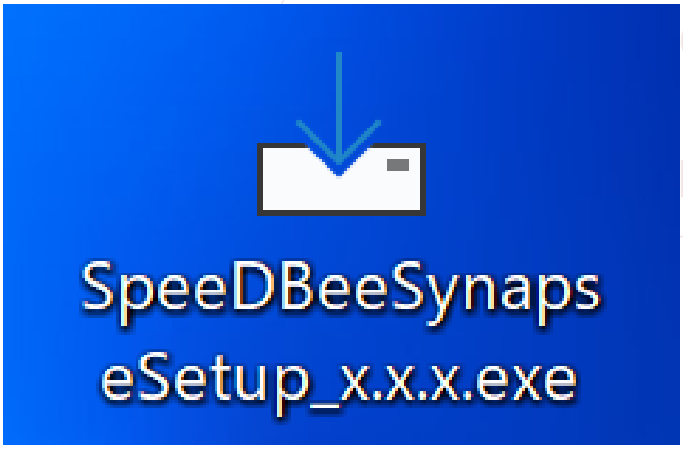
설치 준비

아래 설치 프로그램을 사용합니다. 사전에 설치 위치의 Windows PC의 데스크톱에 파일을 배치하십시오.

- SpeeDBeeSynapseSetup_x.x.x.exe （x.x.x 부분은 버전에 따라 다릅니다）

설치 및 기동

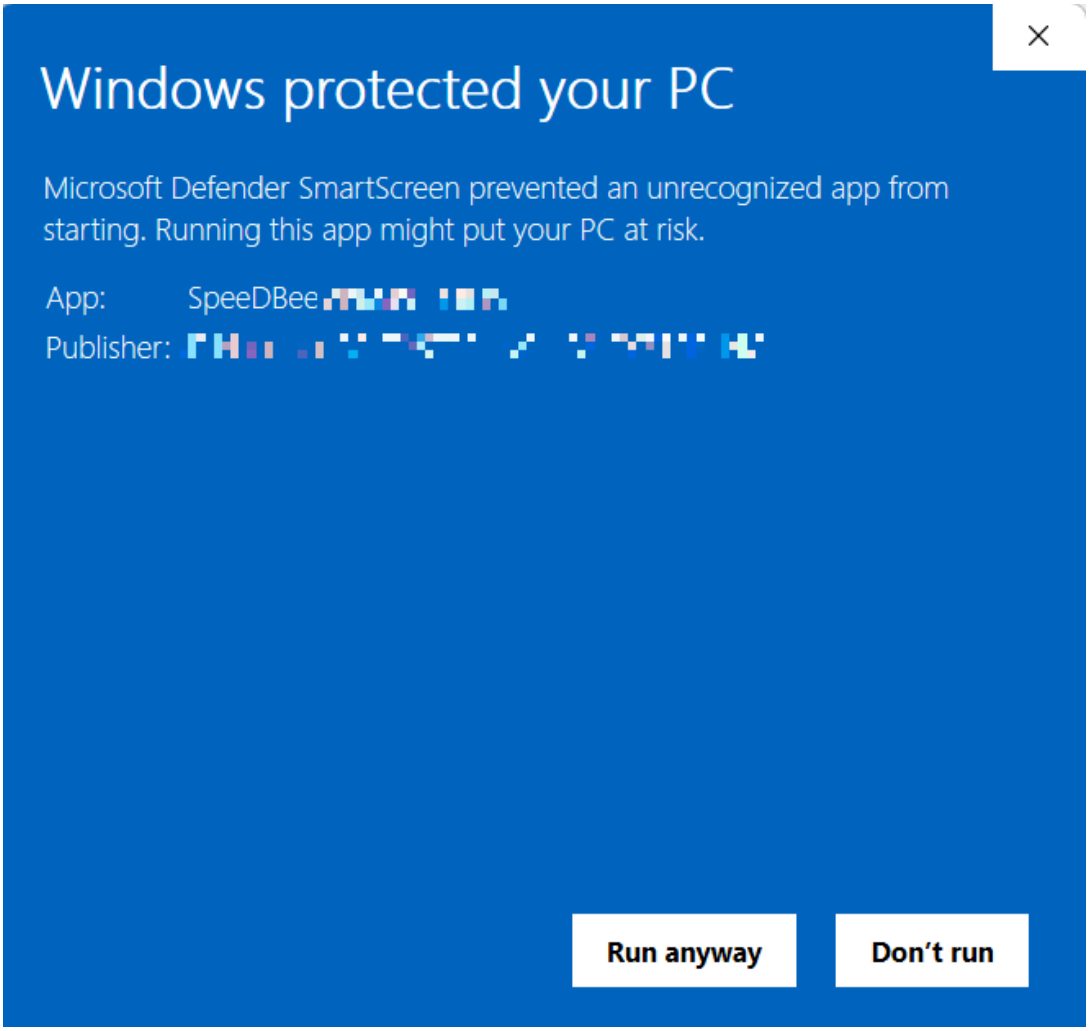
1. SpeeDBee Synapse의 설치 프로그램을 실행합니다.



역본
nce Only-

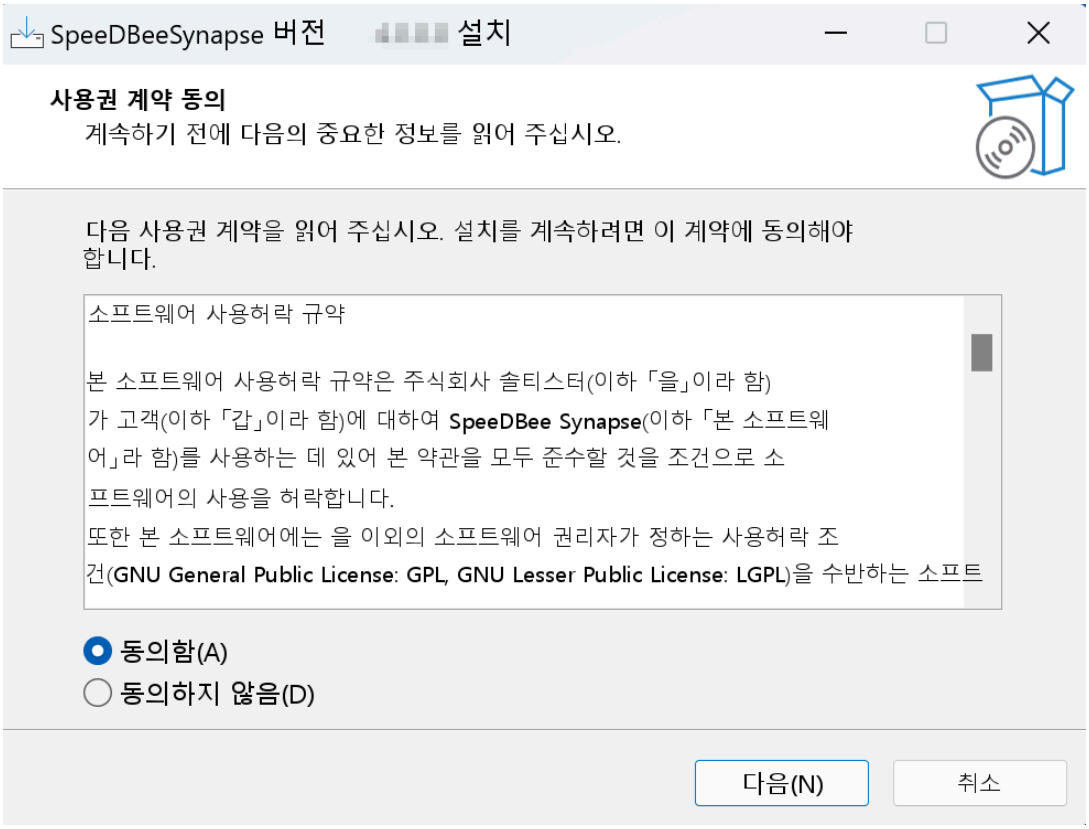
   - 설치 프로그램의 실행에는, Windows의 관리자 권한이 필요합니다

2. Microsoft Defender 가 유효한 경우에, 경고가 표시되는 경우가 있습니다. 발행원이 「 OMRON Corporation 」인 것을 확인하고, 「 실행 」을 클릭해 주십시오.

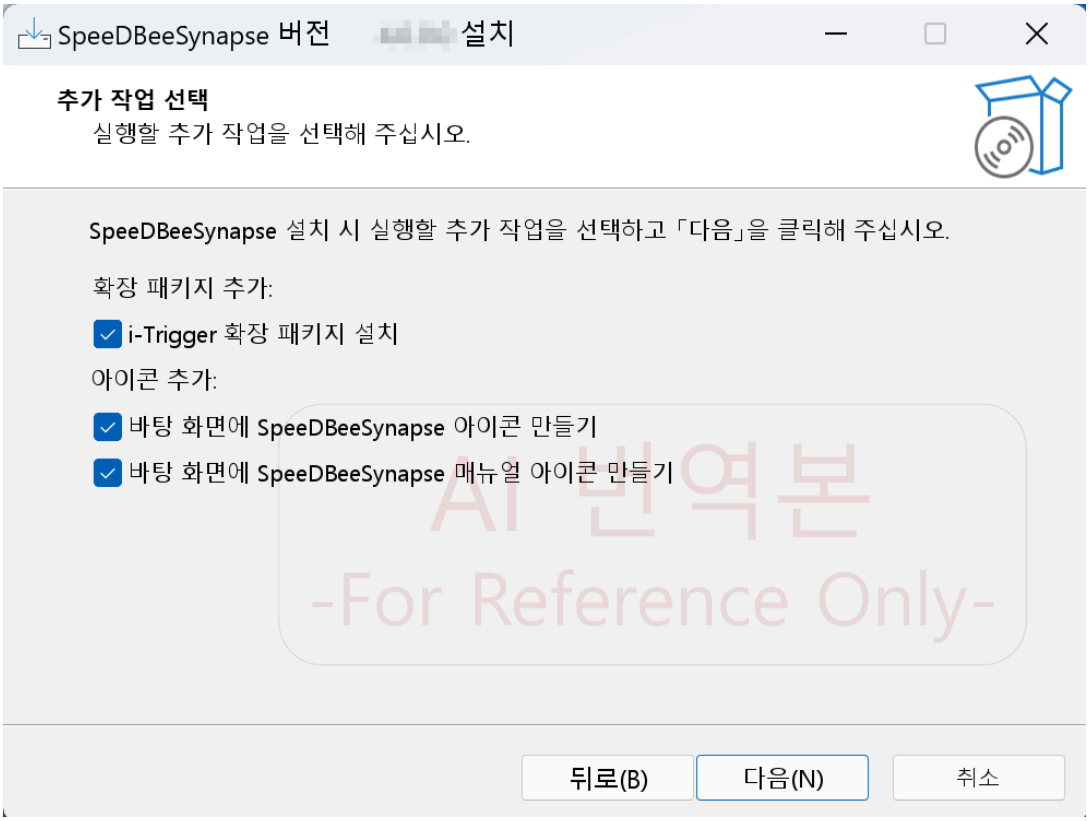


3. 「 소프트웨어 사용 허락 규약 」을 확인하고, 「 동의함 」, 「 다음 」을 클릭합니다.

AI 번역본
-For Reference Only-

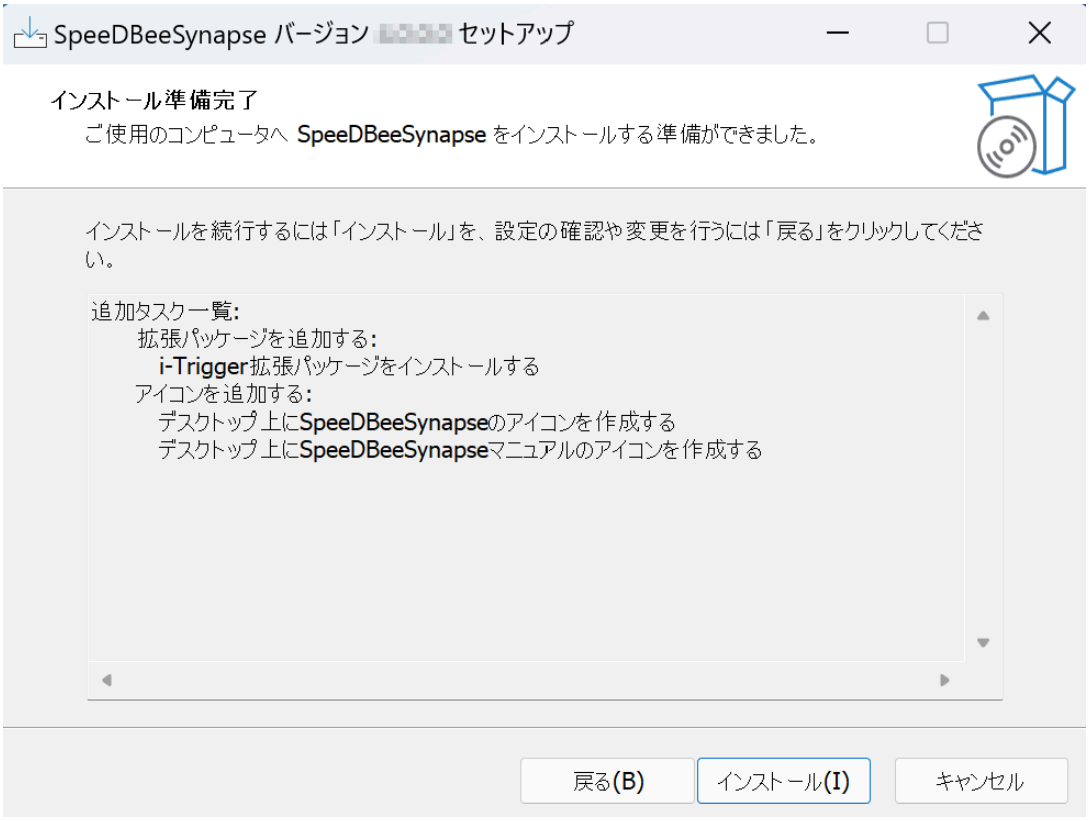


4. 추가 태스크의 선택이 표시됩니다. 「 다음 」을 클릭합니다.

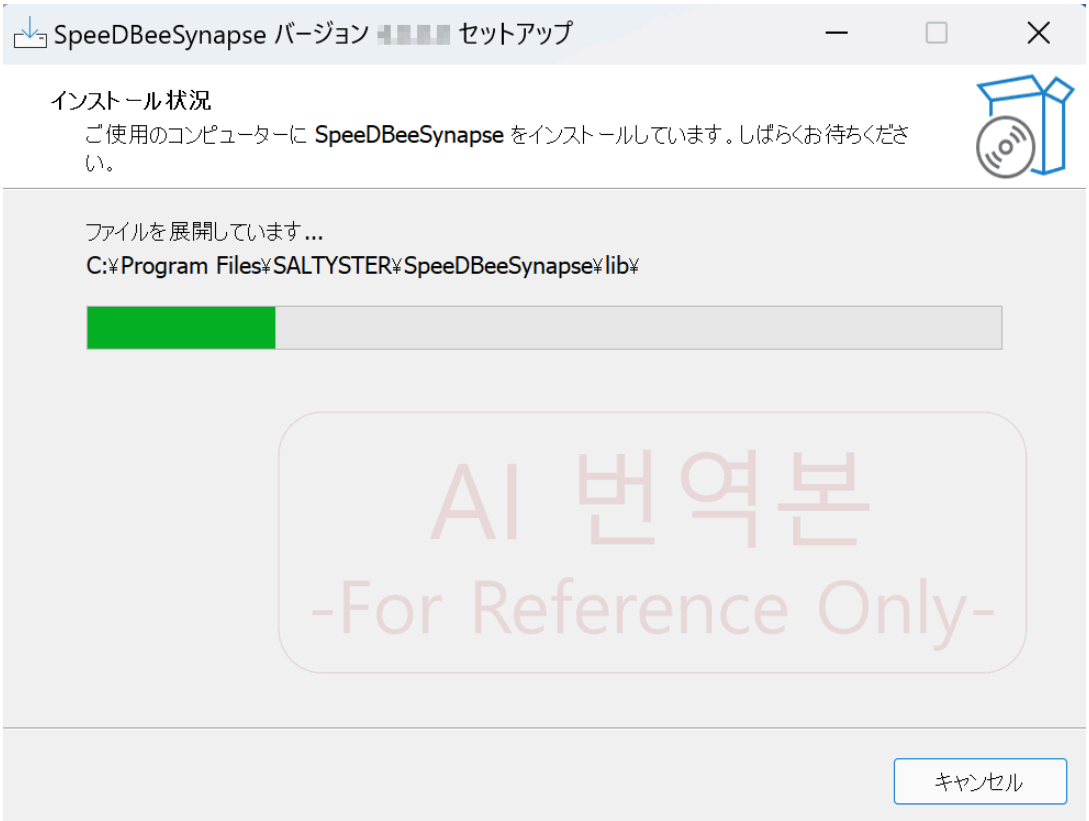


- i-Trigger (드라이브 레코더) 기능이 불필요하면 체크를 해제해 주십시오
- 불필요한 아이콘이 있으면 체크를 해제해 주십시오

5. 표시 내용을 확인하고, 「 설치 」를 실행합니다.



6. 설치가 진행됩니다.



7. 설치가 종료되면, 「 완료 」를 클릭합니다.



설치 후에 PC 재기동을 요구받는 경우가 있습니다. 그때는 재기동을 수행해 주십시오.

브라우저에서 액세스

1. 브라우저에서 아래 URL에 액세스함으로써, SpeeDBee Synapse의 화면을 열 수 있습니다.
   - http://{호스트명or디바이스의 IP 주소}:8120

     예) http://192.168.1.3:8120

> ⚠ SpeeDBee Synapse에 액세스하는 PC에 대해
>
> 설치한 PC, 또는 대상 PC와 동일한 네트워크에 속하는 PC에서, 브라우저 (Google Chrome/Microsoft Edge) 로 SpeeDBee Synapse의 URL에 액세스해 주십시오.
> 호스트명으로 액세스하는 경우, PC가 호스트명으로 액세스할 수 있는 환경 설정이 필요합니다.
> ※일부 iPad 단말 (Safari 사용) 에서도 동작 확인을 수행하고 있으며, 기본적인 조작이 가능합니다.

> ⚠ 업데이트 후의 조작에 대해
>
> SpeeDBee Synapse의 업데이트를 화면을 표시한 채로 실시하고, 그대로 조작을 계속하면 정상적으로 동작하지 않을 가능성이 있습니다. 업데이트 후에 설정을 수행하는 경우는, 화면의 다시 읽기 (리로드) 를 수행해 주십시오. 업데이트 후에, 매뉴얼에 기재되어 있는 새로운 화면 항목이 표시되지 않는 등의 증상이 있는 경우는, 브라우저의 캐시를 모두 삭제하십시오.

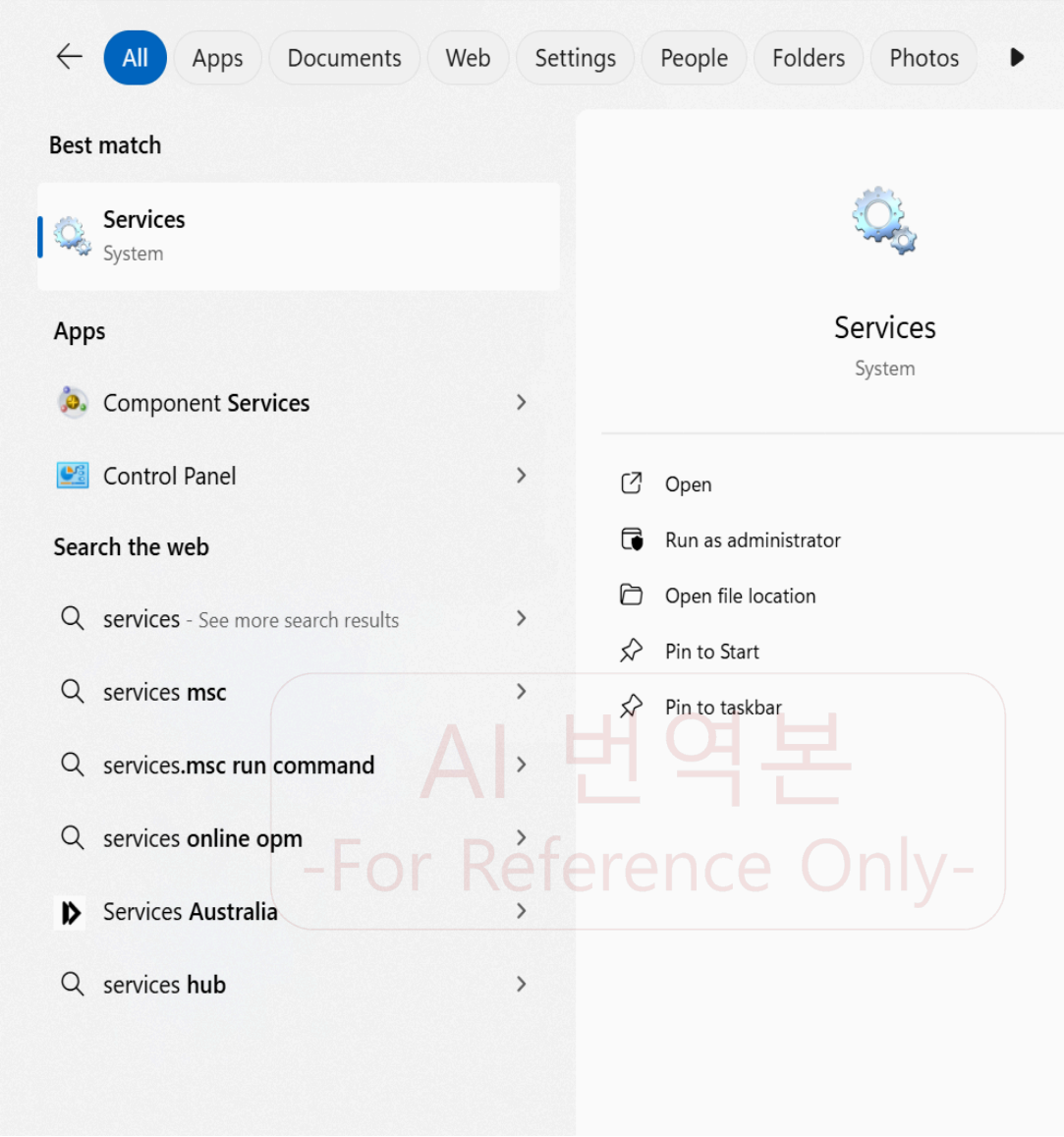
2. http 경유이기 때문에, 첫 액세스 시에는 화면상에 보안에 관한 경고가 표시되는 경우가 있습니다.

> **https 경유로의 액세스**
>
> https 경유로 SpeeDBee Synapse에 액세스하는 것도 가능합니다. 상세는 「HTTP/HTTPS 통신」을 참조하십시오.

4.3.3 SpeeDBee Synapse의 제어

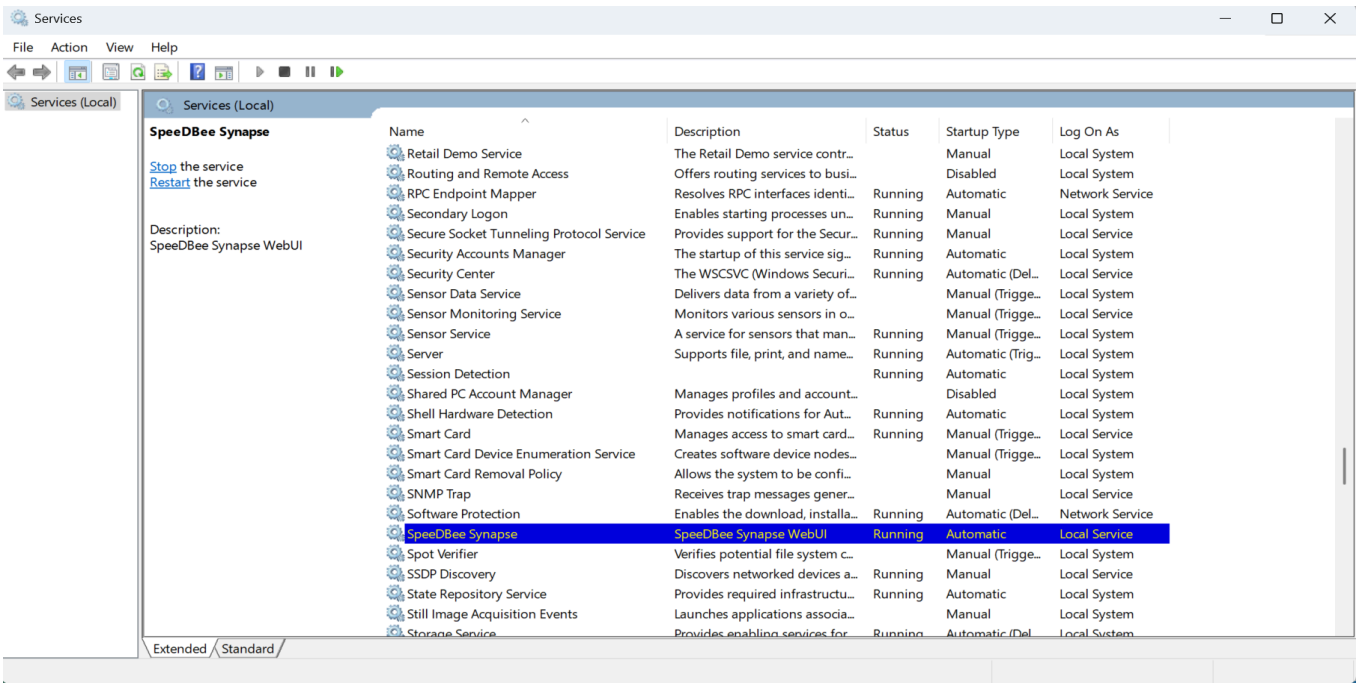
SpeeDBee Synapse는, 설치하면 Windows의 서비스로서 등록됩니다. 그 때문에, 각 제어는 Windows의 서비스에서 조작합니다.

1. 데스크톱 하부의 검색란에 「서비스」라고 입력하여, 검색 결과에서 서비스를 기동합니다.



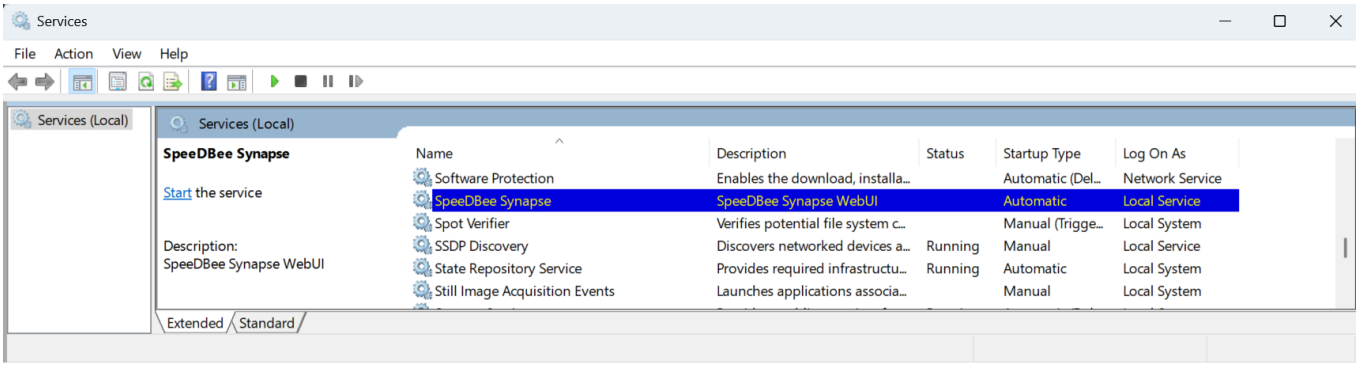
AI 번역본
-For Reference Only-

2. 「SpeeDBee Synapse」를 선택하여, 각종 제어를 수행합니다.

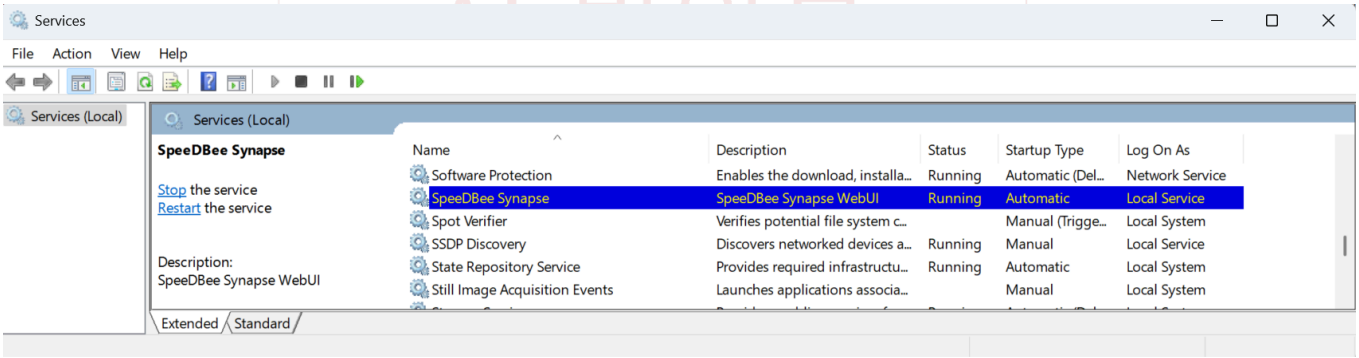


4.3.3.1 기동

1. SpeeDBee Synapse를 선택하여, 「 서비스의 개시 」를 클릭합니다.

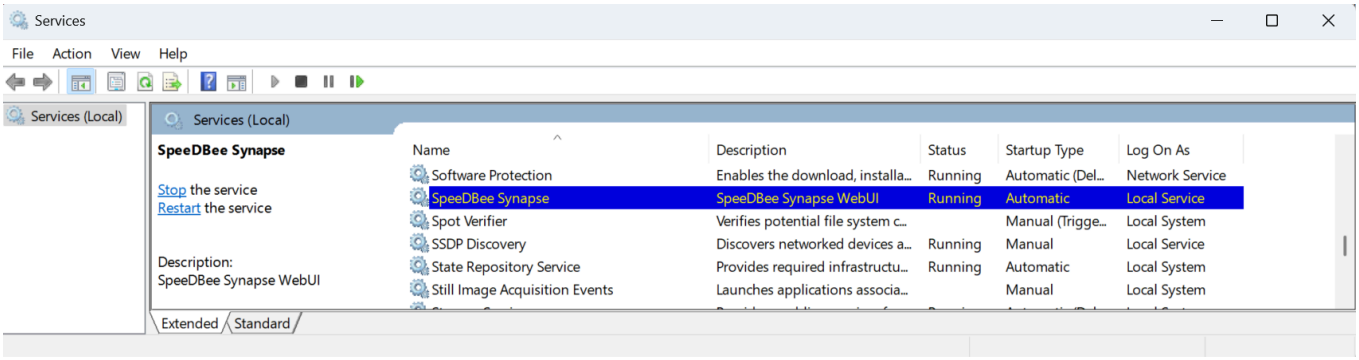


2. 서비스가 기동합니다. 정상적으로 기동하고 있으면, 상태가 「 실행 중 」이 됩니다.

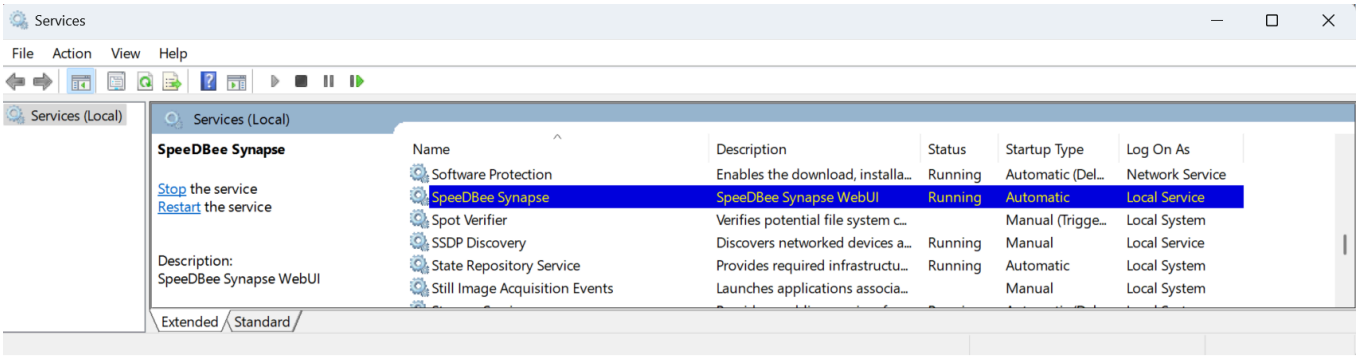


4.3.3.2 재기동

1. SpeeDBee Synapse를 선택하여, 「 서비스의 재기동 」을 클릭합니다.

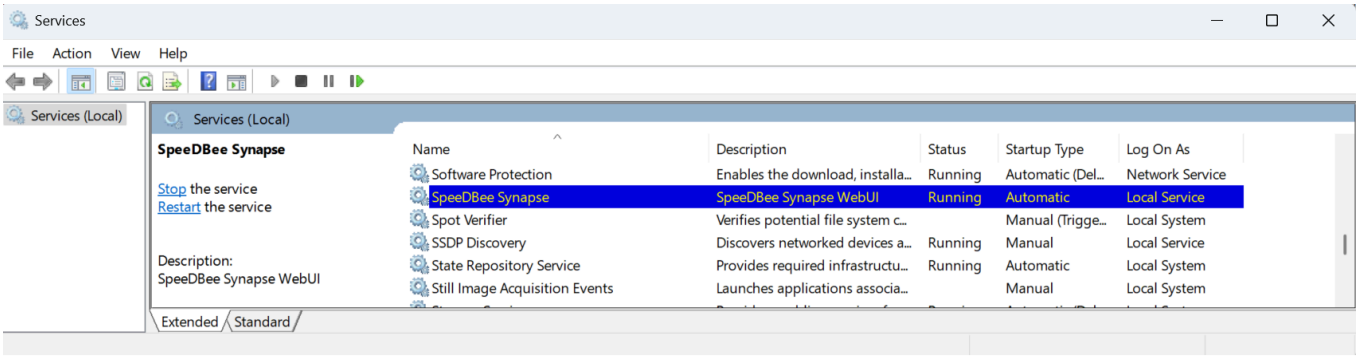


2. 서비스가 재기동합니다.

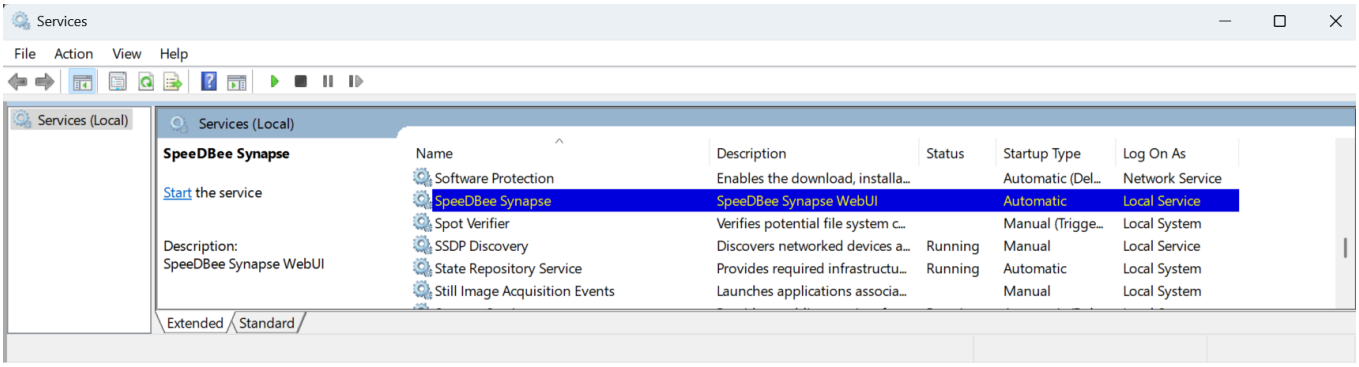


4.3.3.3 정지

1. SpeeDBee Synapse를 선택하여, 「 서비스의 정지 」를 클릭합니다.



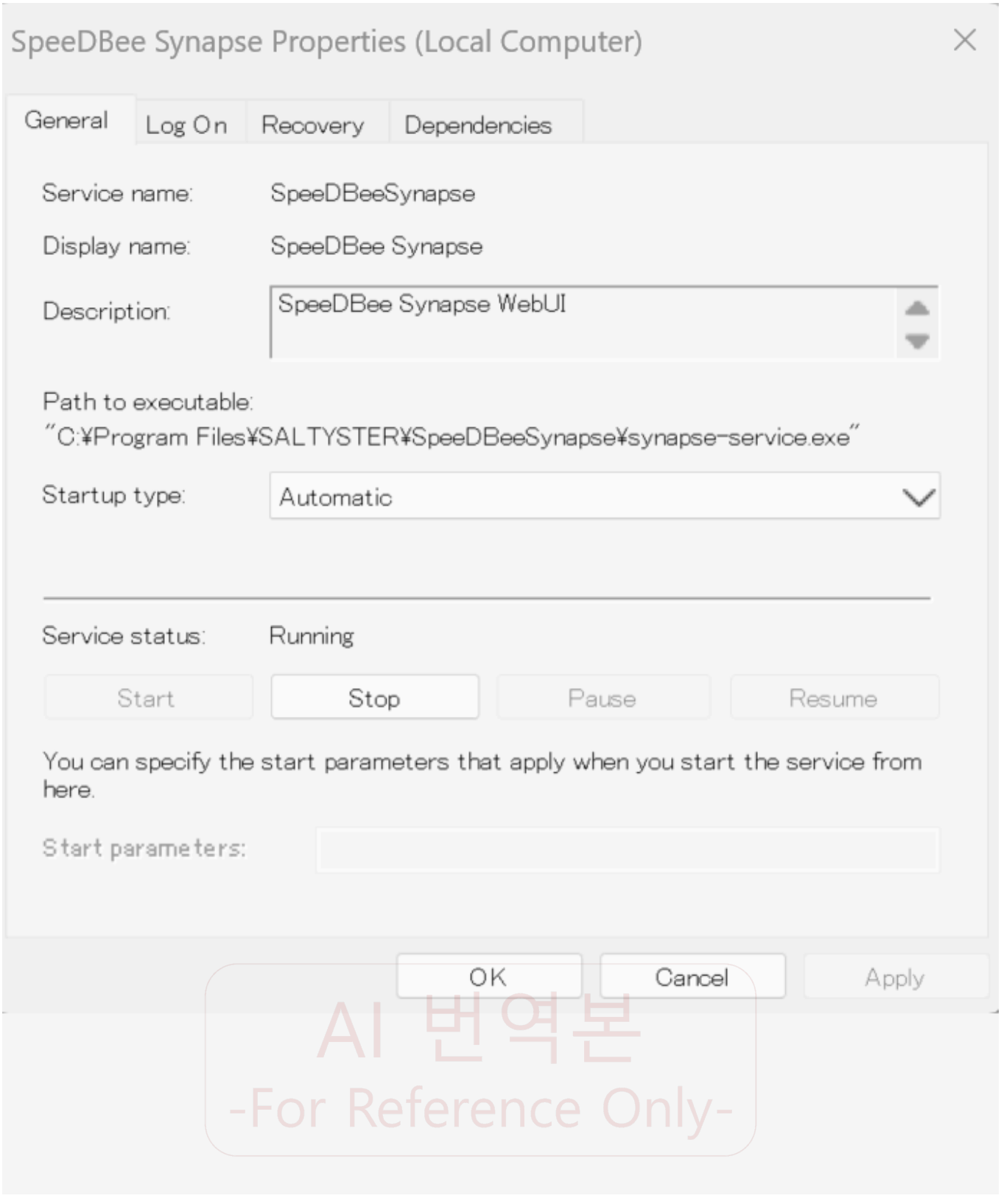
2. 서비스가 정지합니다.



4.3.3.4 자동 기동의 유효화

1. SpeeDBee Synapse를 선택하여, 오른쪽 클릭 메뉴에서「 프로퍼티 」를 클릭합니다.
2.「 스타트업의 종류 」를「 자동 」으로 설정하여,「 OK 」를 클릭합니다.

AI 번역본
-For Reference Only-

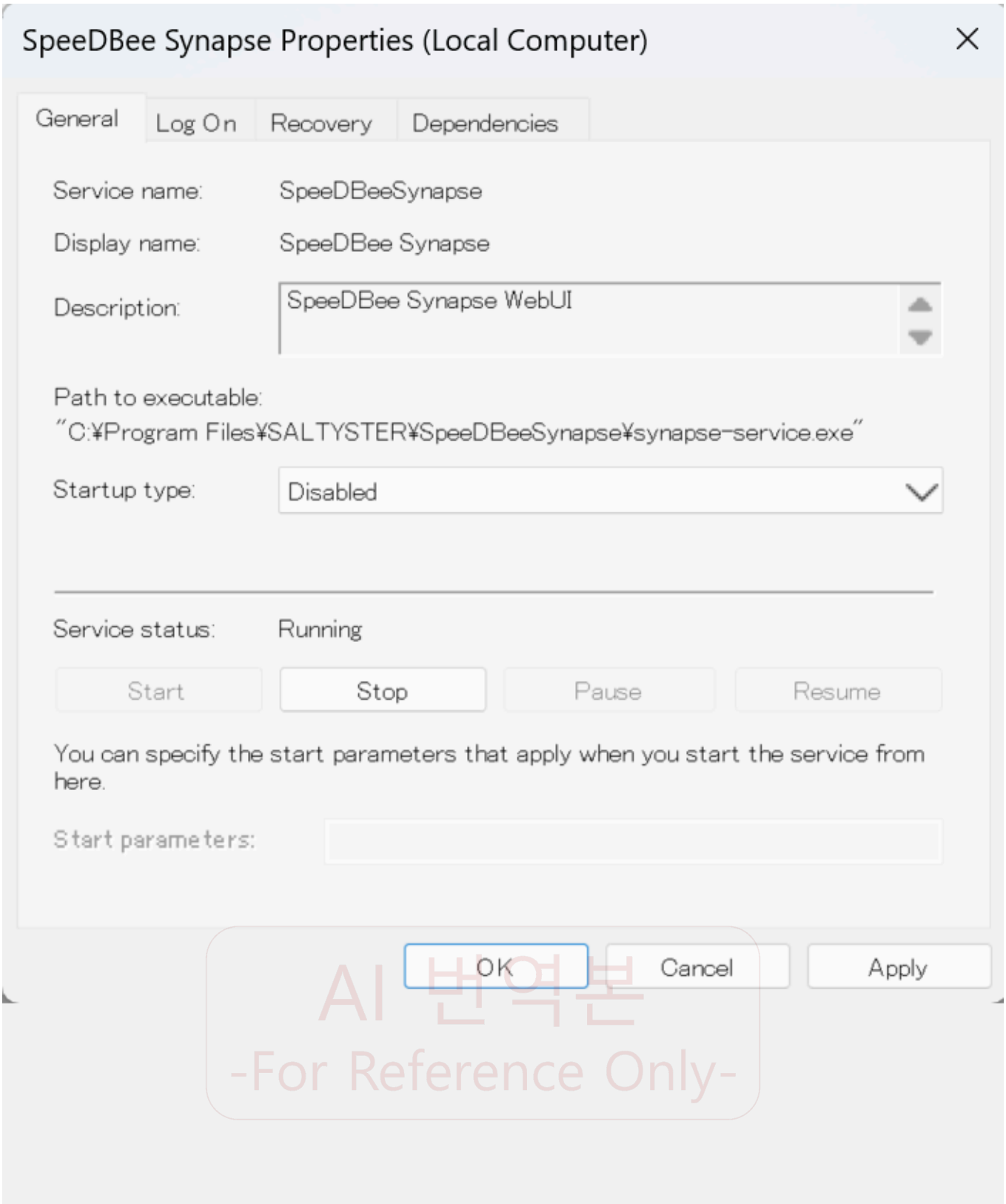


3. 이 설정에 의해 PC 기동 시, SpeeDBee Synapse의 서비스가 자동 기동하게 됩니다.

4.3.3.5 자동 기동의 무효화

1. SpeeDBee Synapse를 선택하여, 오른쪽 클릭 메뉴에서 「 프로퍼티 」를 클릭합니다.

2. 「 스타트업의 종류 」를 「 무효 」로 설정하여, 「 OK 」를 클릭합니다.

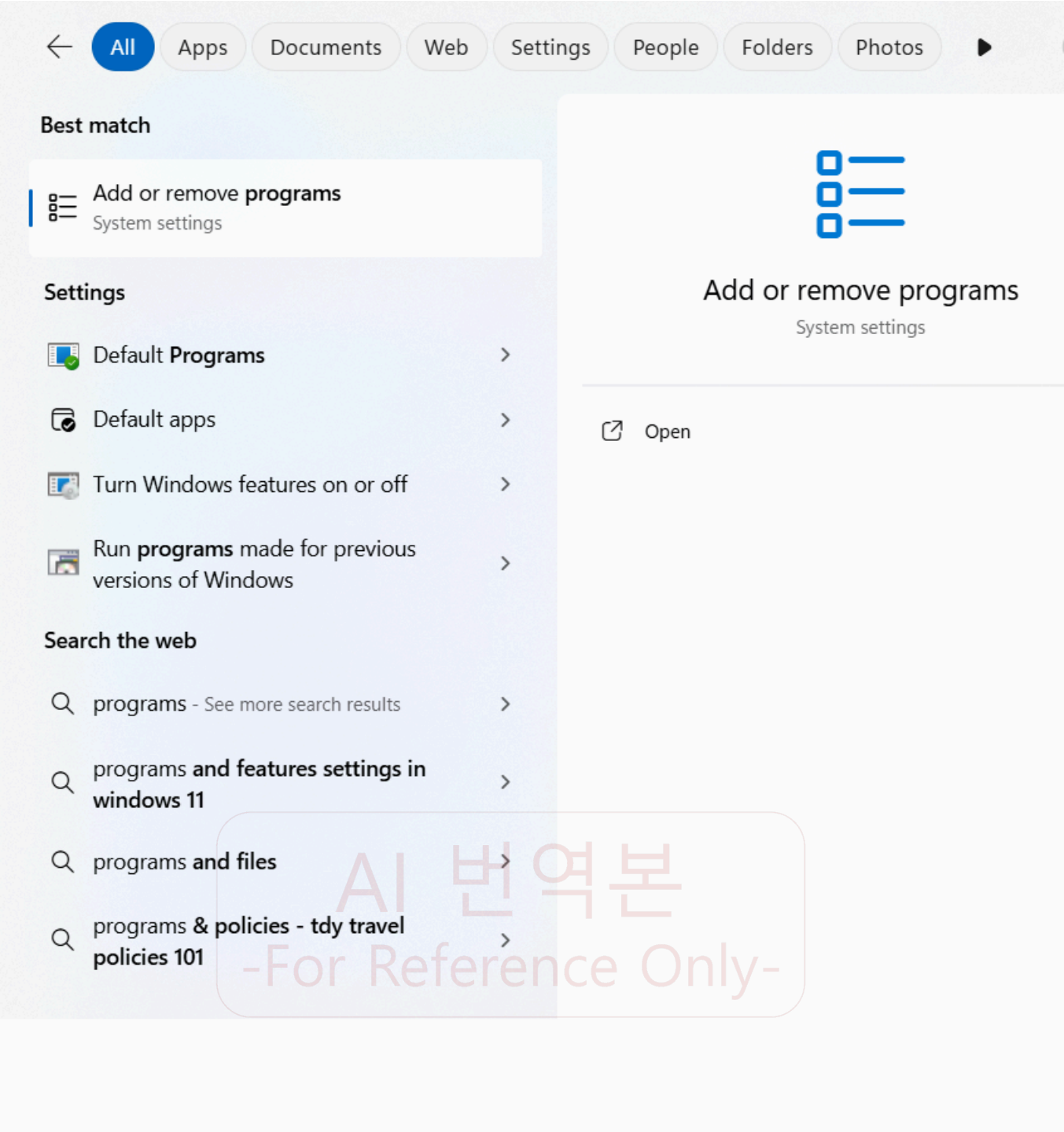


3. 이 설정을 적용하면, PC 기동 시, SpeeDBee Synapse의 서비스는 기동하지 않습니다. 기동하는 경우는, 수동으로 기동 조작을 수행할 필요가 있습니다.

4.3.4 제거

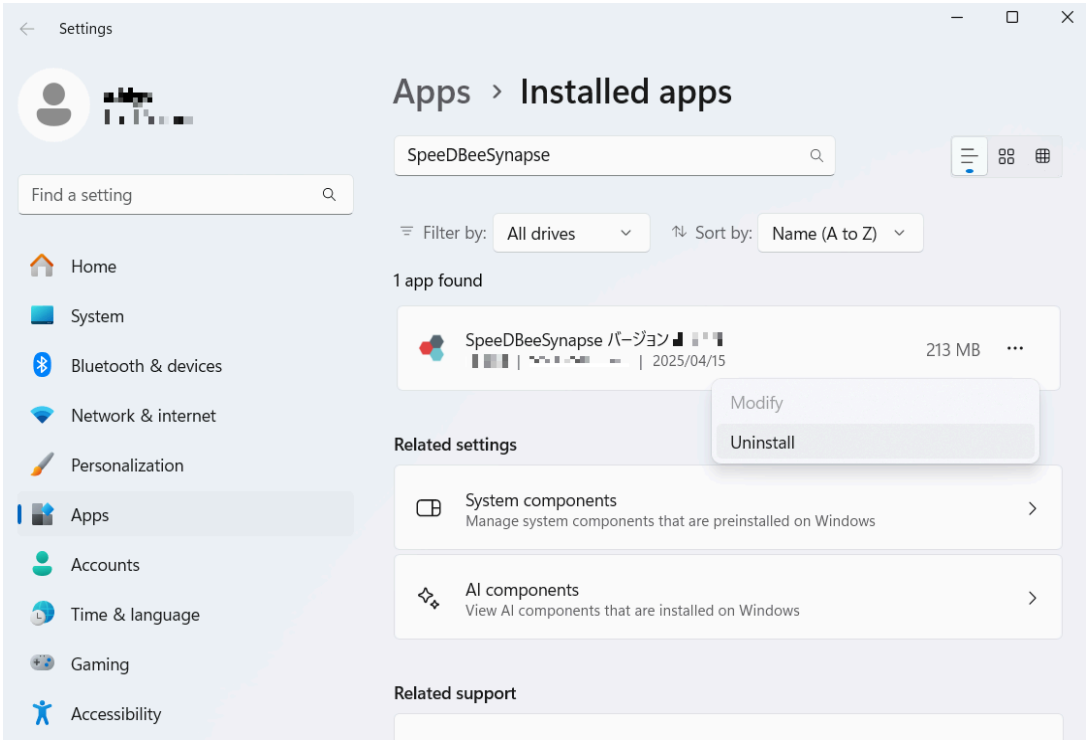
아래 순서로, 디바이스에서 SpeeDBee Synapse를 제거할 수 있습니다.

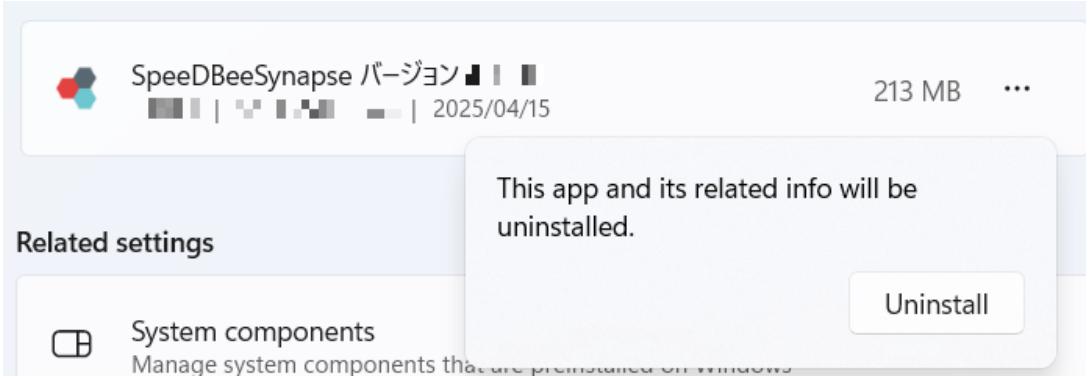
1. 데스크톱 하부의 검색란에 「 프로그램 추가 및 삭제 」라고 입력하여, 검색 결과에서 「 프로그램 추가 및 삭제 」를 기동합니다.



AI 번역본
-For Reference Only-

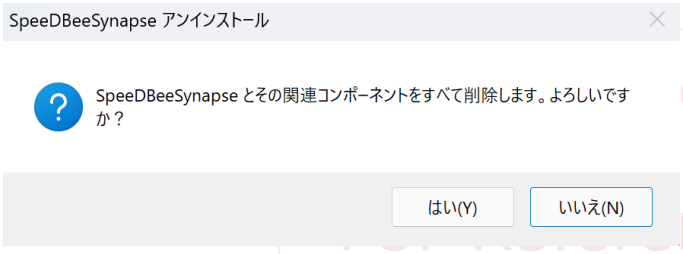
2. 설치되어 있는 앱의 검색란에 「 SpeeDBee Synapse 」라고 입력하여, 검색 결과에서 SpeeDBee Synapse의 메뉴에서 「 제거 」를 클릭합니다.





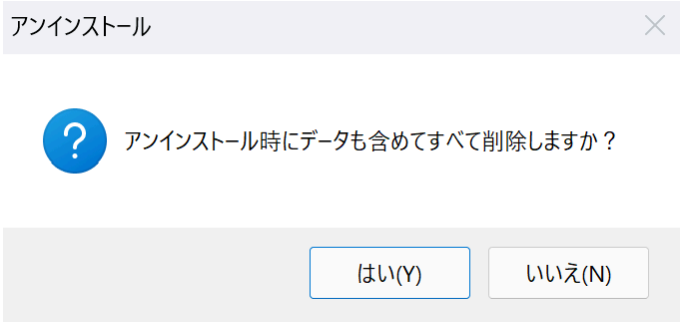
◦ 제거의 실행에는, Windows의 관리자 권한이 필요합니다

3. 확인 메시지가 표시되면,「 예 」를 클릭합니다.



역본
-For Reference Only-

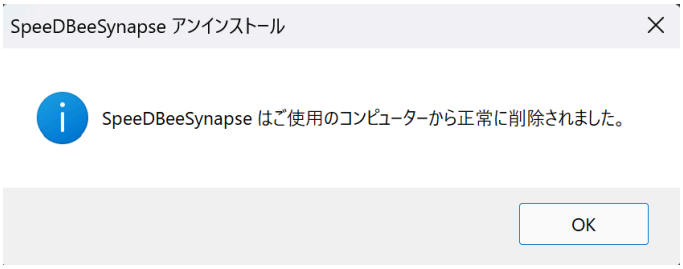
4. 이어서, 다음 메시지 내용을 확인합니다. 데이터（ ProgramData 내에 저장되는 정보 ）를 모두 삭제하는 경우는,「 예 」를 클릭합니다. 데이터를 남기는 경우는,「 아니오 」를 클릭해 주십시오.



5. 제거 처리가 실행됩니다.



6. 완료 후에,「 OK 」를 클릭합니다. 이것으로 SpeeDBee Synapse의 제거는 완료입니다.



AI 번역본
-For Reference Only-

4.4 초기 설정

여기에서는 초기 설정（라이선스의 등록, admin 사용자의 패스워드 변경）의 방법과, 사용자를 추가하는 방법을 설명합니다.

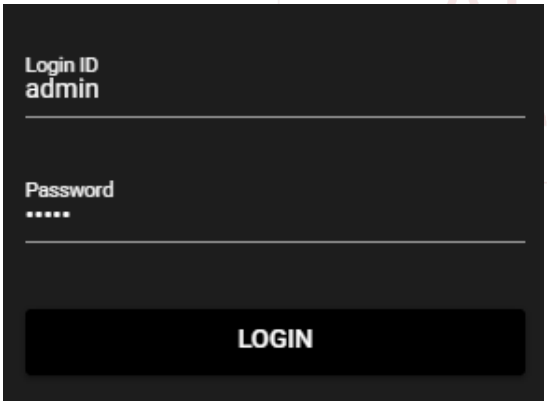
4.4.1 관리자 패스워드의 변경

초기 설정으로서, 라이선스를 등록하고, admin 사용자의 패스워드를 디폴트 설정값에서 변경합니다. admin 은, SpeeDBee Synapse에 디폴트로 등록되어 있는 사용자입니다. admin 사용자의 디폴트 설정값은, 이하와 같습니다.

항목	설명
로그인 ID	admin
사용자명	admin
권한	시스템 관리자
패스워드	admin
E메일 주소	admin@example.com

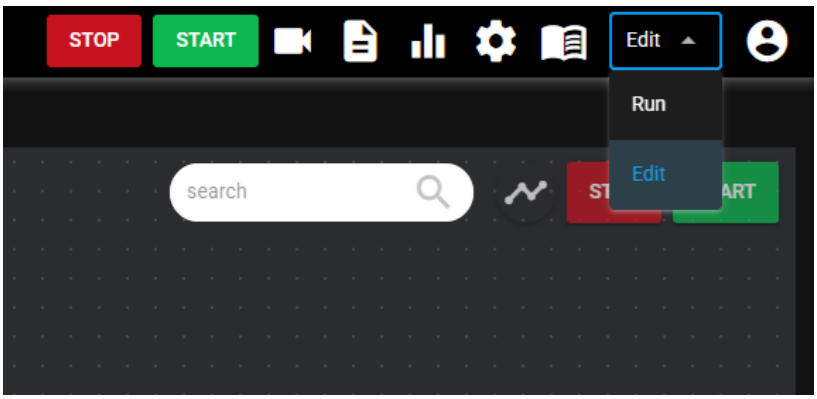
Data Flow Controller판, OMRON-SBS for Windows판에서는 초기 상태로 라이선스 등록 완료로 되어 있습니다. 그 경우, 순서 7부터 확인하십시오.

1. 로그인 화면에서, 로그인 ID와 패스워드에「admin」이라고 입력하고「로그인」을 누릅니다.



2. 헤더의 모드 선택에서 "Edit Mode"를 선택하고, Edit 모드의 화면을 엽니다.

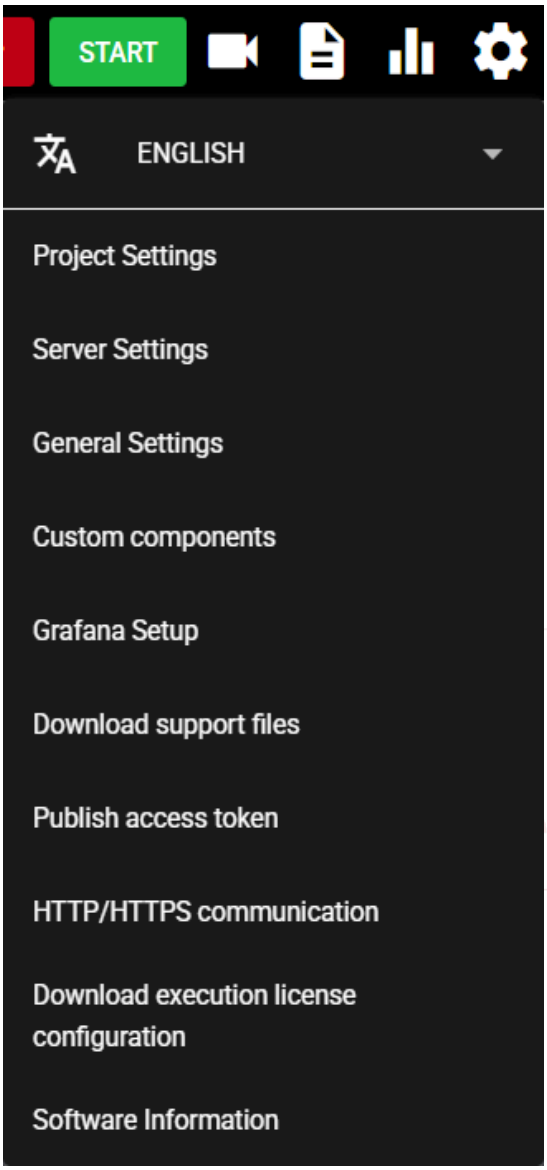
AI 번역본
-Reference Only-



3. 헤더의 설정 메뉴 아이콘을 누릅니다.

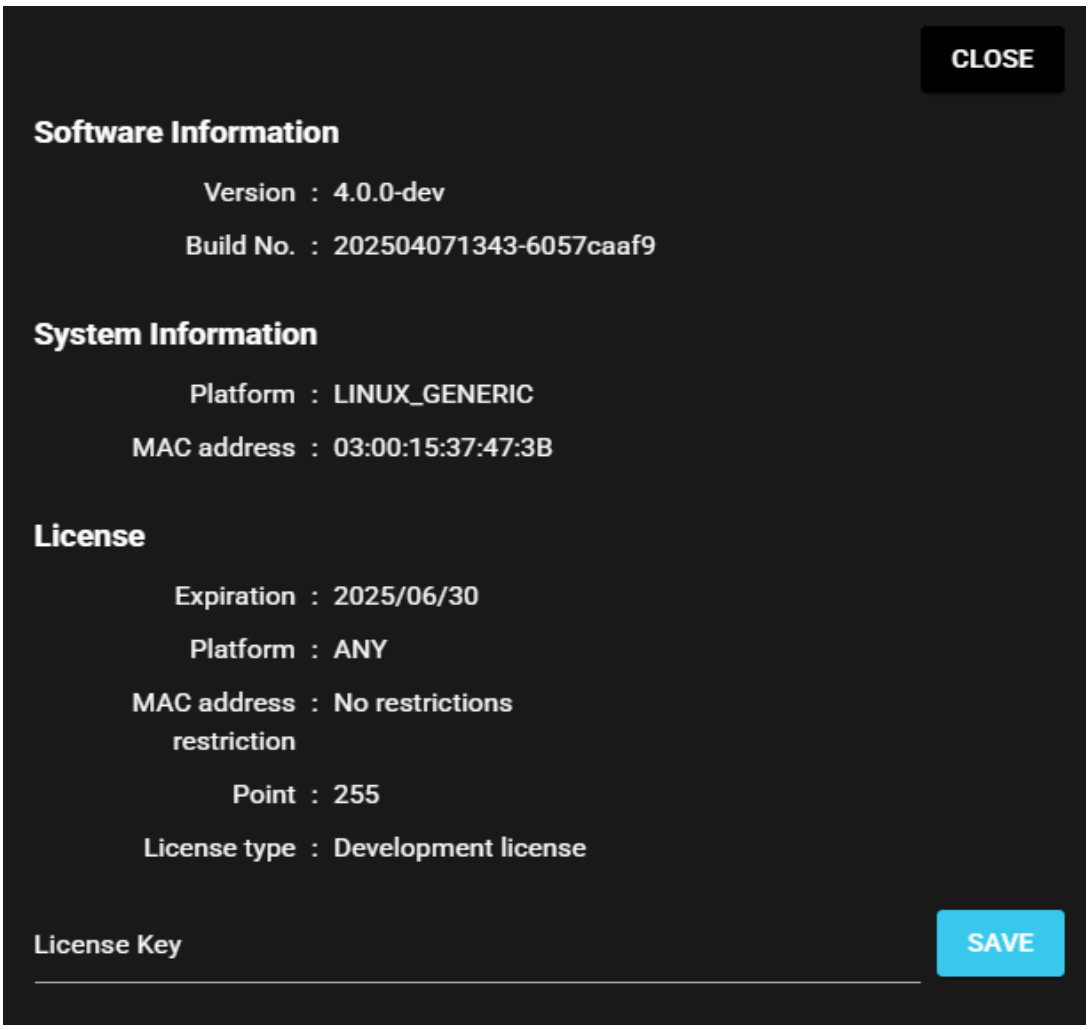


4.「 소프트웨어 정보 」를 선택합니다.

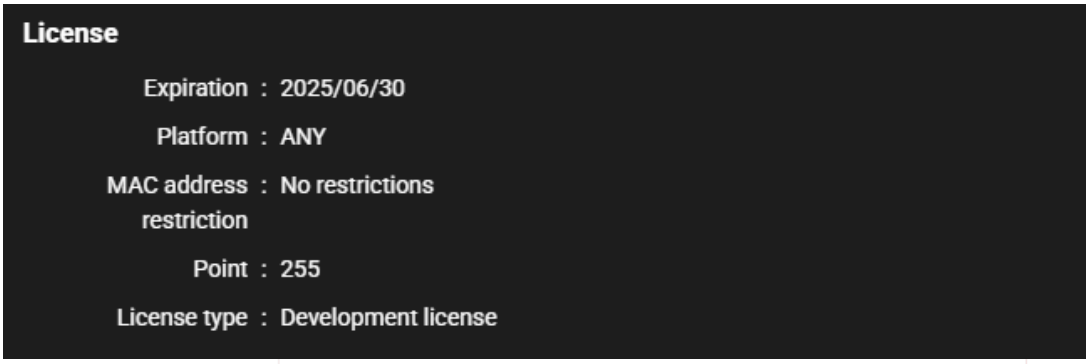


번역본
ference Only-

5. 라이선스 키 입력란에 라이선스 키를 입력하고,「 저장 」을 누릅니다.

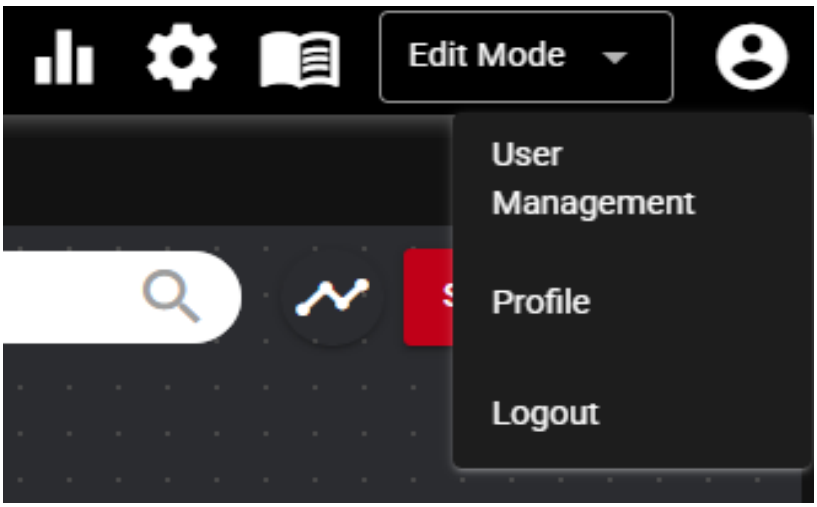


6. 올바른 라이선스 키가 입력되면, 입력된 라이선스의 정보가 표시됩니다.

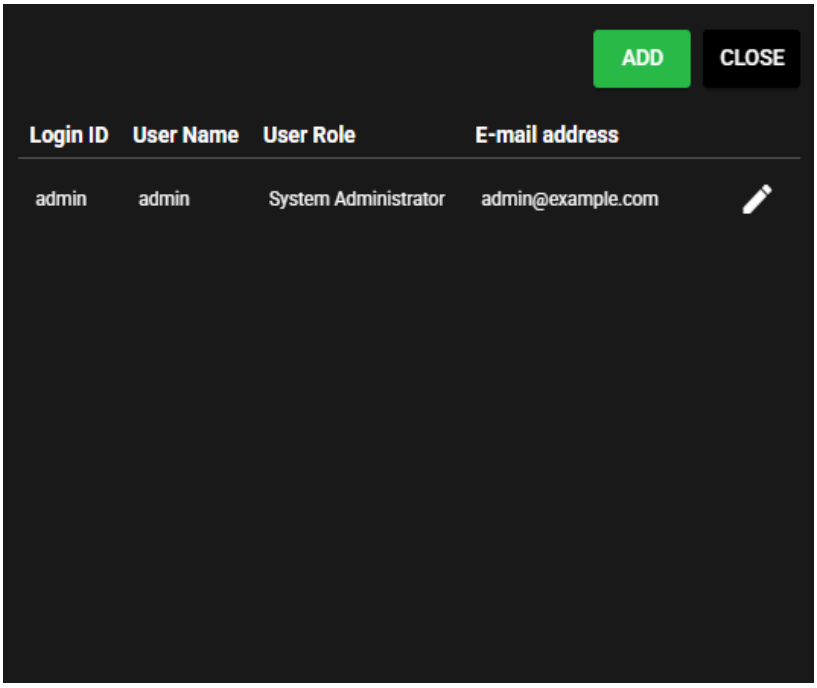


7. 다음으로, 사용자 아이콘을 누르고「사용자 관리」를 선택하여, 사용자 관리 다이얼로그를 엽니다.

For Reference Only

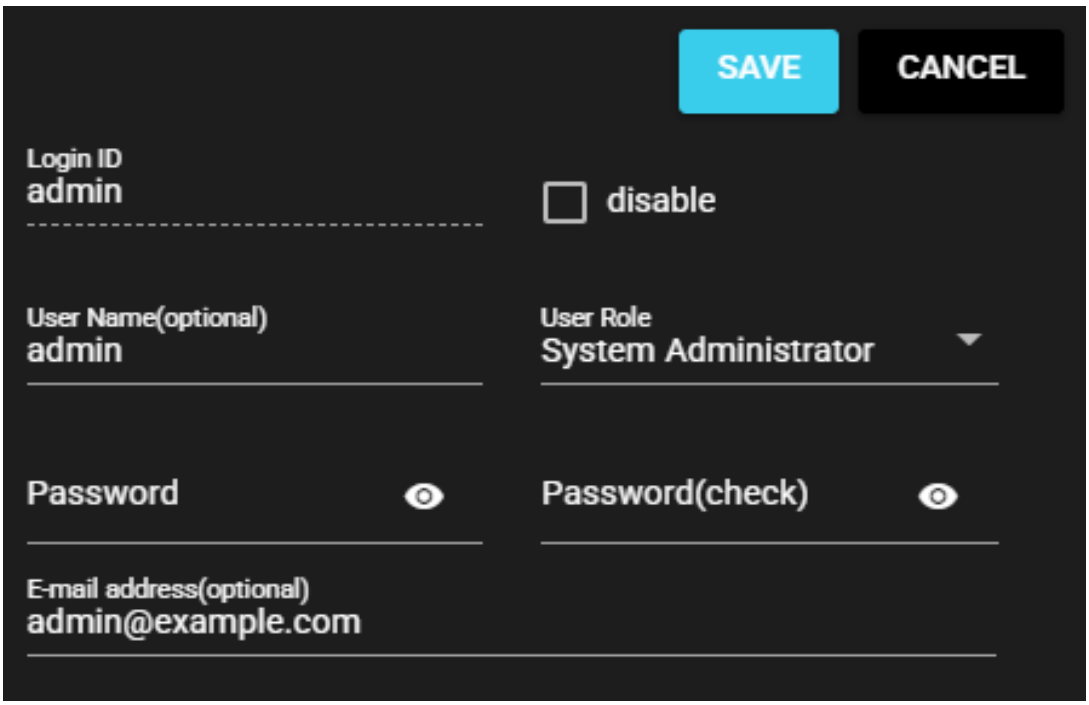


8. "admin" 사용자의 편집 아이콘을 누릅니다.



9. 「 패스워드 」와 「 패스워드(확인) 」에, "admin"과는 다른 임의의 패스워드를 입력하고, 「 저장 」을 누릅니다.

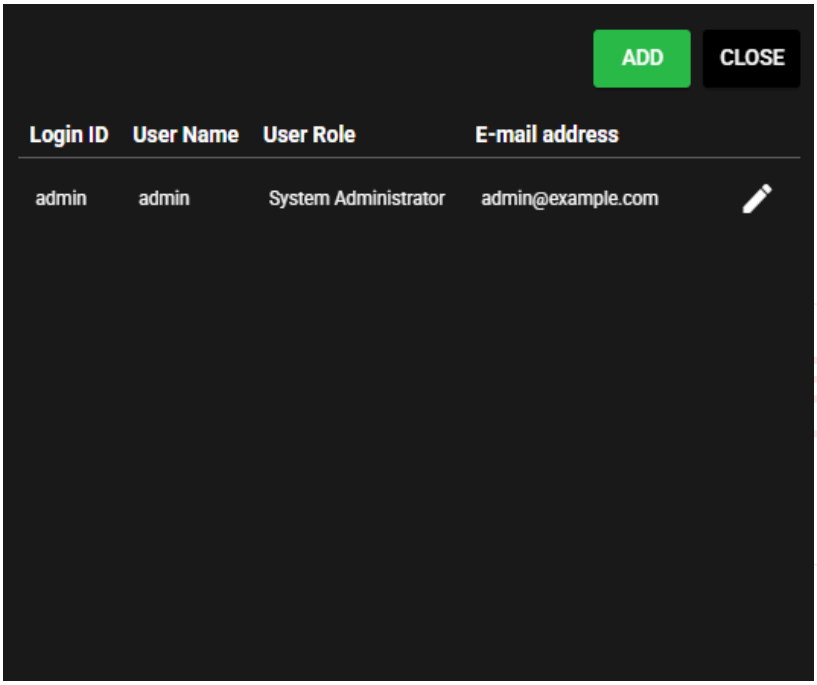
AI 번역본
-For Reference Only-



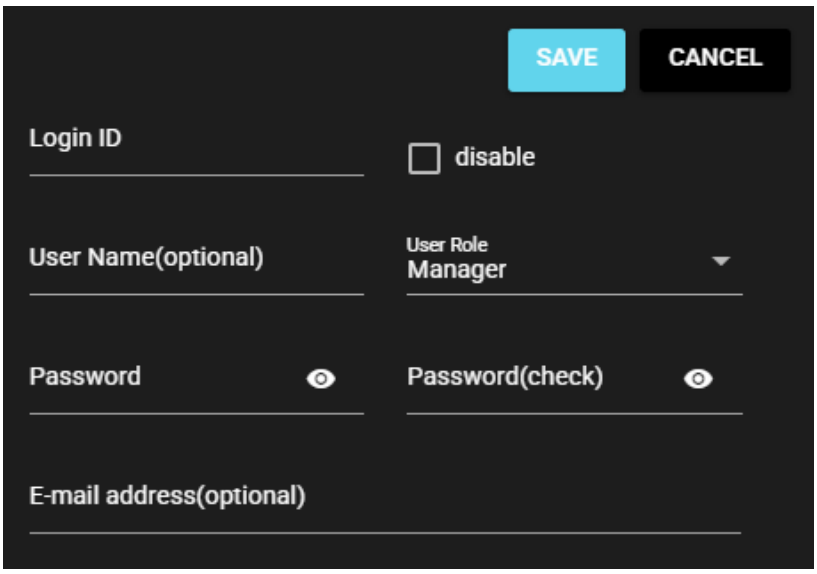
4.4.2 사용자의 추가

다음 순서로 사용자를 추가할 수 있습니다.

1. 사용자 관리 다이얼로그에서,「 추가 」버튼을 누릅니다.



2. 이하의 항목을 입력합니다. 권한은「 관리자 」를 선택합니다. 입력이 완료되면,「 저장 」버튼을 누릅니다.

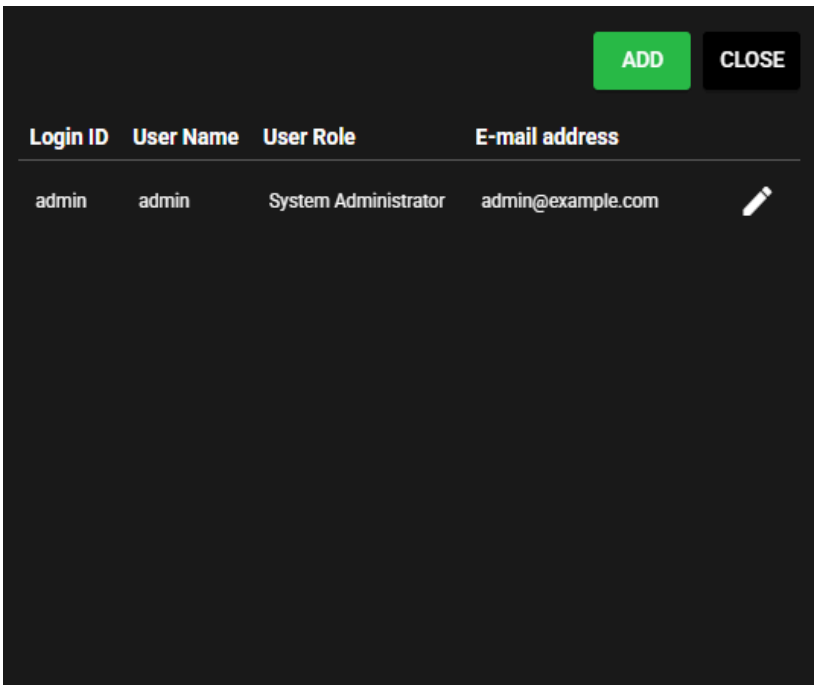


항목	설명
무효로 한다	체크를 붙이면 사용자가 무효가 되어 로그인을 할 수 없게 됩니다.
로그인 ID	로그인 시에 입력하는 ID입니다.
사용자명	사용자의 이름입니다. ※입력을 생략하는 것이 가능합니다.
권한	아래 표 참조
패스워드	로그인 시에 입력하는 패스워드입니다.
E메일 주소	사용자에 연결하는 E메일 주소입니다. ※입력을 생략하는 것이 가능합니다.

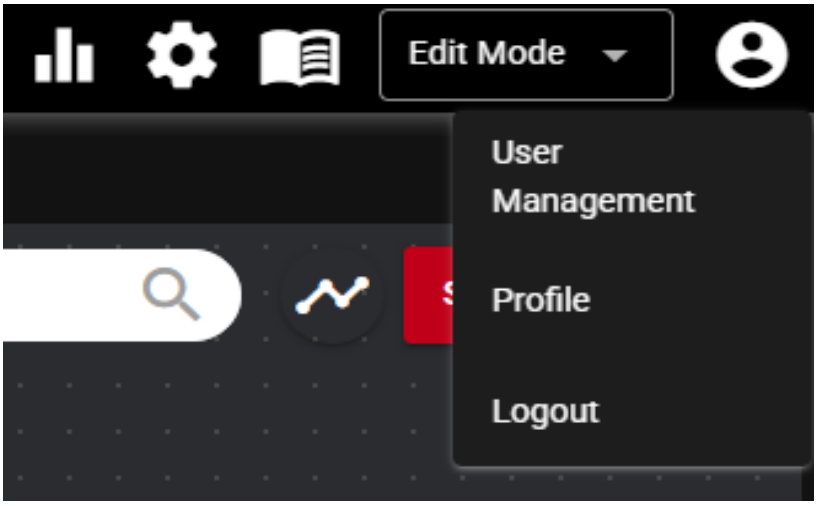
사용자의 권한은 4종류가 있습니다.

권한	설명
시스템 관리자	모든 조작이 가능합니다. 초기 사용자 "admin"에게 부여되어 있으며, 1명 이상 필수입니다.
관리자	모든 조작이 가능합니다. 시스템 관리자의 추가・갱신・삭제는 불가입니다.
오퍼레이터	Run Mode의 열람, 및 컴포넌트의 기동・정지가 가능합니다.
열람자	Run Mode의 열람이 가능합니다.

3. 사용자 관리 다이얼로그의「닫기」를 누릅니다.

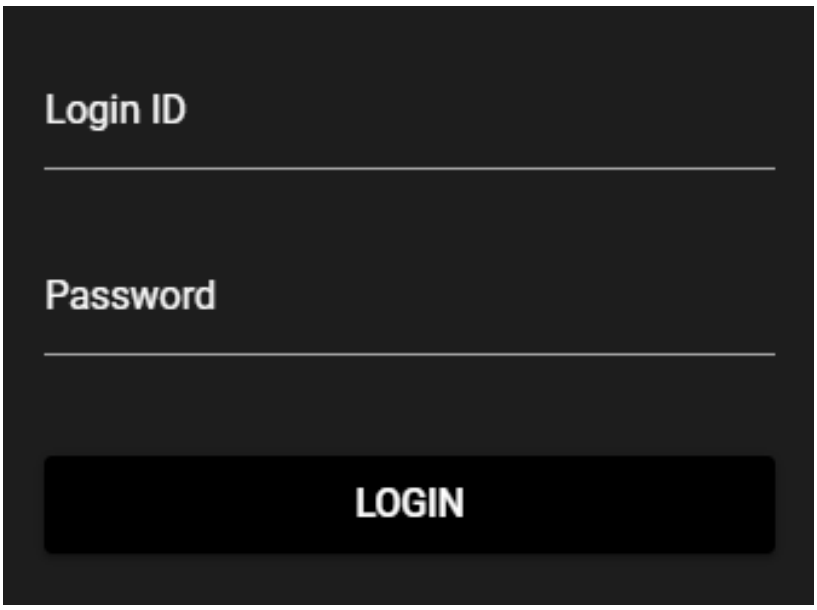


4. 사용자 아이콘을 누르고,「 로그아웃 」을 선택합니다.

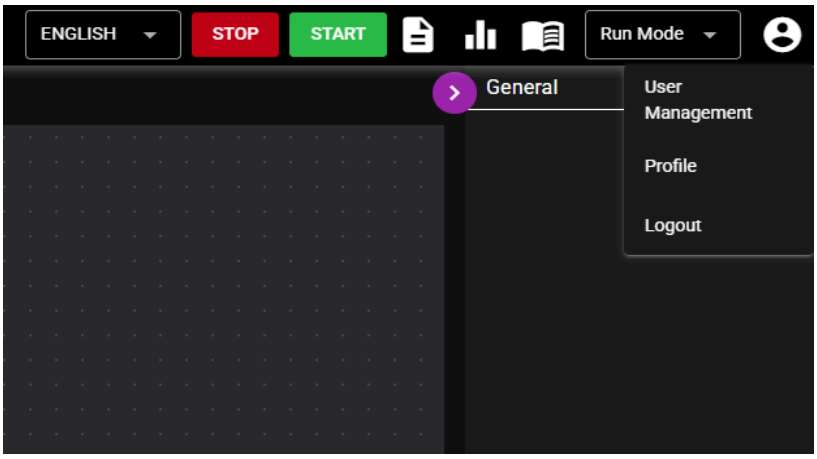


5. 로그인 화면에서, 새롭게 추가한 사용자의 로그인 ID와 패스워드를 입력하고,「 로그인 」을 누릅니다. 이 조작은, 해당하는 사용자 본인이 조작합니다.

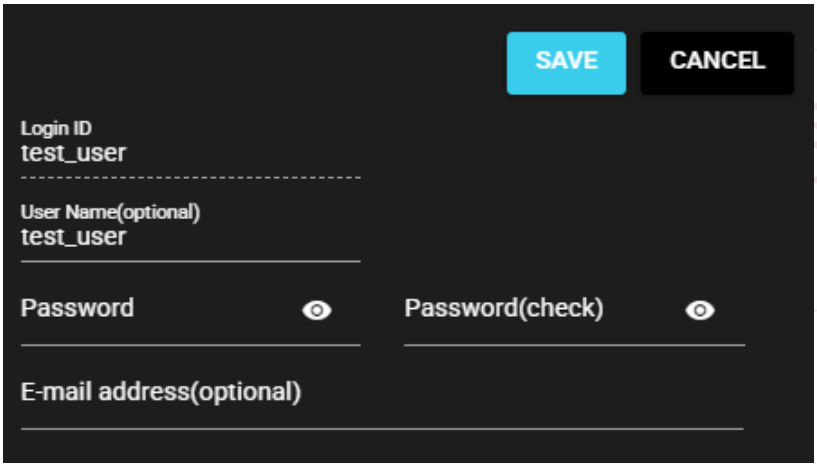
AI 번역본
-For Reference Only-



6. 사용자 아이콘을 누르고,「 프로필 」을 선택합니다.

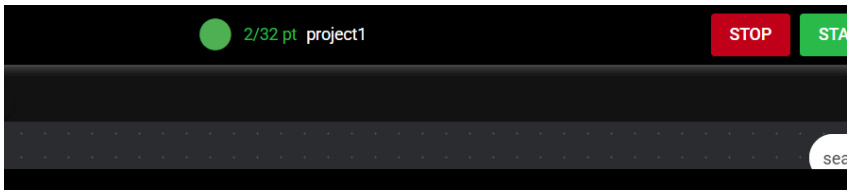


7. 현재 로그인하고 있는 사용자의 정보가 표시되어, 위에서 추가한 사용자로 로그인하고 있는 것을 알 수 있습니다. 설정값을 변경하고「 저장 」을 누르면, 현재 로그인하고 있는 사용자의 설정을 변경할 수 있습니다.



4.4.3 프로젝트명의 설정

SpeeDBee Synapse는, 설정한 데이터 플로우를 프로젝트라는 단위로 관리합니다. 설정한 데이터 플로우를 판별하기 쉽게 하기 위해 프로젝트명을 설정하십시오.
（예：「라인A 생산 정보 가시화」,「설비 연계 시스템」등 임의로 설정）
초기는 project1이라는 이름으로 설정되어 있으며, 프로젝트명은 헤더에 표시됩니다.



설정 방법은, 이하를 참고하십시오.

설정 정보의 관리(프로젝트)

AI 번역본
-For Reference Only-

4.5 컴포넌트의 설정

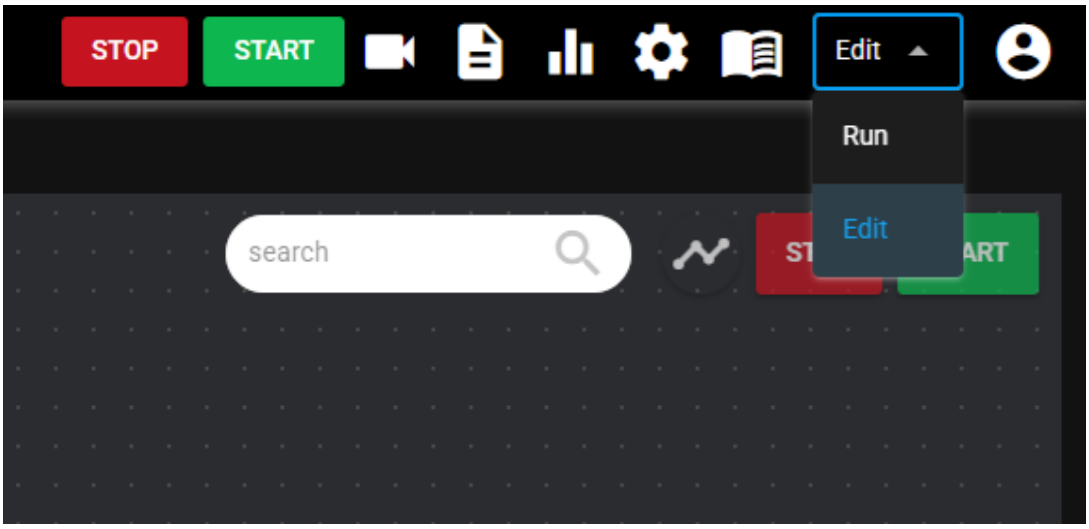
여기에서는, 수집한 데이터를 파일에 저장하기까지의 설정 방법을 설명합니다.

4.5.1 리소스 컬렉터의 설정

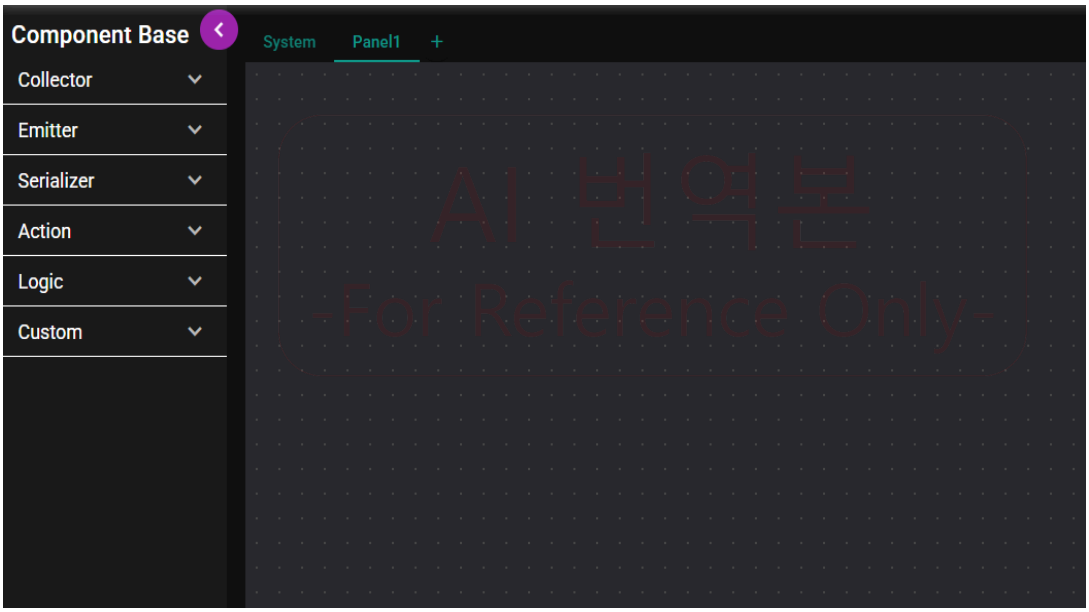
가동 머신의 CPU 사용률을 수집하기 위해, "리소스 컬렉터"를 설정하는 방법을 설명합니다. "리소스 컬렉터"는, 가동 머신의 시스템 리소스（CPU 사용률, 메모리 소비량 등）를 수집하는 컴포넌트입니다.

다음 순서로, 리소스 컬렉터를 설정할 수 있습니다.

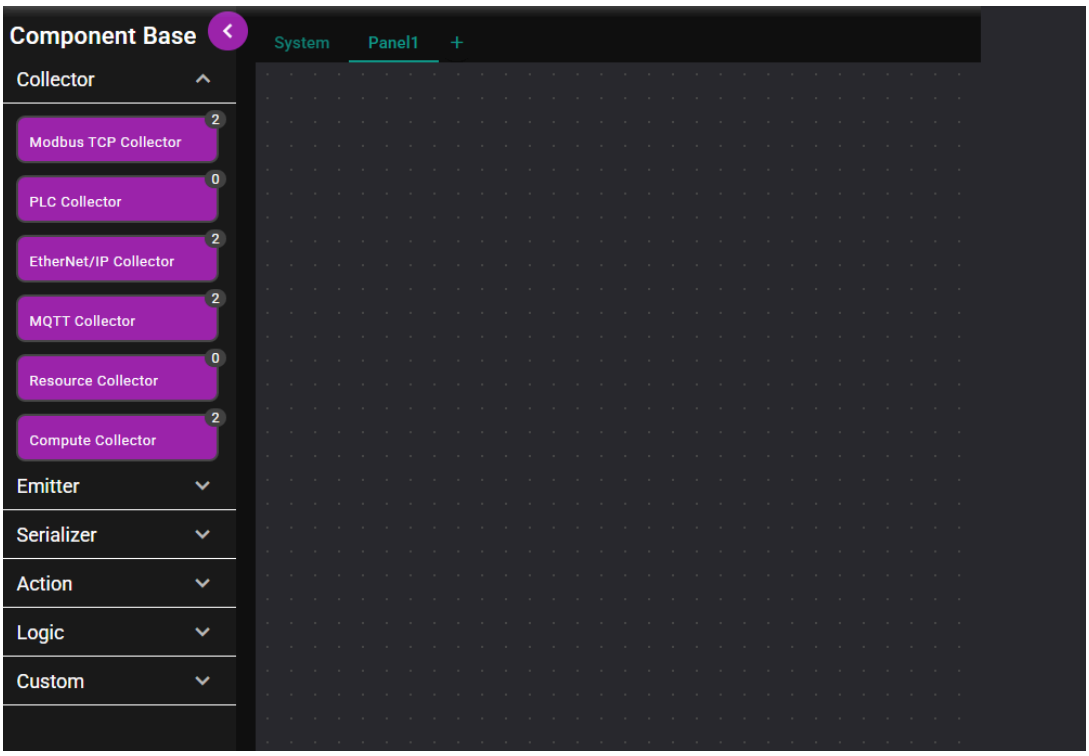
1. 헤더의 모드 선택에서 "Edit mode"를 선택하고, Edit 모드의 화면을 엽니다.



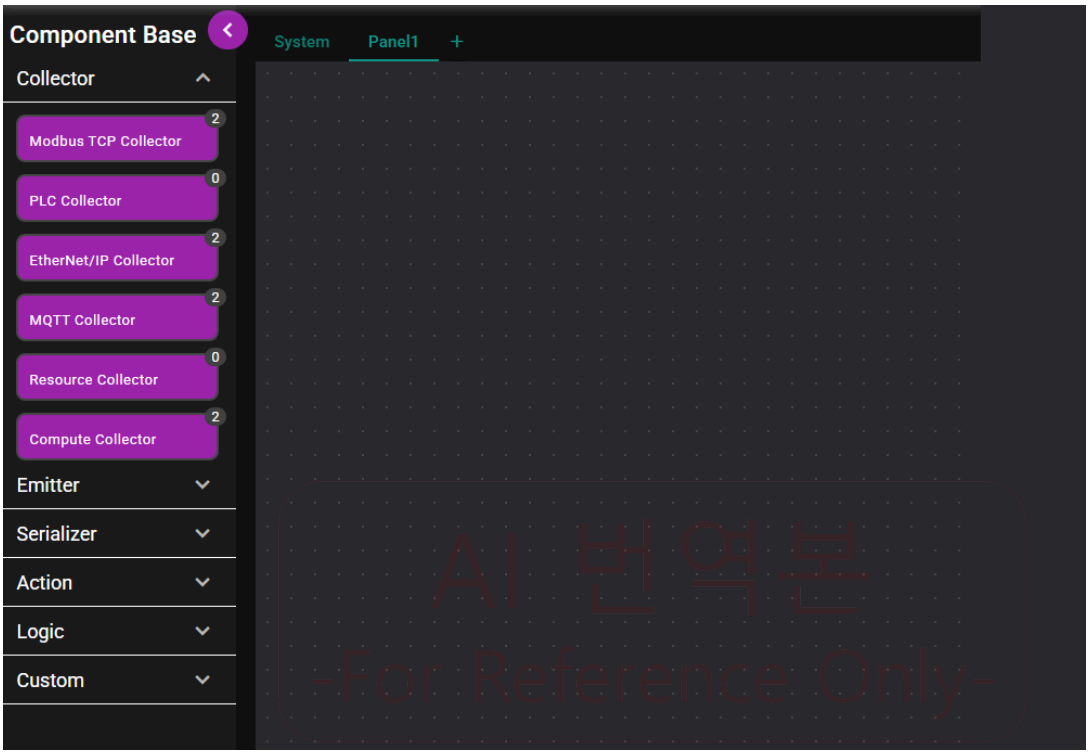
2. 패널1을 표시합니다.



3. 좌측 메뉴에서「Collector」의 인디케이터를 누르고, 컬렉터 목록을 표시합니다.

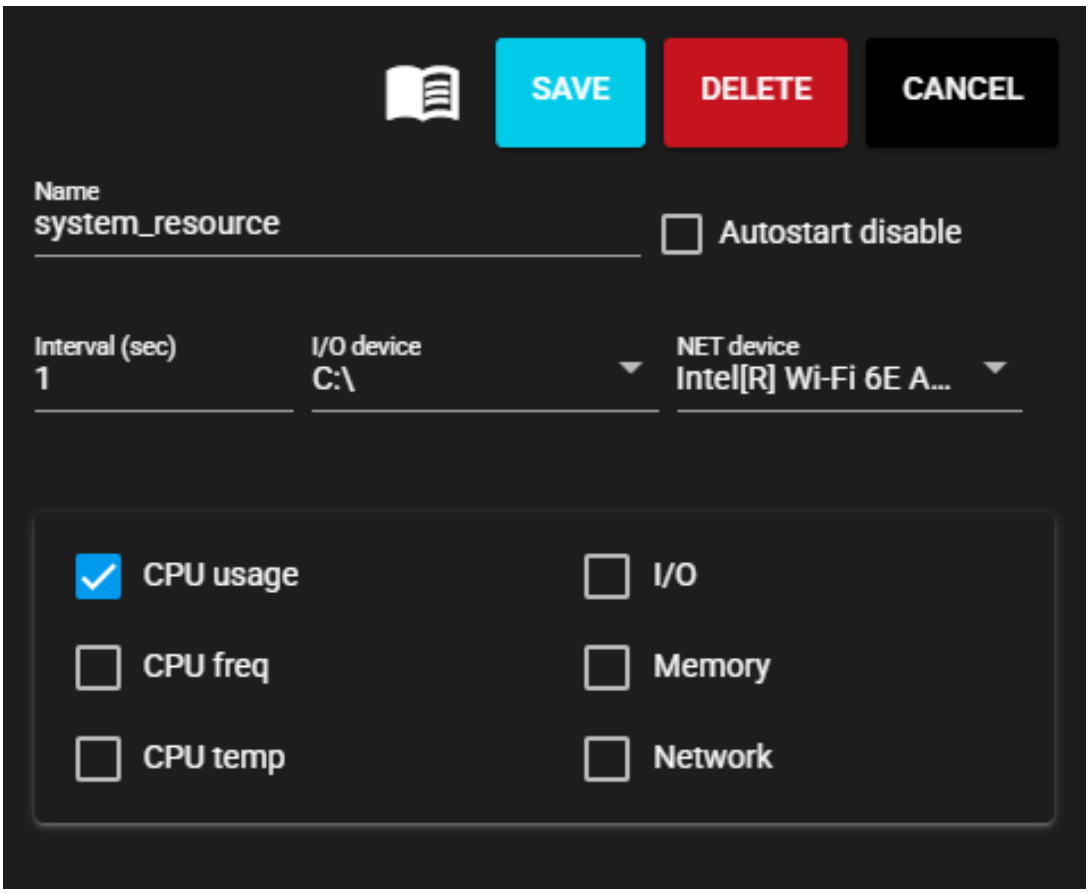


4. 「 리소스 컬렉터 」를 드래그하고, 패널상에 드롭합니다.



5. 설정 다이얼로그가 자동으로 열립니다. "명칭"을 임의로 입력하고, "CPU 사용률"에 체크를 붙이고「 저장 」버튼을 누릅니다.

명칭 예 : 시스템 리소스



4.5.2 CSV 시리얼라이저의 설정

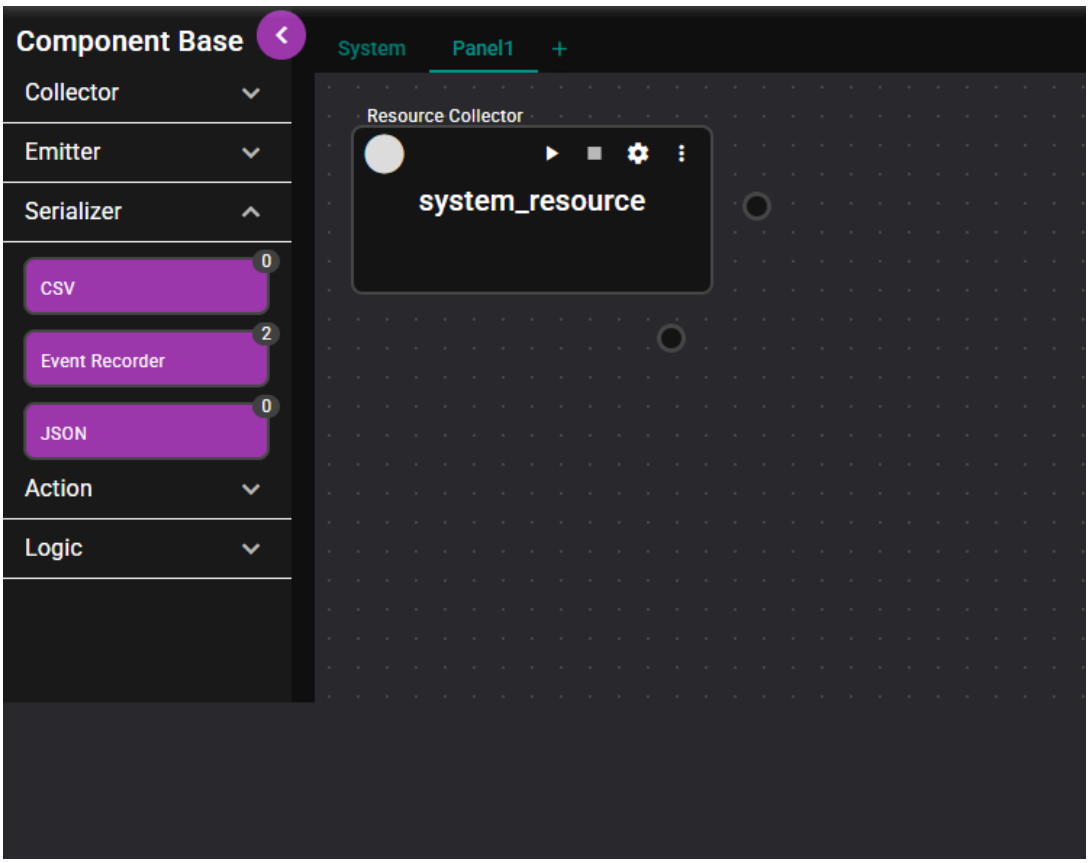
리소스 컬렉터로 수집한 CPU 사용률을, CSV 형식으로 파일 출력하기 위해, "CSV 시리얼라이저"를 설정하는 방법을 설명합니다.
"CSV 시리얼라이저"는, 임의의 데이터를 받아들여, CSV 형식으로 가공하여 재출력하는 컴포넌트
입니다.

컴포넌트 간의 데이터의 흐름은, 플로우 링크로 정의합니다. 플로우 링크란, 2개의 컴포넌트를 연
결하고, 연결한 컴포넌트 간에 데이터를 흘려보내는 것입니다.

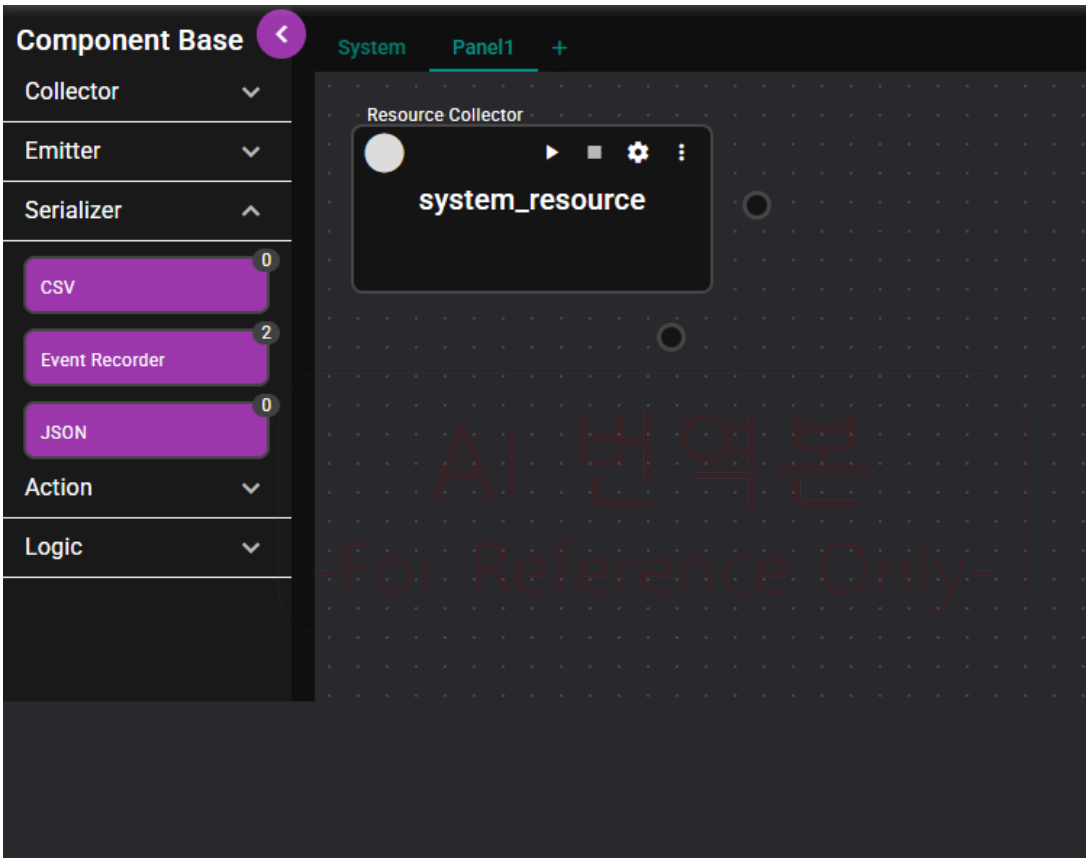
다음 순서로, CSV 시리얼라이저를 설정할 수 있습니다.

1. Edit 모드의 화면에서, 좌측 메뉴에서 「 Serializer 」의 인디케이터를 누르고, 시리얼라이저 목록을 표시시킵니다.

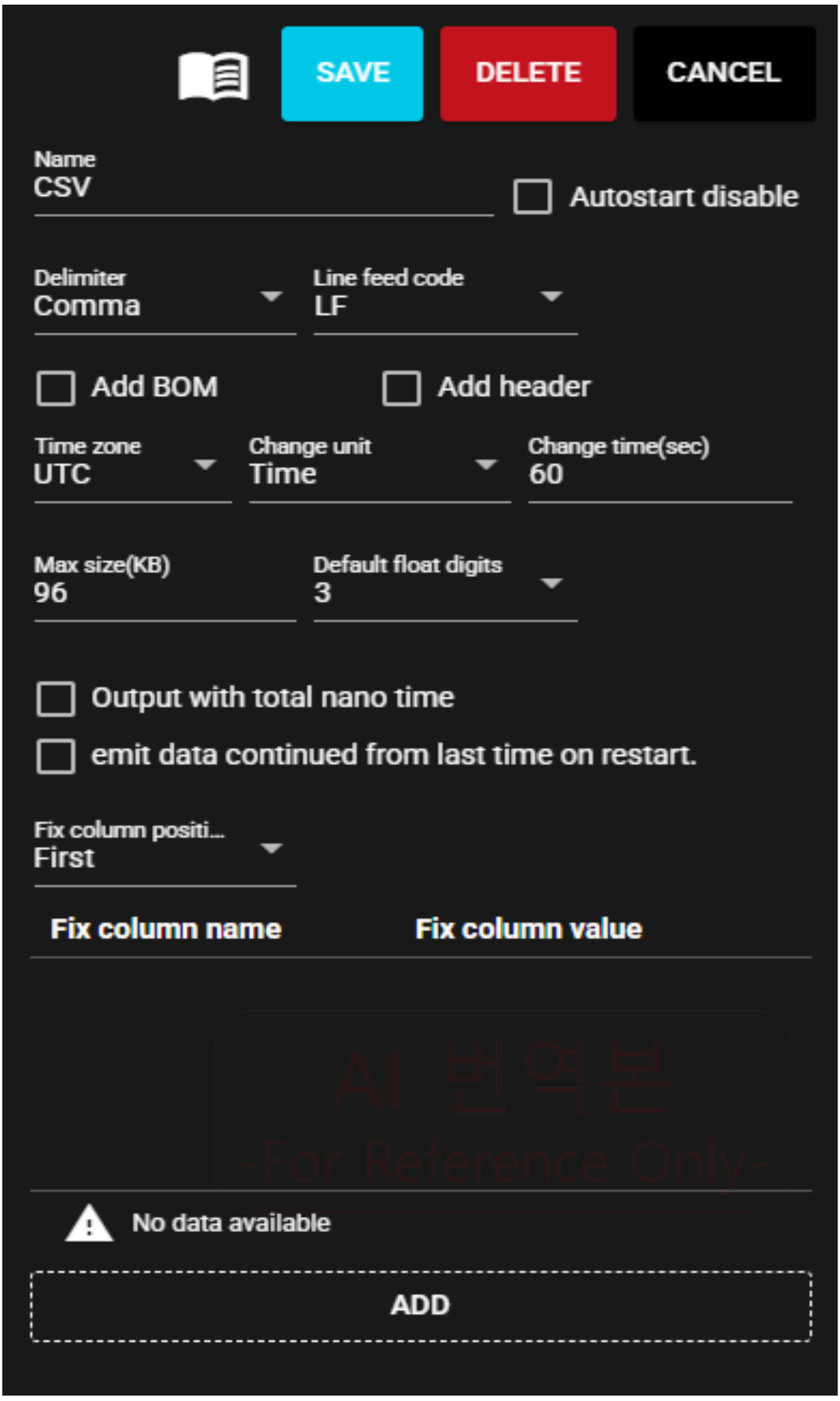
AI 번역본
-For Reference Only-



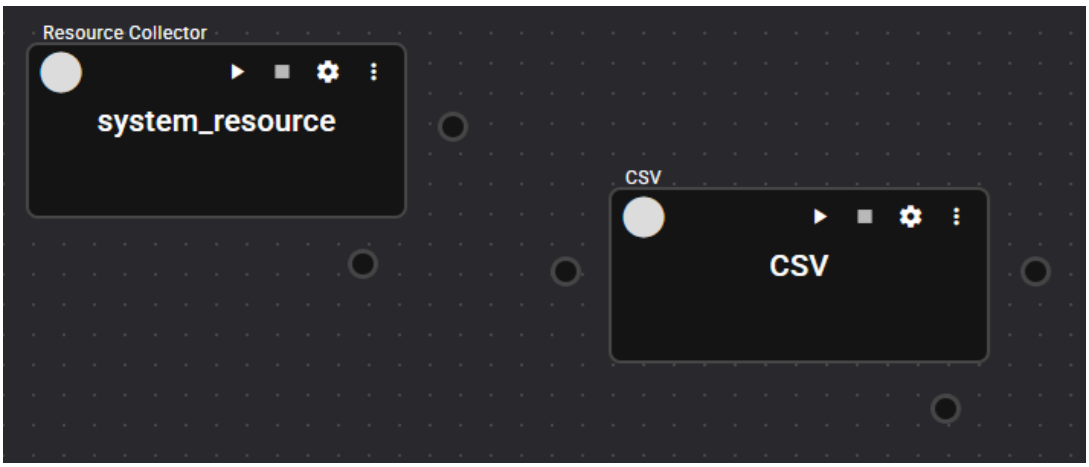
2. 「 CSV 시리얼라이저 」를 드래그하고, 패널상에 드롭합니다.



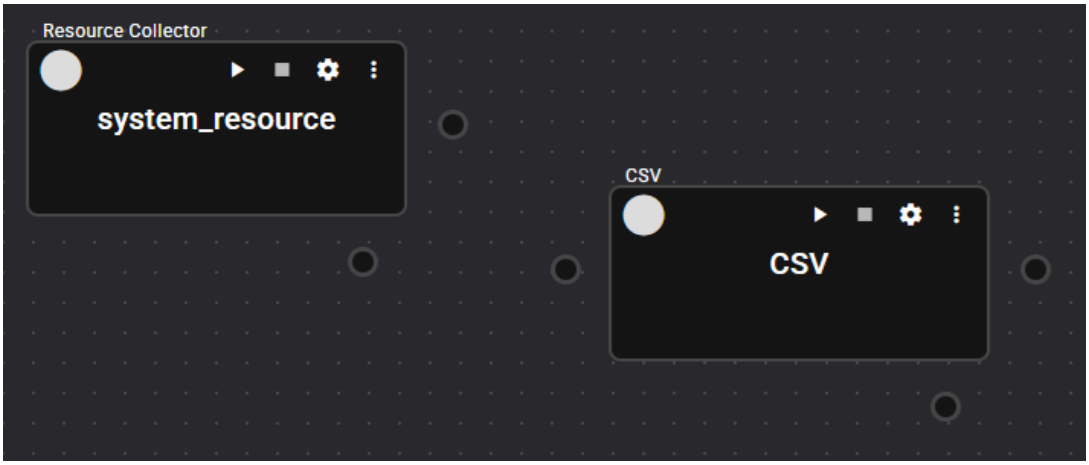
3. 설정 다이얼로그가 자동으로 열립니다. "명칭"을 임의로 입력하고,「 저장 」버튼을 누릅니다.
   명칭 예：CSV 포맷



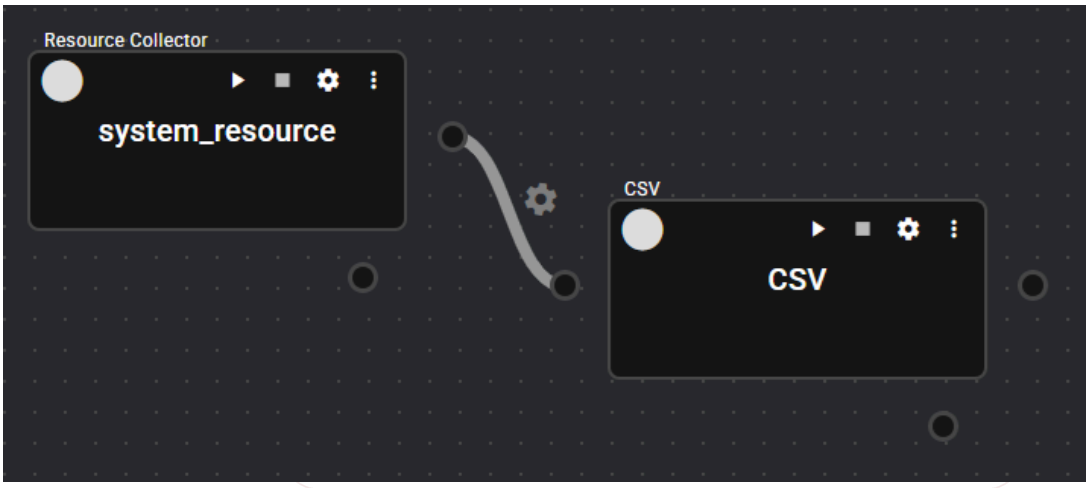
4. 리소스 컬렉터의 출력 포트를 드래그합니다.



5. CSV 시리얼라이저의 입력 포트에서 마우스의 버튼을 뗍니다.



6. 컴포넌트 사이에 플로우 링크가 추가됩니다.

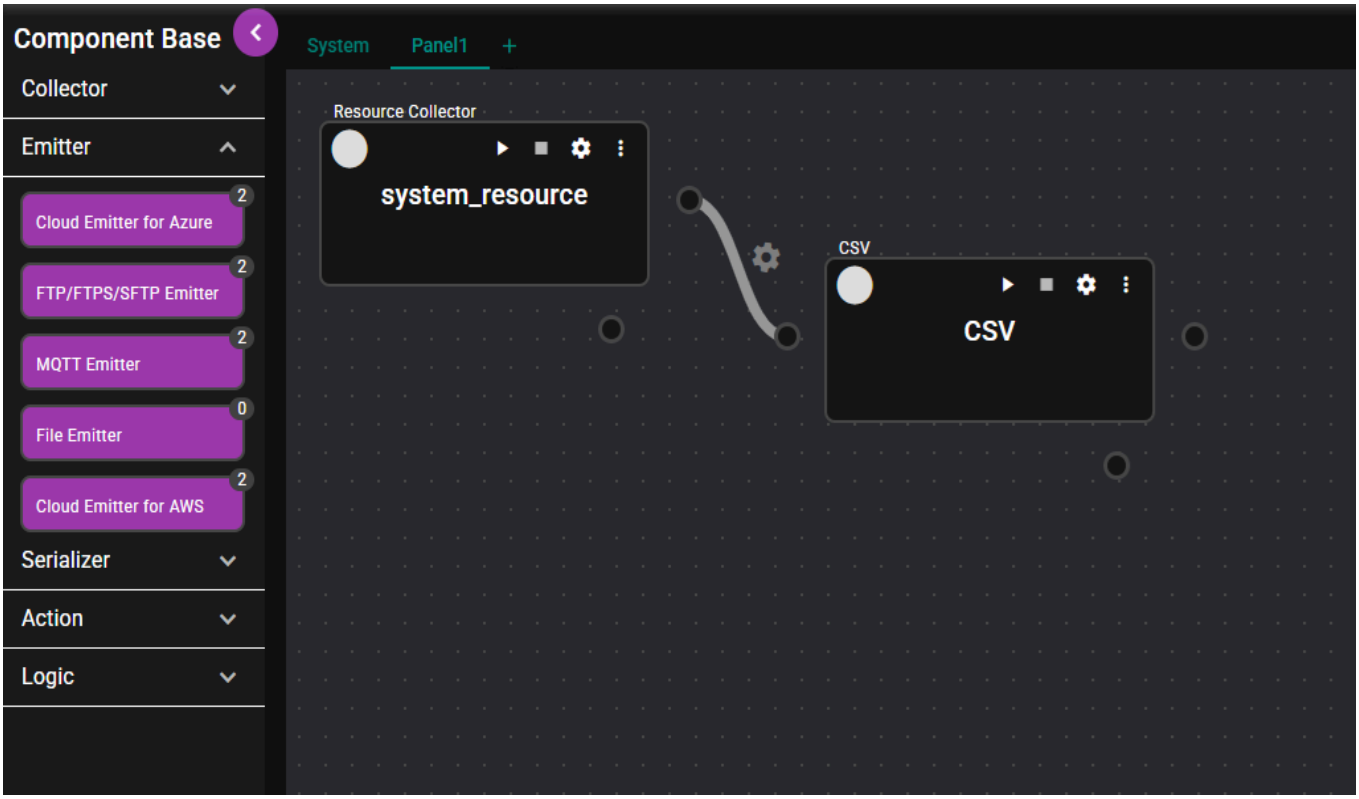


4.5.3 File 에미터의 설정

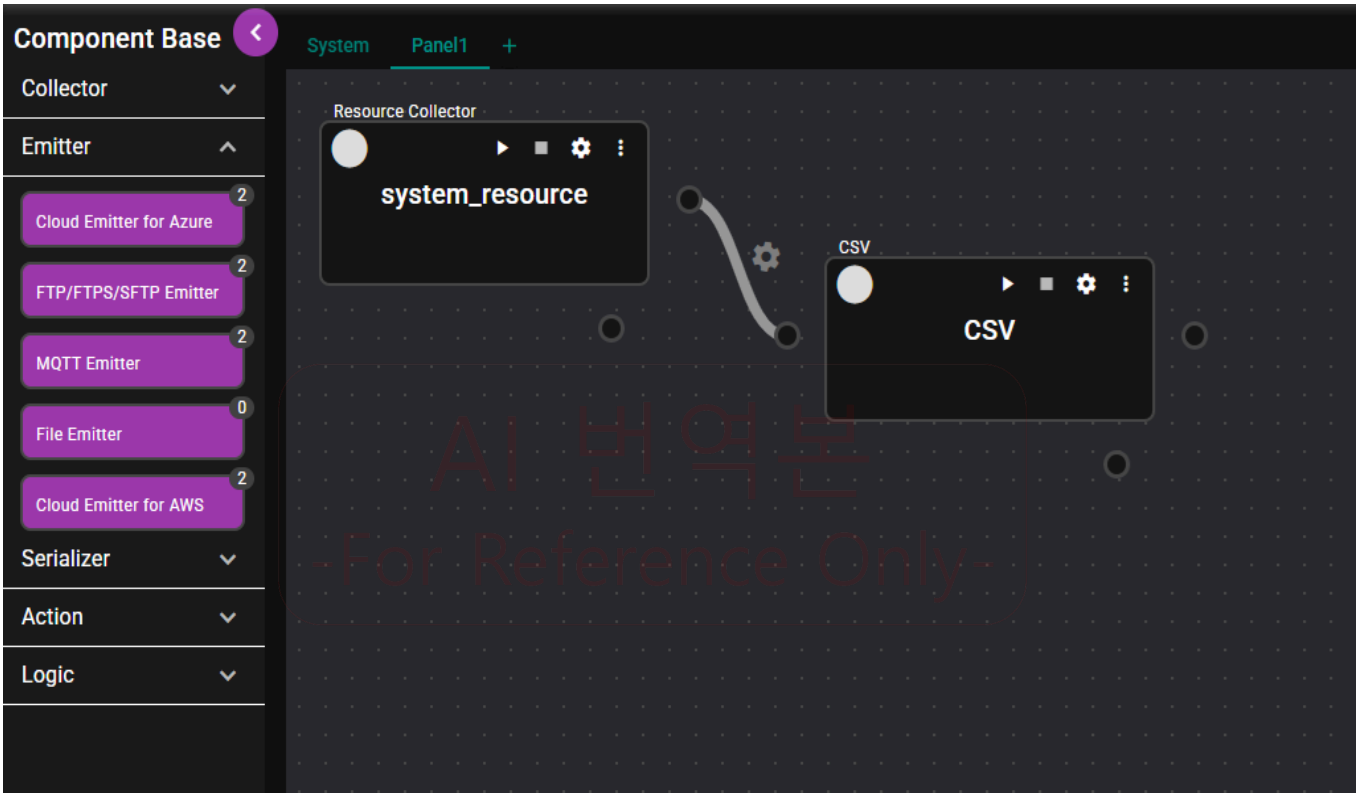
이어서 "File 에미터"를 설정하는 방법을 설명합니다. "File 에미터"는, 시리얼라이저에 의해 가공된 데이터를 받아들여, 로컬 환경에 파일 출력하는 컴포넌트입니다.

다음 순서로, File 에미터를 설정할 수 있습니다.

1. Edit 모드의 화면에서, 좌측 메뉴에서 「 Emitter 」의 인디케이터를 누르고, 에미터 목록을 표시시킵니다.

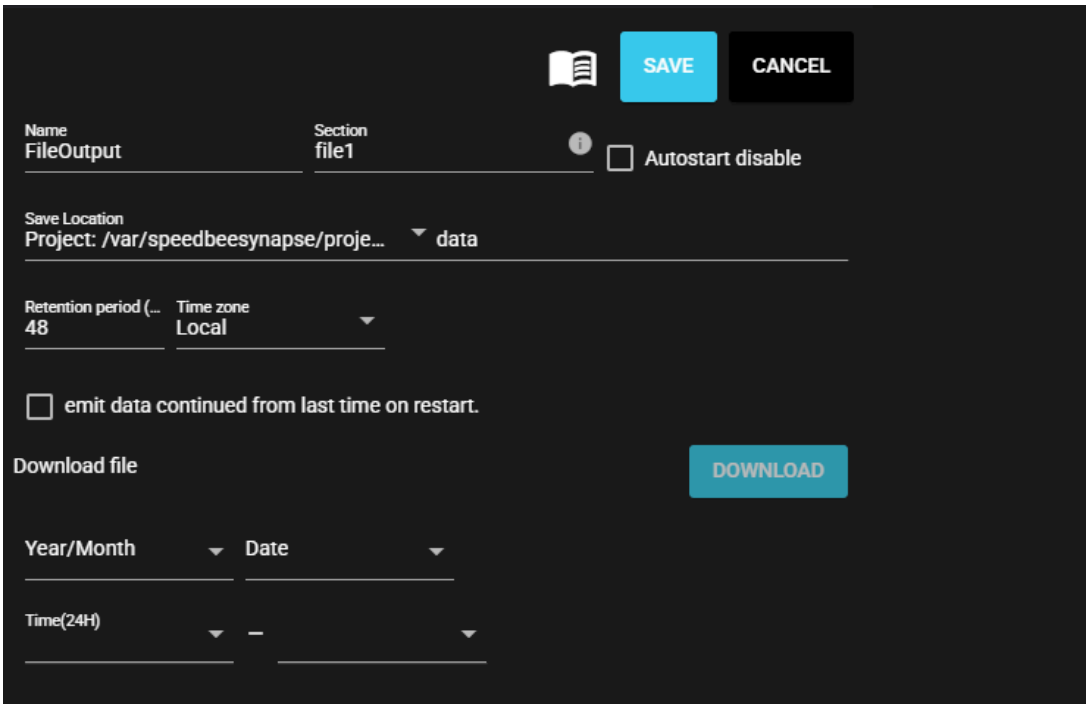


2. 「 File 에미터 」를 드래그하고, 패널상에 드롭합니다.

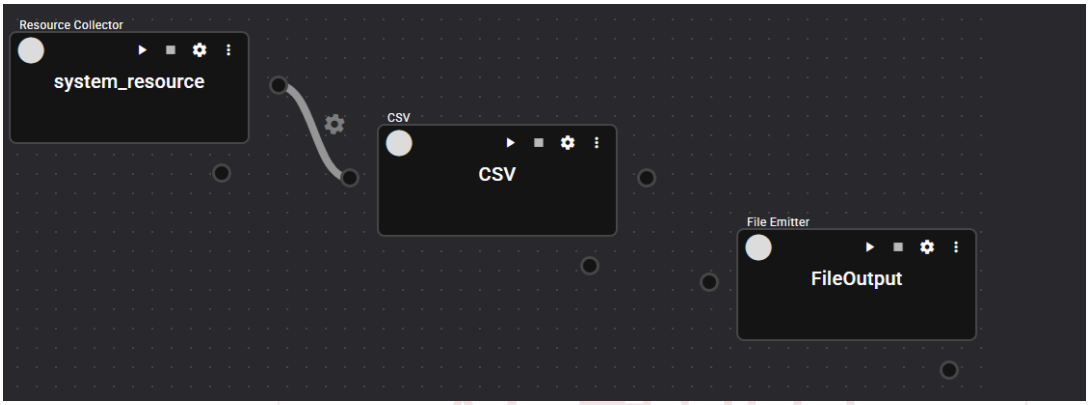


AI 번역본
- For Reference Only -

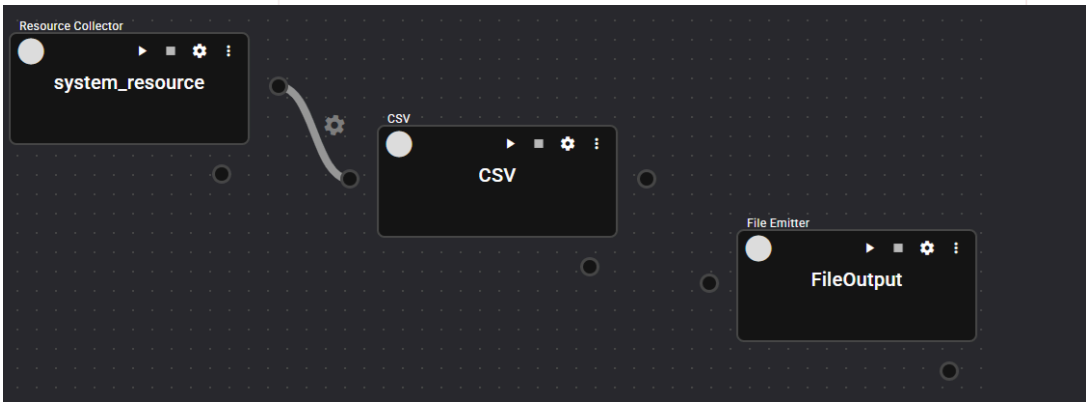
3. 설정 다이얼로그가 자동으로 열립니다. "명칭"과 "구분"을 임의로 입력하고,「 저장 」버튼을 누릅니다.
   명칭 예 : 파일 출력
   구분 예 : file1
   ※"구분"은 반각만 입력 가능합니다.



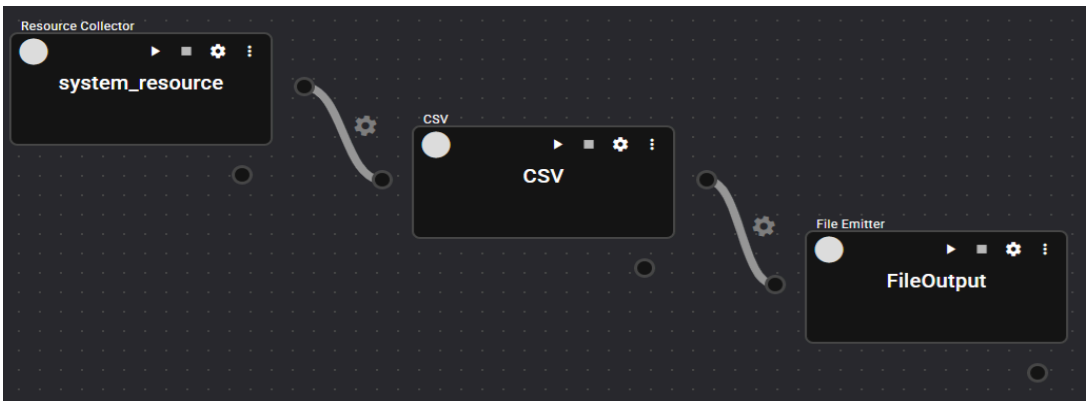
4. CSV 시리얼라이저의 출력 포트를 드래그합니다.



5. File 에미터의 입력 포트에서 마우스의 버튼을 뗍니다.



6. 컴포넌트 사이에 플로우 링크가 추가됩니다.



AI 번역본
-For Reference Only-

4.6 수집 서비스의 실행

여기서는 컴포넌트를 기동하여, 수집 서비스를 개시하는 방법을 설명합니다.

또한, 수집 서비스를 개시한 후, 전항까지 설정한 컬렉터・시리얼라이저・에미터가 정상적으로 가동되고 있는 것을 확인하는 방법과, 설정을 튜닝하는 방법을 설명합니다.

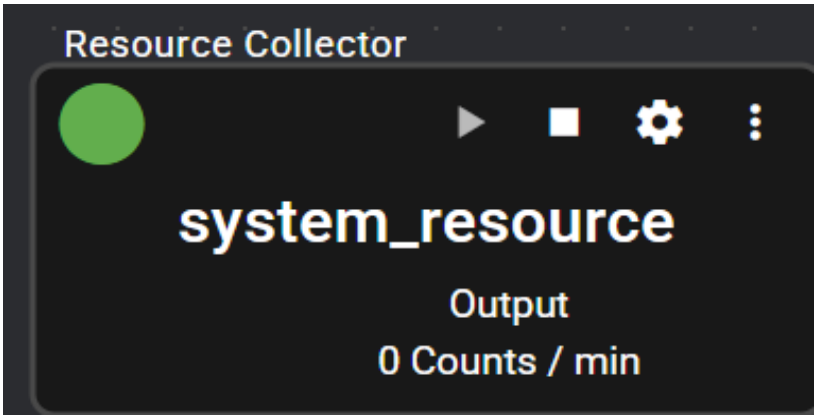
4.6.1 수집 서비스의 개시/정지

컴포넌트를 기동함으로써, 수집 서비스를 개시할 수 있습니다. 컴포넌트의 기동은, 컴포넌트 단위, 패널 전체, SpeeDBee Synapse 전체의 3가지 범위로 수행할 수 있습니다.

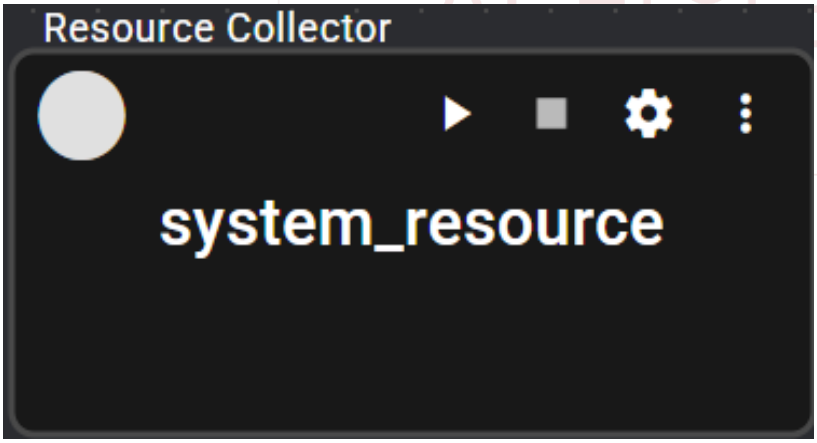
4.6.1.1 컴포넌트의 기동/정지

다음 순서로, 하나의 컴포넌트를 기동/정지할 수 있습니다.

1. 컴포넌트의 기동 아이콘을 누르면, 컴포넌트를 기동할 수 있습니다. 컴포넌트가 기동하면, 스테이터스 램프가 녹색이 됩니다.



2. 컴포넌트의 정지 아이콘을 누르면, 컴포넌트를 정지할 수 있습니다. 컴포넌트가 정지하면, 스테이터스 램프가 회색이 됩니다.



시 범 영 본
Only-

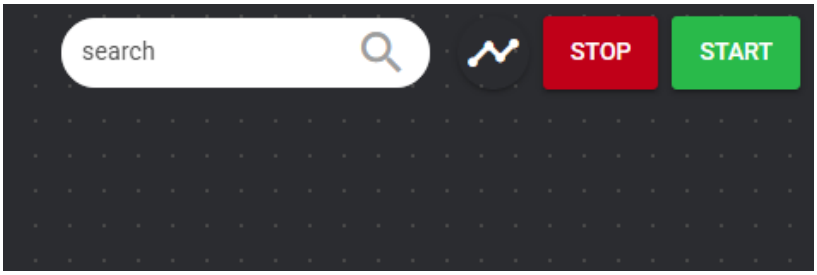
4.6.1.2 패널 상의 전 컴포넌트의 기동/정지

다음 순서로, 하나의 패널 상의 컴포넌트를 기동/정지할 수 있습니다.

1. 패널의 「 기동 」 버튼을 누르면, 패널 상의 「 자동 기동 무효 」가 OFF인 컴포넌트를 모두 기동할 수 있습니다.



2. 패널의 「 정지 」 버튼을 누르면, 패널 상의 모든 컴포넌트를 정지할 수 있습니다.



4.6.1.3 SpeeDBee Synapse 상의 전 컴포넌트의 기동/정지

다음 순서로, SpeeDBee Synapse 상의 컴포넌트를 기동/정지할 수 있습니다.

1. 헤더의 「 기동 」 버튼을 누르면, SpeeDBee Synapse 상의 「 자동 기동 무효 」가 OFF인 컴포넌트를 모두 기동할 수 있습니다.



2. 헤더의 「 정지 」 버튼을 누르면, SpeeDBee Synapse 상의 모든 컴포넌트를 정지할 수 있습니다.



AI 번역본
Only-

4.6.2 동작 확인

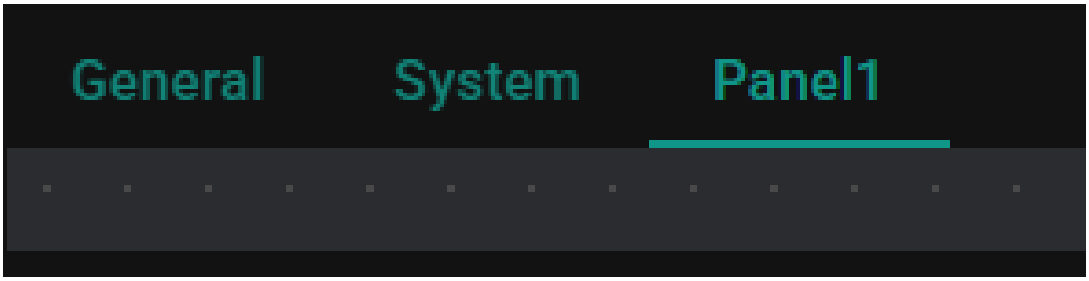
여기서는, 리소스 컬렉터가 수집한 데이터의 확인 방법, File 에미터가 출력한 파일을 다운로드하는 방법에 대해 설명합니다.

리소스 컬렉터의 동작 확인

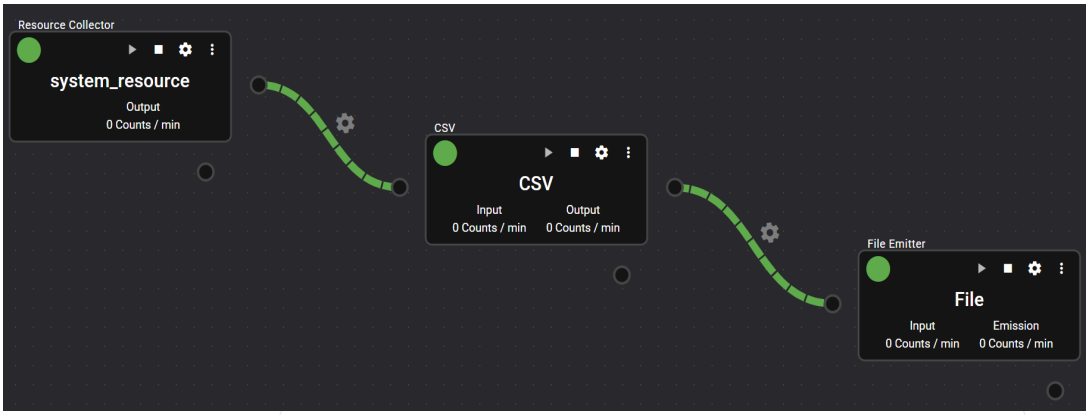
1. 헤더의 모드 선택에서 "Run"을 선택하고, Run 모드의 화면을 엽니다.



2. 패널1을 표시합니다.

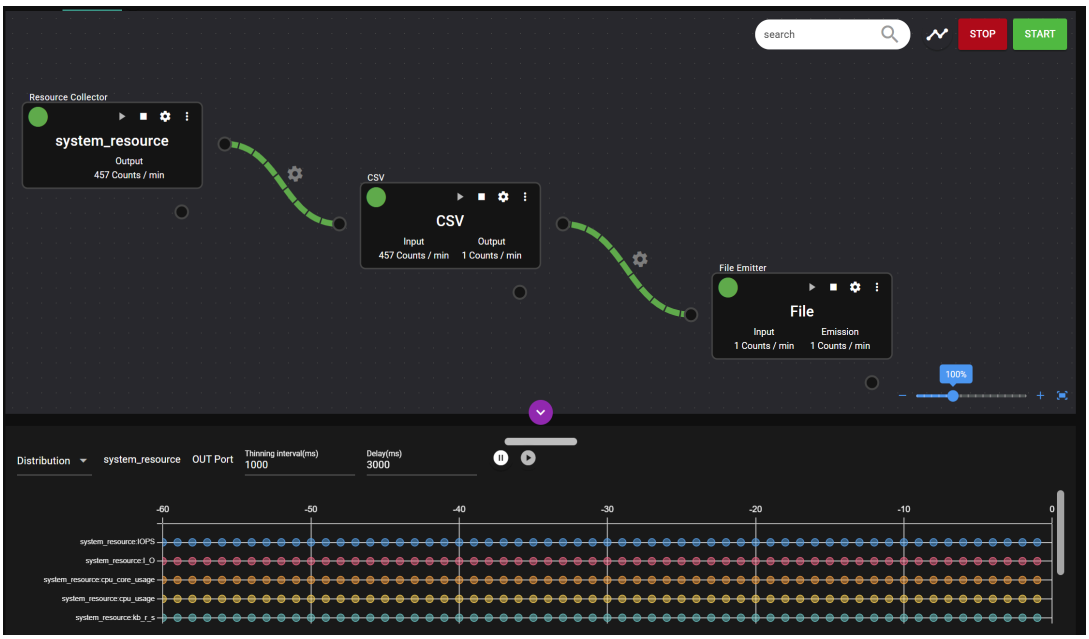


3. 리소스 컬렉터의 출력 포트를 클릭합니다.



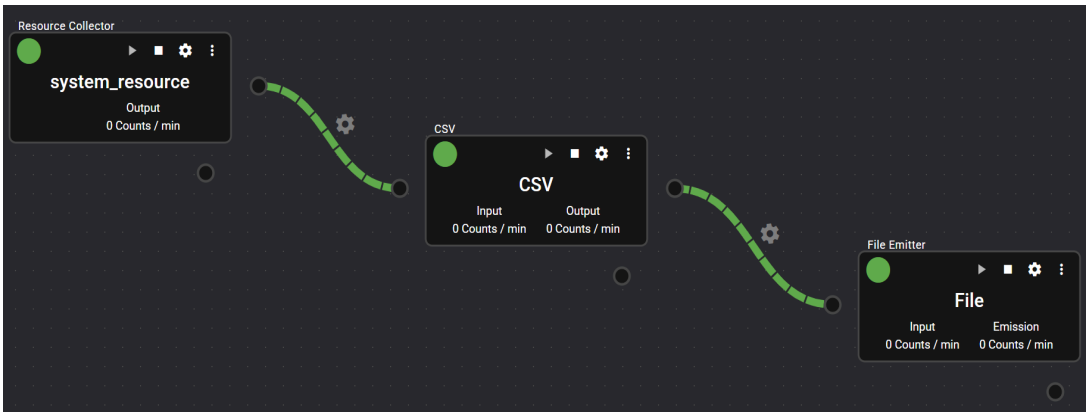
4. 화면 하부의 인디케이터를 클릭하면 데이터 모니터가 표시되고, 리소스 컬렉터가 수집하고 있는 데이터를 확인할 수 있습니다.

AI 번역본
-For Reference Only-



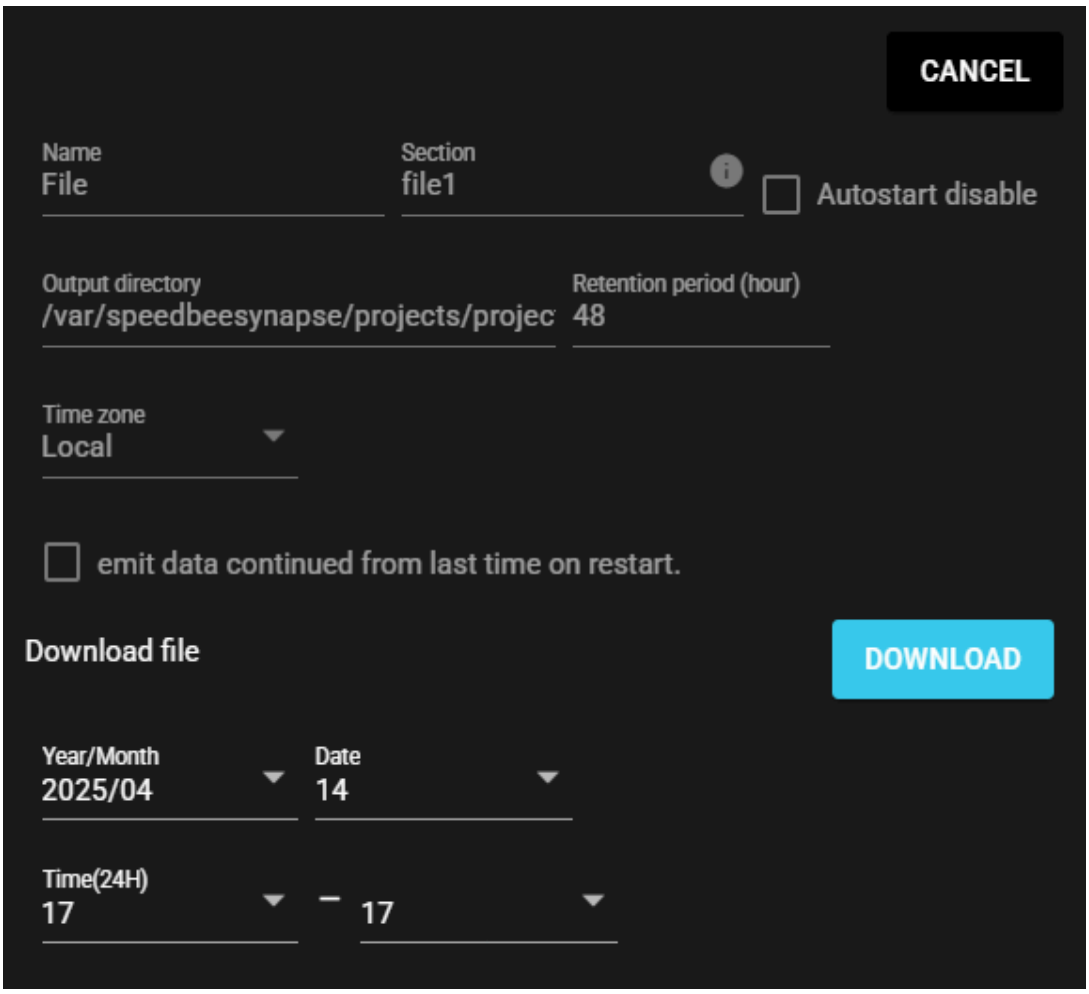
File 에미터의 동작 확인

1. File 에미터의 설정 아이콘을 누릅니다.



2. 시간 범위를 입력하고,「 다운로드 」를 누릅니다. 지정한 시간 범위 사이에 File 에미터가 출력한 파일이, zip 형식으로 다운로드됩니다.

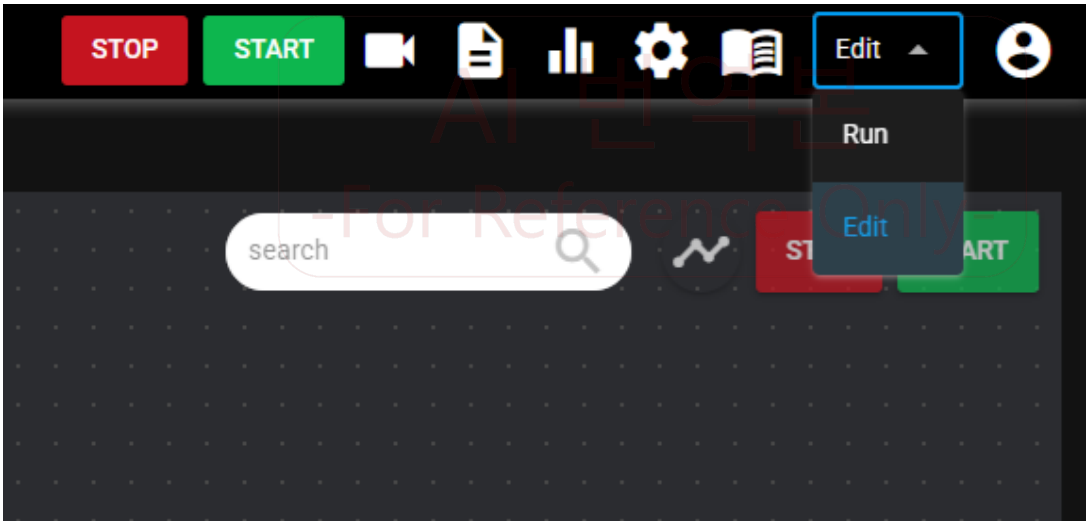
AI 번역본
-For Reference Only-



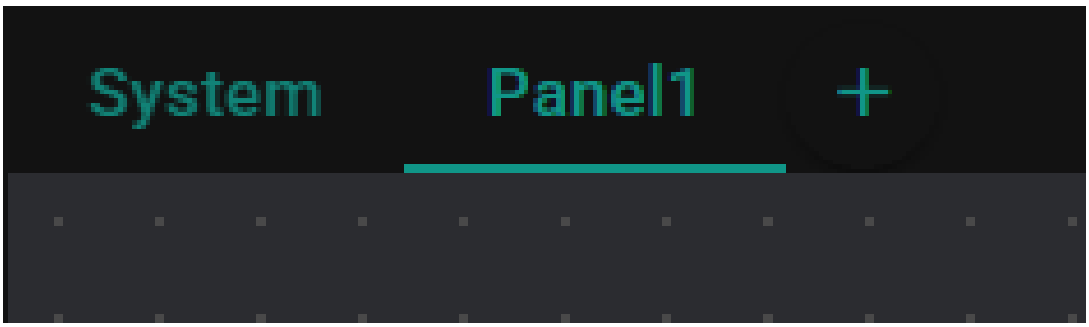
4.6.3 설정 튜닝

여기서는, 컴포넌트의 설정을 편집하는 방법, 컴포넌트를 삭제하는 방법에 대해 설명합니다.

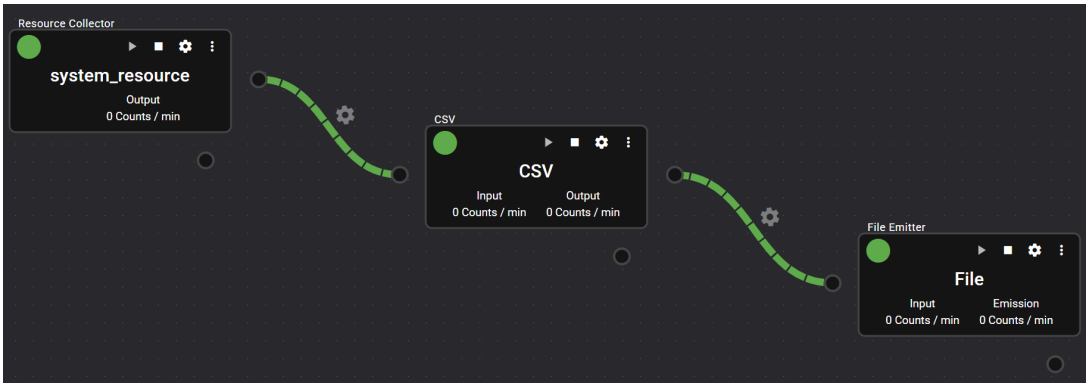
1. 헤더의 모드 선택에서 "Edit mode"를 선택하고, Edit 모드의 화면을 엽니다.



2. 패널1을 엽니다.

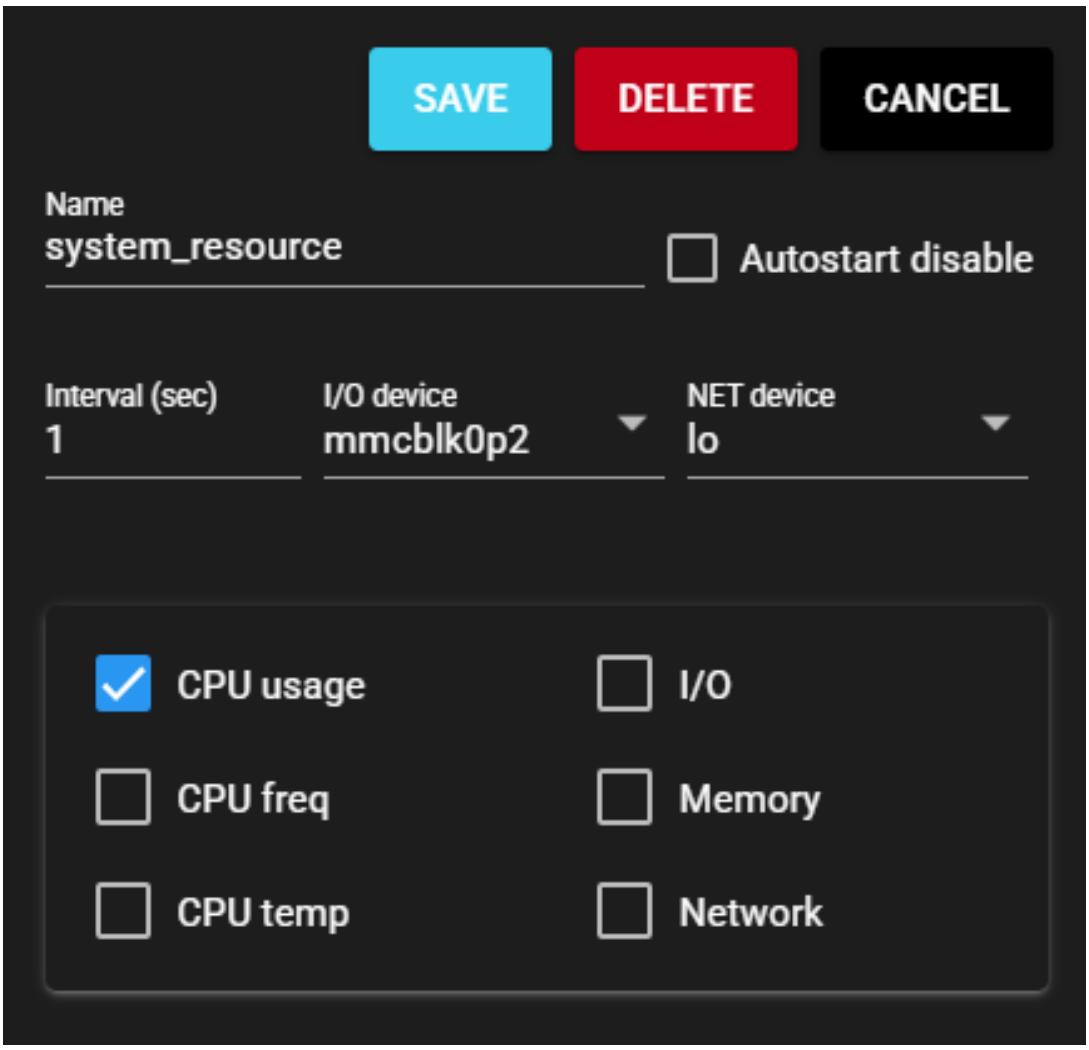


3. 설정을 변경할 컴포넌트의 설정 아이콘을 누릅니다.



4. 설정 다이얼로그에서 설정을 변경하고, 「 저장 」을 누릅니다. 여기서는 리소스 컬렉터의 이미지를 사용하고 있습니다. 예를 들어, 메모리 사용량 등을 수집하는 경우, 체크를 ON으로 합니다. 또한, 컴포넌트를 삭제할 경우는, 「 삭제 」를 누르고, 확인 다이얼로그에서 「 예 」를 선택합니다.

AI 번역본
-For Reference Only-



5. 설정을 변경한 컴포넌트를 정지하고, 다시 기동함으로써, 설정을 수집 서비스에 반영시킬 수 있습니다. 컴포넌트의 기동/정지에 대해서는 「 수집 서비스의 개시/정지 」를 참조하십시오.

4.6.4 수집 서비스의 종료

SpeeDBee Synapse 상의 전 컴포넌트를 정지함으로써, 수집 서비스는 대기 상태가 됩니다. 순서에 대해서는, 「 SpeeDBee Synapse 상의 전 컴포넌트의 기동/정지 」를 참조하십시오.

AI 번역본
-For Reference Only-

5 각종 기능 설명

AI 번역본
-For Reference Only-

5.1 각종 기능 설명

본 장에서는, 각종 기능에 대해 설명합니다.

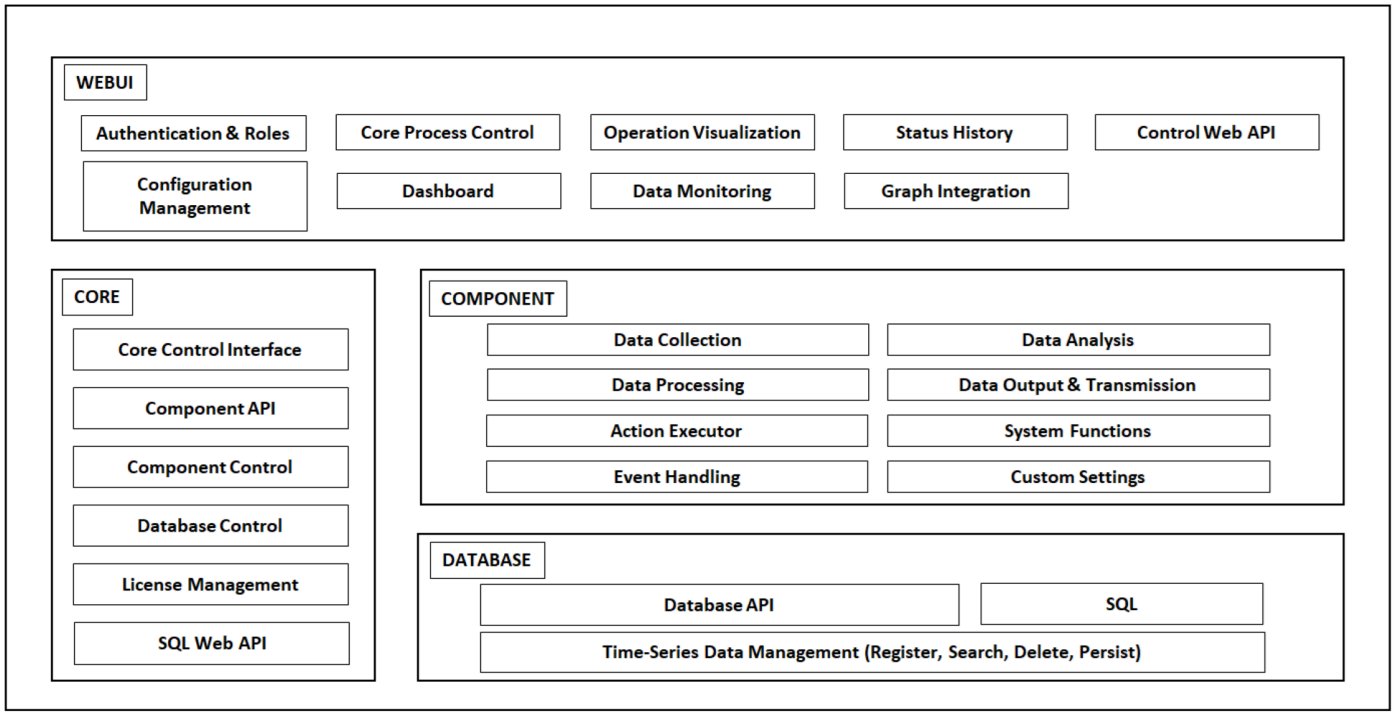
기재 내용은 다음과 같습니다.

1. 엔진 기능의 전체상
2. 플로우 에디터
3. 컬렉터
4. 시리얼라이저
5. 에미터
6. 로직（이벤트）
7. 로직（실시간 분석/데이터 조작）
8. 액션
9. 개발용 컴포넌트
10. 시스템 컴포넌트
11. 컴포넌트 공통의 설정 항목
12. 스테이터스 타임라인
13. 각종 로그
14. DB쿼리（SQL）
15. 연산식・조건식
16. 설정 정보의 관리(프로젝트)

AI 번역본
-For Reference Only-

5.2 엔진 기능의 전체상

여기서는, 엔진 기능의 전체상에 대해 설명합니다. 이 절은, 내부의 처리에 대해, 자세히 알고 싶은 경우에 참조하십시오. 다음 그림은, SpeeDBee Synapse의 구성입니다.



5.2.1 기능 목록

여기서는, 구성도에 나타낸 내용에 대해 해설합니다.

5.2.1.1 코어

코어는, SpeeDBee Synapse의 중심으로, 각 기능의 제어를 수행하고 있는 부분입니다. 코어는 다음 기능을 제공합니다.

항목	설명
코어 제어 인터페이스	SpeeDBee Synapse의 각 기능을 제어합니다. WEBUI로부터의 요구를 접수하여 처리합니다. 사용자가 직접 사용하는 일은 없습니다.
컴포넌트API	컴포넌트가 사용하는 API를 제공합니다. 컴포넌트 개발자는, 이들 API를 이용하여 프로그래밍을 수행할 필요가 있습니다.
컴포넌트 제어	컴포넌트(데이터의 수집, 가공, 분석, 송신 등을 실현하는 모듈군)의 처리를 제어합니다. 코어가 내부에서 자동적으로 처리합니다.

항목	설명
데이터베이스 제어	SpeeDBee Synapse에 탑재되어 있는 시계열 데이터베이스를 제어합니다. 코어가 내부에서 자동 제어합니다.
라이선스 제어	라이선스에 관한 제어를 수행합니다. 올바른 라이선스 아래 SpeeDBee Synapse를 이용할 수 있도록 제어합니다.
SQL Web-API	SpeeDBee Synapse에 탑재되어 있는 시계열 데이터베이스에 대해, SQL을 실행할 수 있는 Web-API를 제공합니다. 상세는「DB쿼리（SQL）」,「DB쿼리 서비스」를 참조하십시오.

5.2.1.2 컴포넌트

컴포넌트는, SpeeDBee Synapse에 있어서, 임의의 처리를 수행하는 모듈군입니다. 컴포넌트를 사용함으로써, 이하의 것이 가능합니다.

항목	설명
데이터 수집	특정 대상으로부터 데이터를 수집할 수 있습니다. 상세는「컬렉터」를 참조하십시오.
데이터 분석	수집한 데이터를 분석할 수 있습니다. 상세는「로직（실시간 분석/데이터 조작）」을 참조하십시오.
이벤트	수집한 데이터에 따른 이벤트를 처리합니다. 상세는「로직（이벤트）」를 참조하십시오.
데이터 가공	데이터를 특정 포맷으로 가공할 수 있습니다. 상세는「시리얼라이저」를 참조하십시오.
데이터 출력・송신	데이터를 특정 대상으로 출력・송신할 수 있습니다. 상세는「에미터」를 참조하십시오.
액션	임의의 액션을 실행합니다. 예를 들어, OS 커맨드의 실행이나 메일 송신 등이 가능합니다. 상세는「액션」을 참조하십시오.
시스템	시스템에 관련된 기능을 제공합니다. 초기 상태로 배치되어 있으며, 사용자가 새로운 인스턴스를 추가할 수는 없습니다. 상세는「시스템 컴포넌트」를 참조하십시오.

AI 번역본
-For Reference Only-

항목	설명
커스텀	사용자 자신이 컴포넌트를 개발할 수 있습니다. 상세는「커스텀 컴포넌트의 작성・이용」의「개요」를 참조하십시오.

5.2.1.3 WEBUI

WEBUI는, 사용자가 조작하는 SpeeDBee Synapse의 화면을 제공합니다. WEBUI는, 다음 기능을 제공합니다.

항목	설명
인증・롤	SpeeDBee Synapse를 정당한 사용자가 이용할 수 있도록, 인증 기능을 제공합니다. 또한, 사용자의 권한에 맞춘 롤（롤）을 할당할 수 있습니다. 상세는,「사용자 관리」를 참조하십시오.
코어 프로세스 제어	코어 프로세스의 제어를 수행합니다. 사용자의 조작 및 코어 프로세스의 이상 시에, 프로세스 제어를 수행합니다.
대시보드	데이터 수집 플로우를 시각적으로 표시하고, 인터랙티브하게 커스터마이즈할 수 있는 인터페이스입니다. 상세는,「패널」을 참조하십시오.
설정 정보 관리	SpeeDBee Synapse의 설정 정보를 관리할 수 있습니다. 사용자가 입력한 정보는, 설정 파일로서 관리됩니다.
가동 정보 가시화	SpeeDBee Synapse 및 컴포넌트의 가동 상태를 표시합니다. 상세는,「스테이터스 목록」을 참조하십시오.
스테이터스 이력	컴포넌트의 가동 상태의 이력을 표시합니다. 상세는,「스테이터스 타임라인」을 참조하십시오.
제어Web-API	SpeeDBee Synapse를 제어하는 Web-API를 제공합니다. 상세는,「제어 프로그램과의 연계」를 참조하십시오.
데이터 모니터	수집한 데이터를 모니터로 열람할 수 있습니다. 상세는,「데이터 모니터」를 참조하십시오.
그래프 연계	수집한 데이터를 바탕으로, 그래프를 작성할 수 있습니다. 상세는, 「grafana 연계」를 참조하십시오.

AI 번역본
-For Reference Only-

5.2.1.4 데이터베이스

SpeeDBee Synapse에는, 시계열 데이터베이스가 탑재되어 있습니다. 시계열 데이터베이스는, 다음 기능을 제공합니다.

항목	설명
데이터베이스API	시계열 데이터베이스를 조작하는 API를 제공합니다. SpeeDBee Synapse에 있어서, 사용자가 이 API를 의식할 필요는 없습니다.
SQL	시계열 데이터베이스에 대해, SQL을 실행할 수 있습니다. 상세는「DB쿼리（SQL）」를 참조하십시오.
시계열 데이터 처리 (등록, 검색, 삭제, 영속화)	수집한 데이터를 등록, 검색, 삭제, 영속화할 수 있습니다. 코어에 의해, 이것들은 제어됩니다.

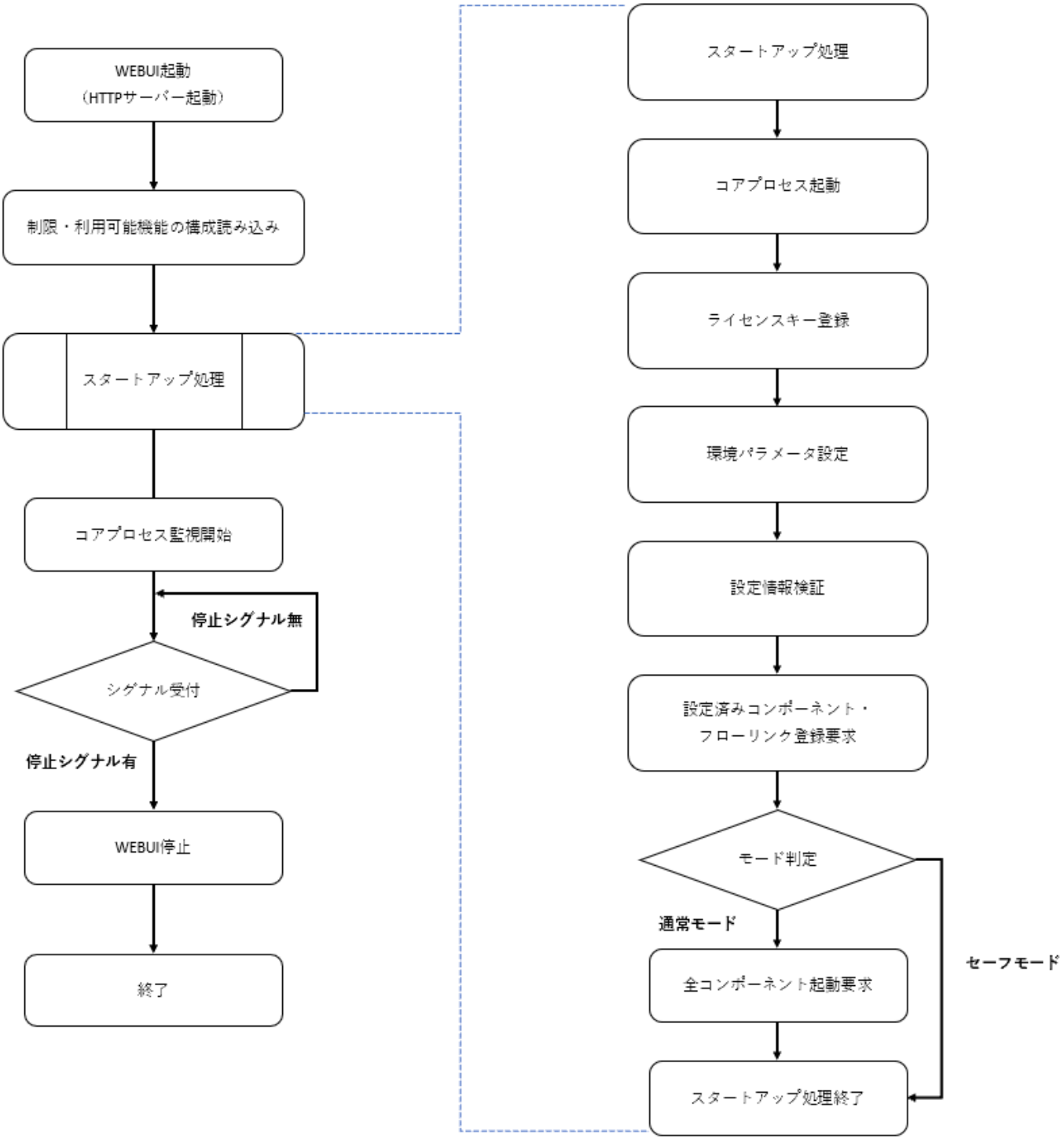
5.2.2 동작 사양

본 항에서는, SpeeDBee Synapse의 동작 사양에 대해 설명합니다.

5.2.2.1 기본 동작

SpeeDBee Synapse의 기본 동작 사양은 다음과 같습니다.

AI 번역본
-For Reference Only-



-For Reference Only-

상기 플로우 도표에서 이루어지는 처리의 상세를 아래 표에 나타냅니다.

- 기본 동작

처리	설명
WEBUI 기동	SpeeDBee Synapse의 WEBUI를 기동합니다. 내부적으로는 HTTP 서버가 동작합니다.

처리	설명
제한・이용 가능 기능의 구성 불러오기	구성 정보를 읽어들입니다. 유효한 기능이나 각종 제한값에 관한 정보가 대상입니다. 기본적으로는, 사용자가 준비・설정 등을 수행하는 것이 아닙니다.
스타트업 처리	코어 프로세스의 기동 및 SpeeDBee Synapse를 사용 가능하게 하기 위한 처리를 수행합니다. 상세는 후술합니다.
코어 프로세스 감시 개시	코어 프로세스의 감시 처리를 기동합니다. 코어 프로세스의 상태 감시를 수행하고, 이상 발생 시에는, 코어 프로세스의 재기동을 수행합니다. 이상이 계속되는 경우는, 코어 프로세스를 기동하지 않고 세이프 모드로 천이합니다.
시그널 접수	OS로부터의 시그널을 접수합니다. 정지 시그널을 수신하면, 종료 처리로 이행합니다.
WEBUI 정지	WEBUI를 정지합니다. 코어 프로세스 감시 등의 종료 처리나 각 처리의 뒷정리를 수행합니다.

- 스타트업 처리（상세）

처리	설명
코어 프로세스 기동	SpeeDBee Synapse의 데이터 엔진이 되는 코어 프로세스（백그라운드 프로세스）를 기동합니다. 상세는 후술합니다.
라이선스 키 등록	SpeeDBee Synapse의 라이선스 키를 등록합니다. WEBUI에서, 설정된 라이선스 키가 적용되어, 사용 가능한 기능 등이 제어됩니다.
환경 파라미터 설정	WEBUI에서 설정된 데이터베이스의 파라미터를 코어 프로세스에 적용합니다.
설정 정보 검증	WEBUI에서 설정되어 있는 컴포넌트 등의 정보를 검증합니다. 설정의 모순이 없는지 확인하여 정합성을 유지합니다.
설정 완료 컴포넌트・플로우 링크 등록 요구	사용자가 설정한 컴포넌트 및 플로우 링크를 코어 프로세스에 적용합니다.
모드 판정	모드 판정을 수행합니다. 모드에는 다음 종류가 있습니다. ・통상 모드 （스타트업 처리 내에서 컴포넌트의 기동까지를 수행하는 통상 모드입니다） ・세이프 모드 （이상 시에 전 컴포넌트가 기동할 수 없는 경우의 모드입니다）

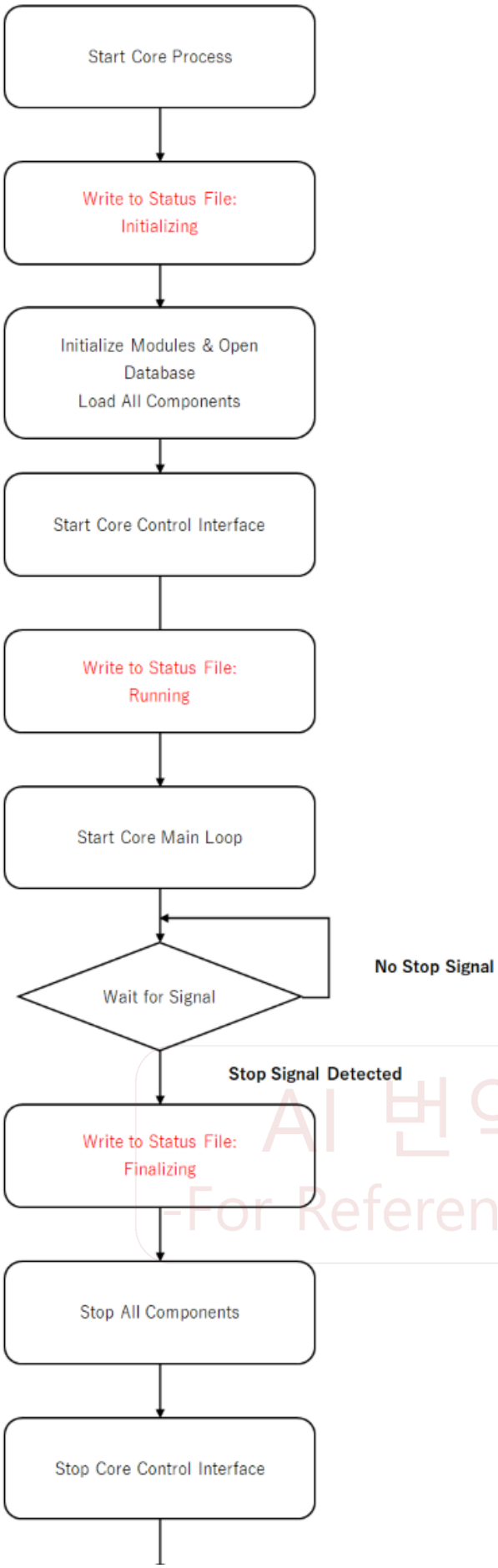
AI 번역본
-For Reference Only-

처리	설명
전 컴포넌트 기동 요구	등록 완료된 전 컴포넌트（자동 기동 무효인 것을 제외）를 기동합니다. 이에 따라 데이터 수집・가공・출력・송신 등의 처리가 실행되는 상태가 됩니다. 세이프 모드에서는 이 처리는 스킵됩니다.

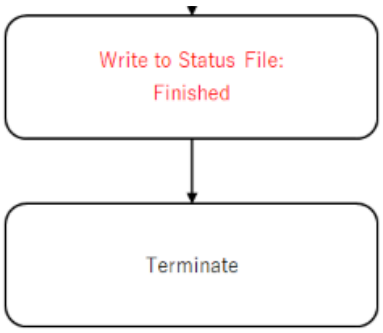
코어 프로세스 기동의 처리 상세는 다음과 같습니다.

- 코어 프로세스 기동（상세）

AI 번역본
-For Reference Only-



AI 번역본
-For Reference Only-



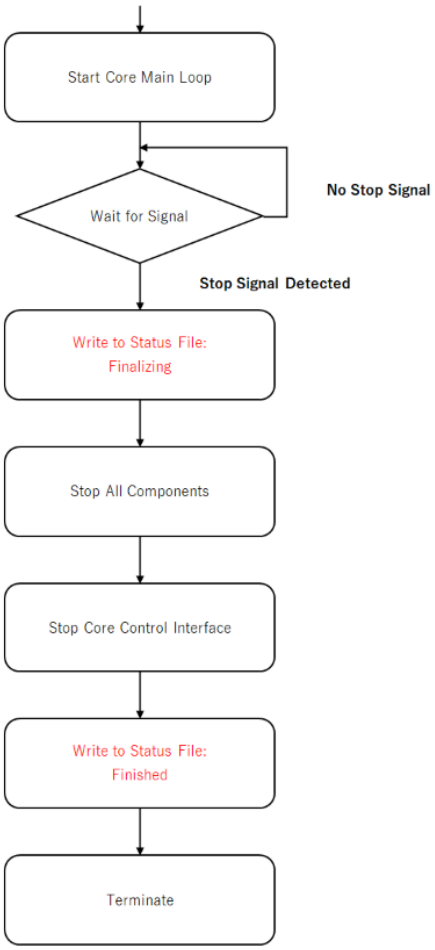
처리	설명
스테이터스 파일에 기록	코어의 기동・정지 상태 기록용 스테이터스 파일에 기록을 기입합니다. 이 단계에서는, 스테이터스는「Initializing」이 됩니다.
모듈 초기화・DB 오픈 전 컴포넌트의 읽어들이기	코어 모듈의 초기화를 수행합니다. 초기 중에, 데이터베이스의 오픈, 전 컴포넌트의 불러오기를 수행합니다. 이 단계에서, 문제가 있는 컴포넌트는 로드되지 않습니다. 배치는 되어 있지만, 초기화를 실시할 수 없는 것이 불러오기 대상 외가 됩니다.
코어 제어IF 개시	코어 프로세스를 제어하기 위한 인터페이스를 유효화합니다. WEBUI로부터의 요구를 수신할 수 있는 상태가 됩니다.
스테이터스 파일에 기록	코어의 기동・정지 상태 기록용 스테이터스 파일에 기록을 기입합니다. 이 단계에서는, 스테이터스는「Running」이 됩니다. 이 스테이터스가 기록됨으로써 정상적으로 기동했다는 것을 판별할 수 있습니다.
코어 메인 루프 개시	코어의 메인 루프가 개시됩니다. 코어 제어IF에 대한 요구에 따라, 각종 처리를 수행합니다.
시그널 접수	OS로부터의 시그널을 접수합니다. 정지 시그널을 수신하면, 종료 처리로 이행합니다.
스테이터스 파일에 기록	코어의 기동・정지 상태 기록용 스테이터스 파일에 기록을 기입합니다. 이 단계에서는, 스테이터스는「Finalizing」이 됩니다. 이 스테이터스가 기록됨으로써 종료 처리가 이루어졌다는 것을 판별할 수 있습니다. 다만, 종료 처리가 완전히 종료된 후에 다음 스테이터스가 기록되므로, 이 상태 그대로 스테이터스가 남아 있는 경우는 이상으로 판단됩니다.
전 컴포넌트 정지	기동 중인 전 컴포넌트를 정지, 파기합니다.
코어 제어IF 정지	코어 제어IF를 정지합니다. 외부로부터의 요구는 접수할 수 없게 됩니다.

AI 번역본
For reference only

처리	설명
스테이터스 파일에 기록	코어의 기동・정지 상태 기록용 스테이터스 파일에 기록을 기입합니다. 이 단계에서는, 스테이터스는「Finished」가 됩니다. 이 스테이터스가 기록됨으로써 종료 처리가 정상적으로 완료되었다는 것을 판별할 수 있습니다. 다음 회 코어 프로세스 기동 시에는, 이 스테이터스를 참조하여, 전회 기동 시의 이상 진단을 수행합니다.「Finished」 이외의 스테이터스가 기록된 것을 검지하면, 코어 인터널 컴포넌트는 이상 상태를 스테이터스 등록합니다. 이상이 있었던 경우의 정보는, 스테이터스 타임라인에서 참조할 수 있습니다.

5.2.2.2 정상 정지 시

정상 정지 시의 동작에 대해 해설합니다. 아래 그림은, 코어 프로세스 기동（상세）의 발췌입니다. 빨간 테두리 부분이 중요한 처리입니다.



번역본
eference Only-

정상적인 정지 처리가 실행됨으로써, 스테이터스 파일에 올바른 기록이 이루어집니다. 스테이터스 파일에 「 Finished 」가 기록되어 있는 상태가 올바른 정지가 이루어졌음을 나타냅니다. 상세는, 코어 프로세스 기동（ 상세 ）를 참조하십시오.

5.2.2.3 이상 정지 시(전원 차단 등)

이상 시 정지 시의 동작에 대해 해설합니다. 이상 시란, SpeeDBee Synapse 가동 중에 전원 차단에 의해 호스트 머신이 정지한 것 등을 가리킵니다.

- 코어 프로세스의 이상 정지 진단

정상 정지 시의 설명에 기재한 바와 같이, 가동 도중에 SpeeDBee Synapse가 정지한 경우는, 스테이터스 파일에 올바른 기록이 이루어지지 않습니다. 다음 회 코어 프로세스 기동 시에, 스테이터스 파일의 진단을 수행하여, 정상 정지가 확인되지 않는 경우는, 코어 인터널의 에러 스테이터스로서 상태가 등록됩니다. 에러 정보에 대해서는, 스테이터스 타임라인에서 참조할 수 있습니다.

- 데이터베이스 진단

데이터베이스 기입 중에, SpeeDBee Synapse가 이상 정지를 일으킨 경우, 타이밍에 따라서는 데이터베이스 내의 데이터 결손이 발생할 가능성이 있습니다. 데이터 결손이란, 메모리 상에, 보유하고 있던 데이터를 스토리지에 기입 하고 있는 도중에, 전원 차단 등에 의해 정상적으로 기입이 종료되지 않아 일부 데이터가 소실되는 것을 가리킵니다. 이 진단도, 다음 회 코어 프로세스 기동 시에 내부에서 진단됩니다. 진단 결과에서, 이상이 발견되면 코어 인터널의 에러 스테이터스로서 상태가 등록됩니다. 에러 정보에 대해서는, 스테이터스 타임라인에서 참조할 수 있습니다.

5.2.2.4 데이터 수집에 관련된 동작 사양에 대해

SpeeDBee Synapse에서는, 컴포넌트의 데이터 수집을 기점으로 처리를 수행합니다. 다만, 설정이나 조작의 타이밍에 따라 주의해야 할 동작 사양이 있으므로, 이하에 기재합니다.

AI 번역본
-For Reference Only-

컴포넌트의 설정 변경이 적용되는 타이밍에 대해

설정을 변경한 경우, 그것이 즉시 적용되는 것과, 일단 컴포넌트를 정지한 후, 재기동 후에 적용되는 것이 있습니다. 또한, 적용 타이밍이 특수한 설정이 있습니다.

1. 컴포넌트 재기동 후에, 새로운 설정이 적용되는 것

- 컴포넌트 각각의 고유 설정

컴포넌트 고유의 설정 항목은, 재기동 후에 반영됩니다.
예를 들어, PLC 컬렉터 등에서 주소를 추가하여, 설정을 저장해도, 그대로는 설정이 적용되지 않습니다. 일단,
PLC 컬렉터를 정지하고, 다시 기동함으로써 새로운 설정이 반영된 상태로 동작합니다.

2. 설정 변경 후, 즉시 새로운 설정이 적용되는 것

- 플로우 링크의 필터
- 플로우 링크의 연결/삭제
- 입력 포트 설정

3. 설정 변경 후, 특수한 타이밍으로 새로운 설정이 적용되는 것

- 출력 포트

출력 포트에 관해서는, 설정 변경 후에 컬럼이 작성됨으로써, 새로운 설정이 적용됩니다.
이는 기동 시에, 컬럼을 작성하는 컴포넌트라면 재기동될 때까지 적용되지 않지만, 정규 표현식 등으로 설정을 하고 있는 경우이자
특수한 타이밍으로 컬럼을 생성하는 컴포넌트의 경우는, 도중에 반영될 가능성이 있습니다.

예를 들어, MQTT 컬렉터 등은 데이터를 수신하여, 비로소 컬럼을 생성하는 경우가 있습니다. 이 경우에, 출력 포트의 설정을 정규 표현식 등으로 지정하고 있으면, 새로운 설정이 적용될 가능성이 있습니다.
※확실하게 설정을 적용하고 싶은 경우는, 해당 컴포넌트를 재기동해 주십시오.

플로우 링크 연결/삭제 시의 데이터 흐름에 대해

데이터 수집을 수행하는 컴포넌트를 기동한 후, 다른 컴포넌트로 플로우 링크를 연결해도 연결하기 전의 데이터는
수집되지 않습니다.
또한, 플로우 링크를 도중에 삭제하면, 그 타이밍에 연결처로 해당 데이터가 흐르지 않게 되어, 해당 컴포넌트가 처리하는 데이터가 변화합니다. 이는 설정을 변경하지 않아도, 흐르는 데이터가 변화하기 때문에 동작하고 있는 컴포넌트의 거동이 변화합니다.

AI 번역본
-For Reference Only-

예를 들어, PLC 컬렉터와 Modbus TCP 컬렉터를 JSON 시리얼라이저에 플로우 링크를 각각 연결하고 있다고 하겠습니다. 이때, PLC 컬렉터와 JSON 시리얼라이저의 플로우 링크를 삭제하면, JSON 시리얼라이저가 참조하는 데이터가, 도중에 Modbus TCP 컬렉터의 데이터만이 됩니다.

데이터의 영속화에 대해

데이터 수집하는 컴포넌트가 취득한 데이터는, 디폴트로는 DB에 영속화하지 않습니다. DB에 영속화하면,
Grafana나 DB쿼리 서비스를 사용하여, 영속화되어 있는 기간의 과거 데이터를 참조할 수 있습니다. 영속화하지 않은 경우는, 현재 메모리 상에 보유하고 있는 데이터만이 참조 가능하게 됩니다.

이들 과거 데이터 참조를 하는 데이터를 영속화하고 싶은 경우는,「출력 포트」의「영속화의 설정」의 설정을 수행해 주십시오.

데이터 참조 시의 지연 시간에 대해

입력 포트에는,「지연 시간」이라는 설정이 있습니다.
이는 데이터 참조 시에, 취득하는 데이터의 시간 범위를 현재 시각에 대해 몇 초 늦추어 수행할지 설정하는 것입니다.
데이터 수집과 데이터 참조가 비동기로 병렬 처리되기 때문에, 데이터 수집이 완전히 완료되지 않은 데이터를 놓치는 것을 방지하는 설정입니다.
데이터 수집에 시간이 걸리는 컴포넌트의 데이터를 다루고 있는 경우는, 이 지연 시간을 조정하십시오.
컴포넌트에 따라서는, 데이터 수집의 시간 보증을 수행하여, 이 지연 시간보다 짧은 간격으로 데이터를 참조할 수 있는 경우도 있습니다.
이는, SpeeDBee Synapse 내부에서 자동으로 이루어집니다.

지연 시간의 설정을 수행하려면,「입력 포트」를 참조하십시오.

컴포넌트 정지 시의 데이터에 대해

컴포넌트를 정지하면, 그 조작 직전의 데이터는 완전히 보증되지 않습니다.

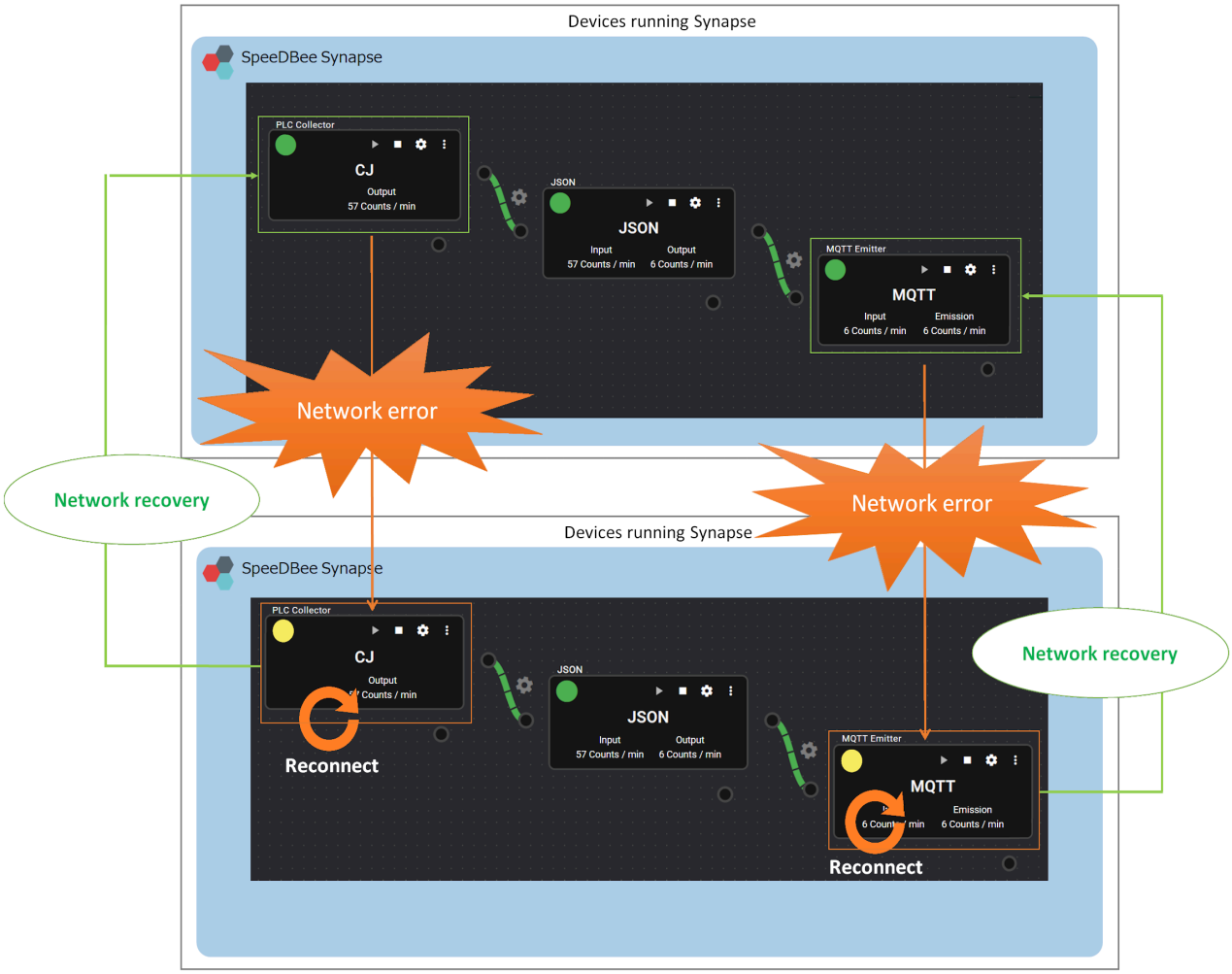
예를 들어 임의의 컬렉터로부터 시리얼라이저로 플로우 링크를 연결하고 있는 경우에, 시리얼라이저의 정지를 수행하면, 시리얼라이저는 되도록 빨리 정지하도록 동작합니다. 따라서, 정지 중이 된 시리얼라이저는, 데이터 참조를 실행하지 않기 때문에, 데이터는 보증되지 않습니다.
처리하고 싶은 데이터가 흐르고 있는 동안, 부주의하게 컴포넌트를 정지하지 마십시오.

AI 번역본
-For Reference Only-

5.2.2.5 네트워크 이상, 복구 시의 동작 사양에 대해

SpeeDBee Synapse의 네트워크 이상 시, 그리고, 네트워크 이상으로부터의 복구 시에 관한 동작 사양에 대해 설명합니다.

기본적으로는, 통신을 사용하는 컴포넌트는 네트워크 이상이 발생한 경우, 통신의 재접속을 수행합니다.



1. 데이터 수집을 수행하는 컴포넌트

   통신을 사용한 컬렉터 계열의 컴포넌트는, 데이터 수집 중에 네트워크 이상이 발생한 경우는, 통신의 재접속을 시도합니다.

   통신에 이상이 있는 동안은, 네트워크 이상 상태가 되어 스테이터스 램프가 경고 상태가 되니다. 네트워크 이상이 발생하고 있는 동안의 데이터는 수집할 수 없습니다. 가동 머신의 네트워크가 복구 되고, 재접속에 성공하면, 통상의 데이터 수집을 재개합니다.

   해당하는 컴포넌트는, 다음과 같습니다.

   - PLC 컬렉터
   - Modbus TCP 컬렉터
   - EtherNet/IP 컬렉터
   - VIDEO 컬렉터
   - MQTT 컬렉터

AI 번역본
-For Reference Only-

2. 수집한 데이터를 송신하는 컴포넌트
   통신을 사용한 액션 계열 컴포넌트, 에미터 계열 컴포넌트는, 동작 중에 네트워크 이상이 발생한 경우는, 통신의 재접속을 시도합니다. 통신에 이상이 있는 동안은, 네트워크 이상 상태가 되어 스테이터스 램프가 경고 상태가 됩니다.

   액션 계열 컴포넌트
   네트워크 이상이 발생하고 있는 동안의 데이터는 송신할 수 없습니다.
   해당하는 컴포넌트는, 다음과 같습니다.

   - PLC 쓰기
   - Modbus 쓰기
   - EtherNet/IP 쓰기
   - 메일 송신

   에미터 계열 컴포넌트
   네트워크 이상이 발생하고 있는 동안의 데이터는 송신할 수 없습니다. 다만, 재시도 모드의 설정에 의해 송신되지 않았던 데이터의 재송신을 수행하는 설정도 가능합니다.
   해당하는 컴포넌트는, 다음과 같습니다.

   - Cloud 에미터(Azure)
   - Cloud 에미터(AWS)
   - FTP/FTPS/SFTP 에미터
   - MQTT 에미터

FTP/FTPS/SFTP 에미터는, 항상 미송신 데이터는 항상 재송신을 시도합니다. 그 외 에미터의 재시도 모드에 대해서는, 해당 컴포넌트의 기능 설명을 참조하십시오.
또한, 사용자가 작성한 커스텀 컴포넌트나 셸 커맨드로 통신 처리를 수행하는 경우의 동작은, 실장에 따라 달라지집니다.

네트워크 이상 시에, 통신의 재접속을 수행할 필요가 있는 경우는, 각각의 실장 시에, 재접속하는 처리를 실장할 필요가 있습니다.

AI 번역본
-For Reference Only-

5.3 플로우 에디터

여기서는, 플로우 에디터에 대해 설명합니다. 플로우 에디터는, SpeeDBee Synapse의 조작 화면입니다.

5.3.1 모드에 대해

플로우 에디터에는 2종류의 모드가 존재합니다.

모드	설명
Edit Mode	데이터 수집 플로우의 설정 등을 수행합니다. 권한이 시스템 관리자・관리자인 사용자만 이용할 수 있습니다.
Run Mode	데이터 수집 플로우의 운용을 수행합니다. 모든 권한의 사용자가 이용할 수 있습니다.

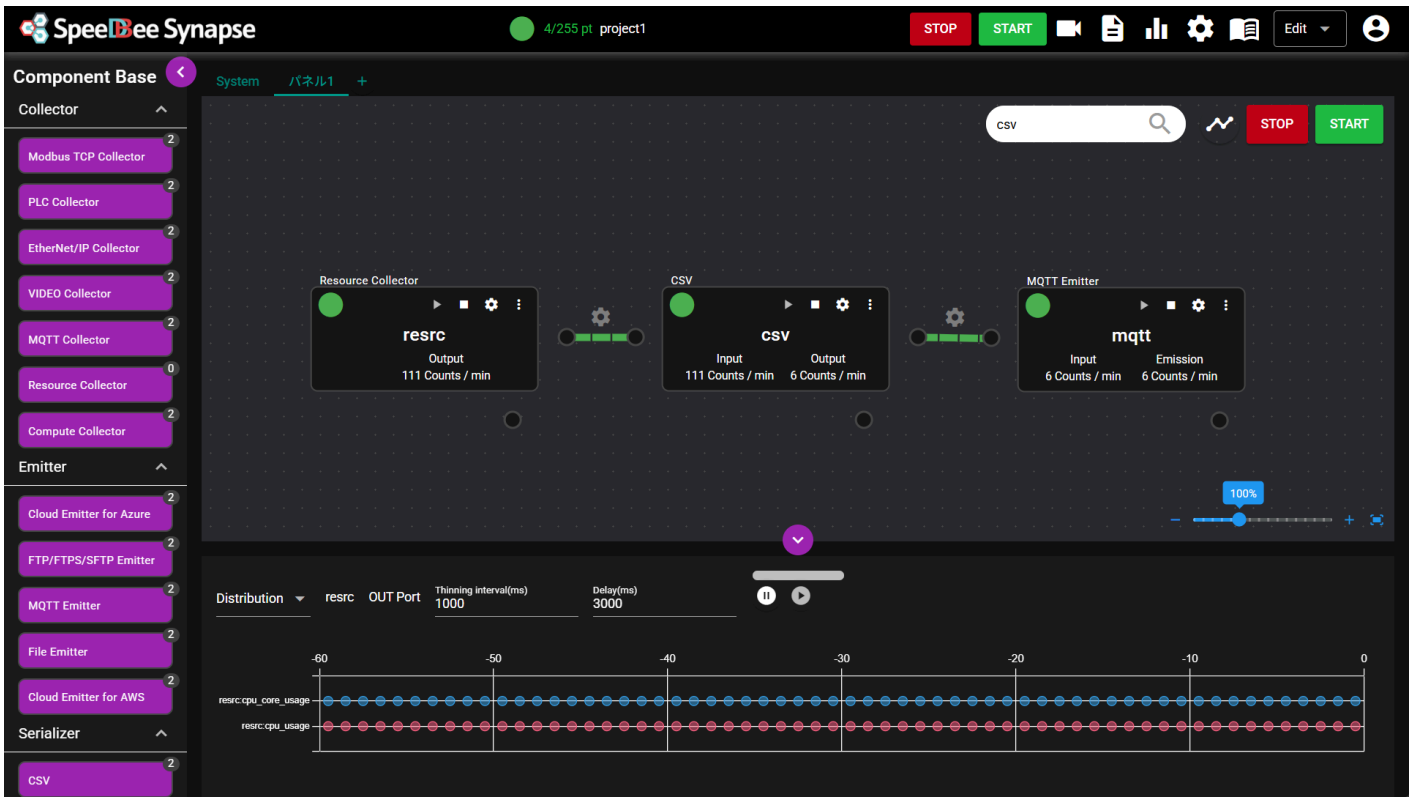
5.3.2 화면 항목

여기서는, 플로우 에디터의 화면 항목에 대해 설명합니다.

5.3.2.1 Edit Mode

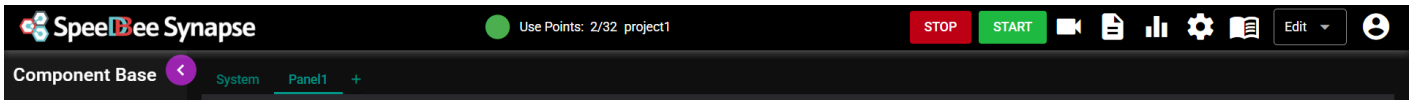
Edit Mode의 플로우 에디터의 화면 항목은 다음과 같습니다.

AI 번역본
-For Reference Only-



Edit Mode의 화면은, ① 헤더, ② 컴포넌트 베이스, ③ 패널, ④ 데이터 모니터 로 구성되어 있습니다.

① 헤더



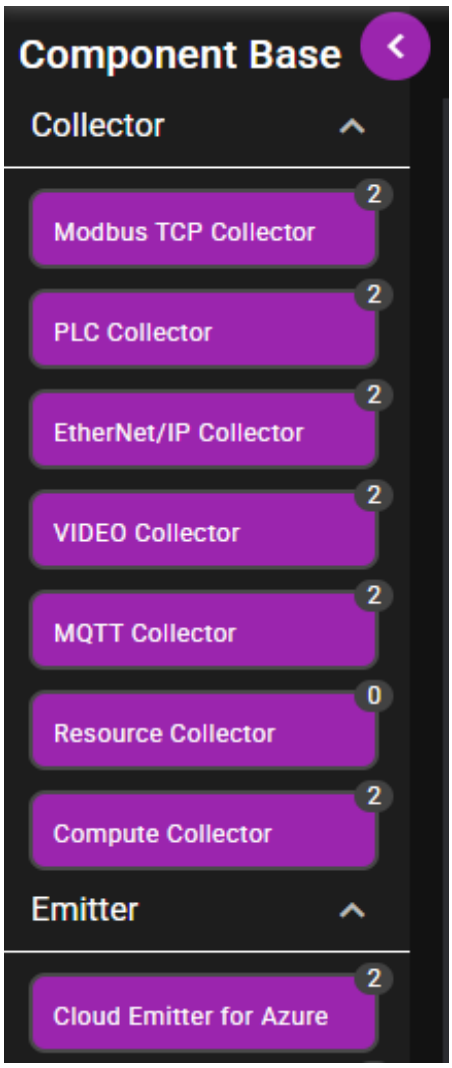
번호	항목	설명
①	사용 포인트 수	등록된 라이선스의 포인트 수와, 사용된 포인트 수가 표시됩니다.
②	현재의 프로젝트	현재 선택 중인 프로젝트가 표시됩니다. ※상세는「설정 정보의 관리(프로젝트)」를 참조하십시오.
③	정지 버튼	클릭하면, SpeeDBee Synapse에 등록된 모든 컴포넌트가 정지합니다.
④	기동 버튼	클릭하면, SpeeDBee Synapse에 등록되어 있는,「자동 기동 무효」가 OFF인 컴포넌트가 모두 기동합니다.
⑤	뷰어 아이콘	클릭하면, 뷰어 화면으로 이동합니다. ※확장 패키지「i-Trigger」를 적용한 경우에만 표시됩니다. 상세는「i-Trigger（확장 패키지）」를 참조하십시오.

AI 번역본
-For Reference Only-

번호	항목	설명
⑥	로그 아이콘	클릭하면, SpeeDBee Synapse의 조작 로그 화면이 표시됩니다. ※상세는「각종 로그」를 참조하십시오.
⑦	그래프 아이콘	클릭하면, Grafana의 화면이 표시됩니다. ※상세는「Grafana와의 연계」를 참조하십시오.
⑧	설정 메뉴 아이콘	클릭하면, 설정 메뉴가 표시됩니다. ※상세는「설정 메뉴」를 참조하십시오.
⑨	매뉴얼	클릭하면, 매뉴얼 화면이 표시됩니다.
⑩	모드 선택	플로우 에디터의 표시 모드를, 다음 중에서 선택할 수 있습니다. ・Edit Mode ・Run Mode ※권한이 시스템 관리자・관리자인 사용자로 로그인한 경우에만 표시되됩니다.
⑪	사용자 관리 아이콘	클릭하면, 사용자 관리의 메뉴가 표시됩니다. ※상세는「사용자 관리」를 참조하십시오.

② 컴포넌트 베이스

AI 번역본
-For Reference Only-



번호	항목	설명
①	개폐 아이콘	클릭함으로써, 컴포넌트 베이스의 표시・비표시를 전환할 수 있습니다.
②	인디케이터	클릭함으로써, 각 항목의 표시・비표시를 전환할 수 있습니다.
③	컴포넌트 베이스	패널에 등록할 수 있는 컴포넌트입니다. 중앙에, 베이스명이 표시됩니다. 오른쪽 위에, 등록 시에 소비되는 포인트 수가 표시됩니다. 또한, 마우스 포인터를 맞추면 컴포넌트의 설명, 입출력 포트의 유무가 추가로 표시됩니다.

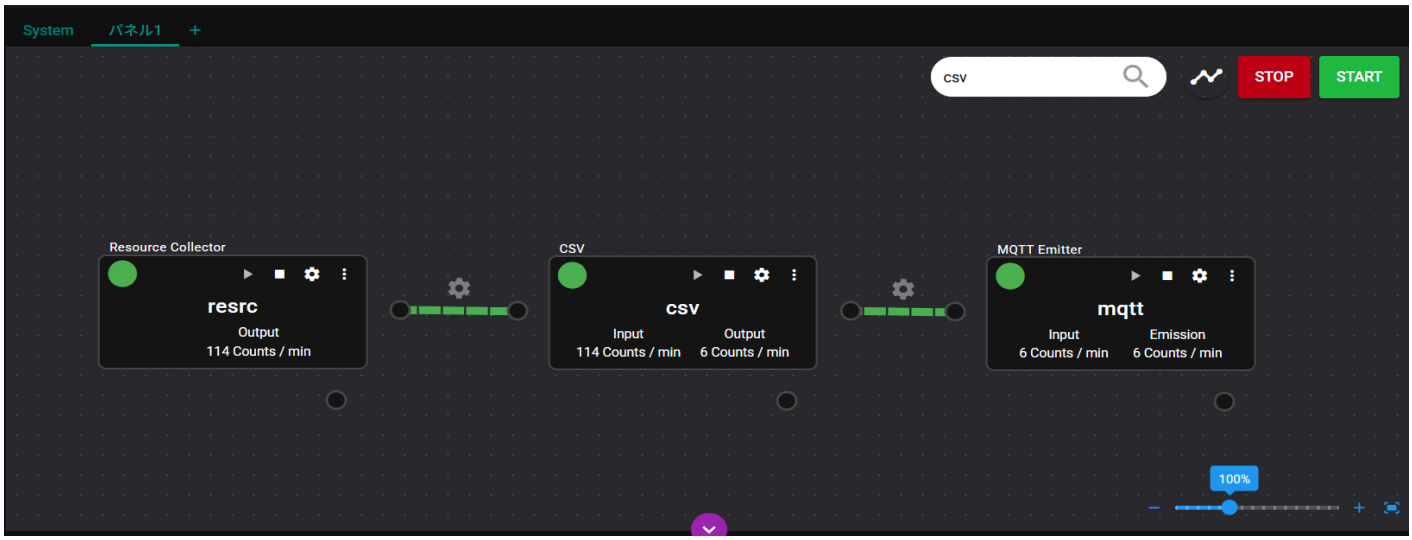
AI 번역본
-For Reference Only-

- 컴포넌트 베이스에는 다음 종류가 있습니다.

종류	링크
Collector	상세는「컬렉터」를 참조하십시오.

종류	링크
Serializer	상세는「시리얼라이저」를 참조하십시오.
Emitter	상세는「에미터」를 참조하십시오.
Logic	상세는「로직（이벤트）」,「로직（실시간 분석/데이터 조작）」을 참조하십시오.
Action	상세는「액션」을 참조하십시오.
Custom	상세는「커스텀 컴포넌트의 작성・이용」의「개요」를 참조하십시오. ※커스텀 컴포넌트가 추가되어 있는 경우에만 표시됩니다.

③ 패널



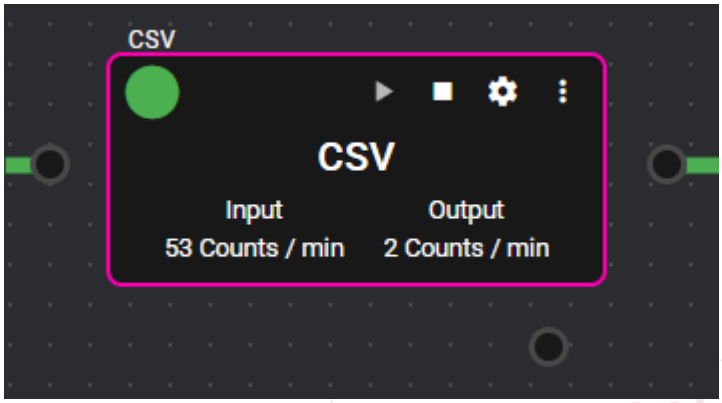
번호	항목	설명
①	탭	표시할 패널을 전환하기 위한 표제입니다.
②	패널 추가 아이콘	클릭하면, 새로운 패널을 추가할 수 있습니다.
③	컴포넌트 검색 박스	컴포넌트의 위치를 검색할 수 있는 박스입니다.
④	스테이터스 타임라인	클릭하면, 컴포넌트의 스테이터스를 시계열 순으로 나열한 화면이 표시됩니다. ※상세는「스테이터스 타임라인」을 참조하십시오.

AI 번역본
For Reference Only-

번호	항목	설명
⑤	정지 버튼	클릭하면, 패널 상의 모든 컴포넌트가 정지합니다.
⑥	기동 버튼	클릭하면, 패널 상의,「 자동 기동 무효 」가 OFF인 컴포넌트가 모두 기동합니다.
⑦	컴포넌트	패널에 등록되어 있는 컴포넌트입니다.
⑧	플로우 링크	패널에 등록되어 있는 플로우 링크입니다. ※플로우 링크는, 2개의 컴포넌트를 연결하여, 컴포넌트 간에 데이터를 흘려보내는 오브젝트입니다. 상세는「 플로우 링크 」를 참조하십시오.
⑨	표시 배율	패널의 표시 배율을 변경할 수 있습니다.

- 상세는「 패널 」을 참조하십시오.
- "시스템"이라는 이름의 패널은, 시스템 컴포넌트 전용 패널입니다.

③-① 컴포넌트

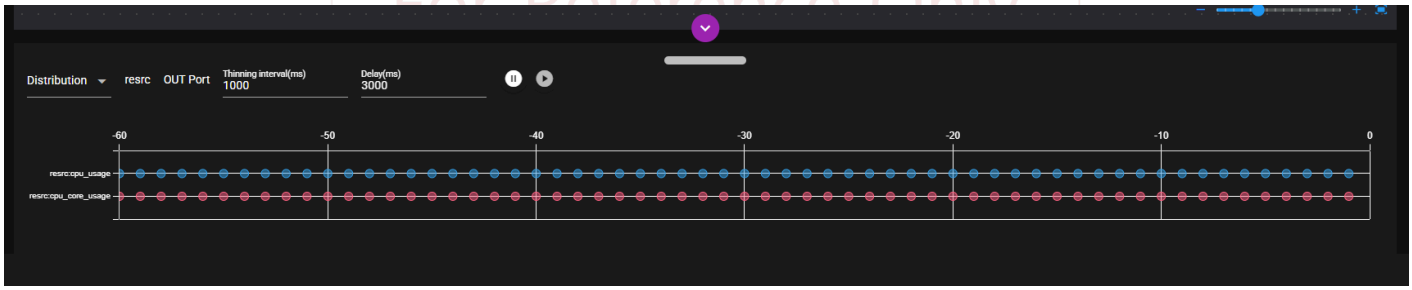


번호	항목	설명
①	입력 포트	데이터를 입력받을 포트입니다. ※1 클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되 됩니다. 상세는「 데이터 모니터 」를 참조하십시오. ※2 오른쪽 클릭하면, 설정을 수행할 수 있습니다. 상세는「 입력 포트 」를 참조하십시오.
②	컴포넌트 베이스명	컴포넌트의 베이스명이 표시됩니다.

번호	항목	설명
③	스테이터스 램프	컴포넌트의 가동 상황에 따른 색이 표시됩니다. ※상세는「 스테이터스 목록 」을 참조하십시오.
④	기동 아이콘	클릭하면, 컴포넌트가 기동합니다.
⑤	정지 아이콘	클릭하면, 컴포넌트가 정지합니다.
⑥	설정 아이콘	클릭하면, 컴포넌트의 설정 내용을 표시합니다.
⑦	메뉴 아이콘	클릭하면, 컴포넌트의 각종 포트 설정, 로그, 스테이터스 타임라인 등의 메뉴를 표시합니다.
⑧	컴포넌트명	컴포넌트명이 표시됩니다.
⑨	데이터량 모니터	컴포넌트가 다루고 있는 데이터량이 표시됩니다.
⑩	출력 포트	데이터를 출력하는 포트입니다. ※1 클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되 됩니다. 상세는「 데이터 모니터 」를 참조하십시오. ※2 오른쪽 클릭하면, 설정을 수행할 수 있습니다. 상세는 「 출력 포트 」를 참조하십시오.
⑪	시스템 포트	시스템에 관련된 데이터를 출력하는 포트입니다. ※1 클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되 됩니다. 상세는「 데이터 모니터 」를 참조하십시오. ※2 오른쪽 클릭하면, 설정을 수행할 수 있습니다. 상세는「 시스템 포트 」를 참조하십시오.

- 상세는「 컴포넌트 」를 참조하십시오.

④ 데이터 모니터



AI 번역본
For Reference Only

번호	항목	설명
①	개폐 아이콘	클릭함으로써, 데이터 모니터의 표시・비표시를 전환할 수 있습니다.
②	표시 사이즈 조정 그립	드래그하여 마우스를 상하로 움직임으로써, 데이터 모니터의 표시 사이즈를 조정할 수 있습니다.
③	표시 종류 선택 폼	데이터 모니터의 표시 방법을 선택할 수 있습니다.
④	선택 포트	선택한 포트가 표시됩니다.
⑤	솎아내기 간격	표시할 데이터의 솎아내기 간격을 설정할 수 있습니다.
⑥	지연 시간	표시할 데이터의 지연 시간을 설정할 수 있습니다.
⑦	모니터 표시 에어리어	선택한 종류의 모니터가 표시됩니다.

- 상세는「 데이터 모니터 」를 참조하십시오.

5.3.2.2 Run Mode

Run Mode의 플로우 에디터의 화면 항목은 다음과 같습니다.

AI 번역본
-For Reference Only-



Run Mode의 화면은, ① 헤더, ② 패널, ③ 패널 스테이터스, ④ 데이터 모니터, ⑤ 시스템 리소스로 구성되어 있습니다.

① 헤더

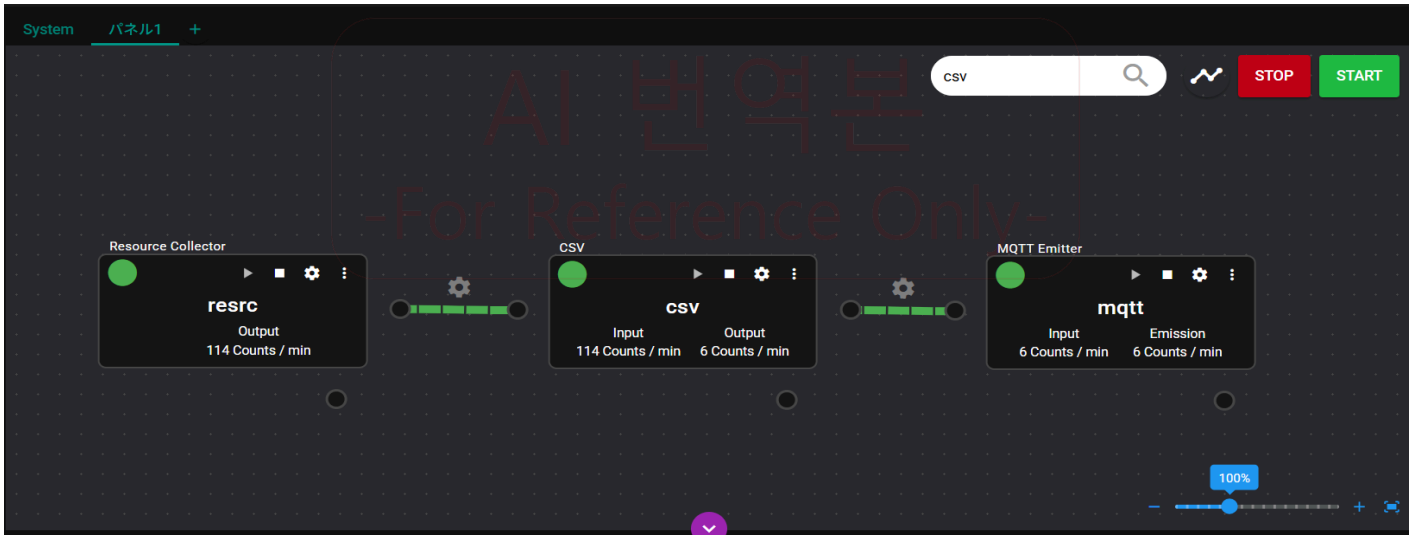


번호	항목	설명
①	사용 포인트 수	등록된 라이선스의 포인트 수와, 사용된 포인트 수가 표시됩니다.
②	현재 프로젝트	현재 선택 중인 프로젝트가 표시됩니다. ※상세는「설정 정보의 관리(프로젝트)」를 참조하십시오.
③	정지 버튼	클릭하면, SpeeDBee Synapse에 등록된 모든 컴포넌트가 정지합니다.
④	기동 버튼	클릭하면, SpeeDBee Synapse에 등록되어 있는,「자동 기동 무효」가 OFF인 컴포넌트가 모두 기동합니다.

AI 번역본
-For Reference Only-

번호	항목	설명
⑤	뷰어 아이콘	클릭하면, 뷰어 화면으로 이동합니다. ※확장 패키지「i-Trigger」를 적용한 경우에만 표시됩니다. 상세는「i-Trigger（확장 패키지）」를 참조하십시오.
⑥	로그 아이콘	클릭하면, SpeeDBee Synapse의 조작 로그 화면이 표시됩니다. ※상세는「각종 로그」를 참조하십시오.
⑦	그래프 아이콘	클릭하면, Grafana의 화면이 표시됩니다. ※상세는「Grafana와의 연계」를 참조하십시오.
⑧	설정 메뉴 아이콘	클릭하면, 설정 메뉴가 표시됩니다. ※선택할 수 있는 메뉴는 언어 설정뿐입니다. 상세는「설정 메뉴」를 참조하십시오.
⑨	매뉴얼	클릭하면, 매뉴얼 화면이 표시됩니다.
⑩	모드 선택	플로우 에디터의 표시 모드를, 다음 중에서 선택할 수 있습니다. ・Edit Mode ・Run Mode ※권한이 시스템 관리자・관리자인 사용자로 로그인한 경우에만 표시됩니다.
⑪	사용자 관리 아이콘	클릭하면, 사용자 관리의 메뉴가 표시됩니다. ※상세는「사용자 관리」를 참조하십시오.

② 패널

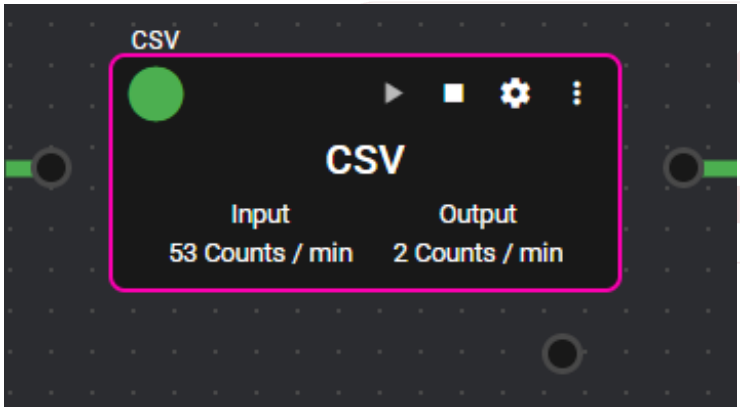


AI 번역본
-For Reference Only-

번호	항목	설명
①	탭	표시할 패널을 전환하기 위한 표제입니다.
②	컴포넌트 검색 상자	컴포넌트의 위치를 검색할 수 있는 상자입니다.
③	스테이터스 타임라인	클릭하면, 컴포넌트의 스테이터스를 시계열순으로 나열한 화면이 표시됩니다. ※상세는「 스테이터스 타임라인 」을 참조하십시오.
④	정지 버튼	클릭하면, 패널상의 모든 컴포넌트가 정지합니다.
⑥	기동 버튼	클릭하면, 패널상의,「 자동 기동 무효 」가 OFF인 컴포넌트가 모두 기동합니다.
⑥	컴포넌트	패널에 등록되어 있는 컴포넌트입니다. ※컴포넌트는, 데이터의 수집, 가공, 분석, 송신 등을 실현하는 모듈군입니다.
⑦	플로우 링크	패널에 등록되어 있는 플로우 링크입니다.
⑧	표시 배율	패널의 표시 배율을 변경할 수 있습니다.

- 상세는「 패널 」을 참조하십시오.
- "시스템"이라는 이름의 패널은, 시스템 컴포넌트 전용 패널입니다.

②-① 컴포넌트



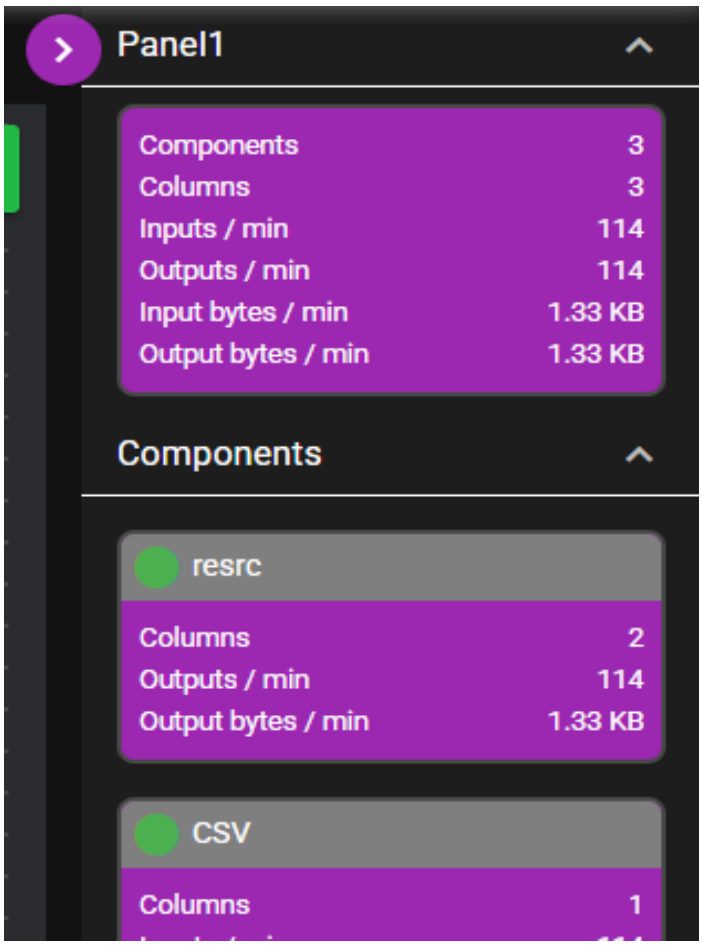
역본
nce Only-

번호	항목	설명
①	입력 포트	데이터를 입력 받을 포트입니다. ※클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되습니다. 상세는 「 데이터 모니터 」를 참조하십시오.
②	컴포넌트 베이스명	컴포넌트의 베이스명이 표시됩니다.
③	스테이터스 램프	컴포넌트의 가동 상황에 따른 색이 표시됩니다. ※상세는 「 스테이터스 목록 」을 참조하십시오.
④	기동 아이콘	클릭하면, 컴포넌트가 기동합니다.
⑤	정지 아이콘	클릭하면, 컴포넌트가 정지합니다.
⑥	설정 아이콘	클릭하면, 컴포넌트의 설정 내용을 표시합니다.
⑦	메뉴 아이콘	클릭하면, 컴포넌트의 로그, 스테이터스 타임라인 등의 메뉴가 표시됩니다.
⑧	컴포넌트명	컴포넌트명이 표시됩니다.
⑨	데이터양 모니터	컴포넌트가 다루고 있는 데이터양이 표시됩니다.
⑩	출력 포트	데이터를 출력하는 포트입니다. ※클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되습니다. 상세는 「 데이터 모니터 」를 참조하십시오.
⑪	시스템 포트	시스템에 관련된 데이터를 출력하는 포트입니다. ※클릭하면, 포트에 흐르고 있는 데이터가 데이터 모니터에 표시되습니다. 상세는 「 데이터 모니터 」를 참조하십시오.

-For Reference Only-

- 상세는 「 컴포넌트 」를 참조하십시오.

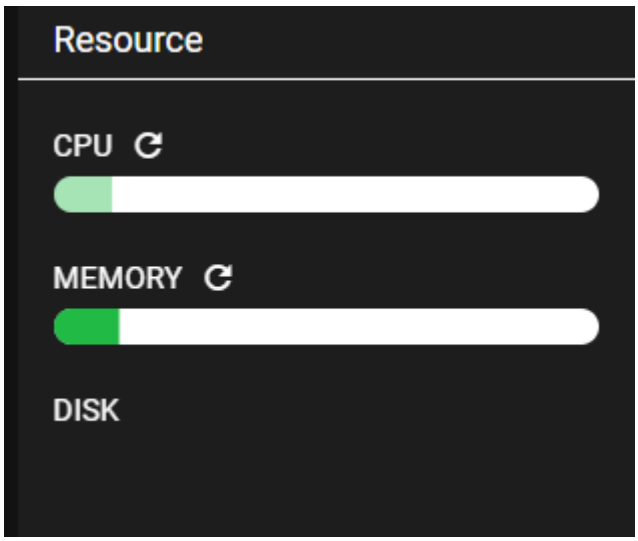
③ 패널 스테이터스



번호	항목	설명
①	개폐 아이콘	클릭함으로써, 패널 스테이터스의 표시・비표시를 전환할 수 있습니다.
②	인디케이터	클릭함으로써, 각 항목의 표시・비표시를 전환할 수 있습니다.
③	패널 스테이터스	패널의 스테이터스가 표시됩니다.
④	컴포넌트 스테이터스	컴포넌트의 스테이터스가 표시됩니다.

- 상세는「패널 스테이터스」를 참조하십시오.

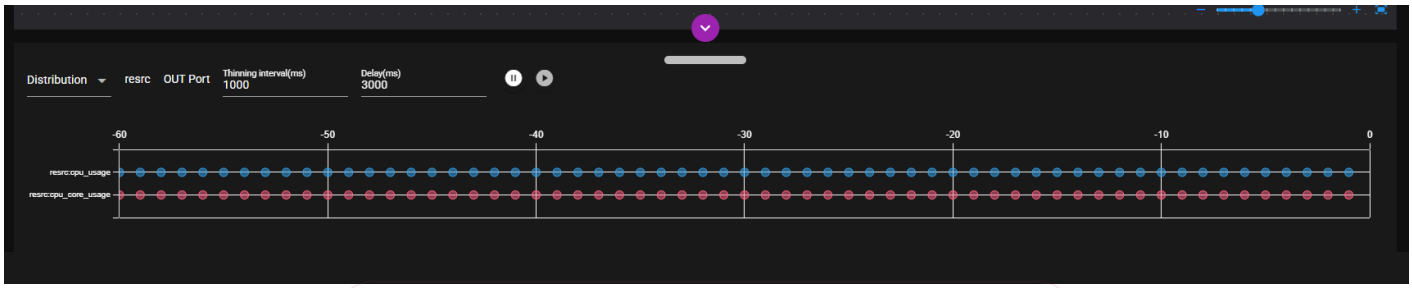
④ 시스템 리소스



번호	항목	설명
①	리프레시 아이콘	클릭하면, 표시 내용이 갱신됩니다.
②	게이지	리소스의 비율이 표시됩니다.

- 상세는「 시스템 리소스 」를 참조하십시오.

⑤ 데이터 모니터



AI 번역본
-For Reference Only-

번호	항목	설명
①	개폐 아이콘	클릭함으로써, 데이터 모니터의 표시・비표시를 전환할 수 있습니다.
②	표시 사이즈 조정 그립	드래그하여 마우스를 상하로 움직임으로써, 데이터 모니터의 표시 사이즈를 조정할 수 있습니다.
③	표시 종류 선택 폼	데이터 모니터의 표시 방법을 선택할 수 있습니다.
④	선택 포트	선택한 포트가 표시됩니다.

번호	항목	설명
⑤	솎아내기 간격	표시할 데이터의 솎아내기 간격을 설정할 수 있습니다.
⑥	지연 시간	표시할 데이터의 지연 시간을 설정할 수 있습니다.
⑦	모니터 표시 에어리어	선택한 종류의 모니터가 표시됩니다.

- 상세는 「 데이터 모니터 」를 참조하십시오.

5.3.3 화면 조작

여기에서는, 플로우 에디터의 조작 방법에 대해 설명합니다.

5.3.3.1 설정 메뉴

설정 메뉴의 조작에 대해 설명합니다.

> 설정 메뉴의 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.1.1 언어 설정

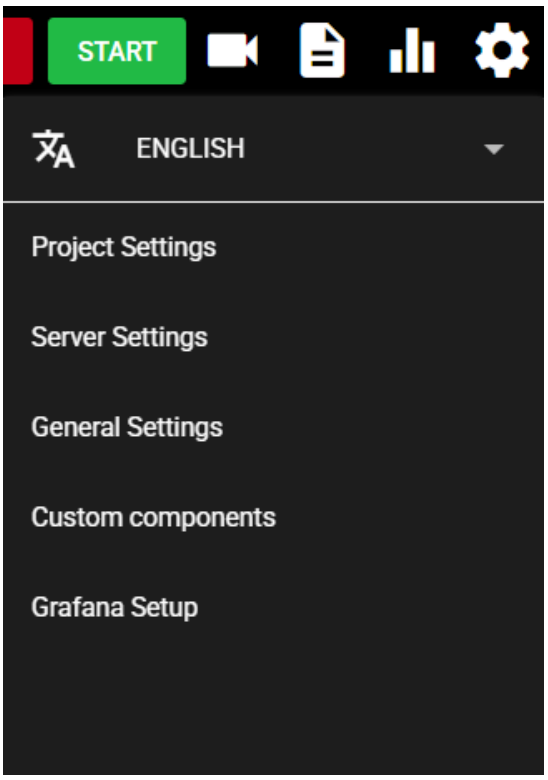
다음 순서로, 언어의 전환을 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

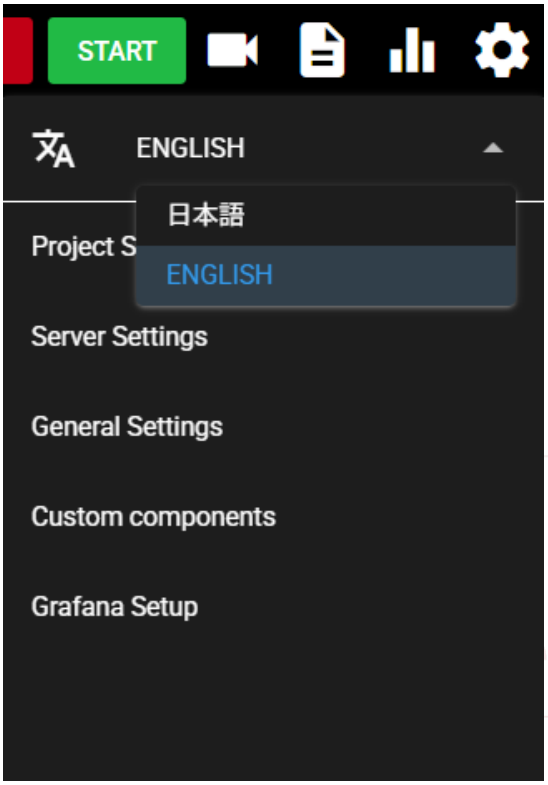


2. 언어 선택 풀다운 메뉴를 선택합니다.

AI 번역본
-For Reference Only-



3. 언어를 선택합니다.



번역본
ference Only-

4. 선택한 언어가 SpeeDBee Synapse에 적용됩니다.

5.3.3.1.2 프로젝트 설정

프로젝트의 설정을 수행할 수 있습니다. 상세는「설정 정보의 관리(프로젝트)」를 참조하십시오.

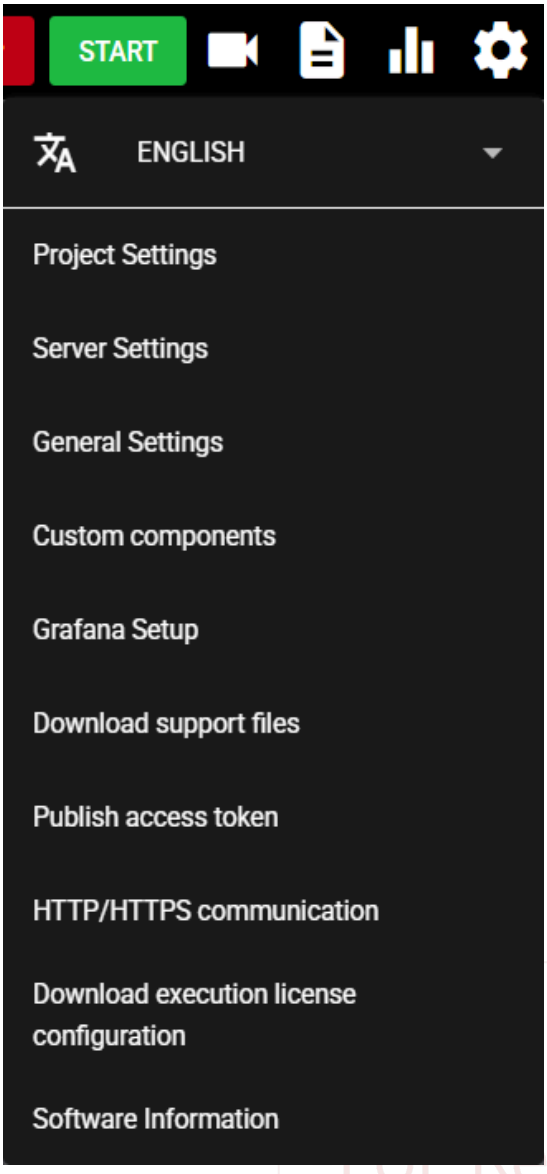
5.3.3.1.3 서버 설정

다음 순서로, 서버 설정의 확인 및 갱신, 데이터베이스의 초기화를 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

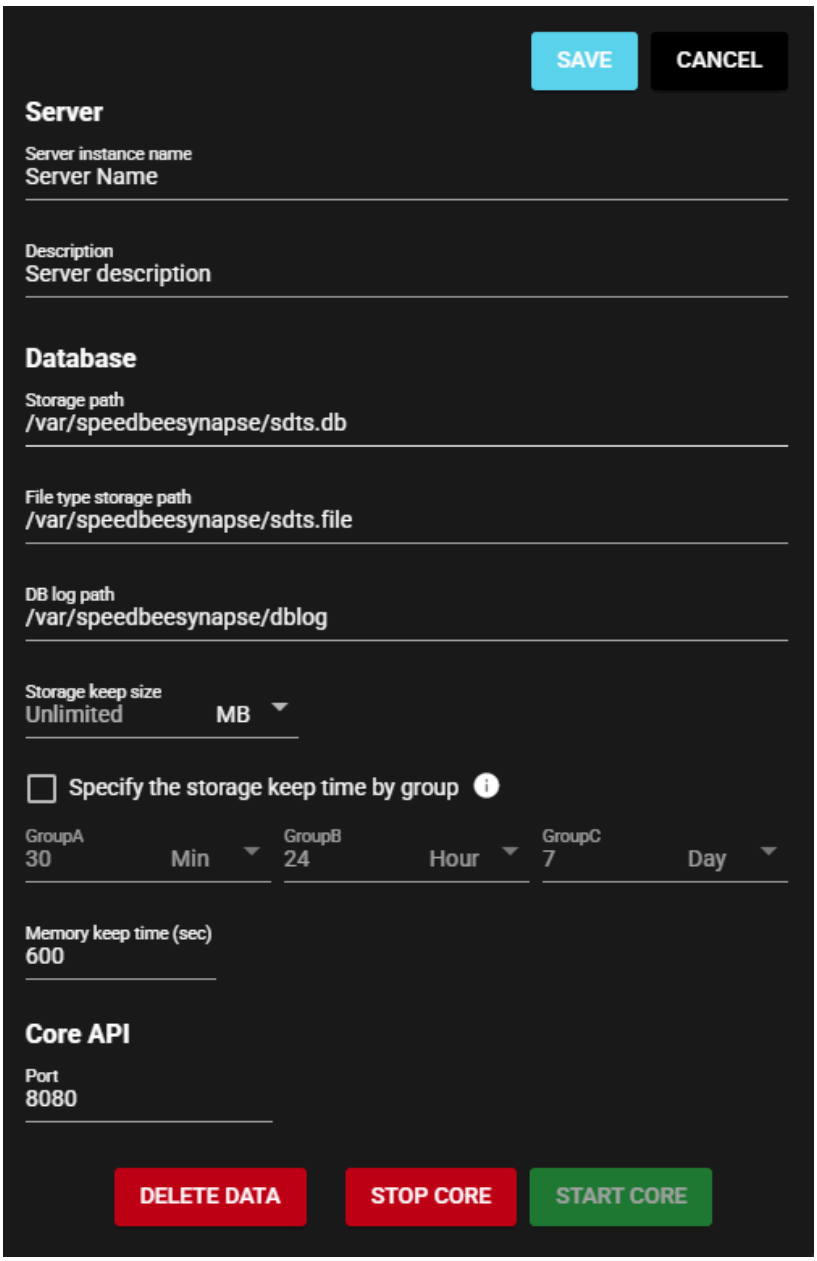


2. 「 서버 설정 」을 선택합니다.



3. 다음 항목 중, 임의의 것을 입력하고 「 저장 」을 누르면, 설정이 갱신됩니다.

번역본
For Reference Only-



항목		설명
서버	서버 인스턴스명	SpeeDBee Synapse에 부여하는 명칭. Run Mode의 전체 패널에 표시된다. (복수 대의 SpeeDBee Synapse를 이용하고 있는 경우, 각각에 명칭을 설정함으로써, 식별에 이용 가능)
	설명	SpeeDBee Synapse의 용도 등의 설명
데이터베이스	스토리지 경로	데이터베이스의 저장 위치 디렉터리 ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.
	파일형 스토리지	파일형 데이터베이스의 저장 위치 디렉터리 ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.

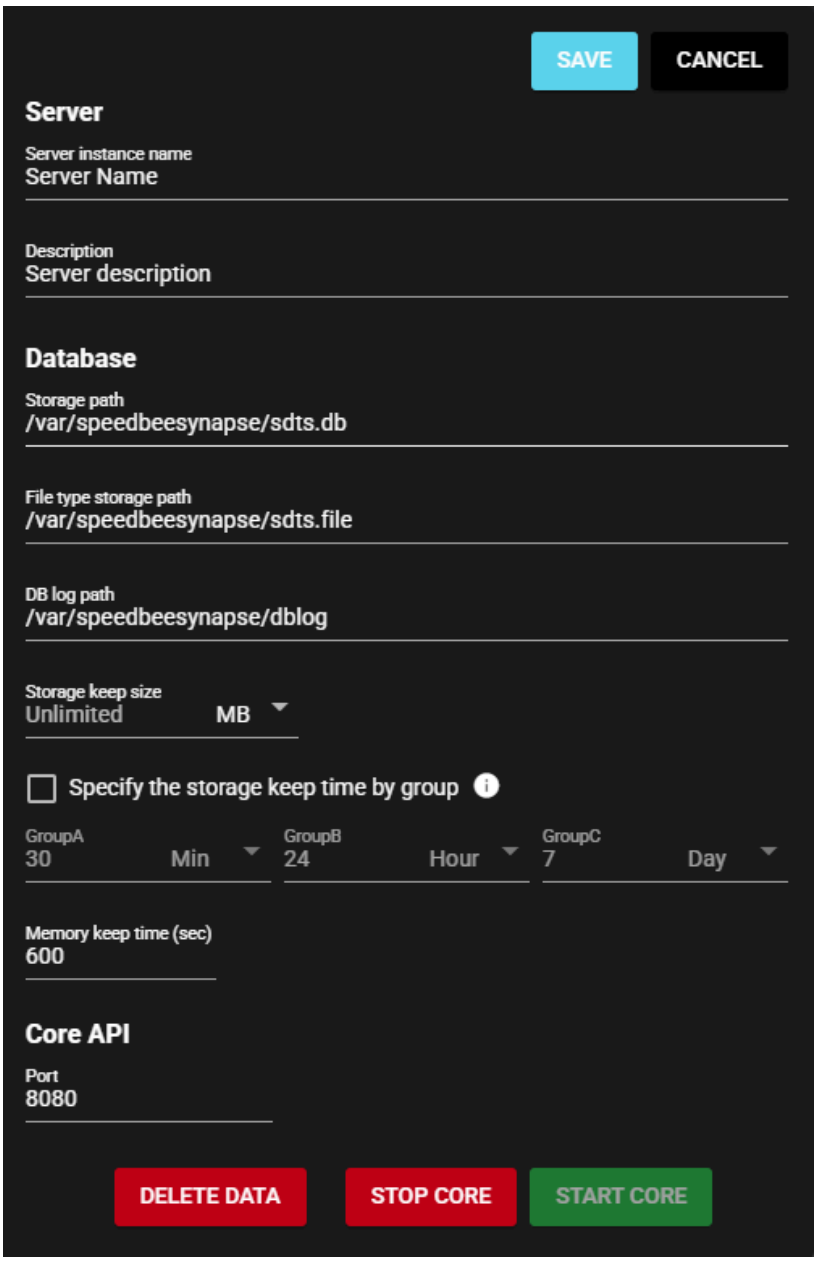
AI 번역본
-For reference only-

	경로	
	DB 로그 경로	데이터베이스의 로그의 저장 위치 디렉터리 ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.
	스토리지 상한 사이즈	스토리지에 데이터를 저장하는 상한 사이즈. 스토리지상의 데이터가 이 상한 사이즈를 초과하는 경우, 오래된 데이터부터 삭제된다. 미지정인 경우는 무제한.
	그룹 단위로 저장 기간을 지정	ON인 경우, 그룹별 스토리지에 데이터를 보유하는 기간을 설정 가능. 저장 기간을 지난 데이터는 삭제된다. 단, 저장 기간 내라도, 스토리지 상한 사이즈를 초과하는 경우는 오래된 데이터부터 삭제된다.
	메모리 저장 기간(초)	메모리상에 데이터를 보유하는 시간.
코어 API	포트	SpeeDBee Synapse가, WEB 화면과 내부 서버 사이에서 주고받기 위해 사용되는 포트 번호. ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다.

데이터 삭제

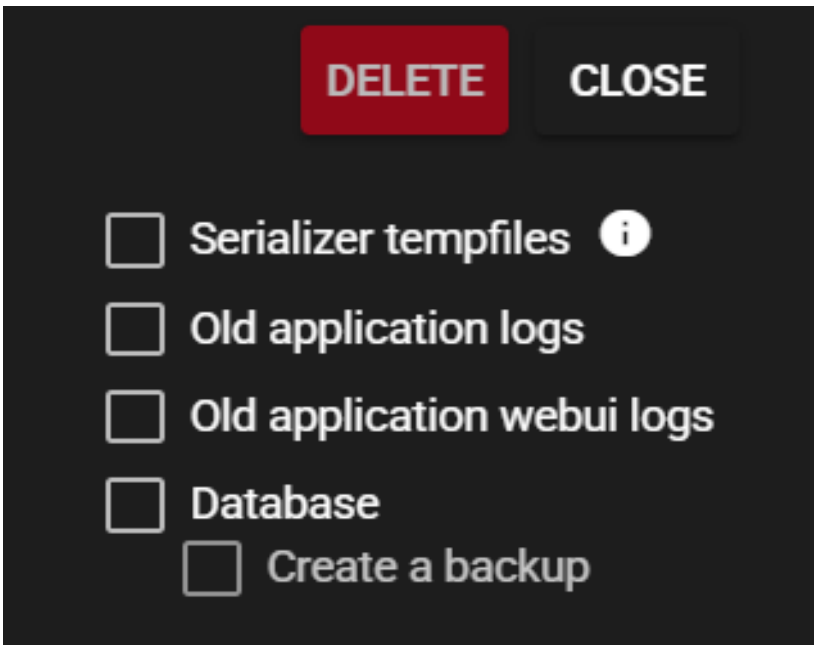
1. 「 데이터 삭제 」를 누릅니다.

AI 번역본
-For Reference Only-



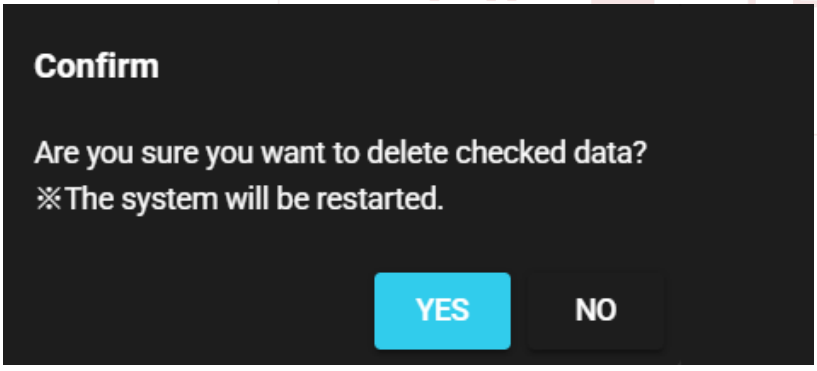
2. 삭제할 데이터를 선택하고, 「 삭제 」를 누릅니다.

AI 번역본
-For Reference Only-



항목	설명
시리얼라이저의 임시 파일	시리얼라이저 컴포넌트가 출력한 임시 파일 ※삭제하는 경우는 시스템이 재기동됩니다.
애플리케이션 로그	코어, 컴포넌트, 데이터베이스의 운용 로그
애플리케이션 로그(UI)	WEBUI의 운용 로그
데이터베이스	SpeeDBee Synapse의 시계열 데이터베이스 ※삭제하는 경우는 시스템이 재기동됩니다.
백업을 작성한다	데이터베이스를 삭제하기 전에 백업을 작성한다 ※데이터베이스에 체크를 넣은 경우에만 선택할 수 있습니다.

3. 「예」를 누릅니다.

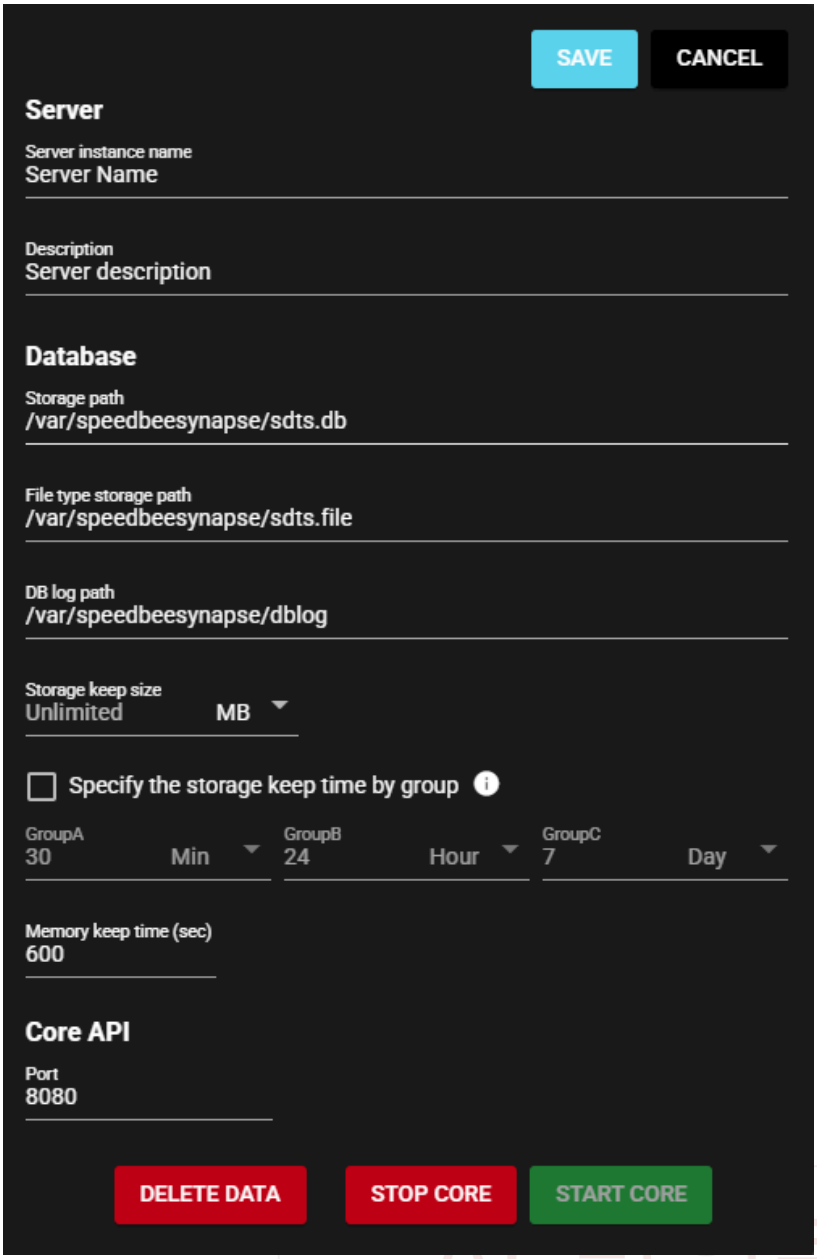


4. 선택한 데이터가 삭제됩니다.

AI 번역본
Only-

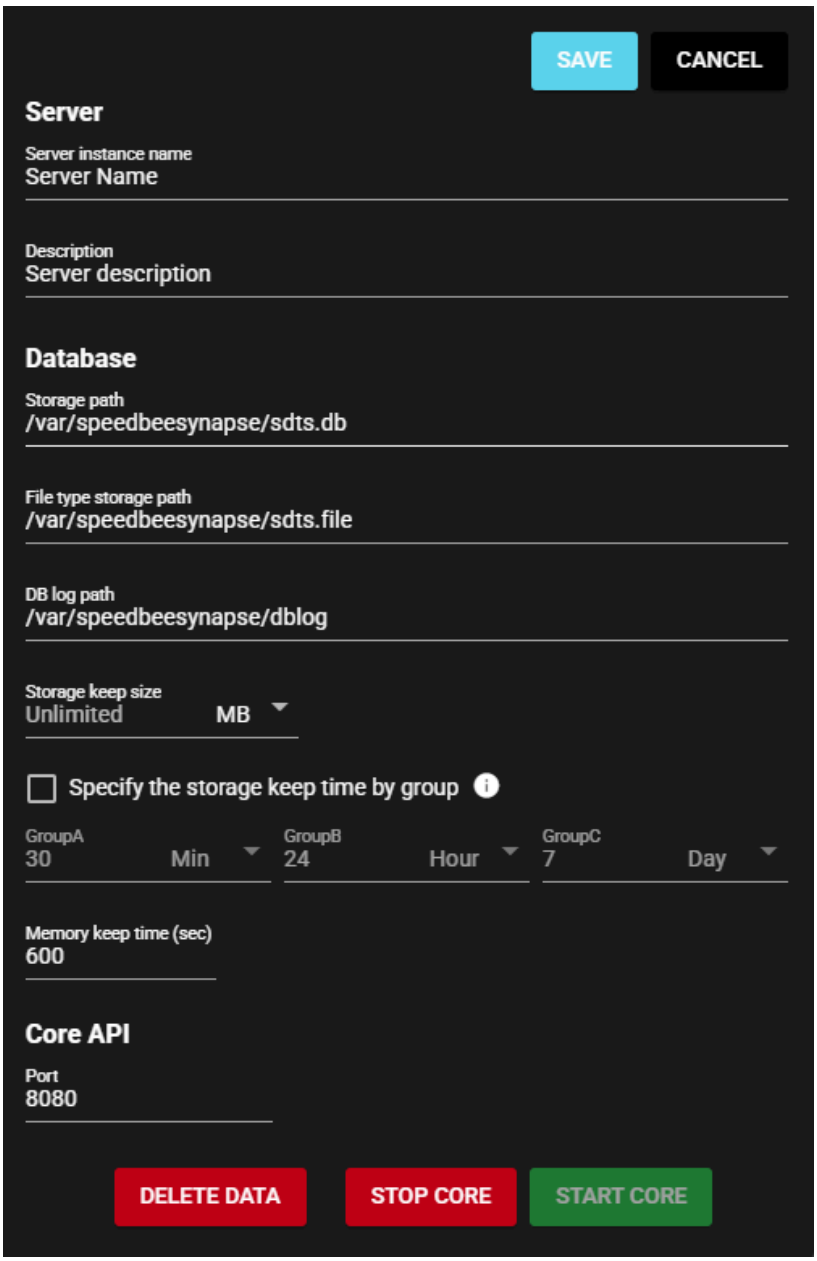
코어 기동・정지

1. 「 코어 정지 」를 누르면, SpeeDBee Synapse의 코어 기능을 정지합니다.



2. 「 코어 기동 」을 누르면, SpeeDBee Synapse의 코어 기능을 기동합니다.

AI 번역본
-For Reference Only-



5.3.3.1.4 전반적인 설정

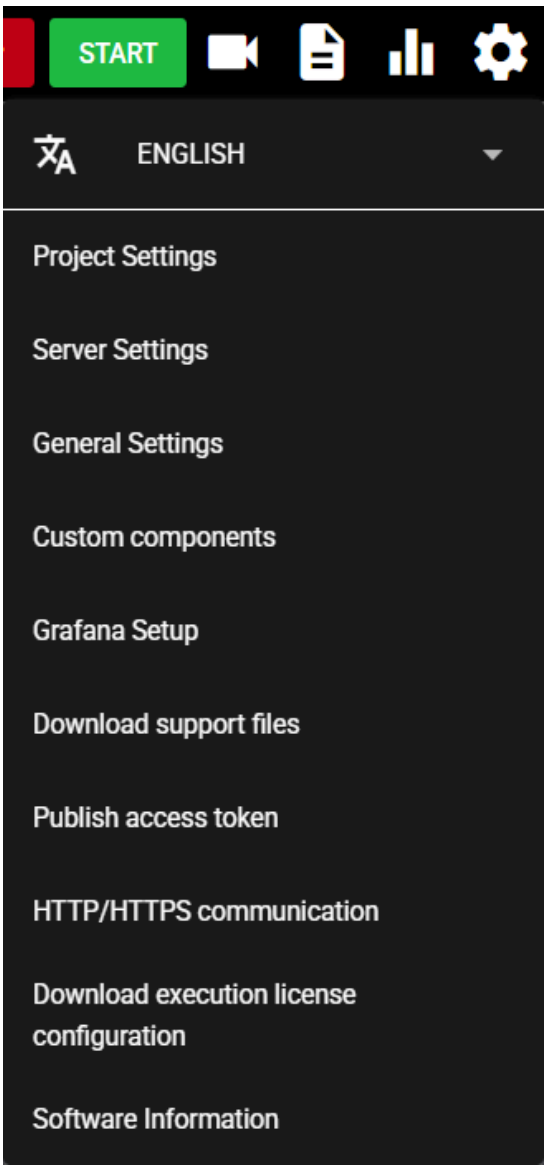
다음 순서로, 전반적인 설정의 확인 및 갱신을 수행할 수 있습니다.

AI 번역본
For Reference Only-

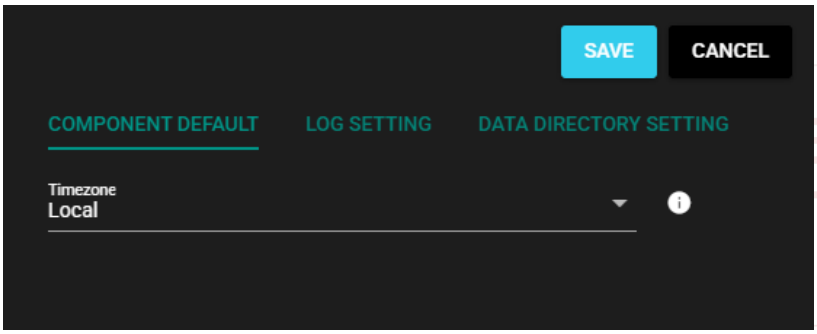
1. 설정 메뉴 아이콘을 누릅니다.



2. 「 전반적인 설정 」을 선택합니다.



3. 탭을 전환하여 임의의 항목을 편집하고,「 저장 」을 누르면 설정이 갱신됩니다.



항목	설명
컴포넌트 공통	컴포넌트의 공통 초기값을 설정합니다.
로그 설정	애플리케이션 로그의 설정을 수행합니다.

데이터 디렉터리 설정	데이터 디렉터리의 설정을 수행합니다.

5.3.3.1.5 컴포넌트 공통

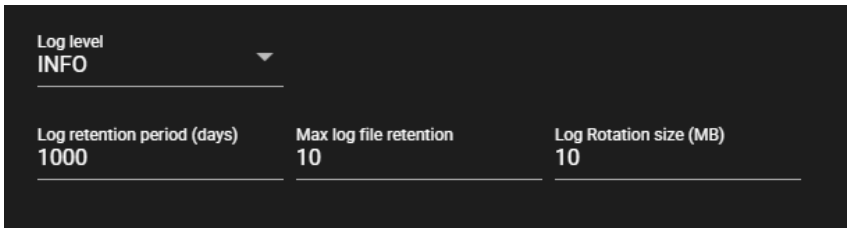
컴포넌트의 공통되는 입력 항목의 초기값을 확인, 편집합니다.



항목	설명
타임존 초기값	CSV, JSON 시리얼라이저, File 에미터에서 설정할 수 있는 타임존의 초기값

5.3.3.1.6 로그 설정

애플리케이션 로그의 설정을 확인, 편집합니다.



항목	설명
로그 레벨	로그 출력의 레벨
로그 저장 기간 (일)	로그의 저장 기간 일수
로그 파일 최대 저장 수	로그 파일의 최대 저장 수
로테이션 사이즈 (MB)	로그 로테이션의 사이즈 (MB)

5.3.3.1.7 데이터 디렉터리 설정

데이터 디렉터리의 설정을 확인, 편집합니다.



항목	설명
데이터 디렉터리	VIDEO 컬렉터, File 에미터, Event Recorder에서 설정할 수 있는 저장처, 출력처의 베이스 디렉터리. ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다.

※데이터 디렉터리를 변경한 경우, 이하의 컴포넌트에서 다음 기동 시부터 변경이 반영되어, 새로운 디렉터리가 사용됩니다.

- VIDEO 컬렉터
- File 에미터
- Event Recorder

5.3.3.1.8 커스텀(Python)

Python 커스텀 컴포넌트의 설정을 수행할 수 있습니다. 상세는「Python 커스텀 컴포넌트 작성」을 참조하십시오.

5.3.3.1.9 커스텀(C)

C 커스텀 컴포넌트의 설정을 수행할 수 있습니다. 상세는「C 언어 커스텀 컴포넌트 작성」을 참조하십시오.

5.3.3.1.10 Grafana 셋업

Grafana의 설정을 수행할 수 있습니다. 상세는「Grafana와의 연계」를 참조하십시오.

5.3.3.1.11 지원용 파일 다운로드

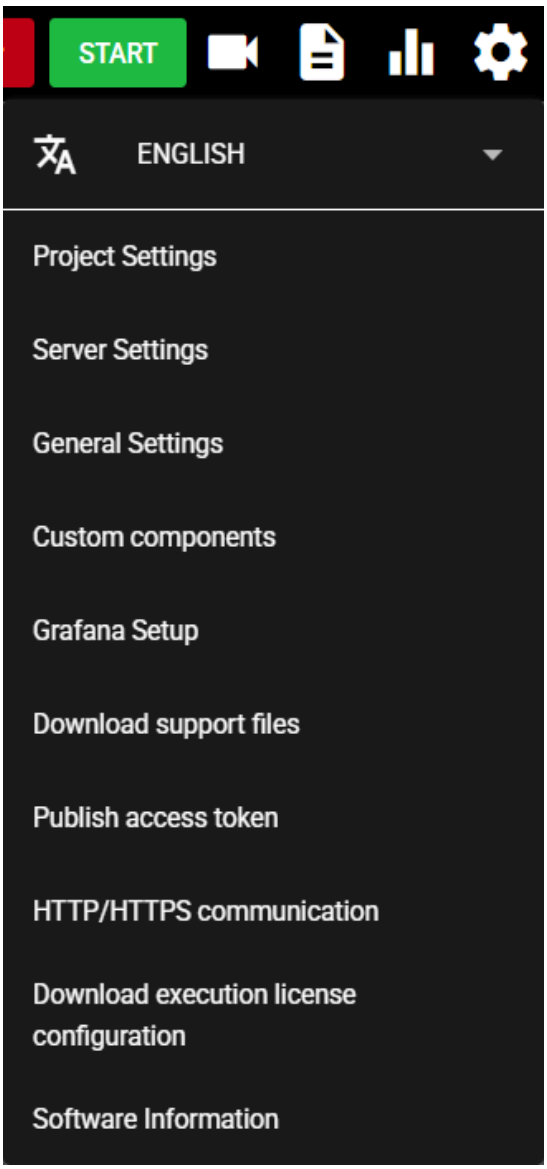
다음 순서로 SpeeDBee Synapse의 지원 담당자에게 전달할 파일을 다운로드할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

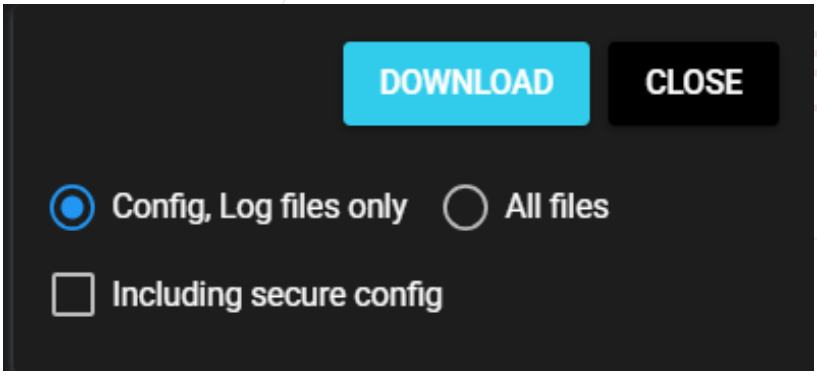


2.「지원용 파일 다운로드」를 선택합니다.

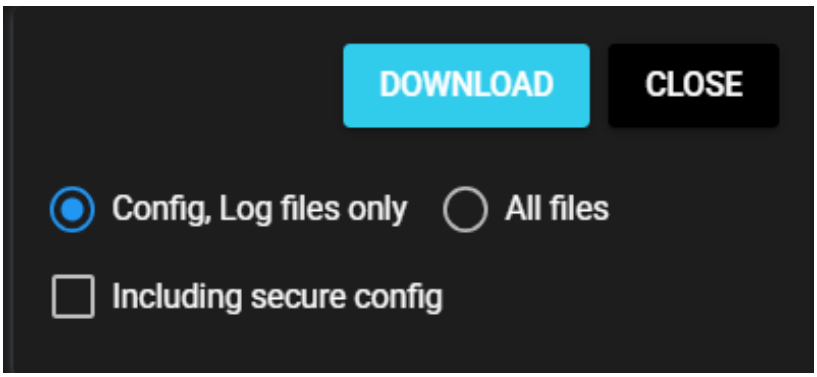
번역본
For Reference Only-



3. 다운로드할 파일을 선택합니다.
※기본적으로는「 설정, 로그 파일만 」을 선택하십시오. 기밀 정보는 지원 담당자로부터 지시가 없는 한 포함할 필요는 없습니다.



4.「 다운로드 」를 누릅니다.



5. 지원용 파일이 다운로드됩니다.

5.3.3.1.12 액세스 토큰 발행

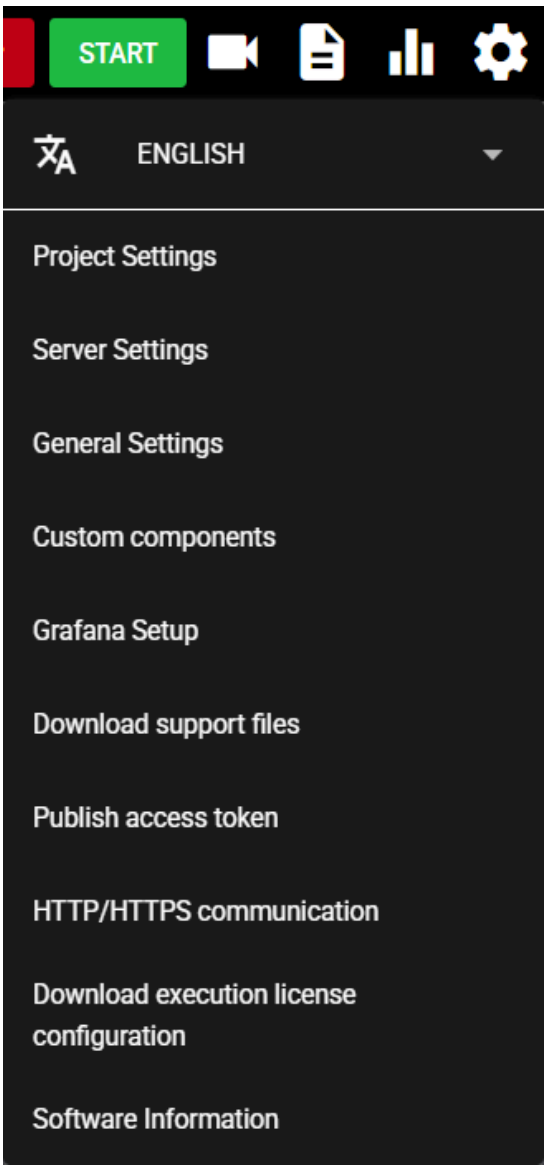
다음 순서로 액세스 토큰을 발행할 수 있습니다. 발행한 액세스 토큰은 「 DB 쿼리 서비스 」, 「 Grafana와의 연계 」, 「 제어 프로그램과의 연계 」에서 사용할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

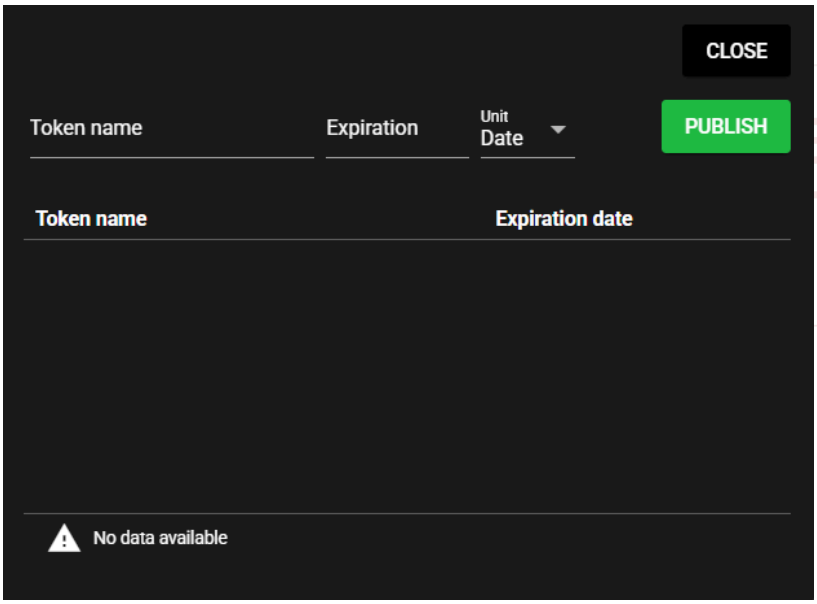


2. 「 액세스 토큰 발행 」을 선택합니다.

AI 번역본
-For Reference Only-



3. 다음 항목을 입력하고「 발행 」을 누릅니다.



항목	설명
토큰명	토큰의 식별명
유효 기한	토큰의 유효 기한

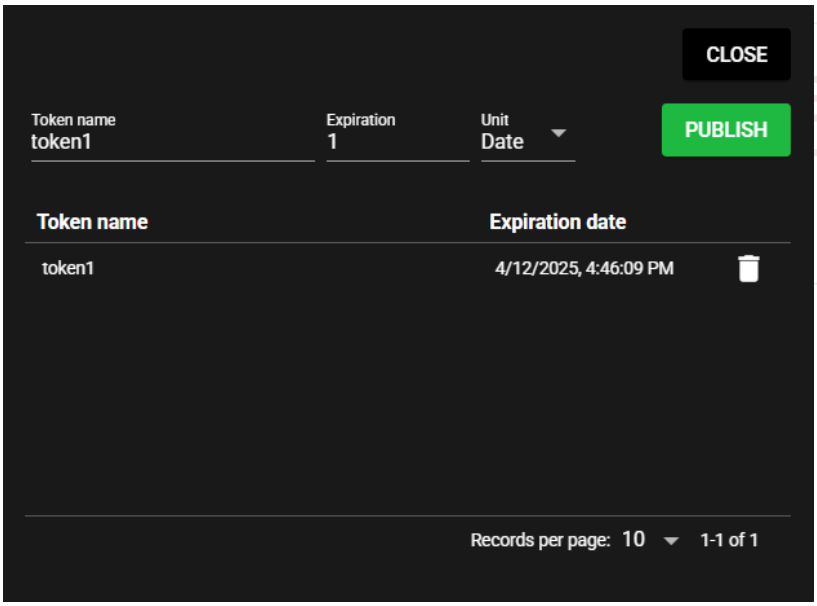
> 액세스 토큰의 유효 기한에 대해
>
> 액세스 토큰의 유효 기한을 길게 한 경우, 토큰이 누설되었을 때의 보안 리스크가 높아집니다. 반대로 유효 기한을 짧게 하면 보안은 강화되지만, 빈번하게 액세스 토큰을 갱신할 필요가 있습니다. 미리 정한 운용 방침에 따라 정기적으로 액세스 토큰을 갱신하는 것도 검토하십시오.

4. 발행된 액세스 토큰을 복사하고 「 닫기 」를 누릅니다.

> 액세스 토큰의 보관에 대해
>
> 액세스 토큰은 한 번밖에 표시되지 않습니다. 이 화면을 닫으면 같은 토큰을 취득할 수 없으므로, 반드시 복사하여 보관하십시오.



5. 발행한 액세스 토큰은 목록에 추가됩니다.



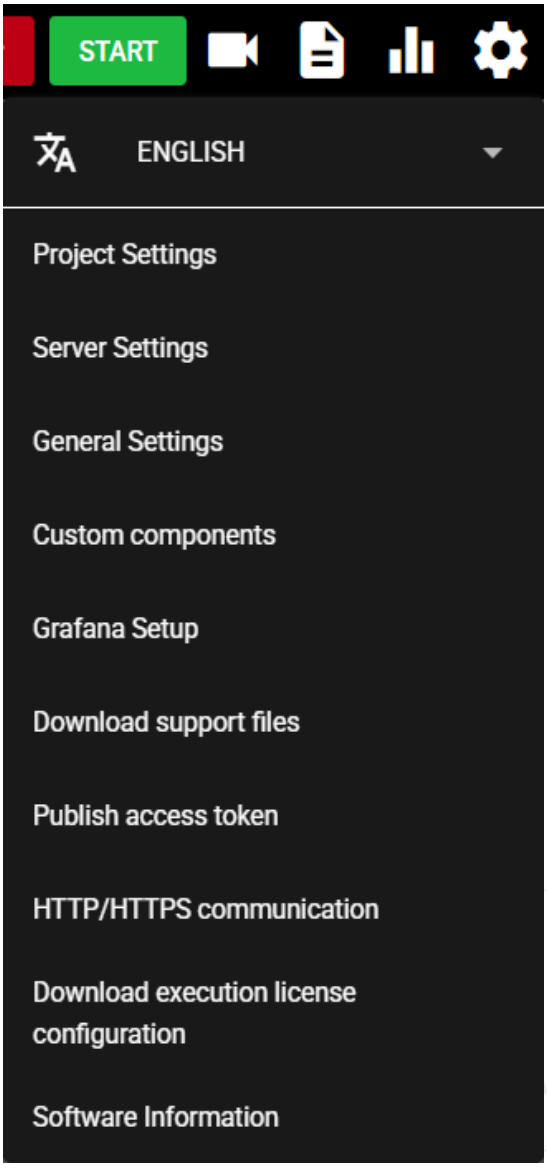
5.3.3.1.13 HTTP/HTTPS 통신

다음 순서로 사용자의 디바이스와 SpeeDBee Synapse의 HTTP 통신에 대한 설정을 수행합니다.

1. 설정 메뉴 아이콘을 누릅니다.

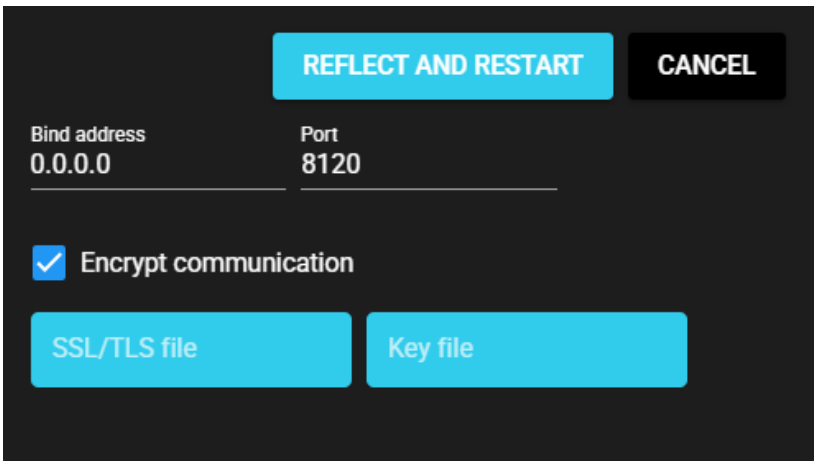


2. 「 HTTP/HTTPS 통신 」을 선택합니다.



번역본
ference Only-

3. 이하의 항목을 입력하고 「 반영하여 재기동 」을 누릅니다.



항목	설명
바인드 주소	SpeeDBee Synapse가 요청을 접수하는 주소 ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.
포트	SpeeDBee Synapse가 요청을 접수하는 포트 ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.
암호화 통신	암호화 통신(HTTPS)을 이용하는 경우 ON ※일부 환경에서는 이 설정을 변경/편집할 수 없습니다.
SSL/TLS 인증서	암호화를 위해 사용하는 디지털 인증서 ※인증서는 CA（인증 기관）로부터 취득할 필요가 있습니다.
비밀키	SSL/TLS 인증서에 연결된 비밀키 ※비밀키는 CA（인증 기관）로부터 취득할 필요가 있습니다.

4. 설정 반영을 위해 시스템 전체의 재기동이 수행됩니다. 재기동 후, 설정 내용에 맞춘 URL로 SpeeDBee Synapse에 액세스할 수 있습니다.

AI 번역본

암호화 통신을 수행하는 경우

```
https://<호스트명or디바이스의 IP 주소>:<SpeeDBee Synapse가 요청을 접수하는 포트>/
```

암호화 통신을 수행하지 않는 경우

```
http://<호스트명or디바이스의 IP 주소>:<SpeeDBee Synapse가 요청을 접수하는 포트>/
```

5.3.3.1.14 프록시 설정

여기에서는 프록시 설정의 조작에 대해 설명합니다. 여기에서 설정한 프록시 정보는 각 컴포넌트의 설정에서 참조함으로써, 해당 컴포넌트의 통신을 프록시 경유로 전환할 수 있습니다.

프록시 대응 컴포넌트 목록

- MQTT 컬렉터
- MQTT 에미터
- Cloud 에미터(AWS용)
- Cloud 에미터(Azure용)

또한, 기본 프록시를 설정함으로써 특정 조작에서의 통신이 프록시 경유로 수행되게 됩니다.

대상이 되는 조작 목록

- Python 커스텀 컴포넌트의 외부 라이브러리 추가

추가

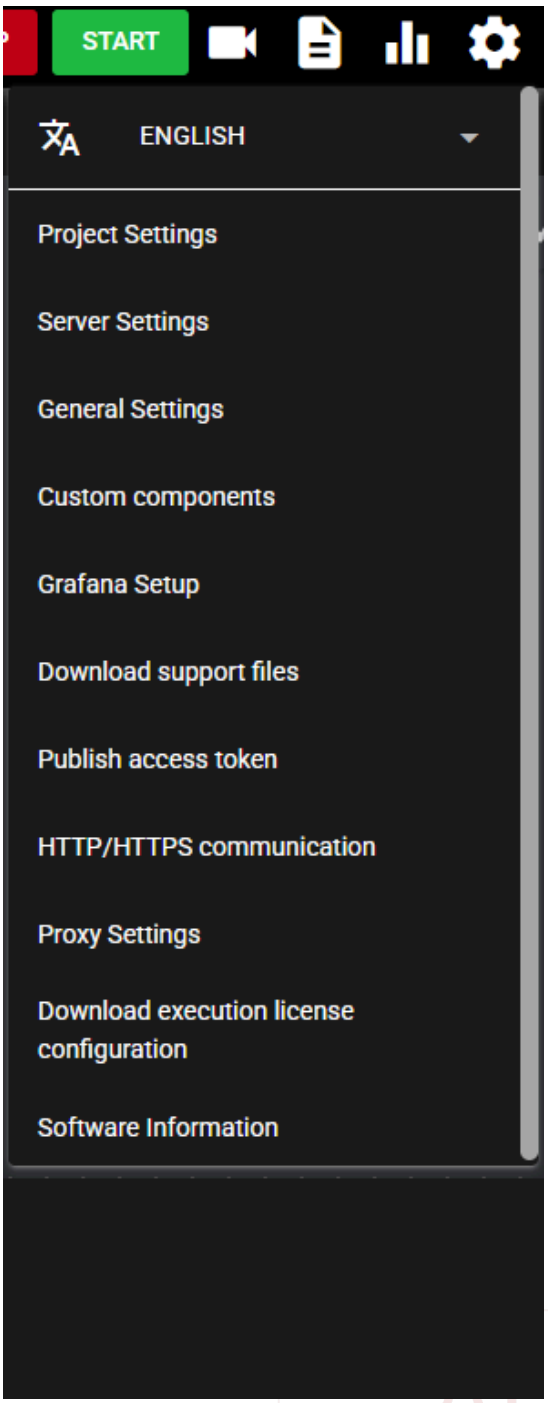
다음 순서로 프록시 설정의 추가를 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.



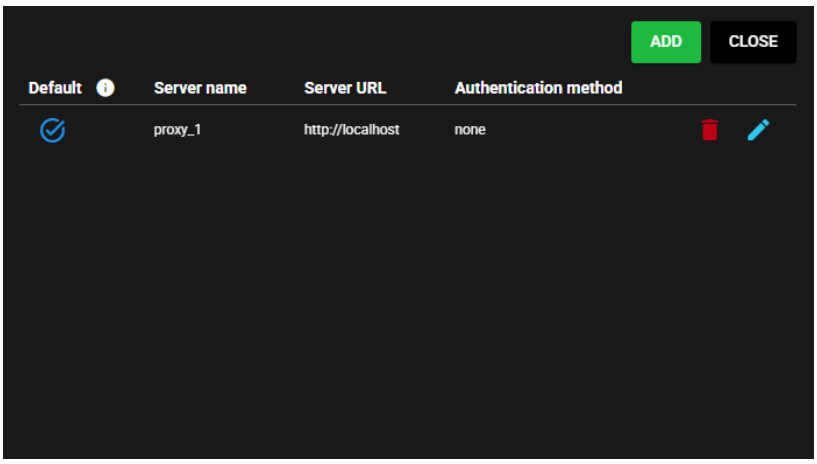
2. 「 프록시 설정 」을 선택합니다.

AI 번역본
-For Reference Only-

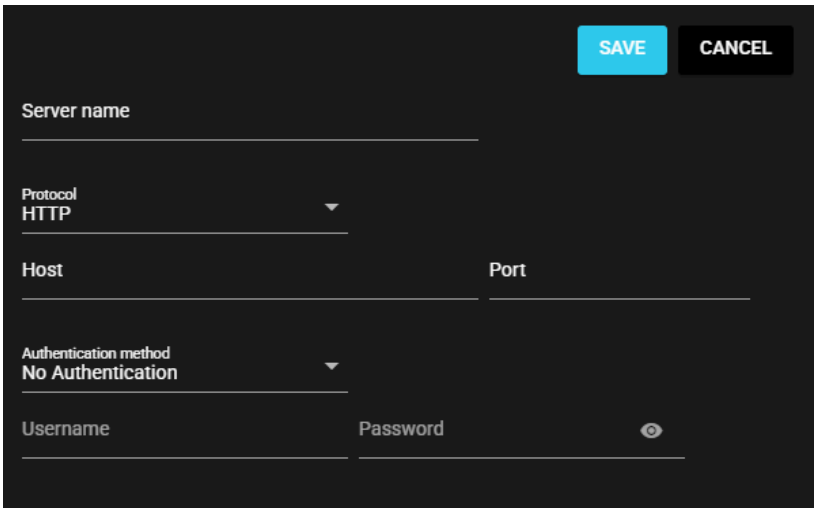


3.「추가」를 누릅니다.

AI 번역본
-For Reference Only-



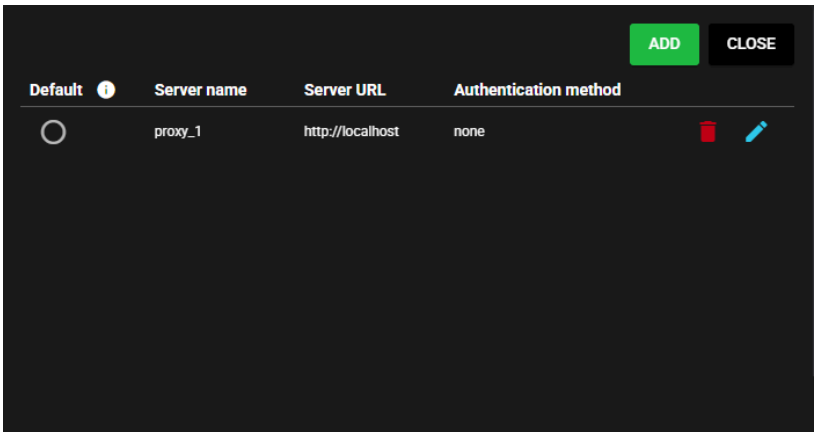
4. 이하의 항목을 입력하고 「 저장 」을 누릅니다.



항목	설명
서버명	임의의 서버 식별명. ※프록시 경유의 통신을 설정하고 있는 컴포넌트는 서버명을 참조하여 설정을 읽어 들이고 있으므로, 변경하면 컴포넌트의 재설정이 필요하게 됩니다.
프로토콜	사용할 프록시의 종류. (HTTP)
호스트	프록시 서버의 호스트명 또는 IP 주소.
포트	프록시 서버의 포트 번호.
인증 방식	프록시 서버의 인증 방식. (인증 없음, Basic 인증)
사용자명	Basic 인증을 사용하는 경우의 사용자명.
패스워드	Basic 인증을 사용하는 경우의 패스워드.

AI 번역본
-For Reference Only-

5. 저장한 설정은 목록에 추가됩니다.



편집

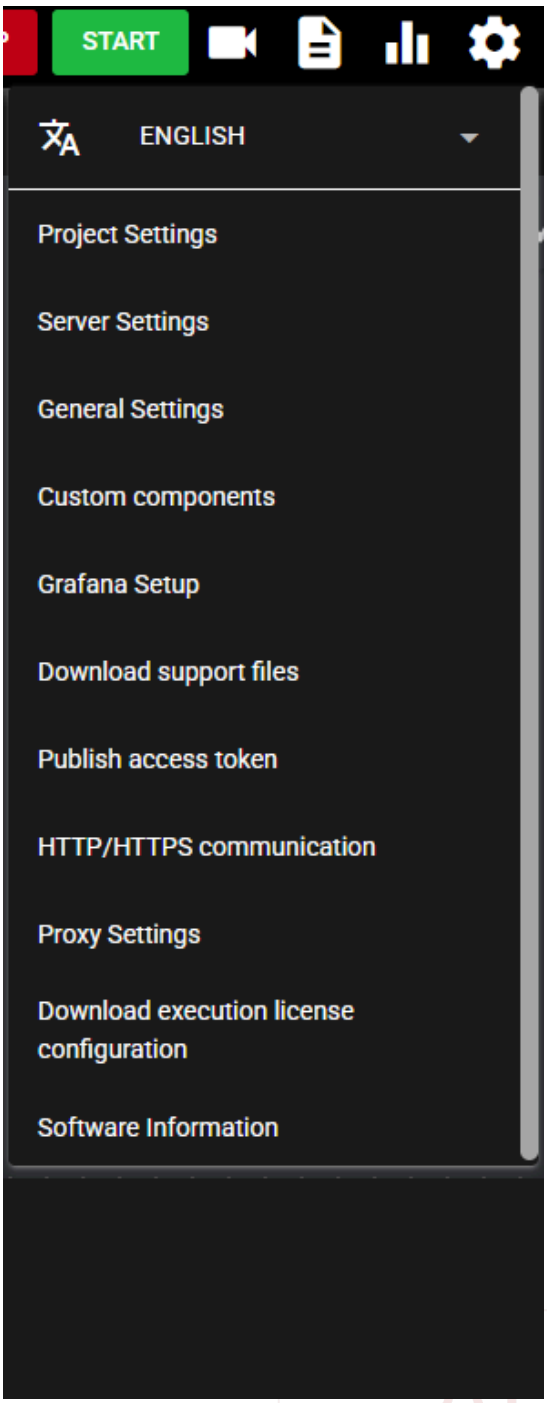
다음 순서로 프록시 설정의 편집을 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.



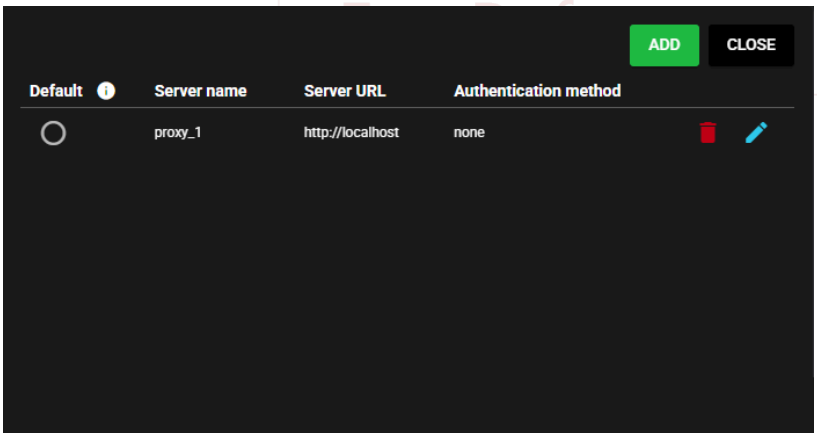
2.「 프록시 설정 」을 선택합니다.

AI 번역본
-For Reference Only-

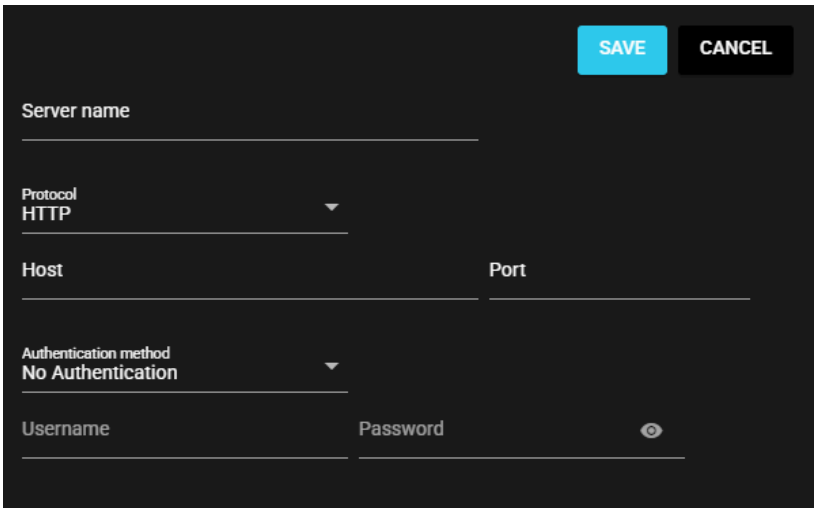


AI 번역본
Only-

3. 편집 아이콘을 누릅니다.



4. 이하의 항목을 입력하고 「 저장 」을 누릅니다.



항목	설명
서버명	임의의 서버 식별명. ※프록시 경유의 통신을 설정하고 있는 컴포넌트는 서버명을 참조하여 설정을 읽어 들이고 있으므로, 변경하면 컴포넌트의 재설정이 필요하게 됩니다.
프로토콜	사용할 프록시의 종류. (HTTP)
호스트	프록시 서버의 호스트명 또는 IP 주소.
포트	프록시 서버의 포트 번호.
인증 방식	프록시 서버의 인증 방식. (인증 없음, Basic 인증)
사용자명	Basic 인증을 사용하는 경우의 사용자명.
패스워드	Basic 인증을 사용하는 경우의 패스워드.

-For Reference Only-

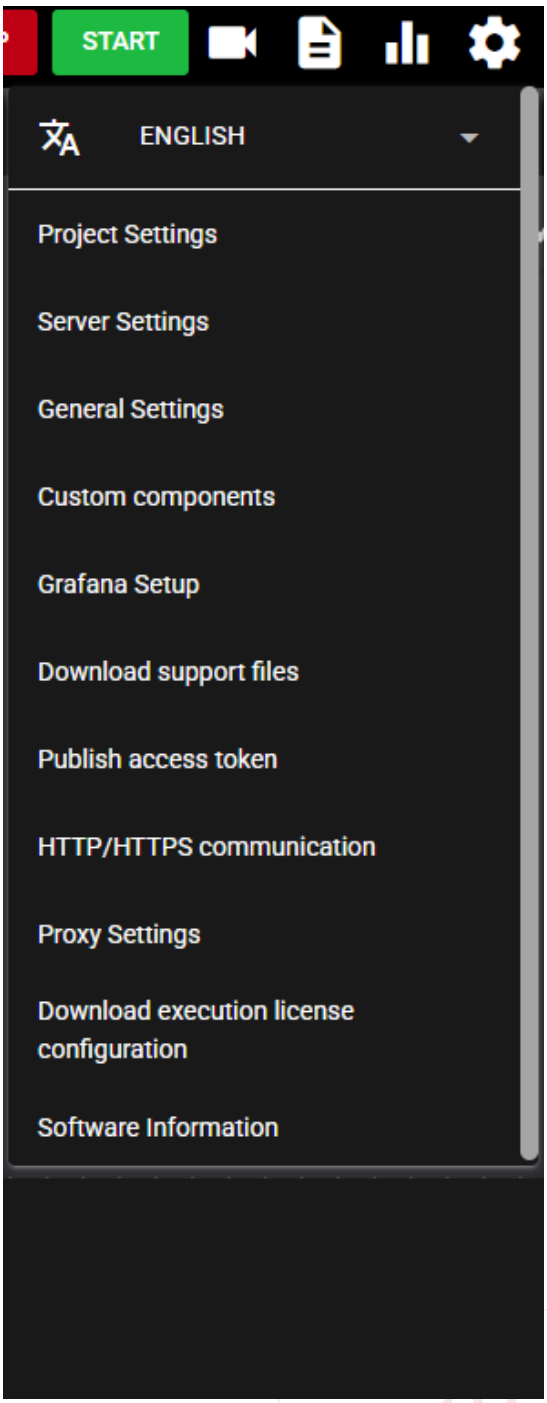
기본 프록시

다음 순서로 기본 프록시를 설정할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

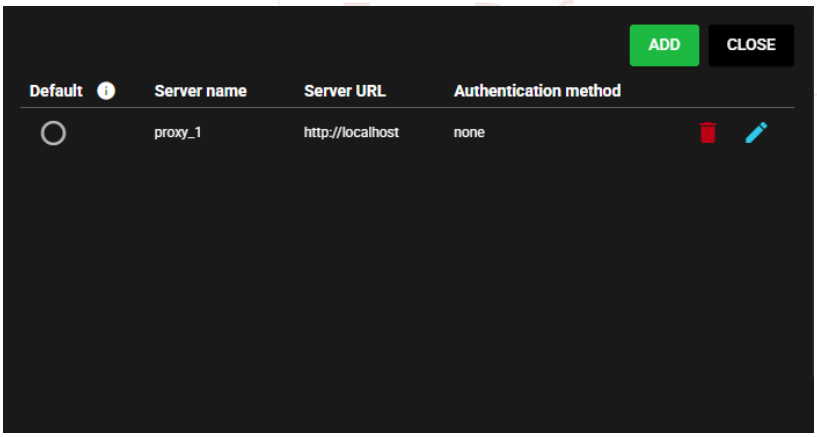


2. 「 프록시 설정 」을 선택합니다.

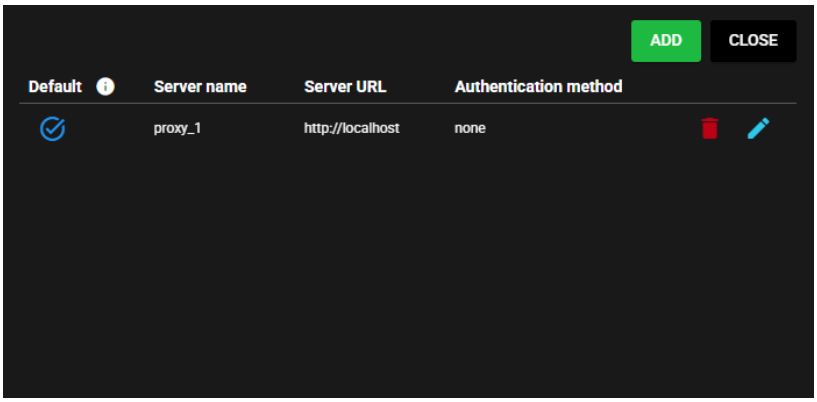


AI 번역본

3. 기본으로 설정하고 싶은 프록시에 체크를 넣습니다.



※기본으로 설정 완료된 프록시를 누르면, 기본 프록시를 미설정 상태로 할 수 있습니다.



삭제

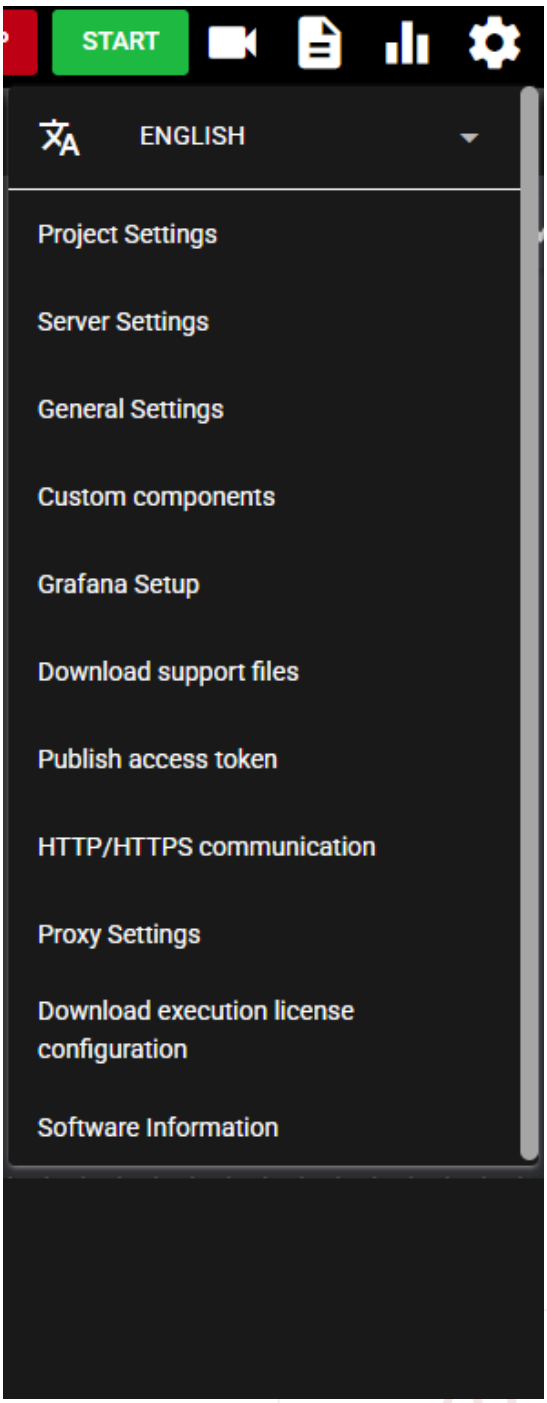
다음 순서로 프록시 설정의 삭제를 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.



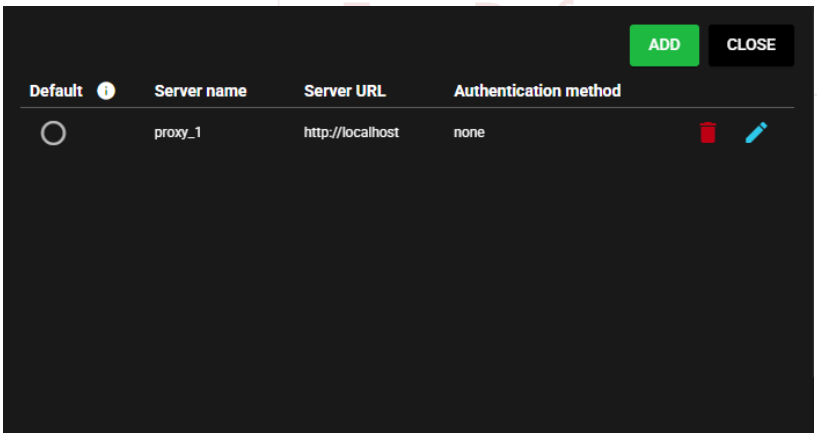
2. 「 프록시 설정 」을 선택합니다.

AI 번역본
-For Reference Only-



AI 번역본
Only-

3. 삭제 아이콘을 누릅니다.



※삭제한 프록시 설정이 컴포넌트로부터 참조되고 있던 경우, 컴포넌트의 재설정이 필요하게 됩니다.

5.3.3.1.15 실행 라이선스 설정 다운로드

실행 라이선스 환경용으로 설정을 다운로드할 수 있습니다. 상세는「 실행 라이선스 」를 참조하십시오.

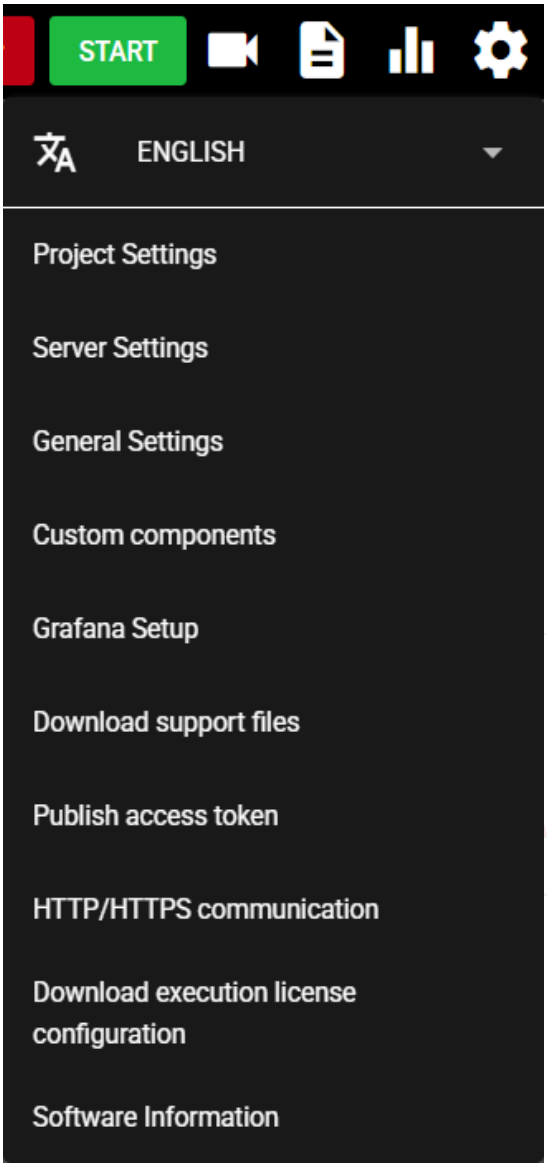
5.3.3.1.16 소프트웨어 정보

다음 순서로 소프트웨어 정보 및 라이선스 정보의 확인・라이선스의 등록을 수행할 수 있습니다.

1. 설정 메뉴 아이콘을 누릅니다.

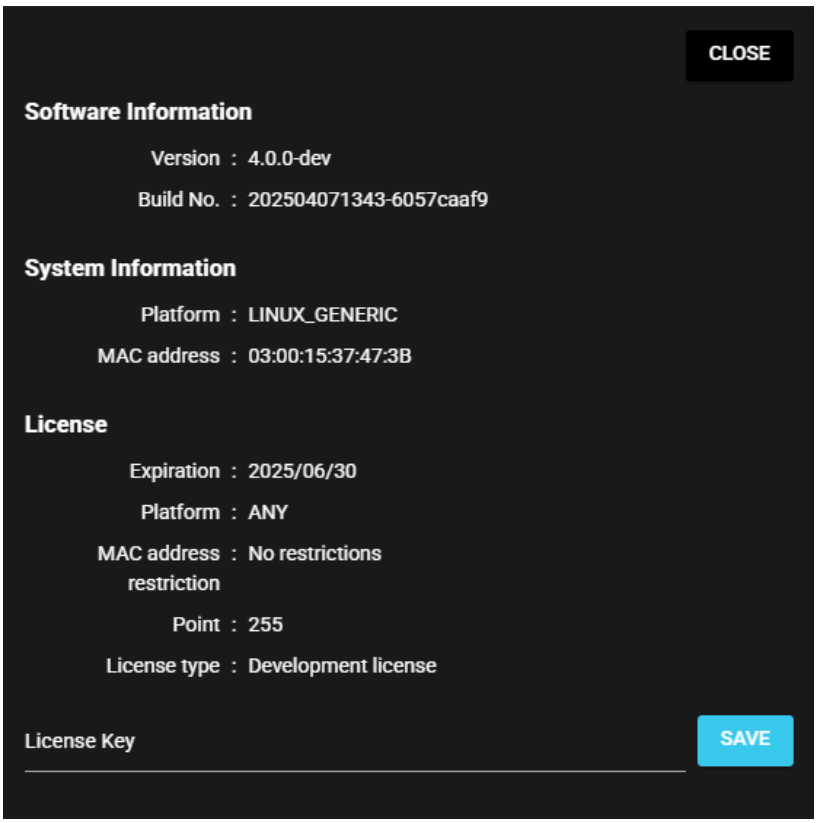


2.「 소프트웨어 정보 」를 선택합니다.



3. 다음 항목이 표시됩니다.

번역본
ference Only-

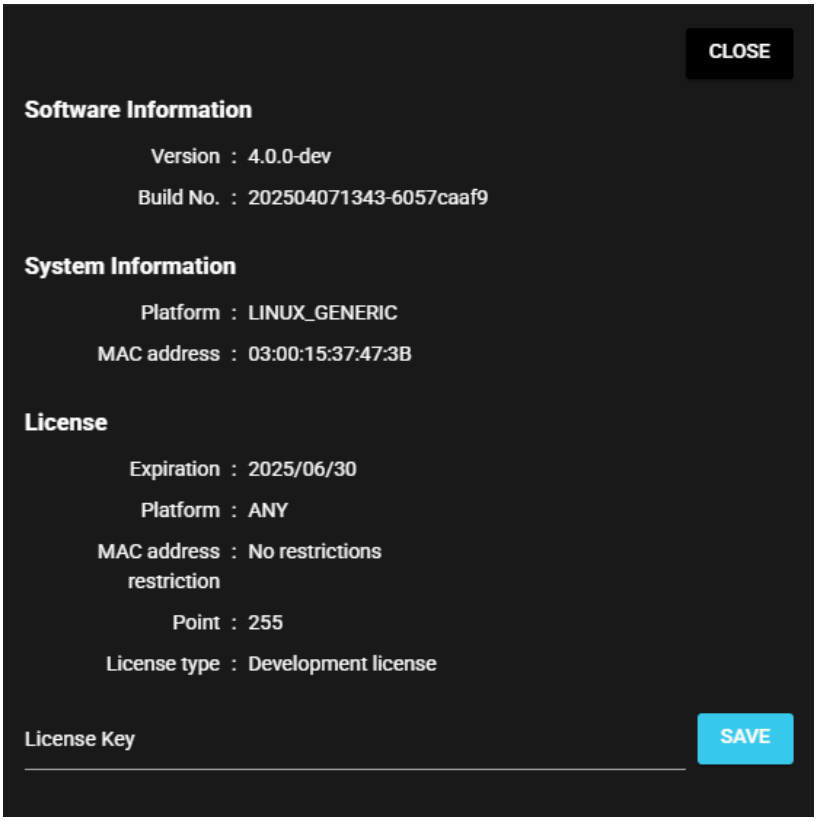


항목		설명
소프트웨어 정보	버전	SpeeDBee Synapse의 버전
	빌드 번호	SpeeDBee Synapse의 빌드 번호
시스템 정보	플랫폼	시스템의 플랫폼
	MAC 주소	시스템의 MAC 주소
라이선스	기한	등록된 라이선스의 유효 기한
	플랫폼	등록된 라이선스가 지원하는 플랫폼
	MAC 주소 제한	등록된 라이선스가 지원하는 MAC 주소
	포인트	등록된 라이선스에 부여되어 있는 포인트
	라이선스 종별	등록된 라이선스의 종별

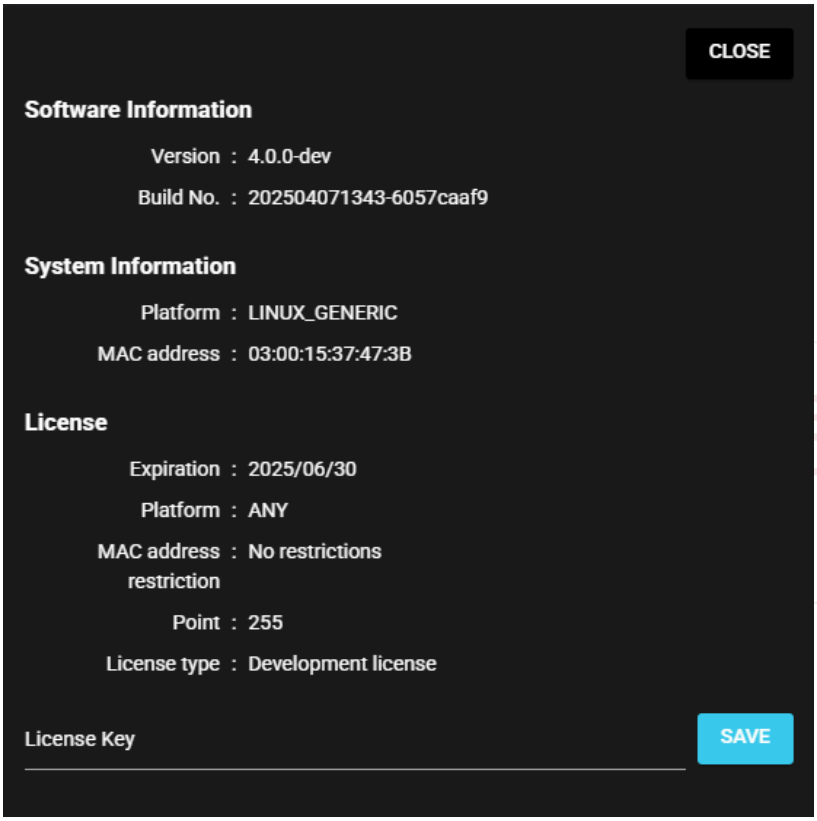
AI 번역본
-For Reference Only-

라이선스의 등록

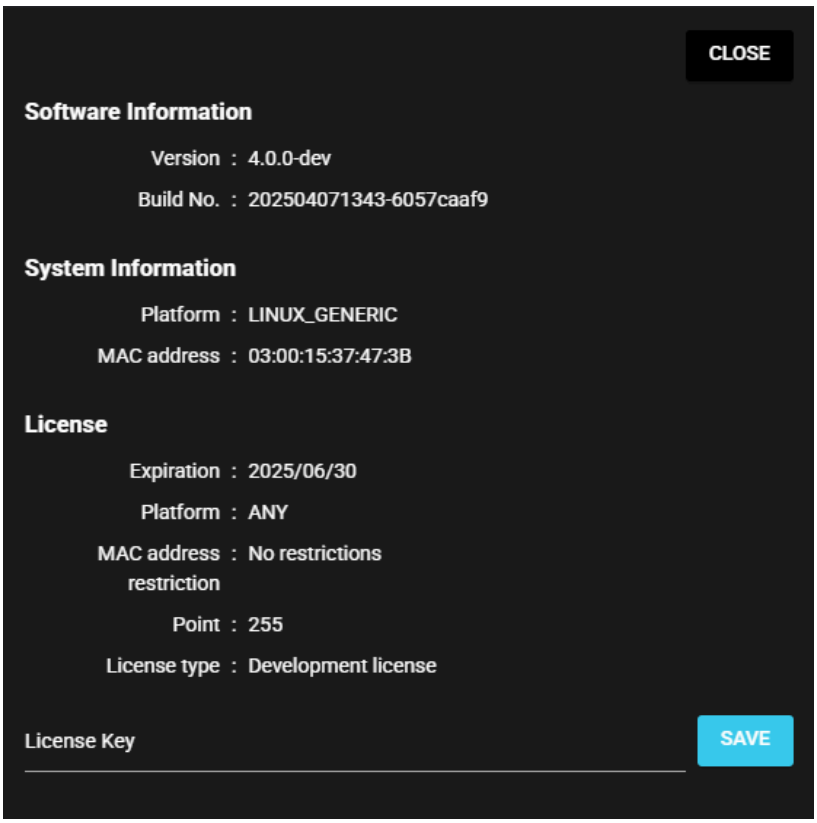
1. 라이선스 키 입력란에 라이선스 키를 입력합니다.



2. 「 저장 」을 누릅니다.



3. 라이선스가 등록됩니다. 등록한 라이선스의 정보가 화면에 반영됩니다.

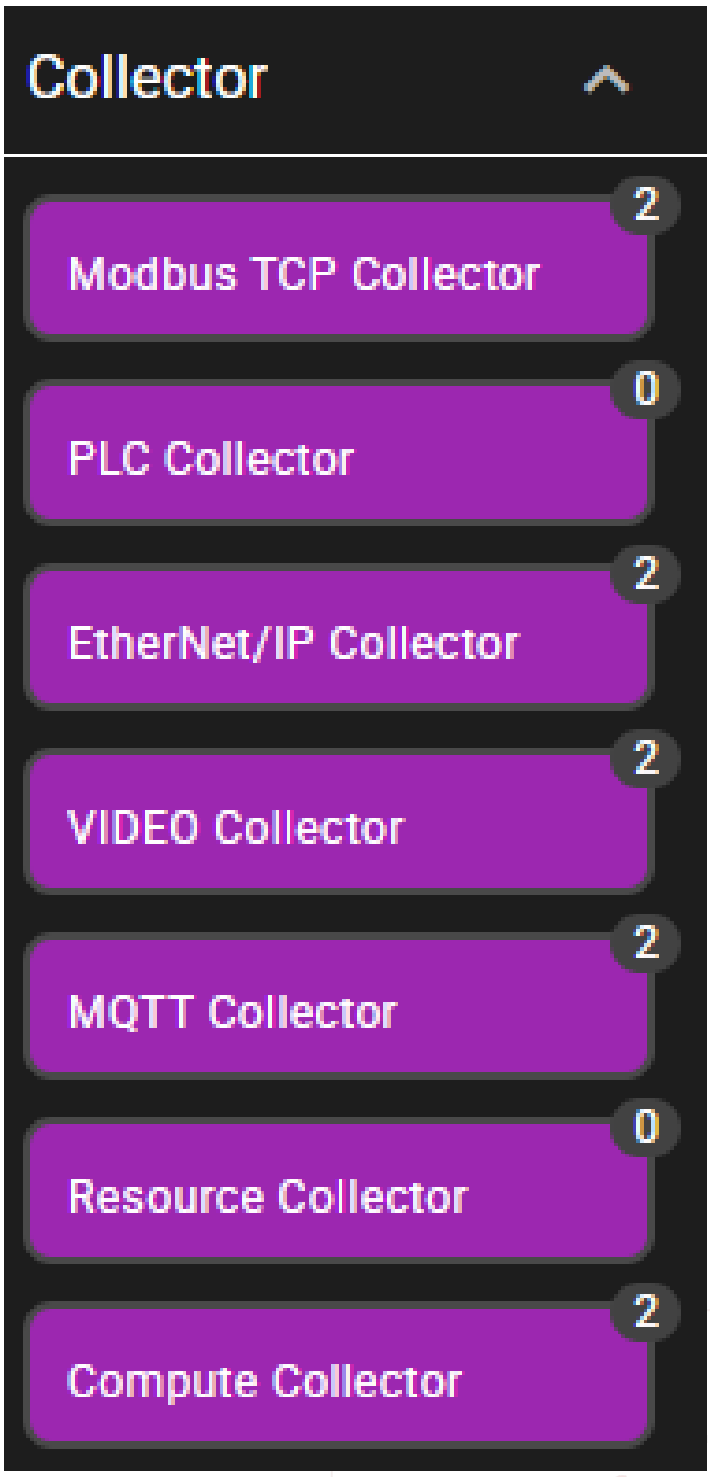


5.3.3.1.17 포인트에 대해

SpeeDBee Synapse의 라이선스에는 각 기능을 이용하기 위한 포인트가 부여되어 있습니다. SpeeDBee Synapse의 기능을 이용하면 그 기능별 포인트 수가 소비됩니다.

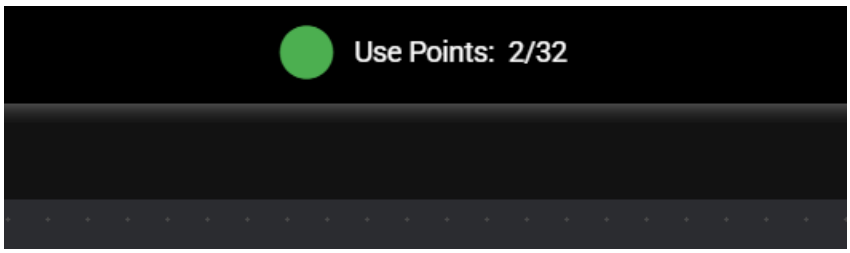
> Data Flow Controller판, OMRON-SBS for Windows판의 주의 사항
>
> Data Flow Controller판, OMRON-SBS for Windows판은 포인트를 무제한으로 이용할 수 있습니다.
> 그러나 무제한으로 동작을 보증하는 것은 아닙니다. 이용하시는 하드웨어 스펙, 동작시키는 애플리케이션, 관련되는 시스템 등의 영향을 받습니다.

AI 번역본
-For Reference Only-

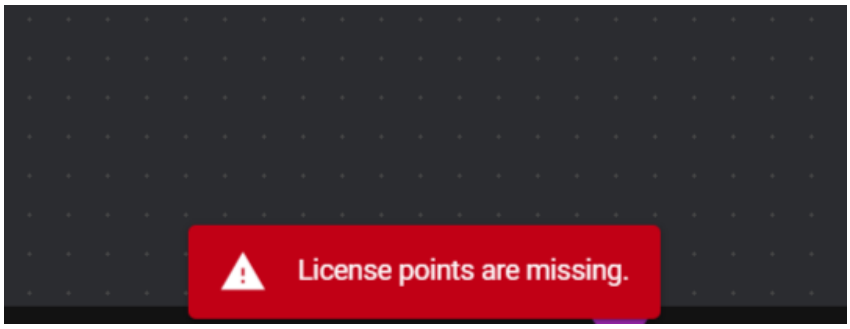


역본
-For Reference Only-

포인트가 0으로 되어 있는 것은 포인트를 소비하지 않습니다.
포인트가 2로 되어 있는 것은 설정을 저장하고 가동시키면 2P 소비합니다.
또한 같은 기능을 동시에 이용한 경우의 소비량도 2P 그대로입니다.
예를 들어, MQTT 에미터를 2 개 동작시켜도 소비되는 것은 2P가 됩니다.



동시에 이용하고 싶은 기능분의 포인트가 부족한 경우, 컴포넌트를 동작할 수 없습니다.



> 같은 기능을 동시에 사용하는 경우의 주의 사항
>
> 같은 기능을 동시에 이용하는 경우, 포인트 소비 수는 변하지 않지만, 무제한으로 동작을 보증하는 것은 아닙니다. 이용하시는 하드웨어 스펙, 동작시키는 애플리케이션, 관련되는 시스템 등의 영향을 받습니다.

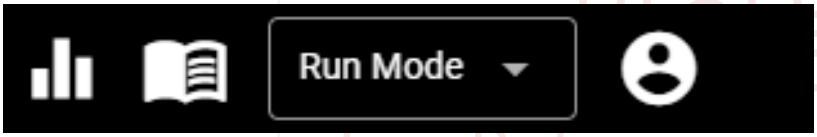
5.3.3.2 사용자 관리

여기에서는 사용자 관리의 조작에 대해 설명합니다.

5.3.3.2.1 추가

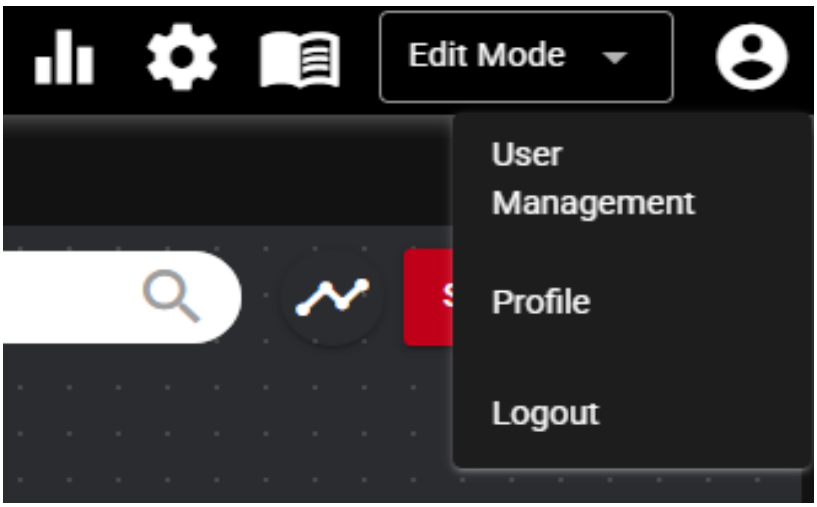
다음 순서로 사용자를 추가할 수 있습니다.

1. 사용자 관리 아이콘을 누릅니다.

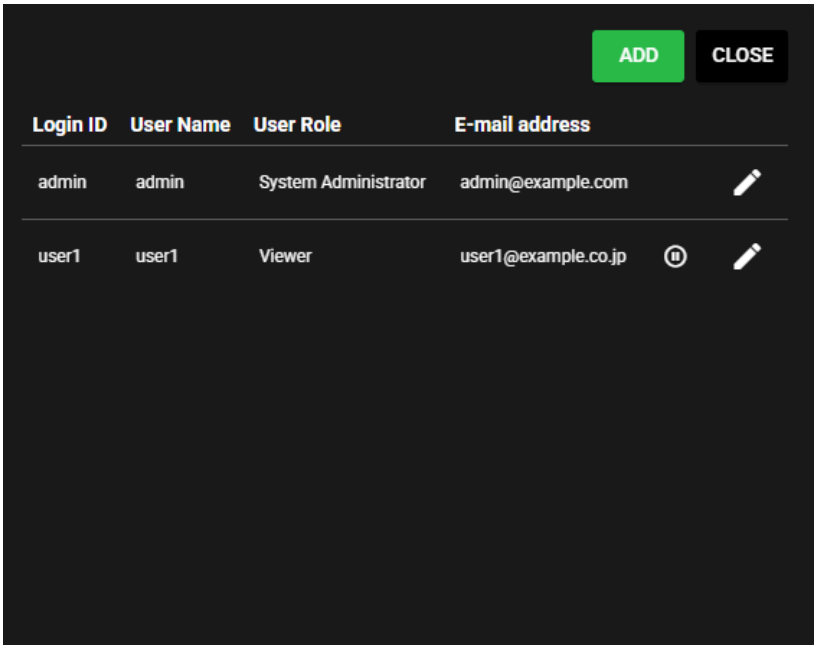


2. 「 사용자 관리 」를 선택합니다.

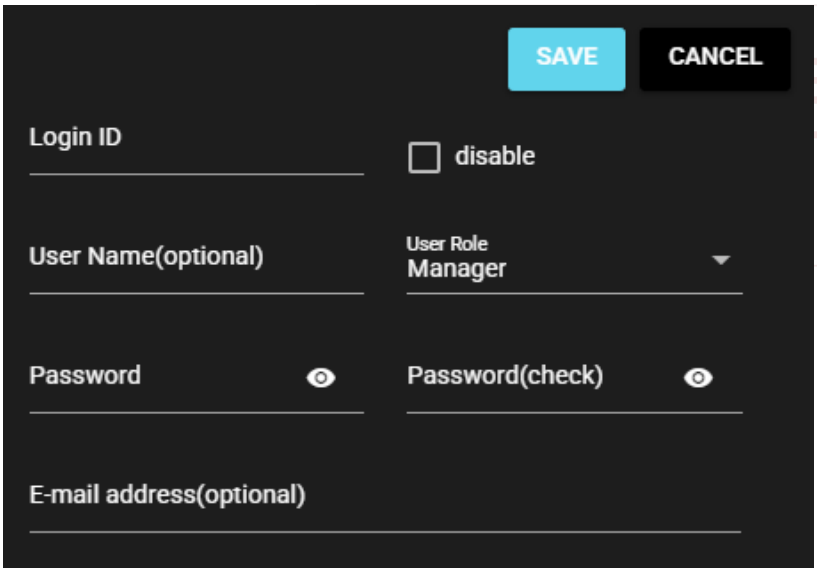
-For Reference Only-



3. 「 추가 」를 누릅니다.



4. 다음 항목을 입력하고 「 저장 」을 누릅니다.



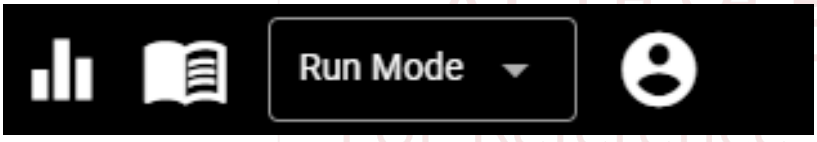
항목	설명
비활성화로 한다	체크하면 사용자가 비활성화가 되어 로그인을 할 수 없게 됩니다.
로그인 ID	로그인 시에 입력하는 ID
사용자명	사용자에게 부여하는 이름※생략 가능
권한	시스템 관리자：모든 조작이 가능합니다. 초기 사용자 "admin"에게 부여되어 있으며, 1명 이상 필수입니다. 관리자：모든 조작이 가능합니다. 시스템 관리자의 추가・갱신・삭제는 불가입니다. 오퍼레이터：Run Mode의 열람, 및 컴포넌트의 기동・정지가 가능합니다. 열람자：Run Mode의 열람이 가능합니다.
패스워드	로그인 시에 입력하는 패스워드
E메일 주소	사용자에 연결하는 E메일 주소※생략 가능
열람 가능 패널	사용자가 열람 가능한 패널※권한이 오퍼레이터・열람자인 경우에만 설정합니다.

5. 사용자가 추가됩니다.

5.3.3.2.2 편집

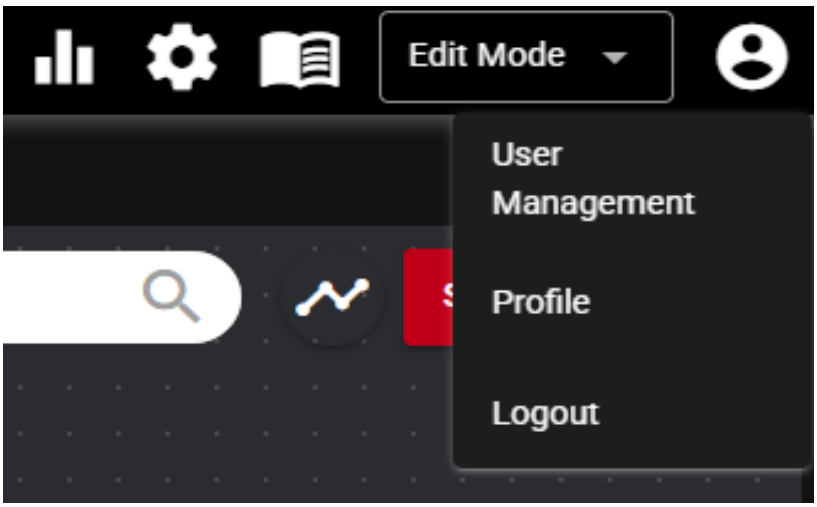
다음 순서로 사용자 정보를 편집할 수 있습니다.

1. 사용자 관리 아이콘을 누릅니다.

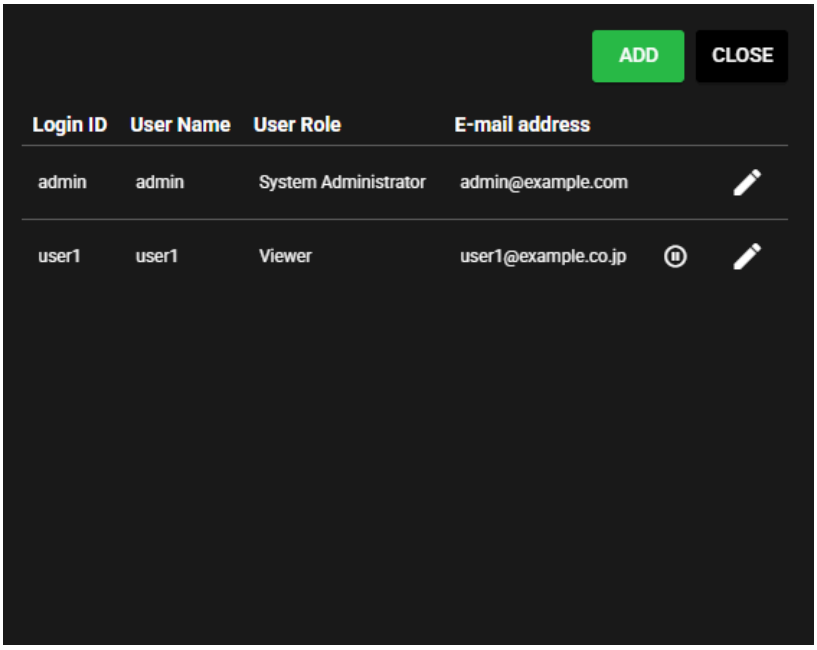


2.「 사용자 관리 」를 선택합니다.

사내 배포본
For Reference Only-

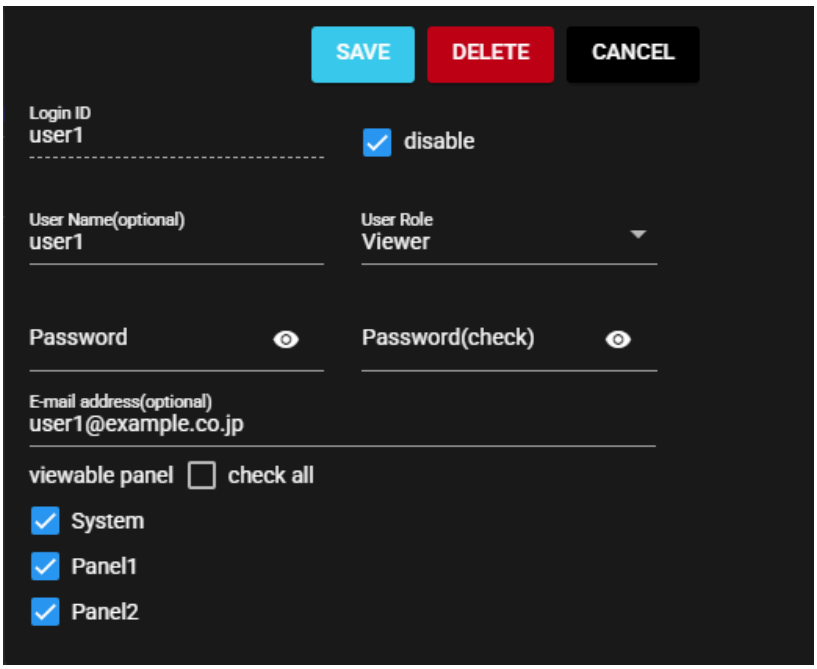


3. 편집할 사용자의 편집 아이콘을 누릅니다.



4. 다음 항목을 편집하고「 저장 」을 누릅니다.

AI 번역본
-For Reference Only-



항목	설명
무효로 한다	체크를 붙이면 사용자가 무효가 되어 로그인을 할 수 없게 됩니다.
로그인 ID	로그인 시에 입력하는 ID
사용자명	사용자에게 부여하는 이름※생략 가능
권한	시스템 관리자：모든 조작이 가능합니다. 초기 사용자 "admin"에게 부여되어 있으며, 1명 이상 필수입니다. 관리자：모든 조작이 가능합니다. 시스템 관리자의 추가・갱신・삭제는 불가입니다. 오퍼레이터：Run Mode의 열람, 및 컴포넌트의 기동・정지가 가능합니다. 열람자：Run Mode의 열람이 가능합니다.
패스워드	로그인 시에 입력하는 패스워드
E메일 주소	사용자에 연결하는 E메일 주소※생략 가능
열람 가능 패널	사용자가 열람 가능한 패널※권한이 오퍼레이터・열람자인 경우에만 설정합니다.

AI 번역본
-For Reference Only-

5. 편집한 내용이 저장됩니다.

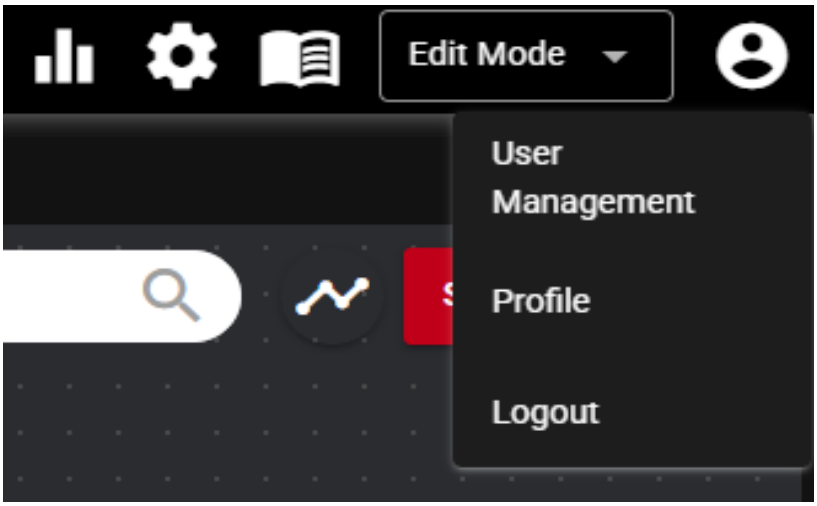
5.3.3.2.3 삭제

다음 순서로 사용자를 삭제할 수 있습니다.

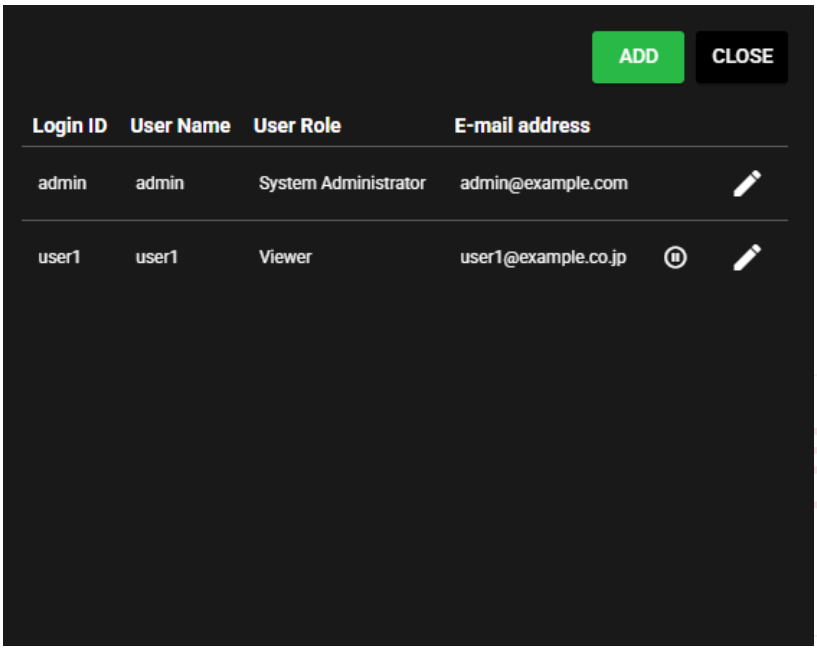
1. 사용자 관리 아이콘을 누릅니다.



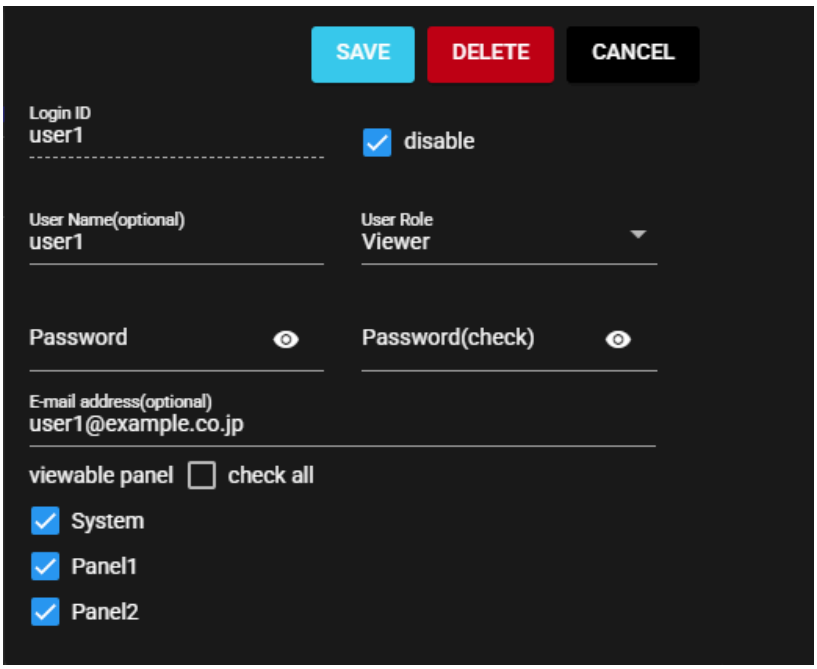
2. 「 사용자 관리 」를 선택합니다.



3. 삭제할 사용자의 편집 아이콘을 누릅니다.



4. 「 삭제 」를 누릅니다.



5. 사용자가 삭제됩니다.

> 시스템 관리자에 대해
>
> 시스템 관리자 사용자는 1명 이상 필수입니다. 시스템 관리자가 1명인 경우는 삭제할 수 없습니다.

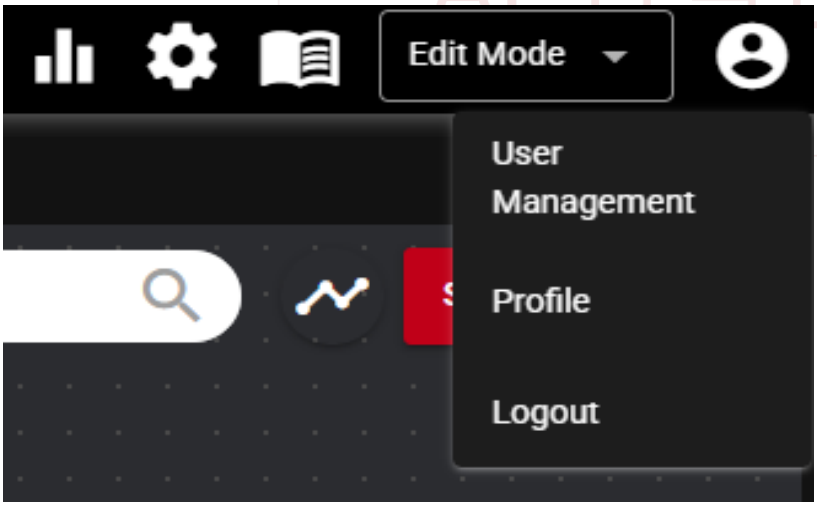
5.3.3.2.4 프로필

다음 순서로 현재 로그인하고 있는 사용자의 정보를 확인할 수 있습니다.

1. 사용자 관리 아이콘을 누릅니다.

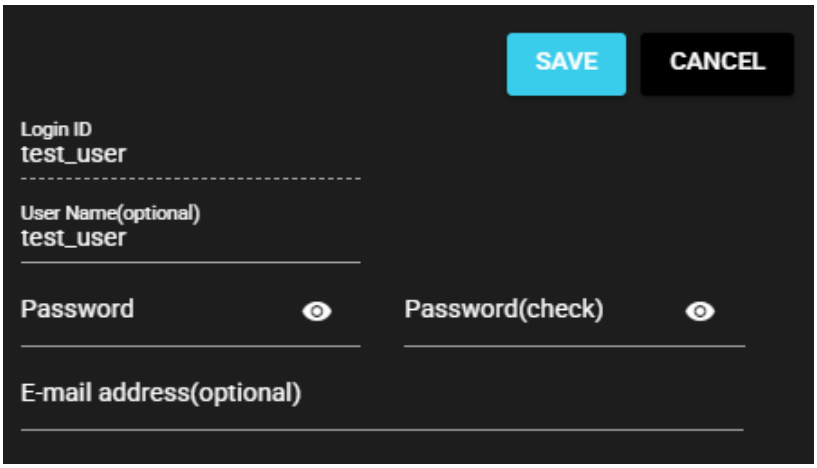


2. 「 프로필 」을 선택합니다.



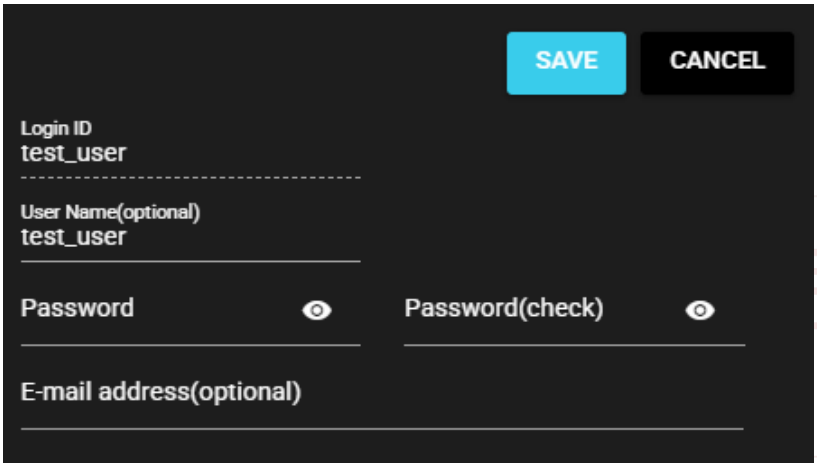
AI 번역본
-Only-

3. 현재 로그인하고 있는 사용자에 대해 다음 항목이 표시됩니다.



항목	설명
로그인 ID	로그인 시에 입력하는 ID
사용자명	사용자에게 부여하는 이름※생략 가능
패스워드	로그인 시에 입력하는 패스워드
E메일 주소	사용자에 연결하는 E메일 주소※생략 가능

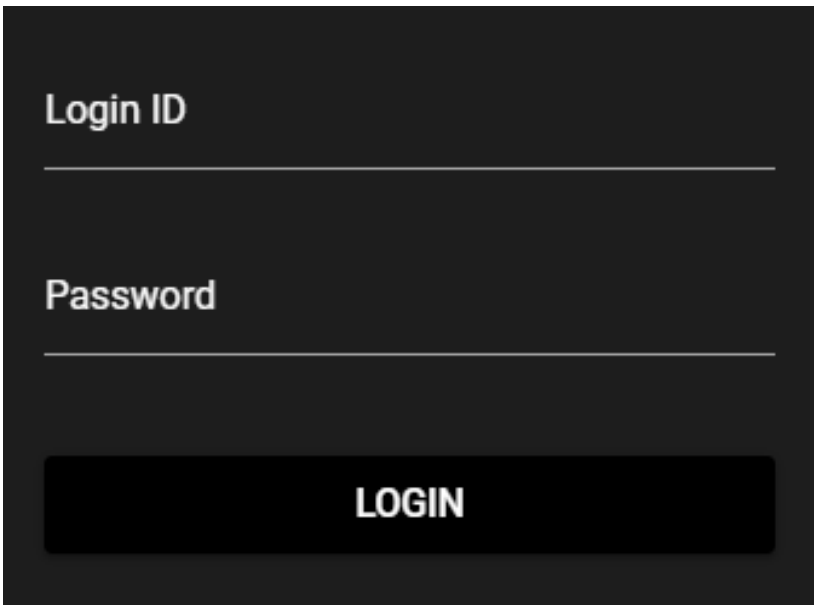
4. 항목을 편집하고 「 저장 」을 누르면 편집한 내용이 저장됩니다.



5.3.3.2.5 로그인

다음 순서로 SpeeDBee Synapse에 로그인할 수 있습니다.

1. 「 http://{ SpeeDBee Synapse가 가동하고 있는 환경의 IP 주소 }:8120 」에 액세스합니다.
2. 사용자의 「 로그인 ID 」와 「 패스워드 」를 입력하고 「 로그인 」을 누릅니다.



3. SpeeDBee Synapse에 로그인합니다.

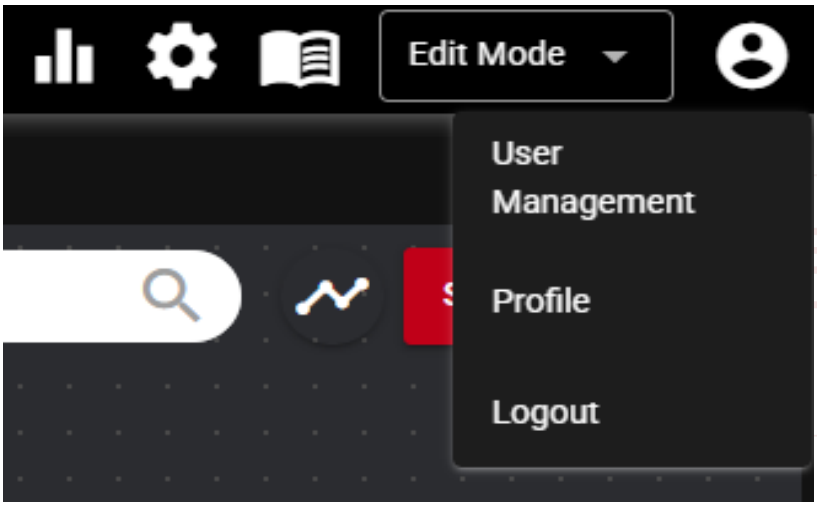
5.3.3.2.6 로그아웃

다음 순서로 SpeeDBee Synapse에서 로그아웃할 수 있습니다.

1. 사용자 관리 아이콘을 누릅니다.



2. 「 로그아웃 」을 선택합니다.



3. SpeeDBee Synapse에서 로그아웃합니다.

5.3.3.3 패널

여기에서는 패널의 조작에 대해 설명합니다.

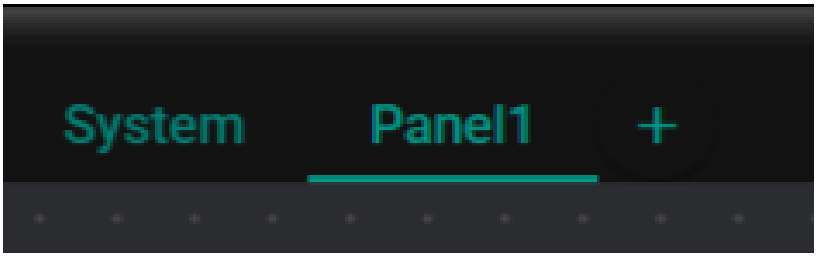
패널의 열람 권한에 대해

오퍼레이터・열람자 사용자인 경우는, 사용자 관리의 설정에서 패널별로 열람 권한을 설정할 수 있습니다. 복수의 패널에서 다른 데이터 수집 ~ 출력・연계까지의 플로우를 작성하여, 오퍼레이터・열람자에게 임의의 패널만을 표시하는 것이 가능합니다.

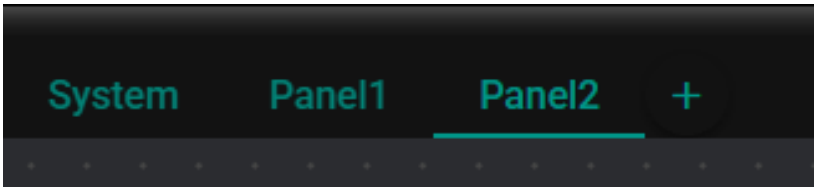
5.3.3.3.1 추가

다음 순서로 패널을 추가할 수 있습니다.

1. 패널의 탭 메뉴에 있는 ＋아이콘을 누릅니다.



2. 패널이 추가됩니다.



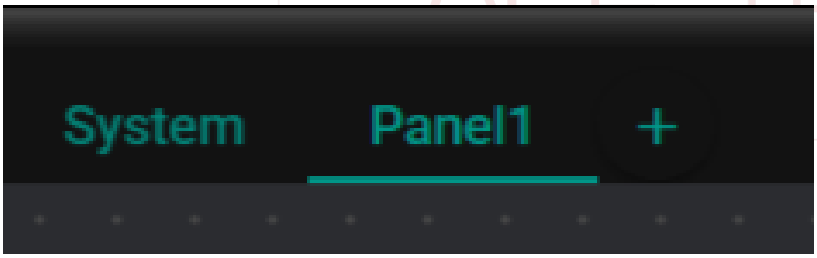
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.3.2 이름의 변경

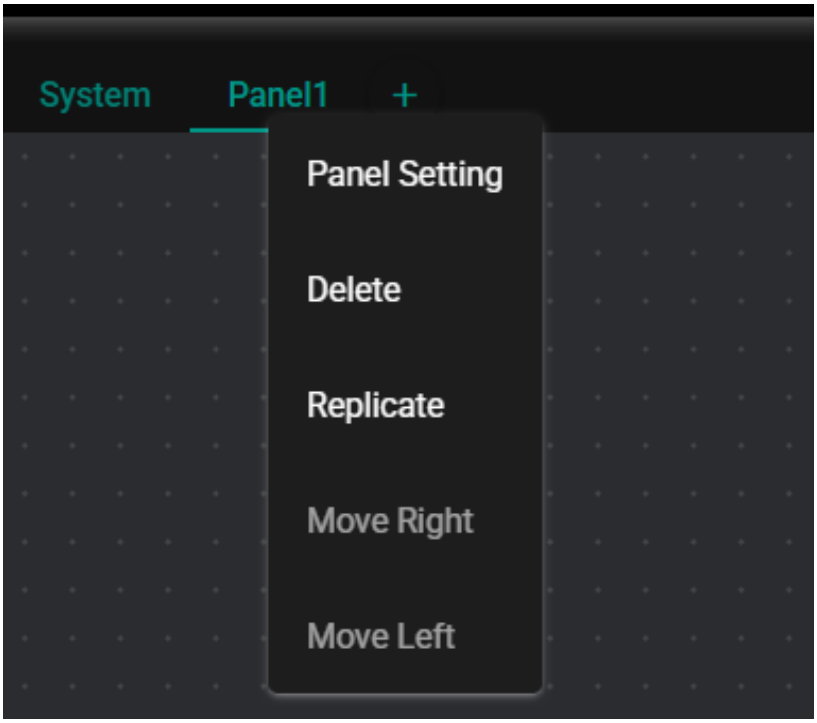
다음 순서로 패널명을 변경할 수 있습니다.

AI 번역본
Only-

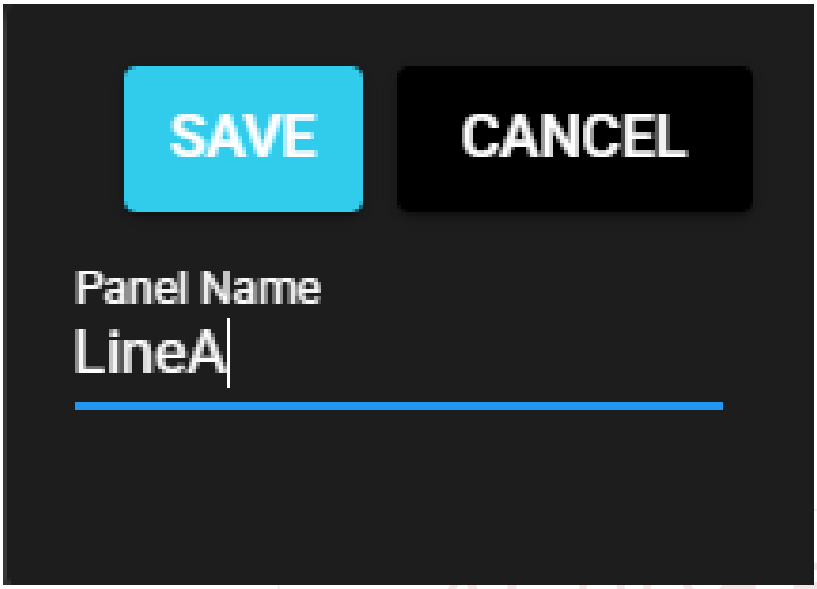
1. 패널 탭을 오른쪽 클릭합니다.



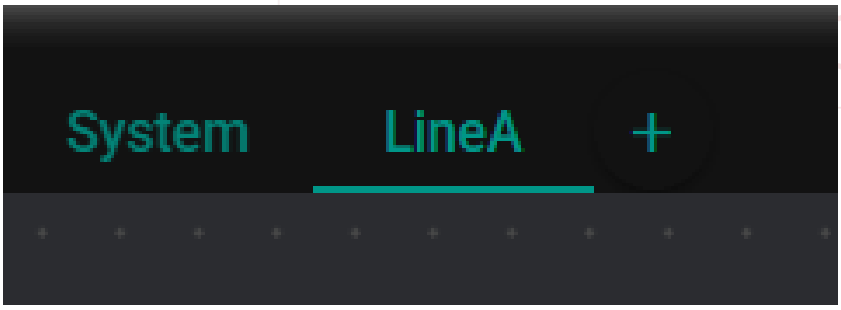
2.「패널 설정」을 선택합니다.



3. 패널명을 입력하고「 저장 」을 누릅니다.



4. 패널명이 입력한 것으로 변경됩니다.



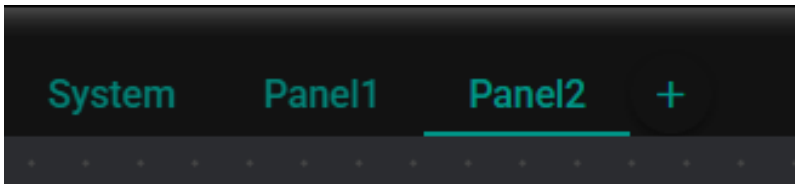
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.3.3 정렬

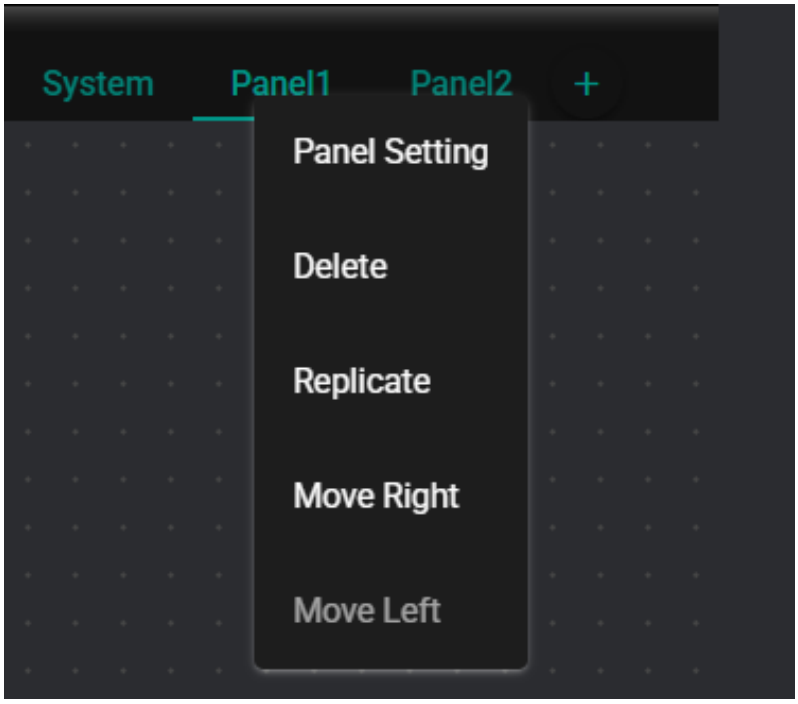
다음 중 어느 하나의 순서로 패널을 정렬할 수 있습니다.

- 패널 탭의 컨텍스트 메뉴에서 정렬

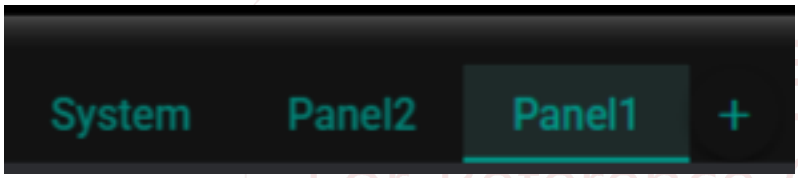
  a. 패널 탭을 오른쪽 클릭합니다.



  b.「 오른쪽으로 이동 」 또는 「 왼쪽으로 이동 」을 선택합니다.



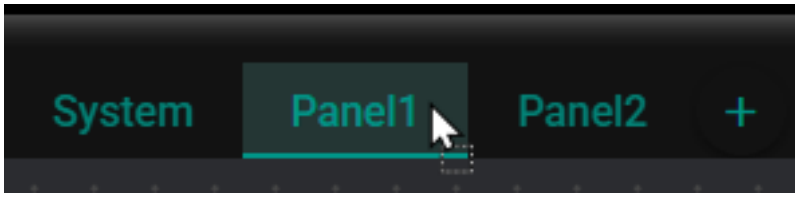
  c. 패널이 정렬됩니다.



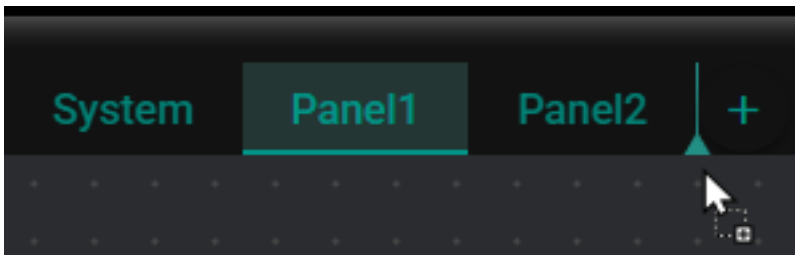
-For Reference Only-

- 패널 탭의 드래그&드롭에 의한 정렬

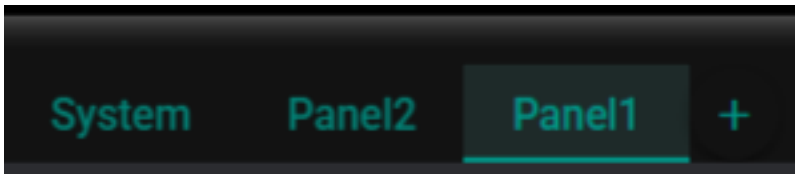
  a. 패널 탭을 드래그합니다.



  b. 이동하고 싶은 위치에 드롭합니다.



c. 패널이 정렬됩니다.

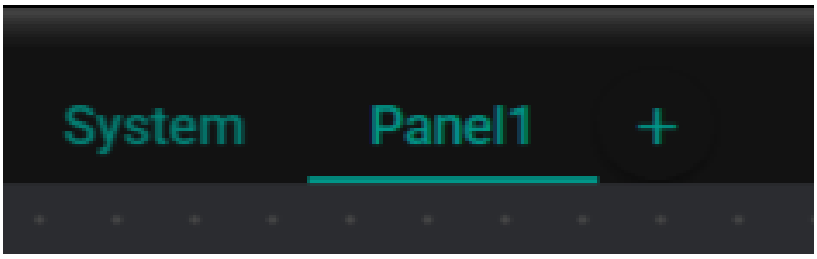


> 이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.3.4 삭제

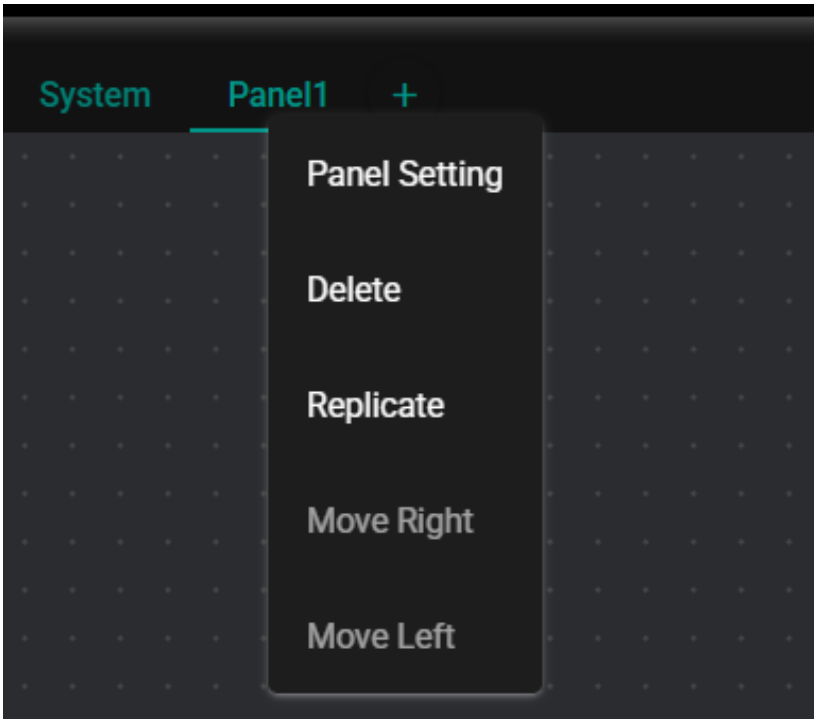
다음 순서로 패널을 삭제할 수 있습니다.

1. 패널 탭을 오른쪽 클릭합니다.

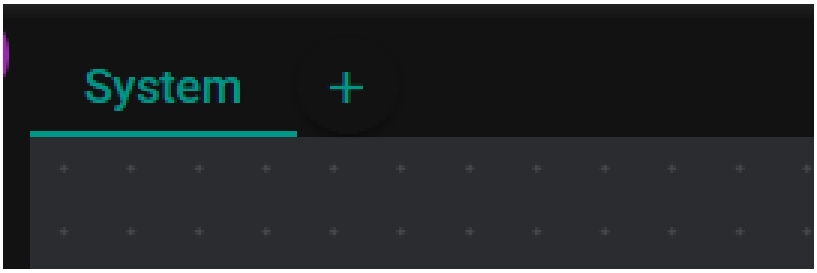


2. 「 삭제 」를 선택합니다.

AI 번역본
-For Reference Only-



3. 패널이 삭제됩니다.



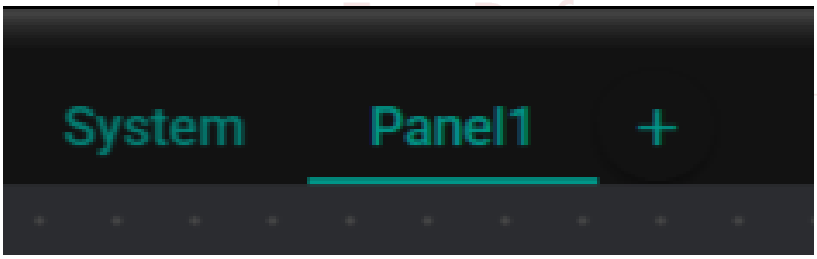
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.3.5 복제

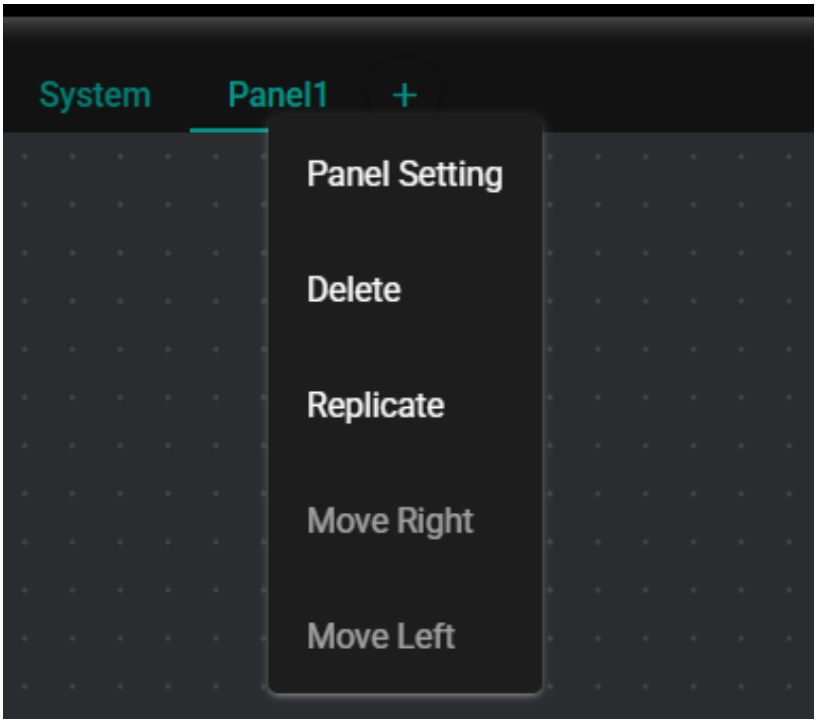
다음 순서로 같은 설정 내용의 패널을 복제할 수 있습니다.

AI 번역본
Only-

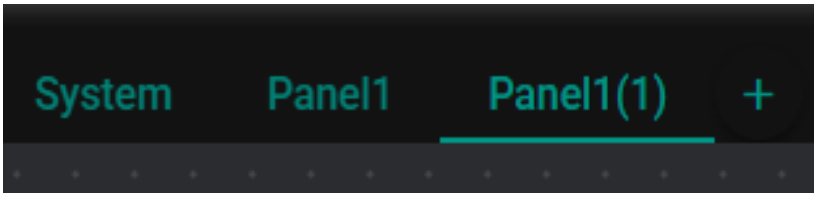
1. 패널 탭을 오른쪽 클릭합니다.



2. 「 복제 」를 선택합니다.



3. 복제원과 같은 설정 내용의 패널이 복제됩니다.

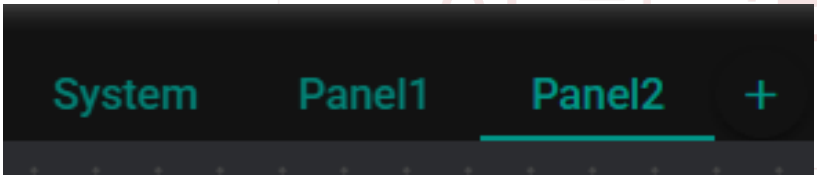


이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.3.6 표시

다음 순서로 패널을 표시할 수 있습니다.

1. 표시할 패널의 탭을 클릭합니다.



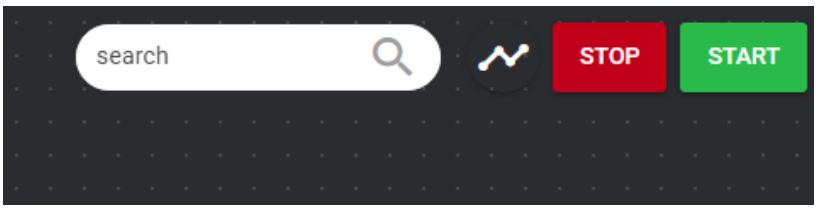
2. 탭을 클릭한 패널이 표시됩니다.

5.3.3.3.7 컴포넌트 검색

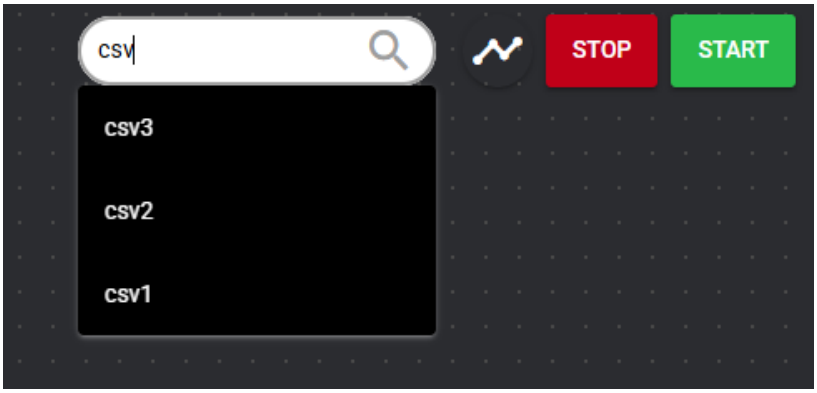
다음 순서로 패널에 등록되어 있는 컴포넌트의 위치를 검색할 수 있습니다.

1. 패널의 우측 상단의 검색 상자에 컴포넌트명을 입력합니다.

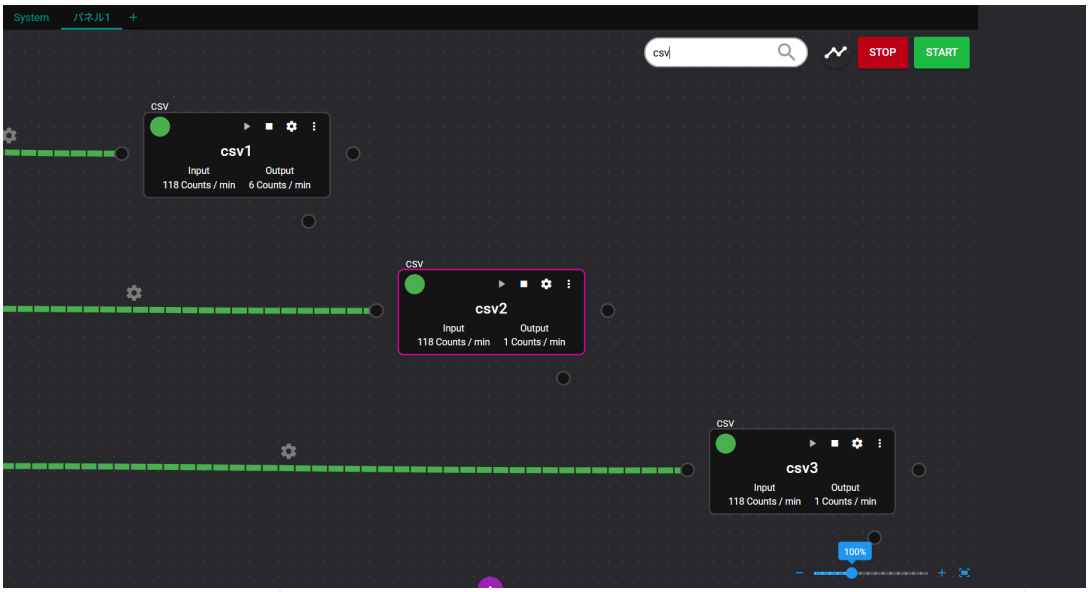
시 버 연 본
Only-



2. 후보 중에서 선택합니다.



3. 선택한 컴포넌트가 패널의 중앙에 표시됩니다.

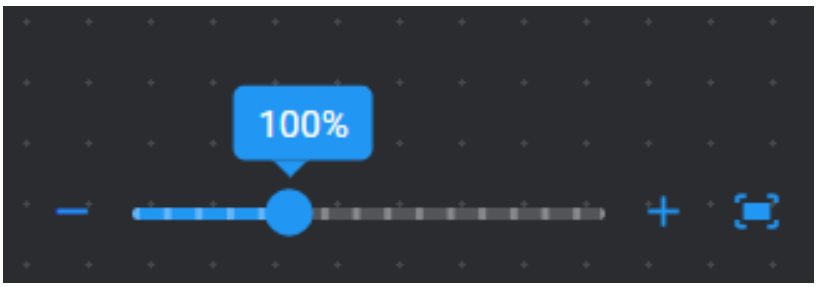


AI 번역본
-For Reference Only-

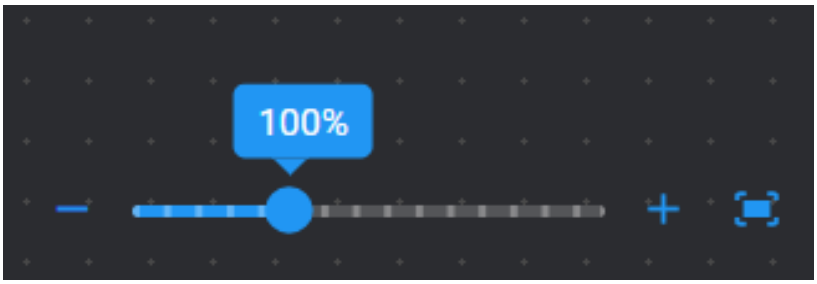
5.3.3.3.8 표시 배율의 조정

다음 순서로 패널의 표시 배율을 조정할 수 있습니다.

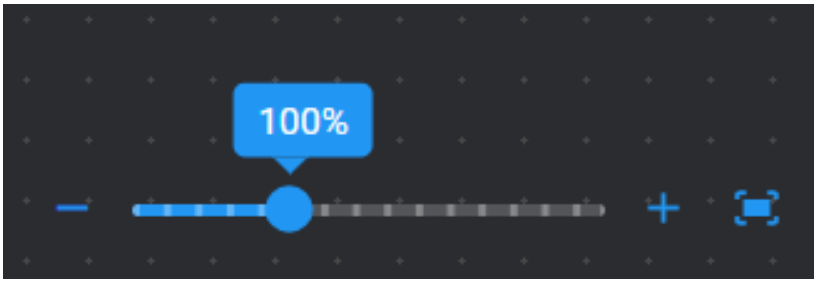
1. 패널의 우측 하단에 패널의 현재 표시 배율이 표시됩니다.



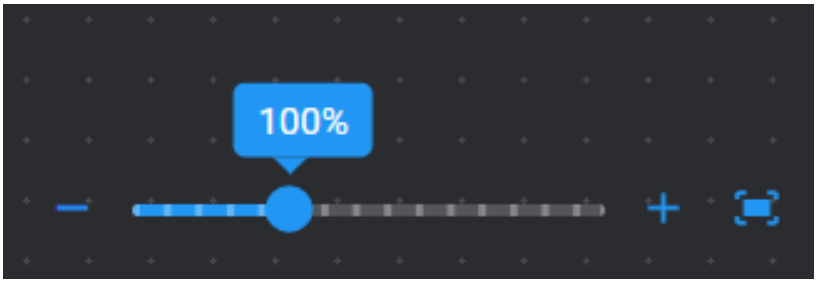
2. 슬라이드 바를 +방향으로 움직이면 표시를 확대하고, -방향으로 움직이면 표시를 축소합니다.



3. +를 누르면 표시 배율을 5% 확대하고, -를 누르면 표시 배율을 5% 축소합니다.



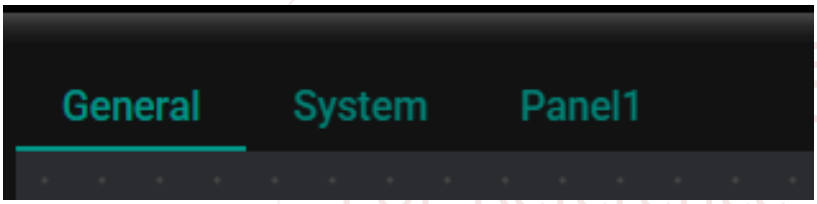
4. 전체 표시 버튼을 누르면 패널에 배치된 컴포넌트 전체를 화면에 표시합니다.



5.3.3.3.9 전체 패널

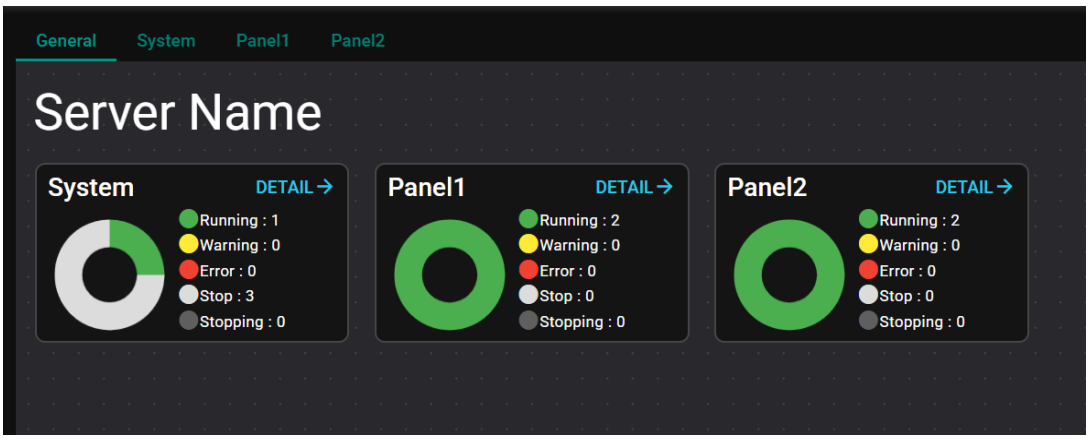
다음 순서로 전체 패널을 표시할 수 있습니다. 전체 패널은 각 패널의 정보를 표시하는 패널입니다.

1. 패널 탭에서 「 전체 」를 선택합니다.

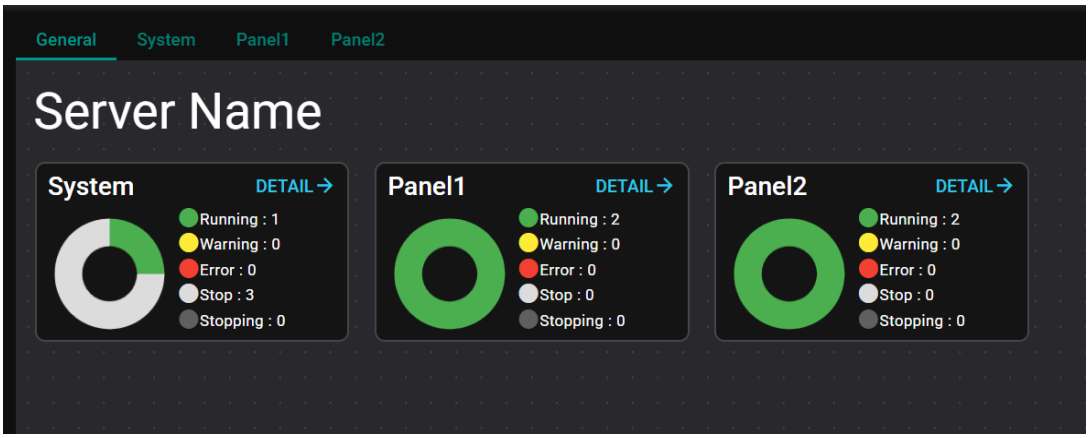


2. 서버 인스턴스명・패널의 목록・각 패널의 컴포넌트의 가동 스테이터스가 표시됩니다.

For Reference Only-



3. 「 DETAIL 」을 누르면 각 패널로 이동합니다.

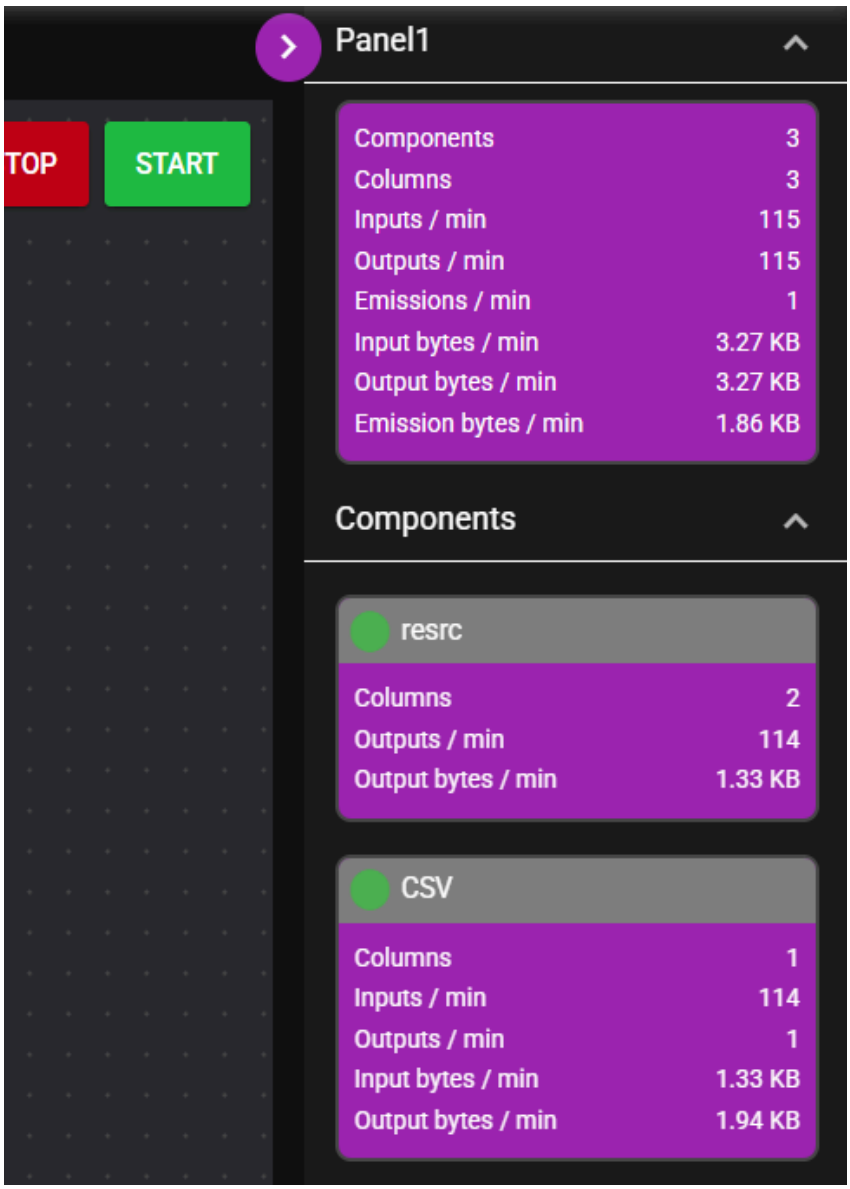


이 조작은 Run Mode에서만 수행할 수 있습니다.

5.3.3.3.10 패널 스테이터스

패널마다 다음 정보가 표시됩니다.

AI 번역본
-For Reference Only-

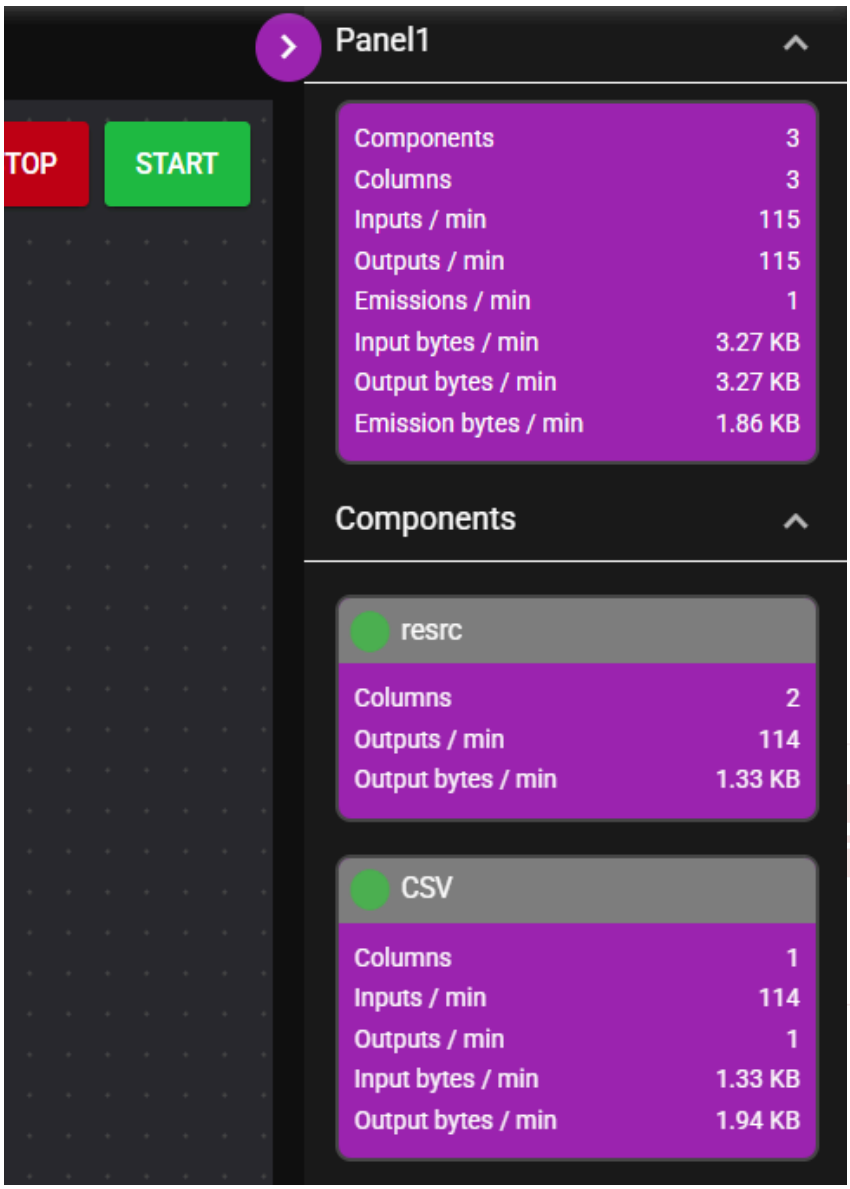


항목	설명
컴포넌트 수	패널상의 모든 컴포넌트의 수
컬럼 수	패널상의 모든 컴포넌트의 컬럼의 수
입력 데이터 건수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 입력 건수의 합계
출력 데이터 건수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 출력 건수의 합계
수신 데이터 건수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 수신 건수의 합계
송신 데이터 건수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 송신 건수의 합계

AI 번역본
-For Reference Only-

항목	설명
입력 데이터 바이트 수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 입력 바이트 수의 합계
출력 데이터 바이트 수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 출력 바이트 수의 합계
수신 데이터 바이트 수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 수신 바이트 수의 합계
송신 데이터 바이트 수/분	패널상의 모든 컴포넌트의 1분마다의 데이터 송신 바이트 수의 합계

패널상에 있는 가동 중인 컴포넌트에 대해 다음 정보가 표시됩니다.



항목	설명
컬럼 수	컴포넌트가 출력하는 데이터의 항목 수
입력 데이터 건수/분	1분마다의 데이터 입력 건수
출력 데이터 건수/분	1분마다의 데이터 출력 건수
수신 데이터 건수/분	1분마다의 데이터 수신 건수
송신 데이터 건수/분	1분마다의 데이터 송신 건수
입력 데이터 바이트 수/분	1분마다의 데이터 입력 바이트 수
출력 데이터 바이트 수/분	1분마다의 데이터 출력 바이트 수
수신 데이터 바이트 수/분	1분마다의 데이터 수신 바이트 수
송신 데이터 바이트 수/분	1분마다의 데이터 송신 바이트 수

이 조작은 Run Mode에서만 수행할 수 있습니다.

5.3.3.4 컴포넌트

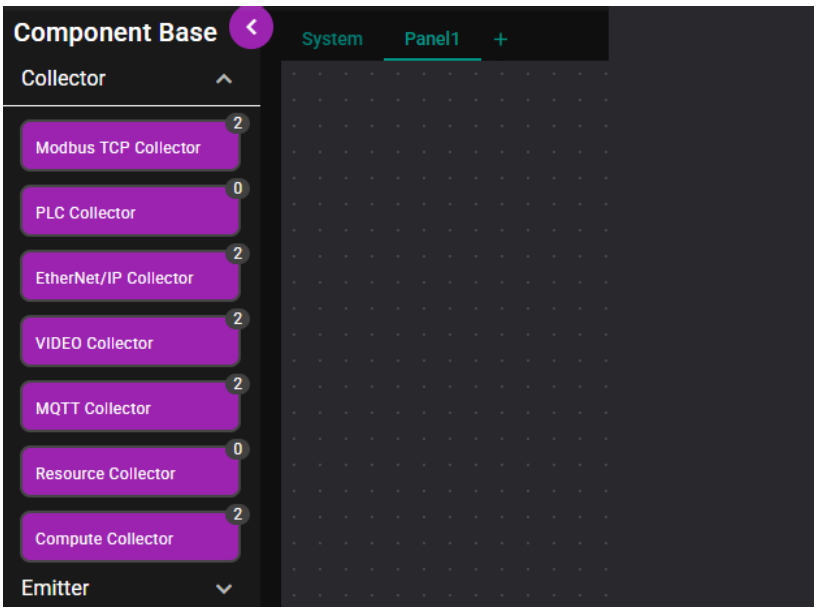
여기에서는 컴포넌트의 조작에 대해 설명합니다.

5.3.3.4.1 추가

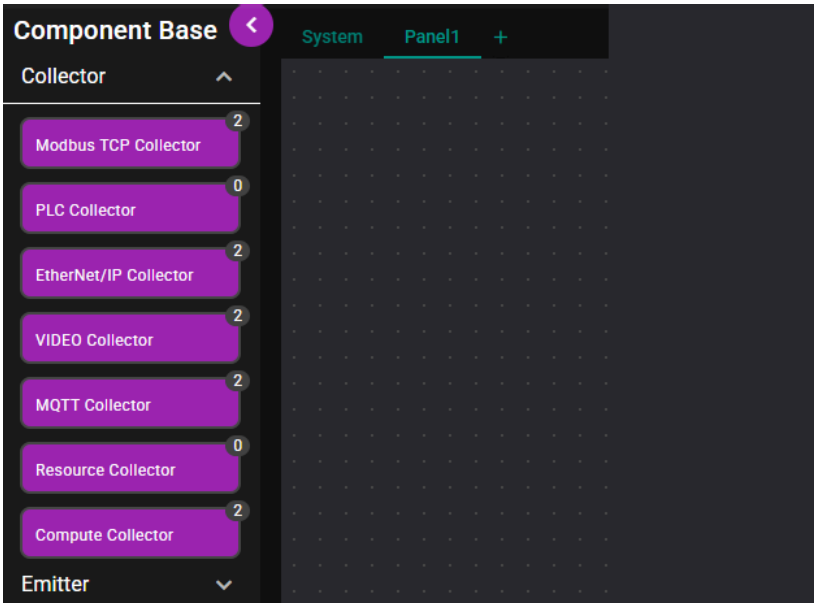
다음 순서로 컴포넌트를 추가할 수 있습니다.

1. 왼쪽 메뉴에서 컴포넌트를 선택하고, 아이콘을 드래그합니다.

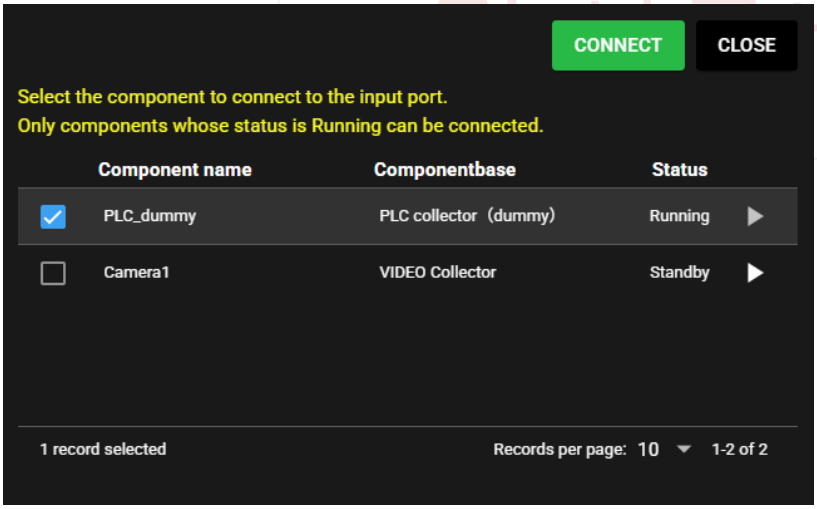
AI 번역본
-For Reference Only-



2. 패널 위에 드롭합니다.

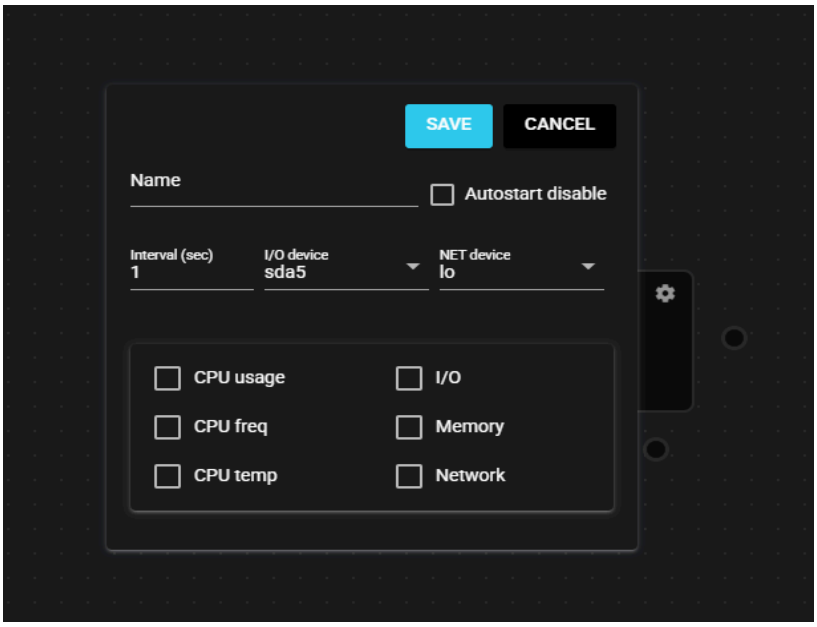


3. 일부 컴포넌트만, 입력 포트에 접속할 컴포넌트의 선택 다이얼로그가 표시됩니다. 접속할 컴포넌트를 선택하고「선택」을 누릅니다.

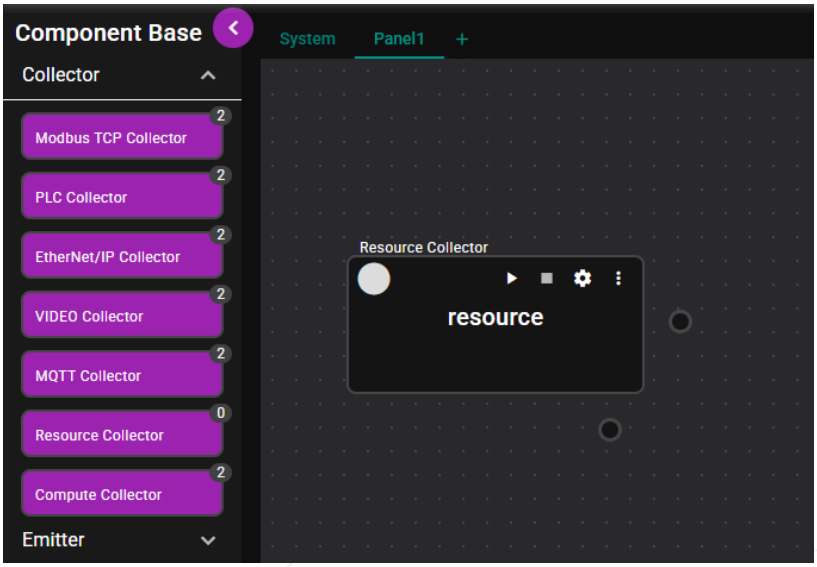


AI 번역 본
-Only-

4. 설정 다이얼로그가 열리므로, 설정 항목을 입력하고 「 저장 」을 누릅니다.



5. 패널에 컴포넌트가 추가됩니다.



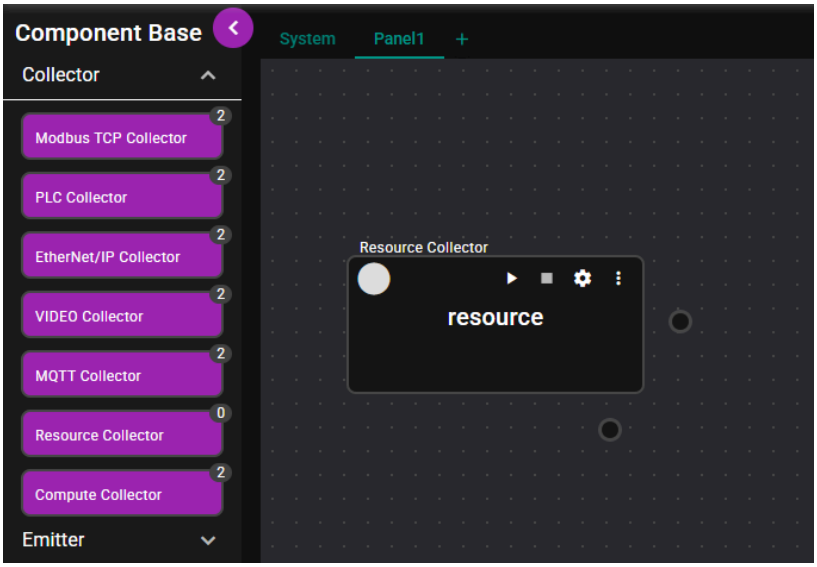
-For Reference Only-

이 조작은 Edit Mode에서만 수행할 수 있습니다.

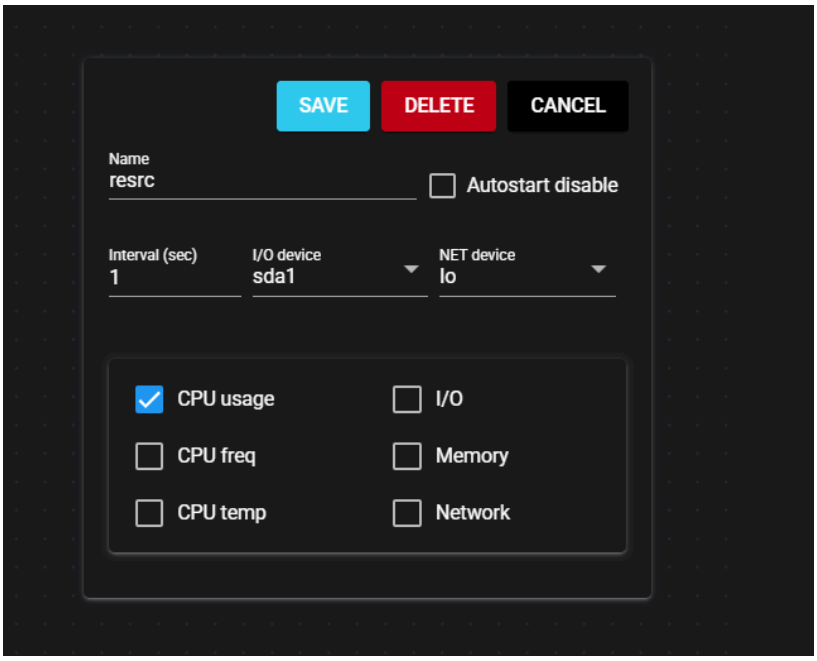
5.3.3.4.2 편집

다음 순서로 컴포넌트의 설정을 편집할 수 있습니다.

1. 컴포넌트의 설정 아이콘을 누릅니다.



2. 설정 다이얼로그가 열리므로, 설정 항목을 편집하고「 저장 」을 누릅니다.



3. 편집한 설정이 저장됩니다.

AI 번역본
For Reference Only

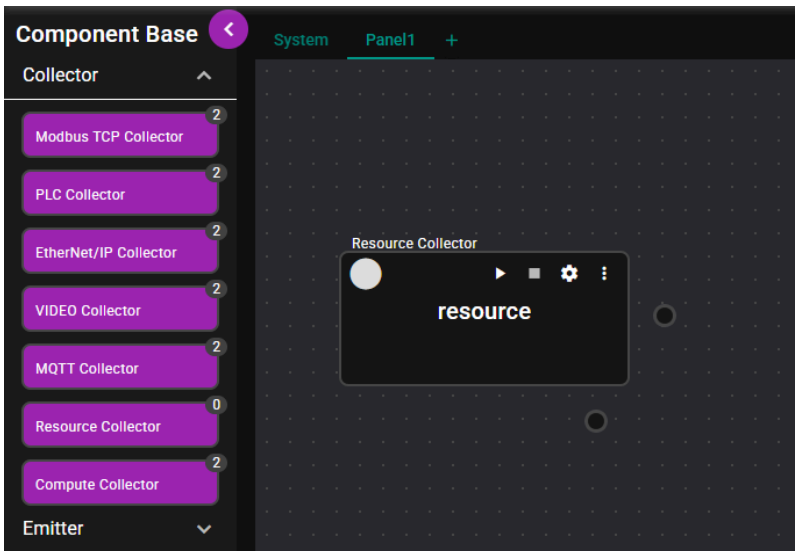
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.4.3 삭제

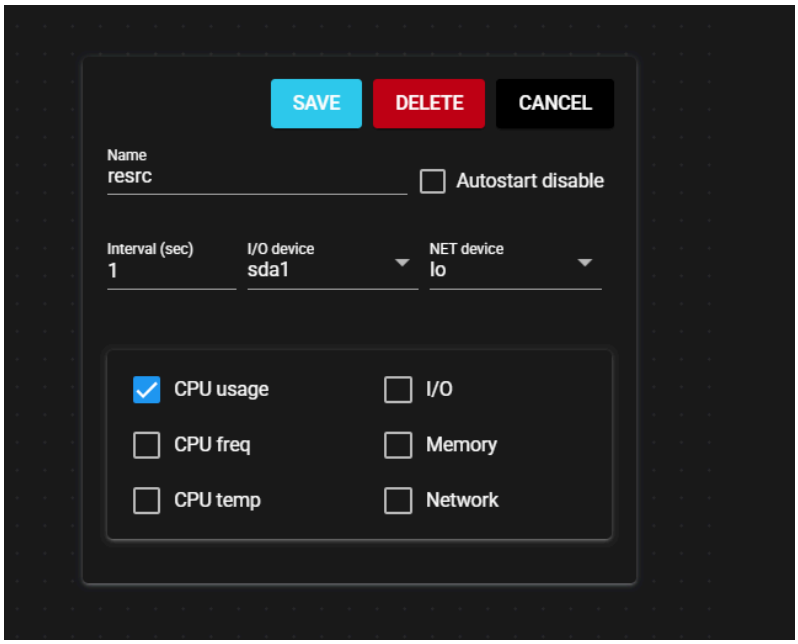
다음 중 어느 하나의 순서로 컴포넌트를 삭제할 수 있습니다.

- 컴포넌트의 설정 아이콘에서 컴포넌트를 삭제하는 방법입니다.

  a. 컴포넌트의 설정 아이콘을 누릅니다.



b. 설정 다이얼로그가 열리므로,「 삭제 」를 누릅니다.

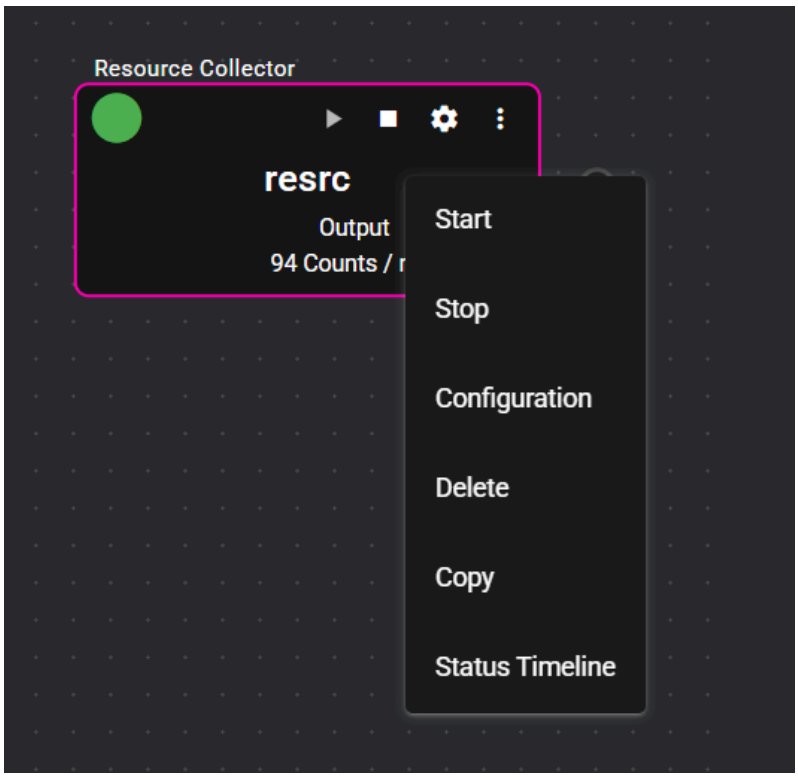


c. 컴포넌트가 삭제됩니다.

- 컴포넌트의 컨텍스트 메뉴상에서 컴포넌트를 삭제하는 방법입니다.

  a. 컴포넌트를 오른쪽 클릭하고,「 삭제 」를 선택합니다.

AI 번역본
-For Reference Only-



b. 컴포넌트가 삭제됩니다.

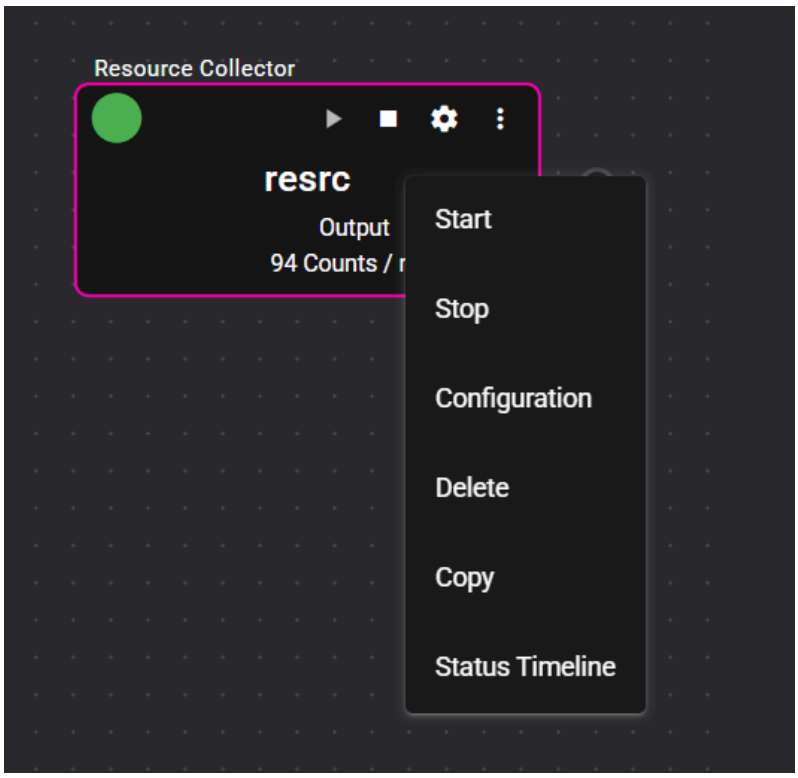
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.4.4 복사&붙여넣기

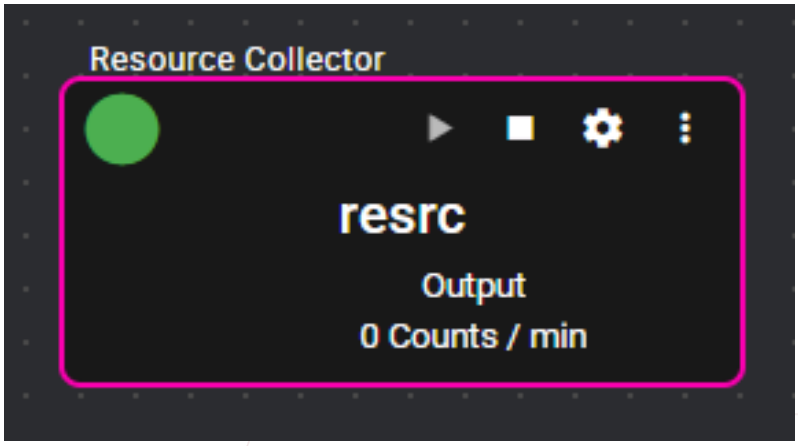
다음 순서로 같은 설정 내용의 컴포넌트를 복제할 수 있습니다.

1. 다음 중 어느 하나의 방법으로 컴포넌트를 복사합니다.
   - 컴포넌트를 오른쪽 클릭하고,「 복사 」를 선택합니다.

AI 번역본
-For Reference Only-

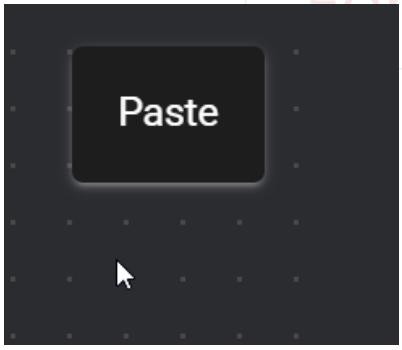


- 컴포넌트를 클릭하여 선택한 상태에서, 키보드의 「 Ctrl 」 키를 누르면서 「 C 」 키를 누릅니다.



2. 다음 중 어느 하나의 방법으로 컴포넌트를 붙여넣습니다.

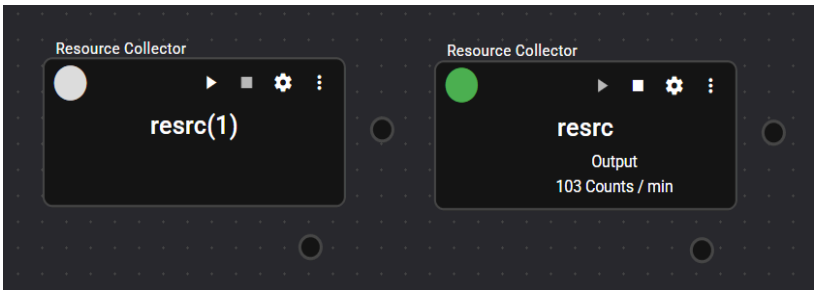
- 패널을 오른쪽 클릭하고, 「 붙여넣기 」 를 선택합니다.



- 키보드의 「 Ctrl 」 키를 누르면서 「 V 」 키를 누릅니다.

AI 번역본
For Reference Only-

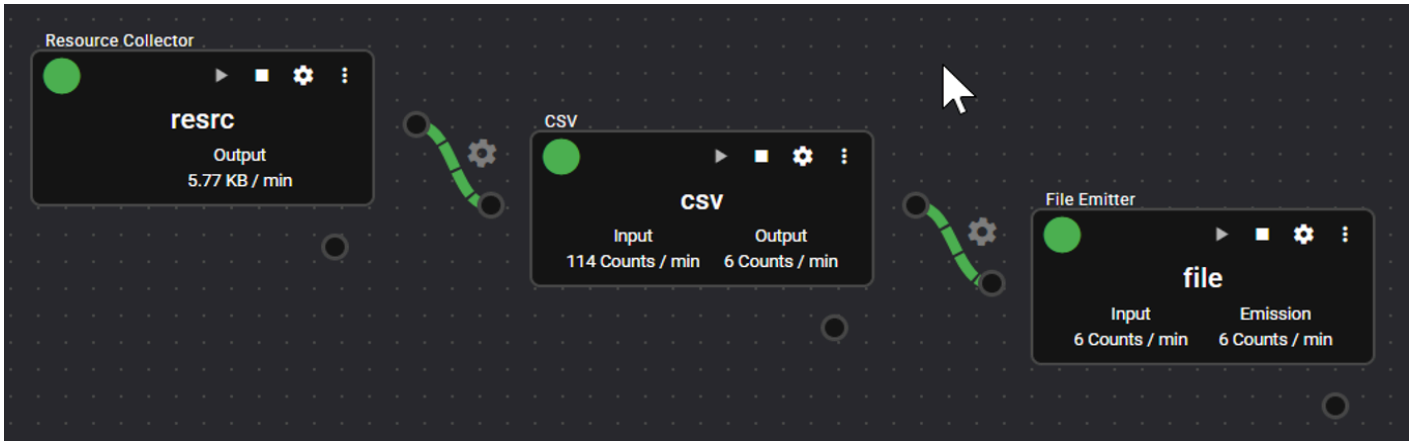
3. 복사한 것과 같은 설정 내용을 가진 컴포넌트가 작성됩니다.



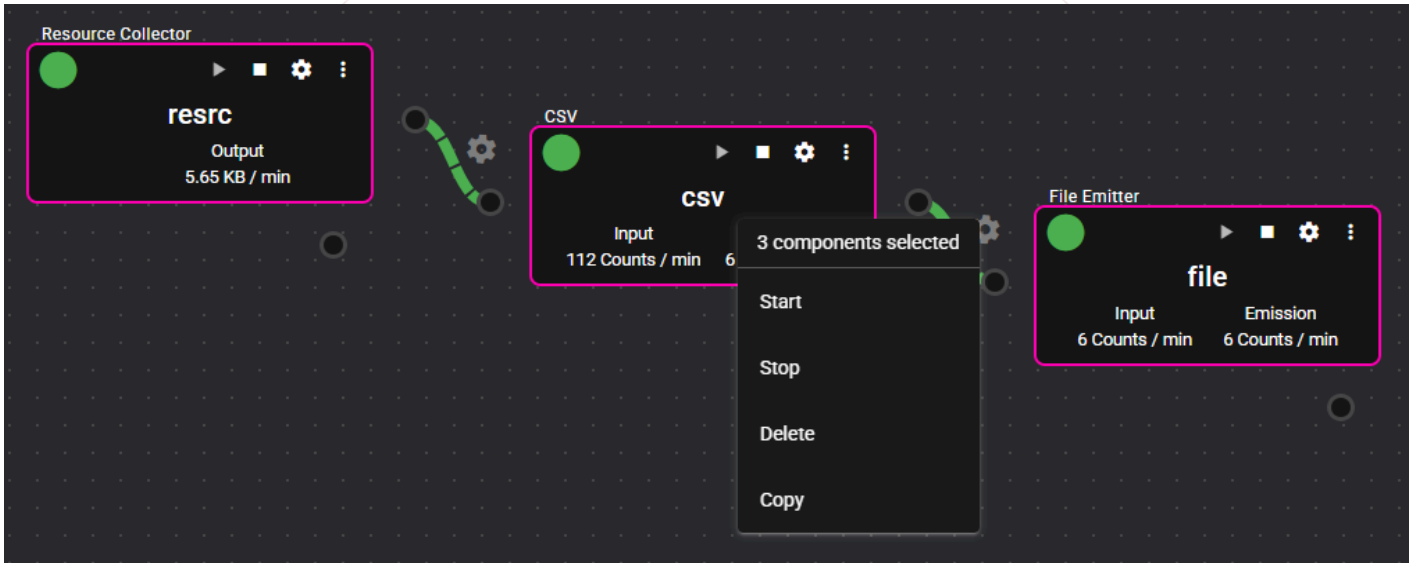
이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.4.5 복수 선택

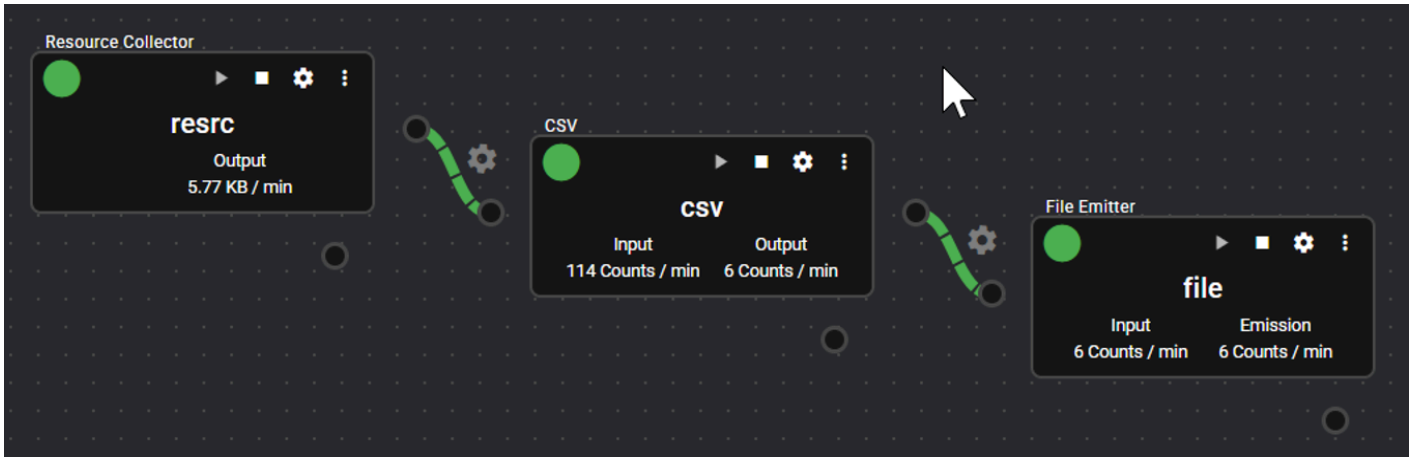
키보드의 「 Ctrl 」 키를 누르면서 각각의 컴포넌트를 클릭하면, 컴포넌트를 복수 선택할 수 있습니다.



복수 선택 후, 컨텍스트 메뉴나 키보드 조작에서 선택한 컴포넌트의 「 복사&붙여넣기 」 나 「 삭제 」 를 수행할 수 있습니다.



복수 선택을 해제하려면, 패널 내의 컴포넌트 이외의 장소를 클릭합니다.

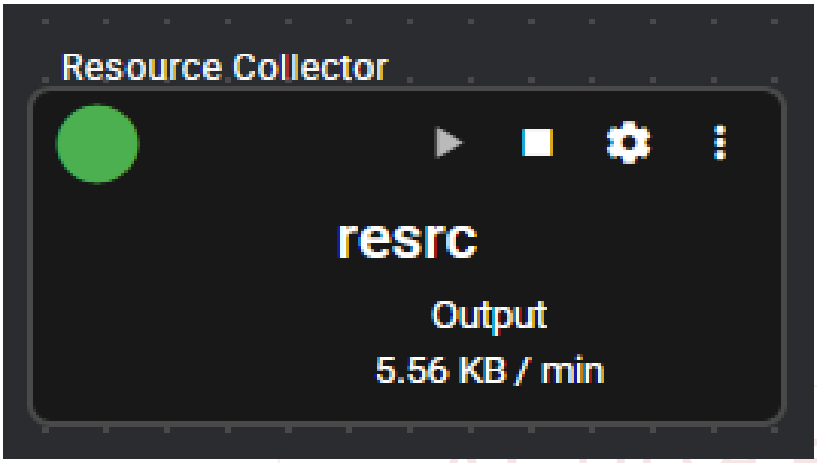


이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.3.3.4.6 설정 내용의 확인

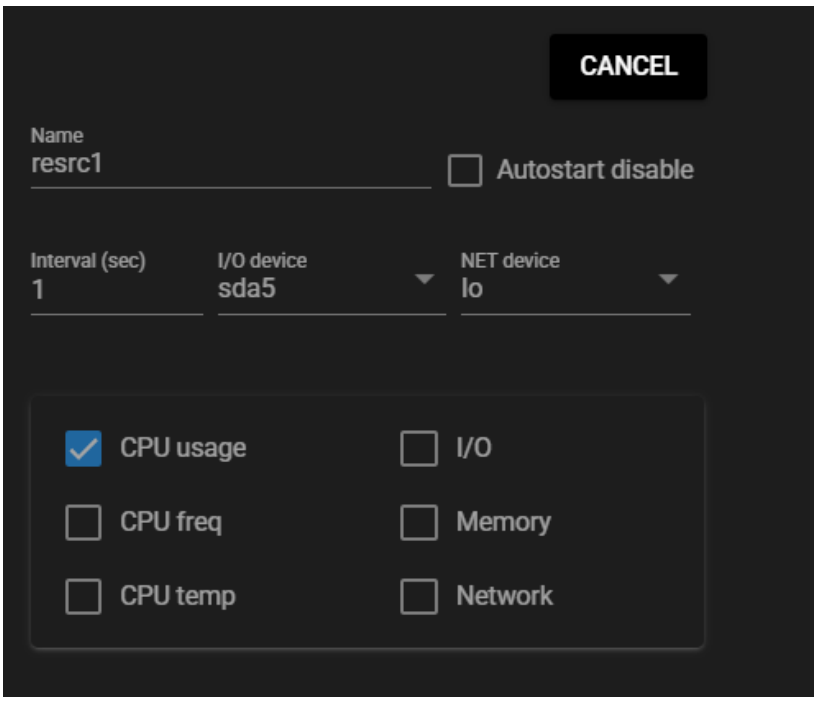
다음 순서로 컴포넌트의 설정 내용을 확인할 수 있습니다.

1. 컴포넌트의 설정 아이콘을 누릅니다.



2. 다이얼로그가 열리고, 컴포넌트의 설정 항목과 그 설정값이 표시됩니다.

AI 번역본
-For Reference Only-



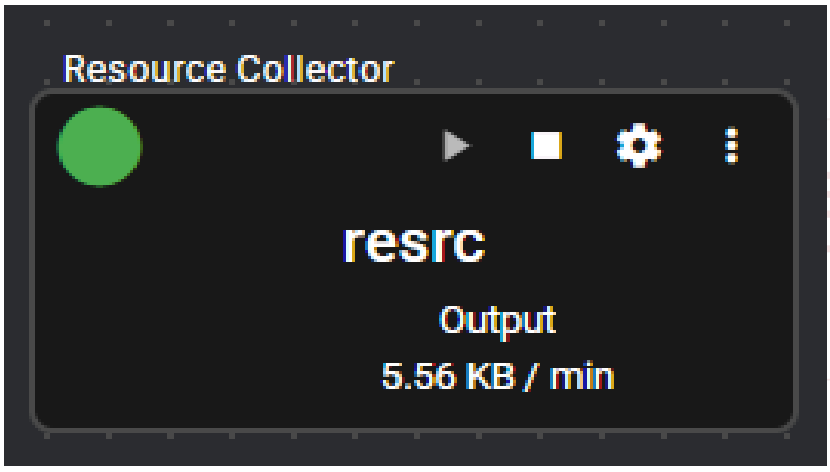
이 조작은 Run Mode에서만 수행할 수 있습니다.

5.3.3.4.7 가동 상황

다음 순서로, 컴포넌트의 가동 상황을 확인할 수 있습니다.

- 스테이터스

  스테이터스 램프에, 컴포넌트의 가동 스테이터스가 표시됩니다. 가동 스테이터스의 상세는,「스테이터스 목록」을 참조하십시오.



- 데이터양

  컴포넌트의 이하의 데이터양이 표시됩니다. 1분마다의

  - 입력・출력 데이터
  - 1분마다의 수신・송신 데이터

클릭하면, 데이터의 건수・바이트 수 중 어느 쪽을 표시할지, 전환할 수 있습니다.



5.3.3.4.8 기동/정지

다음 순서로, 컴포넌트를 기동/정지할 수 있습니다.

(전 컴포넌트)

1. 헤더의 「 기동 」 버튼을 누릅니다.



2. SpeeDBee Synapse에 등록되어 있는, 모든 데이터 수집 플로우의 가동을 개시합니다.



3. 헤더의 「 정지 」 버튼을 누릅니다.

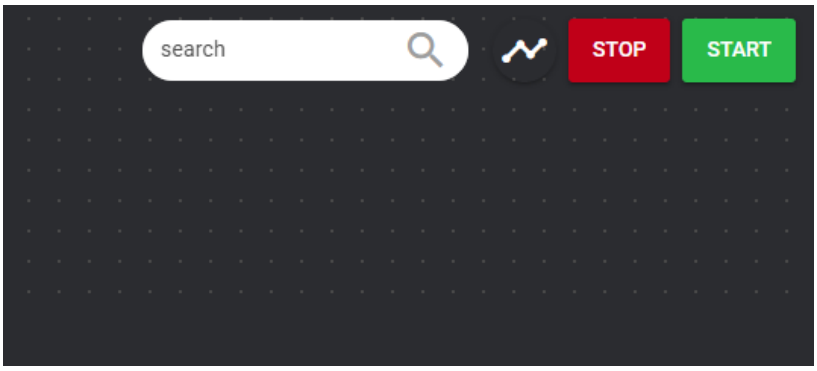


4. SpeeDBee Synapse에 등록되어 있는, 모든 데이터 수집 플로우의 가동을 정지합니다.



(패널 단위)

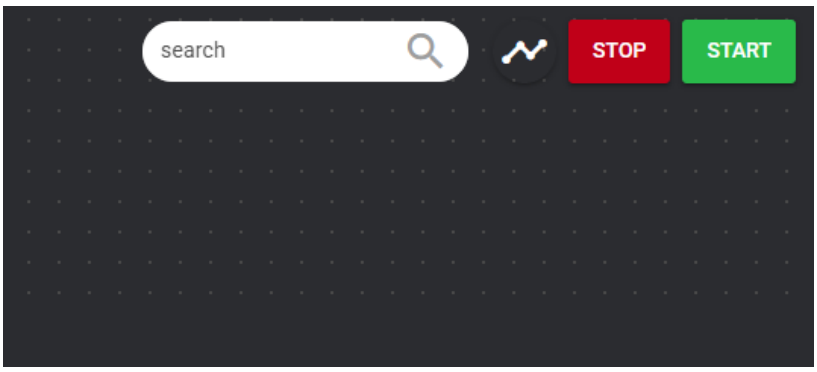
1. 패널의「 기동 」버튼을 누릅니다.



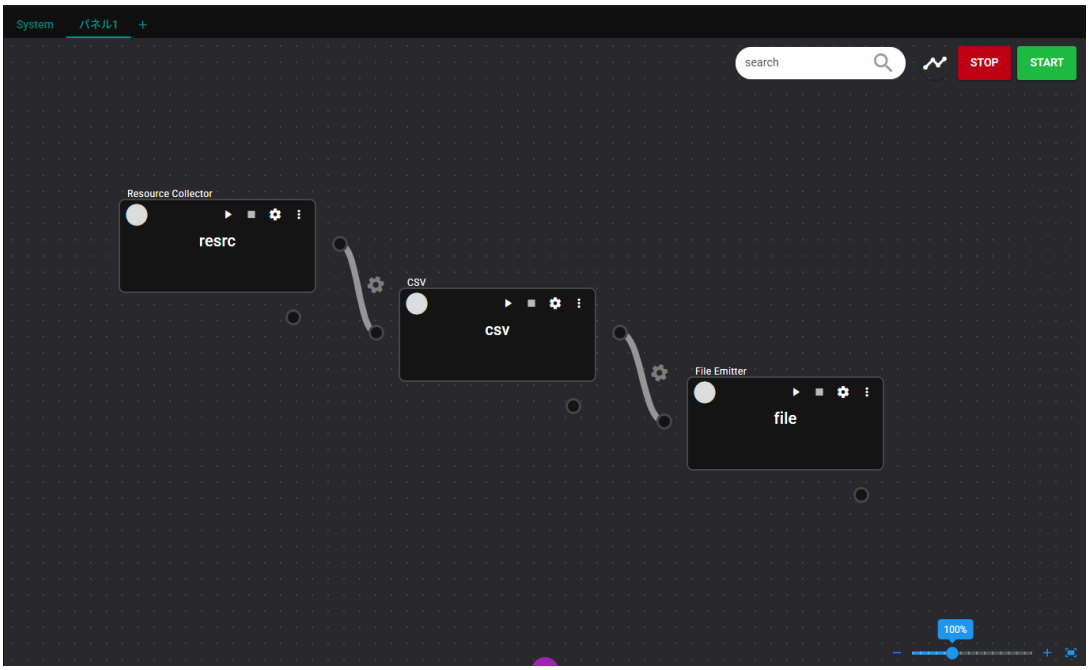
2. 패널에 등록되어 있는 데이터 수집 플로우의 가동을 개시합니다.



3. 패널의「 정지 」버튼을 누릅니다.



4. 패널에 등록되어 있는 데이터 수집 플로우의 가동을 정지합니다.

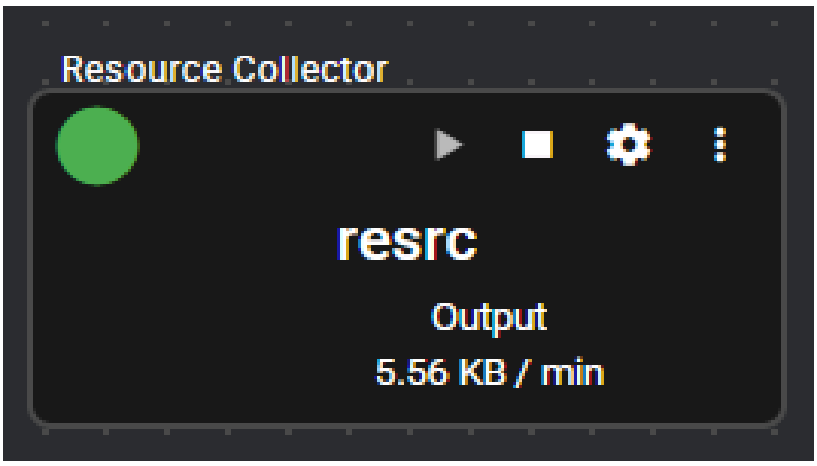


(컴포넌트 단체)

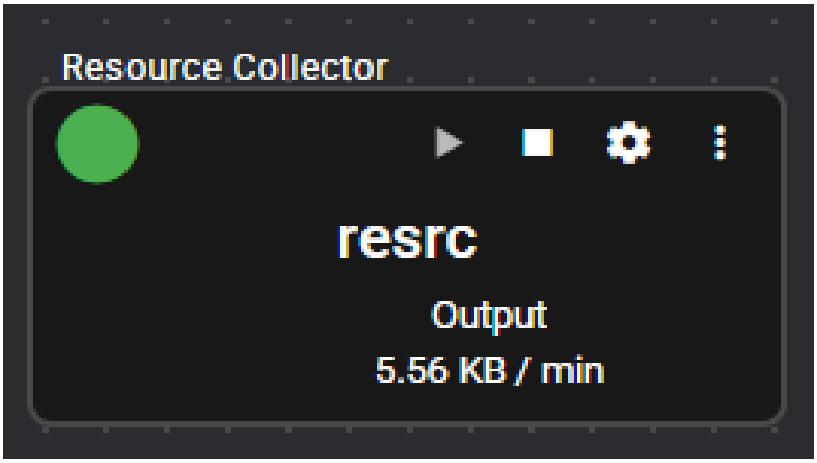
1. 컴포넌트의 기동 아이콘을 누릅니다.



2. 컴포넌트가 기동하여, 처리를 개시합니다.



3. 컴포넌트의 정지 아이콘을 누릅니다.



4. 컴포넌트가 정지하여, 처리를 종료합니다.



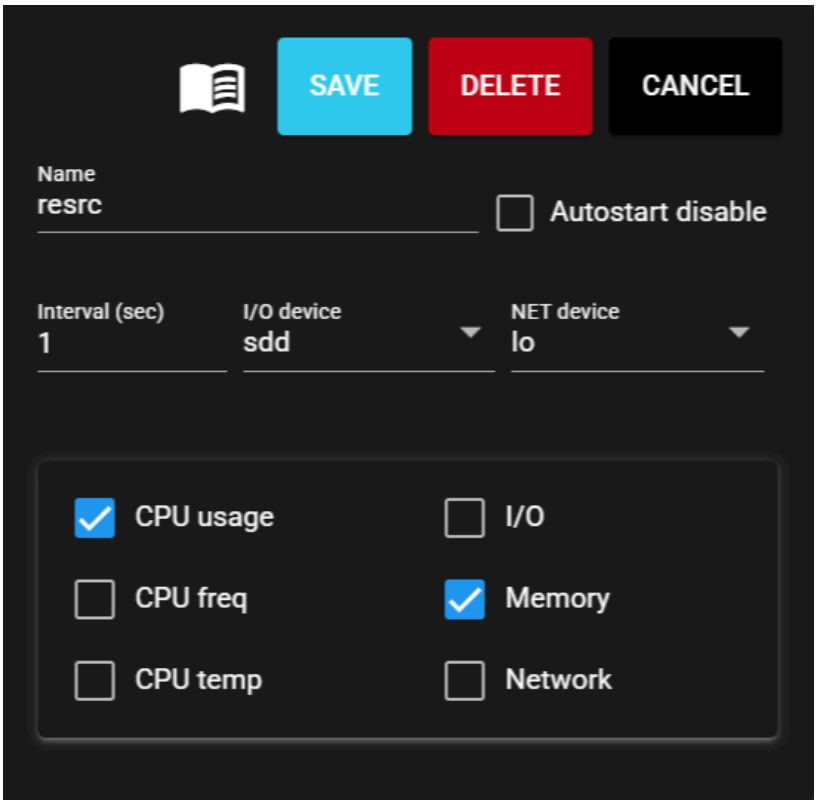
본
Only-

권한이 열람자인 사용자는, 컴포넌트의 기동/정지를 수행할 수 없습니다.

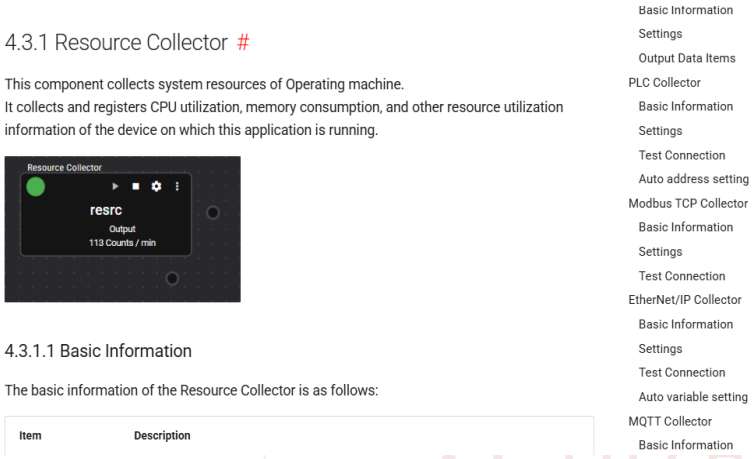
5.3.3.4.9 조작 대상 컴포넌트의 매뉴얼 표시

조작 대상의 컴포넌트 표시 시에, 대상 화면의 매뉴얼을 표시할 수 있습니다.

1. 컴포넌트 표시 시에 매뉴얼 아이콘을 누릅니다.



2. 조작 대상의 컴포넌트의 매뉴얼이 별도 윈도우로 열립니다.


4.3.1 Resource Collector #

This component collects system resources of Operating machine.
It collects and registers CPU utilization, memory consumption, and other resource utilization information of the device on which this application is running.

Resource Collector
resrc
Output
113 Counts / min

4.3.1.1 Basic Information

The basic information of the Resource Collector is as follows:

Item Description

Basic Information
Settings
Output Data Items
PLC Collector
Basic Information
Settings
Test Connection
Auto address setting
Modbus TCP Collector
Basic Information
Settings
Test Connection
EtherNet/IP Collector
Basic Information
Settings
Test Connection
Auto variable setting
MQTT Collector
Basic Information

AI 번역본
-For Reference Only-

5.3.3.5 플로우 링크

여기에서는, 플로우 링크의 조작에 대해 설명합니다.

> 이 조작은 Edit Mode에서만 수행할 수 있습니다.

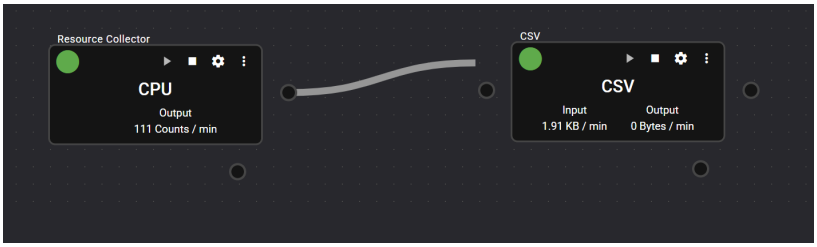
5.3.3.5.1 추가

다음 순서로, 플로우 링크를 추가할 수 있습니다.

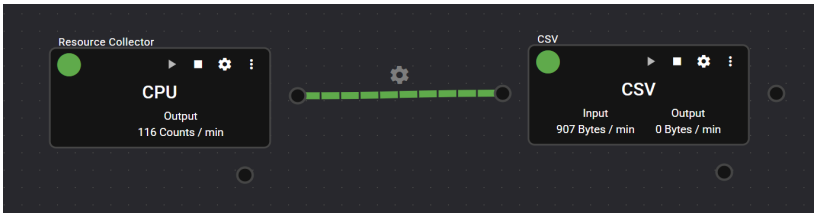
1. 컴포넌트의 출력 포트 또는 시스템 포트를 드래그합니다.



2. 다른 컴포넌트의 입력 포트에서 마우스의 버튼을 뗍니다.



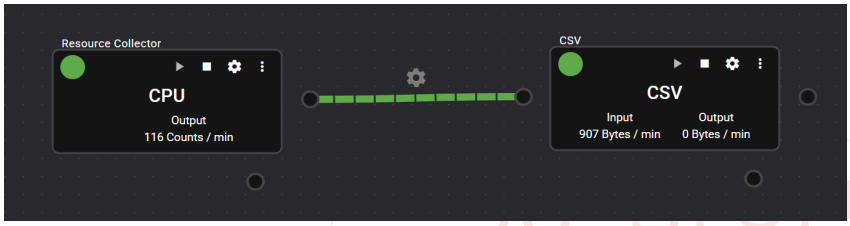
3. 컴포넌트 사이에 플로우 링크가 추가됩니다.



5.3.3.5.2 필터

플로우 링크 아이콘을 왼쪽 클릭, 또는 플로우 링크를 오른쪽 클릭하면 필터를 설정할 수 있습니다.

플로우 링크 위를 흐르는 데이터를 제한하는 설정을 수행할 수 있습니다. 상세는 「 필터 」를 참조하십시오.

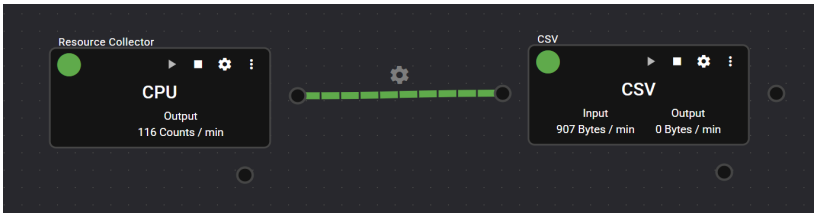


AI 번역본
-For Reference Only-

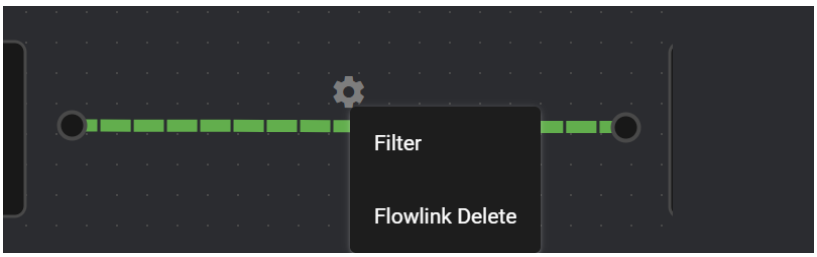
5.3.3.5.3 삭제

다음 순서로, 플로우 링크를 삭제할 수 있습니다.

1. 플로우 링크를 오른쪽 클릭합니다. 또는 아이콘을 왼쪽 클릭합니다.



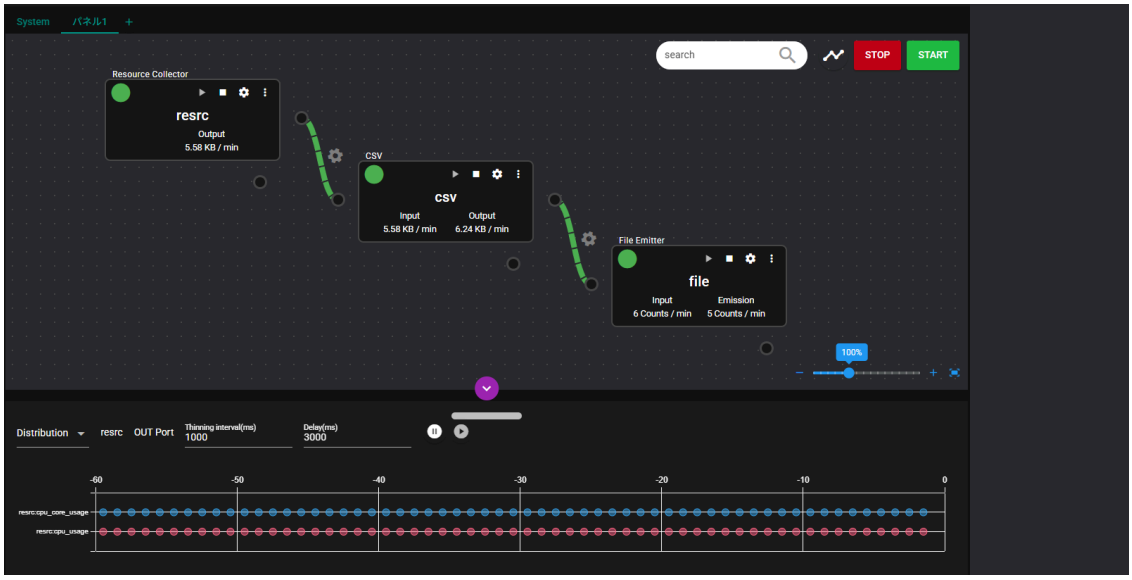
2. 「 플로우 링크 삭제 」를 선택합니다.



3. 플로우 링크가 삭제됩니다.

5.3.3.6 데이터 모니터

컴포넌트의 포트를 클릭하면, 포트에 흐르고 있는 데이터를, 데이터 모니터에서 확인할 수 있습니다.

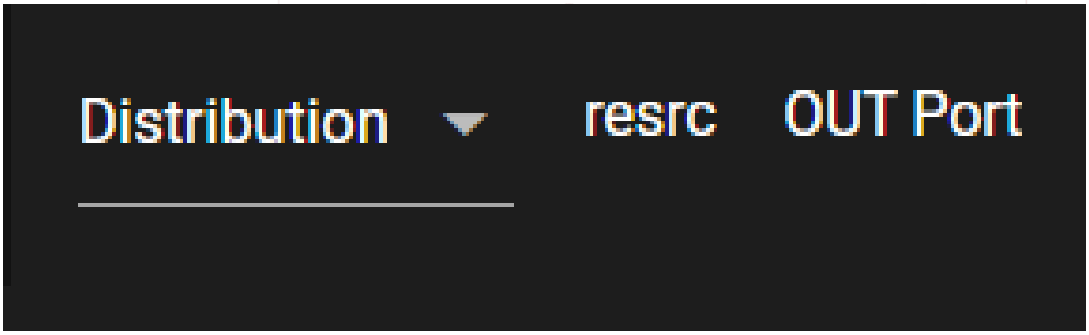


5.3.3.6.1 설정의 변경

종류의 변경

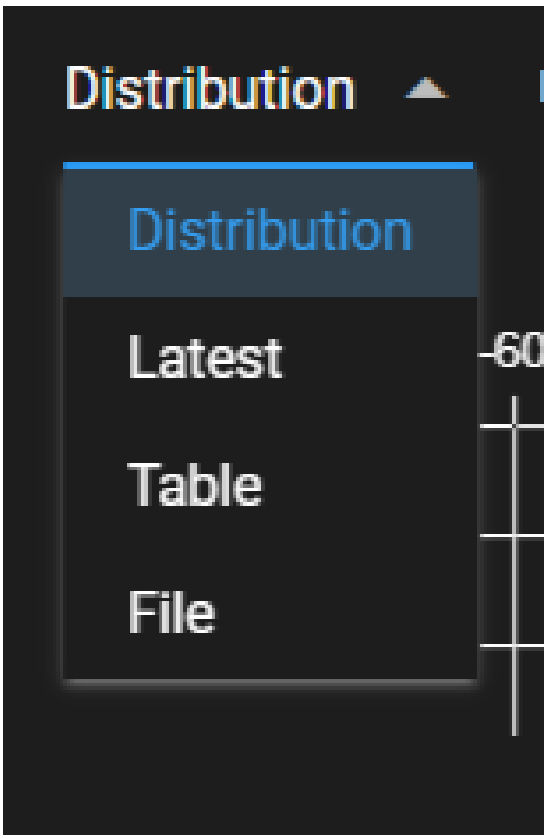
다음 순서로, 데이터 모니터의 종류를 변경할 수 있습니다.

1. 데이터 모니터의 표시 종류 선택 폼을 클릭합니다.



2. 종류를 선택합니다.

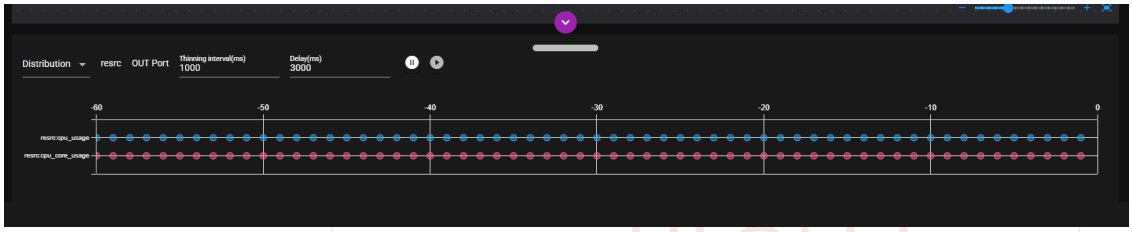
AI 번역본



3. 선택한 종류의 데이터 모니터가 표시됩니다.

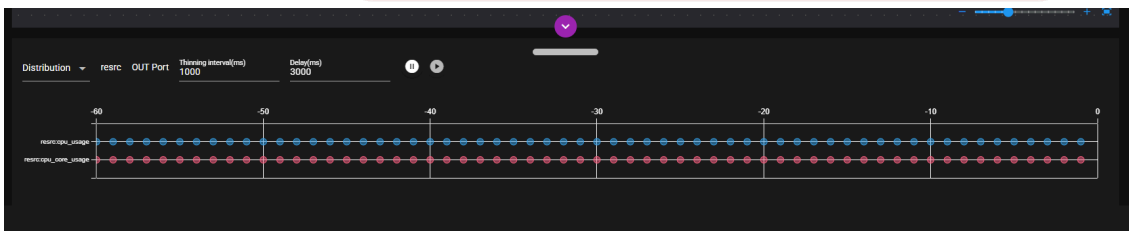
솎아내기 간격의 설정

데이터 모니터는, 일정 간격마다 데이터를 솎아내어 표시하고 있습니다.
「 솎아내기 간격(밀리초) 」에 솎아내기의 간격을 입력하면, 데이터 모니터에 반영됩니다.



지연 시간의 설정

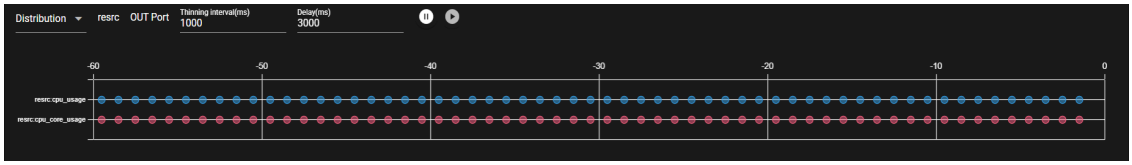
데이터 모니터에는, 현재 시각에서 「 지연 시간(밀리초) 」을 뺀 시각의 데이터가, 최신값으로서 표시됩니다.
「 지연 시간(밀리초) 」에 값을 입력하면, 데이터 모니터에 반영됩니다.



5.3.3.6.2 분포

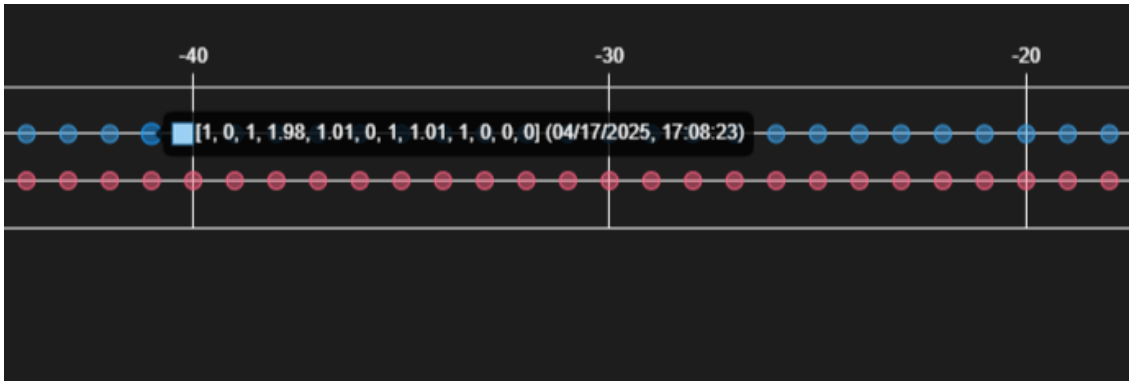
AI 번역본
-For Reference Only-

1분 사이에 포트에 흐른 데이터가, 그래프에 표시됩니다. 그래프에는 이하의 정보가 표시됩니다.



항목	설명
가로축	시각. 0은 현재 시각에서 「지연 시간(밀리초)」을 뺀 시각이고, -x는 그 시각에서 x초 전의 시각.
세로축	데이터의 항목명
점	데이터

점에 커서를 맞추면, 데이터의 내용・데이터가 흐른 시각이 표시됩니다.

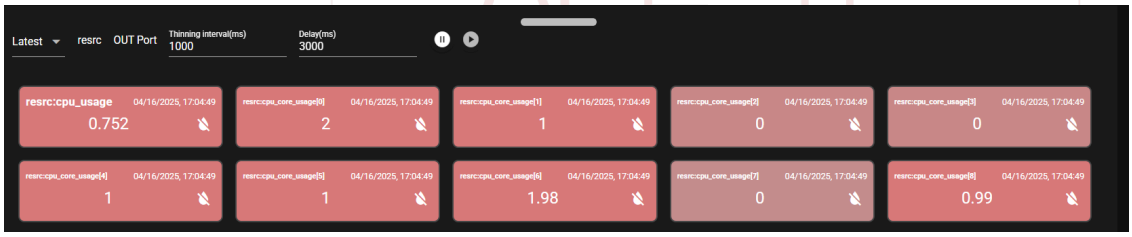


5.3.3.6.3 최신값

포트 위의 모든 데이터 항목에 있어서, 이하의 정보가 표시됩니다.

- 최신값
- 최신값이 흐른 시간

AI 번역본

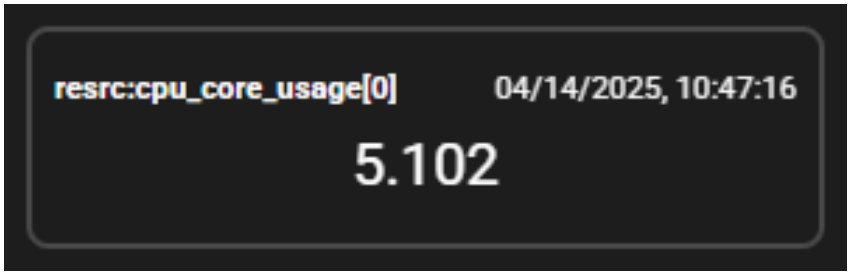


데이터의 값이 변화한 것은, 적색으로 표시됩니다.



클리어 버튼을 누르면, 적색으로 변화한 데이터 항목이, 원래의 색으로 돌아갑니다.

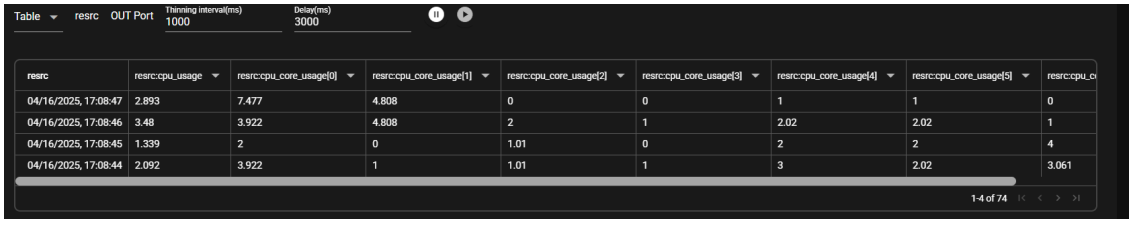




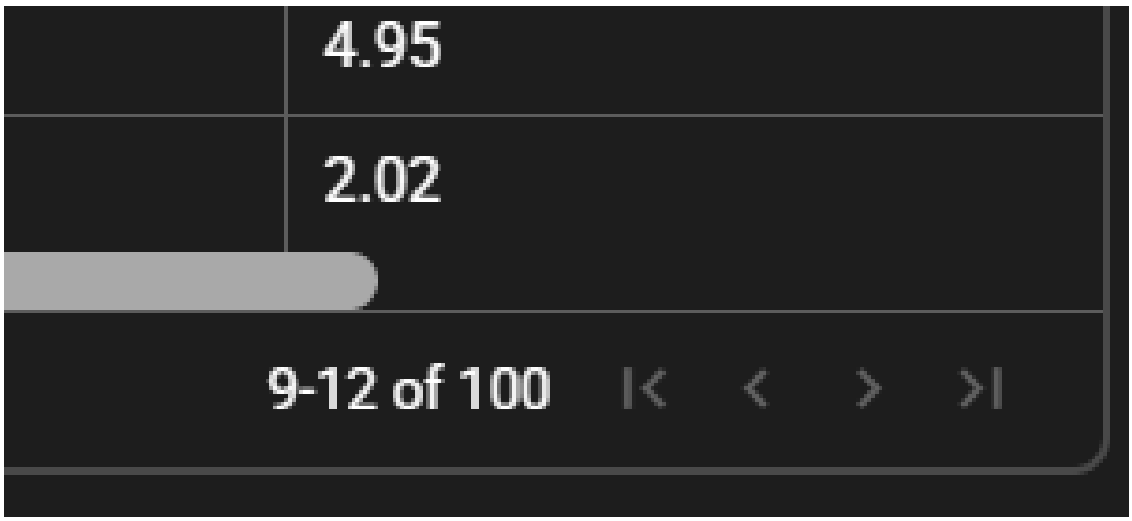
5.3.3.6.4 테이블

포트에 흐른 데이터가, 테이블 형식으로 표시됩니다. 테이블에는 이하의 정보가 표시됩니다. 1페이지마다 표시할 테이블의 행수는, 데이터 모니터의 높이에 따라 자동으로 조정됩니다.

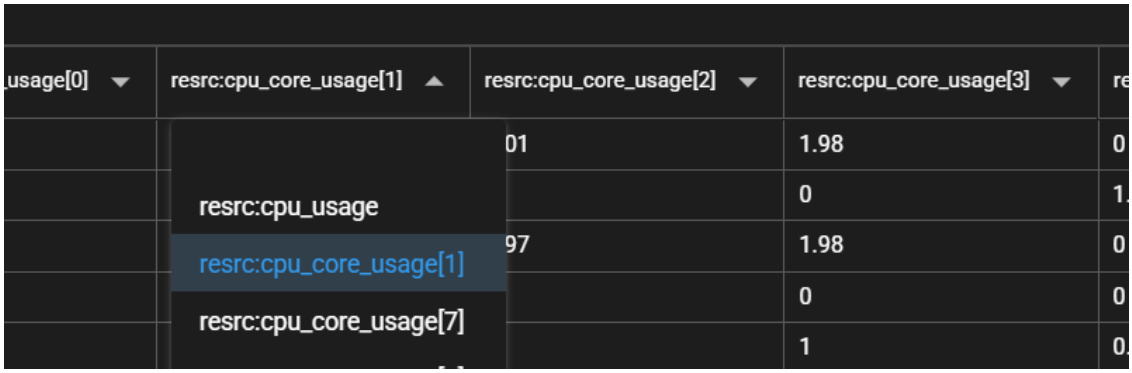
항목	설명
열	데이터의 항목명. 열수는 최대 8.
행	데이터가 흐른 시각. 행수는 최대 100.
내용	데이터



테이블의 행수가 1페이지에 담기지 않는 경우, 페이지를 전환할 수 있습니다.



포트 위의 데이터 항목 수가 8을 넘는 경우, 테이블의 열의 표제를 선택하면, 그 열에 표시할 데이터 항목을 변경할 수 있습니다. 공백을 선택하면, 그 열은 비표시가 됩니다.



5.3.3.6.5 파일

데이터가 파일 형식인 경우, 이하의 파일의 정보가 표시됩니다.

- 파일 사이즈...파일의 사이즈.
- 미디어 타입...파일의 미디어 타입.
- 타임스탬프...파일의 내용에 들어 있는 데이터의 시각 범위 파일의 내용
-



5.3.3.7 시스템 리소스

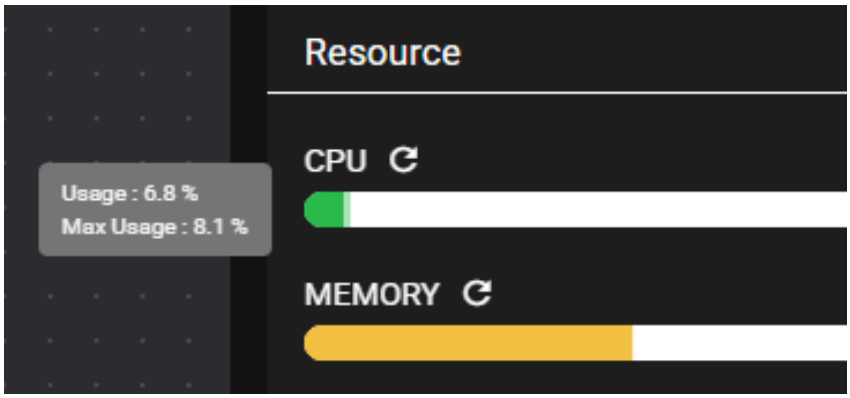
여기에서는, 시스템 리소스에 대해 설명합니다.

AI 번역본

이 조작은 Run Mode에서만 수행할 수 있습니다.

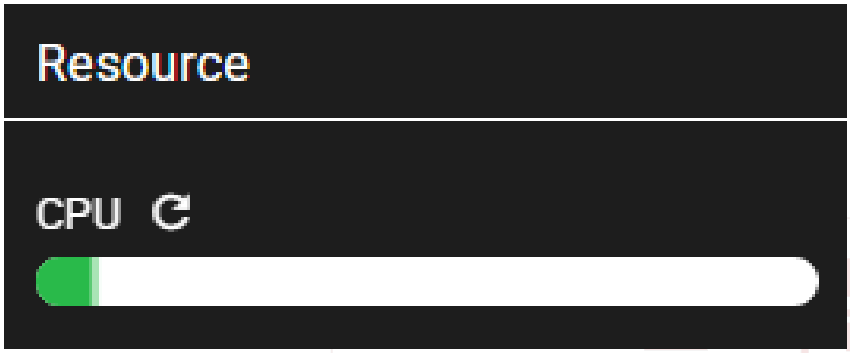
5.3.3.7.1 CPU

게이지에 커서를 맞추면, 이하의 정보가 표시됩니다.



항목	설명
사용률	SpeeDBee Synapse의 현재의 CPU 사용률
최대 사용률	SpeeDBee Synapse의 CPU 사용률의 최대값

게이지의 진한 색의 부분이 현재의 CPU 사용률, 옅은 색의 부분이 동작 중에 도달한 최대 CPU 사용률을 나타냅니다.

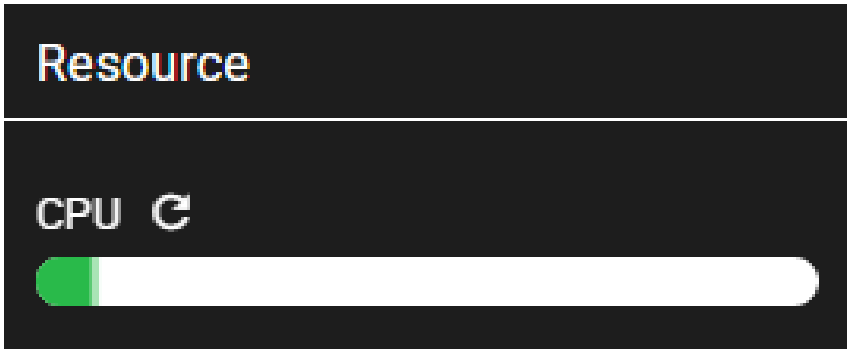


게이지의 색은, CPU 사용률의 퍼센티지에 맞춰, 다음 표와 같이 변화합니다.

퍼센티지	색
0~40	녹색
40.1~70	황색
70.1~100	적색

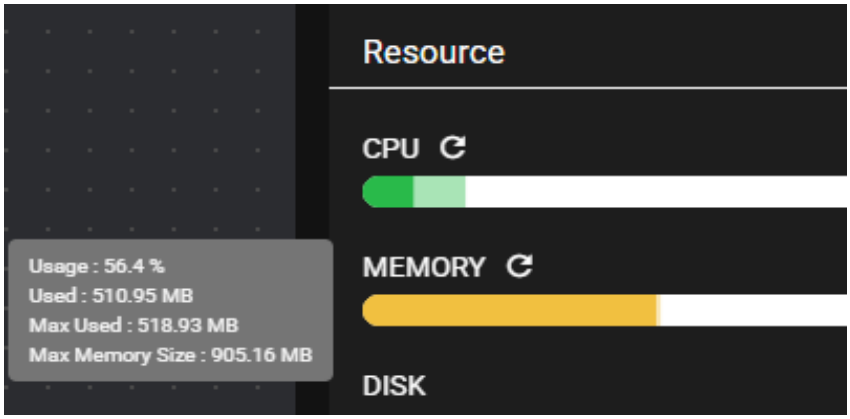
For Reference Only-

리프레시 아이콘을 누르면, 최대 사용률이 리셋됩니다.



5.3.3.7.2 MEMORY

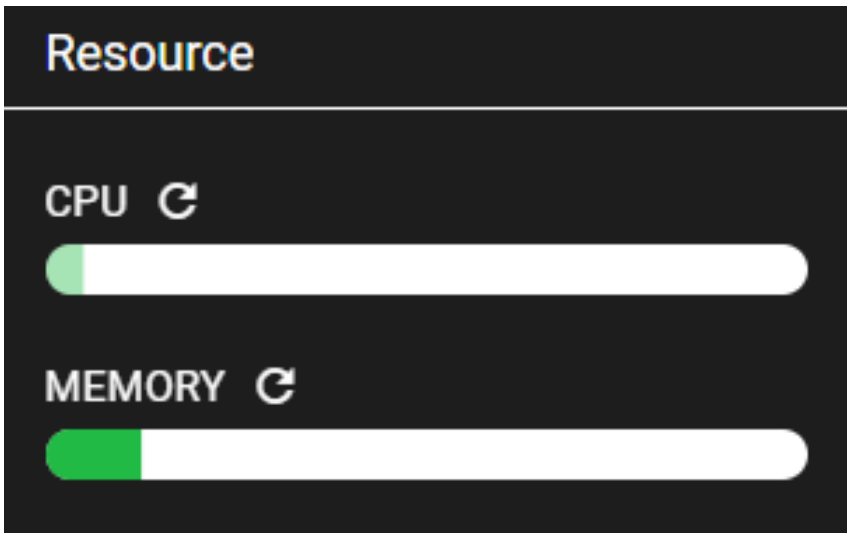
게이지에 커서를 맞추면, 이하의 정보가 표시됩니다.



항목	설명
사용률	SpeeDBee Synapse의 현재의 메모리 사용률
사용량	SpeeDBee Synapse의 현재의 메모리 사용량
최대 사용량	SpeeDBee Synapse의 메모리 사용률의 최대값
최대 메모리 사이즈	시스템 전체의 메모리 사이즈

-For Reference Only-

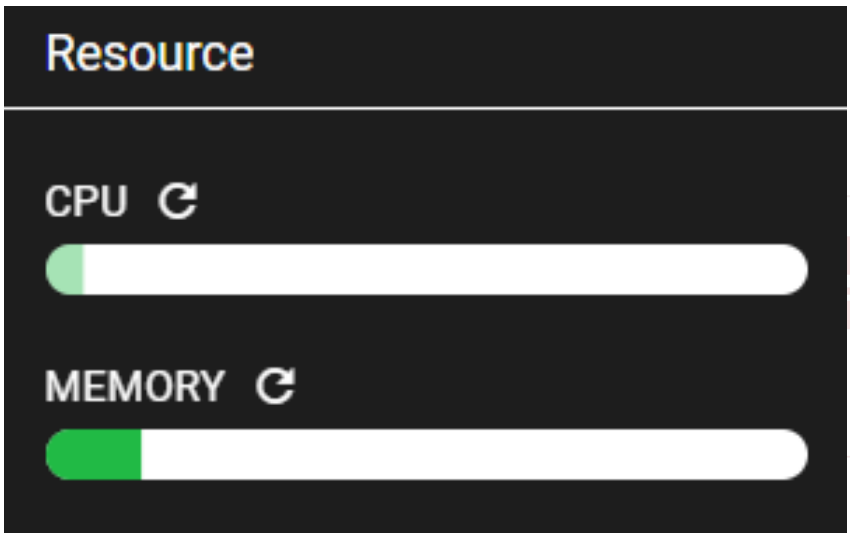
게이지의 색이 있는 부분이, 메모리 사용률을 나타냅니다.



게이지의 색은, 메모리 사용률의 퍼센티지에 맞춰, 다음 표와 같이 변화합니다.

퍼센티지	색
0~40	녹색
40.1~70	황색
70.1~100	적색

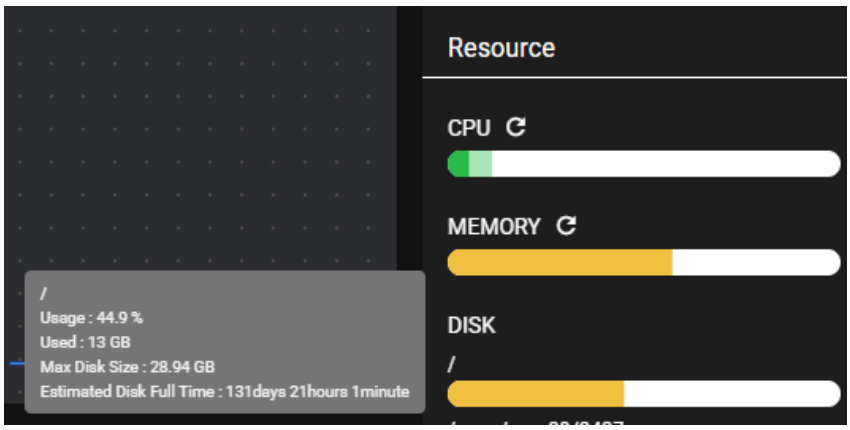
리프레시 아이콘을 누르면, 최대 사용량이 리셋됩니다.



본
Only-

5.3.3.7.3 DISK

게이지에 커서를 맞추면, 이하의 정보가 표시됩니다.



항목	설명
사용률	최대 디스크 용량에 대한, SpeeDBee Synapse의 현재의 디스크 사용의 비율
사용량	SpeeDBee Synapse의 현재의 디스크 사용량
최대 디스크 용량	SpeeDBee Synapse가 사용할 수 있는 디스크의 최대 용량
디스크 포화 예상 시간	1분간의 디스크 사용량의 증가를 바탕으로 산출된, 디스크 사용량이 디스크 최대 용량에 도달할 때까지의 예상 시간 ※디스크 사용량의 증가가 없는 경우는 표시되지 않음

게이지의 색이 있는 부분이, 디스크 사용률을 나타냅니다.



AI 번역본

게이지의 색은, 디스크 사용률의 퍼센티지에 맞춰, 다음 표와 같이 변화합니다.

퍼센티지	색
0~40	녹색
40.1~70	황색
70.1~100	적색

5.3.3.8 시뮬레이션 모드

시뮬레이션 모드에 대해 설명합니다.

5.3.3.8.1 시뮬레이션 모드의 개요

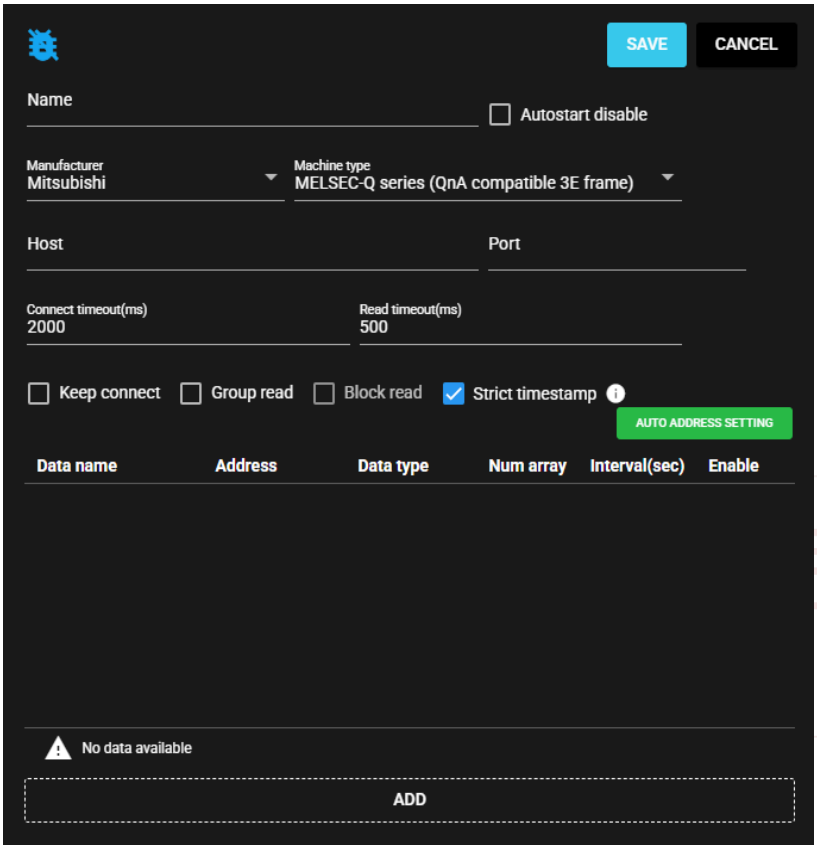
시뮬레이션 모드는, CSV 파일의 내용을 바탕으로 컬럼을 생성하고, 더미 데이터를 등록하는 기능입니다. 본 기능에 의해, 실제의 대향 기기（PLC 등）가 없는 환경에서도 설정이나 동작 검증이 가능해집니다. 시뮬레이션 모드에 대응하고 있는 컴포넌트는 이하와 같습니다.

- Modbus TCP 컬렉터
- PLC 컬렉터
- EtherNet/IP 컬렉터
- MQTT 컬렉터

5.3.3.8.2 시뮬레이션 모드의 설정・전환

여기에서는, PLC 컬렉터를 사용하여 설명합니다.

1. 컴포넌트 설정 화면의 좌측 상단의 시뮬레이션 설정 버튼을 클릭합니다.



2. 시뮬레이션 설정 화면이 열리므로, 설정 항목을 입력하고「닫기」를 클릭합니다.

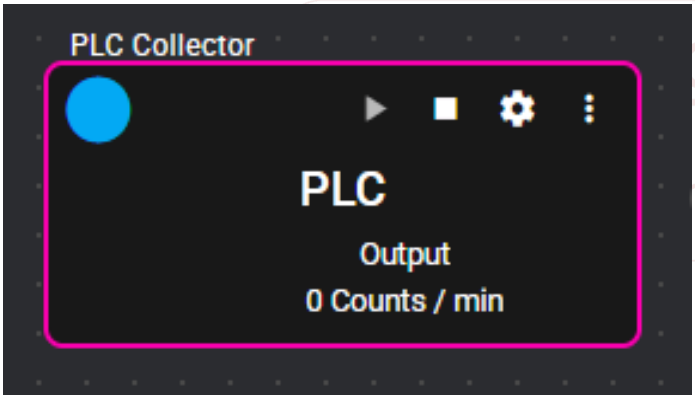


항목	설명
시뮬레이션모드	시뮬레이션 모드의 유효・무효
운전 파일(csv)	시뮬레이션 모드에서 사용할 컬럼이나 데이터를 정의한 CSV 형식의 파일. 상세는, 운전 파일의 포맷을 참조하십시오.

운전 파일은, 한 번 업로드하면, 삭제할 때까지는 그 설정이 유지됩니다.



3. 컴포넌트 설정 화면으로 돌아가므로, 컴포넌트의 설정 항목을 입력하고 「 저장 」을 클릭합니다. 컴포넌트 설정 화면에서 「 저장 」을 클릭할 때까지는, 시뮬레이션 설정의 변경은 반영되지 않습니다.
4. 기동 버튼을 클릭하여 기동합니다. 시뮬레이션 모드 ON의 상태로 기동하고 있는 경우, 컴포넌트의 스테이터스 램프가 청색이 됩니다.



5.3.3.8.3 운전 파일의 포맷

운전 파일은, CSV 형식으로 되어 있습니다. 헤더부（1행째）와 데이터부（2행째 이후）가 있고, 헤더부에서 컬럼 정의, 데이터부에서 타임스탬프와 등록할 데이터를 정의합니다.

example.csv

```
timestamp_ms,data1:bool,data2:int32,data3:string,data4:binary
        1000,      true,          1,       test1,    YWl1ZW8=
        2000,     false,          2,       test2,    YWl1ZW8=
        3000,      true,          3,       test3,    YWl1ZW8=
```

헤더부

```
timestamp_ms,data1:bool,data2:int32,data3:string,data4:binary
```

- 헤더부의 1열째는 timestamp_ms 고정, 2열째 이후에 컬럼 정의를 입력합니다.
- 컬럼 정의는, 컬럼명:데이터형과 같이, : 구분으로 컬럼명과 데이터형을 지정합니다. 배열인 경우는,
- 컬럼명:데이터형[배열 수] 와 같이, [] 로 배열 수를 감싸서 지정합니다.

데이터부

```
        1000,      true,          1,       test1,    YWl1ZW8=
        2000,     false,          2,       test2,    YWl1ZW8=
        3000,      true,          3,       test3,    YWl1ZW8=
```

- 데이터부의 1열째는 timestamp_ms의 값, 2열째 이후에 실제로 등록할 데이터의 값을 입력합니다.
- 기동하고 나서 timestamp_ms의 값 경과 후의 시간에 데이터를 등록합니다. 상기의 CSV의 예라면 기동부터 1000밀리초, 2000밀리초, 3000밀리초 후의 시간에 데이터를 등록합니다.
- timestamp_ms가 전회값보다도 작은（시각이 되돌아가는）경우, 그 행의 데이터는 등록되지 않습니다.
- 파일의 종단에 도달하면, 선두의 데이터로 돌아가, 다시 선두부터 데이터가 등록됩니다.
- 데이터의 값이 공백인 경우, 데이터는 등록되지 않습니다.
- 배열 데이터는, 스페이스 구분으로 지정합니다.

지정 가능한 데이터형	비고
bool	true, false, TRUE, FALSE로 지정.
int8	int8의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
int16	int16의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
int32	int32의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
int64	int64의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
uint8	uint8의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.

AI 번역본
-For Reference Only-

지정 가능한 데이터형	비고
uint16	uint16의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
uint32	uint32의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
uint64	uint64의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
float	float의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
double	double의 범위로 지정. 프리픽스에 0x를 붙임으로써 Hex 지정 가능.
string	CSV에 있어서 특수 문자(, \n 등)를 지정할 때는" 로 감쌉니다. " 의 이스케이프는" .
binary	Base64 형식으로 지정.

운전 파일의 샘플은 이하와 같습니다.

- 전 데이터형 샘플
- 배열 데이터 샘플

5.3.3.8.4 시뮬레이션 모드의 동작

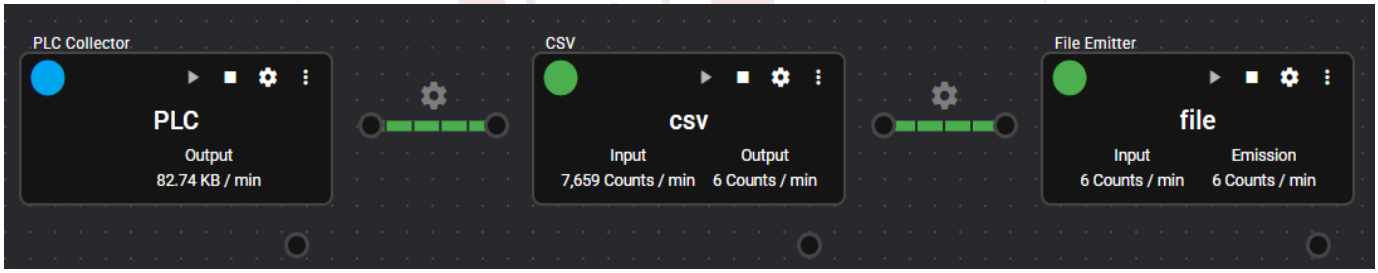
시뮬레이션 모드에서 동작 중은, 더미 데이터의 등록만을 수행합니다. 실제의 기기와의 통신에 의한 데이터 수집 은 수행되지 않습니다. 또한, 시뮬레이션 모드 동작 중의 컴포넌트 스테이터스에 대해서, 스테이터스 램프가 청색으로 표시됩니다만, 상태 자체는 통상의 Running 과 같습니다.

AI 번역본
-For Reference Only-

- 등록된 더미 데이터는, 통상의 컴포넌트와 마찬가지로 데이터 모니터링으로 확인할 수 있습니다.



- 플로우 링크를 연결하여 다른 컴포넌트로 데이터를 전달하는 것도 가능합니다.



시뮬레이션 모드는, 개발 중인 데이터 플로우의 검증에는 문제없습니다만, 실제 운용 환경에서 운용할 때는, OFF로 하는 것을 잊지 않도록 주의하십시오.

5.3.3.9 태블릿 단말에서의 조작 방법

SpeeDBee Synapse는, 태블릿 단말에서의 조작에 대응하고 있습니다. 검증 완료 단말은, 다음과 같습니다.

기종명/모델	OS 버전	브라우저
iPad(A16)	iOS 18.5	Safari 605.1.15
HP Spectre x360 Convertible 14-ea0xxx	Windows 11 24H2	Google Chrome 136.0.7103.114

태블릿 단말에서의 동작은, PC의 마우스 조작과 일부 다릅니다. 이하에 주요한 차이를 정리하므로, 참조하십시오.

기능・조작	PC에서의 조작 방법	태블릿에서의 조작 방법
컨텍스트 메뉴	오른쪽 클릭	대상 요소를 롱탭(길게 누르기)합니다
드래그&드롭	마우스에서의 드래그	손가락으로 대상을 탭하고, 그대로 이동함으로써 드래그 조작을 수행합니다
호버 조작	마우스 오버로 표시	터치 디바이스에서는 호버 조작은 수행할 수 없습니다만, 일부의 조작은 탭이나 롱탭으로 대용되고 있습니다.

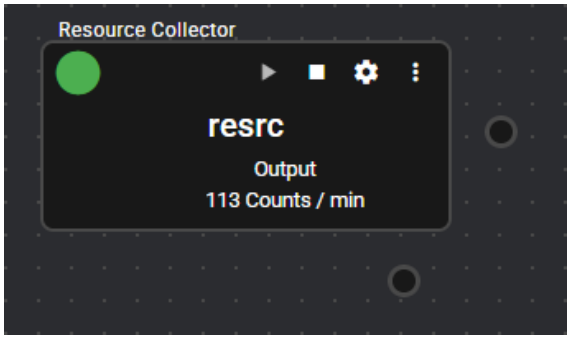
AI 번역본
-For Reference Only-

5.4 컬렉터

여기에서는, 각 컬렉터 컴포넌트에 대해 설명합니다.

5.4.1 리소스 컬렉터

가동 머신의 시스템 리소스를 수집하는 컴포넌트입니다.
본 앱이 가동하고 있는 기기의 CPU 사용률, 메모리 소비 등의 리소스의 사용 상황을 수집하여 등록합니다.



5.4.1.1 기본 정보

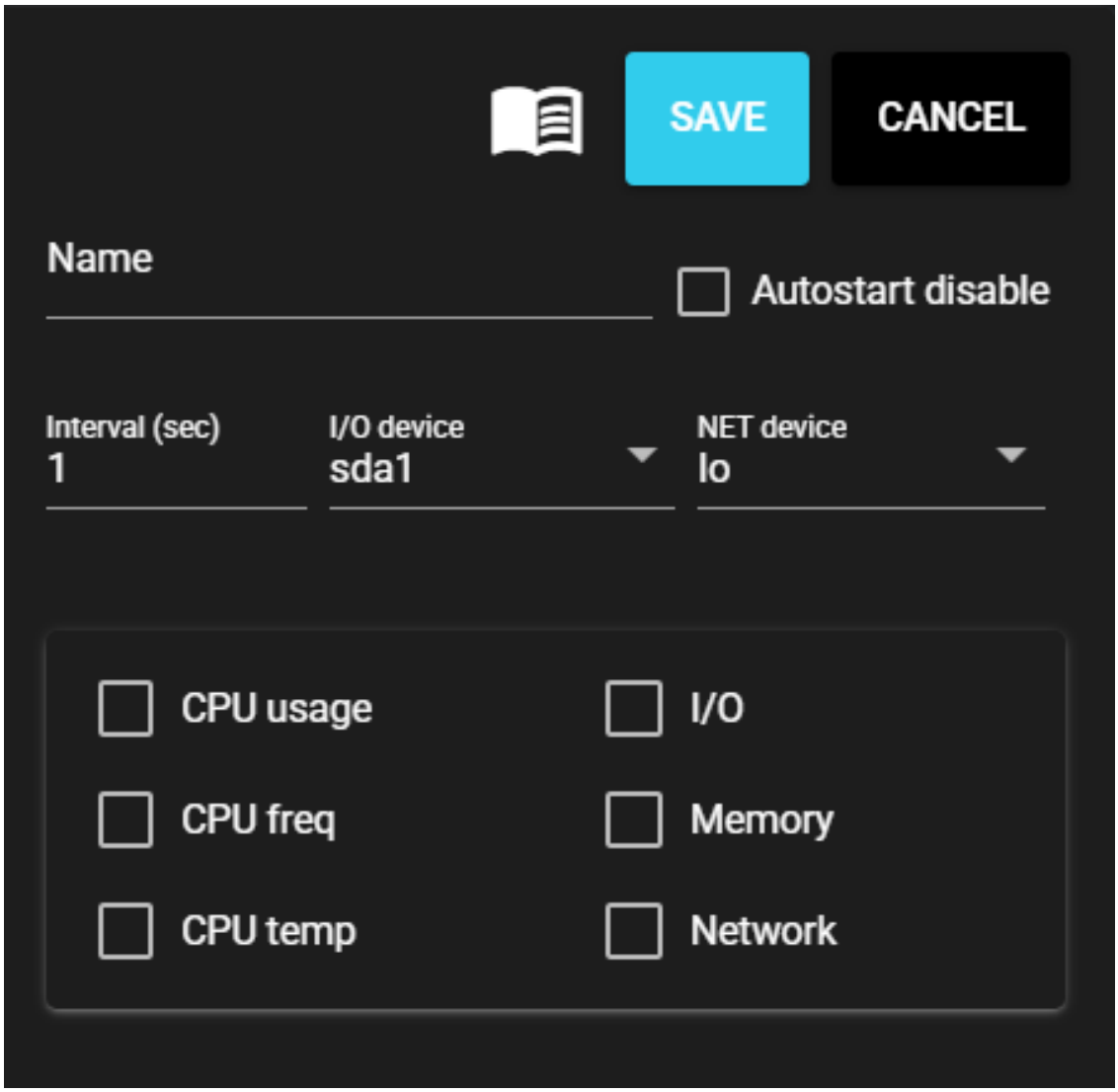
리소스 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	리소스 컬렉터
설명	가동 머신의 시스템 리소스를 수집합니다.
소비 포인트	0
포트	입력 : 0 출력 : 1 수집한 가동 머신의 시스템 리소스를 출력합니다. 출력할 데이터의 상세는,「출력 데이터 항목」을 참조하십시오. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.

AI 번역본
-For Reference Only-

5.4.1.2 설정 항목

리소스 컬렉터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
취득 주기(초)	컬렉터가 데이터를 취득하는 주기를 입력
I/O 디바이스	데이터를 수집할 I/O 디바이스를 선택
NET 디바이스	데이터를 수집할 네트워크 디바이스를 선택

AI 번역본
-For Reference Only-

항목	설명
CPU 사용률	가동 머신의 CPU 사용률을 수집하는 경우 ON
CPU 클록 수	가동 머신의 CPU 클록 수를 수집하는 경우 ON ※Windows에서는 사용할 수 없습니다 ※Linux에서도 사용할 수 없는 경우가 있습니다
CPU 온도	CPU 온도를 수집하는 경우 ON ※Windows에서는 사용할 수 없습니다 ※Linux에서도 사용할 수 없는 경우가 있습니다
I/O	「I/O 디바이스」에서 선택한 디바이스의 I/O의 상태를 수집하는 경우 ON ※Windows판에 있어서, I/O의 수집을 유효로 하면, 취득 주기 1초에서는, 처리가 따라가지 못하는 경우가 있습니다（지연되는 경우, 주기를 5초 이상으로 설정하여 조정하십시오）
메모리	가동 머신의 메모리 사용량을 수집하는 경우 ON
네트워크	「NET 디바이스」에서 선택한 디바이스의 네트워크의 상태를 수집하는 경우 ON

5.4.1.3 출력 데이터 항목

리소스 컬렉터는, 다음의 데이터를 출력합니다.

항목	컬럼명	데이터형	설명
CPU 사용률	cpu_usage	double	CPU 사용률을 수집한다. 단위는 %, 범위는 0.000～100.000
CPU 코어마다의 사용률	cpu_core_usage	double[n]	CPU 코어마다의 사용률을 수집한다. 단위는 %, 범위는 0.000 ~ 100.000
CPU 실행 주파수	cpu_core_freq	int32[n]	CPU 코어마다의 실행 주파수를 수집한다. 단위는 kHz
CPU 온도	cpu_temp	double	CPU 온도를 수집한다. 단위는 ℃
IO 정보 수집 디바이스명	I_O	string	IO 정보 수집 디바이스명(물리만)

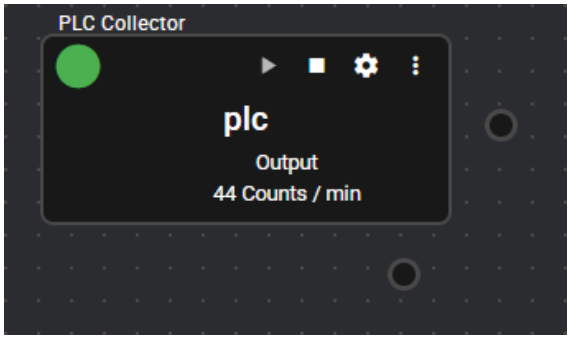
AI 번역본
For Reference Only

항목	컬럼명	데이터형	설명
평균 읽기 요청 처리 수	kb_r_s	double	평균 읽기 요청 처리 수를 수집한다. 단위는 KB
평균 쓰기 요청 처리 수	kb_w_s	double	평균 쓰기 요청 처리 수를 수집한다. 단위는 KB
평균 읽기 · 쓰기 횟수	IOPS	double	평균 읽기 · 쓰기 횟수를 수집한다. 단위는 KB
디스크 여유 용량 (MB)	storage_free_space_mb	double	지정된 디스크의 여유 용량을 수집한다. 단위는 MB.
디스크 사용률(%)	storage_usage	double	지정된 디스크의 사용률을 수집. 단위는 %, 범위는 0.0~100.0
메모리 총 용량	mem_total	int32	메모리 총 용량을 수집한다. 단위는 KB
메모리 사용량	mem_used	int32	메모리 사용량을 수집한다. 단위는 KB
메모리 사용률	mem_usage	double	메모리 사용률을 수집한다. 단위는 %, 범위는 0.0~100.0
SWAP 총 용량	swap_total	int32	SWAP 총 용량을 수집한다. 단위는 KB
SWAP 사용량	swap_used	int32	SWAP 사용량을 수집한다. 단위는 KB
SWAP 사용률	swap_usage	double	SWAP 사용률을 수집한다. 단위는 %, 범위는 0.0~100.0
합계 수신 바이트 수	rx_bytes1	uint64	수신 바이트 수를 수집한다. 단위는 bytes
합계 수신 패킷 수	rx_pckt1	uint64	수신 패킷 수를 수집한다.
합계 송신 바이트 수	tx_bytes1	uint64	송신 바이트 수를 수집한다. 단위는 bytes
합계 송신 패킷 수	tx_pckt1	uint64	송신 패킷 수를 수집한다.

AI 번역본
-For Reference Only-

5.4.2 PLC 컬렉터

PLC의 레지스터/변수 정보에서 데이터 수집을 수행하는 컴포넌트입니다.
오므론이나 미쓰비시 등 다양한 제조사의 PLC와 접속하여, 레지스터/변수 정보에서 데이터를 수집, 등록합니다. 일부의 PLC에서는, 래더 편집 툴로 출력한 파일을 임포트하여 레지스터 정보를 설정할 수 있습니다.



5.4.2.1 기본 정보

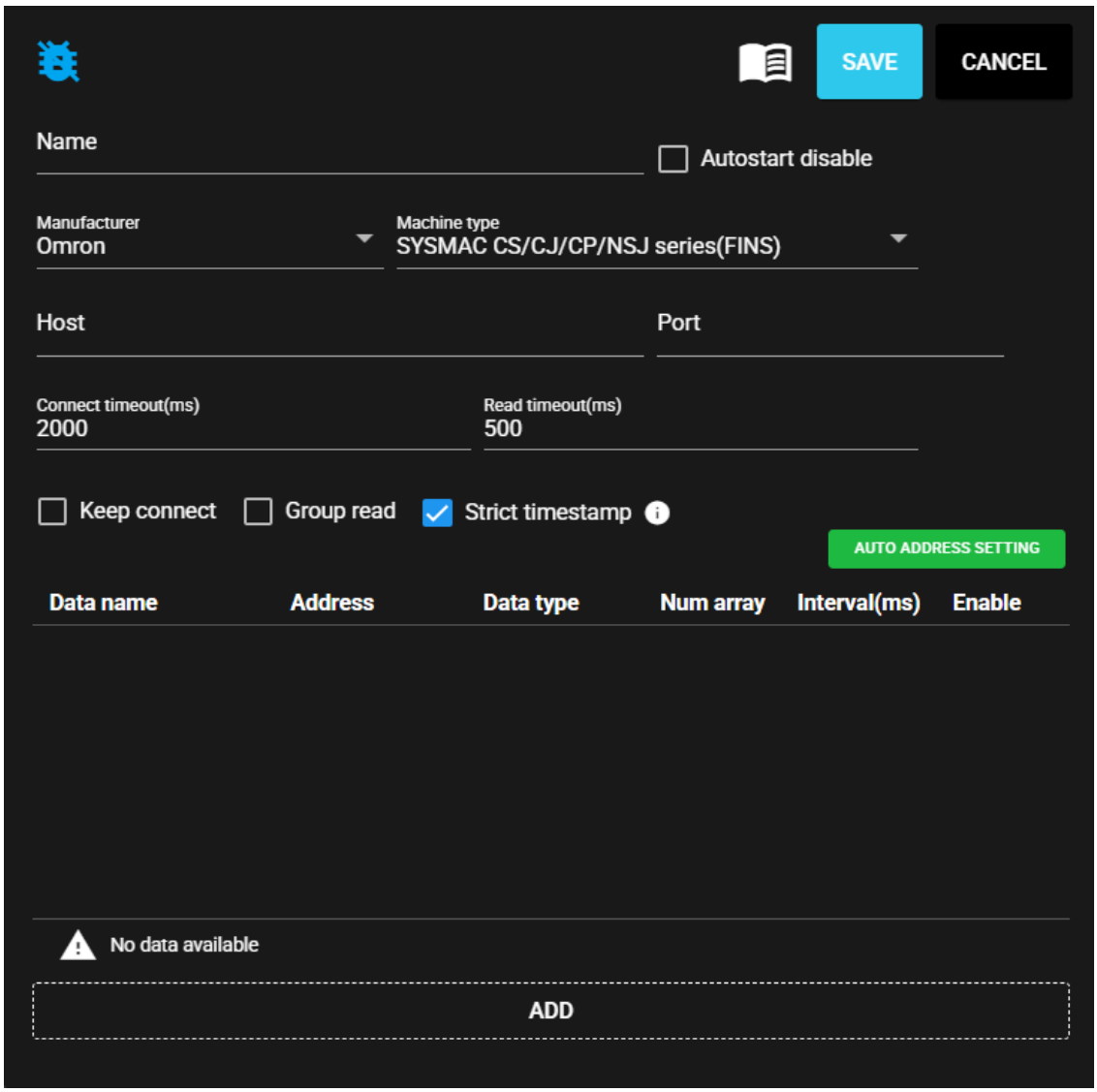
PLC 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	PLC 컬렉터
설명	PLC의 레지스터/변수 정보에서 데이터를 수집합니다.
소비 포인트	0
포트	입력 : 0 출력 : 1 「 설정 항목 」에 따라, PLC에서 수집한 데이터를 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	・PLC에 대한 테스트 접속을 수행할 수 있습니다. 상세는 「 테스트 접속 」을 참조하십시오. ・주소를 자동으로 설정할 수 있습니다. 상세는 「 주소 자동 설정 」을 참조하십시오.

AI 번역본
-For Reference Only-

5.4.2.2 설정 항목

PLC 컬렉터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
제조사	접속할 PLC의 제조사를 선택 ※선택할 수 있는 제조사는 다음과 같습니다. ・미쓰비시전기 ・제이텍트 ・오므론 ・키엔스 ・파나소닉

AI 번역본
-For Reference Only-

항목	설명
기종	접속할 PLC의 기종을 선택 ※선택할 수 있는 기종은 제조사에 따라 다릅니다.
호스트	접속할 PLC의 호스트명 또는 IP 주소를 입력
포트	접속할 PLC의 포트를 입력
접속 타임아웃(밀리초)	접속 시의 타임아웃 시간을 입력
읽기 타임아웃(밀리초)	읽기 시의 타임아웃 시간을 입력
상시 접속한다	PLC 접속을 조회 때마다, 절단하지 않는 경우 ON
일괄 취득	한 번에 복수의 워드 레지스터의 불러오기를 수행하는 경우 ON
일괄 블록 취득	한 번에 복수의 비트 레지스터・워드 레지스터의 불러오기를 수행하는 경우 ON
엄밀한 타임스탬프	수집한 데이터에 보다 엄밀한 타임스탬프를 부여하는 경우 ON

- PLC 컬렉터의 출력 데이터는, 다음의 설정 항목에 의해 정의됩니다.

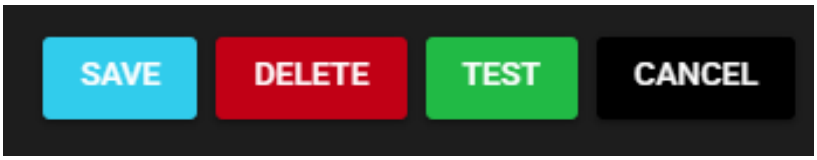
항목	설명
데이터명	데이터의 이름을 입력
주소	PLC의 레지스터 주소를 입력
프로그램	레지스터의 프로그램 번호를 선택 ※제조사가 제이텍트인 경우에만 선택 가능
데이터형	취득할 레지스터 값의 데이터형을 선택
연속 영역 수	입력한 주소를 선두로 하여 연속하는 레지스터 중에서, 값 취득의 대상으로 할 범위를 입력
수집 간격(밀리초)	레지스터 값을 취득하는 간격을 입력
유효	레지스터 값을 취득하는 경우 ON

AI 번역본
-For Reference Only-

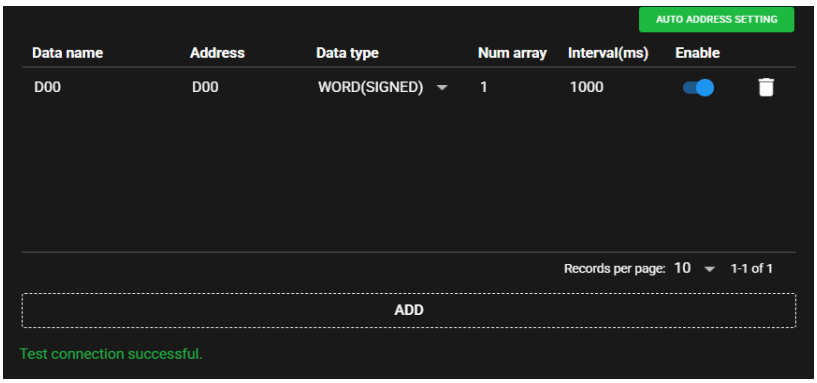
5.4.2.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 PLC 컬렉터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.

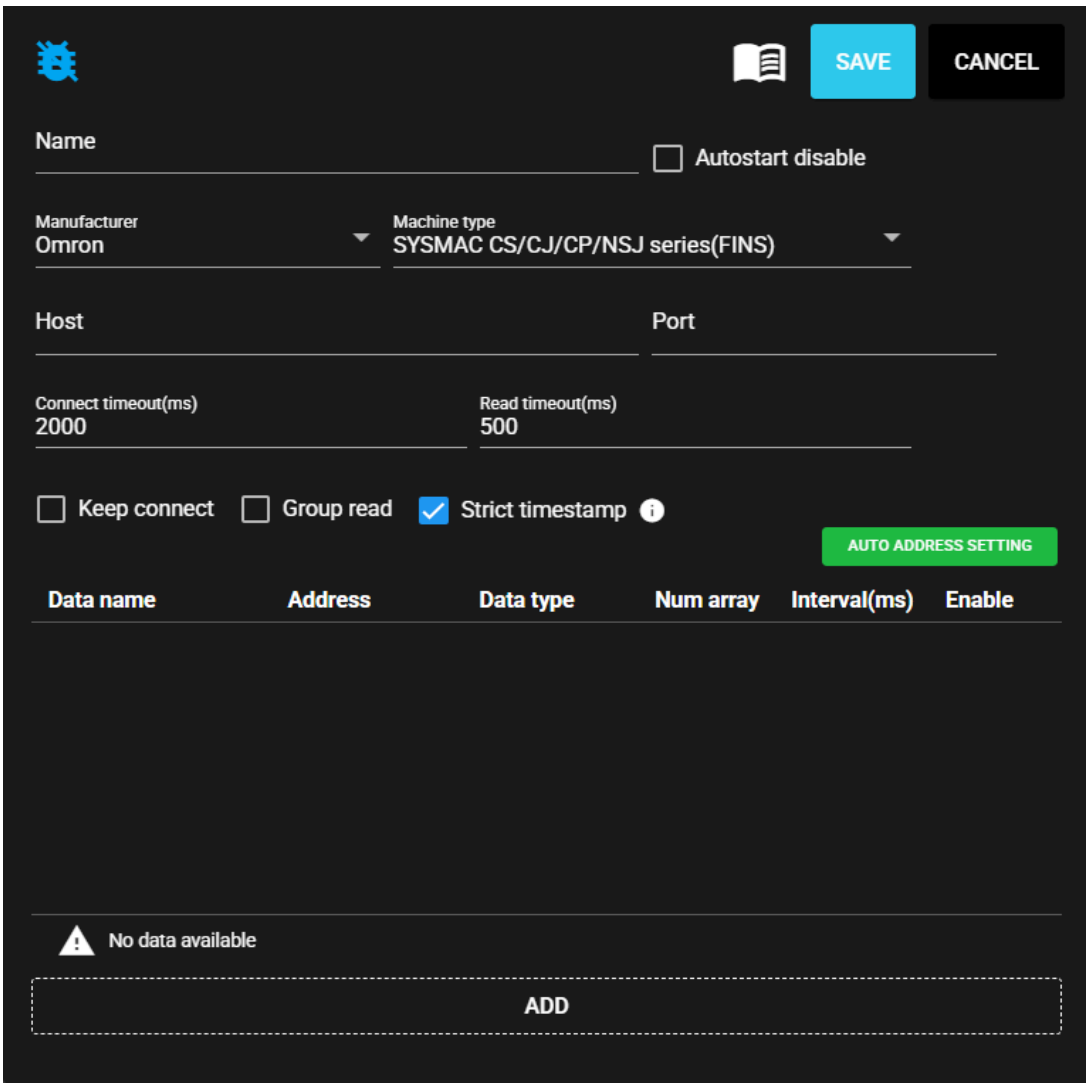


5.4.2.4 주소 자동 설정

일부의 PLC에서는, 래더 편집 툴로 출력한 파일을 읽어들임으로써, 주소를 자동으로 일괄 설정하는 것이 가능합니다.

1. 래더 편집 툴에서, 작성한 래더를 파일로서 출력합니다. 출력할 파일의 형식은, PLC의 제조사・기종에 따라 다릅니다. 파일의 샘플은 다음과 같습니다.
   - 미쓰비시전기 MELSEC-Q 시리즈(QnA 호환 3E 프레임), MELSEC iQ-R 시리즈(QnA 호환 3E 프레임)
     샘플 파일의 다운로드
   - 오므론 SYSMAC CS/CJ/CP/NSJ 시리즈(FINS)
     샘플 파일의 다운로드
2. 「 주소 자동 설정 」을 누릅니다.

AI 번역본
-For Reference Only-



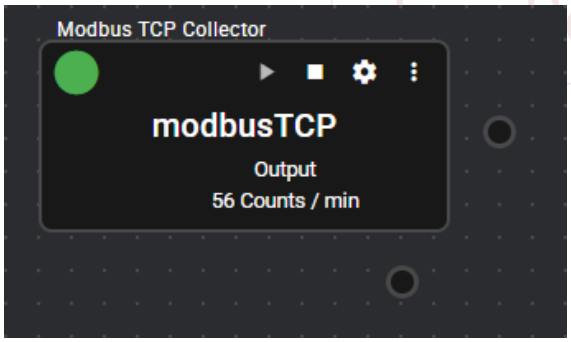
3. 래더 편집 툴로 출력한 파일을 선택합니다.

4. 파일을 바탕으로, 주소가 자동으로 설정됩니다.

5.4.3 Modbus TCP 컬렉터

AI 번역본
For Reference Only

Modbus TCP 대응 기기에서 데이터 수집을 수행하는 컴포넌트입니다.
본 컴포넌트가 Modbus의 마스터가 되어, Modbus의 슬레이브 기기에서 데이터를 수집, 등록합니다.



5.4.3.1 기본 정보

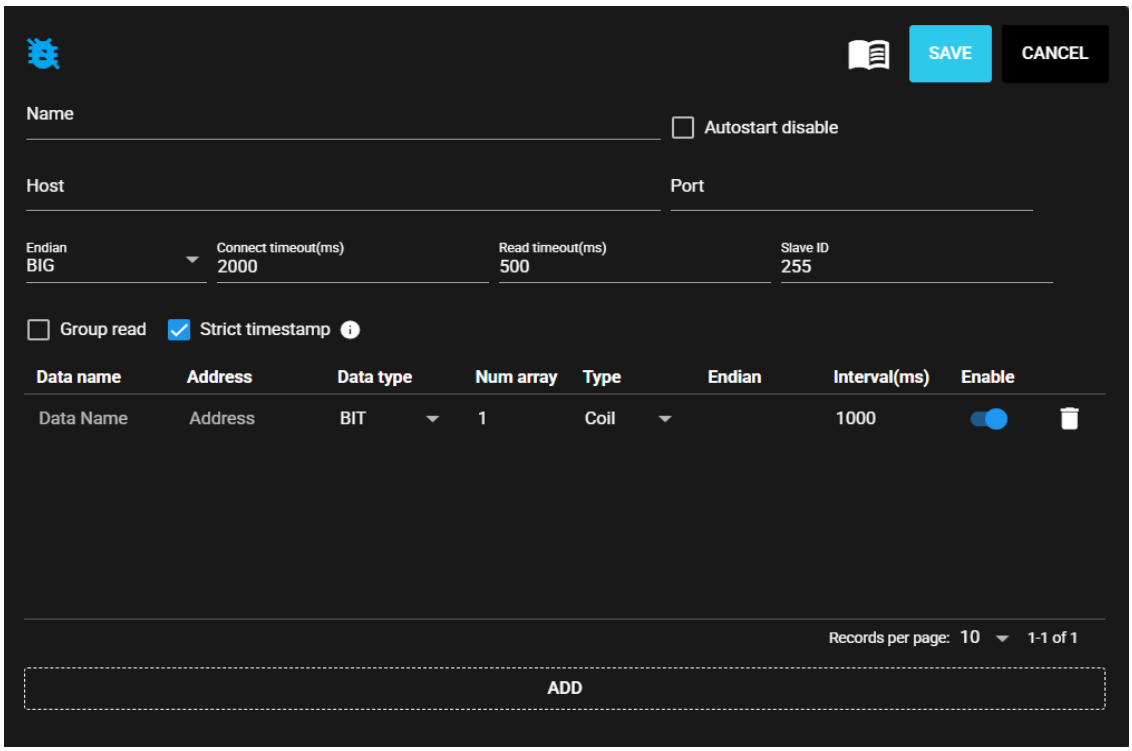
Modbus TCP 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	Modbus TCP 컬렉터
설명	Modbus TCP 대응 기기에서 데이터를 수집합니다.
소비 포인트	2
포트	입력 : 0 출력 : 1 「 설정 항목 」에 따라, Modbus TCP 대응 기기에서 수집한 데이터를 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	Modbus TCP 대응 기기에 대한 테스트 접속을 수행할 수 있습니다. 상세는 「 테스트 접속 」을 참조하십시오.

5.4.3.2 설정 항목

Modbus TCP 컬렉터의 설정 항목은, 다음과 같습니다.

AI 번역본
-For Reference Only-



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	접속할 Modbus 기기의 호스트명 또는 IP 주소를 입력
포트	접속할 PLC의 포트를 입력
엔디안	바이트 오더를 선택
접속 타임아웃(밀리초)	접속 시의 타임아웃 시간을 입력
읽기 타임아웃(밀리초)	읽기 시의 타임아웃 시간을 입력
슬레이브 ID	접속처의 슬레이브 ID를 입력
일괄 취득	한 번에 복수의 워드 레지스터의 불러오기를 수행하는 경우 ON
엄밀한 타임스탬프	수집한 데이터에 보다 엄밀한 타임스탬프를 부여하는 경우 ON

AI 번역본
-For Reference Only-

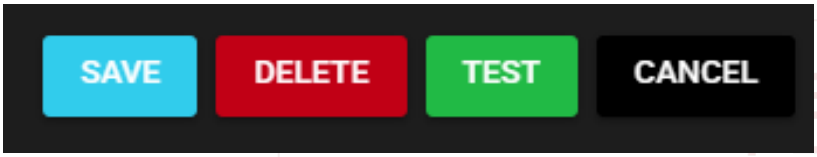
- Modbus TCP 컬렉터의 출력 데이터는, 다음의 설정 항목에 의해 정의됩니다.

항목	설명
데이터명	데이터의 이름을 입력
주소	레지스터 주소를 입력 ※Modbus의 사양상, 주소 범위는 통상「1~65536」으로 되어 있습니다만, SpeeDBee Synapse에서 설정할 때의 주소 범위는「0~65535」와 같이 -1한 주소를 설정할 필요가 있습니다.
데이터형	취득할 레지스터 값의 데이터형을 선택
연속 영역 수	입력한 주소를 선두로 하여 연속하는 레지스터 중에서, 값 취득의 대상으로 할 범위를 입력
종류	레지스터 값의 종류를 선택
엔디안	레지스터 단위에서의 바이트 오더를 선택
수집 간격(밀리초)	레지스터 값을 취득하는 간격을 입력
유효	레지스터 값을 취득하는 경우 ON

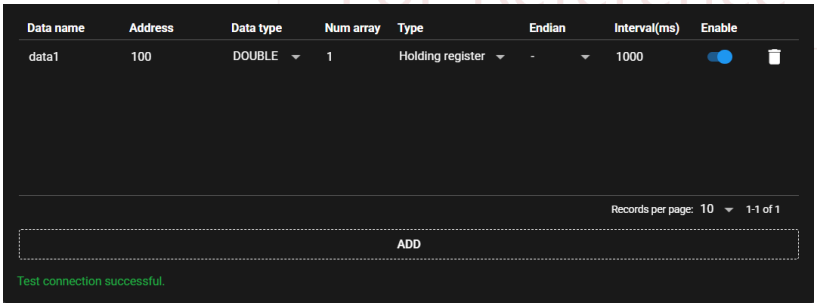
5.4.3.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 Modbus TCP 컬렉터의 설정 화면을 엽니다.
2. 「테스트」를 누릅니다.



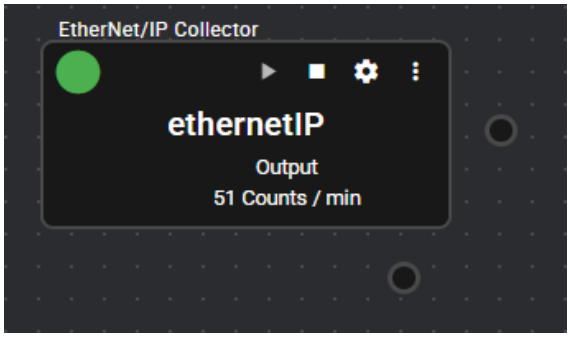
3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



-For Reference Only-

5.4.4 EtherNet/IP 컬렉터

EtherNet/IP 대응 기기에서 데이터 수집을 수행하는 컴포넌트입니다.
본 컴포넌트가 오리지네이터가 되어, Ethernet/IP 대응 기기에서 데이터를 수집, 등록합니다.



5.4.4.1 기본 정보

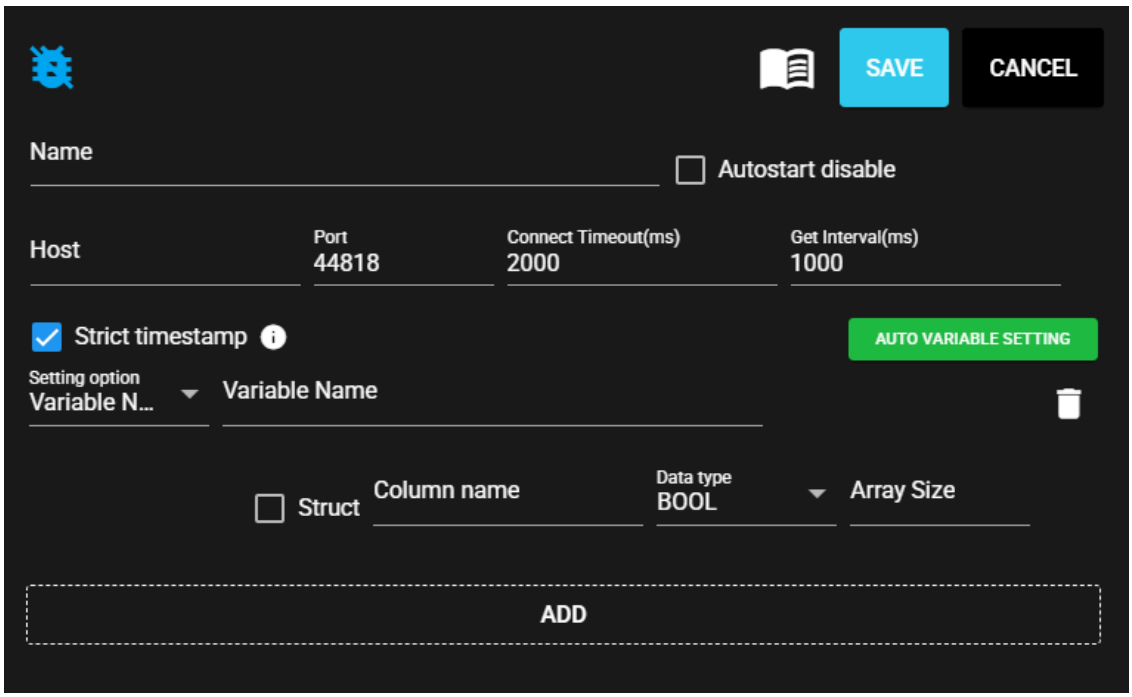
EtherNet/IP 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	EtherNet/IP 컬렉터
설명	EtherNet/IP 대응 기기에서 데이터를 수집합니다.
소비 포인트	2
포트	입력 : 0 출력 : 1 「 설정 항목 」에 따라, EtherNet/IP 대응 기기에서 수집한 데이터를 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	변수를 자동으로 설정할 수 있습니다. 상세는「 변수 자동 설정 」을 참조하십시오.

AI 번역본
-For Reference Only-

5.4.4.2 설정 항목

EtherNet/IP 컬렉터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	EtherNet/IP 대응 기기의 호스트명 또는 IP 주소를 입력
포트	EtherNet/IP 대응 기기의 포트를 입력
접속 타임아웃(밀리초)	접속 시의 타임아웃 시간을 입력
취득 간격(밀리초)	데이터를 취득하는 간격을 입력
엄밀한 타임스탬프	수집한 데이터에 보다 엄밀한 타임스탬프를 부여하는 경우 ON

-For Reference Only-

- EtherNet/IP 컬렉터의 출력 데이터는, 다음의 설정 항목에 의해 정의됩니다.

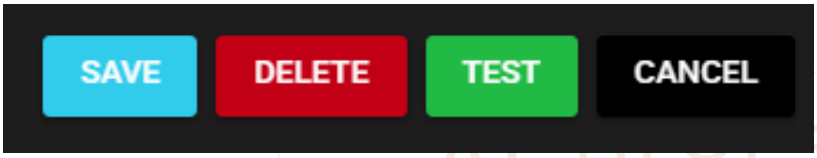
항목	설명
설정 방법	취득할 데이터의 설정 방법을 선택
변수명	태그명을 입력 ※설정 방법이 변수명인 경우에만 입력 가능합니다.

항목	설명
구조체	복수의 멤버를 등록하는 경우 ON ※설정 방법이 변수명인 경우에만 설정 가능합니다.
클래스 ID	CIP 오브젝트의 클래스 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
인스턴스 ID	CIP 오브젝트의 인스턴스 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
어트리뷰트 ID	CIP 오브젝트의 어트리뷰트 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
컬럼명	멤버의 이름
데이터형	멤버의 데이터형
배열 수	멤버의 배열 수

5.4.4.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 EtherNet/IP 컬렉터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



5.4.4.4 변수 자동 설정

Sysmac Studio가 익스포트하는 글로벌 변수 정보 등을, 지정의 포맷의 텍스트 파일로 변환하여 읽어들임으로써, 변수를 자동 설정할 수 있습니다.

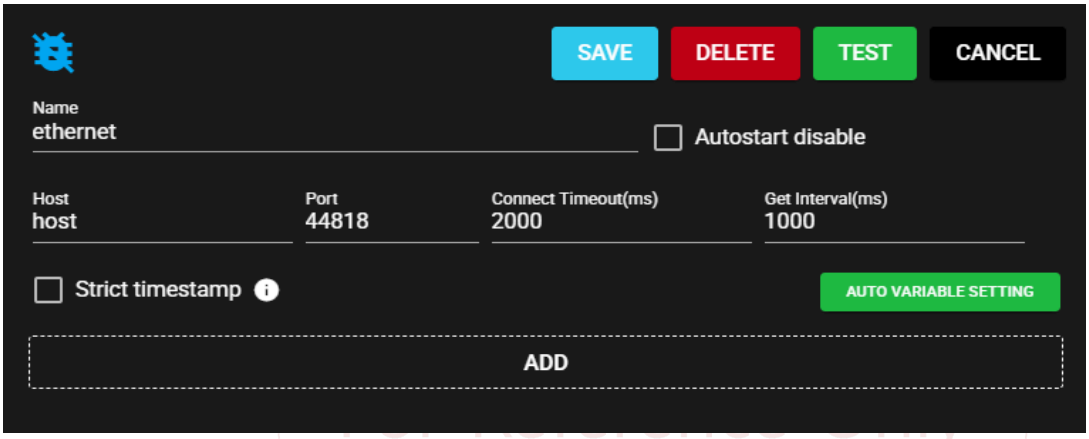
1. Sysmac Studio가 익스포트하는 글로벌 변수 정보 등을, 다음의 열 정보를 갖는 텍스트 파일로 변환합니다.

   - HOST
   - NAME
   - DATATYPE
   - ADDRESS
   - COMMENT
   - TAGLINK
   - RW
   - POU

   예

```
HOST    NAME     DATATYPE     ADDRESS COMMENT TAGLINK RW  POU
        b_test   BOOL         test1           TRUE    RW
        i_test   UINT         test2           TRUE    RW
        s_test   STRING(256)  test3           TRUE    RW
```

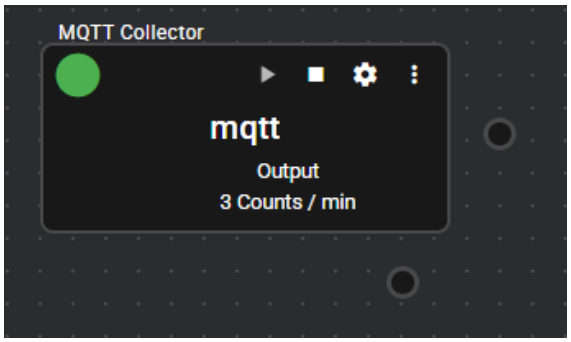
2. 「 변수 자동 설정 」을 누릅니다.



3. 준비한 텍스트 파일을 선택합니다.
4. 파일을 바탕으로, 변수가 자동으로 설정됩니다.

5.4.5 MQTT 컬렉터

MQTT 프로토콜을 사용하여 JSON 데이터의 수집을 수행하는 컴포넌트입니다.
본 컴포넌트는 MQTT에서 특정의 토픽명을 서브스크라이브합니다. 거기에서 수신한 데이터를 JSON 데이터로서 해석하고, 그 내용을 수집 데이터로서 등록합니다.
수신할 수 있는 JSON 데이터의 포맷은, JSON 시리얼라이저의 출력과 같은 형식입니다.
통상의 MQTT 브로커와의 접속 외에, AWS IoT Core와도 접속 가능합니다.



5.4.5.1 기본 정보

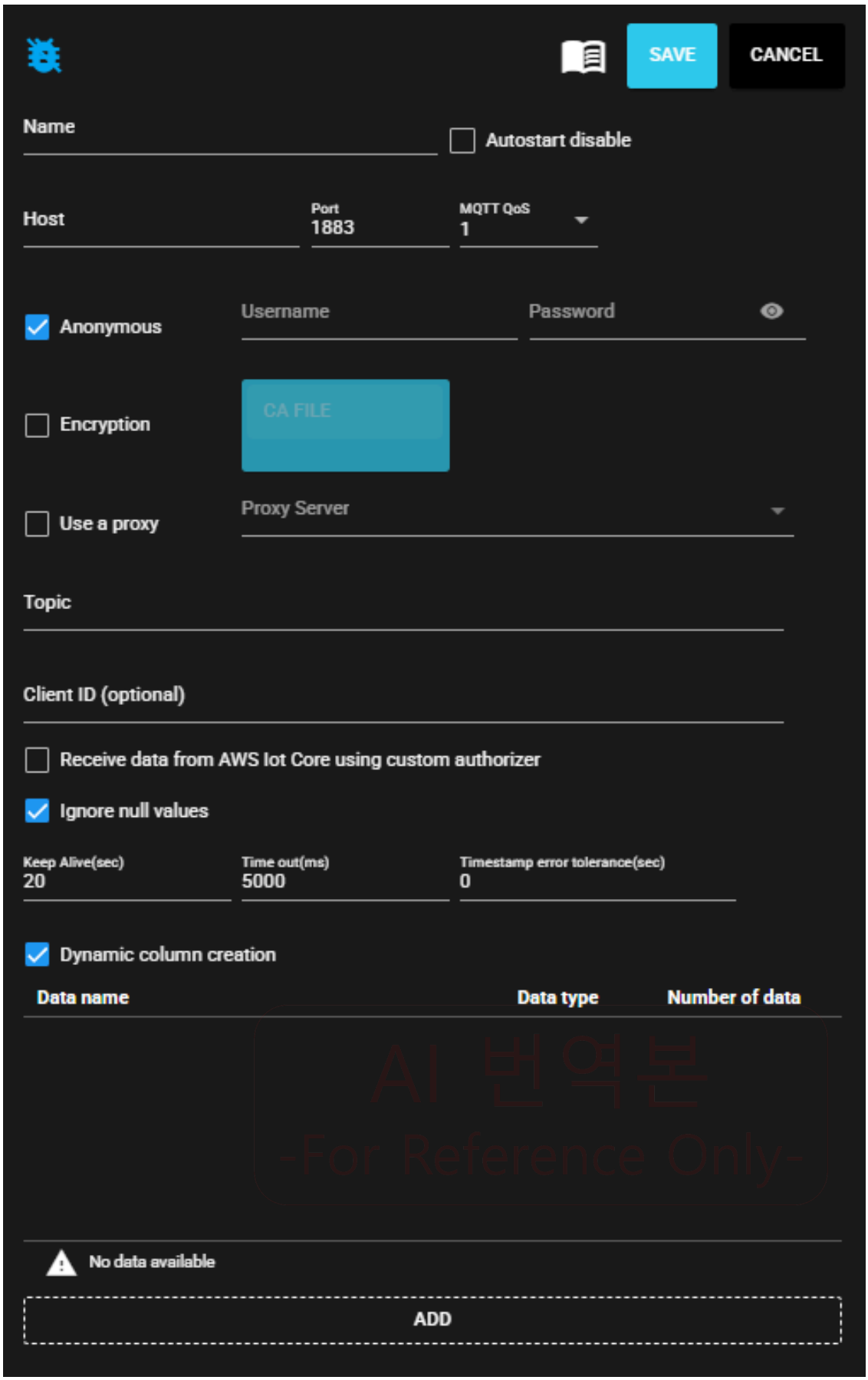
MQTT 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	MQTT 컬렉터
설명	MQTT 프로토콜（Subscribe）에 의한 데이터 수집을 수행합니다.
소비 포인트	2
포트	입력：0 출력：1 「설정 항목」에 의해 지정된 MQTT 브로커로부터 받은 데이터를 출력합니다. 시스템：1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
기타	MQTT 브로커에 대한 테스트 접속을 수행할 수 있습니다. 상세는「테스트 접속」을 참조하십시오.

AI 번역본
For Reference Only

5.4.5.2 설정 항목

MQTT 컬렉터의 설정 항목은, 다음과 같습니다.


SAVE
CANCEL
Name
Autostart disable
Host
Port
1883
MQTT QoS
1
Anonymous
Username
Password
Encryption
CA FILE
Use a proxy
Proxy Server
Topic
Client ID (optional)
Receive data from AWS Iot Core using custom authorizer
Ignore null values
Keep Alive(sec)
20
Time out(ms)
5000
Timestamp error tolerance(sec)
0
Dynamic column creation
Data name
Data type
Number of data
AI 번역본
-For Reference Only-
No data available
ADD

항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	MQTT 브로커의 호스트명 또는 IP 주소를 입력
포트	MQTT 브로커의 포트를 입력
MQTT Qos	MQTT의 Quality of Service를 선택
익명 액세스	사용자 인증을 수행하지 않는 경우 ON
사용자명	인증 사용자명을 입력 ※익명 액세스가 무효인 경우에만 입력이 가능합니다.
패스워드	인증 패스워드를 입력 ※익명 액세스가 무효인 경우에만 입력이 가능합니다.
암호화	암호화를 수행하는 경우 ON
CA 증명서	암호화 통신에 사용하는 CA 증명서를 업로드 ※암호화가 유효한 경우에만 업로드가 가능합니다.
프록시를 이용	MQTT 브로커와의 통신을 프록시 서버를 경유하여 수행하는 경우 ON
프록시 서버	프록시 서버를 선택 ※사전에 프록시 설정에서 사용할 서버를 설정해 둘 필요가 있습니다.
토픽	수집 대상의 토픽을 입력
클라이언트 ID	MQTT 클라이언트 ID ※접속처 사양에 따라서는 지정할 필요가 있습니다. 생략 시는, 랜덤한 문자열 이 사용됩니다. ※복수의 컴포넌트에서, 동일 MQTT 브로커에 접속하는 경우, 클라이언트 ID는 겹치지 않도록 설정하십시오. 통신이 올바르게 수행되지 않게 됩니다.

AI 번역본
-For Reference Only-

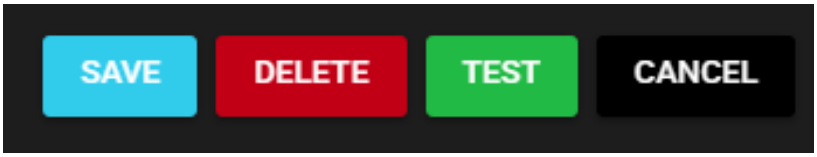
항목	설명
커스텀 인증을 사용하여 AWS Iot Core로부터 데이터를 받아들이다	AWS Iot Core의 커스텀 인증 기능을 사용하여, AWS IoT Core로부터 데이터를 받아들이는 경우 ON ※AWS Iot Core에서 커스텀 인증 설정을 유효하게 해 둘 필요가 있습니다. 자세히는, 이쪽을 참조하십시오.
NULL 값을 무시한다	수신한 데이터 내의 값이 NULL이었을 경우의 거동에 영향을 미칩니다. 체크 ON인 경우는, NULL은 무시되어 에러가 되지 않습니다. 체크 OFF인 경우는, NULL 값을 수신하면, 에러 상태로 천이합니다.
킵얼라이브(초)	MQTT 킵얼라이브 간격을 입력
타임아웃(밀리초)	MQTT 브로커 접속 시의 타임아웃 초수를 입력
타임스탬프 오차 허용 범위(초)	타임스탬프 오차의 허용 범위를 입력
동적 컬럼 생성	MQTT 브로커로부터 받은 데이터로부터, 동적으로 컬럼을 작성하는 경우 ON ※1 MQTT 브로커로부터 데이터를 받아야 비로소 컬럼이 생성됩니다. ※2 동적으로 생성된 컬럼은 컴포넌트가 정지하면 없어집니다. 데이터를 사용하고 싶은 경우는 고정 컬럼으로서 등록하십시오.
데이터명	고정 컬럼의 이름을 입력
데이터형	고정 컬럼의 형을 선택
데이터 수	고정 컬럼의 배열 수를 입력

5.4.5.3 테스트 접속

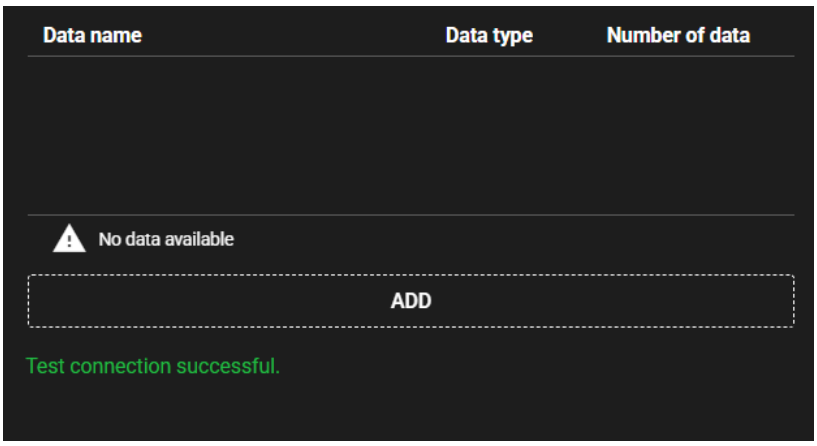
AI 번역본
-For Reference Only-

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 MQTT 컬렉터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



MQTT 컬렉터의 테스트 접속에 관한 주의점

테스트 접속은, 다음의 경우에만 조작 가능합니다.
・컴포넌트가 정지해 있는 경우
・컴포넌트의 스테이터스가 에러인 경우
※스테이터스에 대한 상세는 「 스테이터스 목록 」을 참조하십시오.

5.4.5.4 수신 데이터의 포맷

MQTT 컬렉터의 수신 데이터의 포맷은, JSON 시리얼라이저와 호환성이 있습니다. JSON 시리얼라이저의 출력 데이터 항목을 확인하십시오.

mosquitto를 사용한 데이터 송신 예

MQTT 컬렉터에서 데이터를 수신하려면, JSON의 "records"에 데이터가 격납되는 설정이므로, 키는 "records"를 지정할 필요가 있습니다.
예를 들어 mosquitto_pub를 사용하여 topic에 데이터를 Publish하는 경우, 아래의 커맨드 조작을 수행합니다.

```
$ mosquitto_pub -d -h 192.168.1.10 -p 1883 -t "topic" -m '{"records": [{"data1":1,
"data2":"Hello,World"}, {"data3":[111,222,333], "data4":3.14}]}'
```

json의 제1계층에, "records"가 없으면 데이터는 수신할 수 없습니다. 또한, "records" 이외는, 무시합니다.

-For Reference Only-

5.4.5.5 클라우드 서비스로부터의 데이터 수신

MQTT 컬렉터는, 다음의 클라우드 서비스로부터의 데이터를 수신 가능합니다. 각각 다음의 보충 사항을 확인하십시오.

AWS IoT Core

AWS IoT Core로 송신된 데이터를 MQTT 컬렉터에서 수신하려면,
「 커스텀 인증을 사용하여 AWS Iot Core로부터 데이터를 받아들이다 」의 설정이 필수가 됩니다.
또한, 클라우드측에서의 설정도 필요하게 됩니다.
상세에 대해서는 AWS 공식 페이지의 「 커스텀 인증과 인가 」를 참조하십시오.
이들 정보는, 2024년 6월 시점의 것으로 되어 있습니다.

Azure IoT Hub

Azure IoT Hub의 데이터를 수신하려면, IoT Hub의 사양에 따른 설정이 필요하게 됩니다. 또한, MQTT 컬렉터에서 수신할 수 있는 것은,「 cloud-to-device용으로 송신된 데이터 」가 됩니다.
다른 디바이스측으로부터 IoT Hub용으로 송신된 데이터는 수신할 수 없습니다.
cloud-to-device로부터 복수의 디바이스용으로 브로드캐스트하는 것은 할 수 없습니다.
이들 정보는, 2024년 6월 시점의 것으로 되어 있습니다. 자세한 내용은, 다음의 공식 페이지를 확인하십시오.

- MQTT 프로토콜을 사용한 IoT 허브와의 통신
- IoT 허브에서 cloud-to-device 메시지를 송신한다
- IoT Hub 를 사용한 클라우드에서 디바이스로의 메시지 송신 (Python)
- Shared Access Signature 를 사용하여 IoT Hub 로의 액세스를 제어한다(SAS 토큰)

요약한 설정 내용은 다음과 같습니다. 반드시 위의 공식 페이지를 확인하고 설정하십시오.

- IoT Hub는, Standard 레벨일 필요가 있습니다. Basic, Free에서는 사용할 수 없습니다.
- 호스트：{IoT Hub의 명칭}.azure-devices.net
- 포트：8883
- 사용자명：{IoT Hub의 명칭}.azure-devices.net/{디바이스Id}/?api-version=2021-04-12
- 패스워드：SAS 토큰
- 토픽：devices/{디바이스Id}/messages/events/"
- 클라이언트 ID：{디바이스Id}
- 암호화：유효
- CA 증명서：DigiCert 루트 증명서（ DigiCert Global Root G2 ）

AI 번역본
-For Reference Only-

Azure Event Grid의 MQTT 브로커에는 현재 대응하지 않습니다.

5.4.5.6 각종 에미터로부터의 데이터 수신

JSON 시리얼라이저에서 생성한 데이터를, 각종 에미터에서 데이터 송신하여, 다른 MQTT 컬렉터에서 수신하는 것이 가능합니다.

송신측	수신측	가부
MQTT 에미터	MQTT 컬렉터	○ ※동일한 MQTT 브로커로의 접속 설정이 필요합니다.
Cloud 에미터(AWS용)	MQTT 컬렉터	○ ※MQTT 컬렉터는 커스텀 인증을 설정할 필요가 있습니다.
Cloud 에미터(Azure용)	MQTT 컬렉터	× ※디바이스용 데이터를 MQTT 컬렉터에서는 수신할 수 없습니다.

AI 번역본
-For Reference Only-

다른 에미터로부터 데이터를 수신하는 경우의 주의점

각종 에미터에서 송신된 데이터를 수신하는 경우, 아래의 점에 주의하십시오.

1.송신측에서 1건마다 에미터로 데이터 송신하는 경우：

예）
송신측의 JSON 시리얼라이저의 데이터를 1건으로 전환
수신측의 MQTT 컬렉터의 오차 허용 범위（초）를 10

이 경우, 데이터 등록으로부터 10초 이내에 데이터를 수신할 수 있으면, 원래의 타임스탬프 그대로 데이터는 등록됩니다. 다만, 클라우드로의 데이터 송신이 빈번하게 이루어지는 점에 주의하십시오.

2.송신측에서 집약한 값을 에미터로 송신하는 경우：

예）
송신측의 JSON 시리얼라이저의 데이터를 1건으로 전환
송신측의 JSON 시리얼라이저에서는, 입력 포트에서 집약한다, 임의의 데이터의 평균값으로 한다, 윈도우 폭 60초
수신측의 MQTT 컬렉터의 오차 허용 범위（초）를 10

이 경우, 지정된 폭으로, 값의 평균값이 등록되어, 에미터로 송신됩니다. 컬렉터는, 이것을 수신하여, 원래의 타임스탬프로 데이터를 등록합니다. 전 데이터를 송신하지 않으므로, 네트워크에의 부하는 작아집니다.

3.송신측에서 일정 시간 데이터를 모아 에미터로 송신하는 경우：

예）
송신측의 JSON 시리얼라이저의 데이터를 60초로 전환
수신측의 MQTT 컬렉터의 오차 허용 범위（초）를 0

이 경우, 모든 데이터가 수신한 현재 시각으로 등록됩니다. 원래의 타임스탬프로의 등록이 되지 않는 점에 주의하십시오.

AI 번역본
-For Reference Only-

4.송신측에서 일정 시간 데이터를 모아 에미터로 송신하는 경우（타임스탬프도 원래 그대로 등록）：

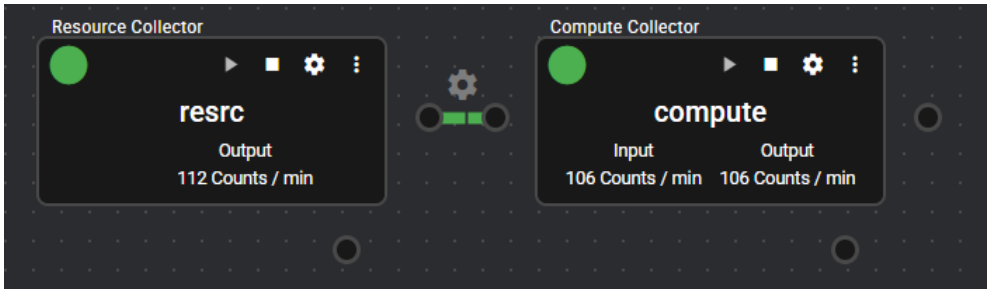
이를 실현하려면, 적어도 아래의 설정이 필요합니다.
예）
송신측의 JSON 시리얼라이저의 데이터를 10~30초 정도로 전환
수신측의 MQTT 컬렉터의 오차 허용 범위（초）를 60
MQTT 컬렉터에 접속되는 컴포넌트의 입력 포트에서, 지연 시간을 61000（밀리초）로 한다

이 설정에 의해, 송신된 데이터의 타임스탬프가 60초 이내에 들어가 있으면, 데이터는 등록할 수 있습니다. 다만, 송수신 확인을 실시하여, 성능적으로 문제없는지, 수신한 데이터 처리가 기대 결과로 되어 있는지를 충분히 검증한 후 이용하십시오.

5.4.6 연산 컬렉터

임의의 계산식으로 연산을 수행하는 컴포넌트입니다.
다른 컴포넌트가 출력한 데이터를 입력 포트로부터 받아, 그 데이터를 임의의 계산식으로 계산한 결과를 신규 데이터로서 출력합니다.

계산식에는 사칙 연산이나 논리 연산 외, 다양한 함수를 사용할 수 있습니다.



5.4.6.1 기본 정보

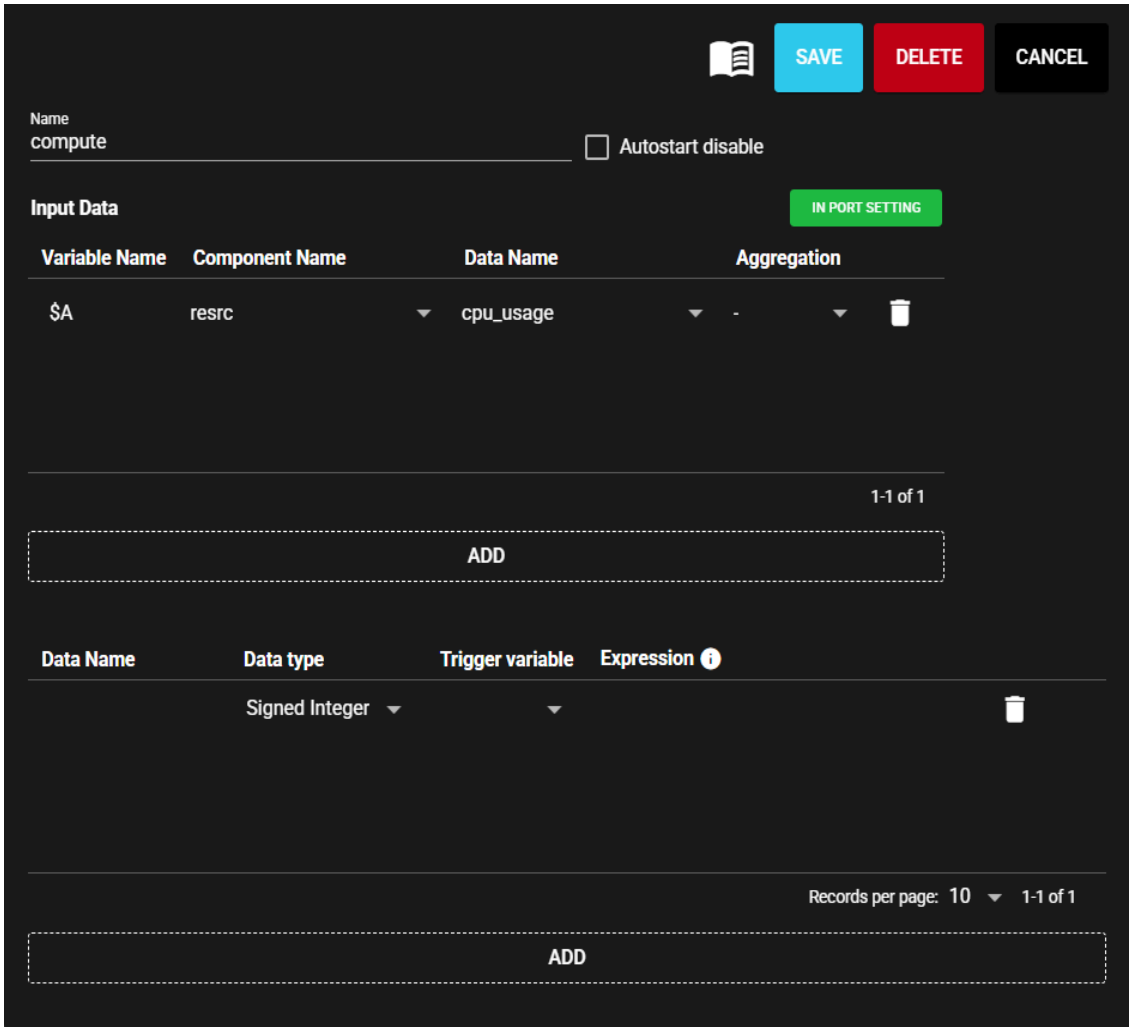
연산 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	연산 컬렉터
설명	다른 컬렉터로부터 데이터를 받아 연산을 수행하여, 새로운 별도 데이터로서 취급합니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 「 설정 항목 」에 따라, 연산 결과를 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는, 「 에러 코드 목록 」을 참조하십시오.
기타	입력 포트의 설정 화면을 열어, 다른 컴포넌트로부터 받은 데이터에 집약 등의 설정을 수행하는 것이 가능합니다. 상세는 「 입력 포트 설정 」을 참조하십시오.

AI 번역본
-For Reference Only-

5.4.6.2 설정 항목

연산 컬렉터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
변수명	입력 데이터에 자동으로 부여되는 이름
컴포넌트명	입력 포트의 데이터 목록으로부터 컴포넌트를 선택
데이터명	위에서 선택한 컴포넌트의 데이터 목록으로부터 데이터를 선택

AI 번역본
-For Reference Only-

항목	설명
집약	집약값을 선택 ※컴포넌트의 데이터에 집약 옵션을 적용하고 있었던 경우에만 선택 가능합니다. 집약의 상세는 「 입력 포트 」를 참조하십시오.

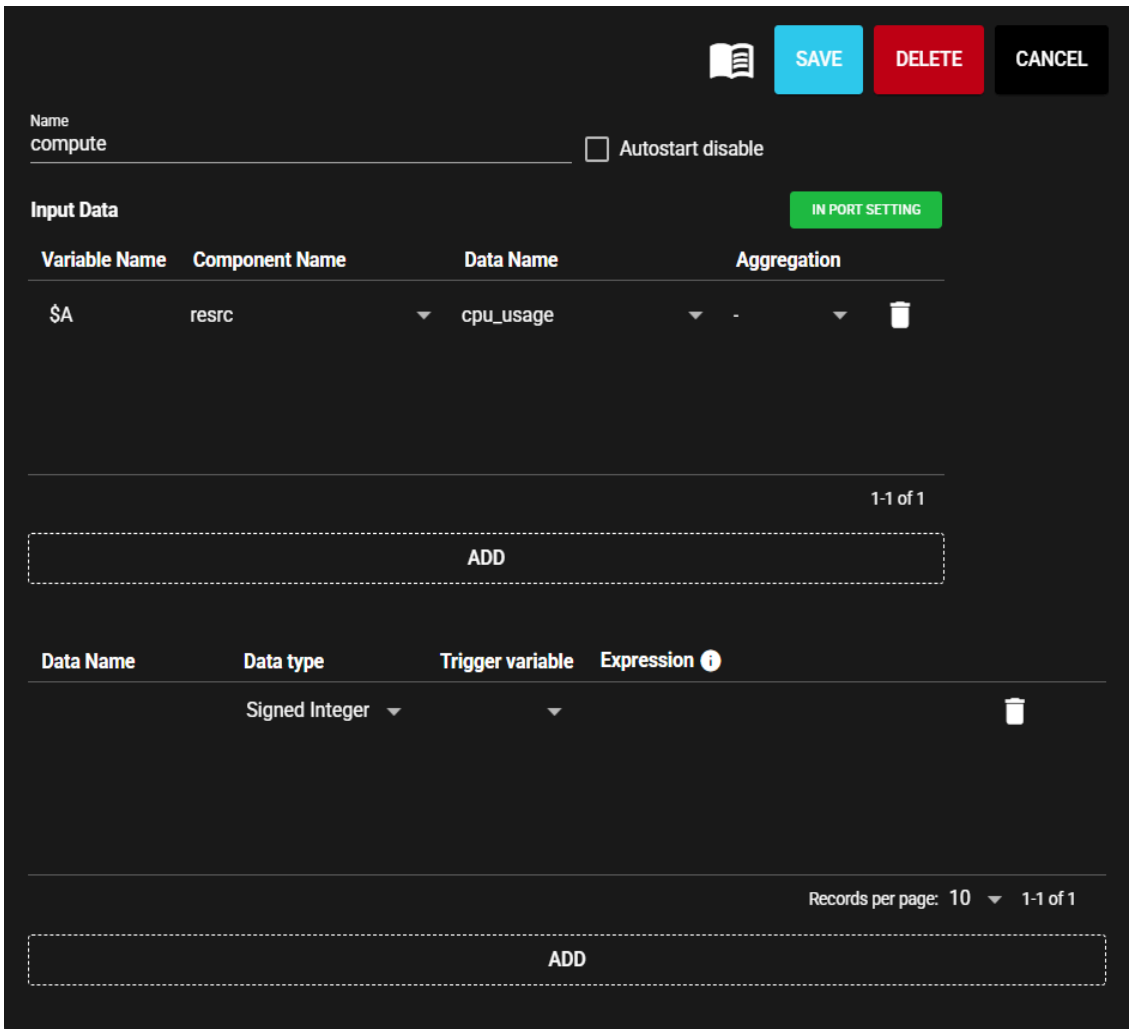
- 연산 컬렉터의 출력 데이터는, 다음의 설정 항목에 의해 정의됩니다.

항목	설명
데이터명	데이터의 이름을 입력
데이터형	데이터의 형을 선택
트리거 변수	연산을 실행하는 트리거가 되는 변수 입력 데이터에서 설정한 변수를 선택 가능합니다. 선택한 변수가 갱신되면 연산이 실행됩니다.
연산식	컬렉터가 실행하는 연산식을 입력 ※표제의「 연산식 」의 우측 아이콘을 클릭하면, 입력 예와 식에 사용할 수 있는 요소 목록이 표시됩니다.

5.4.6.3 입력 포트 설정

추가 완료된 연산 컬렉터에서는, 설정 화면에서「 입력 포트 설정 」을 눌러, 연산 컬렉터의 입력 포트의 설정 화면을 열 수 있습니다.
입력 포트의 설정 화면에서는, 다른 컴포넌트로부터 받은 데이터에 집약 등의 설정을 수행할 수 있습니다. 상세는
「 입력 포트 」를 참조하십시오.

AI 번역본
-For Reference Only-



AI 번역본
-For Reference Only-

5.5 시리얼라이저

여기에서는, 각 시리얼라이저 컴포넌트에 대해 설명합니다.

5.5.1 JSON 시리얼라이저

지정 데이터를 JSON 형식으로 가공하는 컴포넌트입니다.
입력 포트로부터 컴포넌트의 데이터를 받아들여, 일정 시간/건수로 「 컴포넌트명:데이터명 」을 키로 한 JSON 형식으로 가공하여 출력합니다.



5.5.1.1 기본 정보

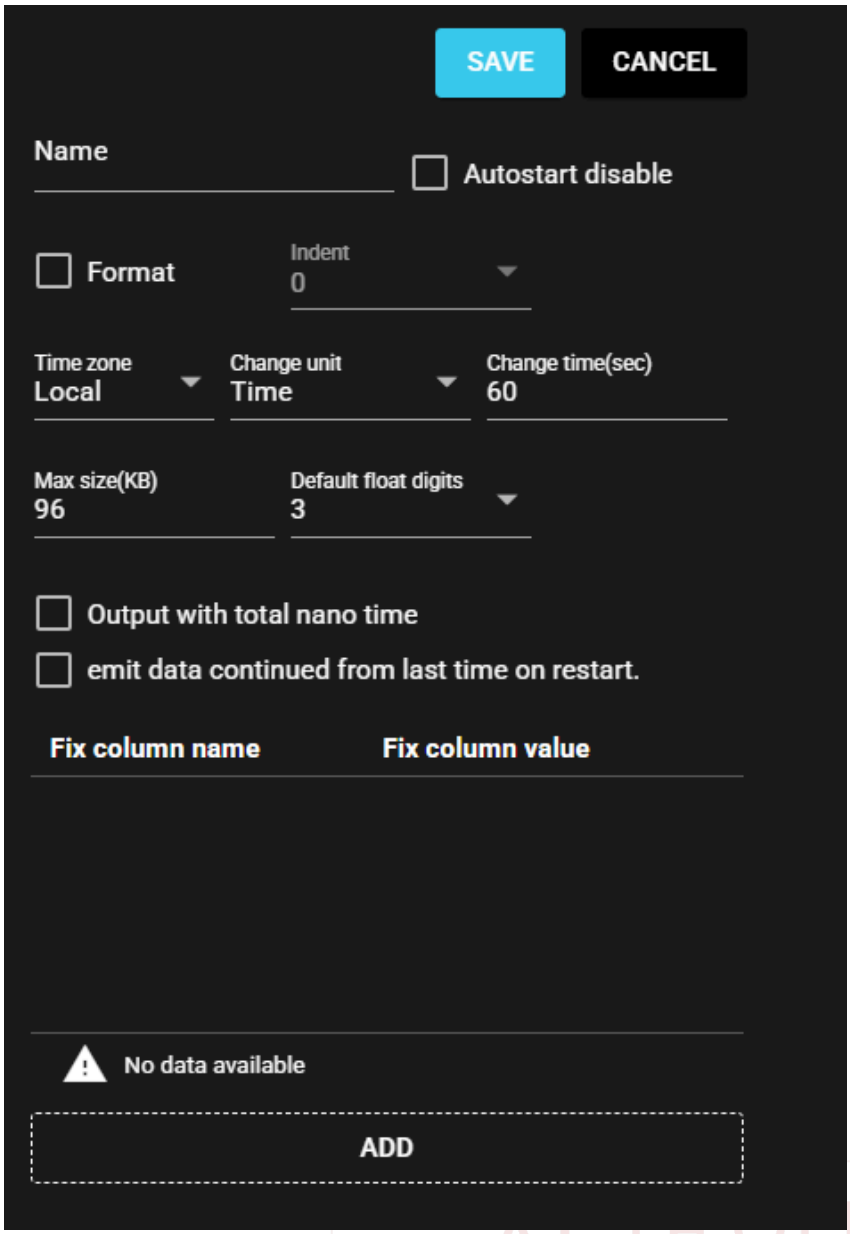
JSON 시리얼라이저의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	JSON
설명	컴포넌트의 데이터를 받아들여 JSON 포맷으로 가공하고, 다른 컴포넌트에 넘깁니다.
소비 포인트	0
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 입력한 데이터를 가공하여 작성한 JSON 파일을 출력합니다. 출력하는 데이터의 상세는, 「 출력 데이터 항목 」을 참조하십시오. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는, 「 에러 코드 목록 」을 참조하십시오.

AI 번역본
For Reference Only-

5.5.1.2 설정 항목

JSON 시리얼라이저의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
정형한다	JSON을 정형하는 경우 ON

항목	설명
인덴트	JSON에 부여하는 인덴트를 지정 ※정형한다를 ON으로 한 경우 입력 가능
타임존	JSON 내의 데이터에 부여하는 타임스탬프의 타임존을 선택
전환 기준	JSON 파일을 전환하는 기준을 선택
전환 시간(초)	1개의 JSON 파일에 몇 초분의 데이터를 기록할지 입력 ※전환 기준이 시간인 경우 입력 가능
전환 건수	1개의 JSON 파일에 몇 건 데이터를 기록할지를 입력 ※전환 기준이 건수인 경우 입력 가능
상한 사이즈(KB)	1개의 JSON 파일의 상한 사이즈를 입력
디폴트 소수 자릿수	데이터가 부동 소수인 경우의 송신하는 자릿수를 설정 각 데이터에 소수 자릿수가 지정되어 있지 않은 경우의 디폴트값으로서 이용한다
통산 나노 시간 첨부로 출력한다	데이터에 통산 나노초를 추가한다
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	컴포넌트를 정지하고, 재개시켰을 때에 전회 송신한 데이터의 계속부터 데이터를 출력한다

임의로 JSON 파일에 고정값을 출력할 수 있습니다.

항목	설명
고정값 컬럼명	고정값의 컬럼명
값	고정값으로서 출력하는 값

5.5.1.3 출력 데이터 항목

JSON 시리얼라이저는, 다음 데이터를 출력합니다.

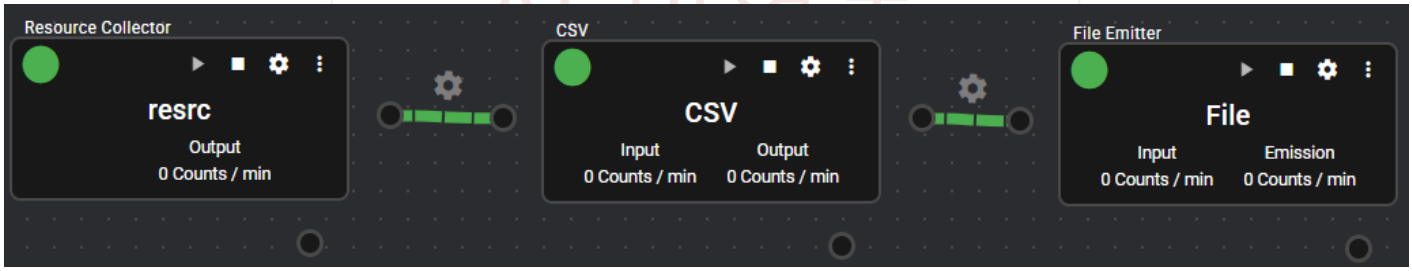
항목	컬럼명	데이터형	설명
JSON 데이터 출력	output file	FILE	작성한 JSON 파일을 출력하는 컬럼. 미디어 타입은"application/json"고정

이하는, 리소스 컬렉터의 I/O 데이터를 받아들여, 생성된 output file의 예입니다. "records"에, 데이터가 격납됩니다.
"timestamp"는, 데이터의 생성 일시입니다. "resrc:~~"는, "resrc"라는 이름의 리소스 컬렉터로부터 받아들인, I/O 데이터입니다.

```
{
  "records": [
    {
      "timestamp": "2023-10-02T05:54:54.837742200Z",
      "resrc:IOPS": 0,
      "resrc:I_O": "sdb",
      "resrc:kb_r_s": 0,
      "resrc:kb_w_s": 0
    },
    {
      "timestamp": "2023-10-02T05:54:55.837652700Z",
      "resrc:IOPS": 0,
      "resrc:I_O": "sdb",
      "resrc:kb_r_s": 0,
      "resrc:kb_w_s": 0
    },
    ~~~
  ]
}
```

5.5.2 CSV 시리얼라이저

지정 데이터를 CSV 형식으로 가공하는 컴포넌트입니다.
입력 포트로부터 컴포넌트의 데이터를 받아들여, 일정 시간/건수로 CSV 형식으로 가공하여 출력합니다. CSV의 나열 순서는 입력 포트의 설정에서 변경할 수 있습니다.



5.5.2.1 기본 정보

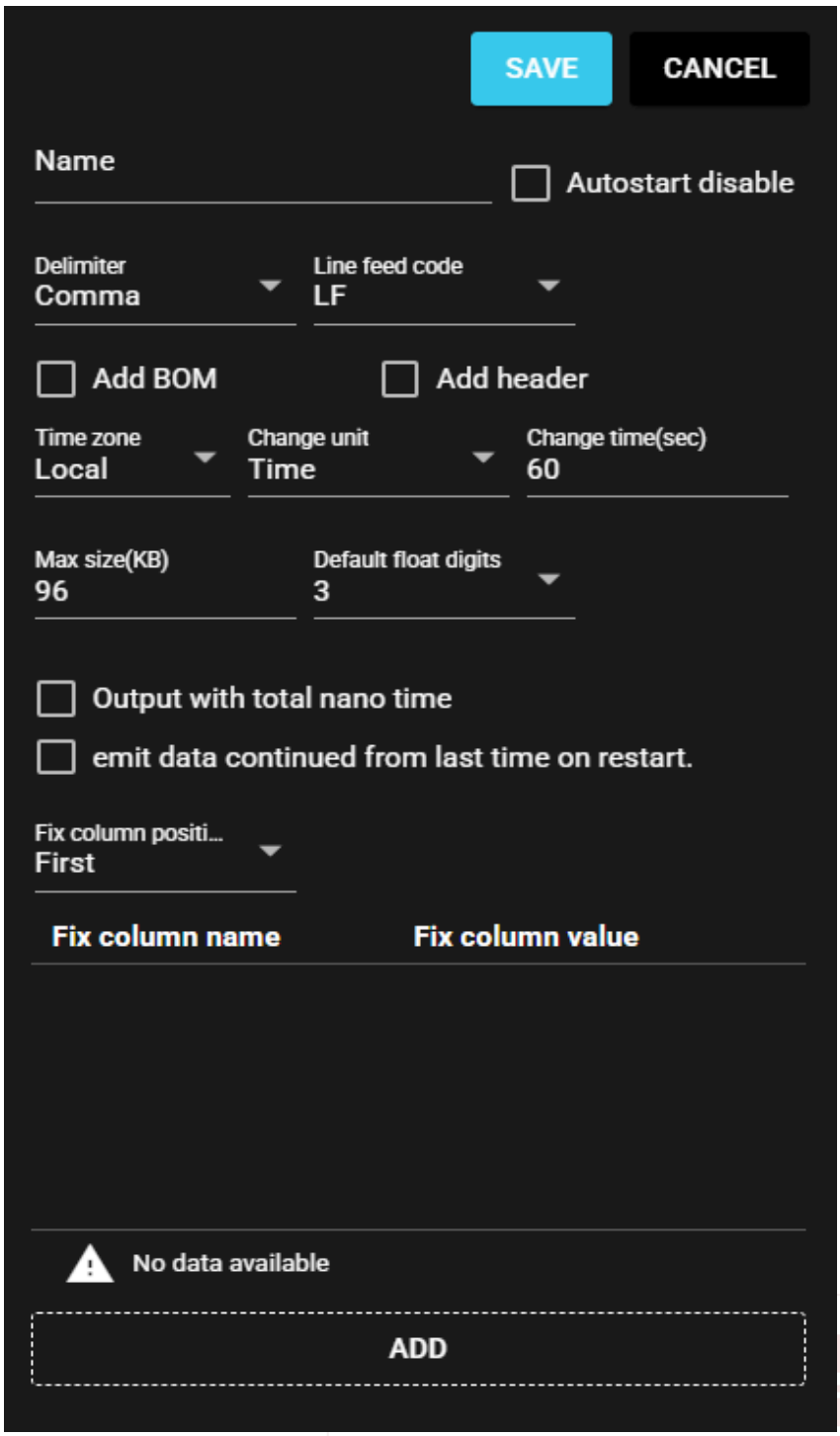
CSV 시리얼라이저의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	CSV
설명	컴포넌트의 데이터를 받아들여 CSV 포맷으로 가공하고, 다른 컴포넌트에 넘깁니다.
소비 포인트	0
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 입력한 데이터를 가공하여 작성한 CSV 파일을 출력합니다. 출력하는 데이터의 상세는,「 출력 데이터 항목 」을 참조하십시오. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

5.5.2.2 설정 항목

CSV 시리얼라이저의 설정 항목은, 다음과 같습니다.

AI 번역본
-For Reference Only-



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
구분 문자	CSV 파일의 구분 문자를 선택

항목	설명
개행 코드	CSV 파일의 개행 코드를 선택
BOM을 붙인다	CSV 파일에 파일 BOM을 붙이는 경우 ON
헤더를 붙인다	CSV 파일에 헤더를 붙이는 경우 ON
타임존	CSV 데이터에 부여하는 타임스탬프의 타임존을 선택
전환 기준	CSV 파일을 전환하는 기준을 선택
전환 시간(초)	1개의 CSV 파일에 몇 초분의 데이터를 기록할지 입력 ※전환 기준이 시간인 경우 입력 가능
전환 건수	1개의 CSV 파일에 몇 건 데이터를 기록할지를 입력 ※전환 기준이 건수인 경우 입력 가능
상한 사이즈(KB)	1개의 JSON 파일의 상한 사이즈를 입력
디폴트 소수 자릿수	데이터가 부동 소수인 경우의 송신하는 자릿수를 설정 각 데이터에 소수 자릿수가 지정되어 있지 않은 경우의 디폴트값으로서 이용한다
통산 나노 시간 첨부로 출력한다	데이터에 통산 나노초를 추가한다
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	컴포넌트를 정지하고, 재개시켰을 때에 전회 송신한 데이터의 계속부터 데이터를 출력한다
고정값 컬럼의 위치	CSV 파일 내에서의 고정값 컬럼의 위치를 선택

AI 번역본
For Reference Only

임의로 CSV 파일에 고정값을 출력할 수 있습니다.

항목	설명
고정값 컬럼명	고정값의 컬럼명
값	고정값으로서 출력하는 값

5.5.2.3 출력 데이터 항목

CSV 시리얼라이저는, 다음 데이터를 출력합니다.

항목	컬럼명	데이터형	설명
CSV 데이터 출력	output file	FILE	작성한 CSV 파일을 출력하는 컬럼. 미디어 타입은"text/csv"고정

이하는, 리소스 컬렉터의 I/O 데이터를 받아들여, 생성된 output file의 예입니다. 가장 왼쪽의 열은, 데이터의 타임스탬프입니다. 그 이외의 열이, 리소스 컬렉터로부터 받아들인, I/O 데이터입니다.

```
2023-10-02T06:19:52.837670200Z,0.000,"sdb",0.000,0.000
2023-10-02T06:19:53.837656800Z,0.000,"sdb",0.000,0.000
2023-10-02T06:19:54.837805700Z,0.000,"sdb",0.000,0.000
2023-10-02T06:19:55.837734700Z,0.000,"sdb",0.000,0.000
~~~
```

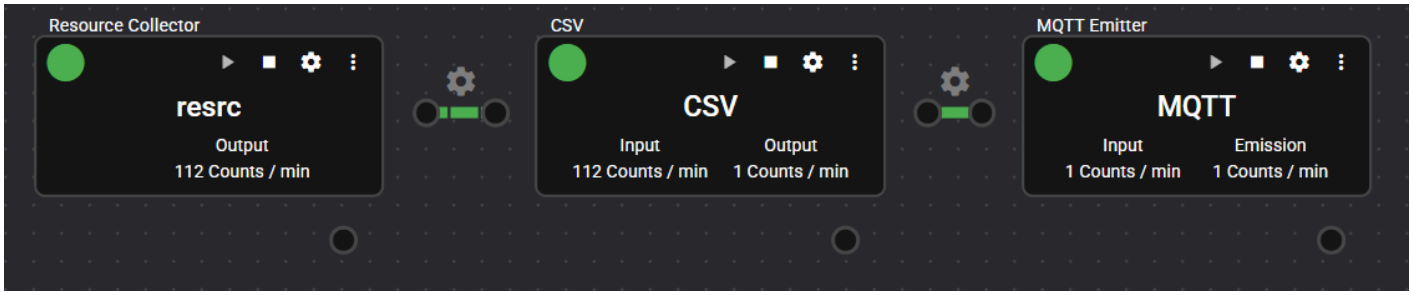
AI 번역본
-For Reference Only-

5.6 에미터

여기에서는, 각 에미터 컴포넌트에 대해 설명합니다.

5.6.1 MQTT 에미터

MQTT 프로토콜로 임의의 데이터 파일을 송신（퍼블리시）하는 컴포넌트입니다. 입력 포트에 연결된 각종 시리얼라이저로부터 데이터를 받아들여, 그 데이터를 송신합니다.



5.6.1.1 기본 정보

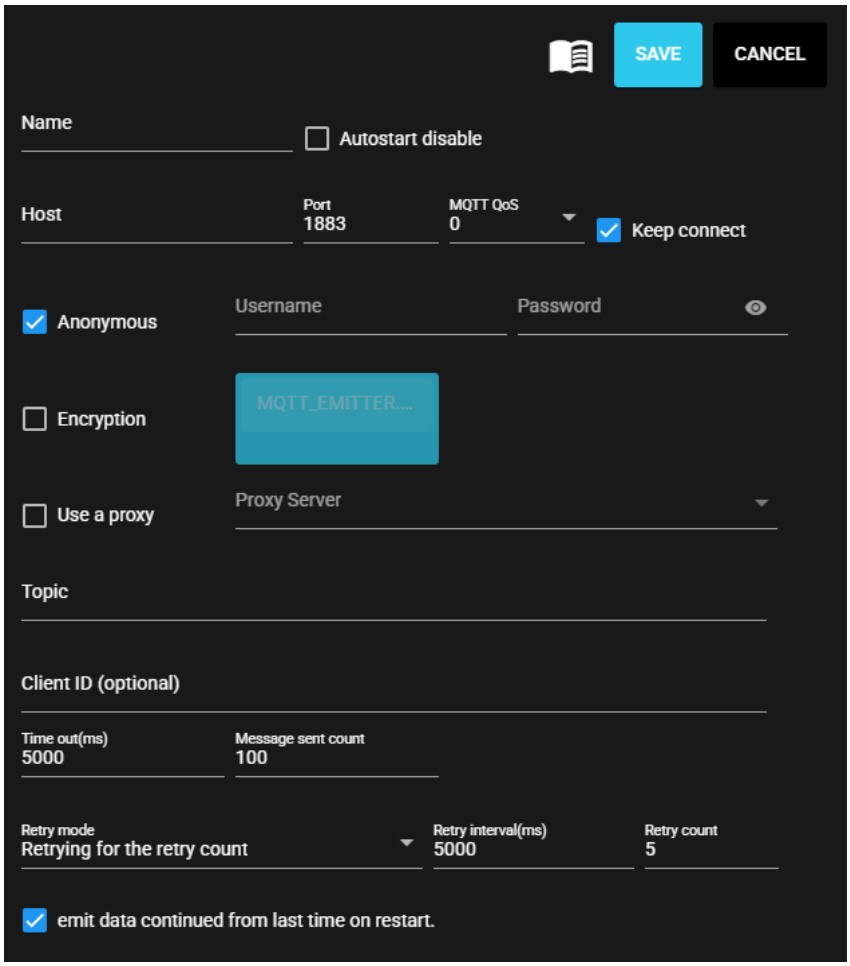
MQTT 에미터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	MQTT 에미터
설명	MQTT 프로토콜（Publish）에 의한 데이터 송신을 수행합니다.
소비 포인트	2
포트	입력：1 시리얼라이저・컴포넌트로부터, 파일 형식의 데이터를 받아들입니다. 받아들인 데이터는, MQTT 프로토콜（Publish）에 의해 송신됩니다. 출력：0 시스템：1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
기타	MQTT 브로커에 대한 테스트 접속을 수행할 수 있습니다. 상세는「테스트 접속」을 참조하십시오.

AI 번역본
-For Reference Only-

5.6.1.2 설정 항목

MQTT 에미터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	MQTT 브로커의 호스트명 or IP 주소를 지정
포트	MQTT 브로커의 포트 번호를 지정
MQTT Qos	사용하는 MQTT의 QoS를 선택（0:수신 확인 없음, 1:수신 확인 있음, 2:수신 확인, 중복 억지）
상시 접속한다	서버 접속을 송신할 때마다, 절단하지 않는다

AI 번역본
-For Reference Only-

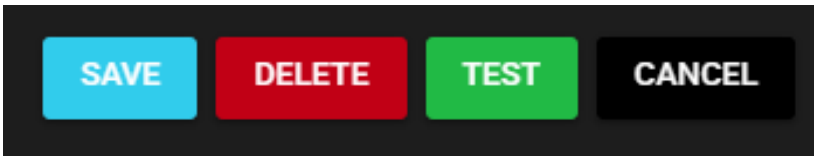
항목	설명
익명 액세스	브로커에 익명 액세스하는 경우는 ON으로 한다
사용자명	익명 액세스가 아닌 경우, 사용자명을 입력
패스워드	익명 액세스가 아닌 경우, 패스워드를 입력
암호화	암호화를 수행하는 경우 ON
CA 증명서	CA 증명서를 업로드 ※암호화가 ON인 경우만 업로드 가능
프록시를 이용	MQTT 브로커와의 통신을 프록시 서버를 경유하여 수행하는 경우 ON
프록시 서버	프록시 서버를 선택 ※사전에 프록시 설정에서 사용할 서버를 설정해 둘 필요가 있습니다.
토픽	Publish 시에 부여하는 토픽
클라이언트 ID	MQTT 클라이언트 ID ※접속처 사양에 따라서는 지정할 필요가 있습니다. 생략 시는, 랜덤한 문자열이 사용됩니다. ※복수의 컴포넌트에서, 동일 MQTT 브로커에 접속하는 경우, 클라이언트 ID는 겹치지 않도록 설정하십시오. 통신이 올바르게 수행되지 않게 됩니다.
타임아웃(밀리초)	송신 처리가 타임아웃될 때까지의 시간을 밀리초 단위로 입력
메시지 송신 횟수	1 회의 이벤트에서의 송신 최대 횟수
리트라이 모드	송신에 실패하여 리트라이할 때의 거동을 선택
리트라이 간격(밀리초)	송신 리트라이의 간격을 입력
리트라이 횟수	송신 리트라이의 횟수를 입력 ※리트라이 모드가「 리트라이 횟수분 재송신 」인 경우만 입력 가능
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	컴포넌트를 정지하고, 재개시켰을 때에 전회 송신한 데이터의 계속부터 데이터를 출력한다

AI 번역본
-For Reference Only-

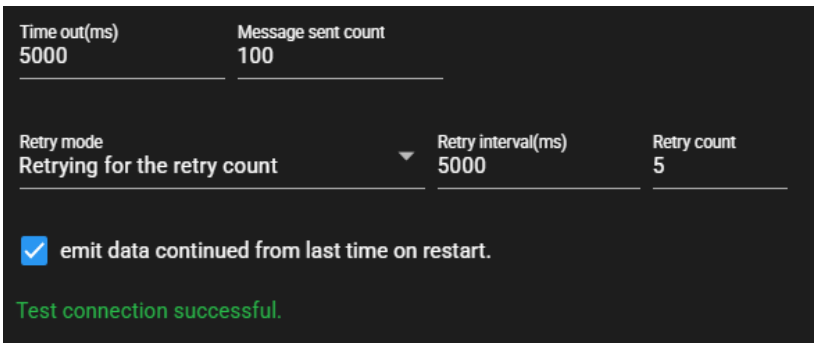
5.6.1.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 MQTT 에미터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.

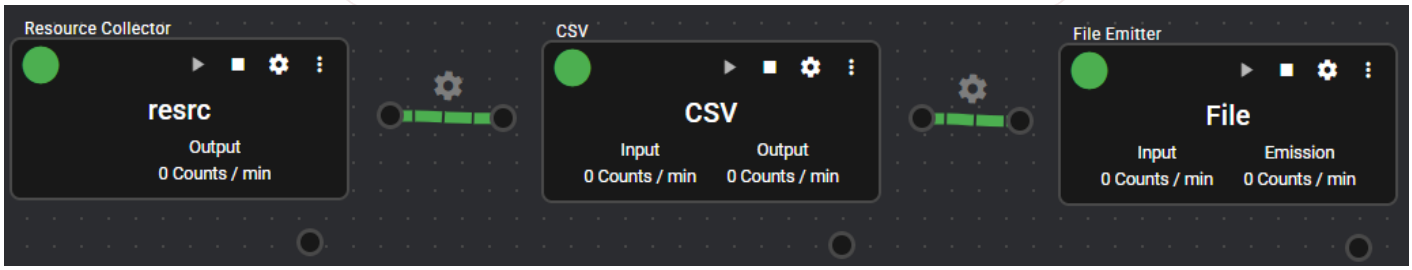


> MQTT 에미터의 테스트 접속에 관한 주의점
>
> 테스트 접속은, 다음의 경우에만 조작 가능합니다.
> ・컴포넌트가 정지해 있는 경우
> ・컴포넌트의 스테이터스가 에러인 경우
> ※스테이터스에 대한 상세는 「 스테이터스 목록 」을 참조하십시오.

5.6.2 File 에미터

수집한 데이터를 파일로서, 스토리지에 저장하는 컴포넌트입니다.
입력 포트에 접속된 각종 시리얼라이저로부터 데이터를 받아들여, 그 데이터를 파라미터로 지정된 파일 경로에 저장합니다.
저장된 파일은, 설정 다이얼로그에서 시간 범위를 지정하여 다운로드할 수 있습니다.

AI 번역본
For Reference Only



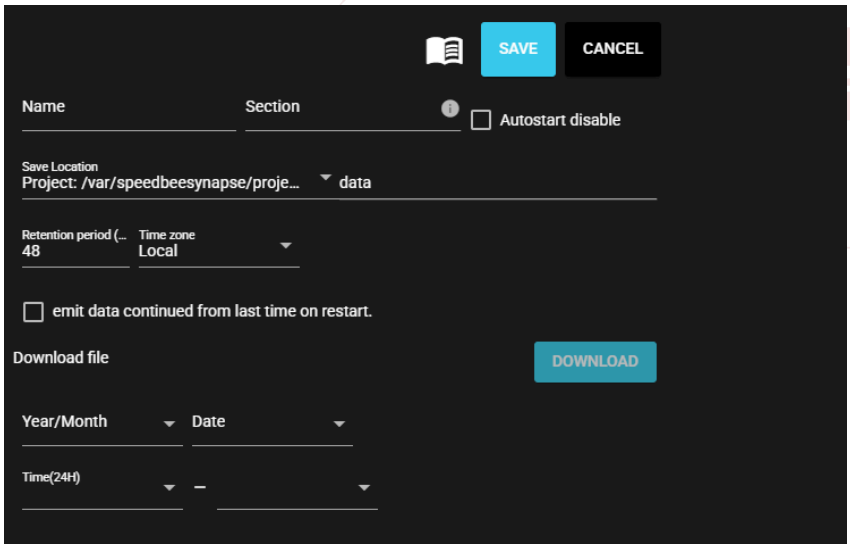
5.6.2.1 기본 정보

File 에미터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	File 에미터
설명	수집 데이터를 스토리지에 저장합니다.
소비 포인트	0
포트	입력 : 1 시리얼라이저 · 컴포넌트로부터, 파일 형식의 데이터를 받아들입니다. 받아들인 데이터는, 스토리지에 저장됩니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	출력한 파일을 다운로드할 수 있습니다. 상세는「 파일 다운로드 」를 참조하십시오.

5.6.2.2 설정 항목

File 에미터의 설정 항목은, 다음과 같습니다.



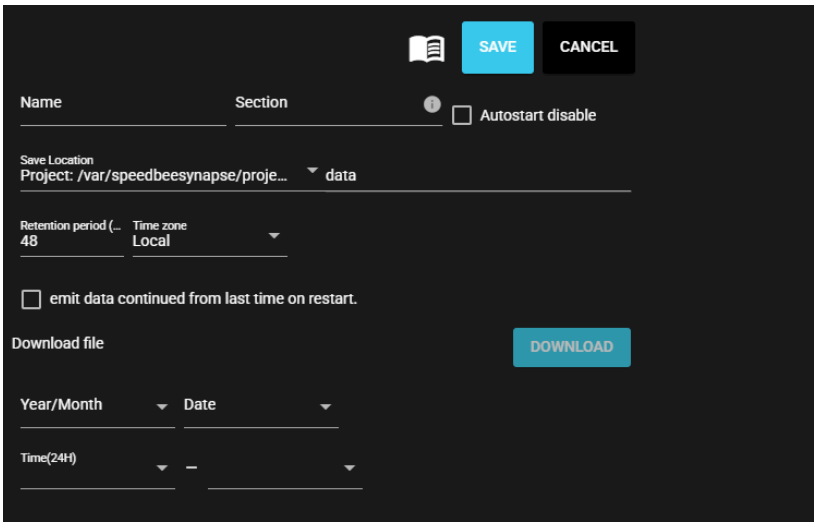
항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
구분	File 에미터마다 고유하게 부여하는 임의의 값을 입력 ※구분은 출력처 디렉터리의 서브 디렉터리명으로서 사용됩니다. 반각만 입력 가능합니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
출력처 디렉터리	파일을 출력하는 디렉터리. 베이스 디렉터리와, 거기부터의 상대 경로를 지정합니다. 베이스 디렉터리는 이하에서 선택 가능합니다. • 프로젝트 경로（프로젝트 하위에 저장됩니다. 프로젝트를 복제 등을 한 경우, 새로운 프로젝트의 하위가 저장 위치가 됩니다.） ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다. • 임의의 경로（통상은 베이스 디렉터리에 대한 상대 경로를 지정합니다만, 임의의 절대 경로를 직접 지정합니다. 프로젝트를 복제 등을 해도 지정한 경로가 저장 위치가 됩니다.）
보존 기간(시간)	파일을 저장해 둘 기간을 입력
타임존	파일의 타임존을 선택
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	기동 후는 전회 송신한 데이터의 계속부터 데이터를 출력한다

5.6.2.3 파일 다운로드

다음 순서로, File 에미터가 출력한 파일을 다운로드할 수 있습니다.

AI 번역본
-For Reference Only-

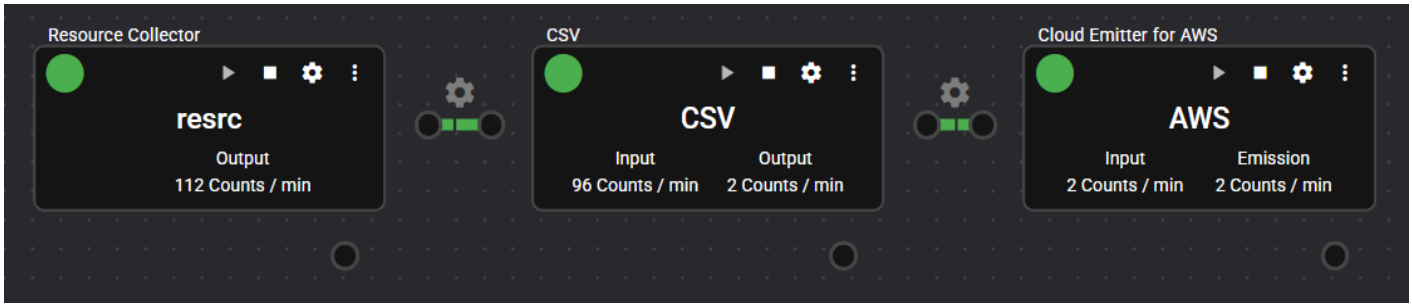
1. 「 파일 다운로드 」 란에서, 시간 범위를 입력합니다.



2. 「 다운로드 」 를 누르면, 위에서 입력한 시간 범위 내에 출력된 파일이, zip 형식으로 다운로드할 수 있습니다.

5.6.3 Cloud 에미터(AWS용)

AWS IoT Core용으로 데이터 송신을 수행하는 컴포넌트입니다.
입력 포트에서 각종 시리얼라이저로부터 데이터를 받아들여, 그 데이터를 AWS IoT Core에 송신합니다.



5.6.3.1 기본 정보

AI 번역본

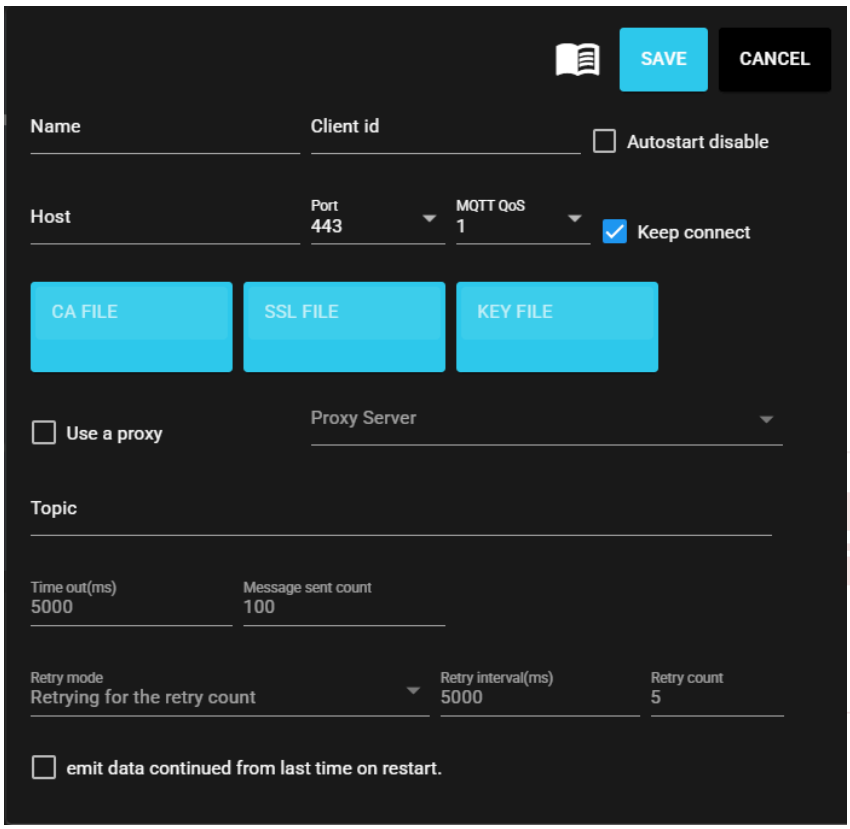
Cloud 에미터(AWS용)의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	Cloud 에미터(AWS용)
설명	AWS IoT Core용으로 데이터 송신을 수행합니다.
소비 포인트	2

항목	설명
포트	입력 : 1 시리얼라이저・컴포넌트로부터, 파일 형식의 데이터를 받아들입니다. 받아들인 데이터는, AWS IoT Core용으로 송신됩니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
기타	AWS IoT Core에 대한 테스트 접속을 수행할 수 있습니다. 상세는「테스트 접속」을 참조하십시오.

5.6.3.2 설정 항목

Cloud 에미터(AWS용)의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
클라이언트 ID	AWS IoT Core의 클라이언트 ID
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	AWS IoT Core의 호스트명을 입력
포트	AWS IoT Core의 포트 번호를 선택（445 or 8883）
MQTT Qos	사용하는 MQTT의 QoS를 선택（0:수신 확인 없음, 1:수신 확인 있음）
상시 접속한다	서버 접속을 송신할 때마다, 절단하지 않는다
CA 증명서	Amazon 루트 CA1을 선택
SSL 증명서	디바이스 증명서를 선택
비밀키	프라이빗 키 파일을 선택
프록시를 이용	AWS IoT Core와의 통신을 프록시 서버를 경유하여 수행하는 경우 ON
프록시 서버	프록시 서버를 선택 ※사전에 프록시 설정에서 사용할 서버를 설정해 둘 필요가 있습니다.
토픽	Publish 시에 부여하는 토픽
타임아웃(밀리초)	송신 처리가 타임아웃될 때까지의 시간(밀리초 단위) ※이 항목은 편집할 수 없습니다.
메시지 송신 횟수	1 회의 이벤트에서의 송신 최대 횟수 ※이 항목은 편집할 수 없습니다.
리트라이 모드	송신에 실패하여 리트라이할 때의 거동 ※이 항목은 편집할 수 없습니다.

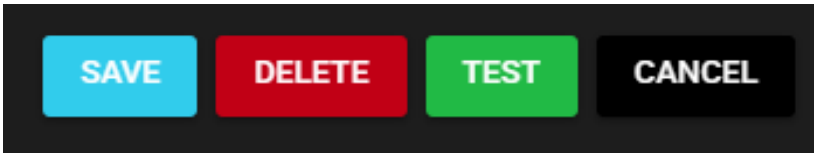
AI 번역본
-For Reference Only-

항목	설명
리트라이 간격(밀리초)	송신 리트라이의 간격 ※이 항목은 편집할 수 없습니다.
리트라이 횟수	송신 리트라이의 횟수 ※이 항목은 편집할 수 없습니다.
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	기동 후는 전회 송신한 데이터의 계속부터 데이터를 출력한다

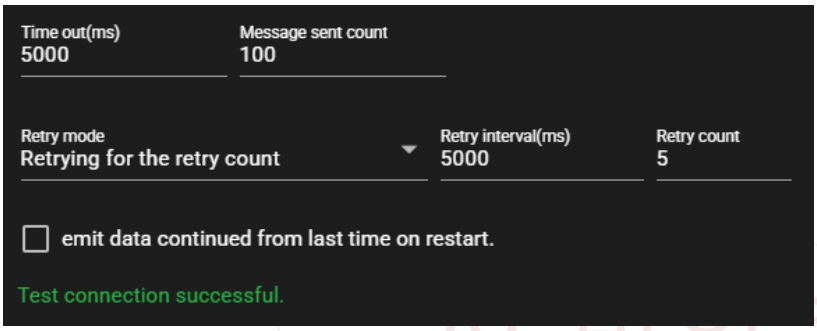
5.6.3.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 Cloud 에미터(AWS용)의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



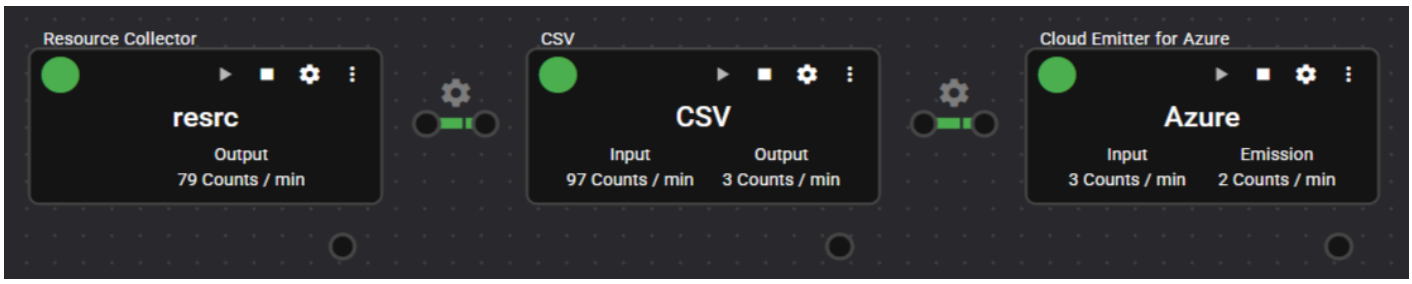
3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



AI 번역본
For Reference Only-

5.6.4 Cloud 에미터(Azure용)

Azure IoT Hub용으로 데이터 송신을 수행하는 컴포넌트입니다.
입력 포트에서 각종 시리얼라이저로부터 데이터를 받아들여, 그 데이터를 Azure IoT Hub에 송신합니다.



5.6.4.1 기본 정보

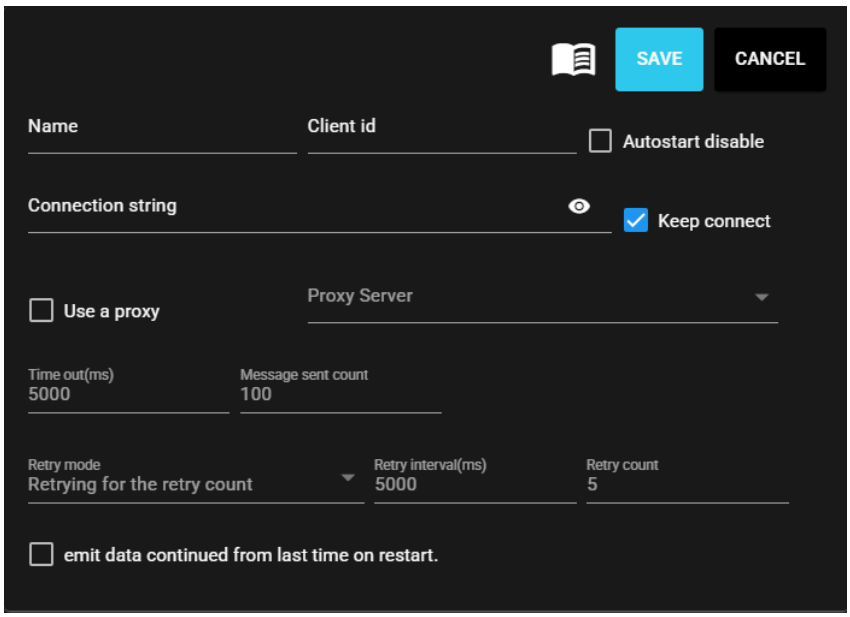
Cloud 에미터(Azure용)의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	Cloud 에미터(Azure용)
설명	Azure IoT Hub용으로 데이터 송신을 수행합니다.
소비 포인트	2
포트	입력 : 1 시리얼라이저・컴포넌트로부터, 파일 형식의 데이터를 받아들입니다. 받아들인 데이터는, Azure IoT Hub용으로 송신됩니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
기타	Azure IoT Hub에 대한 테스트 접속을 수행할 수 있습니다. 상세는「테스트 접속」을 참조하십시오.

-For Reference Only-

5.6.4.2 설정 항목

Cloud 에미터(Azure용)의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
클라이언트 ID	Azure의 클라이언트 ID （MQTT에서 사용하는 클라이언트 ID）
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
접속 문자열	Azure IoT Hub의 접속 문자열을 입력
상시 접속한다	서버 접속을 송신할 때마다, 절단하지 않는다
프록시를 이용	Azure IoT Hub와의 통신을 프록시 서버를 경유하여 수행하는 경우 ON
프록시 서버	프록시 서버를 선택 ※사전에 프록시 설정에서 사용할 서버를 설정해 둘 필요가 있습니다.
타임아웃(밀리초)	송신 처리가 타임아웃될 때까지의 시간(밀리초 단위) ※이 항목은 편집할 수 없습니다.
메시지 송신 횟수	1 회의 이벤트에서의 송신 최대 횟수 ※이 항목은 편집할 수 없습니다.

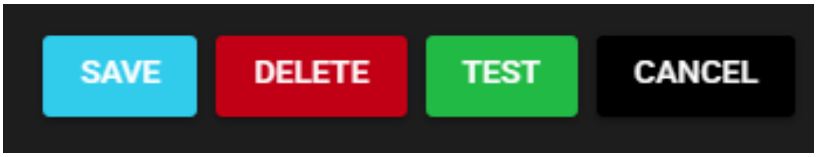
AI 번역본
-For Reference Only-

항목	설명
리트라이 모드	송신에 실패하여 리트라이할 때의 거동 ※이 항목은 편집할 수 없습니다.
리트라이 간격(밀리초)	송신 리트라이의 간격 ※이 항목은 편집할 수 없습니다.
리트라이 횟수	송신 리트라이의 횟수 ※이 항목은 편집할 수 없습니다.
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	기동 후는 전회 송신한 데이터의 계속부터 데이터를 출력한다

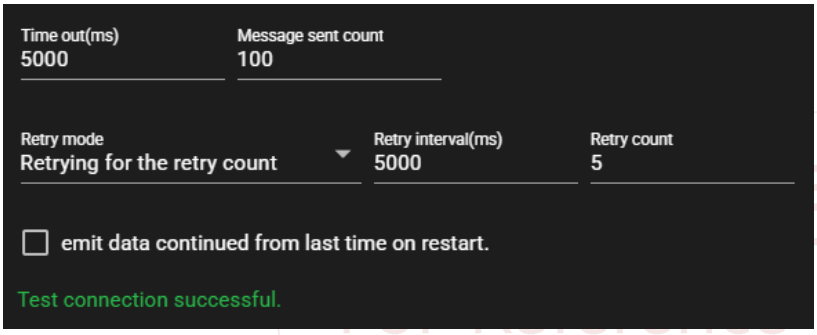
5.6.4.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 Cloud 에미터(Azure용)의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.

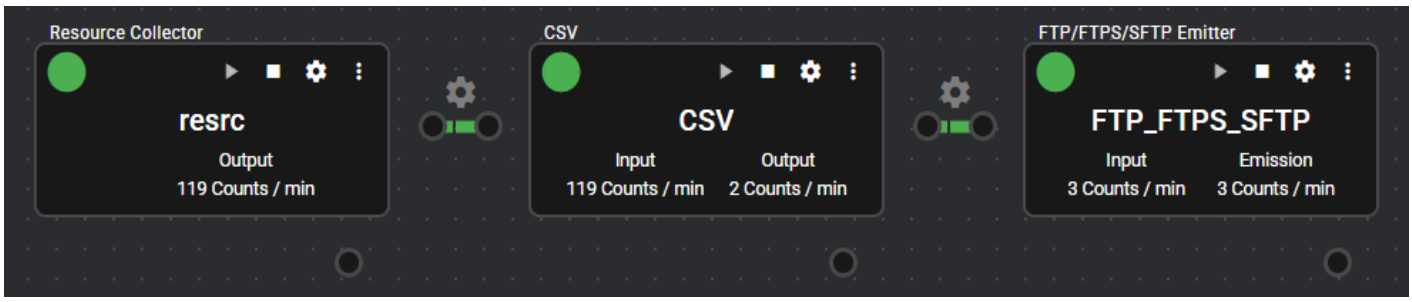


3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



5.6.5 FTP/FTPS/SFTP 에미터

FTP, FTPS 또는 SFTP에 의한 데이터 전송을 수행하는 컴포넌트입니다.
입력 포트에서 각종 시리얼라이저로부터 데이터를 받아들여, 그 데이터를 받아들인 시리얼라이저의 형식의 파일로서 FTP, FTPS 또는 SFTP로 전송합니다.
이 컴포넌트는, Proxy Server를 경유한 통신에는 대응하고 있지 않습니다.



5.6.5.1 기본 정보

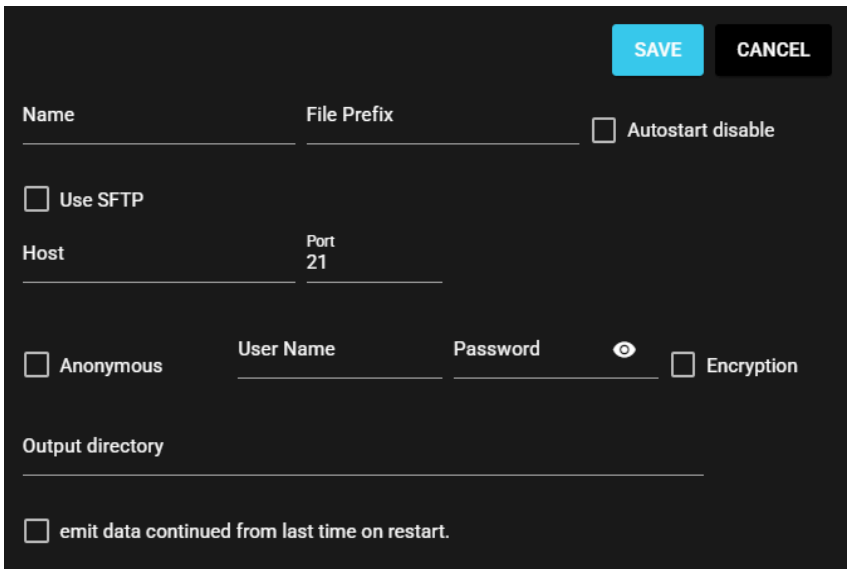
FTP/FTPS/SFTP 에미터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	FTP/FTPS/SFTP 에미터
설명	FTP 또는 SFTP에 의한 데이터 전송을 수행합니다.
소비 포인트	2
포트	입력 : 1 시리얼라이저・컴포넌트로부터, 파일 형식의 데이터를 받아들입니다. 받아들인 데이터는, FTP 또는 SFTP에 의해 전송됩니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
기타	FTP 또는 SFTP 서버에 대한 테스트 접속을 수행할 수 있습니다. 상세는「테스트 접속」을 참조하십시오.

-For Reference Only-

5.6.5.2 설정 항목

FTP/FTPS/SFTP 에미터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
파일 프리픽스	파일명에 부여하는 프리픽스
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
SFTP를 사용한다	SFTP를 사용하는 경우 ON
호스트	송신처 서버의 호스트명이나 IP 주소를 지정
포트	송신처 서버의 포트 번호를 지정
익명 액세스	사용자 인증을 수행하지 않는 경우 ON ※SFTP를 사용한다가 OFF인 경우에만 선택 가능합니다.
사용자명	인증 사용자명을 입력 ※다음의 경우는 입력 가능합니다. ・SFTP를 사용한다가 OFF이고 익명 액세스가 OFF ・SFTP를 사용한다가 ON
패스워드	인증 패스워드를 입력 ※다음의 경우는 입력 가능합니다. ・SFTP를 사용한다가 OFF이고 익명 액세스가 OFF ・SFTP를 사용한다가 ON이고 공개키 인증이 OFF

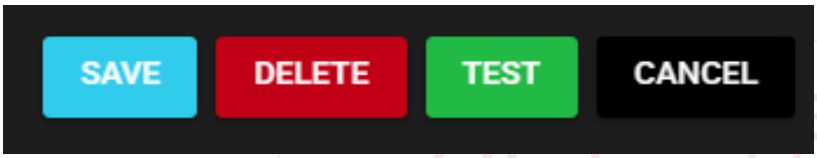
AI 번역본
-For Reference Only-

항목	설명
패스프레이즈	패스프레이즈를 입력 ※SFTP를 사용한다가 ON이고 공개키 인증이 ON인 경우만 입력 가능합니다.
공개키 인증	공개키 인증을 수행하는 경우 ON ※SFTP를 사용한다가 ON인 경우에만 선택 가능합니다.
비밀키	비밀키를 업로드 ※SFTP를 사용한다가 ON이고 공개키 인증이 ON인 경우만 업로드 가능합니다.
암호화	송수신하는 데이터를 암호화하는 경우 ON ※SFTP를 사용한다가 OFF인 경우에만 선택 가능합니다.
출력처 디렉터리	송신처 서버 내의 출력처 디렉터리를 지정
기동 후는 전회 송신하는 데이터의 계속부터 데이터를 출력한다	기동 후는 전회 송신한 데이터의 계속부터 데이터를 출력한다

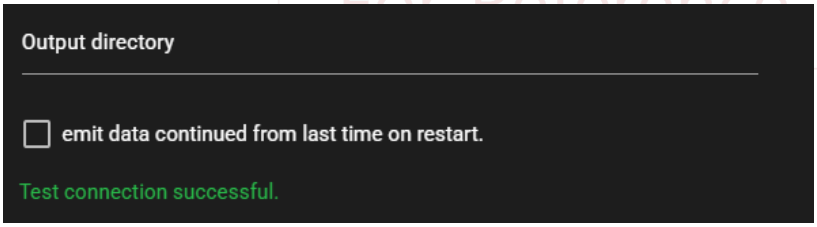
5.6.5.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 FTP/FTPS/SFTP 에미터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



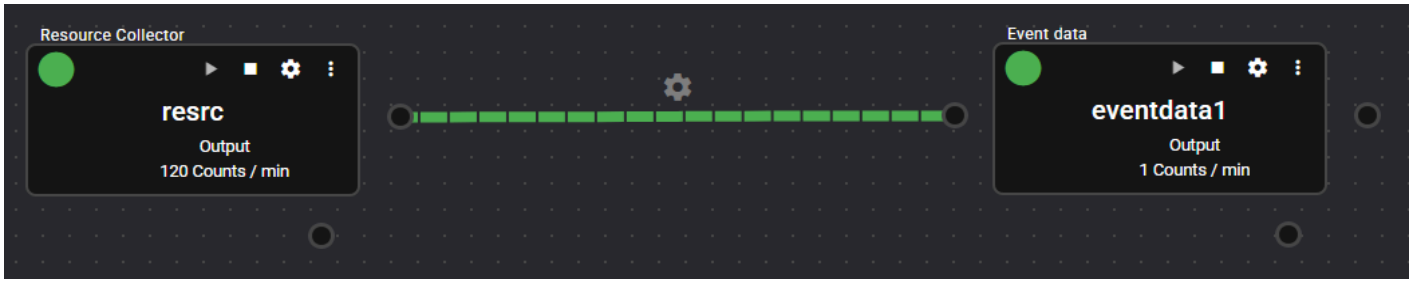
For Reference Only-

5.7 로직（이벤트）

여기에서는, 이벤트에 관한 로직 컴포넌트에 대해 설명합니다.

5.7.1 이벤트 데이터

수집한 데이터를 조건 판정하고, 조건 합치하고 있는 구간의 정보를 출력하는 컴포넌트입니다.
입력 포트의「이벤트를 사용한다」와 조합하여 사용합니다.
입력 포트로부터 데이터를 받아들여, 그 데이터에 대해 그때마다 조건 판정을 수행하고, 조건에 합치하고 있는 구간의 정보를 데이터로서 출력합니다.



5.7.1.1 기본 정보

이벤트 데이터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	이벤트 데이터
설명	수집한 데이터 중, 조건 합치한 것만을, 새롭게 별도 데이터로서 출력합니다. ※별도의 데이터로서 등록하기 때문에, 메모리 사용량 등은 증가합니다. 고샘플링 레이트 데이터를 그대로 다루고 싶은 경우에는 적합하지 않습니다.
소비 포인트	2

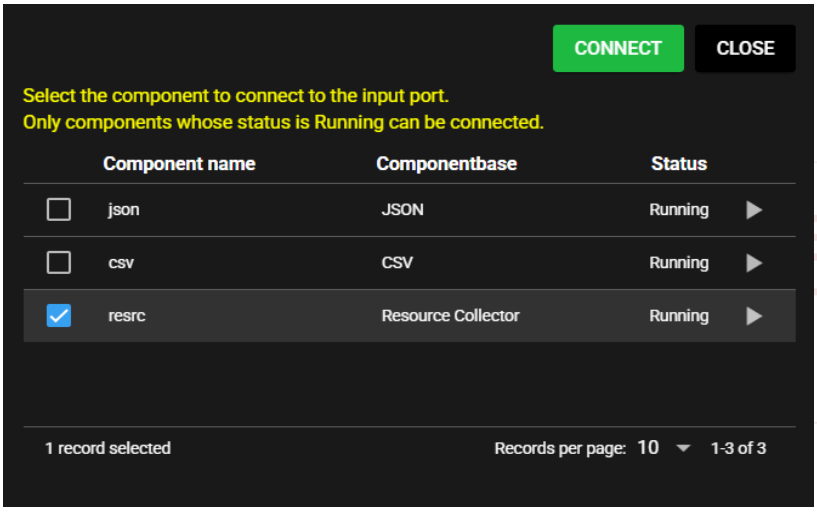
AI 번역본
-For Reference Only-

항목	설명
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 받아들인 데이터 중, 조건 합치한 것을 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	이벤트 데이터를 사용한 설정 예는,「 설정 예 」를 참조하십시오.

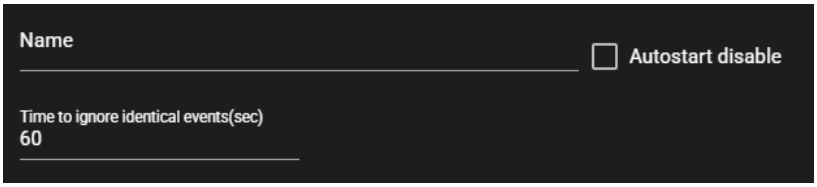
5.7.1.2 설정 방법

다음 순서로, 이벤트 데이터의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
2. 추가한 컴포넌트를 기동하고, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.
3. 이벤트 데이터를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.

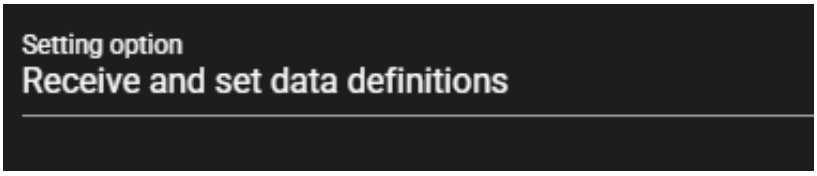


5. 다음 컴포넌트 정보를 입력합니다.



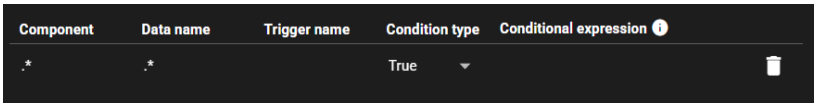
항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
동일 이벤트를 무시하는 시간(초)	동일 이벤트를 무시하는 시간을 입력

6. 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별로 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

7. 다음 항목을 입력하고, 이벤트 조건을 설정합니다.



항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식 ※설정 방법이 「 정규 표현식으로 설정한다 」인 경우만 입력 가능
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식 ※설정 방법이 「 정규 표현식으로 설정한다 」인 경우만 입력 가능
트리거명	트리거의 이름을 입력
조건 종별	True : 조건식이 참이면 트리거 발생 False : 조건식이 거짓이면 트리거 발생

AI 번역본
For Reference Only

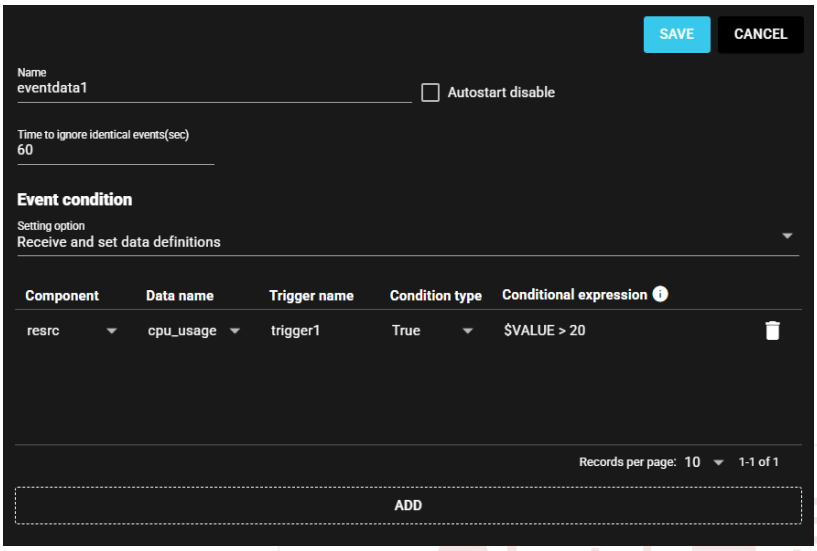
항목	설명
	Fall：조건식이 참→거짓일 때에 트리거 발생 Raise：조건식이 거짓→참일 때에 트리거 발생 Anytime：값의 등록 시에 발생
조건식	트리거의 조건식을 입력 ※조건 종별이 Anytime인 경우 입력 불가

정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

- 「data1」에만 매칭： data1
- 이름에「abc」를 포함하는 데이터에 매칭： .*abc.*
- 이름이「abc」로 시작하는 데이터에 매칭： abc.*
- 이름이「abc」로 끝나는 데이터에 매칭： .*abc

8.「저장」을 누릅니다.



9. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「컴포넌트의 기동/정지」를 참조하십시오.

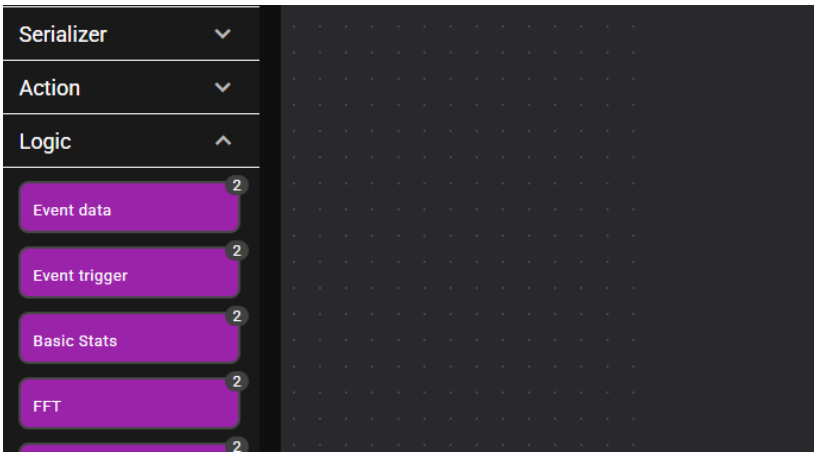
-For Reference Only-

5.7.1.3 설정 예

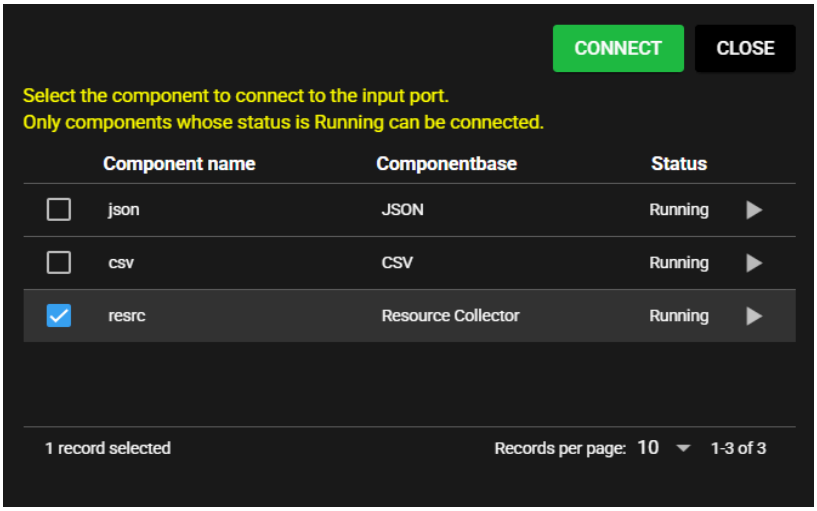
이벤트 데이터의 설정 예는 다음과 같습니다. 여기에서는 먼저, 리소스 컬렉터를 사용하여 CPU 사용률을 수집합니다. 거기에 이벤트 데이터를 사용하여, CPU 사용률이 20%를 초과하는 경우에만 새롭게 별도 데이터로서 등록되도록 합니다.

1. 사전에, 컬렉터를 추가합니다. 여기에서는 "resrc"라는 이름의 리소스 컬렉터를 작성하고, cpu 사용률을 출력하도록 설정합니다.

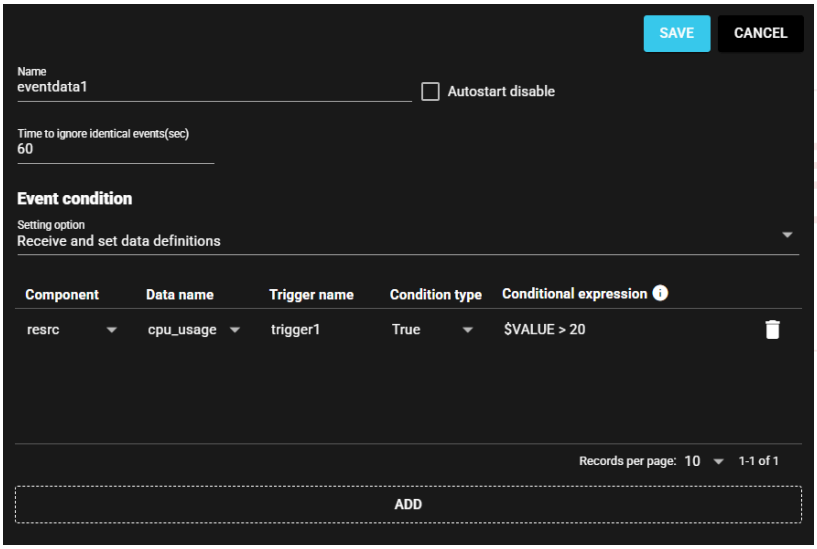
2. 이벤트 데이터의 아이콘을, 패널상으로 드래그 앤 드롭합니다.



3. 입력 포트에 연결할 컴포넌트를 선택합니다. 이번에는, "resrc"라는 이름의 리소스 컬렉터를 선택합니다.



4. 다음 항목을 입력하고, 저장을 누릅니다.



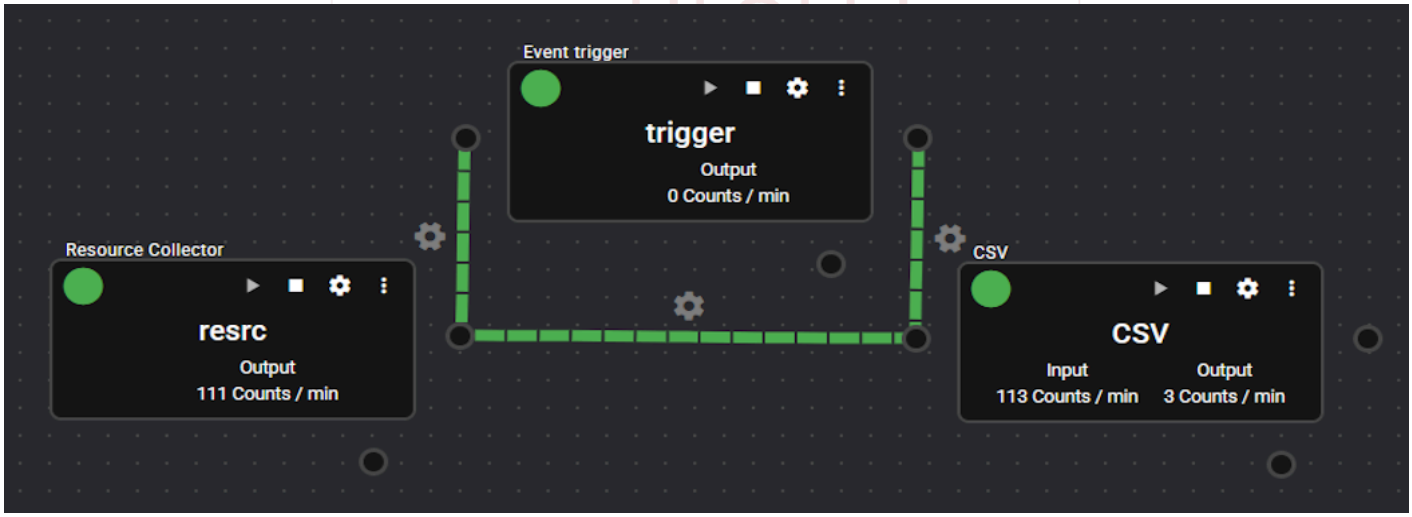
항목	설명
명칭	이벤트 데이터1
자동 기동 무효	OFF
동일 이벤트를 무시하는 시간(초)	60
설정 방법	데이터 정의를 수신하여 설정한다
컴포넌트	resource
데이터명	cpu_usage
트리거명	trigger1
조건 종별	True
조건식	$VALUE > 20

5. 리소스 컬렉터로부터 cpu 사용률을 받아들여, 20%를 초과했을 때, 새롭게 별도 데이터로서 등록됩니다. 등록된 데이터는, 출력 포트로부터 출력됩니다.

5.7.2 이벤트 트리거

수집한 데이터를 조건 판정하고, 조건 합치한 경우만, 새롭게 별도 데이터로서 등록하는 컴포넌트입니다.

입력 포트로부터 데이터를 받아들여, 그 데이터에 대해 그때마다 조건 판정을 수행하고, 조건에 합치하고 있는 경우는 새롭게 별도 데이터로서 출력합니다.



5.7.2.1 기본 정보

이벤트 트리거의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	이벤트 트리거
설명	수집한 데이터를 조건 판정하고, 조건 합치하고 있는 구간의 정보를, 데이터로서 출력합니다. 입력 포트의 「 이벤트를 사용한다 」와 조합하여 사용하는 컴포넌트입니다. 이벤트 중의 데이터를 플 로우 링크처로 전달하려면, 별도로 플로우 링크를 접속할 필요가 있습니다. 원본 데이터를 별도로 등록하는 이벤트 데이터와는 달리, 이벤트 구간의 정보를 등록하기 때문에 메모리 사용량 등은 억제됩니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 받아들인 데이터를 조건 판정하고, 조건에 합치했는지 여부를 진위값으로서 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	이벤트 트리거를 사용한 설정 예는,「 설정 예 」를 참조하십시오.

5.7.2.2 설정 방법

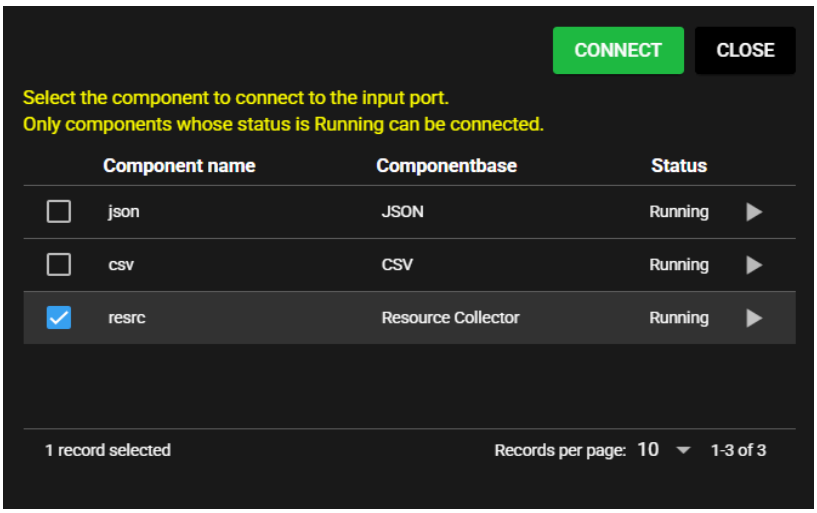
AI 번역본
-For Reference Only-

다음 순서로, 이벤트 트리거의 설정을 수행할 수 있습니다.

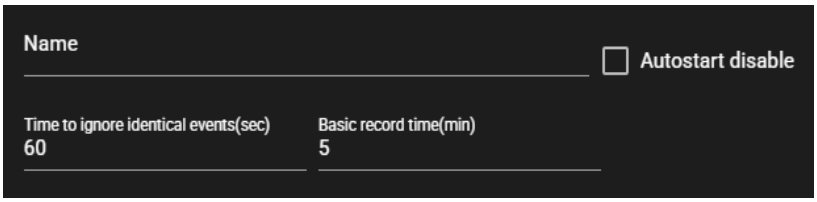
1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

2. 추가한 컴포넌트를 기동하고, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

3. 이벤트 트리거를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.

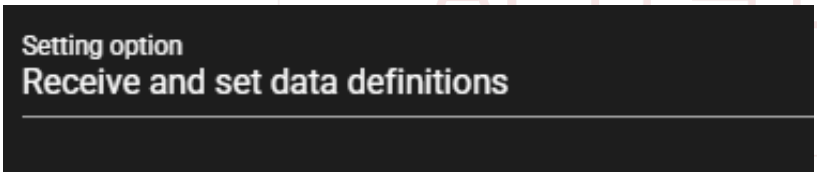


5. 다음 컴포넌트 정보를 입력합니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
동일 이벤트를 무시하는 시간(초)	동일 이벤트를 무시하는 시간을 입력
기본 기록 시간(분)	기본 기록 시간을 분 단위로 입력

6. 개시 이벤트의 「 설정 방법 」을, 다음 중에서 선택합니다.

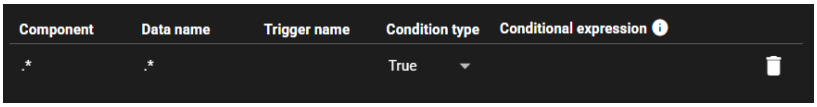


항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별로 설정을 수행한다

AI 번역본
-Only-

항목	설명
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

7. 다음 항목을 입력하고, 개시 이벤트를 설정합니다.



항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식 ※설정 방법이 「정규 표현식으로 설정한다」인 경우만 입력 가능
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식 ※설정 방법이 「정규 표현식으로 설정한다」인 경우만 입력 가능
트리거명	트리거의 이름을 입력
조건 종별	True：조건식이 참이면 트리거 발생 False：조건식이 거짓이면 트리거 발생 Fall：조건식이 참→거짓일 때에 트리거 발생 Raise：조건식이 거짓→참일 때에 트리거 발생 Anytime：값의 등록 시에 발생
조건식	트리거의 조건식을 입력 ※조건 종별이 Anytime인 경우 입력 불가

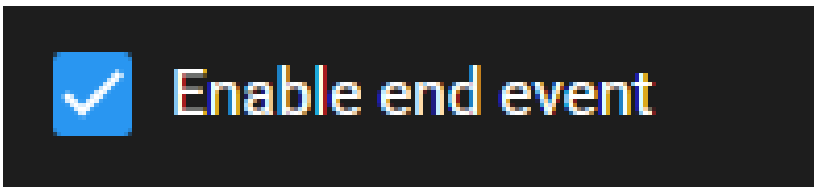
AI 번역본

정규 표현식의 입력 예

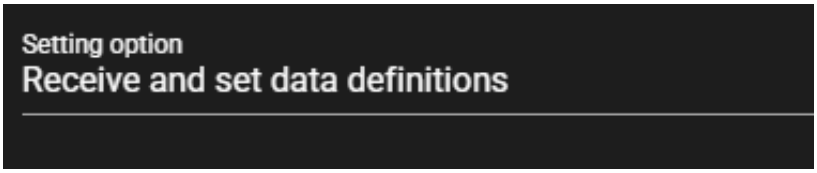
정규 표현식의 입력 예는 다음과 같습니다.

- 「data1」에만 매칭： data1
- 이름에 「abc」를 포함하는 데이터에 매칭： .*abc.*
- 이름이 「abc」로 시작하는 데이터에 매칭： abc.*
- 이름이 「abc」로 끝나는 데이터에 매칭： .*abc

8. 종료 이벤트를 설정하는 경우, 「종료 이벤트」를 ON으로 합니다.

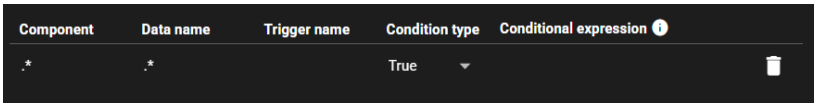


9. 종료 이벤트를 설정하는 경우, 종료 이벤트의 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별로 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

10. 종료 이벤트를 설정하는 경우, 다음 항목을 입력하고, 종료 이벤트를 설정합니다.



항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식 ※설정 방법이 「 정규 표현식으로 설정한다 」인 경우만 입력 가능
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식 ※설정 방법이 「 정규 표현식으로 설정한다 」인 경우만 입력 가능
트리거명	트리거의 이름을 입력
조건 종별	True : 조건식이 참이면 트리거 발생 False : 조건식이 거짓이면 트리거 발생 Fall : 조건식이 참→거짓일 때에 트리거 발생 Raise : 조건식이 거짓→참일 때에 트리거 발생 Anytime : 값의 등록 시에 발생

AI 번역본
-For Reference Only-

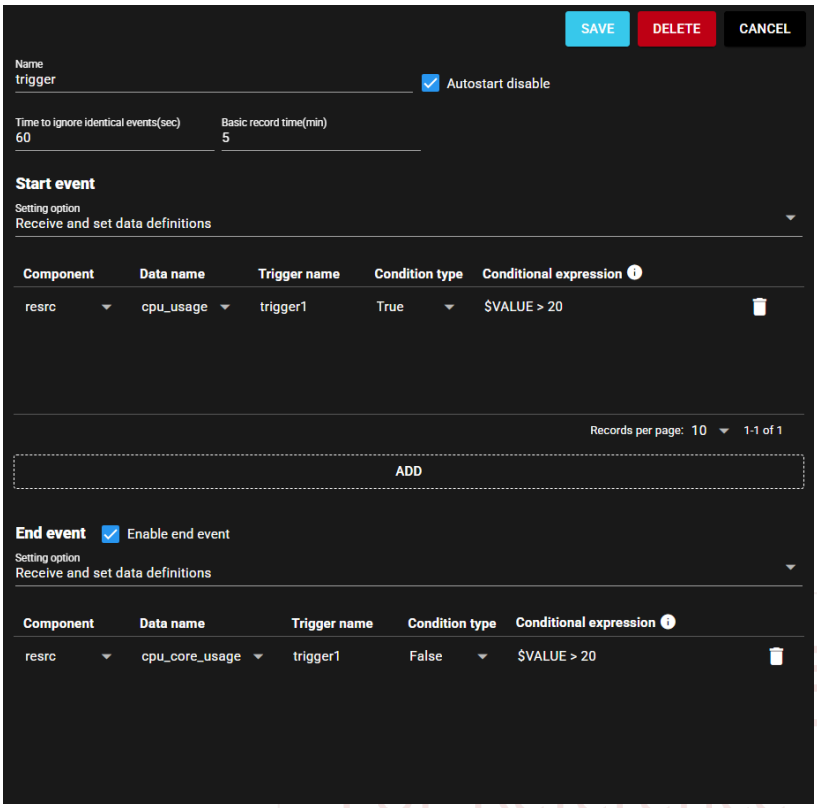
항목	설명
조건식	트리거의 조건식을 입력 ※조건 종별이 Anytime인 경우 입력 불가

정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

- 「data1」에만 매칭 : data1
- 이름에「abc」를 포함하는 데이터에 매칭 : .*abc.*
- 이름이「abc」로 시작하는 데이터에 매칭 : abc.*
- 이름이「abc」로 끝나는 데이터에 매칭 : .*abc

11. 「저장」을 누르고, 설정을 저장합니다.



12. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「컴포넌트의 기동/정지」를 참조하십시오.

13. 이벤트 트리거로 설정한 이벤트를, 다른 컴포넌트가 데이터를 받아들일 때의 조건 판정에 사용하는 것이 가능합니다. 상세는「입력 포트」의「이벤트의 설정」을 참조하십시오.

-For Reference Only-

이벤트 트리거의 이벤트 구간을 사용할 수 없는 컴포넌트

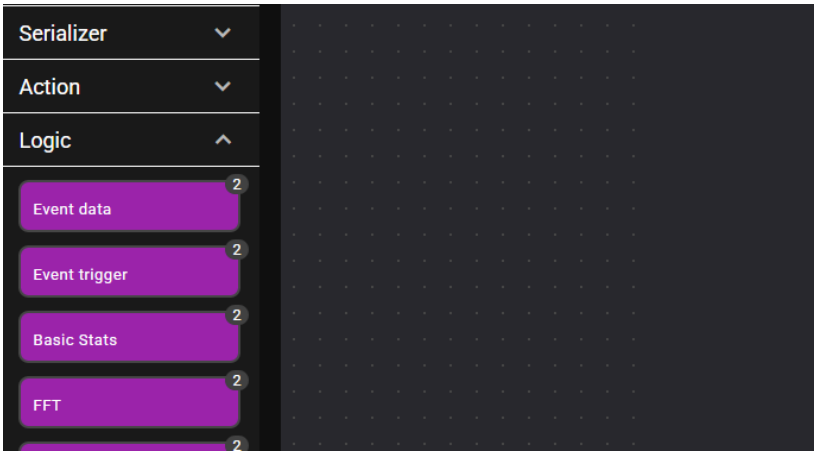
아래 컴포넌트는 이벤트 트리거에 접속 가능(설정 가능)합니다만, 이벤트 구간을 이용할 수는 없습니다.
・로직 컴포넌트 (이벤트 데이터 · 기본 통계 · FFT · 이동 평균)

5.7.2.3 설정 예

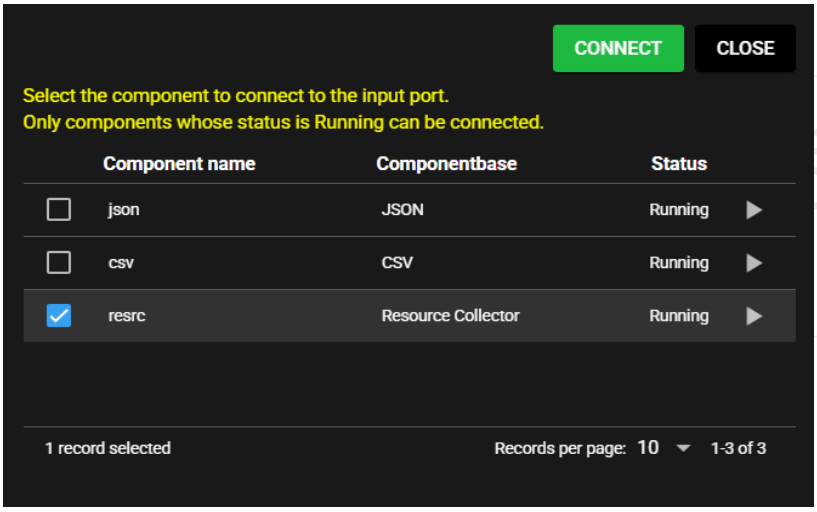
이벤트 트리거의 설정 예는 다음과 같습니다. 여기에서는 먼저, 리소스 컬렉터가 수집한 CPU 사용률을 CSV 시리얼라이저로 가공하도록 설정합니다. 거기에 이벤트 트리거를 사용하여, CPU 사용률이 20%를 초과하는 경우에만 가공이 이루어지도록 합니다.

1. 사전에, 컬렉터를 추가합니다. 여기에서는 "resrc"라는 이름의 리소스 컬렉터를 작성하고, cpu 사용률을 출력하도록 설정합니다.

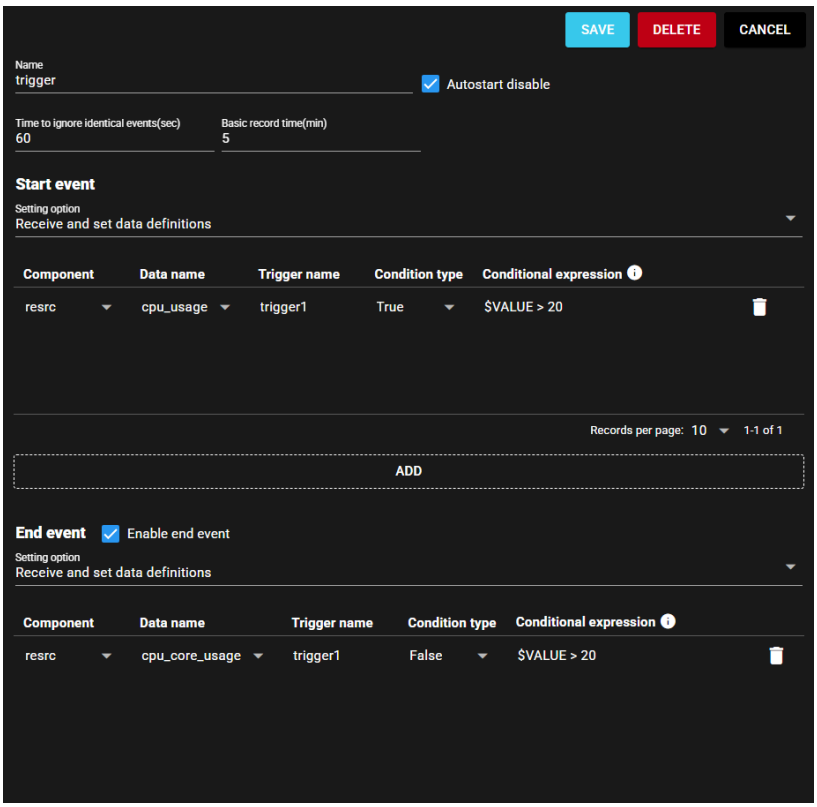
2. 왼쪽 메뉴 안에서 이벤트 트리거의 아이콘을 드래그하여, 패널상으로 드롭합니다.



3. 입력 포트에 연결할 컴포넌트를 선택합니다. 이번에는, "resrc"라는 이름의 리소스 컬렉터를 선택합니다.



4. 다음 항목을 입력하고, 저장을 누릅니다.

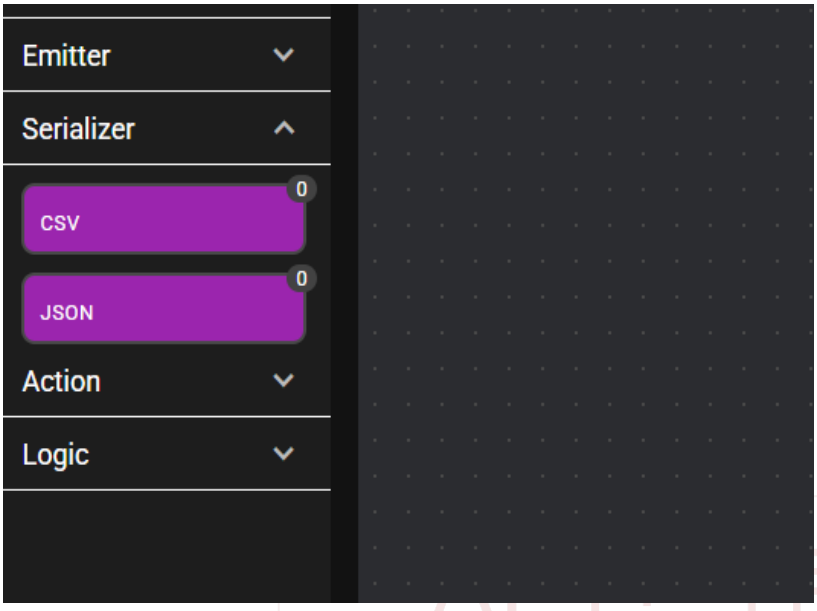


항목	설명
명칭	이벤트 트리거1
자동 기동 무효	OFF
동일 이벤트를 무시하는 시간(초)	60
기본 기록 시간(분)	5
설정 방법	데이터 정의를 수신하여 설정한다
컴포넌트	resource
데이터명	cpu usage
트리거명	trigger1
조건 종별	True
조건식	$VALUE > 20

AI 번역본
-For Reference Only-

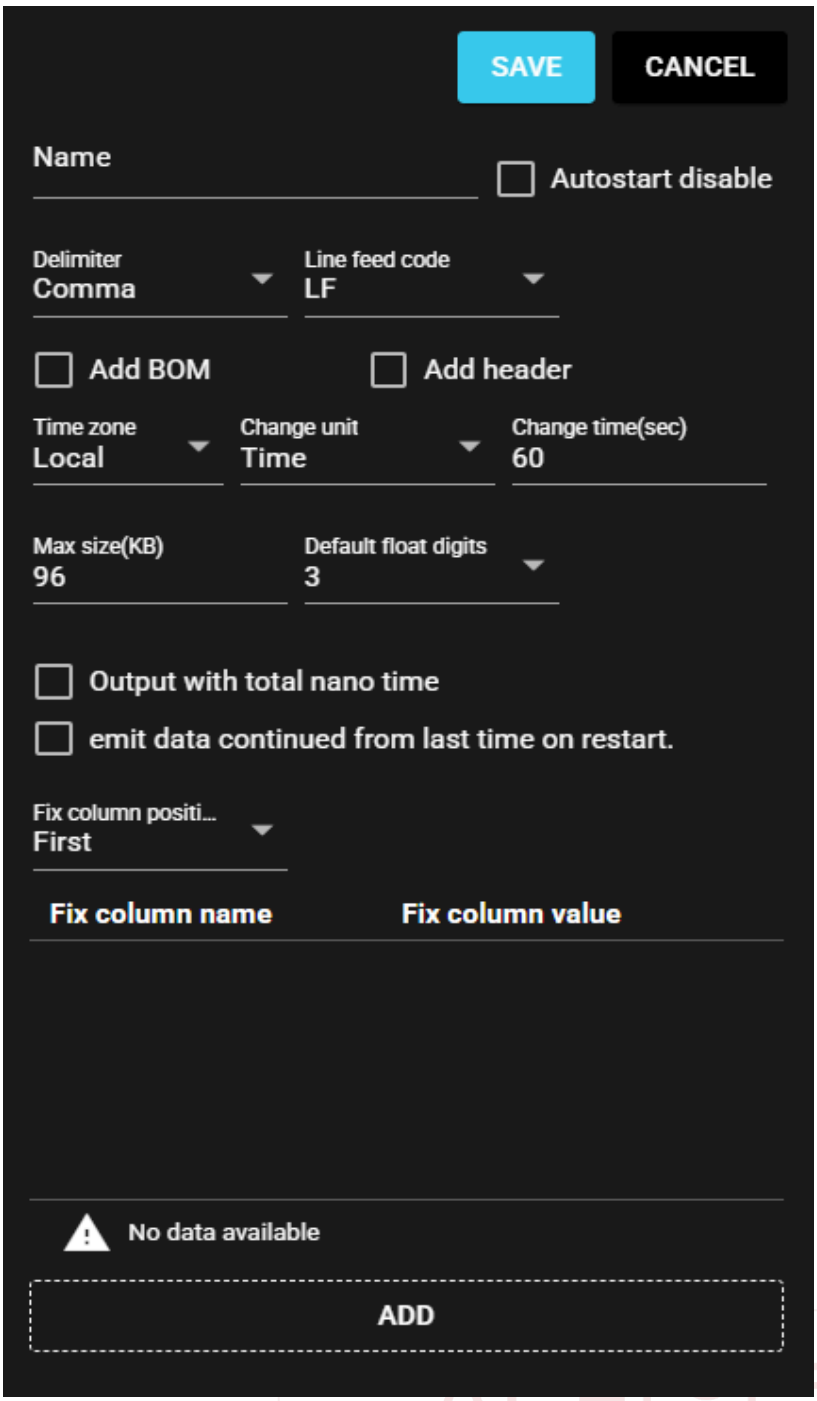
항목	설명
종료 이벤트를 설정한다	ON
설정 방법	데이터 정의를 수신하여 설정한다
컴포넌트	resource
데이터명	cpu_usage
트리거명	trigger1
조건 종별	False
조건식	$VALUE > 20

5. 왼쪽 메뉴 안에서 CSV 시리얼라이저의 아이콘을 드래그하여, 패널상으로 드롭합니다.

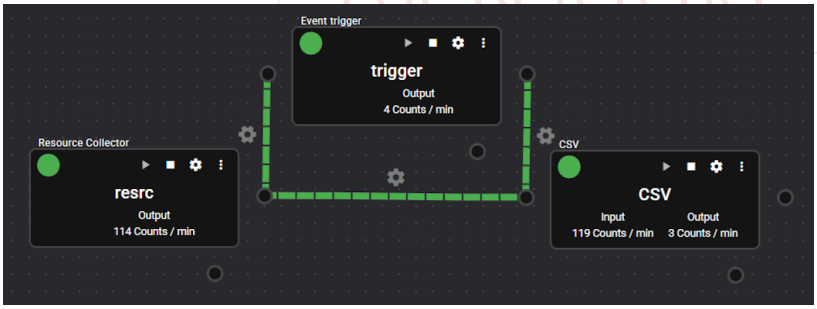


6. 임의의 설정 항목을 입력하고, 저장을 누릅니다.

-For Reference Only-

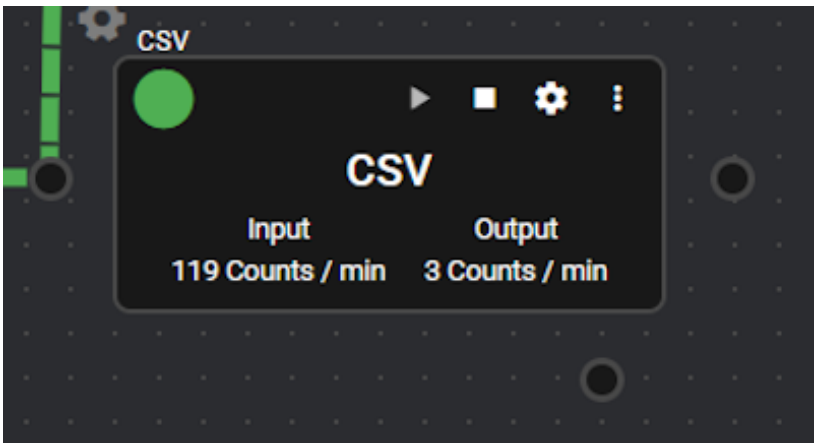


7. 리소스 컬렉터 · 이벤트 트리거의 출력 포트와, CSV 시리얼라이저의 입력 포트를, 플로우 링크로 연결합니다.

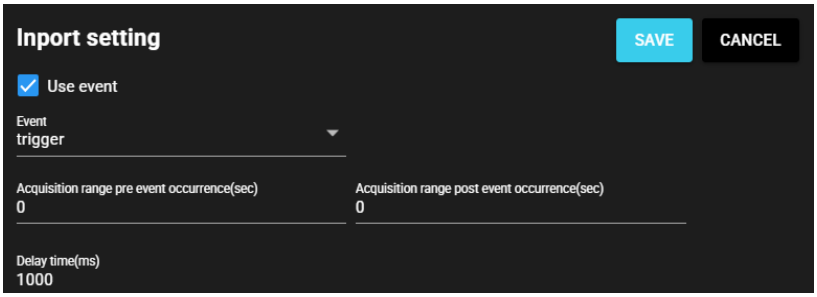


8. CSV 시리얼라이저의 입력 포트를 오른쪽 클릭합니다.

AI 학습용
-For Reference Only-



9. 「 이벤트를 사용한다 」에 체크를 표시하고, 「 이벤트 」에서 "이벤트 트리거1"을 선택하고, 저장을 누릅니다.



10. 이 상태에서, 컴포넌트를 가동시킴으로써, 리소스 컬렉터가 수집한 CPU 사용률이, 20%를 초과한 경우에만, CSV 시리얼라이저에 의해 CSV 파일로 가공됩니다.

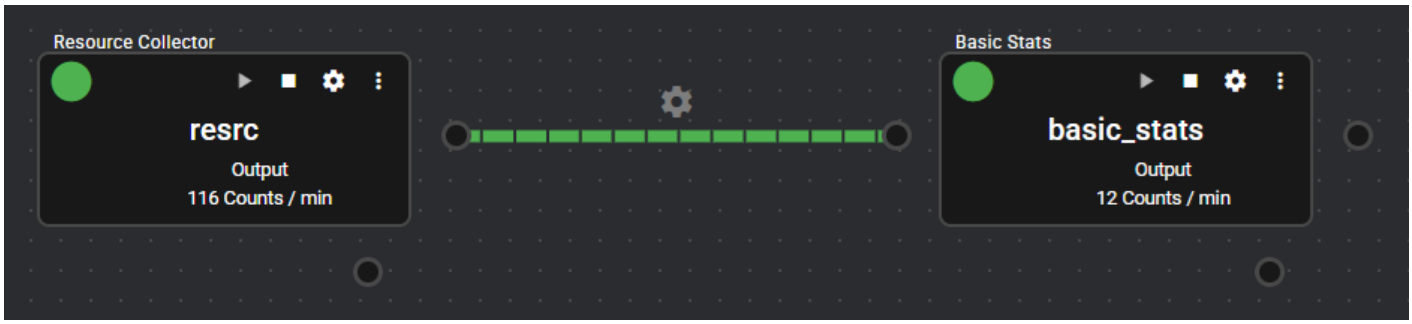
AI 번역본
-For Reference Only-

5.8 로직（실시간 분석/데이터 조작）

여기에서는, 실시간 분석에 관한 로직 컴포넌트에 대해 설명합니다.

5.8.1 기본 통계

수집 데이터에 대한 기본 통계를 수행한 결과를 별도 데이터로서 등록하는 컴포넌트입니다.
입력 포트로부터 받아들인 데이터에 대해 일정 범위（건수/초）로 기본 통계를 수행하고, 그 결과를 별도 데이터로서 출력합니다.



5.8.1.1 기본 정보

기본 통계의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	기본 통계
설명	수집 데이터에 기본 통계를 수행한 결과를, 데이터로서 출력합니다.
소비 포인트	2
포트	입력：1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력：1 「설정 방법」에 의해 설정된 내용을 토대로, 입력 데이터에 기본 통계를 수행한 결과를 출력합니다. 시스템：1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.

AI 번역본
For Reference Only-

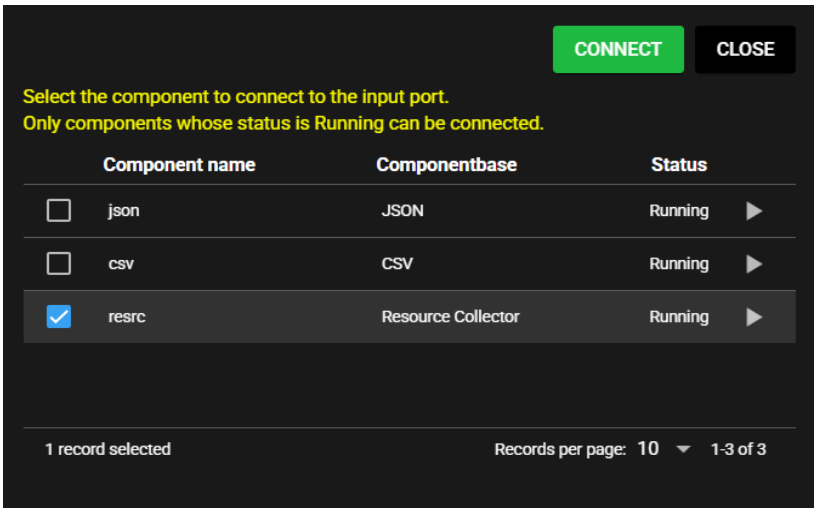
5.8.1.2 설정 방법

다음 순서로, 기본 통계의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

2. 추가한 컴포넌트를 기동하고, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

3. 기본 통계를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.

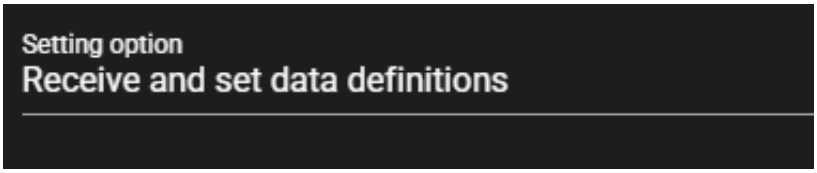


5. 다음 컴포넌트 정보를 입력합니다.



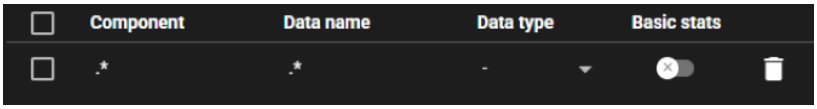
항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON

6.「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별로 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

7. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합다.



항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터형

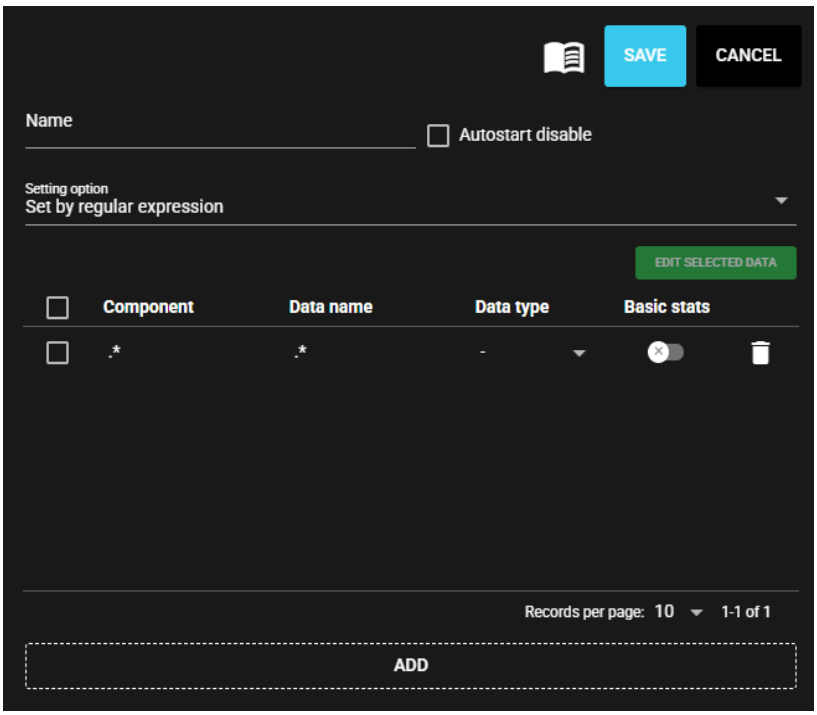
정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

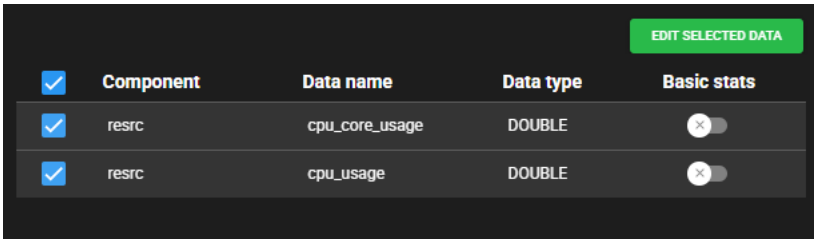
- 「 data1 」에만 매칭 : data1
- 이름에 「 abc 」를 포함하는 데이터에 매칭 : .*abc.*
- 이름이 「 abc 」로 시작하는 데이터에 매칭 : abc.*
- 이름이 「 abc 」로 끝나는 데이터에 매칭 : .*abc

-For Reference Only-

8. 다음 중 어느 하나의 방법을 수행하여, 기본 통계의 설정 다이얼로그를 엽니다.
   - 기본 통계의 설정을 수행할 데이터의 「 기본 통계 」 아이콘을 누릅니다.



- 기본 통계의 설정을 수행할 데이터에 체크를 표시하고, 「 선택 중인 데이터를 편집 」을 누릅니다. 체크를 표시인 데이터에, 같은 설정 내용이 적용됩니다.



9. 다음 항목을 입력하고, 기본 통계의 설정을 수행합니다.

AI 번역본
-For Reference Only-



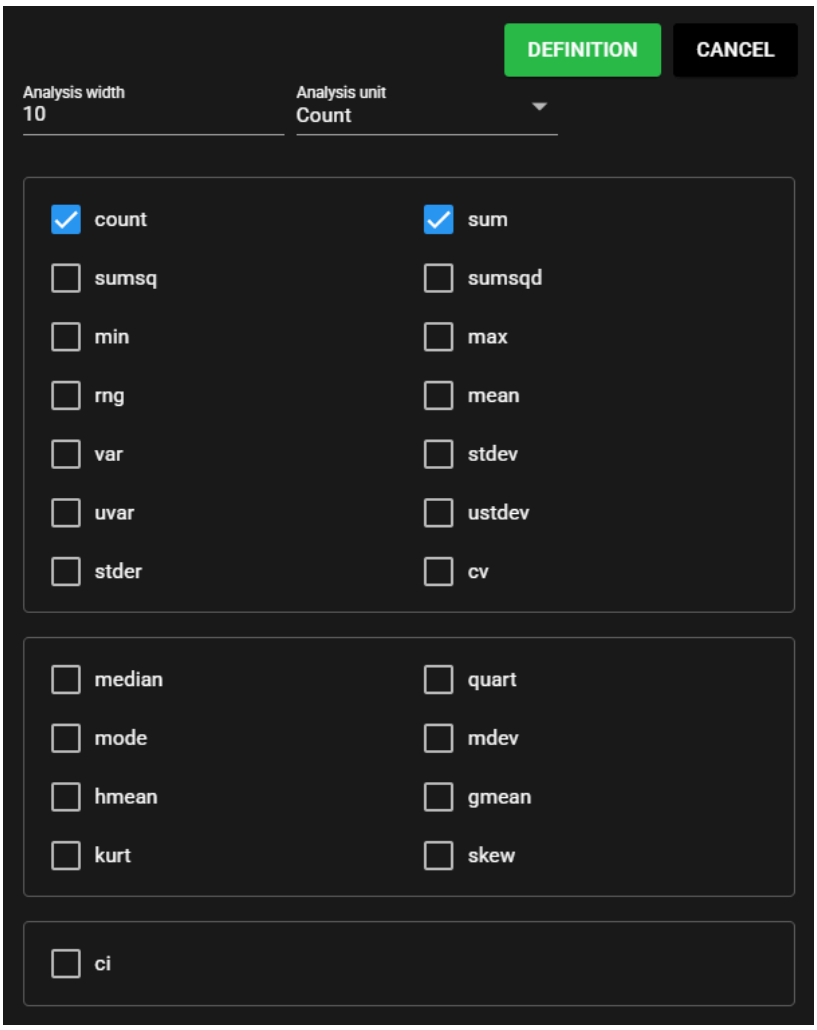
항목	설명
분석 폭	분석의 폭을 설정
분석 단위	분석 폭의 단위를 설정 (건수 or 초)
건수	분석 대상의 데이터 건수
합계	분석 대상의 데이터의 합계값
제곱합	분석 대상의 각 데이터의 제곱을 합계한 값
편차 제곱합	분석 대상의 각 데이터가 평균에서 얼마나 떨어져 있는지의 편차를 제곱하여 합계한 값
최소값	분석 대상의 데이터 내에서 가장 작은 값
최대값	분석 대상의 데이터 내에서 가장 큰 값

AI 번역본
-For Reference Only-

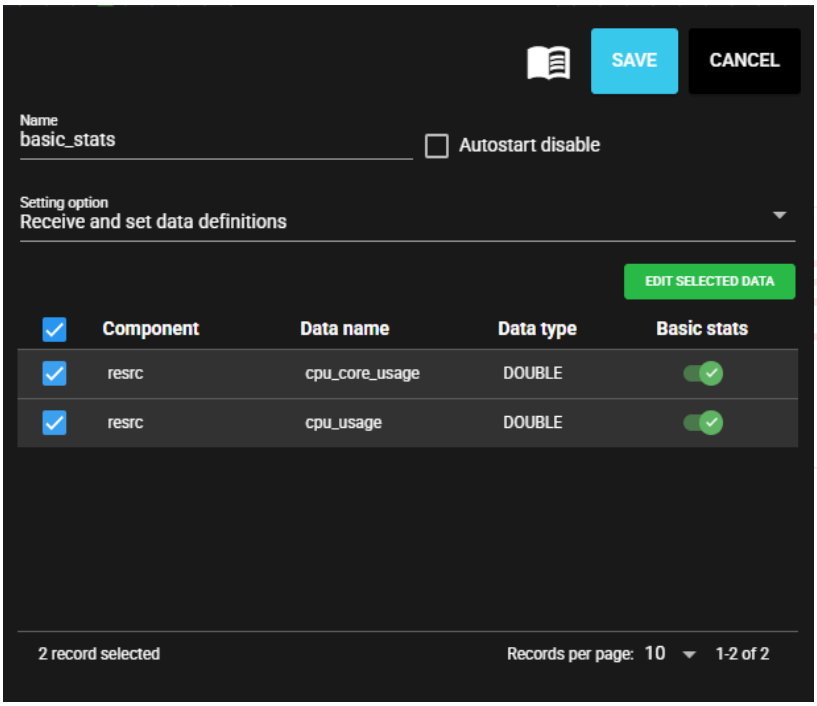
항목	설명
범위	분석 대상의 데이터의 최대값과 최소값의 차
산술평균	분석 대상의 데이터의 평균값
분산	분석 대상의 각 데이터가 평균에서 얼마나 떨어져 있는지의 평균의 제곱
표준편차	분산의 제곱근
불편분산	분산을 건수-1로 나눈 값
표본 불편 표준편차	표본의 표준편차
표준오차	표본 평균의 표준편차
변동계수	표준편차를 평균으로 나눈 값
중앙값	분석 대상의 데이터를 작은 순으로 나열하여, 중앙에 있는 값
사분위수	분석 대상의 데이터를 작은 순으로 나열하여, 4등분한 위치에 있는 값
최빈값	분석 대상의 데이터 내에서 가장 빈번하게 출현하는 값
평균편차	분석 대상의 각 데이터가 평균에서 얼마나 떨어져 있는지의 평균
조화평균	분석 대상의 각 데이터의 역수의 평균의 역수
기하평균	분석 대상의 각 데이터의 곱의 N제곱근 ※N은 건수
첨도	분포의 뾰족한 정도
왜도	분포의 왜곡
신뢰구간	분석 대상의 데이터의 평균값이 포함된다고 신뢰할 수 있는 수치의 범위

AI 번역본
-For Reference Only-

10.「확정」을 누릅니다.



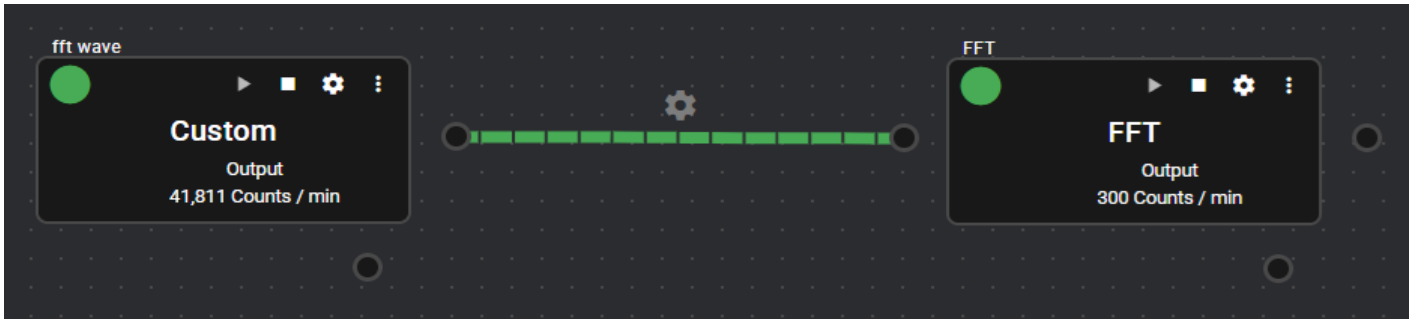
11. 「 저장 」을 누릅니다.



12. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

5.8.2 FFT

수집 데이터를 FFT한 결과를 데이터로서 등록하는 컴포넌트입니다.
입력 포트로부터 데이터를 받아들여, 그 데이터에 FFT 해석을 수행한 결과를 별도 데이터로서 출력합니다.
이 컴포넌트는, 샘플링 레이트가 정의되어 있는 컬럼 타입 미들의 데이터에 대해서만 이용할 수 있니다.



> 컬럼 타입 미들의 데이터에 대해
>
> 표준 기능의 컴포넌트에서는, 샘플링 레이트가 정의되어 있는 컬럼 타입 미들을 사용하는 컴포넌트는 존재하지 않습니다. 컬럼 타입 미들（샘플링 레이트가 정주기）의 데이터를 다루는 커스텀 컴포넌트를
> 개발하여 사용합니다.

5.8.2.1 기본 정보

FFT의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	FFT
설명	수집 데이터에 FFT 해석을 수행한 결과를, 데이터로서 출력합니다.
소비 포인트	2
포트	입력：1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력：1 「설정 방법」에 의해 설정된 내용을 토대로, 입력 데이터에 FFT 해석을 수행한 결과를 출력합니다. 시스템：1

AI 번역본
-For Reference Only-

항목	설명
	컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

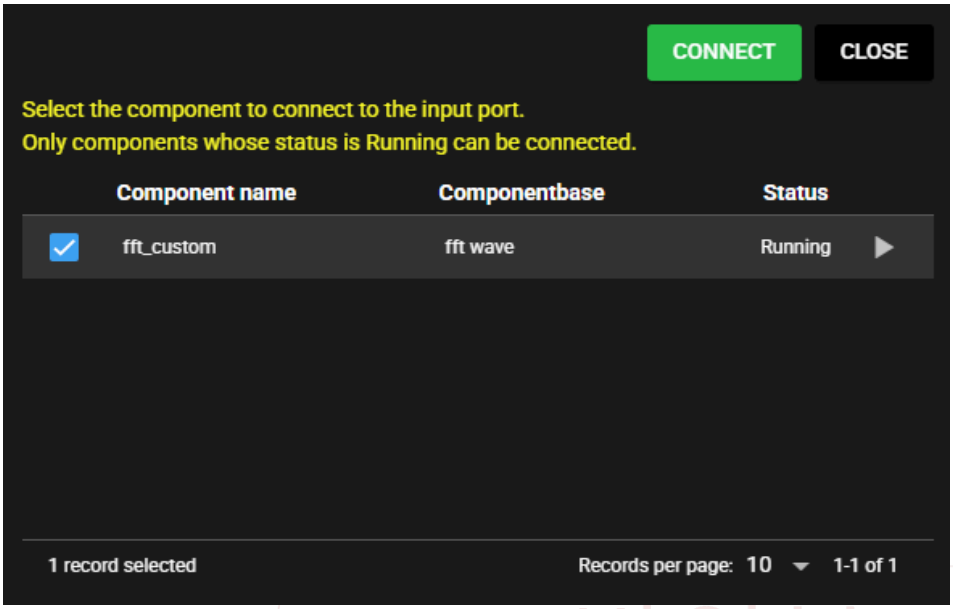
5.8.2.2 설정 방법

다음 순서로, FFT의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

2. 추가한 컴포넌트를 기동하고, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

3. FFT를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.



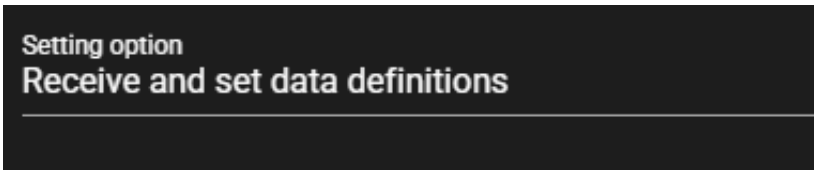
5. 다음 컴포넌트 정보를 입력합니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON

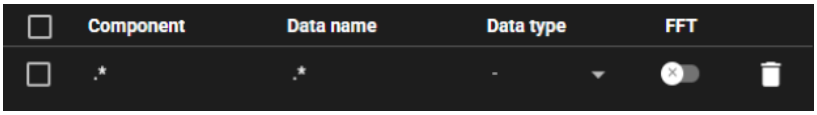
AI 번역본
-Only-

6.「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별로 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

7.「 설정 방법 」이「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합다.



항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터형

AI 번역본

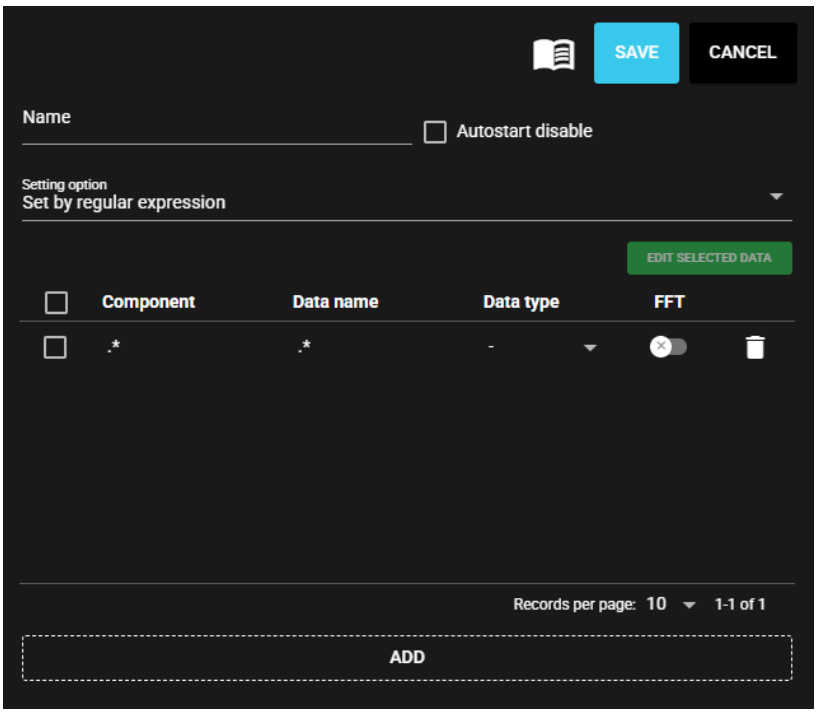
정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

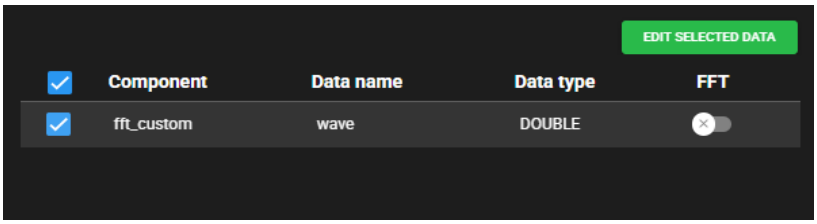
- 「 data1 」에만 매칭 ： data1
- 이름에「 abc 」를 포함하는 데이터에 매칭 ： .*abc.*
- 이름이「 abc 」로 시작하는 데이터에 매칭 ： abc.*
- 이름이「 abc 」로 끝나는 데이터에 매칭 ： .*abc

8. 다음 중 어느 하나의 방법을 수행하여, FFT의 설정 다이얼로그를 엽니다.

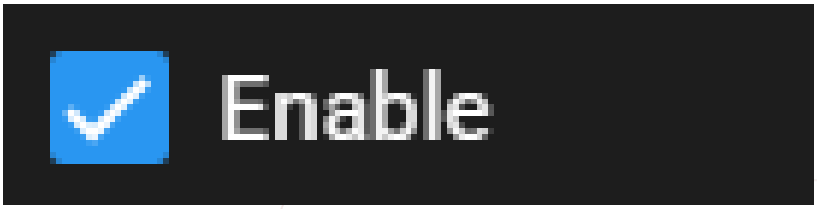
- FFT의 설정을 수행할 데이터의「 FFT 」아이콘을 누릅니다.



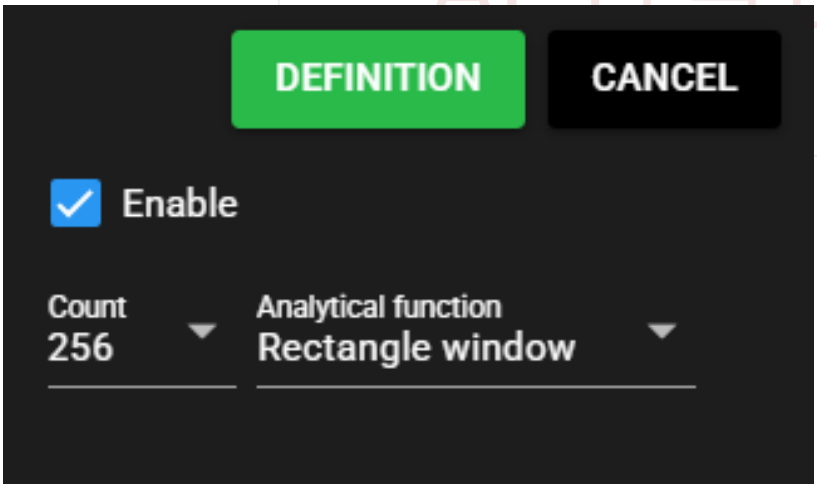
◦ FFT의 설정을 수행할 데이터에 체크를 붙이고,「 선택 중인 데이터를 편집 」을 누릅니다. 체크를 붙인 데이터에, 같은 설정 내용이 적용됩니다.



9.「 FFT 해석을 수행한다 」를 ON으로 합니다.



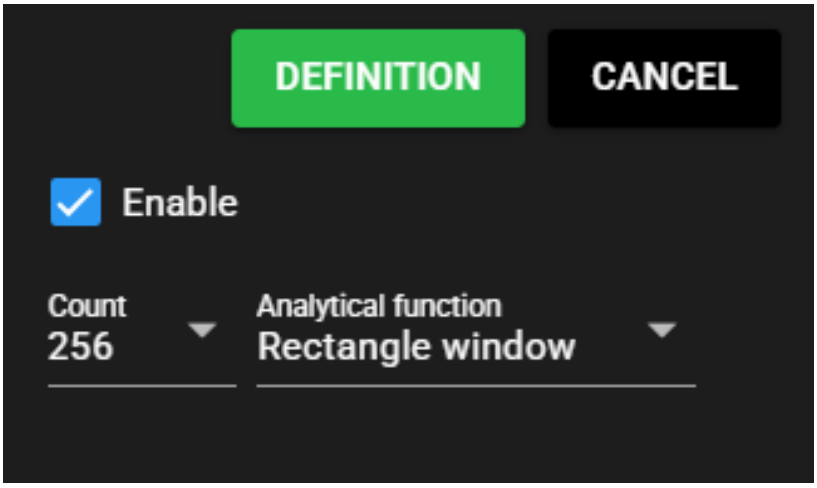
10. 다음 항목을 입력하고, FFT의 설정을 수행합니다.



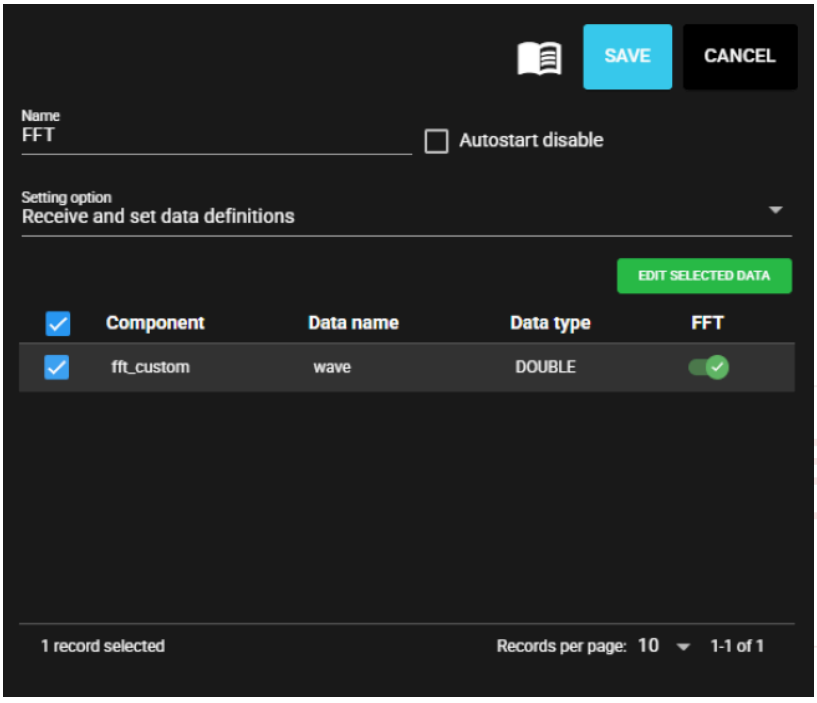
AI 번역본
-Only-

항목	설명
건수	FFT를 실행할 데이터의 건수
윈도우 함수	FFT를 실행하기 전에 데이터에 대해 적용되는 함수

11. 「 확정 」을 누릅니다.



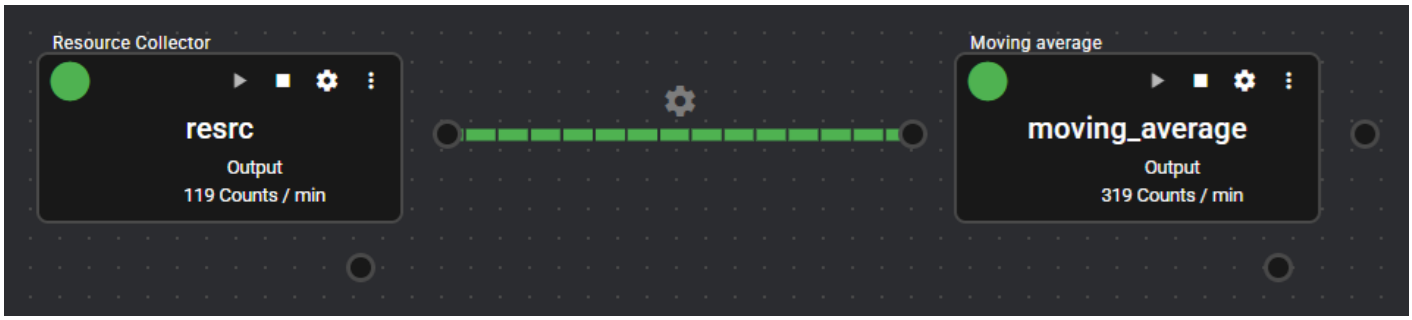
12. 「 저장 」을 누릅니다.



13. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

5.8.3 이동 평균

수집 데이터의 이동 평균값을 데이터로서 등록하는 컴포넌트입니다.
입력 포트에서 데이터를 받아들여, 그 데이터로부터 기간 내의 이동 평균값을 계산하여 별도 데이터로서 출력합니다.



5.8.3.1 기본 정보

이동 평균의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	이동 평균
설명	수집 데이터의 이동 평균값을, 데이터로서 출력합니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 「 설정 방법 」에 의해 설정된 내용을 바탕으로, 입력 데이터의 이동 평균값을 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

-For Reference Only-

5.8.3.2 설정 방법

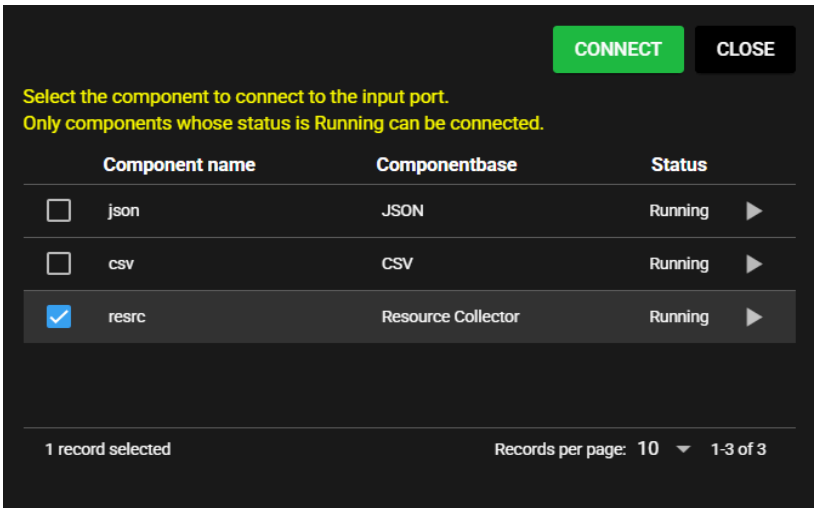
다음 순서로, 이동 평균의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

2. 추가한 컴포넌트를 기동하여, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

3. 이동 평균을 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.

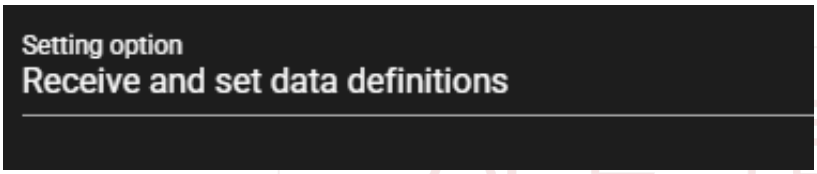


5. 다음 컴포넌트 정보를 입력합니다.



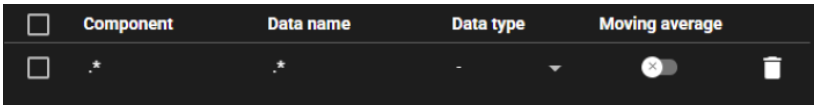
항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON

6.「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별적으로 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해 설정을 수행한다

7. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.

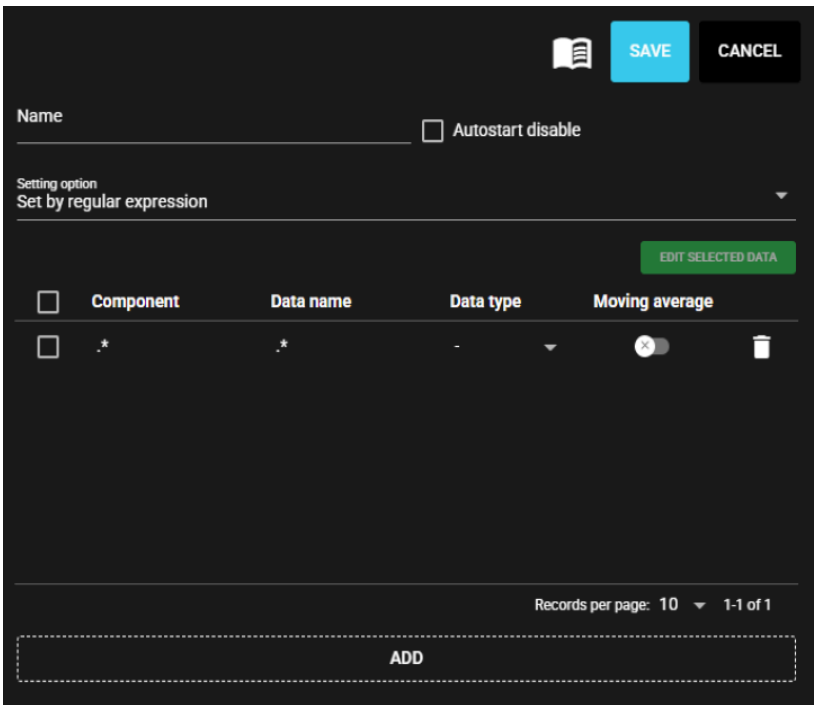


항목	설명
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터형

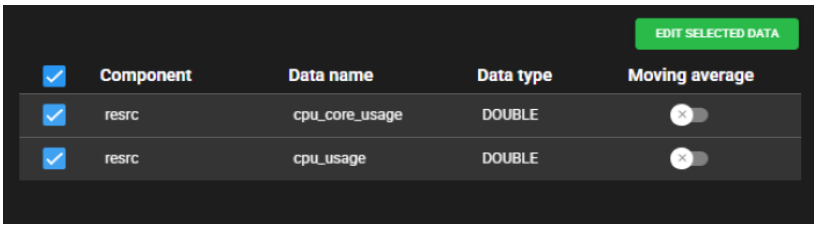
> 정규 표현식의 입력 예
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「 data1 」에만 매칭 : data1
> - 이름에 「 abc 」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이 「 abc 」로 시작하는 데이터에 매칭 : abc.*
> - 이름이 「 abc 」로 끝나는 데이터에 매칭 : .*abc

8. 다음 중 어느 하나의 방법을 수행하여, 이동 평균의 설정 다이얼로그를 엽니다.
   - 이동 평균의 설정을 수행할 데이터의 「 이동 평균 」의 아이콘을 누릅니다.

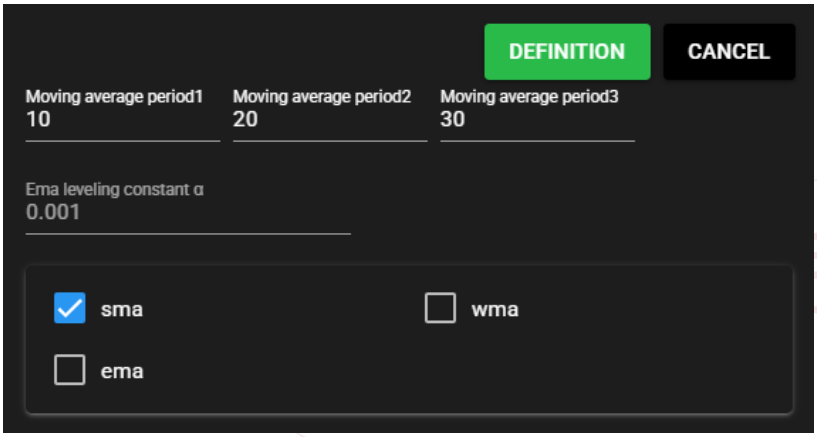
AI 번역본
-For Reference Only-



- 이동 평균의 설정을 수행할 데이터에 체크를 붙이고,「 선택 중인 데이터를 편집 」을 누릅니다. 체크를 붙인 데이터에, 같은 설정 내용이 적용됩니다.



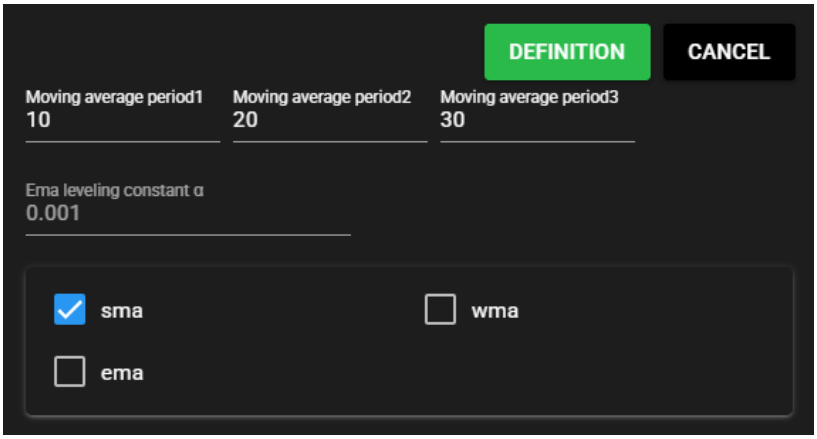
9. 다음 항목을 입력하고, 이동 평균의 설정을 수행합니다.



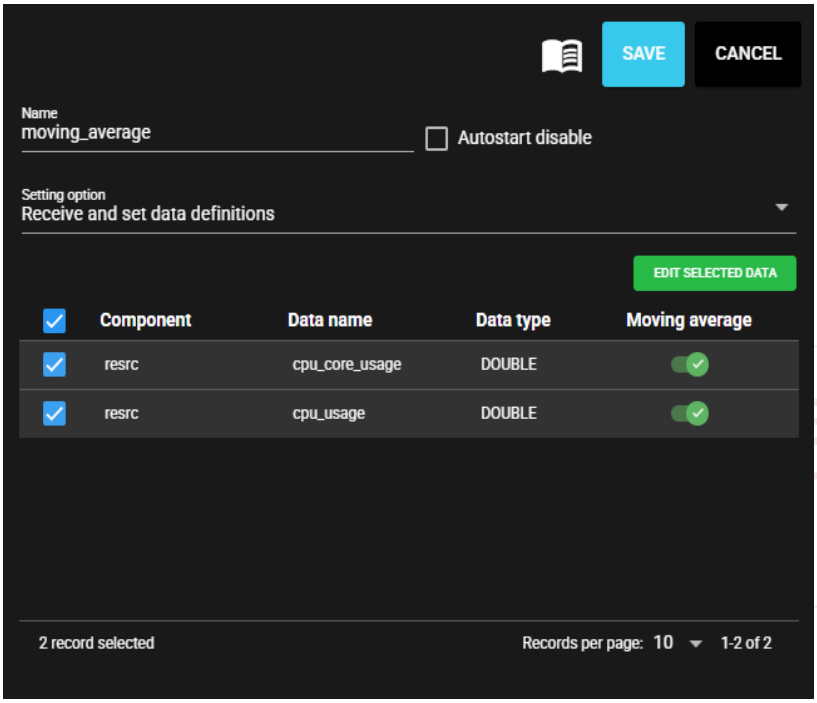
항목	설명
이동 평균값 기간1~3	계산 대상으로 하는 기간
지수 가중 이동 평균값의 표준화 상수α	지수 가중 이동 평균값의 밸런스를 잡기 위한 계수 ※지수 가중 이동 평균값이 ON인 경우만 입력 가능

항목	설명
단순 이동 평균값	기간 내의 데이터의 단순한 평균값
선형 가중 이동 평균값	기간 내의 각 데이터에 대해 선형인 가중치를 곱하여 계산한 평균값
지수 가중 이동 평균값	전 기간의 데이터에 대해 지수적인 가중치를 곱하여 계산한 평균값

10. 「 확정 」을 누릅니다.



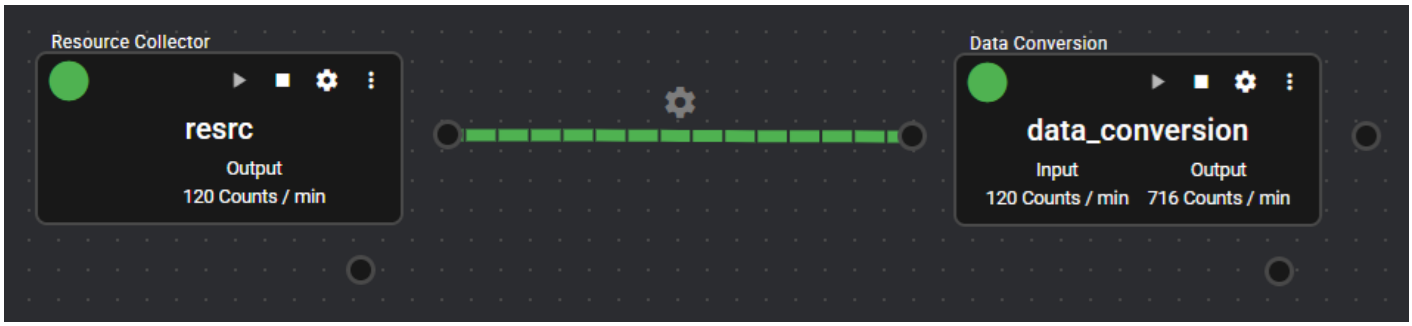
11. 「 저장 」을 누릅니다.



12. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

5.8.4 데이터 변환

입력 포트의 컬럼으로부터 다양한 형의 데이터를 취득하고, 그것에 대해 각종 변환을 통해 출력 포트의 컬럼에 출력하는 컴포넌트입니다.



현재 제공하고 있는 데이터 변환 기능은 아래와 같습니다.

변환 기능	설명
배열 분해	배열 컬럼의 데이터를 개개의 요소로 분해하여 개별 컬럼에 등록합니다. 분해 후의 각 컬럼의 명칭은 디폴트로 자동 설정됩니다. [대상 컬럼] INT8/INT16/INT32/INT64 배열 UINT8/UINT16/UINT32/UINT64 배열 BOOLEAN/FLOAT/DOUBLE 배열
비트 변환	컬럼의 데이터의 bitN~bitM까지를 정수화하여 컬럼에 등록합니다. 정수형 스칼라인 경우, 그 정수의 bitN~bitM을 그대로 다룹니다. bool형 배열인 경우, 배열의 각 요소[N]~[M]을 bitN~bitM으로서 다룹니다 INT8/UINT8형 배열인 경우, 각 요소는 8bit 있지만 0 or !0으로 bit화하고, 각 요소[N]~[M]을 bitN~bitM으로서 다룹니다 [대상 컬럼] 정수형 스칼라 BOOLEAN형 배열 INT8/UINT8형 배열 (단 각 요소는 0 or 1로서 다룬다)
데이터명 변경	입력 데이터를 그대로 별도 데이터로서 명칭을 설정하여 컬럼에 등록합니다. [대상 컬럼] 입력 데이터의 제한 없음

AI 번역본
-For Reference Only-

5.8.4.1 기본 정보

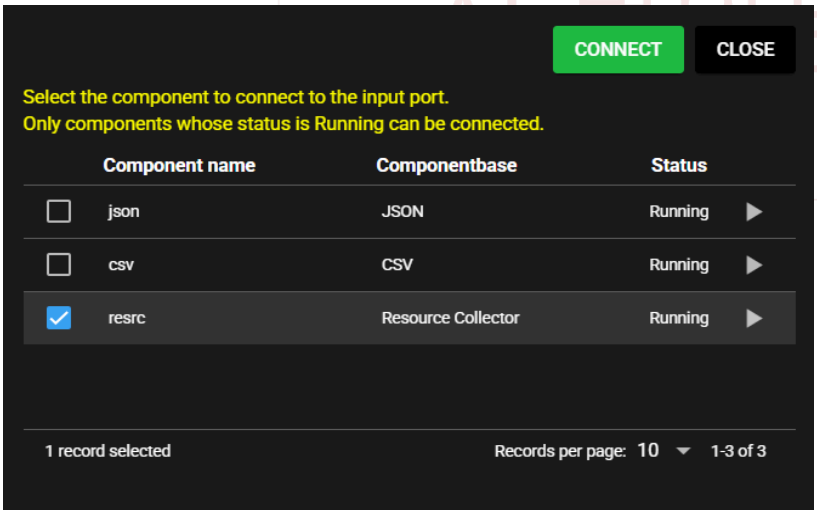
데이터 변환의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	데이터 변환
설명	다양한 형의 데이터를 취득하고, 각종 변환을 통해 출력합니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 1 「 설정 방법 」에 의해 설정된 내용을 바탕으로, 입력 데이터의 이동 평균값을 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

5.8.4.2 설정 방법(공통)

다음 순서로, 데이터 변환의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
2. 추가한 컴포넌트를 기동하여, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.
3. 데이터 변환을 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.

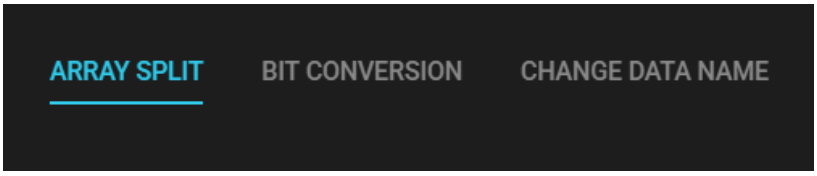


5. 다음 컴포넌트 정보를 입력합니다.



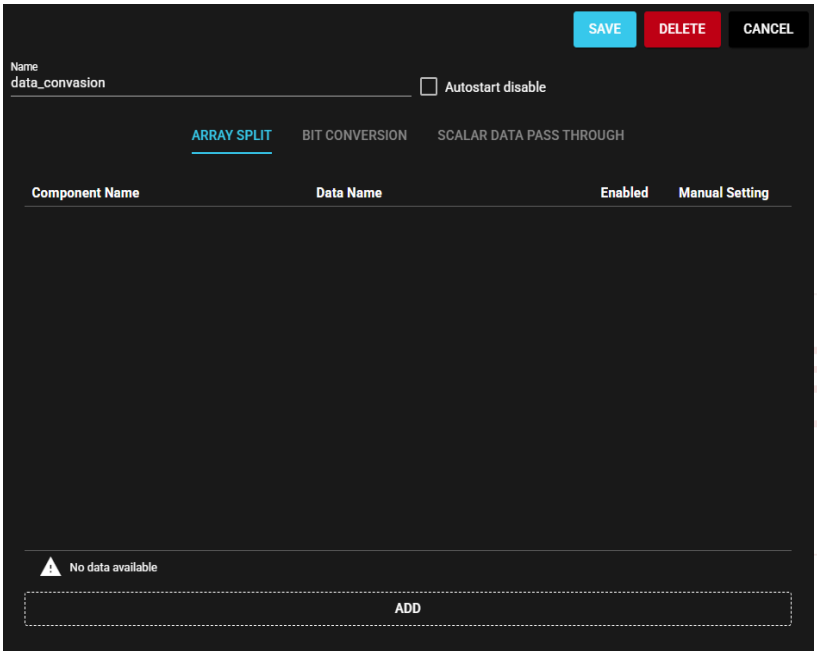
항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON

6. 사용할 데이터 변환 기능을 선택합니다.

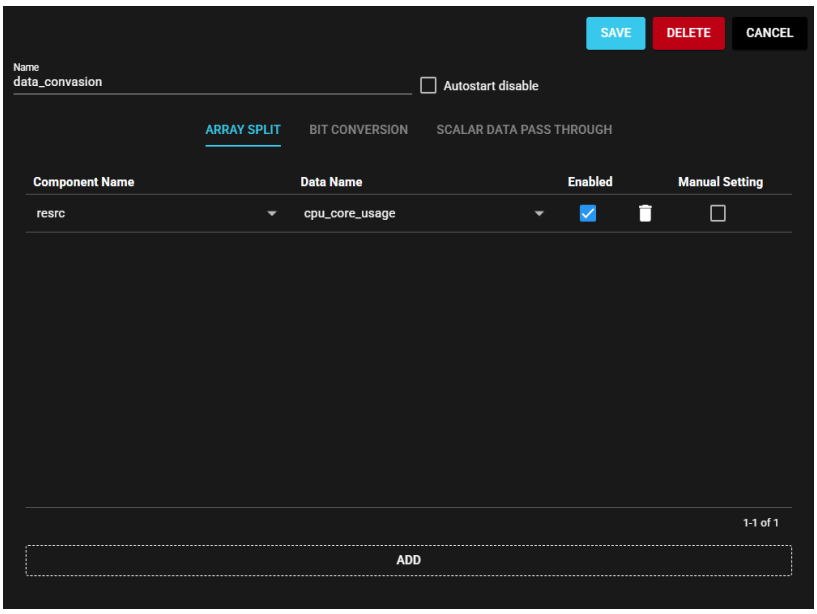


5.8.4.3 설정 방법(배열 분해)

1. 추가 버튼을 눌러 입력 데이터 란을 추가한다.



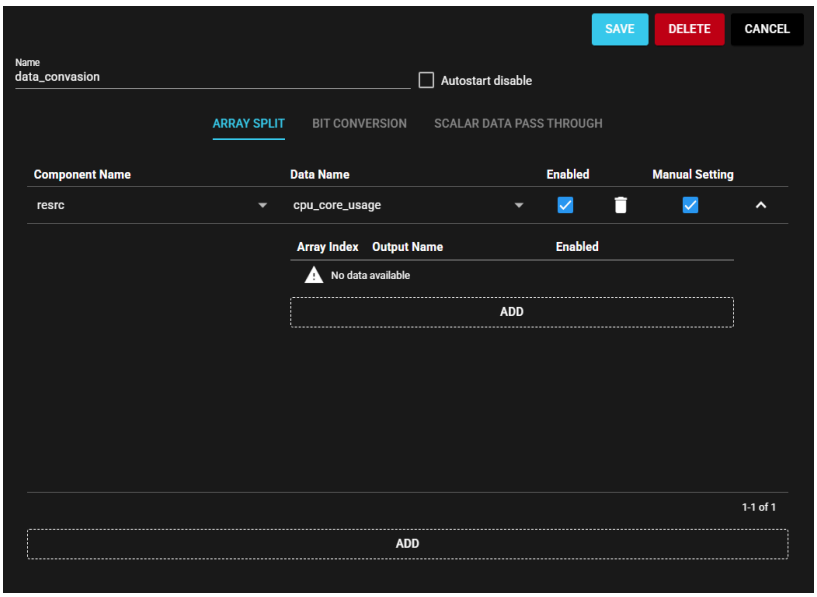
2. 입력할 컴포넌트명과 데이터명을 입력한다.
(드롭다운 선택 또는 수동으로 입력)



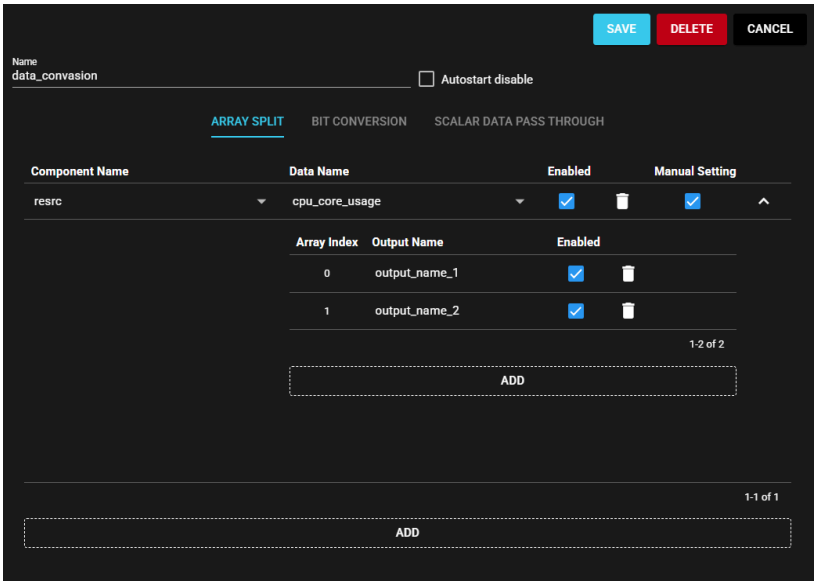
항목	설명
컴포넌트명	입력 포트에 접속한 컴포넌트명을 설정
데이터명	입력 포트에 접속한 컴포넌트명의 데이터명을 설정 (배열 컬럼만 설정 가능)
유효	설정한 입력 데이터 설정을 무효로 할 때는 OFF로 하십시오
삭제	설정한 입력 데이터 설정을 삭제할 때에 누르십시오
수동 설정	배열 분해 후의 출력 데이터명을 수동 설정하고 싶을 때는 ON으로 하십시오 (ON으로 하면 출력 데이터 설정의 전개 버튼이 유효하게 됩니다)

3. (임의)추가 버튼을 눌러 출력 데이터 란을 추가한다.
   (수동 설정을 ON으로 하여 출력 데이터 설정의 전개 버튼을 누르면 추가 버튼이 표시됩니다)

AI 번역본
-For Reference Only-



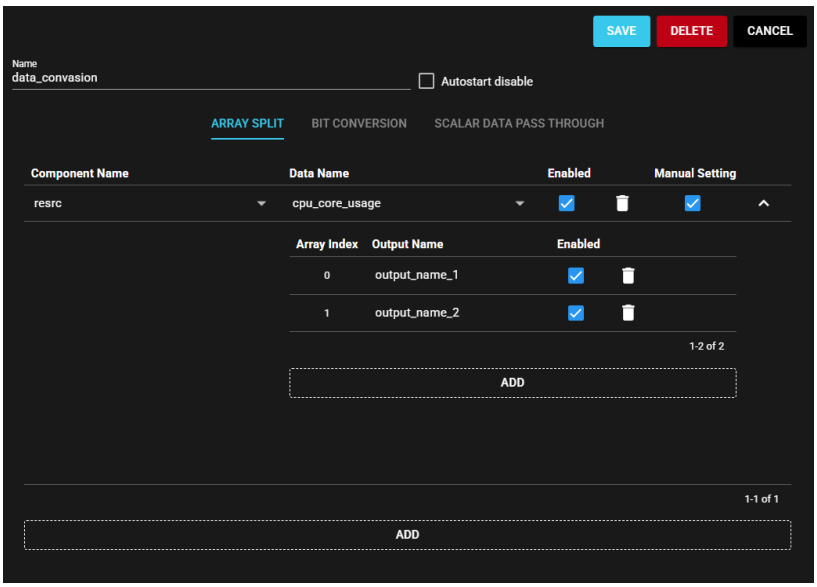
4. (임의)출력할 데이터명을 입력한다.



항목	설명
배열 번호	배열 번호가 자동 채번되어 표시됩니다 (편집 불가)
출력 데이터명	출력 데이터명을 설정
유효	설정한 출력 데이터 설정을 무효로 할 때는 OFF로 하십시오
삭제	설정한 출력 데이터 설정을 삭제할 때에 누르십시오

AI 번역본
-For Reference Only-

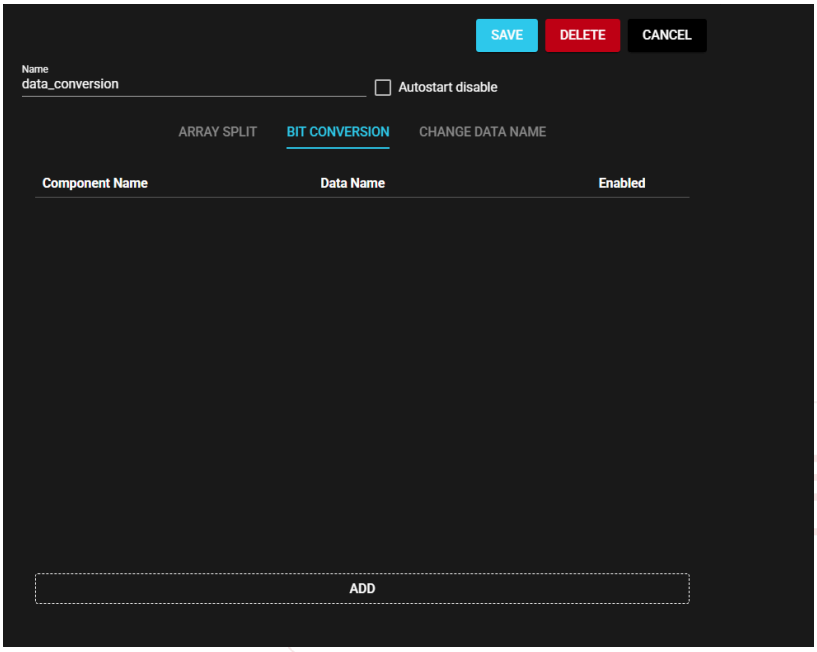
5. 「 저장 」을 누릅니다.



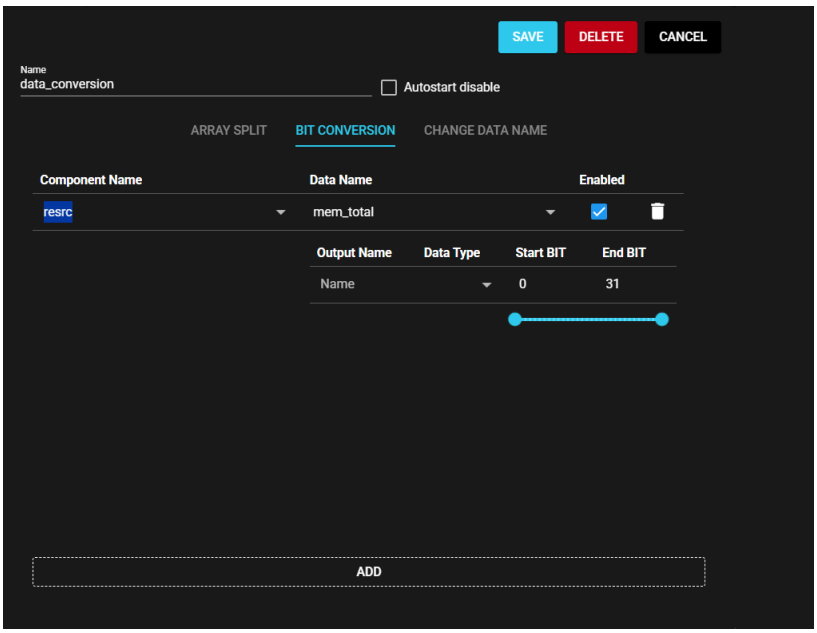
6. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

5.8.4.4 설정 방법(비트 변환)

1. 추가 버튼을 눌러 입력 데이터 란을 추가한다.

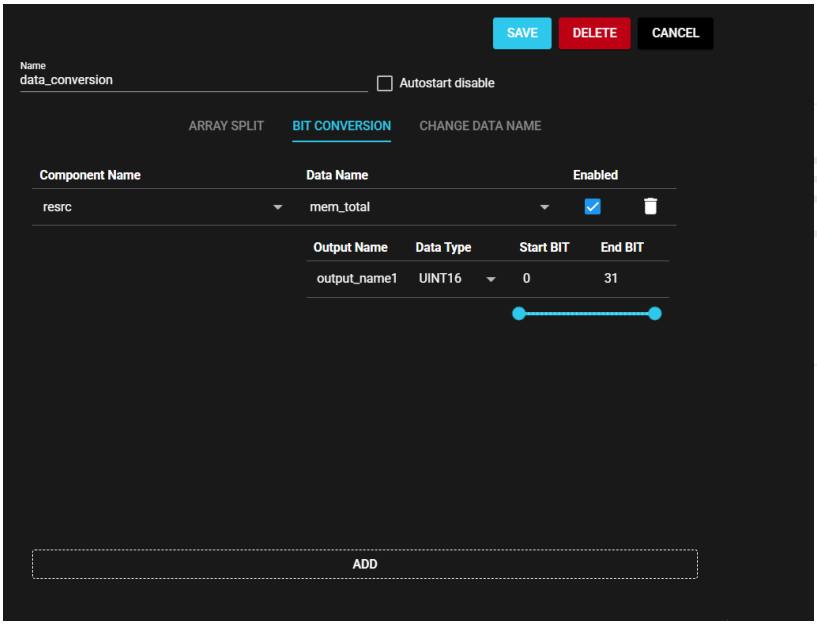


2. 입력할 컴포넌트명과 데이터명을 입력한다.
(드롭다운 선택 또는 수동으로 입력)



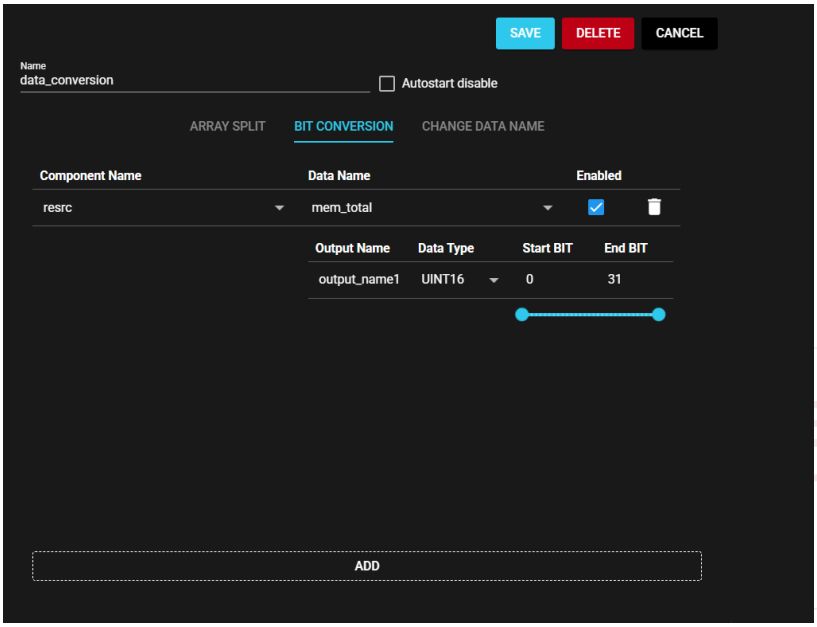
항목	설명
컴포넌트명	입력 포트에 접속한 컴포넌트명을 설정
데이터명	입력 포트에 접속한 컴포넌트명의 데이터명을 설정
유효	설정한 입력 데이터 설정을 무효로 할 때는 OFF로 하십시오
삭제	설정한 입력 데이터 설정을 삭제할 때에 누르십시오

3. 출력할 데이터의 설정을 입력한다.



항목	설명
출력 데이터명	출력 데이터명을 설정
데이터형	출력할 데이터형을 설정
개시BIT	입력할 데이터로부터 취득할 데이터의 개시BIT를 설정. 수동 입력 또는 비트 레인지 설정을 사용하십시오. 최하위 비트는 0, 최상위 비트는 입력할 데이터형의 비트 폭에 따라, 7, 15, 31, 63으로 변동합니다
종료BIT	입력할 데이터로부터 취득할 데이터의 종료BIT를 설정. 수동 입력 또는 비트 레인지 설정을 사용하십시오. 최하위 비트는 0, 최상위 비트는 입력할 데이터형의 비트 폭에 따라, 7, 15, 31, 63으로 변동합니다
비트 레인지 설정	개시BIT, 종료BIT와 연동하고 있으며, 어느 한쪽의 방법으로 입력. 비트 레인지의 최대값은 스칼라 데이터인 경우, 입력할 데이터형에 연동, 배열 데이터인 경우는 입력할 배열 수에 연동

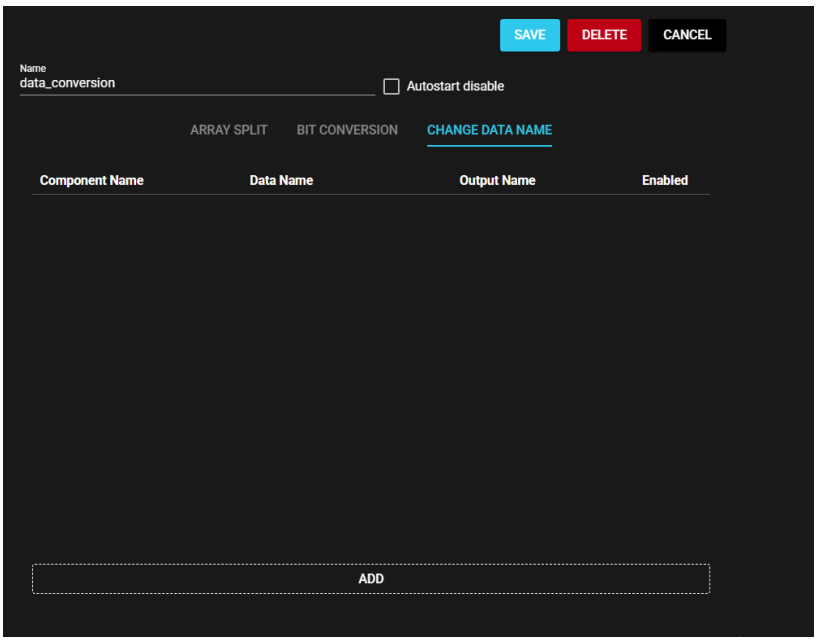
4. 「 저장 」을 누릅니다.



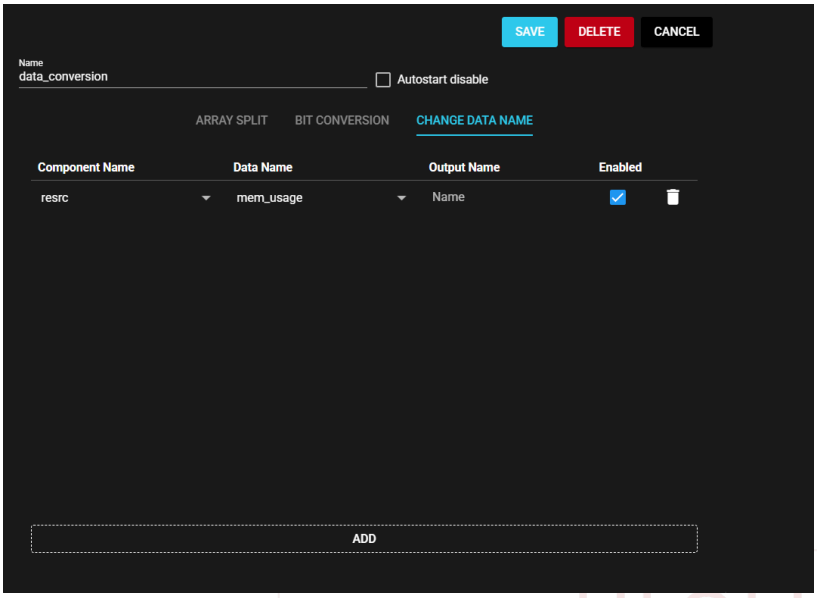
5. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은, 「 컴포넌트의 기동/정지 」를 참조하십시오.

5.8.4.5 설정 방법(데이터명 변경)

1. 추가 버튼을 눌러 입력 데이터 란을 추가한다.



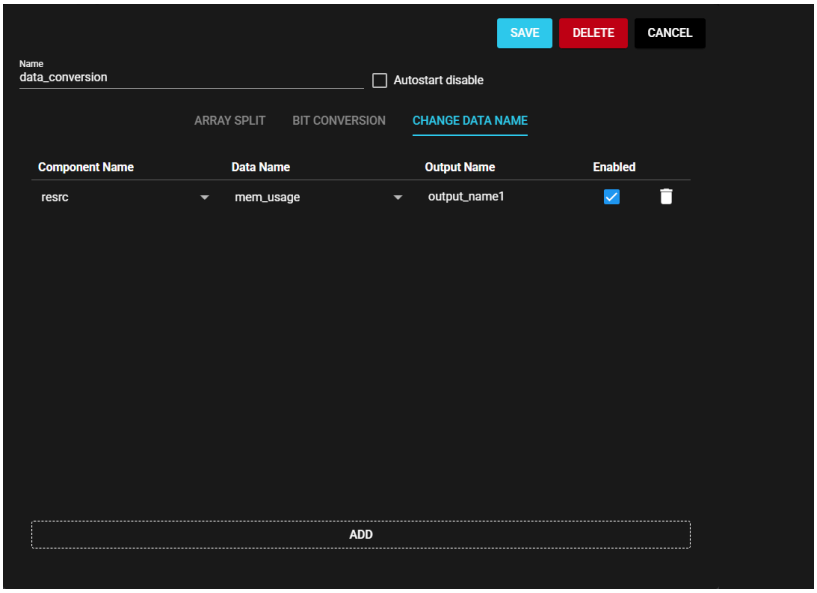
2. 입력할 컴포넌트명과 데이터명을 입력한다.
(드롭다운 선택 또는 수동으로 입력)



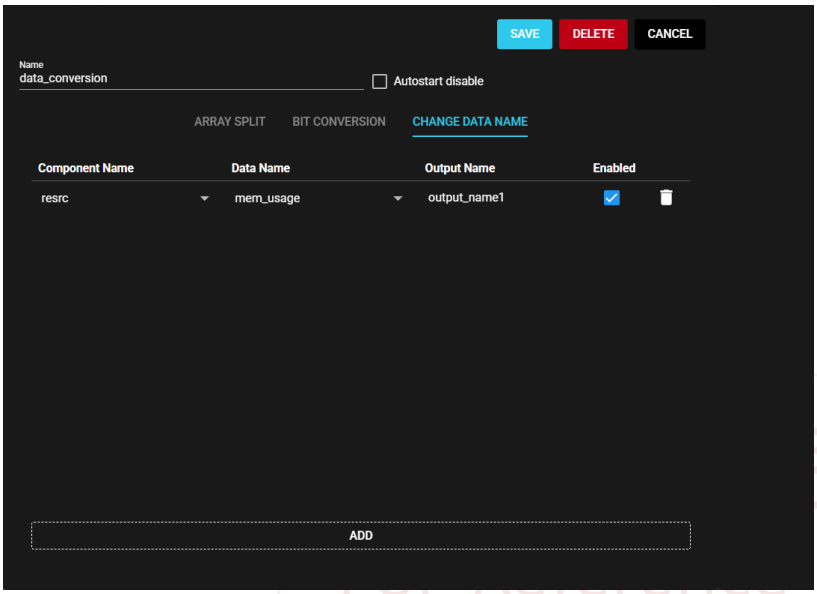
항목	설명
컴포넌트명	입력 포트에 접속한 컴포넌트명을 설정
데이터명	입력 포트에 접속한 컴포넌트명의 데이터명을 설정
출력 데이터명	출력 데이터명을 설정
유효	설정한 입력 데이터 설정을 무효로 할 때는 OFF로 하십시오

항목	설명
삭제	설정한 입력 데이터 설정을 삭제할 때에 누르십시오

3. 출력할 데이터명을 입력한다.



4. 「 저장 」을 누릅니다.



5. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은, 「 컴포넌트의 기동/정지 」를 참조하십시오.

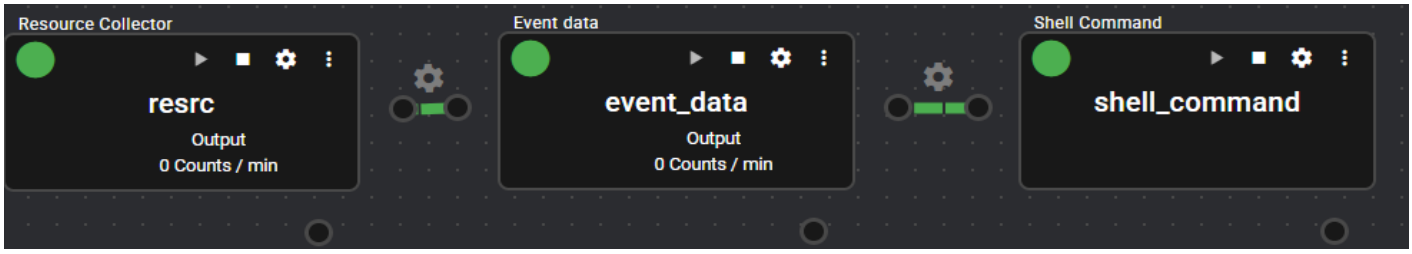
For Reference Only-

5.9 액션

여기에서는, 각 액션 컴포넌트에 대해 설명합니다.

5.9.1 셸 커맨드

수집한 데이터를 사용하여, 임의의 셸 커맨드를 실행하는 컴포넌트입니다.
입력 포트에서 데이터를 받아들이면, 파라미터로 지정된 임의의 셸 커맨드를 실행합니다.
셸 커맨드를 지정하는 문자열에는, 받아들인 데이터명이나 형, 값, 타임스탬프 등을 사용할 수 있습니다.



5.9.1.1 기본 정보

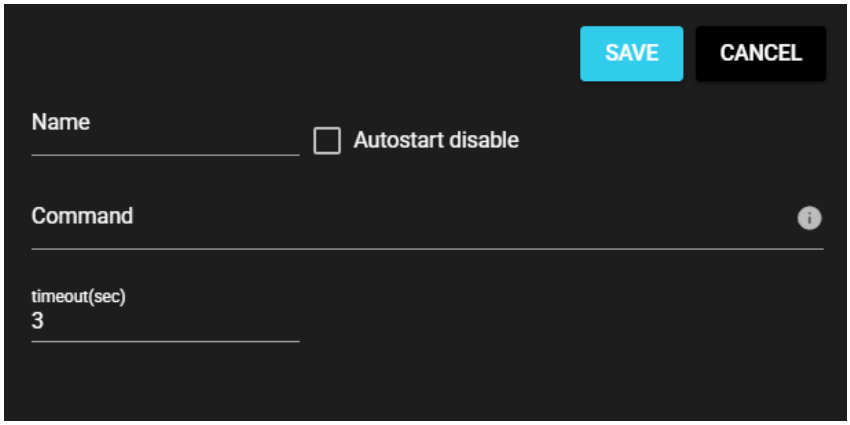
셸 커맨드의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	셸 커맨드
설명	데이터의 입력을 실행 트리거로 하여, 셸 커맨드를 실행합니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

AI 번역본
-For Reference Only-

5.9.1.2 설정 항목

셸 커맨드의 설정 항목은, 다음과 같습니다.

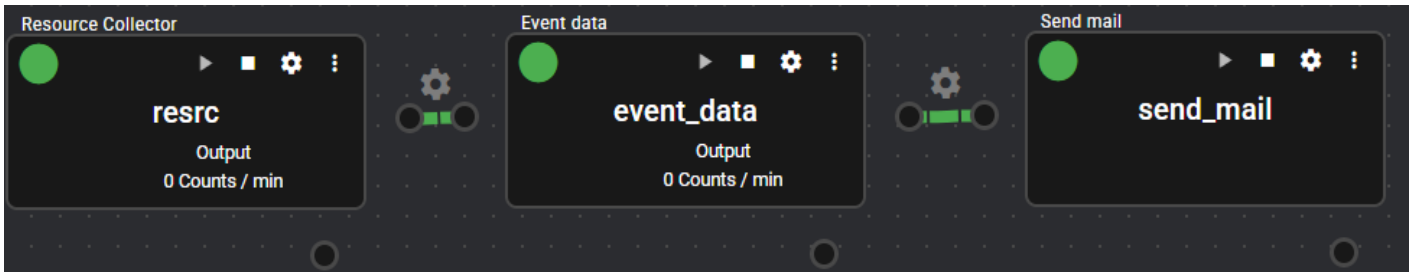


항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
커맨드	실행할 셸 커맨드 ※1 변수를 사용하면, 대응하는 정보로 변환할 수 있습니다. ※2 OS에 따라 지정 방법이 다릅니다. **Linux** $HIVE_COMPONENT：데이터의 송신원 컴포넌트명 $HIVE_DATANAME：데이터명 $HIVE_DATATYPE：데이터형 $HIVE_DATAVALUE：값 $HIVE_TIMESTAMP：데이터의 타임스탬프 $HIVE_DATAARRAYSIZE：데이터의 배열 수 **Windows** %HIVE_COMPONENT%：데이터의 송신원 컴포넌트명 %HIVE_DATANAME%：데이터명 %HIVE_DATATYPE%：데이터형 %HIVE_DATAVALUE%：값 %HIVE_TIMESTAMP%：데이터의 타임스탬프 %HIVE_DATAARRAYSIZE%：데이터의 배열 수
타임아웃(초)	커맨드 실행 시의 타임아웃 시간을 입력

AI 번역본
-For Reference Only-

5.9.2 메일 송신

수집한 데이터를 사용하여 메일 송신을 수행하는 컴포넌트입니다.
입력 포트에서 데이터를 받아들이면, 지정된 수신처, 내용으로 메일을 송신합니다.



5.9.2.1 기본 정보

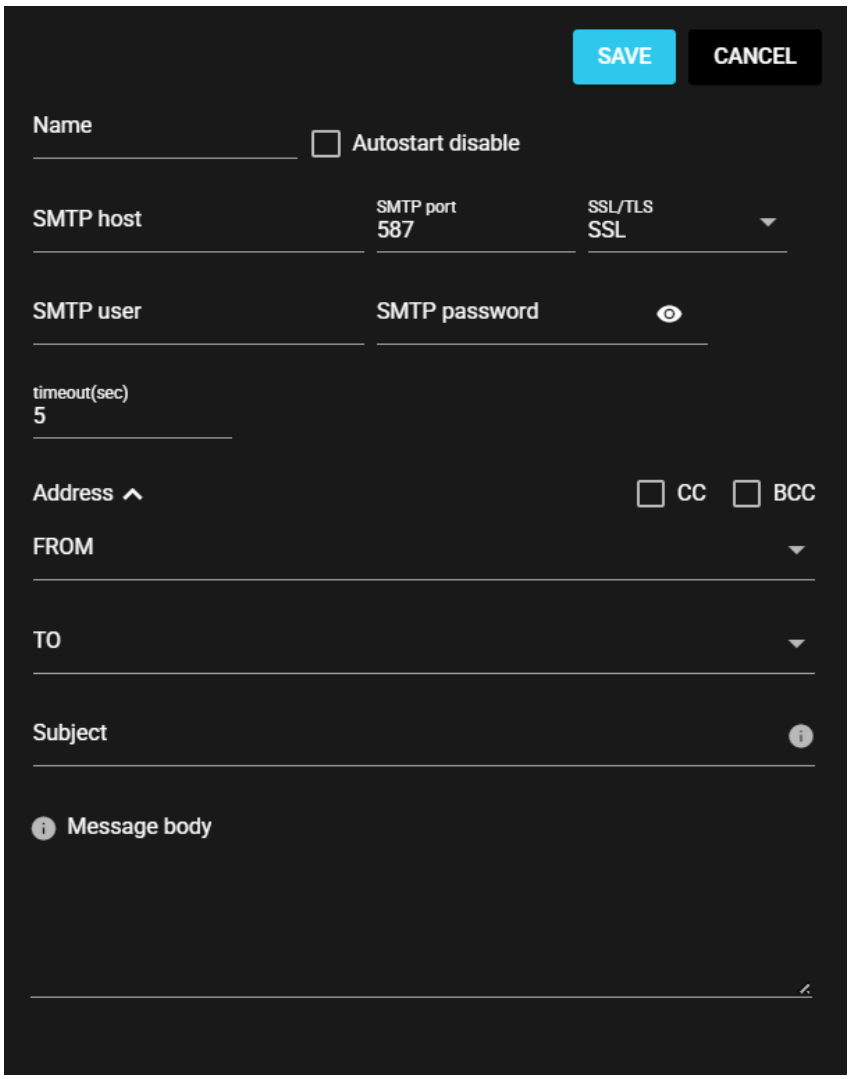
메일 송신의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	메일 송신
설명	데이터의 입력을 실행 트리거로 하여, 내용과 수신처를 지정한 메일을 송신합니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	메일 서버에 대한 테스트 접속을 수행할 수 있습니다. 상세는「 테스트 접속 」을 참조하십시오.

For Reference Only

5.9.2.2 설정 항목

메일 송신의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
SMTP 호스트	메일 송신을 수행할 SMTP 서버의 호스트명orIP 주소
SMTP 포트	메일 송신을 수행할 SMTP 서버의 포트
SSL/TLS	메일의 암호화 방식
SMTP 사용자명	메일 송신 시의 SMTP 인증에 필요한 사용자명

AI 번역본
-For Reference Only-

항목	설명
SMTP 패스워드	메일 송신 시의 SMTP 인증에 필요한 패스워드
타임아웃 (초)	메일 송신에 걸리는 시간의 상한
FROM	송신할 메일의 발신인의 메일 주소
TO	송신할 메일의 수신처의 메일 주소
CC	송신할 메일의 수신처(CC)의 메일 주소
BCC	송신할 메일의 수신처(BCC)의 메일 주소
제목	송신할 메일의 제목 ※변수를 사용하면, 대응하는 정보로 변환할 수 있습니다. 상세는「메일 송신에서 사용 가능한 변수」를 참조하십시오.
본문	송신할 메일의 본문 ※변수를 사용하면, 대응하는 정보로 변환할 수 있습니다. 상세는「메일 송신에서 사용 가능한 변수」를 참조하십시오.

5.9.2.2.1 메일 송신에서 사용 가능한 변수

변수명	설명	변환 후의 문자열 예
$HIVE_COMPONENT	데이터의 송신원 컴포넌트명	plc-collector
$HIVE_DATANAME	데이터명	D110
$HIVE_DATATYPE	데이터형	double
$HIVE_DATAVALUE	값	123.45
$HIVE_DATAARRAYSIZE	데이터의 배열 수	0
$HIVE_TIMESTAMP	데이터의 타임스탬프	1733356200123456789
$HIVE_TS	데이터의 타임스탬프（초 단위）	1733356200

AI 번역본
-For Reference Only-

변수명	설명	변환 후의 문자열 예
$HIVE_TS_MSEC	데이터의 타임스탬프（밀리초 단위）	1733356200123
$HIVE_TS_USEC	데이터의 타임스탬프（마이크로초 단위）	1733356200123456
$HIVE_TS_NSEC	데이터의 타임스탬프（나노초 단위）	1733356200123456789
$HIVE_TS_DATE_LOCAL	데이터의 타임스탬프의 연월일（로컬 타임존）	2024-12-05
$HIVE_TS_DATE_UTC	데이터의 타임스탬프의 연월일（UTC）	2024-12-04
$HIVE_TS_TIME_LOCAL	데이터의 타임스탬프의 시분초（로컬 타임존）	08:50:00
$HIVE_TS_TIME_UTC	데이터의 타임스탬프의 시분초（UTC）	23:50:00
$HIVE_TS_DATETIME_LOCAL	데이터의 타임스탬프의 연월일시분초（로컬 타임존）	2024-12-05T08:50:00
$HIVE_TS_DATETIME_UTC	데이터의 타임스탬프의 연월일시분초（UTC）	2024-12-04T23:50:00
$HIVE_TS_TZ_DIFF	로컬 타임존의 시차	+0900

변수명의 기술에 대해

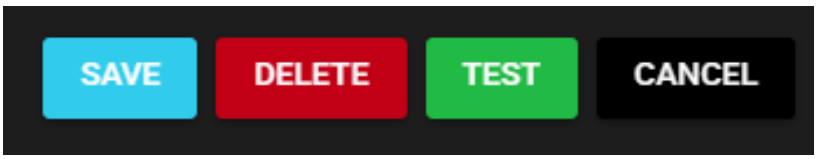
- 변수명의 HIVE_는 SYNAPSE_로 기술하는 것도 가능합니다.
- 변수의 기술 형식은 $HIVE_COMPONENT、${HIVE_COMPONENT}、%HIVE_COMPONENT%의 3종류의 기술 형식을 사용할 수 있습니다.

-For Reference Only-

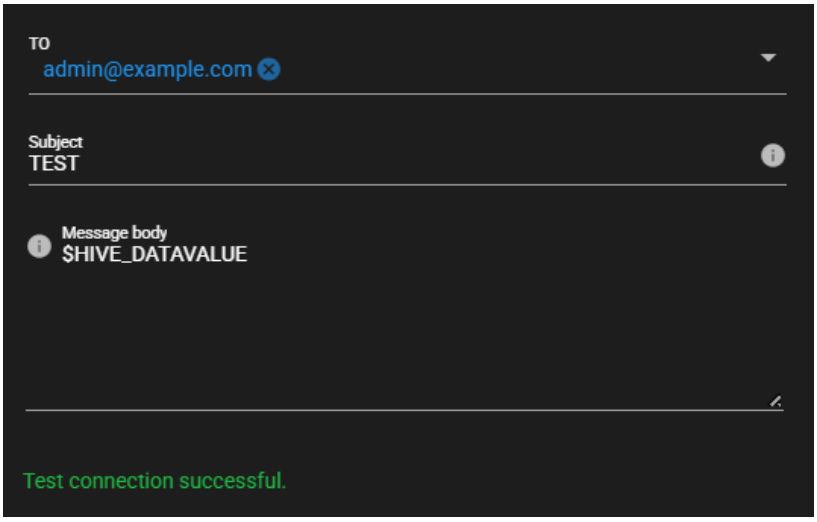
5.9.2.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 메일 송신의 설정 화면을 엽니다.
2.「테스트」를 누릅니다.

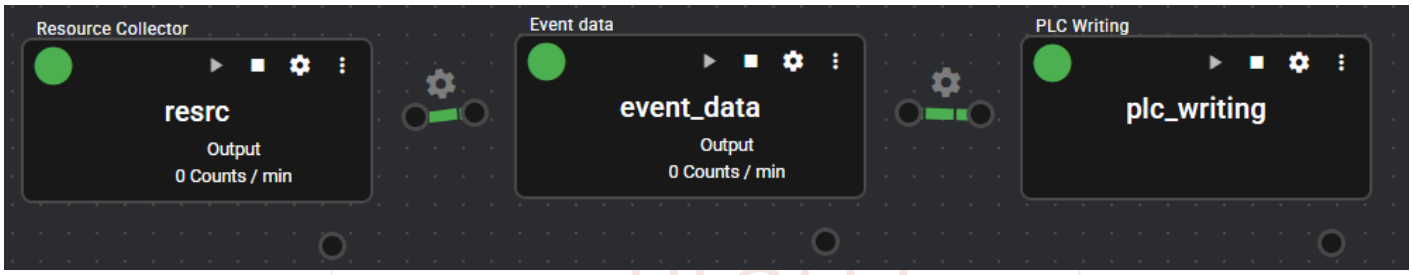


3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



5.9.3 PLC 쓰기

PLC의 레지스터에 값을 쓰는 컴포넌트입니다.
입력 포트에서 트리거가 되는 데이터를 받아들이면, PLC의 지정 레지스터에 임의의 값을 쓸 수 있습니다. 쓰는 값으로서는, 고정값이나 트리거가 된 데이터의 값, 다른 컴포넌트의 데이터의 값을 지정 가능합니다.



AI 번역본
-For Reference Only-

5.9.3.1 기본 정보

PLC 쓰기의 기본 정보는, 다음과 같습니다.

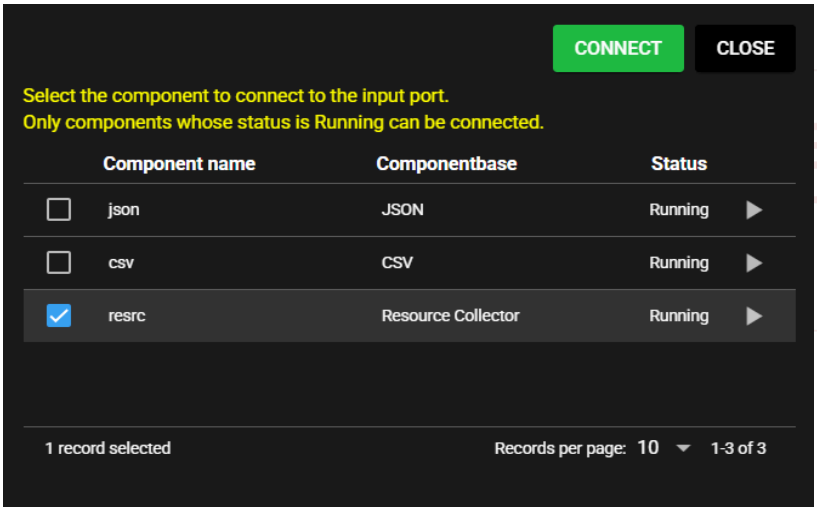
항목	설명
베이스명	PLC 쓰기
설명	데이터의 입력을 실행 트리거로 하여, PLC의 지정 레지스터에, 임의의 값을 씁니다.

항목	설명
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	PLC에 대한 테스트 접속을 수행할 수 있습니다. 상세는 「 테스트 접속 」을 참조하십시오.

5.9.3.2 설정 방법

다음 순서로, PLC 쓰기의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
2. 추가한 컴포넌트를 기동하여, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.
3. PLC 쓰기를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.



5. 다음 컴포넌트 정보를 입력합니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
제조사	접속할 PLC의 제조사를 선택 ※선택할 수 있는 제조사는 다음과 같습니다. ・미쓰비시전기 ・제이텍트 ・오므론 ・키엔스 ・파나소닉
기종	접속할 PLC의 기종을 선택 ※선택할 수 있는 기종은 제조사에 따라 다릅니다.
호스트	접속할 PLC의 호스트orIP 주소
포트	접속할 PLC의 포트
접속 타임아웃（밀리초）	PLC에의 접속에 걸리는 시간의 상한
쓰기 타임아웃（밀리초）	값 쓰기에 걸리는 시간의 상한
상시 접속한다	PLC에의 접속을 조회할 때마다, 절단하지 않는다

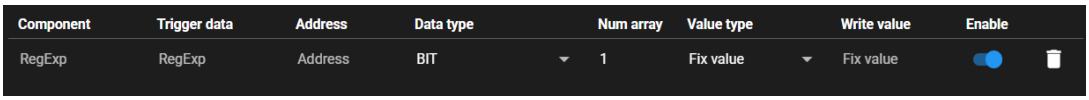
-For Reference Only-

6.「 트리거 데이터 지정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터의 목록으로부터, 쓰기의 트리거로 할 데이터를 선택한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것을 쓰기의 트리거로 한다

7. 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
(설정 방법이 「 정규 표현식으로 설정한다 」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 컴포넌트명을 나타내는 정규 표현식
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 데이터명을 나타내는 정규 표현식
(설정 방법이 「 데이터 정의를 수신하여 설정한다 」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
(공통)	
주소	값의 쓰기 위치인 레지스터의 주소
프로그램	값의 쓰기 위치인 레지스터의 프로그램 번호 ※제조사가 제이텍트인 경우만 선택 가능
데이터형	값의 쓰기 위치인 레지스터의 데이터형

AI 번역본
-For Reference Only-

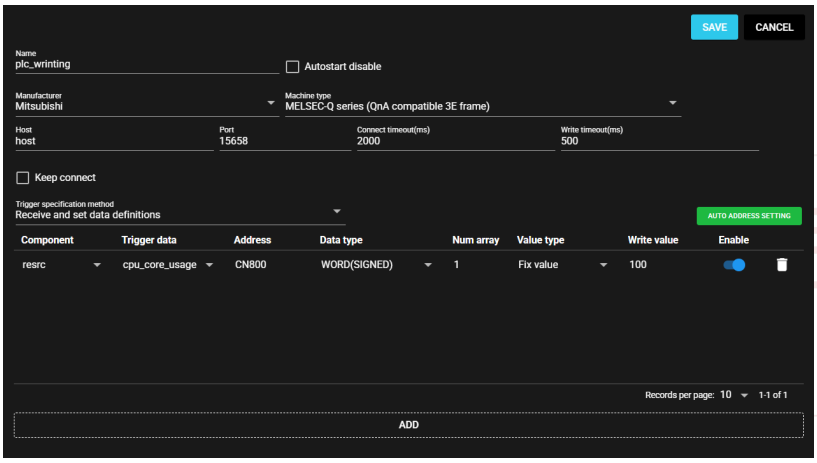
항목	설명
연속 영역 수	입력한 주소를 선두로 하여 연속하는 레지스터 중에서, 값 쓰기의 대상으로 하는 범위
값 종별	・고정값 : 쓰는 값으로서, 고정값을 입력한다 ・컴포넌트 : 쓰는 값으로서, 입력 포트에서 받아들인 데이터를 선택한다 ・이벤트 : 트리거 데이터의 값이 쓰인다
쓰는 값	선택한 값 종별에 따른 값을 입력 ※값 종별이 이벤트인 경우는 입력 불가
유효	쓰기를 수행할지 여부

정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

- 「data1」에만 매칭 : data1
- 이름에「abc」를 포함하는 데이터에 매칭 : .*abc.*
- 이름이「abc」로 시작하는 데이터에 매칭 : abc.*
- 이름이「abc」로 끝나는 데이터에 매칭 : .*abc

8. 「저장」을 누릅니다.

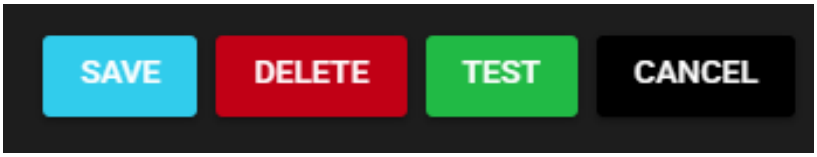


9. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「컴포넌트의 기동/정지」를 참조하십시오.

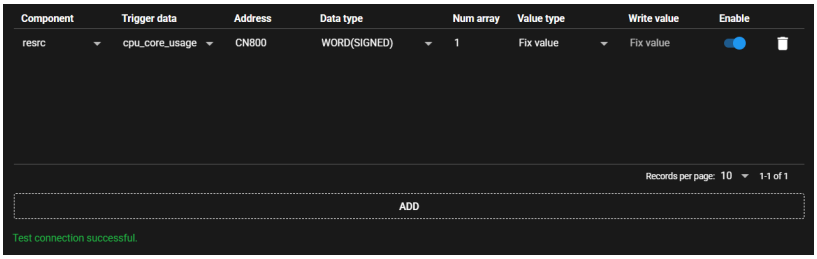
5.9.3.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 PLC 쓰기의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



5.9.3.4 주소 자동 설정

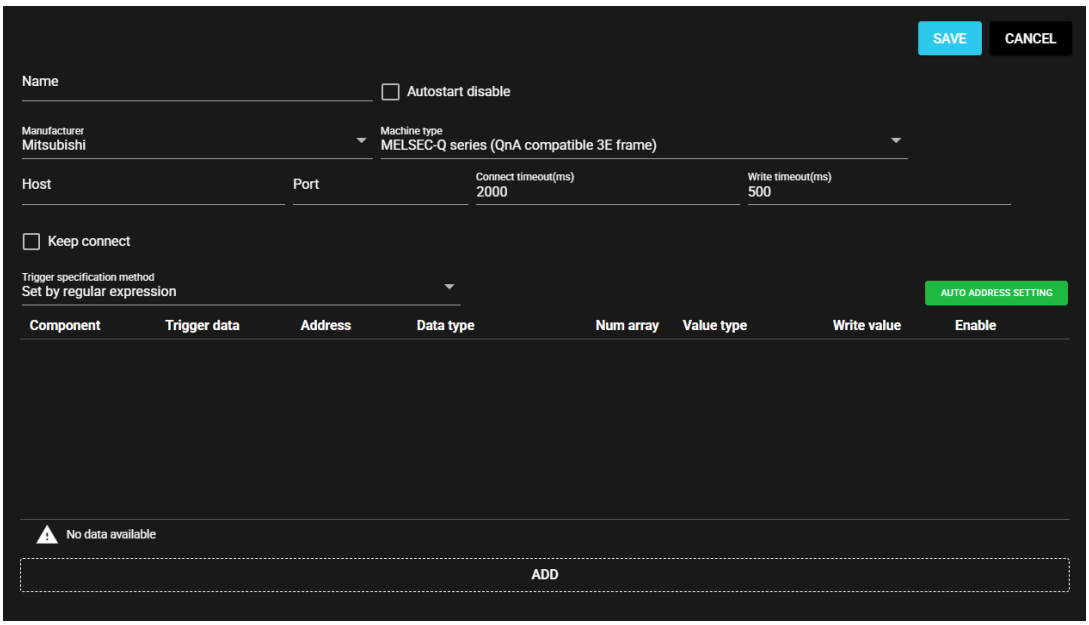
일부의 PLC에서는, 래더 편집 툴로 출력한 파일을 읽어 들임으로써, 주소를 자동으로 일괄 설정할 수 있습니다.

1. 래더 편집 툴로, 작성한 래더를 파일로서 출력합니다. 출력할 파일의 형식은, PLC의 제조사・기종에 따라 다릅니다. 파일의 샘플은 다음과 같습니다.
   - 미쓰비시전기 MELSEC-Q 시리즈(QnA 호환3E 프레임), MELSEC iQ-R 시리즈(QnA 호환3E 프레임)

     샘플 파일의 다운로드
   - 오므론 SYSMAC CS/CJ/CP/NSJ 시리즈(FINS)

     샘플 파일의 다운로드
2. 「 주소 자동 설정 」을 누릅니다.

AI 번역본
-For Reference Only-

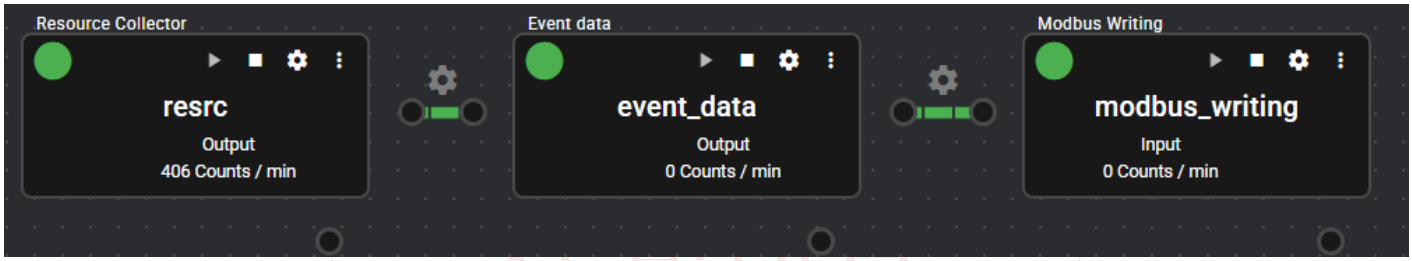


3. 래더 편집 툴로 출력한 파일을 선택합니다.

4. 파일을 바탕으로, 주소가 자동으로 설정됩니다.

5.9.4 Modbus 쓰기

Modbus TCP 대응 기기의 지정 주소에 값을 쓰는 컴포넌트입니다.
입력 포트에서 트리거가 되는 데이터를 받아들이면, 본 컴포넌트가 Modbus의 마스터가 되어, Modbus 슬레이브 기기의 지정 레지스터에 임의의 값을 씁니다.
쓰는 값으로서는, 고정값이나 트리거가 된 데이터의 값, 다른 컴포넌트의 데이터의 값을 지정 가능합니다.



AI 번역본
-For Reference Only-

5.9.4.1 기본 정보

Modbus 쓰기의 기본 정보는, 다음과 같습니다.

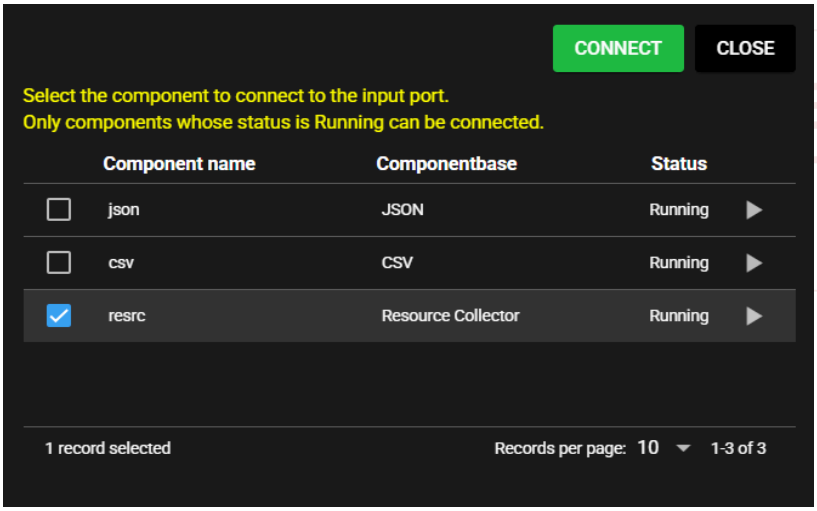
항목	설명
베이스명	Modbus 쓰기
설명	데이터의 입력을 실행 트리거로 하여, Modbus TCP 대응 기기의 지정 레지스터에, 임의의 값을 씁니다.

항목	설명
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
기타	Modbus TCP 대응 기기에 대한 테스트 접속을 수행할 수 있습니다. 상세는「 테스트 접속 」을 참조하십시오.

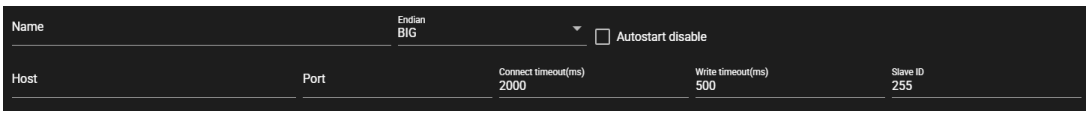
5.9.4.2 설정 방법

다음 순서로, Modbus 쓰기의 설정을 수행할 수 있습니다.

1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
2. 추가한 컴포넌트를 기동하여, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.
3. Modbus 쓰기를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.



5. 다음 컴포넌트 정보를 입력합니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
엔디안	쓰는 값의 바이트 오더
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	접속할 기기의 호스트orIP 주소
포트	접속할 기기의 포트
접속 타임아웃（밀리초）	기기에의 접속에 걸리는 시간의 상한
쓰기 타임아웃（밀리초）	값 쓰기에 걸리는 시간의 상한
슬레이브ID	접속처의 슬레이브ID

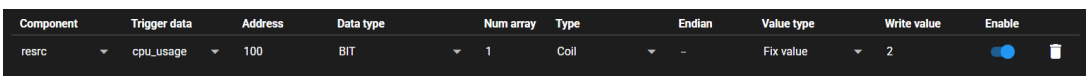
6.「트리거 데이터 지정 방법」을, 다음 중에서 선택합니다.



AI 번역본
For Reference Only

항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터의 목록으로부터, 쓰기의 트리거로 할 데이터를 선택한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것을 쓰기의 트리거로 한다

7. 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



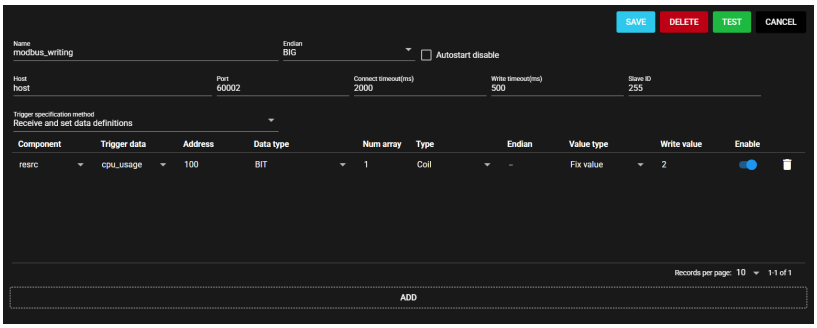
항목	설명
(설정 방법이「정규 표현식으로 설정한다」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 컴포넌트명을 나타내는 정규 표현식
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 데이터명을 나타내는 정규 표현식
(설정 방법이「데이터 정의를 수신하여 설정한다」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
(공통)	
주소	값의 쓰기 위치인 레지스터의 주소 ※Modbus의 사양상, 주소 범위는 통상「1~65536」으로 되어 있습니다만, SpeeDBee Synapse에서 설정할 때의 주소 범위는「0~65535」와 같이 -1한 주소를 설정할 필요가 있습니다.
데이터형	값의 쓰기 위치인 레지스터의 데이터형
연속 영역 수	입력한 주소를 선두로 하여 연속하는 레지스터 중에서, 값 쓰기의 대상으로 하는 범위
종류	값의 쓰기 위치인 레지스터의 종류
엔디안	쓰는 데이터의 바이트 오더 ※「-」인 경우, 전체의 엔디안이 적용된다
값 종별	・고정값：쓰는 값으로서, 고정값을 입력한다 ・컴포넌트：쓰는 값으로서, 입력 포트에서 받아들인 데이터를 선택한다 ・이벤트：트리거 데이터의 값이 쓰인다

AI 번역본
-For Reference Only-

항목	설명
쓰는 값	선택한 값 종별에 따른 값을 입력 ※값 종별이 이벤트인 경우는 입력 불가
유효	쓰기를 수행할지 여부

> 정규 표현식의 입력 예
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「data1」에만 매칭： data1
> - 이름에「abc」를 포함하는 데이터에 매칭： .*abc.*
> - 이름이「abc」로 시작하는 데이터에 매칭： abc.*
> - 이름이「abc」로 끝나는 데이터에 매칭： .*abc

8. 「저장」을 누릅니다.

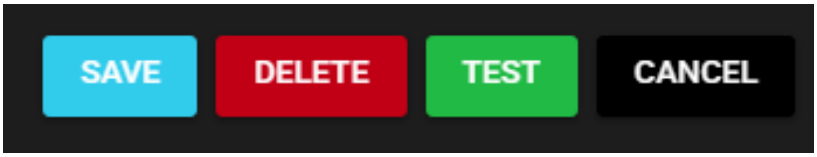


9. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「컴포넌트의 기동/정지」를 참조하십시오.

5.9.4.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 Modbus 쓰기의 설정 화면을 엽니다.
2. 「테스트」를 누릅니다.



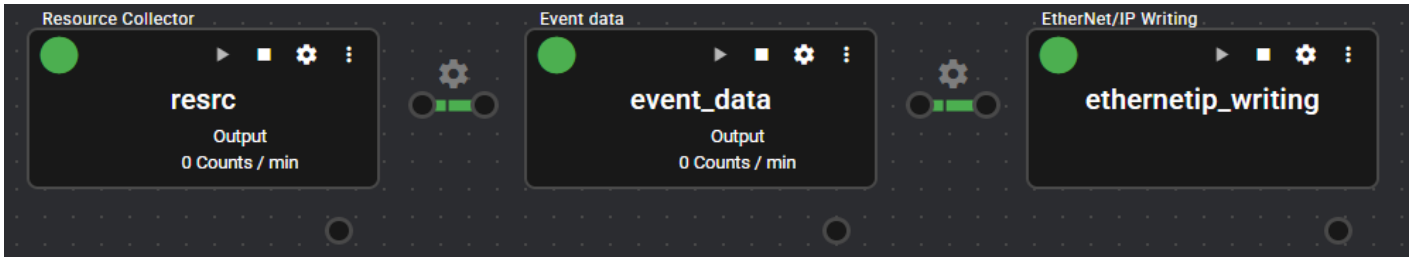
3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.

AI 번역본
-For Reference Only-



5.9.5 EtherNet/IP 쓰기

EtherNet/IP 대응 기기의 지정 주소에 값을 쓰는 컴포넌트입니다.
입력 포트에서 트리거가 되는 데이터를 받아들였을 때, 본 컴포넌트가 오리지네이터가 되어 EtherNet/IP 대응 기기에 접속하여, 지정 변수에 임의의 값을 씁니다.



5.9.5.1 기본 정보

EtherNet/IP 쓰기의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	EtherNet/IP 쓰기
설명	데이터의 입력을 실행 트리거로 하여, EtherNet/IP 대응 기기의 지정한 변수에, 임의의 값을 씁니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트로부터 데이터를 받아들입니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

AI 번역본
-For Reference Only-

항목	설명
기타	변수를 자동으로 설정할 수 있습니다. 상세는「 변수 자동 설정 」을 참조하십시오.

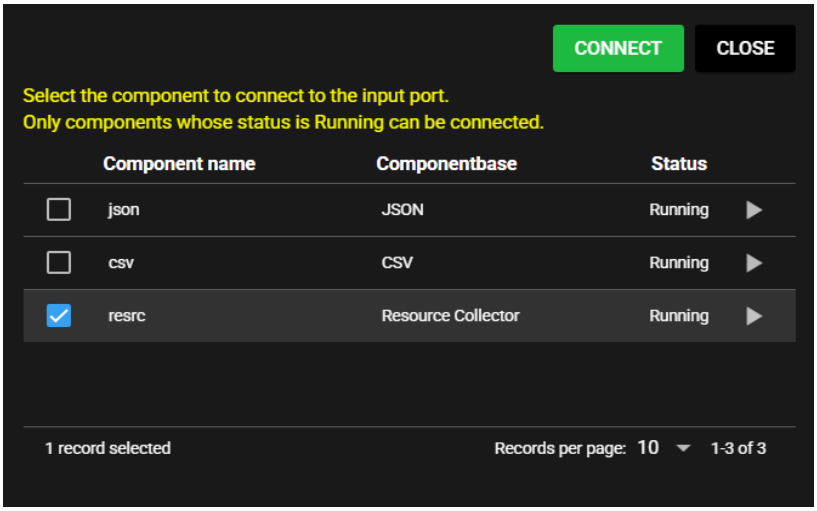
5.9.5.2 설정 방법

다음 순서로, EtherNet/IP 쓰기의 설정을 수행할 수 있습니다.

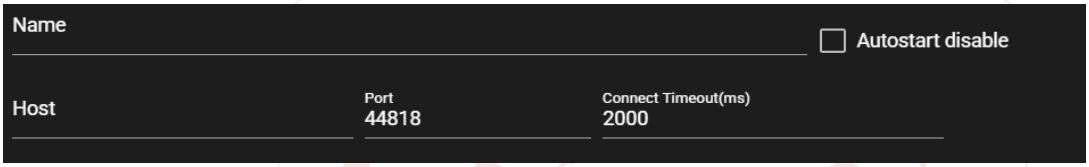
1. 출력 포트가 존재하는 임의의 컴포넌트를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.

2. 추가한 컴포넌트를 기동하여, 데이터를 출력시킵니다. 컴포넌트의 기동 방법은,「 컴포넌트의 기동/정지 」를 참조하십시오.

3. EtherNet/IP 쓰기를 추가합니다. 컴포넌트의 추가 방법은,「 컴포넌트의 추가 」를 참조하십시오.
4. 입력 포트에 접속할 컴포넌트에, 추가한 컴포넌트를 선택합니다.



5. 다음 컴포넌트 정보를 입력합니다.

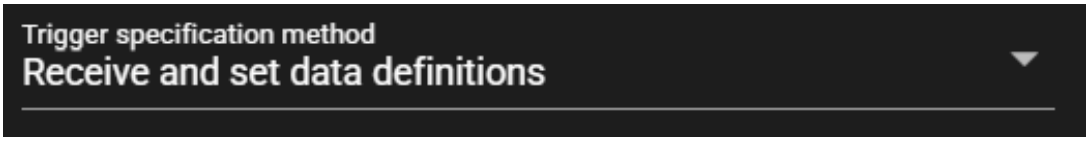


-For Reference Only-

항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
호스트	EtherNet/IP 대응 기기의 호스트명 또는 IP 주소를 입력

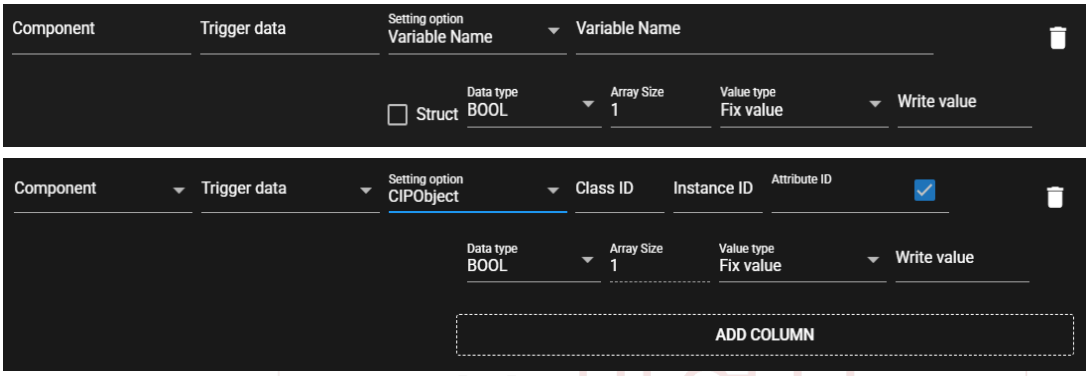
항목	설명
포트	EtherNet/IP 대응 기기의 포트를 입력
접속 타임아웃(밀리초)	접속 시의 타임아웃 시간을 입력

6. 「 트리거 데이터 지정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터의 목록으로부터, 쓰기의 트리거로 할 데이터를 선택한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것을 쓰기의 트리거로 한다

7. 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
(설정 방법이 「 정규 표현식으로 설정한다 」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 컴포넌트명을 나타내는 정규 표현식
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것의 데이터명을 나타내는 정규 표현식

항목	설명
(설정 방법이「데이터 정의를 수신하여 설정한다」인 경우)	
컴포넌트	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
트리거 데이터	입력 포트에서 받아들이고 있는 데이터 중, 쓰기의 트리거로 할 것을 선택
(공통)	
설정 방법	취득할 데이터의 설정 방법을 선택
변수명	태그명을 입력 ※설정 방법이 변수명인 경우에만 입력 가능합니다.
구조체	복수의 멤버를 등록하는 경우 ON ※설정 방법이 변수명인 경우에만 설정 가능합니다.
클래스 ID	클래스 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
인스턴스 ID	인스턴스 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
어트리뷰트 ID	어트리뷰트 ID를 입력 ※설정 방법이 CIPObject인 경우에만 입력 가능합니다.
어트리뷰트 설정	어트리뷰트 ID를 입력할 때 체크 ※설정 방법이 CIPObject인 경우에만 조작 가능합니다.
데이터형	멤버의 데이터형
배열 수	멤버의 배열 수 ※설정 방법이 변수명인 경우에만 조작 가능합니다.
값 종별	・고정값：쓰는 값으로서, 고정값을 입력한다 ・컴포넌트：쓰는 값으로서, 입력 포트에서 받아들인 데이터를 선택한다 ・이벤트：트리거 데이터의 값이 쓰인다

AI 번역 버전
-For Reference Only-

항목	설명
쓰는 값	선택한 값 종별에 따른 값을 입력 ※값 종별이 이벤트인 경우는 입력 불가

> 정규 표현식의 입력 예
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「data1」에만 매칭 : data1
> - 이름에「abc」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이「abc」로 시작하는 데이터에 매칭 : abc.*
> - 이름이「abc」로 끝나는 데이터에 매칭 : .*abc

8. 「저장」을 누릅니다.



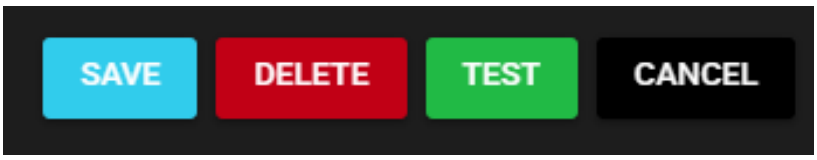
9. 컴포넌트를 기동합니다. 컴포넌트의 기동 방법은,「컴포넌트의 기동/정지」를 참조하십시오.

AI 번역본
For Reference Only-

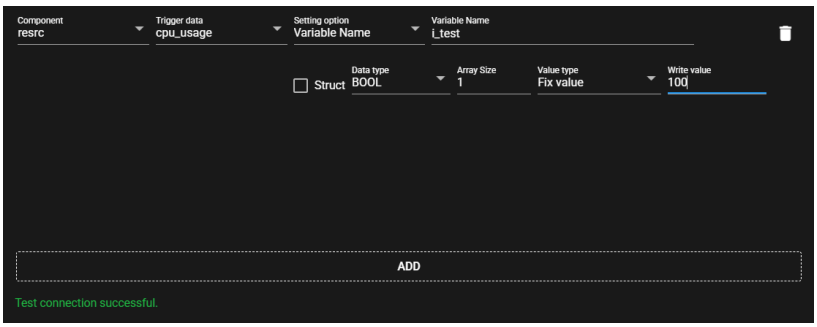
5.9.5.3 테스트 접속

다음 순서로, 설정한 접속처에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 EtherNet/IP 쓰기의 설정 화면을 엽니다.
2. 「테스트」를 누릅니다.



3. 설정한 접속처에의 테스트 접속이 수행되어, 설정 화면의 하부에 결과가 표시됩니다.



5.9.5.4 변수 자동 설정

Sysmac Studio가 익스포트하는 글로벌 변수 정보 등을, 지정의 포맷의 텍스트 파일로 변환하여 읽어들임으로써, 변수를 자동 설정할 수 있습니다.

1. Sysmac Studio가 익스포트하는 글로벌 변수 정보 등을, 다음의 열 정보를 갖는 텍스트 파일로 변환합니다.

   - HOST
   - NAME
   - DATATYPE
   - ADDRESS
   - COMMENT
   - TAGLINK
   - RW
   - POU

   예

```
HOST    NAME    DATATYPE    ADDRESS COMMENT TAGLINK RW  POU
        b_test  BOOL        test1           TRUE    RW
        i_test  UINT        test2           TRUE    RW
        s_test  STRING(256) test3           TRUE    RW
```

2. 「변수 자동 설정」을 누릅니다.



3. 준비한 텍스트 파일을 선택합니다.

4. 파일을 바탕으로, 변수가 자동으로 설정됩니다.

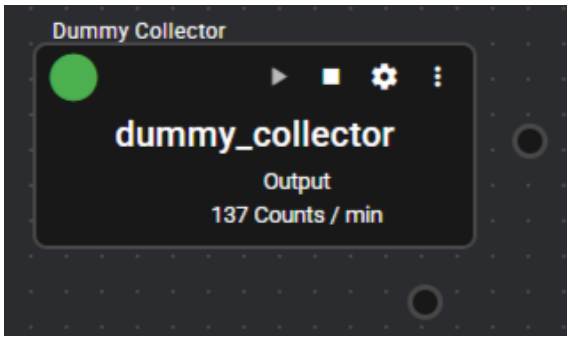
AI 번역본
-For Reference Only-

5.10 개발용 컴포넌트

여기서는, 각 개발용 컴포넌트에 대해 설명합니다. 개발용 컴포넌트는, SCCDE 테스트 실행에서만 이용 가능합니다.

5.10.1 더미 컬렉터

더미 데이터의 등록을 수행하는 컴포넌트입니다.
지정된 운전 파일(csv)의 내용을 바탕으로, 컬럼을 작성하여 더미 데이터를 등록합니다.



5.10.1.1 기본 정보

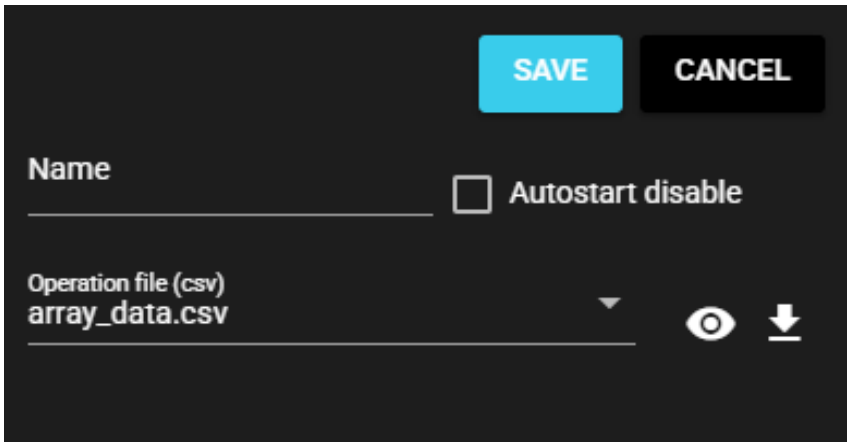
더미 컬렉터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	더미 컬렉터
설명	더미 데이터를 등록합니다.
소비 포인트	0
포트	입력 : 0 출력 : 1 운전 파일(csv)의 내용을 바탕으로 더미 데이터를 출력합니다. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

AI 번역본
-For Reference Only-

5.10.1.2 설정 항목

더미 컬렉터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
운전 파일(csv)	작성할 컬럼이나 등록할 데이터를 정의한 csv 형식의 파일을 선택

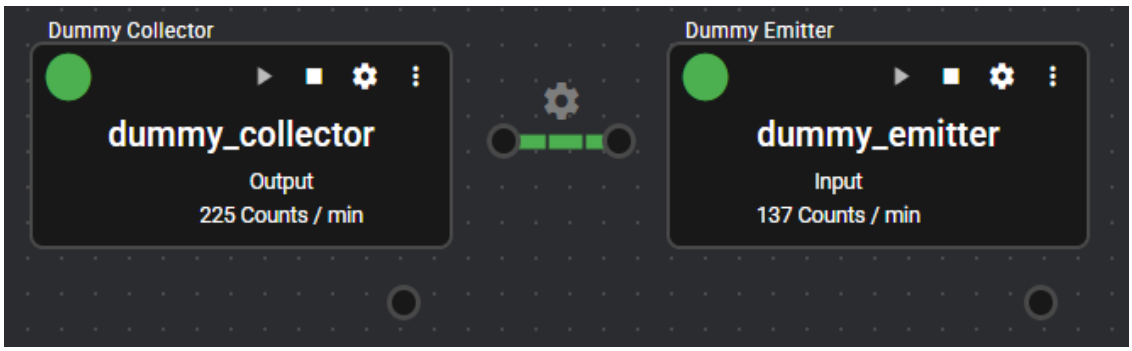
운전 파일은, sccde의 작업 디렉터리 내에 있는 simulation_data 디렉터리에 배치되어 있습니다. 초기 상태에서는, 이하의 파일을 선택하는 것이 가능하지만, 임의의 파일을 추가하여 선택하는 것도 가능합니다. 운전 파일의 포맷에 대해서는,
「 운전 파일의 포맷 」을 참조하십시오.

파일명	설명
all_primitive_data_type.csv	프리미티브형의 데이터를 등록합니다.
array_data.csv	배열의 데이터를 등록합니다.

For Reference Only

5.10.2 더미 에미터

수집한 데이터를 CSV 파일로서, 스토리지에 저장하는 컴포넌트입니다.
입력 포트에 접속된 각종 컴포넌트로부터 프리미티브형의 데이터를 받아들여, 그 데이터를 특정 디렉터리에 저장합니다.



5.10.2.1 기본 정보

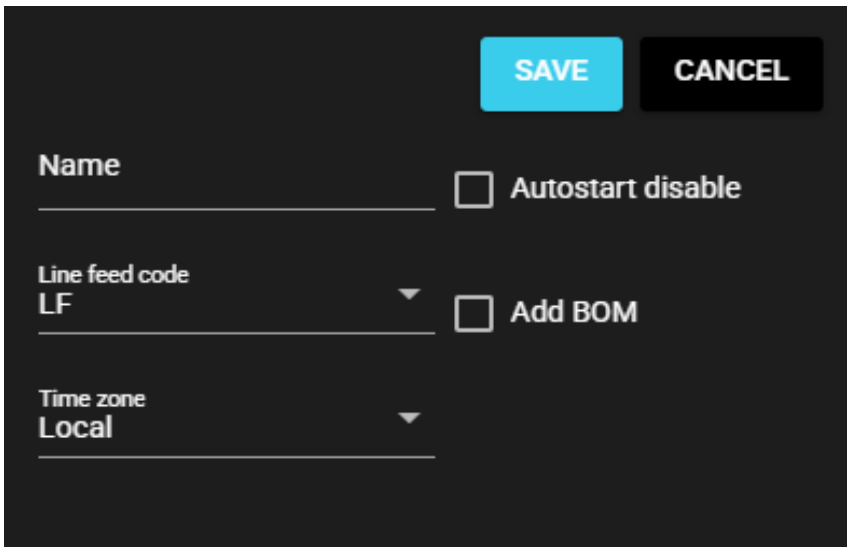
더미 에미터의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	더미 에미터
설명	수집 데이터를 CSV 형식으로 스토리지에 저장합니다.
소비 포인트	0
포트	입력 : 1 컴포넌트로부터, 프리미티브형의 데이터를 받아들입니다. 받아들인 데이터는, CSV 형식으로 스토리지에 저장됩니다. 출력 : 0 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.

AI 번역본
-For Reference Only-

5.10.2.2 설정 항목

더미 에미터의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트에 부여하는 이름 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
개행 코드	CSV 파일의 개행 코드를 선택
BOM을 붙인다	CSV 파일에 파일 BOM을 붙이는 경우 ON
타임존	파일, CSV 데이터에 부여하는 타임스탬프의 타임존을 선택

출력처는, sccde의 작업 디렉터리 내에 있는 .synapse/var-speedbeesynapse/projects/sccde/data 디렉터리
에, 등록되어 있는 더미 에미터마다 디렉터리가 작성되어, 파일이 출력됩니다.

5.10.2.3 파일의 전환 타이밍

AI 번역본
-For Reference Only-

더미 에미터는 기동할 때마다 파일이 새롭게 전환됩니다. 기동 후는, 기본적으로 같은 파일에 계속 출력하지만, 이하의 경우에만 파일을 새롭게 전환합니다.

- 파일의 사이즈가 상한(10MB)을 초과한 경우.
- 입력 포트에서 받아들인 데이터의 컬럼 정의（컬럼 수, 컬럼명 등）가 변화한 경우.

※출력된 파일은, 자동 삭제되지 않으므로, 필요에 따라 삭제하십시오.

5.11 시스템 컴포넌트

여기서는, 각 시스템 컴포넌트에 대해 설명합니다. 시스템 컴포넌트는, 다른 컴포넌트와 달리 복수의 인스턴스를 설정할 수 없습니다. 시스템 패널에 배치되어 있는 컴포넌트를 설정하여 이용합니다.

5.11.1 DB 쿼리 서비스

SpeeDBee Synapse 내에서 수집한 데이터에 대한 조회를 접수하는 컴포넌트입니다. 이 컴포넌트는, 데이터 수집이나 데이터 참조를 수행하지 않습니다. 외부로부터의 SQL을 처리하기 위해 사용합니다. 상세는, DB 쿼리（SQL）를 참조하십시오.

쿼리 실행을 수행할 때는, DB 쿼리 서비스 이용 시의 주의점을 한 번 읽어볼 것을 권장합니다.

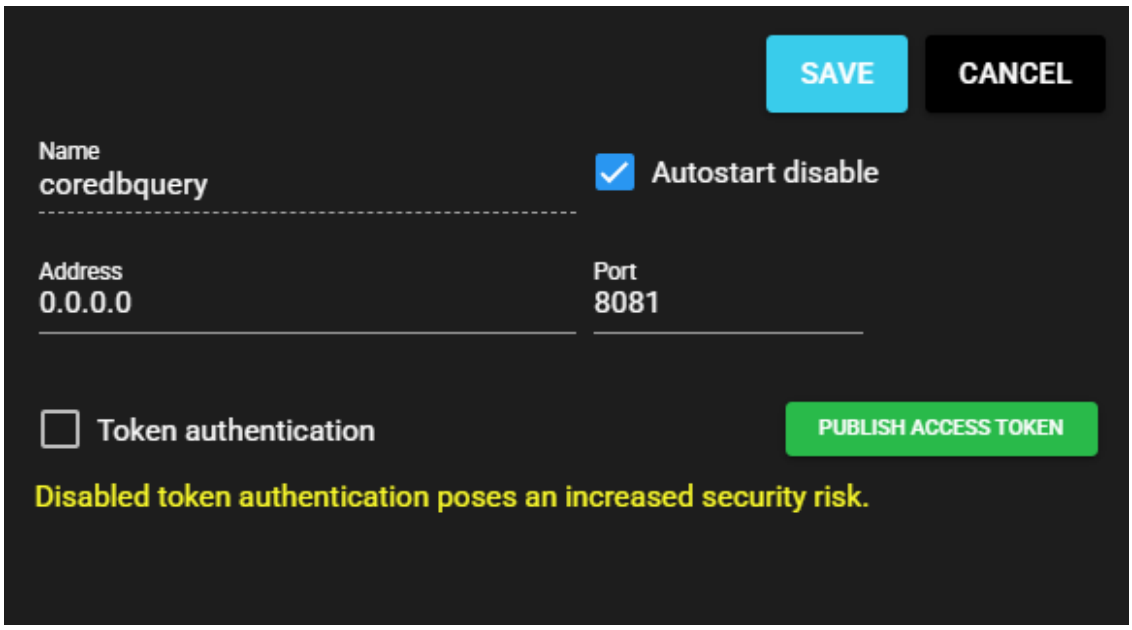
5.11.1.1 기본 정보

DB 쿼리 서비스의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	DB 쿼리 서비스
설명	데이터베이스에 대한 조회를 접수하는 컴포넌트입니다. 기동하면 HTTP 경유로 DB로의 쿼리를 발행할 수 있습니다. DB 쿼리에 대한 상세는,「DB 쿼리（SQL）」를 참조하십시오. 이 컴포넌트는, SpeeDBee Synapse에서 1개밖에 설정할 수 없습니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
소비 포인트	0
포트	입력：0 출력：0

5.11.1.2 설정 항목

DB 쿼리 서비스의 설정 항목은, 다음과 같습니다.

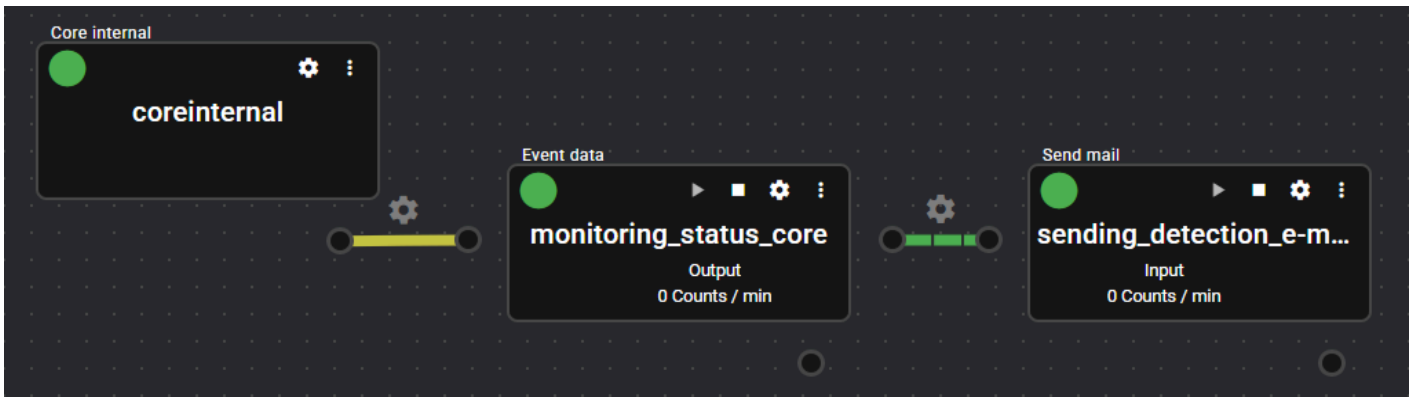


항목	설명
명칭	컴포넌트의 이름（coredbquery 고정）
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
주소	수신측의 IP 주소를 입력, 0.0.0.0이면 모든 네트워크로부터의 통신을 허가 ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다.
포트	DB 쿼리를 접수하는 포트를 입력 ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다.
토큰 인증	Web API 요청 시의 인증용 액세스 토큰을 선택
액세스 토큰 발행	액세스 토큰 발행 화면을 표시 ※액세스 토큰 화면의 상세는,「액세스 토큰 발행」을 참조하십시오.

-For Reference Only-

5.11.2 코어 인터널

SpeeDBee Synapse 내의 내부 스테이터스를 관리하는 컴포넌트입니다. 이 컴포넌트는, 사용자가 기동하거나 정지를 수행할 수는 없습니다. 자동적으로 기동되어, 상시 동작하는 특수한 컴포넌트입니다.
가동 머신의 전원 차단에 의해 데이터의 결손이 있었던 경우, 검지한 에러를 메일로 통지할 수 있습니다. 순서의 상세는, 트러블슈트를 참조하십시오.



5.11.2.1 기본 정보

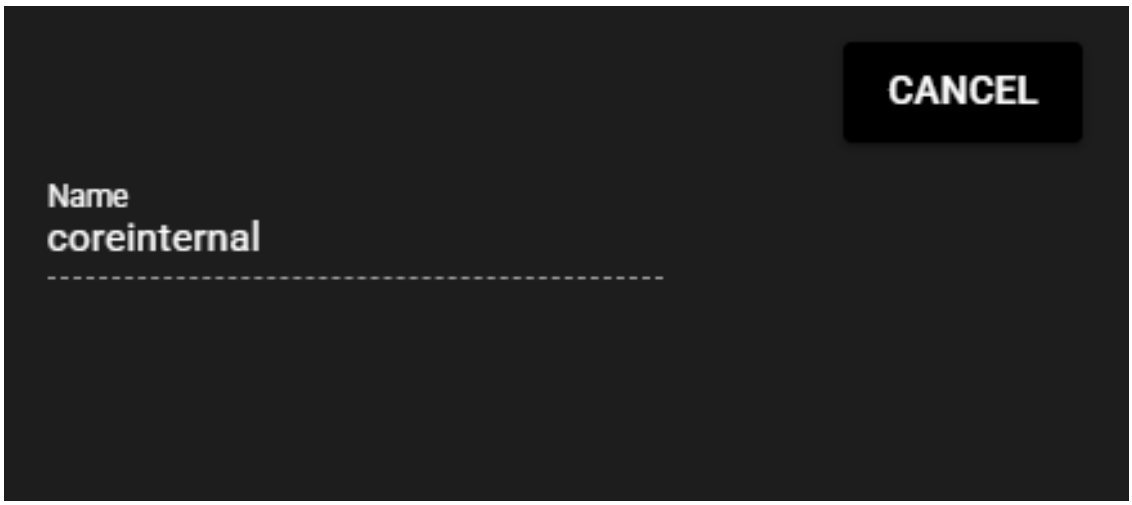
코어 인터널의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	코어 인터널
설명	코어 내에서 정기적인 실행이 필요한 처리를 수행하는 컴포넌트입니다. 이 컴포넌트는, SpeeDBee Synapse에서 1개밖에 존재하지 않습니다. 또한, 설정 변경 및 정지 등 제어하는 것도 할 수 없습니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.
소비 포인트	0
포트	입력 : 0 출력 : 0

AI 번역본
-For Reference Only-

5.11.2.2 설정 항목

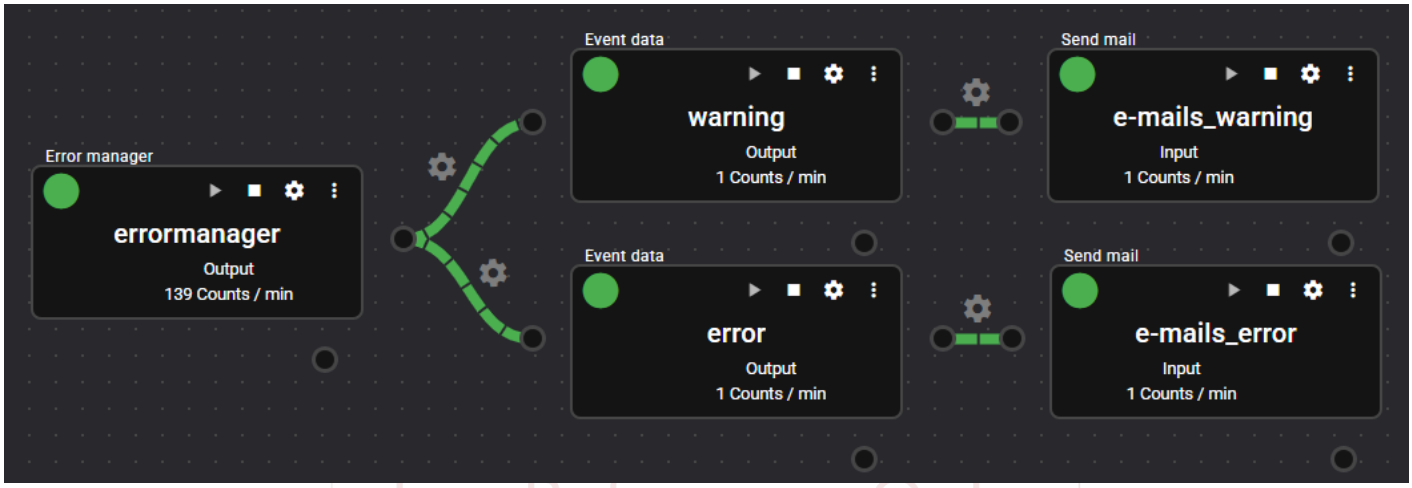
코어 인터널은 설정을 변경할 수 없는 특수한 컴포넌트입니다.



항목	설명
명칭	컴포넌트의 이름 (coreinternal 고정)

5.11.3 에러 매니저

SpeeDBee Synapse 내에서 발생한 전 컴포넌트의 에러를 검지할 수 있는 컴포넌트입니다.
검지한 에러를 메일로 통지할 수 있습니다. 순서의 상세는, 트러블슈트를 참조하십시오.



-For Reference Only-

5.11.3.1 기본 정보

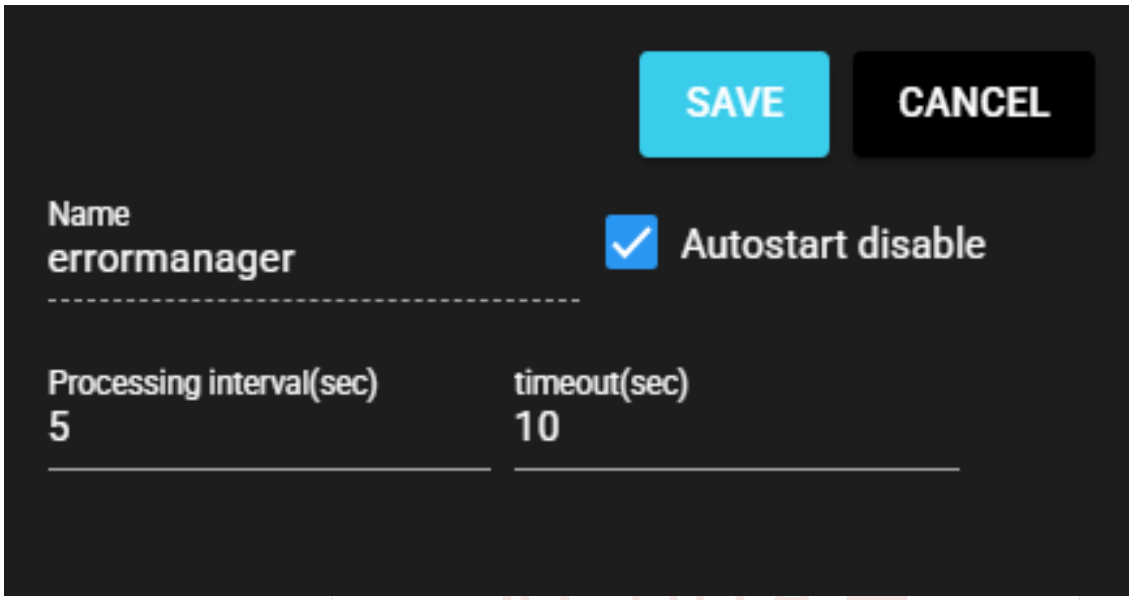
에러 매니저의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	에러 매니저

항목	설명
설명	SpeeDBee Synapse 내에서 발생한 전 에러를 검지하는 컴포넌트입니다. 이 컴포넌트는, SpeeDBee Synapse에서 1개밖에 존재하지 않습니다. 출력하는 에러 정보의 상세는, 「에러 코드 목록」을 참조하십시오.
소비 포인트	0
포트	입력 : 0 출력 : 1

5.11.3.2 설정 항목

에러 매니저의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름（errormanager 고정）
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
처리 간격	에러 정보를 수집하는 처리 간격（초 단위）
타임아웃	에러 정보를 수집하는 처리의 타임아웃 시간（초 단위）

5.11.3.3 출력 데이터 항목

에러 매니저는, 다음 데이터를 출력합니다.

항목	컬럼명	데이터형	설명
정지 컴포넌트 수	standby_component_count	uint16	지난번 에러 정보 수집부터 이번까지의, 정지하고 있는 컴포넌트의 수의 변화
가동 컴포넌트 수	running_component_count	uint16	지난번 에러 정보 수집부터 이번까지의, 가동하고 있는 컴포넌트의 수의 변화
경고 컴포넌트 수	warning_component_count	uint16	지난번 에러 정보 수집부터 이번까지의, 경고 상태로 가동하고 있는 컴포넌트의 수의 변화
이상 컴포넌트 수	error_component_count	uint16	지난번 에러 정보 수집부터 이번까지의, 에러 상태로 정지하고 있는 컴포넌트의 수의 변화
경고 컴포넌트명 목록	warning_components	string	지난번 에러 정보 수집부터 이번까지의, 경고 상태로 가동하고 있는 컴포넌트의 이름의 목록（콤마 구분）의 변화
이상 컴포넌트명 목록	error_components	string	지난번 에러 정보 수집부터 이번까지의, 에러 상태로 정지하고 있는 컴포넌트의 이름의 목록（콤마 구분）의 변화
최신의 정지 컴포넌트 수	latest_standby_component_count	uint32	에러 정보 수집 시의, 정지하고 있는 컴포넌트의 수
최신의 가동 컴포넌트 수	latest_running_component_count	uint32	에러 정보 수집 시의, 가동하고 있는 컴포넌트의 수
최신의 경고 컴포넌트 수	latest_warning_component_count	uint32	에러 정보 수집 시의, 경고 상태로 가동하고 있는 컴포넌트의 수
최신의 이상 컴포넌트 수	latest_error_component_count	uint32	에러 정보 수집 시의, 에러 상태로 정지하고 있는 컴포넌트의 수

AI 번역본
-For Reference Only-

항목	컬럼명	데이터형	설명
최신의 경고 컴포넌트명 목록	latest_warning_components	string	에러 정보 수집 시의, 경고 상태로 가동하고 있는 컴포넌트의 이름의 목록（콤마 구분）
최신의 이상 컴포넌트명 목록	latest_error_components	string	에러 정보 수집 시의, 에러 상태로 정지하고 있는 컴포넌트의 이름의 목록（콤마 구분）

5.11.4 이벤트 데이터 매니저

이 컴포넌트는, i-Trigger에서 기록되는 이벤트 데이터의 정기 삭제 등을 수행하는 특수한 컴포넌트입니다. i-Trigger를 사용하지 않는 경우는, 기동할 필요는 없습니다.

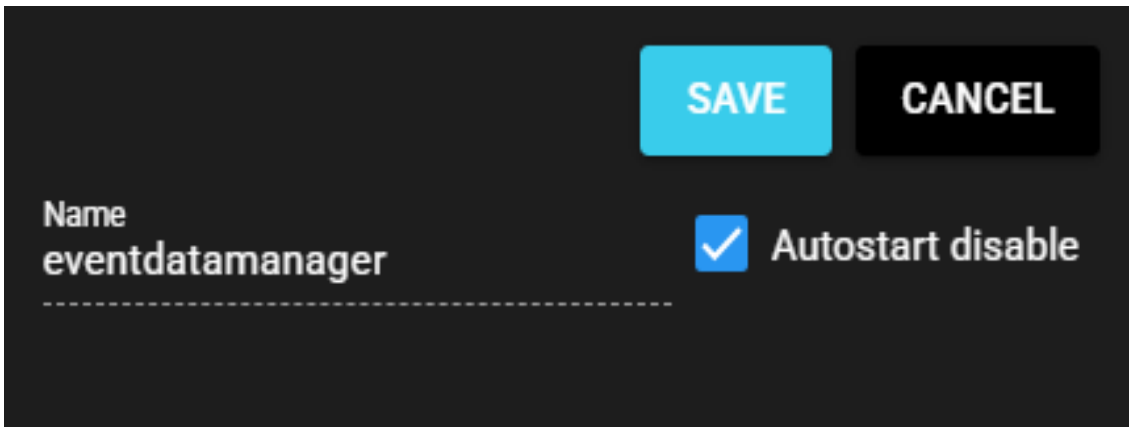
5.11.4.1 기본 정보

이벤트 데이터 매니저의 기본 정보는, 다음과 같습니다.

항목	설명
베이스명	이벤트 데이터 매니저
설명	이벤트 데이터의 데이터 관리를 수행합니다. Event Recorder에 의해 저장된 각종 데이터를 정기 삭제합니다. 또한, 내부 데이터베이스의 백업, 리스토어에 대해서도 처리합니다. 이 컴포넌트는, SpeeDBee Synapse에서 1 개밖에 존재하지 않습니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.
소비 포인트	0
포트	입력：0 출력：0

5.11.4.2 설정 항목

이벤트 데이터 매니저의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름（eventdatamanager 고정）
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON

5.11.4.3 데이터 관리의 동작 사양

이벤트 데이터 매니저는, 데이터 관리를 다음과 같이 수행합니다. 본 항은, 이벤트 데이터 매니저의 데이터 관리에 대해, 확인하고 싶은 경우에만 참조하십시오.

Event Recorder가 저장한 데이터의 정기 삭제

Event Recorder의 설정에는,「저장 시간」의 설정 항목이 있습니다. Event Recorder가 저장한 데이터（이벤트 데이터 JSON 파일, 동영상 파일）를, 지정된 저장 시간에 따라 삭제합니다.

- 저장 시간을 도중에 변경한 경우, 최신의 저장 시간을 기준으로 삭제를 수행합니다
- Event Recorder의 인스턴스를 삭제해도, 이벤트 데이터 매니저가 기동하고 있는 한, 저장 시간을 경과한 데이터는 삭제됩니다

뷰어용 내부 데이터베이스 파일의 백업・리스토어

Event Recorder는, 이벤트의 기록 시의 데이터를 내부의 데이터베이스에서 관리합니다. 이 데이터베이스는, i-Trigger 전용입니다. 이 데이터베이스를 정기적으로 백업하여, 파손 시는 리스토어를 수행합니다.

- 백업은, 1일 단위로 처리됩니다（일시의 전환 10분 이내에 처리됩니다）
- 파손 시의 리스토어는, 1일 전의 데이터베이스로부터 복원되기 때문에, 최신의 데이터는 소실됩니다
- 디스크 풀 시는, 백업/리스토어도 에러가 됩니다

5.12 컴포넌트 공통의 설정 항목

여기에서는, 컴포넌트의 포트의 설정, 플로우 링크의 필터의 설정에 대해 설명합니다.

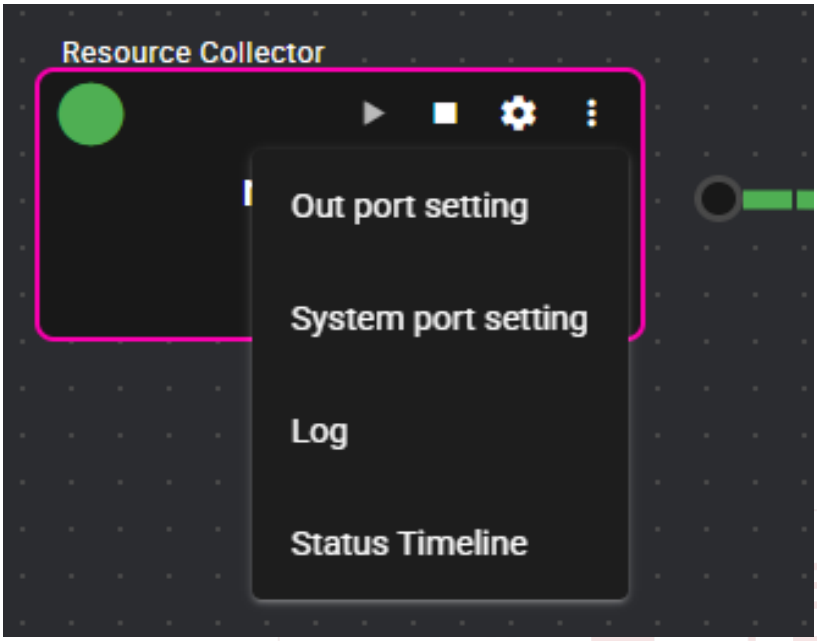
5.12.1 출력 포트

출력 포트는, 플로우 링크의 접속원이며, 컴포넌트가 데이터를 출력하는 포트입니다. 출력 포트에서는, 출력되는 데이터에 대해, 영속화, 등록 조건, 수치 변환의 유무 등의 설정을 수행할 수 있습니다.

5.12.1.1 영속화의 설정

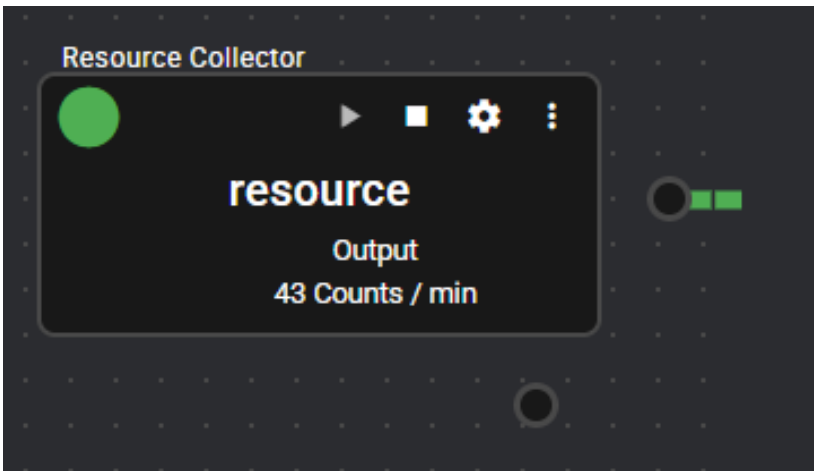
영속화를 유효로 한 데이터는, 스토리지 내에 저장되고, 저장되어 있는 동안, SQL에서의 참조, Grafana 연계에서의 참조가 가능해집니다. 다음 순서로, 컴포넌트의 출력 데이터에 대해 영속화의 설정을 수행할 수 있습니다.

1. 다음 중 어느 하나의 방법으로 출력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여「 출력 포트 설정 」을 선택

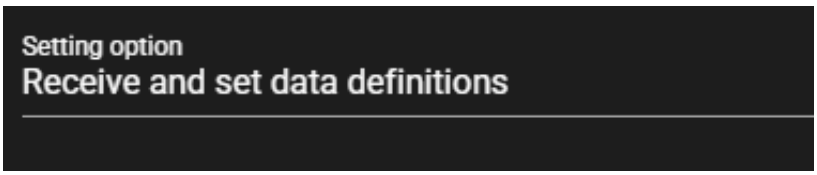


   - 출력 포트를 오른쪽 클릭

-For Reference Only-

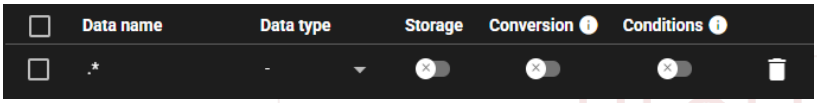


2. 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정하다	출력 데이터를 목록으로 표시하고, 개별적으로 영속화의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 출력 데이터 중에서 매칭된 것에 대해, 영속화의 설정을 수행한다

3. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
데이터명	출력 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	출력 데이터 중, 설정 대상으로 하는 것의 데이터형

정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

- 「data1」에만 매칭 : data1
- 이름에「abc」를 포함하는 데이터에 매칭 : .*abc.*
- 이름이「abc」로 시작하는 데이터에 매칭 : abc.*
- 이름이「abc」로 끝나는 데이터에 매칭 : .*abc

4. 다음 중 어느 하나의 방법을 수행하여, 영속화의 설정 다이얼로그를 엽니다.
   - 영속화의 설정을 수행할 데이터의「영속화」의 아이콘을 누릅니다.



   - 영속화의 설정을 수행할 데이터에 체크를 붙이고,「선택 중인 데이터를 편집」을 눌러「영속화」를 선택합니다. 체크를 붙인 데이터에, 동일한 설정 내용이 적용됩니다.

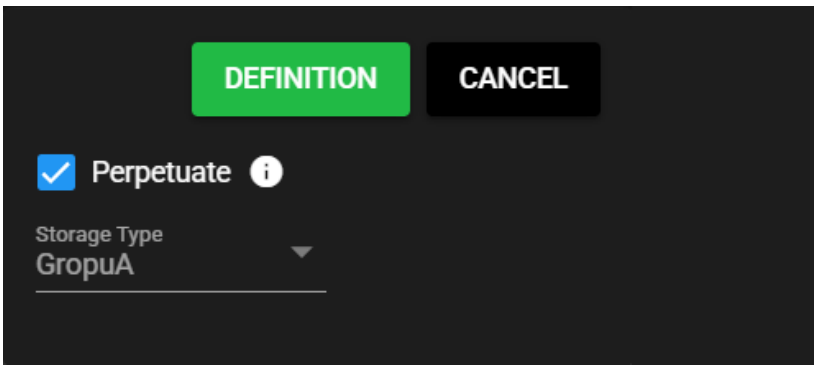


5.「영속화한다」를 ON으로 합니다.



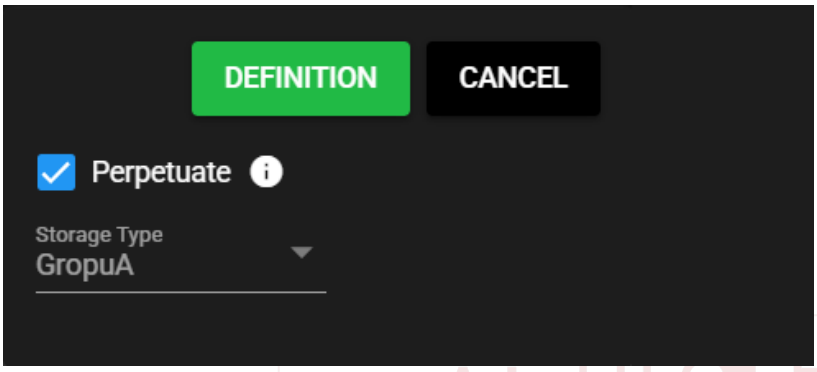
6. 다음 항목을 입력합니다.

AI 번역본
-Only-



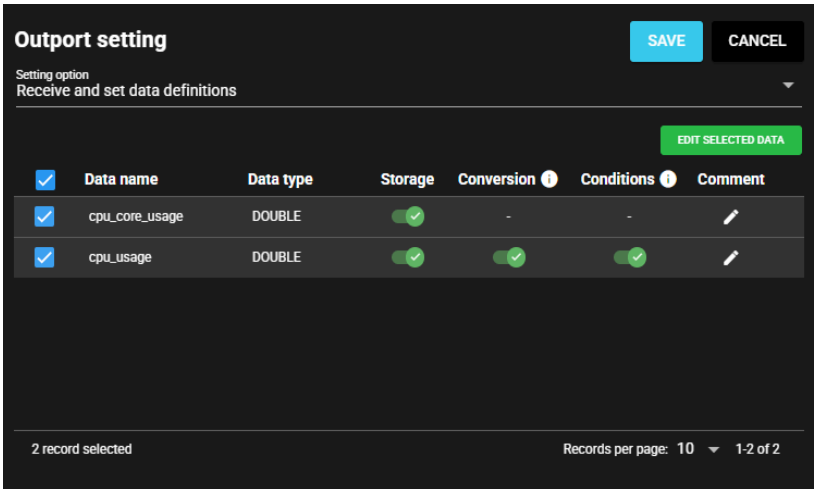
항목	설명
스토리지 타입	시계열 데이터베이스에 저장하는 기간. 서버 설정의「그룹 단위로 저장 기간을 지정」에서, GroupA・GroupB・GroupC의 저장 기간을 확인・변경할 수 있습니다.「그룹 단위로 저장 기간을 지정」이 ON인 경우만 변경할 수 있습니다. 디폴트 값 ・GroupA : 30분 ・GroupB : 24시간 ・GroupC : 7일간

7.「확정」을 누릅니다.



8.「저장」을 누릅니다.

AI 번역본
-For Reference Only-

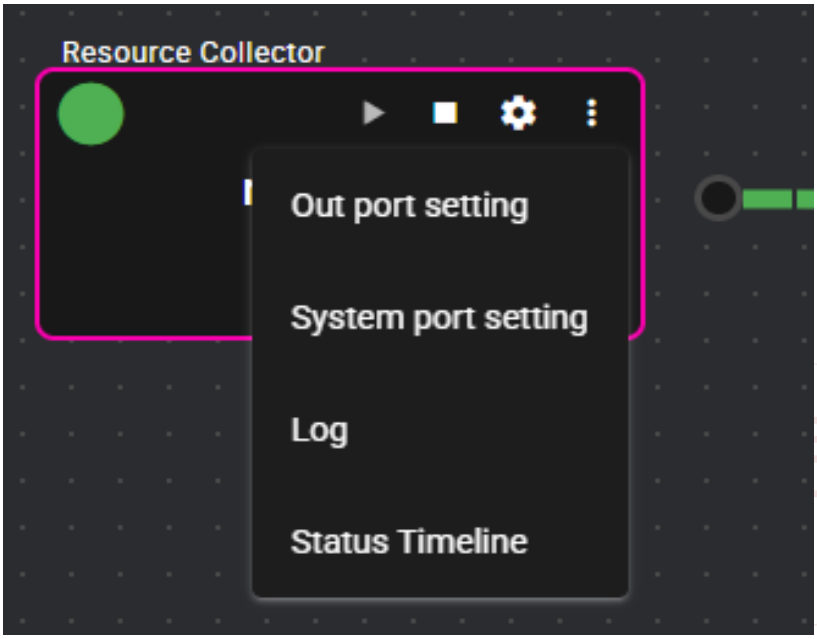


9. 영속화의 설정이 적용됩니다.

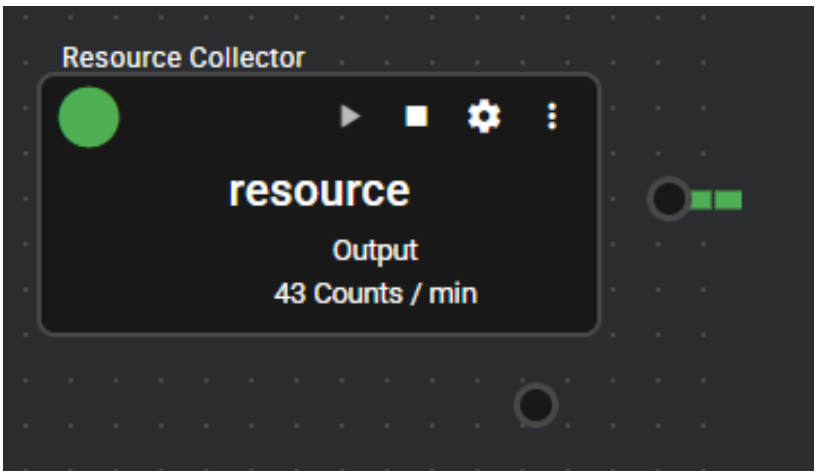
5.12.1.2 수치 변환의 설정

수치 변환은, 원래의 데이터를 그대로 등록하는 것이 아니라, 연산한 결과를 데이터 등록하기 위한 기능입니다. 다음 순서로, 컴포넌트의 출력 데이터에 대해 수치 변환의 설정을 수행할 수 있습니다.

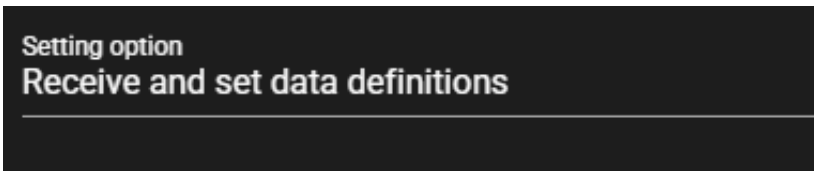
1. 다음 중 어느 하나의 방법으로 출력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여 「 출력 포트 설정 」을 선택



   - 출력 포트를 오른쪽 클릭



2. 「 설정 방법 」을, 다음 중에서 선택합니다.



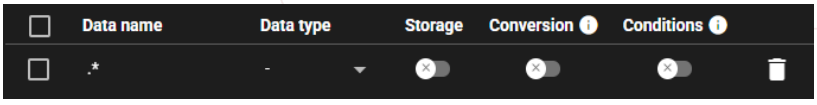
항목	설명
데이터 정의를 수신하여 설정 한다	출력 데이터를 목록으로 표시하고, 개별적으로 수치 변환의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 출력 데이터 중에서 매칭된 것에 대해, 수치 변환의 설정을 수행한다

배열형의 출력 데이터는 대상 외

배열형의 출력 데이터는, 수치 변환의 대상 외입니다.
정규 표현식의 조건에서, 데이터명이 해당하는 경우에도, 배열형이었던 경우는 수치 변환은 처리되지 않습니다. 예를 들면, 리소스 컬렉터의 "cpu_core_usage"가 해당합니다.

AI 번역본
-For Reference Only-

3. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
데이터명	출력 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식

항목	설명
데이터형	출력 데이터 중, 설정 대상으로 하는 것의 데이터형

> **정규 표현식의 입력 예**
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「data1」에만 매칭 : data1
> - 이름에「abc」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이「abc」로 시작하는 데이터에 매칭 : abc.*
> - 이름이「abc」로 끝나는 데이터에 매칭 : .*abc

4. 다음 중 어느 하나의 방법을 수행하여, 수치 변환의 설정 다이얼로그를 엽니다.
   - 수치 변환의 설정을 수행할 데이터의「수치 변환」의 아이콘을 누릅니다.



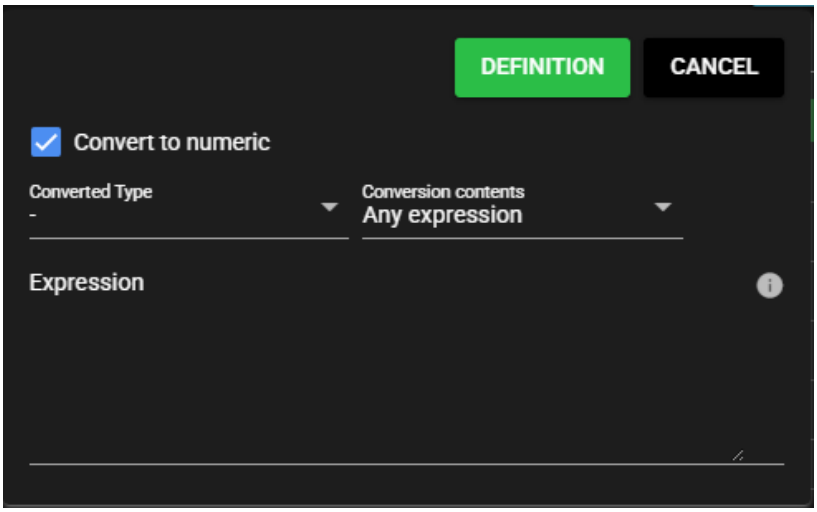
   - 수치 변환의 설정을 수행할 데이터에 체크를 붙이고,「선택 중인 데이터를 편집」을 눌러「수치 변환」을 선택합니다. 체크를 붙인 데이터에, 동일한 설정 내용이 적용됩니다.



5.「수치 변환을 수행한다」를 ON으로 합니다.

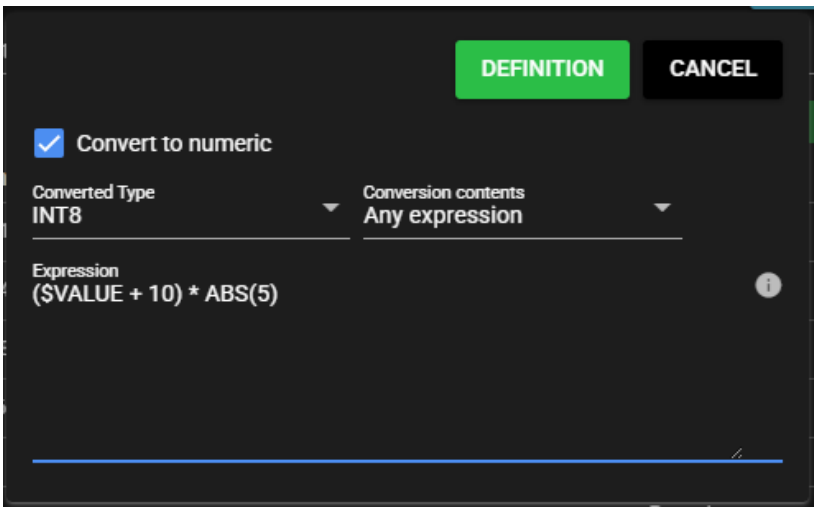


6. 다음 항목을 입력합니다.

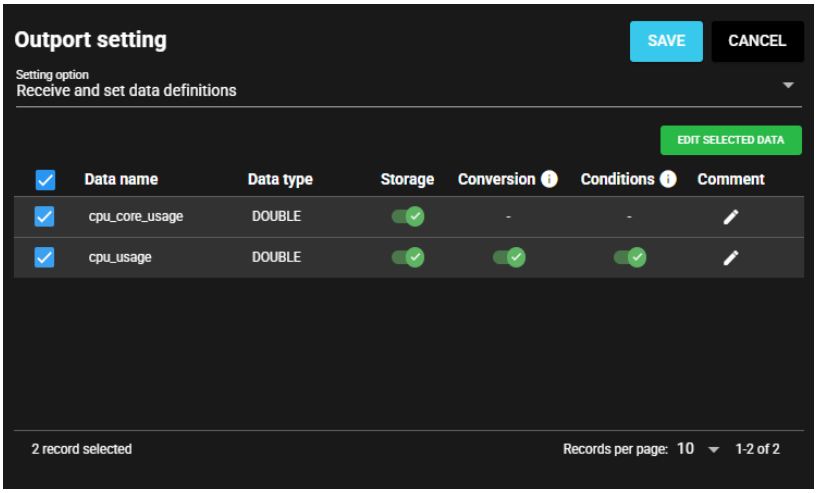


항목	설명
변환 후의 형	수치 변환 후의 데이터형
변환 내용	어떠한 수치 변환식을 기술할지의 선택 ※선택할 수 있는 변환 내용은 다음과 같습니다. ・임의의 수식 ・BCD(4자리) → INT ・BCD(8자리) → INT ・BCD(16자리) → INT ※「 임의의 수식 」을 선택하면 수치 변환에서 실행할 변환식을 임의로 입력할 수 있습니다. ※「 BCD(4자리) → INT 」,「 BCD(8자리) → INT 」,「 BCD(16자리) → INT 」를 선택하면 BCD 값을 정수값으로 변환하는 수식이 자동으로 입력됩니다. 이들을 선택하고 있는 상태에서는 변환식의 편집을 수행할 수 없습니다. 변환식의 편집을 수행하는 경우에는 「 임의의 수식 」을 선택하십시오 ※BCD 값에서 정수값으로 변환한 데이터는 정수형이 됩니다. PLC 쓰기나 EtherNet/IP 쓰기에서 변환 후의 값을 BCD를 다루는 주소에 쓰면 올바른 값이 되지 않습니다
변환식	수치 변환에서 실행하는 식 ※입력 에어리어 오른쪽 끝의 아이콘을 클릭하면, 입력 예와 식에 사용할 수 있는 요소 목록이 표시됩니다.

7.「 확정 」을 누릅니다.



8. 「 저장 」을 누릅니다.



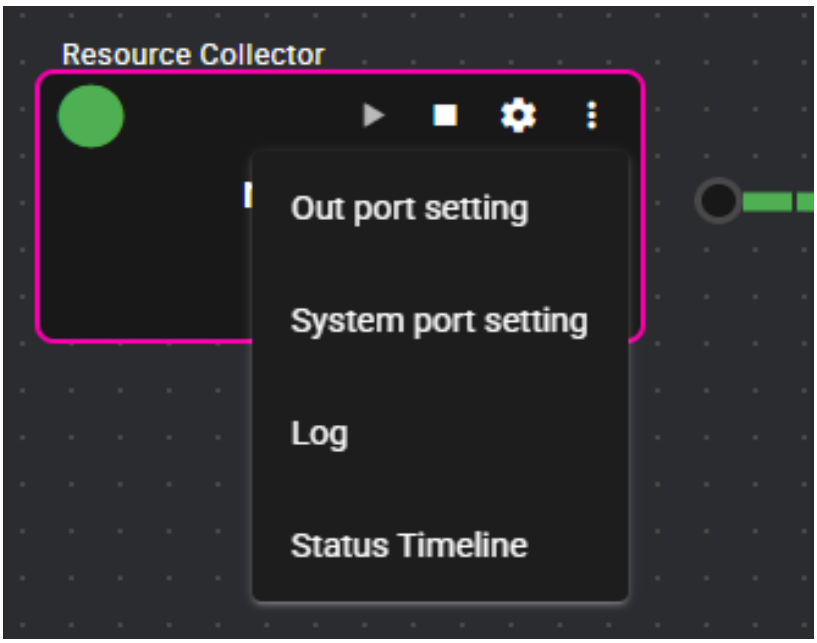
9. 수치 변환의 설정이 적용됩니다.

5.12.1.3 등록 조건의 설정

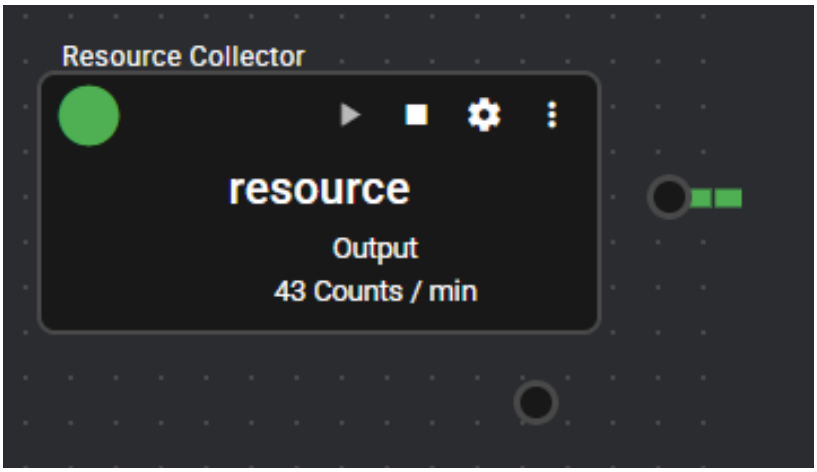
등록 조건은, 조건에 합치한 데이터만을 등록하는 기능입니다. 조건에 합치하지 않는 데이터는, 비해당 데이터로서 등록되지 않습니다. 다음 순서로, 컴포넌트의 출력 데이터에 대해 등록 조건의 설정을 수행할 수 있습니다.

AI 번역본
-For Reference Only-

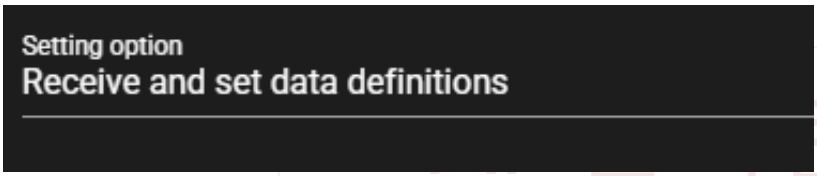
1. 다음 중 어느 하나의 방법으로 출력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여 「 출력 포트 설정 」을 선택



- 출력 포트를 오른쪽 클릭



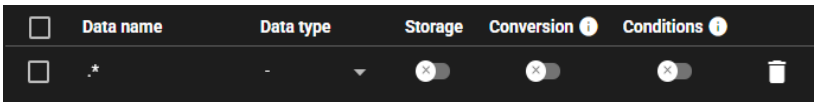
2. 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	출력 데이터를 목록으로 표시하고, 개별적으로 등록 조건의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 출력 데이터 중에서 매칭된 것에 대해, 등록 조건의 설정을 수행한다

> **배열형의 출력 데이터는 대상 외**
>
> 배열형의 출력 데이터는, 등록 조건의 대상 외입니다.
> 정규 표현식의 조건에서, 데이터명이 해당하는 경우에도, 배열형이었던 경우는 등록 조건은 처리되지 않습니다. 예를 들면, 리소스 컬렉터의 "cpu_core_usage"가 해당합니다.

3. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
데이터명	출력 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	출력 데이터 중, 설정 대상으로 하는 것의 데이터형

> **정규 표현식의 입력 예**
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「 data1 」에만 매칭 : data1
> - 이름에 「 abc 」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이 「 abc 」로 시작하는 데이터에 매칭 : abc.*
> - 이름이 「 abc 」로 끝나는 데이터에 매칭 : .*abc

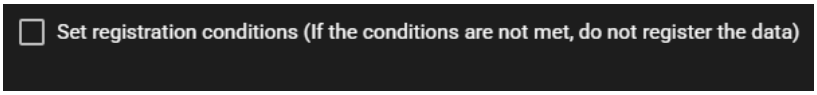
4. 다음 중 어느 하나의 방법을 수행하여, 등록 조건의 설정 다이얼로그를 엽니다.
   - 등록 조건의 설정을 수행할 데이터의 「 등록 조건 」의 아이콘을 누릅니다.



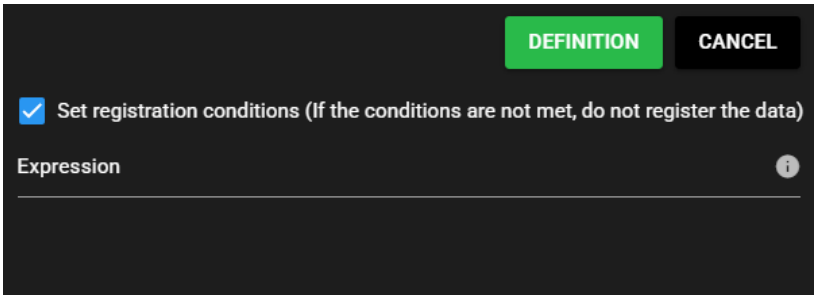
   - 등록 조건의 설정을 수행할 데이터에 체크를 붙이고, 「 선택 중인 데이터를 편집 」을 눌러 「 등록 조건 」을 선택합니다. 체크를 붙인 데이터에, 동일한 설정 내용이 적용됩니다.



5. 「 등록 조건을 설정한다 」를 ON으로 합니다.

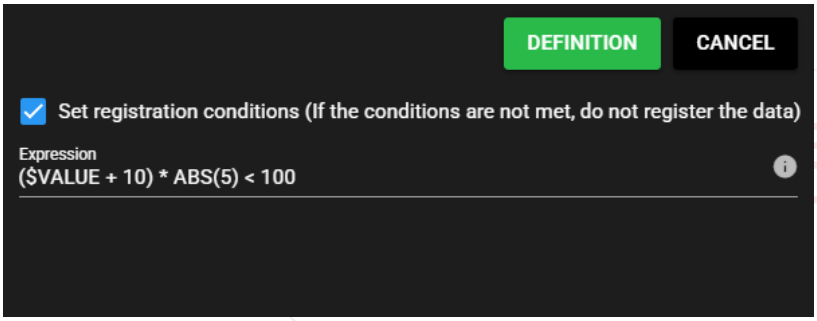


6. 다음 항목을 입력합니다.

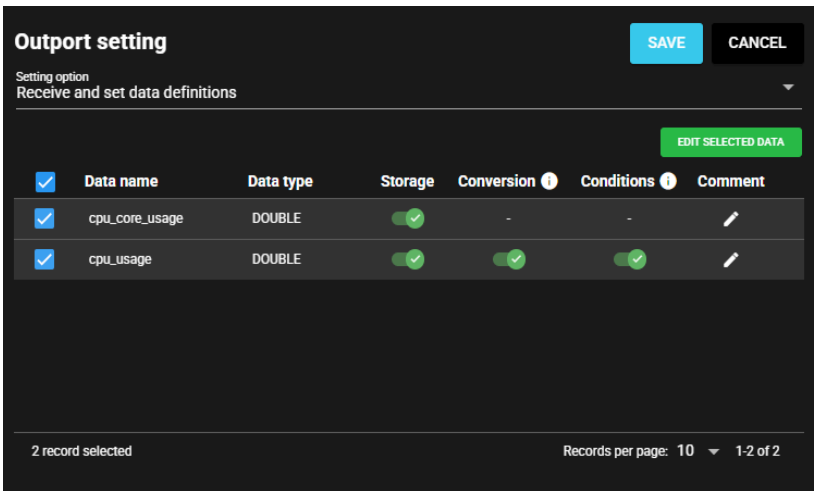


항목	설명
조건식	등록할 데이터를 판정하는 조건식 ※입력 에어리어 오른쪽 끝의 아이콘을 클릭하면, 입력 예와 식에 사용할 수 있는 요소 목록이 표시됩니다.

7. 「 확정 」을 누릅니다.



8. 「 저장 」을 누릅니다.



9. 등록 조건의 설정이 적용됩니다.

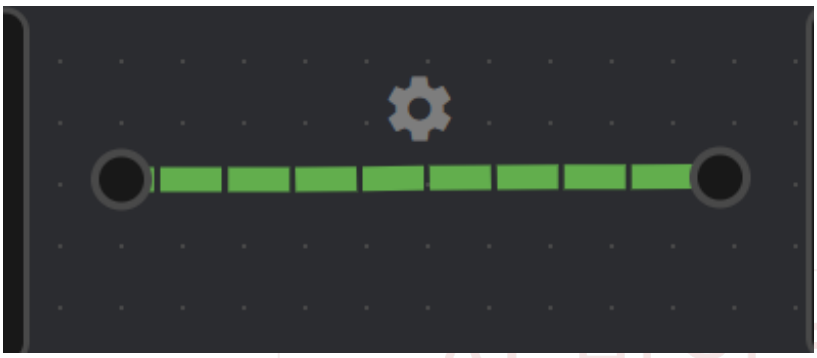
5.12.2 필터

필터는, 각 데이터를 플로우 링크상에서 통과시킬지 여부를 선택할 수 있는 기능입니다.

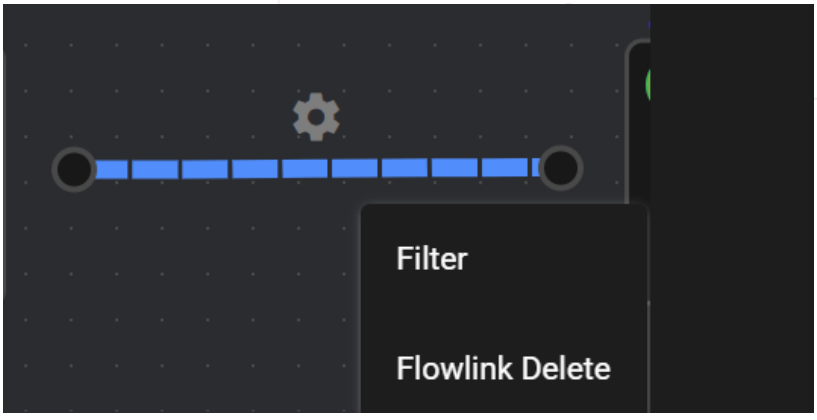
5.12.2.1 설정 방법

다음 순서로, 필터를 설정할 수 있습니다.

1. 플로우 링크를 오른쪽 클릭합니다.

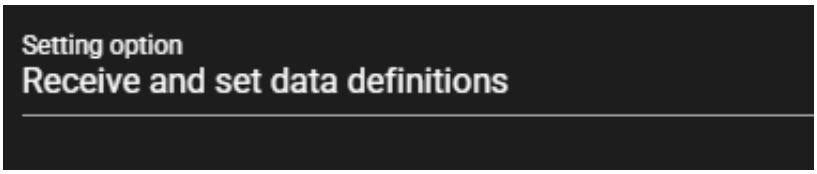


2. 「 필터 」를 선택합니다.



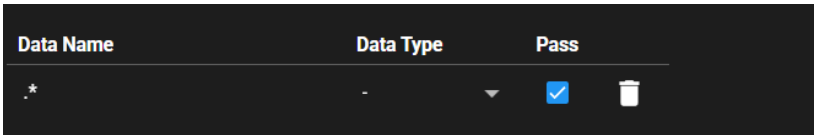
AI 학습본
-Only-

3. 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정한다	플로우 링크의 접속원의 데이터를 목록으로 표시하고, 개별적으로 필터의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 플로우 링크의 접속원의 데이터 중에서 매칭된 것에 대해, 필터의 설정을 수행한다

4. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



항목	설명
데이터명	플로우 링크의 접속원의 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	플로우 링크의 접속원의 데이터 중, 설정 대상으로 하는 것의 데이터형

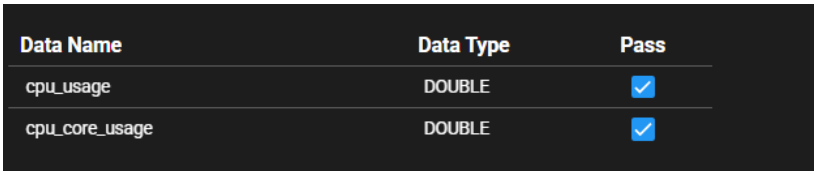
정규 표현식의 입력 예

정규 표현식의 입력 예는 다음과 같습니다.

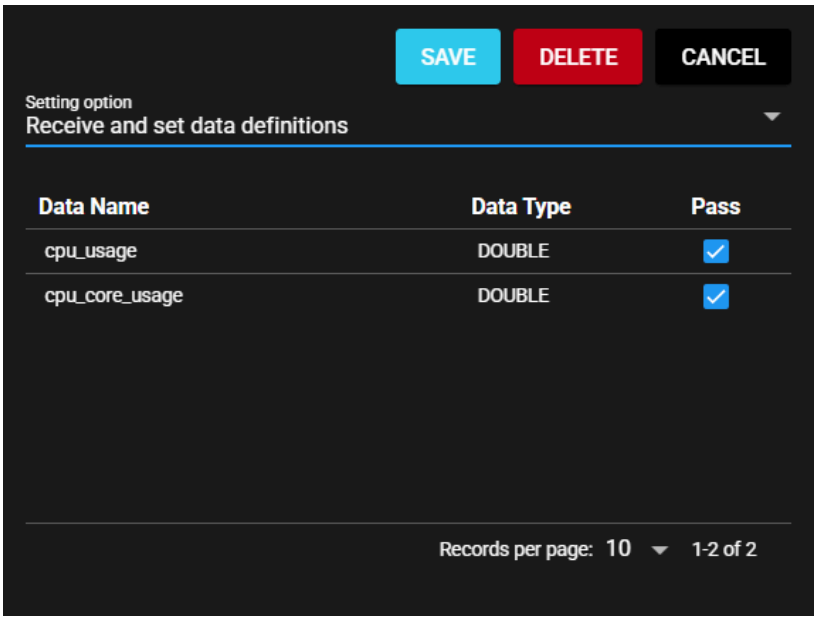
- 「 data1 」에만 매칭 : data1
- 이름에 「 abc 」를 포함하는 데이터에 매칭 : .*abc.*
- 이름이 「 abc 」로 시작하는 데이터에 매칭 : abc.*
- 이름이 「 abc 」로 끝나는 데이터에 매칭 : .*abc

AI 번역본
-For Reference Only-

5. 통과시켜, 플로우 링크의 접속처로 넘길 데이터의 「 통과 」를 ON으로 합니다.



6. 「 저장 」을 누릅니다.



7. 필터의 설정이 적용됩니다. 설정이 적용된 플로우 링크는, 아이콘의 색이 변화합니다.



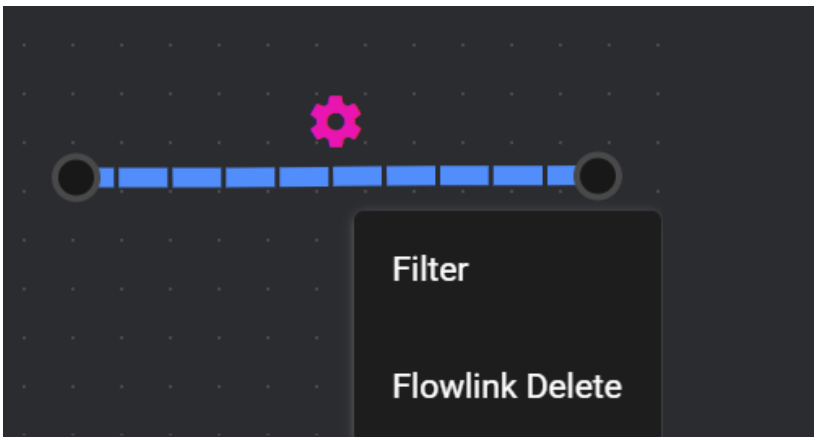
5.12.2.2 삭제

다음 순서로, 필터를 삭제할 수 있습니다.

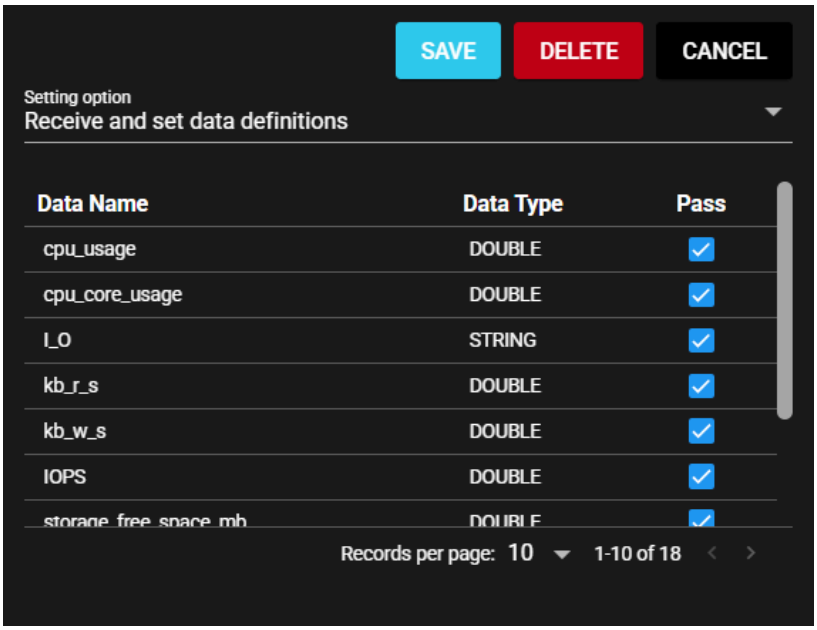
1. 필터가 설정된 플로우 링크를 오른쪽 클릭합니다.



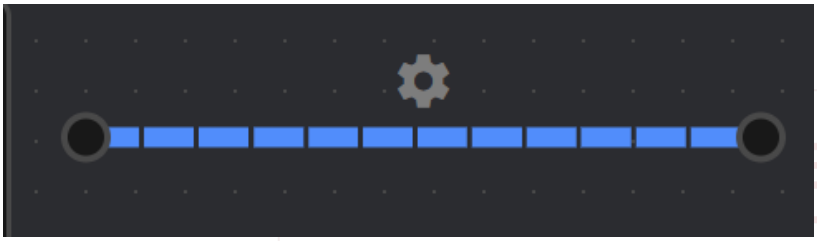
2. 「 필터 」를 선택합니다.



3. 「 삭제 」를 누르고, 확인 다이얼로그에서 「 예 」를 선택합니다.



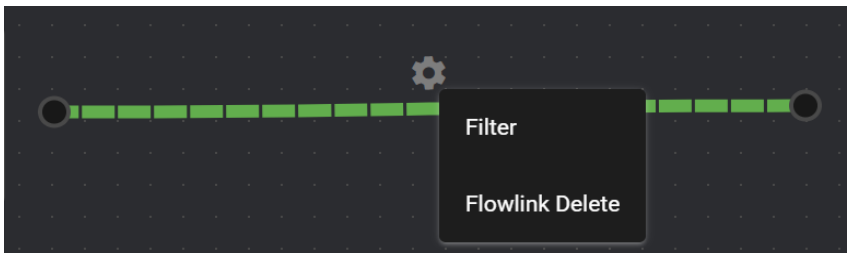
4. 필터의 설정이 삭제됩니다. 아이콘의 색이 변화합니다.



-For Reference Only-

5.12.3 플로우 링크 아이콘

플로우 링크의 접속 시에 출현하는 아이콘을 왼쪽 클릭하면, 플로우 링크를 오른쪽 클릭했을 때와 동일한 조작이 가능해집니다. 필터 설정에 관한 상세는 「 필터 」를 참조하십시오. 플로우 링크 삭제에 관한 상세는「 삭제 」를 참조하십시오.



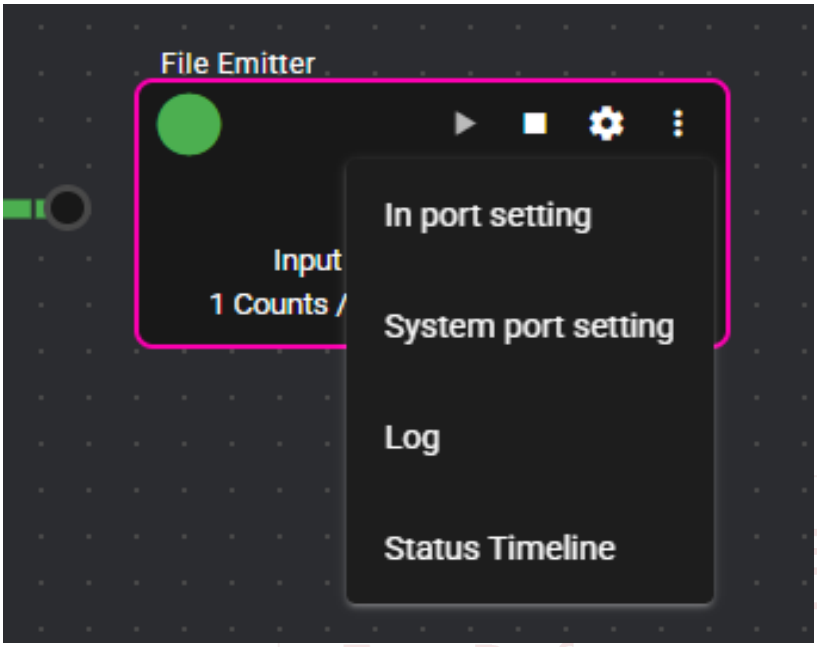
5.12.4 입력 포트

입력 포트는, 플로우 링크의 접속처이며, 컴포넌트가 데이터를 받아들이는 포트입니다. 입력 포트에서는, 받아들인 데이터에 대해, 이벤트의 사용・집약의 유무 등의 설정을 수행할 수 있습니다.

5.12.4.1 지연 시간의 설정

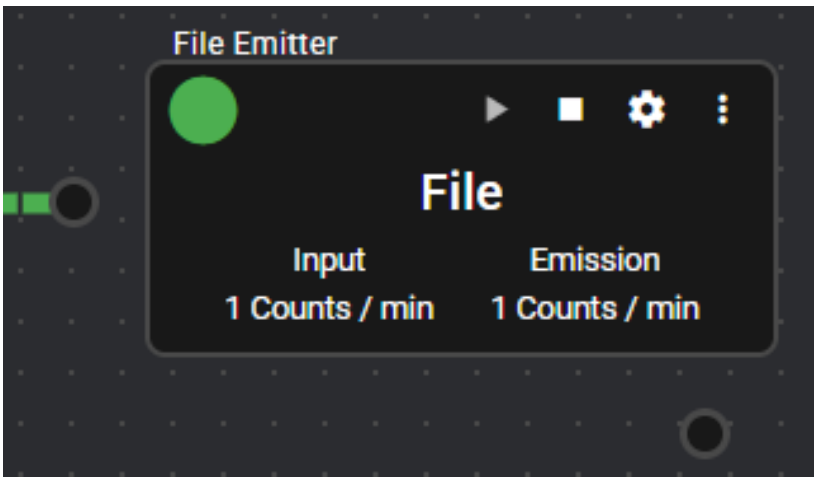
다음 순서로, 컴포넌트가 받아들인 데이터에 대해 지연 시간의 설정을 수행할 수 있습니다.

1. 다음 중 어느 하나의 방법으로 입력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여「입력 포트 설정」을 선택



   - 입력 포트를 오른쪽 클릭

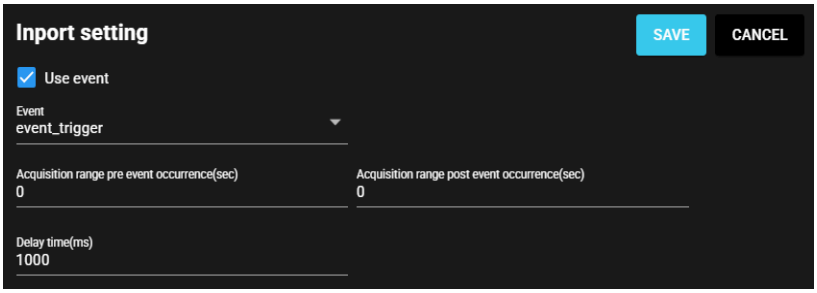
-For Reference Only-



2. 「 지연 시간(밀리초) 」에, 데이터의 수집 지연 시간을 밀리초 단위로 입력합니다.
   자세히는 「 데이터 수집에 관련된 동작 사양에 대해 」의 「 데이터 참조 시의 지연 시간에 대해 」를 참조하십시오



3. 「 저장 」을 누릅니다.



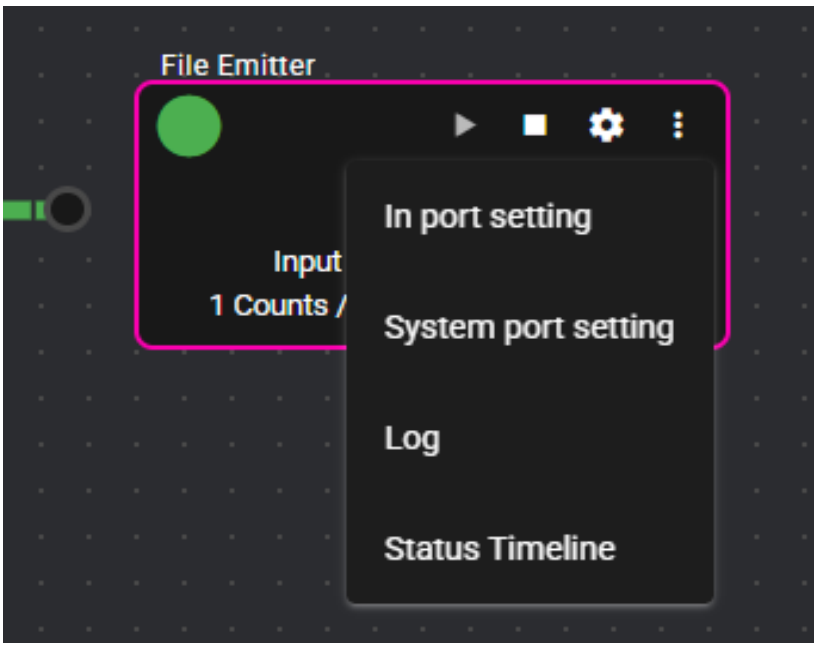
4. 지연 시간의 설정이 적용됩니다.

5.12.4.2 이벤트의 설정

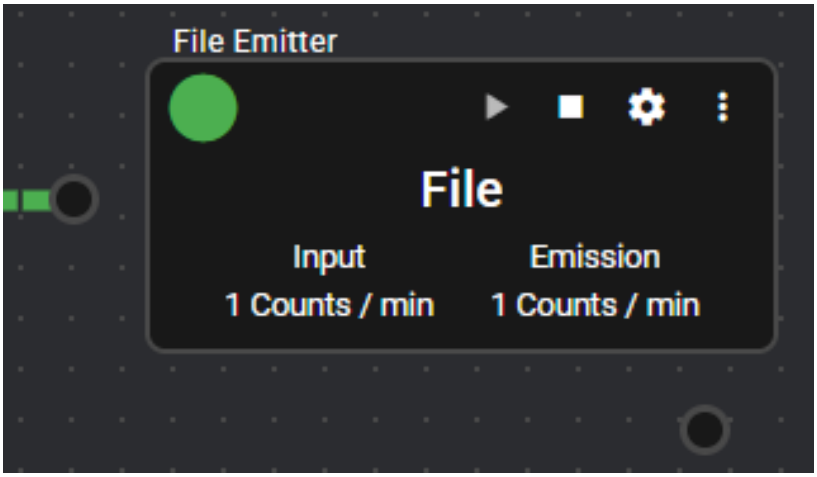
다음 순서로, 컴포넌트가 받아들인 데이터에 대해 이벤트의 설정을 수행할 수 있습니다.

1. 이벤트 트리거를 등록합니다. 상세는 「 이벤트 트리거 」를 참조하십시오.
2. 이벤트 트리거에서 컴포넌트로 플로링을 연결합니다. 상세는 「 플로우 링크의 추가 」를 참조하십시오.
3. 다음 중 어느 하나의 방법으로 입력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여 「 입력 포트 설정 」을 선택

AI 번역본
-For Reference Only-



◦ 입력 포트를 오른쪽 클릭



4. 「 이벤트를 사용한다 」를 ON으로 합니다.

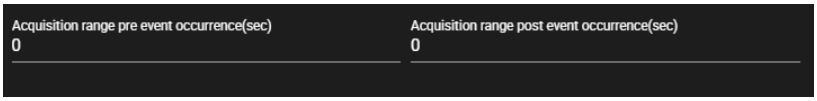


AI 번역본
For Reference Only-

5. 「 이벤트 」에, 등록한 이벤트 트리거를 선택합니다.

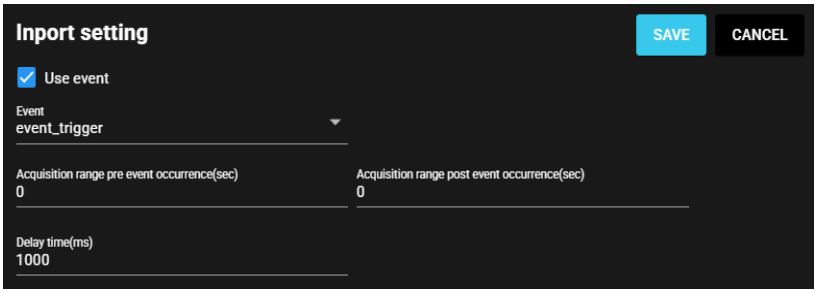


6. 다음 항목을 입력합니다.



항목	설명
이벤트 발생 전의 취득 범위(초)	이벤트 발생 전의 데이터를 취득하는 범위
이벤트 종료 후의 취득 범위(초)	이벤트 발생 후의 데이터를 취득하는 범위

7. 「 저장 」을 누릅니다.

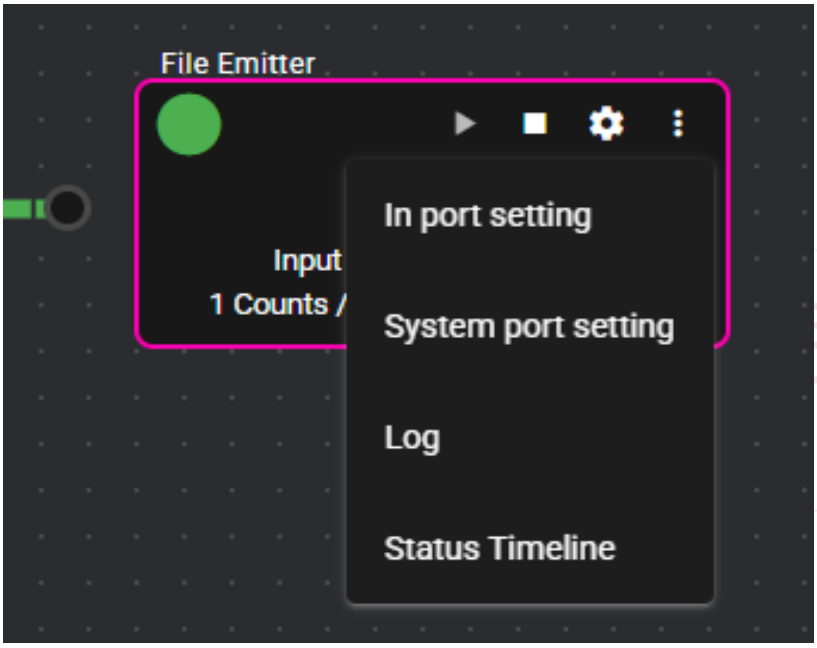


8. 이벤트의 설정이 적용됩니다.

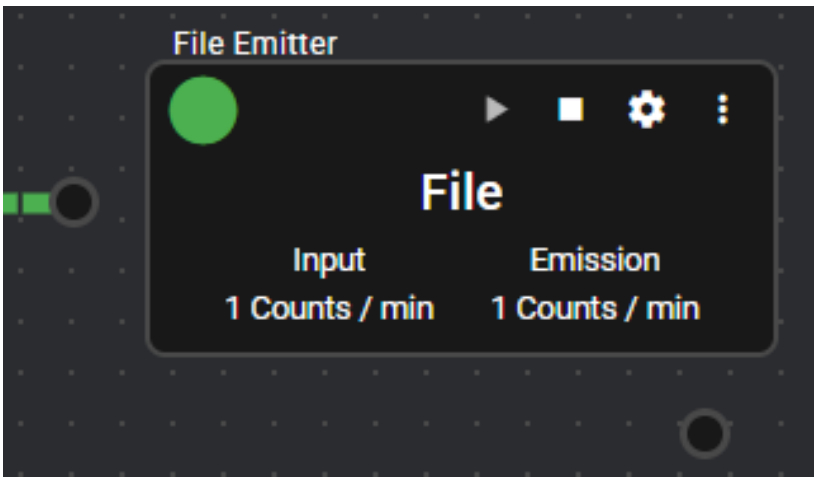
5.12.4.3 집약의 설정

다음 순서로, 집약의 설정을 수행할 수 있습니다.

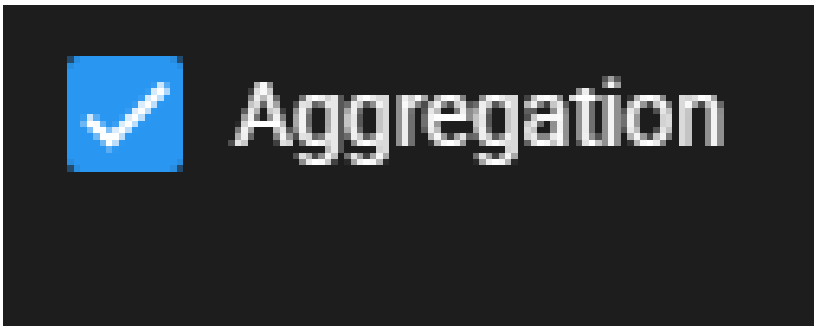
1. 다음 중 어느 하나의 방법으로 입력 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여 「 입력 포트 설정 」을 선택



   - 입력 포트를 오른쪽 클릭



2.「 집약한다 」를 ON으로 합니다.

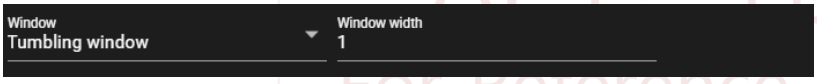


3.「 윈도우 함수 」를 다음 중에서 선택합니다.



항목	설명
텀블링 윈도우	지정한 시간 폭으로, 집약 대상의 데이터의 범위를 구분한다

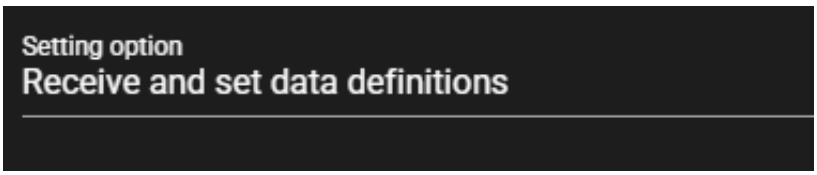
4. 윈도우 폭을 지정합니다. 선택한 윈도우 함수에 따라, 입력하는 항목이 다릅니다.



항목		설명
(텀블링 윈도우를 선택한 경우)	윈도우 폭	집약 대상의 시간 폭을 입력(초 단위)

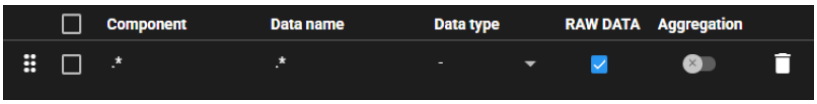
5.「 설정 방법 」을, 다음 중에서 선택합니다.

AI 사업부
-For Reference Only-



항목	설명
데이터 정의를 수신하여 설정한다	입력 포트에서 받아들이고 있는 데이터를 목록으로 표시하고, 개별적으로 집약의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 입력 포트에서 받아들이고 있는 데이터 중에서 매칭된 것에 대해, 집약의 설정을 수행한다

6. 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.

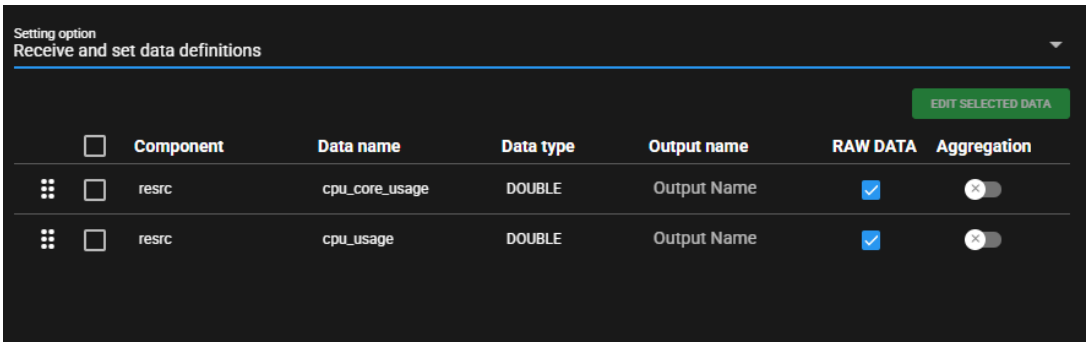


항목	설명
컴포넌트명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 컴포넌트명을 나타내는 정규 표현식 ※설정 방법이「정규 표현식으로 설정한다」인 경우만 입력 가능
데이터명	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식 ※설정 방법이「정규 표현식으로 설정한다」인 경우만 입력 가능
데이터형	입력 포트에서 받아들이고 있는 데이터 중, 설정 대상으로 하는 것의 데이터형 ※설정 방법이「정규 표현식으로 설정한다」인 경우만 선택 가능
출력명	기존의 데이터명을 덮어쓰는 새로운 이름 ※설정 방법이「데이터 정의를 수신하여 설정한다」인 경우만 선택 가능
RAW DATA	가공하지 않은 데이터를 취득한다

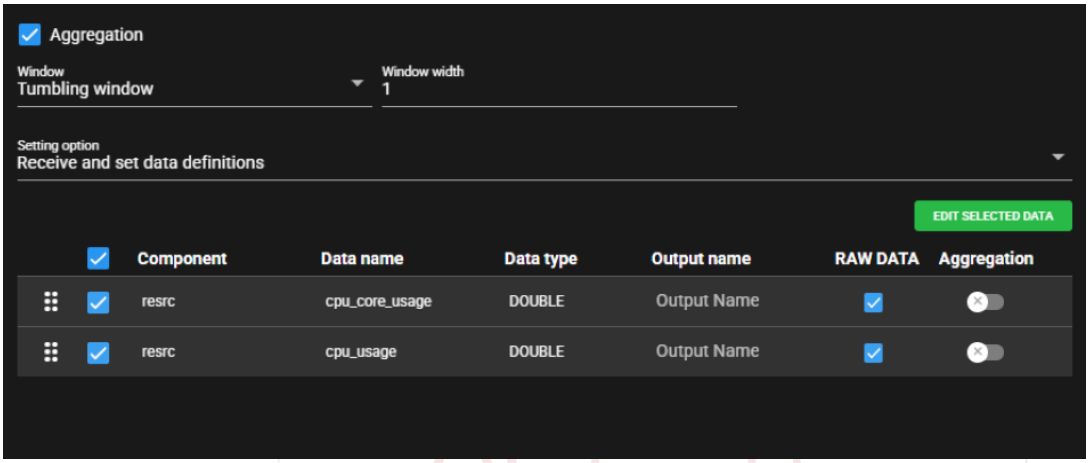
AI 번역본
-For Reference Only-

> **정규 표현식의 입력 예**
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「data1」에만 매칭 : data1
> - 이름에 「abc」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이 「abc」로 시작하는 데이터에 매칭 : abc.*
> - 이름이 「abc」로 끝나는 데이터에 매칭 : .*abc

7. 다음 중 어느 하나의 방법을 수행하여, 집약의 설정 다이얼로그를 엽니다.
   - 집약의 설정을 수행할 데이터의 「집약」의 아이콘을 누릅니다.

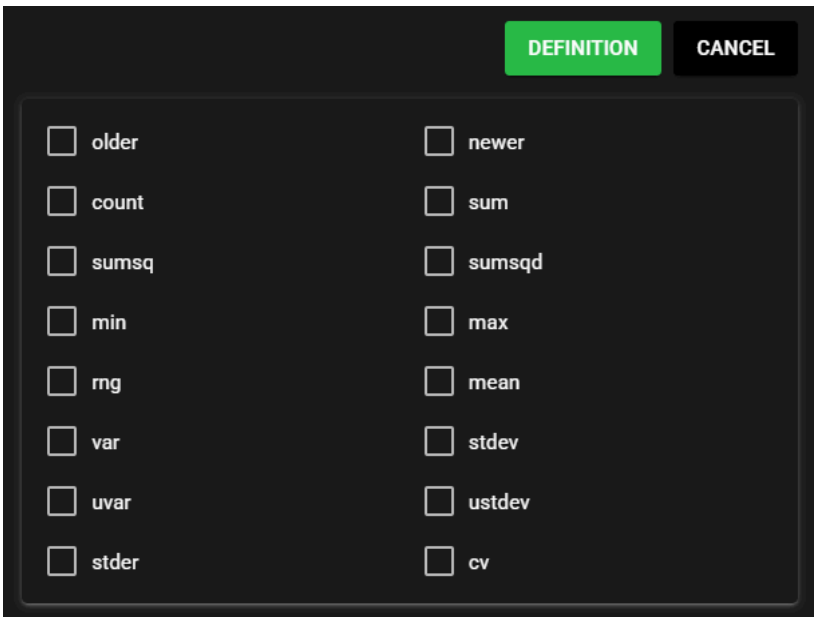


   - 집약의 설정을 수행할 데이터에 체크를 붙이고, 「선택 중인 데이터를 편집」을 누릅니다. 체크를 붙인 데이터에, 동일한 설정 내용이 적용됩니다.



8. 취득할 집약값에 체크를 붙입니다.

-For Reference Only-

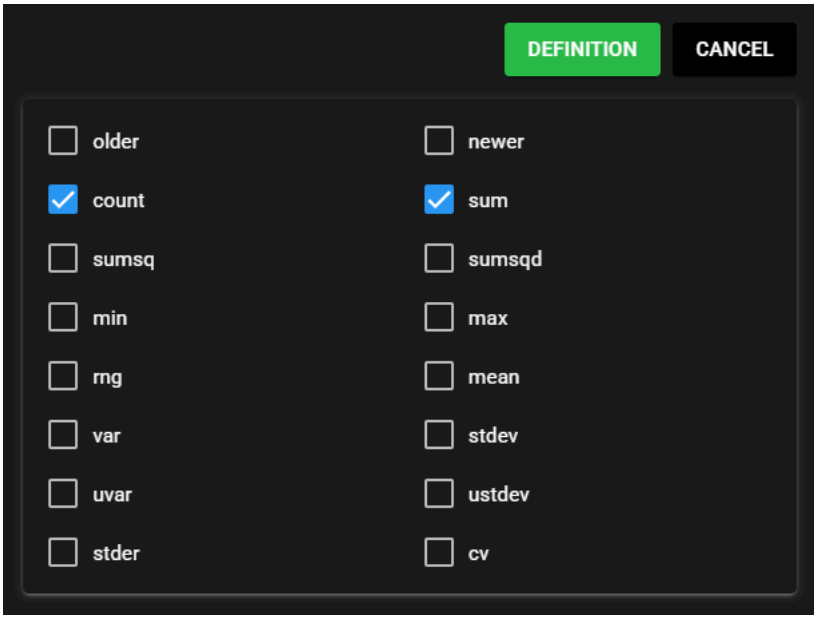


항목	설명
가장 오래된 값	대상 데이터 내의 가장 오래된 값
최신값	대상 데이터 내의 가장 새로운 값
건수	대상의 데이터 건수
합계	대상의 데이터의 합계값
2제곱합	대상의 각 데이터의 제곱을 합계한 값
편차 제곱합	대상의 각 데이터가 평균에서 얼마나 떨어져 있는지의 편차를 제곱하여 합계한 값
최소값	대상의 데이터 내에서 가장 작은 값
최대값	대상의 데이터 내에서 가장 큰 값
범위	대상의 데이터의 최대값과 최소값의 차
산술평균	대상의 데이터의 평균값
분산	대상의 각 데이터가 평균에서 얼마나 떨어져 있는지의 평균의 제곱
표준편차	분산의 제곱근
불편분산	분산을 건수-1로 나눈 값

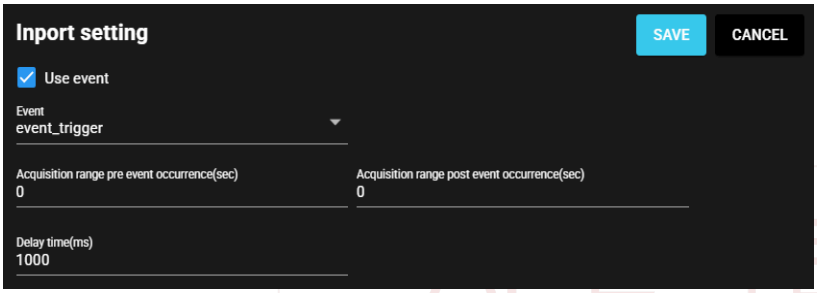
AI 번역본
-For Reference Only-

항목	설명
표본 불편 표준편차	표본의 표준편차
표준오차	표본 평균의 표준편차
변동계수	표준편차를 평균으로 나눈 값

9. 「 확정 」을 누릅니다.



10. 「 저장 」을 누릅니다.

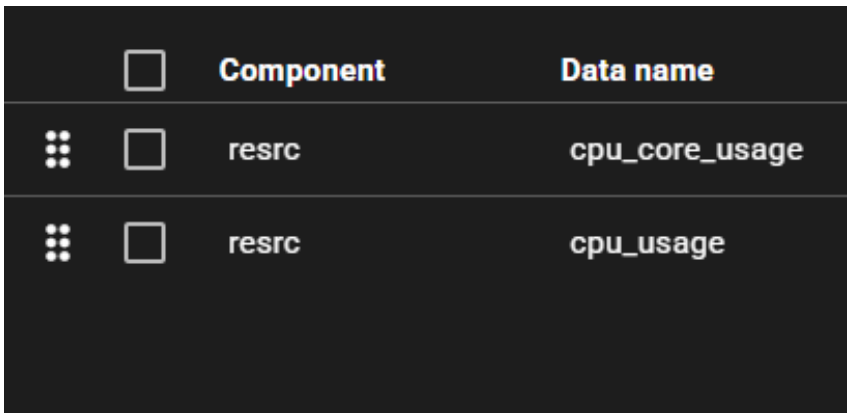


11. 집약의 설정이 적용됩니다.

-For Reference Only-

5.12.4.4 데이터의 정렬

데이터의 왼쪽 끝의 아이콘을 클릭하여 이동시킴으로써, 데이터의 정렬 순서를 변경할 수 있습니다.



5.12.5 시스템 포트

시스템 포트는, 플로우 링크의 접속원이며, 컴포넌트의 시스템 정보(에러 정보나 가동 정보)를 출력하는 포트입니다. 시스템 포트에서는, 시스템 정보에 대해, 영속화의 유무 등의 설정을 수행할 수 있습니다.

> ⚠ 시스템 포트의 설정 변경에 대해
>
> 이 시스템 포트에서 설정할 수 있는 설정 내용에 대해서는, 에러의 기록 기간 등에 영향을 주기 때문에, 필요가 없는 한은 변경하지 마십시오.

5.12.5.1 설정 가능한 시스템 정보

시스템 포트에서 출력되는, 컴포넌트의 시스템 정보는 다음과 같습니다.

항목	데이터형	설명
errors	COMPONENT_STATUS	에러 정보
errors_run_status	UINT8	정수값에 의한 가동 상태 0: 정지 1: 실행 중（에러 없음） 2: 실행 중（에러 있음） 3: 에러 정지
total_input_data_count	UINT64	입력 데이터의 건수
total_input_data_bytes	UINT64	입력 데이터의 바이트 수
total_output_data_count	UINT64	출력 데이터의 건수
total_output_data_bytes	UINT64	출력 데이터의 바이트 수

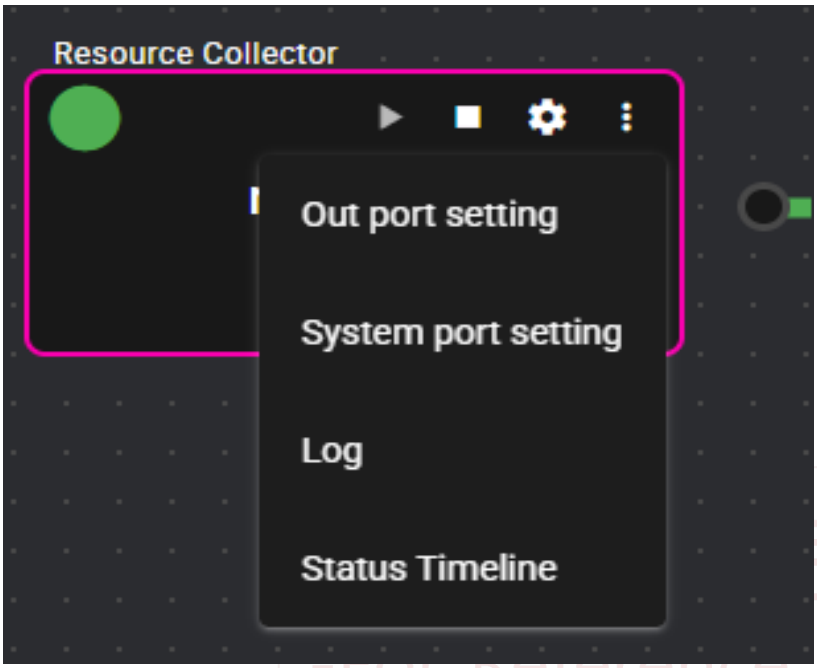
AI 번역로
-For Reference Only-

항목	데이터형	설명
total_received_count	UINT64	컴포넌트가 수신한 데이터의 건수
total_received_bytes	UINT64	컴포넌트가 수신한 데이터의 바이트 수
total_emission_count	UINT64	컴포넌트가 송신한 데이터의 건수
total_emission_bytes	UINT64	컴포넌트가 송신한 데이터의 바이트 수

5.12.5.2 영속화의 설정

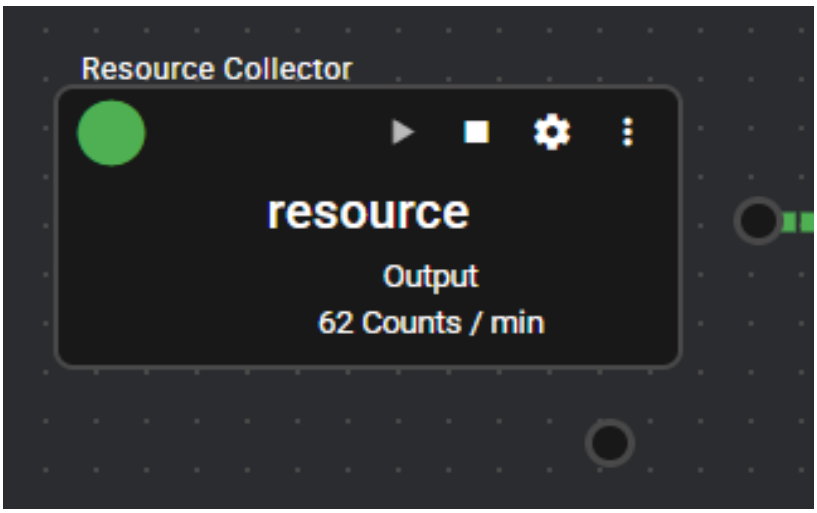
영속화를 유효로 한 데이터는, 스토리지 내에 저장되고, 저장되어 있는 동안, SQL에서의 참조, Grafana 연계에서의 참조가 가능해집니다. 다음 순서로, 영속화의 설정을 수행할 수 있습니다.

1. 다음 중 어느 하나의 방법으로 시스템 포트의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하고,「 시스템 포트 설정 」을 선택

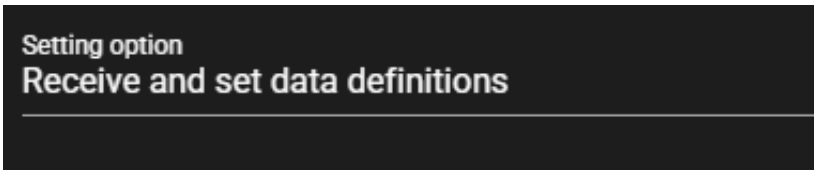


   - 컴포넌트의 시스템 포트를 오른쪽 클릭

-For Reference Only-

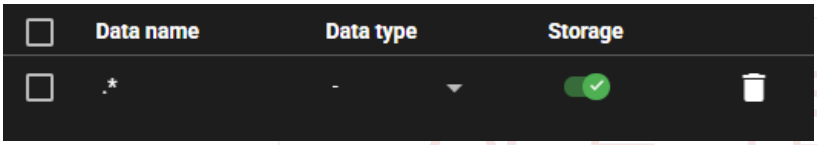


2. 「 설정 방법 」을, 다음 중에서 선택합니다.



항목	설명
데이터 정의를 수신하여 설정 한다	시스템 정보를 목록으로 표시하고, 개별적으로 영속화의 설정을 수행한다
정규 표현식으로 설정한다	정규 표현식을 입력하고, 시스템 정보 중에서 매칭된 것에 대해, 영속화의 설정을 수행한다

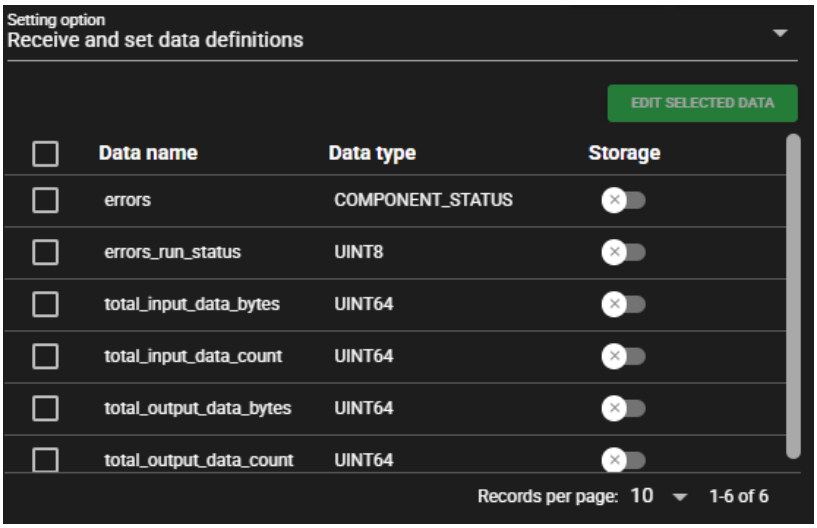
3. 「 설정 방법 」이 「 정규 표현식으로 설정한다 」인 경우만, 설정 대상으로 하는 데이터에 대해, 다음 항목을 입력합니다.



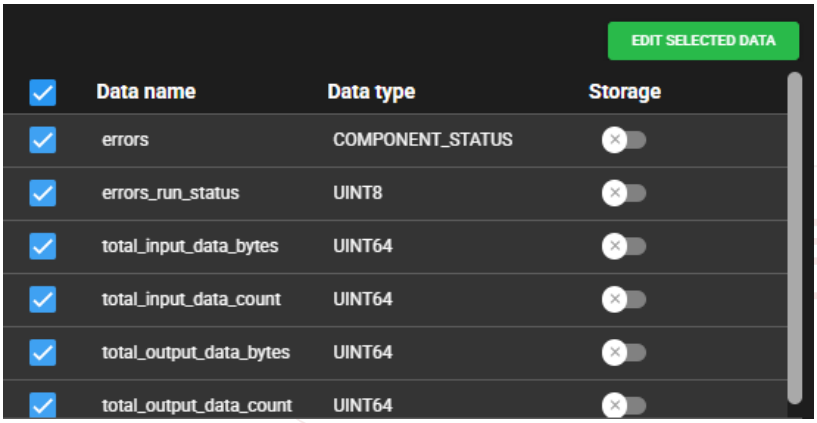
항목	설명
데이터명	시스템 정보 중, 설정 대상으로 하는 것의 데이터명을 나타내는 정규 표현식
데이터형	시스템 정보 중, 설정 대상으로 하는 것의 데이터형

> **정규 표현식의 입력 예**
>
> 정규 표현식의 입력 예는 다음과 같습니다.
>
> - 「data1」에만 매칭 : data1
> - 이름에 「abc」를 포함하는 데이터에 매칭 : .*abc.*
> - 이름이 「abc」로 시작하는 데이터에 매칭 : abc.*
> - 이름이 「abc」로 끝나는 데이터에 매칭 : .*abc

4. 다음 중 어느 하나의 방법을 수행하여, 영속화의 설정 다이얼로그를 엽니다.
   - 영속화의 설정을 수행할 데이터의 「영속화」의 아이콘을 누릅니다.



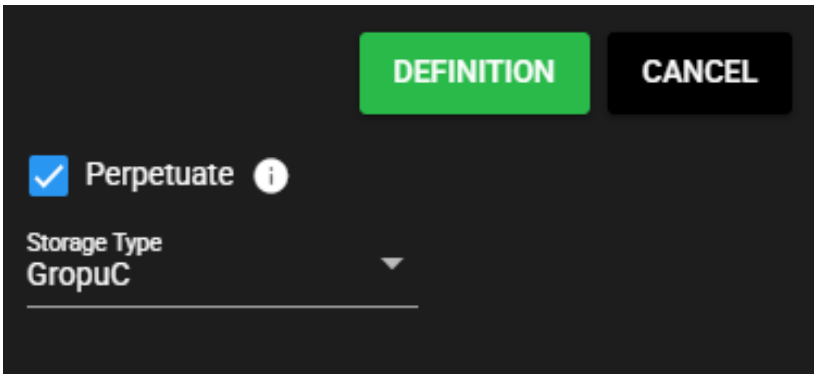
   - 영속화의 설정을 수행할 데이터에 체크를 붙이고, 「선택 중인 데이터를 편집」을 누릅니다. 체크를 붙인 데이터에, 동일한 설정 내용이 적용됩니다.



5. 「영속화한다」를 ON으로 합니다.

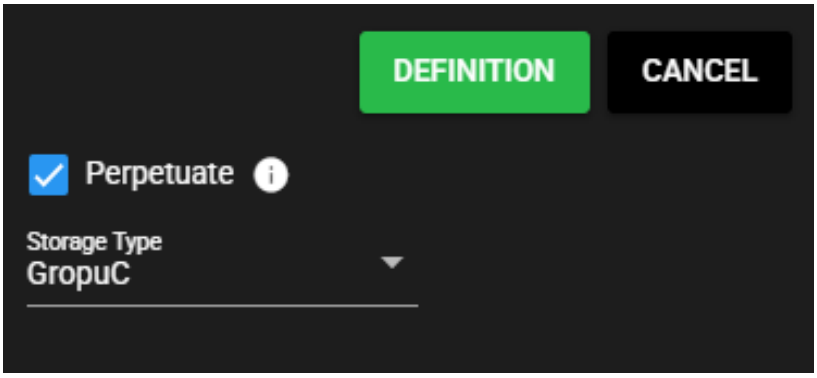


6. 다음 항목을 입력합니다.



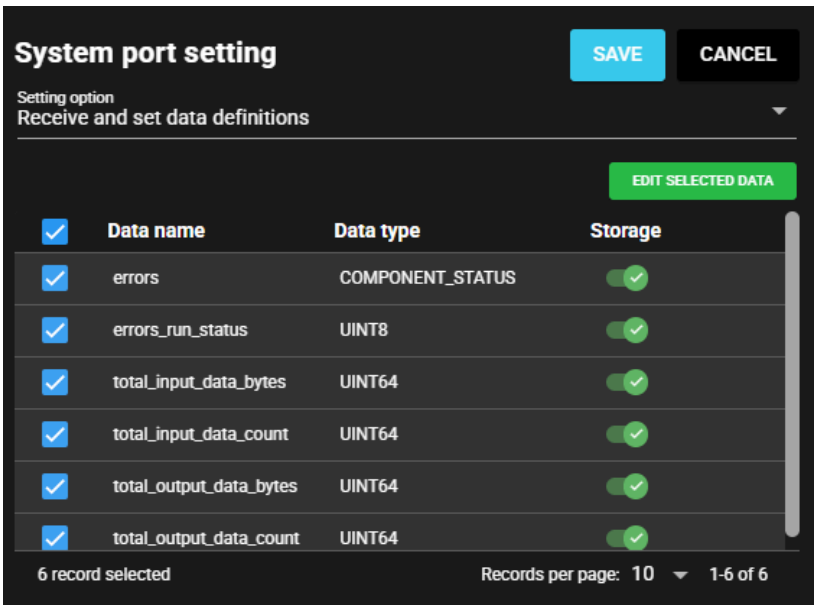
항목	설명
스토리지 타입	시계열 데이터베이스에 저장하는 기간. 서버 설정의 「 그룹 단위로 저장 기간을 지정 」에서, GroupA・GroupB・GroupC의 저장 기간을 확인・변경할 수 있습니다. 「 그룹 단위로 저장 기간을 지정 」이 ON인 경우만 변경할 수 있습니다.

7. 「 확정 」을 누릅니다.



8. 「 저장 」을 누릅니다.

AI 번역본
-For Reference Only-



9. 영속화의 설정이 적용됩니다.

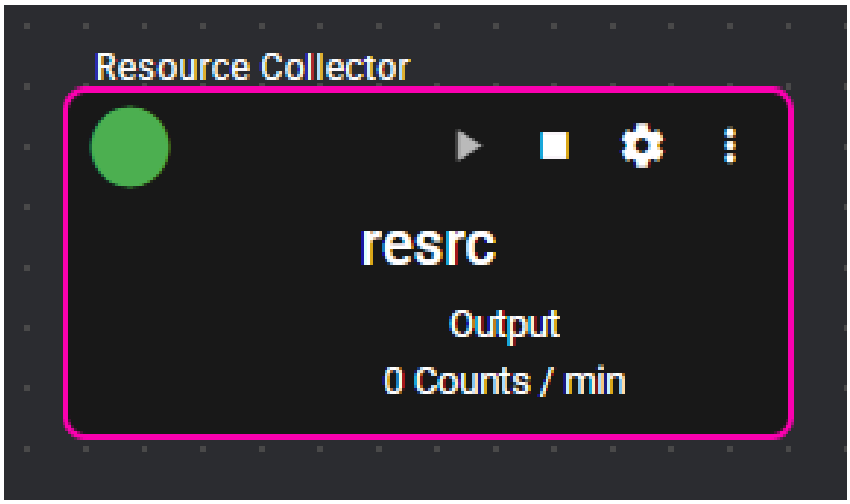
AI 번역본
-For Reference Only-

5.13 스테이터스 타임라인

여기에서는, 가동 스테이터스의 목록, 스테이터스 타임라인에 대해 설명합니다.

5.13.1 스테이터스 목록

컴포넌트의 스테이터스 램프에서, 가동 스테이터스를 확인할 수 있습니다.



가동 스테이터스의 종류는, 다음과 같습니다.

램프	스테이터스 명	설명
	정상	컴포넌트가 정상으로 가동하고 있다
	경고	문제가 발생했지만, 컴포넌트의 가동은 멈추지 않았다
	에러	문제가 발생하여, 컴포넌트의 가동이 멈춰 있다
	정지	컴포넌트가 정지하고 있다
	정지 중	컴포넌트가 가동하고 있던 상태에서, 정지 상태로 이행하고 있다

AI 번역본
-For Reference Only-

램프	스테이터스 명	설명
▬▬	로드 실패	커스텀 컴포넌트가 등록되어 있지 않거나, 혹은 로드 시에 에러가 되어 있어, 컴포넌트를 이용할 수 없다

5.13.2 스테이터스 타임라인

컴포넌트의 가동 스테이터스의 천이 이력을 확인할 수 있습니다. 스테이터스가 경고・에러인 경우는, 에러 코드를 확인할 수 있습니다.

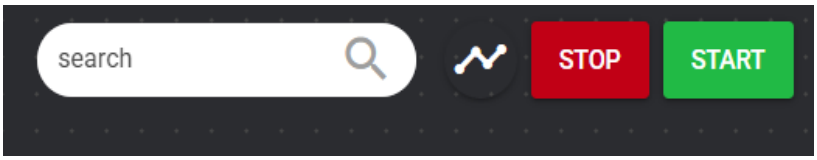
5.13.2.1 열람

여기에서는, 스테이터스 타임라인의 열람 방법을 설명합니다. 스테이터스 타임라인에 전 컴포넌트의 정보를 표시시키는 방법과, 개별 컴포넌트의 정보를 표시시키는 방법이 있습니다.

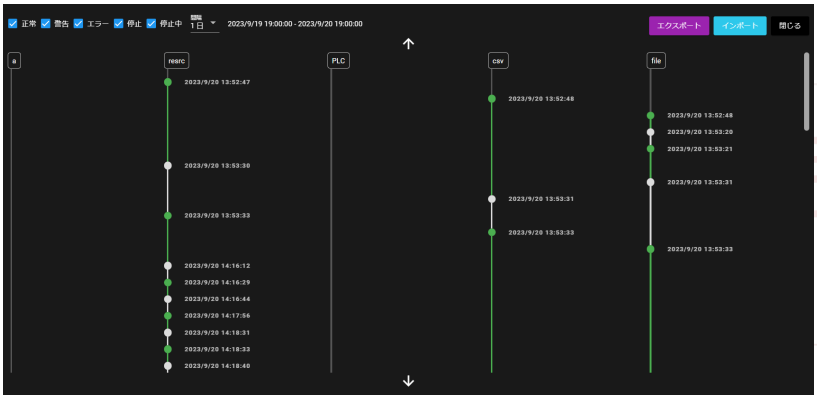
5.13.2.1.1 전 컴포넌트

다음 순서로, 전 컴포넌트의 정보를 표시한 스테이터스 타임라인을 표시할 수 있습니다.

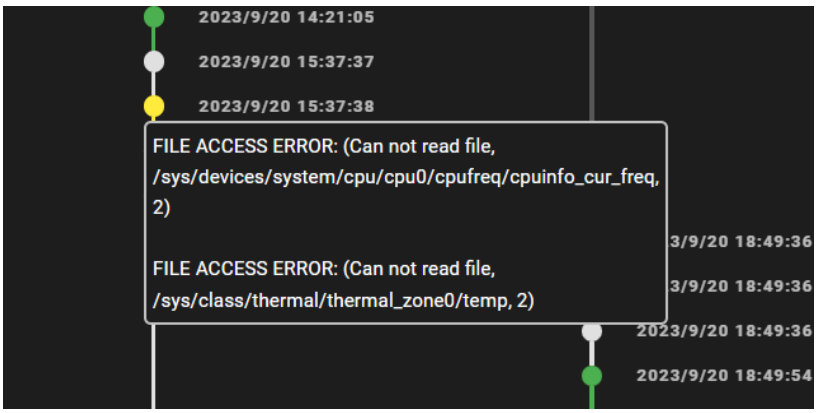
1. 스테이터스 타임라인의 아이콘을 누릅니다.



2. 전 컴포넌트의 스테이터스의 천이 이력이 표시됩니다.



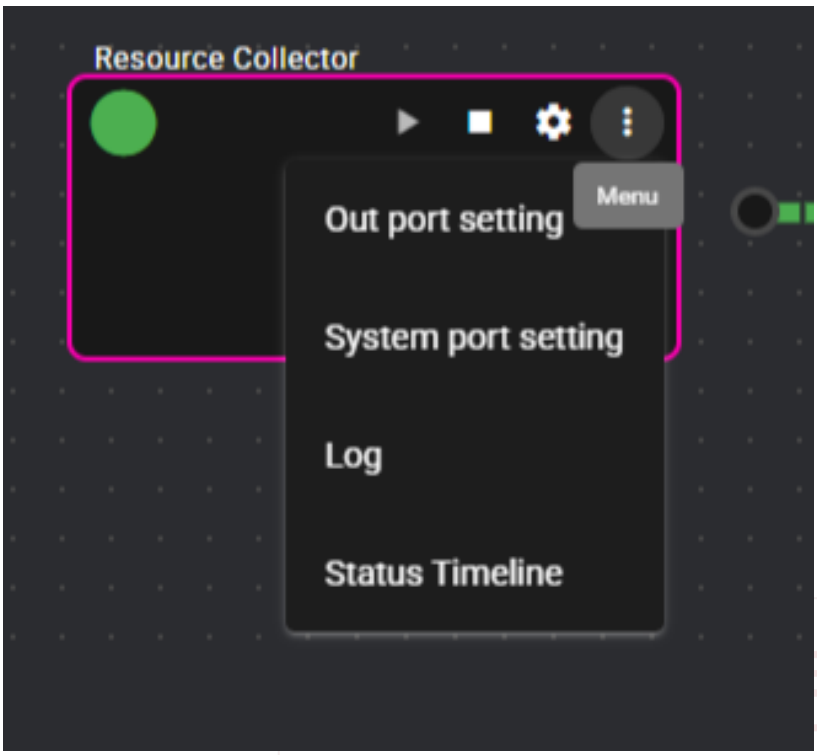
3. 경고・에러의 스테이터스에 마우스를 맞추면, 에러 코드가 표시됩니다.



5.13.2.1.2 개별 컴포넌트

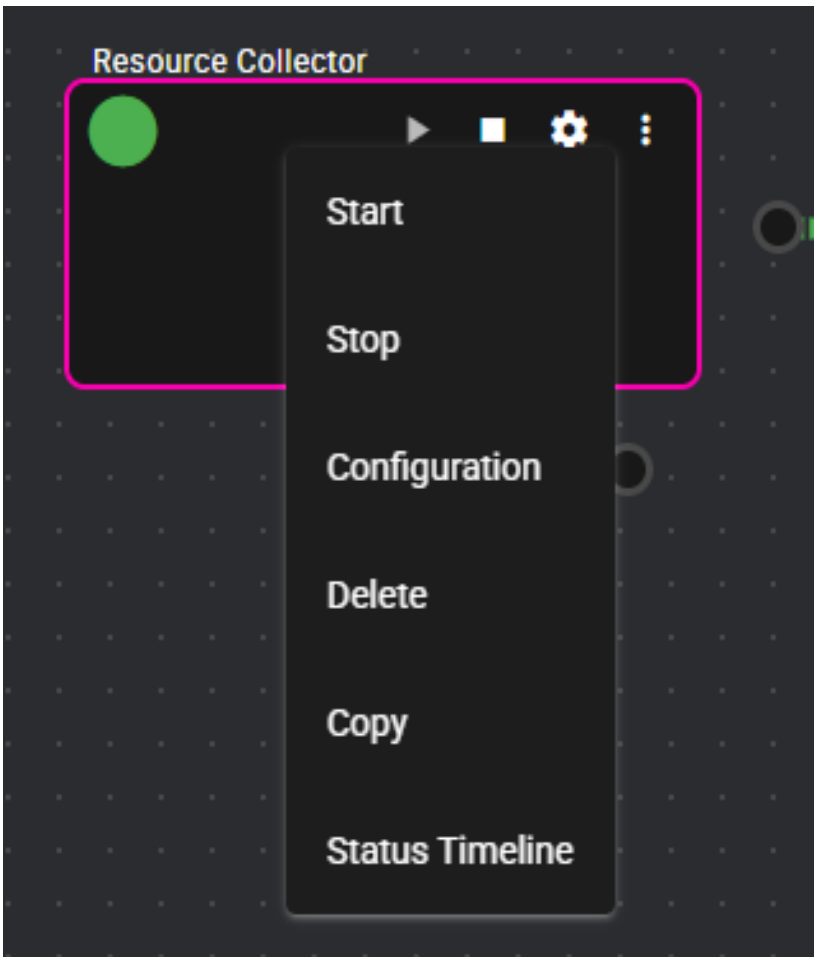
다음 순서로, 개별 컴포넌트의 정보를 표시한 스테이터스 타임라인을 표시할 수 있습니다.

1. 이하의 어느 하나로 스테이터스 타임라인의 다이얼로그를 엽니다.
   - 메뉴 아이콘을 왼쪽 클릭하여 「 스테이터스 타임라인 」을 선택

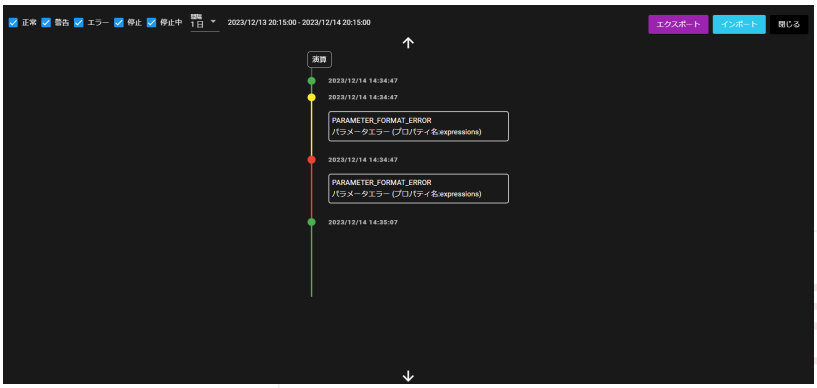


   - 컴포넌트를 오른쪽 클릭하여 「 스테이터스 타임라인 」을 선택

For Reference Only-



2. 개별 컴포넌트의 스테이터스의 천이 이력이 표시됩니다. 경고・에러의 스테이터스에는, 에러 코드가 표시됩니다.



-For Reference Only-

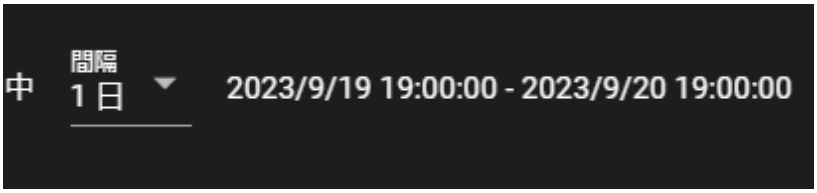
5.13.2.2 좁히기

여기에서는, 스테이터스 타임라인에 표시시킬 내용을 좁히는 방법에 대해 설명합니다.

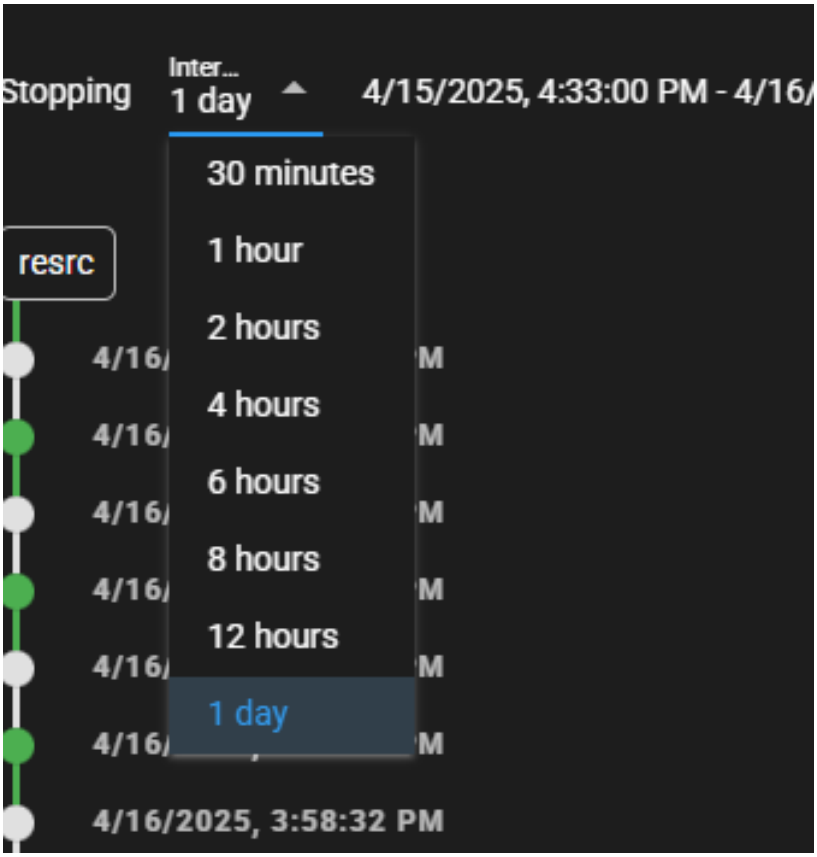
표시 간격

다음 순서로, 스테이터스 타임라인의 표시 간격을 변경할 수 있습니다.

1. 「 간격 」을 누릅니다.



2. 시간 단위를 선택합니다.



3. 선택한 시간 단위의 범위의 분량만큼, 스테이터스의 천이 이력이 표시됩니다.



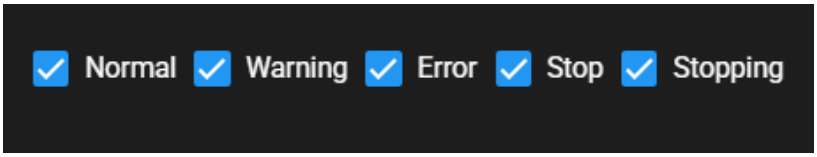
4. 위 화살표 아이콘을 누르면 과거, 아래 화살표 아이콘을 누르면 미래의 이력을 표시합니다.



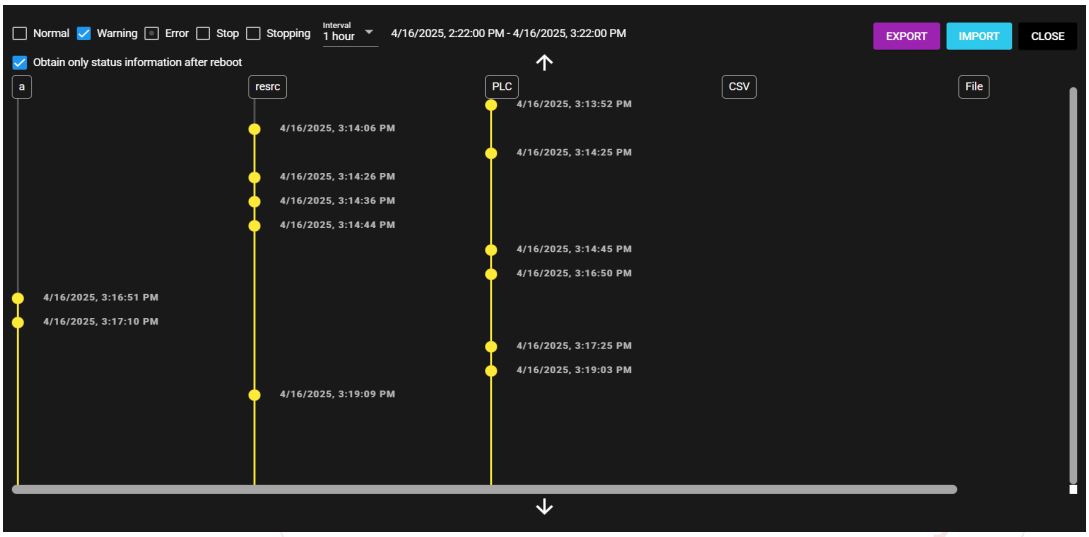
스테이터스

다음 순서로, 지정한 스테이터스만 스테이터스 타임라인에 표시시킬 수 있습니다.

1. 스테이터스에 체크를 붙이거나, 또는 해제합니다.



2. 스테이터스의 천이 이력 중, 체크를 붙인 것만이 표시됩니다.

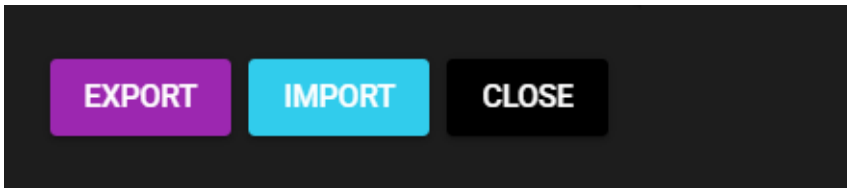


5.13.2.3 익스포트・임포트

스테이터스 타임라인의 표시 내용을, 파일로서 출력할 수 있습니다. 출력한 파일을 다른 환경의 SpeeDBee Synapse에 취입하여 표시시킴으로써, 가동 스테이터스의 정보를 공유할 수 있게 됩니다.

익스포트

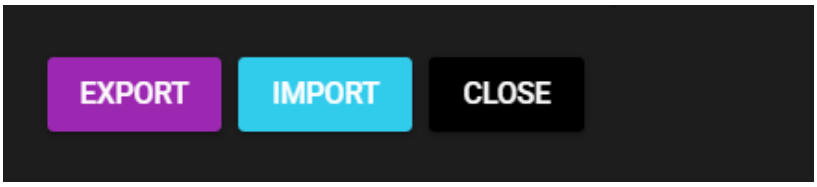
「 익스포트 」를 누르면, 스테이터스 타임라인의 표시 내용을, 파일로서 출력할 수 있습니다.



임포트

다음 순서로, 익스포트한 파일에서 정보를 취득하여, 스테이터스 타임라인을 표시할 수 있습니다.

1. 「 임포트 」를 누릅니다.



2. 익스포트한 파일을 선택합니다.
3. 파일에서 정보를 취득하여, 스테이터스 타임라인이 표시됩니다.

AI 번역본
-For Reference Only-

5.14 각종 로그

여기에서는, SpeeDBee Synapse의 각종 로그에 대해 설명합니다.

5.14.1 로그의 종별

SpeeDBee Synapse가 취급하는 로그의 종별은 다음과 같습니다.

종별	설명	화면에서의 열람	다운로드
SpeeDBee Synapse의 조작 로그	사용자 등에 의한 SpeeDBee Synapse의 조작 이력이 기록됩니다. ※상세는 「 조작 로그 」를 참조하십시오.	○	○
WEBUI의 운용 로그	WEBUI의 운용 정보가 기록됩니다.	×	○
코어, 컴포넌트, 데이터베이스의 운용 로그	코어, 컴포넌트, 데이터베이스의 운용 정보가 기록됩니다.	×	○
컴포넌트 개별의 운용 로그	컴포넌트의 운용 정보가 기록됩니다.	○	○

5.14.2 조작 로그

사용자가 SpeeDBee Synapse상에서 수행한 조작은 모두, 조작 로그에 기록됩니다. 조작 로그에는, SpeeDBee Synapse나 외부 API에 의해 수행된 조작도 기록됩니다.

AI 번역본
-For Reference Only-

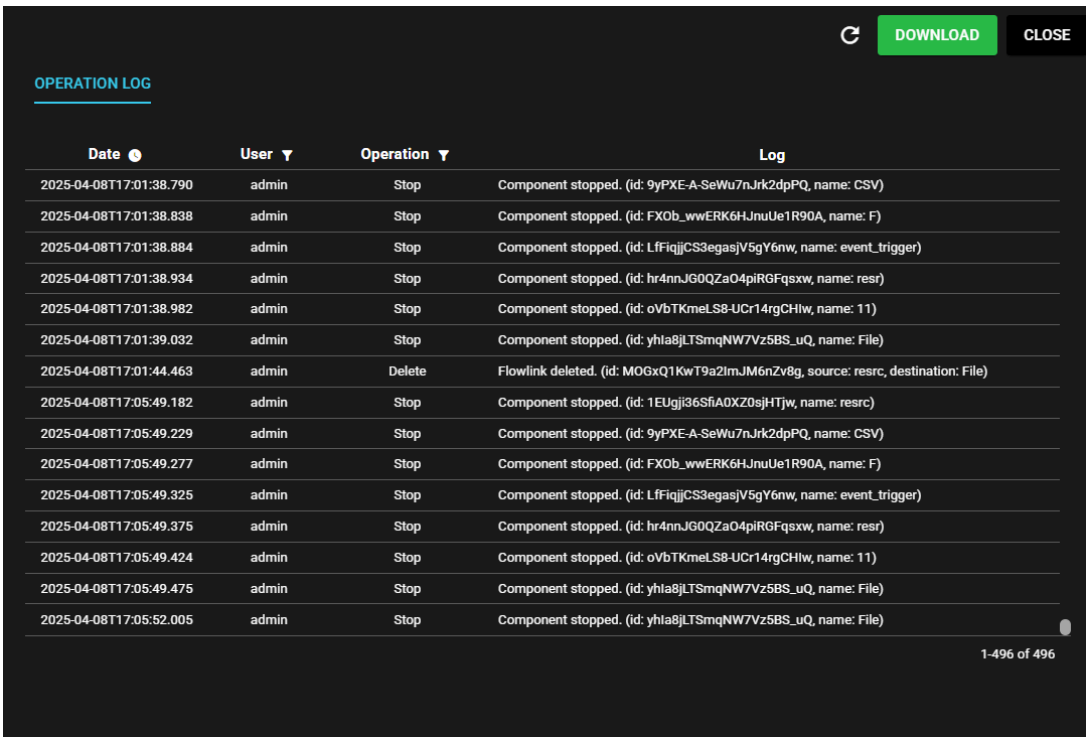
5.14.2.1 열람

다음 순서로, 조작 로그를 열람할 수 있습니다.

1. 헤더의 로그 아이콘을 누릅니다.



2. 조작 로그가 표시됩니다.



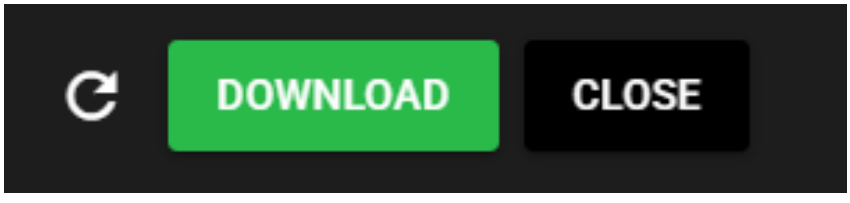
조작 로그에 표시되는 내용은, 다음과 같습니다.

항목	설명
시각	조작을 수행한 시각
사용자	조작을 수행한 사용자 ※1 "시스템"은 SpeeDBee Synapse에 의해 수행된 조작을 나타냅니다 ※2 "외부 API"는 외부 API에 의해 수행된 조작을 나타냅니다.
오퍼레이션	조작의 종류
로그	조작의 내용

AI 번역본
-For Reference Only-

5.14.2.2 리프레시

리프레시 버튼을 누르면, 최신 로그를 취득할 수 있습니다.



5.14.2.3 좁히기

시각, 사용자, 조작의 종류에 의해, 열람하는 조작 로그를 좁힐 수 있습니다.

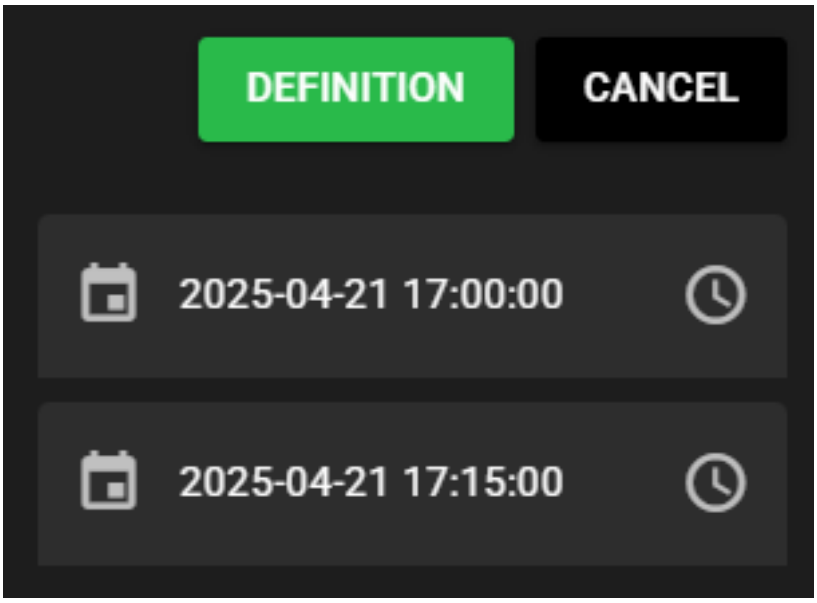
시각

다음 순서로, 지정한 시간 범위의 조작 로그만을 열람할 수 있습니다.

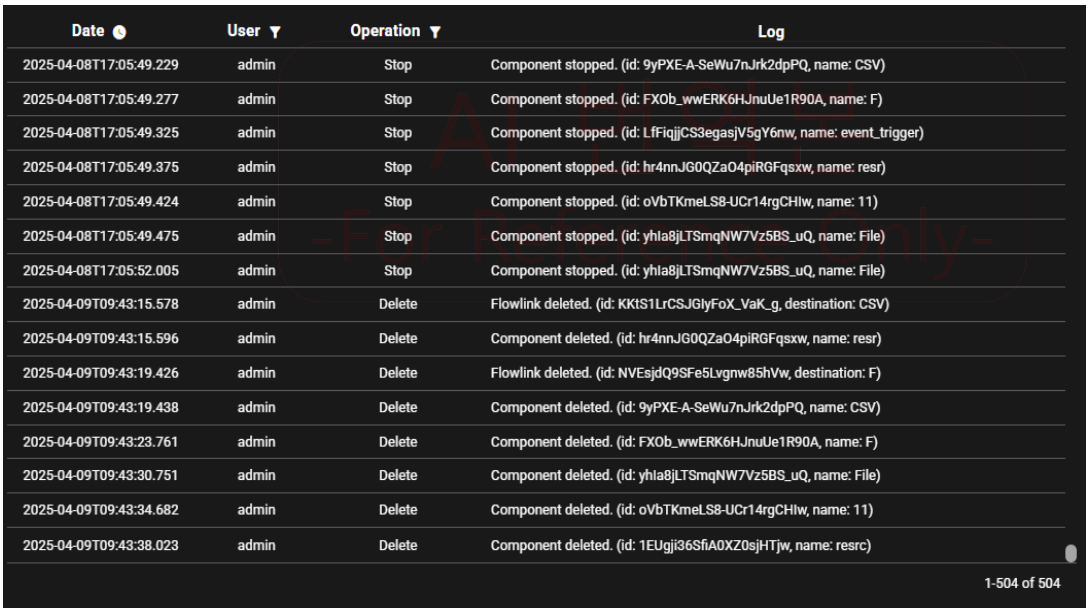
1. 「 시각 」의 옆의 아이콘을 누릅니다.



2. 시간 범위를 입력하고, 「 확정 」을 누릅니다.



3. 입력한 시간 범위의 로그가 표시됩니다.



Date	User	Operation	Log
2025-04-08T17:05:49.229	admin	Stop	Component stopped. (id: 9yPXE-A-SeWu7nJrk2dpPQ, name: CSV)
2025-04-08T17:05:49.277	admin	Stop	Component stopped. (id: FXOb_wwERK6HJnuUe1R90A, name: F)
2025-04-08T17:05:49.325	admin	Stop	Component stopped. (id: LfFiqjjCS3egasjV5gY6nw, name: event_trigger)
2025-04-08T17:05:49.375	admin	Stop	Component stopped. (id: hr4nnJG0QZaO4piRGFqsxw, name: resr)
2025-04-08T17:05:49.424	admin	Stop	Component stopped. (id: oVbTKmeLS8-UCr14rgCHIw, name: 11)
2025-04-08T17:05:49.475	admin	Stop	Component stopped. (id: yhIa8jLTSmqNW7Vz5BS_uQ, name: File)
2025-04-08T17:05:52.005	admin	Stop	Component stopped. (id: yhIa8jLTSmqNW7Vz5BS_uQ, name: File)
2025-04-09T09:43:15.578	admin	Delete	Flowlink deleted. (id: KKtS1LrCSJGlyFoX_VaK_g, destination: CSV)
2025-04-09T09:43:15.596	admin	Delete	Component deleted. (id: hr4nnJG0QZaO4piRGFqsxw, name: resr)
2025-04-09T09:43:19.426	admin	Delete	Flowlink deleted. (id: NVEsjdQ9SFe5Lvgnw85hVw, destination: F)
2025-04-09T09:43:19.438	admin	Delete	Component deleted. (id: 9yPXE-A-SeWu7nJrk2dpPQ, name: CSV)
2025-04-09T09:43:23.761	admin	Delete	Component deleted. (id: FXOb_wwERK6HJnuUe1R90A, name: F)
2025-04-09T09:43:30.751	admin	Delete	Component deleted. (id: yhIa8jLTSmqNW7Vz5BS_uQ, name: File)
2025-04-09T09:43:34.682	admin	Delete	Component deleted. (id: oVbTKmeLS8-UCr14rgCHIw, name: 11)
2025-04-09T09:43:38.023	admin	Delete	Component deleted. (id: 1EUgji36SfiA0XZ0sjHTjw, name: resrc)

1-504 of 504

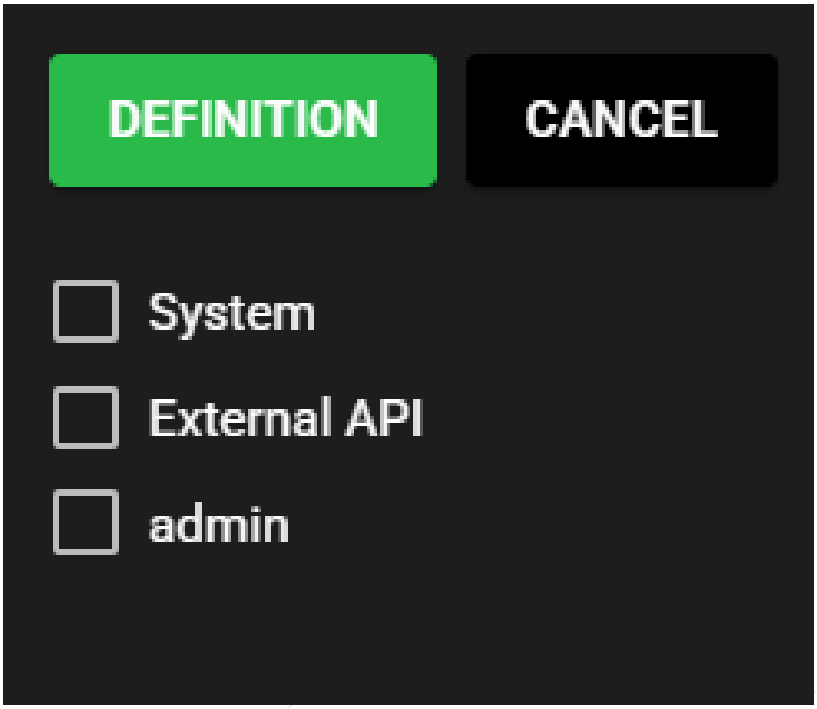
사용자

다음 순서로, 지정한 사용자의 조작 로그만을 열람할 수 있습니다.

1. 「 사용자 」의 옆의 아이콘을 누릅니다.



2. 사용자를 선택하고, 「 확정 」을 누릅니다.



3. 선택한 사용자가 수행한 조작의 로그가 표시됩니다.

AI 번역본
-For Reference Only-

Date	User	Operation	Log
2025-04-21T08:01:09.502	admin	Stop	Component stopped. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:09.511	admin	Stop	Component stopped. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:11.037	admin	Start	Component started. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:12.040	admin	Start	Component started. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:16.554	admin	Stop	Component stopped. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:16.561	admin	Stop	Component stopped. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:17.499	admin	Start	Component started. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:18.513	admin	Start	Component started. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:26.045	admin	Stop	Component stopped. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:26.055	admin	Stop	Component stopped. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:27.694	admin	Start	Component started. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)
2025-04-21T08:01:28.847	admin	Start	Component started. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:33.716	admin	Stop	Component stopped. (id: 6MnFCqwrT1miCnO3-GQDhQ, name: resrc)
2025-04-21T08:01:33.722	admin	Stop	Component stopped. (id: hwUAKaQ8T_GnQ1yJ0yVDxA, name: PLC)

1-14 of 14

오퍼레이션

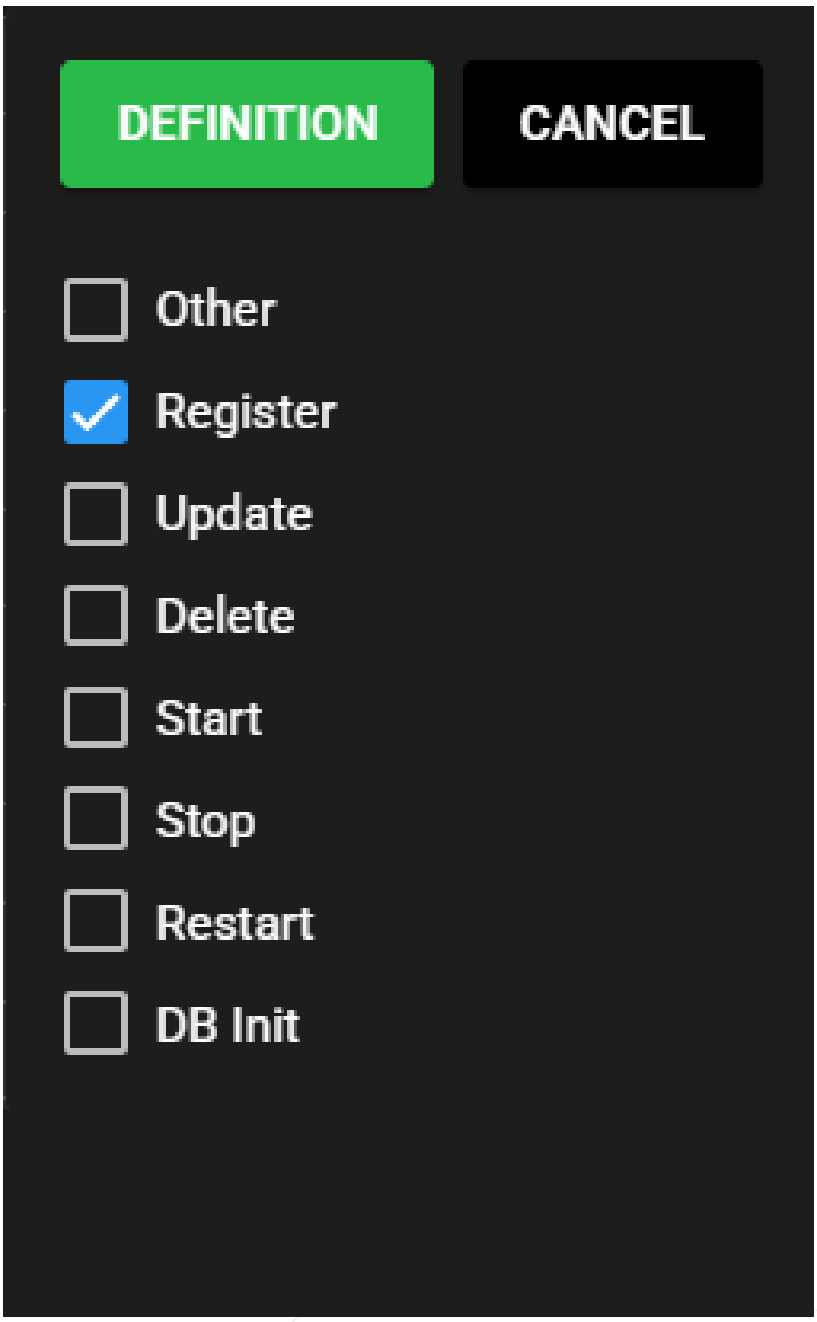
다음 순서로, 지정한 종류의 조작의 로그만을 열람할 수 있습니다.

1. 「 오퍼레이션 」의 옆의 아이콘을 누릅니다.



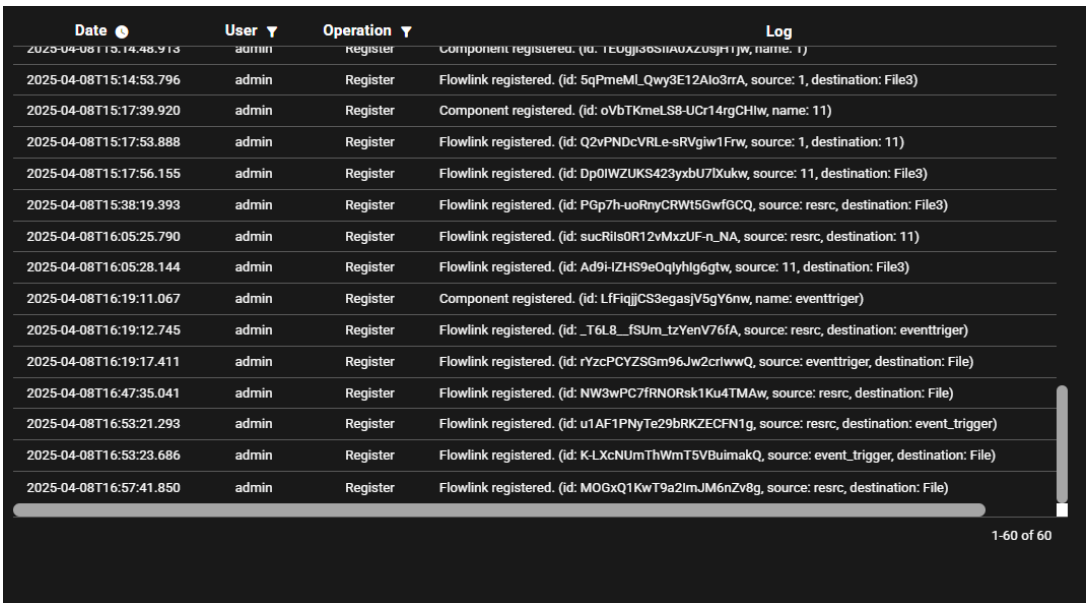
2. 조작 로그의 종류를 선택하고, 「 확정 」을 누릅니다.

AI 번역본
-For Reference Only-



3. 선택한 종류의 조작 로그가 표시됩니다.

AI 번역본
-For Reference Only-



Date	User	Operation	Log
2025-04-08T15:14:48.913	admin	Register	Component registered. (id: TEOgjf36SfIA0XZ0sjH1jw, name: 1)
2025-04-08T15:14:53.796	admin	Register	Flowlink registered. (id: 5qPmeMl_Qwy3E12Alo3rrA, source: 1, destination: File3)
2025-04-08T15:17:39.920	admin	Register	Component registered. (id: oVbTKmeLS8-UCr14rgCHlw, name: 11)
2025-04-08T15:17:53.888	admin	Register	Flowlink registered. (id: Q2vPNDcVRLe-sRVgiw1Frw, source: 1, destination: 11)
2025-04-08T15:17:56.155	admin	Register	Flowlink registered. (id: Dp0IWZUKS423yxbU7lXukw, source: 11, destination: File3)
2025-04-08T15:38:19.393	admin	Register	Flowlink registered. (id: PGp7h-uoRnyCRWt5GwfGCQ, source: resrc, destination: File3)
2025-04-08T16:05:25.790	admin	Register	Flowlink registered. (id: sucRils0R12vMxzUF-n_NA, source: resrc, destination: 11)
2025-04-08T16:05:28.144	admin	Register	Flowlink registered. (id: Ad9i-IZHS9eOqlyhlg6gtw, source: 11, destination: File3)
2025-04-08T16:19:11.067	admin	Register	Component registered. (id: LfFiqjjCS3egasjV5gY6nw, name: eventtriger)
2025-04-08T16:19:12.745	admin	Register	Flowlink registered. (id: _T6L8__fSUm_tzYenV76fA, source: resrc, destination: eventtriger)
2025-04-08T16:19:17.411	admin	Register	Flowlink registered. (id: rYzcPCYZSGm96Jw2crIwwQ, source: eventtriger, destination: File)
2025-04-08T16:47:35.041	admin	Register	Flowlink registered. (id: NW3wPC7fRNORsk1Ku4TMAw, source: resrc, destination: File)
2025-04-08T16:53:21.293	admin	Register	Flowlink registered. (id: u1AF1PNyTe29bRKZECFN1g, source: resrc, destination: event_trigger)
2025-04-08T16:53:23.686	admin	Register	Flowlink registered. (id: K-LXcNUmThWmT5VBuimakQ, source: event_trigger, destination: File)
2025-04-08T16:57:41.850	admin	Register	Flowlink registered. (id: MOGxQ1KwT9a2lmJM6nZv8g, source: resrc, destination: File)

1-60 of 60

5.14.3 컴포넌트 개별 로그

컴포넌트에 의해 출력된 로그가 기록됩니다.

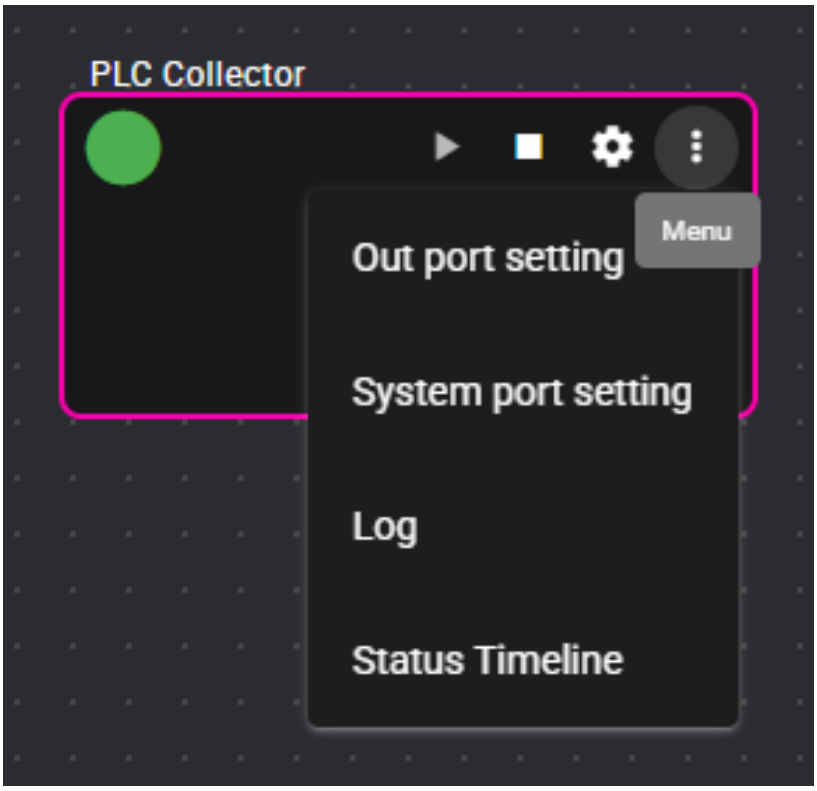
또한 본 기능의 로그 열람 화면에서 컴포넌트 개별로 로그 출력의 레벨을 설정할 수 있습니다. 설정 방법은 「 개별 로그 레벨 설정 」을 참조하십시오.

5.14.3.1 열람

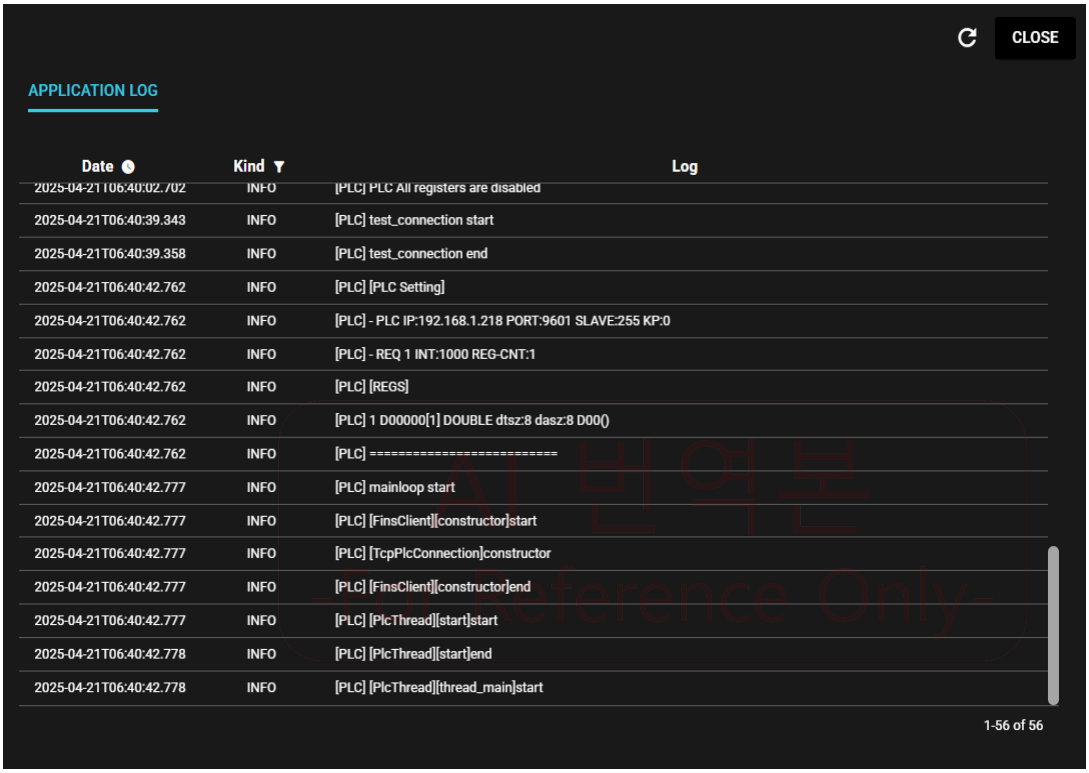
다음 순서로 컴포넌트 개별 로그를 열람할 수 있습니다.

AI 번역본
-For Reference Only-

1. 메뉴 아이콘을 왼쪽 클릭하여「로그」를 선택.



2. 컴포넌트 개별 로그가 표시됩니다.



컴포넌트 개별 로그에 표시되는 내용은, 다음과 같습니다.

항목	설명
시각	컴포넌트에 의해 로그가 출력된 시각
종별	로그 출력의 레벨
로그	로그의 내용

5.14.3.2 리프레시

리프레시 버튼을 누르면, 최신 로그를 취득할 수 있습니다.



5.14.3.3 좁히기

시각, 종별에 의해, 열람하는 컴포넌트 개별 로그를 좁힐 수 있습니다.

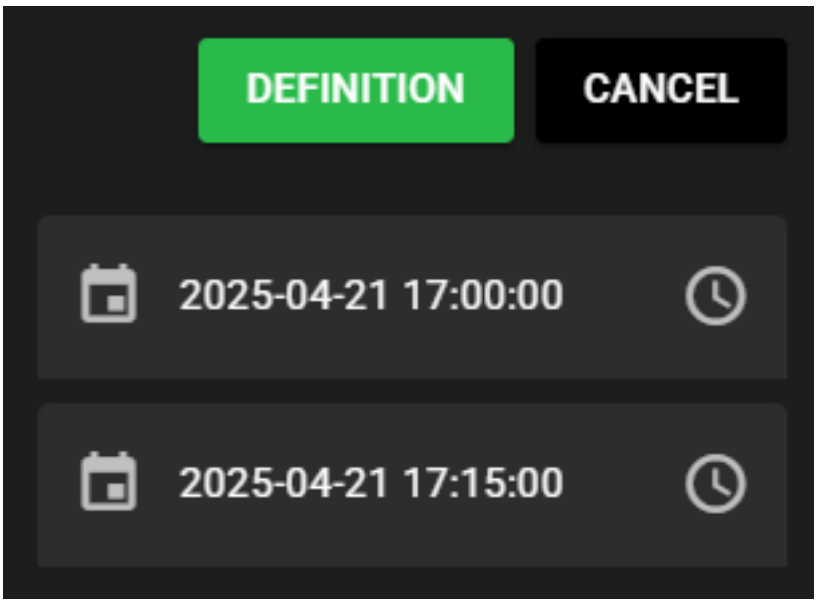
시각

다음 순서로, 지정한 시간 범위의 컴포넌트 개별 로그만을 열람할 수 있습니다.

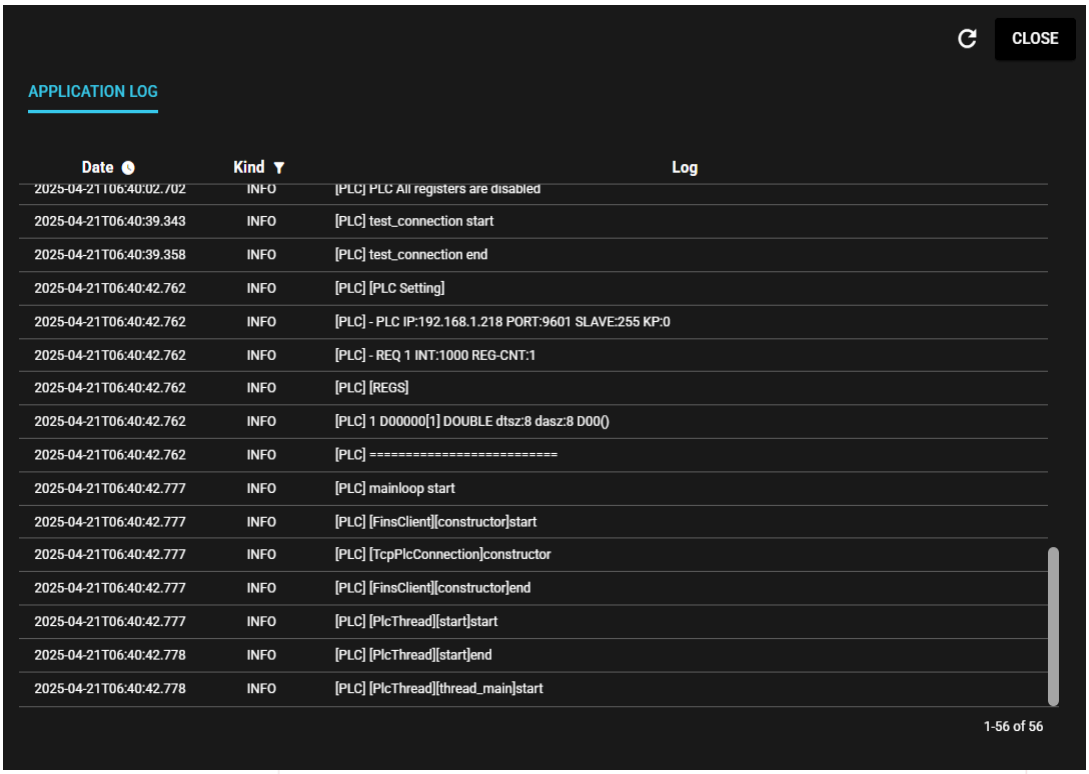
1. 「 시각 」의 옆의 아이콘을 누릅니다.



2. 시간 범위를 입력하고, 「 확정 」을 누릅니다.



3. 입력한 시간 범위의 로그가 표시됩니다.

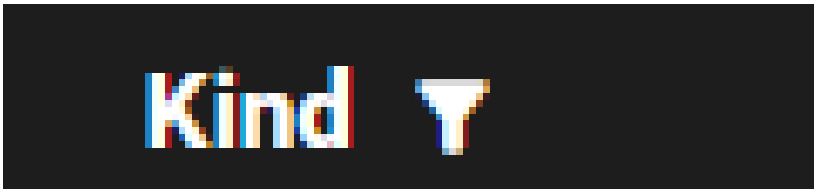


-For Reference Only-

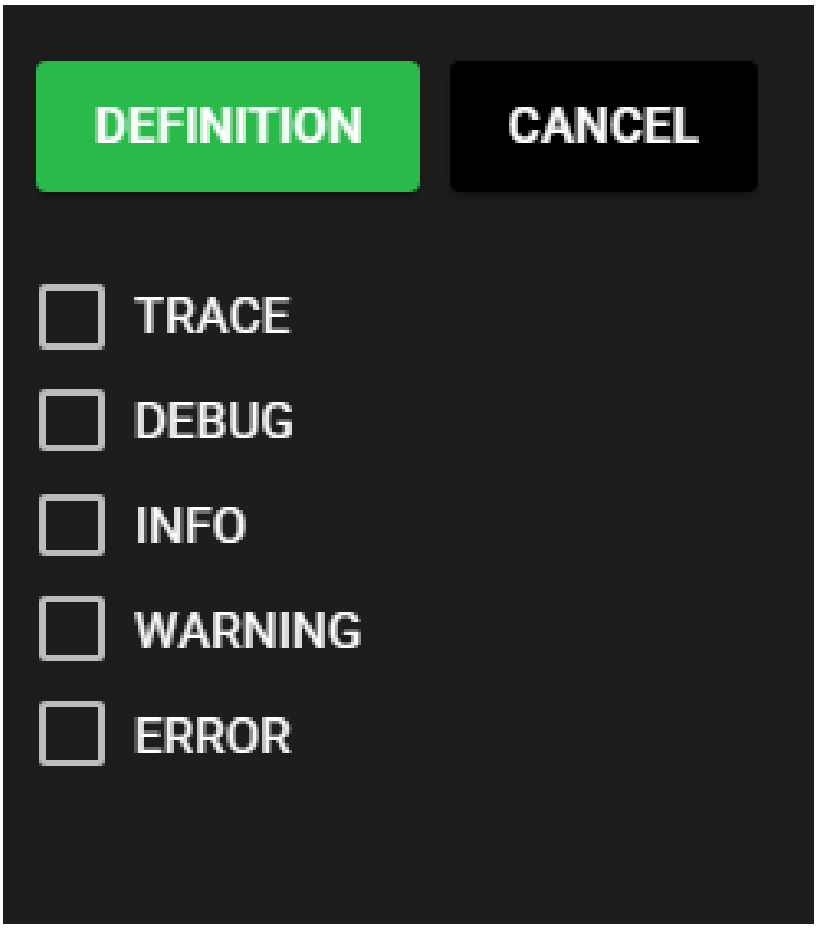
종별

다음 순서로 지정한 종별（로그 레벨）의 컴포넌트 개별 로그만을 열람할 수 있습니다.

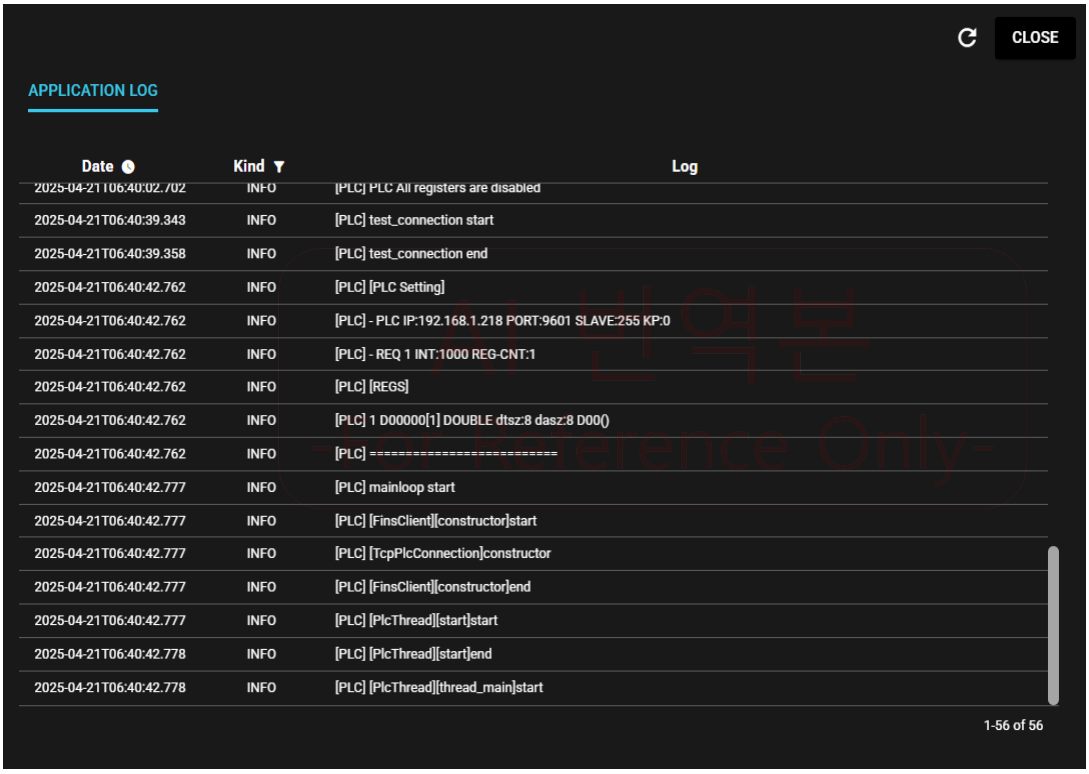
1. 「종별」의 옆의 아이콘을 누릅니다.



2. 종별 (로그 레벨) 을 선택하고, 「 확정 」을 누릅니다.



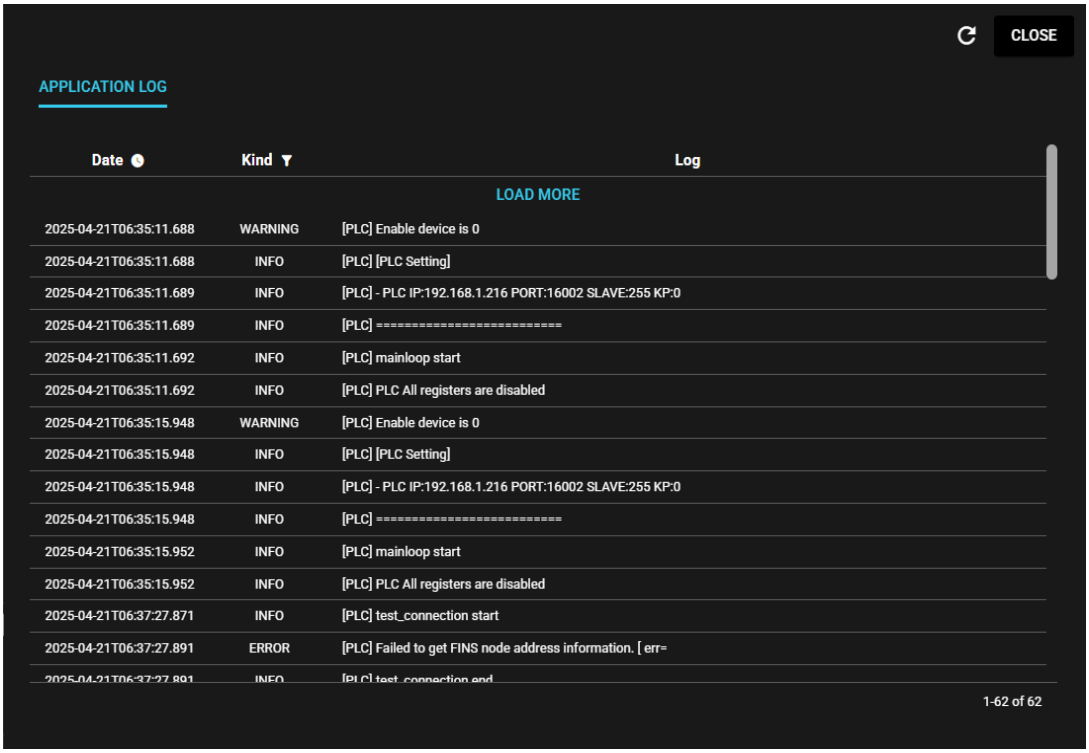
3. 선택한 종별 (로그 레벨) 의 컴포넌트 개별 로그가 표시됩니다.



5.14.3.4 더 불러오기

다음 순서로 현재 표시되어 있는 컴포넌트 개별 로그보다 더 과거의 로그를 추가로 표시할 수 있습니다.

1. 컴포넌트 개별 로그를 맨 위까지 스크롤하면 「 더 불러오기 」가 표시되므로 누릅니다.



※이미 가장 과거의 로그까지 표시하고 있는 경우, 「 더 불러오기 」는 표시되지 않습니다.

5.14.3.5 개별 로그 레벨 설정

컴포넌트 개별로 로그 출력의 레벨을 설정할 수 있습니다.

이 설정을 유효로 한 경우에는 「 전반적인 설정 」의 「 로그 설정 」에서 설정된 로그 레벨에 영향받지 않고, 개별로 설정한 로그 레벨 미만의 로그는 출력되지 않게 됩니다.

이 설정을 무효로 한 경우에는 「 전반적인 설정 」의 「 로그 설정 」에서 설정된 로그 레벨에 따라 로그가 출력됩니다.

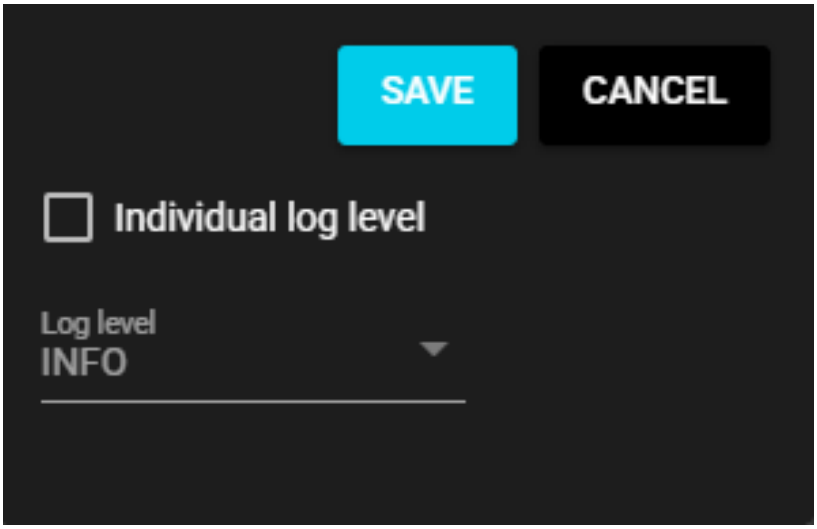
AI 번역본
-For Reference Only-

다음 순서로 현재 표시되어 있는 컴포넌트 개별의 로그 레벨을 설정할 수 있습니다.

1. 설정 아이콘을 누릅니다.



2. 개별 로그 레벨 설정이 표시됩니다.

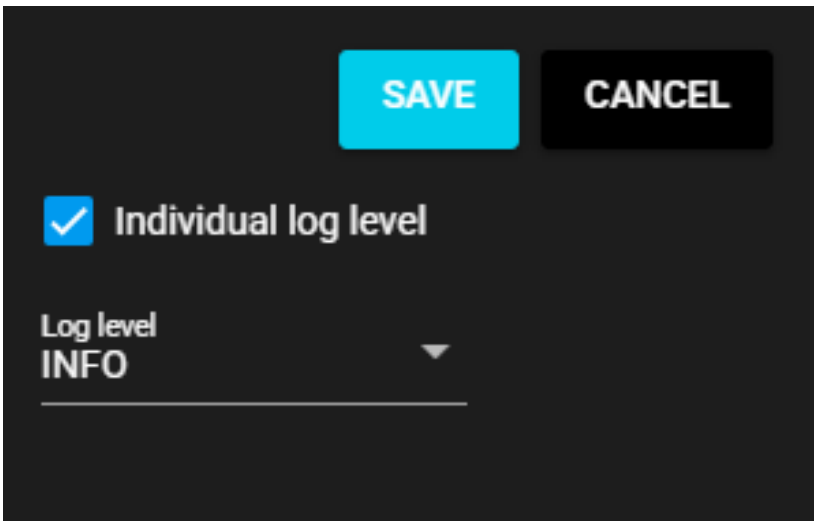


개별 로그 레벨 설정에 표시되는 내용은, 다음과 같습니다.

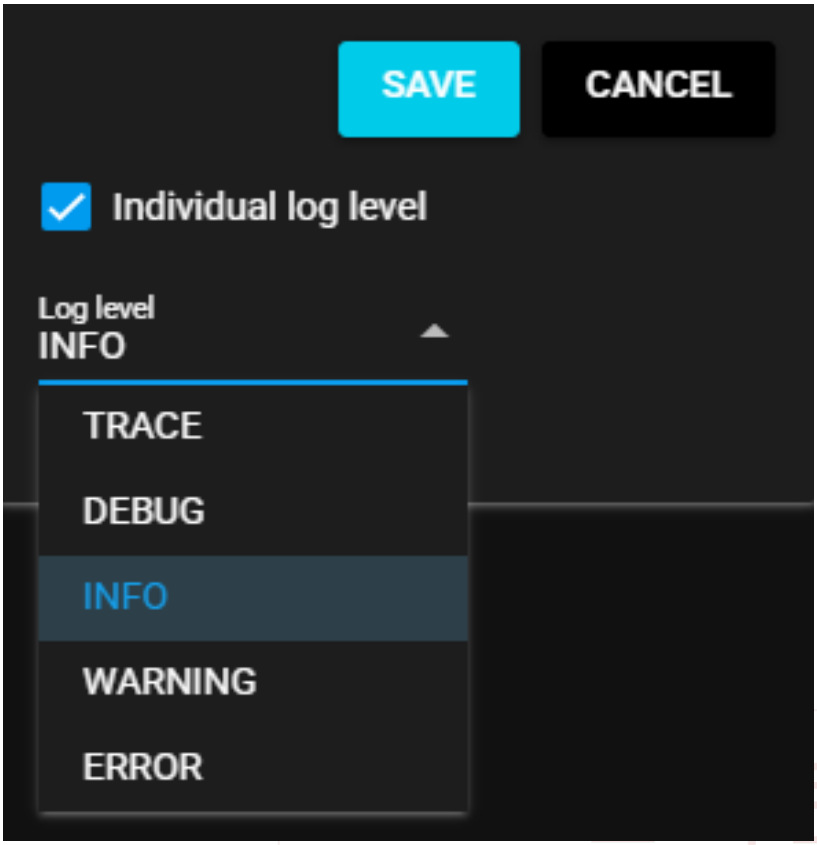
항목	설명
저장	설정 내용을 저장
취소	설정 내용을 파기
개별 로그 레벨	개별 로그 레벨의 유효/무효
로그 레벨	로그 출력의 레벨

3. 「 개별 로그 레벨 」을 유효로 합니다.

-For Reference Only-

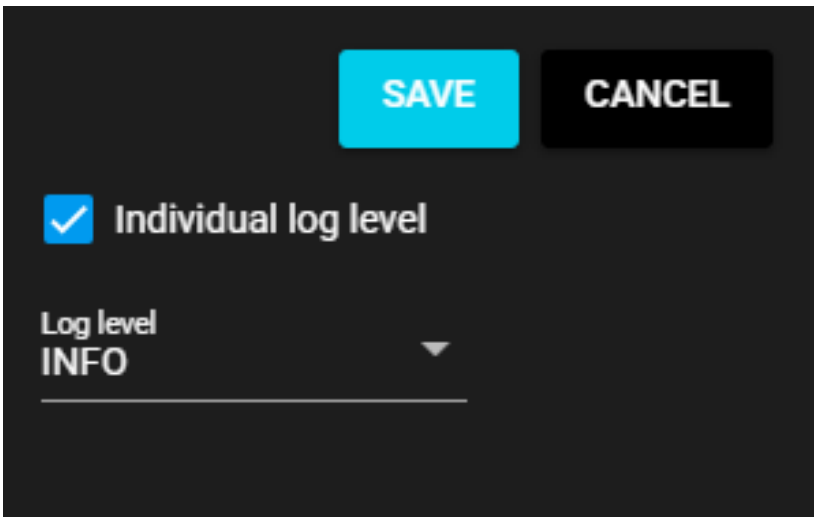


4. 「 로그 레벨 」을 선택합니다.



5. 「 저장 」을 누릅니다.

-For Reference Only-



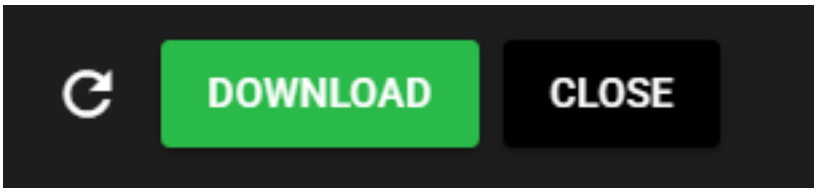
설정을 변경하지 않는 경우는 「 취소 」를 누르십시오.

> 이 조작은 Edit Mode에서만 수행할 수 있습니다.

5.14.4 다운로드

다음 순서로 SpeeDBee Synapse의 각종 로그를 다운로드할 수 있습니다. 컴포넌트 개별 로그도 다운로드 대상에 포함되지만, 모든 컴포넌트나 SpeeDBee Synapse의 가동 로그도 포함된 파일이 됩니다.

1. 다운로드 버튼을 누릅니다.



2. ZIP 파일이 다운로드됩니다. ZIP 파일은 아래의 구성으로 로그 파일을 격납하고 있습니다.

ZIP 파일 구성

```
├──webui
|   ├──operation_log.YYYYMM.csv（SpeeDBee Synapse의 조작 로그）
|   └──speedbeesynapse-webui-backend.log（WEBUI의 운용 로그）
├── speedbeesynapse.log（코어, 컴포넌트, 데이터베이스의 운용 로그）
└── speedbeesynapse-YYYYMMDDHHMMSS.log（speedbeesynapse.log의 오래된 버전）
```

5.15 DB 쿼리（SQL）

여기에서는, DB 쿼리（SQL）에 대해 설명합니다.

5.15.1 DB 쿼리의 개요

DB 쿼리 서비스를 기동하면, 시계열 데이터베이스에 대해 Web-API（HTTP） 경유로 SQL（Structured Query Language）에 의한 조회가 가능합니다. 구문은, SpeeDBee Synapse 독자적인 것이지만, 일반적인 RDB에서 사용되는 SQL과 가까운 구문 사양으로 되어 있습니다.

5.15.2 DB 쿼리의 상세

여기에서는, Web-API 사양 및 구문에 대해 설명합니다.

5.15.2.1 Web-API 사양

DB 쿼리 서비스의 Web-API 사양은, 이쪽을 참조하십시오.

5.15.2.2 구문 사양

5.15.2.2.1 SELECT

SELECT 구는, 시계열 데이터베이스 내에 등록되어 있는 데이터를 참조하는 구문입니다.

AI 번역본
-For Reference Only-

1. 예문：리소스 컬렉터 r1의 cpu_usage를 현재 시각까지를 범위로 하여, 5건 취득한다

```
SELECT "r1:cpu_usage" FROM SPDB WHERE _ts < NOW LIMIT 5;
```

2. 예문：리소스 컬렉터 r1의 cpu_usage를 현재 시각까지를 범위로 하고, 또한 cpu_usage가 10 이상인 데이터를 취득한다

```
SELECT "r1:cpu_usage" FROM SPDB WHERE _TS < NOW AND "r1:cpu_usage" >= 10
```

3. 예문：PLC 컬렉터 plc1의 dataA의 최대값과 최소값에 대해, 분석의 시간 폭 5초로 취득한다

```
SELECT MAX("plc1:dataA"),MIN("plc1:dataA") FROM SPDB WHERE _ts < NOW GROUP BY TUMBLING(5,sec)
LIMIT 10;
```

구문 사양

```
SELECT <select_list> FROM <table> [WHERE <condition>] [GROUP BY <window_condition> [HAVING
<condition>]] [ORDER BY _TS [ASC/DESC]] [LIMIT <number> [OFFSET <start>]|LIMIT <start>,<row_count>]
```

예약어에 대해서는, 대문자, 소문자를 구별하지 않습니다.

- <select_list>

추출할 컬럼이나 식을 지정합니다. 지정은 콤마 구분으로 복수의 지정이 가능합니다. 또한, SpeeDBee Synapse에서 사용하는 컬럼은, 반드시 더블 쿼테이션 (") 으로 컬럼명을 감싸서 지정하십시오. 또한 더블 쿼테이션으로
감싸인 컬럼명은 대문자・소문자를 구별합니다.

예1：컬럼명

```
"plcA:data1", "computeB:DATAa"
```

예2：정수, 부동소수, 문자열

```
3, -5, 1.52, 0.32e2, 'TEST-STR'
```

SQL 내의 정수는 int64로서 취급합니다. 부동소수는 double로서 취급하며, 지수 표현도 가능합니다. 문자열은, 싱글 쿼테이션으로 감쌉니다. 더블 쿼테이션은, 컬럼명 등의 식별자로서 취급됩니다.

AI 번역본
-For Reference Only-

예3：식

```
"plcA:data2" + 3
```

사용 가능한 연산자	설명
+	가산
-	감산

사용 가능한 연산자	설명
*	승산
/	제산
%	나머지

예4：함수

```
MAX("compute1:dataA"), AVG("compute2:dataB")
```

사용 가능한 수학 함수	반환값	설명
ABS	INT64	절대값 (정수)
ACOS	DOUBLE	삼각함수 역여현
ASIN	DOUBLE	삼각함수 역정현
ATAN	DOUBLE	삼각함수 역정접
CEIL	DOUBLE	소수점 이하 올림
COS	DOUBLE	삼각함수 여현
COSH	DOUBLE	쌍곡선 여현
EXP	DOUBLE	밑이 네이피어 수 e인 지수함수
FABS	DOUBLE	절대값 （ 배정밀도 부동소수 ）
FLOOR	DOUBLE	소수점 이하 버림
FMOD	DOUBLE	나머지
LOG	DOUBLE	자연로그
LOG10	DOUBLE	10을 밑으로 하는 로그
POW	DOUBLE	거듭제곱

AI 번역본
-For Reference Only-

사용 가능한 수학 함수	반환값	설명
ROUND	DOUBLE	소수점 이하 반올림
SIN	DOUBLE	삼각함수 정현
SINH	DOUBLE	쌍곡선 정현
SQRT	DOUBLE	제곱근
TAN	DOUBLE	삼각함수 정접
TANH	DOUBLE	쌍곡선 정접

사용 가능한 집약 · 통계 함수	반환값	설명
AVEDEV	DOUBLE	평균편차
AVG	DOUBLE	산술평균
COUNT	INT64	건수
CV	DOUBLE	변동계수
DOF	INT64	자유도
MAX	DOUBLE	최대
MIN	DOUBLE	최소
SE	DOUBLE	표준오차
STDEV	DOUBLE	불편 표준편차
STDEVP	DOUBLE	표준편차
SUM	DOUBLE	합계
SUMSQ	DOUBLE	2제곱합

AI 번역본
-For Reference Only-

사용 가능한 집약 · 통계 함수	반환값	설명
VAR	DOUBLE	불편분산
VARP	DOUBLE	표본분산

사용 가능한 대표값	반환값	설명
LAST	값의 데이터형	윈도우 최종값
TOP	값의 데이터형	윈도우 선두값

예5 : 상수

```
M_PI
```

사용 가능한 상수	반환값	설명
M_E	DOUBLE	네이피어 수, 자연밑수의 밑 (e)
M_LOG2E	DOUBLE	log_e(e)
M_LOG10E	DOUBLE	log_10(e)
M_LN2	DOUBLE	log_e(2)
M_LN10	DOUBLE	log_e(10)
M_PI	DOUBLE	원주율 (Pi)
M_PI_2	DOUBLE	Pi/2
M_PI_4	DOUBLE	Pi/4
M_1_PI	DOUBLE	1/Pi
M_2_SQRTPI	DOUBLE	2/Pi
M_SQRT2	DOUBLE	sqrt(2)

AI 번역본
-For Reference Only-

사용 가능한 상수	반환값	설명
M_SQRT1_2	DOUBLE	1/sqrt(2)

예6 : 의사 컬럼

```
_WID, _TS
```

사용 가능한 의사 컬럼	설명
_WID	윈도우 ID (GROUP BY 사용 시의 분석 폭 단위로 변화하는 ID)
_TS	타임스탬프 (나노초)

- <table>

시계열 데이터의 검색 대상을 지정합니다.

메모리와 파일의 양쪽을 검색하는 경우

```
SPDB
```

메모리만 검색하는 경우

```
SPDB.MEM
```

파일만 검색하는 경우

```
SPDB.FILE
```

- <condition>

조건식을 지정합니다.

예1 : 메모리 사용률이 50 미만 또는 CPU 사용률이 30 이하

```
"r1:mem_usage" < 50 OR "r1:cpu_usage" <= 30
```

AI 번역본
-For Reference Only-

사용 가능한 비교 연산자	설명
=	등가이다
!=	불일치

사용 가능한 비교 연산자	설명
<>	불일치
<	미만
>	초과
<=	이하
>=	이상
IS [NOT] NULL	NULL이다, NULL이 아니다
BETWEEN AND	범위 내
BETWEEN EXCLUSIVE AND EXCLUSIVE	지정 범위를 포함하지 않음

사용 가능한 논리 연산자	설명
AND	논리곱
OR	논리합
NOT	부정

_TS에 지정 가능한 연산자	설명
_TS = 100000000	지정한 나노초와 등가
_TS > 100000000	지정한 나노초를 초과
_TS >= 100000000	지정한 나노초 이상
_TS < 100000000	지정한 나노초 미만
_TS <= 100000000	지정한 나노초 이하
_TS BETWEEN 1000000000 AND 9000000000	지정한 나노초의 범위

AI 번역본
-For Reference Only-

_TS에 지정 가능한 연산자	설명
_TS BETWEEN 1000000000 EXCLUSIVE AND 9000000000 EXCLUSIVE	지정한 나노초를 제외한 범위
_TS > '20230808T01:23:45.111222333'	ISO8601 포맷에서의 문자열 지정에서의 비교
_TS < NOW	현재 시각과의 비교
_TS < NOW + 1 hour	현재 시각+1시간까지를 범위로 한다
_TS < NOW + 10 minutes	현재 시각부터 +10분까지를 범위로 한다
_TS < NOW + 5 sec	현재 시각부터 +5초까지를 범위로 한다
_TS < NOW - 100 msec	현재 시각부터 -100밀리초까지를 범위로 하다
_TS < NOW - 100 usec	현재 시각부터 -100마이크로초까지를 범위로 한다
_TS < NOW - 100 nsec	현재 시각부터 -100나노초까지를 범위로 하다

※_TS는 좌변에만 사용할 수 있습니다.

또한, 시간 조건은 1 개의 SELECT 구에서 복수 지정할 수는 없습니다. 다른

조건과 사용하는 경우는, AND로 접속합니다. OR 접속은 할 수 없습니다.

- <window_condition>

AI 번역본
-For Reference Only-

윈도우 분석의 조건을 지정합니다. 윈도우는 2종류 있으며, 시간 폭으로 분석하는 윈도우를 구획하는 텀블링 윈도우와 값이 변화한 것으로 윈도우를 구획하는 밸류 체인지 윈도우를 사용할 수 있습니다.

예1：텀블링 윈도우（5초로 분석 윈도우를 분할）

```
GROUP BY TUMBLING(5,sec)
```

예2：밸류 체인지 윈도우（plcA:pressure가 변화하면 분석 윈도우를 구획）

```
GROUP BY VCHG("plcA:pressure")
```

예3：밸류 체인지 윈도우（plcA:pressure가 변화, 또는 30초 경과로 분석 윈도우를 구획）

```
GROUP BY VCHG("plcA:pressure", 30, sec)
```

예4 : 텀블링 윈도우 + HAVING (5초로 분석 윈도우를 분할, plcA:pressure의 최대값이 10 이상인 것을 추출)

```
GROUP BY TUMBLING(10,sec) HAVING MAX("plcA:pressure") >= 10
```

예5 : 밸류 체인지 윈도우+HAVING (plcA:pressure가 변화, 또는 30초 경과로 분석 윈도우를 구획하고, plcA:pressure의 최소값이 5 미만인 것을 추출)

```
GROUP BY VCHG("plcA:pressure", 30, sec) HAVING MIN("plcA:pressure") < 5
```

- <row_count>

출력 행수를 정수로 지정합니다.

취득할 데이터 건수를 3건으로 하는 경우

```
LIMIT 3
```

- <start>

개시 위치를 건너뛸 행수를 정수로 지정합니다.

선두부터 5건째 이후의 3건을 취득하는 경우

```
LIMIT 5,3
```

- ORDER BY에 대해

ORDER BY는 _TS만 지정 가능합니다. ORDER BY의 지정이 없는 경우, ORDER BY _TS ASC에 의한 오름차순 취득이 지정된 것이 됩니다.

내림차순 지정

```
ORDER BY _TS DESC
```

AI 번역본
-For Reference Only-

- 배열형의 컬럼에 대한 지정에 대해

예1 : com1:a가 배열형인 경우, 모든 배열 요소를 참조합니다.

```
SELECT "com1:a" FROM SPDB;
```

예2 : com1:a가 배열형인 경우, 선두 요소와 2번째의 요소를 참조합니다.
또한 조건식에 배열 요소를 지정하고 있습니다.

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE "com1:a"[0] > 1;
```

예3 : com1:a가 배열형인 경우, 조건식에 배열을 지정하고 있습니다.
이 경우, 암묵적으로 com1:a[0]과 같은 의미가 됩니다.

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE "com1:a" > 1;
```

예4：배열 검색 ANY의 예입니다. com1:a의 요소 수는 2로 합니다. 다음의 2개의 예문은, 같은 의미가 됩니다.
ANY는 모든 요소와의 비교를 수행하여, 어느 하나라도 참이 되는 경우에, 참이 성립하는 지정입니다.

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE 1 < ANY("com1:a");
```

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE 1 < "com1:a"[0] OR 1 < "com1:a"[1];
```

※ANY는 우변에만 사용할 수 있습니다.

예5：배열 검색 ALL의 예입니다. com1:a의 요소 수는 2로 합니다. 다음의 2개의 예문은, 같은 의미가 됩니다.
ALL은 모든 요소와의 비교를 수행하여, 모두가 참이 되는 경우에, 참이 성립하는 지정입니다.

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE 1 < ALL("com1:a");
```

```
SELECT "com1:a"[0], "com1:a"[1] FROM SPDB WHERE 1 < "com1:a"[0] AND 1 < "com1:a"[1];
```

※ALL은 우변에만 사용할 수 있습니다.

5.15.2.2.2 SHOW

SHOW 구는, 시계열 데이터베이스 내의 메타 정보를 참조하는 구문입니다. 현재는, 에일리어스명（SpeeDBee Synapse에서 작성되는 컬럼명에 상당）만 참조가 가능합니다.
ORDER BY를 생략하면, 정렬 순서는 보장되지 않습니다.

- SHOW ALIAS

예1：에일리어스명의 목록을 취득（오름차순）

```
SHOW ALIAS ORDER BY ALIAS_NAME ASC
```

예2：에일리어스명의 목록을 취득（내림차순）

```
SHOW ALIAS ORDER BY ALIAS_NAME DESC
```

구문 사양

```
SHOW ALIAS [ORDER BY ALIAS_NAME [ASC/DESC]]
```

AI 번역본
-For Reference Only-

5.15.2.2.3 SET

글로벌 변수에 값을 대입하는 구문입니다. SQL 내에서는, 글로벌 변수의 선두에는, 반드시 $를 부여할 필요가 있습니다.

예1 : 정수의 대입

```
SET $GVAR0 TO 10;
```

예2 : 마이너스 값의 대입

```
SET $PVAR0 -111;
```

예3 : 부동소수의 대입

```
SET $GVAR1 1.235;
```

예4 : 지수 표현에서의 대입

```
SET $PVAR1 -1.234E-2
```

예5 : 문자열의 대입

```
SET $PVAR2 'abcd'
```

대입한 글로벌 변수의 참조 예

```
SELECT \"com1:a\",$GVAR1, $PVAR2 FROM SPDB WHERE _ts BETWEEN NOW - 1 sec AND NOW ORDER BY _TS DESC
LIMIT 1;
```

구문 사양

```
SET 글로벌 변수명 [TO] 리터럴값
```

5.15.2.2.4 INSERT

INSERT 구는, 시계열 데이터베이스에 데이터를 등록하는 구문입니다. 존재하는 컴포넌트의 컬럼에 대해, 등록이 가능합니다. 등록 가능한 컬럼명의 목록은, SHOW로 취득할 수 있습니다.
INSERT 문의 [INTO] 직후의 테이블 지정은, SPDB 고정이 됩니다.

AI 번역본
-For Reference Only-

예1 : 리소스 컬렉터 「 resource1(리소스 컬렉터) 」의 컬럼 「 cpu_usage 」에, 값 「 2 」를 삽입한다

```
INSERT INTO SPDB (\"resource1:cpu_usage\") VALUES (2);
```

※타임스탬프를 생략한 경우, 등록되는 타임스탬프는 가동 머신의 시스템 시각이 사용됩니다.

예2 : 리소스 컬렉터 「 resource1(리소스 컬렉터) 」의 컬럼 「 cpu_usage 」에 시각 「 2025년 04월 03일 13시 02분 09초 」를 지정하여, 값 「 10 」을 삽입한다

ISO8606 형식으로 지정하는 경우

```
INSERT INTO SPDB (_TS, \"resource1:cpu_usage\") VALUES ('2025-04-03T13:02:09', 10);
```

에포크 나노초로 지정하는 경우

```
INSERT INTO SPDB (_TS, \"resource1:cpu_usage\") VALUES (1743685329000000000, 10);
```

예3 : 정수형의 값 「 10 」을 삽입한다

```
INSERT INTO SPDB (\"customcomp1:columnA\") VALUES (10);
```

예4 : 부동소수형의 값 「 3.14 」를 삽입한다

```
INSERT INTO SPDB (\"customcomp1:columnB\") VALUES (3.14);
```

예5 : 문자형의 값 「 ABCDE 」를 삽입한다

```
INSERT INTO SPDB (\"customcomp1:columnC\") VALUES ('ABCDE');
```

예6 : 배열의 값 {3,4}를 삽입한다

```
INSERT INTO SPDB (\"customcomp1:columnD\") VALUES ({3,4});
```

예7 : 하나의 컬렉터 & 복수의 컬럼에, 복수의 데이터를 삽입한다

```
INSERT INTO SPDB (\"customcomp1:columnA\",
\"customcomp1:columnB\",\"customcomp1:columnC\",\"customcomp1:columnD\",\"customcomp1:columnE\")
VALUES (10,3.14,'ABCDE',{3,4},'00000000000000000000000000')
```

AI 번역본
-For Reference Only-

예8 : 복수의 컬렉터 & 복수의 컬럼에, 복수의 데이터를 삽입한다

```
INSERT INTO SPDB (\"resource1:cpu_usage\", \"customcomp1:columnA\",
\"customcomp1:columnB\",\"customcomp1:columnC\",\"customcomp1:columnD\",\"customcomp1:columnE\")
VALUES (99,10,3.14,'ABCDE',{3,4},'00000000000000000000000000');
```

구문 사양

```
INSERT [INTO] SPDB (_TS, <column_name>,...) VALUES (타임스탬프값, value,...)
```

5.15.3 DB 쿼리 서비스 이용 시의 주의점

DB 쿼리 서비스 이용 시의 주의점을 설명합니다.

SpeeDBee Synapse의 DB 쿼리 서비스는, 고부하 상황에서도 안정 가동을 유지하는 설계를 지향하고 있습니다만, DB 쿼리의 실행에 관해서는, 시스템의 부하 상황에 크게 좌우되는 특성이 있습니다.
특히, 시스템이 고부하로 동작하고 있는 경우, DB 쿼리의 응답 성능이 일시적으로 저하될 가능성이 있습니다.

> ⚠ 쿼리 조건이 미치는 시스템에의 부하
>
> 데이터베이스에 대한 쿼리는, 지정하는 데이터의 대상 범위가 넓어지면 넓어질수록, 시스템에의 부하가 높아집니다. 예를 들어 기간을 넓게 잡는 쿼리나, 좁히기 조건이 적은 쿼리는, 대량의 데이터를 주사할 필요가 있으므로, 응답 시간이 길어질 뿐만 아니라, 다른 처리에도 영향을 미칠 가능성이 있습니다.

안정된 시스템 운용을 위해, 쿼리 실행 시에는 이하의 점에 유의하십시오.

> ✎ 쿼리 실행의 유의 사항
>
> - 액세스 토큰을 설정한다
> 액세스 토큰을 발행하여, DB 쿼리 서비스에 적용함으로써, DB 쿼리를 이용할 수 있는 이용자를 제한할 수 있습니다.
>
> - 필요 최소한의 범위에서 쿼리를 실행한다
> 필요한 데이터로 좁히고, 기간이나 조건을 가능한 한 구체적으로 설정하십시오.
> - 고부하 시를 피한다
> 가능하다면, 시스템 이용이 집중되는 시간대를 피하여 쿼리를 실행할 것을 권장합니다.

AI 번역본
-For Reference Only-

5.16 연산식・조건식

여기에서는, SpeeDBee Synapse에서 이용 가능한 연산식, 조건식에 대해 설명합니다.

5.16.1 연산식・조건식의 개요

SpeeDBee Synapse 내에서, 연산식과 조건식을 사용할 수 있는 기능을 나타냅니다.

항목	연산식	조건식	설명
연산 컬렉터	○	-	임의의 연산 결과를 컬렉터의 데이터로서 등록
이벤트 데이터	-	○	조건식이 참이 되는 경우에, 조건에 합치한 데이터로서 등록
이벤트 트리거	-	○	조건식이 참이 되는 경우에, 그 이벤트 구간의 정보를 출력
출력 포트 수치 변환	○	-	임의의 연산 결과를 컬렉터의 데이터로서 등록
출력 포트 등록 조건	-	○	조건식이 참이 되는 경우에, 대상의 데이터를 등록

5.16.2 연산식과 조건식（공통）

연산식과 조건식, 공통으로 사용할 수 있는 식에 대해, 설명합니다.

항목	
예	($VALUE + 10) * ABS(5)
사용 가능한 비교 연산	=,<>,!=,<,>,<=,>=
사용 가능한 산술 연산	(),+,-,*,/,%
사용 가능한 논리 연산	AND,OR,NOT
제어문	IF

사용 가능한 변수	설명
$VALUE	컬렉터에서 받아들인 값, 혹은, 변환식에 의한 계산 결과
$PVALUE	지난번 DB에 등록된 값
$GVAR0～63	글로벌 변수
$PVAR0～63	영속화 글로벌 변수
	※글로벌 변수에 대해서는 이쪽을 참조하십시오.

사용 가능한 상수	설명
M_E	네이피어 수, 자연수의 밑（e）
M_LOG2E	log_e(e)
M_LOG10E	log_10(e)
M_LN2	log_e(2)
M_LN10	log_10(e)
M_PI	원주율（Pi）
M_PI_2	Pi/2
M_PI_4	Pi/4
M_1_PI	1/Pi
M_2_SQRTPI	2/Pi
M_SQRT2	sqrt(2)
M_SQRT1_2	1/sqrt(2)

AI 번역본
-For Reference Only-

사용 가능한 함수	설명
ABS	절대값 (정수)
ACOS	삼각함수 역여현
ASIN	삼각함수 역정현
ATAN	삼각함수 역정접
CEIL	소수점 이하 올림
COS	삼각함수 여현
COSH	쌍곡선 여현
EXP	밑이 네이피어 수 e인 지수함수
FABS	절대값 (배정밀도 부동소수)
FLOOR	소수점 이하 버림
FMOD	나머지
LOG	자연로그
LOG10	10을 밑으로 하는 로그
POW	거듭제곱
ROUND	소수점 이하 반올림
SIN	삼각함수 정현
SINH	쌍곡선 정현
SQRT	제곱근
TAN	삼각함수 정접
TANH	쌍곡선 정접

AI 번역본
-For Reference Only-

5.16.3 연산식

연산식에 대해, 설명합니다. 연산식은, 임의의 연산 결과를 산출하는 경우에 지정하는 식입니다. 컬렉터에서 취득한 값에, 어떠한 연산이 필요한 경우 등에 이용할 수 있습니다.

항목	
예	($VALUE + 10) * ABS(5)

글로벌 변수의 대입에 대해（참고）

GVAR1 :=(식)

- 대입할 글로벌 변수의 선두에는 $는 필요 없습니다.
- "GVAR1 := $VALUE * 2"와 같이 대입을 지정하면, 연산 결과로서도 $VALUE에 2를 승산한 값이 반환됩니다.

예
GVAR1 := ($VALUE * 3)
($VALUE를 2로 가정한 경우)
①GVAR1 에는 2 * 3 = 6 의 값이 격납됩니다.
②연산 결과는, GVAR1에 대입된 값（6）과 같아집니다.

IF문을 사용한 대입

IF(조건식,참인 경우의 값,거짓인 경우의 값)

GVAR1 := IF($VALUE = 1,$GVAR1 + 1, $GVAR1)

상기의 경우, $VALUE가 1인 경우에, 참이 되면 $GVAR1+1의 결과가 반환됩니다.
거짓인 경우는, $GVAR1이 원래의 값 그대로 대입됩니다.

주의점으로서,「참인 경우의 값」과「거짓인 값의 경우」내에서, 대입을 수행하면, 참거짓에 관계없이 매번 평가되어 버립니다.

IF($VALUE=1,GVAR1 := $GVAR1 + 1, $GVAR1)
이러한 지정을 하면, 참거짓에 관계없이「참인 경우의 값」이 식으로서 평가되어, 매번 대입되어 버립니다. 의도하지 않은 값이 될 가능성이 있으므로, 대입은, IF문의 밖에서 지정하십시오.

AI 번역본
For Reference Only

5.16.4 조건식

조건식에 대해, 설명합니다. 조건식은, 임의의 조건의 참거짓을 얻는 경우에 지정하는 식입니다. 참거짓을 얻음으로써, 그 앞의 출력에의 제어에 이용할 수 있습니다.

항목	
예	($VALUE + 10) * ABS(5) < 100

글로벌 변수의 대입에 대해（참고）

1 < (PVAR5 :=(식))

- 대입할 글로벌 변수의 선두에는 $는 필요 없습니다.
- "PVAR5 := ($VALUE * 2)"와 같이 대입만을 지정할 수는 없습니다. 조건식으로서의 지정이 필요합니다.

예
GVAR1 := ($VALUE * 3)
($VALUE를 2로 가정한 경우)
①GVAR1 에는 2 * 3 = 6 의 값이 격납됩니다.
②연산 결과는, GVAR1에 대입된 값（6）과 같아집니다.

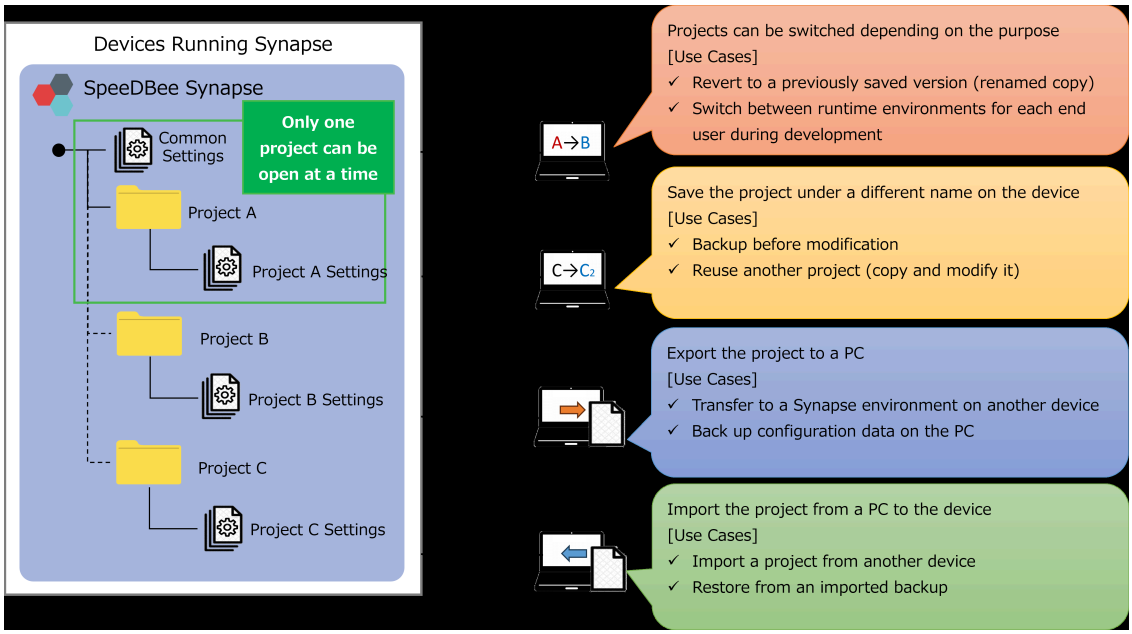
AI 번역본
-For Reference Only-

5.17 설정 정보의 관리(프로젝트)

5.17.1 프로젝트의 개요

프로젝트는, SpeeDBee Synapse의 설정 내용을 관리하는 구조입니다. SpeeDBee Synapse의 설정 내용은, 공통 설정과 프로젝트 개별의 설정으로 나뉩니다. 프로젝트를 전환함으로써, 동일한 디바이스상에서 복수의 설정을 관리할 수 있게 됩니다. 또한 프로젝트 설정을 익스포트하여, 다른 디바이스에 임포트하는 등의 기능이 있습니다.

이하는 프로젝트 기능의 이미지도입니다.

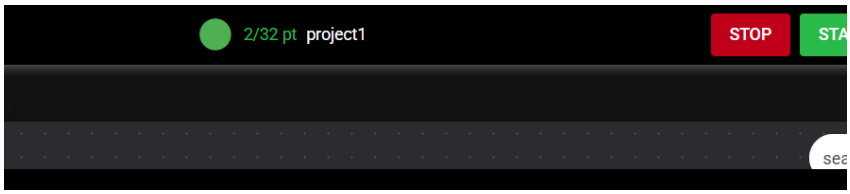


이하는 공통 설정과 프로젝트 개별의 설정의 목록입니다.

공통/프로젝트 고유	설정 내용
공통	・라이선스 정보 ・사용자 관리/인증 ・암호화 통신 ・서버 설정 ・로그 ・데이터베이스 ・액세스 토큰
프로젝트 고유	・설정되어 있는 데이터 플로우(패널, 컴포넌트, 플로우 링크) ・등록한 커스텀 컴포넌트 ・등록한 의존 모듈（Python） ・컴포넌트가 출력한 데이터 ・Grafana 설정

AI 번역본
-For Reference Only-

SpeeDBee Synapse의 동작 중은, 반드시 1 개의 프로젝트가 선택되어 있습니다. 현재 선택 중인 프로젝트는 헤더에 표시됩니다.



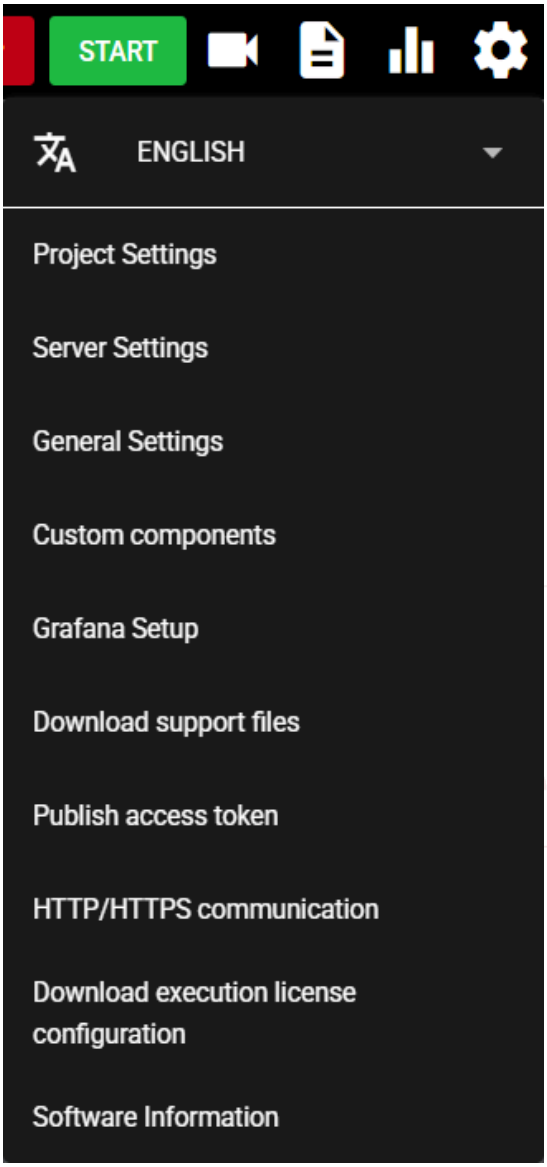
현재 선택 중인 프로젝트에 대해 이하의 조작을 수행할 수는 없습니다.

・삭제

5.17.2 신규 작성

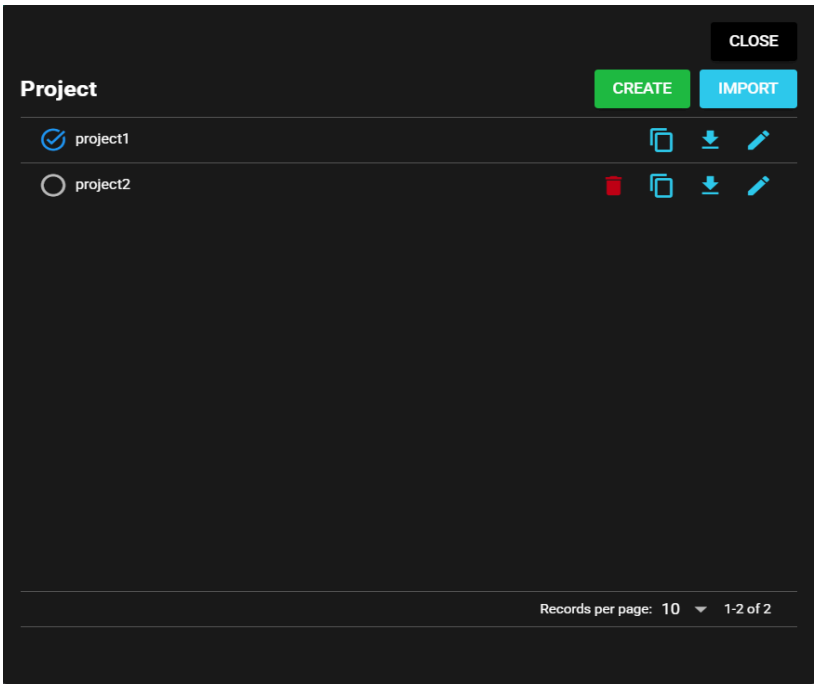
다음 순서로 프로젝트를 작성할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.

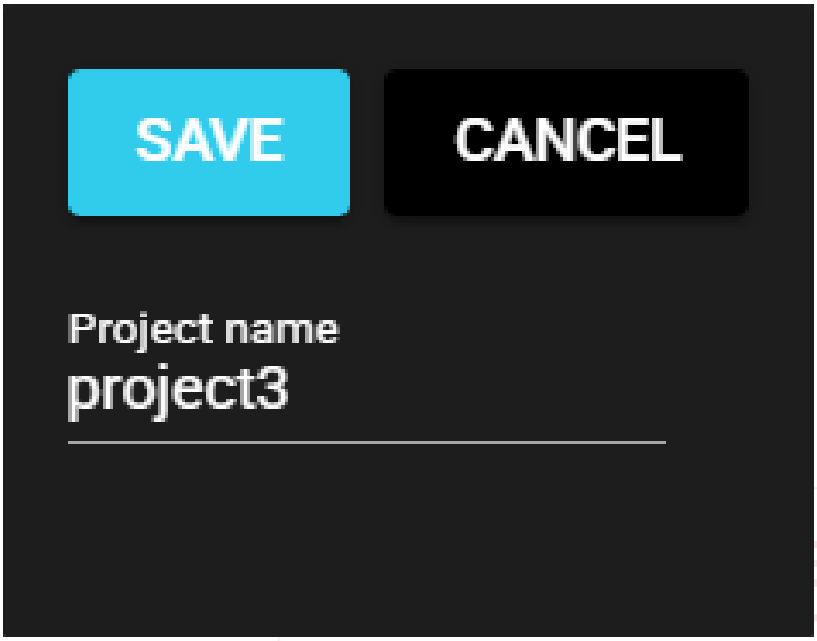


번역본
ference Only-

2.「 신규 작성 」 버튼을 누릅니다.

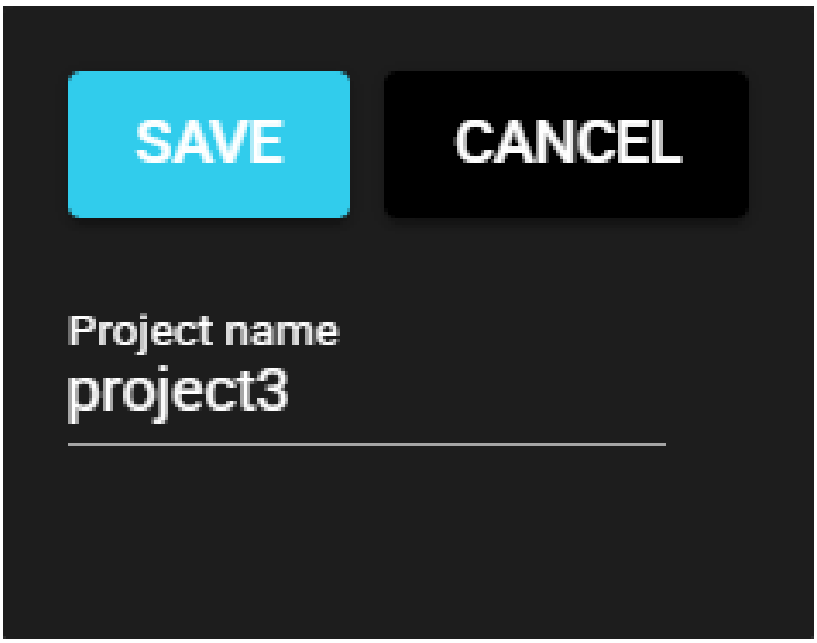


3. 신규 작성할 프로젝트의 이름을 입력합니다.



4.「 저장 」을 누릅니다.

-For Reference Only-



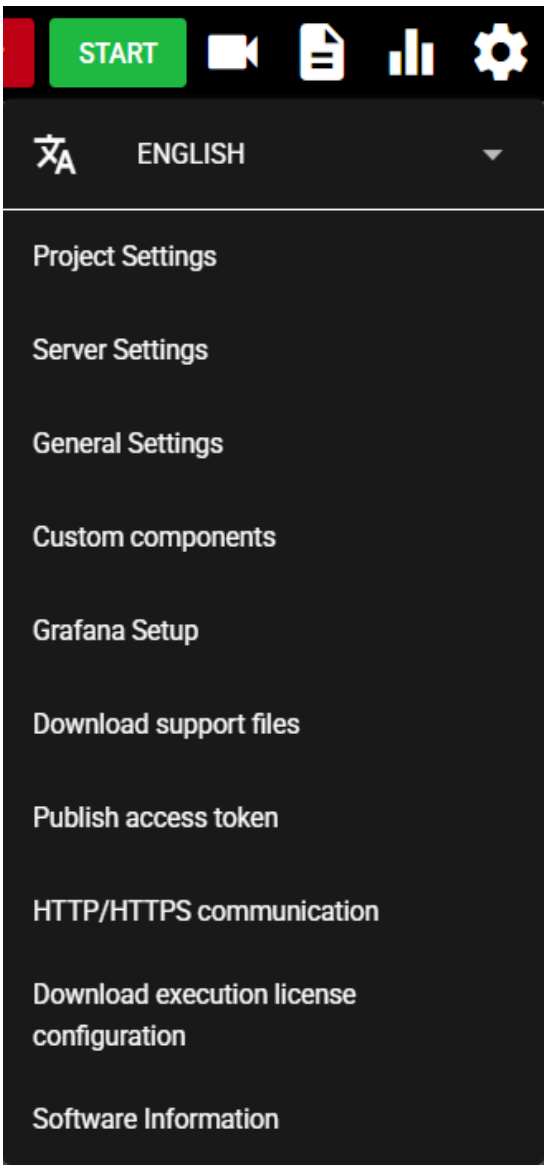
5. 새로운 프로젝트가 작성됩니다.

5.17.3 복제

다음 순서로 프로젝트의 카피를 작성할 수 있습니다.

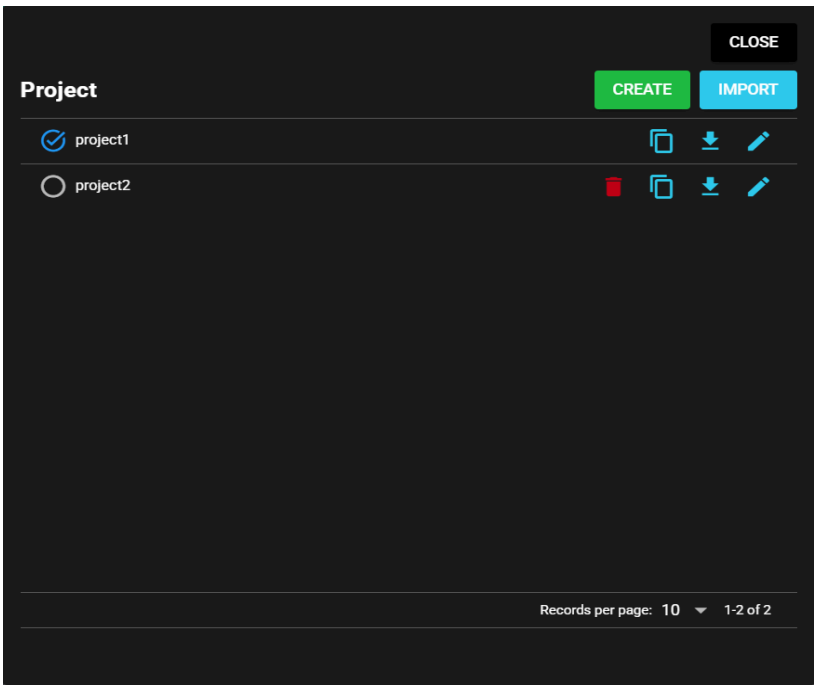
1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.

AI 번역본
-For Reference Only-

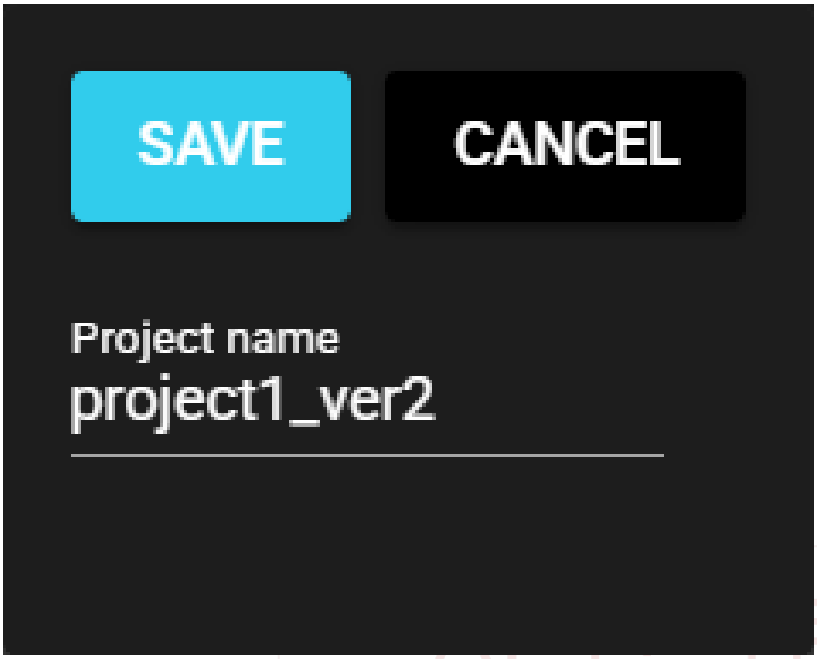


2. 복제할 프로젝트의「카피 아이콘（복제）」을 누릅니다.

AI 번역본
-For Reference Only-

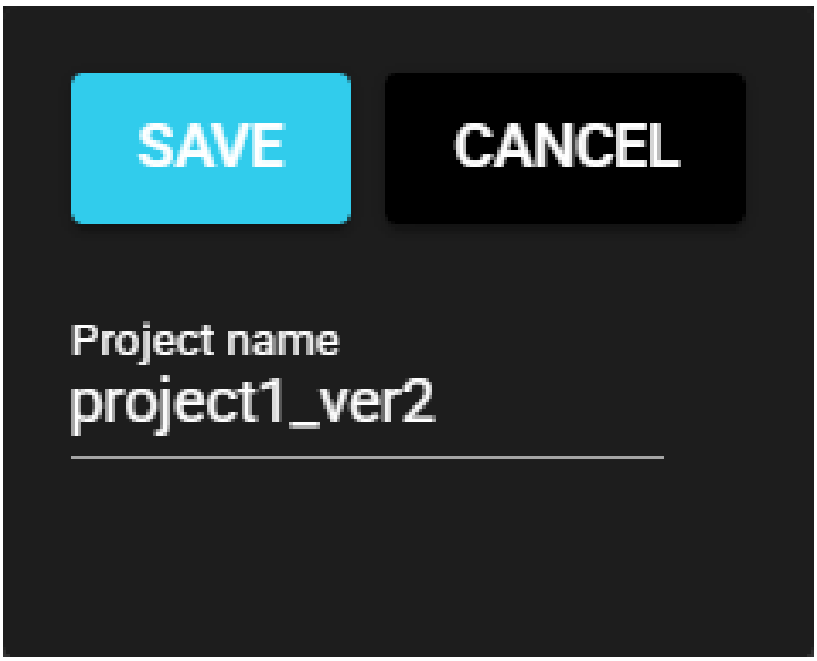


3. 복제의 프로젝트명을 입력합니다.



4.「저장」을 누릅니다.

-For Reference Only-



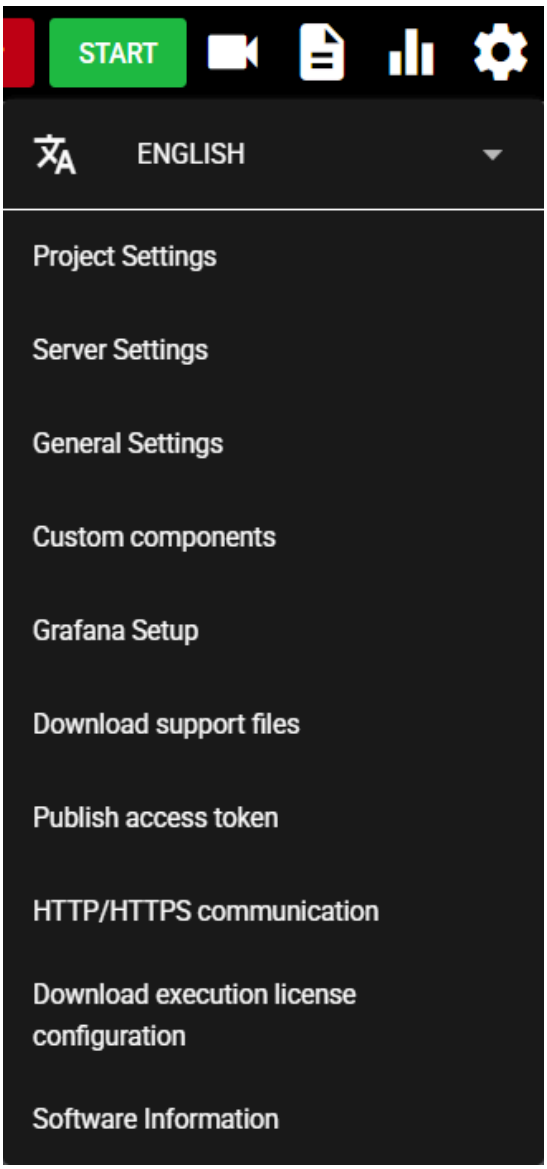
5. 프로젝트의 복제가 작성됩니다.

5.17.4 임포트

다음 순서로 프로젝트를 임포트할 수 있습니다.

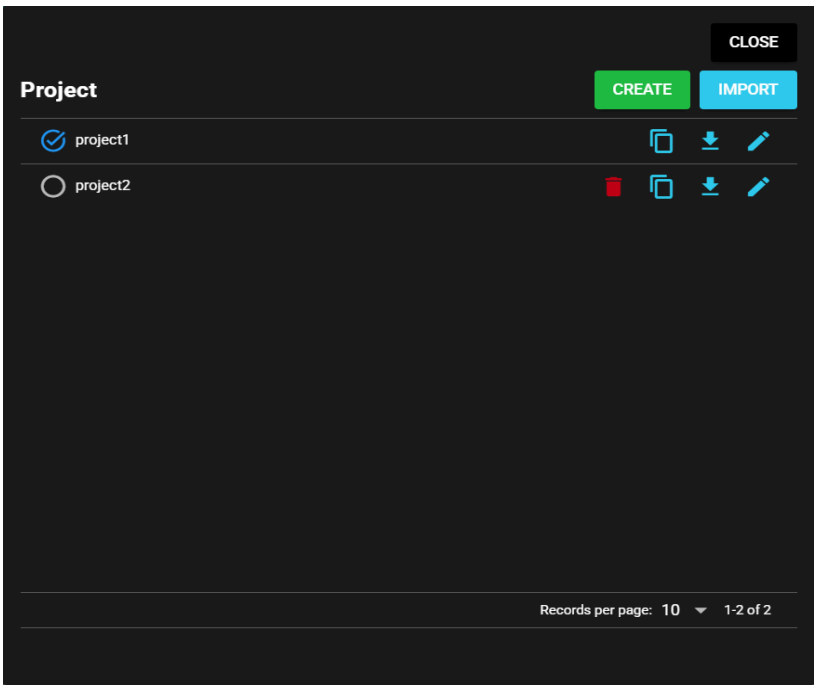
1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.

AI 번역본
-For Reference Only-

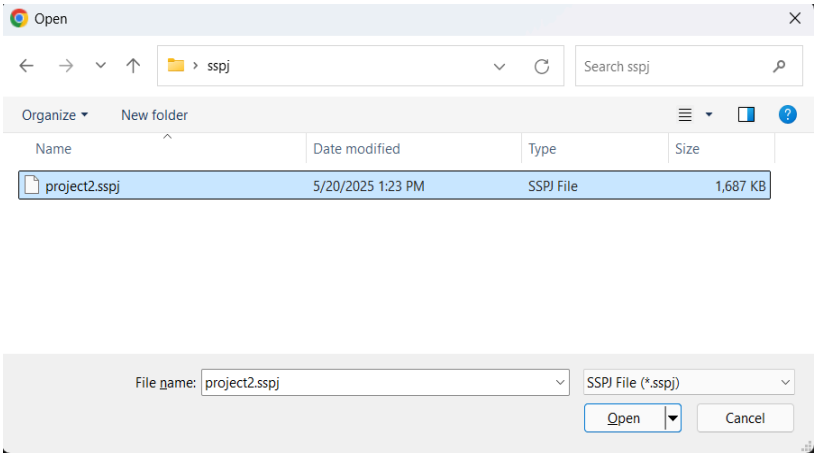


2.「 임포트 」버튼을 누릅니다.

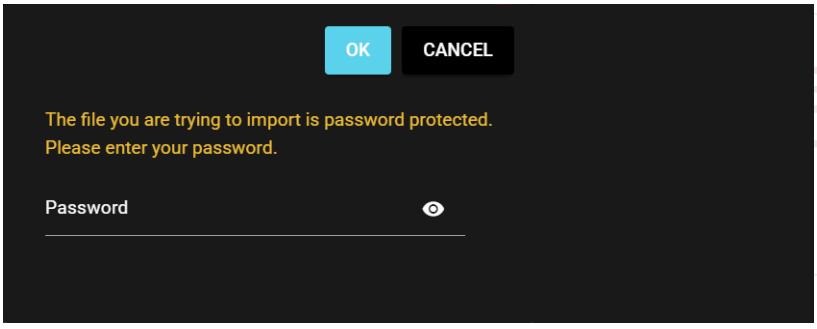
AI 번역본
-For Reference Only-



3. 「 익스포트 」의 순서로 익스포트한 파일을 선택합니다.



4. 파일이 패스워드로 보호되어 있는 경우는, 패스워드를 입력하여 「 OK 」를 선택합니다.



5. 프로젝트가 임포트됩니다.

주의사항

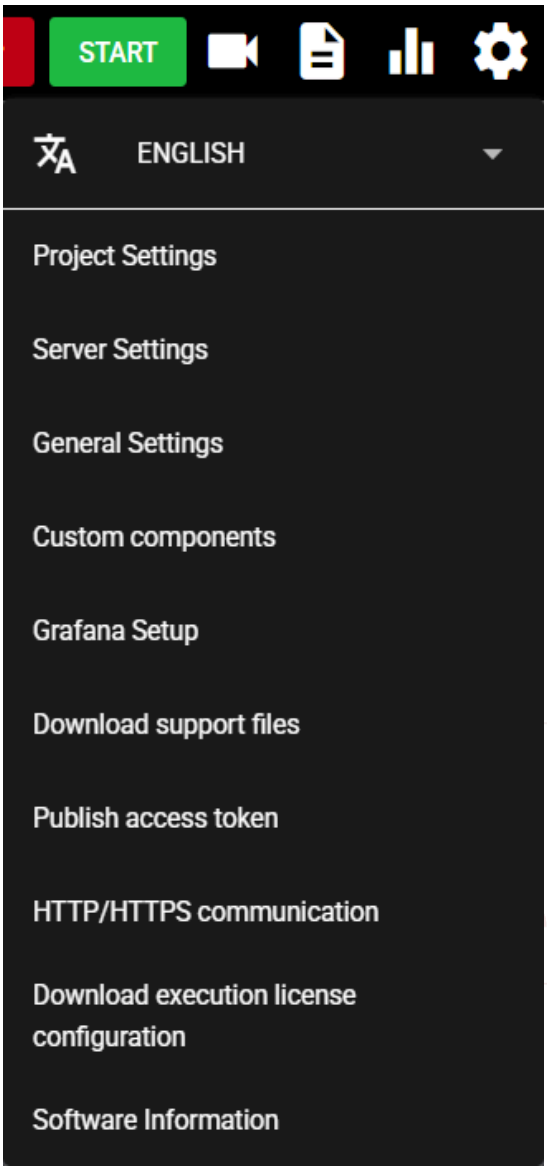
・Python 커스텀 컴포넌트의 의존 모듈은, 임포트처의 플랫폼 또는 Python 버전이 다른 경우는 인터넷 경유로 설치됩니다.
인터넷 접속되어 있지 않은 경우는, Python 모듈은 설치되지 않습니다.

・C언어 커스텀 컴포넌트는, 임포트처의 플랫폼이 다른 경우는 임포트되지 않습니다.

5.17.5 익스포트

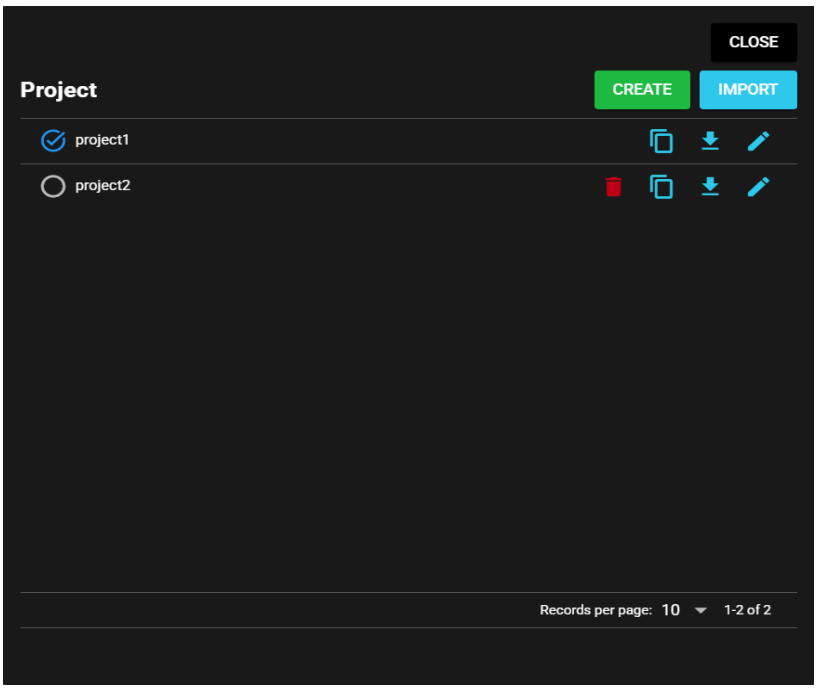
다음 순서로 프로젝트를 익스포트할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.

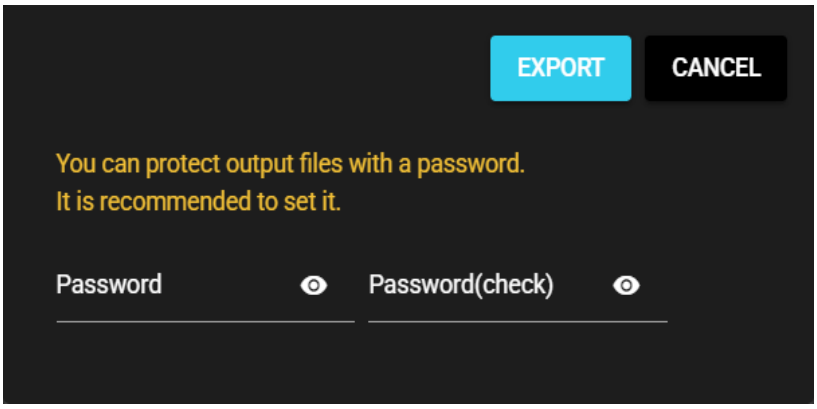


번역본
ference Only-

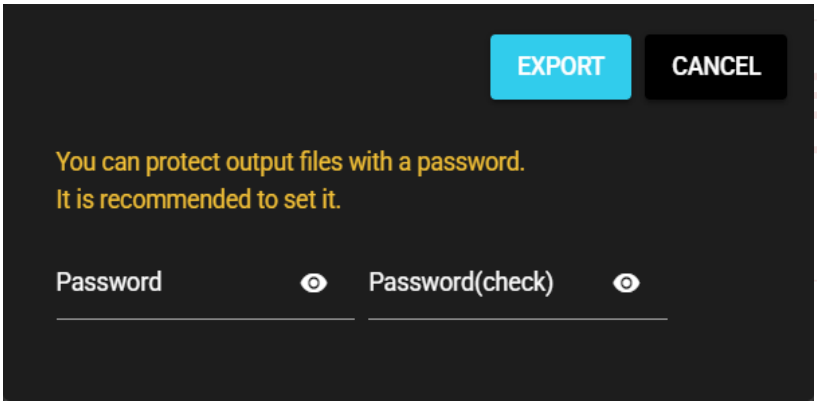
2. 익스포트할 프로젝트의「 export 아이콘（ 익스포트 ）」아이콘을 누릅니다.



3. 익스포트할 프로젝트를 패스워드로 보호하는 경우는, 임의의 패스워드를 입력합니다.
※이 순서는 생략 가능합니다.



4. 「 익스포트 」를 선택합니다.

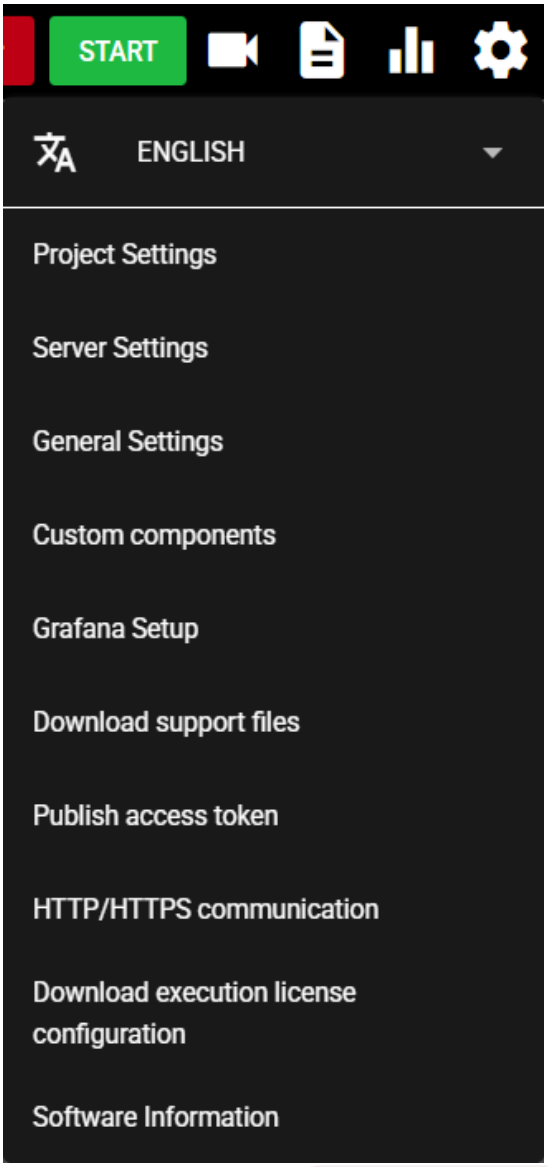


5. 프로젝트가 익스포트됩니다.

5.17.6 전환

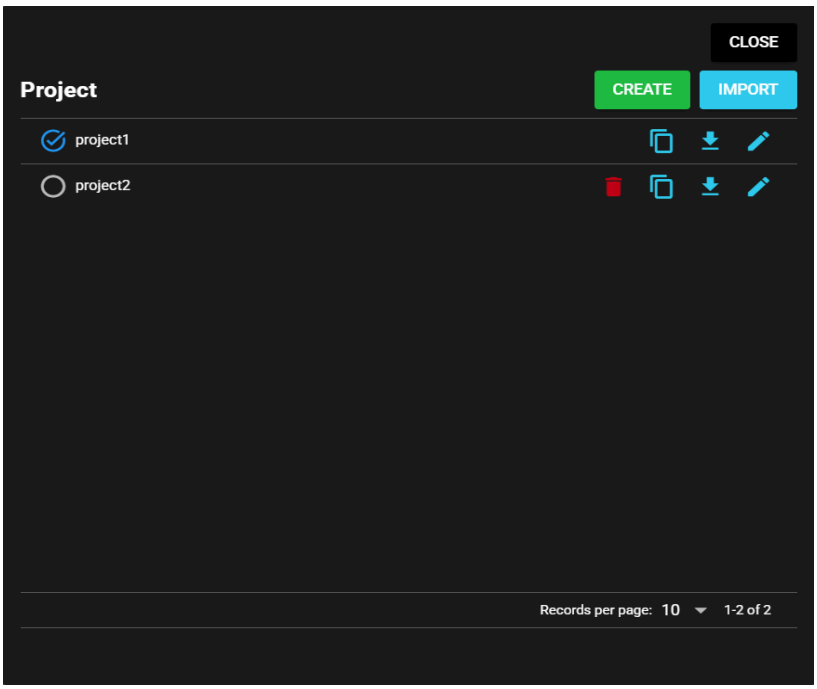
다음 순서로 프로젝트를 전환할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.



2. 전환처의 프로젝트의 라디오 버튼을 선택합니다.

AI 번역본
-For Reference Only-



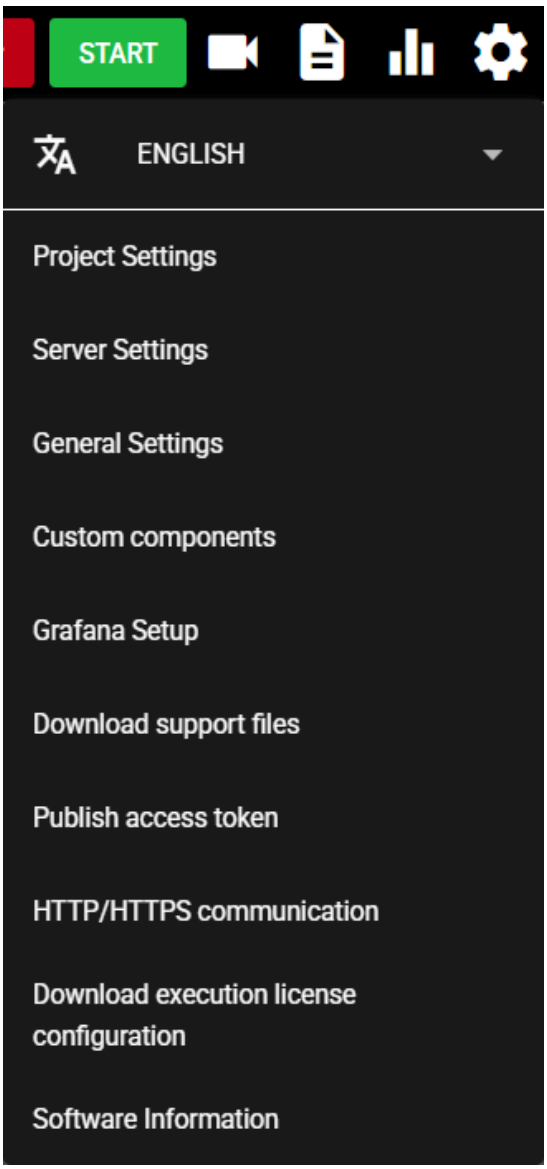
3. 확인 다이얼로그에서 「 예 」를 선택합니다.
4. 프로젝트가 전환됩니다.

5.17.7 정보 갱신

다음 순서로 프로젝트의 정보를 갱신할 수 있습니다. 갱신할 수 있는 정보는 프로젝트명뿐입니다.

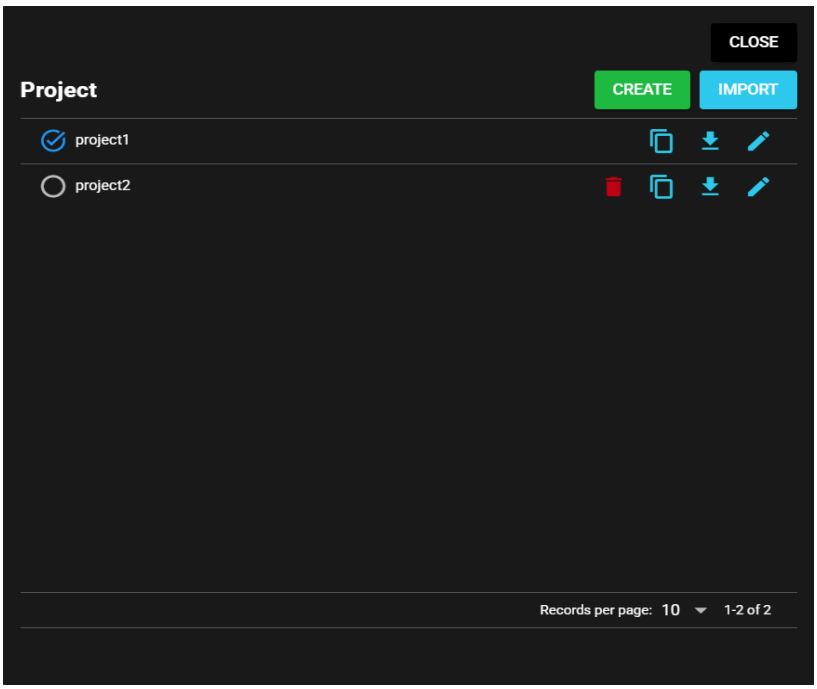
1. 설정 메뉴 아이콘을 누르고, 「 프로젝트 설정 」을 선택합니다.

AI 번역본
-For Reference Only-

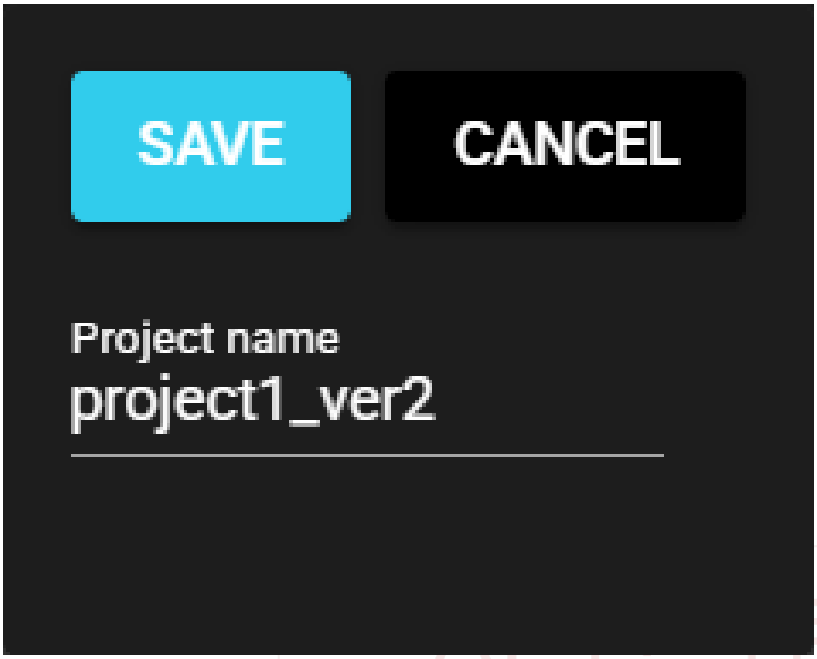


2. 정보를 갱신할 프로젝트의「에디트 아이콘（편집）」아이콘을 누릅니다.

AI 번역본
-For Reference Only-

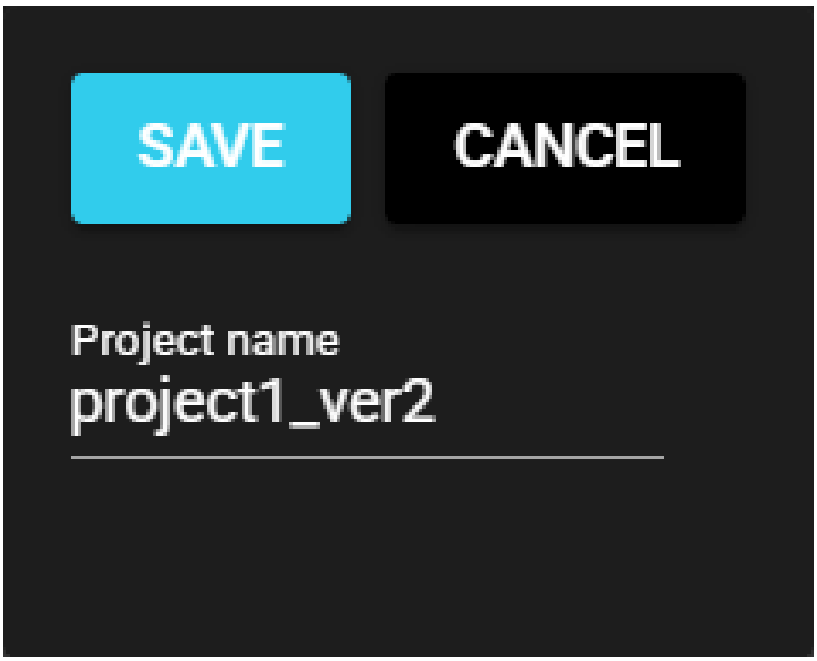


3. 새로운 프로젝트명을 입력합니다.



4. 「 저장 」을 누릅니다.

-For Reference Only-



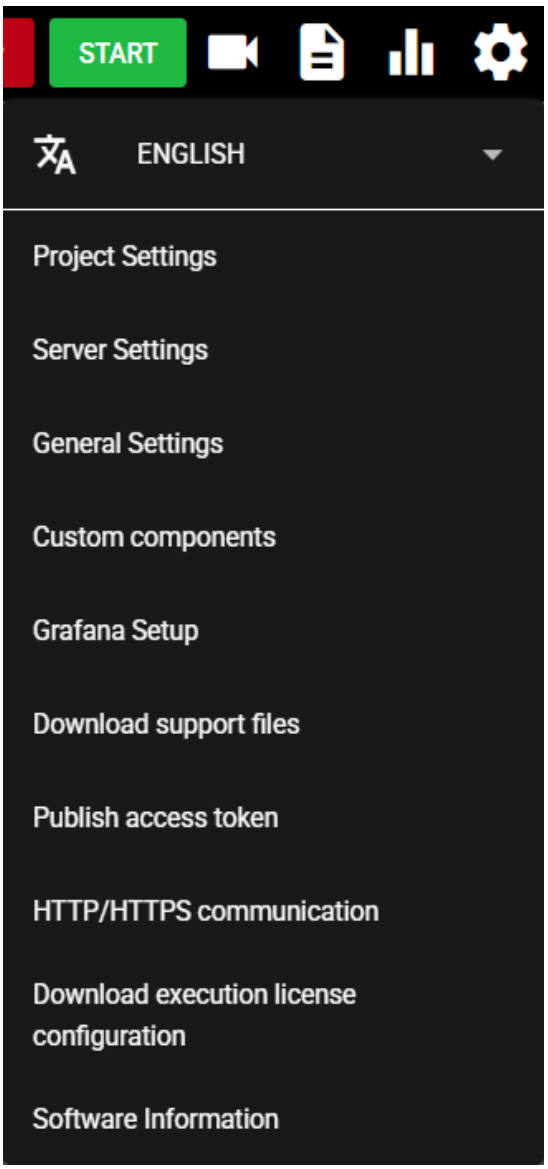
5. 프로젝트명이 갱신됩니다.

5.17.8 삭제

다음 순서로 프로젝트를 삭제할 수 있습니다.

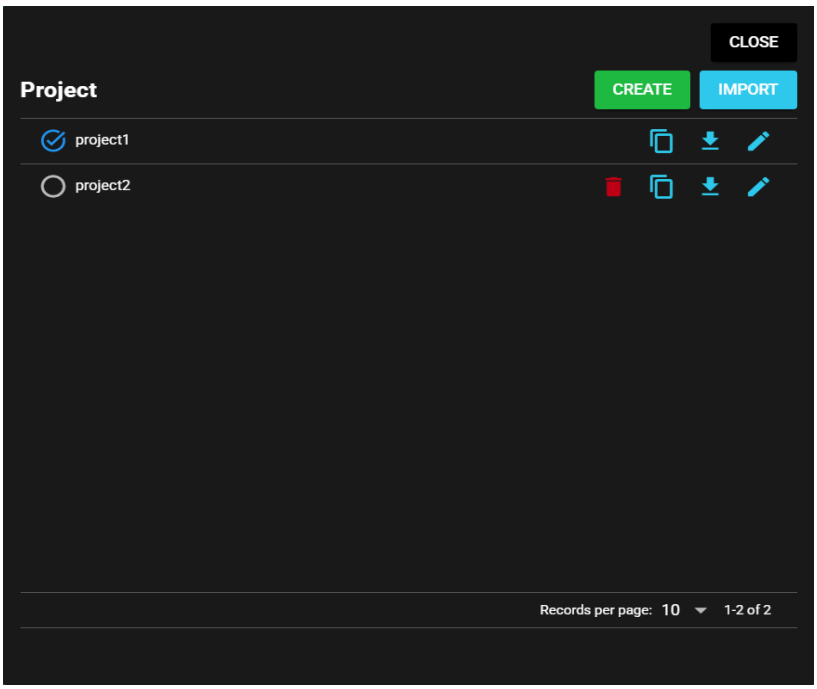
1. 설정 메뉴 아이콘을 누르고,「 프로젝트 설정 」을 선택합니다.

AI 번역본
-For Reference Only-



2. 삭제할 프로젝트의「휴지통 아이콘（삭제）」을 누릅니다.

AI 번역본
-For Reference Only-



※현재 선택 중인 프로젝트를 삭제할 수는 없습니다.

3. 확인 다이얼로그에서「예」를 선택합니다.

4. 프로젝트가 삭제됩니다.

AI 번역본
-For Reference Only-

6 응용적인 이용 방법

AI 번역본
-For Reference Only-

6.1 응용적인 이용 방법

본 장에서는, 응용적인 이용 방법에 대해 설명합니다.

여기에서 설명하는 내용은, 보다 발전적인 시스템을 구축하기 위해 도움이 되는 것입니다. 기재 내용은 다음과 같습니다.

1. 커스텀 컴포넌트의 작성・이용
2. 제어 프로그램과의 연계（WebAPI）
3. Grafana와의 연계
4. 글로벌 변수

AI 번역본
-For Reference Only-

6.2 커스텀 컴포넌트의 작성・이용

AI 번역본
-For Reference Only-

6.2.1 커스텀 컴포넌트의 작성・이용

여기에서는, 커스텀 컴포넌트의 작성・이용 방법에 대해 설명합니다.

이 커스텀 기능을 이용함으로써, 독자적인 기능을 SpeeDBee Synapse에 추가할 수 있게 됩니다. 기재 내용은 다음과 같습니다.

1. 커스텀 컴포넌트의 개요
2. Python 커스텀 컴포넌트 작성
3. C언어 커스텀 컴포넌트 작성
4. 커스텀 UI 작성
5. 커스텀 컴포넌트 개발 환경(SCCDE)

AI 번역본
-For Reference Only-

6.2.2 커스텀 컴포넌트의 개요

6.2.2.1 커스텀 컴포넌트란

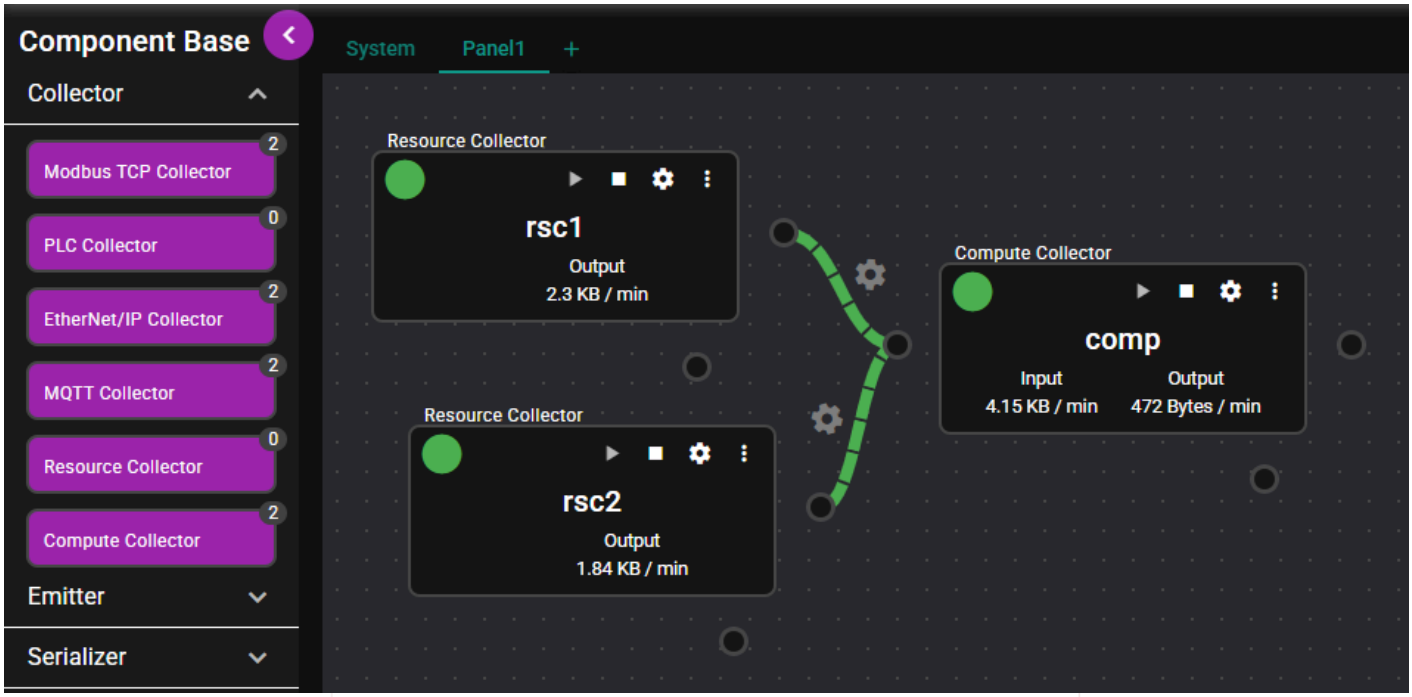
SpeeDBee Synapse에서는, 데이터의 수집이나 송신을 담당하는 모듈을「컴포넌트」라고 부릅니다.

표준으로는 PLC 등의 데이터를 수집하는 컴포넌트나, 클라우드에 데이터를 송신하는 컴포넌트가 준비되어 있습니다만, 사용자가 독자적으로 컴포넌트를 개발, 운용하는 것도 가능합니다.

컴포넌트의 개발에는 C, 혹은, Python을 사용할 수 있습니다만, 본 장에서는 언어에 관계없이, 커스텀 컴포넌트의 개발에 공통으로 필요한 기본 개념을 설명합니다.

6.2.2.2 컴포넌트 베이스와 컴포넌트 인스턴스

통상은 한데 묶어「컴포넌트」라고 호칭합니다만, 커스텀 컴포넌트를 실장하는 데 있어서는,「컴포넌트 베이스」와「컴포넌트 인스턴스」의 차이를 이해할 필요가 있습니다.



For Reference Only

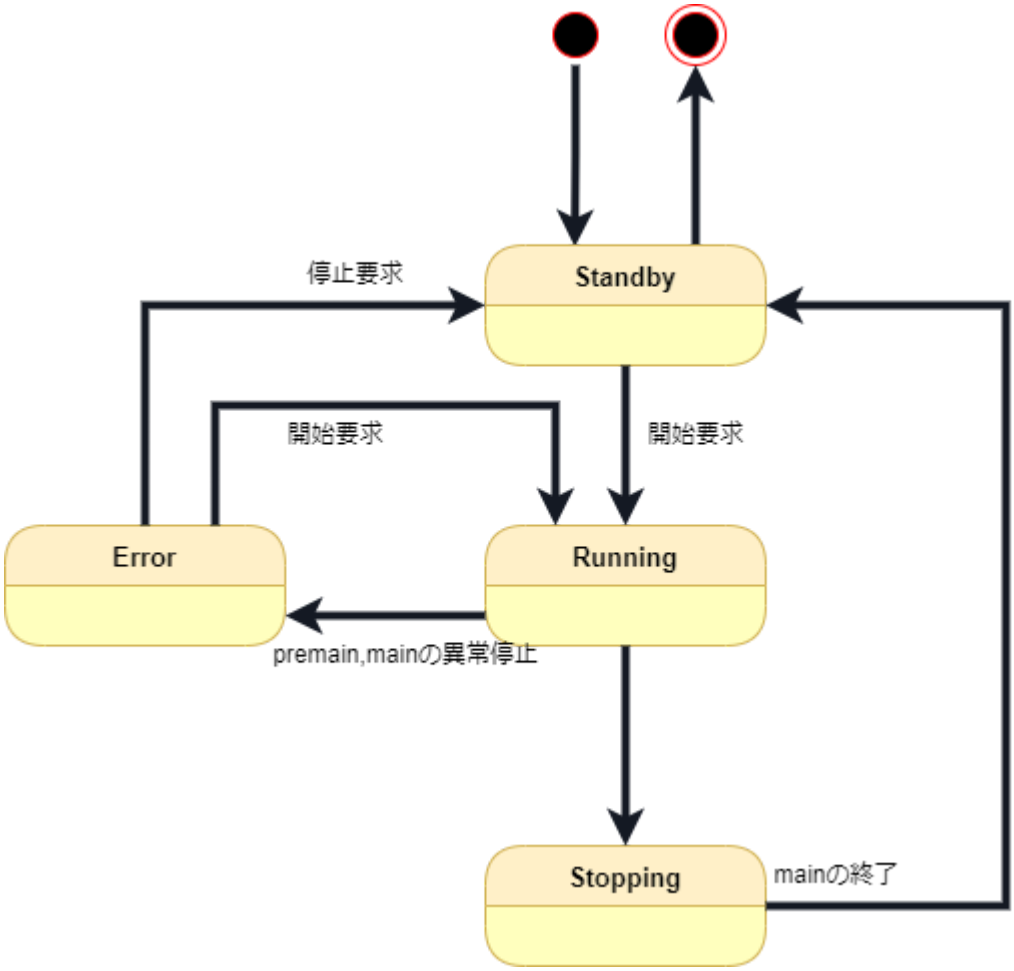
좌측에 1 열로 나열되어 있는 항목이,「컴포넌트 베이스」, 중앙 ~ 우측에 있는 3 개의 요소가「컴포넌트 인스턴스」입니다.

「컴포넌트 인스턴스」는, 어느 하나의「컴포넌트 베이스」를 드래그 앤 드롭함으로써 작성됩니다. 특수한 것을 제외하고, 1 개의「컴포넌트 베이스」에서 복수의「컴포넌트 인스턴스」를 작성할 수 있습니다.

커스텀 컴포넌트를 실장하는 경우도 이것은 마찬가지가 됩니다. 1 개의 커스텀 컴포넌트를 작성하면, 화면의 좌측의 열에 새로운「컴포넌트 베이스」가 1 개 추가됩니다. 이 추가된 컴포넌트 베이스를 드래그 앤 드롭함으로써,「컴포넌트 인스턴스」를 복수 작성할 수 있습니다.

6.2.2.3 컴포넌트 인스턴스의 라이프사이클

모든 컴포넌트 인스턴스는, 작성되고 나서 삭제될 때까지 아래 그림의 라이프사이클로 동작합니다.



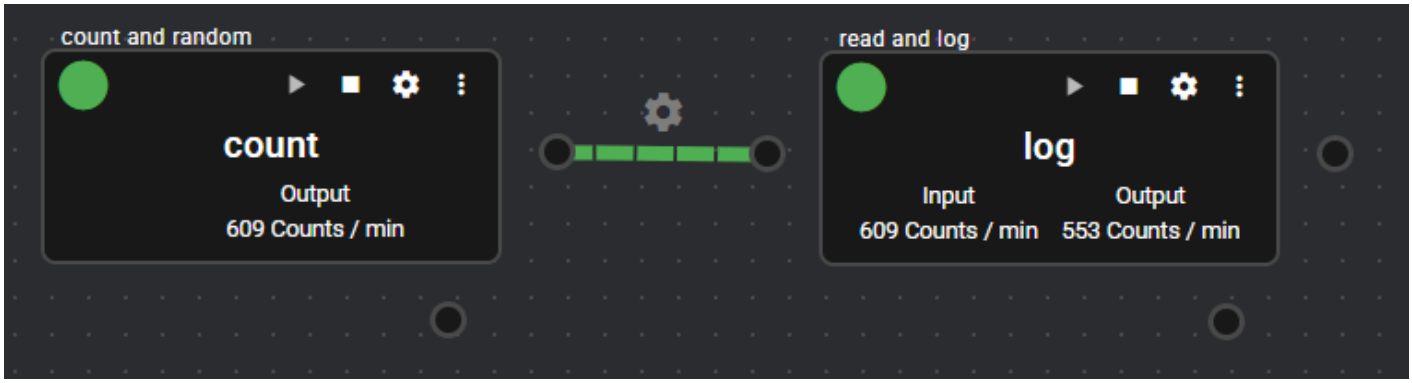
- 컴포넌트 인스턴스의 작성 직후는 Standby 상태입니다
  이 상태에서는 컴포넌트는 아무것도 동작하고 있지 않습니다. 아무 데이터도 출력하지 않고, 아무 데이터도 입력되지 않습니다.
- 컴포넌트 인스턴스의 기동 아이콘을 클릭하면, Running 상태로 천이합니다
  이 상태에 있으면, 컴포넌트 인스턴스는 계속적으로 데이터를 출력하거나, 다른 컴포넌트로부터 데이터를 받아들이거나 할 수 있습니다.
- 컴포넌트 인스턴스의 정지 아이콘을 클릭하면, Stopping 상태가 되고, 그 후 모든 처리를 완료하면 Standby 상태로 돌아갑니다.
- Running 도중에 컴포넌트 인스턴스의 처리를 계속할 수 없는 에러 등이 발생한 경우는, Error 상태로 이행하고, 처리를 정지합니다

AI 번역본
-For Reference Only-

커스텀 컴포넌트의 개발에서는, 이 라이프사이클 속에서, 다양한 처리를 프로그래밍할 수 있습니다. 이후의 장에서는 프로그래밍 언어별 실장 방법을 설명합니다.

6.2.2.4 컴포넌트의 입력 포트와 출력 포트

각 컴포넌트는, 출력 포트와 입력 포트를 갖추고 있습니다. 출력 포트는, 그 컴포넌트가 생성하거나, 외부에 액세스하거나 하여 수집한 데이터를 SpeeDBee Synapse에 출력하기 위한 요소. 입력 포트는, 다른 컴포넌트가 출력한 데이터를 받아들여, 그 컴포넌트에서 이용하기 위한 요소입니다.



커스텀 컴포넌트의 실장에 있어서는, 각각의 포트 수를 지정할 필요가 있습니다. 데이터를 출력하는 컴포넌트의 경우는 출력 포트 수를 1로, 데이터를 받아들이는 컴포넌트의 경우는 입력 포트 수를 1로, 각각의 포트가 필요 없는 경우는 0으로 하십시오. 장래적으로는 복수의 포트를 다룰 수 있도록 할 가능성이 있습니다만, 현시점에서는 각 포트는 최대 1 개로 제한되어 있습니다.

AI 번역본
-For Reference Only-

6.2.3 Python 커스텀 컴포넌트 작성

본 항에서는, Python에 의한 커스텀 컴포넌트의 작성 방법에 대해 설명합니다.

6.2.3.1 전제 지식

커스텀 컴포넌트를 개발하는 데 있어서의 전제 지식으로서, Python 언어의 튜토리얼의 내용은 대체로 파악할 수 있을 것, 아울러 클래스 정의, 메서드 정의, 클래스의 상속 등에 대해 이해하고 있을 것을 상정하고 있습니다. 입문
레벨의 정보에 대해서는, 아래의 Python 공식 도큐먼트를 참조하십시오.

Python 공식 도큐먼트： https://docs.python.org/ja/3/

6.2.3.2 샘플 커스텀 컴포넌트

Python에 의한 커스텀 컴포넌트의 실장 방법 설명을 위해, 심플한 사양의 커스텀 컴포넌트를 샘플로서 제공하고 있습니다.

샘플 커스텀 컴포넌트의 다운로드

혹은, SpeeDBee Synapse를 설치 완료한 경우, 커스텀 컴포넌트 샘플도 아래의 장소에 설치되어 있습니다.

```
$ ls /usr/local/speedbeesynapse/share/sample_component_py/
sin_and_random.py  read_and_log.py
```

6.2.3.2.1 샘플의 내용

커스텀 컴포넌트의 샘플로서 본 장에서는 아래의 샘플을 설명합니다.

- `sin_and_random.py`

  출력 포트에 대해 이하의 2 개의 컬럼을 작성하고, 데이터를 계속 출력하는 컴포넌트입니다.

컬럼명	데이터형	내용
sin	DOUBLE	1Hz의 SIN파의 값을 초당 10건 등록합니다. SIN파의 주기는 파라미터로 변경 가능합니다
rand	INT32	1 초에 1 회, [0, 100)의 범위 내의 어느 하나의 값을 랜덤하게 등록합니다

- `read_and_log.py`

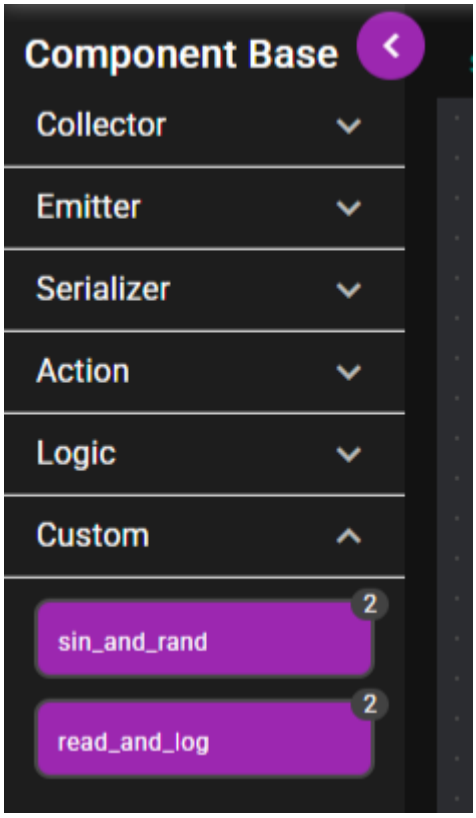
  입력 포트에서 데이터를 받아들여, 그것을 문자열형 컬럼과 로그에 출력합니다.

AI 번역본

6.2.3.2.2 실행

`*.py` 파일을 그대로 화면에서 등록하면, 에러 등이 없으면 컴포넌트 베이스가 추가된 상태로 됩니다. 등록 방법은 「 커스텀 컴포넌트를 사용한다 」를 참조하십시오.

등록 후에는 일반 컴포넌트와 마찬가지로 D&D로 컴포넌트 인스턴스를 생성할 수 있습니다.



`sin_and_random` 및 `read_and_log` 를 연결하여 실행하면, 각각이 출력한 컬럼을 표시할 수 있습니다. 아래는 `sin_and_random` 의 출력 포트를 모니터링하고 있습니다. 값의 변화를 확인하려면 「 솎아내기 간격 」을 100밀리초로 변경하십시오.

AI 번역본
-For Reference Only-



6.2.3.3 커스텀 컴포넌트 실장 상세

이후에서는 샘플인 sin_and_random 및 read_and_log 의 소스 코드를 예로, 커스텀 컴포넌트의 실장 내용을 설명해 나갑니다.

6.2.3.3.1 커스텀 컴포넌트 클래스 정의

Python에 의한 커스텀 컴포넌트는 아래와 같이 HiveComponentBase 클래스를 상속한 새 클래스를 정의하면 그 컴포넌트 베이스의 이름이나 식별 정보, 실행 시의 처리를 정의할 수 있습니다.

```
from speedbeesynapse.component.base import HiveComponentBase, HiveComponentInfo, DataType
  :

@HiveComponentInfo(uuid='00000000-0000-0000-0000-000000000000', name='컴포넌트명', inports=0,
outports=1)
class HiveComponent(HiveComponentBase):
    def main(self, param):
        :
```

For Reference Only

클래스명은 반드시 HiveComponent 로 하십시오. 또한 컴포넌트의 정보를 @HiveComponentInfo 데코레이터로 래핑해야 합니다. 여기서 정의한 정보에 의해 화면에서도 컴포넌트 베이스의 정보를 다룰 수 있게 됩니다. 아래 설명을 참고하여 설정하십시오.

멤버명	정의 정보	설명
uuid	UUID	컴포넌트를 식별하기 위한 UUID입니다. 무작위 ID를 생성하여 여기에 설정하십시오.

멤버명	정의 정보	설명
`name`	컴포넌트명	컴포넌트의 이름을 설정합니다. 화면에서는 이 이름이 표시되므로 되도록 다른 컴포넌트와 겹치지 않고 알기 쉬운 이름으로 하십시오.
`tag`	태그	컴포넌트에 태그를 부여합니다. "collector","emitter","serializer","action","logic" 중 하나를 지정하면 컴포넌트 베이스의 배치처를 지정할 수 있습니다. 생략한 경우에는 Custom에 배치됩니다.
`outports`	출력 포트 수	이 컴포넌트가 갖는 출력 포트의 수를 설정합니다. 다른 컴포넌트의 입력 포트에 데이터를 전달하는 경우에는 이것을 1로 하십시오.
`inports`	입력 포트 수	이 컴포넌트가 갖는 입력 포트의 수를 설정합니다. 다른 컴포넌트의 출력 포트로부터 데이터를 받는 경우에는 이것을 1로 하십시오.

UUID에 대해서는 반드시 무작위 ID를 생성하십시오. 다른 컴포넌트와 같은 UUID로 해 버리면 어느 컴포넌트도 사용할 수 없게 되는 경우가 있습니다.

6.2.3.3.2 HiveComponent 클래스의 라이프사이클

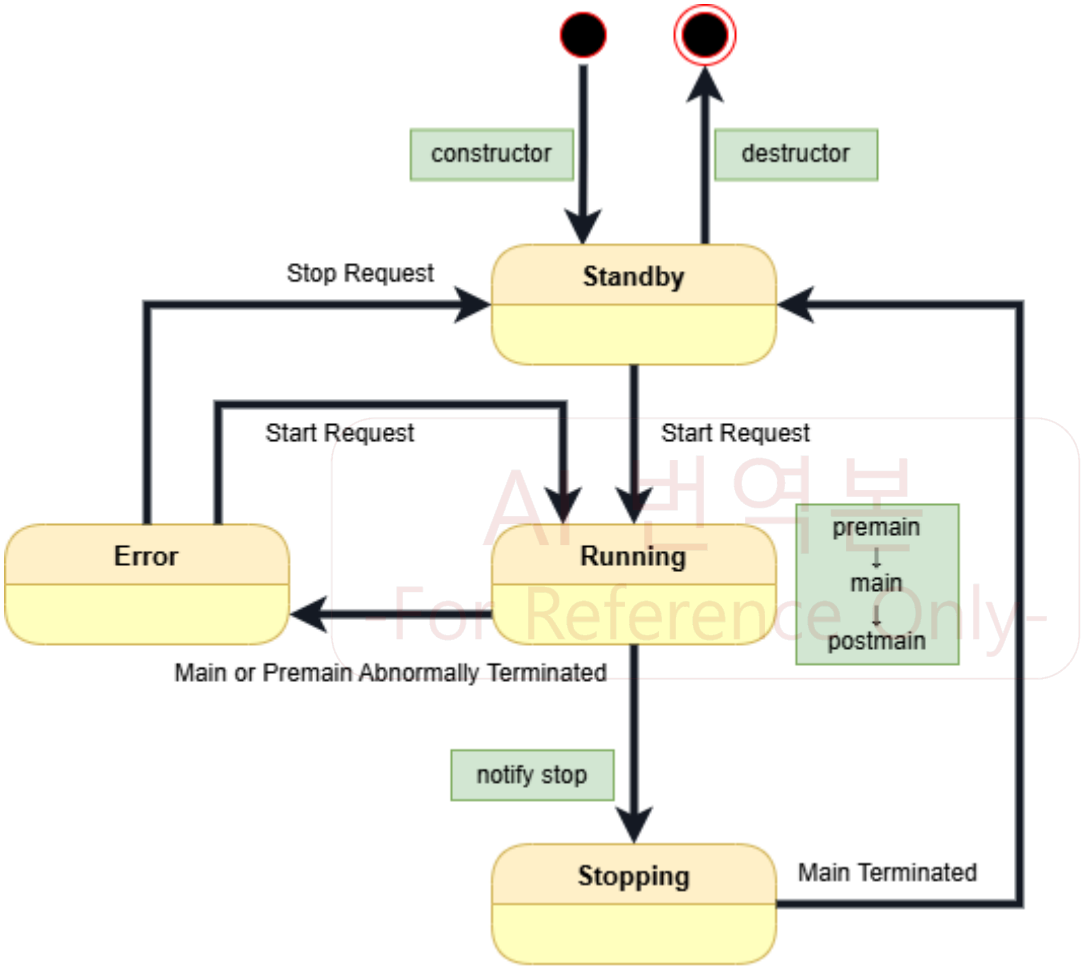
여기서 정의한 HiveComponent 클래스가 하나의 컴포넌트 베이스가 됩니다. Python 스크립트 파일 내에서 이 클래스를 인스턴스화하는 처리는 필요하지 않습니다. 대신 화면상에서 컴포넌트 베이스를 D&D하여 컴포넌트 인스턴스를 생성하면 자동으로 여기서 정의한 HiveComponent 클래스의 인스턴스가 생성됩니다. 또한 화면상에서 컴포넌트 인스턴스를 삭제하면 이 클래스의 인스턴스가 파기됩니다.

따라서 HiveComponent 클래스 내에 __init__() 메서드나 __del__() 메서드를 정의해 두면 화면상에서의 컴포넌트 인스턴스의 생성・삭제에 맞춰 그 메서드가 실행되게 됩니다.

AI 번역본
-For Reference Only-

```
@HiveComponentInfo(uuid='00000000-0000-0000-0000-000000000000', name='サンプルコンポ', inports=0,
outports=0)
class HiveComponent(HiveComponentBase):
    def __init__(self):
        # called at creating component instance
        pass
    def __del__(self):
        # called at deleting component instance
        pass
    def premain(self, param):
        # called before main()
        pass
    def postmain(self, param):
        # called after main() finished
        pass
    def main(self, param):
        # called at starting component instance
        while self.is_runnable():
          pass
    def notify_stop(self):
        # called before the component received stop request from system
        pass
```

그 밖에도 화면에서의 컴포넌트 조작에 의해 호출되는 메서드가 있습니다. 호출되는 타이밍과
그 메서드에 대해서는 아래 그림, 아래 표를 참조하십시오.



콜백 함수	설명
`__init__`	컴포넌트의 인스턴스가 생성되었을 때 호출됩니다
`__del__`	컴포넌트의 인스턴스가 파기될 때 호출됩니다
`premain`	컴포넌트를 개시할 때 main보다 앞서 호출됩니다
`main`	컴포넌트의 메인 처리를 실장하는 함수입니다 컴포넌트의 종료 요청을 받을 때까지 이 함수는 종료하지 않고 루프 처리를 계속할 필요가 있습니다
`postmain`	컴포넌트 종료 시, main 종료 직후에 호출됩니다
`notify_stop`	main 실행 중, 컴포넌트의 정지 요청을 받았을 때 실행됩니다 이 메서드는 main() 실행 중에 호출되므로 별도 스레드에서 실행되는 것에 주의하십시오.

`main` 은 컴포넌트 처리의 본체가 되므로 반드시 정의해야 합니다. 이후에서는 샘플 프로그램의 `main` 함수 실장 방법을 설명합니다.

6.2.3.3.3 출력 포트의 컬럼 작성/데이터 등록(sin_and_random)

아래는 커스텀 컴포넌트의 샘플 sin_and_random 의 메인 처리입니다. 이를 바탕으로 데이터를 출력하는 흐름을 설명합니다.

```
def main(self, param):
     :
    count = 0

    # Create columns.
    self.sin = self.out_port1.Column('sin', DataType.DOUBLE)
    self.rand = self.out_port1.Column('rand', DataType.INT32)

    # Repeat at 100msec intervals.
    for [ts, skip] in self.interval_iteration(100000000):

        # Compute sin wave value and insert it into 'sin' column.
        val = math.sin(2*math.pi*sin_hz*count/10)
        self.sin.insert(val, ts)

        # Insert random value into 'rand' column only once in 10.
        if count % 10 == 0:
            self.rand.insert(random.randrange(0, 100), ts)

        count += 1
```

AI 번역본
-For Reference Only-

1. 컬럼 작성

먼저 컬럼을 작성합니다.

```
self.sin = self.out_port1.Column('sin', DataType.DOUBLE)
self.rand = self.out_port1.Column('rand', DataType.INT32)
```

이 컴포넌트에서는 배정밀도 부동소수점수의 컬럼 sin 과 32bit 정수의 컬럼 rand 를 이용하므로,
self.out_port1 에 있는 Column() 메서드에서 컬럼명, 데이터형 등을 지정하여 컬럼 오브젝트를 생성
하고 있습니다. 반환값은 나중에 컬럼에 데이터를 등록할 때 사용하므로 변수에 격납해 둡니다.

컴포넌트가 출력하는 모든 데이터는 이렇게 생성한 컬럼에 대해 등록해 나가야 합니다.

2. 정기적인 처리 반복

정기적으로 데이터를 등록하기 위해 for문으로 반복 처리를 실장하고 있습니다.

```
for [ts, skip] in self.interval_iteration(100000000):
    :
```

루프의 판정 조건에서 호출하고 있는 self.interval_iteration() 은 인수에 지정한 시간 간격으로 반복 값을
반환하는 제너레이터로 되어 있습니다. 이에 따라 이 for문의 내부는 일정 간격으로 실행되게 됩니다. 인수에 지정하는 것은

나노초 단위이므로, 이 예의 경우는 0.1초(100000000나노초)에 1회, 루프 처리가
실행되게 됩니다.

화면에서 컴포넌트를 정지하지 않는 한, 이 for문의 루프는 끝나지 않습니다.

3. 컬럼으로의 데이터 등록

반복 처리 안에서 컬럼에 대해 데이터를 등록합니다.

```
val = math.sin(2*math.pi*sin_hz*count/10)
self.sin.insert(val, ts)

if count % 10 == 0:
    self.rand.insert(random.randrange(0, 100), ts)
```

데이터 등록은 사전에 작성한 컬럼 오브젝트의 insert() 메서드 호출로 할 수 있습니다. 제1 인수에는 등록할 값을, 제2 인수에는 그 값을 등록할 시각을 지정하십시오. 제2 인수는 생략하면 현재 시각으로 취급됩니다.

AI 번역본
-For Reference Only-

> insert로 등록할 수 있는 최대 사이즈
>
> insert()로 등록할 수 있는 데이터의 최대 사이즈는 512바이트(STRING은 NULL을 포함하지 않는 511바이트)입니다. 상한 사이즈를 초과한 값의 등록은 예외가 스로우됩니다.

6.2.3.3.4 입력 포트로부터의 데이터 불러오기 (read_and_log)

아래는 커스텀 컴포넌트의 샘플 read_and_log 의 메인 처리입니다. 이를 바탕으로 다른 컴포넌트가 출력한 데이터를 받는 처리 흐름을 설명합니다.

```
def main(self, param):
    # Create a column for logging received data.
    self.logclm = self.out_port1.Column('log', DataType.STRING)

    # Logging column information received from 'in_port1'.
    columns = self.in_port1.get_columns()
    for column in columns:
        self.log.info(column)

    # Start receving column values from 'in_port1'.
    with self.in_port1.ContinuousReader(start=self.get_timestamp()) as reader:
        # Loop until the component received 'stop' request.
        while self.is_runnable():
            # Get a group of records of column data (called as window_data).
            window_data = reader.read()
            if not window_data:
                # window_data may be None.
                self.log.info("no data yet")
                continue

            # Process all records in a window_data one by one.
            for record in window_data.records:
                # Convert timestamp as a string.
                ts = datetime.datetime.fromtimestamp(record.timestamp / 1000000000)

                # Process all column's value in a record one by one.
                for columnvalue in record.data:
                    if columnvalue.value:
                        # Make string from columnvalue, logging it and insert it into the column
                        logmsg = f"{columnvalue.column}[{str(ts)}]: {columnvalue.value}"
                        self.log.info(logmsg)
                        self.logclm.insert(logmsg)
                    else:
                        pass # ignore None
```

맨 앞에서 작성하고 있는 log 컬럼은 문자열로 정보를 출력하기 위한 것입니다. 다른 컴포넌트로부터 데이터를 받기만 한다면 필수가 아니므로 생략합니다.

AI 번역본
-For Reference Only-

1. 입력 포트의 컬럼 정보 취득

   in_port1 에 있는 get_columns() 메서드에 의해 그 컴포넌트의 입력 포트에 접속된 컴포넌트가 출력하고 있는 데이터의 컬럼 정보를 취득할 수 있습니다.

   ```
   columns = self.in_port1.get_columns()
   for column in columns:
       self.log.info(column)
   ```

   위 예에서는 취득한 컬럼을 이터레이션 루프로 로그에 출력하고 있습니다.

   컴포넌트로의 접속을 바꾸거나 접속된 컴포넌트를 정지하거나 하면, 이 get_columns() 로 취득할 수 있는 컬럼 정보가 변화합니다. 항상 최신 정보가 필요한 경우에는 정기적으로 취득하여 내용을 체크하도록 하십시오.

2. ContinuousReader 의 개시와 루프

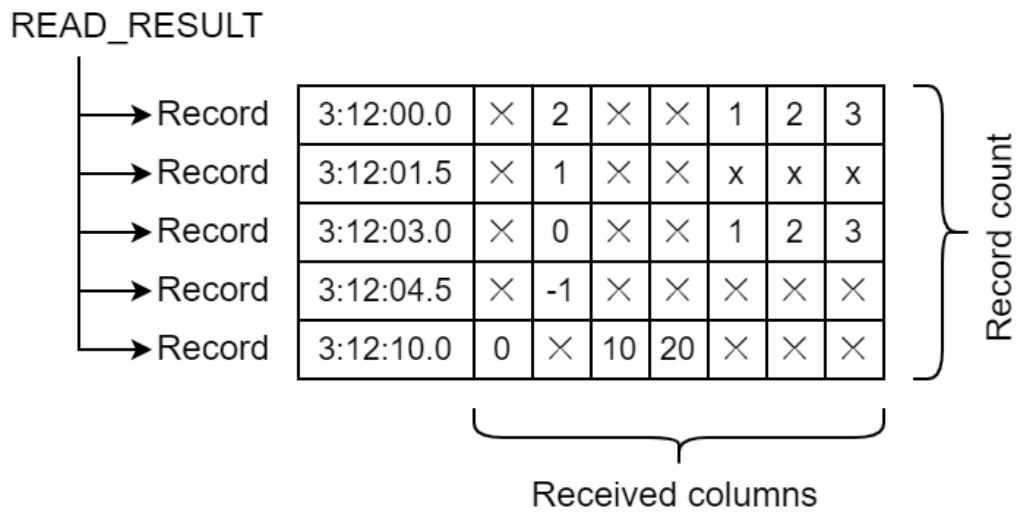
입력 포트에서 계속적으로 데이터를 계속 취득하는 경우, ContinuousReader 오브젝트를 이용합니다. 아래의 와 같이 with구문으로 사용하고, 또한 반복하여 read() 를 호출함으로써 입력 포트에 도착한 데이터를 취득할 수 있습니다.

```
with self.in_port1.ContinuousReader(start=self.get_timestamp()) as reader:
    while self.is_runnable():
        window_data = reader.read()
        if not window_data:
            self.log.info("no data yet")
            continue
        :
```

read() 의 반환값은 None이 되는 경우도 있으므로 그 경우에는 건너뛰십시오

3. 취득한 데이터의 레코드 참조

ContinuousReader 의 read() 메서드는 HiveWindowData 오브젝트를 반환합니다. 이 오브젝트는,
입력 포트에 들어온 데이터를 어느 시간 폭만큼 모아서 유지한 것으로, 아래 그림과 같이 동일 시각의 데이터를 모은 복수의 레코드로 이루어지며, 하나의 레코드에 복수의 컬럼 데이터가 담겨 있습니다.



따라서 아래와 같이 이중 for 루프문으로 모든 데이터에 액세스할 수 있습니다.

```
for record in window_data.records:
    ts = datetime.datetime.fromtimestamp(record.timestamp / 1000000000)

    for columnvalue in record.data:
        if columnvalue.value:
            logmsg = f"{columnvalue.column}[{str(ts)}]: {columnvalue.value}"
            self.log.info(logmsg)
            self.logclm.insert(logmsg)
        else:
            pass # ignore None
```

각 데이터의 상세는 HiveRecordData 오브젝트를 참조하십시오.

6.2.3.3.5 로그 출력

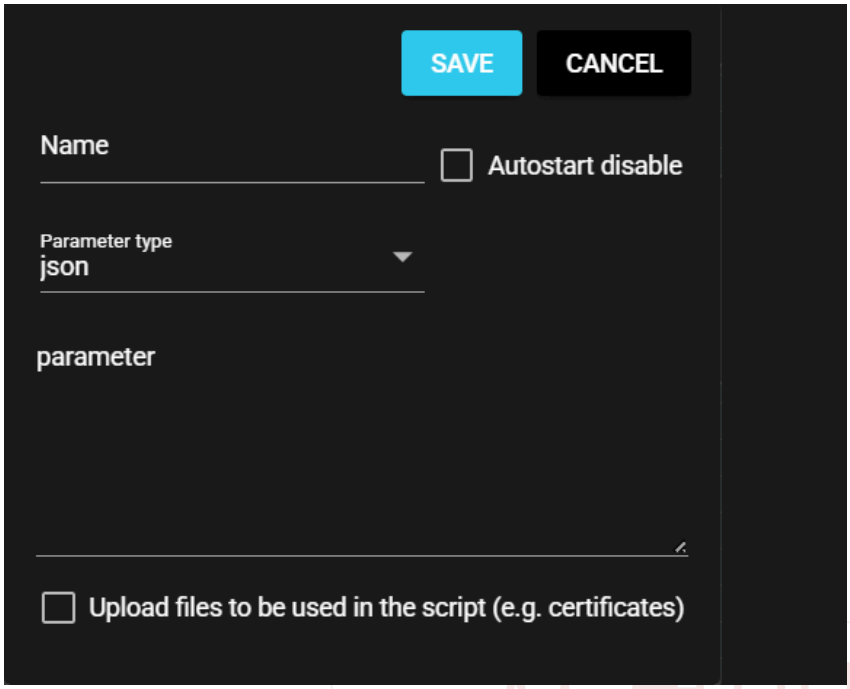
커스텀 컴포넌트에서는 아래와 같이 self.log 의 메서드를 이용함으로써 임의의 타이밍에 로그를 출력할 수 있습니다. 상세는 HiveLog 를 참조하십시오.

```
self.log.info('create columns')
self.log.info(f'insert value={val} into {column_name}')
self.log.error('invalid data received')
```

출력한 로그는 화면에서 다운로드할 수 있습니다.「각종 로그」를 참조하십시오.

6.2.3.3.6 컴포넌트 파라미터

화면상에서 컴포넌트 인스턴스를 생성하면 그 설정 항목으로서 컴포넌트의 실행 시 파라미터를 설정할 수 있습니다. 인스턴스마다 파라미터 설정을 바꿈으로써 같은 컴포넌트 베이스의 실장
이라도 컴포넌트의 동작을 바꿀 수 있습니다.



화면에서 지정할 때, 파라미터의 타입은 STRING 또는 JSON을 선택할 수 있습니다. 실장 시 다루기 쉬운 타입 을 선택하십시오.

AI 번역본
-For Reference Only-

파라미터 타입	설명
STRING	임의의 문자열을 설정할 수 있습니다. 컴포넌트의 각 메서드의 제2 인수에서는 str 형으로 참조할 수 있습니다.
JSON	JSON 문자열을 설정할 수 있습니다. 컴포넌트의 각 메서드의 제2 인수에서는 dict 형으로 참조할 수 있습니다.

컴포넌트 인스턴스에 설정한 파라미터는 정의한 HiveComponent 클래스의 premain() , main() , postmain() 메서드의 제2 인수로서 취득할 수 있습니다.

```
@HiveComponentInfo(uuid='00000000-0000-0000-0000-000000000000', name='サンプルコンポ', inports=0,
outports=0)
class HiveComponent(HiveComponentBase):
    def premain(self, param):
        // use 'param' here.
        pass
    def postmain(self, param):
        // use 'param' here.
        pass
    def main(self, param):
        // use 'param' here.
        while ...
```

이들의 취급은 커스텀 컴포넌트의 실장자에 따라 다릅니다. 예로 sin_and_random 에서는 파라미터에 수치 문자열을 sin 컬럼의 파형 주파수로 이용하고 있으므로 참고하십시오.

6.2.3.4 커스텀 컴포넌트용 API

본 항에서 소개한 함수 외에도 다수의 API 함수가 준비되어 있습니다. 상세는 부록의 「커스텀 컴포넌트 Python-API 레퍼런스」를 참조하십시오.

6.2.3.5 커스텀 컴포넌트를 사용한다

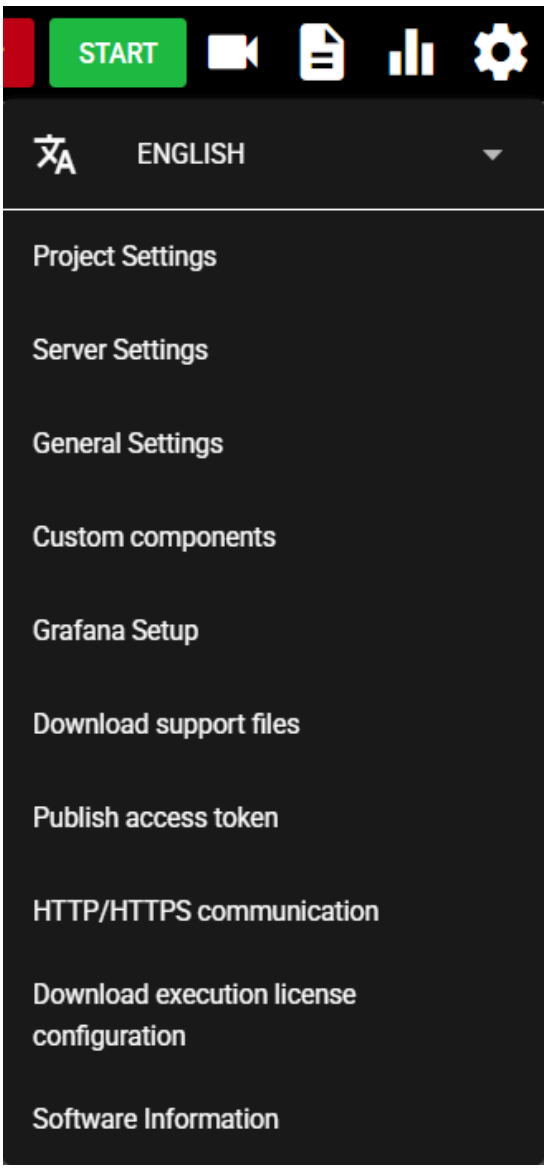
여기서는 작성한 커스텀 컴포넌트를 SpeeDBee Synapse에서 사용하는 방법에 대해 설명합니다.

6.2.3.5.1 등록

다음 순서로 작성한 커스텀 컴포넌트를 SpeeDBee Synapse에 등록할 수 있습니다.

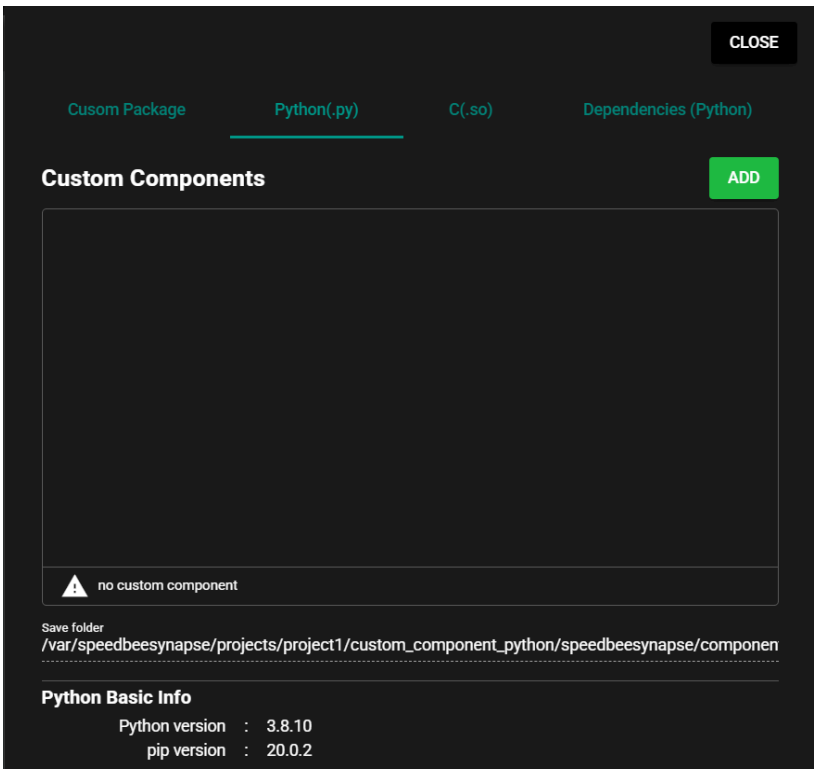
1. 설정 메뉴 아이콘을 누르고, 「커스텀 컴포넌트」를 선택합니다.

AI 번역본
-For Reference Only-

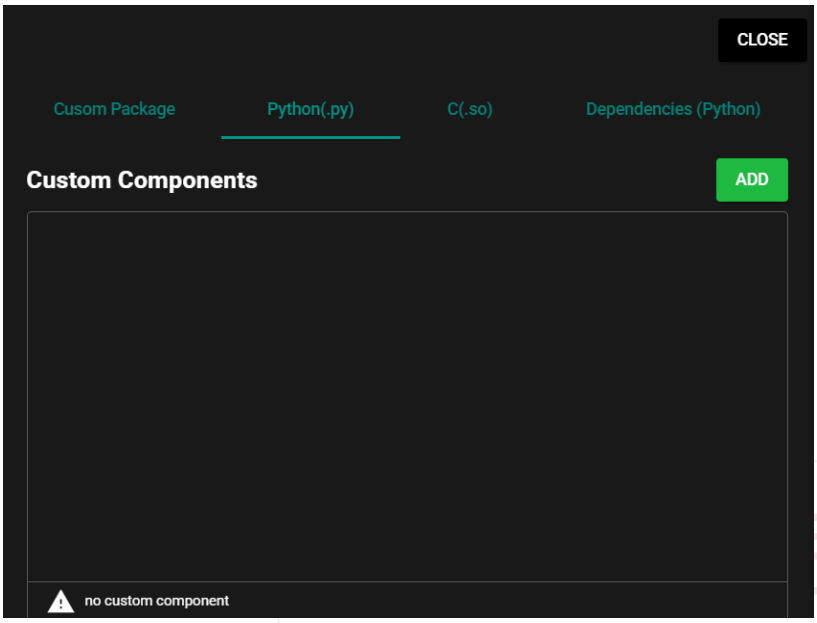


2. 다이얼로그 상단 탭의「 Python(.py) 」를 선택합니다.

AI 번역본
-For Reference Only-

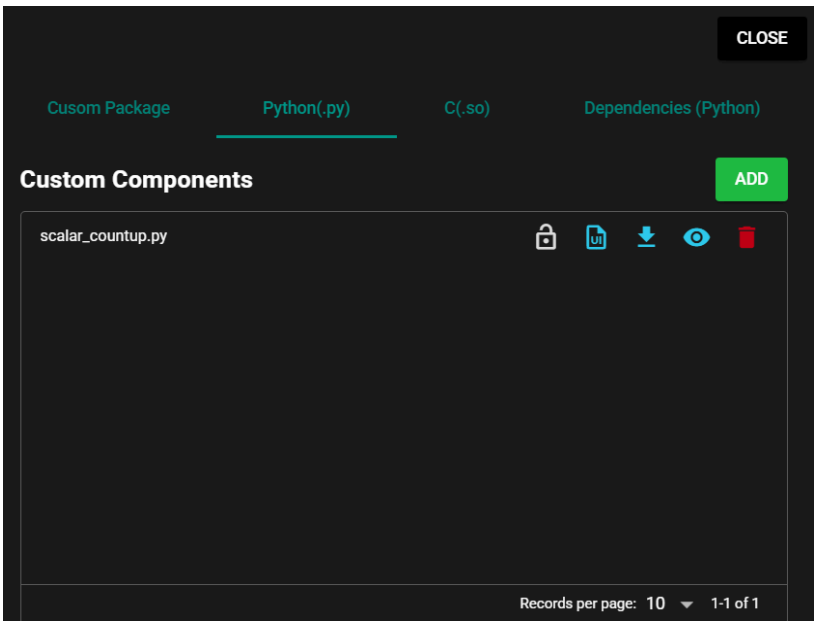


3. 커스텀 컴포넌트의「 추가 」를 누르고, 커스텀 컴포넌트의 파일을 등록합니다.

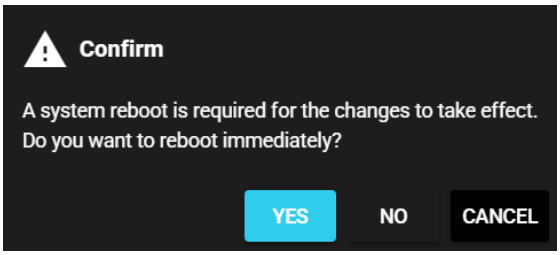


4.「 닫기 」를 누릅니다.

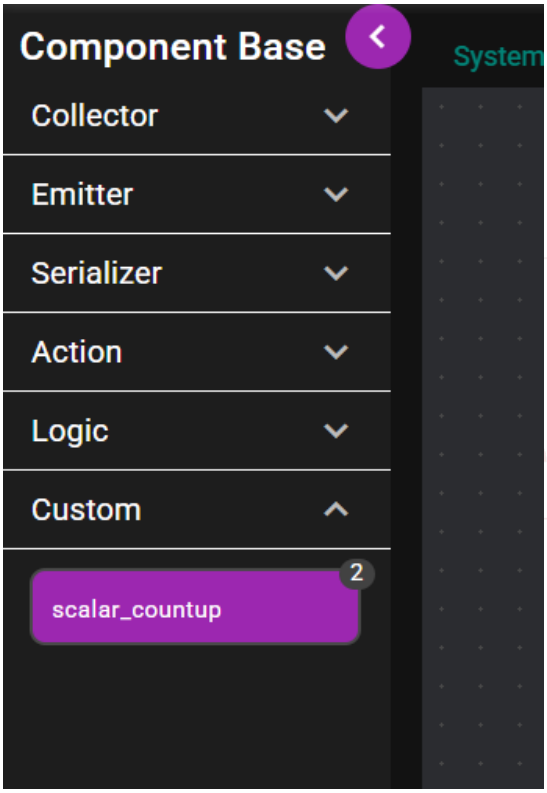
-For Reference Only-



5. 확인 다이얼로그에서 「 예 」를 선택하고, 변경을 반영하기 위해, 시스템을 재기동시킵니다.



6. 좌측 메뉴에, 등록한 커스텀 컴포넌트가 반영됩니다.



번역본
-ference Only-

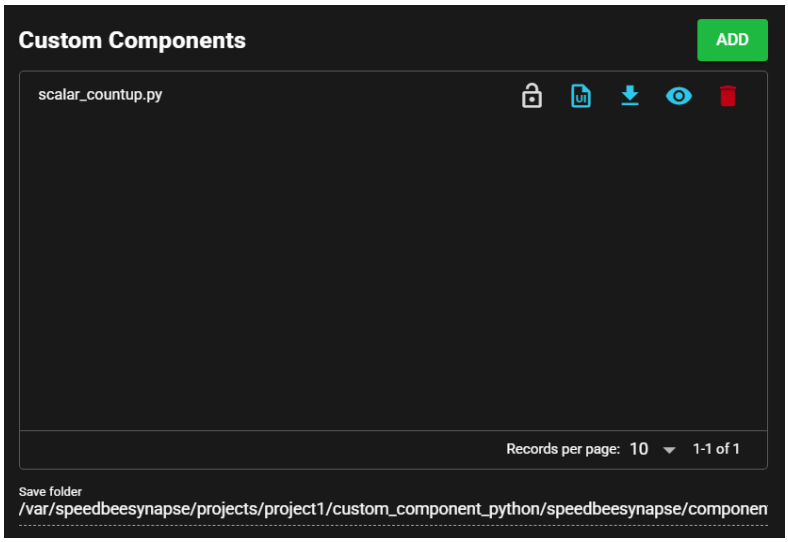
파일의 저장 위치

등록한 커스텀 컴포넌트의 파일은, 다음 디렉터리에 저장됩니다.

Linux

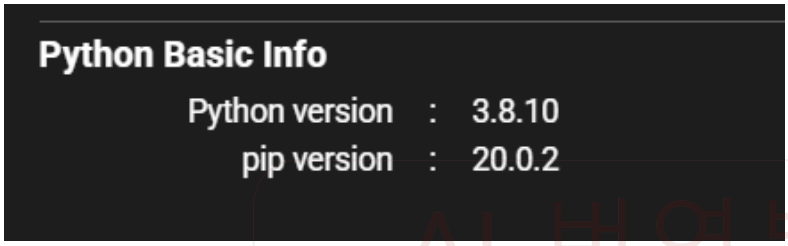
/var/speedbeesynapse/custom_component_python/speedbeesynapse/component/custom

저장 위치 디렉터리는, 화면 항목「저장 위치」에 표시됩니다.



Python 버전

커스텀 컴포넌트를 실행하는 Python의 버전은 화면 항목「Python 기본 정보」의「Python 버전」에 표시됩니다.



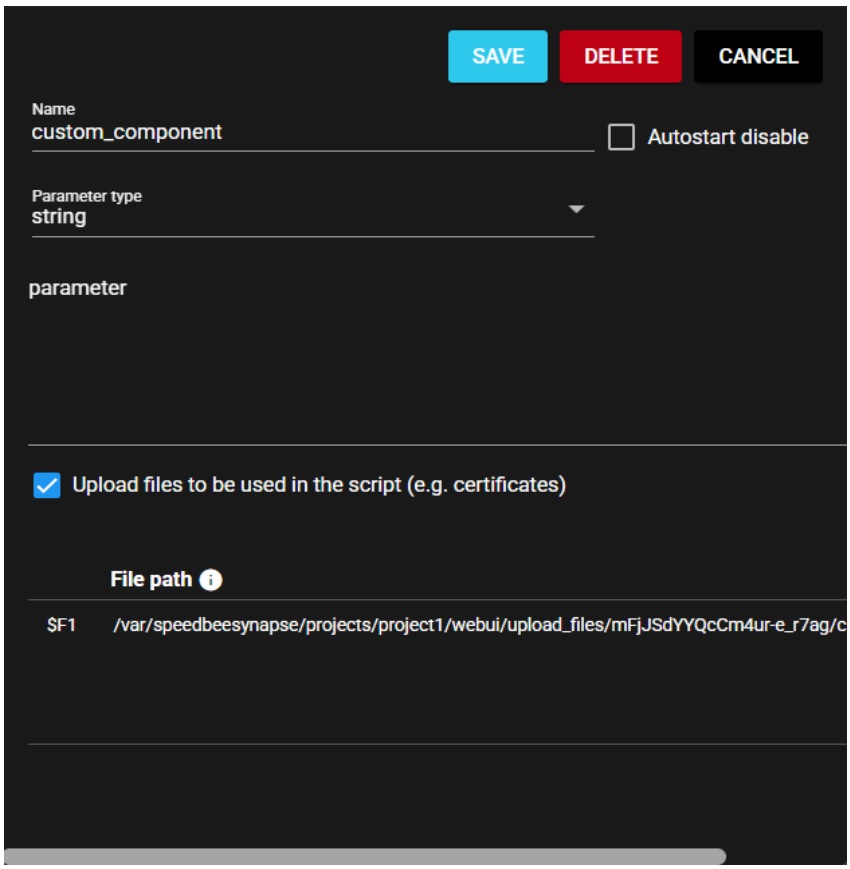
AI 번역본
-For Reference Only-

Python 파일명

커스텀 컴포넌트는 SpeeDBee Synapse에서는 모듈로 취급하고 있으며, 파일명을 모듈명으로 하고 있습니다. 따라서 파일명에 "-"를 사용한 경우 올바르게 다룰 수 없습니다.
Python의 명명 규칙에 따라 파일명은 짧게 모두 소문자로 하고 구분은 "_"를 사용하십시오.

6.2.3.5.2 설정 화면

커스텀 컴포넌트의 설정 화면의 항목은 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
파라미터 타입	파라미터의 종류를 다음 중에서 선택 ・STRING：문자열 ・JSON：JSON 문자열
파라미터	컴포넌트 파라미터를 입력
스크립트에서 사용할 파일을 업로드한다(증명서 등)	커스텀 컴포넌트의 처리에서 사용하는, 증명서 등의 파일을 업로드하는 경우 ON
파일 추가	누르면, 스크립트에서 사용할 파일을 업로드할 수 있습니다. ※새롭게 추가된 파일은, 컴포넌트의 설정을 저장했을 때에 업로드됩니다.

AI 번역본
-For Reference Only-

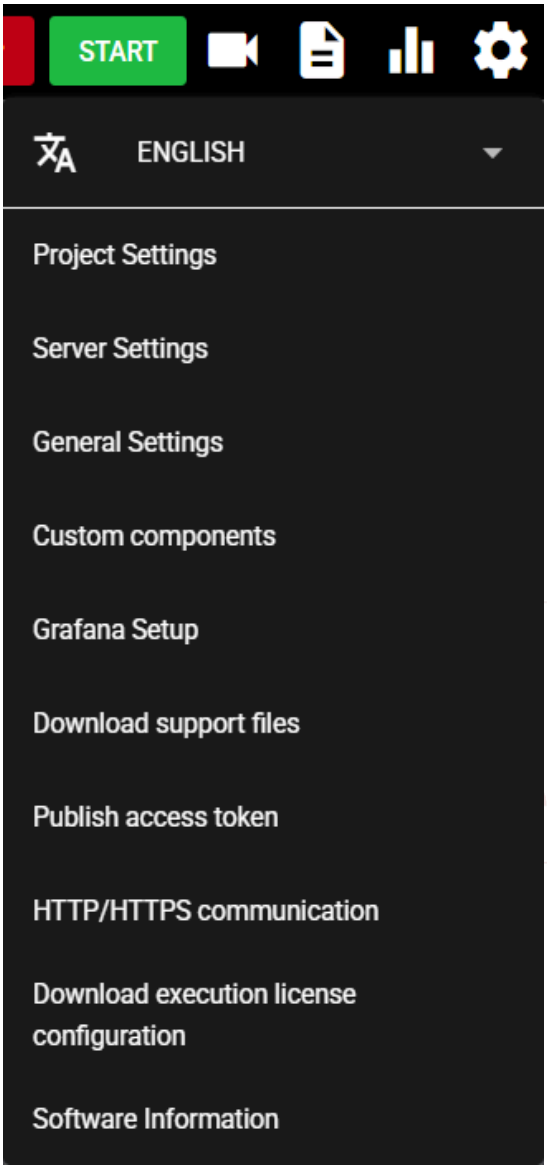
항목	설명
파일 경로	업로드한 파일의 저장 위치 ※파라미터에서 파일 경로를 사용하는 경우, 변수를 사용할 수 있습니다. 상세는, 표제의「파일 경로」의 오른쪽 아이콘을 클릭하면 확인할 수 있습니다.

설정 화면의 파라미터 입력 부분에, 사용자가 정의한 화면 항목을 표시하도록 변경할 수 있습니다. 상세는
「커스텀 UI 작성」을 참조하십시오.

6.2.3.5.3 삭제

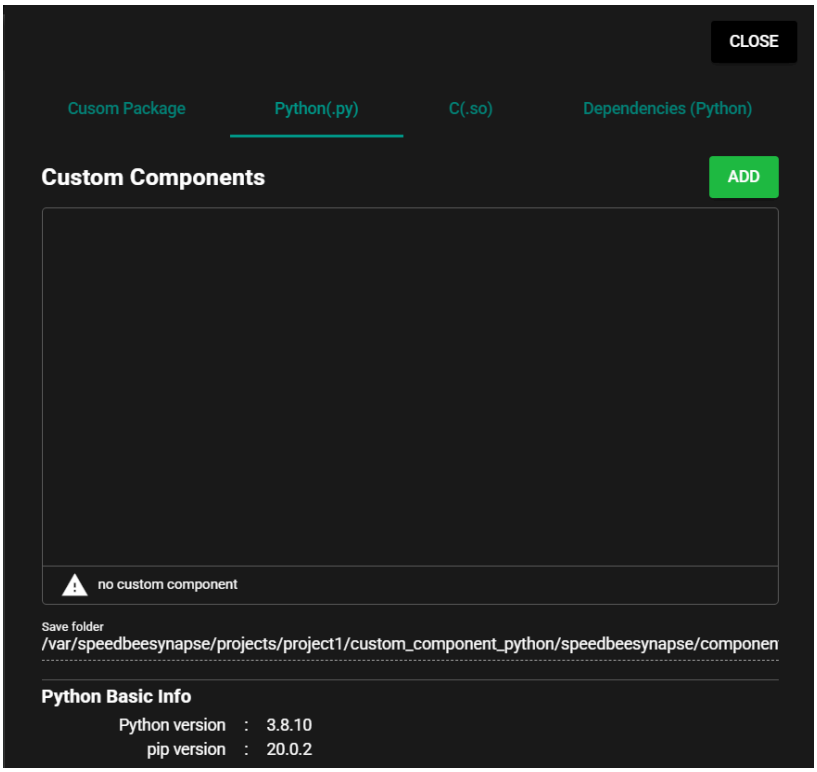
다음 순서로, 등록한 커스텀 컴포넌트를 SpeeDBee Synapse에서 삭제할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「커스텀 컴포넌트」를 선택합니다.

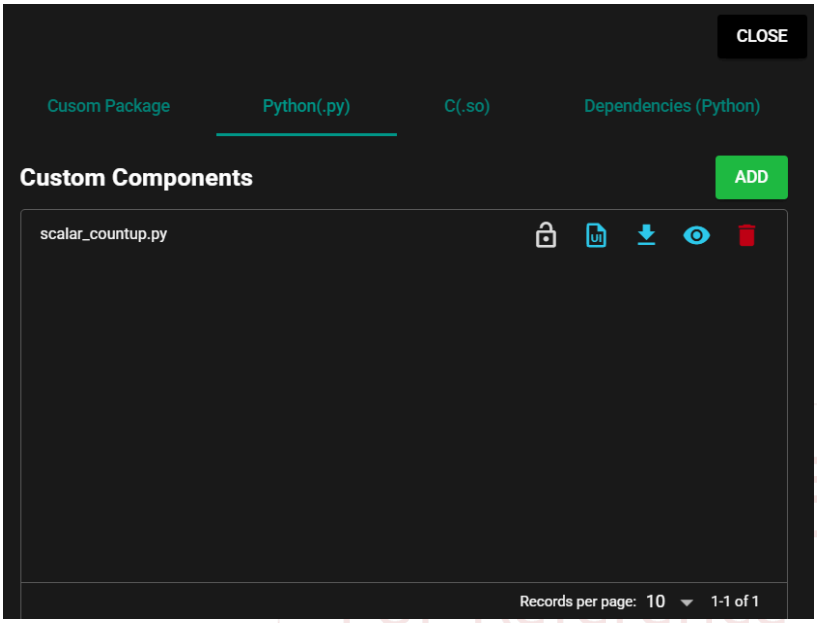


번역본
ference Only-

2. 다이얼로그 상단 탭의 「 Python(.py) 」를 선택합니다.

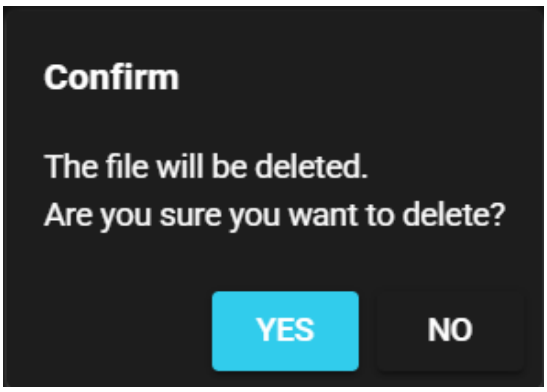


3. 커스텀 컴포넌트의 「 휴지통 아이콘 (삭제) 」을 누릅니다.



4. 확인 다이얼로그에서 「 예 」를 선택합니다.

For Reference Only-

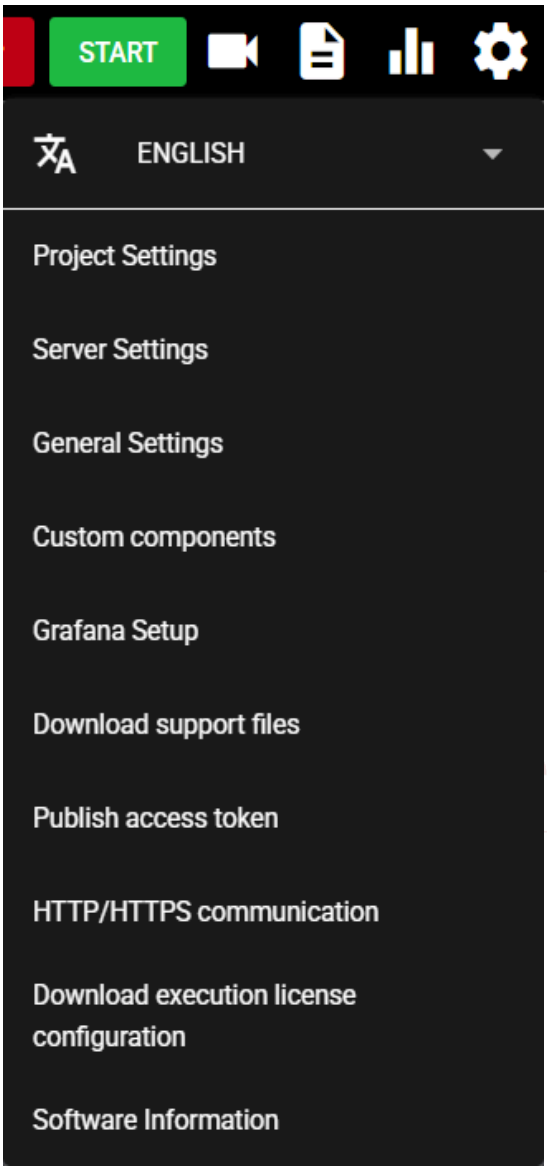


5. 커스텀 컴포넌트가 삭제됩니다.

6.2.3.5.4 표시

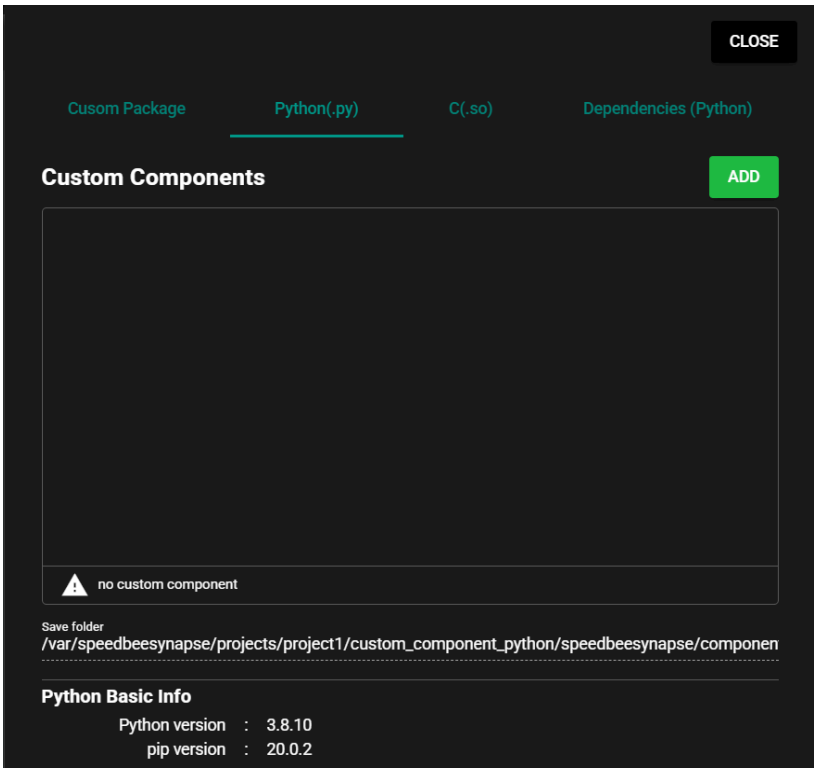
다음 순서로 등록한 커스텀 컴포넌트의 내용을 표시할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

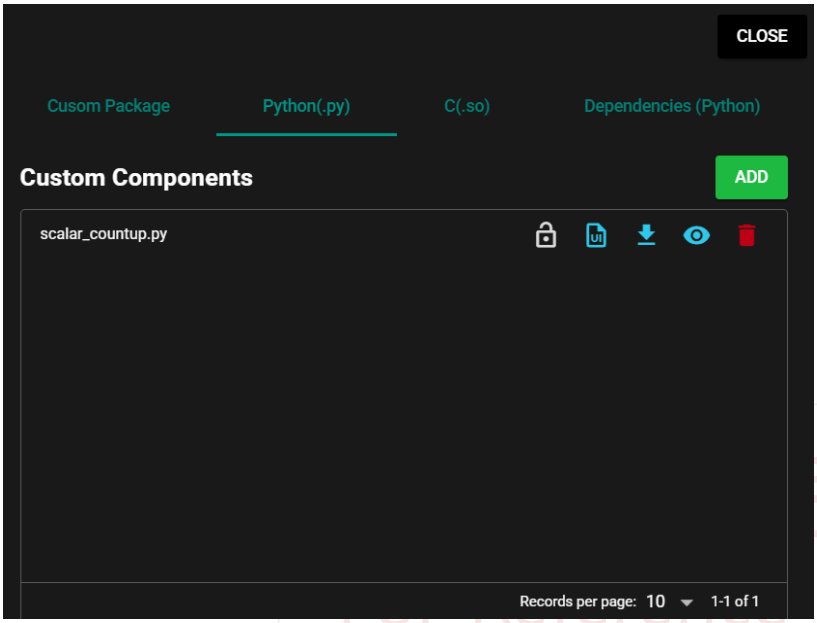


번역본
ference Only-

2. 다이얼로그 상단 탭의 「 Python(.py) 」를 선택합니다.



3. 커스텀 컴포넌트의 「 비저빌리티 아이콘(표시) 」을 누릅니다.



4. 커스텀 컴포넌트의 소스 코드가 화면에 표시됩니다.

For Reference Only-

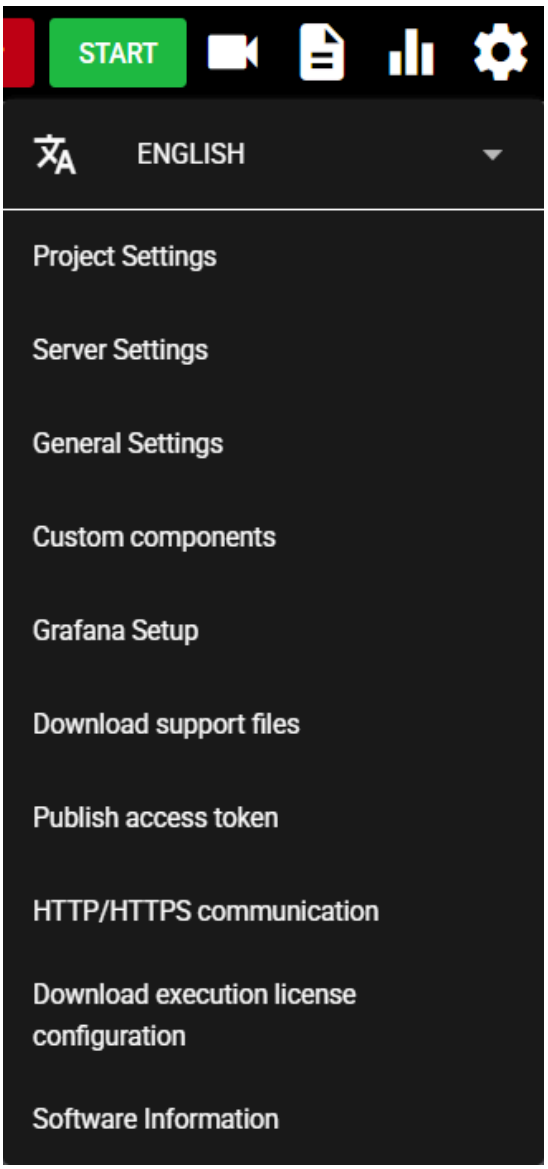


6.2.3.5.5 다운로드

다음 순서로, 등록한 커스텀 컴포넌트를 다운로드할 수 있습니다.

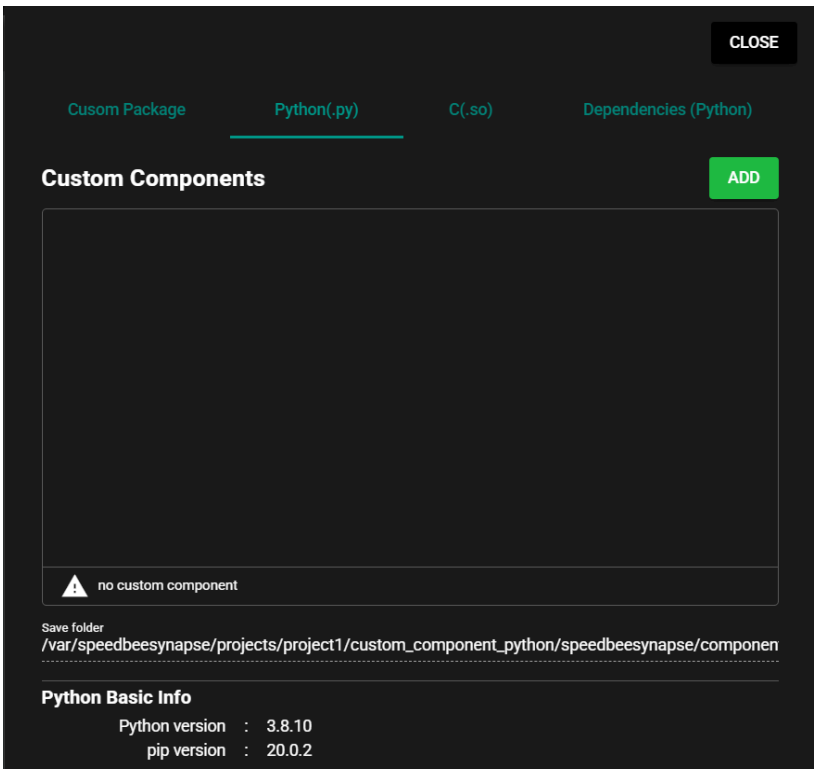
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

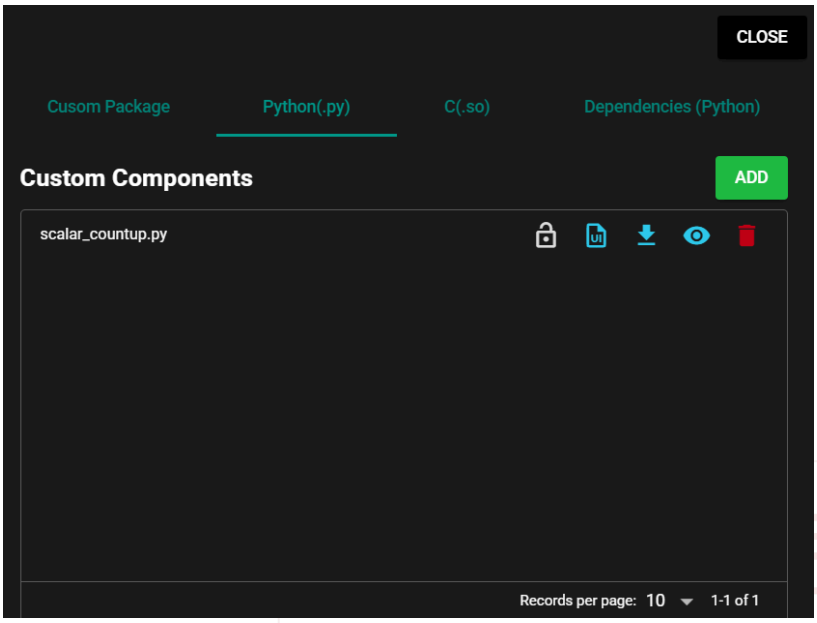


2. 다이얼로그 상단 탭의 「 Python(.py) 」를 선택합니다.

AI 번역본
-For Reference Only-



3. 커스텀 컴포넌트의「다운로드 아이콘（취득）」을 누릅니다.



4. 커스텀 컴포넌트의 파일이 다운로드됩니다.

For Reference Only-

6.2.3.5.6 보호의 부여

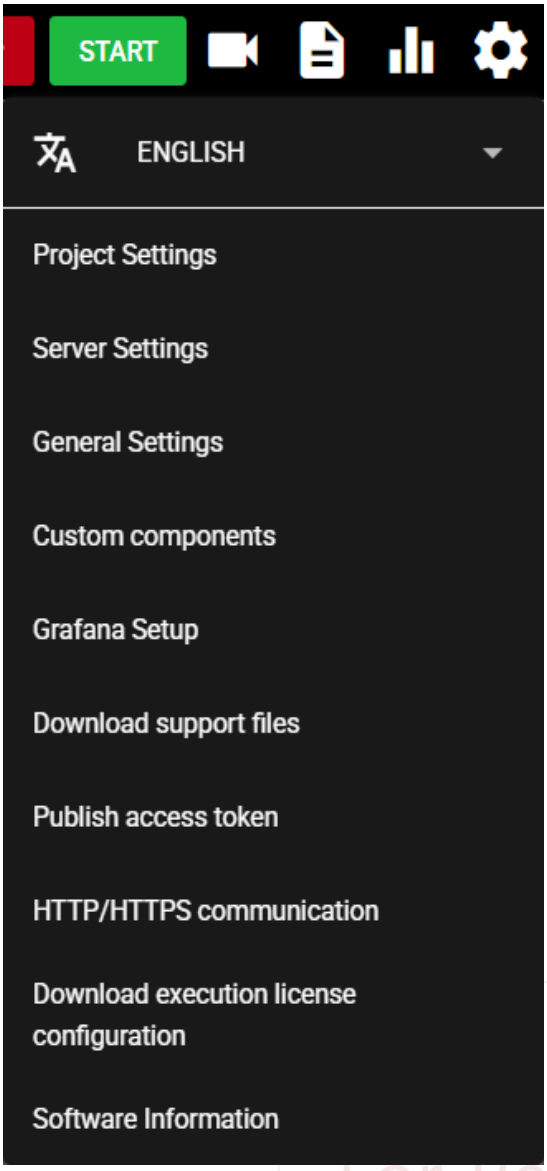
등록한 컴포넌트에는 보호를 걸 수 있습니다.
보호된 컴포넌트는 다음 조작을 수행할 수 없게 되므로, 의도하지 않은 갱신을 방지할 수 있습니다.

- 등록
- 삭제

- 표시
- 다운로드
- 커스텀 UI의 설정

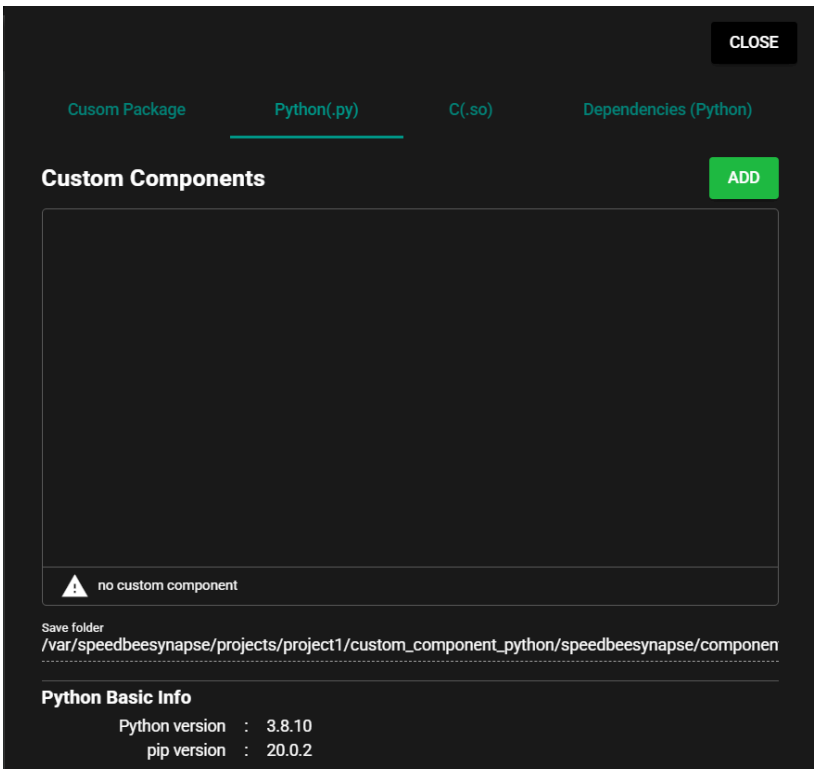
다음 순서로, 커스텀 컴포넌트에 보호를 걸 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

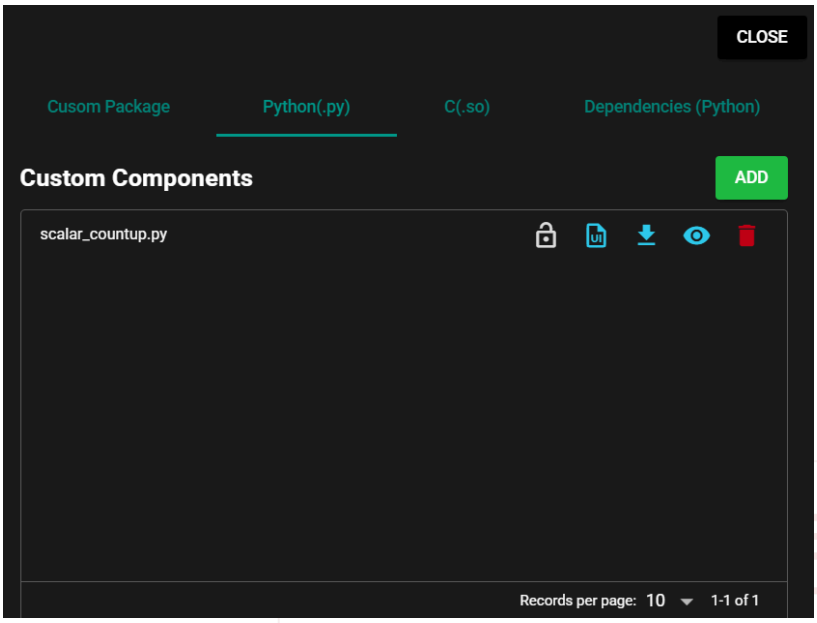


2. 다이얼로그 상단 탭의「 Python(.py) 」를 선택합니다.

번역본
-For Reference Only-

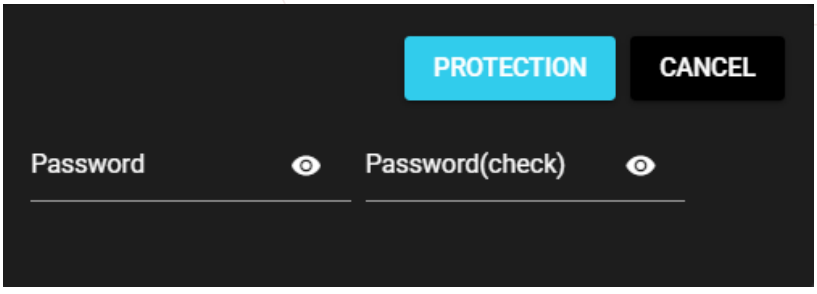


3. 커스텀 컴포넌트의「자물쇠 아이콘（보호）」을 누릅니다.



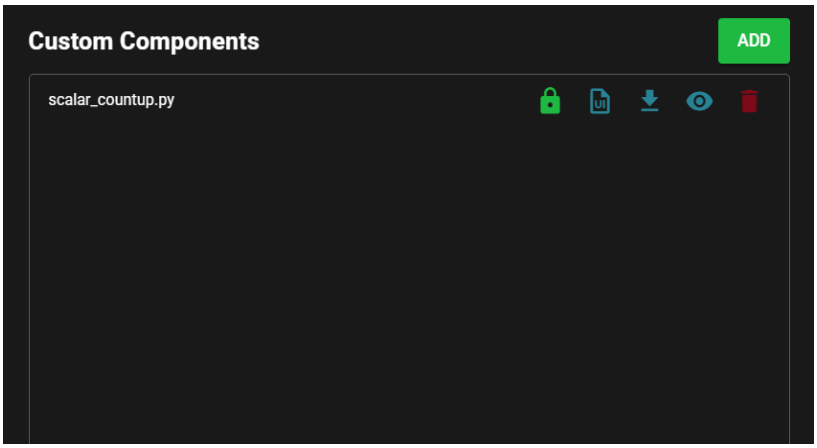
For Reference Only-

4. 보호 해제할 때에 필요하게 되는 패스워드를 입력하고,「보호」를 누릅니다.



> ⚠ 경고
>
> - 패스워드는 복원할 수 없는 형식으로 저장되며, 한 번 설정하면 보호를 해제할 때까지 재설정할 수 없습니다. 보호
> - 설정한 패스워드를 잊은 경우, 해당 커스텀 컴포넌트는 조작 불능이 됩니다.
>
> 패스워드의 취급에는 충분히 주의하십시오.

5. 커스텀 컴포넌트가 보호 상태가 됩니다.

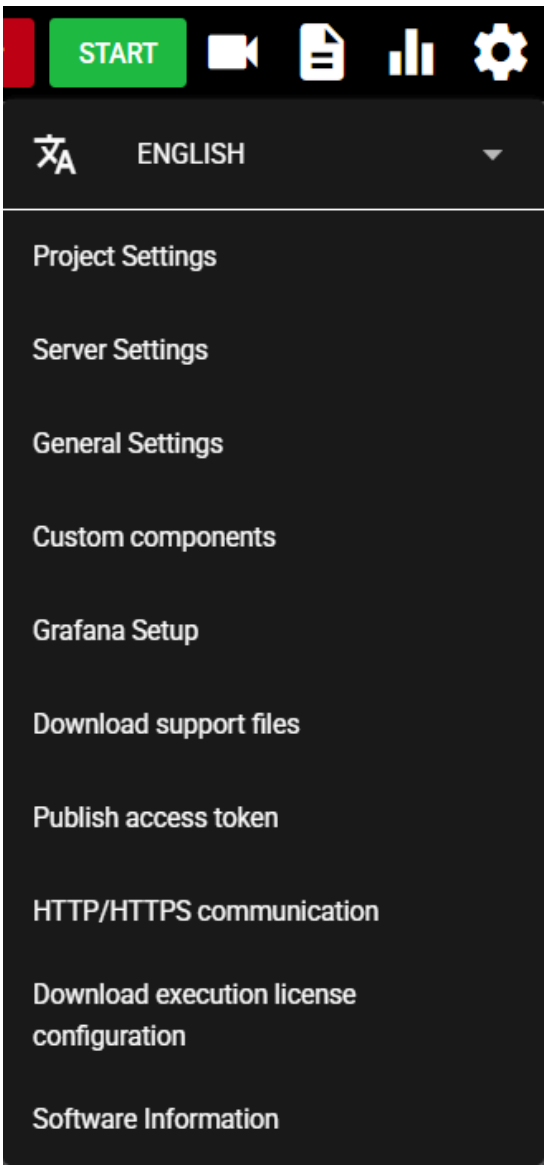


6.2.3.5.7 보호의 해제

다음 순서로, 보호된 커스텀 컴포넌트의 보호를 해제할 수 있습니다.

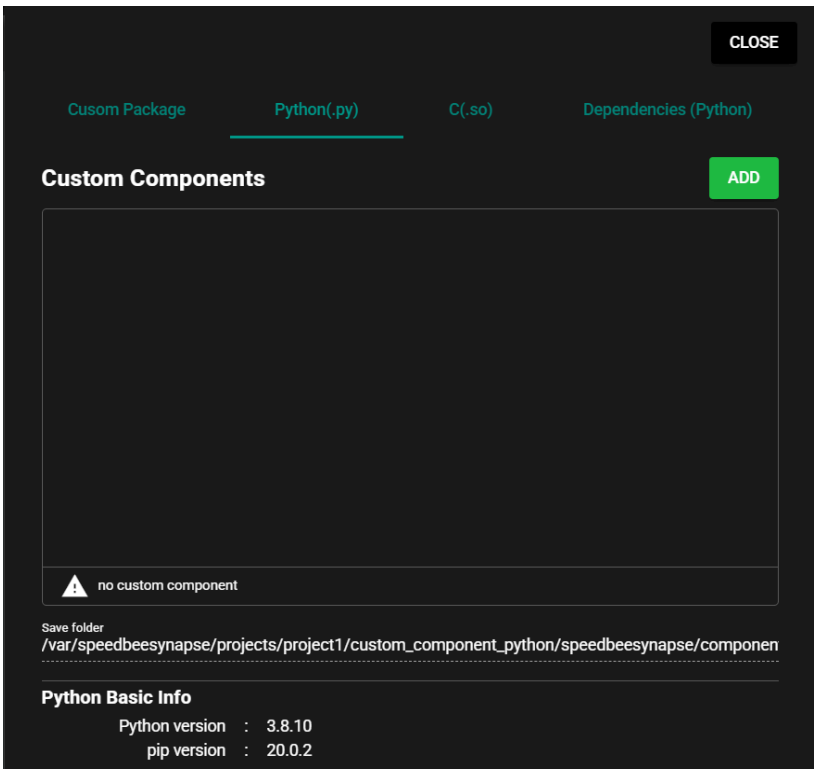
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

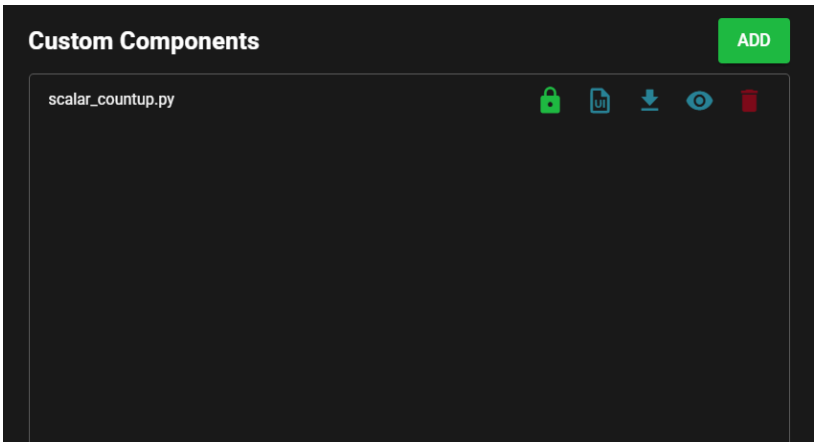


2. 다이얼로그 상단 탭의「 Python(.py) 」를 선택합니다.

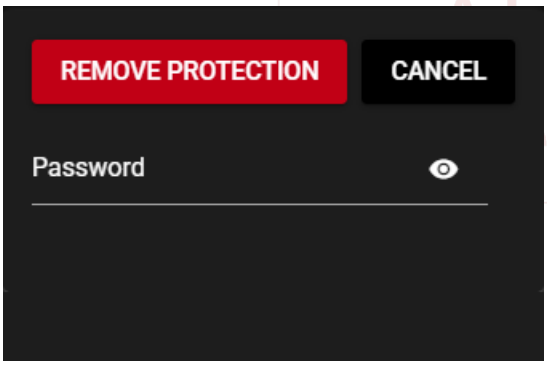
AI 번역본
-For Reference Only-



3. 커스텀 컴포넌트의「자물쇠 아이콘（보호 해제）」를 누릅니다.

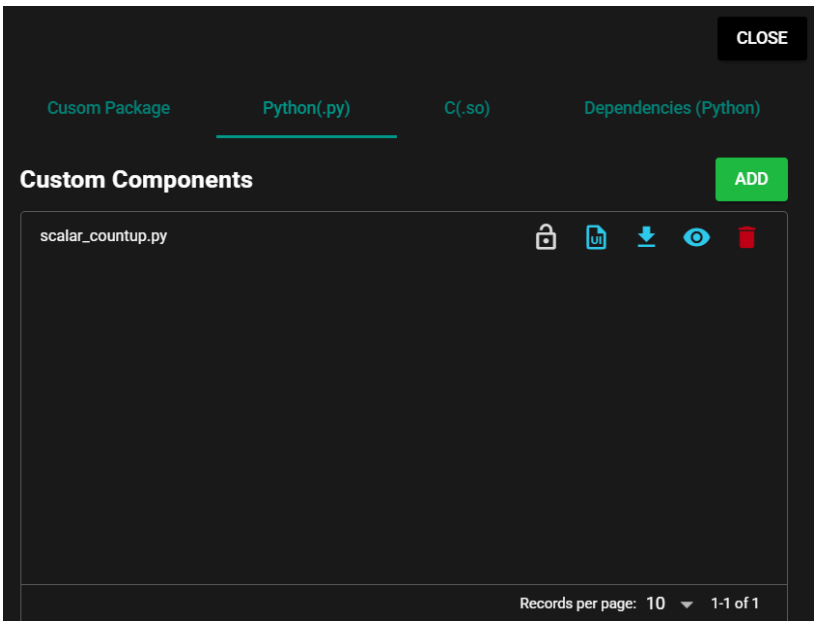


4. 보호를 걸 때에 설정한 패스워드를 입력하고,「보호 해제」를 누릅니다.



5. 커스텀 컴포넌트의 보호가 해제됩니다.

AI 번역본
-For Reference Only-

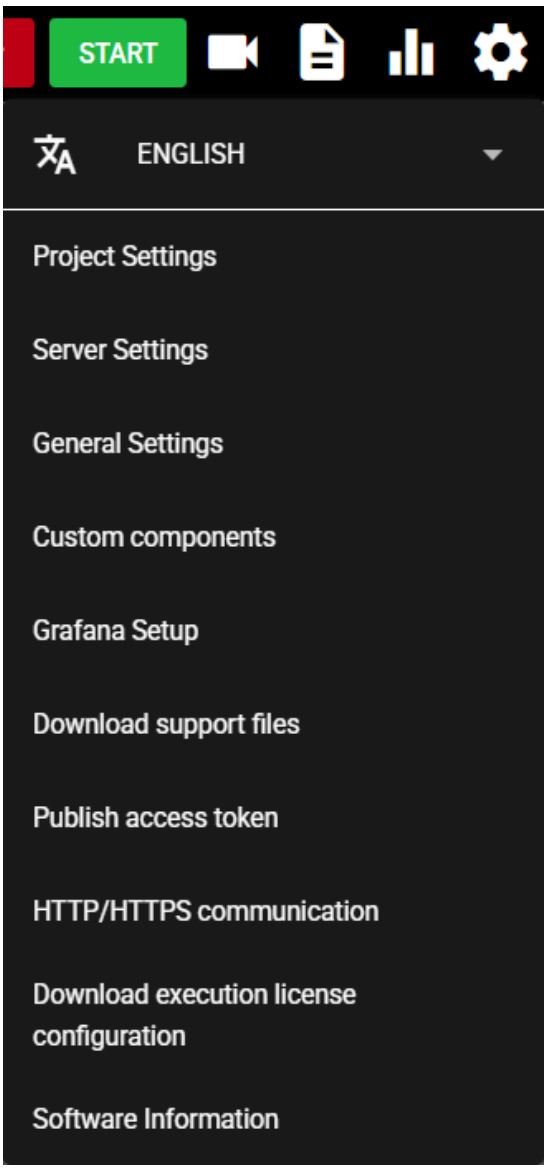


6.2.3.5.8 외부 라이브러리의 추가

다음 순서로 커스텀 컴포넌트가 사용하는 외부 라이브러리를 SpeeDBee Synapse에 추가할 수 있습니다.

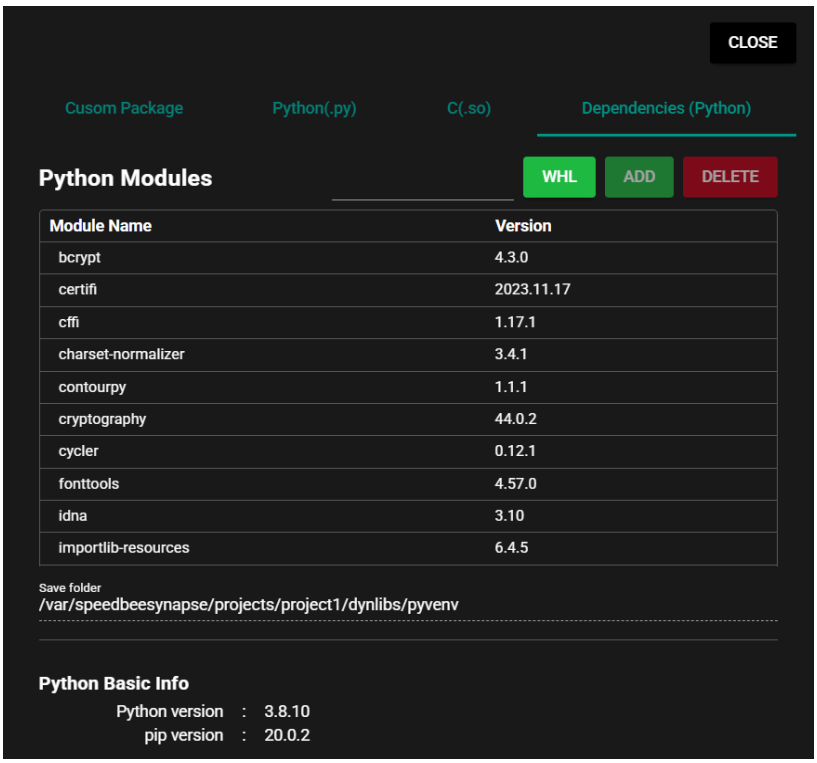
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

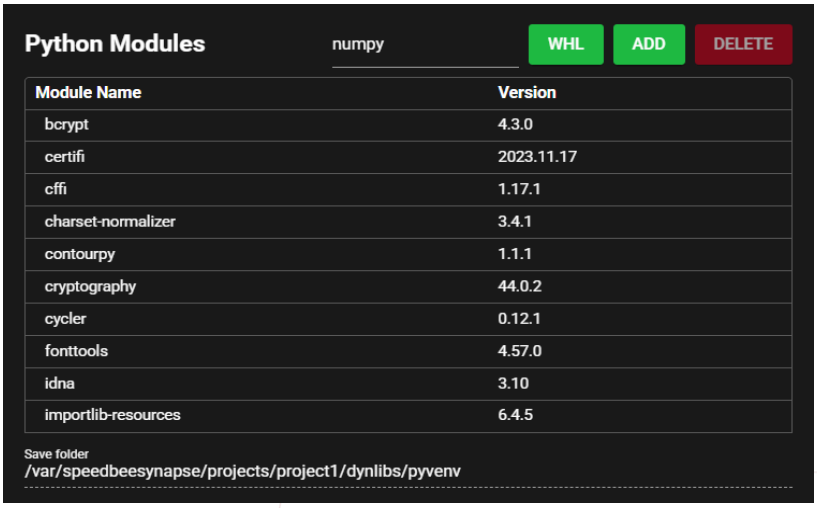


2. 다이얼로그 상단 탭의 「 의존 모듈(Python) 」을 선택합니다.

AI 번역본
-For Reference Only-

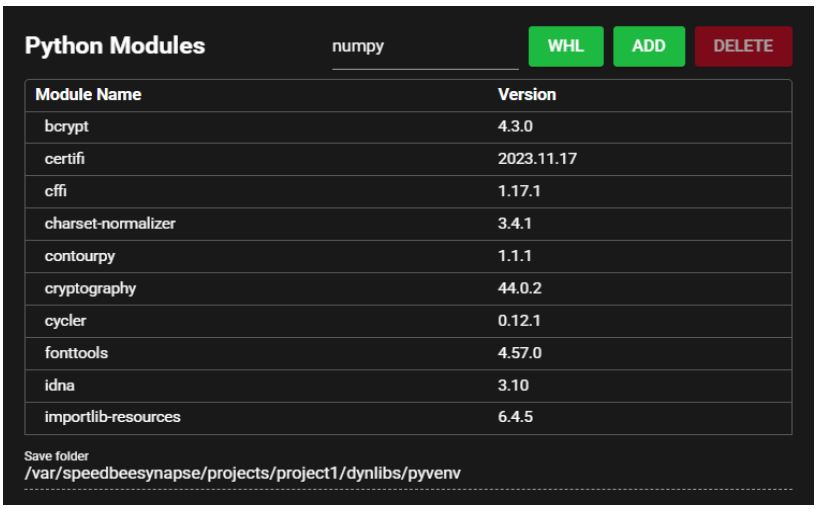


3. Python 모듈의 입력란에 등록할 라이브러리의 이름을 입력합니다.

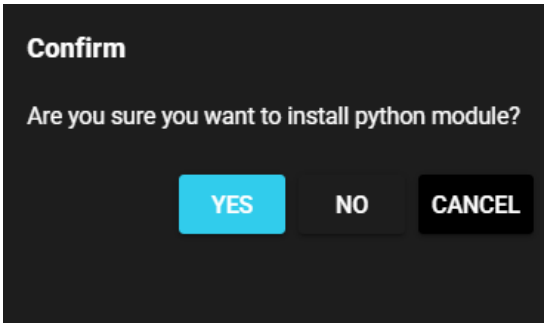


4. Python 모듈의 「 추가 」를 누릅니다.

AI 번역본
-For Reference Only-



5. 확인 다이얼로그에서 「 예 」를 선택합니다.



6. 라이브러리가 등록됩니다.

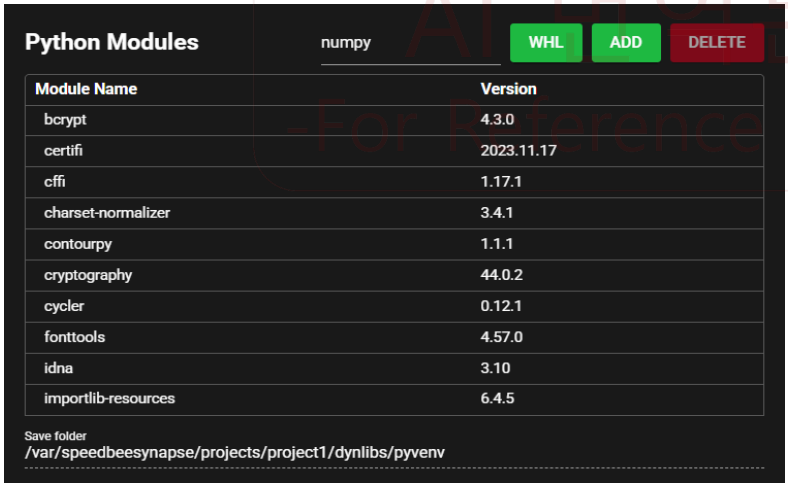
외부 라이브러리의 저장처

등록한 외부 라이브러리는 다음 디렉터리에 저장됩니다.

Linux

/var/speedbeesynapse/dynlibs/pyvenv

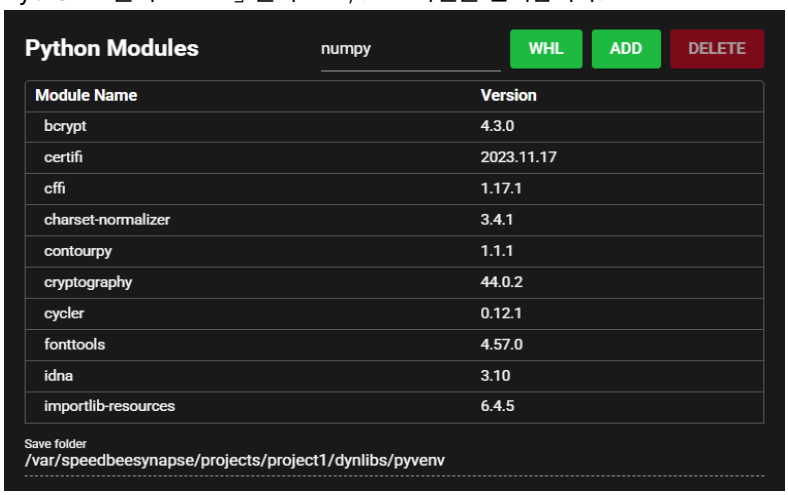
저장 위치 디렉터리는, 화면 항목 「 저장 위치 」에 표시됩니다.



AI 번역본
-For Reference Only-

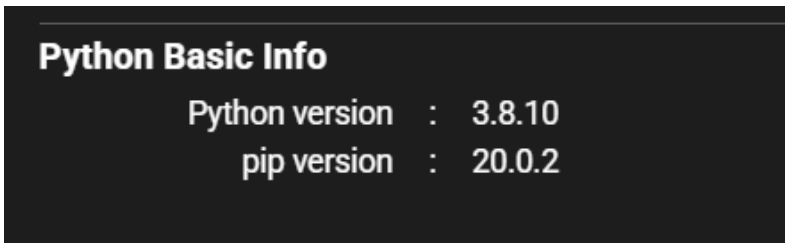
.whl 파일에서 추가한다

Python 모듈의 「 whl 」을 누르고, .whl 파일을 선택합니다.



pip 버전

외부 라이브러리 관리에서 사용하는 pip의 버전은 화면 항목 「 Python 기본 정보 」의 「 pip 버전 」에 표시됩니다.

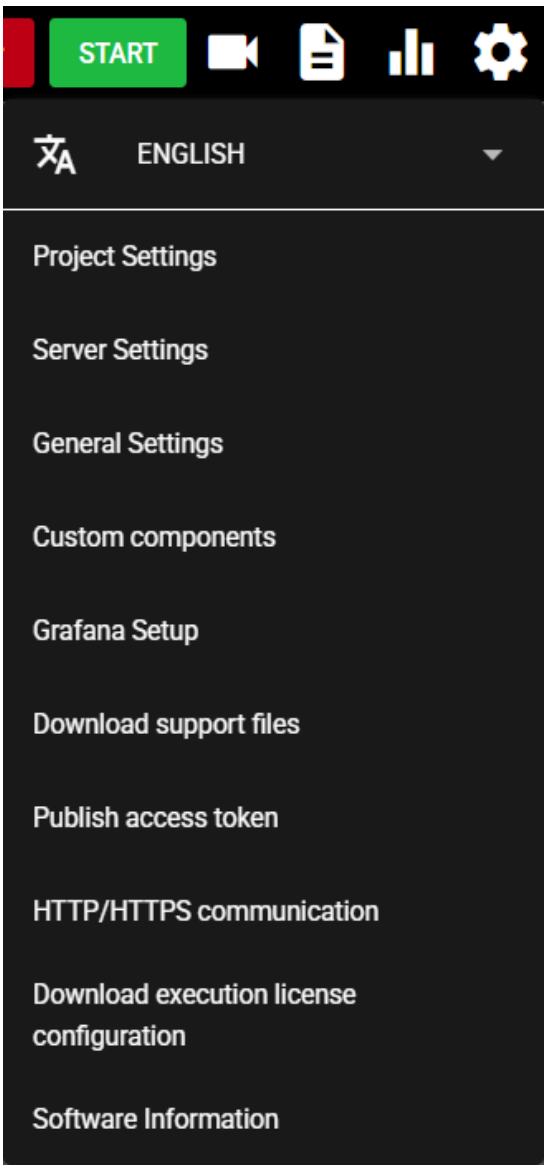


6.2.3.5.9 외부 라이브러리의 삭제

다음 순서로 추가한 외부 라이브러리를 SpeeDBee Synapse에서 삭제할 수 있습니다.

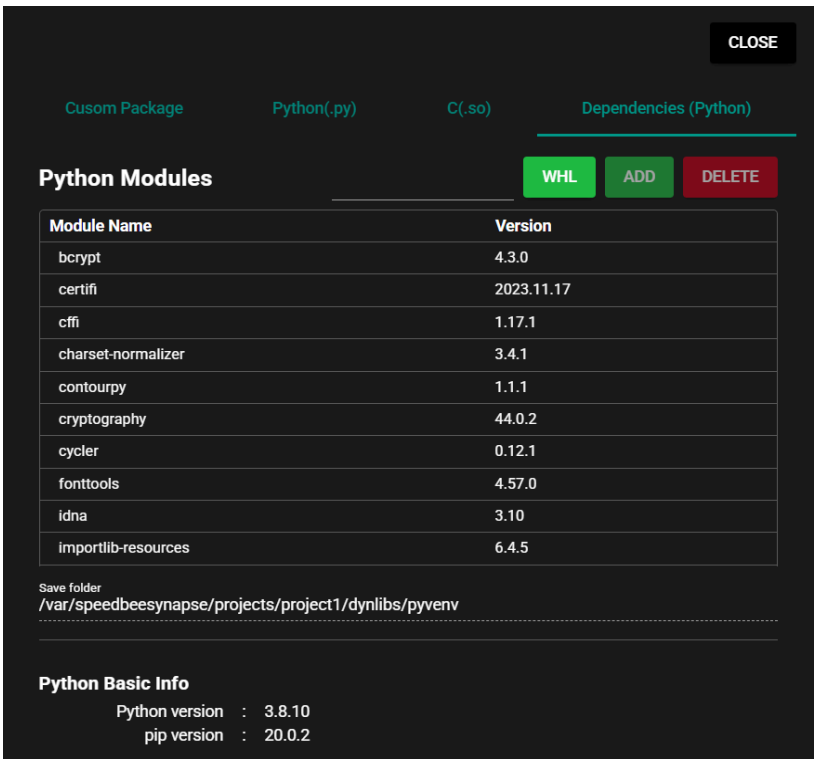
1. 설정 메뉴 아이콘을 누르고, 「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-



2. 다이얼로그 상단 탭의 「 의존 모듈(Python) 」을 선택합니다.

AI 번역본
-For Reference Only-



3. 삭제할 라이브러리를 클릭합니다.

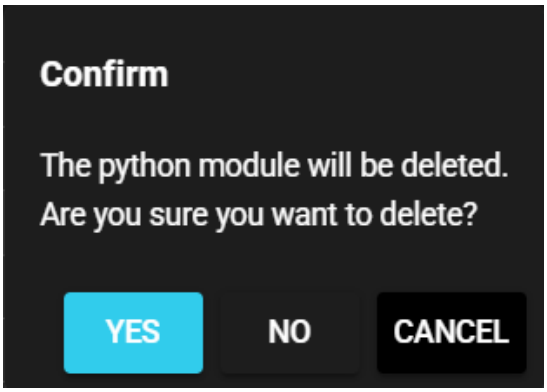


4. Python 모듈의 「 삭제 」를 누릅니다.



AI 버전 본
-Only-

5. 확인 다이얼로그에서 「 예 」를 선택합니다.



6. 라이브러리가 삭제됩니다.

AI 번역본
-For Reference Only-

6.2.4 C언어 커스텀 컴포넌트 작성

본 항에서는 C언어에 의한 커스텀 컴포넌트의 작성 방법에 대해 설명합니다.

6.2.4.1 전제 지식

본 항의 내용은 이하에 기재하는 지식・경험을 전제로 하고 있습니다.

- C언어를 이용한 일반적인 프로그램 개발 경험
- C언어 개발에 필요한 툴류의 준비
- Makefile에 의한 빌드 실행 방법

또한 본 항에서는 SpeeDBee Synapse가 가동되는 환경 내에서 C 프로그램을 빌드하는 것을 전제로 기재하고 있습니다.
SpeeDBee Synapse가 가동되는 환경과는 다른 환경에서 프로그램을 빌드하여 이동하는 경우에는 그 환경에 맞춰 크로스 빌드가 필요합니다. 크로스 빌드 방법은 가동되는 환경의 매뉴얼 등을 확인하십시오.

6.2.4.2 샘플 커스텀 컴포넌트

C언어에 의한 커스텀 컴포넌트의 실장 방법 설명을 위해 심플한 사양의 커스텀 컴포넌트를 샘플 로 제공하고 있습니다.

샘플 커스텀 컴포넌트의 다운로드

혹은, SpeeDBee Synapse를 설치 완료한 경우, 커스텀 컴포넌트 샘플도 아래의 장소에 설치되어 있습니다.

```
$ ls /usr/local/speedbeesynapse/share/sample_component_c/*
/usr/local/speedbeesynapse/share/sample_component_c/cmake_project:
CMakeLists.txt  README.md  countandrandom.c  readandlog.c

/usr/local/speedbeesynapse/share/sample_component_c/makefile_project_for_linux:
Makefile  README.md  countandrandom.c  readandlog.c

/usr/local/speedbeesynapse/share/sample_component_c/makefile_project_for_windows:
Makefile  README.md  countandrandom.c  readandlog.c
```

For Reference Only

샘플 커스텀 컴포넌트의 디렉터리는 빌드 환경에 맞춰 아래 3종류가 준비되어 있습니다.

- `cmake_project`

  CMake를 사용하여 빌드하기 위한 프로젝트 디렉터리입니다. Linux/Windows 모두 같은 순서로 빌드할 수 있습니다.

- `makefile_project_for_linux`

  Make를 사용하여 빌드하기 위한 Linux 환경 전용 프로젝트 디렉터리입니다.

- `makefile_project_for_windows`

  Windows의 NMAKE를 사용하여 빌드하기 위한 프로젝트 디렉터리입니다.

6.2.4.2.1 샘플 컴포넌트의 빌드

각 프로젝트 디렉터리에서의 빌드 방법은 각각의 README.md 파일을 참조하십시오. 어느 프로젝트 디렉터리도 빌드하여 생성되는 컴포넌트는 동일한 것이 됩니다.

SpeeDBee Synapse를 실행하는 환경이 Windows인 경우, cmake_project 또는 makefile_project_for_windows 중 하나를 선택하십시오.
실행하는 환경이 Linux인 경우, cmake_project 또는 makefile_project_for_linux 중 하나를 선택하십시오.

6.2.4.2.2 샘플의 내용

커스텀 컴포넌트의 샘플로서 본 항에서는 아래 샘플을 설명합니다.

- `count_and_random` (countandrandom.c)

  출력 포트에 대해 이하의 2 개의 컬럼을 작성하고, 데이터를 계속 출력하는 컴포넌트입니다.

컬럼명	내용
count	3초에 1회, 1부터 순서대로 카운트업한 값을 등록해 나갑니다
random	3초에 1회, [0.0, 100.0) 범위 내의 값을 무작위로 등록합니다

- `read_and_log` (readandlog.c)

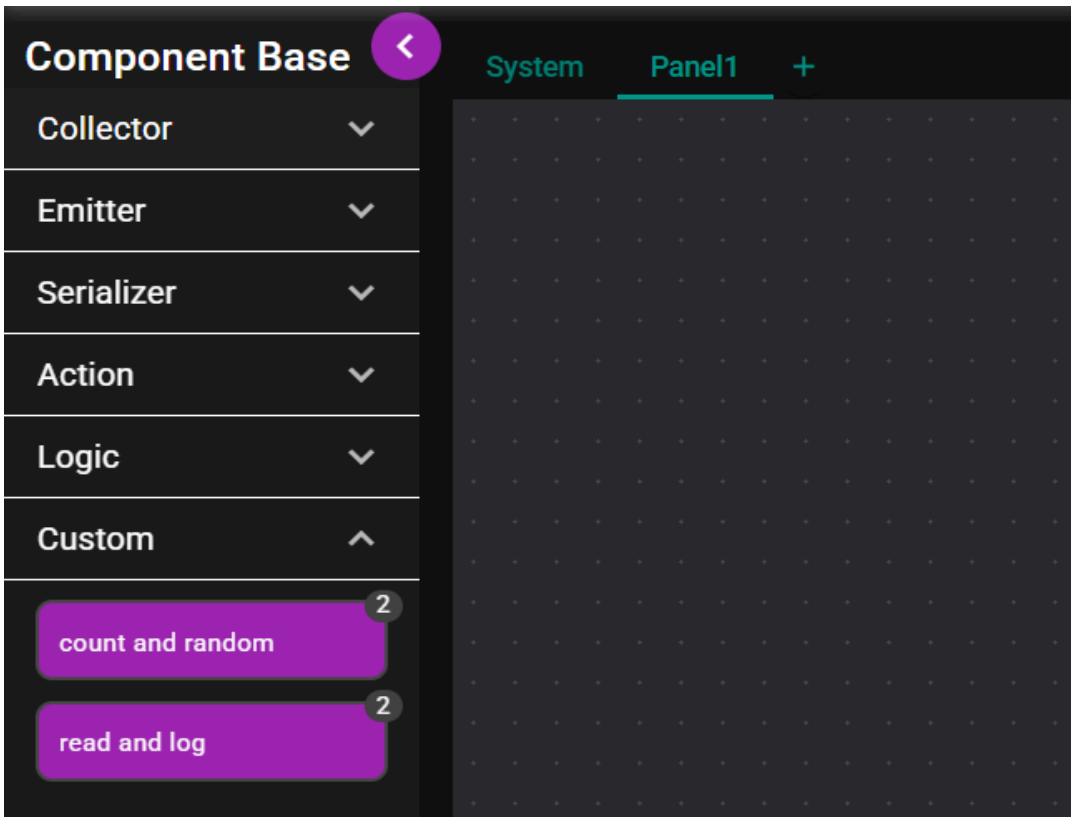
  입력 포트에서 데이터를 받아들여, 그것을 문자열형 컬럼과 로그에 출력합니다.

AI 번역본
For Reference Only

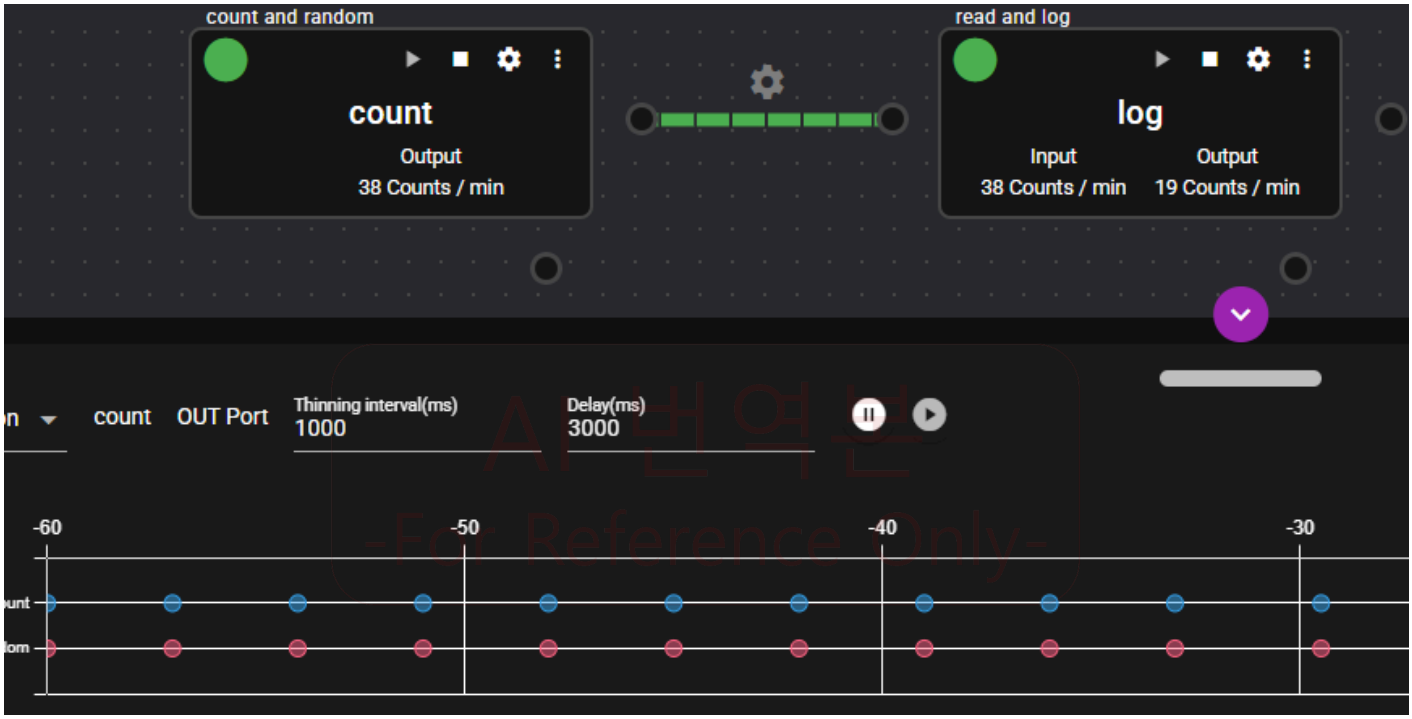
6.2.4.2.3 실행

샘플 컴포넌트의 빌드에 성공하면, Linux라면 .so 파일이, Windows라면 .dll 파일이 컴포넌트마다 생성됩니다. 생성된 .so or .dll 파일을 WEBUI에서 등록함으로써 SpeeDBee Synapse에서 컴포넌트로 이용할 수 있게 됩니다. 등록 순서는 커스텀 컴포넌트를 사용한다를 참조하십시오.

파일을 등록하고 코어가 재기동되면, SpeeDBee Synapse의 화면에서 다른 컴포넌트와 마찬가지로 새로운 커스텀 컴포넌트를 사용할 수 있게 되어 있습니다.



count_and_random 및 read_and_log 를 연결하여 실행하면, 각각이 출력한 컬럼을 표시할 수 있습니다. 아래는 count_and_random 의 출력 포트를 모니터링하고 있습니다.



6.2.4.3 커스텀 컴포넌트 실장 상세

이후에서는 샘플인 count_and_random 및 read_and_log 의 소스 코드를 예로, 커스텀 컴포넌트의 실장 내용을 설명해 나갑니다.

6.2.4.3.1 인클루드 헤더

C에 의한 커스텀 컴포넌트 개발에서는 헤더 파일 hiveframework.h 를 인클루드해야 합니다.

```
#include <hiveframework.h>

  :   이후, 커스텀 컴포넌트의 실장
```

6.2.4.3.2 커스텀 컴포넌트 정보 정의

C에 의한 커스텀 컴포넌트의 소스 코드에서는 아래의 전역 변수 정의에 의해 그 컴포넌트의 이름이나 식별 정보, 라이프사이클 콜백 함수의 정의가 필요합니다.

```
  :   커스텀 컴포넌트의 각종 함수 정의

const static HIVE_COMPONENT_DEFINITION info = {
  .uuid            = "2e80ad02-1730-4bdc-b321-2fbc2d1b942e",
  .name            = "count and random",
  .parameter_type  = HIVE_COMPONENT_PARAMETER_NONE,
  .in_ports        = 0,
  .out_ports       = 1,
  .functions     = {
    .constructor         = NULL,      // optional
    .premain             = premain,   // optional
    .main                = mainloop,  // required
    .postmain            = postmain,  // optional
    .destructor          = NULL,      // optional
    .stop                = NULL,      // optional
  },
};
EXPORT_COMPONENT_DEFINITION(info);
```

AI 번역본
-For Reference Only

멤버명	정의 정보	설명
uuid	UUID	컴포넌트를 식별하기 위한 UUID입니다. 무작위 ID를 생성하여 여기에 설정하십시오.
name	컴포넌트명	컴포넌트의 이름을 설정합니다. 화면에는 이 이름이 표시되므로 되도록 다른 컴포넌트와 겹치지 않는 것이 좋습니다.

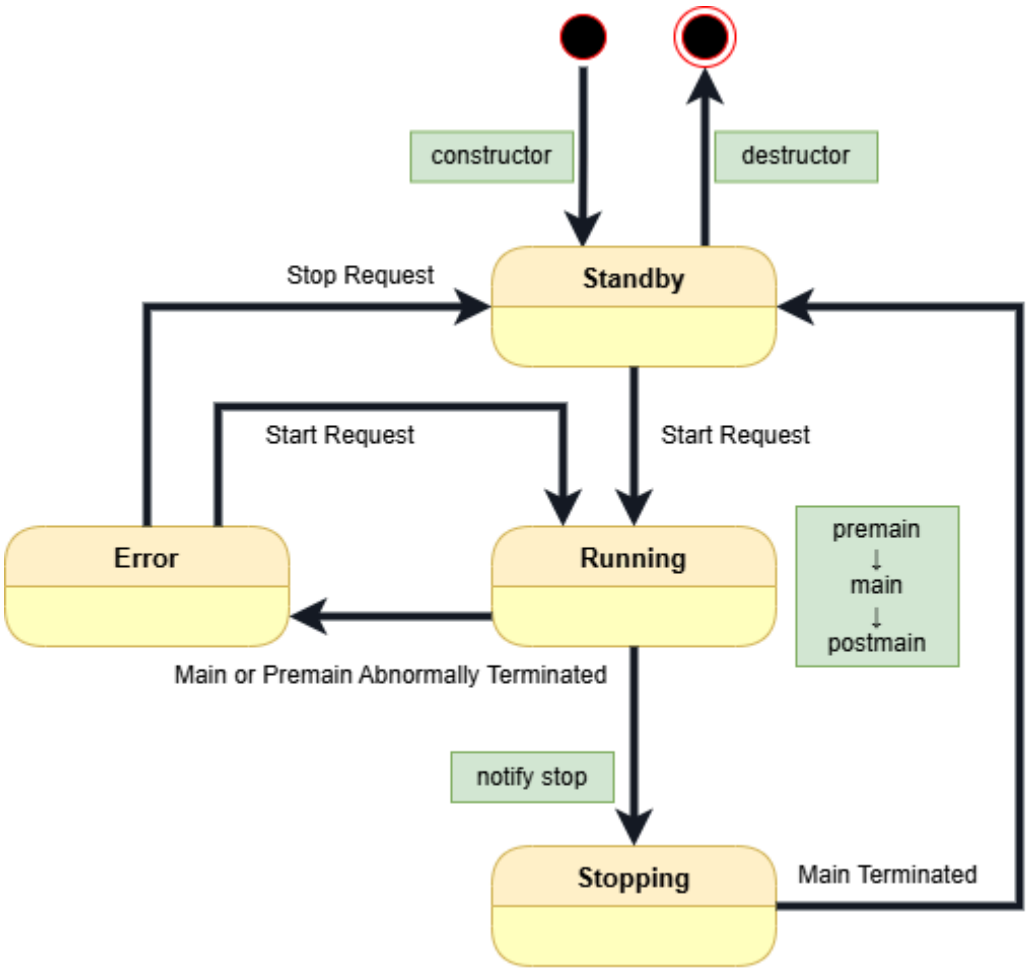
멤버명	정의 정보	설명
`tag`	태그	컴포넌트에 태그를 부여합니다. "collector","emitter","serializer","action","logic" 중 하나를 지정함으로써 컴포넌트 베이스의 배치처를 지정할 수 있습니다. 생략한 경우에는 Custom에 배치됩니다.
`parameter_type`	파라미터 종별	이 컴포넌트에 전달하는 파라미터의 형식을 NONE, TEXT, JSON 중에서 선택할 수 있지만, 현재 이 항목은 사용되지 않습니다.
`out_ports`	출력 포트 수	이 컴포넌트가 갖는 출력 포트의 수를 설정합니다. 다른 컴포넌트의 입력 포트에 데이터를 전달하는 경우에는 이것을 1로 하십시오.
`in_ports`	입력 포트 수	이 컴포넌트가 갖는 입력 포트의 수를 설정합니다. 다른 컴포넌트의 출력 포트로부터 데이터를 받는 경우에는 이것을 1로 하십시오.
`functions`	콜백 함수 리스트	이 컴포넌트의 동작을 실장한 함수의 리스트를 설정합니다. 다음 절 참조.

UUID에 대해서는 반드시 무작위 ID를 생성하십시오. 다른 컴포넌트와 같은 UUID로 해 버리면 어느 컴포넌트도 사용할 수 없게 되는 경우가 있습니다.

6.2.4.3.3 라이프사이클 콜백 정의

앞 절의 functions 멤버는 컴포넌트 인스턴스의 라이프사이클에 기반하여 호출되는 함수 포인터를 설정할 수 있습니다.

AI 번역본
-For Reference Only-



콜백 함수	설명
`constructor`	컴포넌트의 인스턴스가 생성되었을 때 호출됩니다
`destructor`	컴포넌트의 인스턴스가 파기될 때 호출됩니다
`premain`	컴포넌트를 개시할 때 main보다 앞서 호출됩니다
`main`	컴포넌트의 메인 처리를 실장하는 함수입니다 컴포넌트의 종료 요청을 받을 때까지 이 함수는 종료하지 않고 루프 처리를 계속할 필요가 있습니다
`postmain`	컴포넌트 종료 시, main 종료 직후에 호출됩니다
`stop`	main 실행 중, 컴포넌트의 정지 요청을 받았을 때 실행됩니다

AI 번역본
For Reference Only

앞 절의 functions 정의는 샘플 컴포넌트인 count_and_random 의 끝에 정의되어 있습니다.

이 컴포넌트에서는 constructor , destructor 는 불필요하므로 생략(NULL 지정)하고 있습니다. premain , postmain 도 불필요하면 생략할 수 있습니다.(이 샘플에서도 실제로는 로그 출력밖에 실장되어 있지 않습니다)

main 은 컴포넌트 처리의 본체가 되므로 필수입니다. 실제 함수 정의명은 mainloop 등 다른 이름으로 하십시오. main으로 하면 C의 실행 형식 본체인 main 함수와 충돌해 버립니다.

이후에서는 샘플 프로그램의 main 함수 실장 내용을 설명합니다.

6.2.4.3.4 출력 포트의 컬럼 작성/데이터 등록 (count_and_random)

아래는 커스텀 컴포넌트 count_and_random 의 메인 처리입니다. 이를 바탕으로 데이터를 출력하는 흐름을 설명합니다.

```
static bool mainloop(HIVE_COMPONENT comp, const char *param, HIVE_STATUS status) {
  HIVE_LOG_INFO("create columns");
  HIVE_OUTCOLUMN clm_count = hive_outport_create_column(comp, OUTPORT1, "count",
HIVE_DATA_SCALAR(HIVE_TYPE_INT32), HIVE_COLUMN_OPTION_NONE);
  HIVE_OUTCOLUMN clm_random = hive_outport_create_column(comp, OUTPORT1, "random",
HIVE_DATA_SCALAR(HIVE_TYPE_DOUBLE), HIVE_COLUMN_OPTION_NONE);

  int count = 1;
  while (hive_component_runnable(comp)) {
    HIVE_LOG_INFO("insert count");
    if (!hive_outcolumn_insert(clm_count, &count)) {
      HIVE_API_ERROR err = hive_get_api_error();
      HIVE_LOG_ERROR("insert 'count' failed: errcode=0x%08x", err.code);
    }

    HIVE_LOG_INFO("insert random");
    double r = 100 * (double)rand() / (double)RAND_MAX;
    if (!hive_outcolumn_insert(clm_random, &r)) {
      HIVE_API_ERROR err = hive_get_api_error();
      HIVE_LOG_ERROR("insert 'count' failed: errcode=0x%08x", err.code);
    }

    usleep(3000000);
    count++;
  }

  return true;
}
```

For Reference Only

1. 컬럼 작성

   먼저 컬럼을 작성하고 있습니다.

```
HIVE_OUTCOLUMN clm_count = hive_outport_create_column(comp, OUTPORT1, "count",
HIVE_DATA_SCALAR(HIVE_TYPE_INT32), HIVE_COLUMN_OPTION_NONE);
HIVE_OUTCOLUMN clm_random = hive_outport_create_column(comp, OUTPORT1, "random",
HIVE_DATA_SCALAR(HIVE_TYPE_DOUBLE), HIVE_COLUMN_OPTION_NONE);
```

이 컴포넌트에서는 32bit 정수의 컬럼 count 와 배정밀도 부동소수점의 컬럼 random 을 이용하므로, hive_outport_create_column() 함수를 사용하여 인수에 컬럼명, 데이터형 등을 지정해서 컬럼을 생성하고 있습니다. 반환값은 나중에 컬럼에 데이터를 등록할 때 사용하므로 변수에 유지해 두십시오.

컴포넌트가 출력하는 모든 데이터는 이렇게 생성한 컬럼에 대해 등록해 나가야 합니다.

2. 정기적인 처리 반복

정기적으로 데이터를 등록하기 위해 while 루프로 반복 처리를 실장하고 있습니다

```
int count = 1;
while (hive_component_runnable(comp)) {
  :

  usleep(3000000);
  count++;
}
```

루프의 판정 조건에서 호출하고 있는 hive_component_runnable() 은 컴포넌트가 실행 상태인 동안은 true를 반환합니다. 화면에서 컴포넌트의 정지를 요청하면 이 함수가 false를 반환하게 되므로, 그 경우에는 신속하게 이 mainloop 함수를 종료시킬 필요가 있습니다.

반복 처리에는 이러한 while 루프 외에 hive_component_interval_call() 을 사용할 수도 있습니다. 사용 방법은 링크처를 참조하십시오.

3. 컬럼으로의 데이터 등록

반복 처리 안에서 컬럼에 대해 데이터를 등록합니다.

```
if (!hive_outcolumn_insert(clm_count, &count)) {
  HIVE_API_ERROR err = hive_get_api_error();
  HIVE_LOG_ERROR("insert 'count' failed: errcode=0x%08x", err.code);
}

double r = 100 * (double)rand() / (double)RAND_MAX;
if (!hive_outcolumn_insert(clm_random, &r)) {
  HIVE_API_ERROR err = hive_get_api_error();
  HIVE_LOG_ERROR("insert 'count' failed: errcode=0x%08x", err.code);
}
```

데이터 등록은 hive_outcolumn_insert() 로 할 수 있습니다. 제1 인수에는 사전에 작성한 컬럼 변수를, 제2 인수에는 등록할 데이터의 포인터를 지정합니다. 이때의 포인터는 사전에 컬럼을 작성했을 때 지정한 데이터형과 맞출 필요가 있습니다. 다른 형의 데이터를 지정하면 소프트웨어가 이상 종료하는 원인이 되는 경우가 있으므로 주의하십시오.

> **insert로 등록할 수 있는 최대 사이즈**
>
> hive_outcolumn_insert()로 등록할 수 있는 데이터의 최대 사이즈는 512바이트(STRING은 NULL을 포함하지 않는 511바이트)입니다. 상한 사이즈를 초과한 값의 등록은 반환값이 false가 됩니다.

6.2.4.3.5 입력 포트로부터의 데이터 불러오기 (read_and_log)

아래는 커스텀 컴포넌트 read_and_log 의 메인 처리입니다.(주석이나 로그 출력은 생략)

```
static bool mainloop(HIVE_COMPONENT comp, const char *param, HIVE_STATUS status) {
  HIVE_OUTCOLUMN clm_log = hive_outport_create_column(comp, OUTPORT1, "log",
HIVE_DATA_SCALAR(HIVE_TYPE_STRING), HIVE_COLUMN_OPTION_NONE);

  HIVE_CONTINUOUS_READER *creader
    = hive_inport_continuous_reader(comp, INPORT1, hive_timestamp()+5000000000UL);

  hive_time_t last_updated = 0;
  while (hive_component_runnable(comp)) {
    const HIVE_COLUMN_READ_RESULT *read_result = hive_continuous_reader_read(creader);
    if (!read_result) {
      continue;
    }

    if (last_updated < read_result->incolumn_list->updated_at) {
      last_updated = read_result->incolumn_list->updated_at;
    }

    HIVE_RECORD_ITERATOR iter = HIVE_GET_RECORD_ITERATOR(read_result);
    const HIVE_RECORD *record;
    while ((record = HIVE_RECORD_ITERATOR_GET_NEXT(&iter)) != NULL) {
      for (int i=0; i < record->data_count; i++) {
        const HIVE_RECORD_DATA *rd = HIVE_RECORD_GET_DATA(record, i);
        if (rd && rd->data_size != 0) {
          log_record(clm_log, record->timestamp, &read_result->incolumn_list->incolumns[i], rd);
        }
      }
    }
  }
  hive_continuous_reader_release(creader);

  return true;
}
```

1. 컬럼 작성

   이 컴포넌트에서도 먼저 컬럼을 작성하고 있습니다.

   ```
   HIVE_OUTCOLUMN clm_log = hive_outport_create_column(comp, OUTPORT1, "log",
   HIVE_DATA_SCALAR(HIVE_TYPE_STRING), HIVE_COLUMN_OPTION_NONE);
   ```

   이 컬럼은 문자열을 출력하기 위한 것입니다.

2. 데이터 취득 루프

   입력 포트로부터의 데이터를 반복하여 받는 처리는 아래와 같습니다.

AI 번역본

```
HIVE_CONTINUOUS_READER *creader
  = hive_inport_continuous_reader(comp, INPORT1, hive_timestamp()+5000000000UL);

:
while (hive_component_runnable(comp)) {
  const HIVE_COLUMN_READ_RESULT *read_result = hive_continuous_reader_read(creader);
  if (!read_result) {
    continue;
  }

  :
}
hive_continuous_reader_release(creader);
```

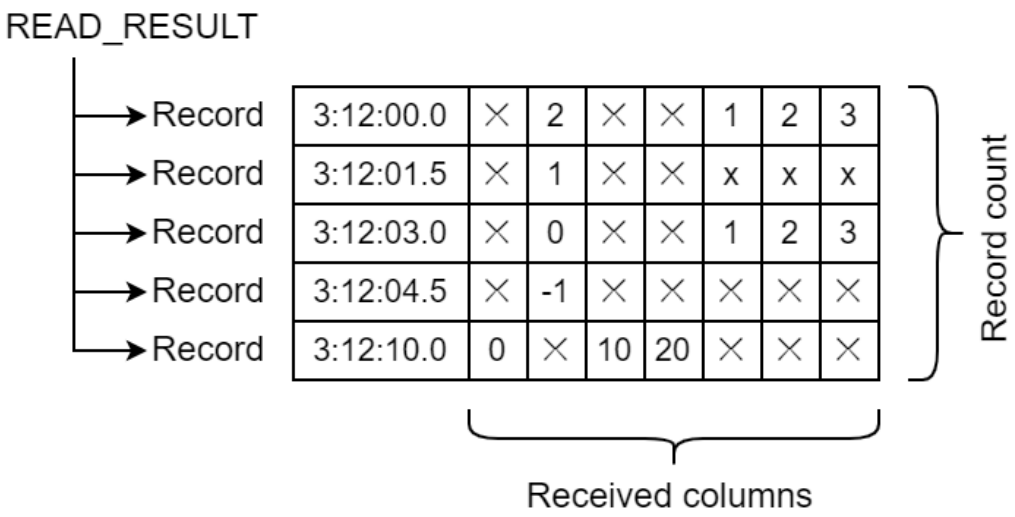
여기서는 4개의 함수를 이용하고 있습니다

- `hive_inport_continuous_reader()`
  계속적으로 입력 포트로부터의 데이터를 취득하기 위한 핸들을 작성합니다
- `hive_continuous_reader_release()`
  데이터 취득을 정지하는 경우, 이 함수로 핸들을 해제합니다
- `hive_continuous_reader_read()`
  작성한 핸들로부터 데이터를 취득합니다. 취득한 데이터는 HIVE_COLUMN_READ_RESULT 구조체의 포인터로 반환되지만, 데이터가 아직 없는 경우에는 NULL 이 반환되는 경우도 있으므로, 그 경우에는 무시하십시오.

- `hive_component_runnable()`
  이것은 앞 절에서도 사용했습니다. 컴포넌트가 실행 상태인 동안은 true를 반환합니다. 화면에서 컴포넌트의 정지를 요청하면 이 함수가 false를 반환하게 되므로, 그 경우에는 신속하게 이 mainloop 함수를 종료시킬 필요가 있습니다.

3. 취득한 데이터의 레코드 참조

hive_continuous_reader_read() 함수에 의해 입력 포트에 들어온 데이터를 어느 시간 폭만큼 모아서 취득하여 HIVE_COLUMN_READ_RESULT 구조체의 포인터, read_result 에 격납합니다. HIVE_COLUMN_READ_RESULT 구조체는 아래 그림과 같이 동일 시각의 데이터를 모은 복수의 레코드로 이루어지며, 하나의 레코드에 복수의 컬럼 데이터가 담겨 있습니다.



아래와 같은 이중 루프에 의해 이 구조체에서 1레코드, 1데이터씩 처리할 수 있습니다.

`HIVE_GET_RECORD_ITERATOR ()와 HIVE_RECORD_ITERATOR_GET_NEXT 가 레코드의 루프, 그 내부에 있는 변수 i`

에 의한 for문이 레코드 내 데이터 처리의 반복입니다

```
HIVE_RECORD_ITERATOR iter = HIVE_GET_RECORD_ITERATOR(read_result);
const HIVE_RECORD *record;
while ((record = HIVE_RECORD_ITERATOR_GET_NEXT(&iter)) != NULL) {
  for (int i=0; i < record->data_count; i++) {
    const HIVE_RECORD_DATA *rd = HIVE_RECORD_GET_DATA(record, i);
    if (rd && rd->data_size != 0) {
      // 여기서 rd가 1컬럼분의 데이터를 유지하고 있다
      log_record(clm_log, record->timestamp, &read_result->incolumn_list->incolumns[i], rd);
    }
  }
}
```

4. 레코드 내 1데이터의 취급

`하나의 데이터 정보는 HIVE_RECORD 구조체와 HIVE_COLUMN_READ_RESULT 구조체의 incolumn_list 멤버로부터`

취득할 수 있습니다.

항목	형	본 샘플 프로그램에서의 참조 예	내용
데이터명	const char*	`read_result->incolumn_list->incolumns[i]->data_name`	데이터를 등록하고 있는 컬럼명
컴포넌트명	const char*	`read_result->incolumn_list->incolumns[i]->source_name`	데이터를 생성한 컴포넌트명
데이터형	HIVE_DATA_TYPE	`read_result->incolumn_list->incolumns[i]->data_type`	데이터를 등록하고 있는 컬럼의 데이터형
데이터 사이즈	uint32_t	`HIVE_RECORD_GET_DATA(record, i)->data_size`	실데이터의 데이터 사이즈
데이터 주소	char *	`HIVE_RECORD_GET_DATA(record, i)->data`	실데이터를 격납한 주소

-For Reference Only-

위와 같이 실데이터는 const char 포인터로 되어 있습니다. 다양한 데이터형을 받을 가능성이 있으므로, HIVE_DATA_TYPE 로 식별하여 각각의 형에 맞춰 캐스트해서 참조할 필요가 있음에 주의하십시오.

6.2.4.3.6 로그 출력

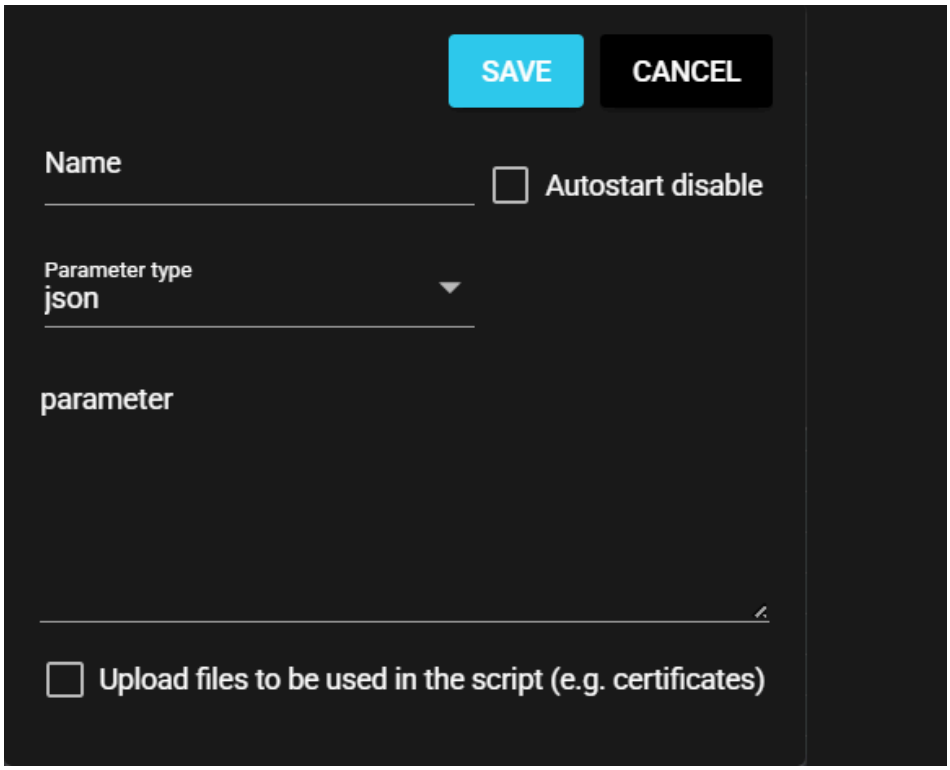
컴포넌트에서는 임의의 타이밍에 로그를 출력할 수 있습니다.

```
  HIVE_LOG_INFO("create columns");
  HIVE_LOG_INFO("x=%d, y=%f, z=%s", x, 12.34, "aiueo");
  HIVE_LOG_ERROR("insert 'count' failed: errcode=0x%08x", err.code);
```

위와 같이 표준 라이브러리의 printf() 와 같은 형식으로 로그를 출력할 수 있습니다. 출력한 로그는 화면에서 다운로드할 수 있습니다.「 각종 로그 」를 참조하십시오.

6.2.4.3.7 컴포넌트 파라미터

화면상에서 컴포넌트 인스턴스를 생성하면 그 설정 항목으로서 컴포넌트의 실행 시 파라미터를 설정할 수 있습니다.



파라미터의 종별로서는 JSON이든 STRING이든 이용 가능합니다. 어느 쪽을 지정한 경우라도, C로 실장된 컴포넌트에 있어서는 main 함수의 제2 인수에 const char* 형의 문자열로 전달되므로, 이를 표준 라이브러리의 sscanf로 해석하거나 직접 문자열로 취급하거나 하는 것이 가능합니다.

AI 번역본

```
static bool mainloop(HIVE_COMPONENT comp, const char *param, HIVE_STATUS status) {
  int val1, val2;
  sscanf(param, "%d,%d", &val1, &val2); // perform parameter as a number string

  // some process using 'val1' and 'val2'.
}
```

JSON 형식의 파라미터를 다루는 경우도, param에 문자열로 JSON 형식이 격납되므로 이를 파싱할 필요가 있습니다. JSON의 취급은 본 시스템의 API로서는 준비되어 있지 않으므로, 범용적인 라이브러리 사용을 검토
하십시오.

참고

- json-c
- jansson

6.2.4.4 커스텀 컴포넌트용 API

본 항에서 소개한 함수 외에도 다수의 API 함수가 준비되어 있습니다. 상세는 부록의 커스텀 컴포넌트 C-API 레퍼런스를 참조하십시오.

6.2.4.5 커스텀 컴포넌트를 사용한다

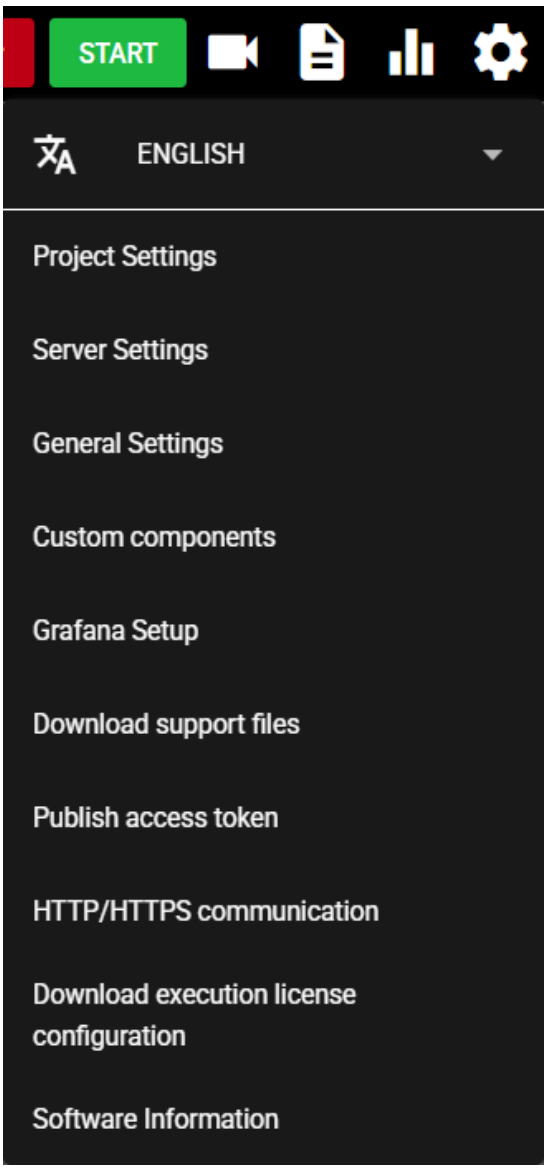
여기서는 작성한 커스텀 컴포넌트를 SpeeDBee Synapse에서 사용하는 방법에 대해 설명합니다.

6.2.4.5.1 등록

다음 순서로 빌드한 커스텀 컴포넌트를 SpeeDBee Synapse에 등록할 수 있습니다.

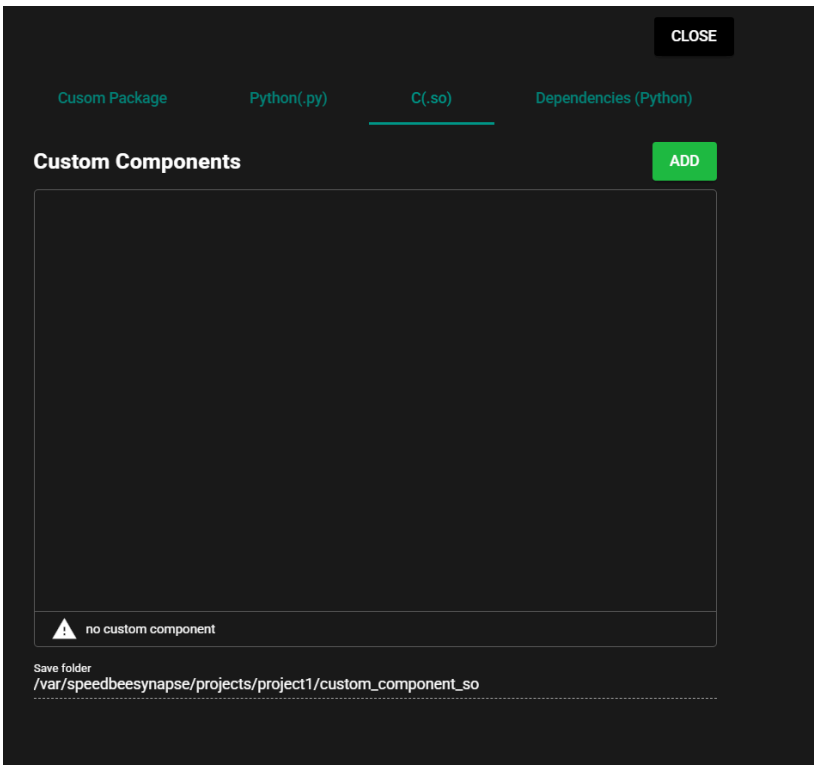
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

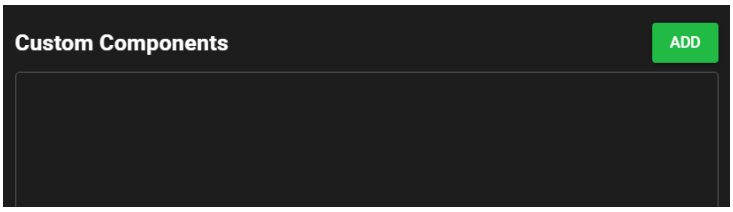


2. 다이얼로그 상단 탭의「C(.so)」를 선택합니다.

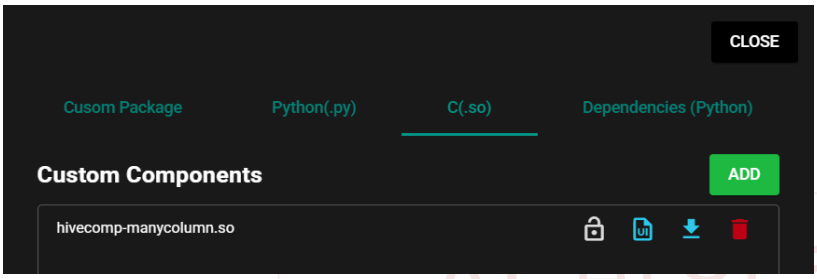
AI 번역본
-For Reference Only-



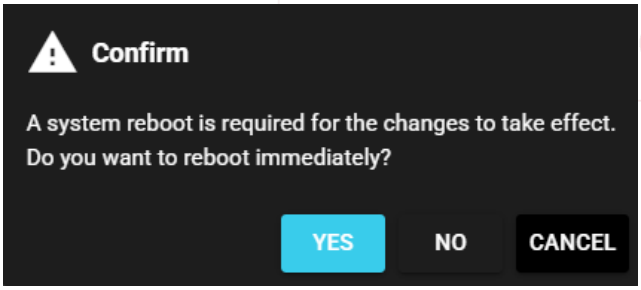
3. 「 추가 」를 누르고, 빌드가 완료된 커스텀 컴포넌트의 so 파일을 등록합니다.



4. 「 닫기 」를 누릅니다.

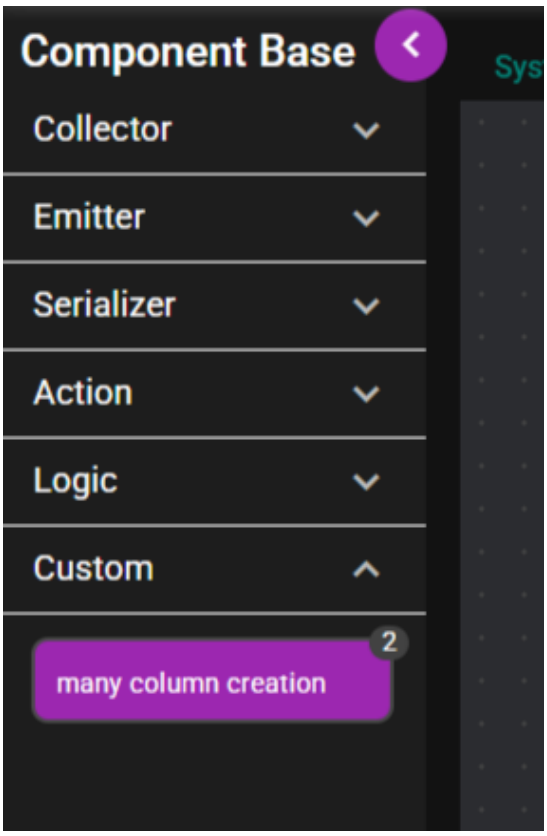


5. 확인 다이얼로그에서 「 예 」를 선택하고, 변경을 반영하기 위해, 시스템을 재기동시킵니다.



6. 좌측 메뉴에, 등록한 커스텀 컴포넌트가 반영됩니다.

AI 학습용
-ence Only-



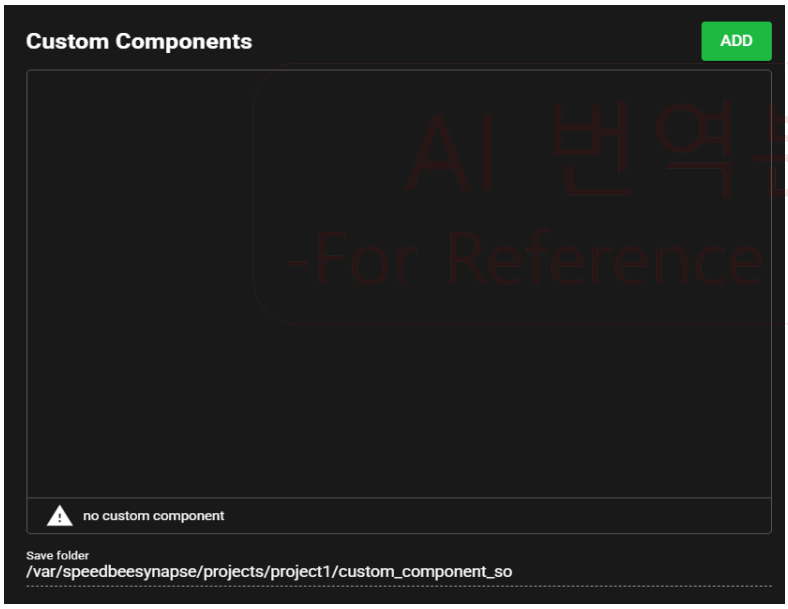
파일의 저장 위치

등록한 커스텀 컴포넌트의 파일은, 다음 디렉터리에 저장됩니다.

Linux

/var/speedbeesynapse/custom_component_so

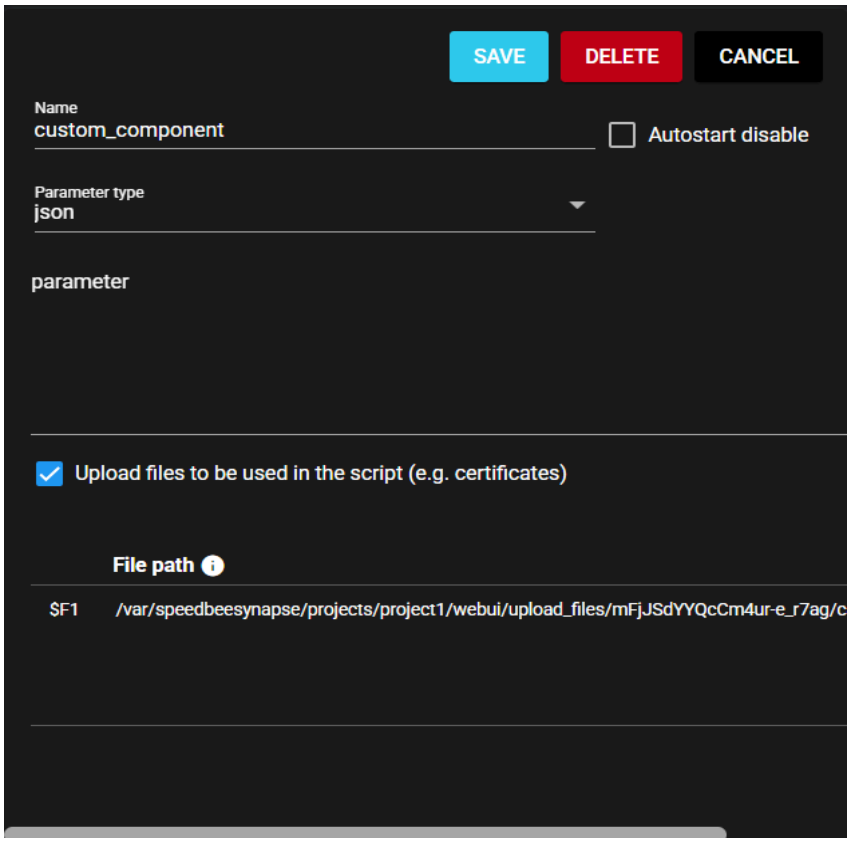
저장 위치 디렉터리는, 화면 항목 「 저장 위치 」에 표시됩니다.



AI 번역본
-For Reference Only-

6.2.4.5.2 설정 화면

커스텀 컴포넌트의 설정 화면의 항목은 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우 ON
파라미터 타입	파라미터의 종류를 다음 중에서 선택 ・STRING：문자열 ・JSON：JSON 문자열
파라미터	컴포넌트 파라미터를 입력
스크립트에서 사용할 파일을 업로드한다(증명서 등)	커스텀 컴포넌트의 처리에서 사용하는, 증명서 등의 파일을 업로드하는 경우 ON

AI 번역본
-For Reference Only-

항목	설명
파일 추가	누르면, 스크립트에서 사용할 파일을 업로드할 수 있습니다. ※새롭게 추가된 파일은, 컴포넌트의 설정을 저장했을 때에 업로드됩니다.
파일 경로	업로드한 파일의 저장 위치 ※파라미터에서 파일 경로를 사용하는 경우, 변수를 사용할 수 있습니다. 상세는, 표제의「파일 경로」의 오른쪽 아이콘을 클릭하면 확인할 수 있습니다.

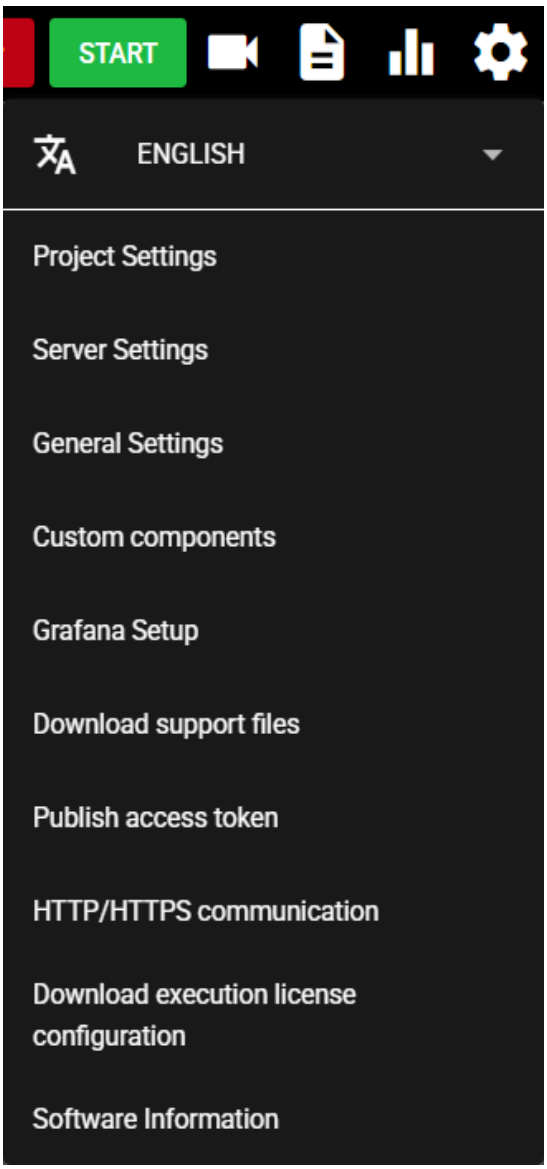
설정 화면의 파라미터 입력 부분에, 사용자가 정의한 화면 항목을 표시하도록 변경할 수 있습니다. 상세는
「커스텀 UI 작성」을 참조하십시오.

6.2.4.5.3 삭제

다음 순서로, 등록한 커스텀 컴포넌트를 SpeeDBee Synapse에서 삭제할 수 있습니다.

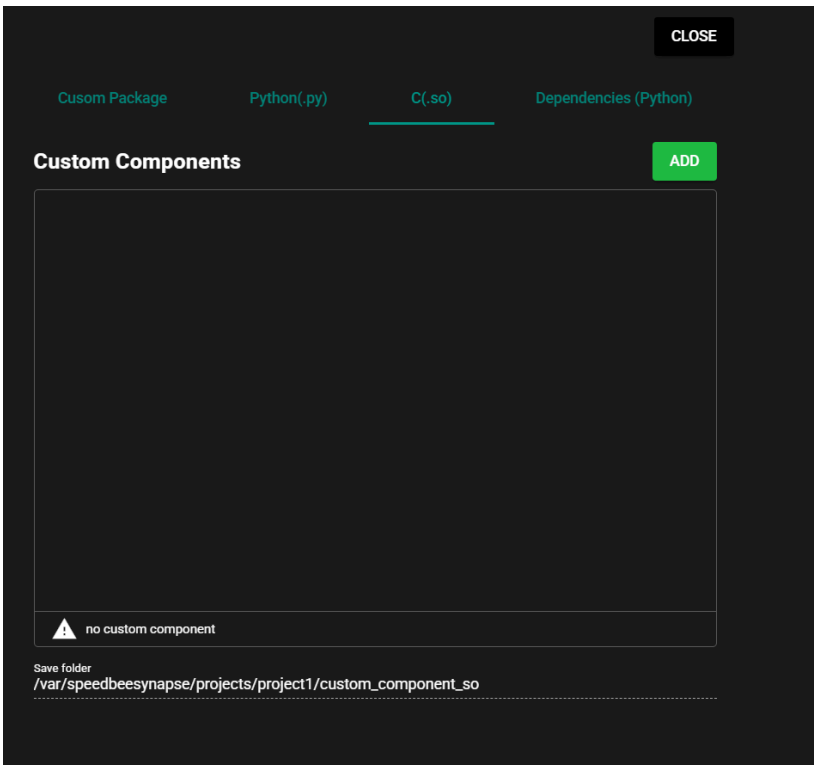
1. 설정 메뉴 아이콘을 누르고,「커스텀 컴포넌트」를 선택합니다.

AI 번역본
-For Reference Only-

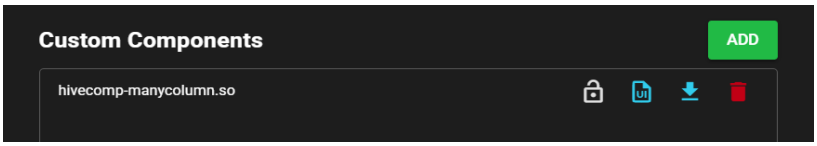


2. 다이얼로그 상단 탭의 「 C(.so) 」를 선택합니다.

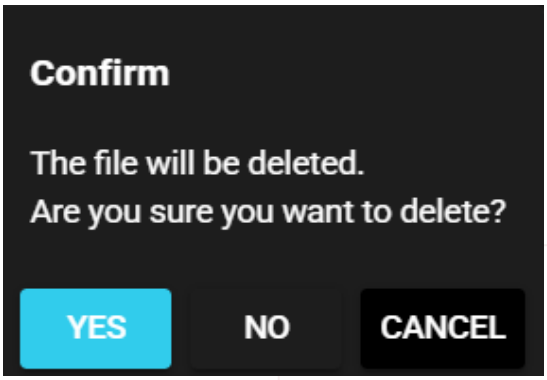
AI 번역본
-For Reference Only-



3. 커스텀 컴포넌트의「 휴지통 아이콘 (삭제) 」을 누릅니다.



4. 확인 다이얼로그에서「 예 」를 선택합니다.



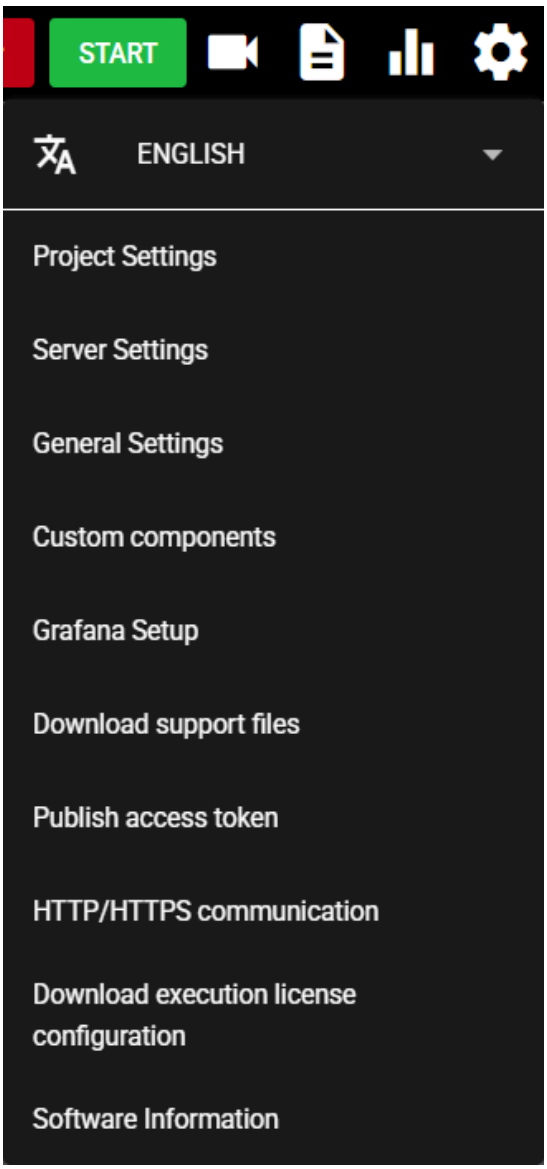
5. 커스텀 컴포넌트가 삭제됩니다.

번역본
-For Reference Only-

6.2.4.5.4 다운로드

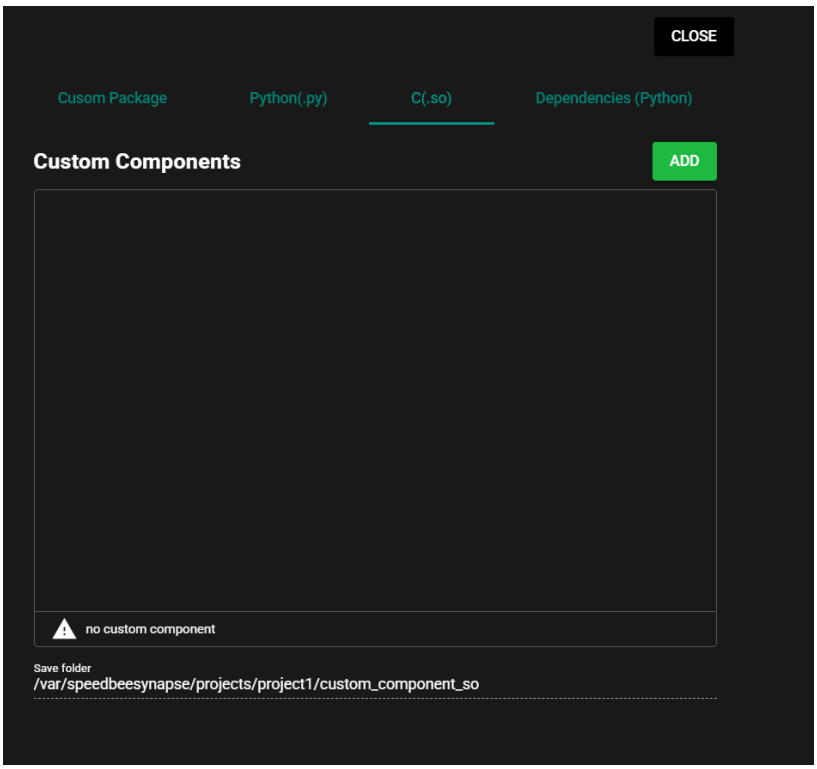
다음 순서로, 등록한 커스텀 컴포넌트를 다운로드할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

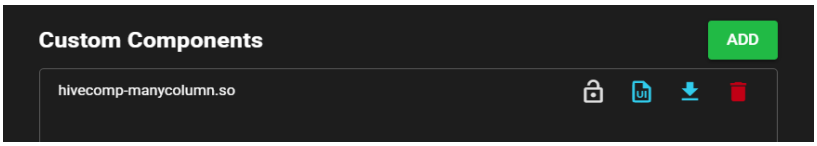


2. 다이얼로그 상단 탭의「C(.so)」를 선택합니다.

AI 번역본
-For Reference Only-



3. 커스텀 컴포넌트의「다운로드 아이콘（취득）」을 누릅니다.



4. 커스텀 컴포넌트의 파일이 다운로드됩니다.

6.2.4.5.5 보호의 부여

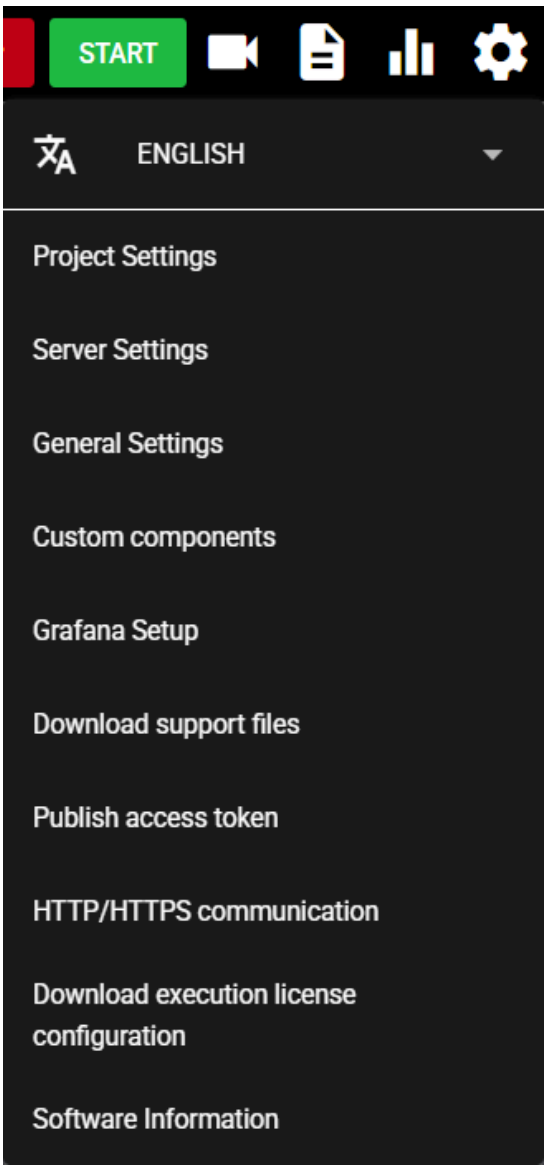
등록한 컴포넌트에는 보호를 걸 수 있습니다.
보호된 컴포넌트는 다음 조작을 수행할 수 없게 되므로, 의도하지 않은 갱신을 방지할 수 있습니다.

- 등록
- 삭제
- 다운로드
- 커스텀 UI의 설정

AI 번역본
-For Reference Only-

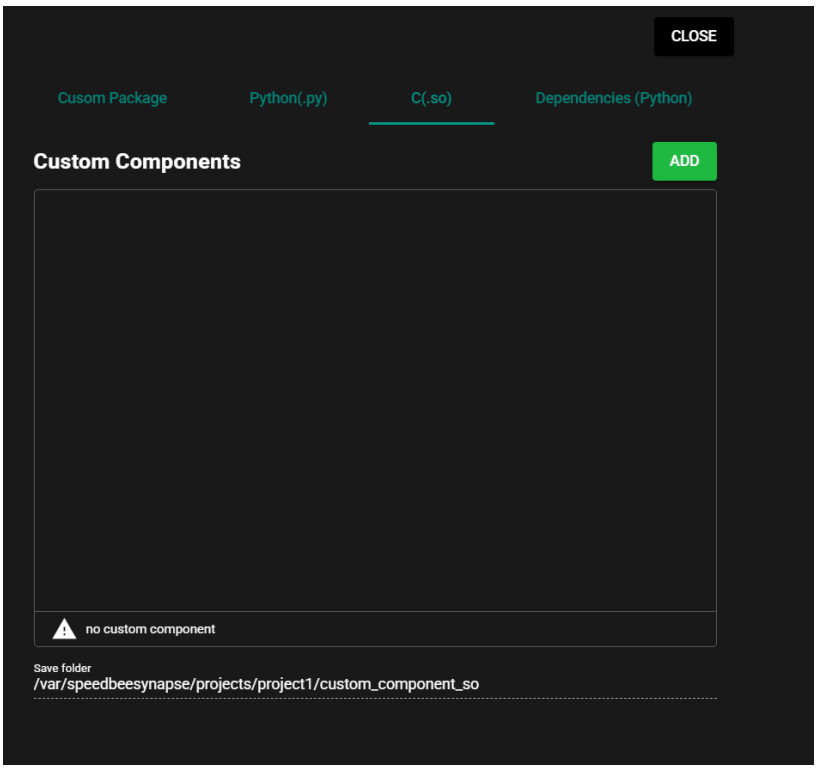
다음 순서로, 커스텀 컴포넌트에 보호를 걸 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「커스텀 컴포넌트」를 선택합니다.

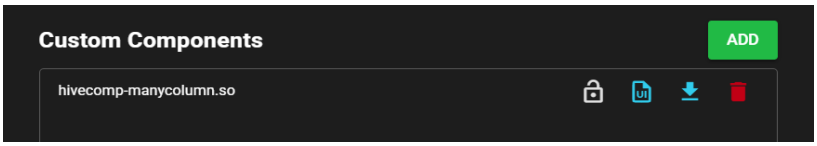


2. 다이얼로그 상단 탭의「C(.so)」를 선택합니다.

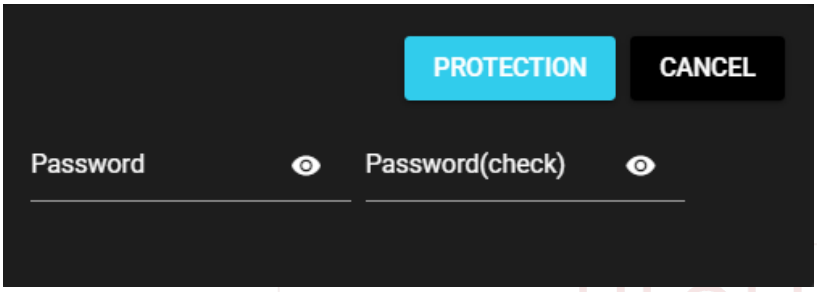
AI 번역본
-For Reference Only-



3. 커스텀 컴포넌트의「자물쇠 아이콘（보호）」을 누릅니다.



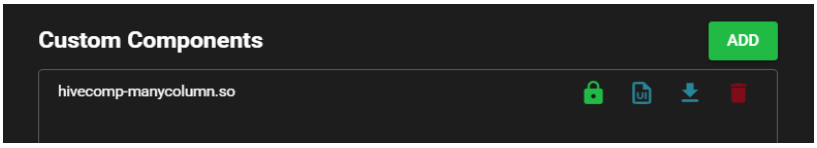
4. 보호 해제할 때에 필요하게 되는 패스워드를 입력하고,「보호」를 누릅니다.



AI 번역본

> 경고
>
> - 패스워드는 복원할 수 없는 형식으로 저장되며, 한 번 설정하면 보호를 해제할 때까지 재설정할 수 없습니다. 보호
> - 설정한 패스워드를 잊은 경우, 해당 커스텀 컴포넌트는 조작 불능이 됩니다.
>
> 패스워드의 취급에는 충분히 주의하십시오.

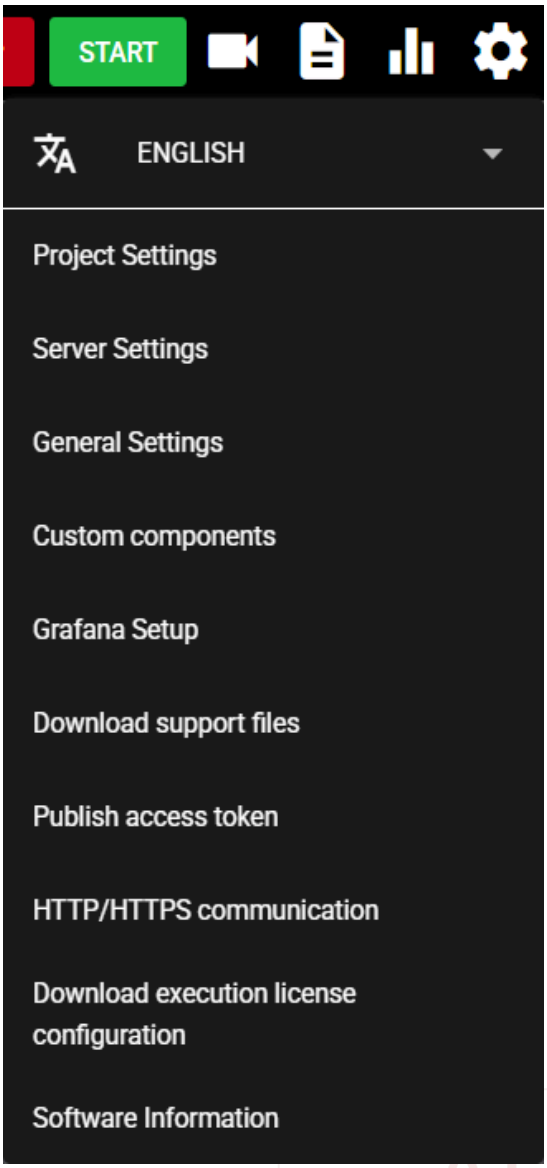
5. 커스텀 컴포넌트가 보호 상태가 됩니다.



6.2.4.5.6 보호의 해제

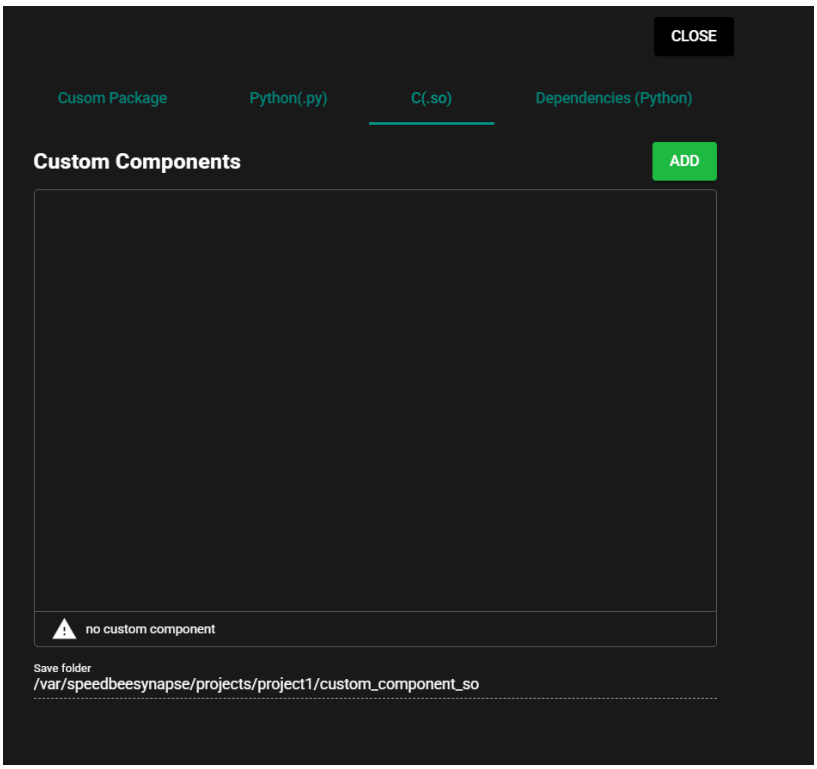
다음 순서로, 보호된 커스텀 컴포넌트의 보호를 해제할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

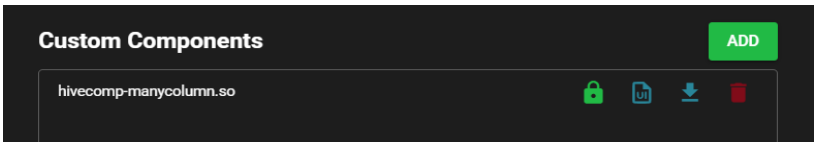


2. 다이얼로그 상단 탭의「 C(.so) 」를 선택합니다.

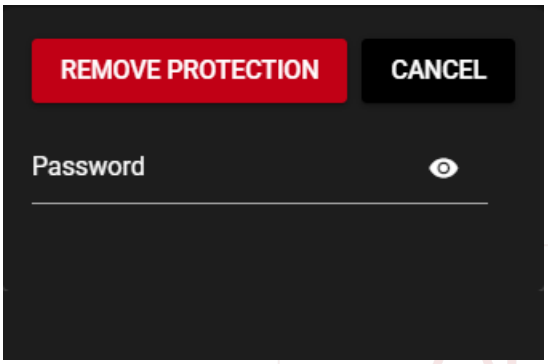
AI 번역본
-For Reference Only-



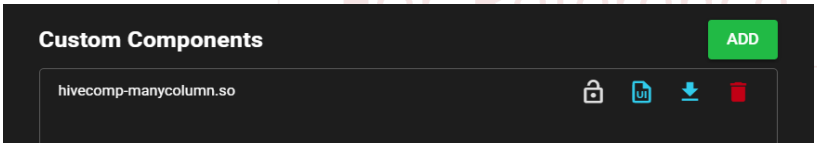
3. 커스텀 컴포넌트의「자물쇠 아이콘（보호 해제）」를 누릅니다.



4. 보호를 걸 때에 설정한 패스워드를 입력하고,「보호 해제」를 누릅니다.



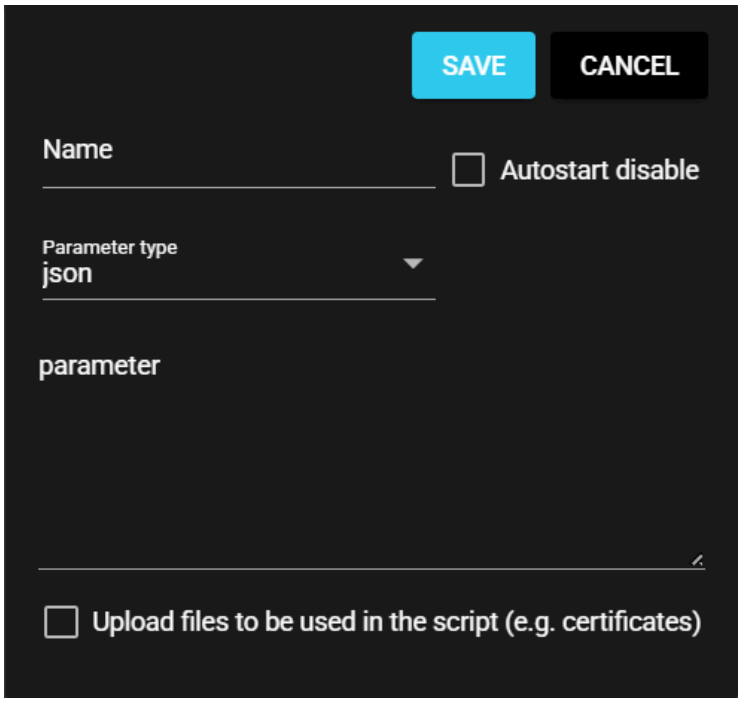
5. 커스텀 컴포넌트의 보호가 해제됩니다.



AI 번역본
For Reference Only-

6.2.5 커스텀 UI 작성

커스텀 컴포넌트의 설정 화면은 디폴트에서는, 파라미터로서 JSON, 또는 문자열을 자유롭게 입력할 수 있게 되어 있습니다. 그러나, 이래서는 작성자와 이용자가 다른 경우, 파라미터에 설정할 항목에 대해
조사하려면 커스텀 컴포넌트의 소스 코드를 확인할 필요가 있어, 수고가 듭니다.



그 때문에, 커스텀 컴포넌트는 설정 화면의 파라미터의 입력란을 커스터마이즈할 수 있습니다. 파라미터의 입력란으로서, 어떠한 항목을 배치할지 사용자가 정의하여, 이용자가 입력할 항목을 알기 쉽게 할 수 있습니다.

6.2.5.1 커스텀 UI의 작성 방법

커스텀 UI는 이하의 방법으로 작성할 수 있습니다. 커스텀 UI에는 이하의 2 가지 형식이 있으며, 각각 작성 방법이 다릅니다.

AI 번역본
For Reference Only

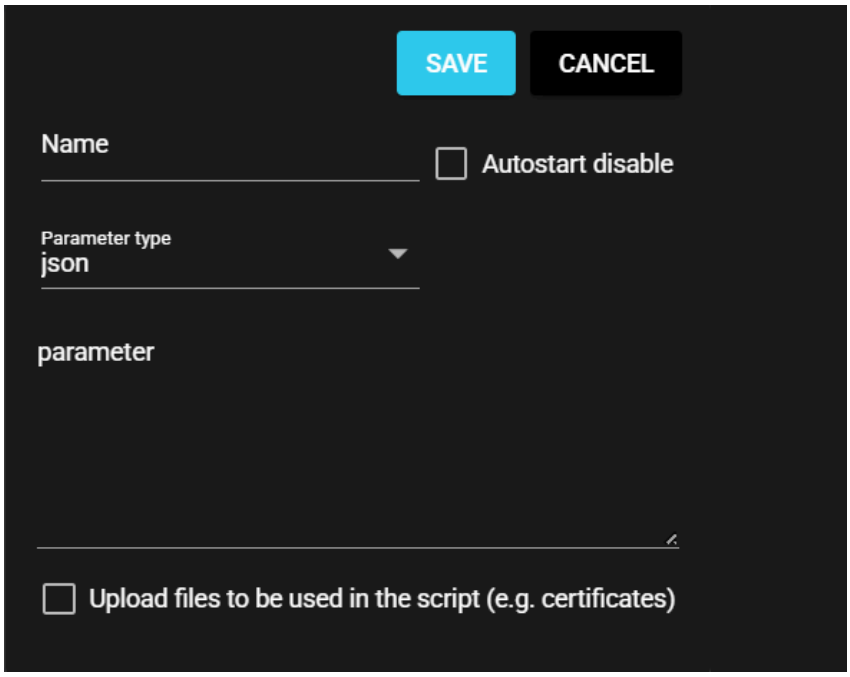
- Synapse 커스텀 UI용 JSON

  컴포넌트에 대한 설정 항목을, JSON 파일에 기재하는 방법입니다. 작성하는 것은 JSON 파일뿐이며,
  수정이 필요할 때도 JSON 파일을 변경하는 것만으로 간편하게 수행할 수 있습니다. 그 대신, Synapse가 제공하는 기능의 범위에서만 설정 항목은 정의할 수 있어, 복잡한 UI의 작성은 할 수 없습니다.

- Synapse 커스텀 UI용 HTML

  컴포넌트에 대한 설정 항목을, HTML과 JavaScript로 작성합니다.
  HTML 파일의 내용은 사용자가 원하는 대로 작성할 수 있으며, JavaScript, CSS, 이미지 등도 활용함으로써 복잡한 UI를 실장할 수 있습니다. 다만, 작성에는 기본적인 HTML/JavaScript의 지식이 필요하게 됩니다. 또한, 단순한 실장이면 Synapse 본체의 화면과는 룩＆필이 달라져 버립니다.

커스텀 UI로서 작성할 수 있는 범위는 이미지의 빨간 테두리 내에 있는 「 파라미터 」에 관련된 요소뿐이며, 바깥쪽 요소는 편집할 수 없습니다.



- 「 저장 」 버튼

  입력한 컴포넌트의 설정을 저장합니다.

- 「 취소 」 버튼

  설정 다이얼로그를 닫습니다.

- 명칭, 자동 기동 무효

  컴포넌트 가동에 필수인 설정 항목입니다.

- 스크립트에서 사용할 파일을 업로드한다(증명서 등)

  컴포넌트 실행 시에 필요한 파일 (증명서 등) 을 관리합니다.

AI 번역본
-For Reference Only-

6.2.5.2 Synapse 커스텀 UI용 JSON 파일의 작성

여기에서는, 커스텀 UI용 JSON 파일의 작성 방법에 대해 설명합니다.

6.2.5.2.1 작성 방법

다음 순서로, 커스텀 UI용 JSON 파일을 작성할 수 있습니다.

1. 텍스트 에디터에서 빈 json 파일을 작성합니다.
2. json 파일 내에, "parameter_defines"라는 배열을 정의합니다.
3. "parameter_defines"의 안에, 커스텀 컴포넌트의 화면에 표시할 입력 요소를 정의합니다. 그 밖에, "window_size", "layout" 등 필요에 따라 정의하십시오.

정의 파일(json)

```
{
  "window_size": "auto",
  "parameter_defines": [
    {
      // 입력 요소1
    },
    {
      // 입력 요소2
    },
    {
      // 입력 요소3
    }
    ...
  ]
}
```

화면에 표시할 입력 요소는, 이하의 항목에 의해 정의합니다.

항목	값의 종류	필수	설명
window_size	string	×	윈도우의 사이즈 { "window_size" : "small" }와 같이 지정 ○ auto/지정 없음 - min-width: "400px" ○ small - min-width: "33%" ○ middle - min-width: "50%" ○ large - min-width:"80%"
parameter_defines	object	○	항목의 부모 요소 （最上位階層から1つ下の階層まで入れ子指定可能 参考）
name	string	○	入力した値を設定ファイルに保持するときのキー
label	string	○	画面に表示される項目名
type	string	○	入力する値の種類を、以下の中から選択 string：文字列 number：整数 real：実数 select：選択肢
max	number	○	入力の上限 ※type が number・real の場合のみ有効

AI 번역본
-For Reference Only-

항목	값의 종류	필수	설명
min	number	×	입력의 하한 ※type이 number・real인 경우에만 유효
select	array	×	선택지 ※1 type이 select인 경우에만 유효 ※2 선택지는, "label"과 "value"의 프로퍼티를 갖는 오브젝트 배열에 의해 정의하십시오. 예：[{ "label":"option1", "value":1}, { "label":"option2", "value":2}]
default	any	×	디폴트값
require	bool	○	입력 필수 여부
array	bool	×	배열화할지 여부
layout	object	×	각 항목의 폭 지정 참고 { "layout" : { "col" : 2 }}와 같이 col에 1～12를 지정 생략 시는 디폴트 6 array,object 생략 시는 디폴트 12

6.2.5.2.2 샘플

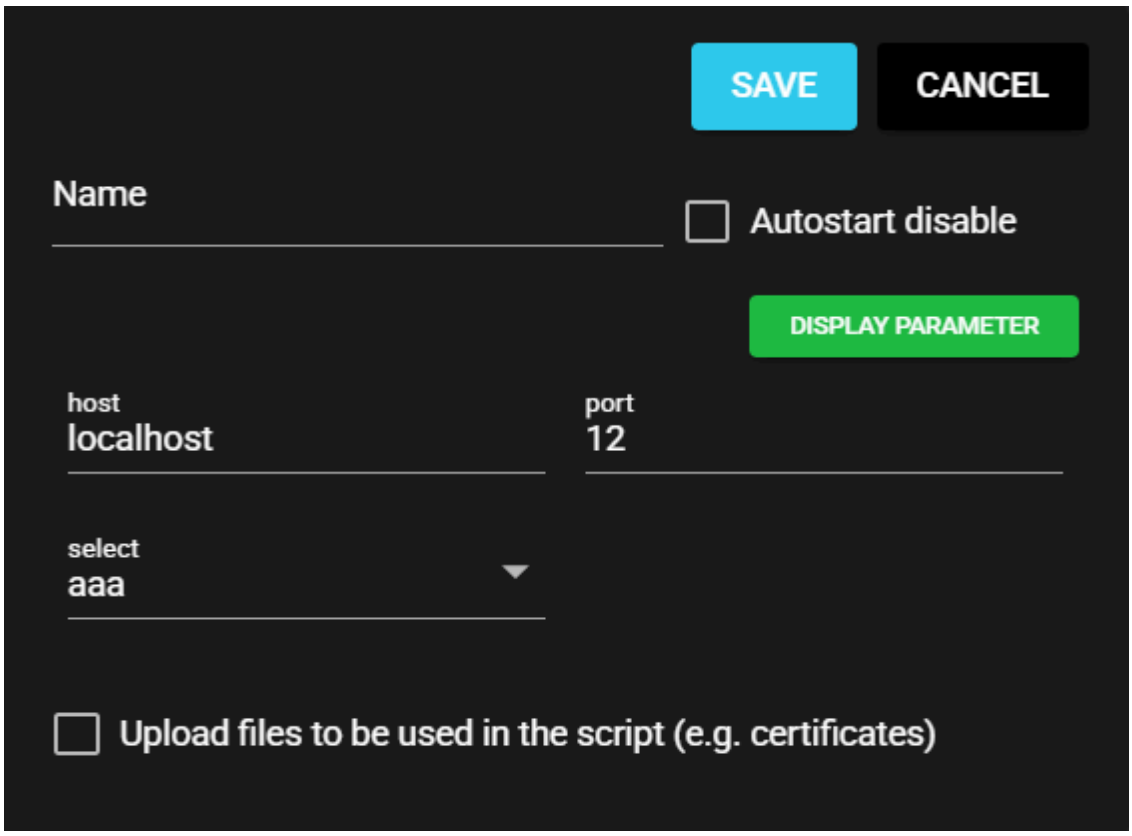
커스텀 UI 정의 파일의 샘플은 다음과 같습니다.

AI 번역본
-For Reference Only-

```
{
  "parameter_defines": [
    {
      "name": "host",
      "label":"호스트",
      "type": "string",
      "default": "localhost",
      "require": true,
      "max":256
    },
    {
      "name": "port",
      "label":"포트",
      "type": "number",
      "max": 65535,
      "min": 1,
      "default": "12",
      "require": true
    },
    {
      "name":"select",
      "label":"선택",
      "type":"select",
      "select":[
        { "label":"aaa", "value":1},
        { "label":"bbb", "value":2}
      ],
      "default": 1,
      "require": false
    }
  ]
}
```

다음과 같은 레이아웃이 됩니다.

AI 번역본
-For Reference Only-



6.2.5.2.3 중첩을 사용한 샘플

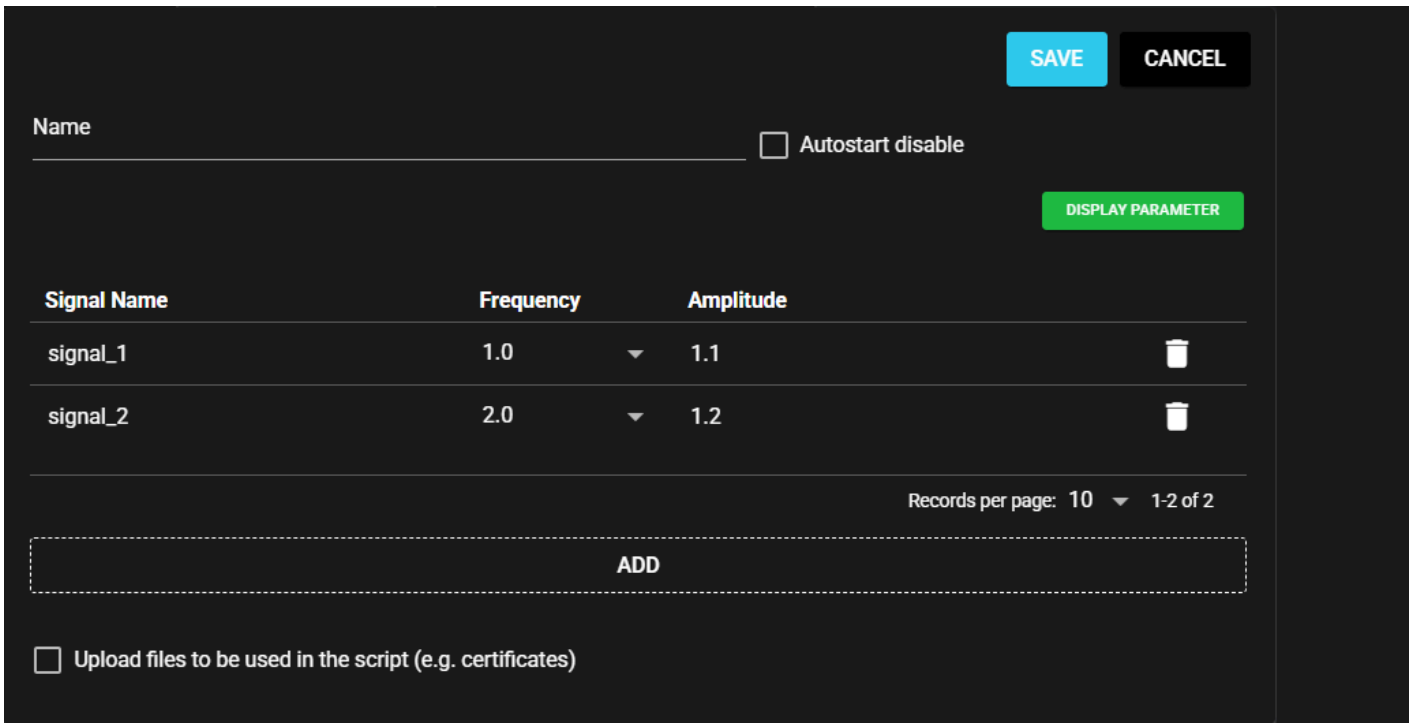
커스텀 UI 정의 파일의 중첩을 사용한 샘플은 다음과 같습니다.

AI 번역본
-For Reference Only-

```
{
  "window_size": "middle",
  "parameter_defines": [
    {
      "name": "nested",
      "label": "중첩",
      "type": "object",
      "array": true,
      "parameter_defines": [
        {
          "name": "name",
          "label": "파명",
          "type": "string",
          "require": true,
          "max": 32,
          "default": "signal_5"
        },
        {
          "name": "freq",
          "label": "주파수",
          "type": "select",
          "require": true,
          "select": [
            { "label": "1.0", "value": 1.0 },
            { "label": "2.0", "value": 2.0 }
          ],
          "default": 1.0
        },
        {
          "name": "amp",
          "label": "진폭",
          "type": "real",
          "require": true,
          "max": 100.0,
          "min": 0.1,
          "default": 1.0
        }
      ],
      "require": false,
      "default": [
        {
          "name": "signal_1",
          "freq": 1,
          "amp": 1.1
        },
        {
          "name": "signal_2",
          "freq": 2,
          "amp": 1.2
        }
      ]
    }
  ]
}
```

AI 번역본
-For Reference Only-

다음과 같은 레이아웃이 됩니다.



6.2.5.2.4 항목의 폭을 지정한 샘플

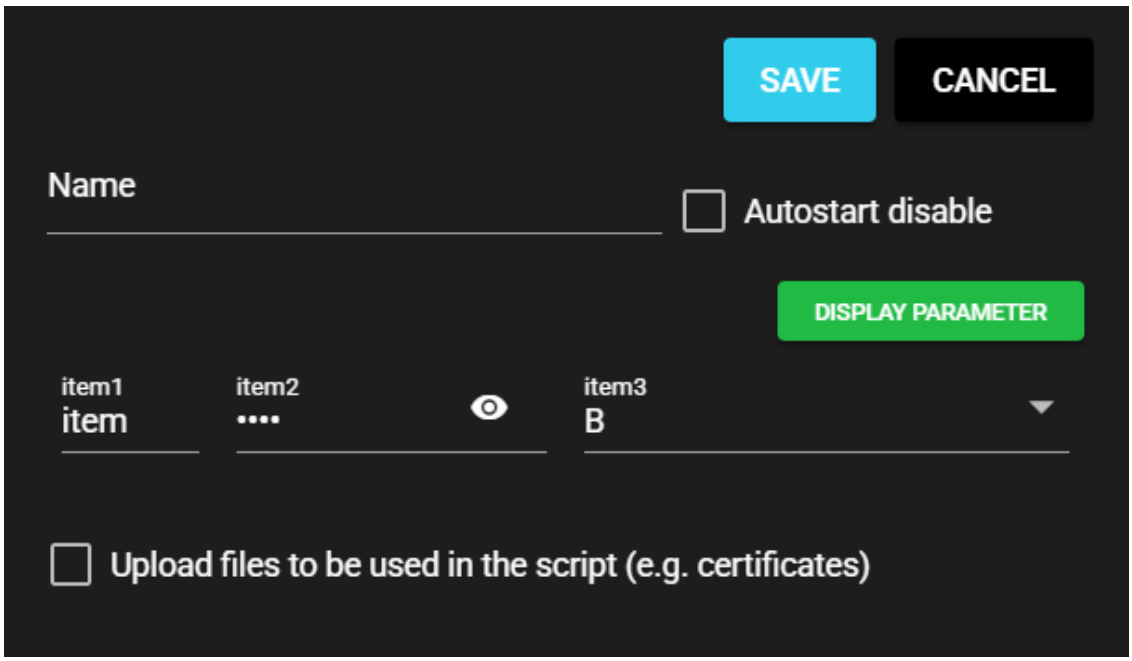
커스텀 UI 정의 파일의 항목 폭을 지정한 샘플은 다음과 같습니다.

AI 번역본
-For Reference Only-

```
{
  "parameter_defines": [
    {
      "name": "item1",
      "label": "항목1",
      "type": "string",
      "require": false,
      "max": 32,
      "default": "항목 1 ",
      "layout": {
        "col": 2
      }
    },
    {
      "name": "item2",
      "label": "항목2",
      "type": "real",
      "require": false,
      "max": 32,
      "default": "12.3",
      "mask": true,
      "layout": {
        "col": 4
      }
    },
    {
      "name": "item3",
      "label": "항목3",
      "type": "select",
      "require": false,
      "max": 32,
      "default": 2,
      "select": [
        { "label": "A", "value": 1},
        { "label": "B", "value": 2},
        { "label": "C", "value": 3}
      ],
      "layout": {
        "col": 6
      }
    }
  ]
}
```

다음과 같은 레이아웃이 됩니다.

-For Reference Only-



6.2.5.3 Synapse 커스텀 UI용 HTML 작성, 적용 방법

여기에서는, 커스텀 UI를 HTML과 JavaScript로 작성하는 방법에 대해 설명합니다.

6.2.5.3.1 작성 방법

HTML 형식에서의 커스텀 UI를 이용하려면, 아래의 환경이 필요합니다. 먼저 이 환경을 준비하십시오.

- 커스텀 컴포넌트 개발 환경(SCCDE)

이하의 어느 하나의 커맨드로, HTML 형식의 커스텀 UI를 포함하는 샘플 컴포넌트를 추가할 수 있습니다.

```
$ sccde add -t collector -u html
$ sccde add -t emitter -u html
```

실제 HTML 파일은, parameter_ui 아래의 컴포넌트 개별의 디렉터리 내에 배치됩니다. 작성된 샘플을 기반으로, html, js, css 파일을 커스터마이즈하여 이용하십시오.

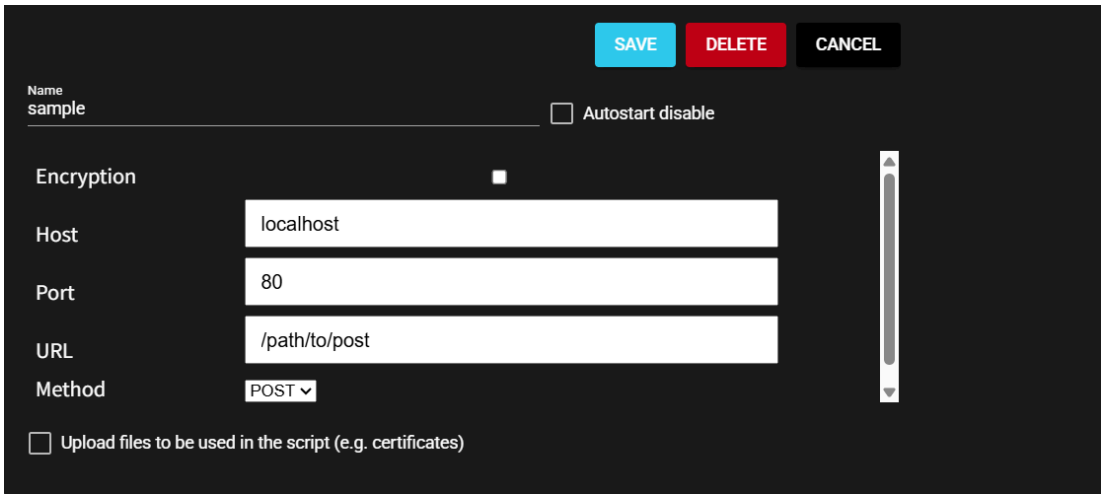
AI 번역본
For Reference Only-

6.2.5.3.2 HTML 형식의 상세

- HTML 파일(index.html)

  통상의 HTML과 마찬가지로, HTML 태그, BODY 태그를 사용하여 정의할 수 있습니다. 이 파일에서 정의한 내용이, 커스텀 컴포넌트의 파라미터 설정 다이얼로그의 문자열 입력부 대신에 표시됩니다.

표시 예



- JavaScript 파일(index.js)

  다이얼로그 표시 시, 및, 파라미터 저장 시에, 소정의 함수 호출이 필요하기 때문에, JavaScript로 그 함수를 정의할 필요가 있습니다. 이 JavaScript 파일은 HTML 파일 내의 <script> 태그로 읽어 들이도록 해 주십시오.

  상세는 후술

- scc-info.json 파일

  커스텀 컴포넌트에서 커스텀 UI(HTML)를 사용하는 것을 시스템에 전달하기 위해, 대상 컴포넌트의 아래의 정의가 필요합니다.

프로퍼티	설명
parameter-ui-type	'html'을 지정
parameter-ui-size	파라미터 다이얼로그의 화면 사이즈 지정. 'small','middle','large','auto'에서 선택 가능. (디폴트는 'middle')
parameter-ui	HTML 파일이나 JavaScript 파일을 포함하는 디렉터리로의 경로를 지정

-For Reference Only-

6.2.5.3.3 JavaScript 파일의 기술

이후에서는, 샘플의 index.js 의 소스 코드를 예로, 커스텀 UI를 HTML로 실장할 때의 JavaScript에서의 처리 방법을 설명합니다.

JavaScript로 Synapse의 정보를 취급할 수 있도록 하기 위해, Synapse 전용의 API 오브젝트를 제공하고 있습니다. 이 API 오브젝트는, 'window.customUiLib`로서 참조할 수 있습니다.

이 오브젝트에 특정 프로퍼티명으로 함수를 등록함으로써, Synapse 본체의 화면과 데이터의 주고받기가 가능하게 됩니다.

```
async function onLoad(editable, param, isNewComponent, componentId) {
  :
}
async function onSave() {
  :
}

// 콜백 함수의 등록
window.customUiLib.onSave = onSave;
window.customUiLib.onLoad = onLoad;
 :
```

그 밖에도 Synapse의 정보를 취득할 수 있는 API가 있습니다.

이들 API의 상세에 대해서는, 커스텀 UI JS-API 레퍼런스를 참조하십시오.

6.2.5.3.4 파라미터 다이얼로그가 표시되었을 때에 실행되는 함수

커스텀 컴포넌트의 파라미터 설정 다이얼로그가 표시될 때에는, 커스텀 UI의 JavaScript에서 등록한 `onLoad()` 함수가 호출됩니다. 이 함수의 인수로서, 이하의 정보가 전달되므로, 필요에 따라 html 내의 요소에 반영하십시오.

- 다이얼로그의 편집 가부

  Run Mode에서는 다이얼로그의 입력 항목을 편집 불가로 하기 위해, 인수 editable 에 false 가 전달됩니다. 상세는 편집 제한의 제어를 참조하십시오.

- 설정 완료된 파라미터

  이미 작성 완료된 컴포넌트 파라미터를 편집할 때, 이전에 설정된 파라미터가 인수 param 에 전달됩니다.

- 신규 작성인가 갱신인가

  컴포넌트의 신규 작성 시에는 true , 갱신 시는 false 가 인수 isNewComponent 에 전달됩니다.

- 컴포넌트 ID

  컴포넌트의 ID가 인수 componentId 에 전달됩니다. 이 정보는 컴포넌트의 신규 작성 시에는 없습니다.

AI 번역본
For Reference Only-

6.2.5.3.5 파라미터 다이얼로그의 저장 시에 실행되는 함수

커스텀 컴포넌트의 파라미터 설정 다이얼로그에서 저장 버튼이 클릭되었을 때에는, 커스텀 UI의 JavaScript에서 등록한 `onSave()` 함수가 호출됩니다. 이 함수에서 저장해야 할 파라미터를 반환하도록 해 주십시오.

6.2.5.3.6 편집 제한의 제어

Synapse에는 Run Mode, Edit Mode의 두 가지 화면이 있으며, Run Mode에서는 데이터 수집 플로우의 운용을, Edit Mode에서는 데이터 수집 플로우의 설정 등을 수행할 수 있습니다.

통상의 컴포넌트나 JSON 파일에서 정의한 커스텀 UI는 Run Mode에서는 편집할 수 없도록 제한이 걸려 있습니다만, HTML로 실장한 커스텀 UI에서는 그 제한을 할 수 없습니다. 그 대신, 편집 가능 여부의 정보를 onLoad() 함수의 인수 editable 에서 받을 수 있으므로, 그것을 바탕으로 사용자가 편집할 수 없도록 실장해 주십시오.

기본적으로는 HTML 요소의 disable을 true로 설정하여 비활성으로 하면 문제없습니다.

```
async function onLoad(editable, param, isNewComponent, componentId) {
  // editable이 false이면 폼의 'host' 항목을 비활성으로 한다
  document.forms['form']['host'].disabled = !editable;
}
window.customUiLib.onLoad = onLoad;
```

참고 : HTML 속성: disabled

6.2.5.3.7 npm을 이용한 샘플

Synapse에서는, Vue 나 Quaser 를 이용하여 화면을 작성하고 있습니다.
순수한 HTML이나 CSS로는 위화감이 있는 화면이 되어 버립니다만, npm 을 이용함으로써, Vue 나 Quaser 같은 외부 패키지를 이용한 UI를 작성할 수 있습니다. 여기까지의 설명에서 이용한 샘플에서는 순수한 HTML, Javascript, CSS만을 이용하고 있었습니다만, Vue, Quaser를 이용하도록 한 샘플도 제공하고 있습니다.

Vue , Quaser 를 이용한 샘플

샘플을 다운로드하셨으면, 아래의 커맨드를 실행함으로써 html이나 js 파일이 dist 디렉터리 아래에 생성됩니다. 이들 파일을 통상의 html과 마찬가지로 SCCDE 환경 내에 배치하여 사용 가능합니다.

```
$ ls
sample_custom_ui_vue.zip
$ unzip sample_custom_ui_vue.zip
$ cd sample_custom_ui_vue
$ npm install
$ npm run build
$ ls dist
assets  index.html
```

> **npm의 이용**
>
> npm 을 이용하려면 Node.js 를 설치할 필요가 있습니다. 설치 방법을 포함한 Node.js에 대한 상세는 Node.js 공식 사이트를 참조하십시오
> Node.js 공식 사이트 : https://nodejs.org/ja

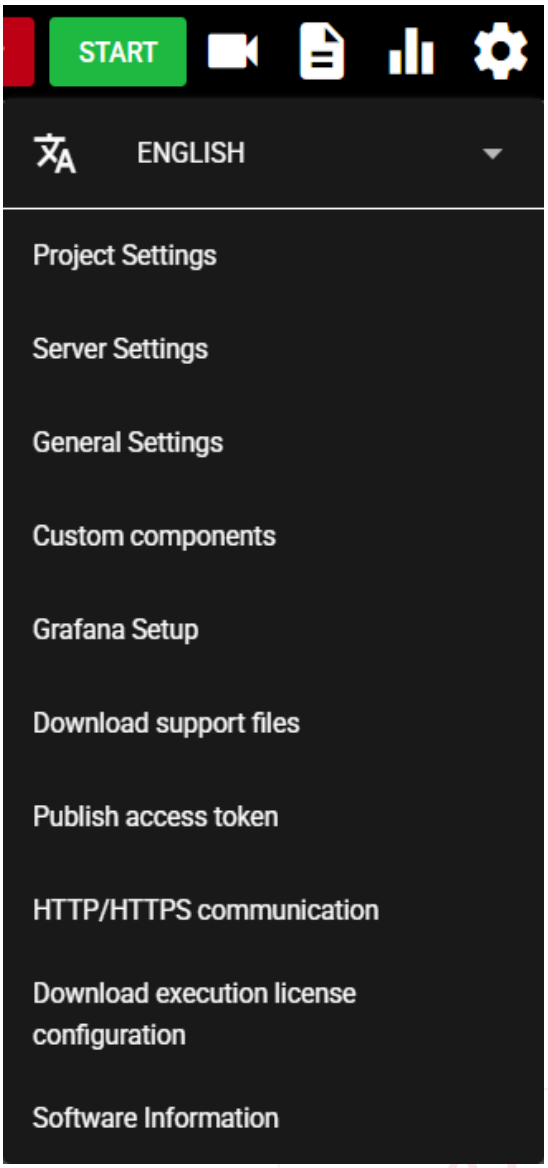
6.2.5.4 Python 커스텀 컴포넌트로의 커스텀 UI(JSON)의 적용

「 Synapse 커스텀 UI용 JSON 파일의 작성 」에서 작성한 정의 파일을 등록함으로써, Python 커스텀 컴포넌트에 커스텀 UI를 설정할 수 있습니다.

6.2.5.4.1 등록

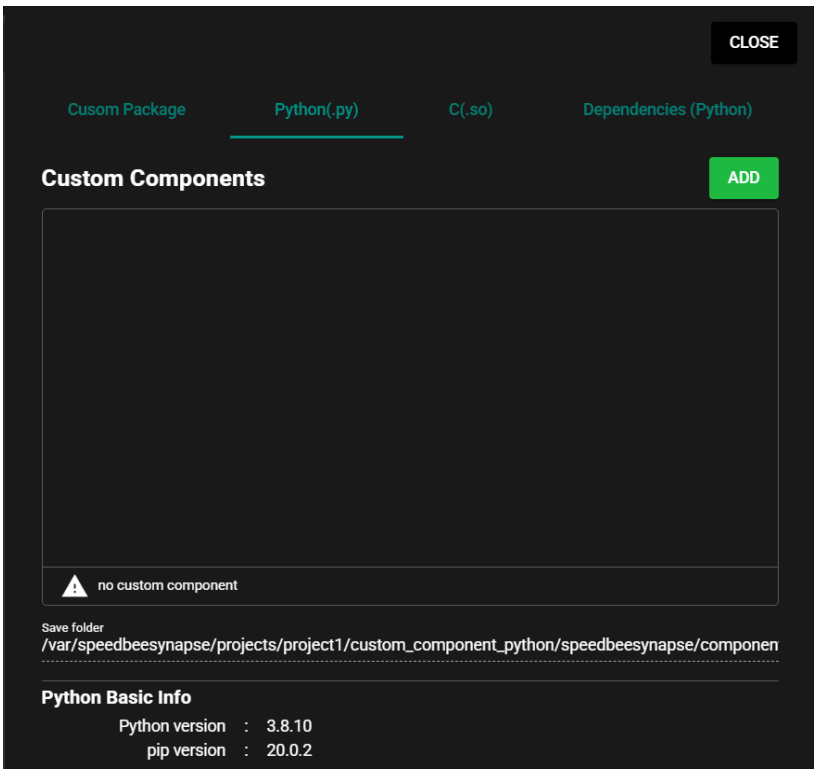
다음 순서로, 커스텀 UI를 등록할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

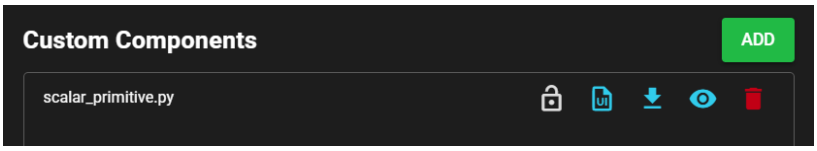


2. 다이얼로그 상단 탭의「 Python(.py) 」를 선택합니다.

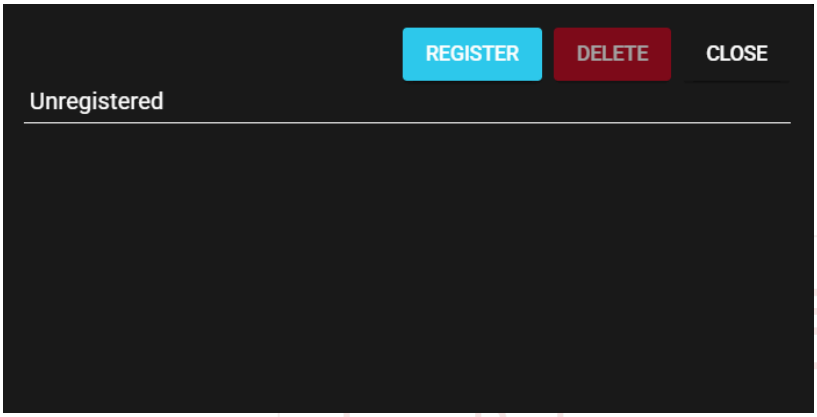
AI 번역본
-For Reference Only-



3. 임의의 커스텀 컴포넌트의「UI 아이콘（UI 설정）」을 누릅니다.

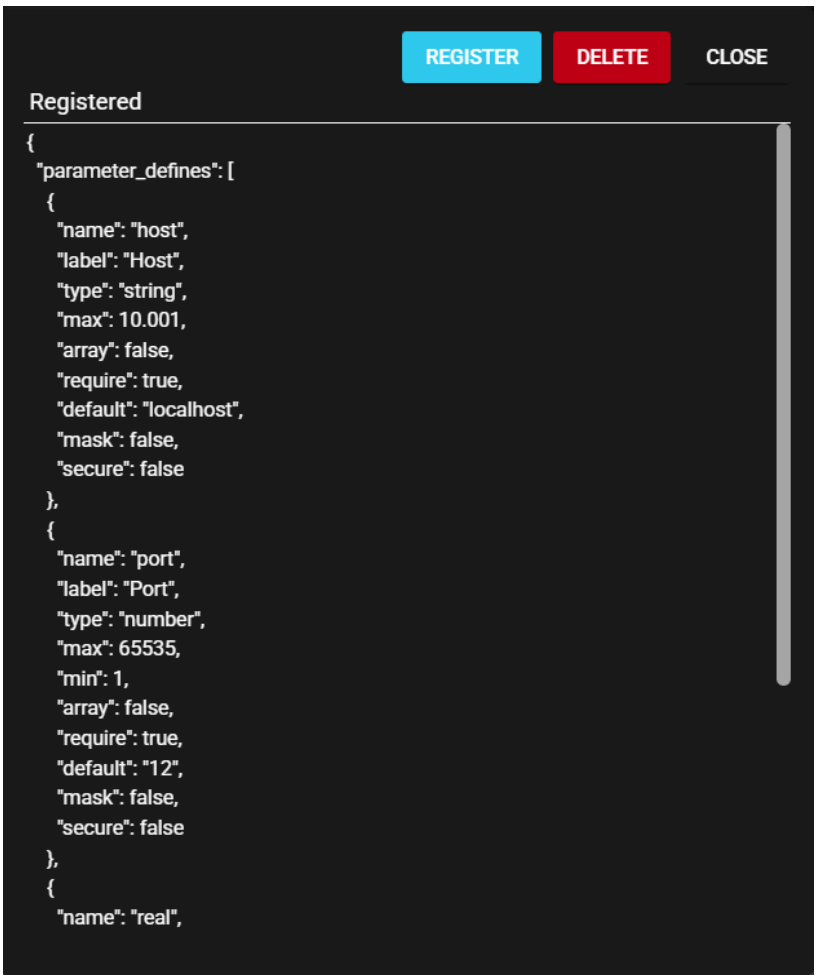


4.「등록」을 누르고, 정의 파일을 등록합니다.

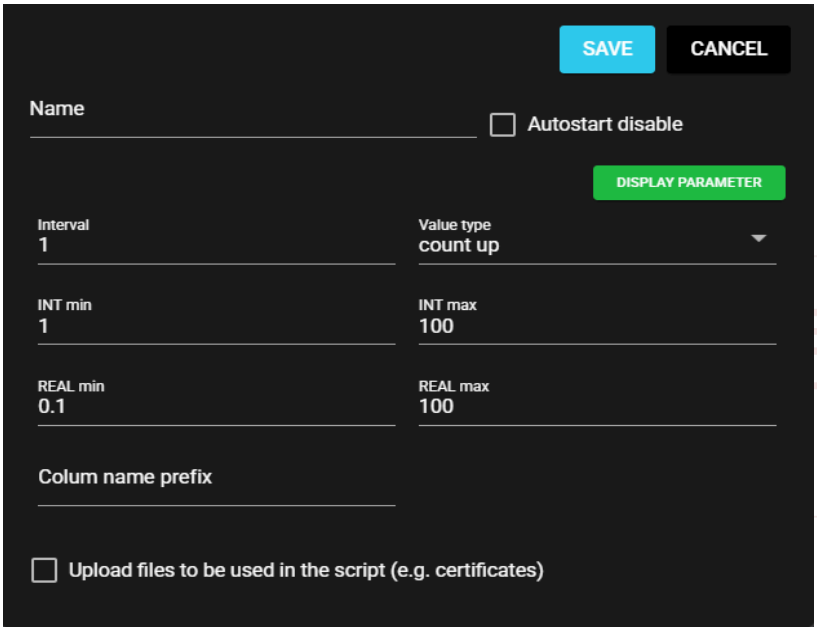


5.「등록 완료」라는 문구와, 정의 파일의 내용이 표시됩니다.

-For Reference Only-

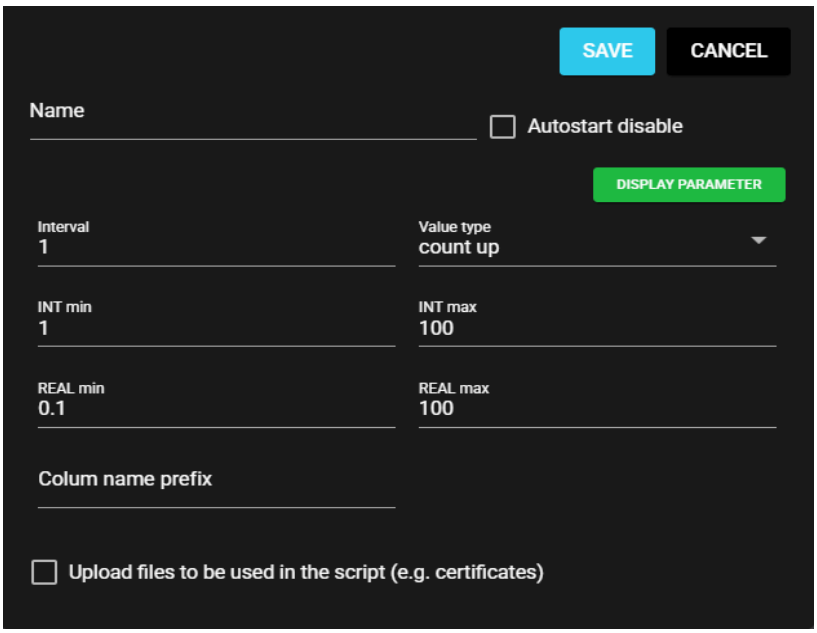


6. 정의 파일을 등록한 커스텀 컴포넌트의 설정 화면에, 파일에서 정의한 항목이 표시됩니다.



파라미터 표시

1. 「 파라미터 표시 」를 누릅니다.



2. 커스텀 UI에 입력한 내용에서 생성되는, 컴포넌트의 실행 시 파라미터를 확인할 수 있습니다.

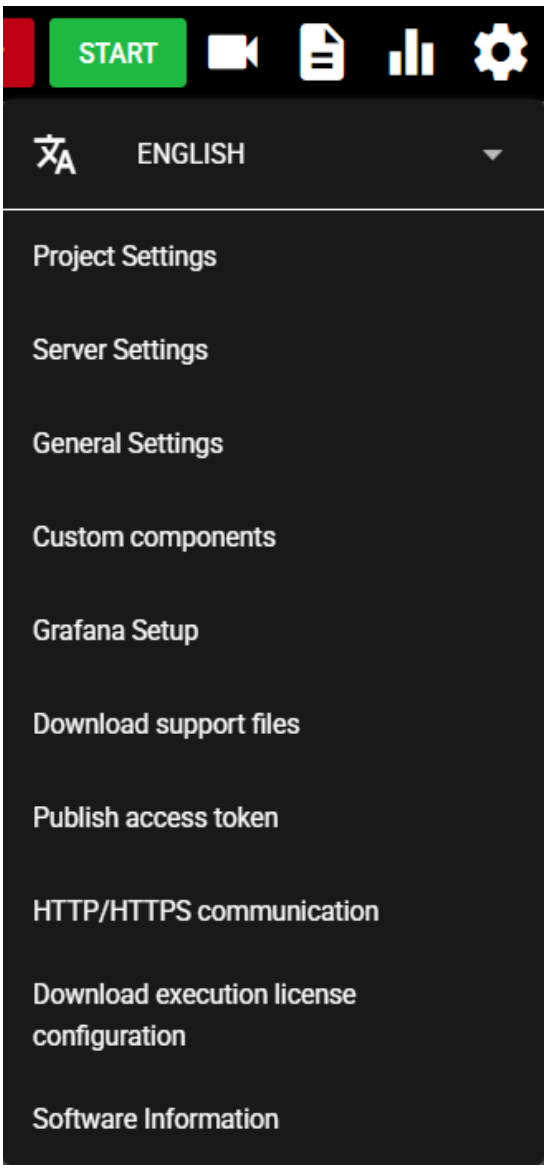


AI 번역본
For Reference Only-

6.2.5.4.2 삭제

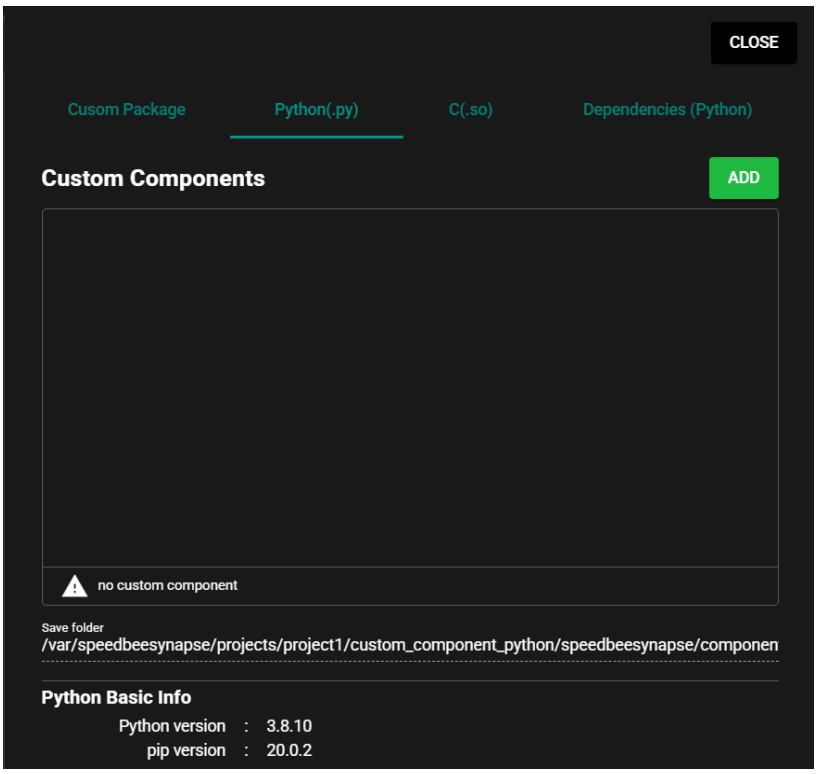
다음 순서로, 등록한 정의 파일을 삭제하고, 화면을 디폴트의 상태로 되돌릴 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

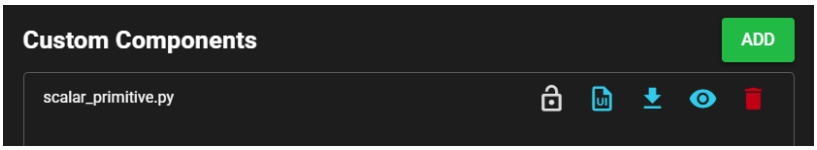


2. 다이얼로그 상단 탭의 「 Python(.py) 」를 선택합니다.

AI 번역본
-For Reference Only-

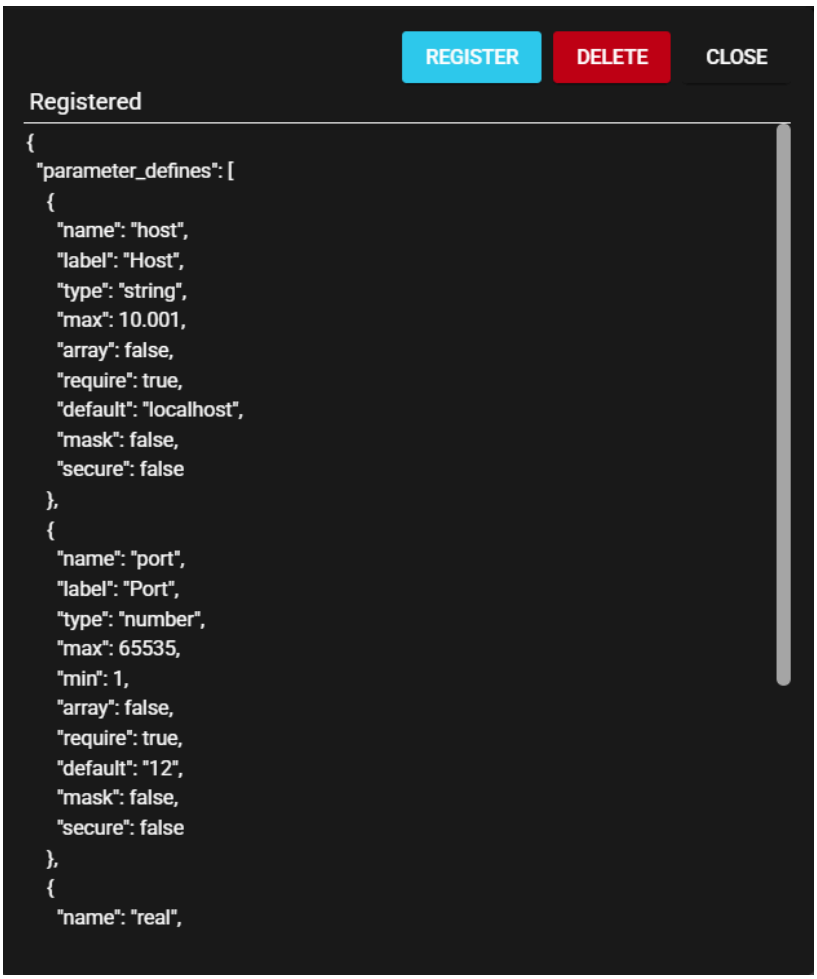


3. 정의 파일을 등록한 커스텀 컴포넌트의「UI 아이콘（UI 설정）」을 누릅니다.

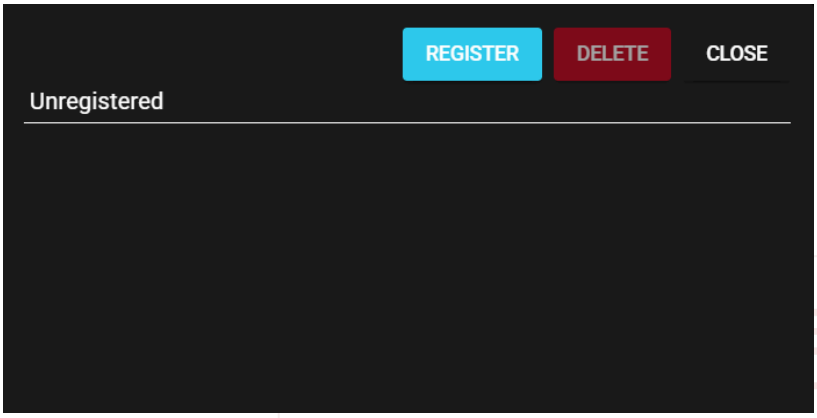


4.「삭제」를 누릅니다.

AI 번역본
-For Reference Only-

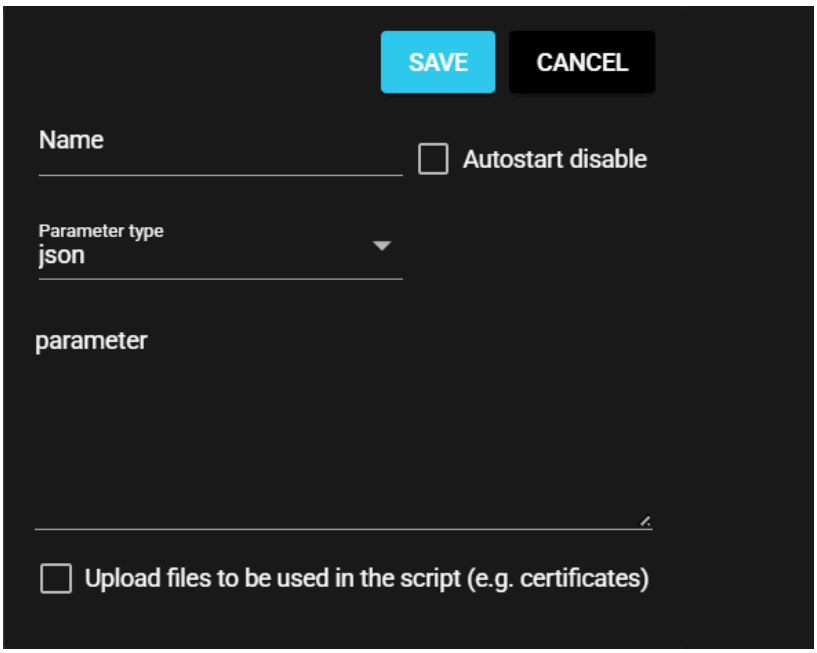


5. 등록한 정의 파일이 삭제되고,「 미등록 」이라고 표시됩니다.



6. 커스텀 컴포넌트의 설정 화면이, 디폴트의 상태로 돌아갑니다.

For Reference Only-



6.2.5.5 C 언어 커스텀 컴포넌트로의 커스텀 UI(JSON)의 적용

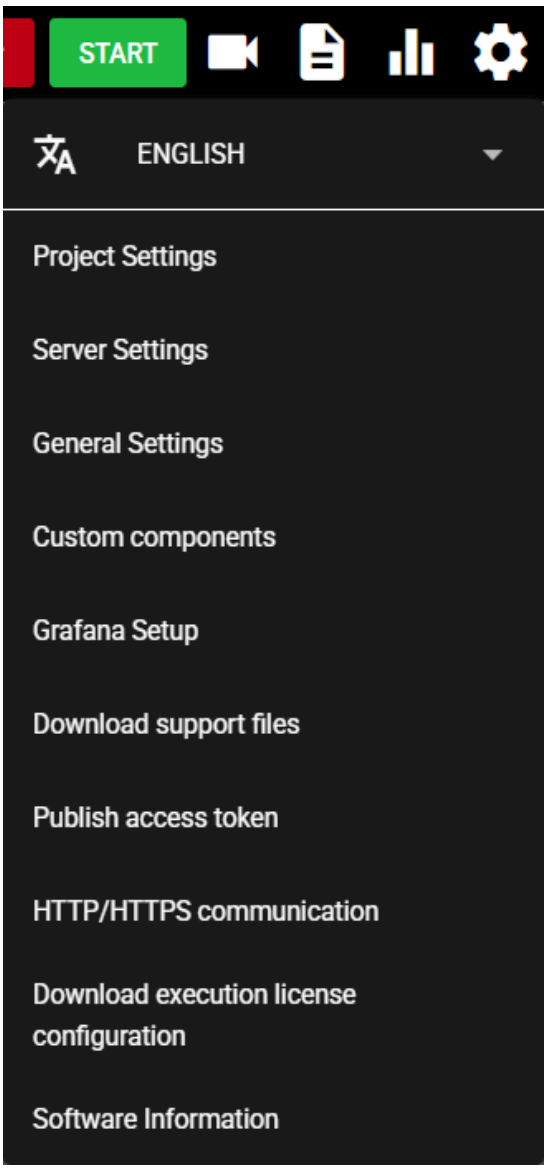
「 Synapse 커스텀 UI용 JSON 파일의 작성 」에서 작성한 정의 파일을 등록함으로써, C 언어 커스텀 컴포넌트에 커스텀 UI를 설정할 수 있습니다.

6.2.5.5.1 등록

다음 순서로, 커스텀 UI를 등록할 수 있습니다.

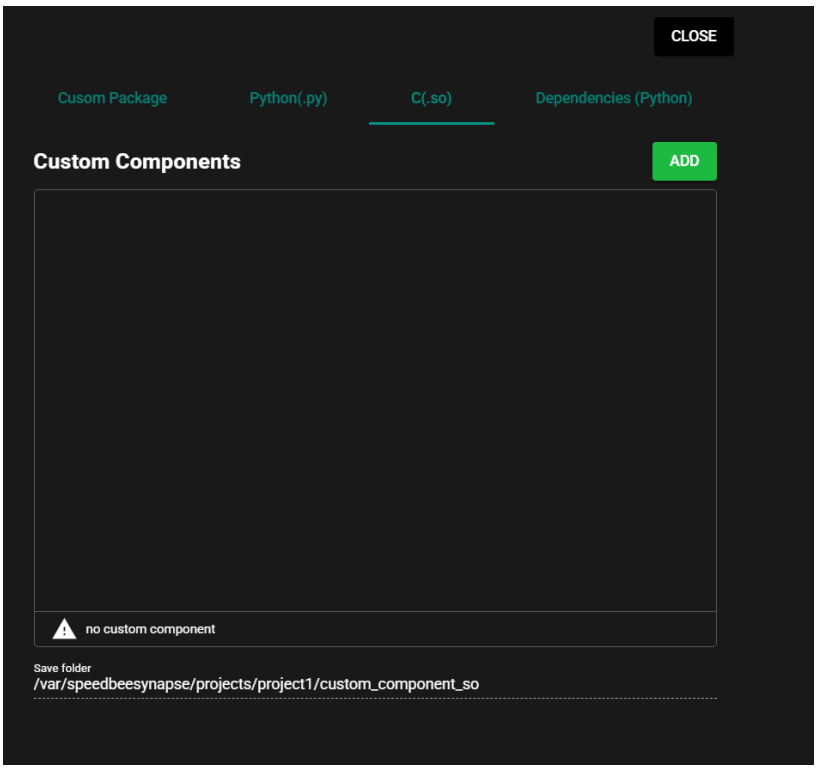
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

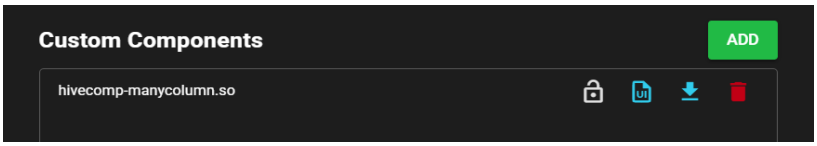


2. 다이얼로그 상단 탭의「C(.so)」를 선택합니다.

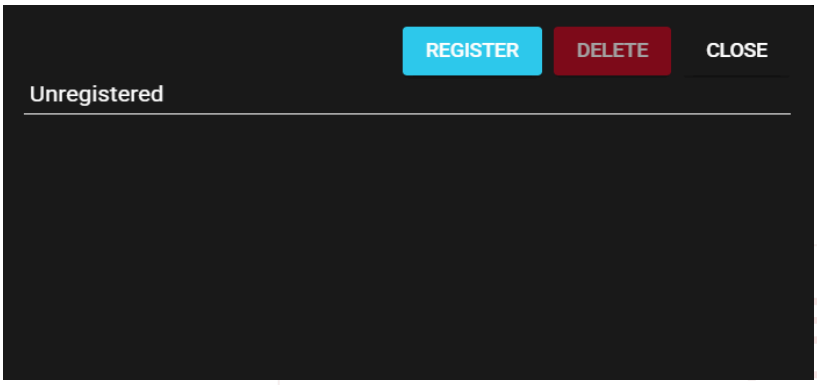
AI 번역본
-For Reference Only-



3. 임의의 커스텀 컴포넌트의「UI 아이콘（UI 설정）」을 누릅니다.

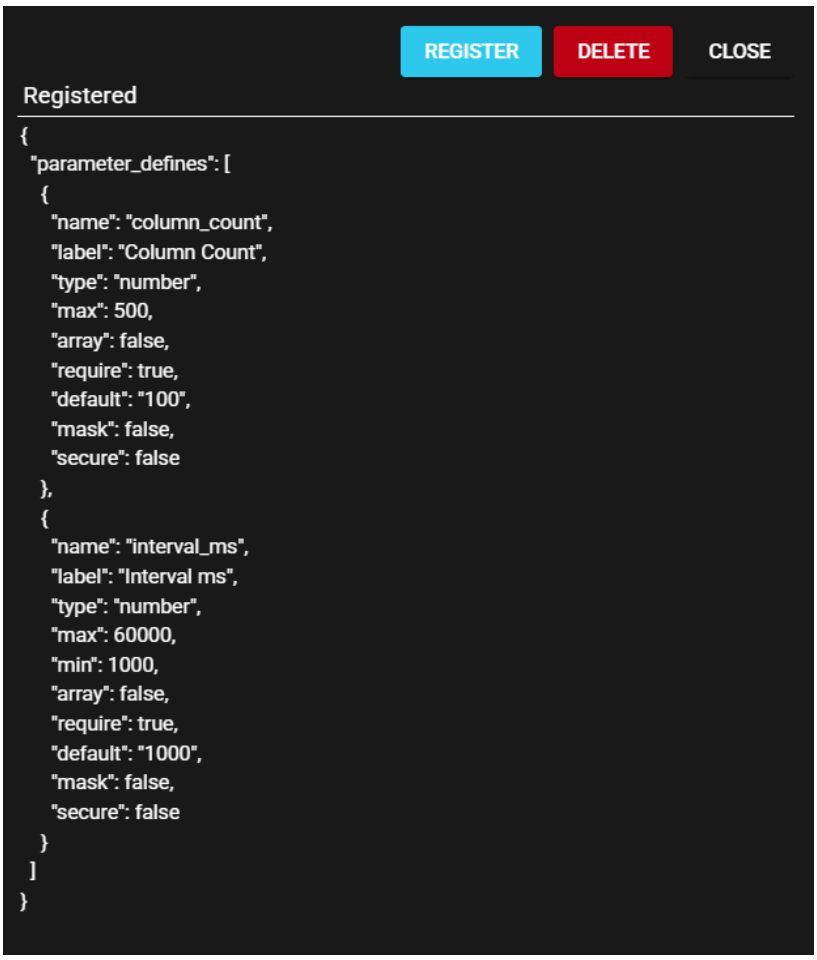


4.「등록」을 누르고, 정의 파일을 등록합니다.

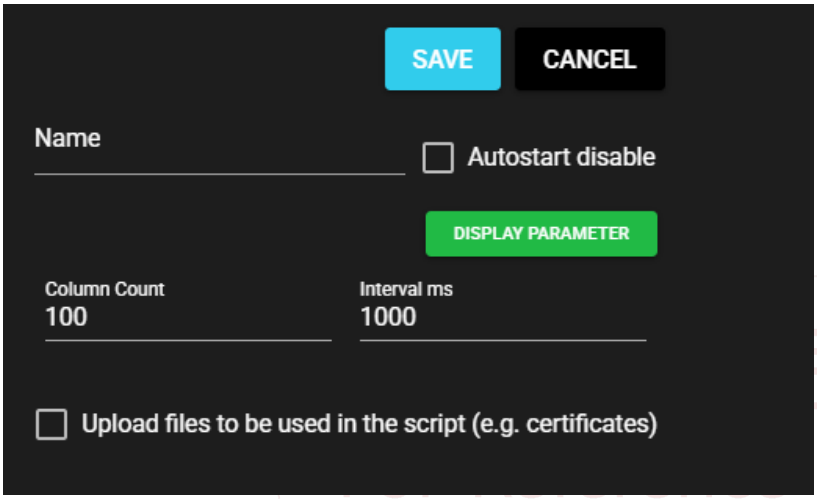


5.「등록 완료」라는 문구와, 정의 파일의 내용이 표시됩니다.

-For Reference Only-

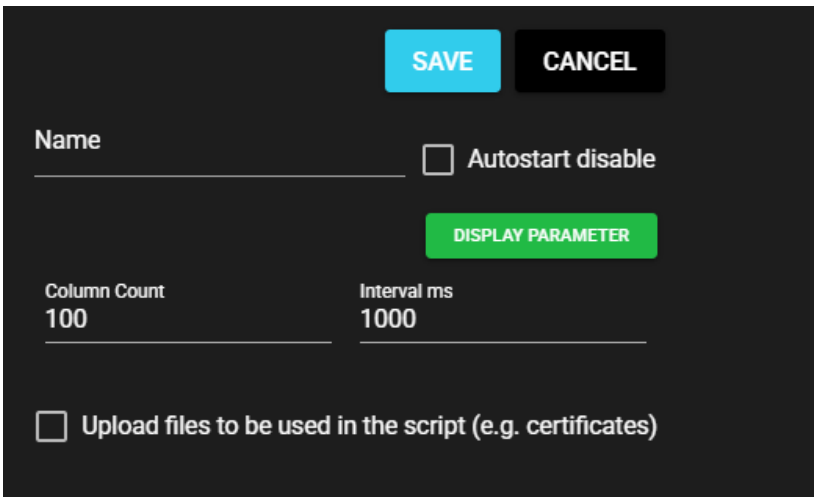


6. 정의 파일을 등록한 커스텀 컴포넌트의 설정 화면에, 파일에서 정의한 항목이 표시됩니다.

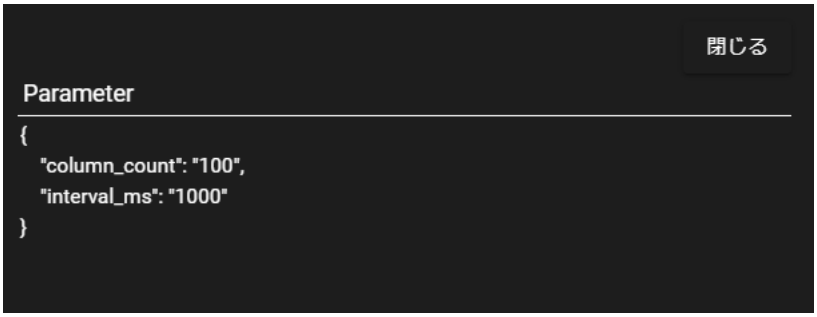


파라미터 표시

1. 「 파라미터 표시 」를 누릅니다.



2. 커스텀 UI에 입력한 내용에서 생성되는, 컴포넌트의 실행 시 파라미터를 확인할 수 있습니다.

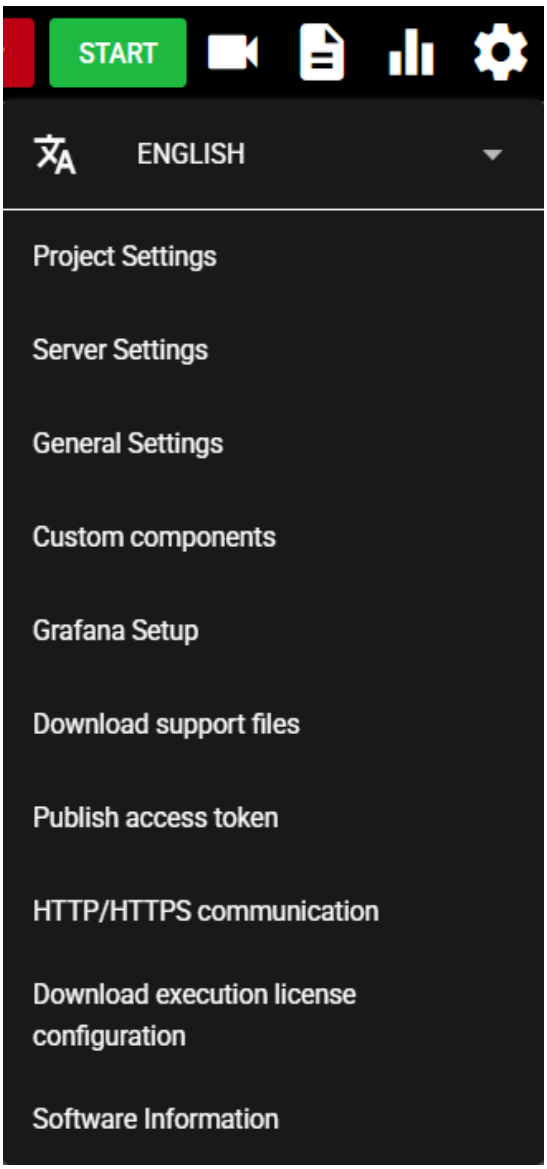


6.2.5.5.2 삭제

다음 순서로, 등록한 정의 파일을 삭제하고, 화면을 디폴트의 상태로 되돌릴 수 있습니다.

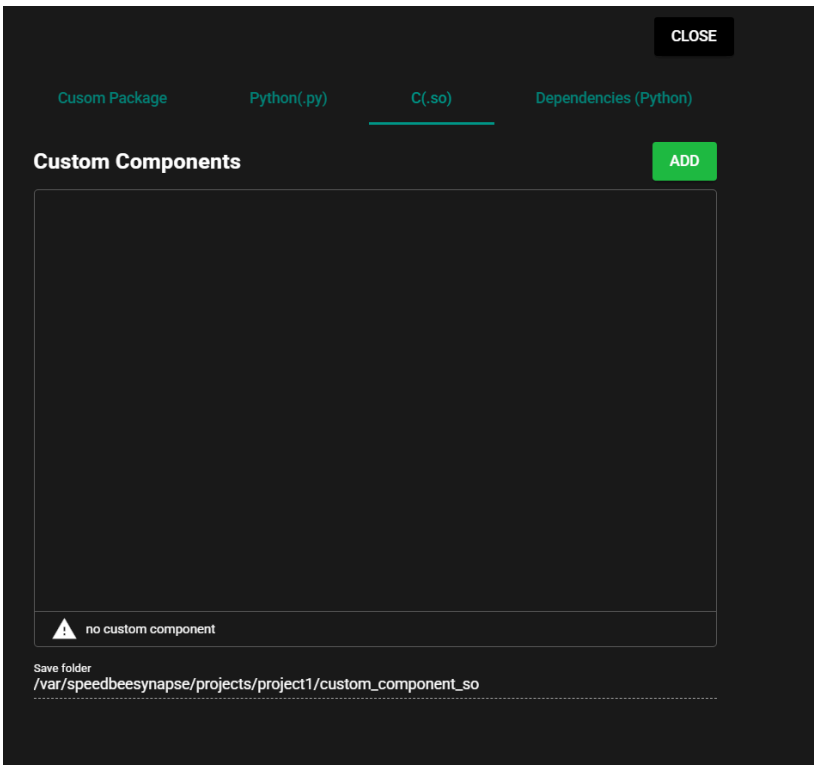
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-

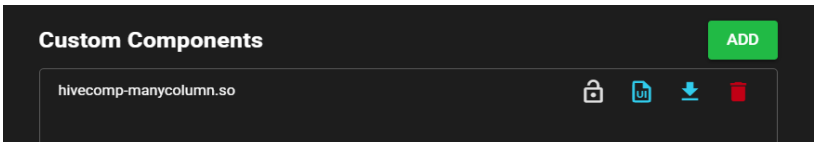


2. 다이얼로그 상단 탭의「C(.so)」를 선택합니다.

AI 번역본
-For Reference Only-

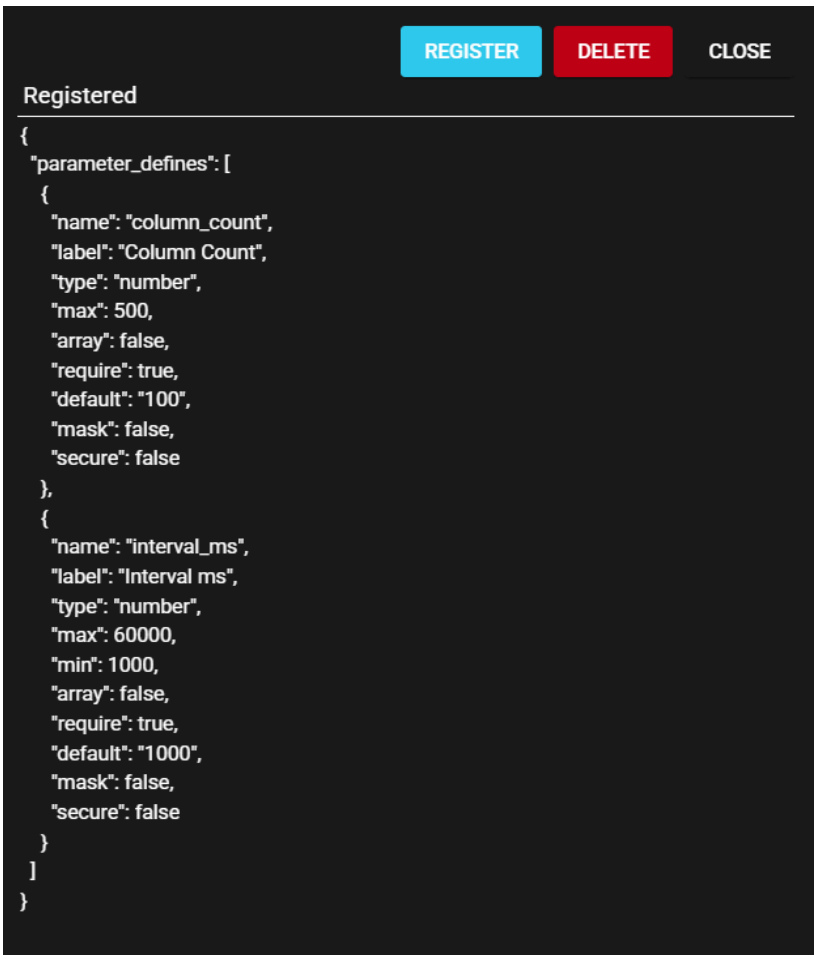


3. 정의 파일을 등록한 커스텀 컴포넌트의「UI 아이콘（UI 설정）」을 누릅니다.

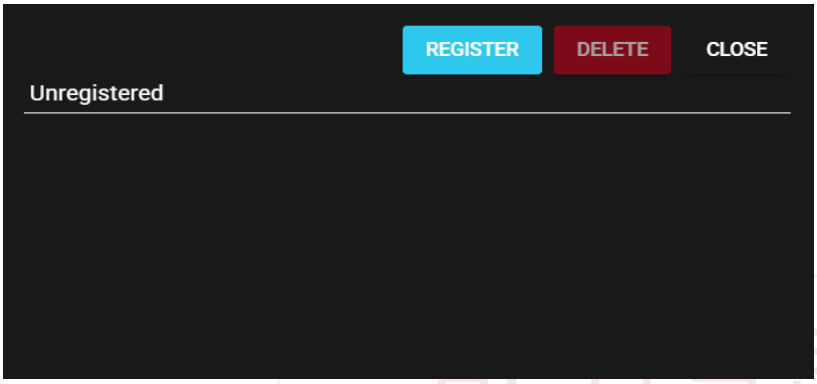


4.「삭제」를 누릅니다.

AI 번역본
-For Reference Only-

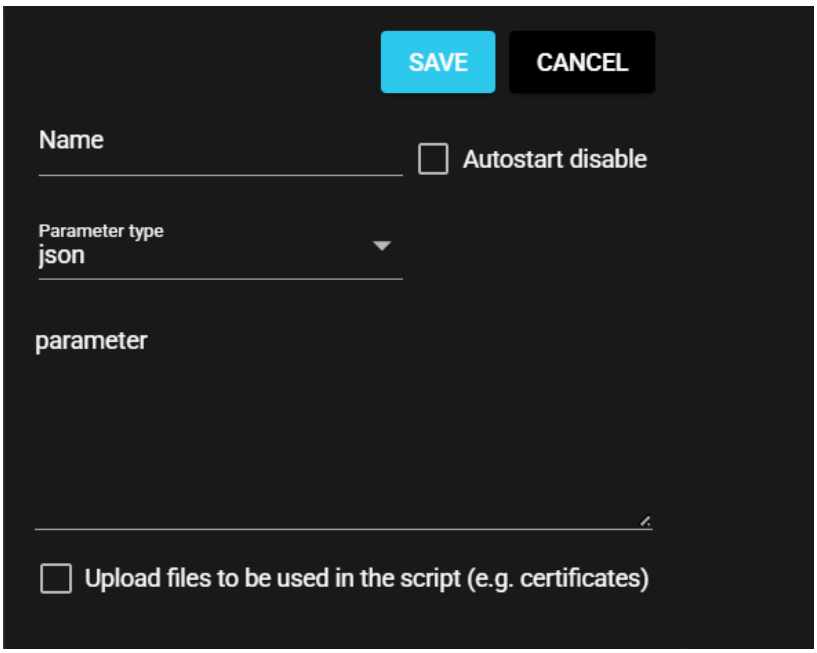


5. 등록한 정의 파일이 삭제되고,「미등록」이라고 표시됩니다.



6. 커스텀 컴포넌트의 설정 화면이, 디폴트의 상태로 돌아갑니다.

-For Reference Only-



AI 번역본
-For Reference Only-

6.2.6 커스텀 컴포넌트 개발 환경(SCCDE)

본 항에서는, Synapse 커스텀 컴포넌트 개발 환경(이하 SCCDE)에 대해 설명합니다.

6.2.6.1 SCCDE의 개요

Synapse 커스텀 컴포넌트 개발 환경(SCCDE)은, Synapse의 커스텀 컴포넌트를 개발하기 위한 다양한
툴・패키지를 집약한 환경입니다. 현재로서는 이하의 기능을 갖고 있습니다.

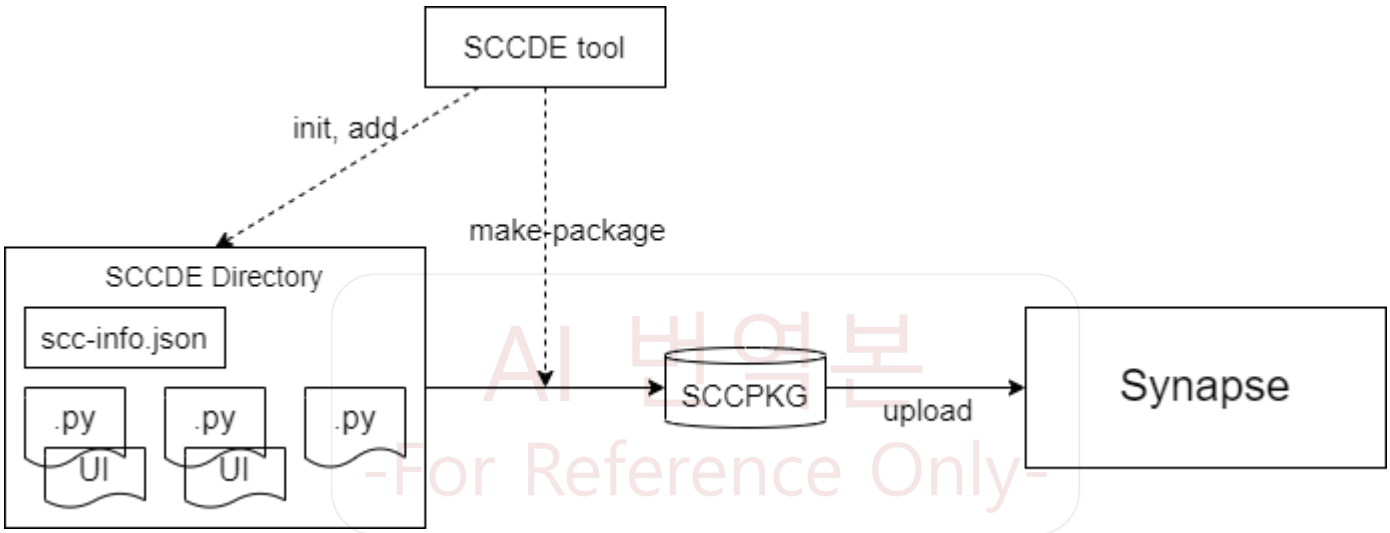
- 커스텀 컴포넌트 관리 디렉터리의 초기화(init)

  관리용 디렉터리는, Git이나 Subversion 등의 버전 관리 시스템에서 이용하기 쉬운 형태로 되어 있습니다. 커스텀 컴포넌트의
- 샘플 프로그램의 추가(add)

  컬렉터, 시리얼라이저, 에미터 등의 간이적인 샘플 프로그램을 추가할 수 있습니다.
- 커스텀 컴포넌트의 패키징(make-package)

  컴포넌트의 패키징을 수행하면, 복수의 커스텀 컴포넌트나 커스텀 UI를 한꺼번에 1 개의 파일(SCCPKG)에 담을 수 있습니다.
  이 SCCPKG 파일을, Synapse의 관리 화면에서 업로드하는 것으로, 복수의 커스텀 컴포넌트를
  한꺼번에 관리할 수 있습니다.
- 커스텀 컴포넌트의 테스트 환경의 실행(serve)

  작성한 커스텀 컴포넌트를 간이적으로 실행하기 위한 테스트 환경을 기동할 수 있습니다.



AI 번역본
-For Reference Only-

현시점에서는 Python으로 구현된 커스텀 컴포넌트에만 대응하고 있습니다. C 언어 실장의 컴포넌트의 대응은 미정입니다.

6.2.6.2 설치 방법

SCCDE는 이하의 방법으로 설치할 수 있습니다.

- Synapse와 함께 배포되는 whl 패키지 파일을 사용하여 설치
- PyPI（pip의 표준 인덱스）에서 설치

설치할 환경은 Windows여도 Linux여도 상관없습니다. 다만 테스트 환경의 실행(serve) 기능은, 동일 디바이스에 Synapse가 설치되어 있을 필요가 있습니다.

Linux

```
$ python3 -m venv ~/sccde
$ . ~/sccde/bin/activate
(sccde) $ pip3 install speedbeesynapse-sccde
```

참고：venv

Windows(pwsh)

```
PS> python -m venv "$HOME\sccde"
PS> . "$HOME\sccde\Scripts\Activate.ps1"
(sccde) PS> pip3 install speedbeesynapse-sccde
```

참고：venv

본 패키지는 표준적인 Python 패키지이므로, 상기의 pip, venv를 사용한 설치 방법 이외에도 다양한 패키지 관리 툴（Poetry, uv 등）을 사용 가능합니다. 선호하시는 툴을 이용하십시오.

※이후에 설명하는 순서는, 상기의 activate를 수행한 (sccde) 환경 내에서 실시하는 것을 상정하고 있습니다.

6.2.6.3 SCCDE의 사용법과 패키징

6.2.6.3.1 커스텀 컴포넌트 관리 디렉터리의 초기화

커스텀 컴포넌트를 관리하기 위해, 빈 디렉터리를 준비하여 초기화할 필요가 있습니다. 이하의 순서로 초기화하십시오.

AI 번역본
-For Reference Only-

Linux

```
(sccde) $ mkdir sccpack-test
(sccde) $ cd sccpack-test
(sccde) $ sccde init
(sccde) $ ls
parameter_ui  source  scc-info.json
```

Windows(pwsh)

```
(sccde) PS> mkdir sccpack-test
(sccde) PS> cd sccpack-test
(sccde) PS> sccde init
(sccde) PS> ls
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          2025/xx/yy    hh:mm                parameter_ui
d----          2025/xx/yy    hh:mm                source
-a---          2025/xx/yy    hh:mm            644 scc-info.json
```

초기화를 수행하면 현재의 디렉터리에 대해 이하의 처리가 수행됩니다.

- scc-info.json 의 생성
- 커스텀 컴포넌트(collector)의 샘플 프로그램의 추가

scc-info.json 은, 이 디렉터리 내에서 관리하는 커스텀 컴포넌트에 관한 정보를 기재하기 위한 파일입니다. JSON 형식으로 되어 있으므로, 필요에 따라 텍스트 에디터 등으로 편집하십시오.

주요한 프로퍼티를 아래에 기재하겠습니다.

프로퍼티	설명
package-name	패키지의 명칭입니다. 임의의 이름을 기재할 수 있습니다.
package-version	패키지의 버전을 나타내는 문자열입니다. 기본적으로는 "." 구분의 3 개의 수치를 기재하십시오. 이 버전 문자열은 Semantic Versioning 2.0.0에 따른 형식이 바람직합니다.
package-uuid	패키지를 식별하기 위한 ID입니다. 초기 상태에서는 랜덤한 UUID가 설정되어 있습니다. 기본적으로는 변경하지 않고 사용하십시오
package-description	패키지의 설명문입니다. 임의의 문자열을 기재해 주십시오.
author	패키지 작성자의 이름을 기재할 수 있습니다.

AI 번역본
-For Reference Only-

프로퍼티	설명
license	이 패키지를 이용할 때의 라이선스 종별을 기입할 수 있습니다.（BSD, MIT, UNLICENSED 등）. 불필요한 경우는 빈 문자열로 하십시오.
license-file	이 패키지의 라이선스 상세를 기술한 파일의 상대 경로를 지정할 수 있습니다. 불필요한 경우는 빈 문자열로 하십시오.
python-components-source	이 디렉터리 내에서 관리하는 Python의 커스텀 컴포넌트의 배치 경로입니다. 기본적으로 변경할 필요는 없습니다.
components	복수의 커스텀 컴포넌트의 정의를 기재하기 위한 오브젝트입니다. key 에는 커스텀 컴포넌트의 UUID를 지정할 필요가 있습니다.
components.*.name	커스텀 컴포넌트의 명칭입니다. 임의의 이름을 기재할 수 있습니다.
components.*.description	커스텀 컴포넌트의 설명문입니다.
components.*.component-type	커스텀 컴포넌트의 실장 종별입니다. 'python' or 'c'를 지정할 필요가 있습니다.
components.*.moduelname	Python 커스텀 컴포넌트의 모듈명입니다. `python-components-source` 에서 지정한 경로를 기준으로 한 경우의 커스텀 컴포넌트를 import하기 위한 모듈명을 기재하십시오.
components.*.parameter-ui-type	커스텀 UI의 정의 파일로 종별을 지정합니다. 'json' or 'html'을 지정 가능합니다.
components.*.parameter-ui	커스텀 UI로의 상대 경로를 지정하십시오. 'json' 형식의 경우는 그 파일로의 경로를, 'html' 형식의 경우는 html 파일이 배치된 디렉터리로의 경로를 지정하십시오.

For Reference Only-

6.2.6.3.2 커스텀 컴포넌트의 추가

관리 디렉터리를 초기화한 후에는, add 서브커맨드에 의해 커스텀 컴포넌트의 샘플 프로그램을 추가할 수 있습니다.

Linux

```
(sccde) $ sccde add -t collector
```

Windows(pwsh)

```
(sccde) PS> sccde add -t collector
```

-t 의 뒤의 문자열은, collector 이외에 serializer , emitter 등으로 치환 가능합니다.

추가한 샘플 프로그램은 초기 상태에서는 source/python 아래에 배치됩니다. 이 샘플은 그대로도 실행 가능하도록 되어 있습니다만, 이것을 베이스로 자유롭게 커스텀 컴포넌트를 개변하여 독자적인 컴포넌트를 만들 수 있습니다.

커스텀 컴포넌트의 실장 방법에 대해서는 아래를 참조하십시오.

- Python 커스텀 컴포넌트 작성
- 커스텀 컴포넌트 Python-API 레퍼런스

또한, 추가된 샘플 컴포넌트에는 커스텀 UI의 샘플도 부속되어 있습니다. 이쪽도 필요에 따라 개변하여 이용하십시오.

- 커스텀 UI 작성

디폴트에서는 json 형식의 커스텀 UI가 부여됩니다. 추가 시의 옵션으로 -u html 을 지정함으로써, HTML 형식의 커스텀 UI도 작성 가능합니다.

6.2.6.3.3 커스텀 컴포넌트의 테스트 실행

SCCDE 커맨드를 설치한 환경에, Synapse도 설치되어 있는 경우는 serve 커맨드로 커스텀 컴포넌트를 테스트 실행할 수 있습니다.

Linux

```
(sccde) $ sccde serve --port 8123
Preparing Synapse data directories..
Synapse server ready: http://127.0.0.1:8123
:
```

Windows(pwsh)

```
(sccde) PS> sccde serve --port 8123
Preparing Synapse data directories..
Synapse server ready: http://127.0.0.1:8123
:
```

최초 실행에서는 데이터 디렉터리의 초기화에 잠시 시간이 걸립니다. ready: http://.. 의 메시지가 표시되면, 브라우저에서 그 URL을 여십시오.

URL을 열면, Synapse의 플로우 에디터의 Edit Mode와 동일한 화면이 표시됩니다. 좌측의 컴포넌트 베이스 목록에서, 작성한 커스텀 컴포넌트를 선택 가능하게 되어 있습니다.

테스트용 Synapse는, Ctrl-C 키로 정지할 수 있습니다.

옵션

옵션	설명
`--host`	테스트용 Synapse 서버가 이용할 IP 주소를 지정할 수 있습니다. 가동하고 있는 디바이스 외부에서도 액세스하고 싶은 경우는 0.0.0.0 을 지정하십시오. 디폴트는 127.0.0.1 로 되어 있습니다.
`--port`	테스트용 Synapse 서버가 이용할 TCP 포트 번호 지정할 수 있습니다. 다른 서비스와 포트 번호가 겹치지 않도록 임의의 번호를 지정하십시오. 디폴트는 8000 으로 되어 있습니다.
`--verbose`	테스트용 Synapse의 실행 로그를 표시합니다.

데이터 디렉터리와 로그

테스트용 Synapse의 실행 중의 데이터는, sccde의 작업 디렉터리 내에 있는 .synapse/var-speedbeesynapse 디렉터리에 배치됩니다.

Linux

```
(sccde) $ ls .synapse/var-speedbeesynapse/
certs  config  dblog  hivecore_exec_status.json  logs  projects  sdts.db  sdts.file  webui_tmp
```

Windows(pwsh)

```
(sccde) PS> ls .\.synapse\var-speedbeesynapse\
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----          2025/xx/yy    hh:mm                certs
d----          2025/xx/yy    hh:mm                config
d----          2025/xx/yy    hh:mm                dblog
d----          2025/xx/yy    hh:mm                logs
d----          2025/xx/yy    hh:mm                projects
d----          2025/xx/yy    hh:mm                sdts.db
d----          2025/xx/yy    hh:mm                sdts.file
d----          2025/xx/yy    hh:mm                webui_tmp
-a---          2025/xx/yy    hh:mm            105 hivecore_exec_status.json
```

AI 번역본
-For Reference Only-

데이터를 클리어하여 초기 상태로 되돌리고 싶은 경우는, .synapse 폴더째 삭제하고 나서 테스트용 Synapse를 기동해 주십시오.

또한, 커스텀 컴포넌트의 디버그 등으로 로그를 참조하고 싶은 경우는, .synapse/var-speedbeesynapse/logs 에 있는 speedbeesynase.log 를 참조할 수 있습니다.

개발용 컴포넌트

테스트용 Synapse 실행 중은, 전용의 개발용 컴포넌트를 이용할 수 있습니다. 개발용 컴포넌트의 상세에 대해서는,
「 개발용 컴포넌트 」를 참조하십시오.

6.2.6.3.4 패키징

커스텀 컴포넌트를 추가한 후에는, make-package 서브커맨드에 의해, 패키지를 작성할 수 있습니다.

Linux

```
(sccde) $ sccde make-package
(sccde) $ ls
custom_component_package_example.sccpkg  parameter_ui  scc-info.json  source
```

Windows(pwsh)

```
(sccde) PS> sccde make-package
(sccde) PS> ls
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----           2025/xx/xx    hh:mm                parameter_ui
d----           2025/xx/xx    hh:mm                source
-a---           2025/xx/xx    hh:mm           2481 custom_component_package_example.sccpkg
-a---           2025/xx/xx    hh:mm            644 scc-info.json
```

.sccpkg 라는 확장자가 붙은 파일이 SCCPKG입니다. 이 파일을 Synapse에 등록할 수 있습니다.

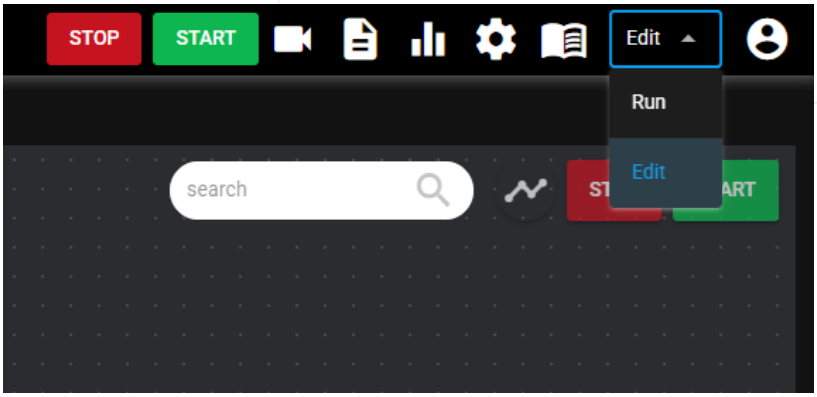
6.2.6.4 Synapse로의 SCCPKG 파일의 등록, 및, 삭제

전항에서 설명한 패키징에 의해 작성한 패키지 파일(SCCPKG)을, Synapse의 관리 화면에서 등록하는 것으로, 패키지에 포함되는 커스텀 컴포넌트를 이용할 수 있게 됩니다.

6.2.6.4.1 등록/갱신

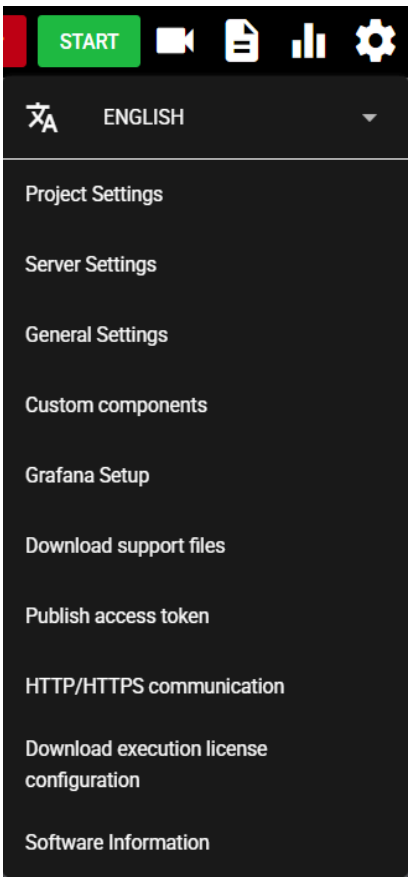
다음 순서로, 작성한 SCCPKG 파일을 SpeeDBee Synapse에 등록할 수 있습니다.

1. 헤더의 모드 선택에서 "Edit Mode"를 선택하고, Edit 모드의 화면을 엽니다.

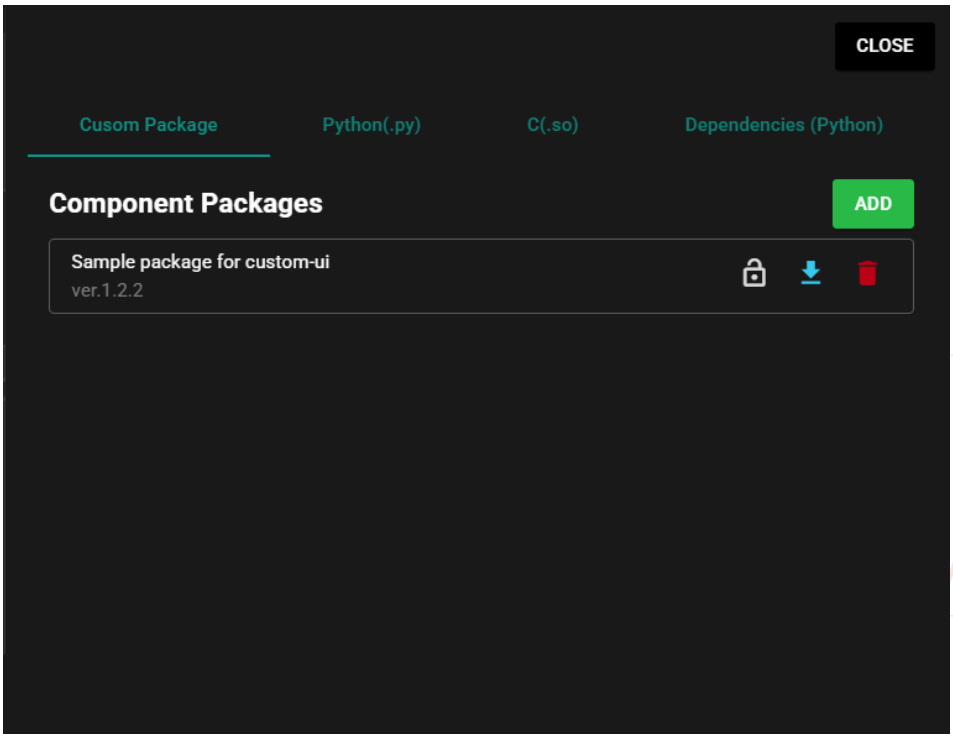


2. 설정 메뉴 아이콘을 누르고, 「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
Only-

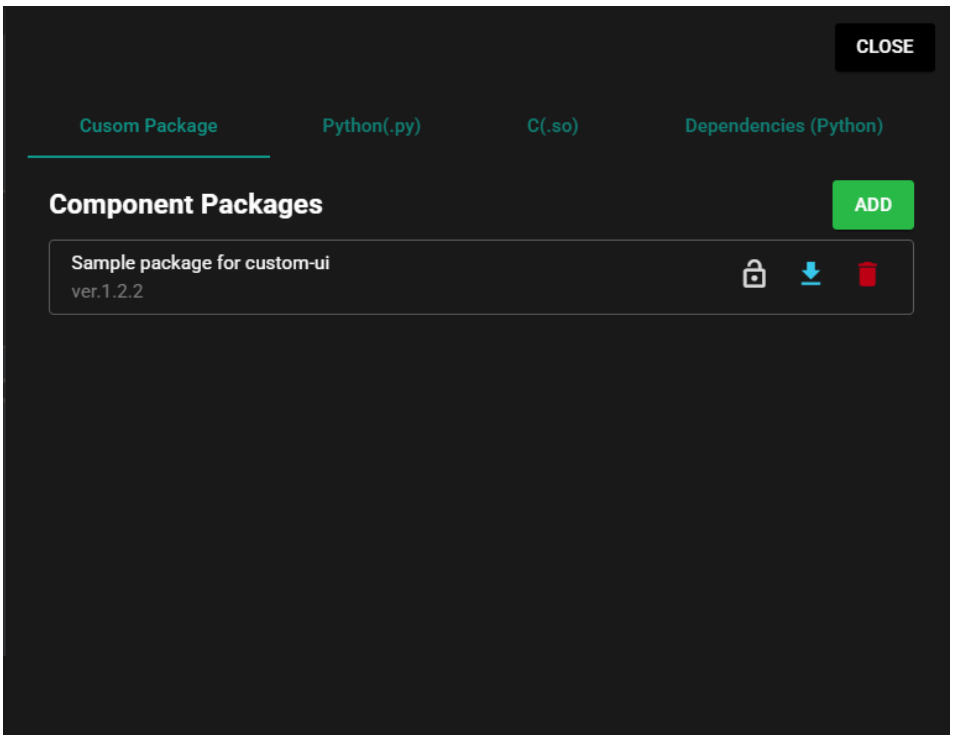


3. 컴포넌트 패키지의「 추가 」를 누르고, 커스텀 컴포넌트의 파일을 등록합니다.

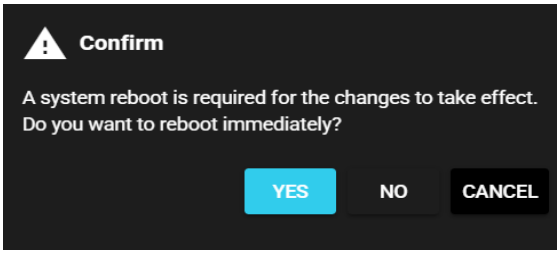


이미 동일한 패키지가 등록 완료된 경우, 여기에서 덮어쓰기를 확인하는 다이얼로그가 표시됩니다. 덮어쓰는 경우는「 예 」를 선택하십시오.

4.「 닫기 」를 누릅니다.

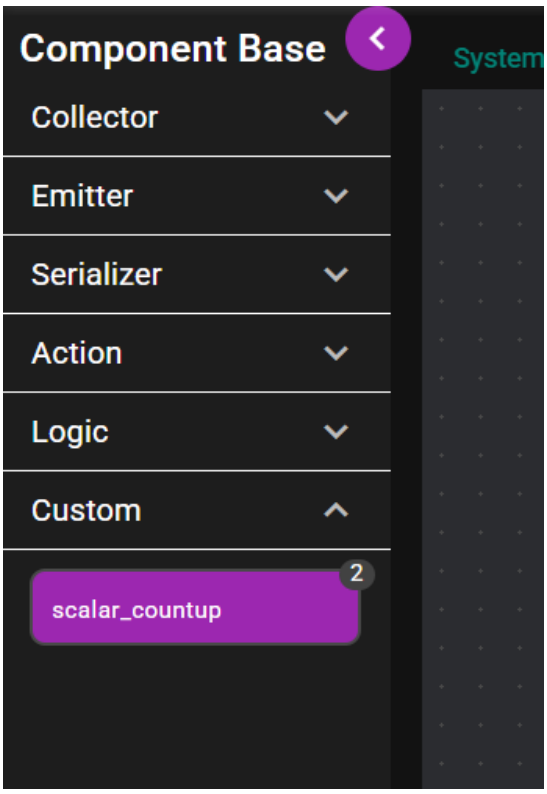


5. 확인 다이얼로그에서 「 예 」를 선택하고, 변경을 반영하기 위해, 시스템을 재기동시킵니다.



6. 좌측 메뉴에, 등록한 커스텀 컴포넌트가 반영됩니다.

AI 번역본
-For Reference Only-

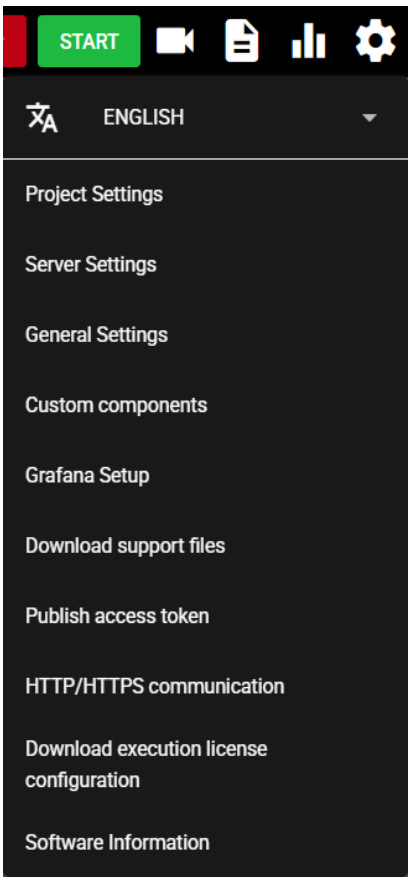


6.2.6.4.2 다운로드

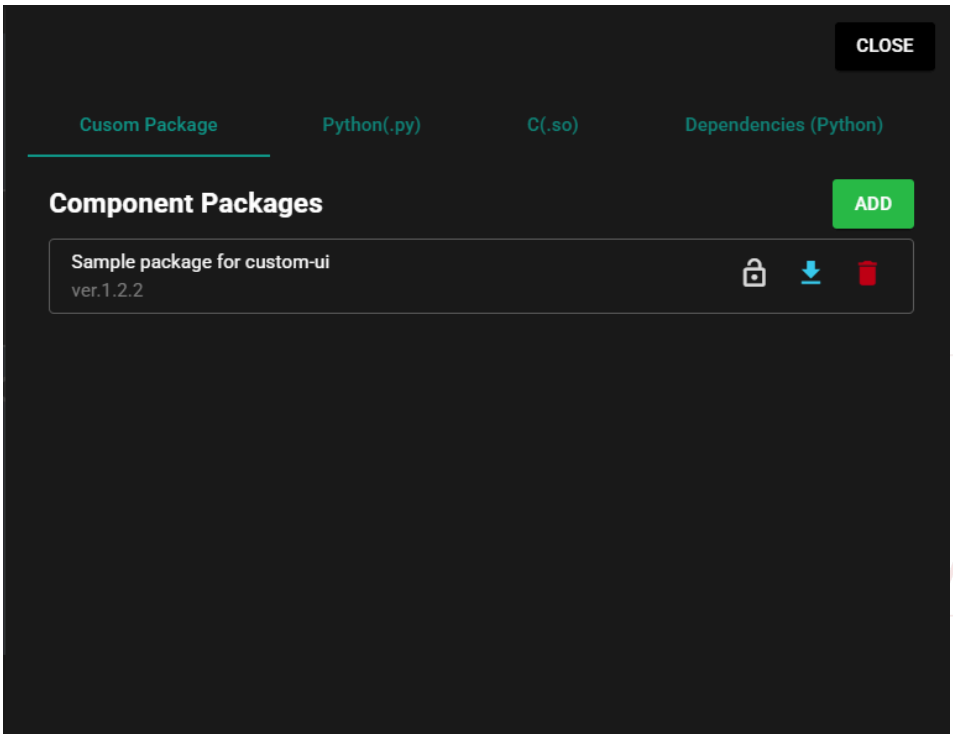
다음 순서로, 등록한 컴포넌트 패키지를 다운로드할 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

AI 번역본
-For Reference Only-



2. 컴포넌트 패키지의 「 다운로드 아이콘 (취득) 」을 누릅니다.

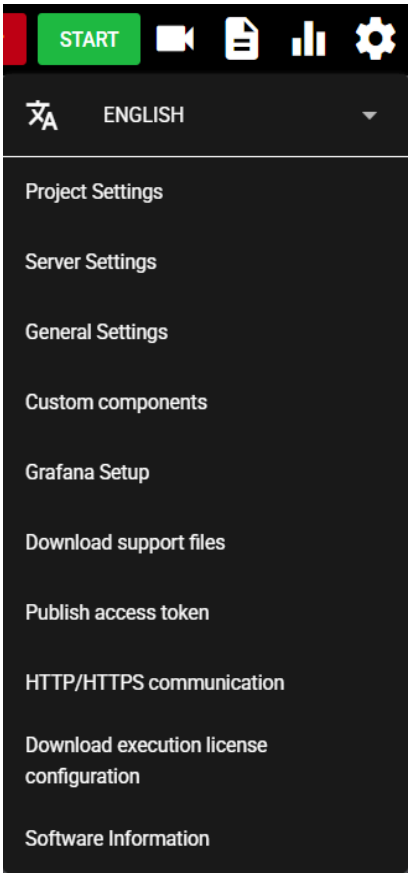


3. 컴포넌트 패키지의 파일이 다운로드됩니다.

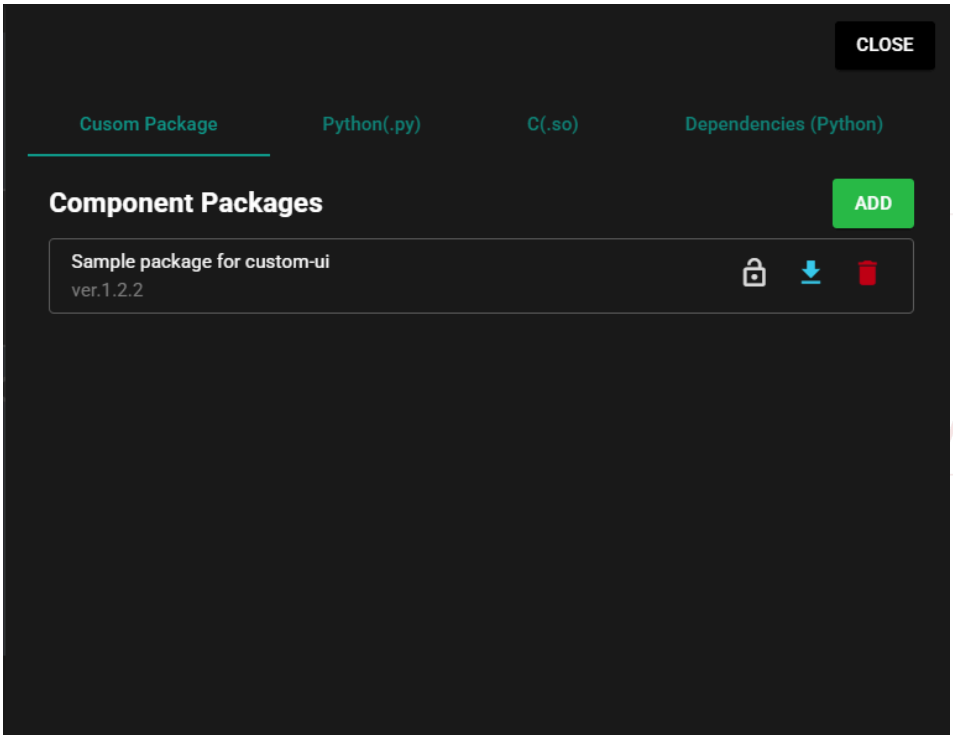
6.2.6.4.3 삭제

다음 순서로, 등록한 SCCPKG 파일을 SpeeDBee Synapse에서 삭제할 수 있습니다.

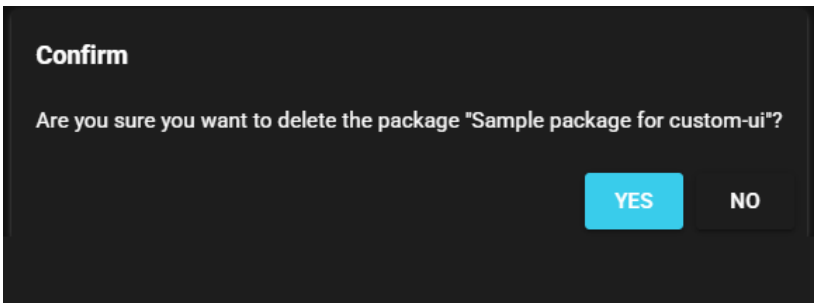
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.



2. 대상 패키지의 오른쪽에 있는「 휴지통 아이콘 (삭제) 」을 누릅니다.



3. 확인 다이얼로그에서「 예 」를 선택합니다.



4. SCCPKG 파일이 삭제됩니다.

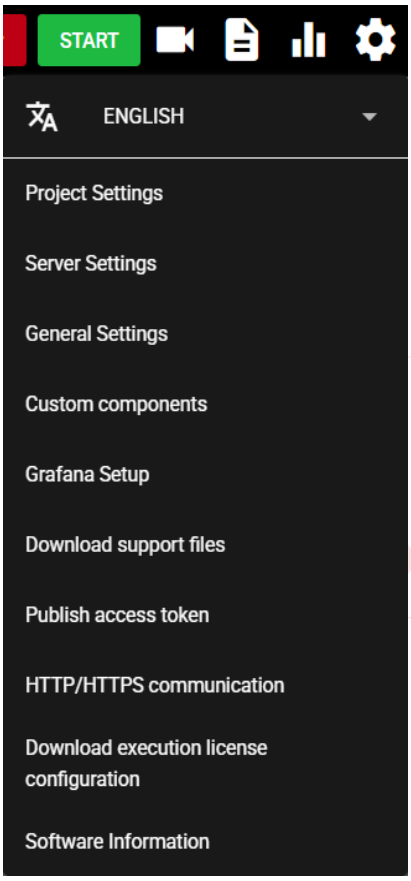
6.2.6.4.4 보호의 부여

등록한 패키지에는 보호를 걸 수 있습니다.
보호된 패키지는 다음 조작을 수행할 수 없게 되기 때문에, 의도하지 않은 갱신을 방지할 수 있습니다.

- 등록/갱신
- 삭제
- 다운로드

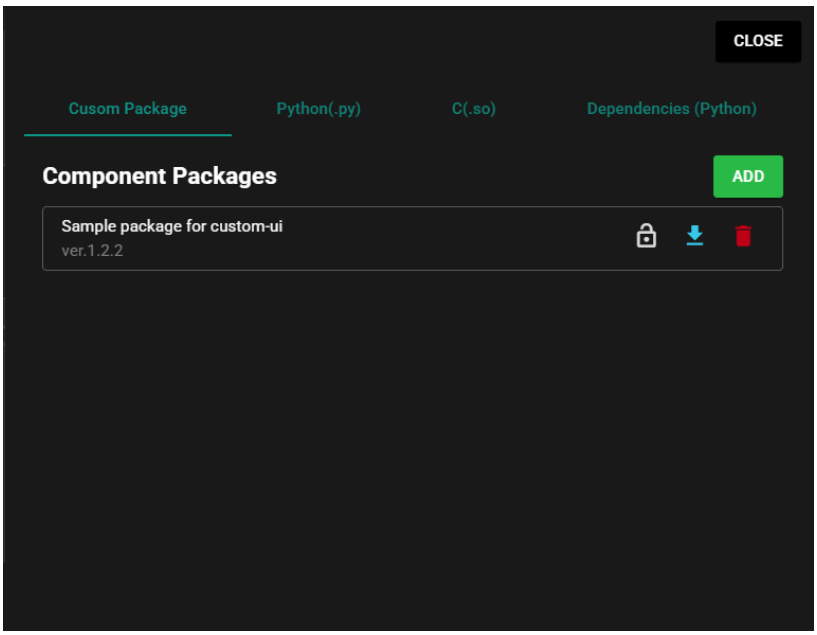
다음 순서로, 패키지에 보호를 걸 수 있습니다.

1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.

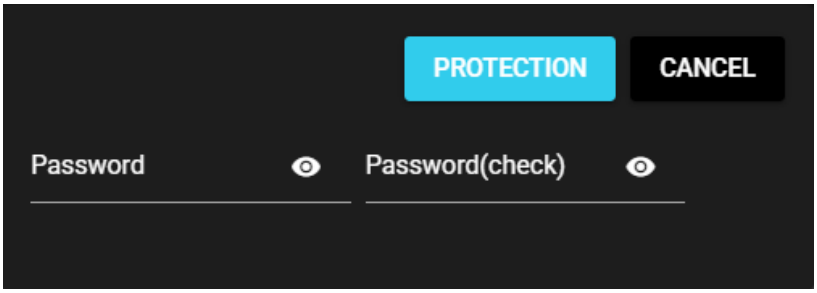


AI 번역본
r Reference Only-

2. 대상 패키지의 오른쪽에 있는「 열쇠 아이콘（ 보호 ）」을 누릅니다.



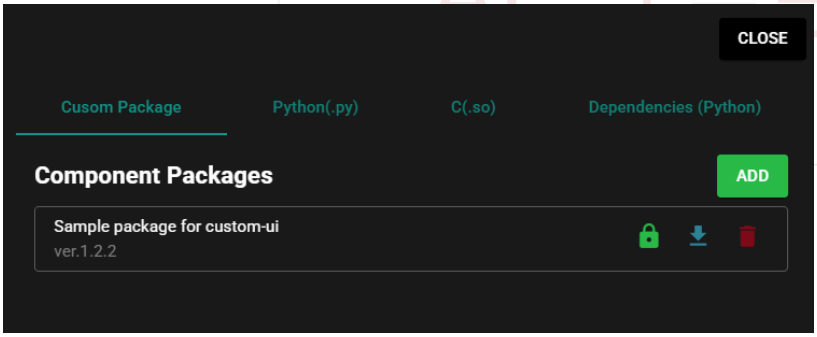
3. 보호 해제할 때 필요하게 되는 패스워드를 입력하고,「 보호 」를 누릅니다.



> 경고
>
> - 패스워드는 복원할 수 없는 형식으로 저장되며, 한 번 설정하면 보호를 해제할 때까지 재설정할 수 없습니다. 보호
> - 설정한 패스워드를 잊은 경우, 해당 패키지는 조작 불능이 됩니다.
>
> 패스워드의 취급에는 충분히 주의하십시오.

AI 버전본
-Only-

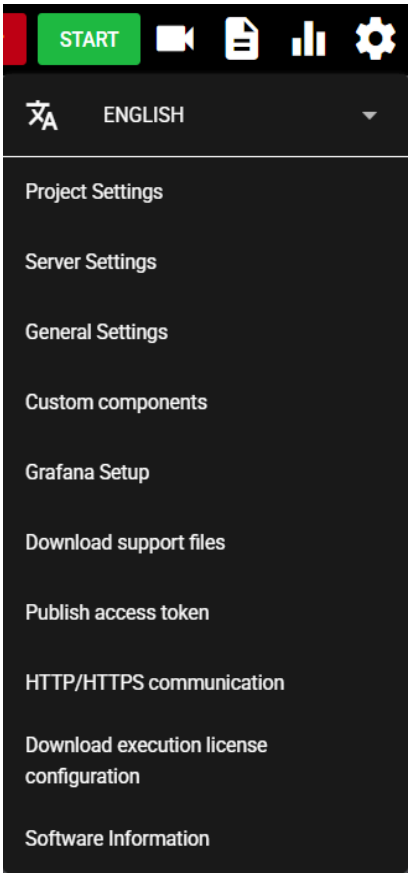
4. 컴포넌트 패키지가 보호 상태가 됩니다.



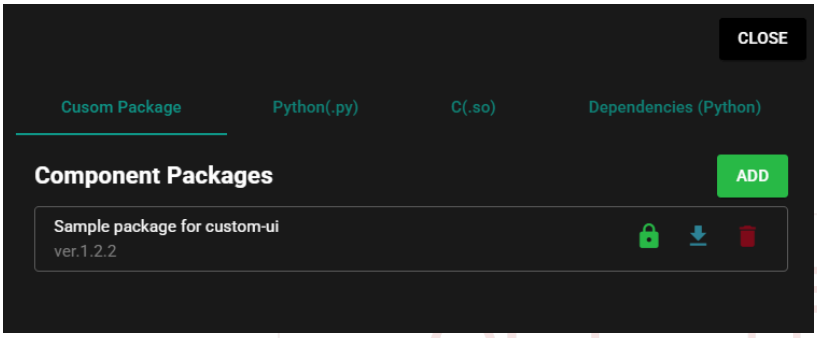
6.2.6.4.5 보호의 해제

다음 순서로, 패키지의 보호를 해제할 수 있습니다.

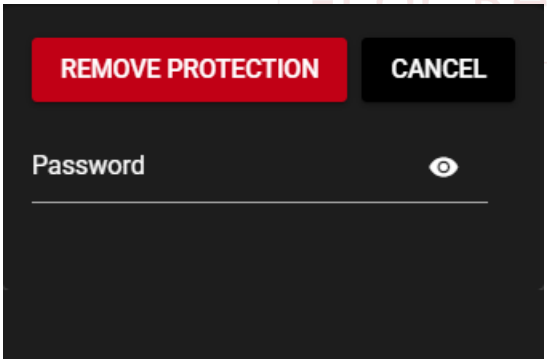
1. 설정 메뉴 아이콘을 누르고,「 커스텀 컴포넌트 」를 선택합니다.



2. 대상 패키지의 오른쪽에 있는「 열쇠 아이콘（ 보호 해제 ）」를 누릅니다.

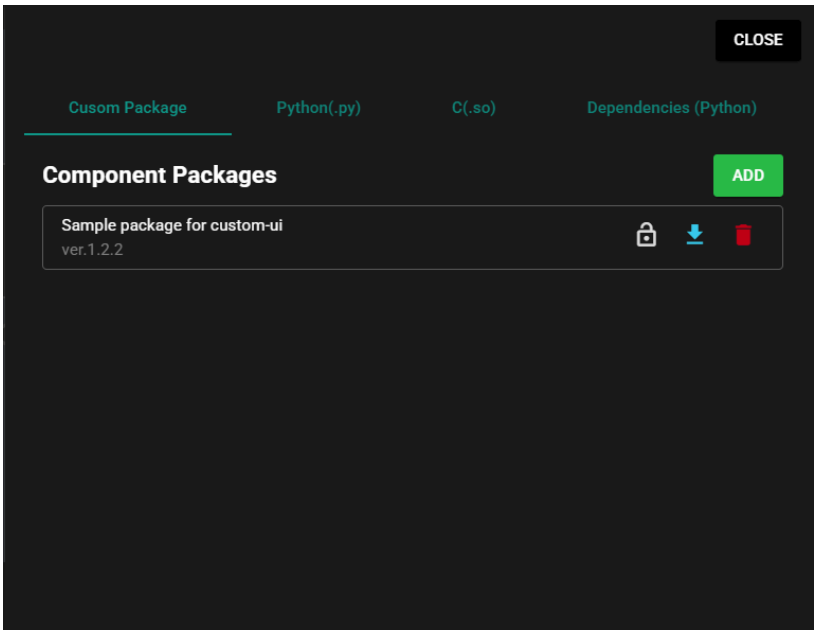


3. 보호를 걸었을 때 설정한 패스워드를 입력하고,「 보호 해제 」를 누릅니다.



-For Reference Only-

4. 컴포넌트 패키지의 보호가 해제됩니다.



AI 번역본
-For Reference Only-

6.3 제어 프로그램과의 연계（WebAPI）

6.3.1 WebAPI의 개요

SpeeDBee Synapse의 일부 조작은, API에 의해 외부에서 수행할 수 있습니다.

6.3.2 WebAPI의 실행 방법

여기에서는 예로서, curl 커맨드에 의해 API를 실행하는 방법을 기재합니다.
curl이 설치되어 있지 않은 경우, curl의 공식 사이트에서 curl을 다운로드하여, 사용 중인 머신에 curl을 설치하십시오.

1. curl 커맨드를 실행하는 환경을 엽니다. Linux의 경우는 터미널을, Windows의 경우는 커맨드 프롬프트를 엽니다.
2. 「API 사양」의 각 API의 「curl 커맨드에 의한 요청」을 복사하고, curl 커맨드를 실행하는 환경에 페이스트합니다.
   ※「curl 커맨드에 의한 요청」의 일부는, 치환할 필요가 있습니다. 상세는, 「공통 사항」을 참조하십시오.
3. Enter 키를 누르면 API가 실행되고, 응답이 표시됩니다. 응답의 상세는, 「API 사양」의 각 API의 「응답」을 참조하십시오.

「API 사양」의 각 API의 「curl 커맨드에 의한 요청」에서는, 다음 항목이 지정되어 있습니다. curl 커맨드 이외의 방법으로 API를 실행하는 경우는, 같은 항목을 지정할 필요가 있습니다.

항목	설명
HTTP 메서드	-X "~~~"라는 형태로 지정되어 있습니다.
요청 URL	"http://"로 시작하는 URL입니다.
HTTP 헤더	-H "~~~"라는 형태로 지정되어 있습니다.
POST 메서드로 송신하는 데이터	-d "~~~"라는 형태로 지정되어 있습니다.

AI 번역본
-For Reference Only-

6.3.3 WebAPI 사양

API에 의해 수행할 수 있는 조작의 목록에 대해, 그 개요・요청 방법・응답을 게재합니다. API 사양을 이해하고, 각종 프로그램에서 동일한 요청을 송신함으로써, 각종 프로그래밍 언어로부터의 이용이나 제어 프로그램과의 연계가 가능해집니다.

> 상세한 API 사양의 도큐먼트는, External API Interface를 참조하십시오.

6.3.3.1 공통 사항

다음 목록은, 이후의 요청 예에 게재되어 있는 내용에 대해, 공통이 되는 사항입니다.

- 는, SpeeDBee Synapse가 가동하고 있는 호스트・포트로 치환하십시오.
- <액세스 토큰>은,「액세스 토큰 발행」에서 발행한 액세스 토큰으로 치환하십시오.
- <패널 ID>는, 취득하고 싶은 패널의 ID를 지정합니다. 그 패널에 등록되어 있는 모든 컴포넌트의 실행 상태를 취득합니다. 패널 ID는,「전 컴포넌트의 실행 상태 취득」에 의해 취득할 수 있습니다.
- <컴포넌트 ID>는, 실행 상태를 취득하고 싶은 컴포넌트의 ID를 지정합니다. 컴포넌트 ID는,「전 컴포넌트의 실행 상태 취득」에 의해 취득할 수 있습니다.

6.3.3.2 전 컴포넌트의 실행 상태 취득

- 개요

  SpeeDBee Synapse에 등록되어 있는 모든 컴포넌트의 실행 상태를 취득합니다. 요청 URL

-

```
http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status
```

- curl 커맨드에 의한 요청

  Linux

```
$ curl -X "GET" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H
"accept: application/json" -H "X-hive-api-key: <액세스 토큰>"
```

  Windows

```
$ curl.exe -X "GET" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H
"accept: application/json" -H "X-hive-api-key: <액세스 토큰>"
```

-For Reference Only-

- 응답

코드	설명
200	정상

코드	설명
403	인증 에러
503	SpeeDBee Synapse가 요청을 처리할 수 없는 상태

정상 응답은, 이하의 형식입니다.

```
{
  "status": "SpeeDBee Synapse의 스테이터스",
  "components": [
    {
      "id": "컴포넌트 ID",
      "name": "컴포넌트명",
      "panel_id": "컴포넌트가 등록되어 있는 패널의 ID",
      "panel_name": "컴포넌트가 등록되어 있는 패널의 이름",
      "status": "컴포넌트의 스테이터스",
      "errors": [
        {
          "id": "에러 ID",
          "parameter": ["에러 메시지"]
        }
      ]
    }
  ]
}
```

※1 "components"에는, 컴포넌트의 실행 상태가 격납됩니다. 컴포넌트가 존재하지 않는 경우는, 빈 배열이 됩니다.

※2 "errors"에는, 컴포넌트의 에러 정보가 격납됩니다. 에러가 발생하지 않은 경우는, 빈 배열이 됩니다.

6.3.3.3 개별 컴포넌트의 실행 상태 취득

AI 번역본
-For Reference Only-

- 개요

  SpeeDBee Synapse에 등록되어 있는 개별 컴포넌트의 실행 상태를 취득합니다. 요청 URL

-

```
http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status
```

- curl 커맨드에 의한 요청
  - 패널마다

Linux

```
$ curl -X "POST" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H
"accept: application/json" -H "X-hive-api-key: <액세스 토큰>" -H 'Content-Type:
application/json' -d "{\"panel_ids\": [<패널 ID>]}"
```

Windows

```
$ curl.exe -X "POST" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H "accept:
application/json" -H "X-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"panel_ids\": [<패널 ID>]}"
```

- 컴포넌트마다

Linux

```
$ curl -X "POST" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H
"accept: application/json" -H "X-hive-api-key: <액세스 토큰>" -H 'Content-Type:
application/json' -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

Windows

```
$ curl.exe -X "POST" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/status" -H "accept:
application/json" -H "X-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

- 응답

코드	설명
200	정상
403	인증 에러
422	유효성 검사 오류
503	SpeeDBee Synapse가 요청을 처리할 수 없는 상태

For Reference Only

정상 응답은, 이하의 형식입니다.

```
{
   "status": "SpeeDBee Synapse의 스테이터스",
   "components": [
      {
         "id": "컴포넌트 ID",
         "name": "컴포넌트명",
         "panel_id": "컴포넌트가 등록되어 있는 패널의 ID",
         "panel_name": "컴포넌트가 등록되어 있는 패널의 이름",
         "status": "컴포넌트의 스테이터스",
         "errors": [
            {
               "id": "에러 ID",
               "parameter": ["에러 메시지"]
            }
         ]
      }
   ]
}
```

※1 "components"에는, 지정된 컴포넌트의 실행 상태가 격납됩니다. 지정된 컴포넌트가 존재하지 않는 경우는, 빈 배열이 됩니다.

※2 "errors"에는, 컴포넌트의 에러 정보가 격납됩니다. 에러가 발생하지 않은 경우는, 빈 배열이 됩니다.

6.3.3.4 컴포넌트 일괄 실행 개시

- 개요

  SpeeDBee Synapse에 등록되어 있는 컴포넌트를 개시합니다.

- 요청 URL

```
http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/start
```

- curl 커맨드에 의한 요청
  - 패널마다

AI 번역본
-For Reference Only-

Linux

```
$ curl -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/start" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"panel_ids\": [<패널 ID>]}"
```

Windows

```
$ curl.exe -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/start" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"panel_ids\": [<패널 ID>]}"
```

- 컴포넌트마다

  Linux

```
$ curl -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/start" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

  Windows

```
$ curl.exe -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/start" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

- 응답

코드	설명
200	정상
400	포인트 부족에 의해 컴포넌트의 실행을 개시할 수 없음
403	인증 에러
422	유효성 검사 오류
503	SpeeDBee Synapse가 요청을 처리할 수 없는 상태

정상 응답은, 이하의 형식입니다.

```
[
  {
    "id": "컴포넌트 ID",
    "panel_id": "패널 ID",
    "success": "컴포넌트 개시의 성공 여부",
    "error": "에러 정보"
  }
]
```

AI 번역본
-For Reference Only-

※ 정상 응답의 배열에는, 요청에서 지정된 컴포넌트의 목록이 격납됩니다. 지정된 컴포넌트가 존재하지 않는 경우는, 빈 배열이 됩니다.

6.3.3.5 컴포넌트 일괄 실행 정지

- 개요

SpeeDBee Synapse에 등록되어 있는 컴포넌트를 정지합니다.

- 요청 URL

```
http://<SpeeDBee Synapse의 호스트 · 포트>/ext_api/v1/components/stop
```

- curl 커맨드에 의한 요청
  - 패널마다

Linux

```
$ curl -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/stop" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"panel_ids\": [<패널 ID>]}"
```

Windows

```
$ curl.exe -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/stop" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"panel_ids\": [<패널 ID>]}"
```

  - 컴포넌트마다

Linux

```
$ curl -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/stop" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

Windows

```
$ curl.exe -X "PUT" "http://<SpeeDBee Synapse의 호스트:포트>/ext_api/v1/components/stop" -H
"accept: application/json" -H "x-hive-api-key: <액세스 토큰>" -H "Content-Type:
application/json" -d "{\"component_ids\": [\"<컴포넌트 ID>\"]}"
```

AI 번역본

- 응답

코드	설명
200	정상
403	인증 에러
422	유효성 검사 오류
503	SpeeDBee Synapse가 요청을 처리할 수 없는 상태

정상 응답은, 이하의 형식입니다.

```
[
  {
    "id": "컴포넌트 ID",
    "panel_id": "패널 ID",
    "success": "컴포넌트 정지의 성공 여부",
    "error": "에러 정보"
  }
]
```

※ 정상 응답의 배열에는, 요청에서 지정된 컴포넌트의 목록이 격납됩니다. 지정된 컴포넌트가 존재하지 않는 경우는, 빈 배열이 됩니다.

AI 번역본
-For Reference Only-

6.4 Grafana와의 연계

6.4.1 Grafana 연계의 개요

Grafana는, 시스템의 데이터를 가시화하기 위한 툴입니다. Grafana와 연계함으로써, SpeeDBee Synapse가 수집한 데이터를 사용하여 그래프를 작성할 수 있습니다.

Grafana에 대한 상세는, 공식 사이트를 참조하십시오.

주의사항

본 장의 설명에서는, 버전 11.5.2의 Grafana를 사용하고 있습니다. 버전의 차이에 의해, 조작의 내용에 차이가 있는 경우가 있습니다.

SpeeDBee Synapse의 Grafana 연계 기능에서는, Grafana의 대시보드에「SpeeDBee Synapse」라는 폴더를 작성합니다. 이 폴더는 Synapse가 관리・갱신하기 때문에, Grafana 화면에서 수동으로 변경하지 않도록 주의하십시오.
변경・저장한 경우, Synapse로부터의 Grafana 갱신 조작에 의해, 사용자에 의한 변경은 파기됩니다. 수동으로 대시보드를 관리하고 싶은 경우는,「SpeeDBee Synapse」폴더 이외의 장소에 임의의 대시보드를 추가하십시오.

6.4.2 Grafana의 설치

・본 항목의 순서는, Grafana를 설치하는 환경에서 수행하십시오.
・이후의 Linux의 순서는, 시스템 관리 권한을 필요로 하는 조작에는, 모두 sudo를 사용하고 있습니다.
・Linux에서, root 권한을 가진 사용자로 조작하는 경우는, sudo를 제외한 커맨드를 입력하십시오.
・Windows에서는, Windows PowerShell을「관리자로 실행」으로 기동해 둡니다.
・관리자 권한에 대해서는, OS의 시스템 관리자에게 상담하십시오.

사내 열람용
-For Reference Only-

Data Flow Controller판에서는, 초기 상태로 Grafana가 설치되어 있기 때문에, 설치는 불필요합니다.

1. Grafana를 설치합니다. 설치 순서는, 공식 사이트를 참고로 하십시오.
2. 이하의 커맨드를 실행하여, 플러그인의 설치 대상 디렉터리를 작성합니다.

Linux

```
$ sudo mkdir -p /var/lib/grafana/plugins/SpeeDBSynapse/
```

Windows

```
PS >New-Item -ItemType Directory -Path "C:\Program
Files\GrafanaLabs\grafana\data\plugins\SpeeDBeeSynapse" -Force
```

Standalone Linux Binaries

```
$ sudo mkdir-p <전개 대상 디렉터리>/grafana-9.1.3/data/plugins/SpeeDBeeSynapse/
```

3. 임의의 디렉터리에, 「 synapse_gplugin.tar.gz 」를 배치합니다.

> 「 synapse_gplugin.tar.gz 」는, Grafana와 SpeeDBee Synapse가 연계하기 위한 플러그인입니다. SpeeDBee Synapse의 설치 프로그램과 함께 배포하고 있습니다. 불명한 경우는 문의해 주십시오.
>
> 문의 폼: https://www.saltyster.com/contact/

4. 이하의 커맨드를 실행하여, 「 synapse_gplugin.tar.gz 」를 배치한 디렉터리로 이동합니다.

Linux

```
$ cd <synapse_gplugin.tar.gz를 배치한 디렉터리>
```

Windows

```
PS >cd <synapse_gplugin.tar.gz를 배치한 디렉터리>
```

Standalone Linux Binaries

```
$ cd <synapse_gplugin.tar.gz를 배치한 디렉터리>
```

5. 이하의 커맨드를 실행하여, 플러그인을 설치합니다.

AI 번역본
For Reference Only

Linux

```
$ sudo tar xzf synapse_gplugin.tar.gz -C /var/lib/grafana/plugins/SpeeDBSynapse/
```

Windows

```
PS >tar xzf synapse_gplugin.tar.gz -C "C:\Program
Files\GrafanaLabs\grafana\data\plugins\SpeeDBeeSynapse"
```

Standalone Linux Binaries

```
$ sudo tar xzf synapse_gplugin.tar.gz -C <전개 대상 디렉터리>/grafana-
9.1.3/data/plugins/SpeeDBeeSynapse/
```

6. 이하의 커맨드를 실행하여, Grafana를 재기동합니다.

Linux

```
$ sudo systemctl restart grafana-server
```

Windows

```
PS >Restart-Service grafana
```

Standalone Linux Binaries

```
$ cd <전개 대상 디렉터리>/grafana-9.1.3
$ sudo ./bin/grafana-server
```

> systemd에의 서비스 등록

systemctl로 서비스를 기동하고 싶은 경우는, 다음 순서에 따라 systemd에 서비스를 등록하십시오.

a. 서비스 유닛 파일을 작성

/lib/systemd/system/grafana-server.service

```
[Unit]
Description=Grafana server

[Service]
Type=simple
WorkingDirectory=<전개 대상 디렉터리>/grafana-9.1.3
ExecStart=<전개 대상 디렉터리>/grafana-9.1.3/bin/grafana-server
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

b. systemd에 서비스를 읽어 들이게 하고, 서비스를 기동.

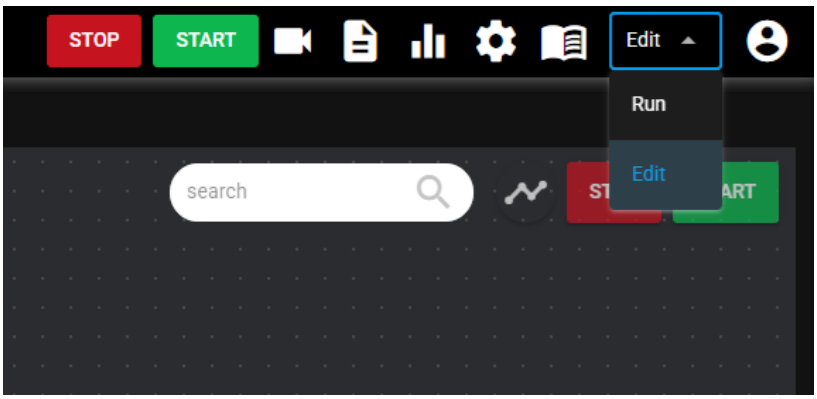
```
$ sudo systemctl daemon-reload
$ sudo systemctl start grafana-server
```

-For Reference Only-

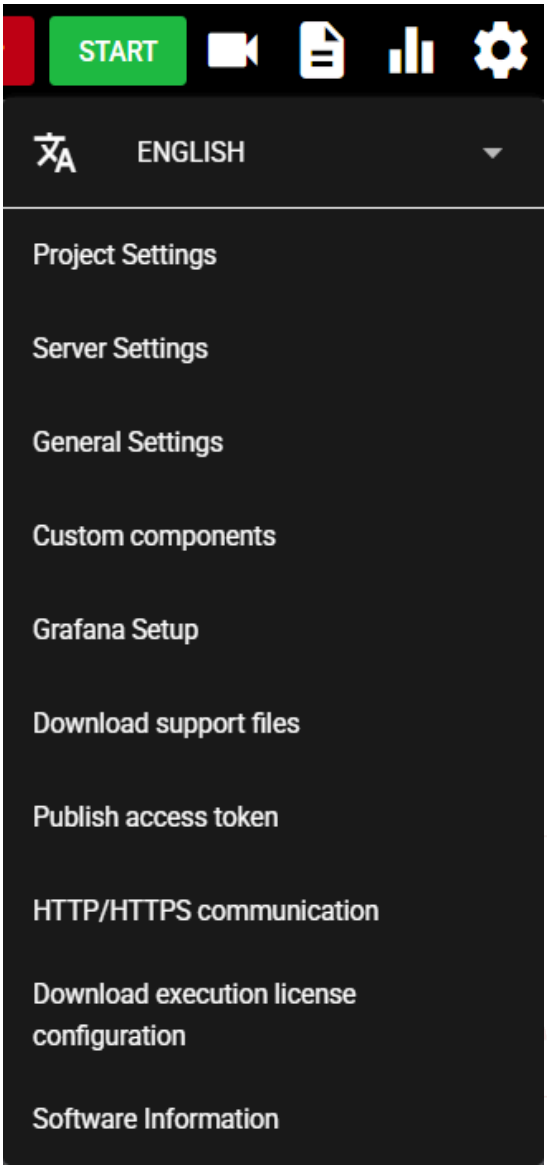
6.4.3 Grafana의 셋업

본 항목의 순서 전에,「Grafana의 설치」가 완료되어 있을 필요가 있습니다.

1. SpeeDBee Synapse의 Edit 모드의 화면을 엽니다.

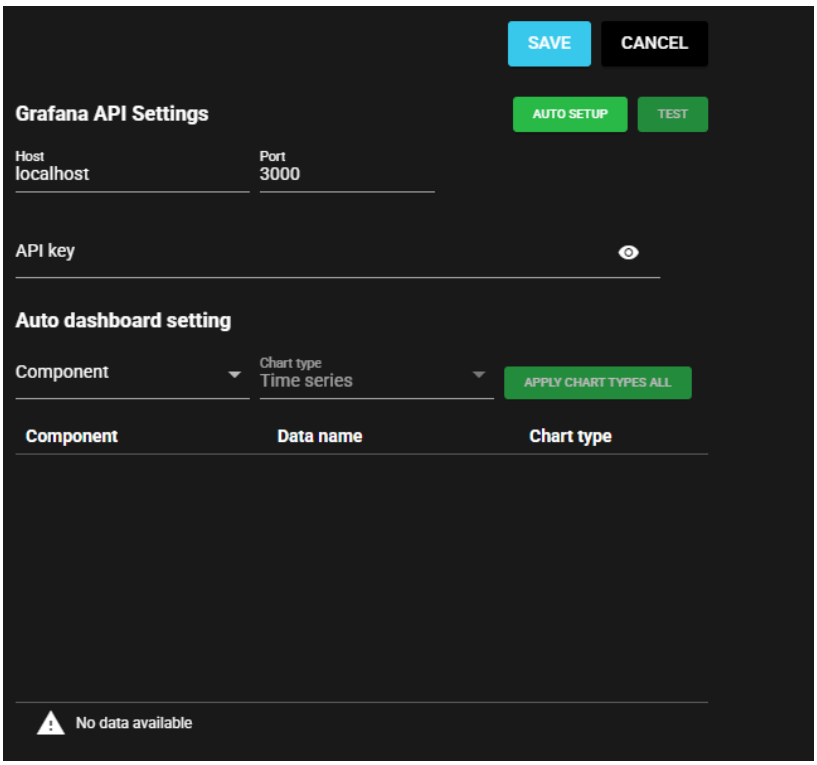


2. 설정 메뉴 아이콘을 누르고,「Grafana 셋업」을 선택합니다.



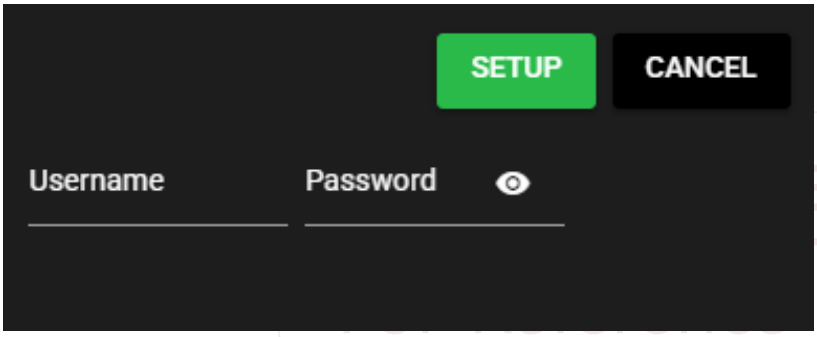
번역본
ference Only-

3. 다음 항목을 입력하고,「자동 셋업」을 누릅니다.



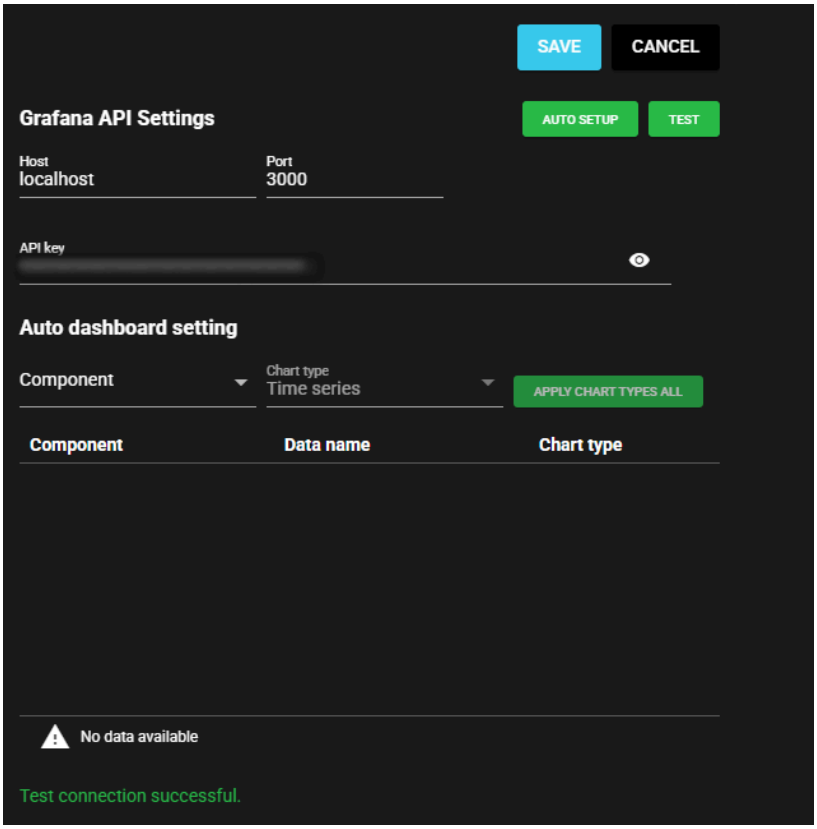
항목	설명
호스트	Grafana를 설치한 환경의 호스트명 또는 IP 주소
포트	Grafana가 가동하고 있는 포트 ※초기 설정에서는, Grafana는 포트 3000에서 가동합니다.

4. 다음 항목을 입력하고,「 셋업 」을 누릅니다.

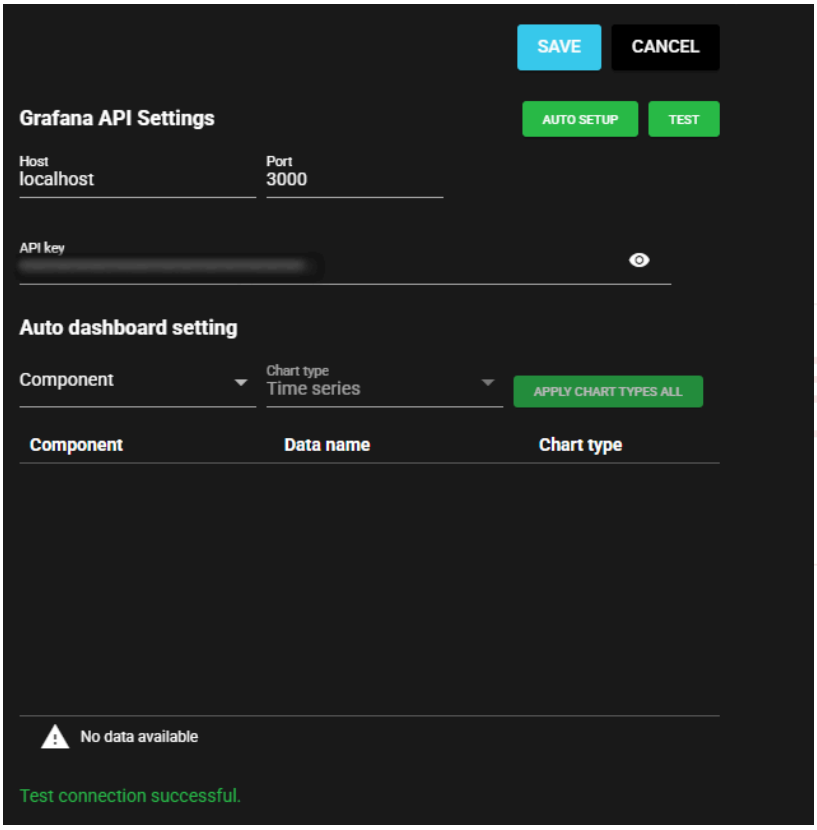


항목	설명
사용자명	Grafana에의 로그인에 필요한 사용자명 ※초기 설정값은 "admin"입니다.
패스워드	Grafana에의 로그인에 필요한 패스워드 ※초기 설정값은 "admin"입니다.

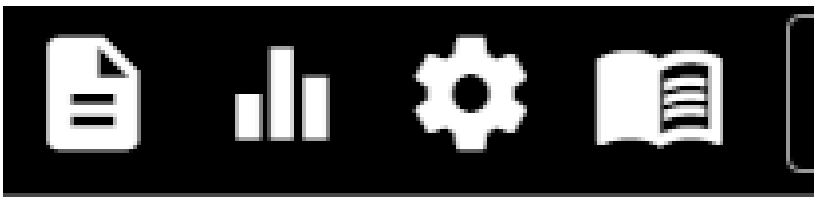
5. 셋업이 완료된 경우, API 키가 자동으로 입력되어 있습니다.
   또한 「 테스트 」를 누르면, 화면 하부에 「 테스트 접속에 성공했습니다. 」라고 표시됩니다.



6. 「 저장 」을 누릅니다.



7. 헤더의 Grafana 아이콘을 누르면, Grafana 화면으로 이동할 수 있게 되어 있습니다.

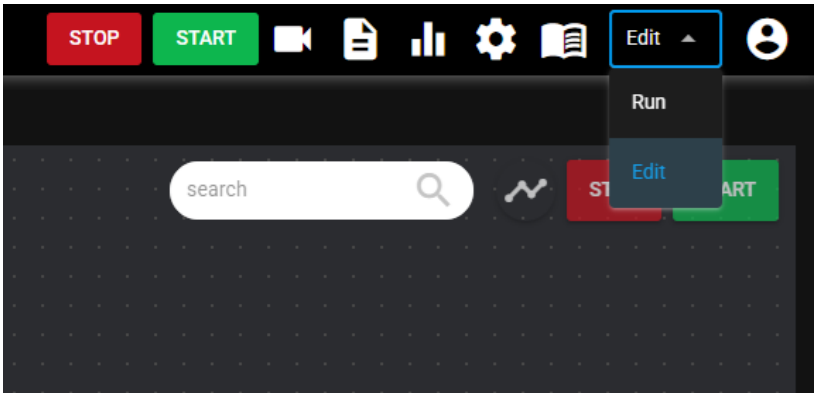


6.4.4 그래프의 작성

본 항목의 순서 전에,「Grafana의 설치」,「Grafana의 셋업」이 완료되어 있을 필요가 있습니다.

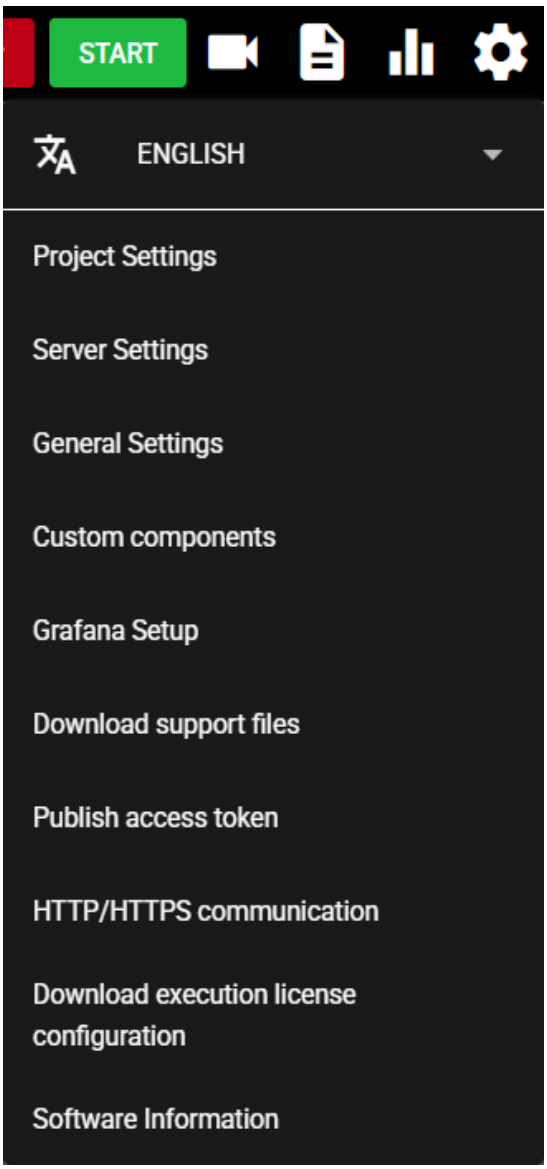
6.4.4.1 데이터의 등록

1. SpeeDBee Synapse의 Edit 모드의 화면을 엽니다.



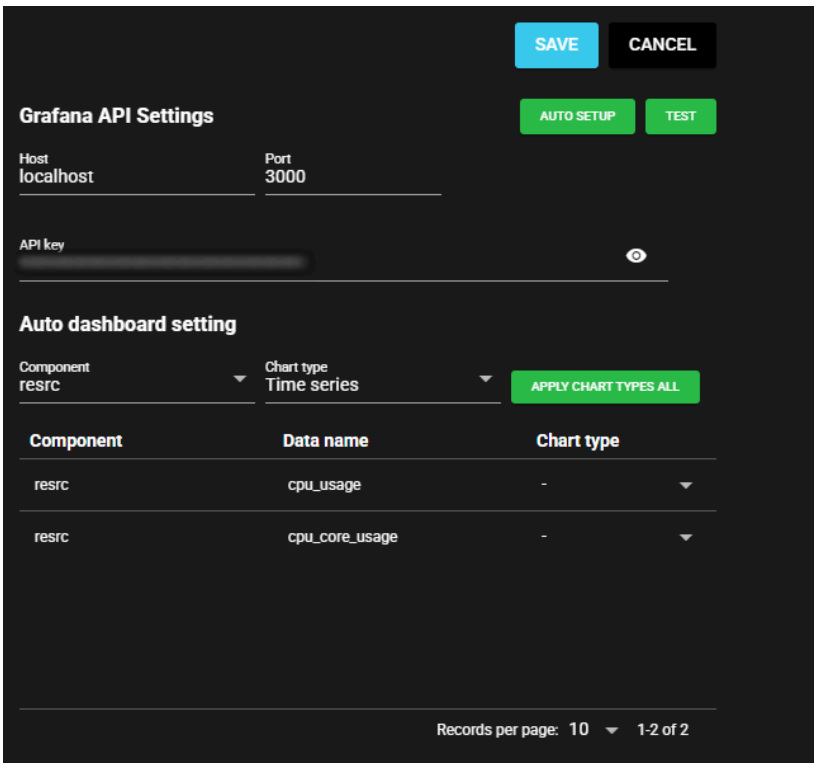
2. 설정 메뉴 아이콘을 누르고,「Grafana 셋업」을 선택합니다.

AI 번역본
-For Reference Only-

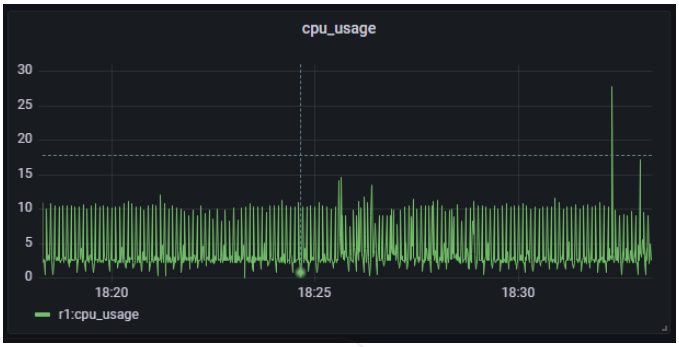


3. 컴포넌트와 그래프 종류를 선택합니다. 그래프 종류는 개별로 선택, 혹은 「 그래프 종류 일괄 적용 」을 사용하여 일괄로 변경하는 것도 가능합니다.

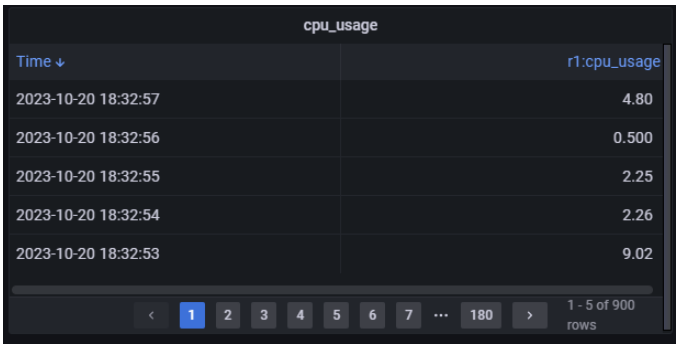
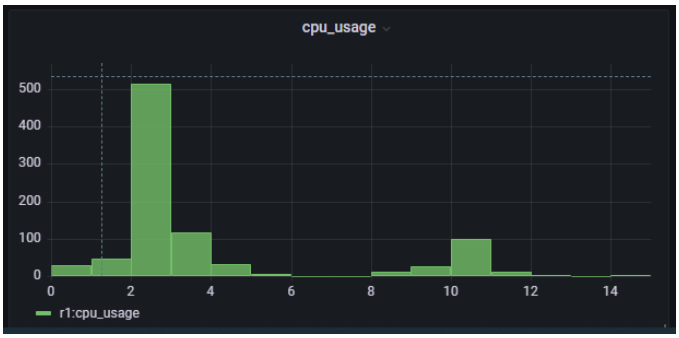
AI 번역본
-For Reference Only-



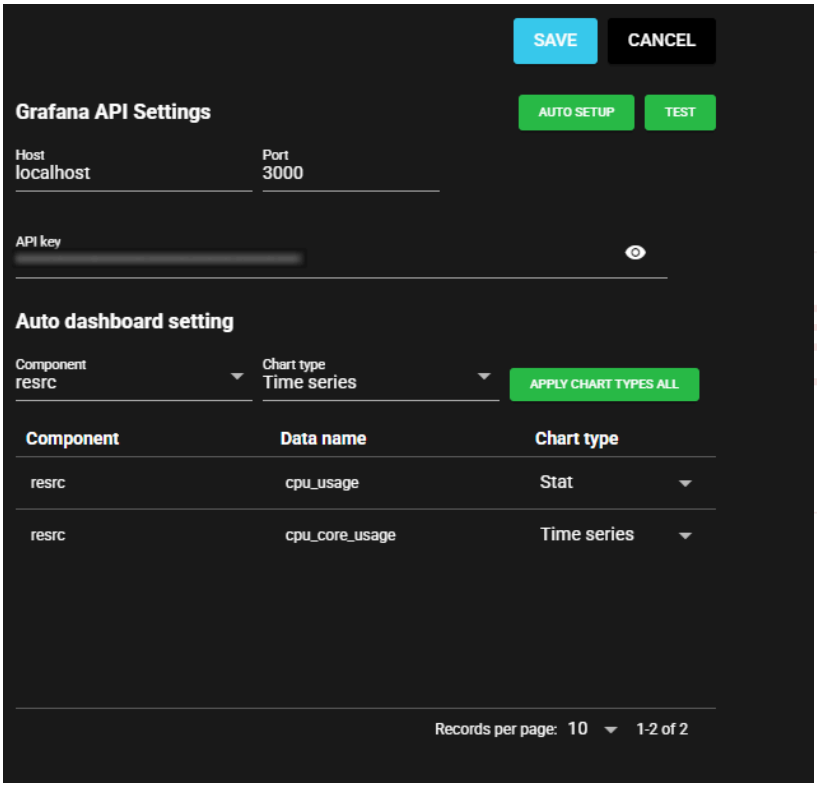
그래프 종류는 다음과 같습니다.

종류	설명	표시 예
꺾은선 그래프	값을 시계열로 표시합니다.	
현재값	꺾은선 그래프를 배경으로 하고, 현재값을 표시합니다.	

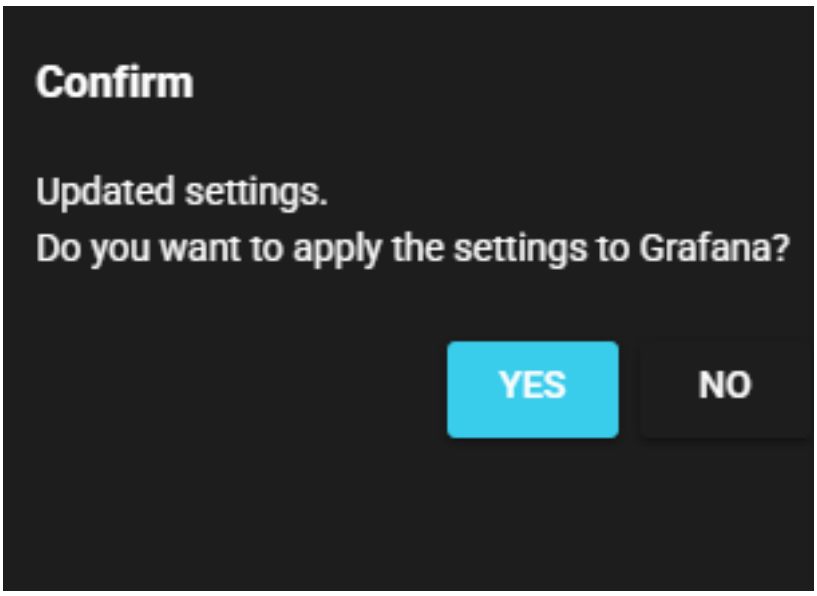
AI 번역본
-For Reference Only

종류	설명	표시 예
테이블	값을 테이블 형식으로 표시합니다.	
히스토그램	값의 분포를 계산하여, 막대 그래프로서 표시합니다.	
-	그래프를 작성하지 않습니다.	

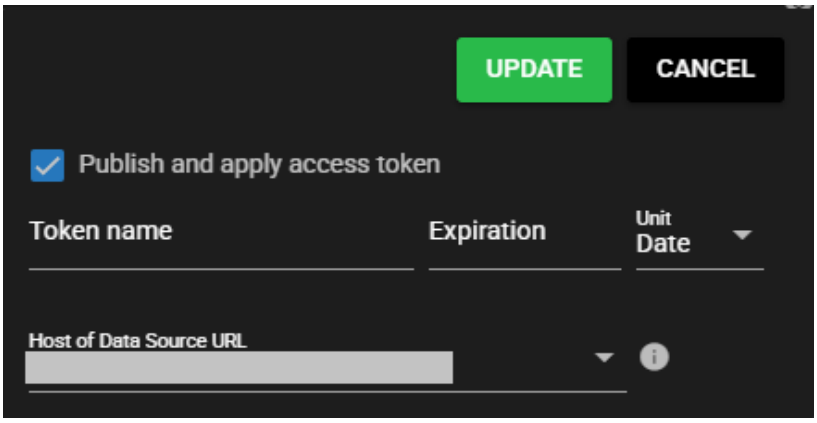
4. 「 저장 」을 누릅니다.



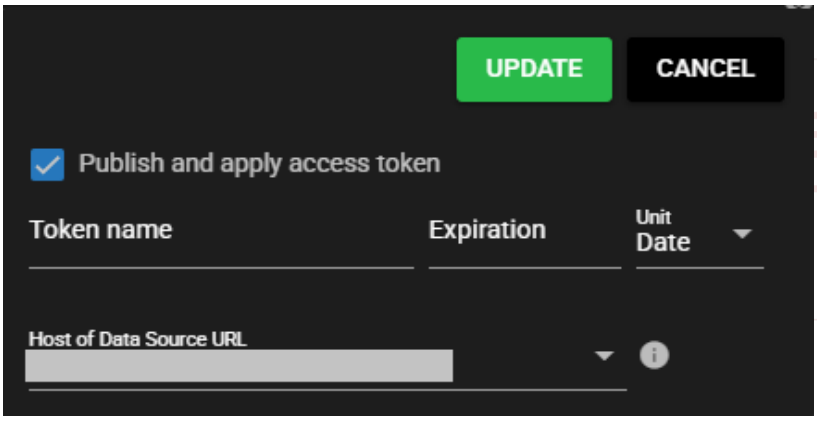
5. 「 예 」를 누릅니다.



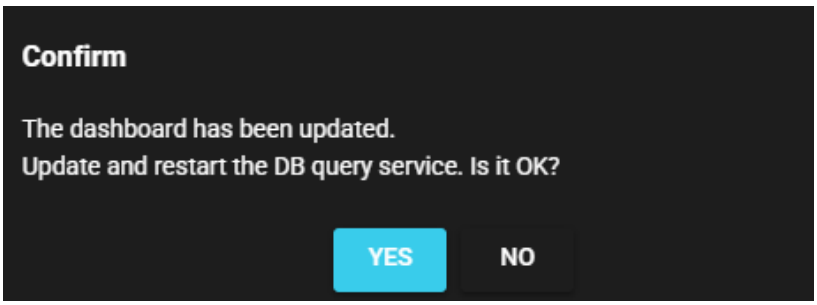
6. 「 액세스 토큰을 발행하여 적용한다 」를 선택하고, 토큰의 이름과 유효 기한을 입력합니다. 또한, 「 데이터 소스 URL의 호스트 」는 Grafana에서 본 SpeeDBee Synapse가 설치된 디바이스의 IP 주소 또는 호스트명이 됩니다.



7. 「 갱신 」을 누릅니다.



8. 「 예 」를 누릅니다.
※액세스 토큰을 발행하여 갱신한 경우, DB 쿼리 서비스의 설정 갱신과 재기동이 필요하게 됩니다. 수동으로 갱신, 기동하는 경우는, 「 DB 쿼리 서비스의 기동 」을 참조하십시오.

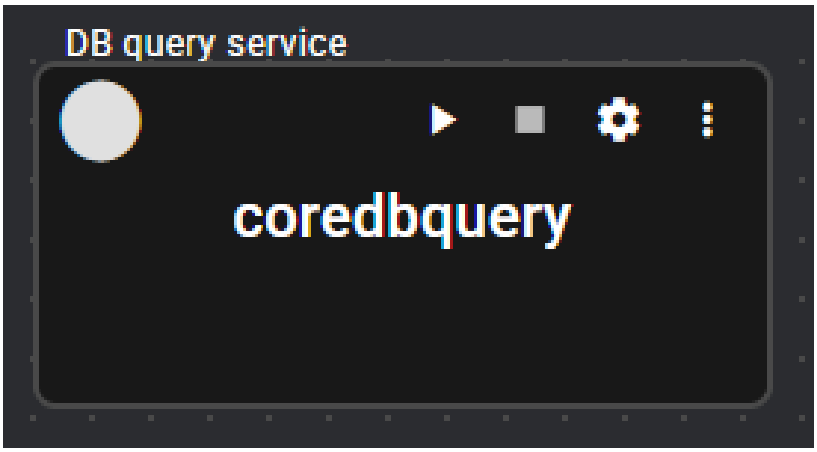


6.4.4.2 DB 쿼리 서비스의 기동

1. 「 시스템 」의 패널 탭을 누르고, 시스템 패널을 엽니다.

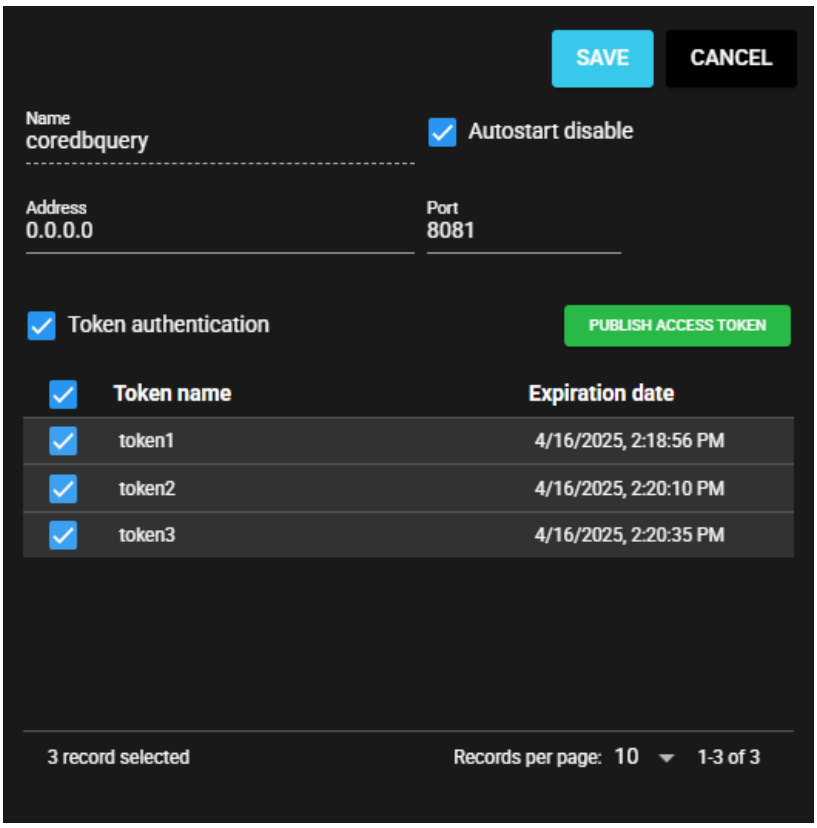


2. DB 쿼리 서비스의 설정 아이콘을 누릅니다.

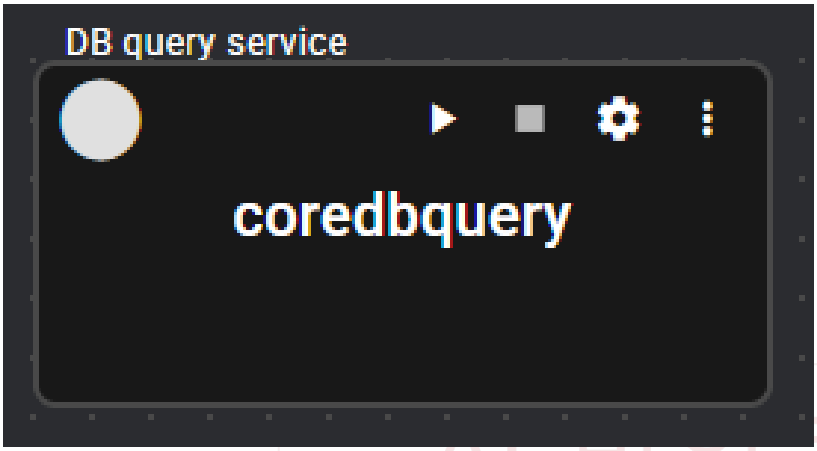


3. 「 토큰 인증 」을 ON으로 하고, 모든 토큰을 선택하여 「 저장 」을 누릅니다.

AI 번역본
-For Reference Only-



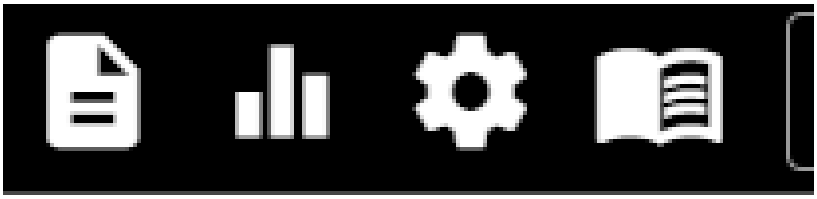
4. 코어 DB 쿼리의 기동 아이콘을 누릅니다. 이미 기동하고 있는 경우는, 정지 아이콘을 누른 후에, 기동 아이콘을 누릅니다.



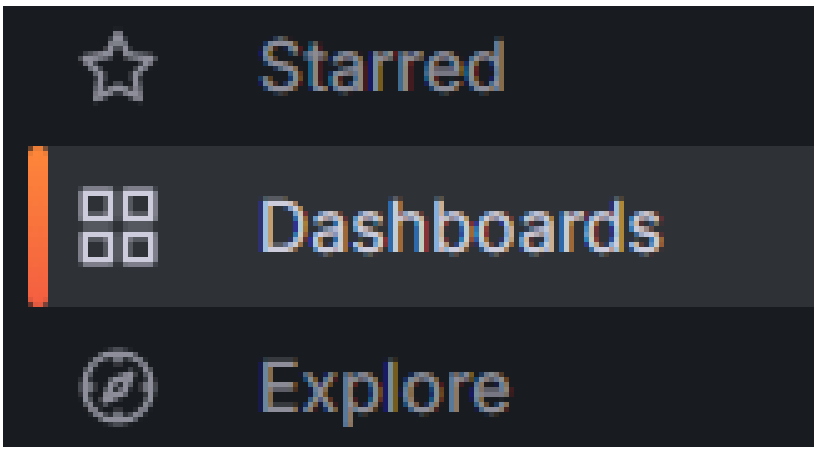
AI 번역본
-For Reference Only-

6.4.5 그래프의 열람

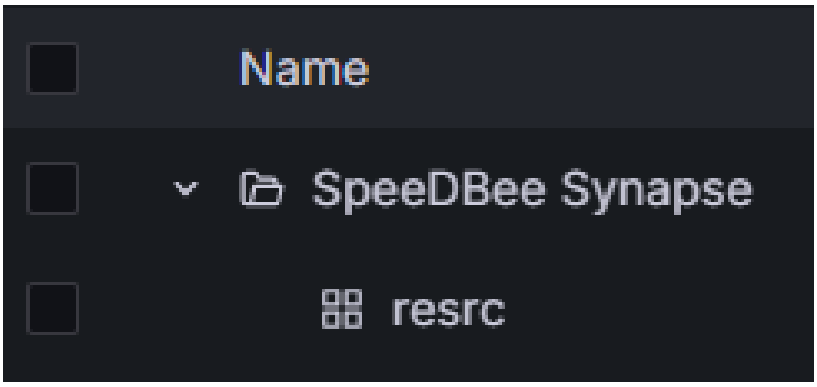
1. 헤더의 Grafana 아이콘을 누르고, Grafana 화면을 엽니다.



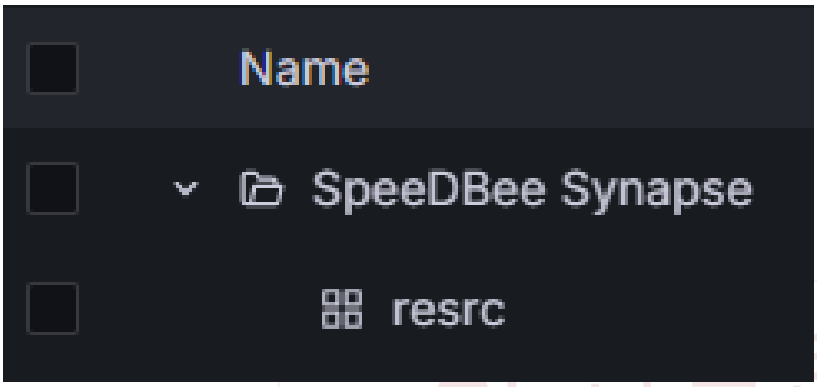
2. 좌측 메뉴의 「 Dashboards 」 아이콘을 누릅니다.



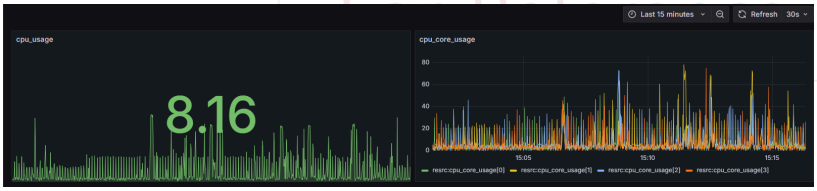
3. 「 SpeeDBee Synapse 」라는 폴더 안에, 「 데이터의 등록 」에서 등록한 컴포넌트가 목록으로 표시됩니다.



4. 컴포넌트를 선택합니다.



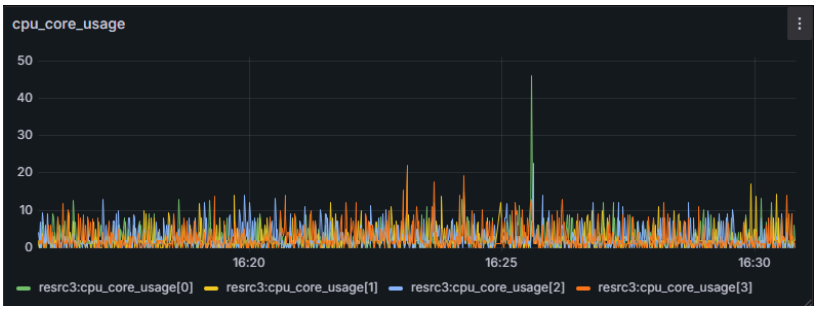
5. 선택한 컴포넌트의 각 데이터의 그래프가, 목록으로 표시됩니다.



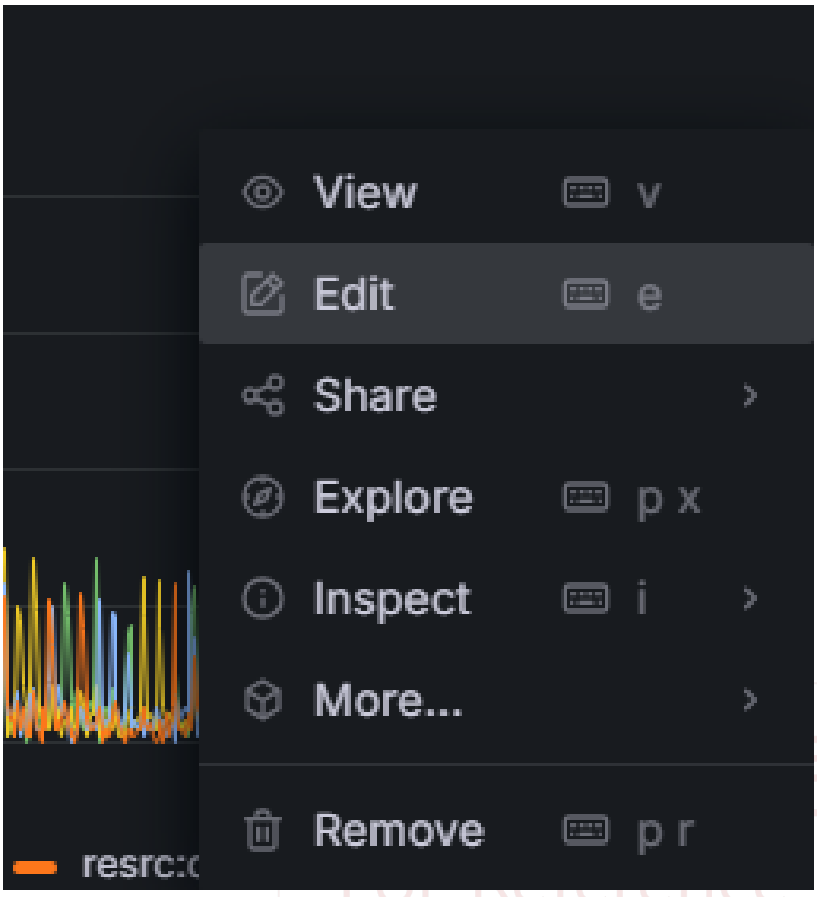
6.4.6 그래프의 데이터 건수 상한의 변경

데이터가 일부분밖에 표시되어 있지 않은 경우, 데이터의 건수가 상한을 초과하고 있는 경우가 있습니다. 아래 순서로, 데이터 건수의 상한을 변경할 수 있습니다.

1. 「 그래프의 열람 」의 순서를 실시하여, 그래프의 목록을 표시합니다.
2. 그래프 우측 상단의 메뉴 아이콘을 누릅니다.

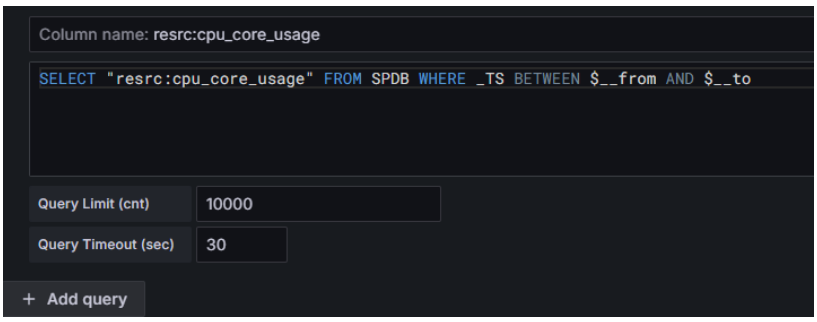


3. 메뉴가 표시됩니다. 그중에서 「 Edit 」를 선택합니다.

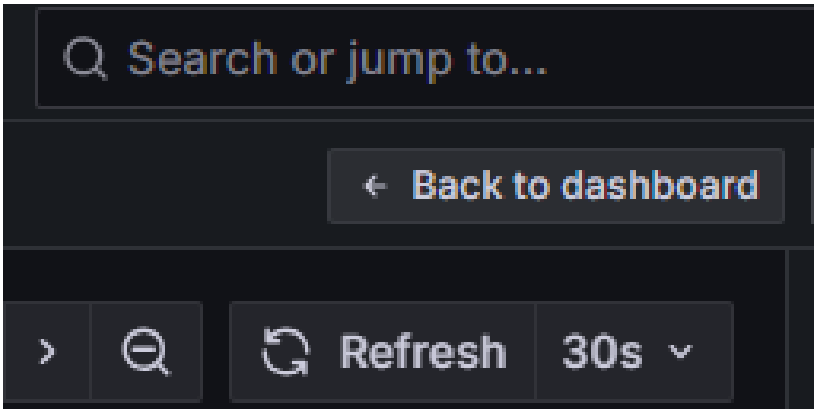


4. 그래프의 설정 화면이 표시됩니다. 이 화면의 「 Query Limit (cnt) 」라는 설정 항목에서, 데이터 건수의 상한이 설정되어 있습니다. 값을 변경하면, 그래프에 반영됩니다.
（ 「 Query Limit (cnt) 」의 디폴트값은 10000입니다. ）

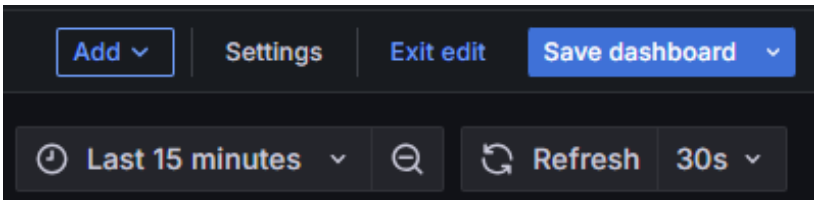
For Reference Only-



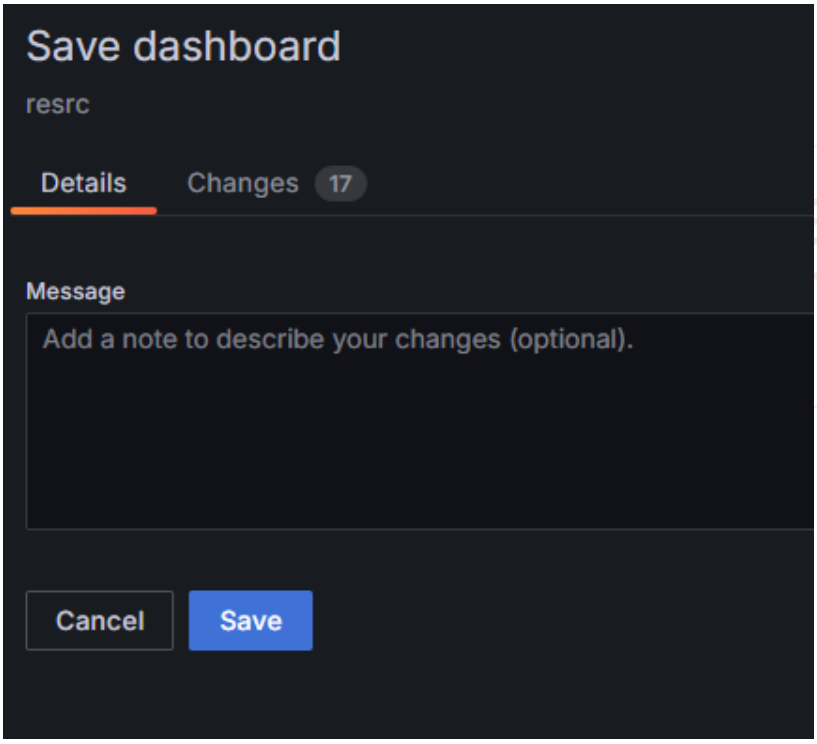
5. 값을 변경한 후는, ESC 키를 누르거나, 화면 우측 상단의 「 Back to dashboard 」 아이콘을 누릅니다.



6. 그래프 목록의 화면으로 돌아갑니다. 화면 우측 상단의 「 Save dashboard 」 아이콘을 누릅니다.



7. 「 Save 」를 누르면, 설정이 저장됩니다.

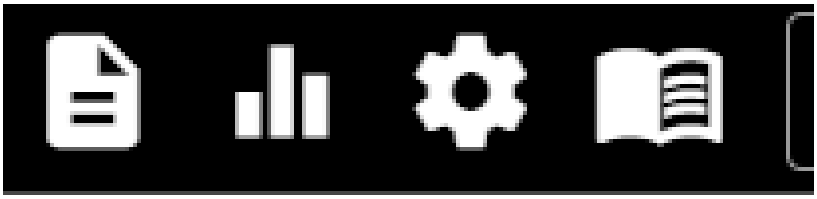


> ⚠ 설정이 리셋되는 타이밍
>
> 설정을 변경한 후,「데이터의 등록」의 순서를 실시하면, 설정이 리셋되므로 주의하십시오.

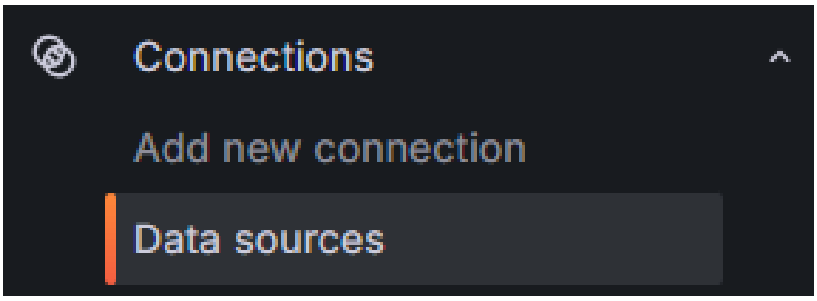
6.4.7 Grafana 데이터 소스의 수동 설정

Grafana를 수동으로 조작하여, 데이터 참조를 수행하는 방법에 대해 설명합니다.
이 설정은, Grafana 셋업에서 설정 완료라면 특별히 설정을 수행할 필요는 없습니다.

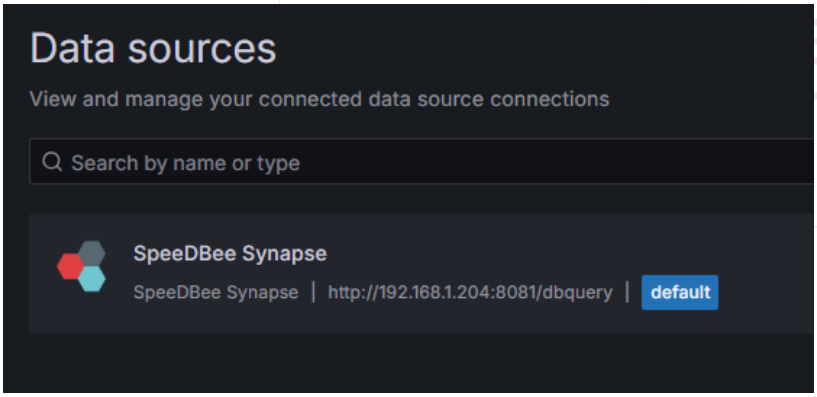
1. 「액세스 토큰 발행」을 참고로, 액세스 토큰을 발행합니다.
   ※발행한 액세스 토큰은 나중에 사용하므로, 메모해 둡니다.
2. 「DB 쿼리 서비스의 기동」의 순서에 따라, DB 쿼리 서비스를 재기동합니다.
3. 헤더의 Grafana 아이콘을 누르고, Grafana 화면을 엽니다.



4. 좌측 메뉴의「Connections」아이콘에 커서를 맞추고, 표시된 메뉴 중에서「Data sources」를 선택합니다.

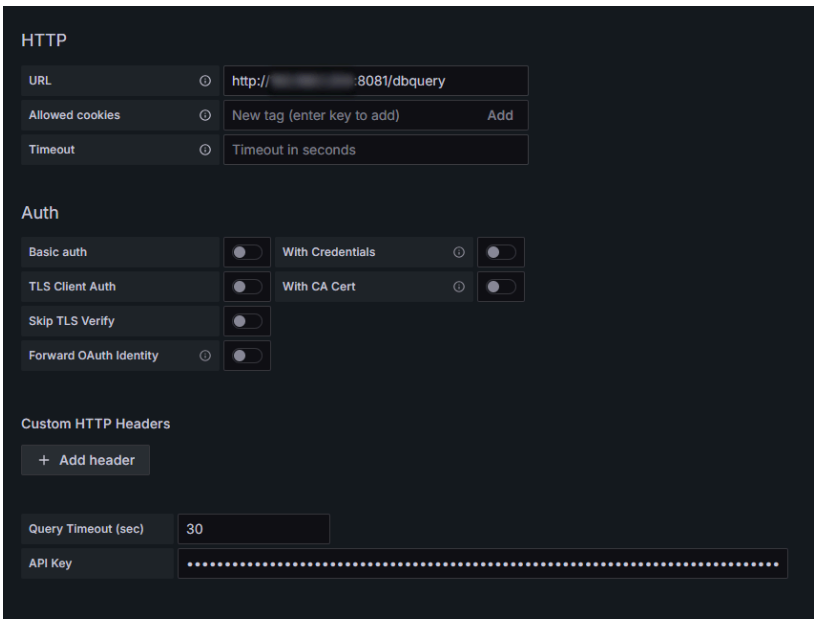


5. 「SpeeDBee Synapse」를 선택합니다.

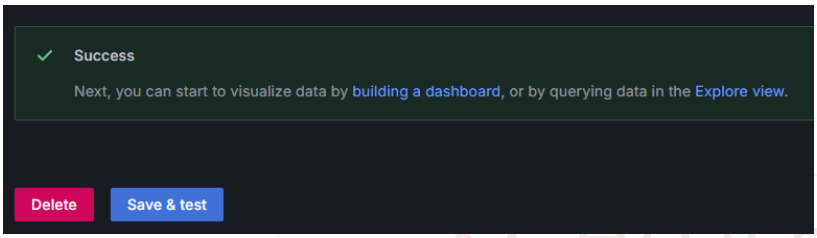


6. 이하의 항목을 입력합니다.

항목	설명
URL	「 http://{ SpeeDBee Synapse가 가동하고 있는 환경의 IP 주소 }:{ DB 쿼리 서비스가 가동하고 있는 포트 }/dbquery 」라고 입력하십시오. ※DB 쿼리 서비스가 가동하고 있는 포트는, DB 쿼리 서비스 컴포넌트의 설정 항목 「 포트 」에서 확인, 변경할 수 있습니다. 상세는 「 DB 쿼리 서비스 」를 참조하십시오.
API Key	위의 순서에서 발행한 액세스 토큰을 입력하십시오.



7. 「 Save & test 」를 누르고, "Success"라는 메시지가 표시되면, 데이터 소스의 설정은 완료입니다.

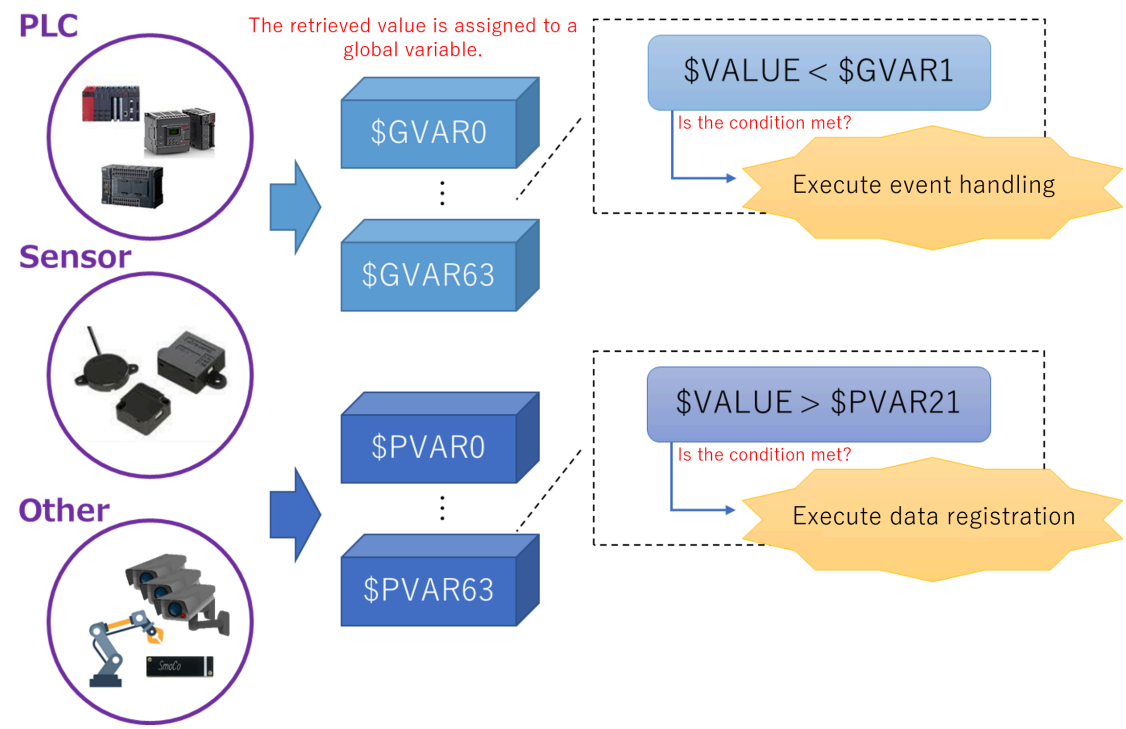


AI 번역본
-For Reference Only-

6.5 글로벌 변수

6.5.1 글로벌 변수의 개요

글로벌 변수는, 빈번하게 변화하는 데이터를 메모리상의 변수에 격납하는 기능입니다. 이 변수를 SpeeDBee Synapse의 각종 기능 (수치 변환, 등록 조건, 컴포넌트의 연산식, 조건식, SQL 등) 에서 이용할 수 있습니다.
이에 의해, 시스템 가동 중의 조건식 임계값의 갱신이나 특정 데이터값에 의한 다른 데이터 등록의 억제 등이 가능해집니다.



글로벌 변수를 이용할 수 있는 것은, 연산식, 조건식, SQL 내가 됩니다. 연산식, 조건식에 대해서는, 이쪽을 참조하십시오.

SQL에 대해서는, 이쪽을 참조하십시오.

AI 번역본
For Reference Only

6.5.2 글로벌 변수의 구문

글로벌 변수에는, 통상의 글로벌 변수와 영속화 글로벌 변수의 2종류의 지정 방법이 있습니다. 어느 쪽이든 64개의 변수를 사용할 수 있습니다.
통상의 글로벌 변수는, SpeeDBee Synapse의 코어를 정지 (가동 중인 머신을 재기동하는 등) 하면, 값은 손실됩니다. 영속화 글로벌 변수는, SpeeDBee Synapse의 코어를 정지해도, 다음 기동 시에 최종값을 참조할 수 있습니다.

- $GVAR0 ~ $GVAR63 (글로벌 변수)

  GVAR (글로벌 변수) 는, SpeeDBee Synapse의 코어를 재기동할 때마다, 0으로 초기화됩니다. 값의 대입이 수행될 때까지 0인 채로 됩니다.

- $PVAR0 ~ $PVAR63 (영속화 글로벌 변수)

PVAR（영속화 글로벌 변수）는, SpeeDBee Synapse의 코어를 처음 기동할 때에, 0으로 초기화됩니다. 값의 대입이 수행되면, 코어를 재기동해도 값을 유지합니다.

SpeeDBee Synapse의 재기동 시에, 전회 기동 시에 대입된 값을 사용하고 싶은 경우는, 영속화 글로벌 변수를 사용하십시오.

1. 대입 방법

```
GVAR0 := $VALUE
```

```
PVAR15 := ($VALUE * 20)
```

※대입할 글로벌 변수의 선두에는「$」는 필요 없습니다.

- 수치 변환에서의 대입 시의 주의점

  ```
  PVAR0 := $VALUE * 2
  ```

  위와 같이 대입을 지정하면, 등록되는 값은 $VALUE에 2를 승산한 값이 됩니다. 등록되는 값이 변화하는 것에 주의하십시오.

- 조건식에서의 대입 시의 주의점

  ```
  GVAR0 := $VALUE
  ```

  조건식에서는, 대입만을 지정할 수는 없습니다. 조건식 내에서, 대입하고 싶은 경우는 이하와 같이 조건식으로서의 지정이 필요합니다.

  ```
  0 < (GVAR0 := $VALUE)
  ```

  이 경우, $VALUE가 0보다 큰 경우 조건 성립이 됩니다. GVAR0에의 대입은 항상 수행됩니다.

2. 참조 방법

```
$GVAR0 < $VALUE
```

```
$PVAR15 > ($VALUE + 20)
```

※참조할 글로벌 변수의 선두에는「$」를 붙입니다.

AI 번역본
-For Reference Only-

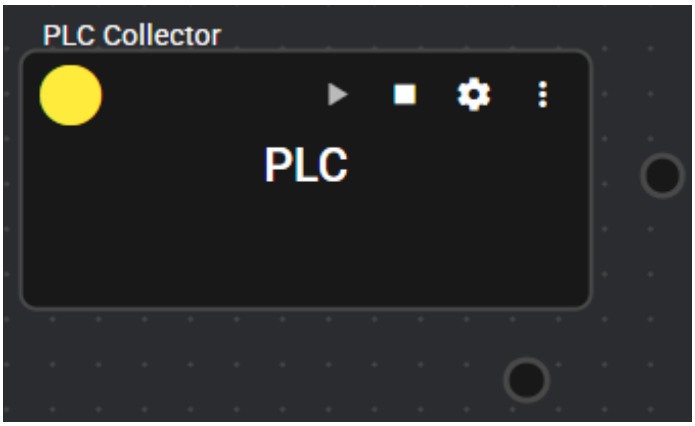
7 트러블슈트

본 장에서는, 트러블 발생 시의 대처 방법에 대해 설명합니다.

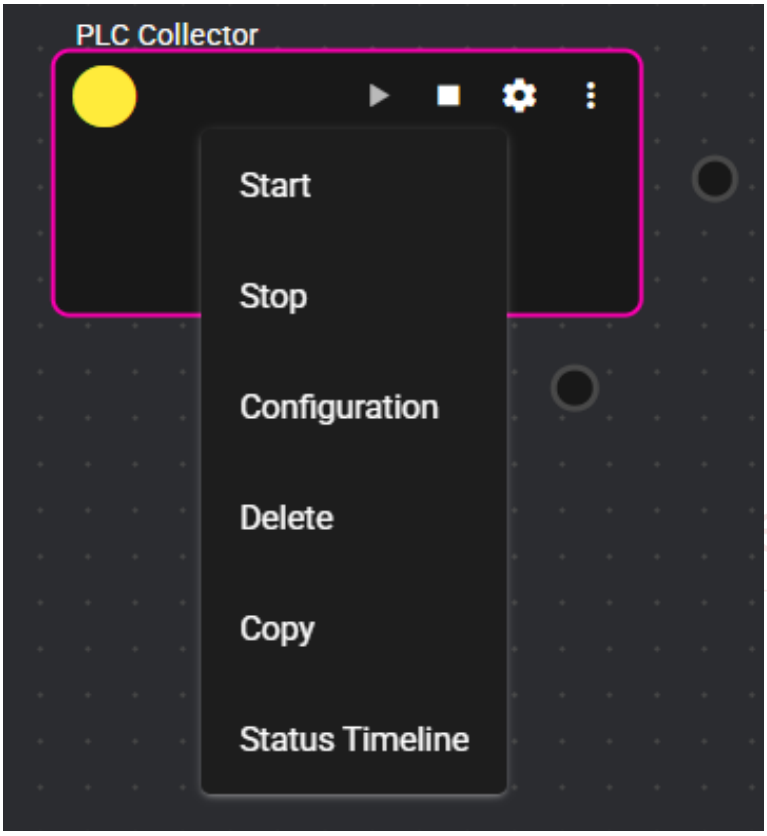
7.1 이상 발생 시 공통 순서

본 절에서는, SpeeDBee Synapse에서 문제가 발생했을 때의, 기본적인 대처 방법을 설명합니다.

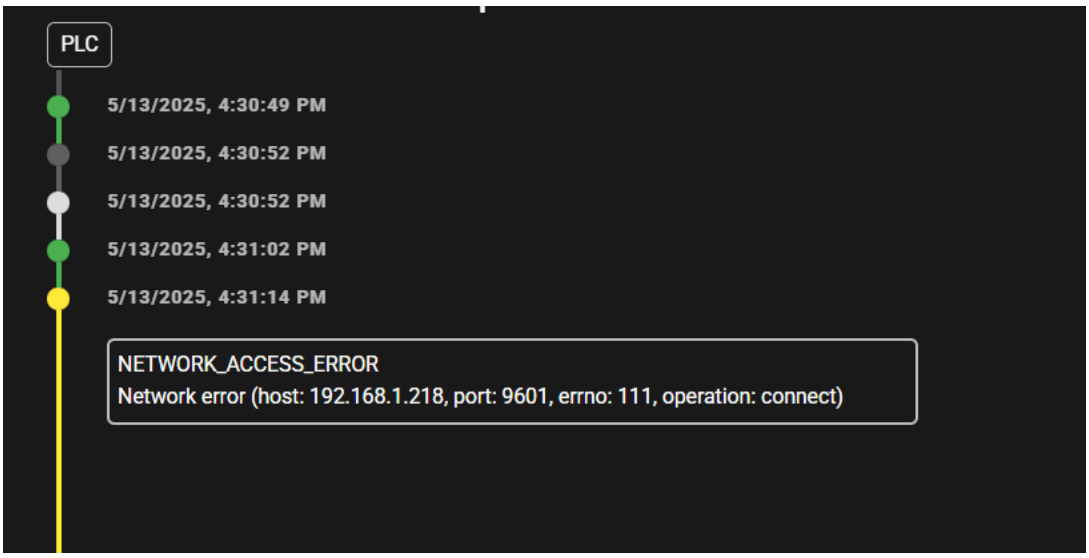
1. 스테이터스 램프가 적색・황색이 되어 있는 컴포넌트를 오른쪽 클릭합니다.



2. 「 스테이터스 타임라인 」을 선택합니다.



3. 타임라인에 표시되어 있는 에러 코드를 확인합니다.



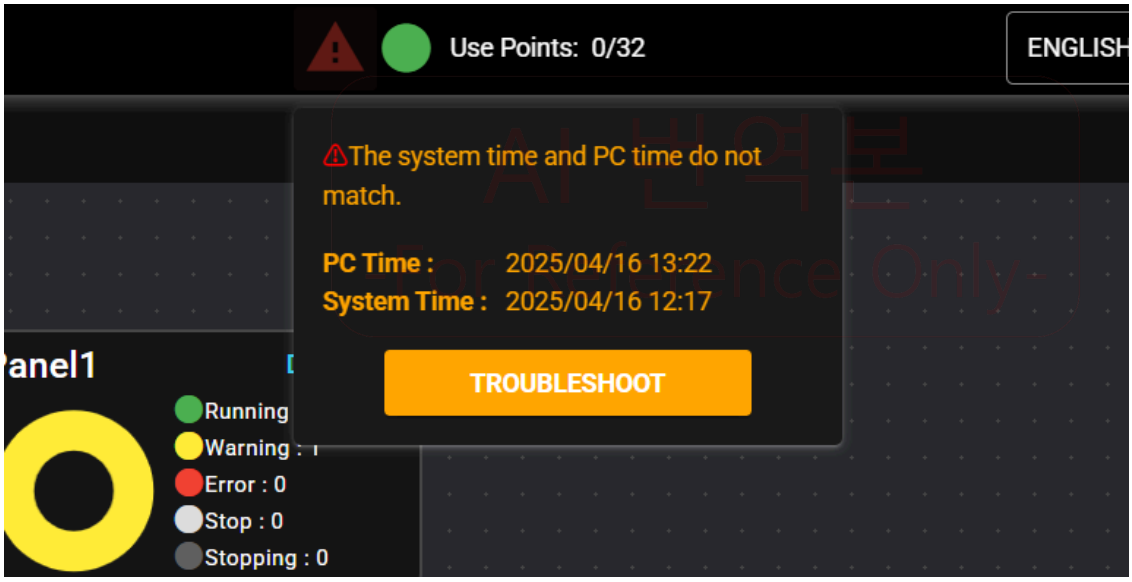
4. 본 매뉴얼의 우측 상단의 검색 상자에서, 타임라인에 표시되어 있던 에러 코드를 검색합니다.
5. 에러 코드의 개요를 파악하고, 처치 방법을 실시합니다.
6. 처치 방법을 실시해도 에러가 해소되지 않는 경우, 「 제조원에 문의 」를 참고로, 제조원에 문의해 주십시오.

7.2 자주 있는 트러블과 그 대처 방법

여기에서는, 자주 있는 트러블과 그 대처 방법에 대해 해설합니다.

7.2.1 시각 설정에 관한 문제와 대처 방법

SpeeDBee Synapse가 가동하고 있는 머신과 사용 브라우저의 시각이 크게 어긋나 있는 경우, 화면 상부에 다음 경고가 표시됩니다.



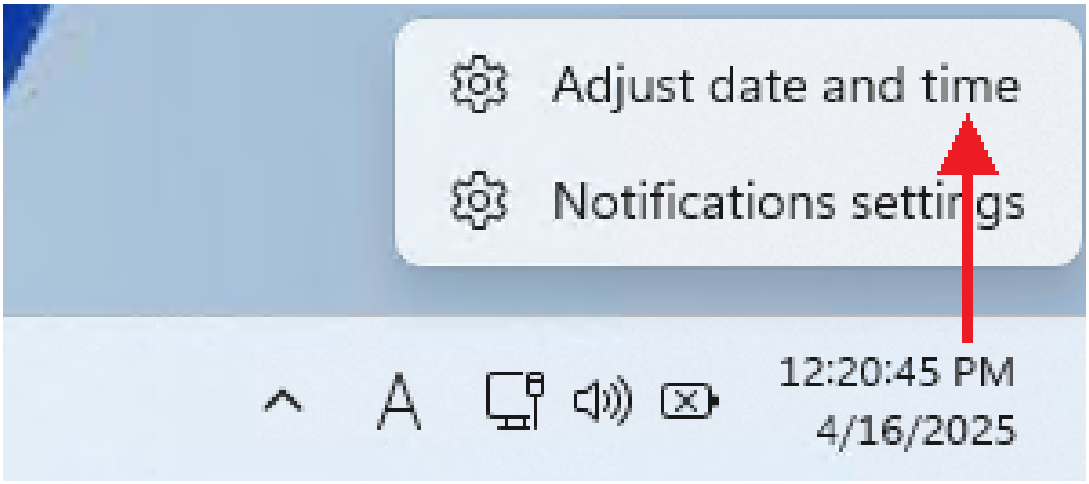
이 경고가 표시되어 있는 경우, SpeeDBee Synapse는, 동작 중에 에러가 발생합니다.
이를 회피하려면, 해당 환경에서 시스템 시각을 설정하는 방법이 있는 경우는, 설정 내용을 PC 시각의 설정과 맞춰 주십시오. 상세는 해당 환경의 매뉴얼을 확인하십시오.

7.2.1.1 SpeeDBee Synapse가 Windows에서 동작하고 있는 경우

Windows의 시각 설정은, 아래 순서로 설정할 수 있습니다.

1. SpeeDBee Synapse가 가동하고 있는 머신에서, Synapse를 정지합니다. 정지 방법에 대해서는, 여기를 참조하십시오.

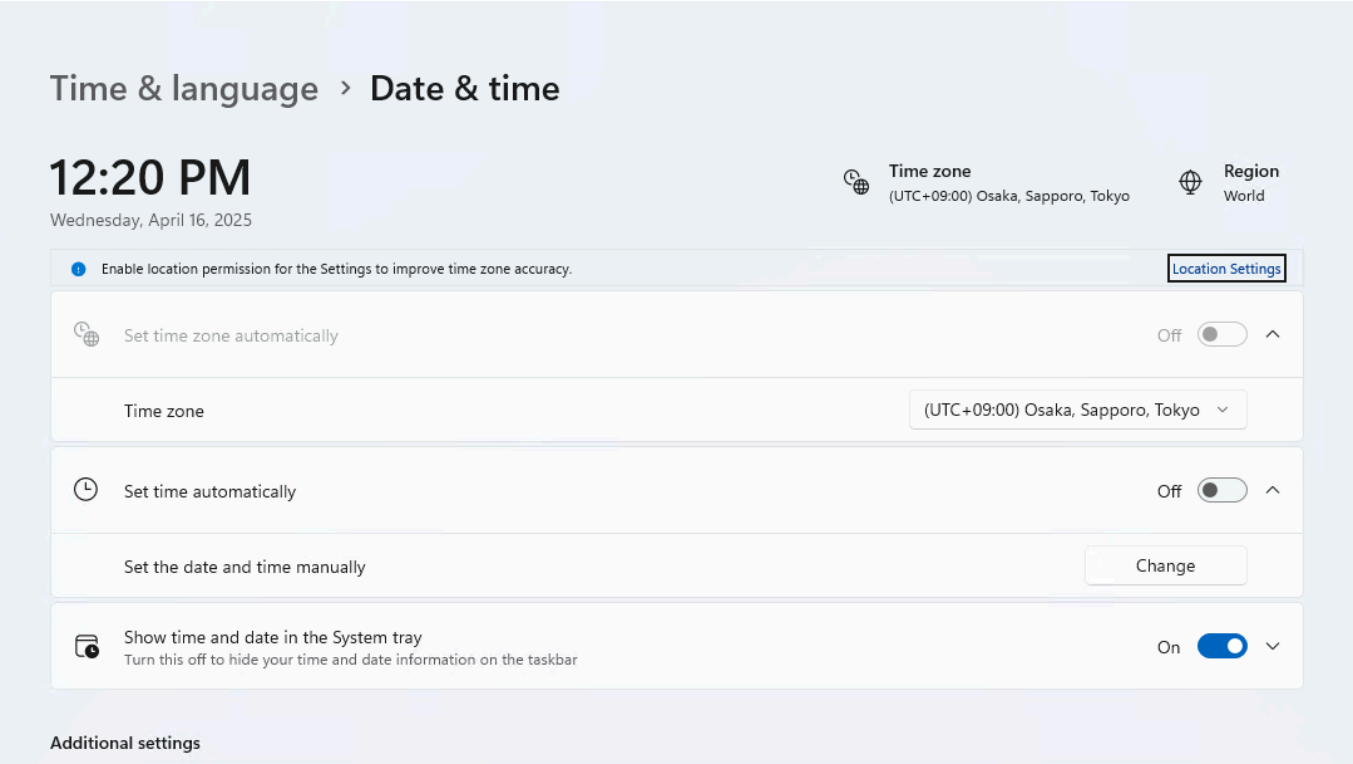
2. 태스크 바의 오른쪽 끝에 표시되어 있는 날짜와 시각을 오른쪽 클릭하고,「 일시를 조정한다 」를 클릭합니다.



PC가 인터넷에 접속하고 있는 경우는 다음 순서로 진행하고, 인터넷에 접속하고 있지 않은 경우는 순서 4부터 수행하십시오.

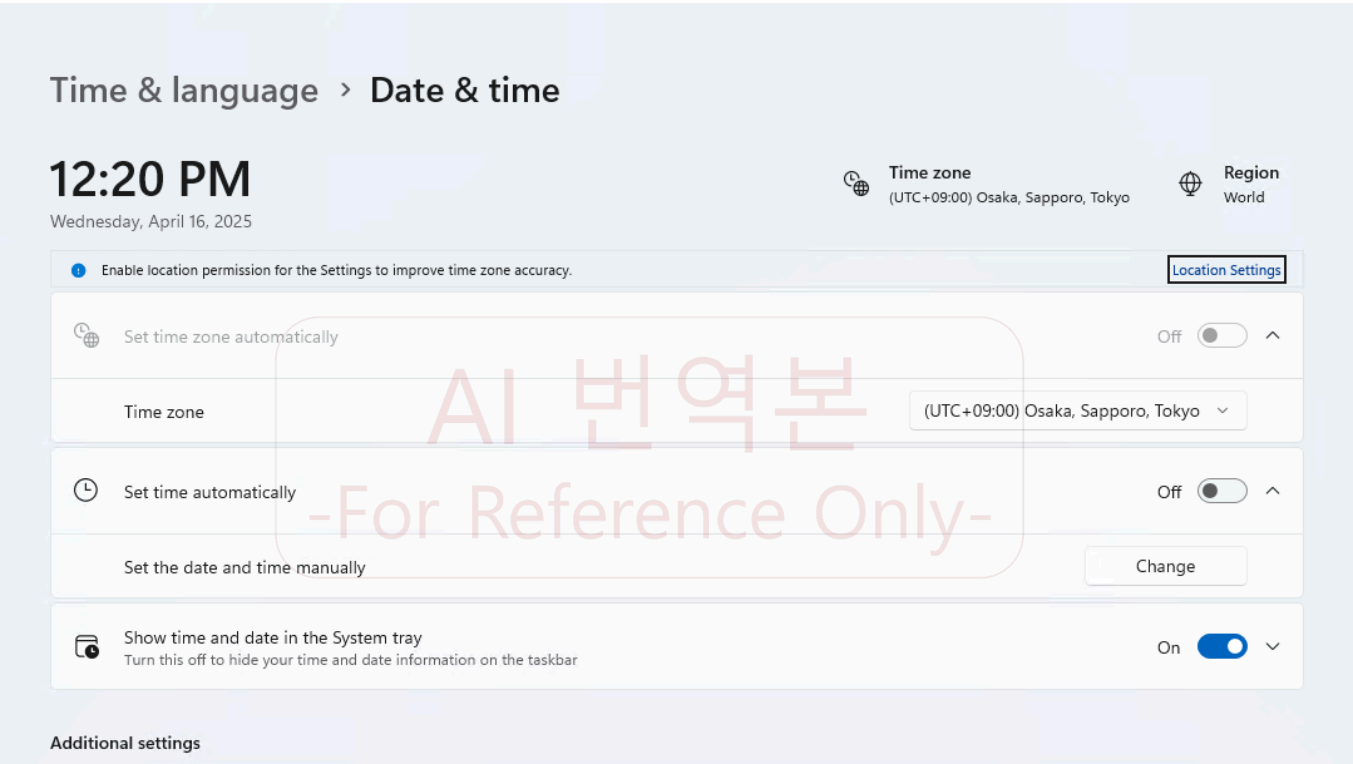
3.「 날짜와 시각 」 화면이 표시됩니다. PC가 인터넷에 접속하고 있는 경우,「 시각을 자동으로 설정한다 」를 온으로 하고 1분 기다립니다. 그 후 순서 6으로 진행하십시오.

AI 번역본
-For Reference Only-



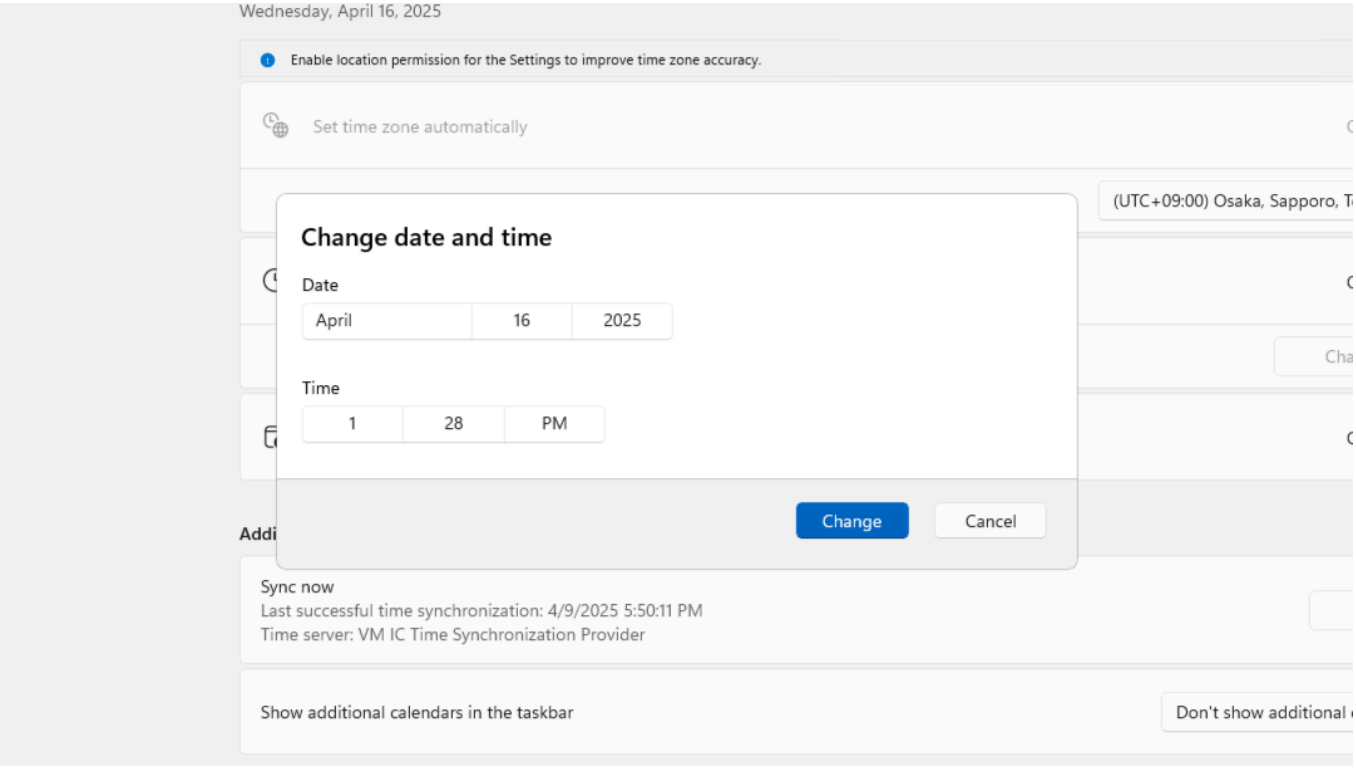
만약 시각의 자동 동기가 되지 않거나, 자동 동기를 수행해도 문제가 해소되지 않는 경우, PC가 인터넷에 접속되어 있지 않은 경우는, 수동으로 시각을 설정할 필요가 있습니다. 다음 순서로 진행하십시오.

4. 「 시각을 자동으로 설정한다 」를 오프로 하고 「 날짜와 시각을 수동으로 설정한다 」의 변경 버튼을 선택하면, 시각의 설정 메뉴가 열립니다.



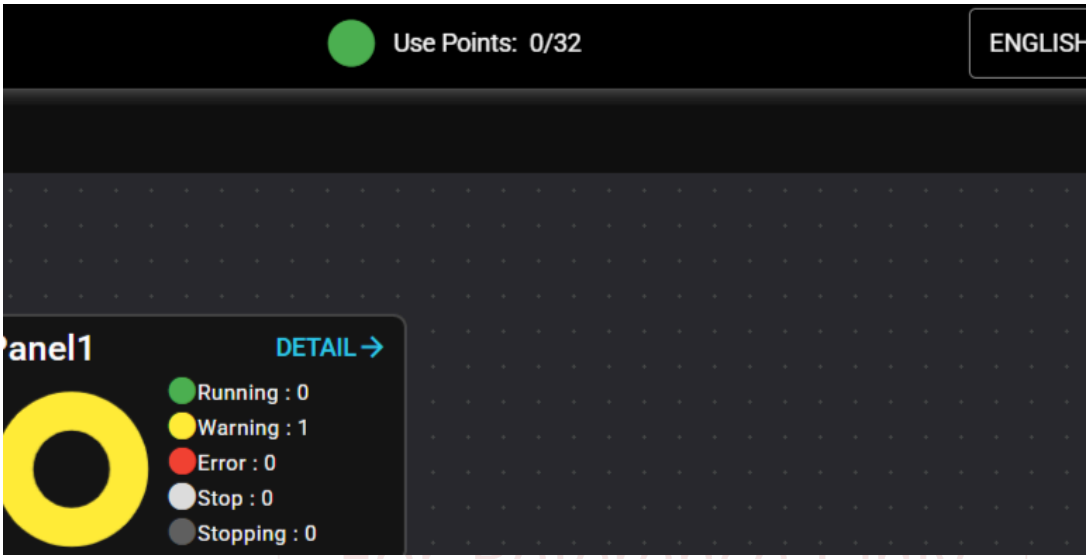
AI 번역본
-For Reference Only-

5. 메뉴에서 현재의 시각을 설정하고, 「 변경 」 버튼을 선택하여 1분 기다립니다.



6. SpeeDBee Synapse가 가동하고 있는 머신에서, Synapse를 기동합니다. 기동 방법에 대해서는, 여기를 참조하십시오.

7. 시각이 올바르게 설정되면, SpeeDBee Synapse의 화면 상부에서 경고가 표시되지 않게 됩니다.



-For Reference Only-

7.2.1.2 SpeeDBee Synapse가 Linux에서 동작하고 있는 경우

Linux의 시각 설정은, 아래 순서로 설정할 수 있습니다.

> Data Flow Controller판에서는, Data Flow Controller의 WebUI에서 설정할 수 있습니다.

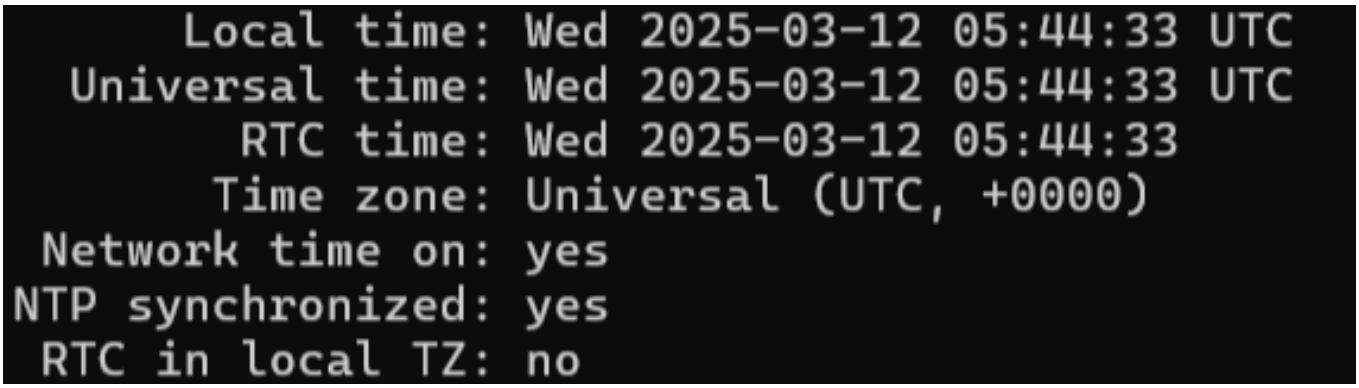
1. SpeeDBee Synapse가 가동하고 있는 머신에서, Synapse를 정지합니다. 정지 방법에 대해서는, 여기를 참조하십시오.

2. Linux의 현재의 시각을 확인합니다. 아래는 현재 시스템에 설정되어 있는 시각을 확인하는 커맨드입니다.

   커맨드

   ```
   $ timedatectl
   ```

   시각이 표시된 결과 예는 이하와 같습니다.「Local time」이 현재 설정되어 있는 시스템 시각이 됩니다. 시각이 어긋나 있는 경우는, 설정을 변경할 필요가 있습니다.

3. 필요에 따라, 타임존을 올바른 지역으로 변경합니다. 이 설정은 임의가 되므로, 타임존이 일치하고 있는지, 설정이 불필요한 경우는 이 순서를 수행하지 말고, 다음 순서로 진행하십시오.
   예를 들어 SpeeDBee Synapse가 가동하고 있는 머신을 JST로 변경하는 경우, 아래 커맨드를 사용합니다.

   커맨드

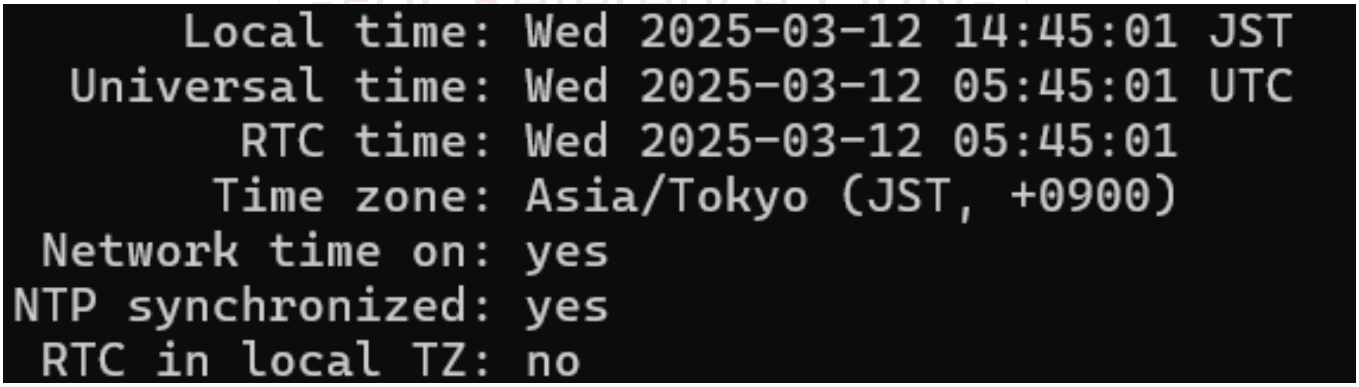
   ```
   $ sudo timedatectl set-timezone Asia/Tokyo
   ```

   설정하고 싶은 타임존이 JST 이외인 경우, 아래 커맨드로 사용할 수 있는 타임존을 확인할 수 있습니다.

   커맨드

   ```
   $ timedatectl list-timezones
   ```

   타임존이 변경된 결과 예는 이하와 같습니다. 순서 1의 커맨드로 설정값을 확인할 수 있습니다.

PC가 인터넷에 접속하고 있는 경우는 다음 순서로 진행하고, 인터넷에 접속하고 있지 않은 경우는 순서 5로진행하십시오.

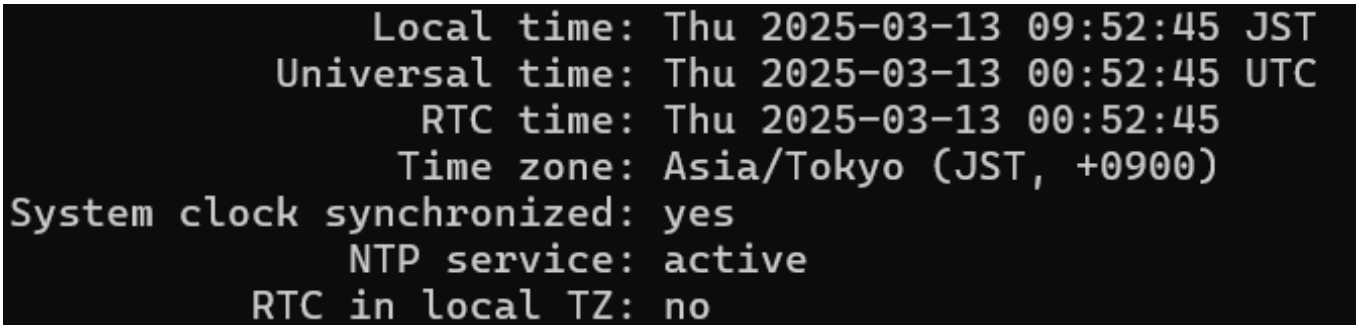
4. PC가 인터넷에 접속하고 있는 경우, NTP Server를 유효하게 합니다. 아래 커맨드를 사용하여 설정 후, 1분 기다리고 순서 7로 진행하십시오.

   커맨드

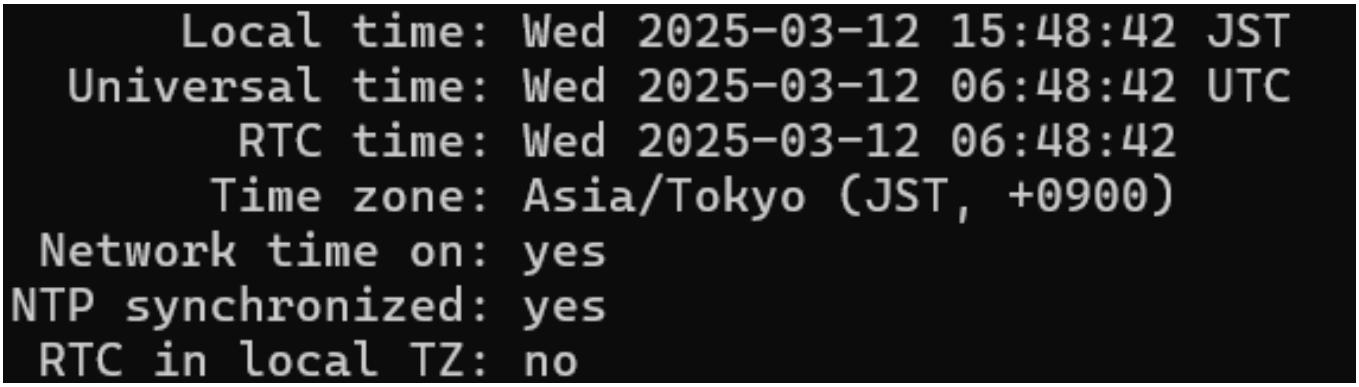
```
$  sudo timedatectl set-ntp yes
```

   NTP Server가 유효화된 결과 예는 이하와 같습니다. 순서 1의 커맨드로 설정값을 확인할 수 있습니다. OS가

   BlackBear 이외인 경우



   OS가 BlackBear인 경우



   만약 NTP server의 동기를 수행해도 문제가 해소되지 않는 경우나, PC가 인터넷에 접속되어 있지 않은 경우는, 수동으로 시각을 설정할 필요가 있습니다. 다음 순서로 진행하십시오.

5. 시각을 수동으로 설정하기 위해, NTP Server를 무효로 합니다. 아래 커맨드를 사용하여 설정할 수 있습니다.

   커맨드

```
$ sudo timedatectl set-ntp no
```

   NTP Server가 무효화된 결과 예는 이하와 같습니다. 순서 1의 커맨드로 설정값을 확인할 수 있습니다. OS가

   BlackBear 이외인 경우

AI 번역본
-For Reference Only-

```
               Local time: Wed 2025-03-12 16:03:22 JST
           Universal time: Wed 2025-03-12 07:03:22 UTC
                 RTC time: Thu 2025-03-13 00:57:15
                Time zone: Asia/Tokyo (JST, +0900)
System clock synchronized: no
              NTP service: inactive
          RTC in local TZ: no
```

OS가 BlackBear인 경우

```
      Local time: Wed 2025-03-12 14:42:04 JST
  Universal time: Wed 2025-03-12 05:42:04 UTC
        RTC time: Wed 2025-03-12 06:48:09
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: no
NTP synchronized: no
 RTC in local TZ: no
```

6. 다음으로 날짜와 시각을 변경합니다. 아래는 2025년 3월 12일 16시 03분 20초로 설정하는 경우의 예입니다. 실제로 사용할 때는, 날짜와 시각을 설정하고 싶은 값으로 고쳐서 이용합니다. 설정 후 1분 기다립니다.

   커맨드

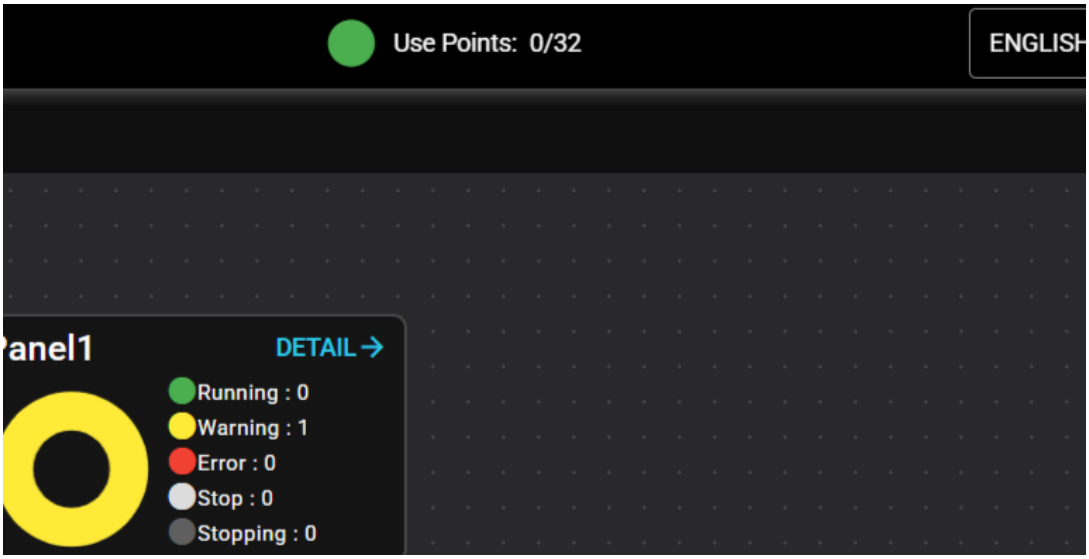
```
$ sudo date -s "2025-3-12 16:03:20"
```

   날짜와 시각이 변경된 결과 예는 이하와 같습니다. 순서 1의 커맨드로 설정값을 확인할 수 있습니다.

```
      Local time: Wed 2025-03-12 16:03:30 JST
  Universal time: Wed 2025-03-12 07:03:30 UTC
        RTC time: Wed 2025-03-12 07:03:30
       Time zone: Asia/Tokyo (JST, +0900)
 Network time on: no
NTP synchronized: no
 RTC in local TZ: no
```

7. SpeeDBee Synapse가 가동하고 있는 머신에서, Synapse를 기동합니다. 기동 방법에 대해서는, 여기를 참조하십시오.

8. 시각이 올바르게 설정되면, SpeeDBee Synapse의 화면 상부에서 경고가 표시되지 않게 됩니다.



7.2.1.3 시각 설정에 관한 주의사항

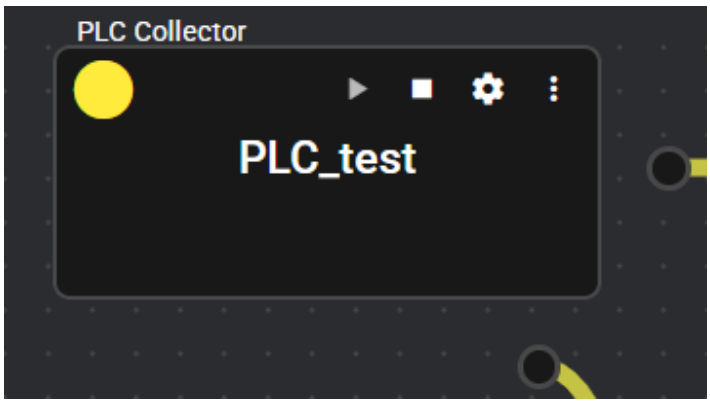
본 항에서 나타낸 대처에 따라 조작함으로써, 시각 설정의 트러블은 회피할 수 있습니다. 그러나 인터넷에 접속되어 있지 않은 머신이나 RTC（Real Time Clock）가 탑재되어 있지 않은 머신에서는 재기동 후에 같은 설정을 수행할 필요가 있습니다. 그 때문에, 다음 중 하나의 검토를 권장합니다.

- 인터넷에 접속하여 시각 동기를 올바르게 수행한다（보안상 문제가 없는 경우에 한한다）
- 가동 머신의 LAN 내에 NTP Server를 기동하여 자동 동기한다
- RTC 모듈에 의해 시각 어긋남을 방지한다

7.2.2 PLC 컬렉터에 접속할 수 없는 경우의 대처 방법

PLC 컬렉터에 접속할 수 없는 경우, 컴포넌트에 황색 스테이터스 램프가 표시되고, 스테이터스 타임라인에 NETWORK_TIMEOUT이 표시됩니다.
스테이터스 확인 방법은, 여기를 참조하십시오.

AI 번역본
-For Reference Only-





7.2.2.1 IP 주소의 확인

사용하고 있는 PLC의 IP 주소를 확인합니다.

> 확인할 수 있는 기기에 대해
>
> 이 장의 내용은, SpeeDBee Synapse가 가동하고 있는 디바이스에서 수행할 필요는 없습니다. 사용하시는 SpeeDBee Synapse의 WebUI를 열람하고 있는 PC에서도 실행 가능합니다.
> 그러나, SpeeDBee Synapse가 가동하고 있는 기기와 같은 네트워크 내의 PC에서 실시하는 편이 확실해집니다.
> 또한, Data Flow Controller판에서는 Data Flow Controller의 WebUI에서 설정할 수 있습니다.

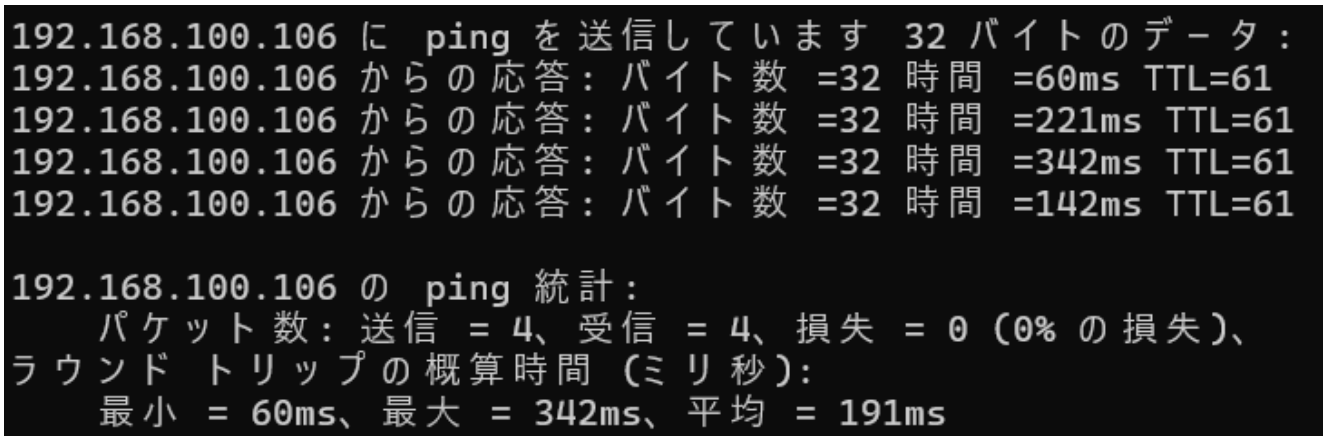
1. 네트워크에 접속하고 있는 PC에서 Windows는 PowerShell(커맨드 프롬프트도 가능)을 열고, Linux는 터미널상에서 다음 커맨드를 실행합니다.

AI 번역본
-For Reference Only-

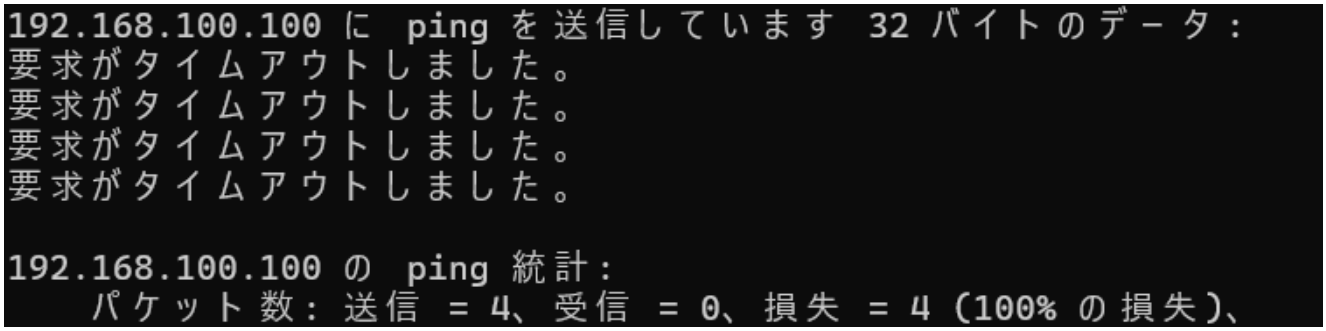
Windows의 경우

```
PS >ping 대상 PLC의 IP 주소
```

- 성공 화면



- 실패 화면

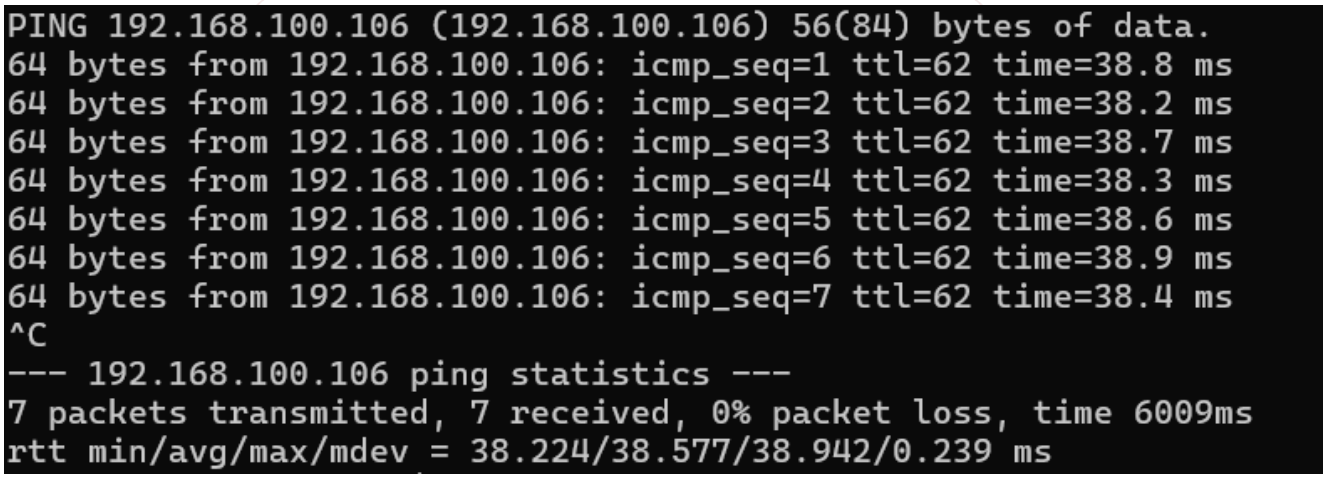


Linux의 경우

```
$ ping 대상 PLC의 IP 주소
```

※Linux는 [ctrl]+[C]를 입력할 때까지, ping을 계속 보냅니다.

- 성공 화면



- 실패 화면

```
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.

--- 192.168.100.100 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3006ms
```

ping의 소통이 확인되지 않은 경우는, 이용하시는 PLC측의 설정을 확인하십시오.

7.2.2.2 포트의 확인

사용하고 있는 PLC의 포트 번호를 확인합니다.

> 확인할 수 있는 기기에 대해
>
> 이 장의 내용은, SpeeDBee Synapse가 가동하고 있는 디바이스에서 수행할 필요는 없습니다. 사용하시는 SpeeDBee Synapse의 WebUI를 열람하고 있는 PC에서도 실행 가능합니다.
> 그러나, SpeeDBee Synapse가 가동하고 있는 기기와 같은 네트워크 내의 PC에서 실시하는 편이 확실해집니다.

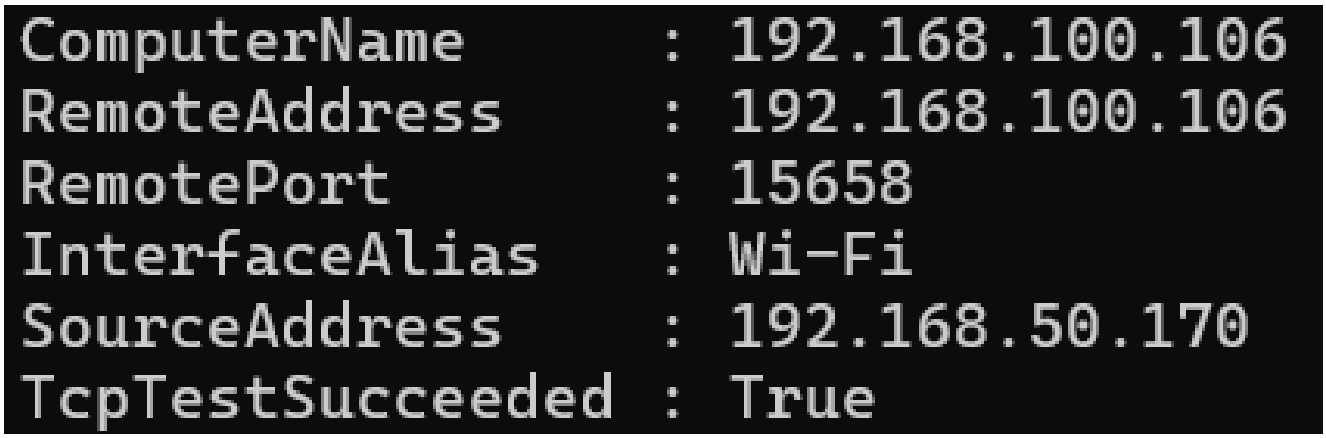
1. 네트워크에 접속하고 있는 PC에서 Windows는 PowerShell(커맨드 프롬프트도 가능)을 열고, Linux는 터미널상에서 다음 커맨드를 실행합니다.

AI 번역본
-For Reference Only-

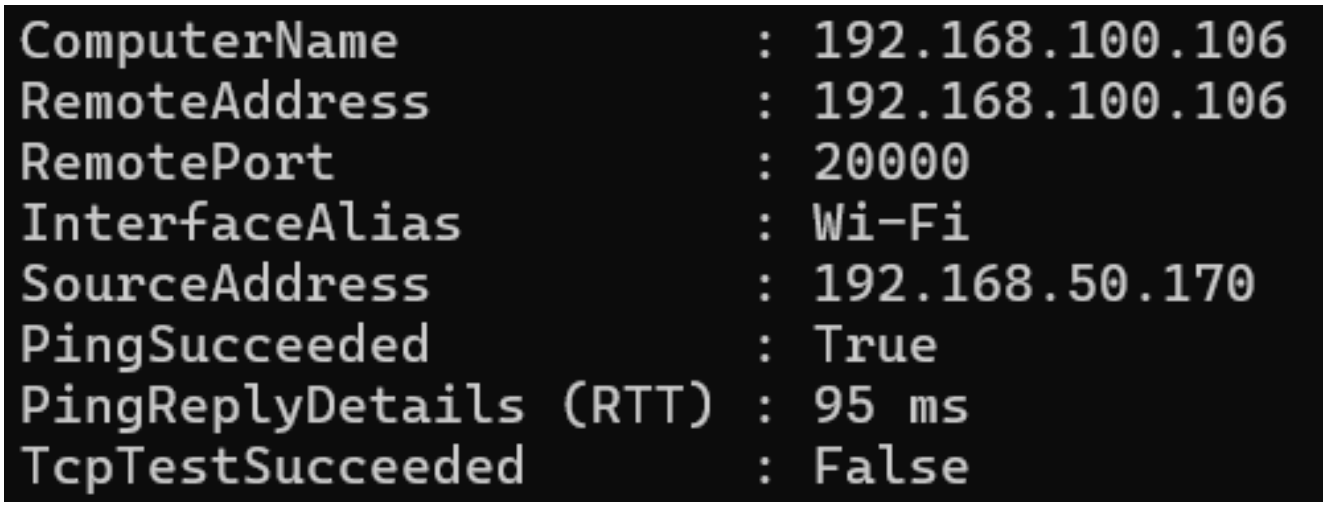
Windows의 경우

```
PS >Test-NetConnection 대상 PLC의 IP 주소-Port 대상 PLC의 포트 번호
```

- 해당 포트가 열려 있을 때의 화면



- 해당 포트가 열려 있지 않을 때의 화면



Linux의 경우

```
$ nc -zv 대상 PLC의 IP 주소대상 PLC의 포트 번호
```

- 해당 포트가 열려 있을 때의 화면



- 해당 포트가 열려 있지 않을 때의 화면



포트가 열려 있는 것을 확인할 수 없었던 경우는, 이용하시는 PLC측의 설정을 확인하십시오.

7.2.2.3 PLC측의 설정

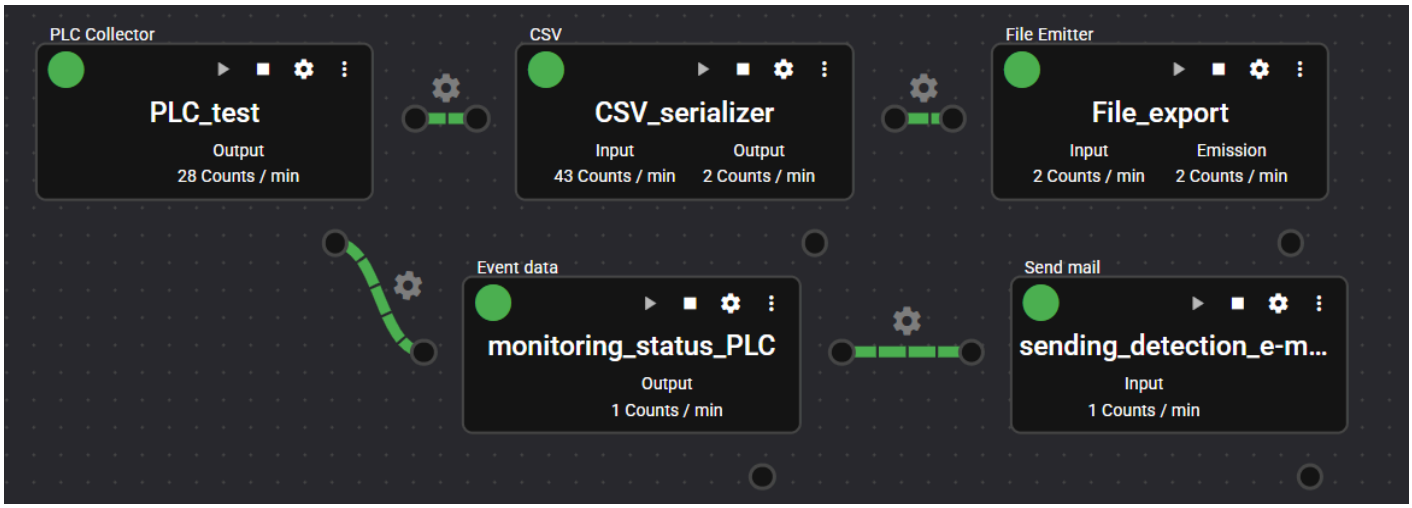
PLC의 IP 주소, 포트의 설정 방법에 대해서는, 이용하시는 PLC의 취급 설명서 등을 확인하십시오.

7.3 이상이 발생한 경우에 검지하는 방법

여기에서는, 컴포넌트에서 이상이 발생한 경우에 검지를 수행하는 방법을 해설합니다.

7.3.1 특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법

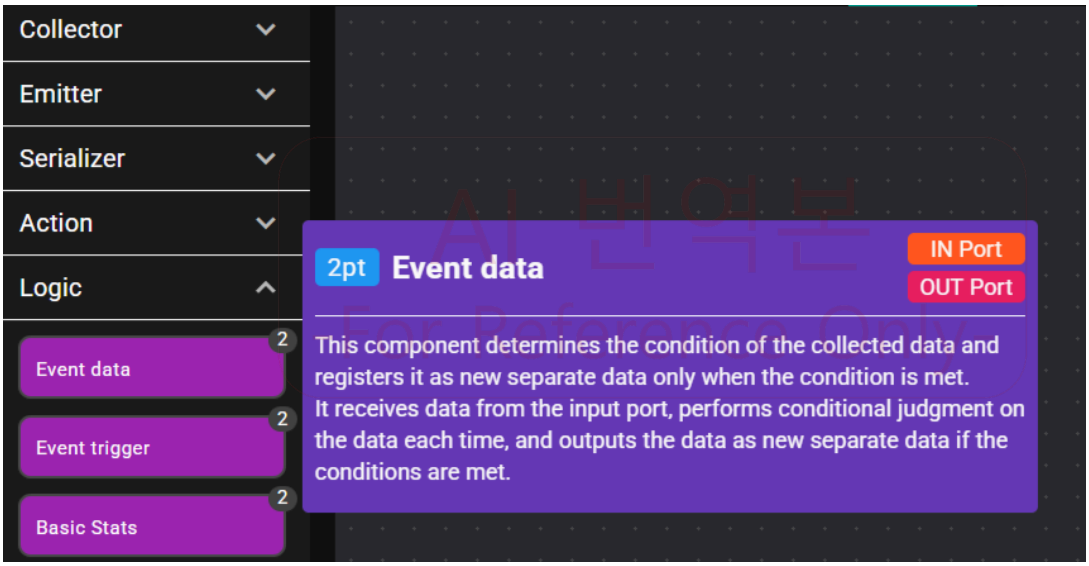
특정 컴포넌트에서 이상이 발생했을 때에, 메일을 송신하여 알리는 방법을 해설합니다.



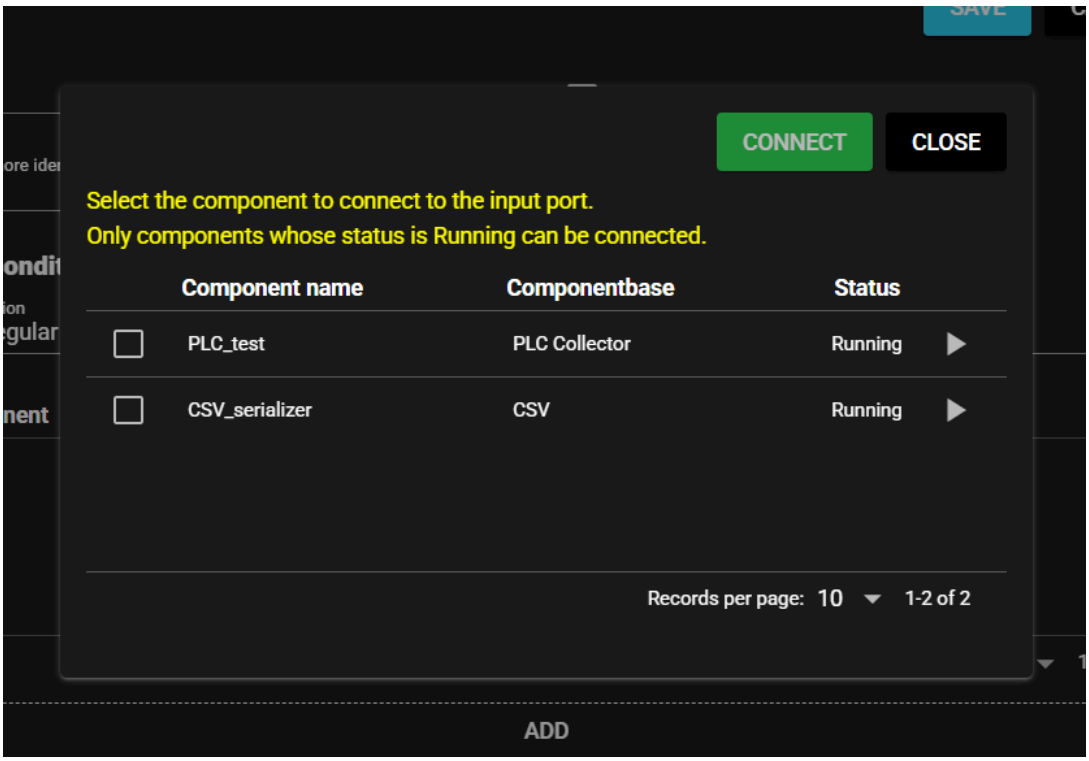
7.3.1.1 이벤트 데이터의 설정

검지 대상 컴포넌트에서 출력되는 errors_run_status의 값이, 정상이나 정지 이외의 값일 때, 이벤트 데이터가 등록되도록 설정합니다. 컴포넌트 스테이터스가 경고, 에러인 경우에 메일 통지를 수행합니다.

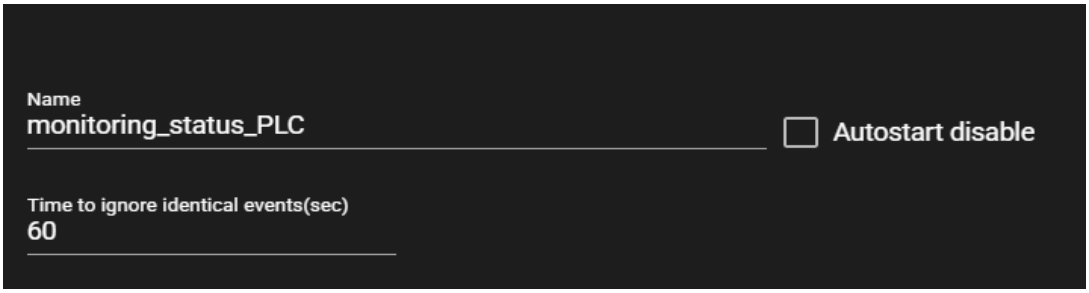
1. 에러를 검지하고 싶은 컴포넌트가 가동하고 있는 패널을 열고, Edit Mode에서 ComponentBase -> Logic -> 이벤트 데이터를 패널상으로 드래그 앤 드롭합니다.



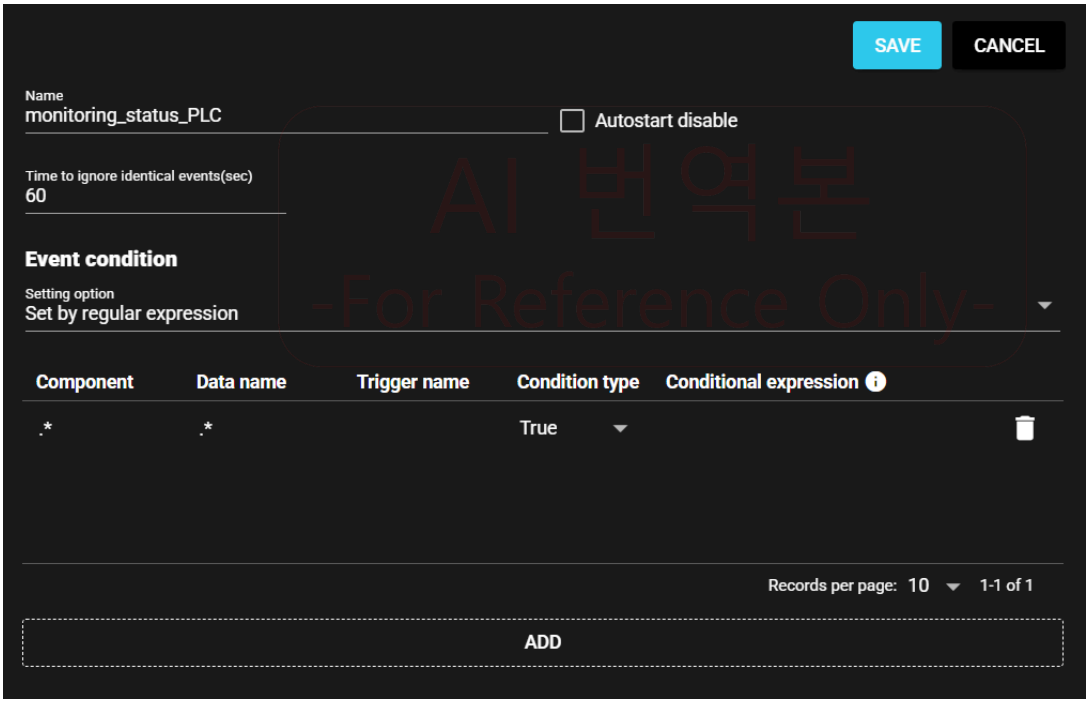
2. 입력 포트의 접속처 메뉴가 열립니다. 이번에는 시스템 포트에 접속하기 위해,「닫기」를 선택합니다.



3. 임의의 명칭을 입력합니다. 이번에는 예로서 「 PLC 상태 감시 」라는 이름으로 합니다.

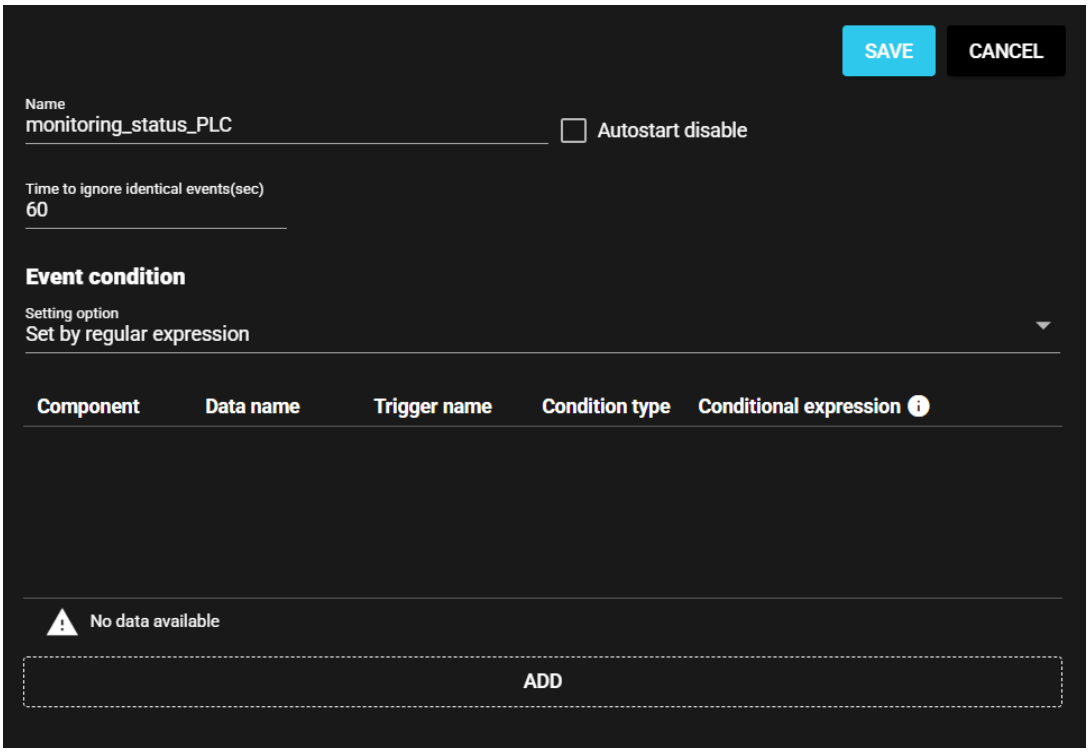


4. 휴지통 마크를 선택하여, 데이터를 삭제합니다.

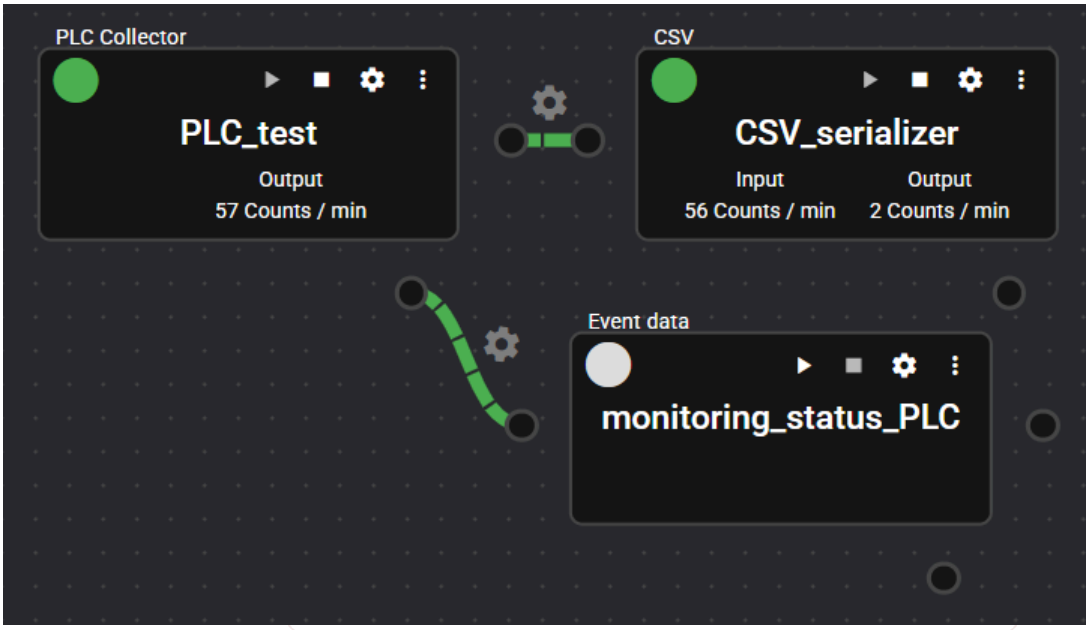


AI 번역본
-For Reference Only-

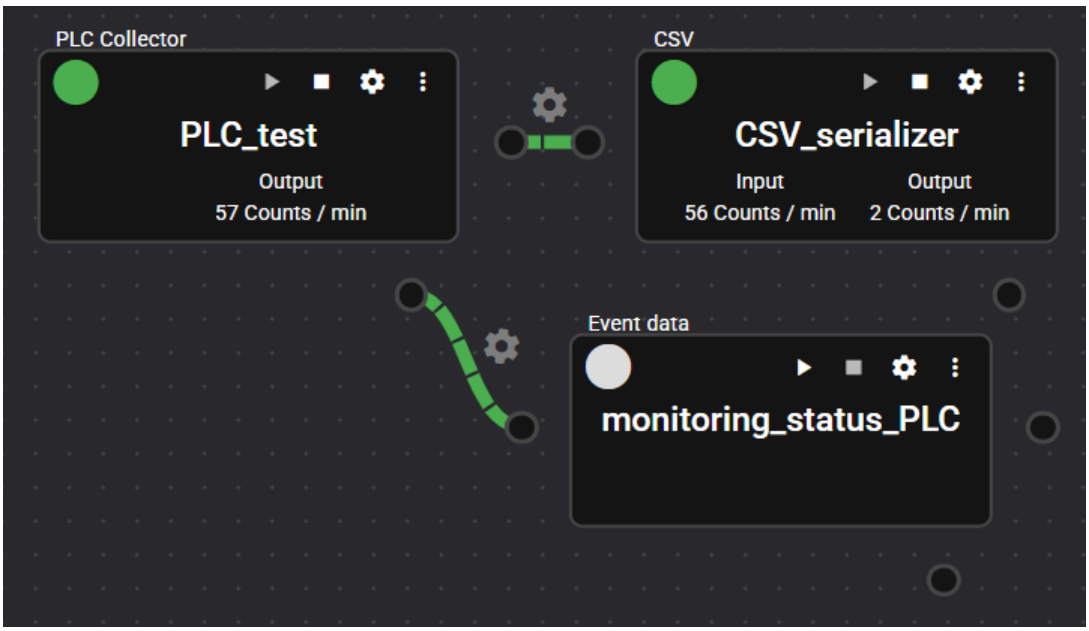
5. 「 저장 」을 선택합니다.



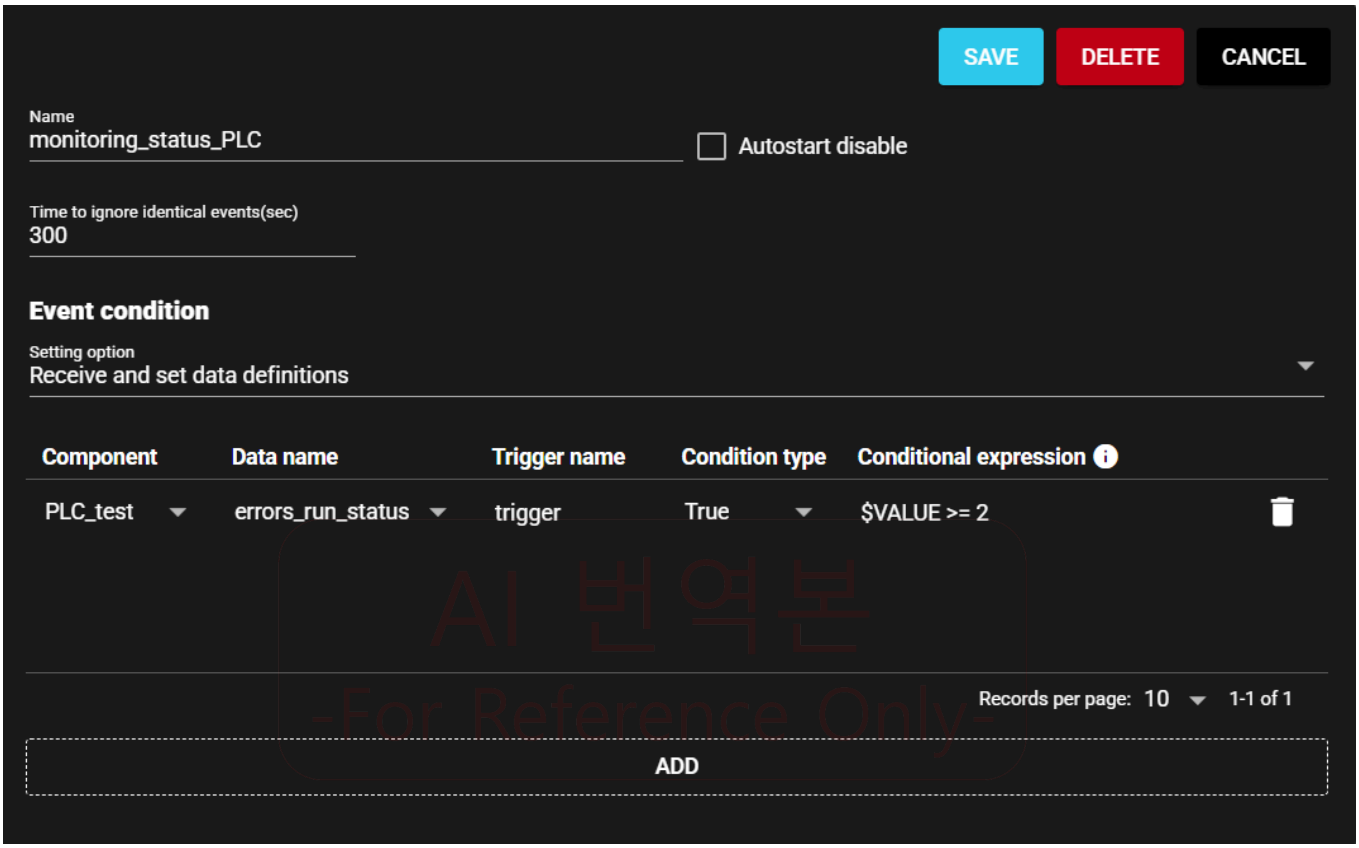
6. 이벤트 데이터를 검지하고 싶은 컴포넌트의 시스템 포트에 접속합니다. 이번에는 예로서 PLC 컬렉터에 이벤트 데이터를 접속합니다.



7. 이벤트 데이터의 설정 메뉴를 엽니다.



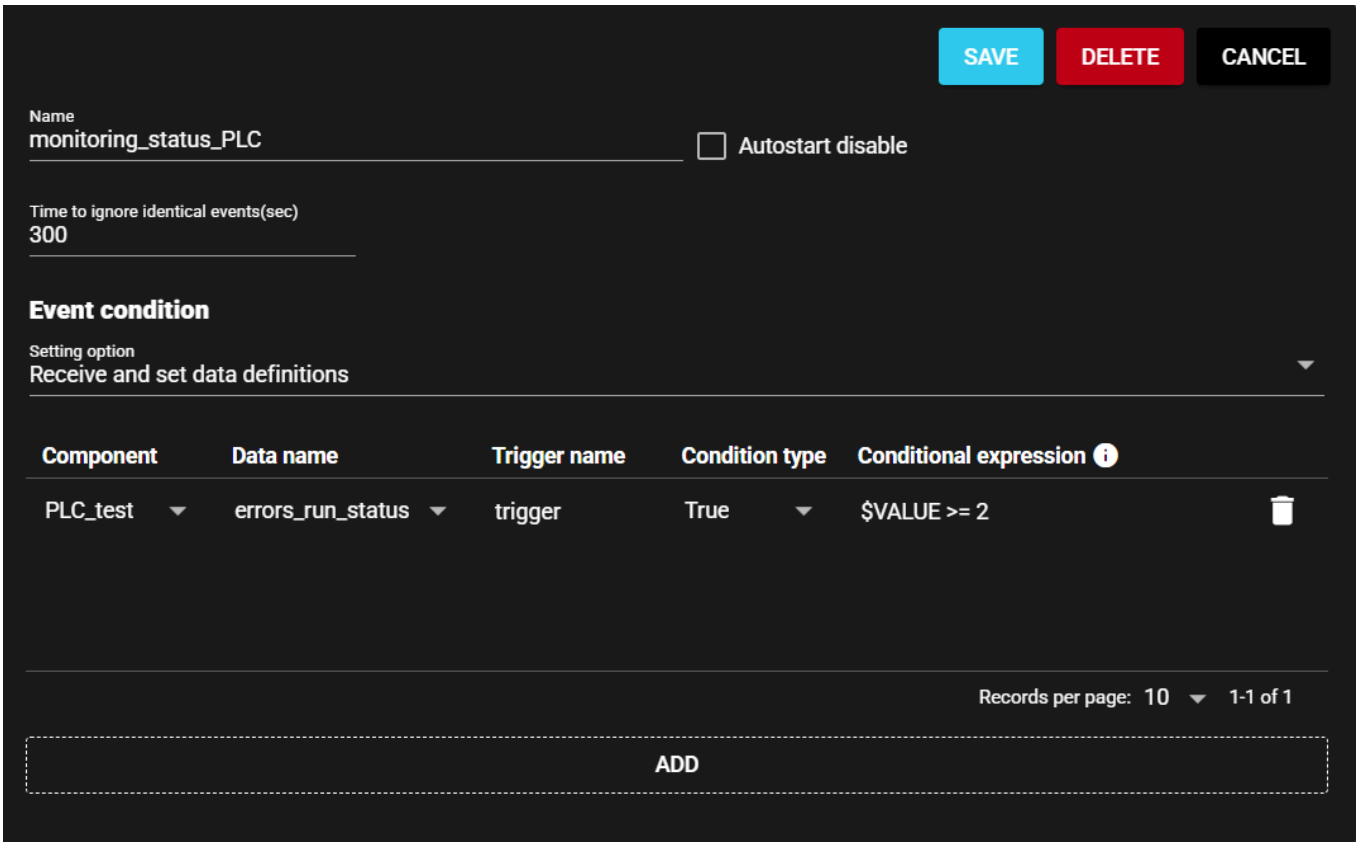
8. 이벤트 조건을 설정합니다. 설정 방법에서 「 데이터 정의를 수신하여 설정 」을 선택합니다. 데이터명을 「 errors_run_status 」, 트리거명을 「 trigger 」, 조건 종별을 「 True 」, 조건식을 「 $VALUE >= 2 」로 설정합니다. 그리고, 동일 이벤트를 무시하는 시간(초)을 「 300 」으로 설정합니다.



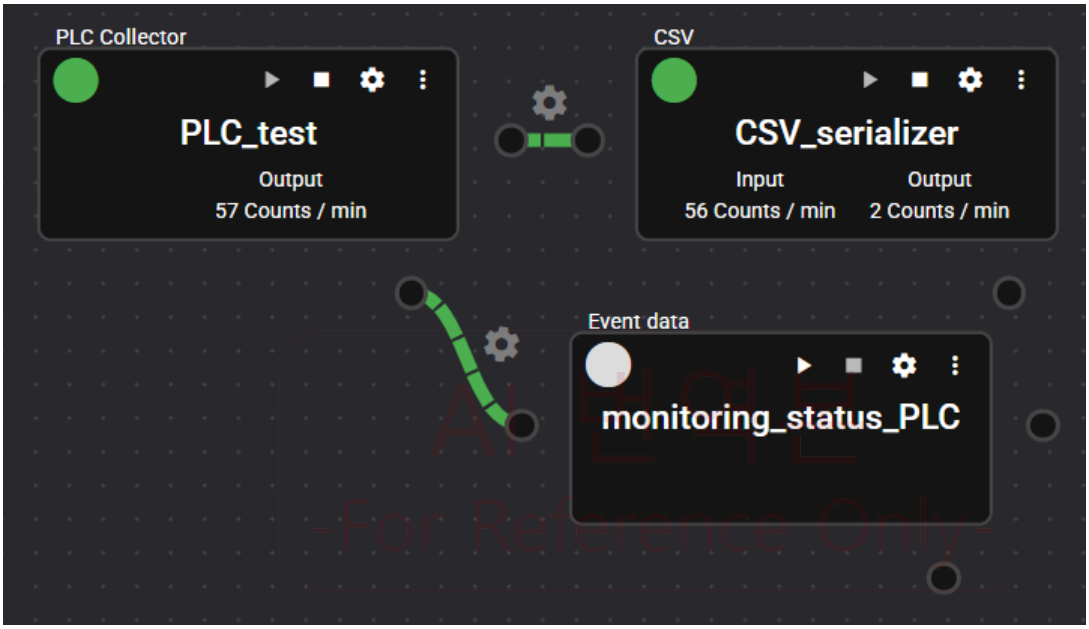
AI 번역본
-For Reference Only-

> 메일 송신에 관한 주의
>
> 이 설정에서는, 5분（300초）경과해도, 이상이 계속되는 한 메일이 항상 송신됩니다. 통지량이 방대해지는 경우가 있으므로 주의하십시오. 이상이 발생한 타이밍에 한 번만 메일 송신하는 경우는, 조건 종별을 「 Raise 」으로 설정합니다.

9. 「 저장 」을 선택합니다.



10. 이벤트 데이터를 기동합니다.



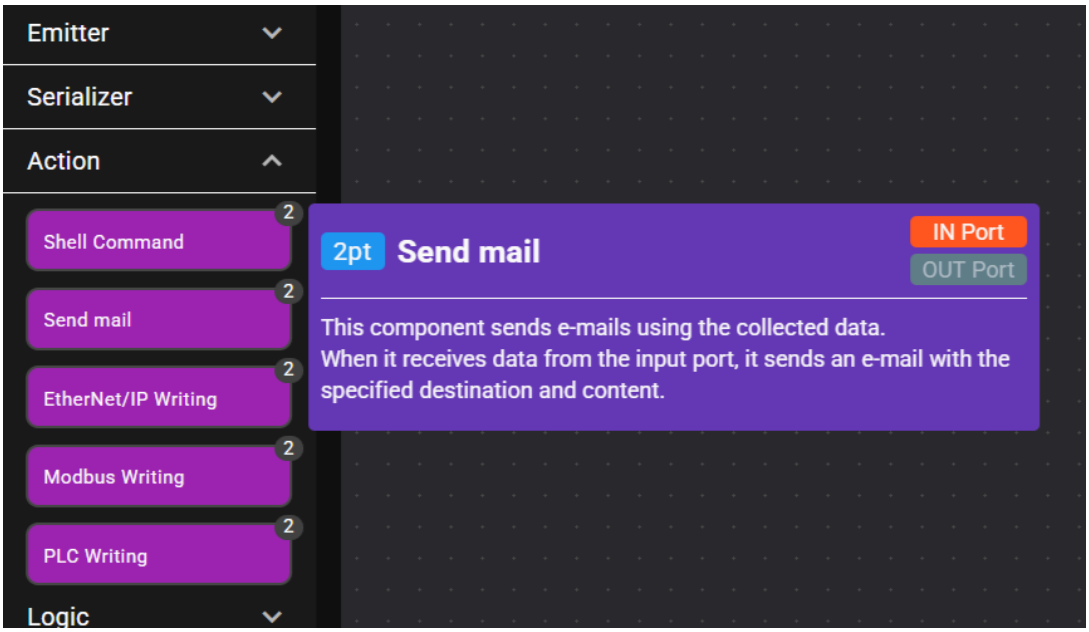
이것으로 이벤트 데이터의 설정이 완료되었습니다.

7.3.1.2 메일 송신의 설정

이벤트 데이터로부터 받은 값을, 메일로 송신하도록 설정합니다.

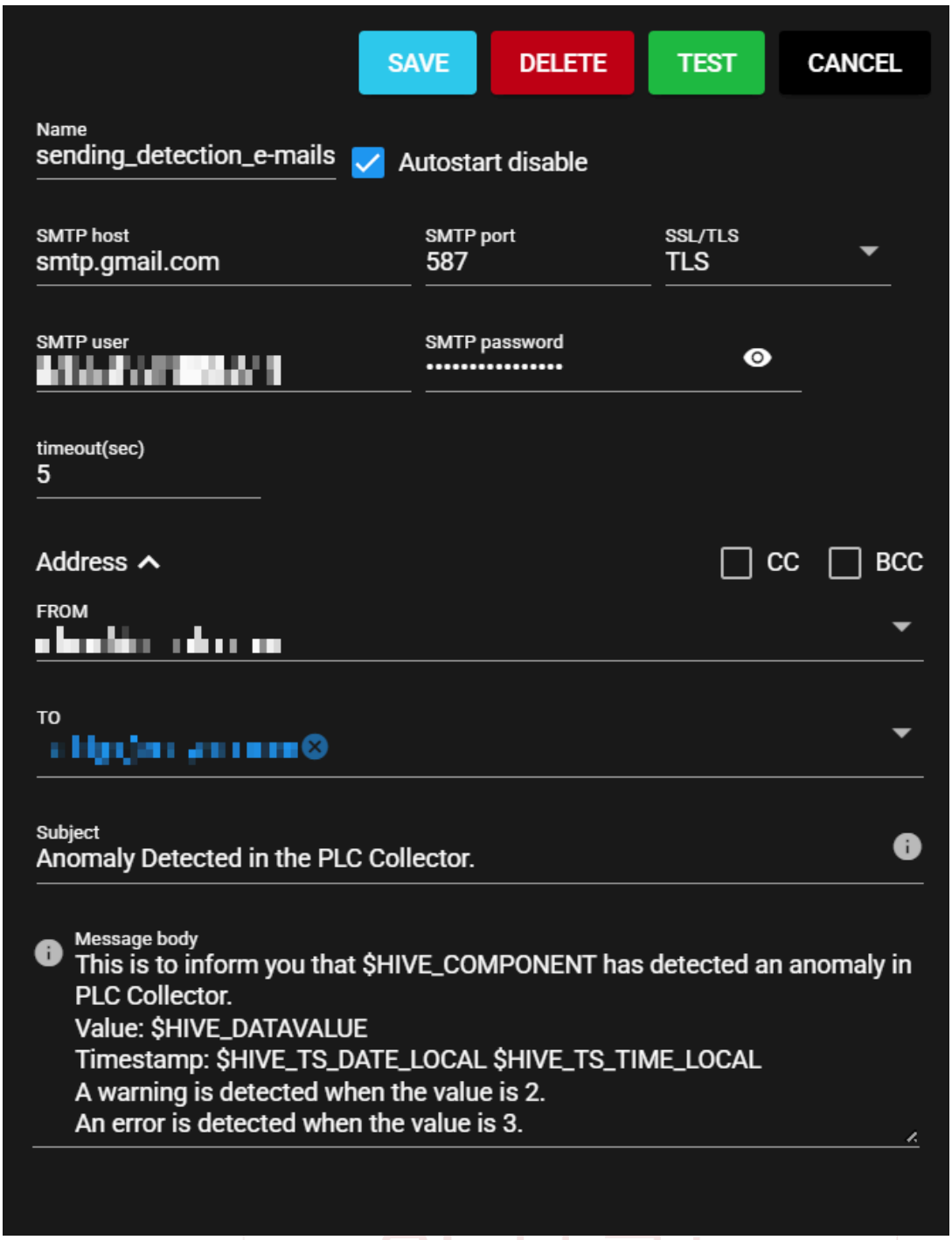
이번에는 gmail로 송신하는 경우의 설정을 해설합니다. 사용할 수 있는 gmail 계정을 준비하고, 2단계 인증을 설정하여 앱 패스워드를 작성해 두십시오.

1. 에러를 검지하고 싶은 컴포넌트가 가동하고 있는 패널을 열고, Edit Mode에서 ComponentBase -> Action -> 메일 송신을 패널상으로 드래그 앤 드롭합니다.

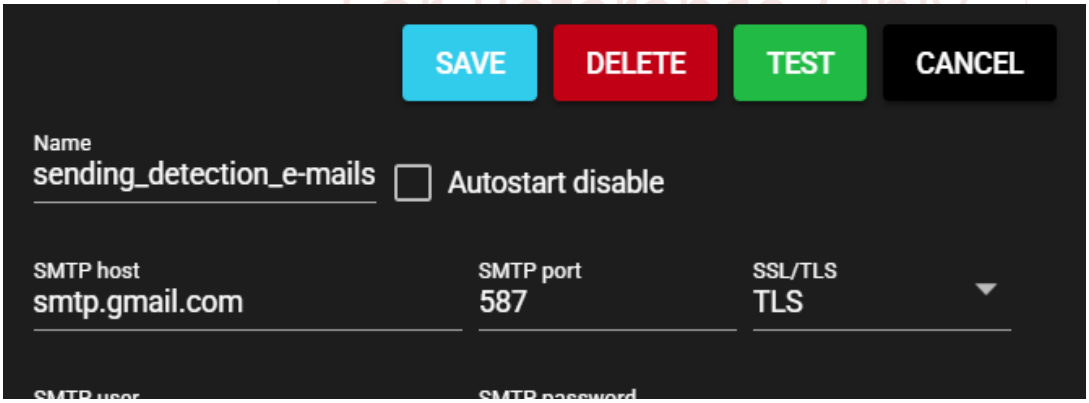


2. 필요한 설정을 수행합니다. gmail의 경우는 SMTP 서버가「 smtp.gmail.com 」, 포트 (TLS)가「 587 」이 됩니다. SMTP 패스워드에는, 작성한 앱 패스워드를 입력하십시오.

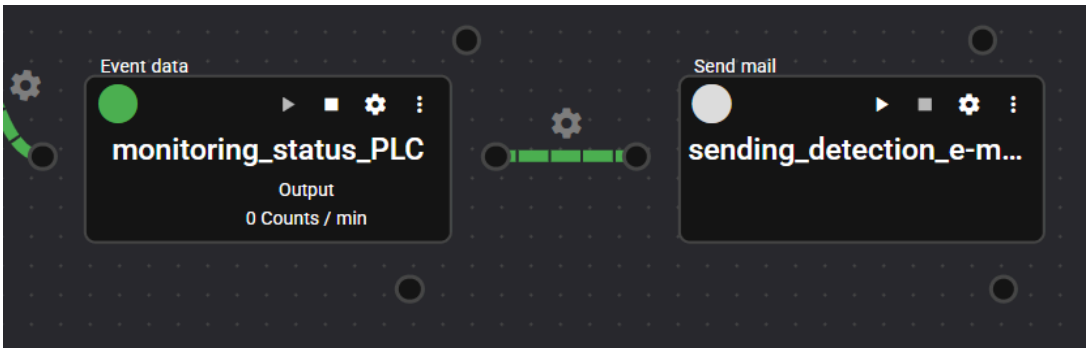
AI 번역본
-For Reference Only-


SAVE
DELETE
TEST
CANCEL
Name
sending_detection_e-mails
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS
SMTP user
SMTP password
timeout(sec)
5
Address
CC
BCC
FROM
TO
Subject
Anomaly Detected in the PLC Collector.
Message body
This is to inform you that $HIVE_COMPONENT has detected an anomaly in PLC Collector.
Value: $HIVE_DATAVALUE
Timestamp: $HIVE_TS_DATE_LOCAL $HIVE_TS_TIME_LOCAL
A warning is detected when the value is 2.
An error is detected when the value is 3.

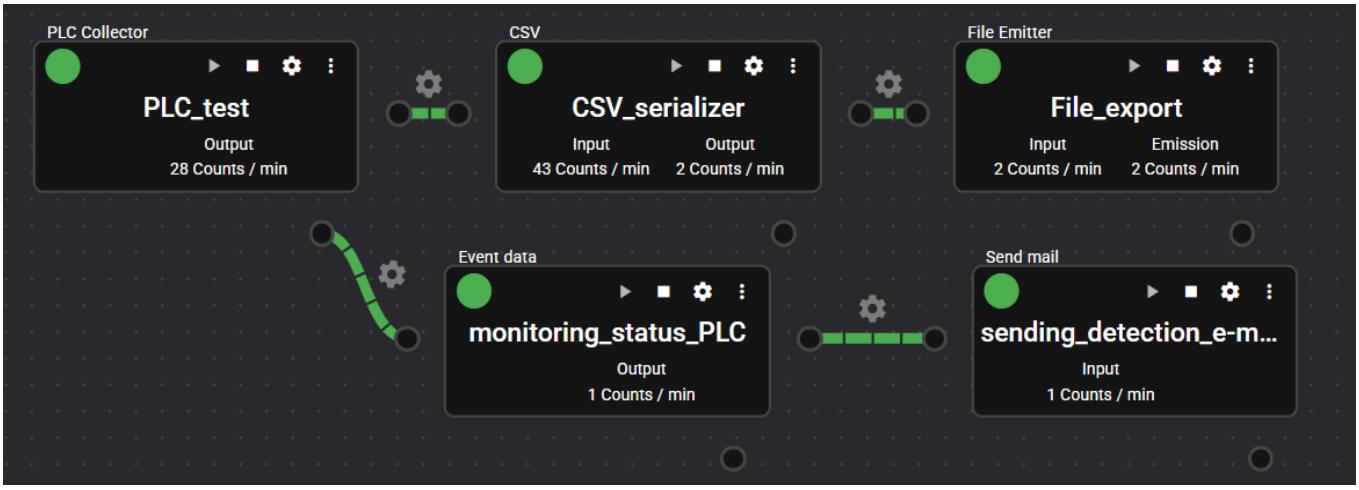
3. 「 저장 」을 선택합니다.


SAVE
DELETE
TEST
CANCEL
Name
sending_detection_e-mails
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS

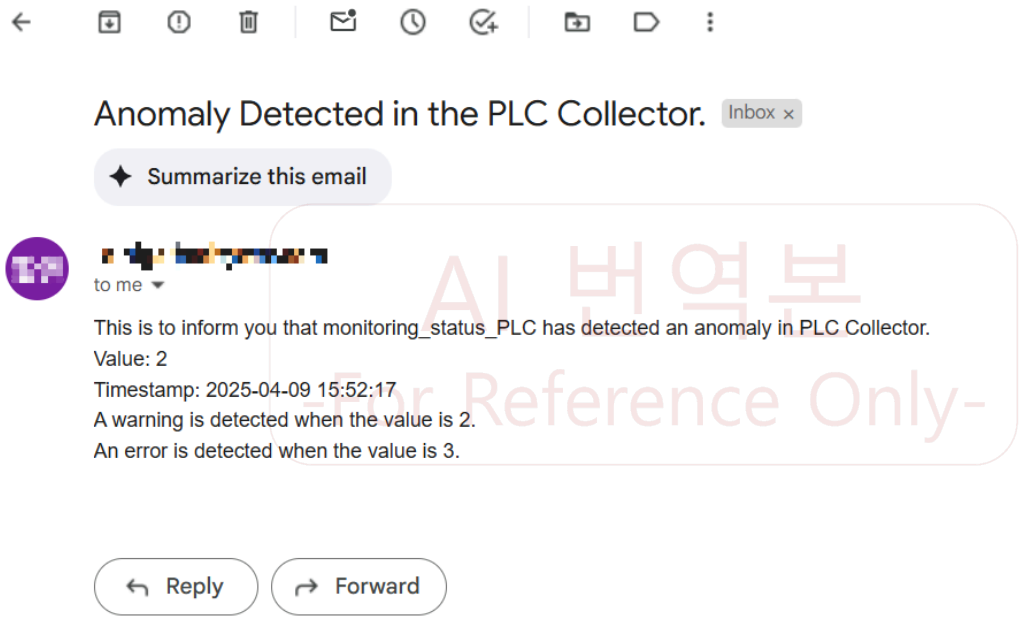
4. 메일 송신을 이벤트 데이터에 접속하고, 컴포넌트를 기동합니다.



이것으로 메일 송신의 설정이 완료되었습니다.



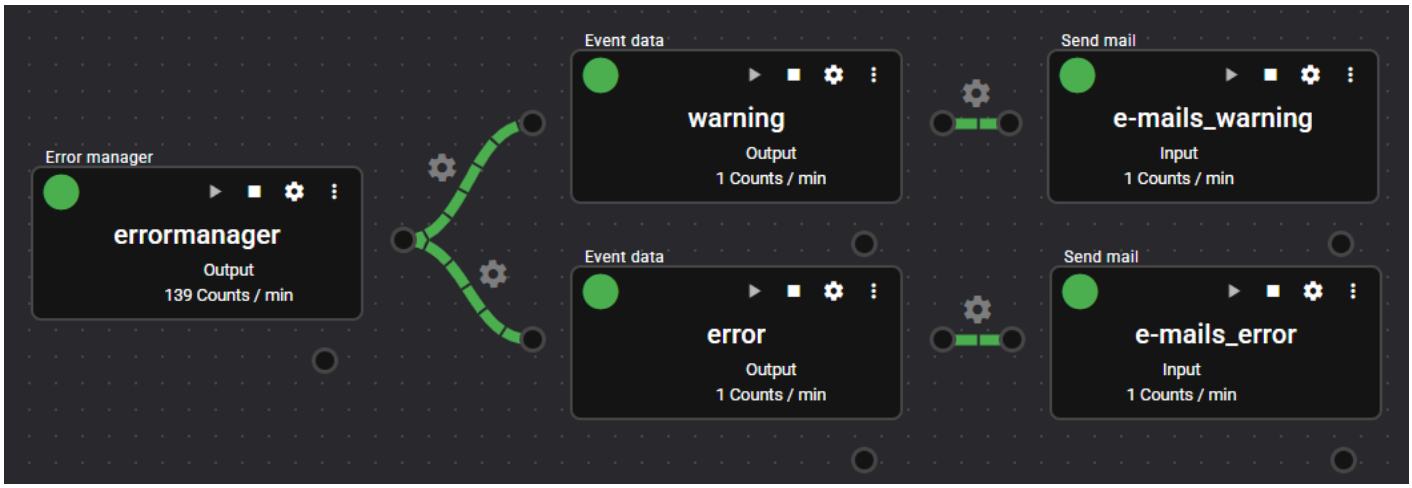
실제로 컴포넌트에서 이상이 발생하면, 설정한 메일 계정으로 데이터가 송신됩니다. 아래는 받은 메일의 예입니다.



AI 번역본
For Reference Only

7.3.2 Synapse 내의 어느 하나의 컴포넌트에서 이상이 발생한 경우에 검지하는 방법

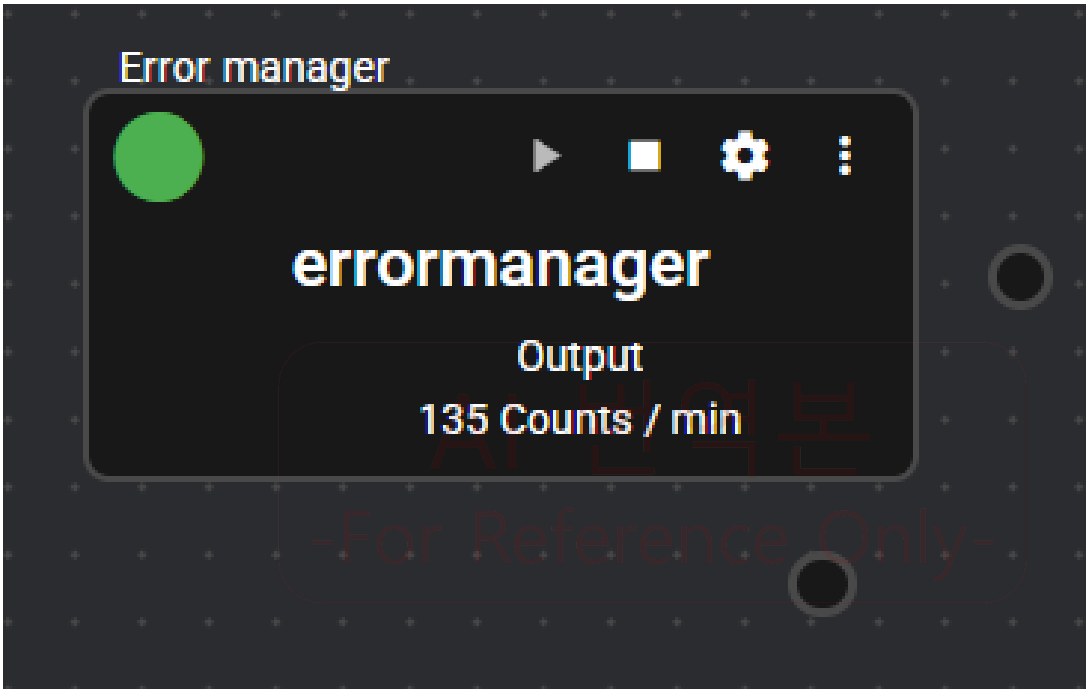
어느 하나의 컴포넌트에서 이상이 발생했을 때에, 메일을 송신하여 알리는 방법을 해설합니다.



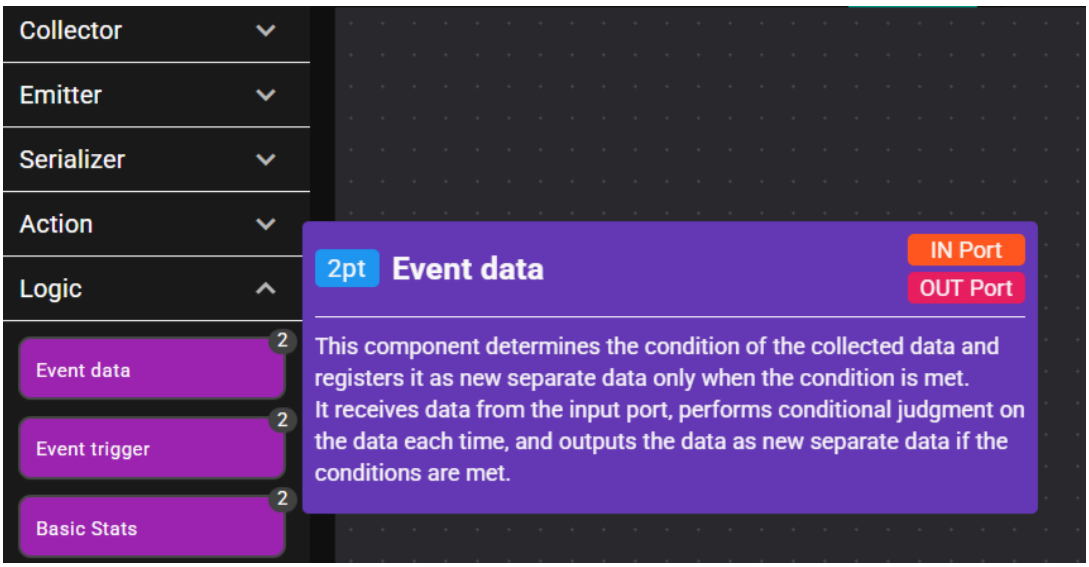
7.3.2.1 이벤트 데이터의 설정

에러 매니저로부터 출력되는「latest_error_component_count」와「latest_warning_component_count」의 값이 증가했을 때, 이벤트 데이터가 등록되도록 설정합니다. Synapse 내에서 어느 하나의 컴포넌트 스테이터스가 경고, 또는 에러인 경우에 메일 통지를 수행합니다.

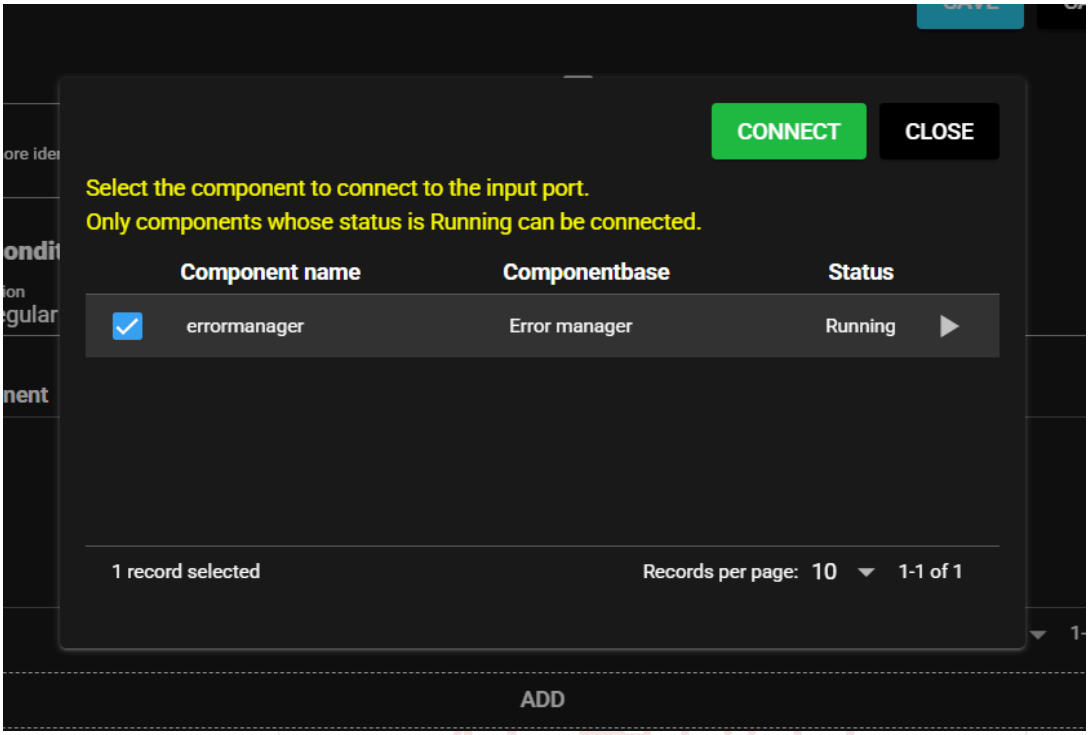
1. Edit Mode에서 시스템 패널을 열고, 에러 매니저를 기동합니다.



2. ComponentBase -> Logic -> 이벤트 데이터를 패널상으로 드래그 앤 드롭합니다.



3. 입력 포트의 접속처 메뉴가 열립니다. errormanager에 체크를 넣고,「 접속 」을 선택합니다.

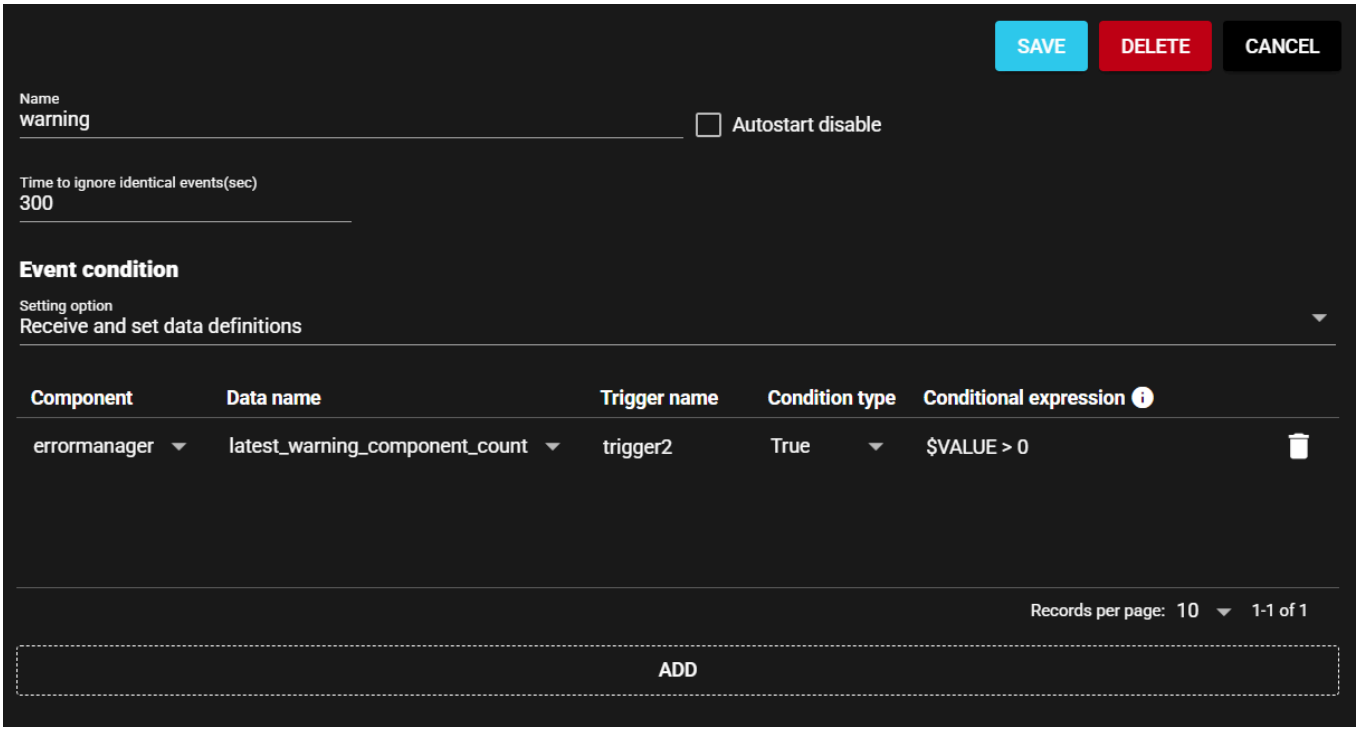


4. 임의의 명칭을 입력한 후, 이벤트 조건을 설정하고「 저장 」을 선택합니다. 이벤트 조건은 아래 조건①②를 설정한 컴포넌트를, 개개에 작성합니다.

AI 번역본
-For Reference Only-

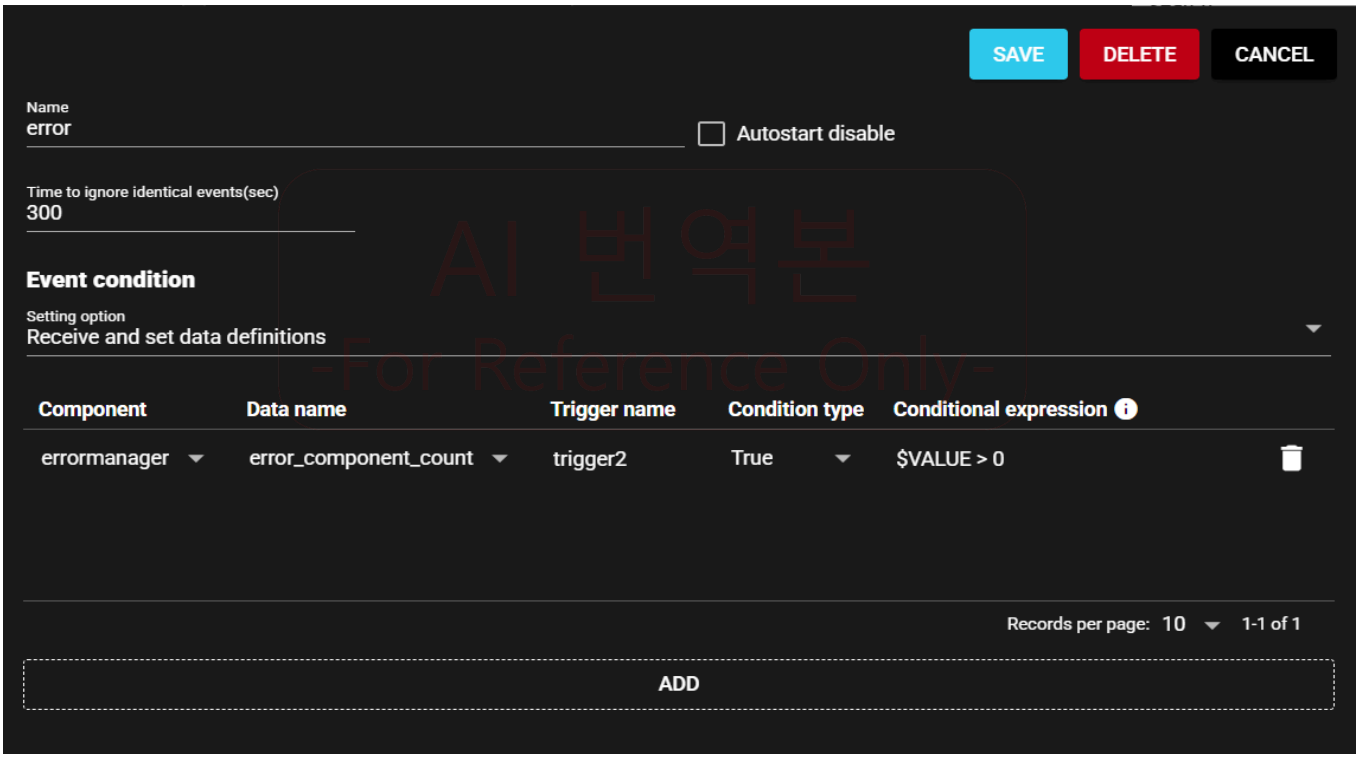
조건①

동일 이벤트를 무시하는 시간(초)「300」, 데이터명「latest_error_component_count」, 트리거명「trigger1」, 조건 종별「True」, 조건식「$VALUE > 0」



조건②

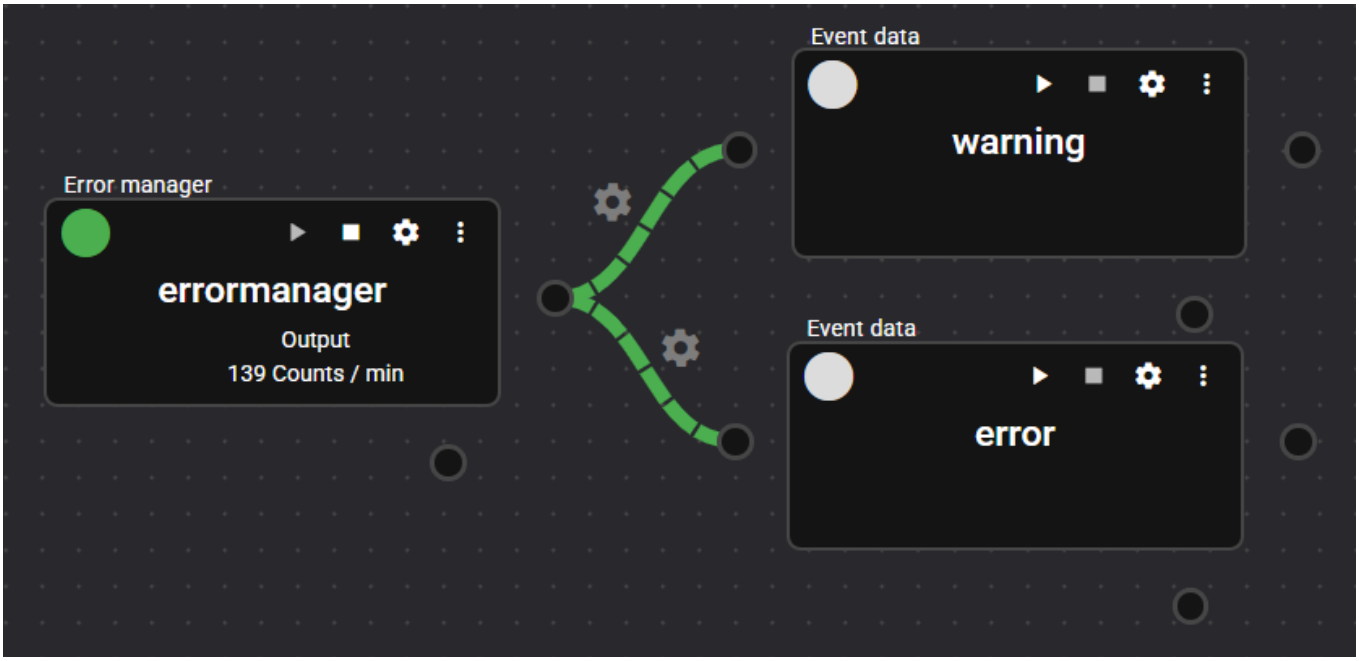
동일 이벤트를 무시하는 시간(초)「300」, 데이터명「latest_warning_component_count」, 트리거명「trigger2」, 조건 종별「True」, 조건식「$VALUE > 0」



AI 번역본
-For Reference Only-

> 메일 송신에 관한 주의
>
> 이 설정에서는, 5분（300초）경과해도, 이상이 계속되는 한 메일이 항상 송신됩니다. 통지량이 방대해지는 경우가 있으므로 주의하십시오. 이상이 발생한 타이밍에 한 번만 메일 송신하는 경우는, 조건 종별을「Raise」으로 설정합니다.

5. 이벤트 데이터를 기동합니다.



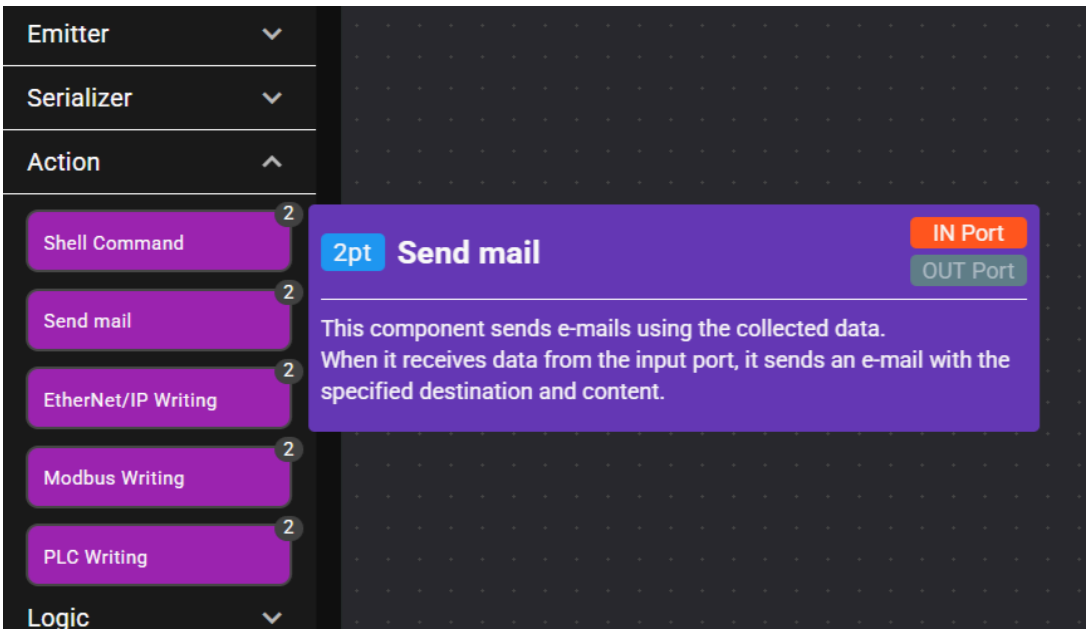
이것으로 이벤트 데이터의 설정이 완료되었습니다.

7.3.2.2 메일 송신의 설정

이벤트 데이터로부터 받은 값을, 메일로 송신하도록 설정합니다.
여기와 마찬가지로, gmail로 송신하는 경우의 설정을 해설합니다. 사용할 수 있는 gmail 계정을 준비하고, 2단계 인증을 설정하여 앱 패스워드를 작성해 두십시오.

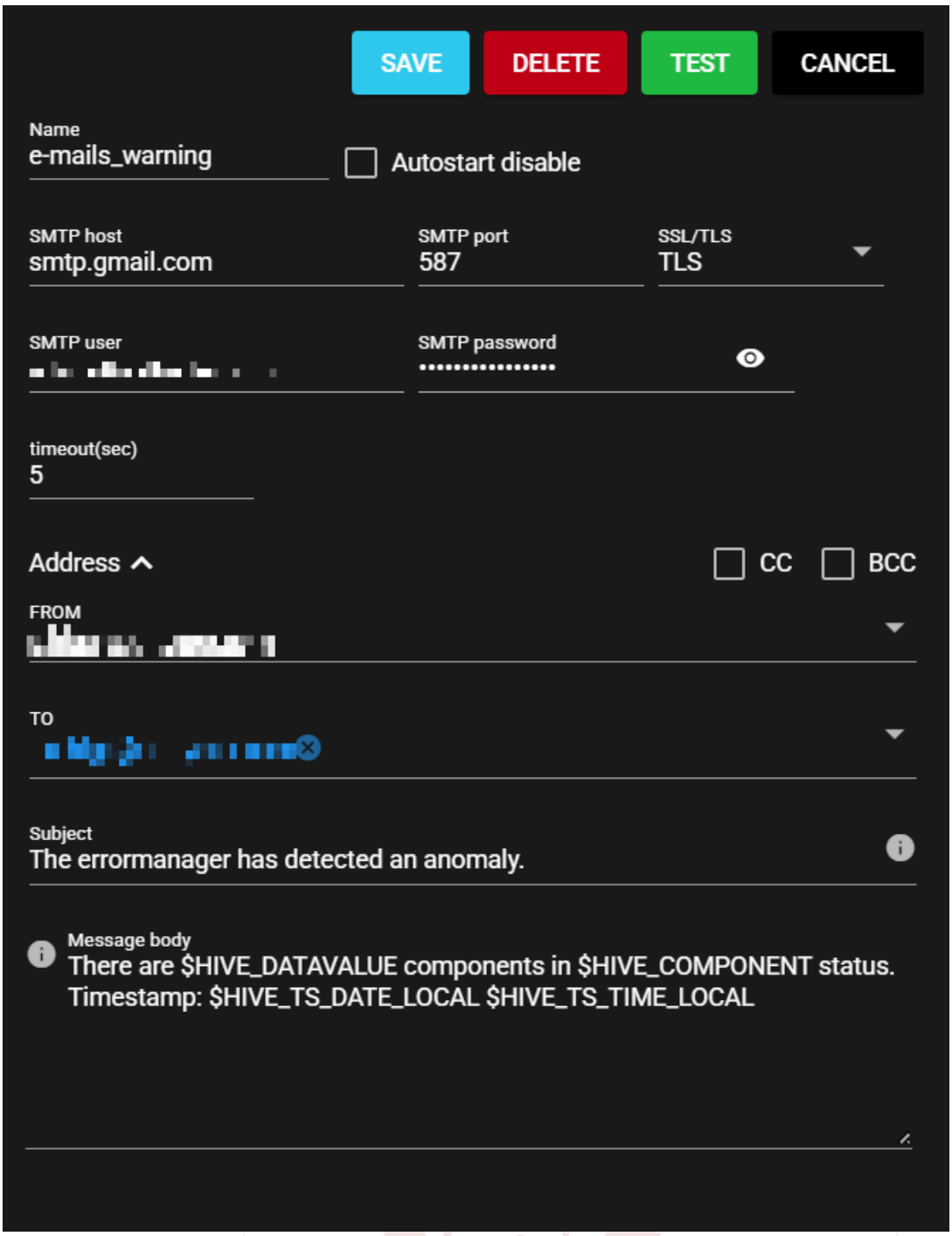
1. 시스템 패널을 열고, Edit Mode에서 ComponentBase -> Action -> 메일 송신을 패널상으로 드래그 앤 드롭합니다.

AI 번역본
-For Reference Only-

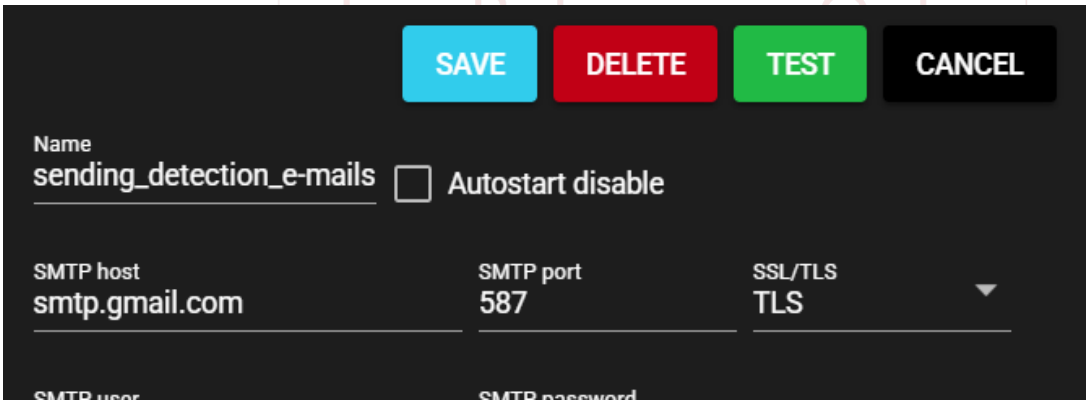


2. 필요한 설정을 수행합니다. gmail의 경우는 SMTP 서버가「 smtp.gmail.com 」, 포트 (TLS)가「 587 」이 됩니다. SMTP 패스워드에는, 작성한 앱 패스워드를 입력하십시오.

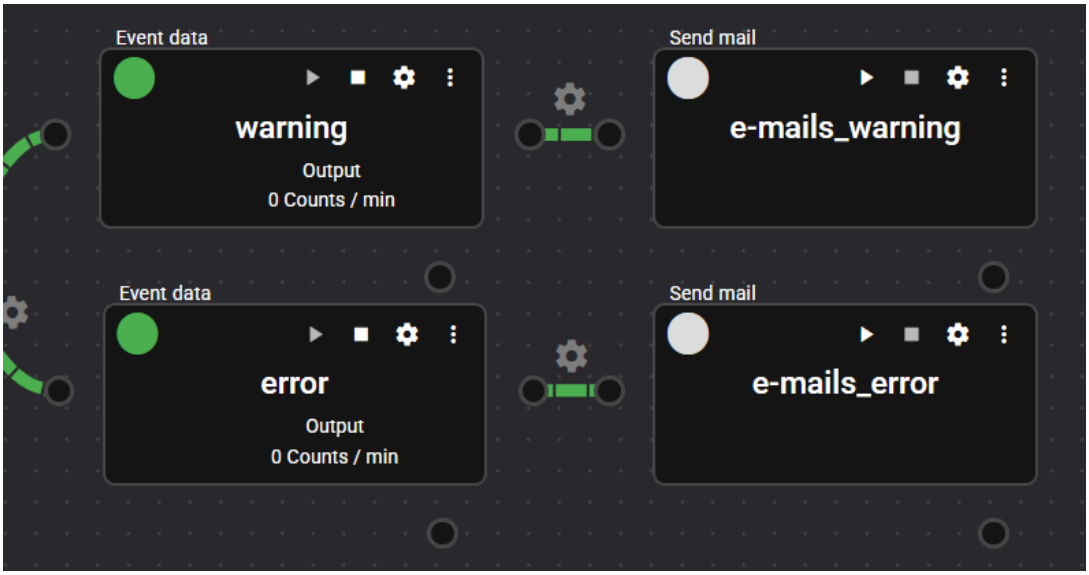
AI 번역본
-For Reference Only-


SAVE
DELETE
TEST
CANCEL
Name
e-mails_warning
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS
SMTP user
SMTP password
timeout(sec)
5
Address
CC
BCC
FROM
TO
Subject
The errormanager has detected an anomaly.
Message body
There are $HIVE_DATAVALUE components in $HIVE_COMPONENT status.
Timestamp: $HIVE_TS_DATE_LOCAL $HIVE_TS_TIME_LOCAL

3. 「 저장 」을 선택합니다.


SAVE
DELETE
TEST
CANCEL
Name
sending_detection_e-mails
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS

4. 같은 순서로 메일 송신 컴포넌트를 하나 더 작성합니다. 그 후, 메일 송신 컴포넌트를 각각의 이벤트 데이터에 접속하고, 컴포넌트를 기동합니다.



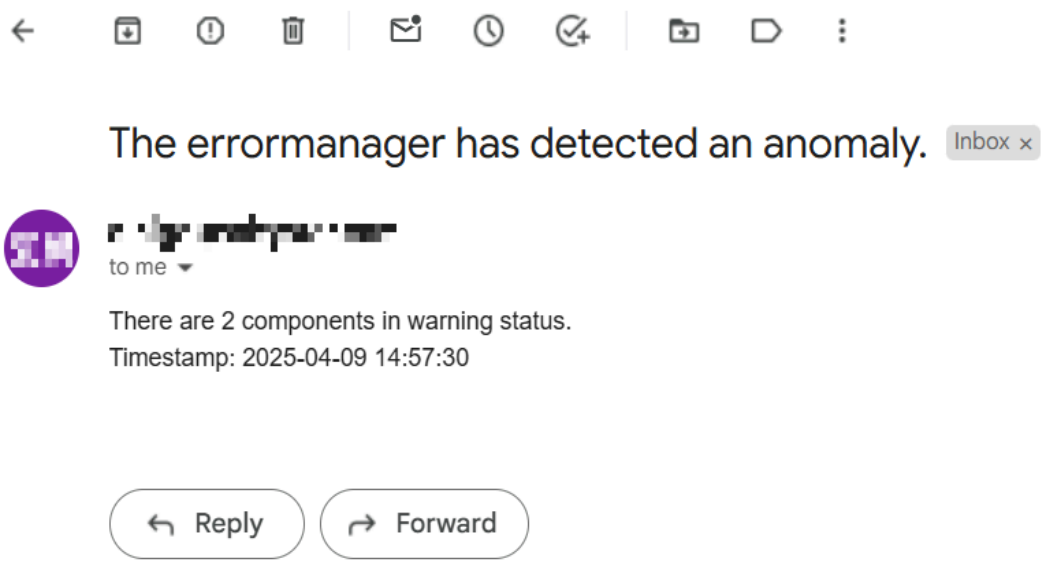
이것으로 메일 송신의 설정이 완료되었습니다.



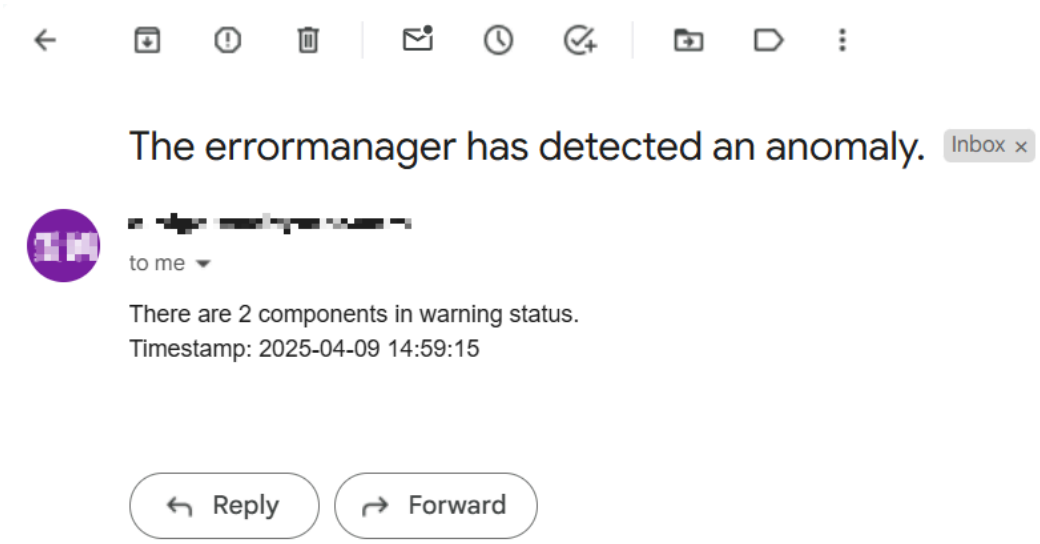
실제로 어느 하나의 컴포넌트에서 이상이 발생하면, 설정한 메일 계정으로 데이터가 송신됩니다. 아래는 받은 메일의 예입니다.

AI 번역본
-For Reference Only-

경고



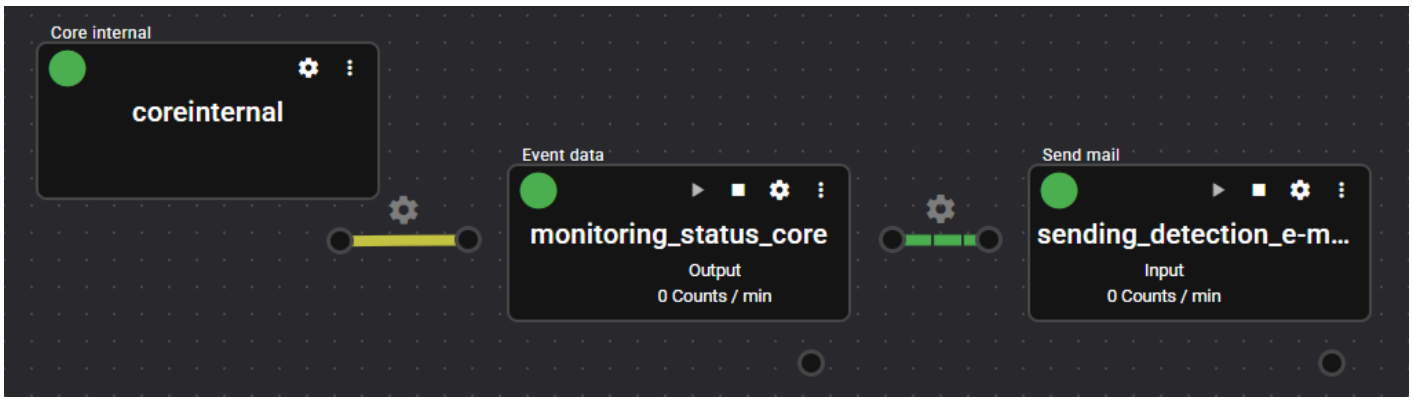
에러



AI 한국어
For Reference Only -

7.3.3 전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법

Synapse에서는, 가동 머신의 전원 차단에 의해 내부에서 사용하고 있는 시계열 데이터베이스에 결손이 발생하면, 코어 인터널은, 다음 기동 시에 그 이상을 경고 상태로서 등록합니다.
시스템 포트에서 이 정보를 포착함으로써, 데이터 결손이 있었을 경우에 메일 통지하는 것이 가능합니다.
여기에서는, 그 순서를 해설합니다.



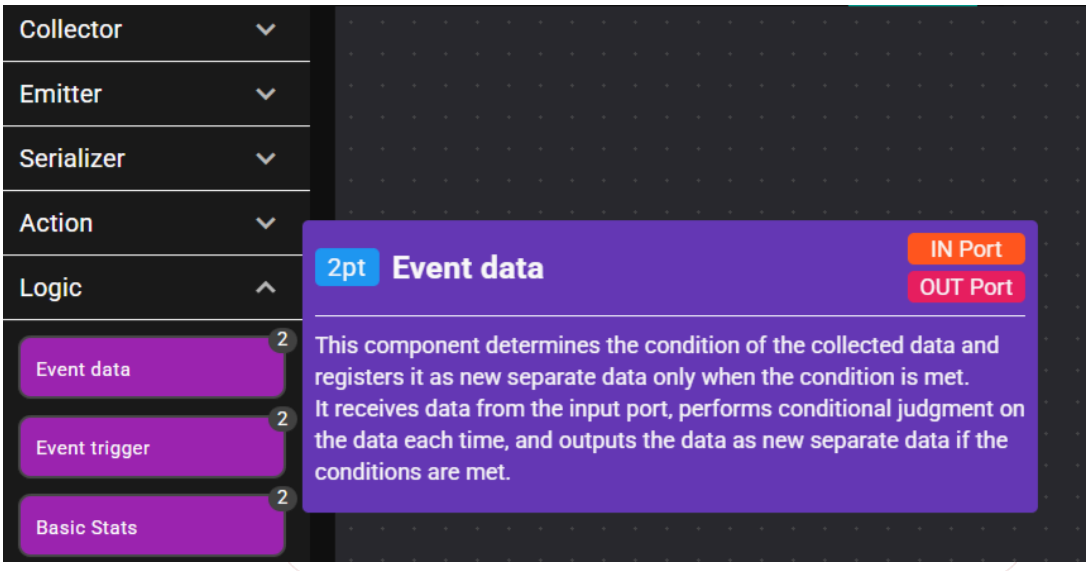
전원 차단에 의한 데이터 결손에 관한 주의

전원 차단에 의한 데이터 결손은 검지 가능하지만, 데이터를 복구하는 것은 할 수 없으므로, 주의하십시오.
어디까지나, 일부의 데이터 저장을 할 수 없었던 것을 검지하는 기능이 됩니다.

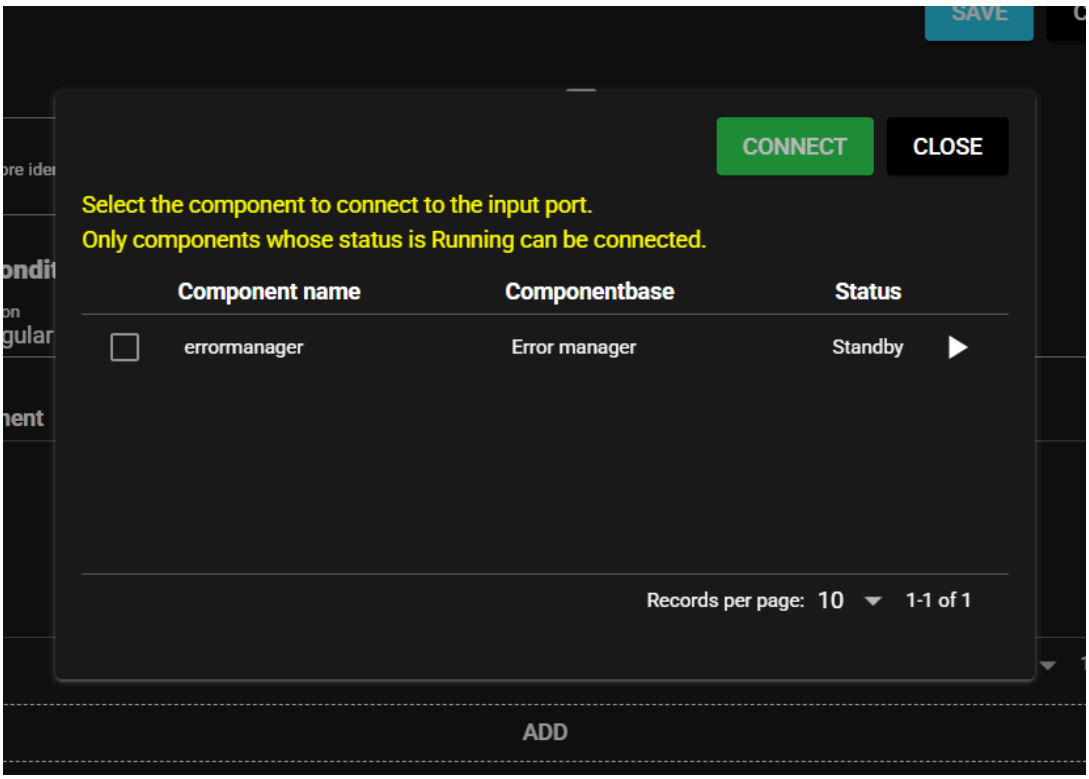
7.3.3.1 이벤트 데이터의 설정

코어 인터널로부터 출력되는 errors_run_status의 값이, 정상이나 정지 이외의 값일 때, 이벤트 데이터가 등록되도록 설정합니다.
스테이터스 램프가 경고 또는 에러인 경우에 메일 통지를 수행하여, 전원 차단을 검지할 수 있습니다.

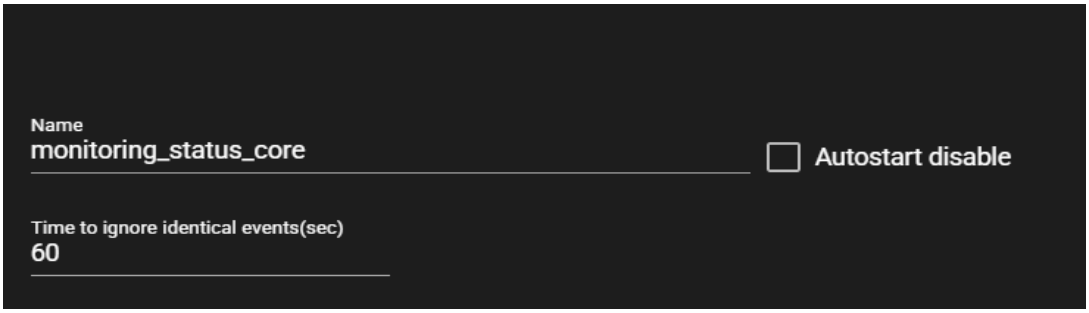
1. 시스템 패널을 열고, Edit Mode에서 ComponentBase -> Logic -> 이벤트 데이터를 패널상으로 드래그 앤 드롭합니다.



2. 입력 포트의 접속처 메뉴가 열립니다. 이번에는 시스템 포트에 접속하기 위해,「닫기」를 선택합니다.

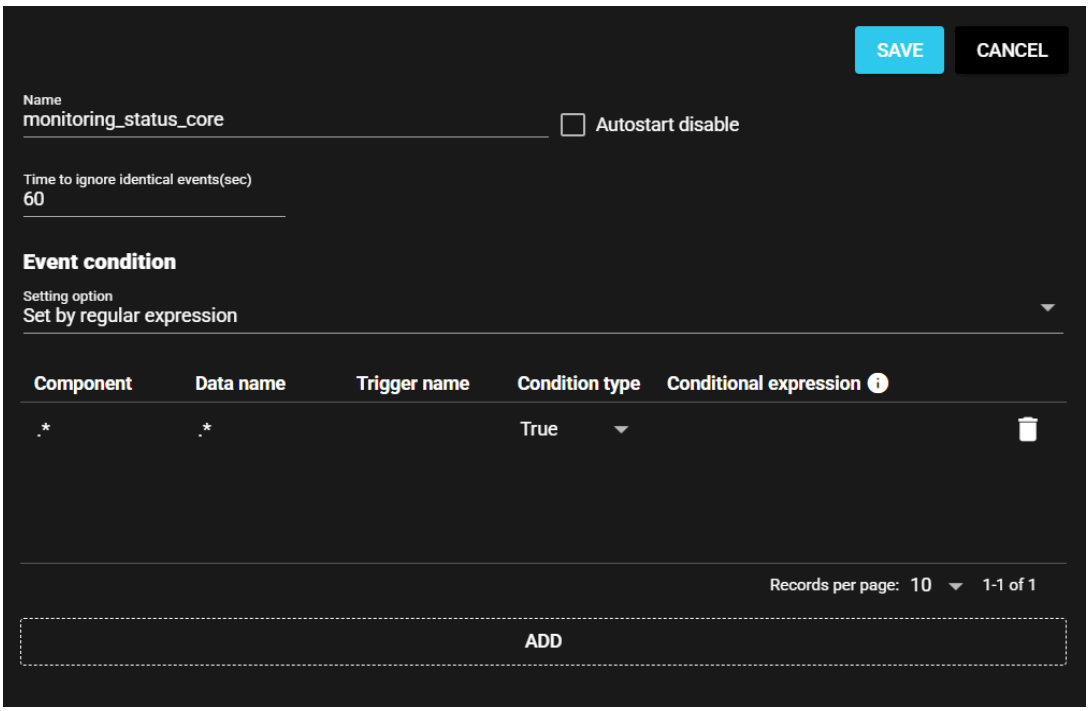


3. 임의의 명칭을 입력합니다. 이번에는 예로서 「 코어 상태 감시 」라는 이름으로 합니다.

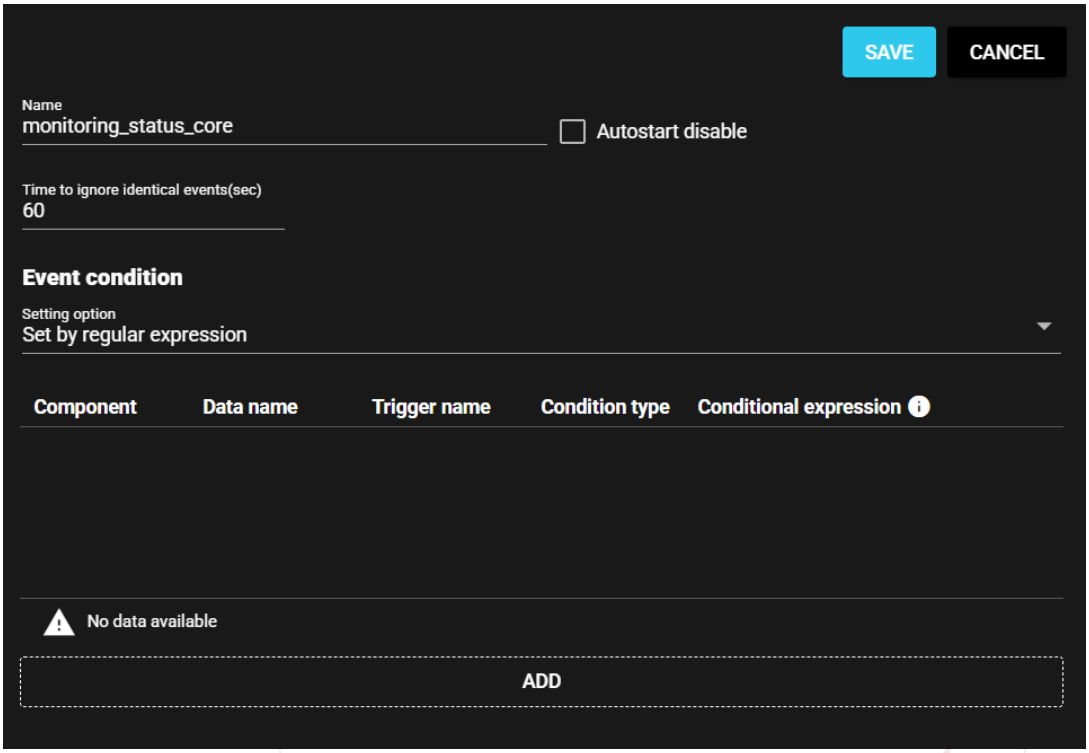


4. 휴지통 마크를 선택하여, 데이터를 삭제합니다.

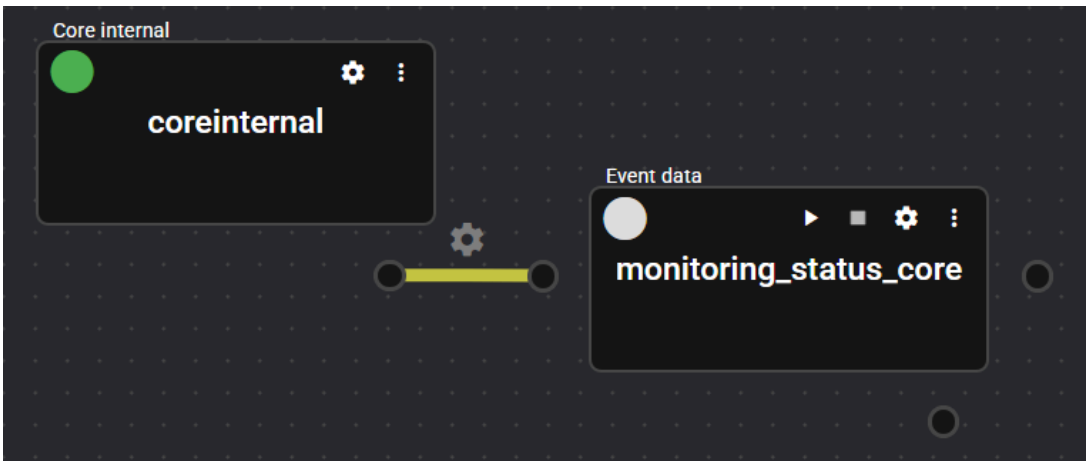
AI 번역본
-For Reference Only-



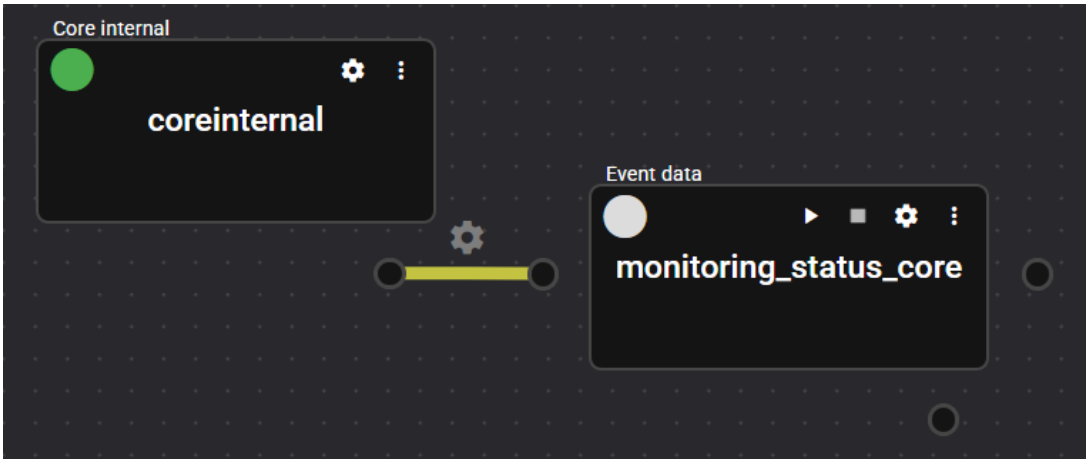
5. 「 저장 」을 선택합니다.



6. 이벤트 데이터를 코어 인터널의 시스템 포트에 접속합니다.

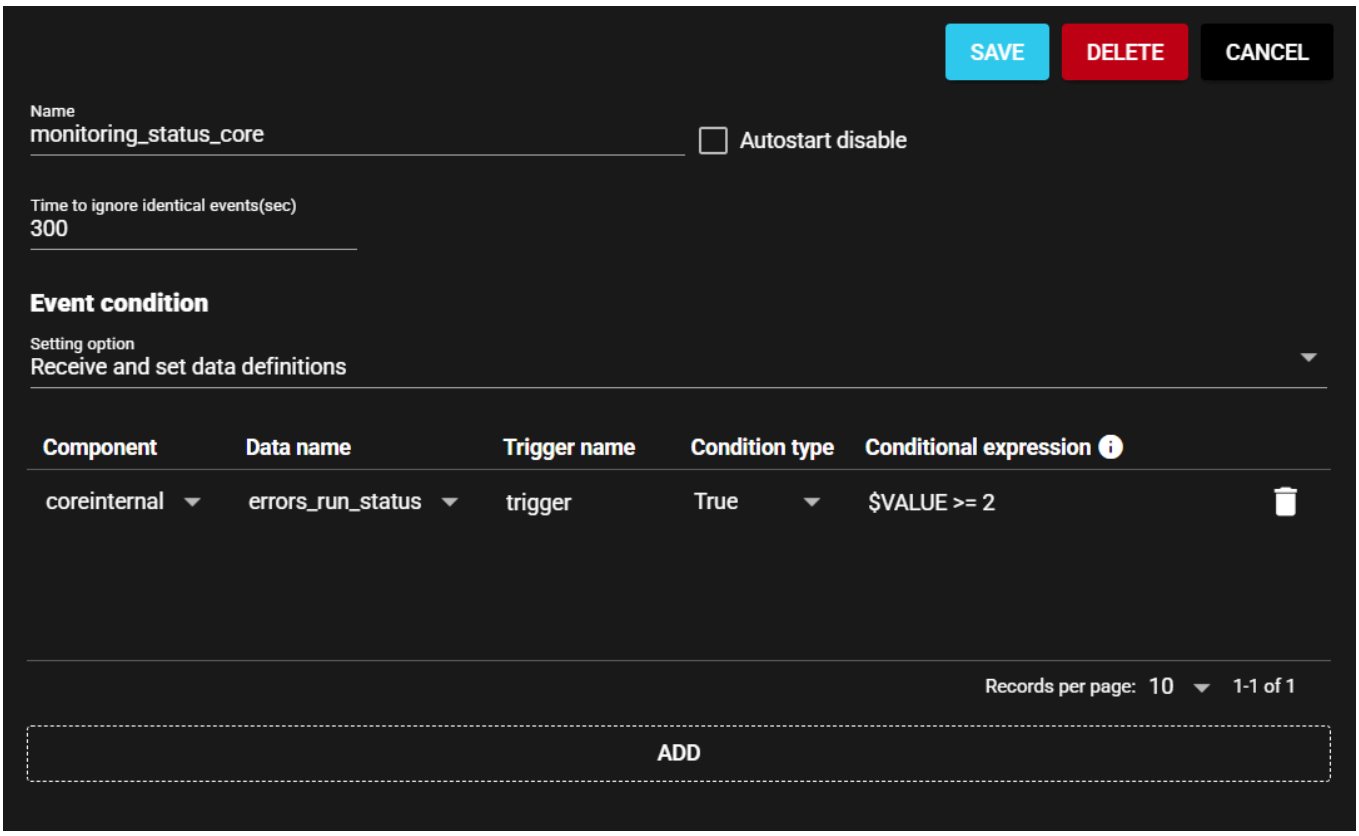


7. 이벤트 데이터의 설정 메뉴를 엽니다.

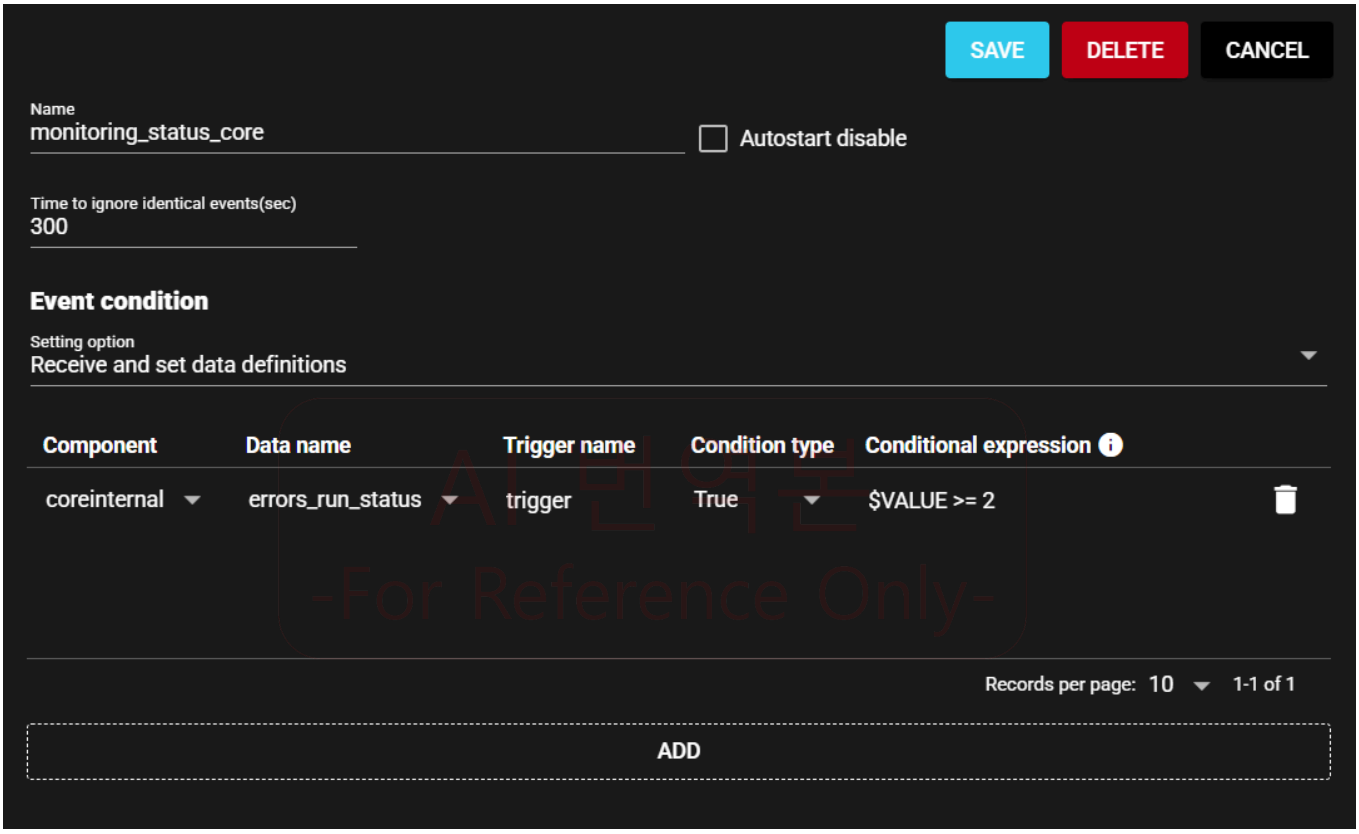


8. 이벤트 조건을 설정합니다. 설정 방법에서 「 데이터 정의를 수신하여 설정 」을 선택합니다. 데이터명을 「 errors_run_status 」, 트리거명을 「 trigger 」, 조건 종별을 「 True 」, 조건식을 「 $VALUE >= 2 」로 설정합니다. 그리고, 동일 이벤트를 무시하는 시간(초)을 「 300 」으로 설정합니다.

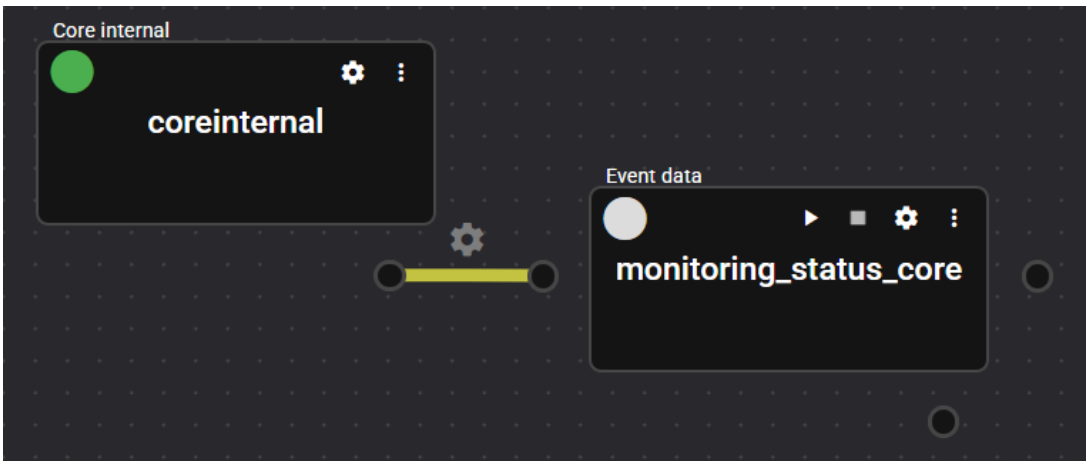
AI 번역본
-For Reference Only-



9. 「 저장 」을 선택합니다.



10. 이벤트 데이터를 기동합니다.

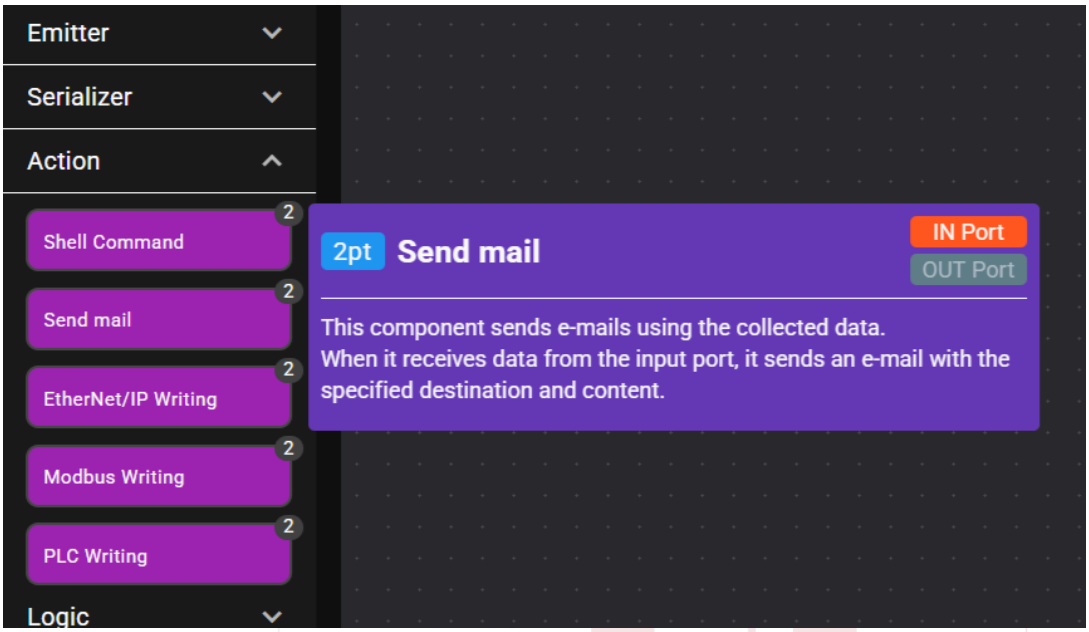


이것으로 이벤트 데이터의 설정이 완료되었습니다.

7.3.3.2 메일 송신의 설정

이벤트 데이터로부터 받은 값을, 메일로 송신하도록 설정합니다.

1. 시스템 패널을 열고, Edit Mode에서 ComponentBase -> Action -> 메일 송신을 패널상으로 드래그 앤 드롭합니다.

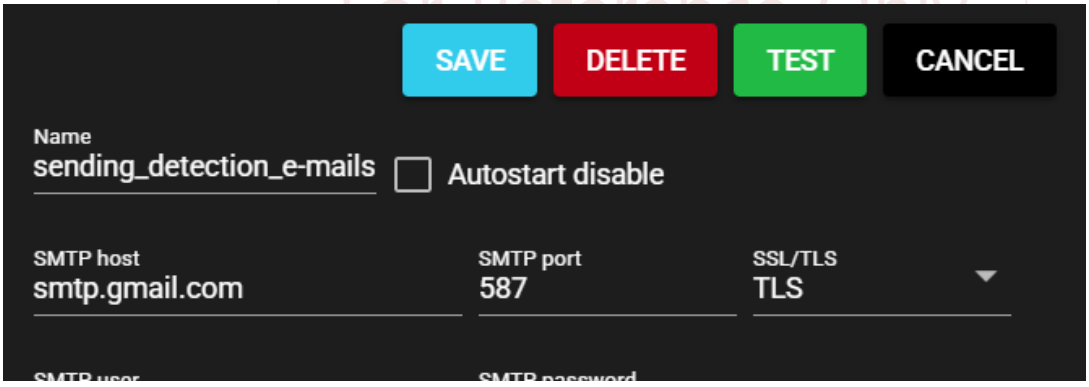


2. 필요한 설정을 수행합니다. gmail의 경우는 SMTP 서버가 「 smtp.gmail.com 」, 포트 (TLS)가 「 587 」이 됩니다. SMTP 패스워드에는, 작성한 앱 패스워드를 입력하십시오.

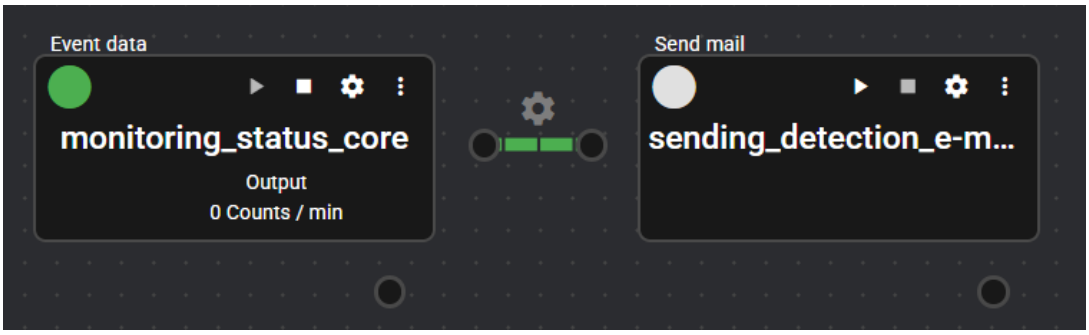
-For Reference Only-


SAVE
DELETE
TEST
CANCEL
Name
sending_detection_e-mails
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS
SMTP user
SMTP password
timeout(sec)
5
Address
CC
BCC
FROM
TO
Subject
Power Disconnection Detected.
Message body
$HIVE_COMPONENT has detected a power disconnection in the core internal system.
Data loss has occurred.
Timestamp: $HIVE_TS_DATE_LOCAL $HIVE_TS_TIME_LOCAL

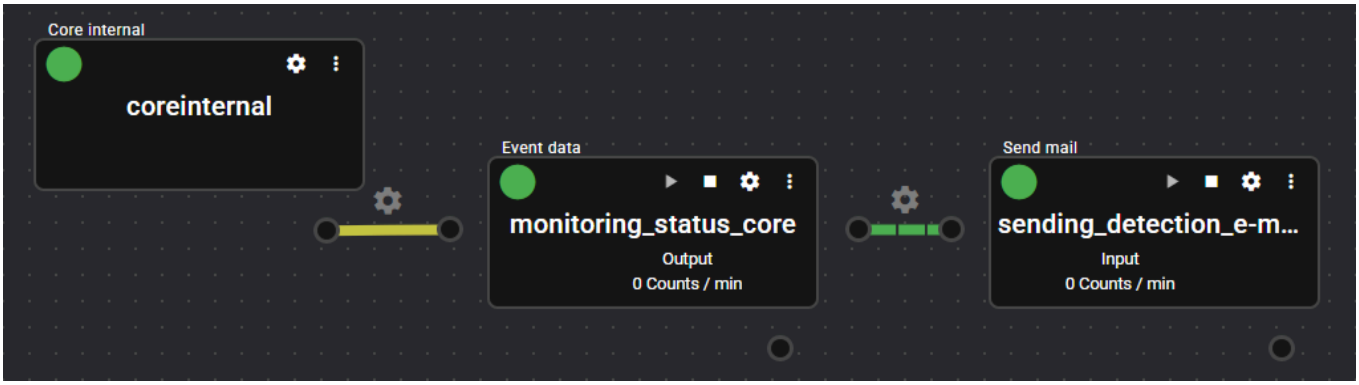
3. 「 저장 」을 선택합니다.


SAVE
DELETE
TEST
CANCEL
Name
sending_detection_e-mails
Autostart disable
SMTP host
smtp.gmail.com
SMTP port
587
SSL/TLS
TLS

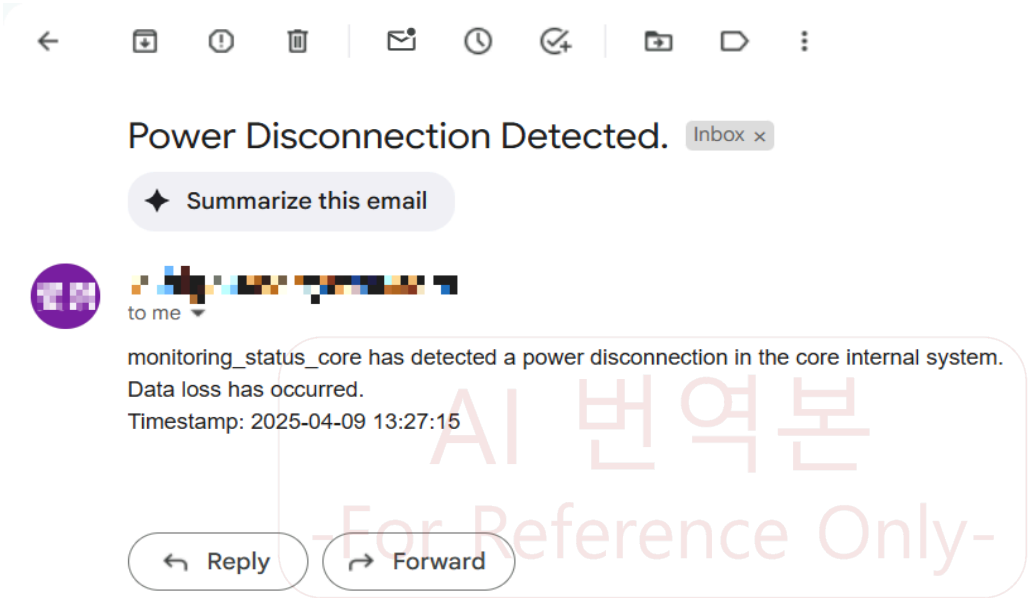
4. 메일 송신을 이벤트 데이터에 접속하고, 컴포넌트를 기동합니다.



이것으로 메일 송신의 설정이 완료되었습니다.



실제로 전원 차단이 발생하면, 설정한 메일 계정으로 데이터가 송신됩니다. 아래는 받은 메일의 예입니다.



AI 번역본
-For Reference Only-

8 보안 이용 가이드

본 장에서는, SpeeDBee Synapse의 안전한 이용 방법에 대해 설명합니다.

8.1 사용자의 관리

보안 수준을 높이기 위한, 사용자 관리의 운용 방법에 대해 설명합니다.

> **admin 사용자의 무효화**
>
> admin 사용자는, SpeeDBee Synapse에 표준으로 등록되어 있는 사용자입니다. 이 사용자에게는 모든 권한이 부여되어 있습니다. 부정한 액세스에 의한 악용을 방지하기 위해, 필요한 경우를 제외하고 무효화할 것을 권장합니다. 무효화하는 경우에는, 별도로 시스템 관리자 권한의 사용자를 추가해 두어야 합니다.
>
> 사용자를 무효로 하는 방법에 대해서는,「 사용자 관리 」를 참조하십시오.

> **강력한 패스워드 정책**
>
> 사용자에게 등록하는 패스워드는, 강력한 것으로 하는 것이 바람직합니다. 아래는, 강력한 패스워드 정책의 예가 됩니다. 패스워드를 자동 생성하는 툴을 사용할 때는, 아래의 조건에 부합하도록 생성할 것을 권장합니다.
>
> - 길이가 8자 이상일 것
> - 영문자의 대문자와 소문자, 숫자, 기호가 각각 1개 이상 사용되어 있을 것
> - 다른 계정의 패스워드와는 다를 것
> - 이름 등의 개인 정보로부터는 추측할 수 없을 것
> - 영단어 등을 그대로 사용하지 않을 것
> - 유추하기 쉬운 배열 방식이나 그 안이한 조합으로 하지 않을 것

AI 번역본
-For Reference Only-

8.2 HTTPS의 이용

SpeeDBee Synapse를 조작할 때에, HTTPS를 이용하여 통신을 암호화함으로써, 보안 수준을 높이는 것이 가능합니다.

HTTPS의 이용 방법에 대해서는,「 HTTP/HTTPS 통신 」을 참조하십시오.

사용하는 증명서에 대해서

증명서를 사용한 암호화 통신을 수행하는 경우, 공적 인증 기관이 발행한 정규 증명서를 이용해 주십시오.
만약 검증 목적 등으로 일시적으로 자체 증명서 등을 이용하는 경우는, SAN(Subject Alternative Name)을 포함하여 증명서를 작성해 주십시오. 각 컴포넌트의 암호화에 이용하는 증명서에 대해서도, 마찬가지로 SAN을 포함해 주십시오.

8.3 토큰의 엄중 관리

액세스 토큰은, 외부로부터 SpeeDBee Synapse에 액세스할 때의, 리소스에 대한 액세스 권한을 나타내는 것입니다. 부정한 공격자에 의해 액세스 토큰이 악용되면, SpeeDBee Synapse의 리소스에 부정하게 액세스될 우려가 있습니다. 액세스 토큰이 제3자의 손에 넘어가는 일이 없도록, 엄중히 관리해 둘 것을 권장합니다.

8.4 암호화 통신의 이용

일부 컴포넌트에서는, 암호화 통신을 이용함으로써, 보안 수준을 높일 수 있습니다.

항목	설명
MQTT 컬렉터	CA 증명서를 준비함으로써, 암호화 통신을 이용할 수 있습니다.
Cloud 에미터(Azure용)	이 에미터에서는, 통신은 암호화되어 있습니다.
Cloud 에미터(AWS용)	이 에미터에서는, 통신은 암호화되어 있습니다.
FTP/FTPS/SFTP 에미터	FTPS 또는 SFTP를 이용함으로써 통신은 암호화됩니다.
메일 송신	TLS를 이용함으로써 더욱 안전하게 통신은 암호화됩니다.

AI 번역본
For Reference Only

8.5 조작 로그의 확인

조작 로그를 정기적으로 확인함으로써, 수상한 액티비티를 조기에 발견할 수 있습니다. 조작 로그의 확인 방법에 대해서는, 「 각종 로그 」를 참조하십시오.

9 후방 호환에 관하여

본 장에서는, 후방 호환에 관하여 설명합니다.

9.1 후방 호환의 대상

SpeeDBee Synapse의 후방 호환에 대한 사고방식에 대해 설명합니다. 후방 호환의 대상은, 최초 2024년도 발매의「 SpeeDBee Synapse 」를 대상으로 합니다. 그 이전의 다른 제품, 유사 제품은 후방 호환의 대상외가 됩니다.（ 제품 호환은 없습니다 ）

대상	설명
설정 정보	사용자가 설정한 각종 정보.（ 패널, 컴포넌트 설정 등, 사용자 관리 정보를 포함 ）
컴포넌트	SpeeDBee Synapse가 제공하는 컴포넌트. 사용자가 추가한 커스텀 컴포넌트를 포함.

9.2 후방 호환

후방 호환의 기본적인 사고방식으로서는, 원칙은 후방 호환을 의식한 제품 제공을 수행합니다.

- 리비전, 마이너 버전 업데이트

기존 설정 그대로 업데이트 가능하며, 후방 호환을 유지합니다.（ 내부 설정의 구조 변경 등은 업데이트 처리에서 수행하는 경우도 있지만, 사용자가 의식할 필요는 없습니다 ）

- 메이저 업데이트

대규모의 기능 확장 등을 수행했을 때의 업데이트입니다. 이 경우, 원칙은 후방 호환의 유지에 힘쓰지만 후방 호환을 지원하지 않는 경우가 있습니다. 그 경우의 업데이트 순서에 대해서는, 업데이트 시에 상세와 함께 개별의 정보 제공을 수행합니다.

AI 번역본
-For Reference Only-

10 실행 라이선스

본 장은, SpeeDBee Synapse의 실행 라이선스에 대해 설명합니다.

10.1 라이선스의 종류

라이선스의 종류에 대해 설명합니다.
현재의 SpeeDBee Synapse에는, 3종류의 라이선스가 존재합니다.

항목	설명
개발 라이선스	SpeeDBee Synapse의 전 기능, 조작이 가능한 라이선스입니다. 시스템 구축은, 통상 이 라이선스를 이용합니다. 본 매뉴얼은 본 장을 제외하고, 개발 라이선스를 전제로 설명하고 있습니다.
실행 라이선스	SpeeDBee Synapse의 일부 기능을 사용할 수 있는 실행 환경용 라이선스입니다. 개발 라이선스에 의해 구축된 설정을 계승하여, 복수 환경에서 동작시킬 수 있습니다.
평가 라이선스	SpeeDBee Synapse를 평가 목적으로 동작시키는 라이선스입니다. 컴포넌트가 기동한 후, 연속 가동할 수 있는 것이 1시간이라는 제약이 있습니다. SpeeDBee Synapse의 구입 전에, 평가 목적으로 사용하는 라이선스입니다.

> 실행 라이선스에 대해서
>
> 실행 라이선스는, 실행 라이선스 단독으로는 이용할 수 없습니다. 최저 1 개 이상의 개발 라이선스를 사용하여 운용하여야 합니다.

AI 번역본
-For Reference Only-

10.2 실행 라이선스 이용 환경의 동작 사양

본 절에서는, 실행 라이선스를 적용한 SpeeDBee Synapse의 동작 사양에 대해 설명합니다. 개발 라이선스와 실행 라이선스에 대해, 대표적인 기능의 가부를 아래 표에 나타냅니다.

항목	개발 라이선스	실행 라이선스
사용자 관리/인증	○	○ ※개발 라이선스의 사용자 정보는 계승되지 않습니다

항목	개발 라이선스	실행 라이선스
Grafana 연계	○	○ ※개발 라이선스의 설정은 계승되지 않습니다
커스텀 컴포넌트 （Python、C）	○	○ ※수동으로 커스텀 컴포넌트나 외부 모듈을 설정할 필요가 있습니다
외부 제어 Web-API	○	○
액세스 토큰 발행	○	○ ※개발 라이선스의 설정은 계승되지 않습니다
암호화 통신	○	○ ※개발 라이선스의 설정은 계승되지 않습니다
DB 쿼리 서비스	○	○ ※설정은 계승되지만, 토큰 인증은 실행 라이선스 측에서 수동으로 재설정할 필요가 있습니다
Run Mode	○	○
Edit Mode	○	△ ※제한이 있는 상태로 이용할 수 있습니다
패널의 추가, 갱신, 삭제	○	×
새로운 컴포넌트의 등록	○	×
설정 완료된 컴포넌트의 설정 갱신	○	○
설정 완료된 컴포넌트의 삭제	○	×
플로우 링크의 추가, 삭제	○	×
각종 포트의 설정	○	○

AI 번역본
-For Reference Only-

항목	개발 라이선스	실행 라이선스
필터의 설정	○	×
데이터 모니터	○	○
스테이터스 타임라인	○	○
로그	○	○
실행 라이선스 설정 다운로드	○	×
프로젝트 설정	○	×

이종 OS 간의 실행 라이선스용 설정 가져오기에 대해서

개발 라이선스 가동 환경과 실행 라이선스 가동 환경의 OS가 다른 경우, 실행 라이선스용 설정을 가져올 때, 일부의 설정이 자동으로 변환됩니다. 단, 아래 이외는 변환되지 않으므로, 수동으로 재설정이 필요해지는 경우가 있습니다.

- 개발 라이선스 측 Linux— > 실행 라이선스 측 Windows의 경우

  설정 파일 중의 경로를 다음과 같이 변환합니다.

  ```
  /var/speedbeesynapse -> C:\ProgramData\SALTYSTER\SpeeDBeeSynapse
  ```

- 개발 라이선스 측 Windows— > 실행 라이선스 측 Linux의 경우

  설정 파일 중의 경로를 다음과 같이 변환합니다.

  ```
  C:\ProgramData\SALTYSTER\SpeeDBeeSynapse -> /var/speedbeesynapse
  ```

※설정을 가져온 후, 이상이 있는 경우는, 수동으로 재설정하십시오.

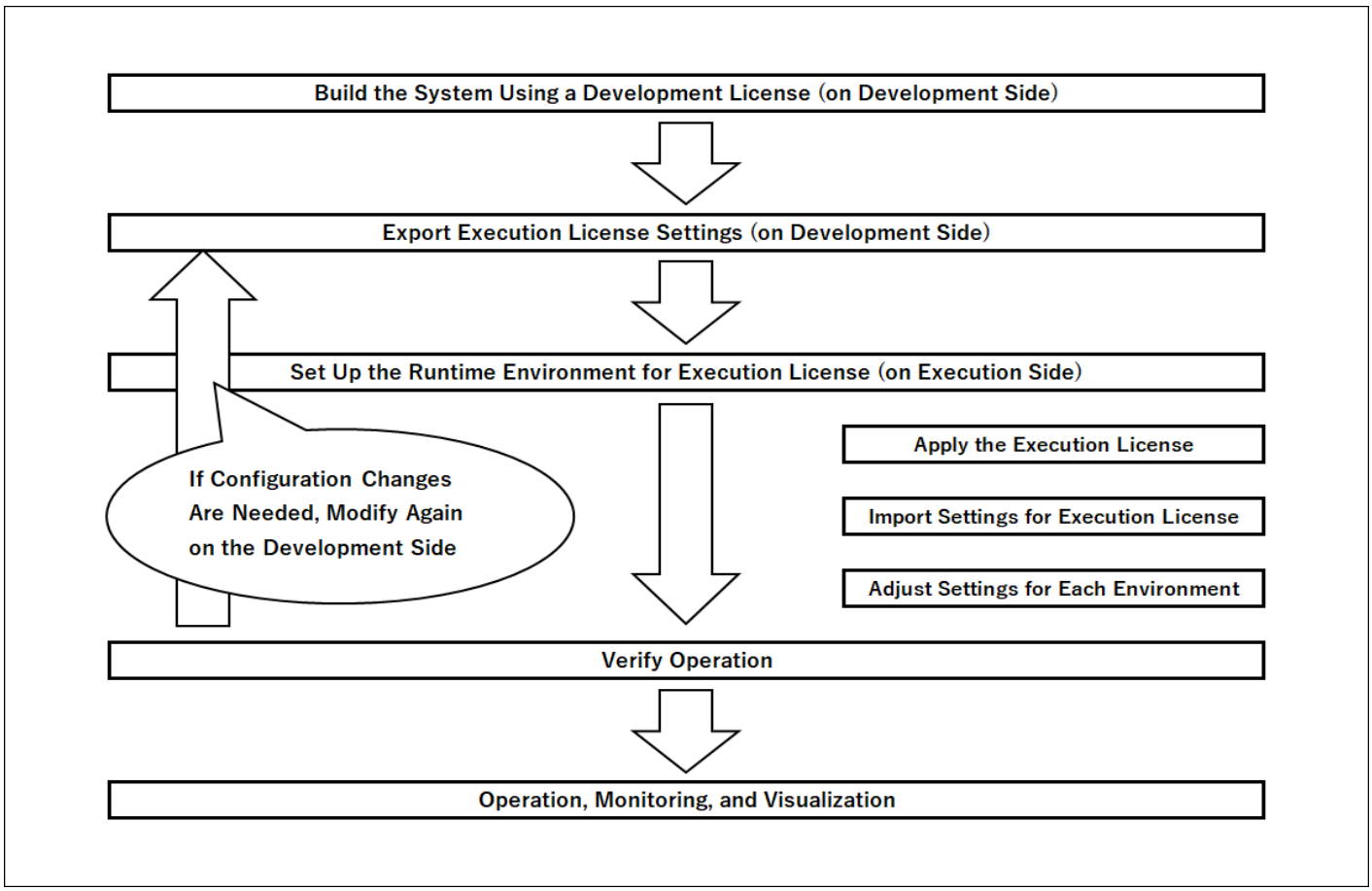
For Reference Only

10.3 실행 라이선스의 운용 방법

본 절에서는, 실행 라이선스의 운용 방법에 대해 설명합니다.

10.3.1 운용까지의 흐름

실행 라이선스의 운용 흐름을 아래 그림에 나타냅니다.



①개발 라이선스를 사용한 시스템 구축（개발 라이선스 측 조작）

먼저, 개발 라이선스를 적용한 SpeeDBee Synapse를 사용하여 시스템을 구축합니다. 데이터 수집, 출력 포맷
, 데이터 송신・연계 등의 데이터 플로우 설정을 수행합니다. 구체적인 설정 내용은, 시스템의 목적마다 다르지만, 참고로서, 도입 플로우를 참조하십시오.

②실행 라이선스 설정의 출력（개발 라이선스 측 조작）

실행 라이선스에서 이용 예정인 데이터 플로우의 설정이 끝나면, 실행 라이선스 설정의 출력을 수행합니다. 이 조작은, 개발 라이선스를 적용한 SpeeDBee Synapse에서만 이용할 수 있습니다.

③실행 라이선스 가동 환경의 셋업（실행 라이선스 측 조작）

실행 라이선스용의 가동 머신 셋업 및 SpeeDBee Synapse의 설치 등을 수행합니다. 설치 후에, 구입한 실행 라이선스의 적용을 수행합니다. 이로써 실행 라이선스용의 설정이 적용되지만,
실행 라이선스 가동 환경마다의 설정을 필요로 하는 경우가 있습니다. 예를 들어, 외부 기기의 IP 주소 설정 등은 이용 환경에 맞춰 설정이 필요한 경우가 있습니다.

AI 번역본
For Reference Only

④동작 확인

실행 라이선스 가동 환경의 설정이 끝나면, 실제로 시스템을 동작시킵니다. 외부 기기의 설정 항목 등의 미비가 없는지 확인합니다.
또한, 이 조작에서는, 새로운 컴포넌트를 추가하여, 새로운 데이터 플로우를 설정하는 것은 할 수 없습니다. 재설정이 필요한 경우, ①개발 라이선스를 사용한 시스템 구축, 및 ②실행 라이선스 설정의 출력을 다시 수행하여, 실행 라이선스용 설정을 가져와야 합니다.

⑤운용・모니터링・가시화

실행 라이선스 가동 환경에서의 운용을 개시합니다. 데이터 플로우의 변경 등이 필요해진 경우는, 다시 개발 라이선스 측에서의 조작을 수행하고, 본 순서에 따라 설정을 변경하십시오.

10.3.2 간이 튜토리얼

여기에서는, 실제로 실행 라이선스 가동 환경을 구축하기까지의 흐름을 해설합니다. 또한, 본 튜토리얼에서는, 개발 라이선스 가동 머신과 실행 라이선스 가동 머신은, 동일한 네트워크 내에 배치되어 있는 것을 전제로 하고 있습니다. 또한, SpeeDBee Synapse는, 각각 설치는 사전에 완료되어 있는 것으로 합니다.

본 튜토리얼에서, 이용하는 데이터 플로우는 다음과 같습니다.

항목	설명
A라인 연계 데이터	PLC 컬렉터로 수집한 데이터를 JSON 형식으로, MQTT에 의해 외부로 데이터 송신

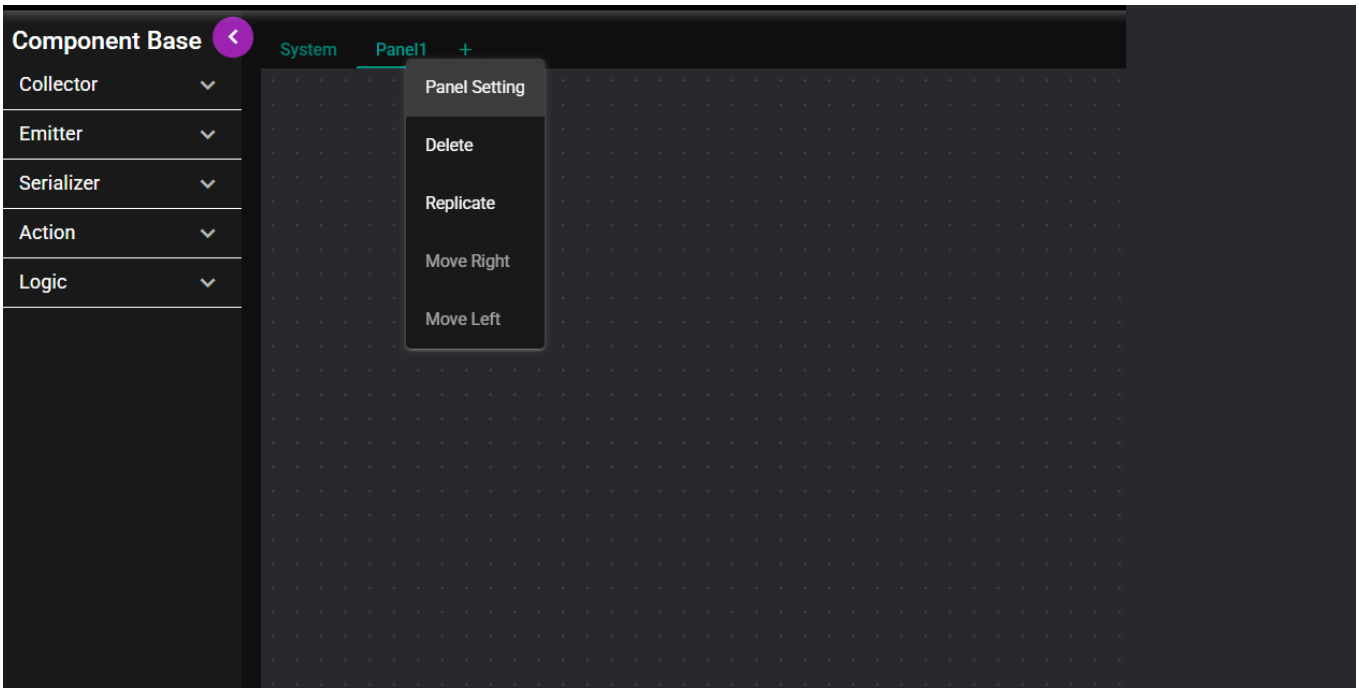


10.3.2.1 개발 라이선스를 사용한 시스템 구축

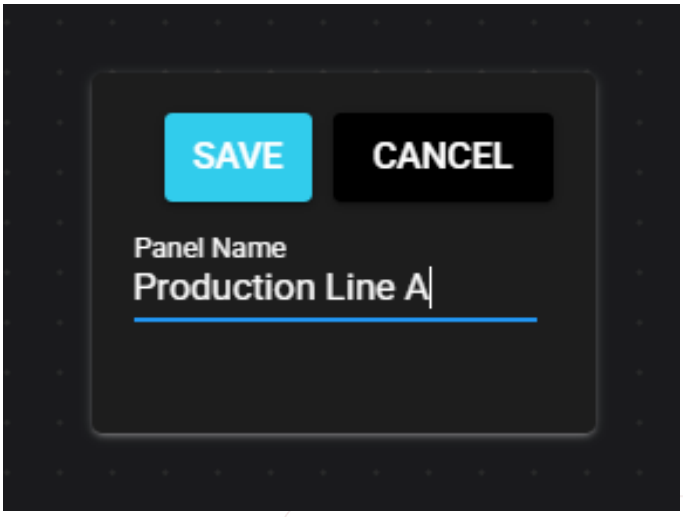
여기에서의 조작은, 개발 라이선스를 사용한 SpeeDBee Synapse에서 수행합니다. 이후를 참고하여, PLC 컬렉터, JSON 시리얼라이저, MQTT 에미터를 설정하십시오.

-For Reference Only-

1. 「 패널1 」을 선택하고, 오른쪽 클릭 메뉴에서 「 패널 설정 」을 클릭합니다.

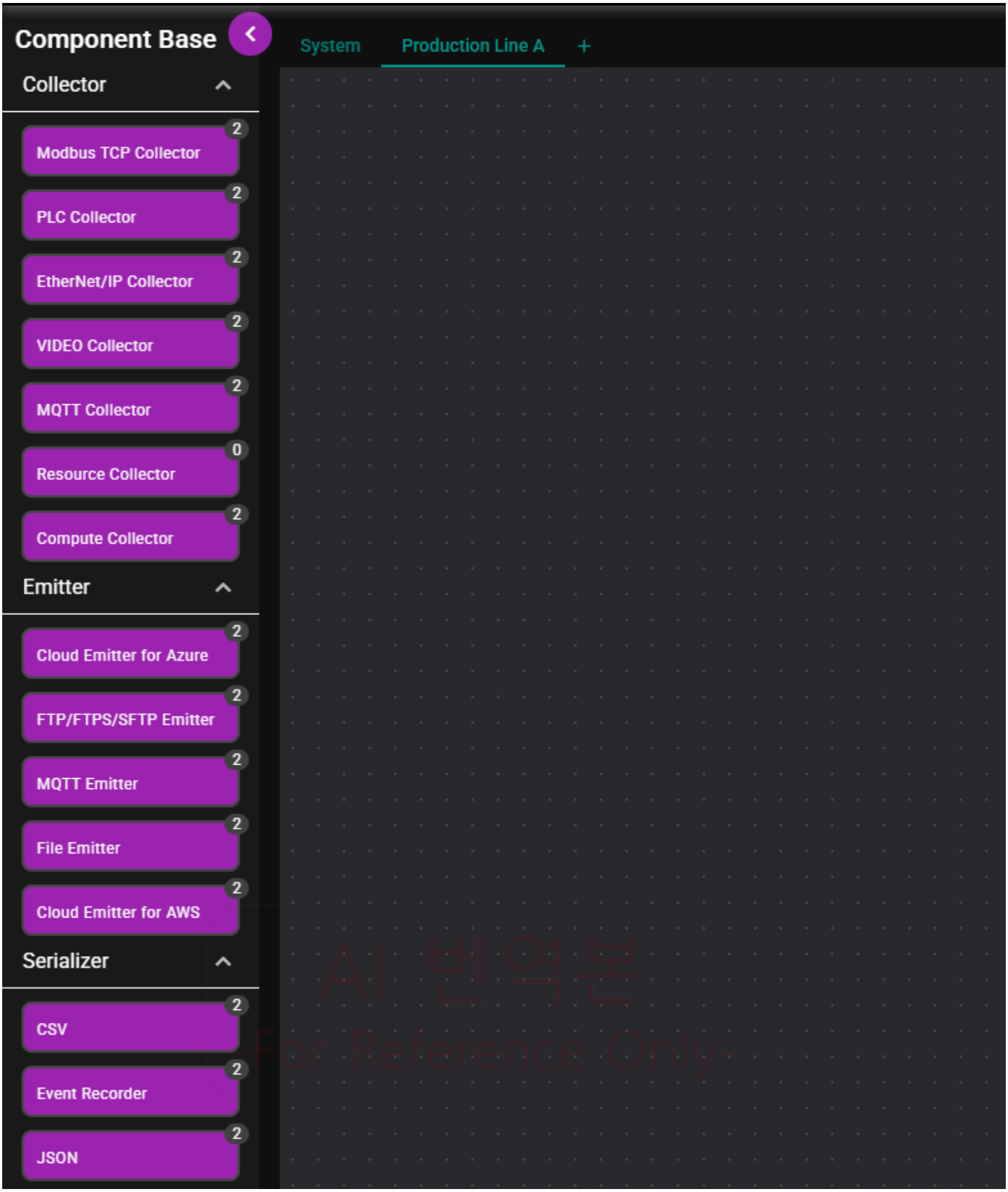


2. 「 패널1 」을 「 A라인 연계 데이터 」로 변경하여 저장합니다.



AI 번역본
-For Reference Only-

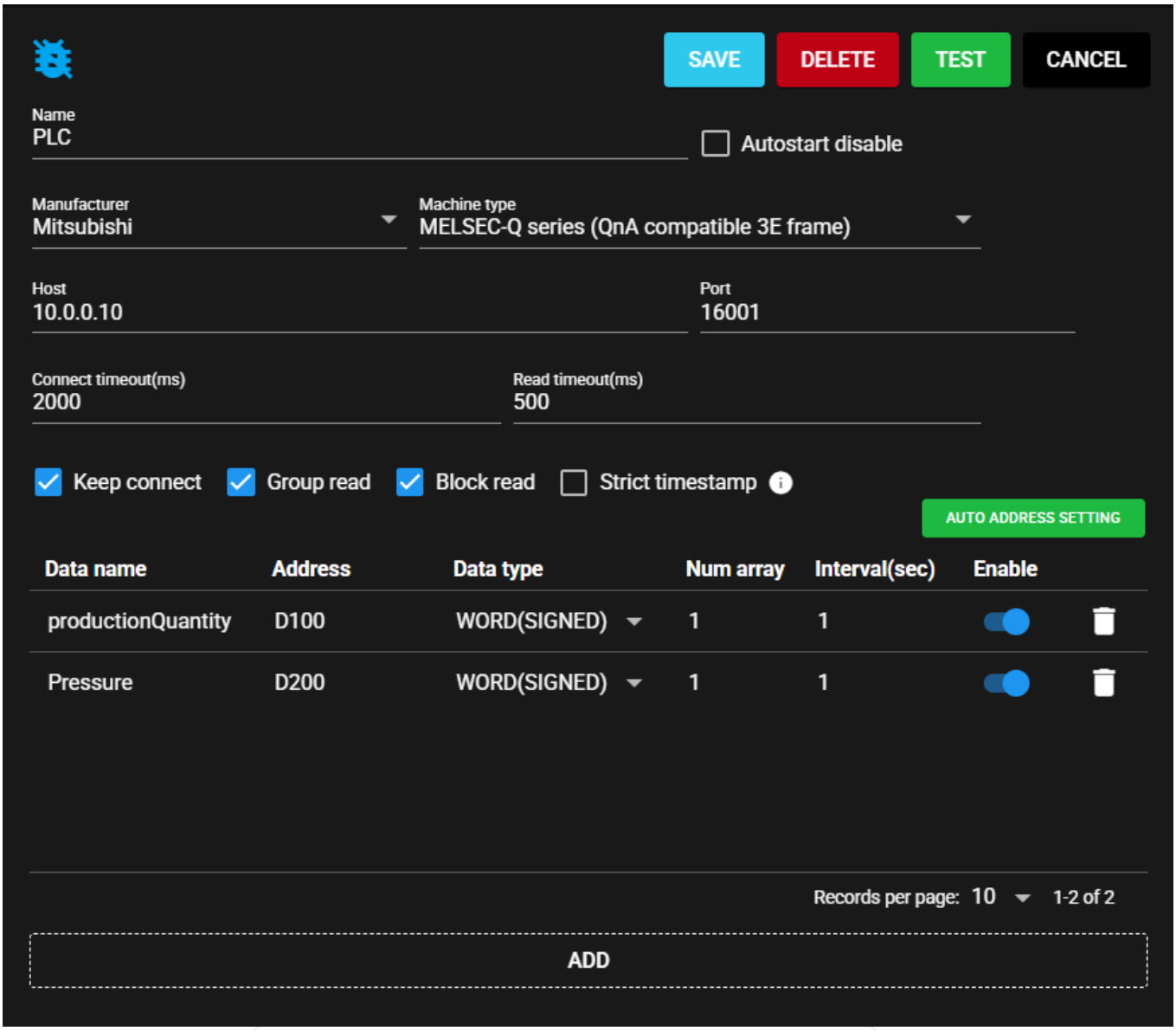
3. PLC 컬렉터의 설정을 수행합니다. 플로우 에디터 좌부에서 「 PLC 컬렉터 」를 플로우 에디터 중앙으로 드래그 & 드롭합니다.



AI 번역본
For Reference Only

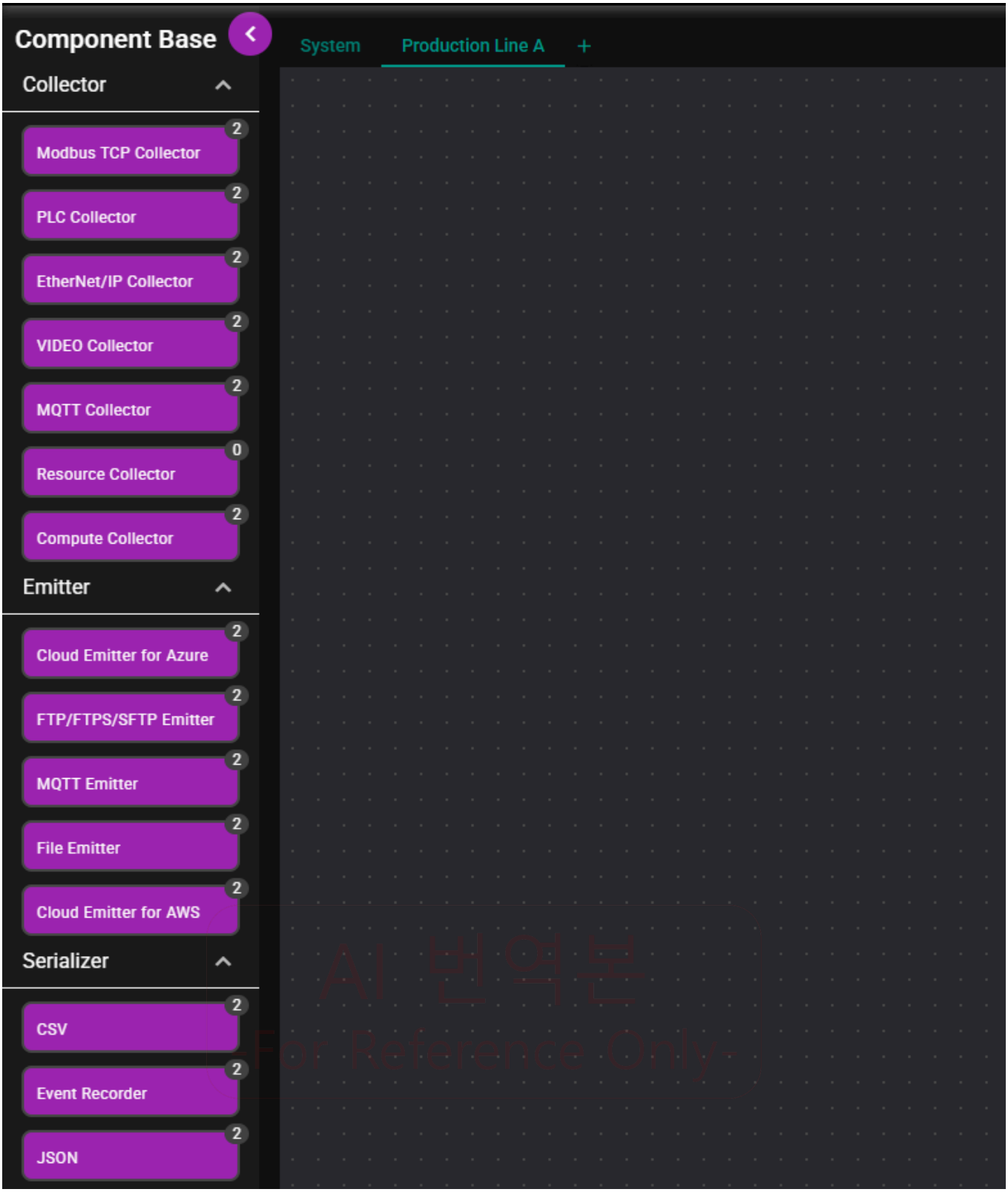
4. PLC 컬렉터의 설정값을 입력하고, 「 저장 」을 클릭합니다. IP 주소나 포트, 데이터 등은 이용하시는 환경에 맞춰 주십시오.

(※만약 PLC 컬렉터를 이용할 수 없는 경우, 리소스 컬렉터 등으로 대체하여 설정하십시오.)



AI 번역본
-For Reference Only-

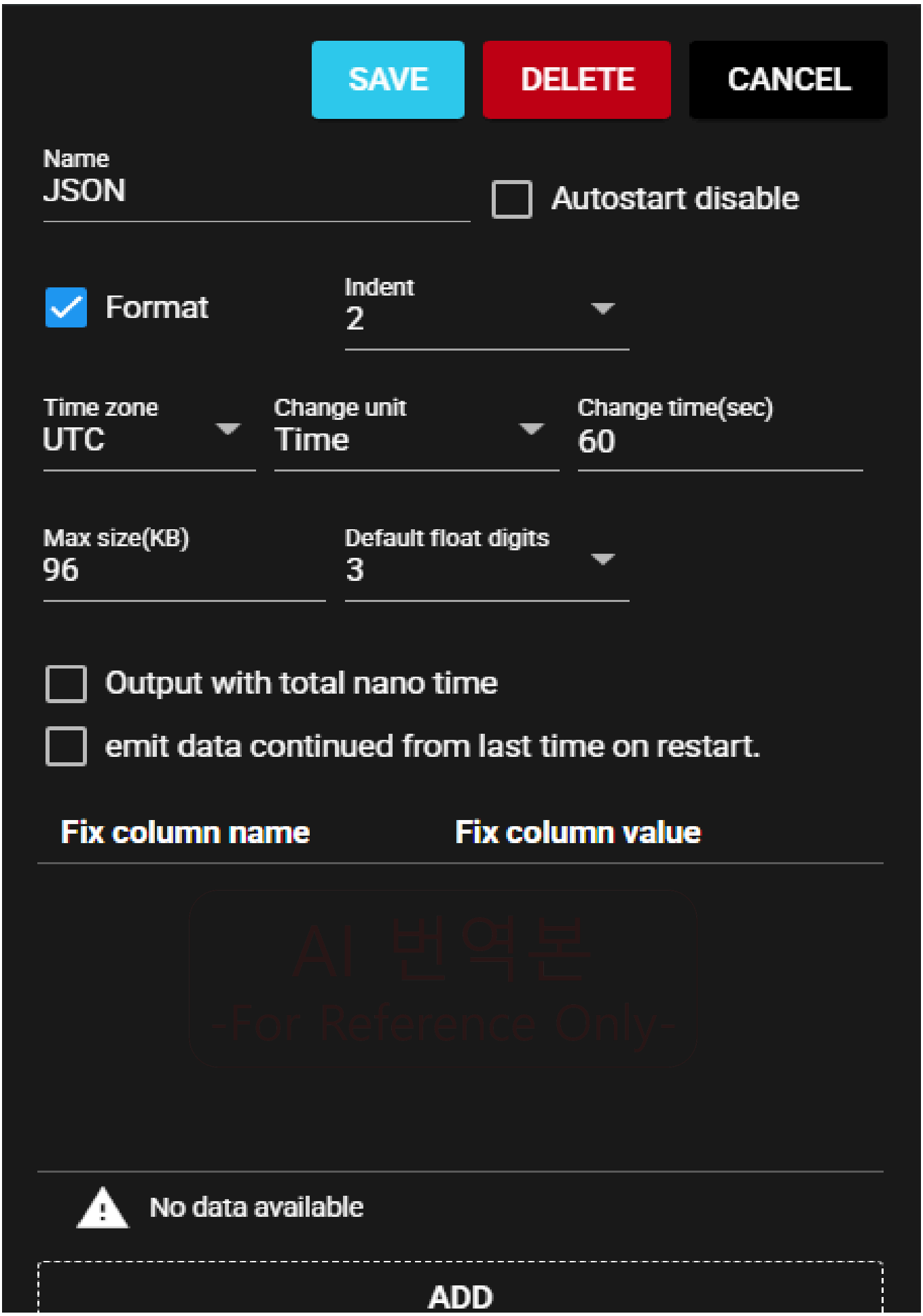
5. 시리얼라이저의 설정을 수행합니다. 플로우 에디터 좌부에서 「 JSON 」을 플로우 에디터 중앙으로 드래그＆드롭 합니다.



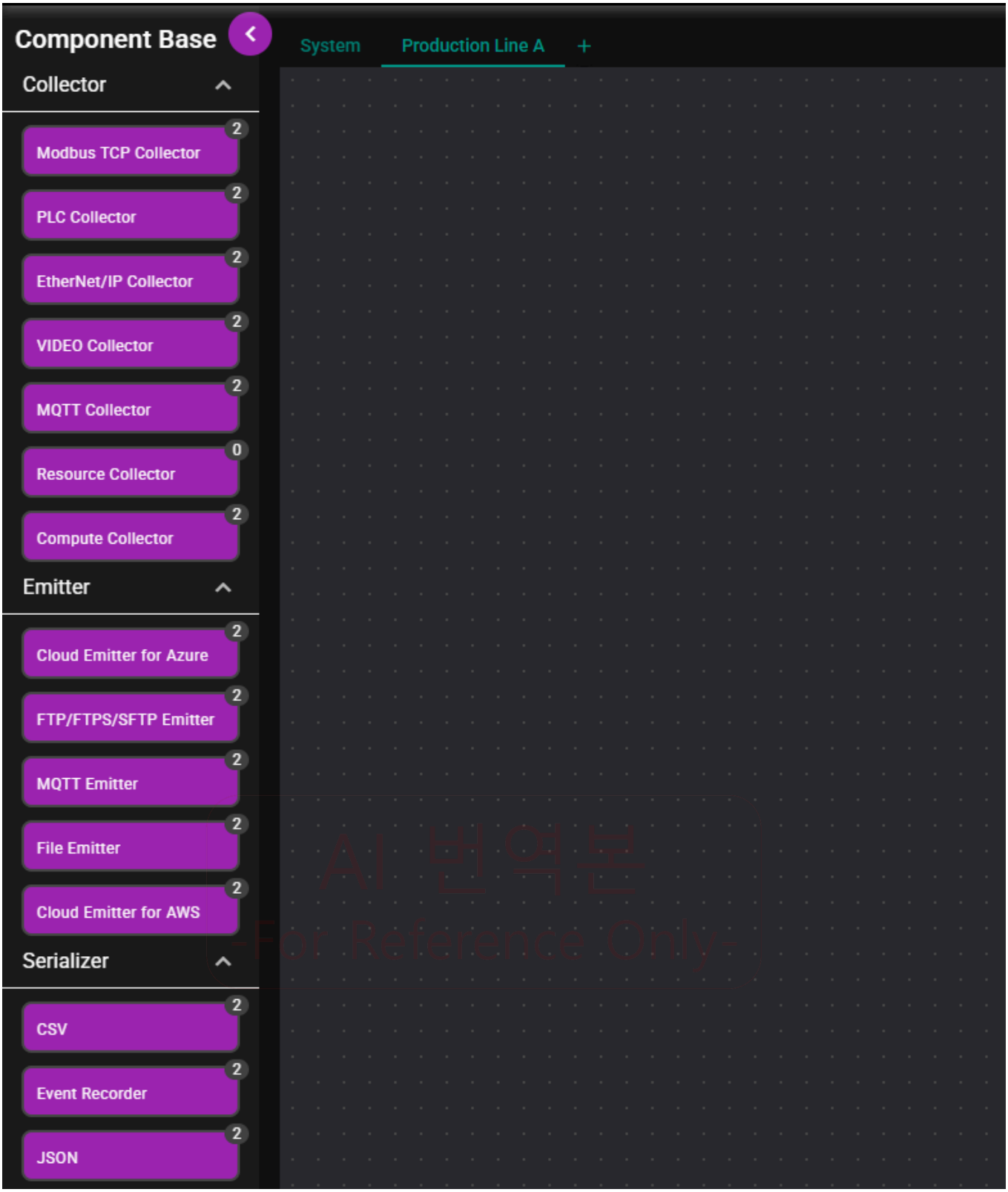
AI 번역본
For Reference Only

6. JSON의 설정값을 입력하고,「 저장 」을 클릭합니다.

AI 번역본
-For Reference Only-


SAVE
DELETE
CANCEL
Name
JSON
Autostart disable
Format
Indent
2
Time zone
UTC
Change unit
Time
Change time(sec)
60
Max size(KB)
96
Default float digits
3
Output with total nano time
emit data continued from last time on restart.
Fix column name
Fix column value
AI 번역본
-For Reference Only-
No data available
ADD

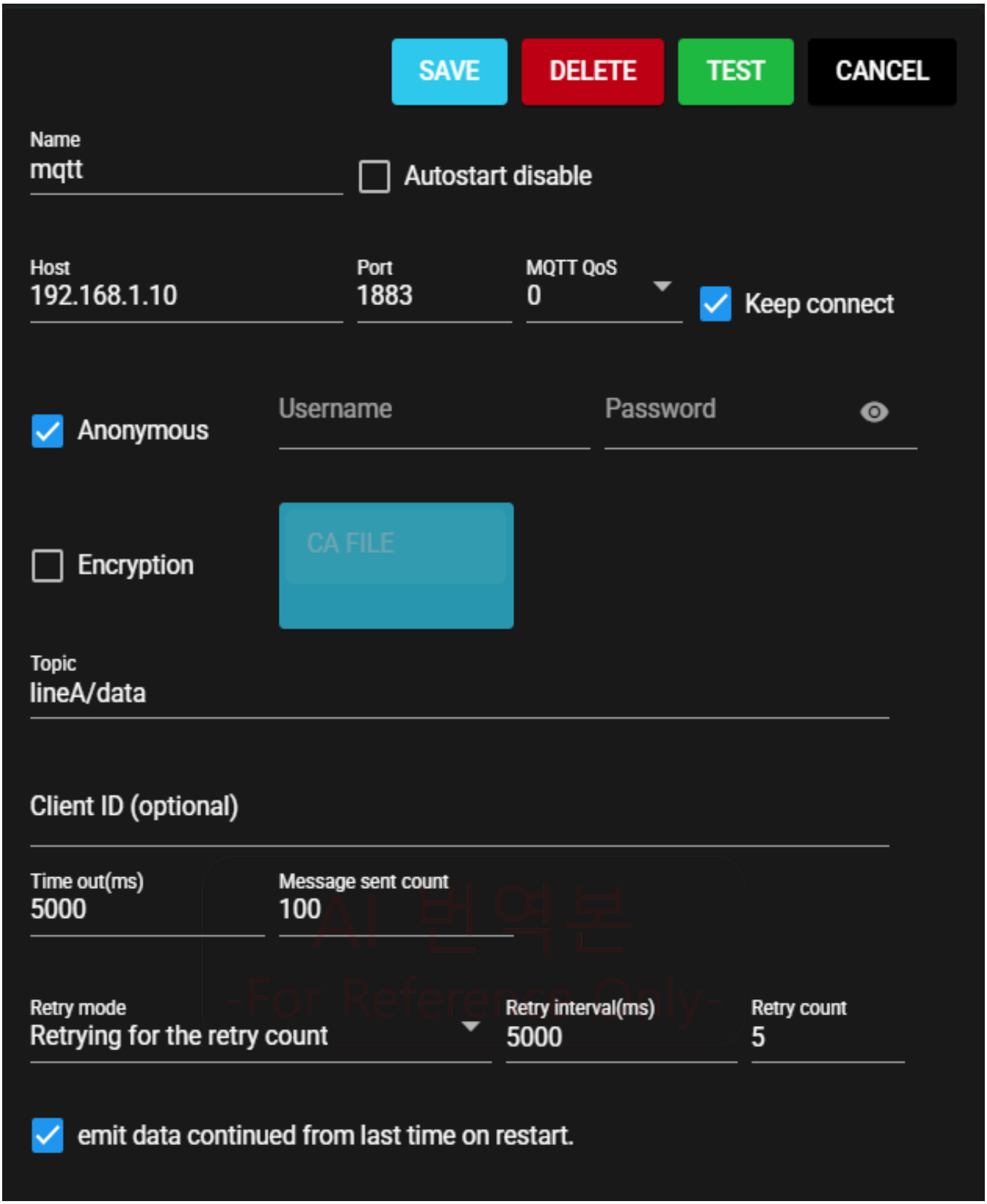
7. MQTT 에미터를 설정합니다. 플로우 에디터 좌부에서 「 MQTT 에미터 」를 플로우 에디터 중앙으로 드래그 & 드롭합니다.



AI 번역본
-For Reference Only-

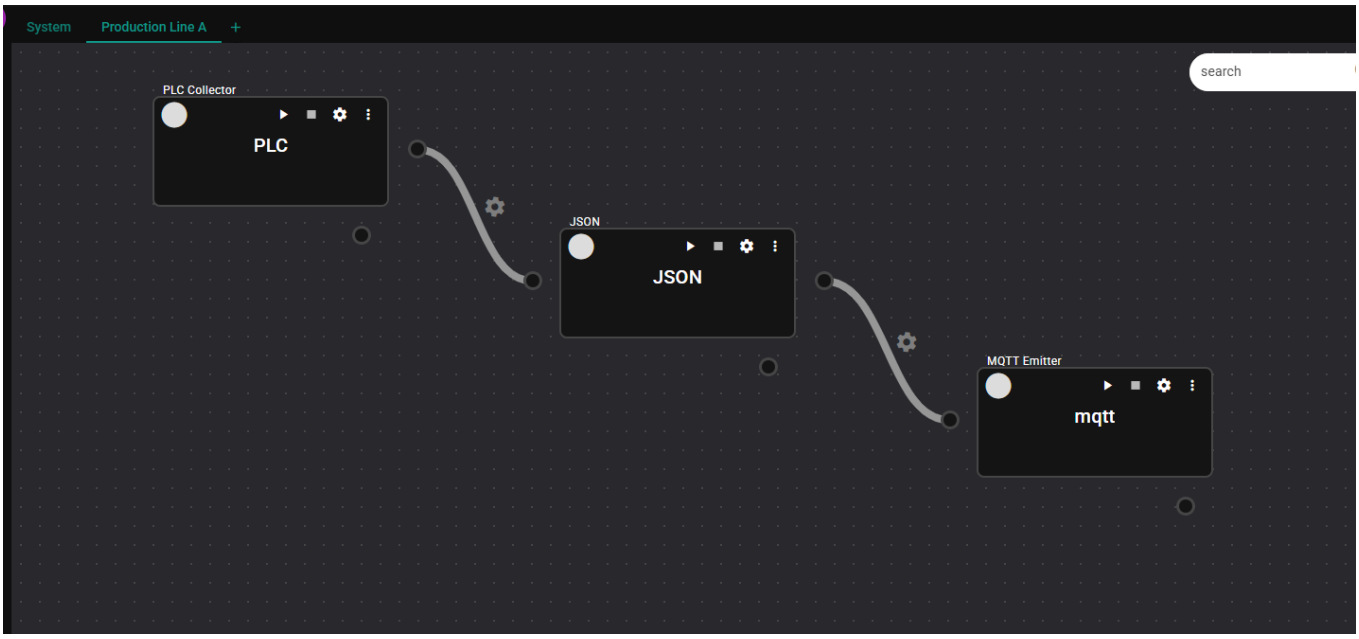
8. MQTT 에미터의 설정값을 입력하고,「 저장 」을 클릭합니다. IP 주소나 포트 등은 이용하시는 환경에 맞춰 주십시오.

(※만약 MQTT 에미터를 이용할 수 없는 경우, File 에미터 등으로 대체하여 설정하십시오.)



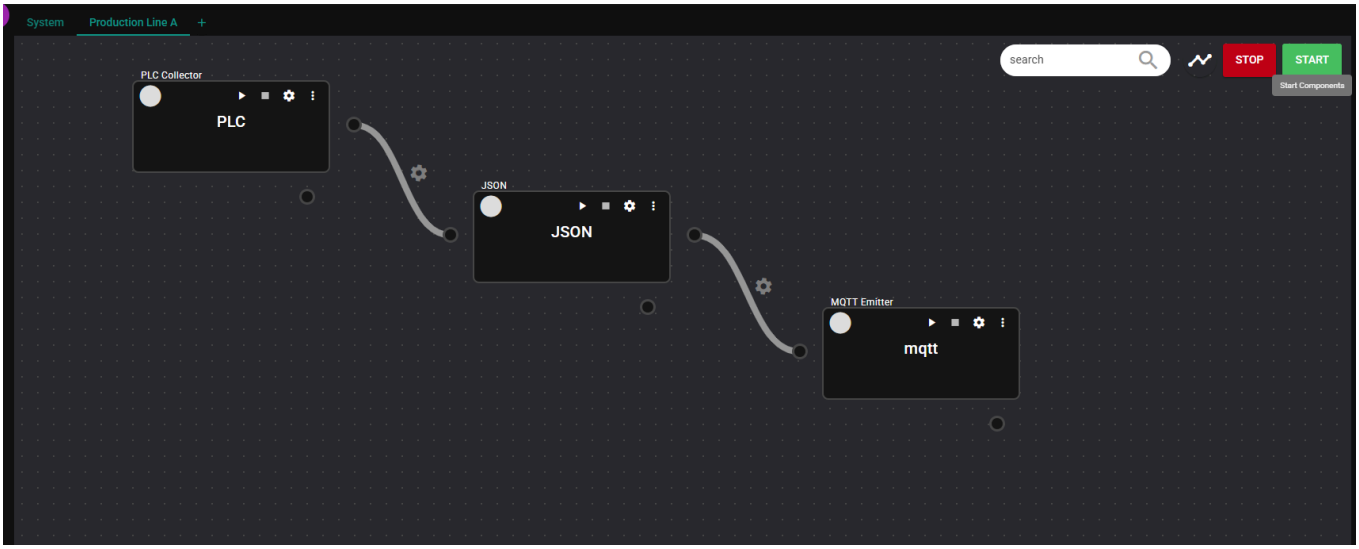
AI 번역본
-For Reference Only-

9. 플로우 링크를 연결합니다.



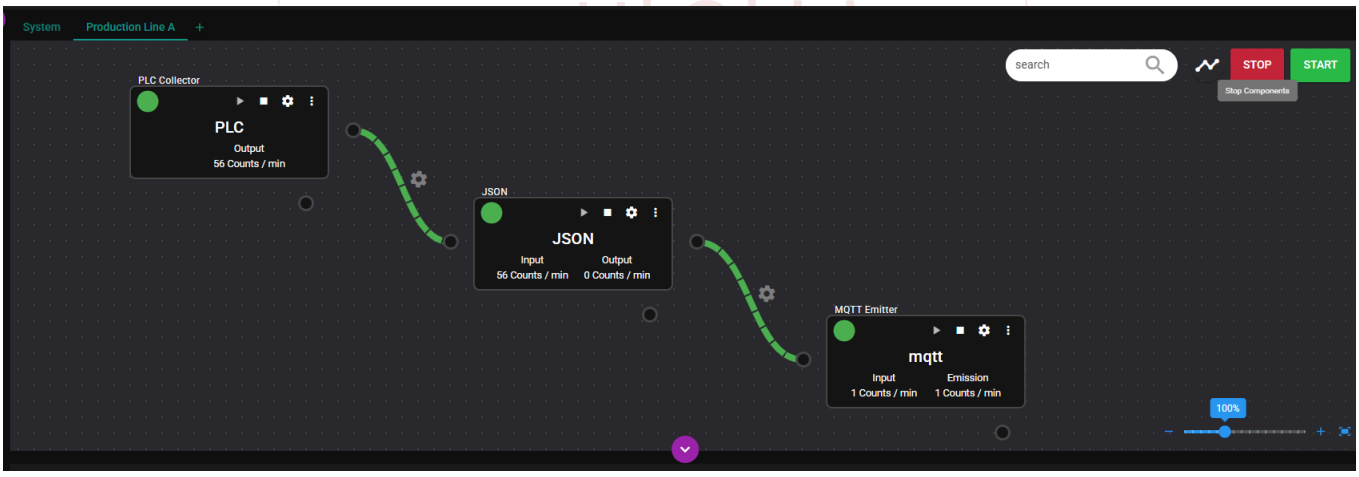
AI 번역본
-For Reference Only-

10. 컴포넌트를 동작시킵니다. 각각의 컴포넌트가 정상적으로 동작하는 것을 확인하십시오. 일정 시간 동작시켜, 이상이 발생하지 않는지 등을 확인합니다.





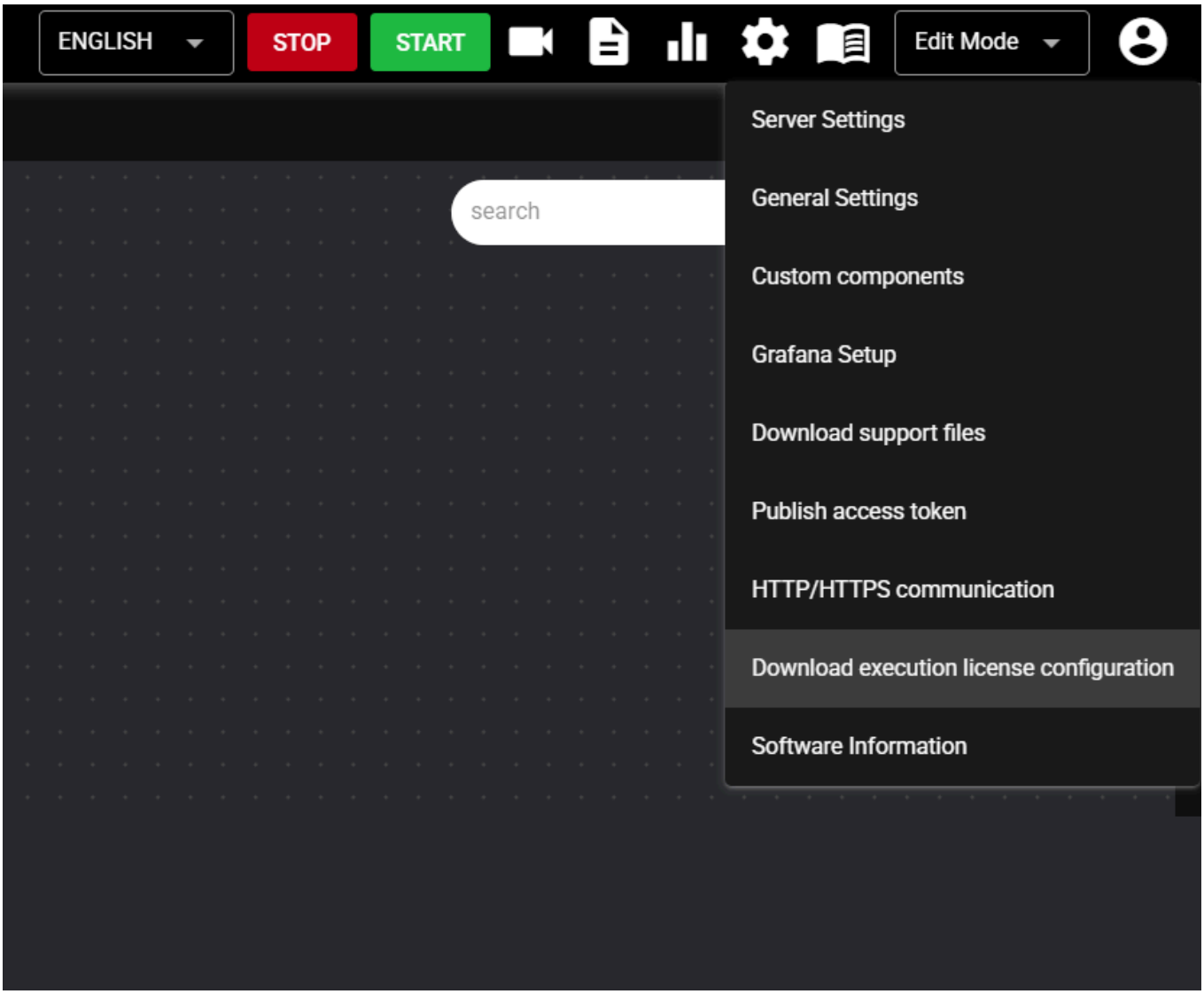
11. 모두 정상적으로 동작하는 것을 확인했으면, 컴포넌트를 정지하십시오.



10.3.2.2 실행 라이선스 설정의 출력

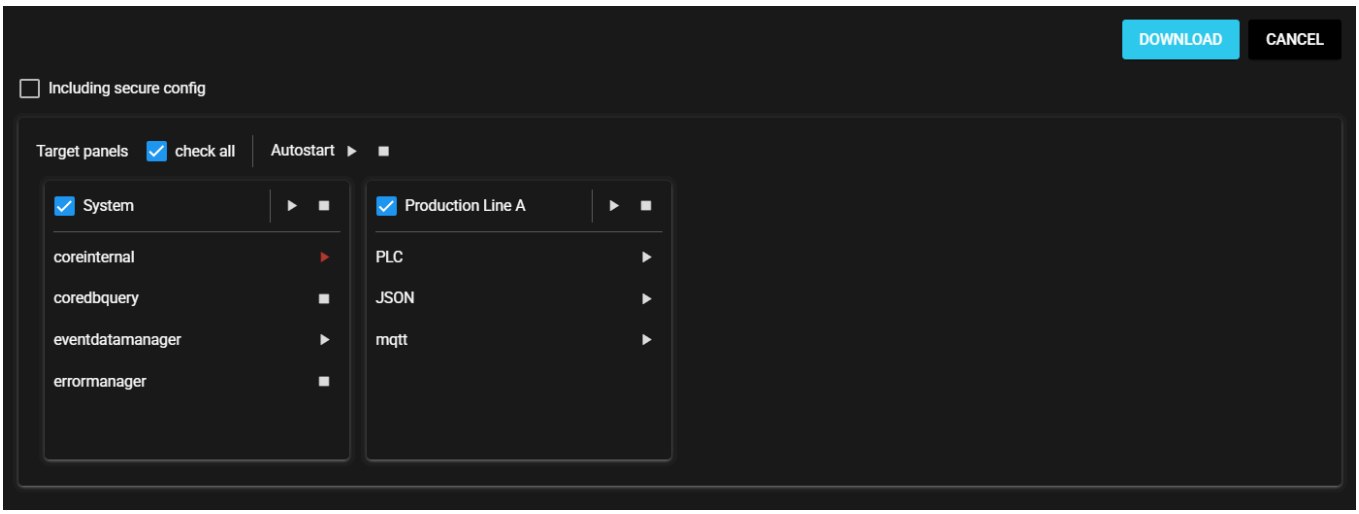
개발 라이선스 가동 환경에서의 설정 구축이 끝나면, 실행 라이선스용 설정의 출력을 수행합니다.

1. 설정 메뉴의 「 실행 라이선스 설정 다운로드 」를 클릭합니다. 이 파일을 실행 라이선스 가동 환경에서 사용합니다.



2. 확인 다이얼로그에서, 「 다운로드 」를 클릭합니다. 기밀 정보를 포함하는 경우는, 「 다운로드 」를 클릭 전에, 「 기밀 정보를 포함 」을 체크하십시오. 선택한 패널만 포함하는 경우는, 포함할 패널을 체크하십시오. 또한, 초기의 자동 기동 설정을 지정할 수 있습니다. ▶（ 자동 기동 ）■（ 자동 기동 무효 ）

AI 번역본
-For Reference Only-



> ⚠ 기밀 정보에 대해서
>
> 기밀 정보（예를 들어, Azure의 접속 문자열이나 MQTT 등의 사용자명, 패스워드 등）를 포함하여, 실행 라이선스 설정 다운로드를 수행하면, 출력 파일에 기밀 정보가 포함되어 있음에 주의하십시오.
> 또한, 각 컴포넌트에 설정한 증명서의 파일 및 커스텀 컴포넌트의 첨부 파일도 기밀 정보의 취급이 됩니다. 기밀 정보를 포함하지 않는 경우는, 이것들이 출력되지 않으므로, 실행 라이선스 가동 머신 측에서, 기밀 정보를 개별로 설정할 필요가 있습니다.

3. 다음과 같은 파일이 다운로드됩니다. 파일명의 형식은,「execution_license_package_서기_월_일.pkg」가 됩니다.

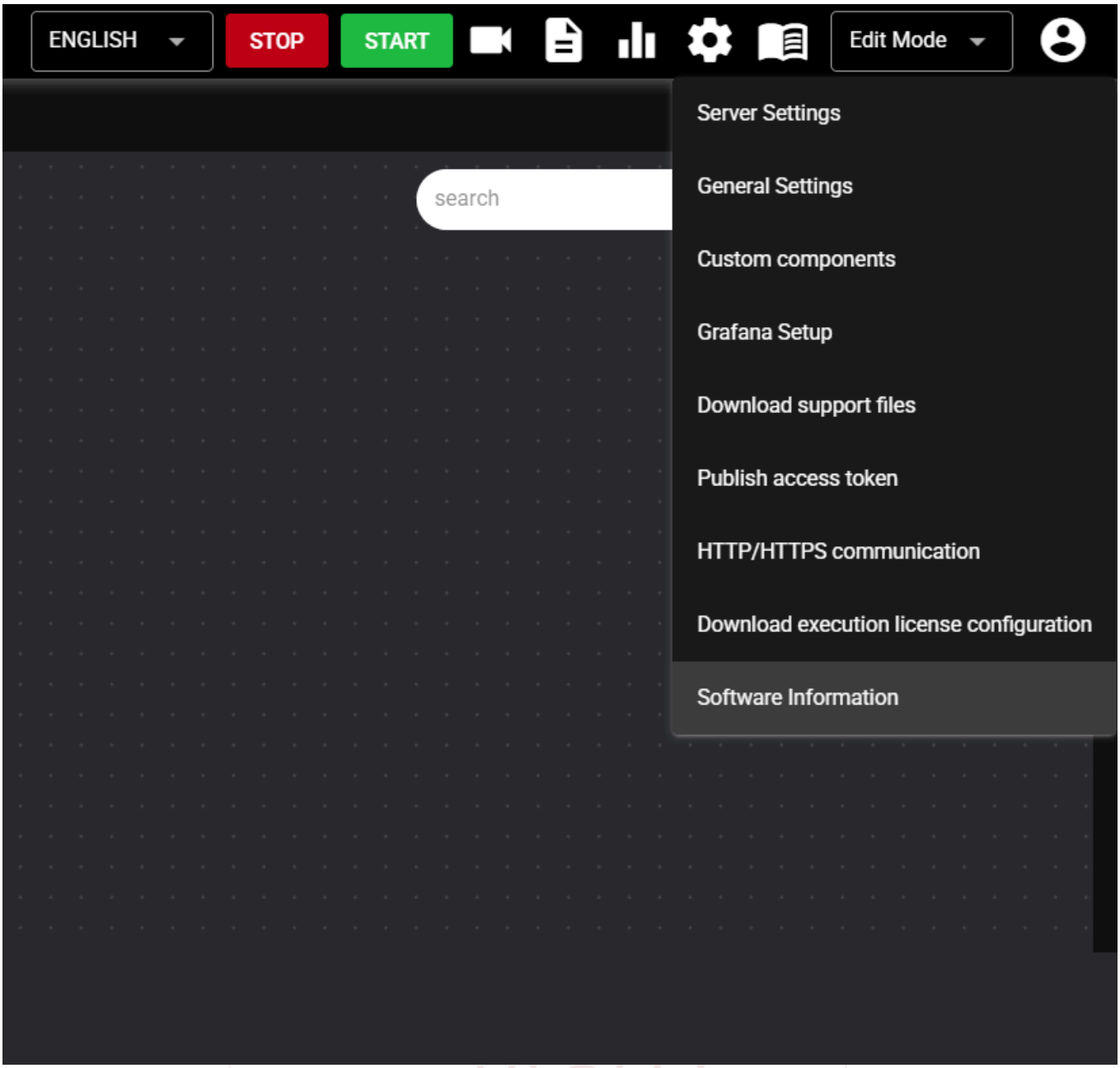


10.3.2.3 실행 라이선스 가동 환경의 셋업

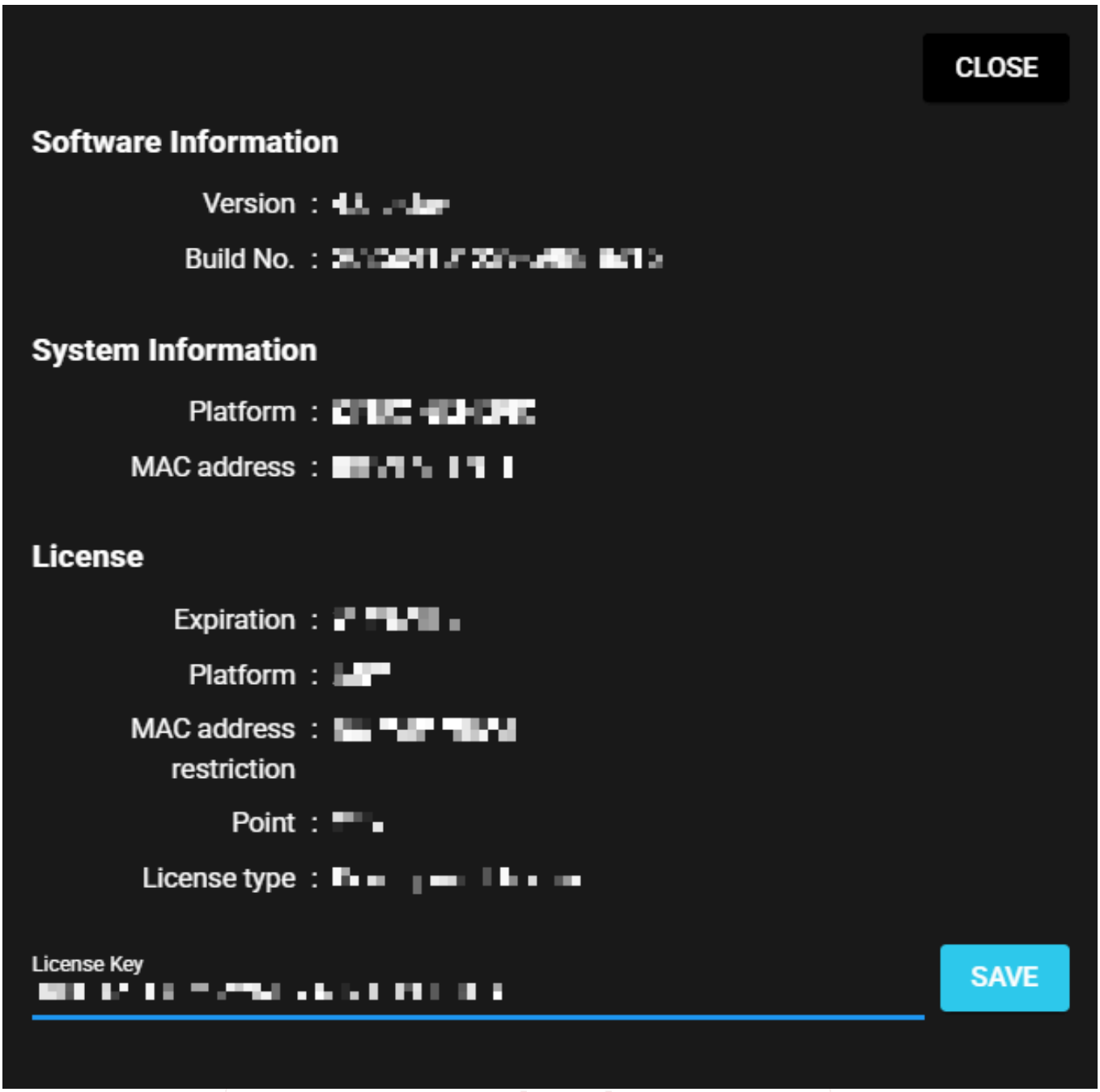
실행 라이선스 가동 환경에서의 설정을 수행합니다. SpeeDBee Synapse의「Edit Mode」에서 조작을 수행합니다.

AI 번역본
-For Reference Only-

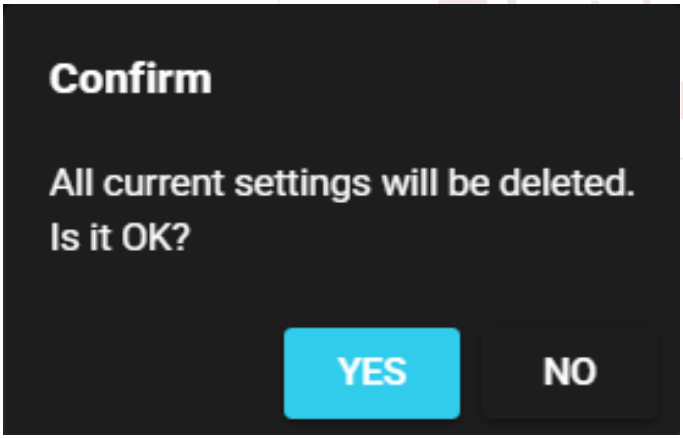
1. 설정 메뉴의 「 소프트웨어 정보 」에서 라이선스 키를 입력하여 저장합니다.



AI 번역본
-For Reference Only-



2. 메시지를 확인하여 문제없으면,「 예 」를 클릭합니다.

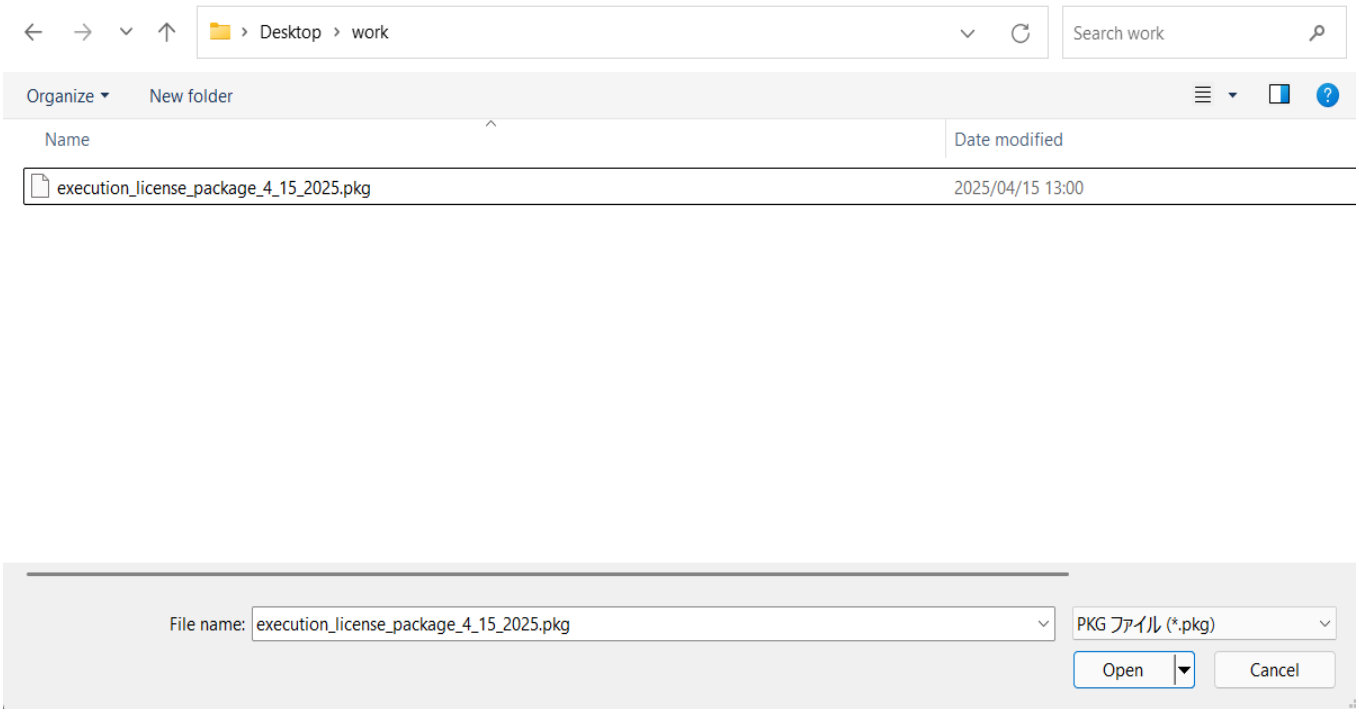


AI 버전 본
-Reference Only-

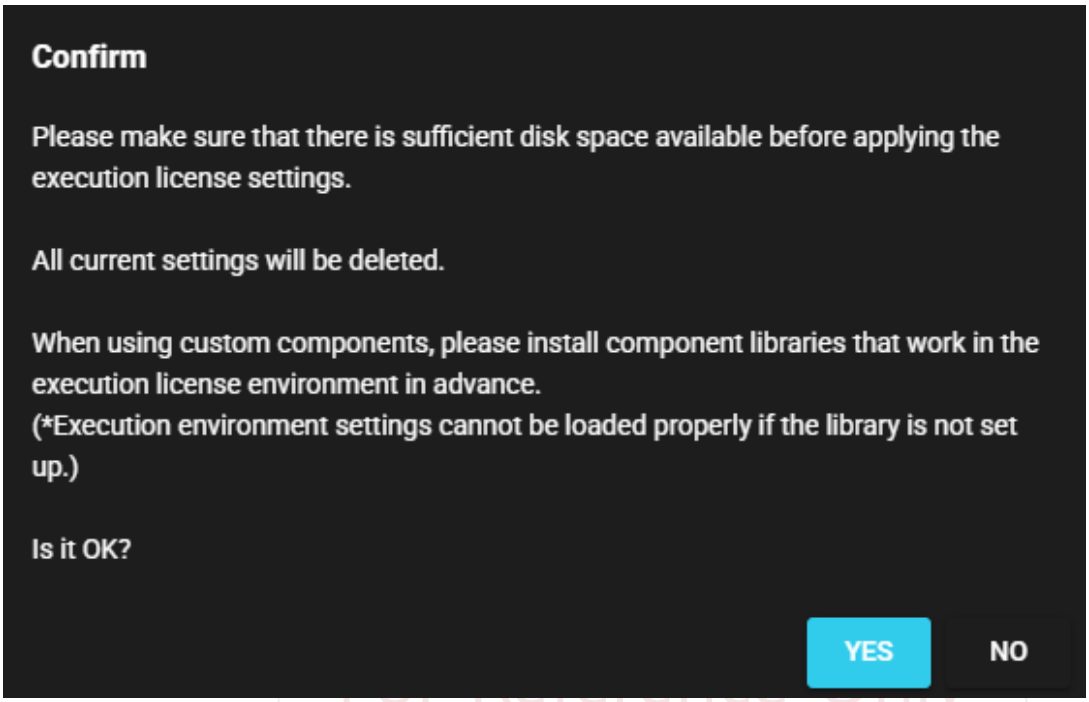
3. 실행 라이선스용 설정 파일을 임포트합니다.「 임포트 」를 클릭합니다.



4. 실행 라이선스용 설정 파일을 선택합니다.

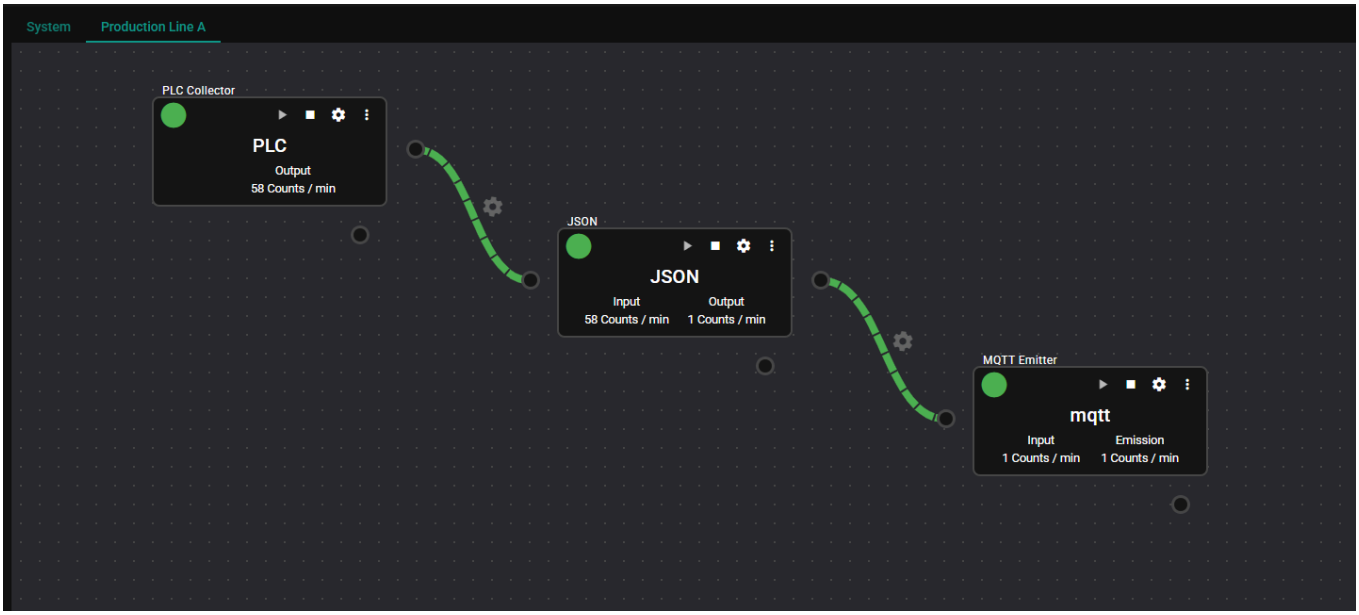


5. 메시지를 확인하여 문제없으면,「 예 」를 클릭합니다.



For Reference Only

6. 일정 시간 경과 후에, 컴포넌트가 기동합니다.
(※표시의 확인은,「 소프트웨어 정보 」를 닫아서 확인합니다.)

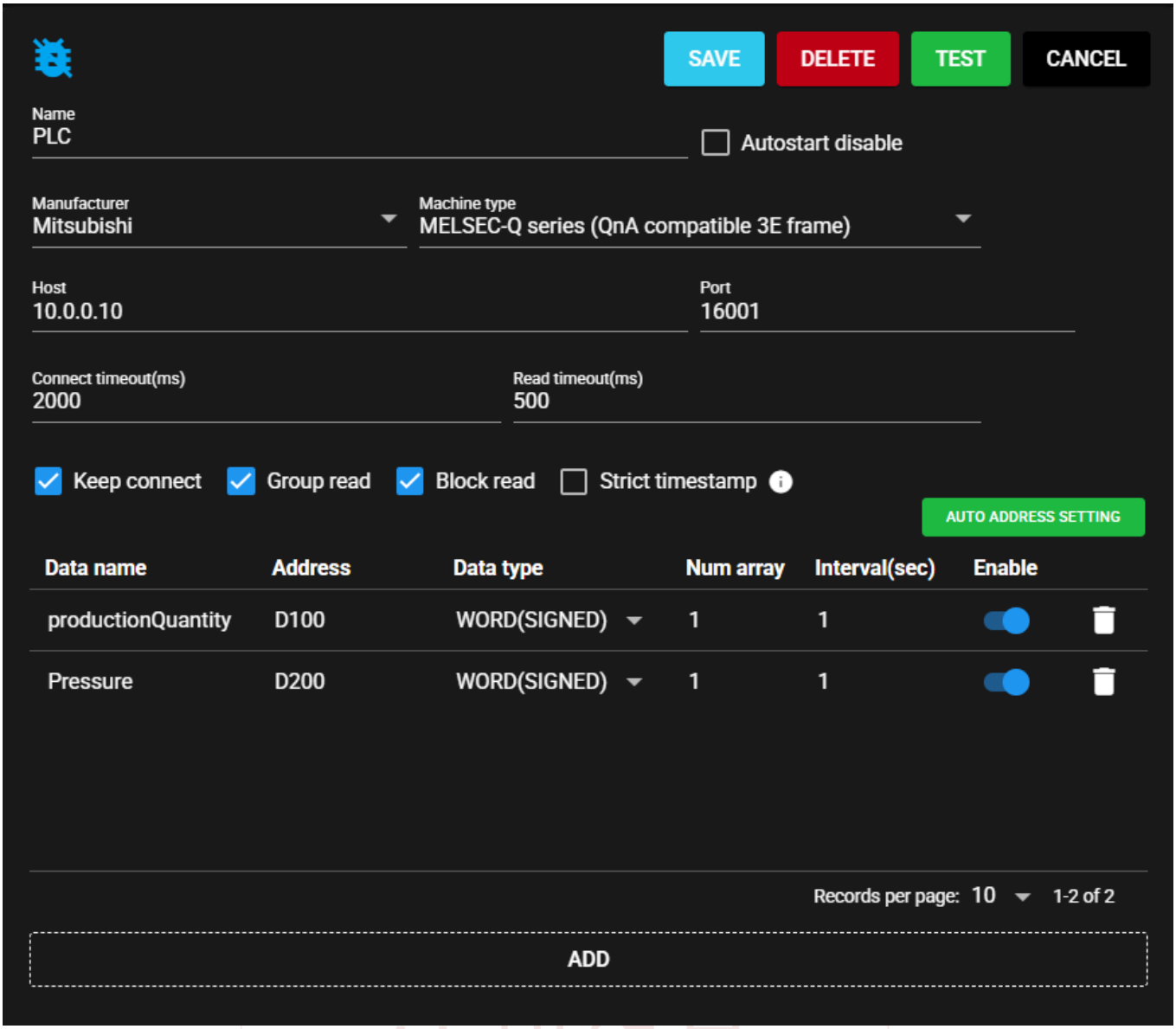


⚠ 기밀 정보를 갖는 컴포넌트에 대해서

기밀 정보를 포함하지 않은 실행 라이선스용 설정을 임포트한 경우, 그것들의 정보가 포함되어 있지 않으므로, 기밀 정보의 설정을 필요로 하는 컴포넌트는, 자동 기동이 무효가 됩니다. 기밀 정보나 첨부 파일에 대해서는, 개별로 재설정하십시오.

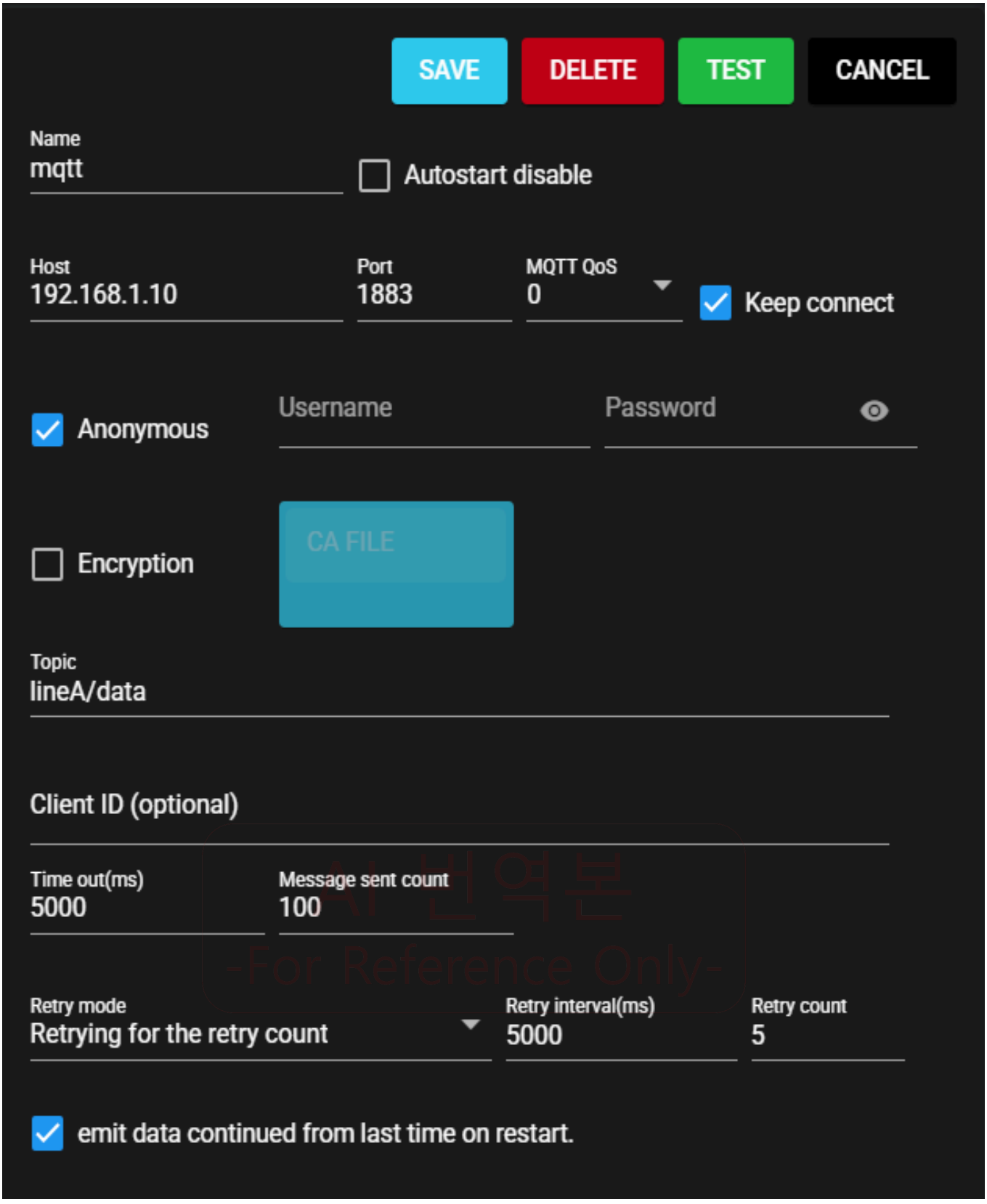
AI 번역본
-For Reference Only-

7. PLC 컬렉터의 설정을 필요에 따라 변경합니다. 개발 라이선스 가동 머신과 실행 라이선스 가동 머신에서는, 접속처가 다른 경우 등의 설정을 수행합니다.
(※튜토리얼에서는, 동일한 네트워크 환경을 전제로 하고 있으므로, 개발 라이선스 측을 정지하고 있으면 변경할 필요는 없습니다.)



AI 번역본
-For Reference Only-

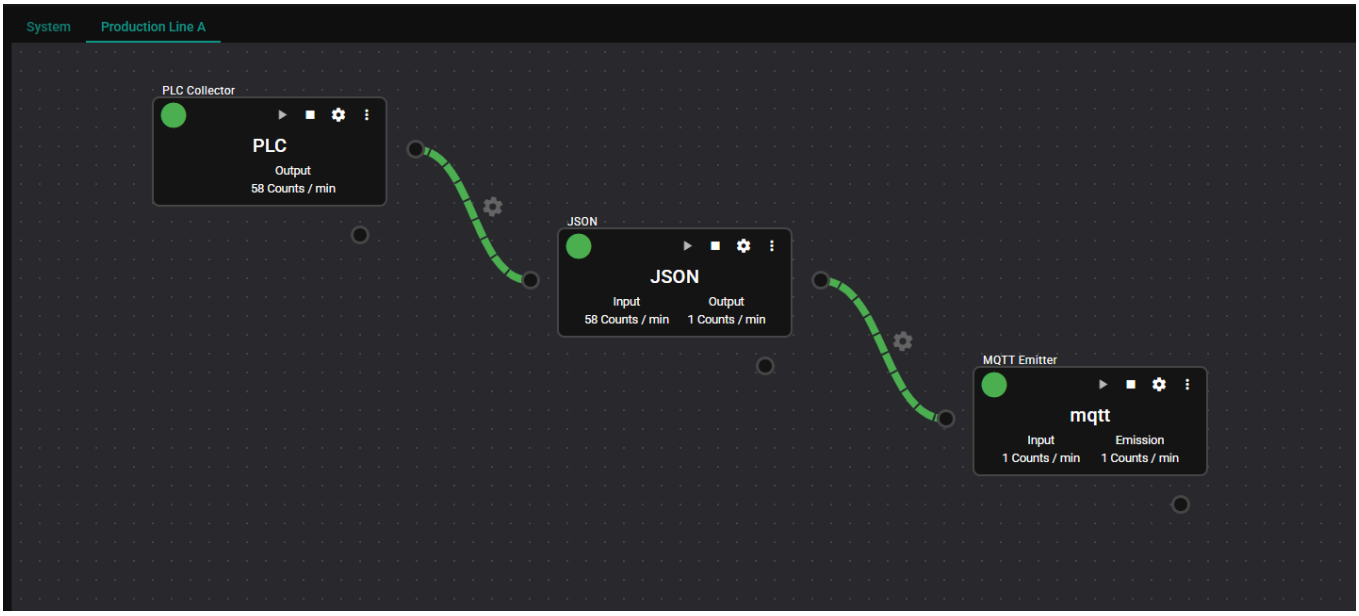
8. MQTT 컬렉터의 설정을 필요에 따라 변경합니다. 개발 라이선스 가동 머신과 실행 라이선스 가동 머신에서는, MQTT 브로커에 대한 접속 정보가 다른 경우 등의 설정을 수행합니다.
(※튜토리얼에서는, 동일한 네트워크 환경을 전제로 하고 있으므로, 개발 라이선스 측을 정지하고 있으면 변경할 필요는 없습니다.)



10.3.2.4 동작 확인

앞 항목까지의 설정으로, 올바르게 동작하는 것을 확인합니다.

1. 컴포넌트의 램프를 확인합니다. 경고나 에러가 없음을 확인합니다. 문제가 있으면 설정의 재검토를 수행합니다.



10.3.2.5 운용・모니터링・가시화

앞 항목까지의 설정으로, 정상적으로 가동할 수 있음이 확인되면 운용에 들어갑니다. 모니터링・가시화에 대해서는, 도입 플로우를 참조하십시오.
실행 라이선스의 운용 튜토리얼은 이상입니다.

10.4 실행 라이선스 가동 환경에 관한 주의사항

실행 라이선스 가동 환경에 관한 주의사항을 다음에 나타냅니다.

AI 번역본
-For Reference Only-

주의사항

- 실행 라이선스를 적용하면, 실행 라이선스에 따른 동작 모드가 되어, 사용할 수 있는 기능이 제한됩니다. (참고: 실행 라이선스 이용 환경의 동작 사양)
- 실행 라이선스용 설정 가져오기를 수행하면, 선택 중인 프로젝트의 그 이전 설정은 삭제됩니다. (커스텀 컴포넌트나 외부 모듈은 삭제되지 않습니다)
- 실행 라이선스로 전환될 때, 선택되지 않은 프로젝트는 모두 삭제됩니다. 남겨 두고 싶은 프로젝트는 사전에 익스포트해 두십시오. (참고: 익스포트)
- 액세스 토큰에 대해서는, 개발 라이선스 측의 설정을 계승하지 않습니다. 또한, 실행 라이선스 측에서 생성한 액세스 토큰은, 실행 라이선스용 설정 가져오기를 수행해도 삭제되지 않습니다.
- DB 쿼리 서비스의 설정은, 실행 라이선스 측으로 계승됩니다. 그러나, 토큰 인증에 사용하는 액세스 토큰은 계승되지 않으므로, 실행 라이선스 측에서 설정을 수행할 필요가 있습니다.
- 실행 라이선스용 설정에서, 커스텀 컴포넌트나 Python의 외부 모듈을 사용하고 있는 경우는, 실행 라이선스용 설정 가져오기 전에, 그것들을 사전에 설정해 두어야 합니다.
- 실행 라이선스용 설정 가져오기 전에, 가동 머신의 스토리지 용량의 여유가 충분한지 등을 확인하십시오. 스토리지 용량이 부족한 경우는, 실행 라이선스용 설정 가져오기에 실패합니다.
- 실행 라이선스 가동 환경에서, 사용자 관리를 수행하는 경우는, 각각의 가동 환경마다 사용자 관리를 수행할 필요가 있습니다. 개발 라이선스 측의 설정은 계승되지 않습니다.

AI 번역본
-For Reference Only-

11 i-Trigger (확장 패키지)

AI 번역본
-For Reference Only-

11.1 i-Trigger（확장 패키지）

여기에서는 SpeeDBee Synapse의 확장 패키지인 i-Trigger에 대해 설명합니다. 기재 내용은

다음과 같습니다.

1. i-Trigger의 개요
2. 이용할 카메라에 대해
3. 설치
4. 동영상 재생용 뷰어
5. 컴포넌트
6. 튜토리얼
7. i-Trigger의 기록 사양

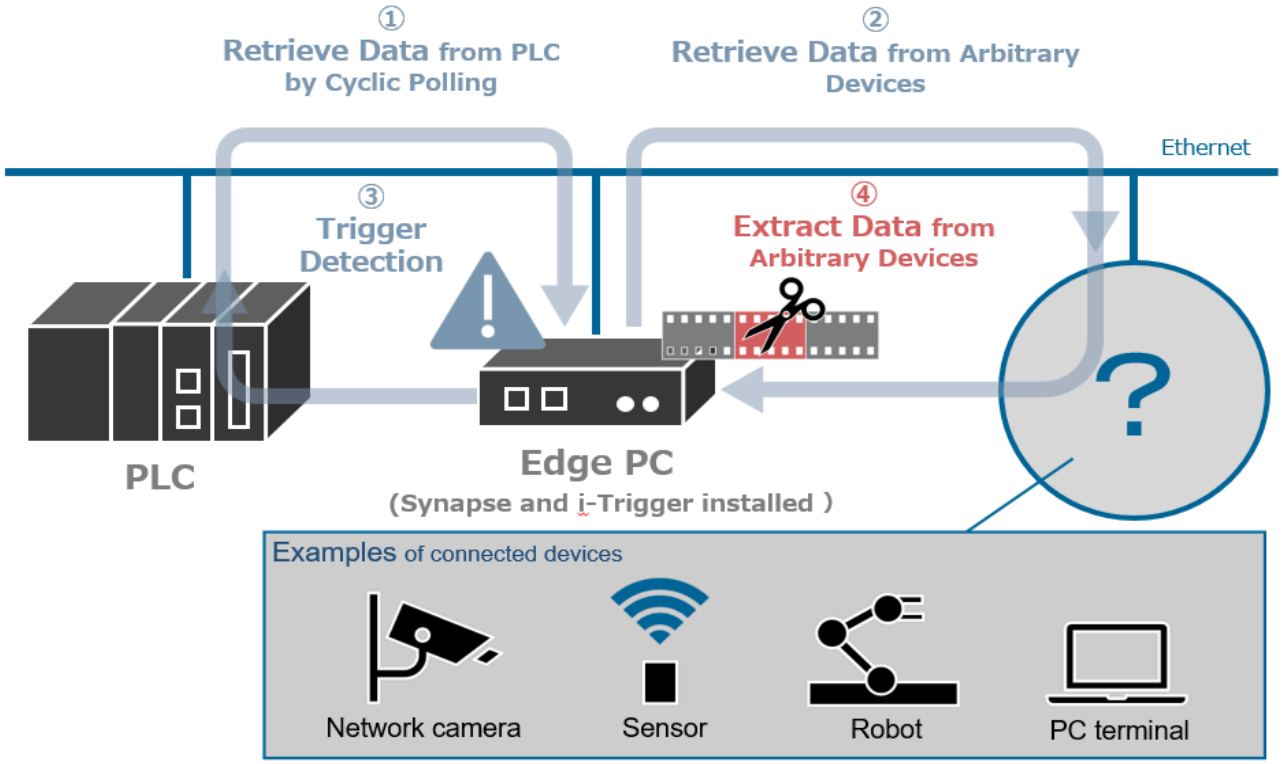
AI 번역본
-For Reference Only-

11.2 개요

여기에서는 i-Trigger의 개요에 대해 설명합니다.

11.2.1 i-Trigger의 특장

i-Trigger（아이트리거）는, 제조업에서의 순간 정지를 감시하는 SpeeDBee Synapse의 확장 패키지입니다.
설비에서의 드라이브 레코더와 같은 기능을 제공합니다. 제조 라인에 이상이 발생했을 때에, 각종 컴포넌트가 수집한 데이터에서, 이상 발생 전후의 수치 데이터, 카메라로 촬영한 동영상 데이터를 자동적으로 잘라내어 지정된 파일 서식으로 저장합니다.



AI 번역본
-For Reference Only-

처리 개요

① 주기 참조로 PLC에서 데이터 수집

트리거 검지 및 기록 대상이 되는 데이터를 PLC에서 취득합니다.

②임의 디바이스에서 데이터 취득

PLC뿐만 아니라, 각 컴포넌트에서 수집한 데이터도 PLC와 마찬가지로 트리거 검지 등에 이용할 수 있습니다.

③트리거 검지

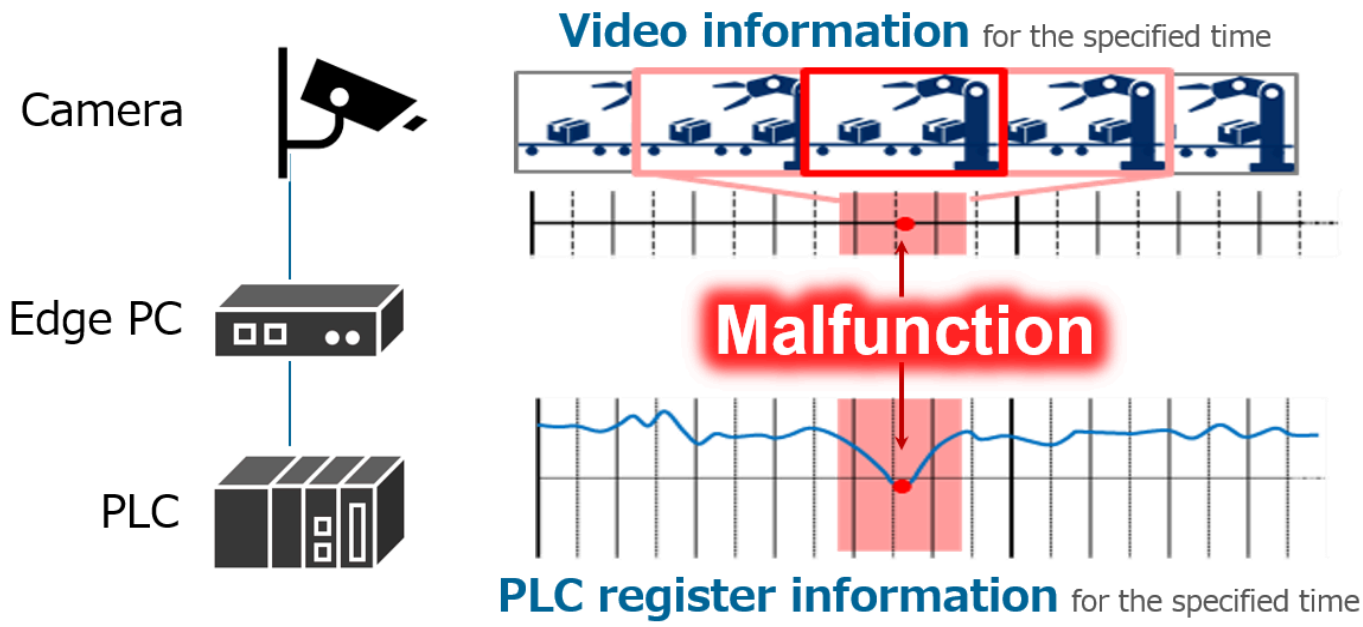
수집한 데이터가, 사전 설정한 조건에 합치하면 트리거 검지로서 잘라내기 처리가 수행됩니다.

④임의 디바이스 데이터의 잘라내기

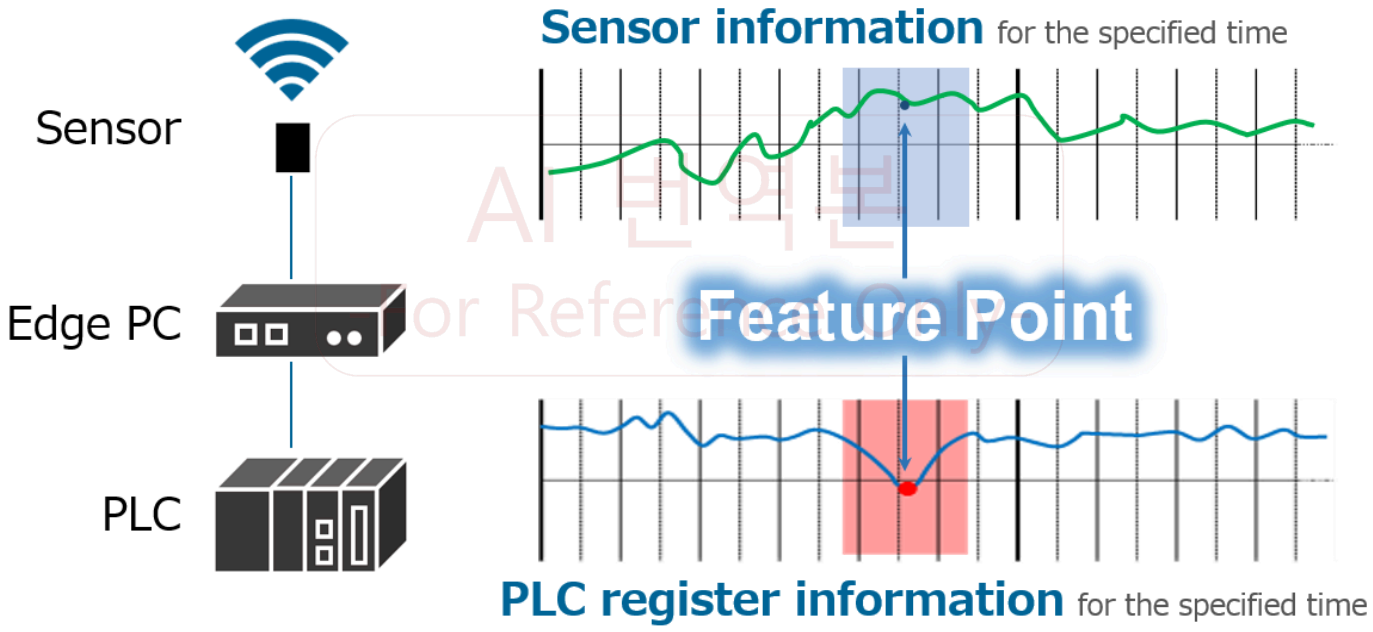
트리거 검지 후에, 사전 설정한 기록 범위에 따라, 대상 디바이스（네트워크 카메라나 PLC의 수집 데이터）
의 데이터를 잘라냅니다. 잘라낸 데이터는 전용 뷰어로 참조할 수 있습니다.

전용 뷰어

i-Trigger에는, 잘라낸 데이터를 열람하기 위한 전용 뷰어가 탑재되어 있습니다. 아래 그림에서 나타내는 바와 같이, 장애 발생 시의 동영상과 수집 데이터를 동시 재생할 수 있는 특장이 있습니다.



동영상과 수집한 다양한 데이터를 같은 시각의 것으로 가시화함으로써, 데이터와 사상의 관련성 해석에 도움이 됩니다.



11.2.2 i-Trigger의 이용에 앞서

i-Trigger는, SpeeDBee Synapse의 확장 패키지이기 때문에, 이용에는 SpeeDBee Synapse의 본체가 필요합니다. 그 때문에, SpeeDBee Synapse와 동일한 환경에서 동작합니다. 상세는, 실행 환경을 참조하십시오.

AI 번역본
-For Reference Only-

11.3 사용할 수 있는 카메라에 대해

본 절에서는, i-Trigger에서 사용할 수 있는 카메라에 대해 설명합니다.

11.3.1 사용할 수 있는 카메라 사양・대응 코덱

i-Trigger에서 사용할 수 있는 카메라는, 다음과 같습니다.

형태	대응하고 있는 비디오 코덱	동작 확인 완료 프로파일
RTSP（Real-Time Streaming Protocol）대응의 네트워크 카메라	H.264	• Baseline Profile • Main Profile • High Profile

11.3.2 카메라의 이용 대수

i-Trigger에서, 동시에 이용할 수 있는 카메라에는 대수의 제한이 있습니다. SpeeDBee Synapse의 라이선스에, 카메라 대수가 부여되어 있으며, 그 대수를 초과하여 카메라를 동시에 녹화할 수는 없습니다.

항목	대수
표준으로 사용할 수 있는 대수	4대
최대로 이용할 수 있는 대수	30대 ※사용하려면, 대수에 따른 라이선스가 필요

AI 번역본

필요한 카메라 대수가 부여된 라이선스를 구입하십시오.

Data Flow Controller판, OMRON-SBS for Windows판에서는, 초기 상태의 라이선스로 접속 가능한 카메라 대수가 30대로 고정되어 있습니다.

11.4 설치

본 절에서는, 설치 순서 및 제거 순서에 대해 설명합니다.

11.4.1 설치 및 업데이트

아래 순서로, i-Trigger를 설치할 수 있습니다. 또한, SpeeDBee Synapse와 i-Trigger는, 반드시 같은 버전을 이용하십시오. 업데이트 시에도 동일한 순서가 됩니다.
Windows에서는, SpeeDBee Synapse의 설치 시에, 동시에 설치합니다.「설치（Windows）」를 참조하십시오.

> Data Flow Controller판에서는, 초기 상태로 i-Trigger가 설치되어 있기 때문에, 설치는 불필요합니다.

파일을 배치

1. i-Trigger의 파일을, 설치 대상 디바이스 내（/tmp 디렉터리）에 배치합니다. 아래는, Windows PowerShell에서 scp 커맨드로 배치하는 예입니다.

AI 번역본
-For Reference Only-

디바이스의 OS가 Ubuntu20(amd64)인 경우

```
PS >scp speedbeesynapse_ubuntu20_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주소):/tmp
PS >scp speedbeesynapse-itrigger_ubuntu20_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주
소):/tmp
```

※동일 버전의 SpeeDBee Synapse의 설치가 완료된 경우는, SpeeDBee Synapse의 전송은 불필요합니다.

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(amd64)인 경우

```
PS >scp speedbeesynapse_ubuntu22_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주소):/tmp
PS >scp speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_amd64.deb (계정명)@(디바이스의 IP 주
소):/tmp
```

※동일 버전의 SpeeDBee Synapse의 설치가 완료된 경우는, SpeeDBee Synapse의 전송은 불필요합니다.

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(arm64)인 경우

```
PS >scp speedbeesynapse_ubuntu22_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmp
PS >scp speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주
소):/tmp
```

※동일 버전의 SpeeDBee Synapse의 설치가 완료된 경우는, SpeeDBee Synapse의 전송은 불필요합니다.

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Raspberry Pi OS(bookworm 64-bit)인 경우

```
PS >scp speedbeesynapse_raspi_bookworm_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주
소):/tmp
PS >scp speedbeesynapse-itrigger_raspi_bookworm_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP
주소):/tmp
```

For Reference Only

※동일 버전의 SpeeDBee Synapse의 설치가 완료된 경우는, SpeeDBee Synapse의 전송은 불필요합니다.

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 debian12(bookworm arm64)인 경우

```
PS >scp speedbeesynapse_debian12_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmpPS >scp
speedbeesynapse-itrigger_debian12_x.x.x-r0_arm64.deb (계정명)@(디바이스의 IP 주소):/tmp
```

※동일 버전의 SpeeDBee Synapse의 설치가 완료된 경우는, SpeeDBee Synapse의 전송은 불필요합니다.

※x.x.x 부분에는, 버전 번호가 들어갑니다.

설치 및 기동

1. ssh를 사용하여 디바이스에 로그인합니다. 아래는 Windows PowerShell에서 ssh 커맨드로 로그인하는 예입니다.

```
PS >ssh (계정명)@(디바이스의 IP 주소)
```

2. 아래 커맨드를 실행하여, i-Trigger를 설치합니다.

AI 번역본
-For Reference Only-

디바이스의 OS가 Ubuntu20(amd64)인 경우

- SpeeDBee Synapse를 동시 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse_x.x.x-r0_amd64.deb /tmp/speedbeesynapse-itrigger_ubuntu20_x.x.x-r0_amd64.deb
```

- i-Trigger만 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse-itrigger_ubuntu20_x.x.x-r0_amd64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(amd64)인 경우

- SpeeDBee Synapse를 동시 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse_ubuntu22_x.x.x-r0_amd64.deb /tmp/speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_amd64.deb
```

- i-Trigger만 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_amd64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Ubuntu22(arm64)인 경우

- SpeeDBee Synapse를 동시 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse_ubuntu22_x.x.x-r0_arm64.deb /tmp/speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_arm64.deb
```

- i-Trigger만 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse-itrigger_ubuntu22_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 Raspberry Pi OS(bookworm 64-bit)인 경우

- SpeeDBee Synapse를 동시 설치 (업데이트) 하는 경우

```
$ sudo apt install /tmp/speedbeesynapse_raspi_bookworm_x.x.x-r0_arm64.deb /tmp/speedbeesynapse-itrigger_raspi_bookworm_x.x.x-r0_arm64.deb
```

- i-Trigger만 설치 (업데이트) 하는 경우

사내 버전 보
For Reference Only-

```
$ sudo apt install /tmp/speedbeesynapse-itrigger_raspi_bookworm_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

디바이스의 OS가 debian12(bookworm arm64)인 경우

- SpeeDBee Synapse를 동시 설치（업데이트）하는 경우

```
$ sudo apt install /tmp/speedbeesynapse_debian12_x.x.x-r0_arm64.deb /tmp/speedbeesynapse-itrigger_debian12_x.x.x-r0_arm64.deb
```

- i-Trigger만 설치（업데이트）하는 경우

```
$ sudo apt install /tmp/speedbeesynapse-itrigger_debian12_x.x.x-r0_arm64.deb
```

※x.x.x 부분에는, 버전 번호가 들어갑니다.

위 커맨드에 의해 SpeeDBee Synapse 본체의 서비스가 재기동합니다.

11.4.2 제거

아래 순서로, 디바이스에서 i-Trigger를 제거할 수 있습니다. Windows에 대해서는,「제거（Windows）」를 참조하십시오.

1. ssh를 사용하여 디바이스에 로그인합니다. 아래는 Windows PowerShell에서 ssh 커맨드로 로그인하는 예입니다.

```
PS >ssh (계정명)@(디바이스의 IP 주소)
```

2. 아래 커맨드를 실행하여, i-Trigger를 제거합니다.

```
$ sudo apt remove speedbeesynapse-itrigger
```

또는

```
$ sudo apt purge speedbeesynapse-itrigger
```

위 어느 커맨드에서도, 제거 시의 동작에 차이는 없습니다.

위 커맨드에 의해 SpeeDBee Synapse 본체의 서비스가 정지합니다. 다시 기동하는 경우는, SpeeDBee Synapse의 제어를 참고로 서비스를 기동하십시오.

AI 번역본

제거 시의 주의

i-Trigger를 제거한 상태에서, SpeeDBee Synapse의 서비스를 기동하면, i-Trigger 관련（VIDEO 컬렉터）등의 설정이 남아 있던 경우, 그러한 설정은 삭제됩니다.

다시 i-Trigger를 설치하는 경우의 순서는,「설치」에 기재된 순서와 동일합니다.

AI 번역본
-For Reference Only-

11.5 동영상 재생용 뷰어

본 절에서는, 기록한 동영상을 재생하는 뷰어에 대해 설명합니다.

11.5.1 뷰어에의 액세스

동영상 재생용 뷰어를 열려면, 다음의 방법이 있습니다.

1. URL 직접 입력

   브라우저에서 아래 URL에 액세스함으로써, 동영상 재생용 뷰어를 열 수 있습니다.

   - `http://{호스트명or디바이스의 IP 주소}:8120/video-viewer`

   예) `http://192.168.1.3:8120/video-viewer`

2. SpeeDBee Synapse의 화면에서 이동한다

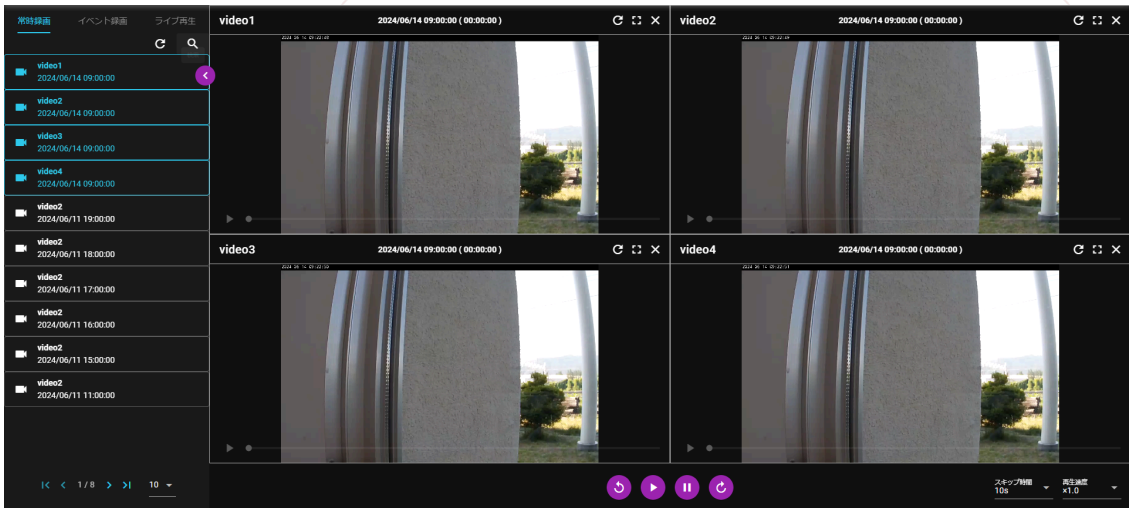
   SpeeDBee Synapse의 헤더에 있는, 동영상 재생용 뷰어로의 이동 아이콘을 누름으로써, 동영상 재생용 뷰어를 열 수 있습니다. 이 아이콘은, i-Trigger가 유효한 경우에만 표시됩니다.



화면을 열면, 처음에 상시 녹화 재생 화면이 표시되어 있습니다. 왼쪽 메뉴의 탭을 선택함으로써, 표시할 화면을 전환할 수 있습니다.

11.5.2 상시 녹화 재생

상시 녹화란, VIDEO 컬렉터가 가동하고 있는 동안에 항상 녹화하고 있는 영상을 가리킵니다. 상시 녹화의 영상을 재생할 수 있습니다.

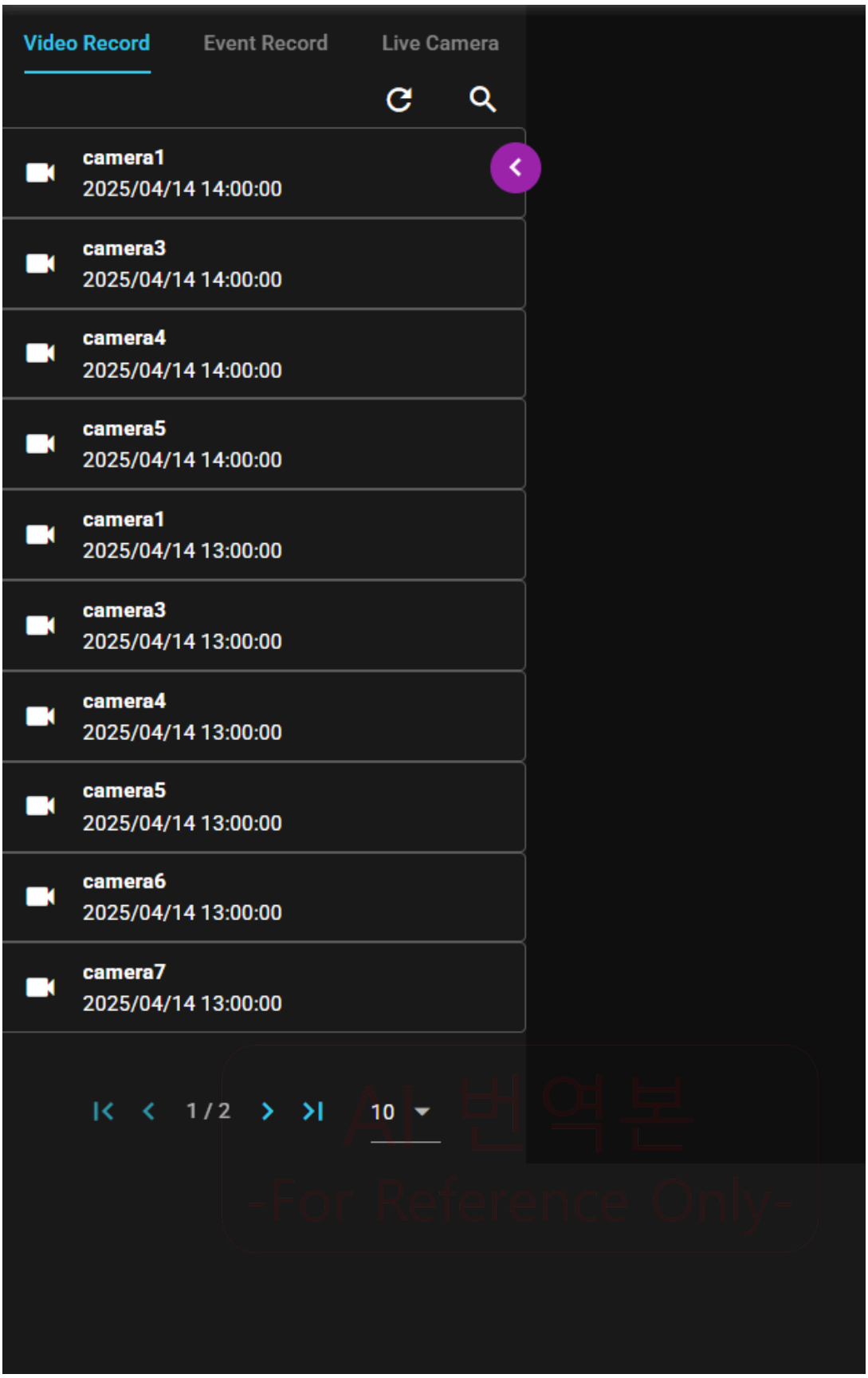


11.5.2.1 목록

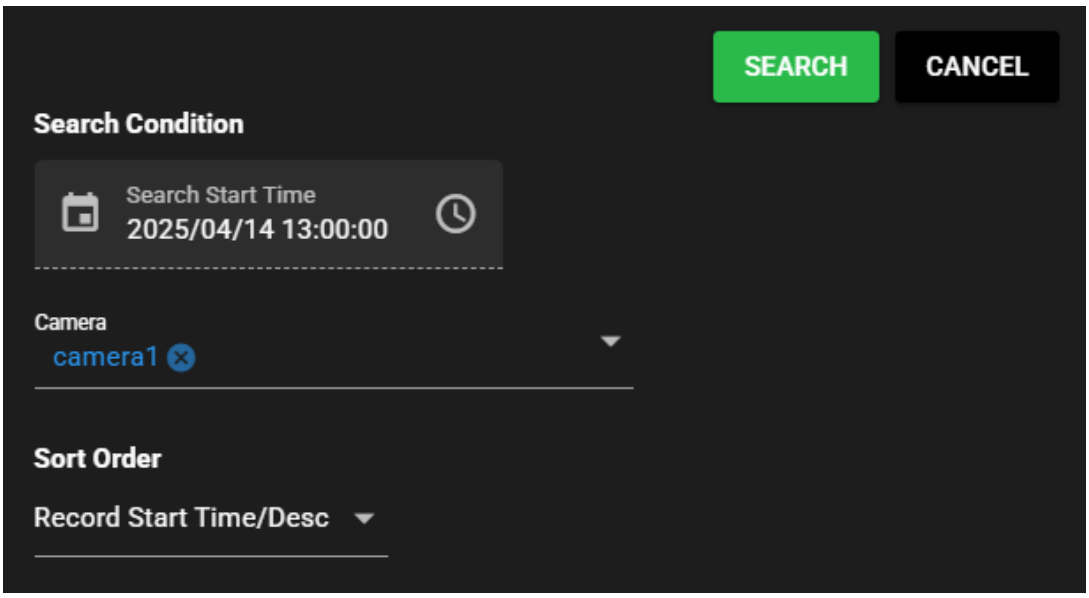
화면의 왼쪽에, 상시 녹화의 목록이 표시됩니다. 초기 상태에서는, 화면을 연 일시 이전에 녹화된 영상이 모두 표시되어 있습니다.
또한, 다음 순서로 검색을 수행할 수 있습니다.

1. 돋보기 아이콘을 누릅니다.

AI 번역본
-For Reference Only-



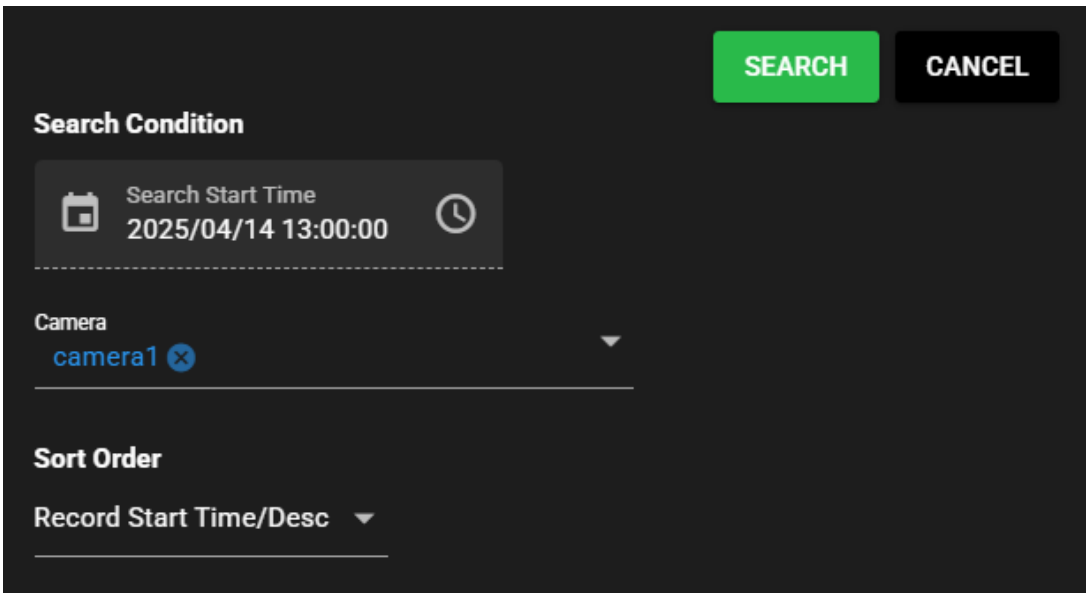
2. 이하의 항목을 입력합니다.



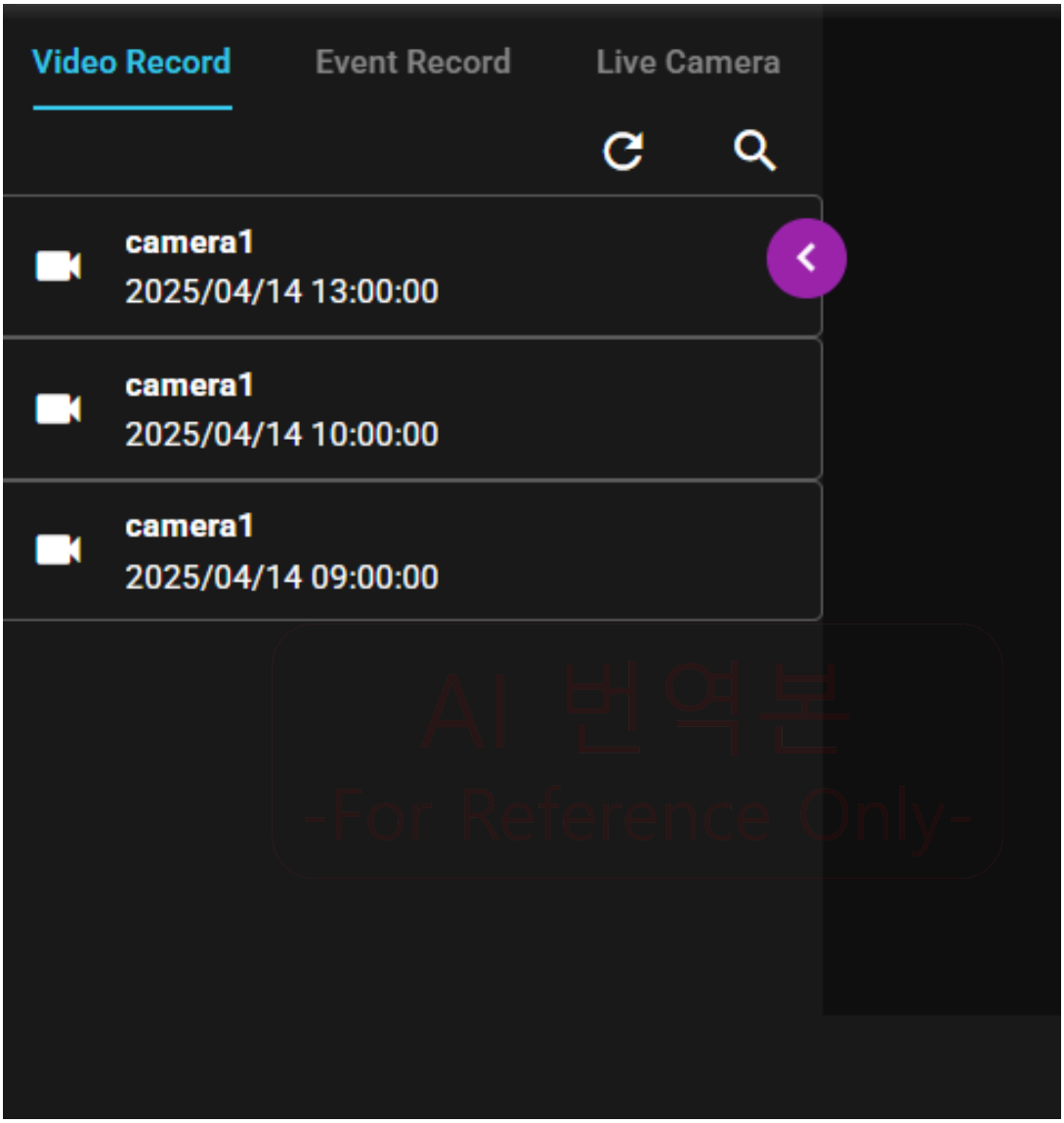
항목	설명
검색 개시 일시	검색 대상으로 하는 시간 범위. ※입력한 시간까지 녹화된 영상을 검색합니다.
카메라	검색 대상으로 하는 VIDEO 컬렉터의 명칭
정렬 순서	정렬의 키로 하는 항목과 순서의 조합 ※1정렬의 키는 다음 중에서 선택합니다. ・녹화 개시 일시 ・카메라명 ※2정렬의 순서는 다음 중에서 선택합니다. ・오름차순 ・내림차순

3.「검색」을 누릅니다.

AI 번역본
-For Reference Only-



4. 검색 결과가 표시됩니다.

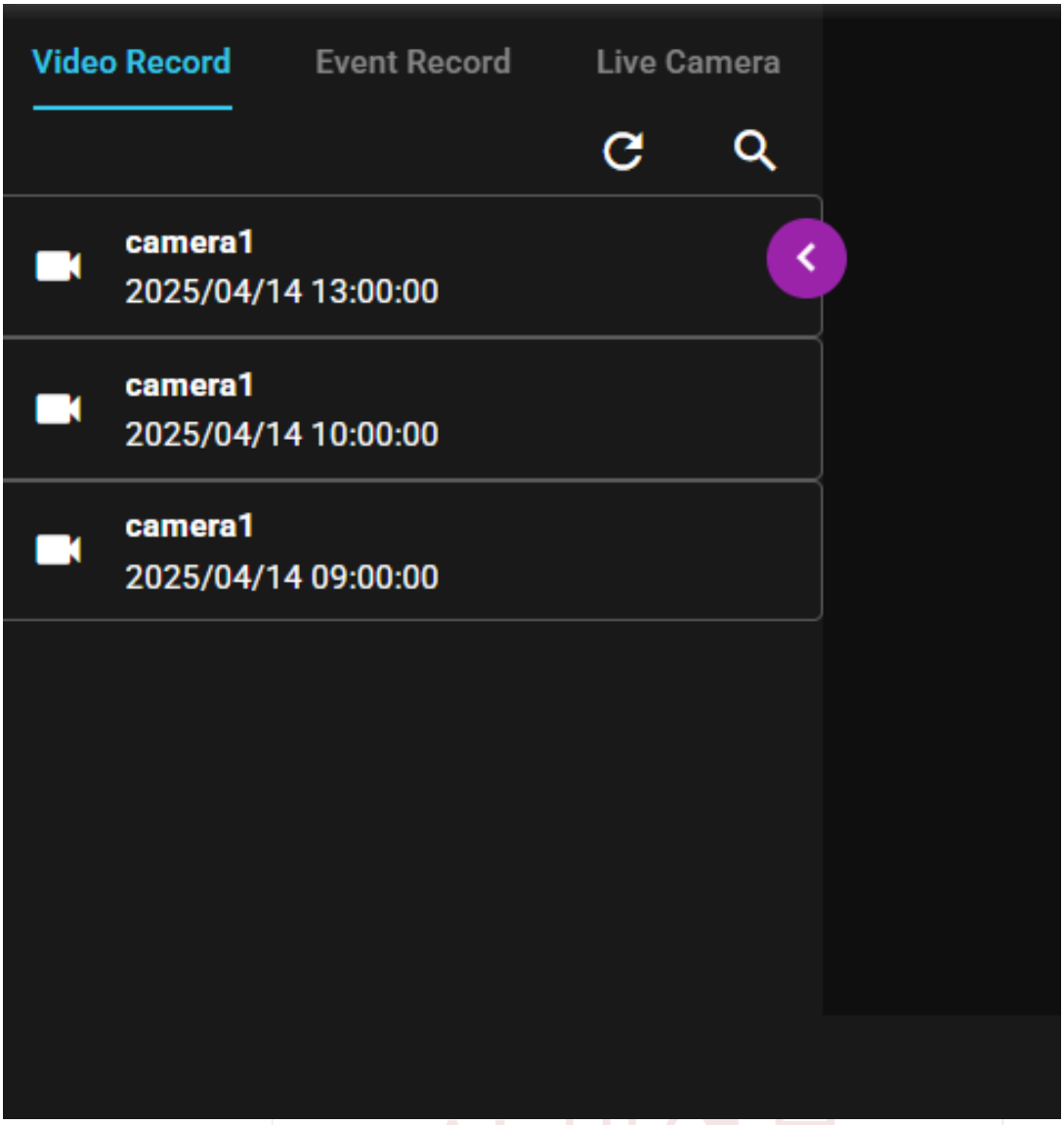


AI 번역본
-For Reference Only-

표시되는 항목은 다음과 같습니다.

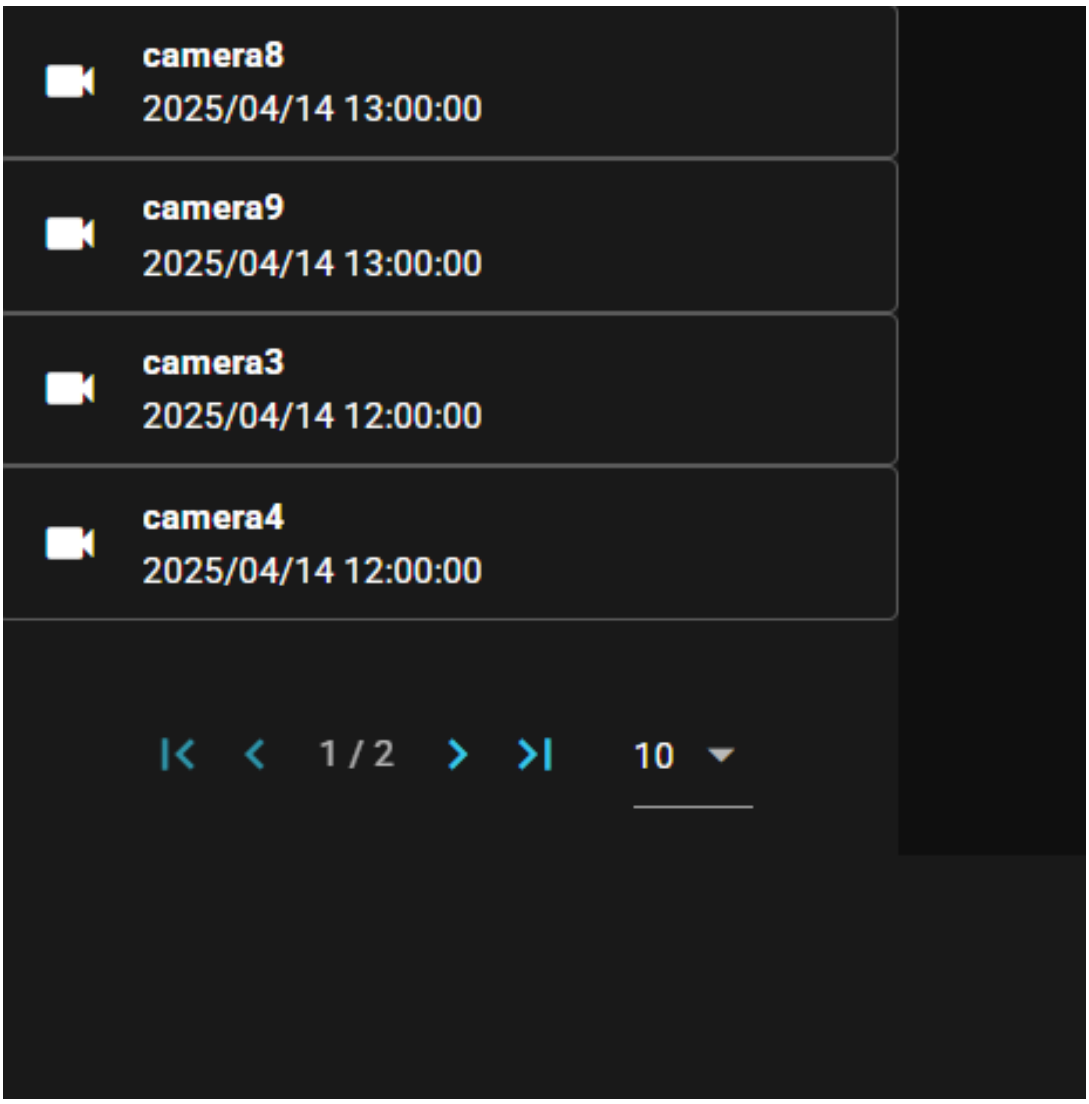
- VIDEO 컬렉터의 명칭
- 녹화 개시 일시

5. 리프레시 아이콘을 누르면, 같은 조건으로 재검색을 수행합니다. 또한, 리프레시는 자동으로도 30초에 1회의 간격으로 수행됩니다.



6. 표시 건수가 많은 경우는, 복수의 페이지로 나뉘어 표시됩니다. 1페이지당 표시 건수를 10, 100, 1000 중에서 선택하는 것이 가능합니다.

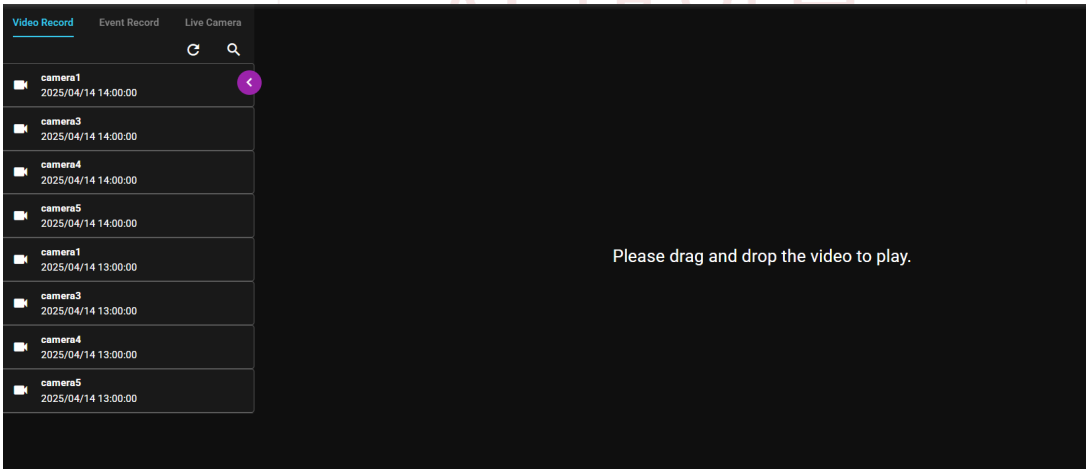
AI 번역문
-For Reference Only-



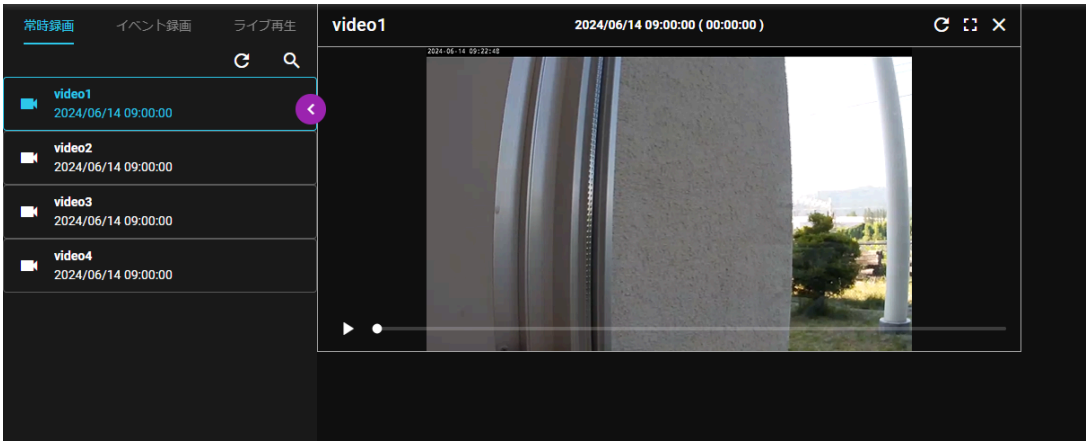
11.5.2.2 재생

다음 순서로, 상시 녹화의 영상을 재생할 수 있습니다. 동시에 10개의 영상을 재생할 수 있습니다.

1. 목록의 상시 녹화를 드래그하여, 오른쪽에 드롭합니다.



2. 상시 녹화의 영상이 표시됩니다. 읽어 들이기가 완료된 후, 재생 아이콘을 누름으로써 재생할 수 있습니다.



화면 항목은 다음과 같습니다.

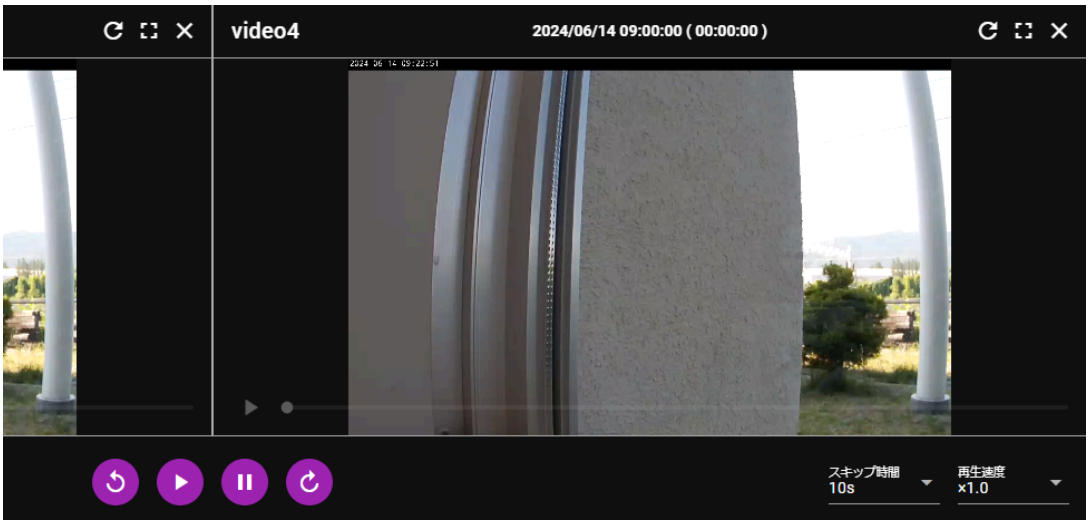


번호	설명
①	VIDEO 컬렉터의 명칭을 표시합니다.
②	녹화를 개시한 일시를 표시합니다.
③	영상을 다시 읽습니다.
④	아이콘을 누른 영상을 전체 화면으로 표시합니다.
⑤	아이콘을 누른 영상을 비표시로 합니다.
⑥	상시 녹화의 영상을 표시합니다.

AI 번역본
-For Reference Only-

번호	설명
⑦	누르면, 영상의 재생・정지를 전환할 수 있습니다.
⑧	현재의 재생 위치가 표시됩니다. 또한, 현재 위치의 아이콘을 드래그 앤 드롭함으로써, 재생 위치를 변경할 수 있습니다.

3. 화면 오른쪽 하부의 영역을 조작함으로써, 복수의 동영상을 동시에 제어할 수 있습니다. 조작 항목은 다음과 같습니다.



항목	설명
①	누르면, 정해진 스킵 시간만큼 재생 위치를 뒤로 되돌립니다.
②	누르면, 표시되어 있는 모든 영상을 재생합니다.
③	누르면, 표시되어 있는 모든 영상을 정지합니다.
④	누르면, 정해진 스킵 시간만큼 재생 위치를 앞으로 진행시킵니다.
⑤	스킵 보내기, 스킵 되돌리기의 초수를, 다음 중에서 선택합니다. ・10초 ・30초 ・60초
⑥	동영상의 재생 속도를, 다음 중에서 선택합니다. ・0.5배속 ・0.75배속 ・1배속 ・1.25배속

AI 번역본
For Reference Only

항목	설명
	・1.5배속 ・2배속

상시 녹화 재생에 대해

상시 녹화 재생에서는, 1시간 단위로 재생 후보를 표시합니다. 복수의 동영상을 동시에 재생할 수 있음으로써, 이하와 같은 사용법이 가능합니다.

- 다른 카메라의 같은 시간대의 영상을 재생（같은 시간대의 영상을 확도가 다른 카메라로 동시 재생하는 등）
- 동일한 카메라로, 다른 시간대의 영상을 재생（예를 들면 정상 시, 이상 시를 함께 비교해 볼 수 있음）

11.5.3 이벤트 녹화 재생

SpeeDBee Synapse의 Event Recorder에서 생성한 이벤트를 확인할 수 있습니다. 이벤트 발생 시의 카메라 영상, 데이터를 확인할 수 있습니다.

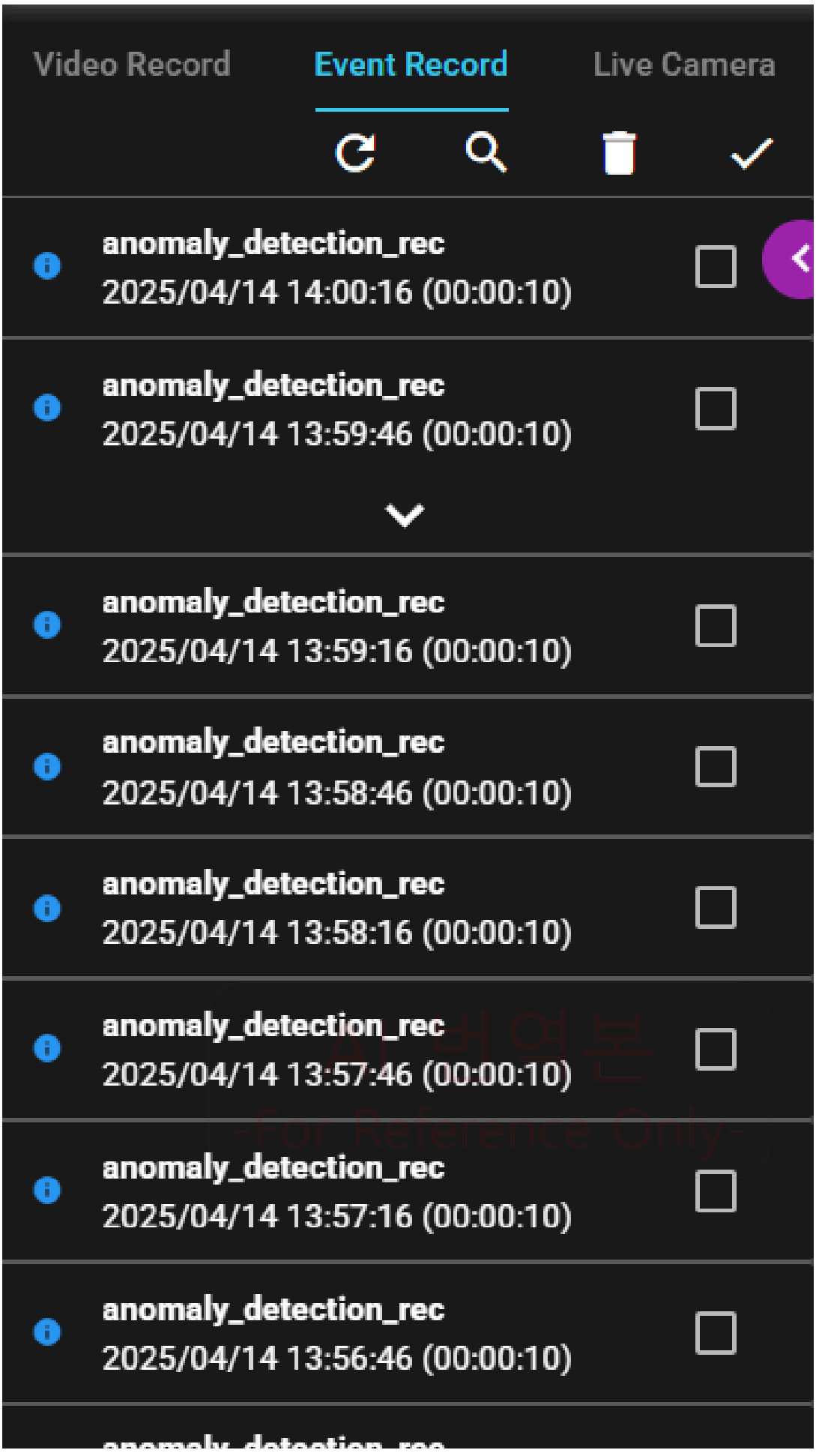


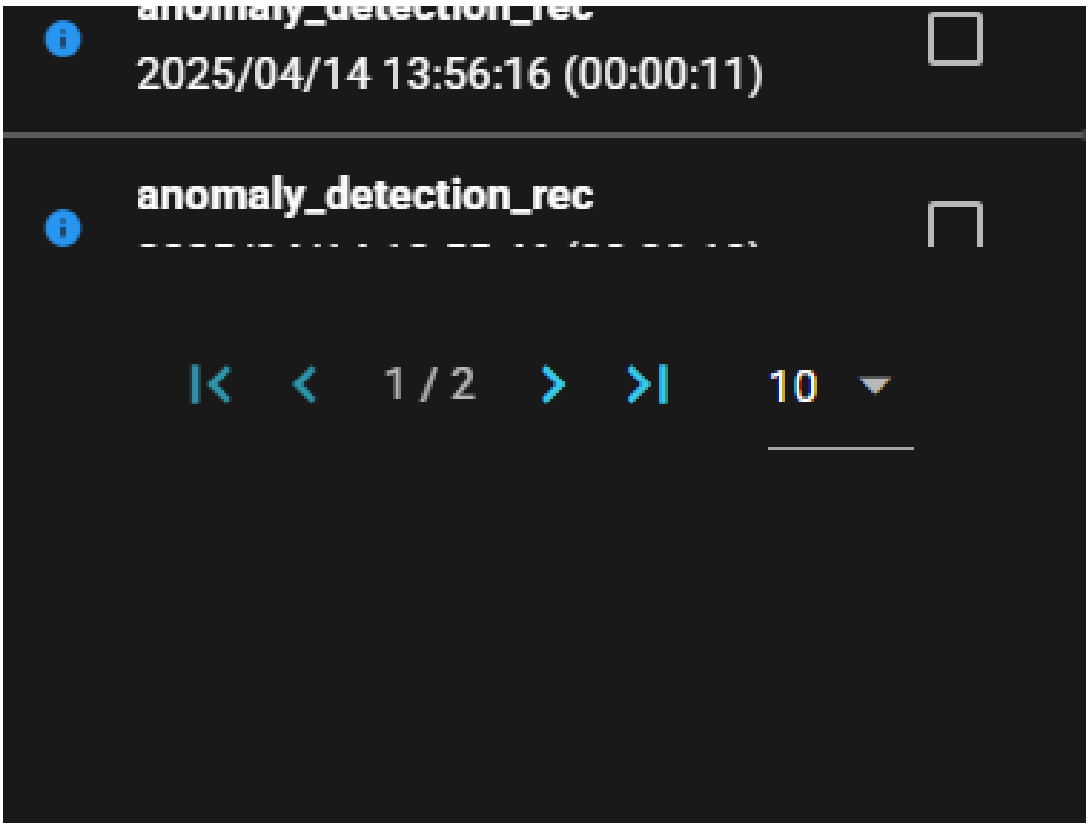
AI 번역본
-For Reference Only-

11.5.3.1 목록

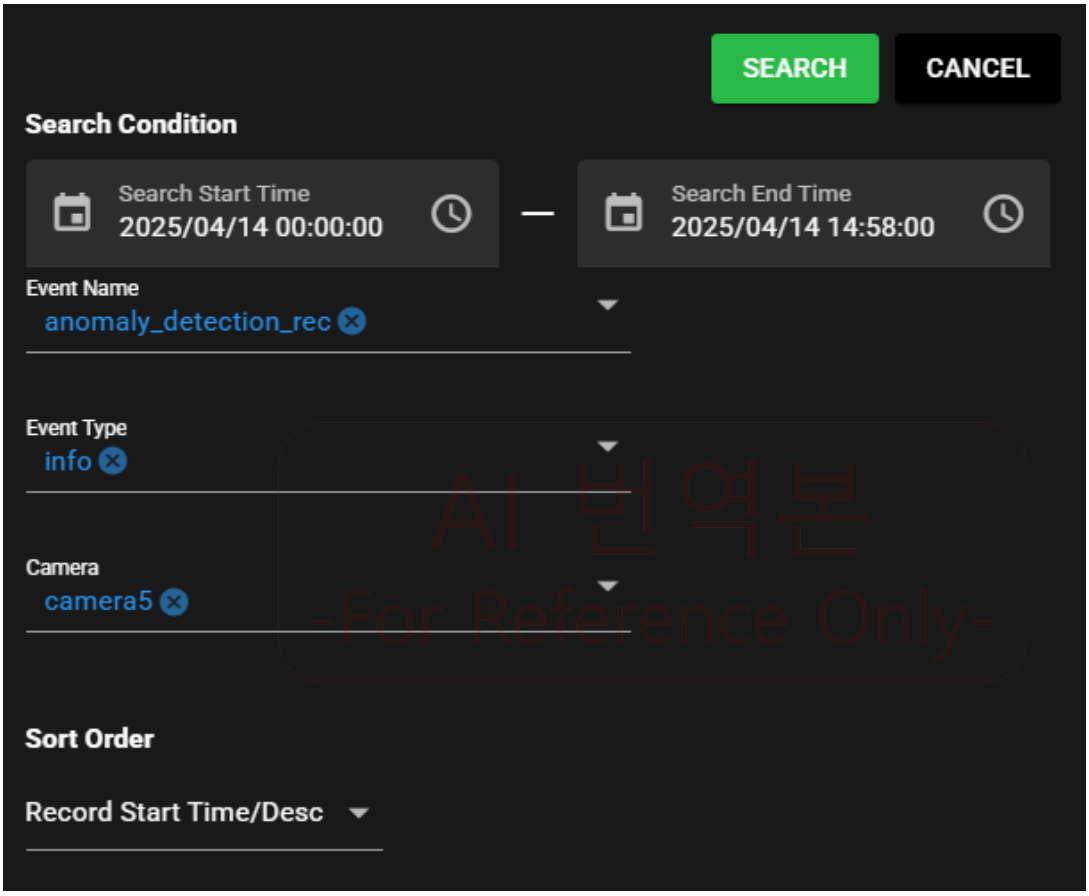
화면의 왼쪽에, 이벤트의 목록이 표시됩니다. 초기 상태에서는, 화면을 연 날의 00:00:00~24:00:00에 발생한 모든 이벤트가 표시되어 있습니다. 또한, 다음 순서로 이벤트를 검색할 수 있습니다.

1. 돋보기 아이콘을 누릅니다.


Video Record
Event Record
Live Camera
anomaly_detection_rec
2025/04/14 14:00:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:59:46 (00:00:10)
anomaly_detection_rec
2025/04/14 13:59:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:58:46 (00:00:10)
anomaly_detection_rec
2025/04/14 13:58:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:57:46 (00:00:10)
anomaly_detection_rec
2025/04/14 13:57:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:56:46 (00:00:10)



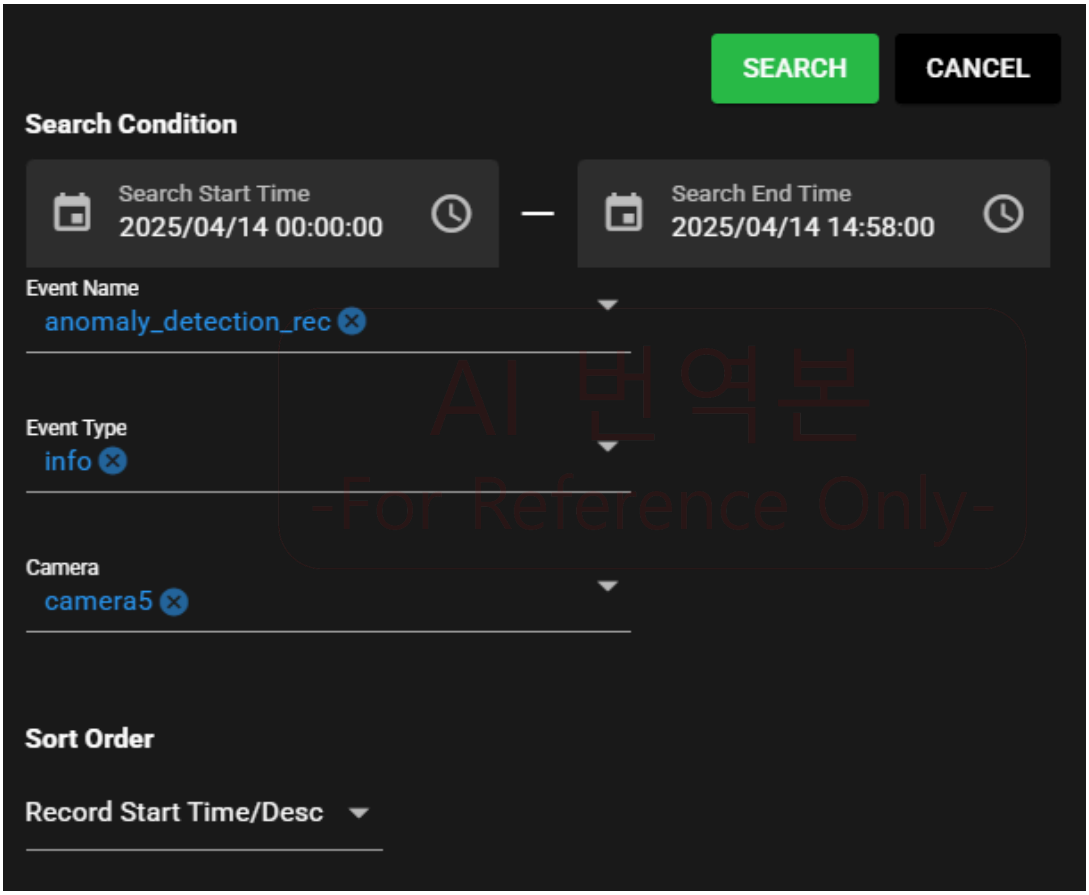
2. 이하의 항목을 입력합니다.



AI 번역본
-For Reference Only-

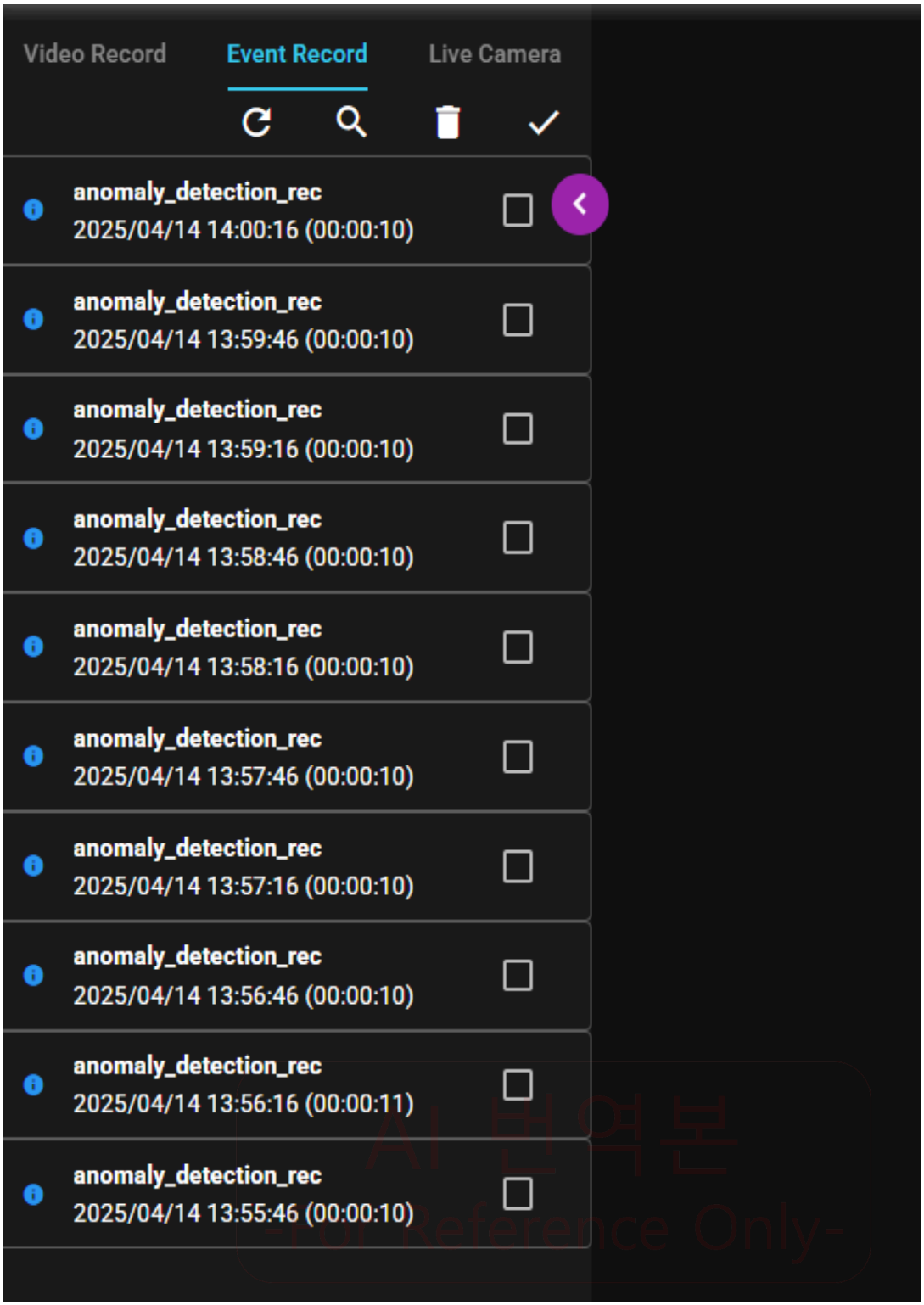
항목	설명
검색 개시 일시・검색 종료 일시	검색 대상으로 하는 시간 범위. ※입력한 시간 범위의 사이에, 개시 이벤트가 발생한 이벤트를 검색합니다.
이벤트명	검색 대상 이벤트의 이름
이벤트 분류	검색 대상 이벤트의 분류
카메라	검색 대상 이벤트에 연결되어 있는 카메라의 이름
정렬 순서	정렬의 키로 하는 항목과 순서의 조합 ※1정렬의 키는 다음 중에서 선택합니다. ・기록 개시 일시 ・이벤트명 ・이벤트 종별 ※2정렬의 순서는 다음 중에서 선택합니다. ・오름차순 ・내림차순

3.「검색」을 누릅니다.



AI 번역본
-For Reference Only-

4. 검색 결과가 표시됩니다.

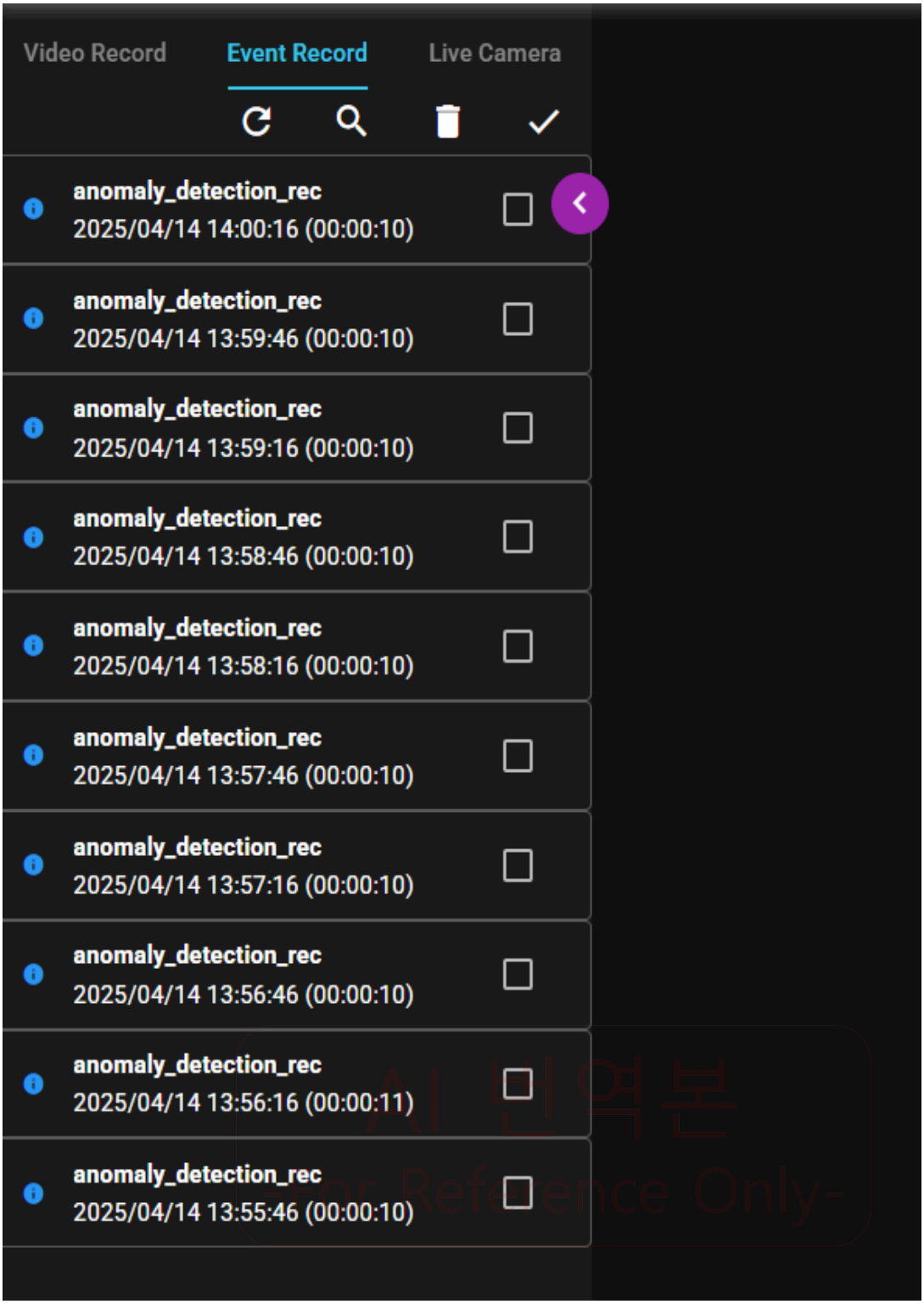


표시되는 항목은 다음과 같습니다.

- 이벤트 분류를 나타내는 아이콘
- 이벤트명
- 개시 이벤트의 발생 일시
- 이벤트의 기록 시간

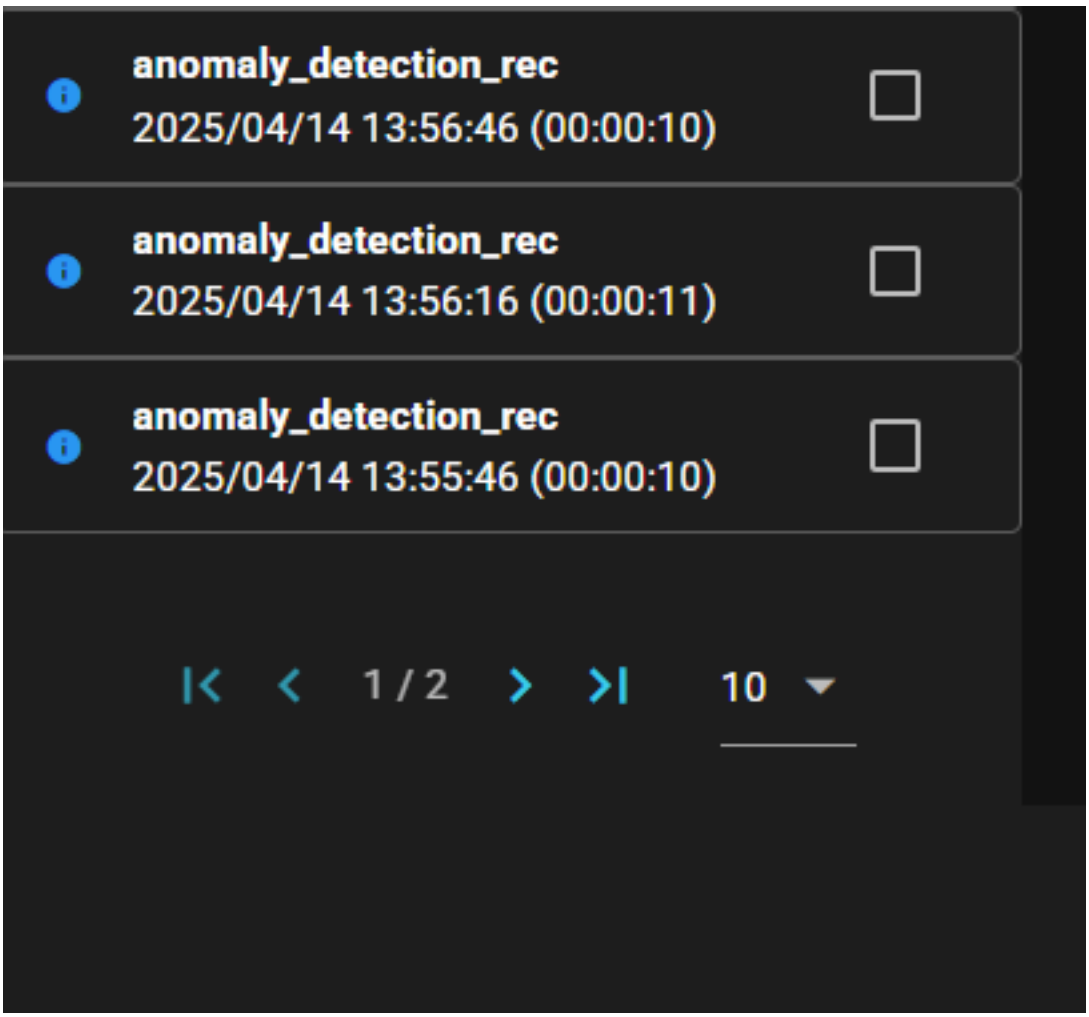
AI 번역 본
-For Reference Only-

5. 리프레시 아이콘을 누르면, 같은 조건으로 재검색을 수행합니다. 또한, 리프레시는 자동으로도 30초에 1회의 간격으로 수행됩니다.



6. 표시 건수가 많은 경우는, 복수의 페이지로 나뉘어 표시됩니다. 1페이지당 표시 건수를 10, 100, 1000 중에서 선택하는 것이 가능합니다.

AI 번역본
-For Reference Only-

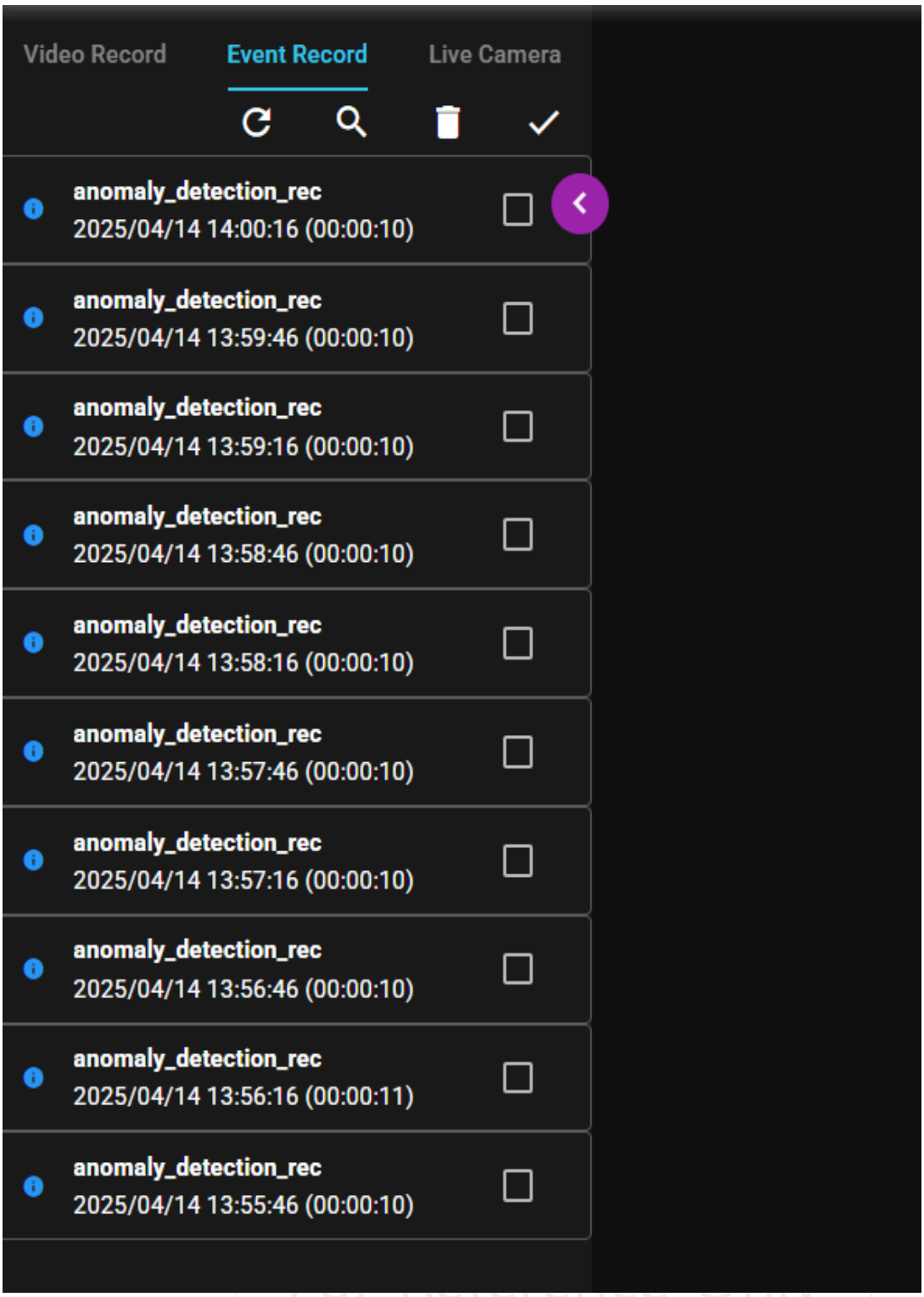


11.5.3.2 재생

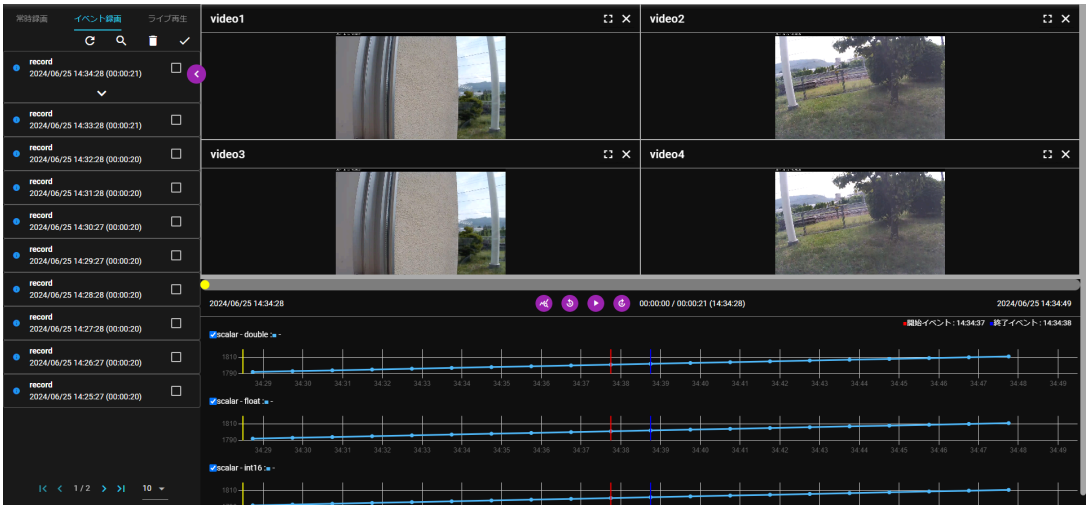
다음 순서로, 이벤트 발생 시의 컴포넌트의 데이터의 그래프와, 카메라의 영상을 재생할 수 있습니다. 동시에 재생할 수 있는 카메라의 대수는 4대입니다.

1. 이벤트를 선택합니다.

AI 번역본
-For Reference Only-

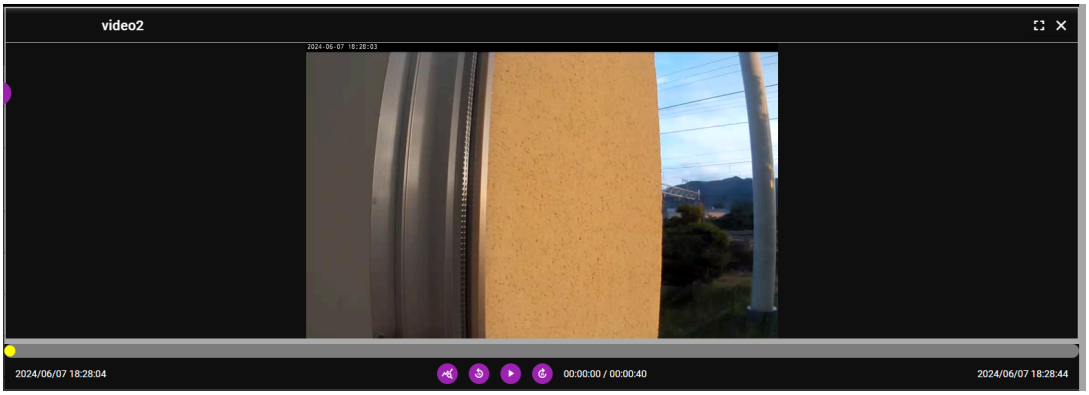


2. 선택한 이벤트의 동영상과 데이터가, 오른쪽 화면에 표시됩니다.



3. 재생 버튼 또는 동영상을 선택하면, 이벤트에 연결된 동영상과 그래프가 재생됩니다. 화면 항목은 다음과 같습니다.

   ◦ 동영상 부분

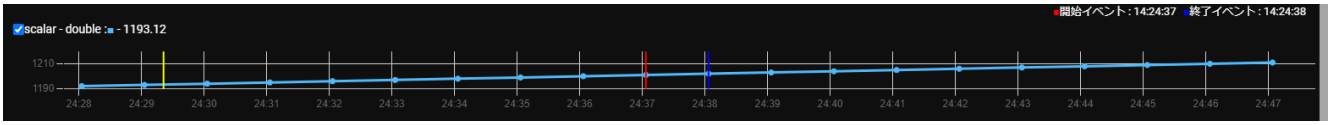


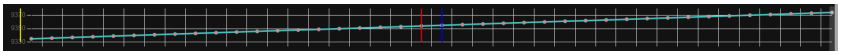
번호	설명
①	VIDEO 컬렉터의 명칭을 표시합니다.
②	아이콘을 누른 동영상을 전체 화면으로 표시합니다.
③	아이콘을 누른 동영상을 비표시로 합니다.
④	이벤트 발생 시의, 이벤트에 등록된 카메라의 영상을 표시합니다. 누름으로써, 이벤트의 재생・정지를 전환할 수 있습니다.
⑤	동영상의 현재의 재생 위치를 표시합니다. 현재 위치의 점을 드래그 앤 드롭함으로써, 재생 위치를 변경하는 것도 가능합니다. 또한, 동영상과의 사이에 있는 영역을 드래그한 상태에서, 상하로 이동시킴으로써, 동영상의 표시 영역을 조정할 수 있습니다.
⑥	이벤트의 기록 개시 일시를 표시합니다.

AI 번역본
-For Reference Only-

번호	설명
⑦	누르면, 그래프의 표시 상태를 다음 중에서 전환합니다. ・전체 그래프...그래프의 모든 데이터를 표시합니다. ・상세 그래프...현재의 재생 위치의 30초 전후의 데이터를 표시합니다.
⑧	누르면, 재생 위치를 10초 전으로 스킵합니다.
⑨	누르면, 동영상과 그래프의 재생을 개시합니다.
⑩	누르면, 재생 위치를 10초 후로 스킵합니다.
⑪	이벤트의 기록 시간과, 재생 경과 시간을 표시합니다.
⑫	이벤트의 기록 종료 일시를 표시합니다.

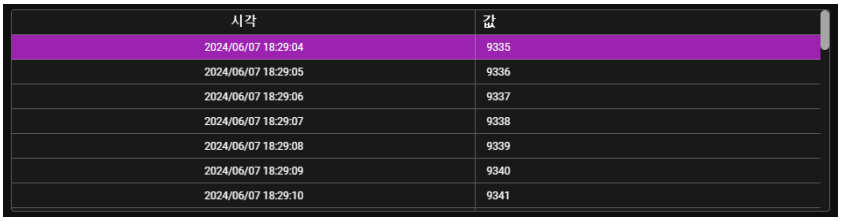
- 그래프 부분



항목	설명
①	체크를 넣은 그래프를 표시하고, 체크를 해제한 그래프를 비표시로 합니다.
②	데이터의 송신원인 컴포넌트의 이름과 데이터명을 표시합니다.
③	데이터의 선의 색과, 현재의 재생 위치의 값을 표시합니다.
④	개시 이벤트, 종료 이벤트의 시간과 선의 색을 표시합니다.
⑤(수치 데이터인 경우)	이벤트 발생 시의, 이벤트에 등록된 컴포넌트의 데이터가, 꺾은선 그래프로 표시됩니다. X축이 시간을, Y축이 값을 나타냅니다. 또한, 이하의 선이 표시됩니다. ・노란 선...동영상의 재생과 연동하여 이동하고, 현재의 재생 위치를 나타냅니다. ・빨간 선...개시 이벤트의 시간을 나타냅니다. ・파란 선...종료 이벤트의 시간을 나타냅니다. 종료 이벤트가 발생한 경우에만 표시됩니다. 
⑤(문자열 데이터인 경우)	이벤트 발생 시의, 이벤트에 등록된 컴포넌트의 데이터가, 시각・값의 2열로 이루어진 테이블로 표시됩니다. 또한, 이하의 표시가 이루어집니다.

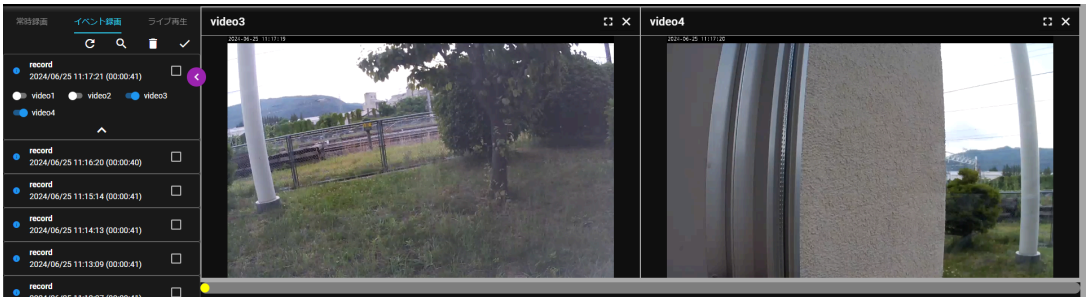
AI 번역본
-For Reference Only-

항목	설명
	・보라색 행...동영상의 재생과 연동하여 이동하고, 현재의 재생 위치를 나타냅니다. ・빨간 표식...시간 옆에 표시되어, 개시 이벤트를 나타냅니다. ・파란 표식...시간 옆에 표시되어, 종료 이벤트를 나타냅니다. 종료 이벤트가 발생한 경우에만 표시됩니다.



시각	값
2024/06/07 18:29:04	9335
2024/06/07 18:29:05	9336
2024/06/07 18:29:06	9337
2024/06/07 18:29:07	9338
2024/06/07 18:29:08	9339
2024/06/07 18:29:09	9340
2024/06/07 18:29:10	9341

4. 목록 내의 선택한 이벤트에 표시되어 있는, 카메라명 옆의 토글 스위치를 전환함으로써, 동영상의 표시・비표시를 전환할 수 있습니다.

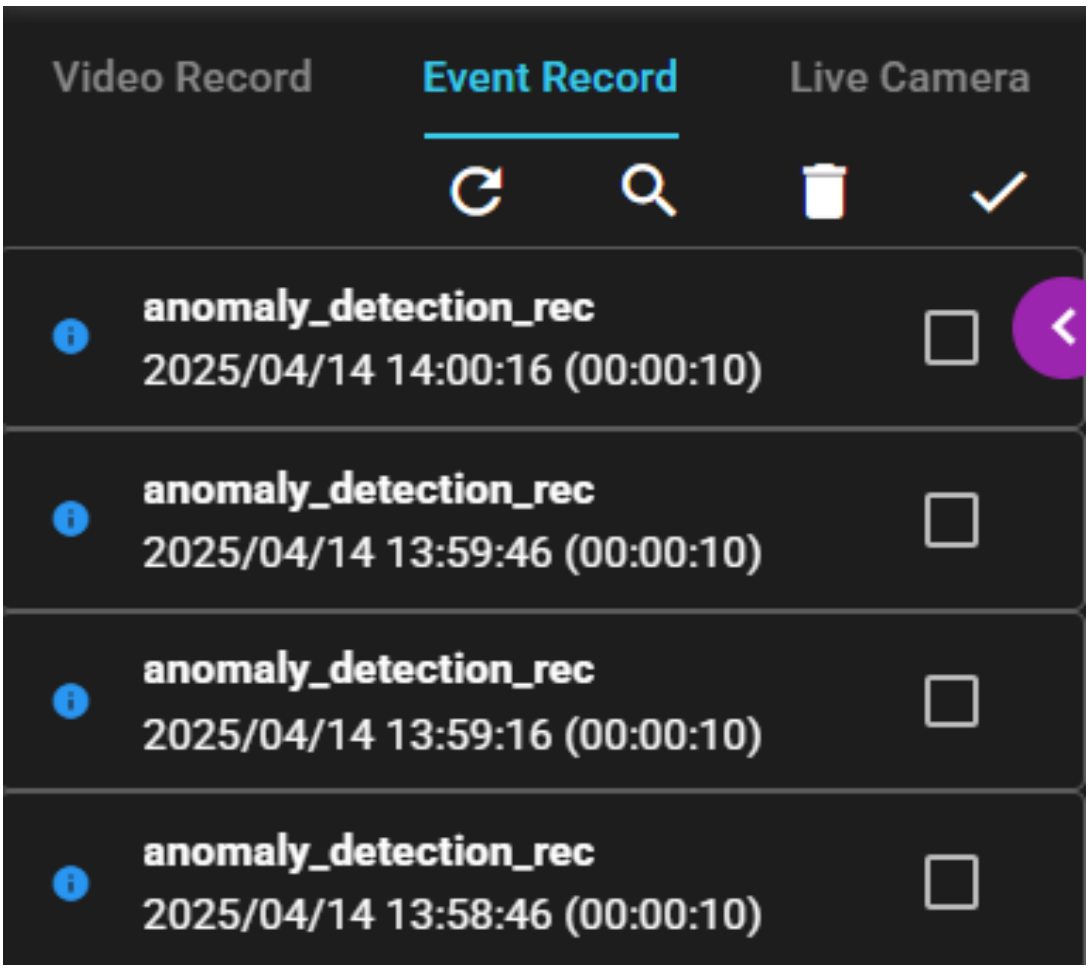


11.5.3.3 삭제

다음 순서로, 이벤트를 삭제할 수 있습니다.

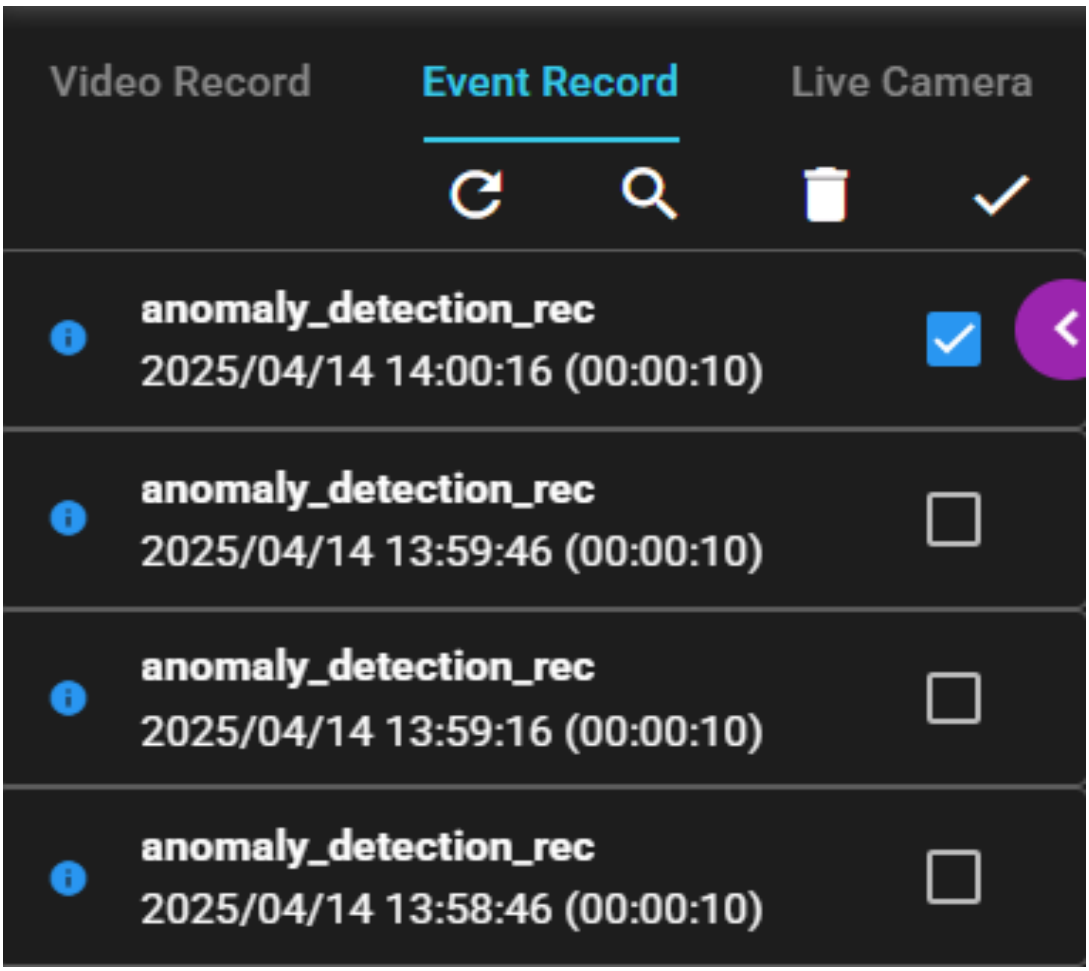
1. 이벤트의 체크박스에 체크를 넣습니다. 헤더의 체크 아이콘을 누르면, 리스트에 표시되어 있는 모든 이벤트에 체크를 넣을 수 있습니다.

AI 번역본
-For Reference Only-

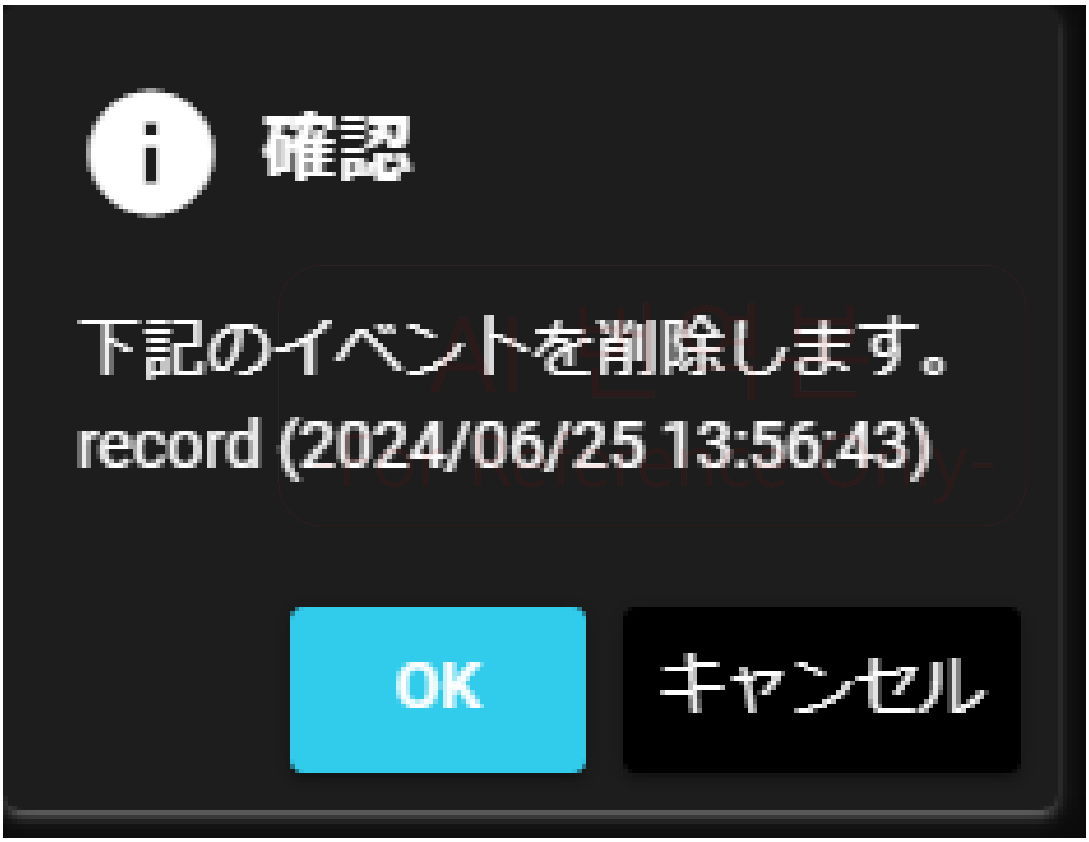


2. 헤더의 삭제 아이콘을 누릅니다.

AI 번역본
-For Reference Only-


Video Record
Event Record
Live Camera
anomaly_detection_rec
2025/04/14 14:00:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:59:46 (00:00:10)
anomaly_detection_rec
2025/04/14 13:59:16 (00:00:10)
anomaly_detection_rec
2025/04/14 13:58:46 (00:00:10)

3.「OK」を選択します.


確認
下記のイベントを削除します。
record (2024/06/25 13:56:43)
OK
キャンセル

4. 체크를 넣은 이벤트가 삭제됩니다.

11.5.3.4 URL 지정에 의한 이벤트 표시

SpeeDBee Synapse의 Event Recorder에서 생성된 이벤트에는, 녹화 개시 일시 (YYYY/MM/DD hh:mm:ss) 를 포함하는 아래의 URL을 지정함으로써 직접 액세스할 수 있습니다. 같은 시각에 복수의 이벤트가 있는 경우는, URL의 말미에 ID가 번호로 부여됩니다.

- http://{호스트명or디바이스의 IP 주소}:8120/video-viewer/event_record/{YYYY-MM-DDThhmmss}

  예1)

  - `http://192.168.1.3:8120/video-viewer/event_record/2025-03-28T165209`

  예2)※시각이 중복되는 이벤트가 있는 경우

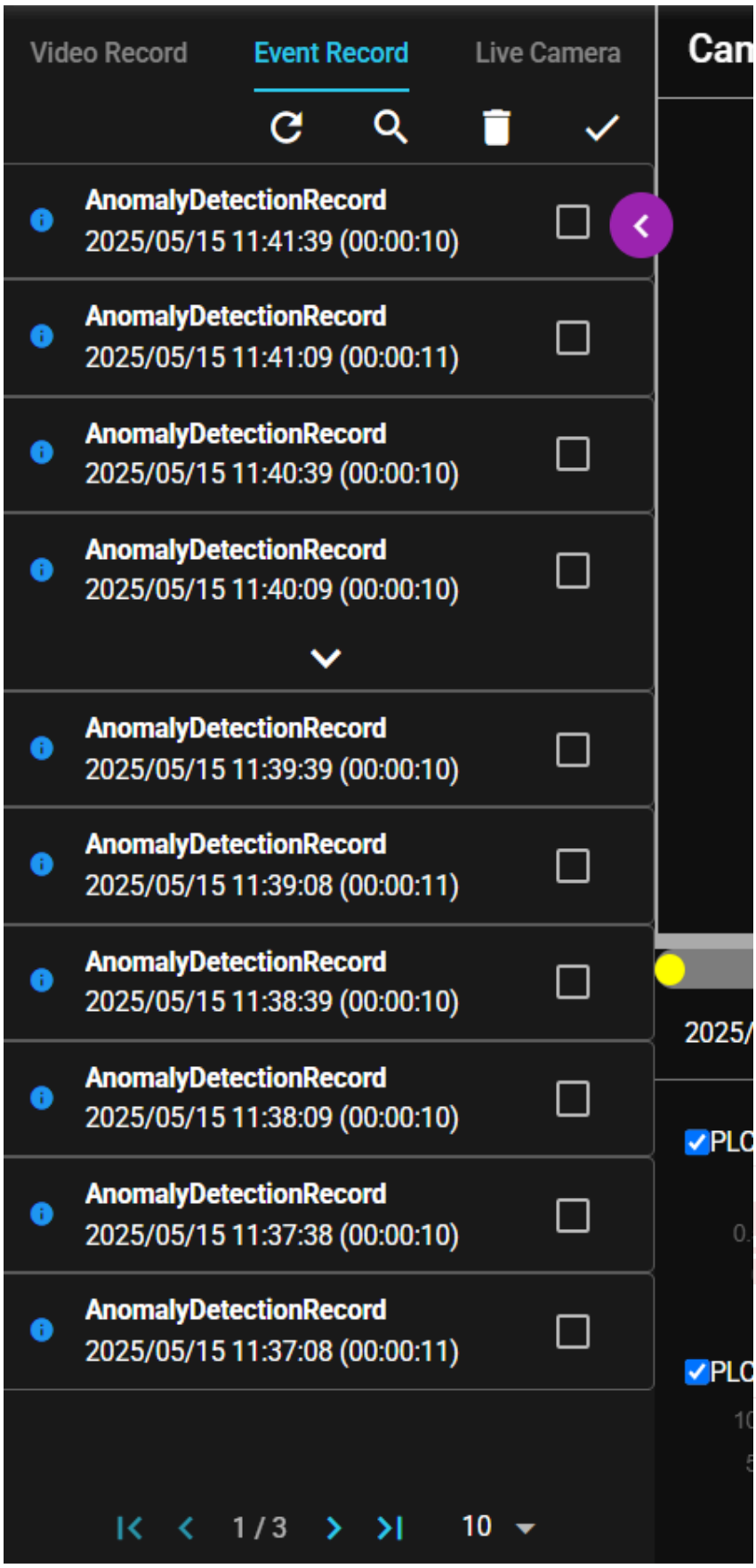
  - `http://192.168.1.3:8120/video-viewer/event_record/2025-03-28T165210/1`
  - `http://192.168.1.3:8120/video-viewer/event_record/2025-03-28T165210/2`

URL은 이벤트의 녹화 개시 일시의 포맷을「 YYYY-MM-DDThhmmss 」로 변환하여 생성하고 있습니다.

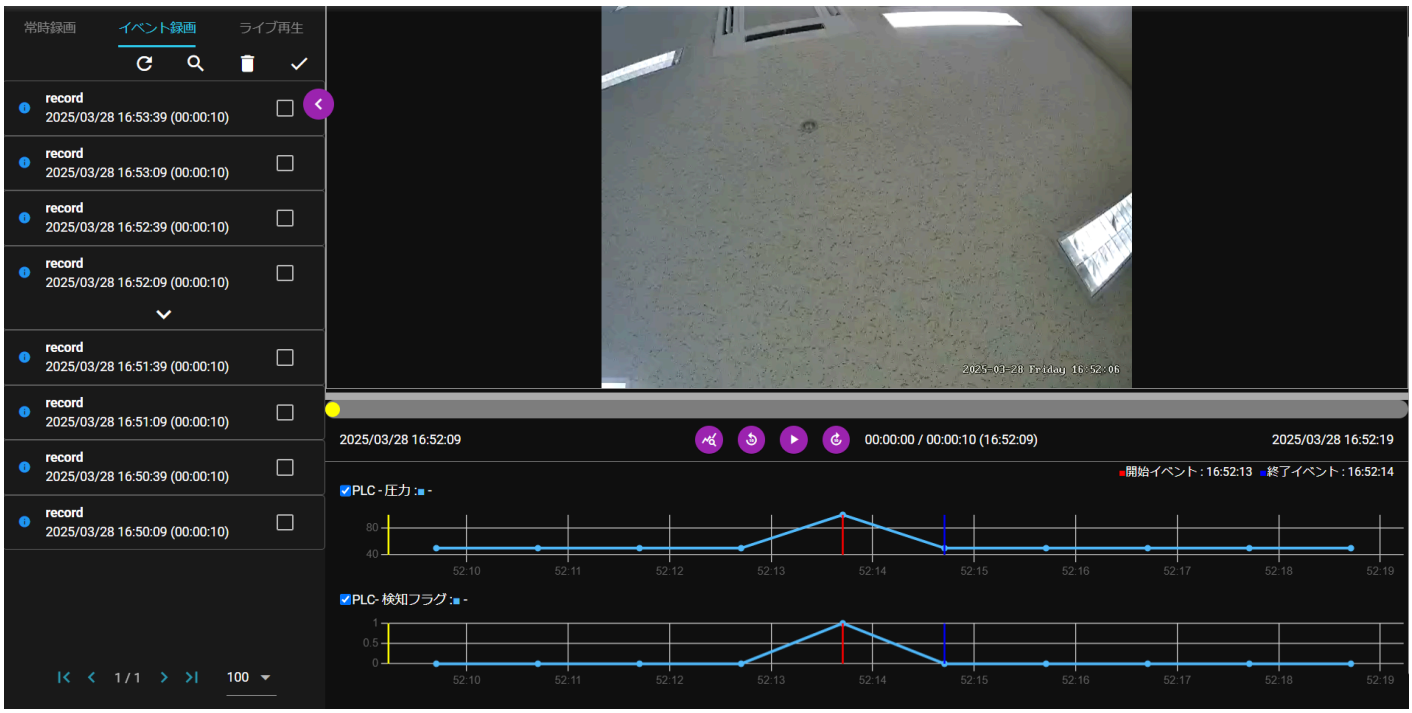


각 이벤트를 누름으로써 URL도 동적으로 갱신됩니다.

AI 번역본
-For Reference Only-

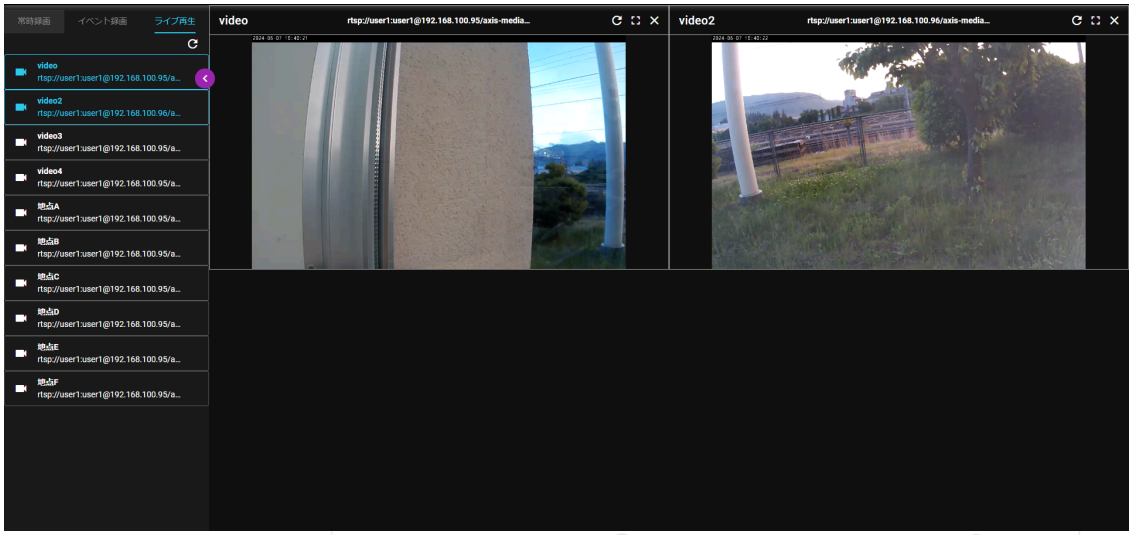


URL을 지정하여 이벤트에 액세스하면, 대상 이벤트의 녹화 개시 시각의 전후 2분 이내에 발생한 이벤트가 목록에 표시됩니다. 액세스 시, 이벤트 목록의 1페이지당 최대 표시 건수는 100건이 되며, 대상 이벤트가 선택된 상태가 됩니다.



11.5.4 라이브 재생

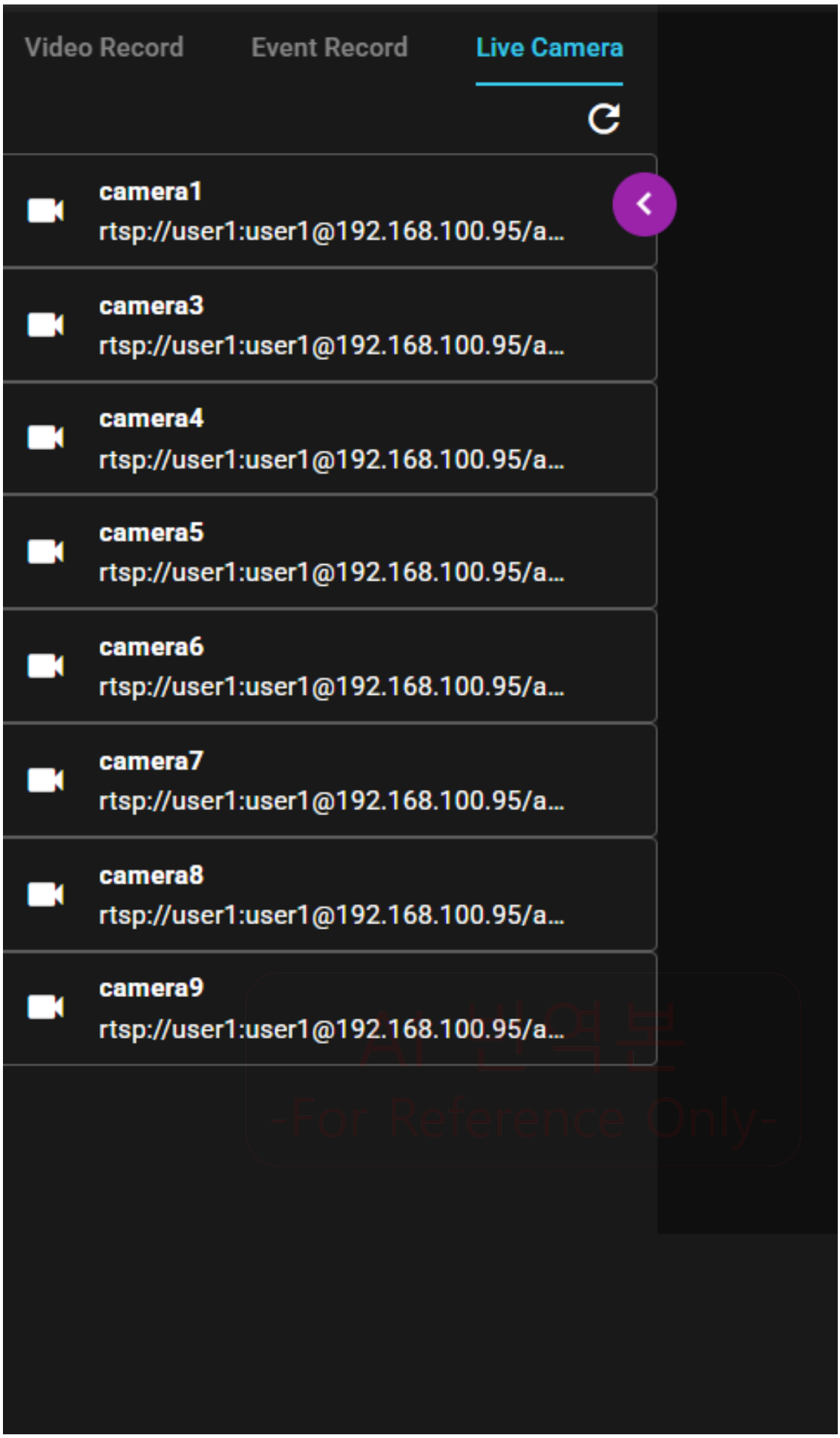
SpeeDBee Synapse에서 가동하고 있는 VIDEO 컬렉터의 영상（현재의 카메라 영상）을 재생할 수 있습니다.



-For Reference Only-

11.5.4.1 목록

화면의 왼쪽에, VIDEO 컬렉터의 목록이 표시됩니다.



AI 번역 본
-For Reference Only-

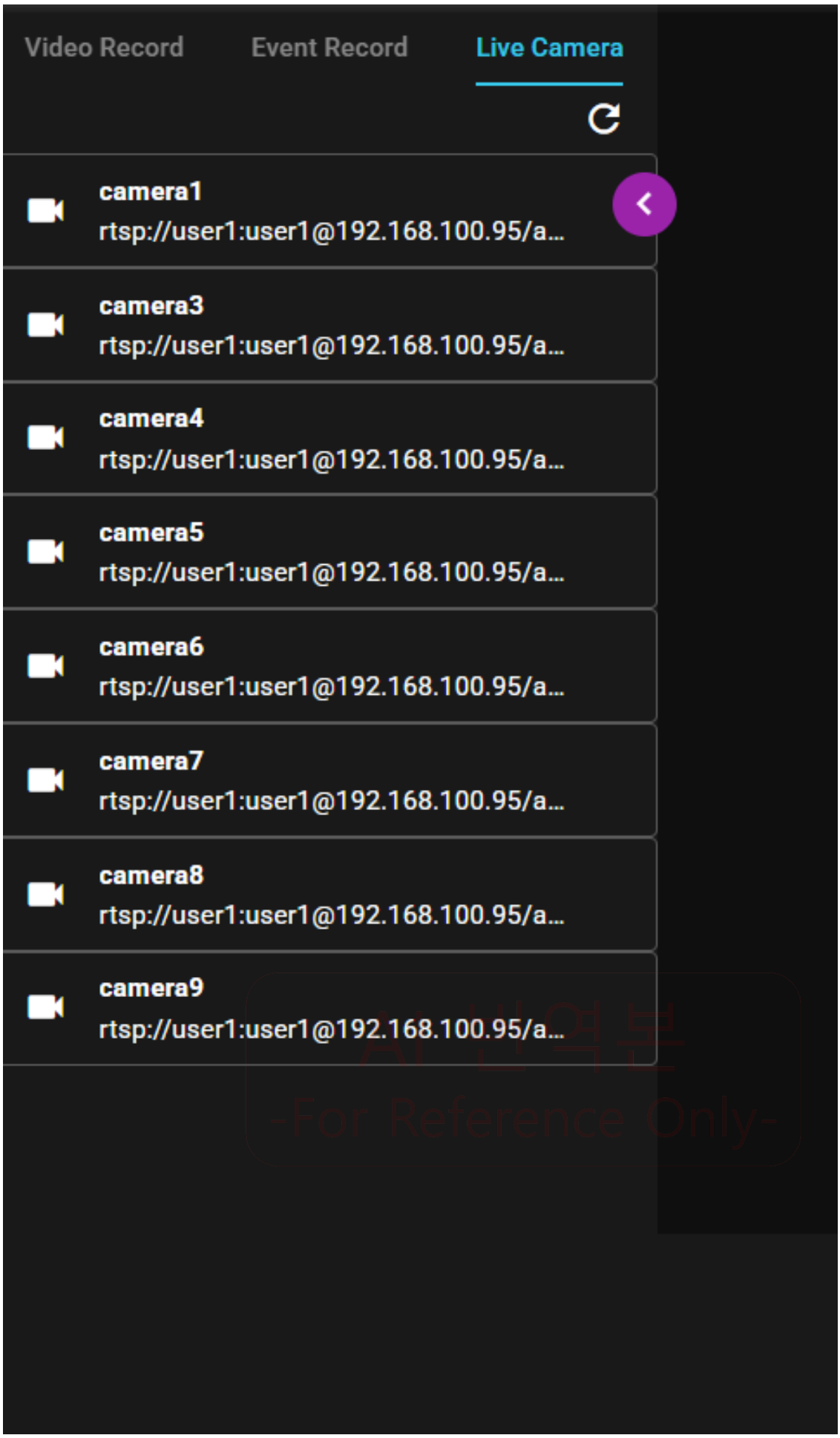
[redacted]

표시되는 항목은 다음과 같습니다.

- VIDEO 컬렉터의 명칭
- VIDEO 컬렉터에 등록되어 있는, 카메라의 RTSP URL

또한, 리프레시 아이콘을 누르면, VIDEO 컬렉터의 최신 상태에 맞추어, 목록의 표시 내용이 갱신됩니다. (등록, 갱신, 삭제된 것이 적용)

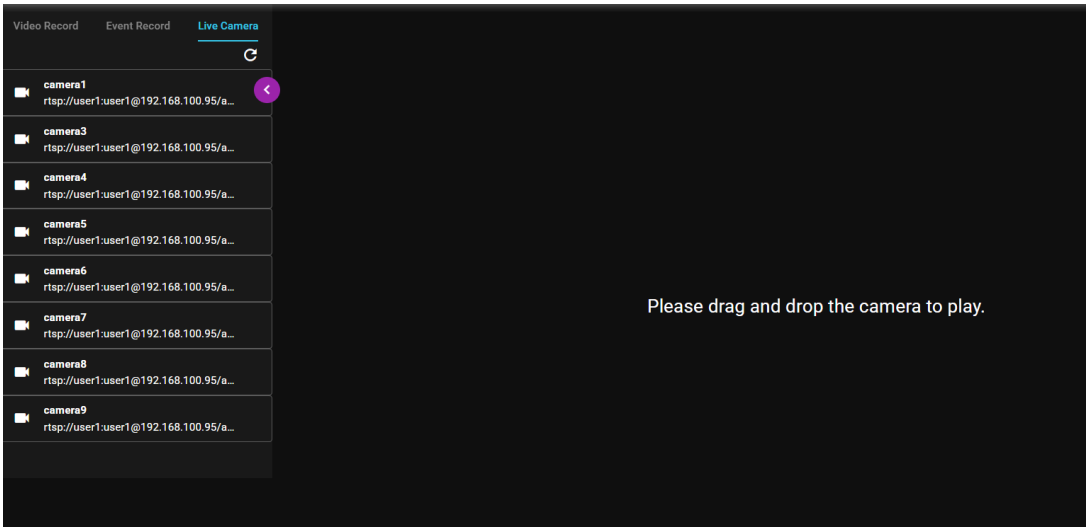
AI 번역본
-For Reference Only-


Video Record
Event Record
Live Camera
camera1
rtsp://user1:user1@192.168.100.95/a...
camera3
rtsp://user1:user1@192.168.100.95/a...
camera4
rtsp://user1:user1@192.168.100.95/a...
camera5
rtsp://user1:user1@192.168.100.95/a...
camera6
rtsp://user1:user1@192.168.100.95/a...
camera7
rtsp://user1:user1@192.168.100.95/a...
camera8
rtsp://user1:user1@192.168.100.95/a...
camera9
rtsp://user1:user1@192.168.100.95/a...
AI 번역 본
-For Reference Only-

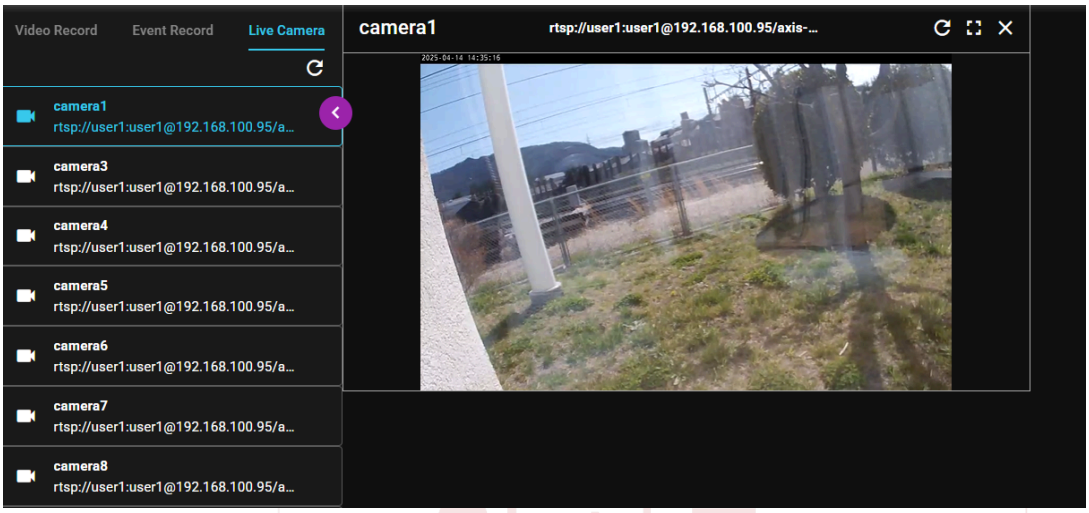
11.5.4.2 재생

다음 순서로, 카메라의 현재의 영상을 재생할 수 있습니다. 동시에 10대의 카메라를 재생할 수 있습니다.

1. 목록의 카메라를 드래그하여, 오른쪽에 드롭합니다.

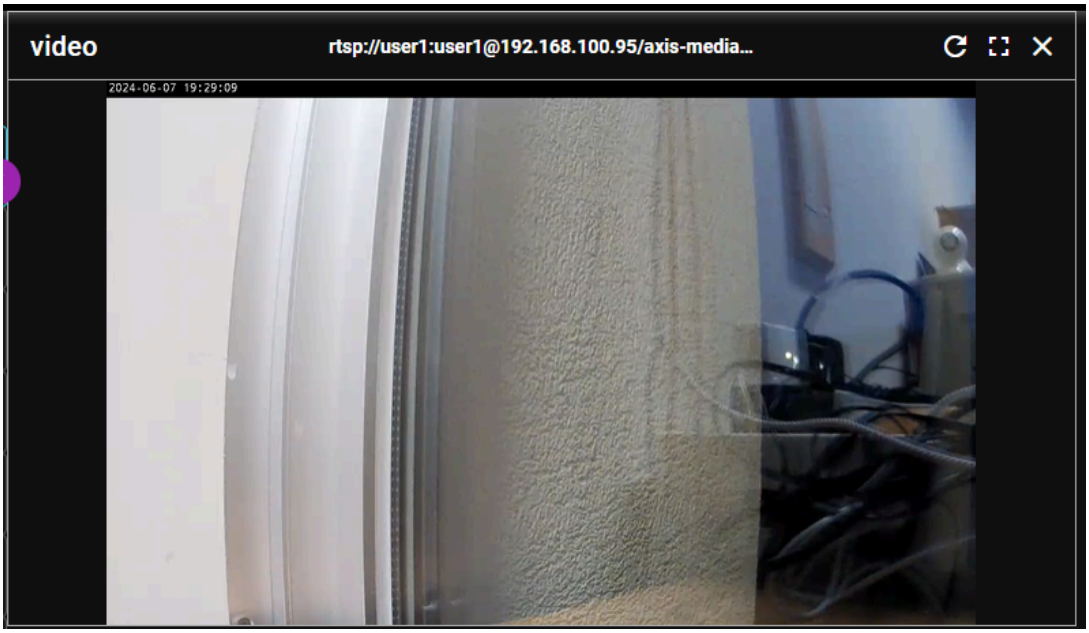


2. 카메라의 현재의 영상이 재생됩니다.



화면 항목은 다음과 같습니다.

-For Reference Only-



번호	설명
①	VIDEO 컬렉터의 명칭을 표시합니다.
②	VIDEO 컬렉터에 등록되어 있는, 카메라의 RTSP URL을 표시합니다.
③	아이콘을 누른 카메라에 재접속합니다.
④	아이콘을 누른 카메라의 영상을 전체 화면으로 표시합니다.
⑤	아이콘을 누른 카메라를 비표시로 합니다.
⑥	카메라의 라이브 영상을 재생합니다.

스토리지의 여유 용량이 없는 경우

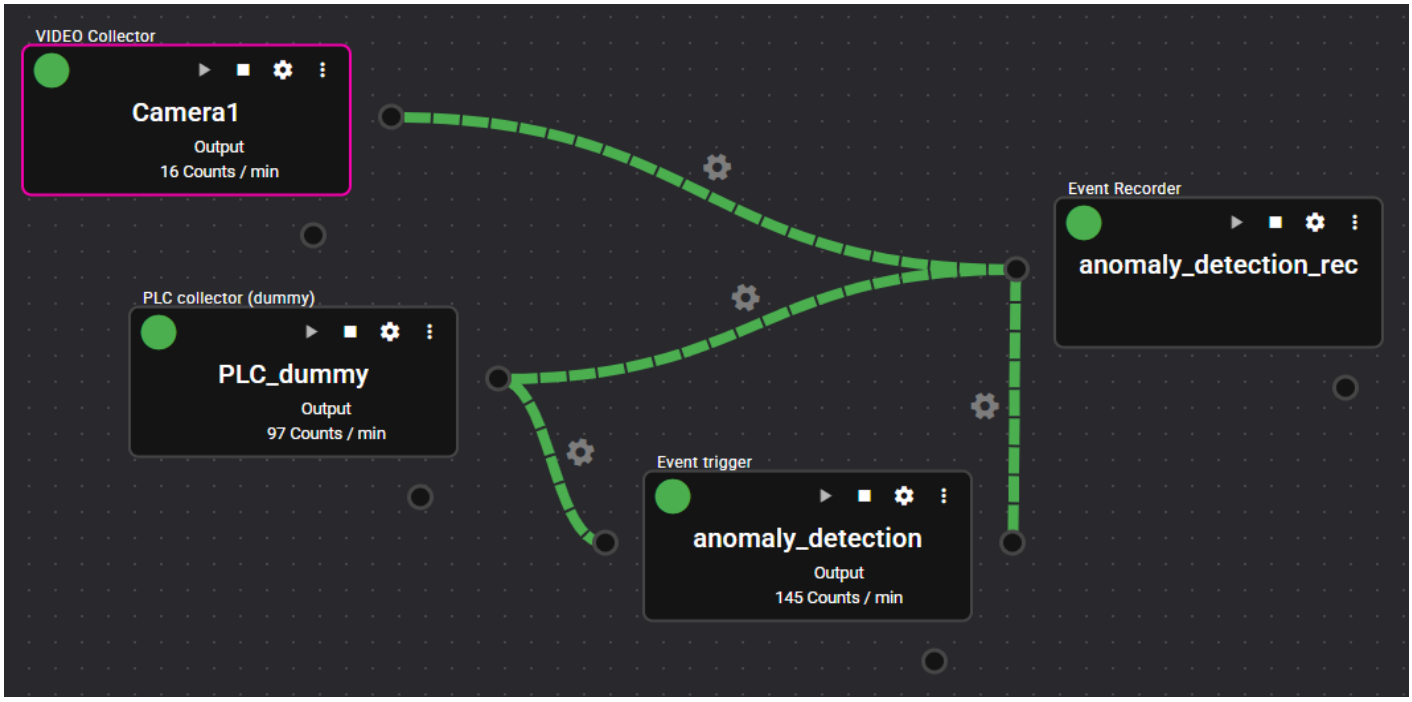
i-Trigger가 가동하고 있는 머신의 스토리지에 여유 용량이 없는 경우, 라이브 재생은 할 수 없습니다.

11.6 관련 컴포넌트

본 절에서는, i-Trigger에서 사용하는 컴포넌트에 대해 설명합니다.

11.6.1 VIDEO 컬렉터

RTSP로 송출되고 있는 네트워크 카메라의 영상을 수신하여, 녹화 파일을 저장하는 컴포넌트입니다.
녹화 파일은 지정 디렉터리 내에 ts 파일로서 저장되는 것 외에, 옵션으로 mp4 형식으로의 출력도 가능합니다.
저장된 동영상 파일은 일정 기간 유지되고 있으며, 그동안은 뷰어 화면에서 재생할 수 있습니다.
이 컴포넌트는, Proxy Server를 경유한 통신에는 대응하고 있지 않습니다.



11.6.1.1 기본 정보

AI 번역본

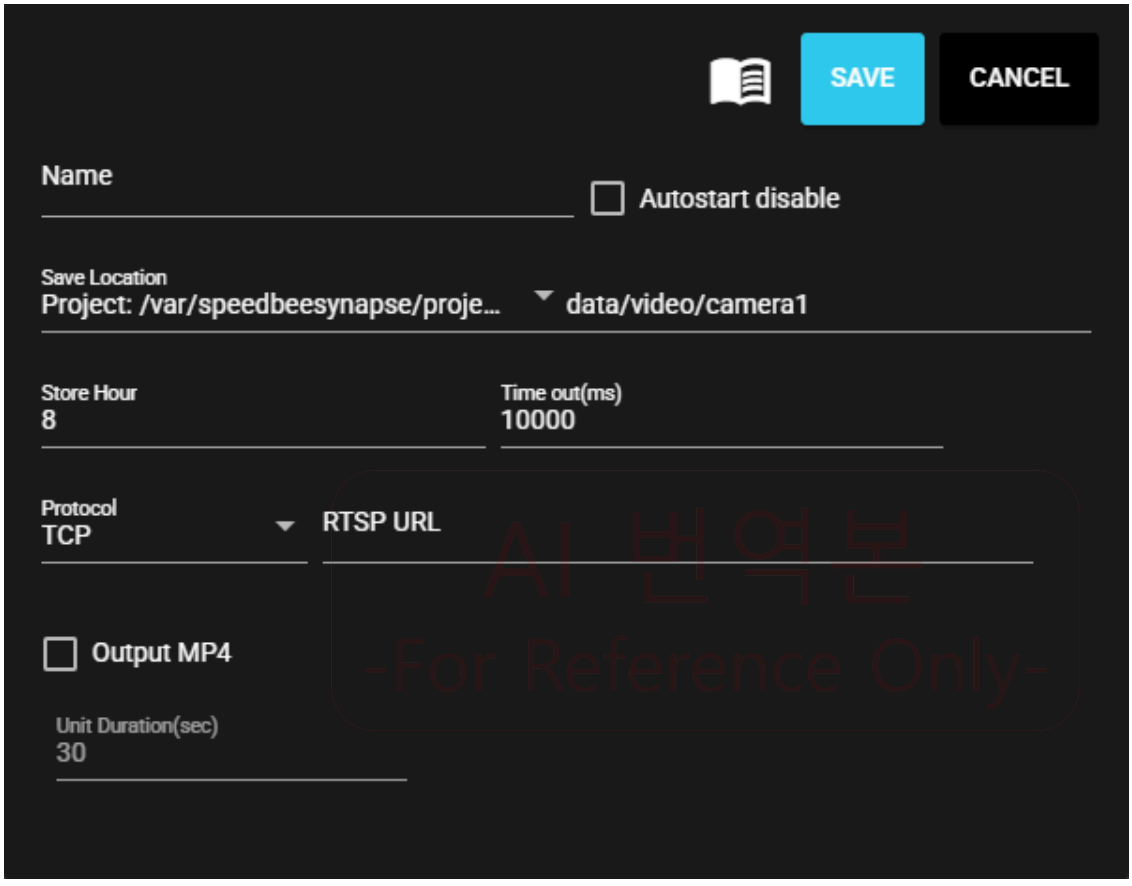
항목	설명
베이스명	VIDEO 컬렉터
설명	네트워크 카메라의 영상을 수신하여, 지정 디렉터리에 TS 파일（MPEG2-TS）을 저장합니다.
소비 포인트	2

항목	설명
포트	입력 : 0 출력 : 1 수신한 영상을 녹화 데이터로서 TS 파일에 출력합니다. 이 파일을 뷰어로 재생할 수 있습니다. 출력하는 데이터의 상세는,「 출력 데이터 항목 」을 참조하십시오. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「 에러 코드 목록 」을 참조하십시오.

11.6.1.2 설정 항목

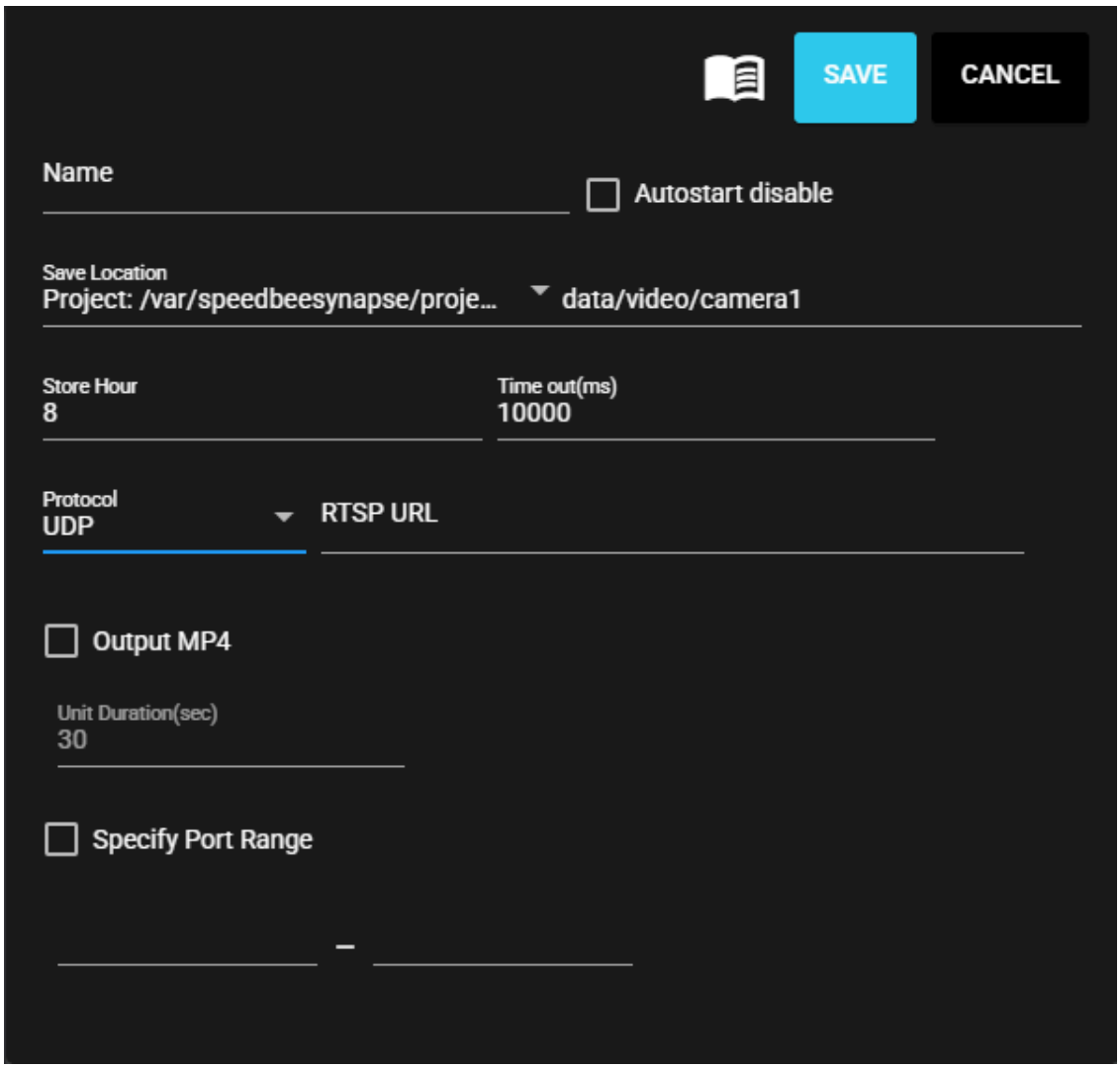
VIDEO 컬렉터의 설정 항목은, 다음과 같습니다.

- 프로토콜「 TCP 」선택 시



AI 번역본
-For Reference Only-

- 프로토콜「 UDP 」선택 시



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우
저장 위치	동영상 파일을 저장하는 디렉터리. 베이스 디렉터리와, 거기에서의 상대 경로를 지정합니다. 베이스 디렉터리는 이하에서 선택 가능합니다. • 프로젝트 경로（프로젝트 아래에 저장됩니다. 프로젝트를 복제하거나 한 경우, 새로운 프로젝트의 아래가 저장 위치가 됩니다.） ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다. • 임의의 경로（통상은 베이스 디렉터리에 대한 상대 경로를 지정합니다만, 임의의 절대 경로를 직접 지정합니다. 프로젝트를 복제하거나 해도 지정한 경로가 저장 위치가 됩니다.）
저장 시간	동영상 파일을 유지하는 기간. 시간 단위로 지정 ※저장 시간을 지나면 동영상은 자동 삭제됩니다

AI 번역본
-For Reference Only-

항목	설명
타임아웃(밀리초)	통신이 타임아웃할 때까지의 시간을 밀리초 단위로 입력
프로토콜	트랜스포트에 사용하는 프로토콜 • TCP • UDP ※일부 환경에서는, 이 설정을 UDP로 변경/편집할 수 없습니다.
RTSP URL	네트워크 카메라의 라이브 스트림 URL
MP4를 출력한다	유효하게 하면, 일정 간격으로 MP4의 동영상 파일을 생성합니다. ※무효인 경우는, TS 파일만이 출력됩니다
녹화 간격	「 MP4를 출력한다 」가 유효한 경우에, MP4의 1파일당 녹화 시간을 초 단위로 지정 ※동영상의 길이는 지정한 설정과 완전히 일치하지 않는 경우가 있습니다
포트 범위를 지정한다	「 UDP 」가 선택되어 있는 경우에, 데이터 수신용의 UDP 포트를 범위로 지정

11.6.1.3 출력 데이터 항목

VIDEO 컬렉터는, 다음의 데이터를 출력합니다.

항목	컬럼명	데이터형	설명
TS 파일	$ss_movie_file	FILE	TS 파일（MPEG2-TS）을 출력하는 컬럼. 미디어 타입은 "video/MP2T" 고정
컨테이너 파일	container_file	FILE	「 MP4를 출력한다 」가 유효한 경우에, MP4 파일을 출력하는 컬럼. 미디어 타입은 "video/mp4" 고정

11.6.1.4 테스트 접속

다음 순서로, 설정한 카메라에 대해 테스트 접속을 수행할 수 있습니다.

1. 등록한 VIDEO 컬렉터의 설정 화면을 엽니다.
2. 「 테스트 」를 누릅니다.



3. 설정한 카메라에의 테스트 접속이 수행되어, 정상적으로 접속할 수 있는 경우에, 카메라의 영상이 라이브 재생됩니다.



카메라의 테스트 접속에 관한 주의점

- 카메라에 접속할 수 없는 경우

RTSP URL 등이 올바른지 확인하십시오. 또한, 이용하고 계신 네트워크 카메라가 사용할 수 있는 카메라인지 확인하십시오. 대응하고 있는 카메라라도, 카메라 측의 설정이 대응하고 있는 비디오 코덱으로 되어 있지 않은 경우는, 카메라 측의 설정을 변경할 필요가 있습니다.
참고 : 사용할 수 있는 카메라에 대해
또한, 타임아웃의 설정을 늘리면, 해소되는 경우가 있습니다.

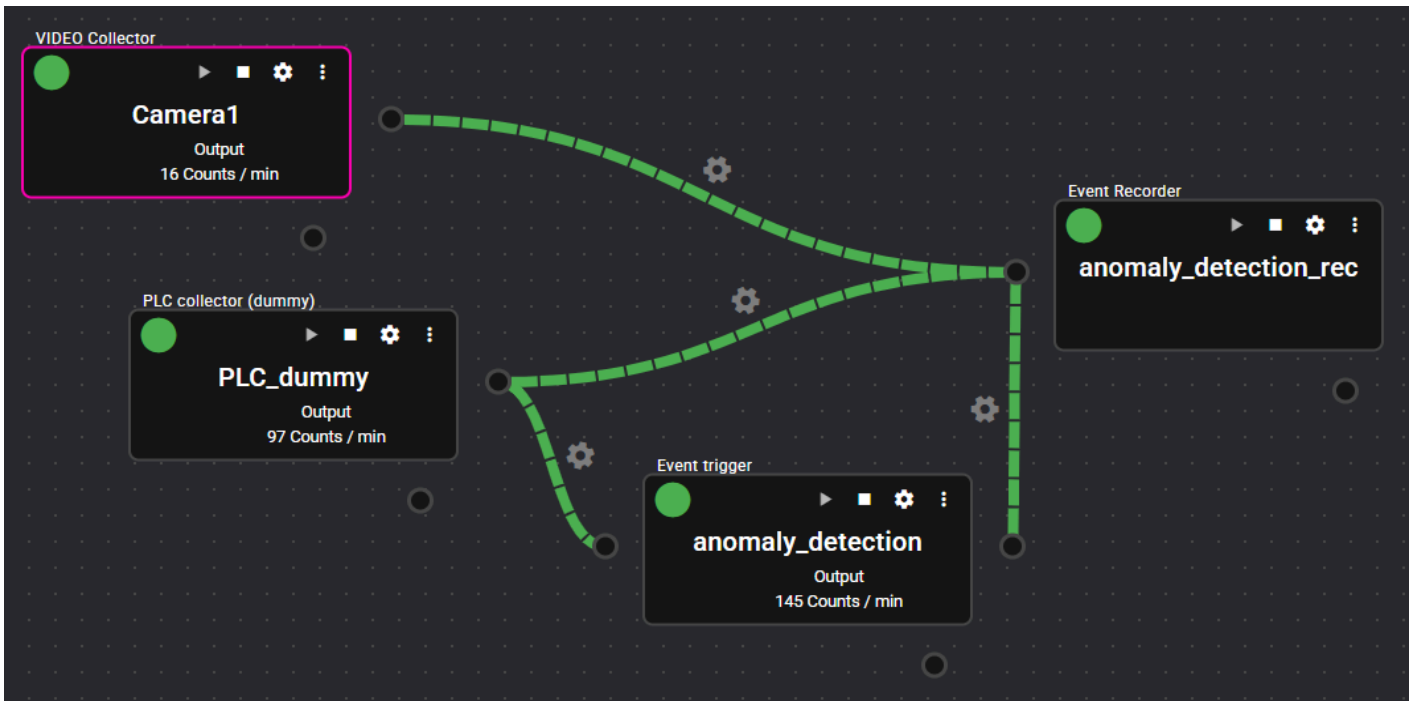
- 동시 접속 수에 대해

카메라에 따라, 동일한 스트리밍에 대해, 동시에 접속할 수 있는 접속 수는 다릅니다. 저사양의 카메라에서는,
1접속만밖에 동시 재생할 수 없는 경우도 있기 때문에, 카메라의 취급 설명서나 카메라의 제품 홈페이지 등을 확인하십시오.

11.6.2 Event Recorder

이벤트 구간의 데이터를 모아, 뷰어 화면의 이벤트 녹화 기능에서 참조 가능한 JSON 형식으로 출력하는 컴포넌트입니다.

VIDEO 컬렉터가 입력 포트에 접속되어 있는 경우는, 구간 내의 동영상 파일도 수집됩니다.
이벤트 구간의 판정은 이벤트 트리거 컴포넌트에서 수행하기 때문에, 입력 포트에 이벤트 트리거를 접속할 필요가 있습니다.



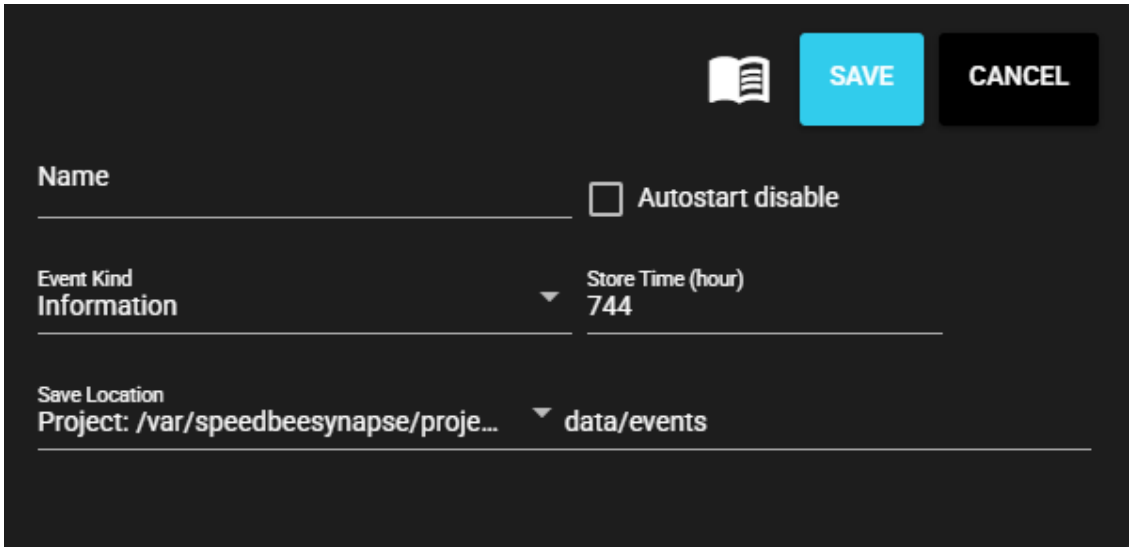
11.6.2.1 기본 정보

항목	설명
베이스명	Event Recorder
설명	이벤트 검출 후의 데이터를 모아, 수집한 데이터를 JSON으로 출력합니다. 같은 구간에, 동영상 파일이 있으면 마찬가지로 정리합니다. 지정 디렉터리에 수집한 데이터를 저장함으로써, 뷰어로 재생할 수 있습니다.
소비 포인트	2
포트	입력 : 1 다른 컴포넌트에서 데이터를 받아들입니다. 출력 : 1 기록 대상이 되는 구간의 데이터를 JSON 파일에 출력합니다. 출력하는 데이터의 상세는,「출력 데이터 항목」을 참조하십시오. 시스템 : 1 컴포넌트의 가동 상황이나 에러 정보를 출력합니다. 출력하는 에러 정보의 상세는,「에러 코드 목록」을 참조하십시오.

AI 번역본
-For Reference Only-

11.6.2.2 설정 항목

Event Recorder의 설정 항목은, 다음과 같습니다.



항목	설명
명칭	컴포넌트의 이름을 입력 ※다른 컴포넌트명과 중복될 수는 없습니다.
자동 기동 무효	컴포넌트의 자동 기동을 무효로 하는 경우
분류	이벤트에 대해 설정할 수 있는 분류. 이벤트의 내용에 맞추어, 임의로 지정합니다. • 인포메이션 • 경고 • 에러
저장 시간	기록한 데이터 및 동영상을 유지하는 기간. 시간 단위로 지정 ※저장 시간을 지나면 동영상은 이벤트 데이터 매니저에 의해 삭제됩니다
저장 위치	이벤트 검지한 구간의 데이터 및 동영상을 저장하는 디렉터리. 베이스 디렉터리와, 거기에서의 상대 경로를 지정합니다. 베이스 디렉터리는 이하에서 선택 가능합니다. • 프로젝트 경로 (프로젝트 아래에 저장됩니다. 프로젝트를 복제하거나 한 경우, 새로운 프로젝트의 아래가 저장 위치가 됩니다.) ※일부 환경에서는, 이 설정을 변경/편집할 수 없습니다. • 임의의 경로 (통상은 베이스 디렉터리에 대한 상대 경로를 지정합니다만, 임의의 절대 경로를 직접 지정합니다. 프로젝트를 복제하거나 해도 지정한 경로가 저장 위치가 됩니다.)

AI 번역본
For Reference Only

11.6.2.3 출력 데이터 항목

Event Recorder는, 다음의 데이터를 출력합니다.

항목	컬럼명	데이터형	설명
이벤트 파일	event_file	FILE	이벤트 기록의 JSON 파일을 출력하는 컬럼. 미디어 타입은 "application/json" 고정

11.6.3 이벤트 트리거

SpeeDBee Synapse 본체에 포함되는 컴포넌트입니다. 자세히는,「 이벤트 트리거 」를 참조하십시오.

11.6.4 이벤트 데이터 매니저

SpeeDBee Synapse 본체에 포함되는 컴포넌트입니다. 자세히는,「 이벤트 데이터 매니저 」를 참조하십시오.

AI 번역본
-For Reference Only-

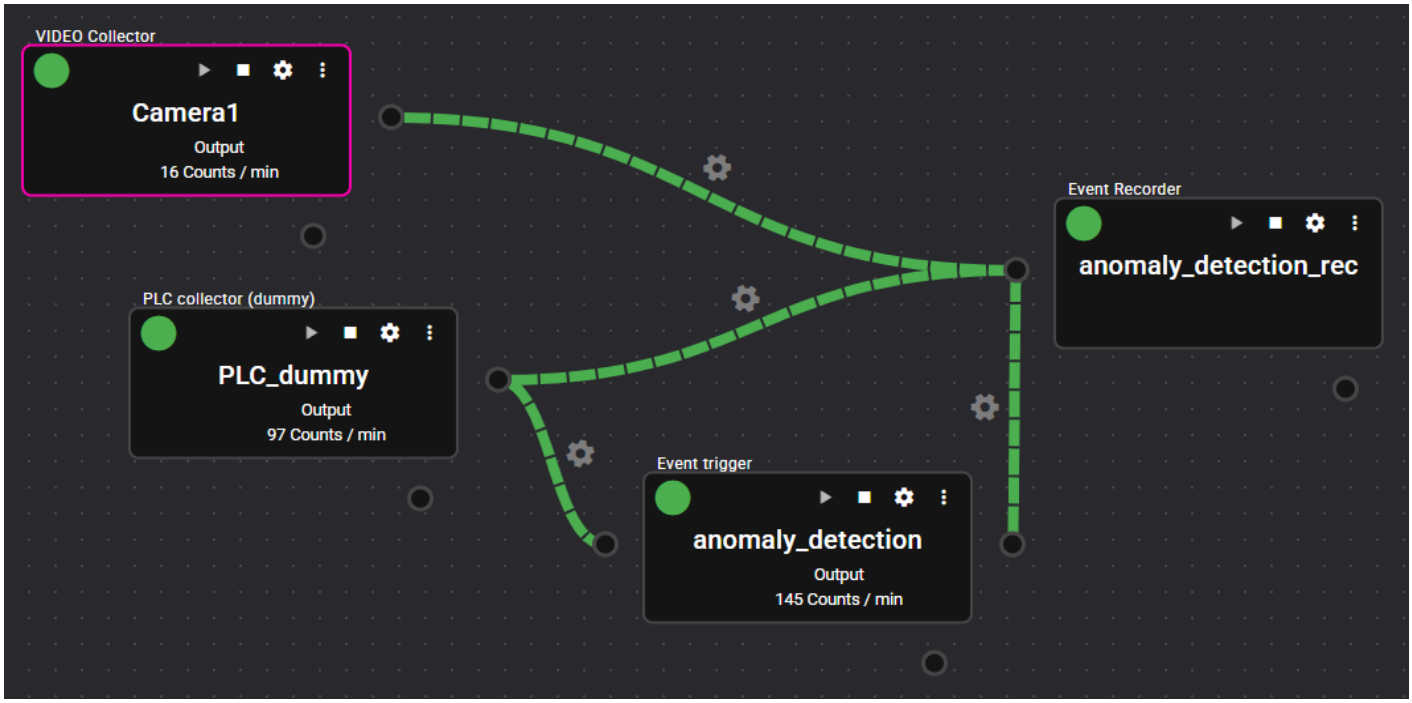
11.7 튜토리얼

본 절에서는, i-Trigger의 간단한 이용 순서에 대해, 튜토리얼 형식으로 해설합니다.

11.7.1 상정하는 유스 케이스

본 튜토리얼에서는, 다음과 같은 유스 케이스로, 이상 검지를 수행합니다.

PLC로부터의 수집 데이터로 이상을 검지하면, 그 전후의 동영상 및 수집한 데이터를 뷰어로 확인한다（※여기에서는, PLC 컬렉터로 간주한 커스텀 컴포넌트를 사용합니다）



1. PLC에서 다음 데이터를 수집한다
   - 이상 검지 플래그
   - 압력
2. 1대의 네트워크 카메라로 영상을 기록한다
3. 이상 검지 플래그가 ON이 되면 이벤트의 기록을 개시
   - 이상 검지 플래그 검지 전 5초, 이상 검지 플래그 OFF부터 뒤 5초를 기록한다
   - 뷰어로 표시하는 데이터는, PLC에서 취득한 이상 검지 플래그, 압력과 그 구간의 영상

AI 번역본
-For Reference Only-

11.7.2 커스텀 컴포넌트의 설정（의사 PLC 컬렉터）

최초에, 이상 검지 및 압력의 데이터를 수집하기 위한 커스텀 컴포넌트를 설정합니다.

11.7.2.1 커스텀 컴포넌트의 코드

다음 코드를 복사하여, 메모장 등의 텍스트 에디터에 붙여넣고, PC 내에「plc_dummy.py」라는 파일명으로 저장해 주십시오. 문자 코드는「UTF-8」을 지정합니다.

```
from speedbeesynapse.component.base import HiveComponentBase, HiveComponentInfo, DataType
import time
import json


@HiveComponentInfo(uuid='12318c73-fe45-fded-150a-b95c894f3692', name='PLC 컬렉터 ( 더미 ) ',
inports=0, outports=1)
class HiveComponent(HiveComponentBase):
    def __init__(self):
        # called at creating component instance
        pass
    def __del__(self):
        # called at deleting component instance
        pass

    def premain(self, param):
        # called before main()
        # 의사 데이터를 등록할 컬럼을 작성한다
        self.data = self.out_port1.Column('압력', DataType.INT16)
        self.flag  = self.out_port1.Column('이상 검지 플래그',              DataType.BOOLEAN)

    def postmain(self, param):
        # called after main() finished
        pass

    def main(self, param):
        value = False
        count = 0
        while self.is_runnable():
            ts = self.get_timestamp()
            pressure = 50
            count += 1
            # count30회마다 이상 데이터를 등록한다
            if (count % 30) == 0:
                value = True
                pressure = 100
            else:
                value = False
            # 의사 데이터를 등록한다
            self.flag.insert(value, ts)
            self.data.insert(pressure, ts)

            time.sleep(1.0)

    def notify_stop(self):
        # called before the component received stop request from system
        pass
```

AI 번역본
For Reference Only

상기의 코드는, 다음 처리를 수행합니다.

- 약 30초 간격으로, 이상 검지의 플래그를 True로 하여 데이터 등록

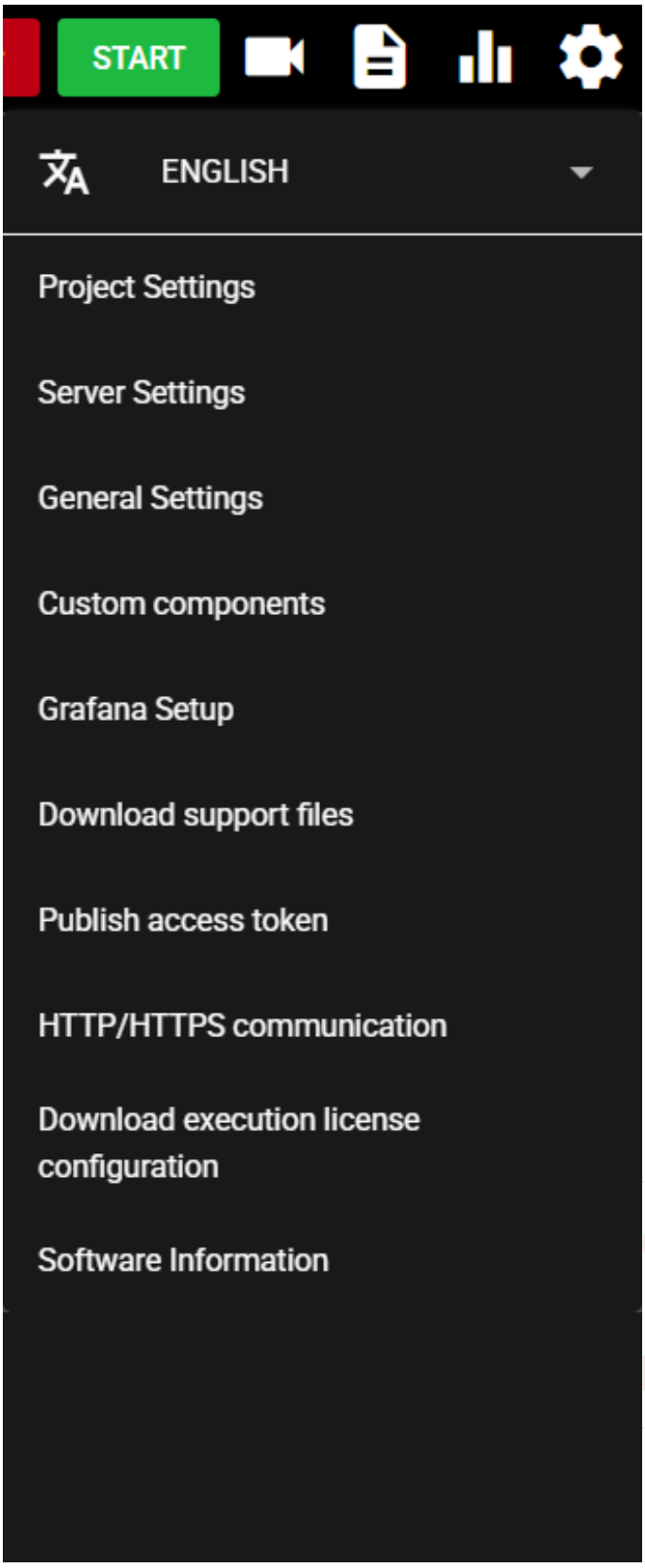
- 압력은 정상 시는 50으로 등록
- 이상 시는 압력의 값을 100으로 등록

11.7.2.2 커스텀 컴포넌트의 등록

저장한 Python의 소스 코드를 SpeeDBee Synapse에서 사용하기 위한 설정을 수행합니다.

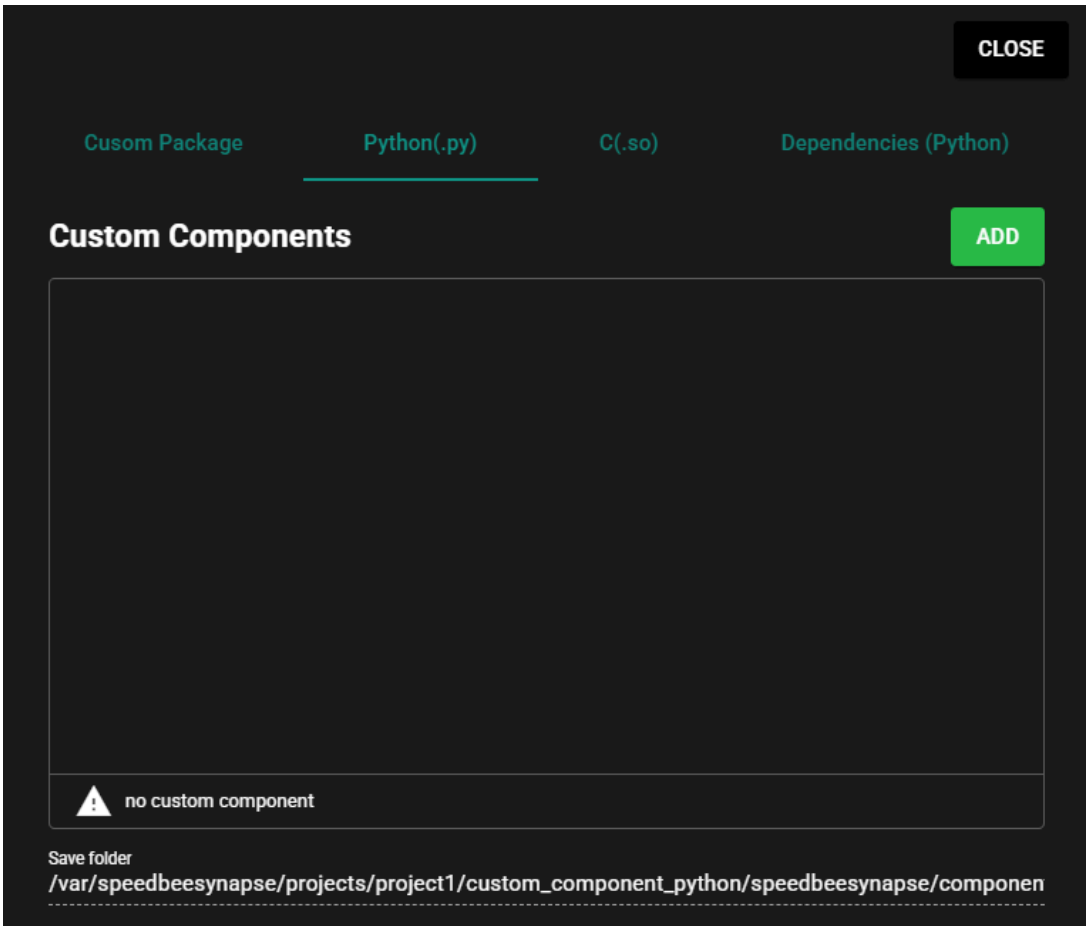
AI 번역본
-For Reference Only-

1. 「 서버 설정 아이콘 」의 「 커스텀 컴포넌트 」를 클릭합니다.

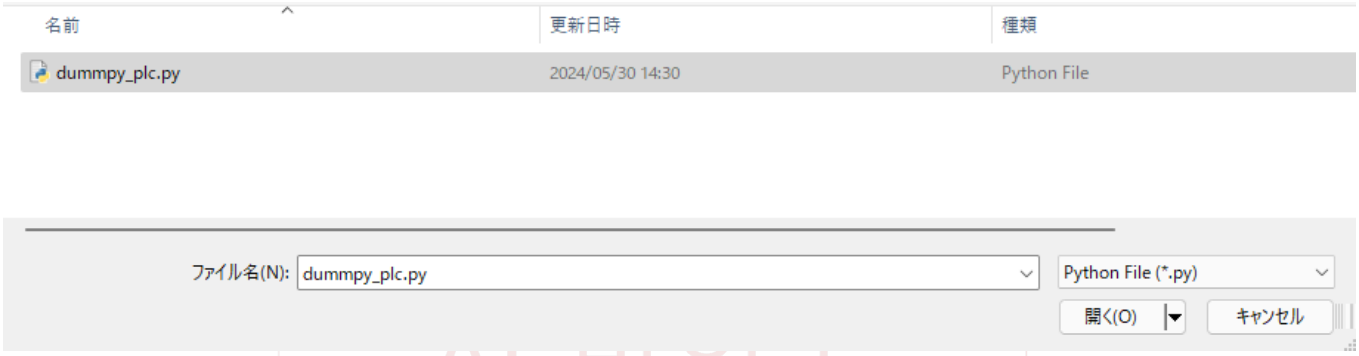


역본
nce Only-

2. 커스텀 컴포넌트의 「 추가 」를 클릭합니다.

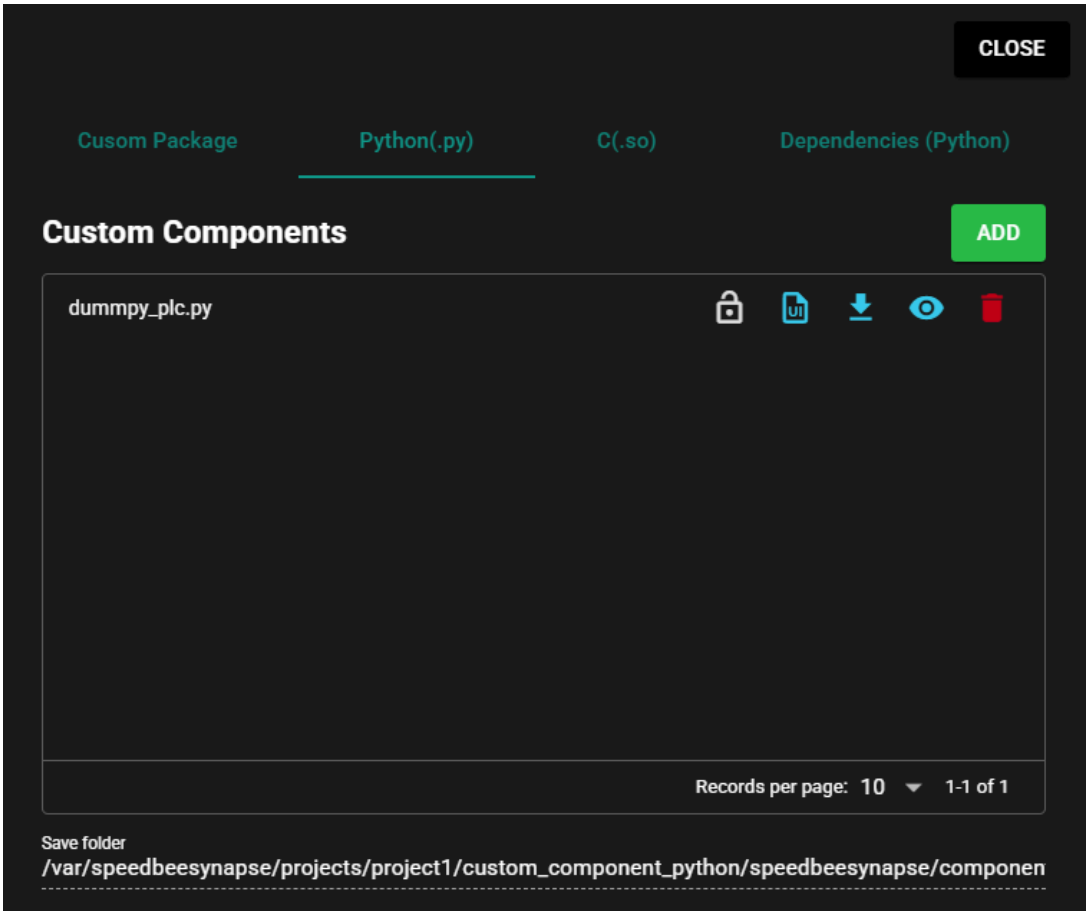


3. 조금 전 저장한 「 dummy_plc.py 」를 선택합니다.

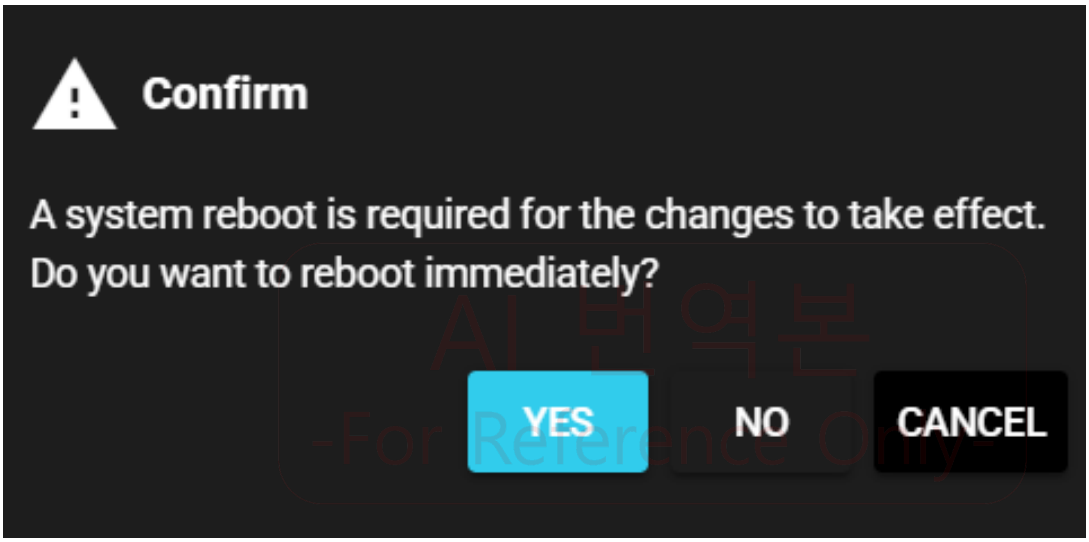


AI 번역본
-For Reference Only-

4. 커스텀 컴포넌트가 표시되는 것을 확인합니다.
「 dummpy_plc.py 」가 표시되어 있으면, 다이얼로그의 「 닫기 」를 클릭합니다.

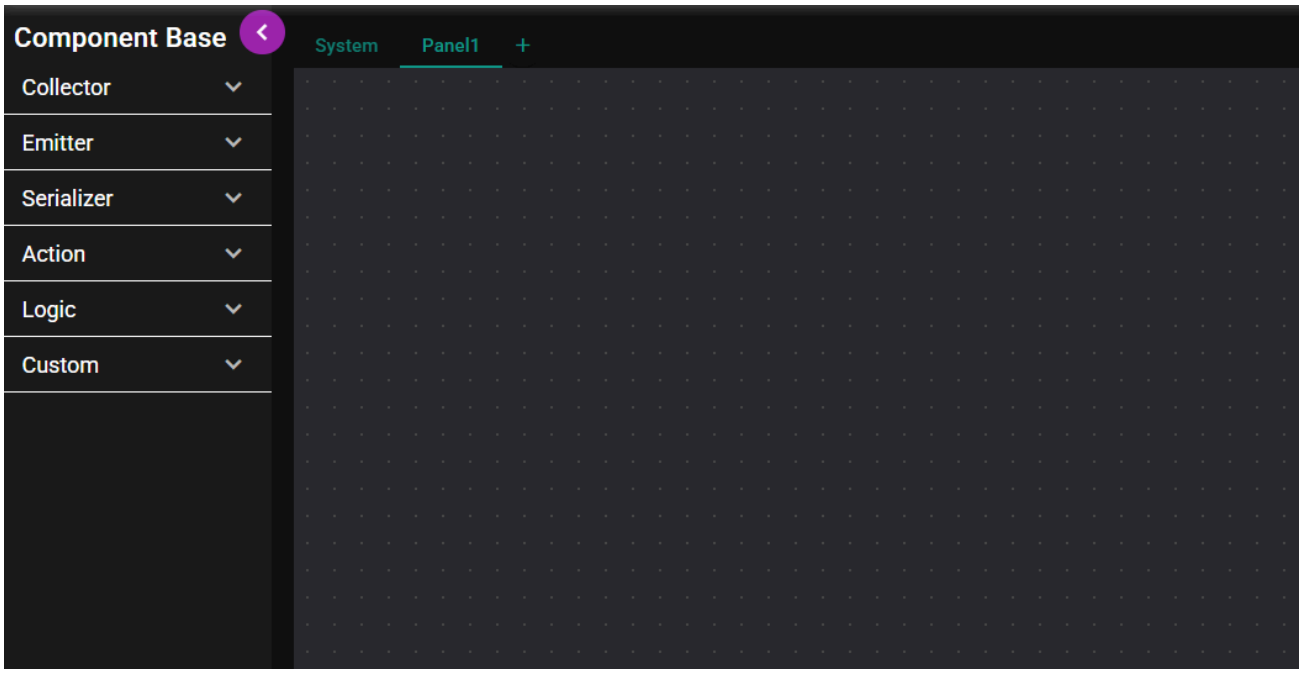


5. 재기동을 촉구하므로, 「 예 」를 클릭합니다.



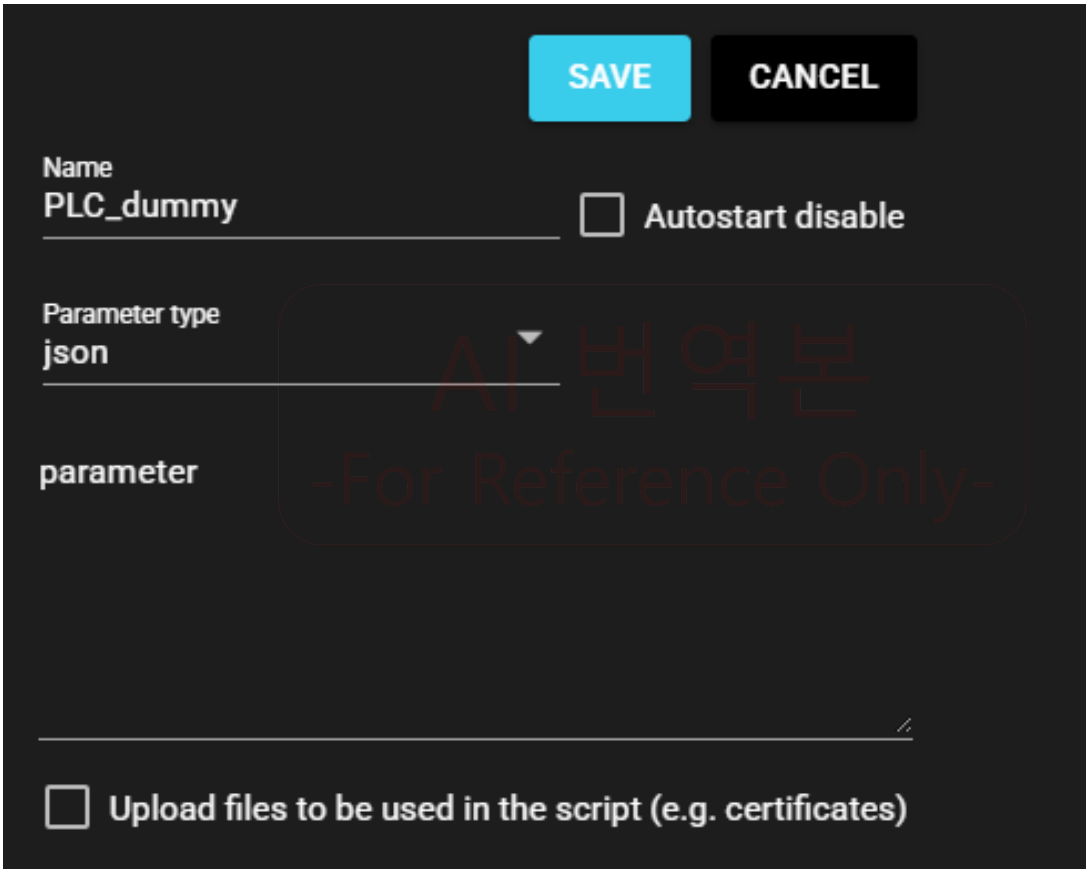
AI 번역 본
-For Reference Only-

6. 플로우 에디터 좌부의「Custom」의「PLC 컬렉터（더미）」를 에디터 중앙부로 드래그＆드롭합니다.



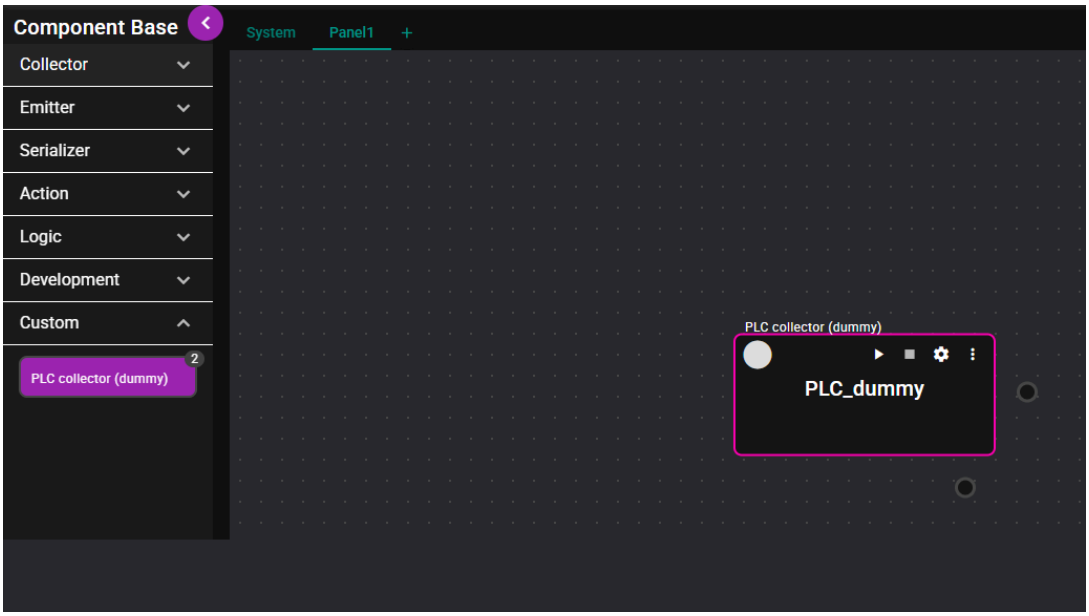
다.

7. 명칭을「PLC 더미」로 설정하고,「저장」을 클릭합니다.



AI 번역본
-For Reference Only-

8. 커스텀 컴포넌트의 준비는 완료입니다.



11.7.3 컴포넌트의 배치・설정

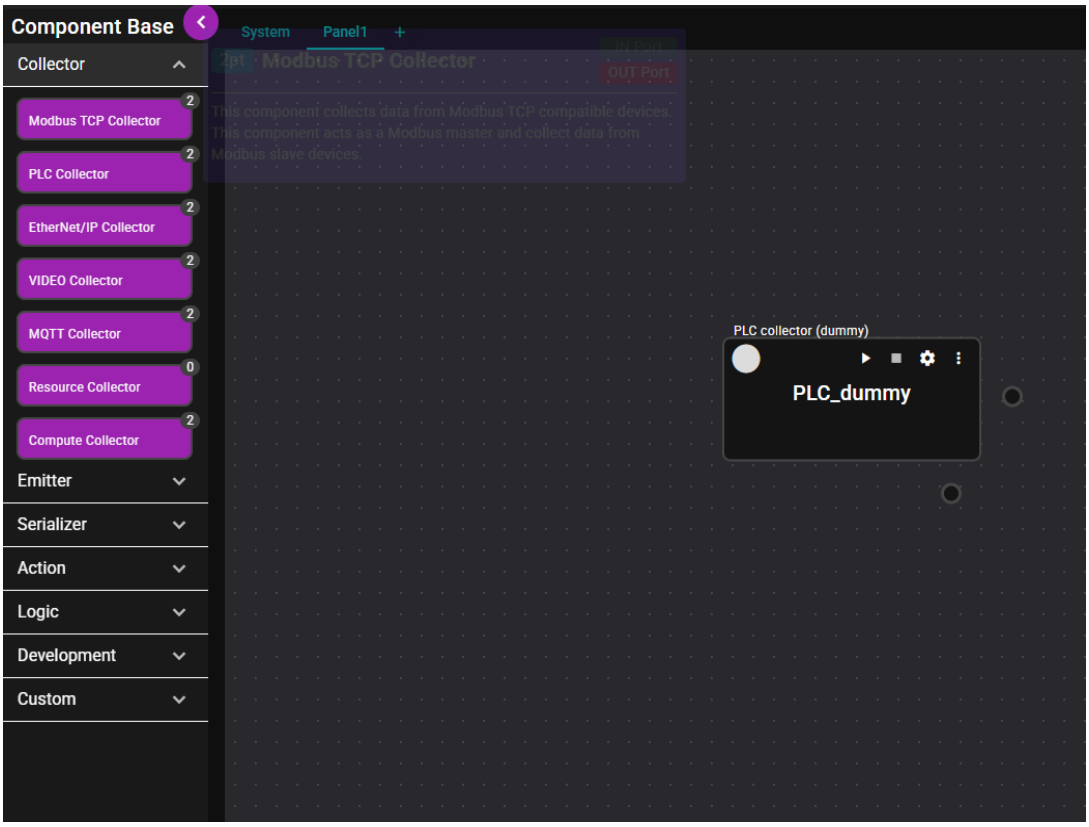
i-Trigger를 사용하기 위한 각종 컴포넌트를 설정해 갑니다.

11.7.3.1 VIDEO 컬렉터의 설정

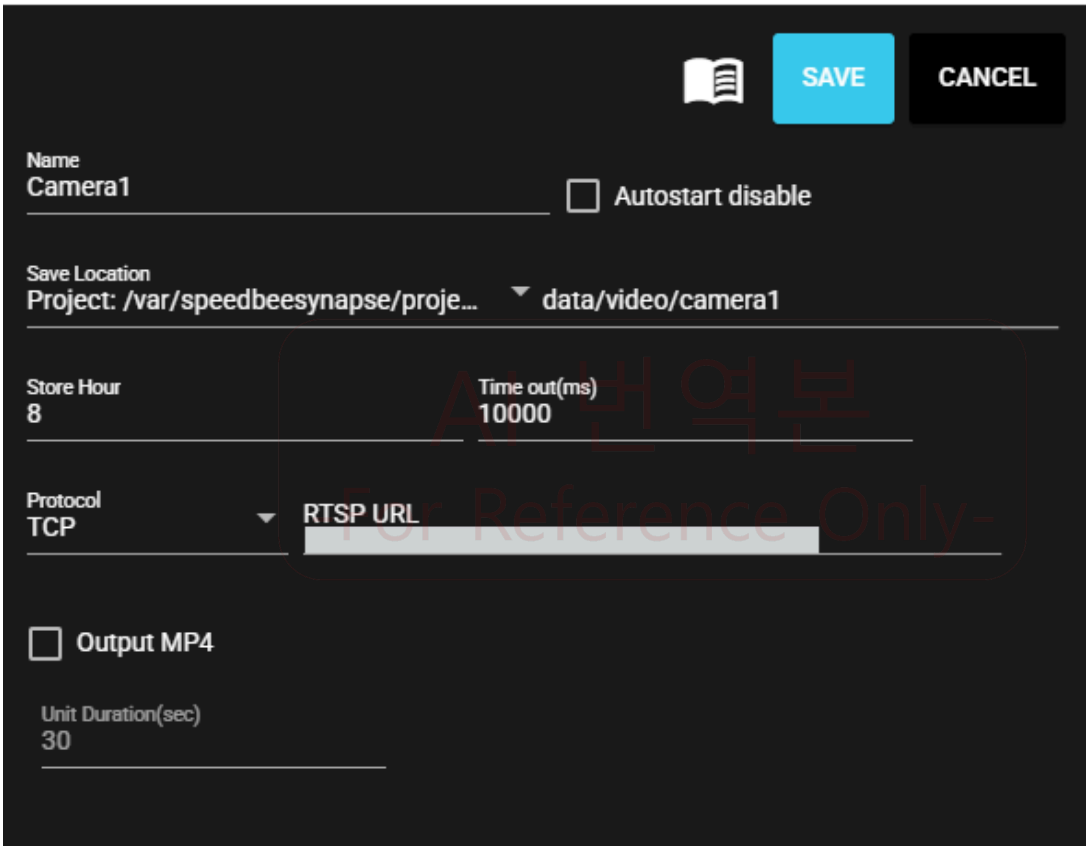
동영상을 기록하기 위해, VIDEO 컬렉터를 설정합니다.
이용할 수 있는 네트워크 카메라가 없는 경우는, 이 설정은 생략해도 상관없습니다.

AI 번역본
-For Reference Only-

1. 플로우 에디터 좌부의 「 Collector 」의 「 Camera1 」을 에디터 중앙부로 드래그＆드롭합니다.

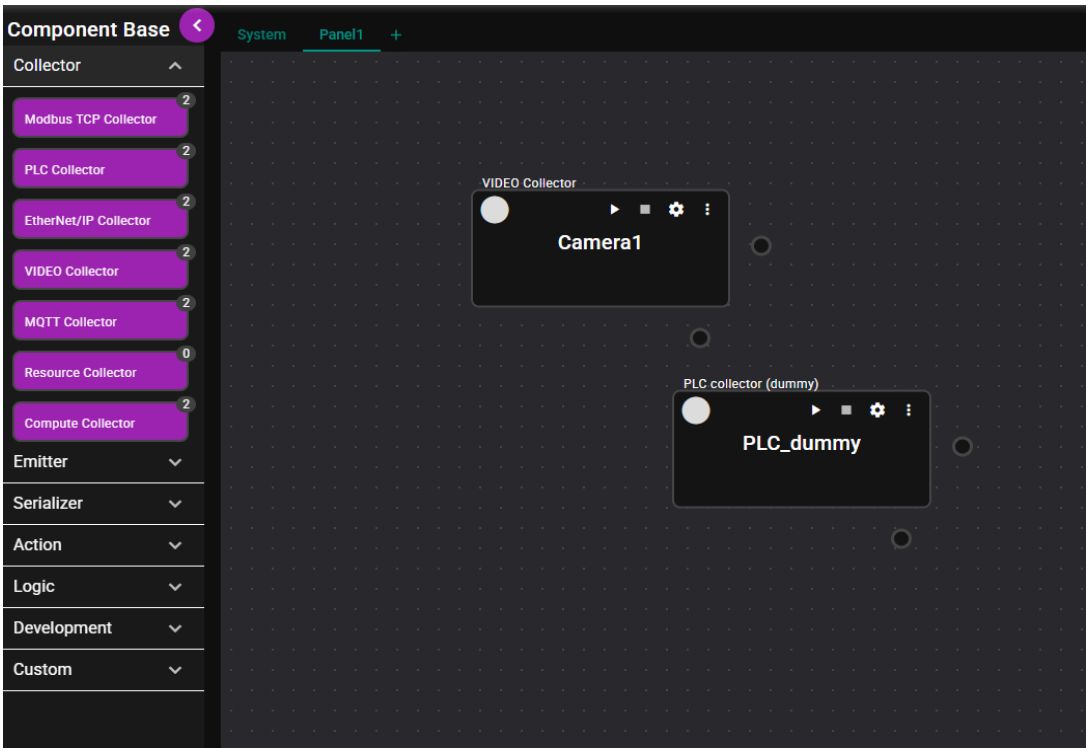


2. 명칭을 「 Camera1 」로 설정하고, 「 RTSP URL 」은, 카메라의 지정 URL을 입력합니다. 그리고, 「 저장 」을 클릭합니다.



사본 역본
For Reference Only

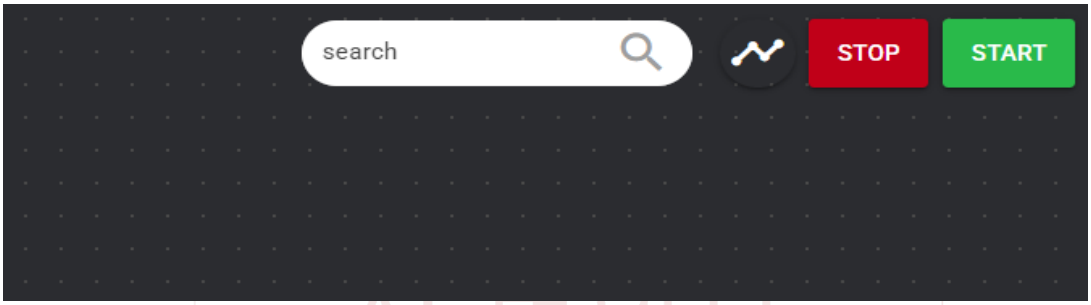
3. Video 컬렉터 「 Camera1 」의 준비는 완료입니다.



11.7.3.2 이벤트 트리거의 설정

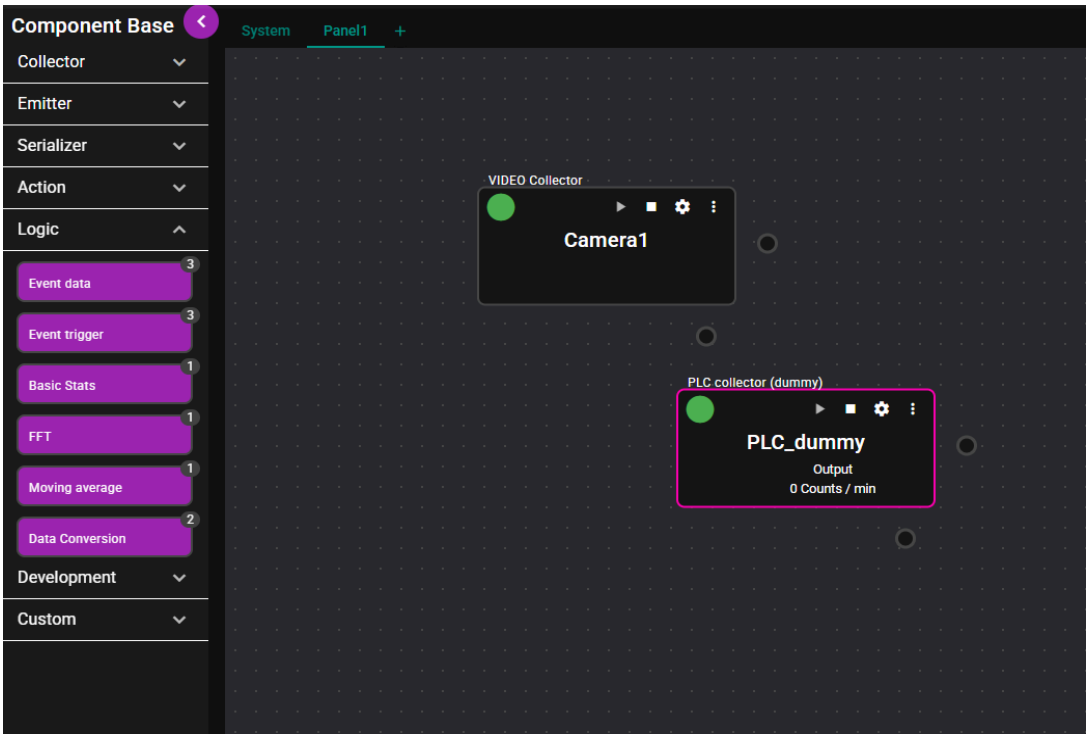
이벤트 트리거로, 이상 검지 시의 조건을 수행합니다.

1. 이벤트 트리거의 설정 전에, 「 PLC 더미 」와 「 Camera1 」은 기동합니다. 패널 우상부의 「 기동 」을 클릭합니다.

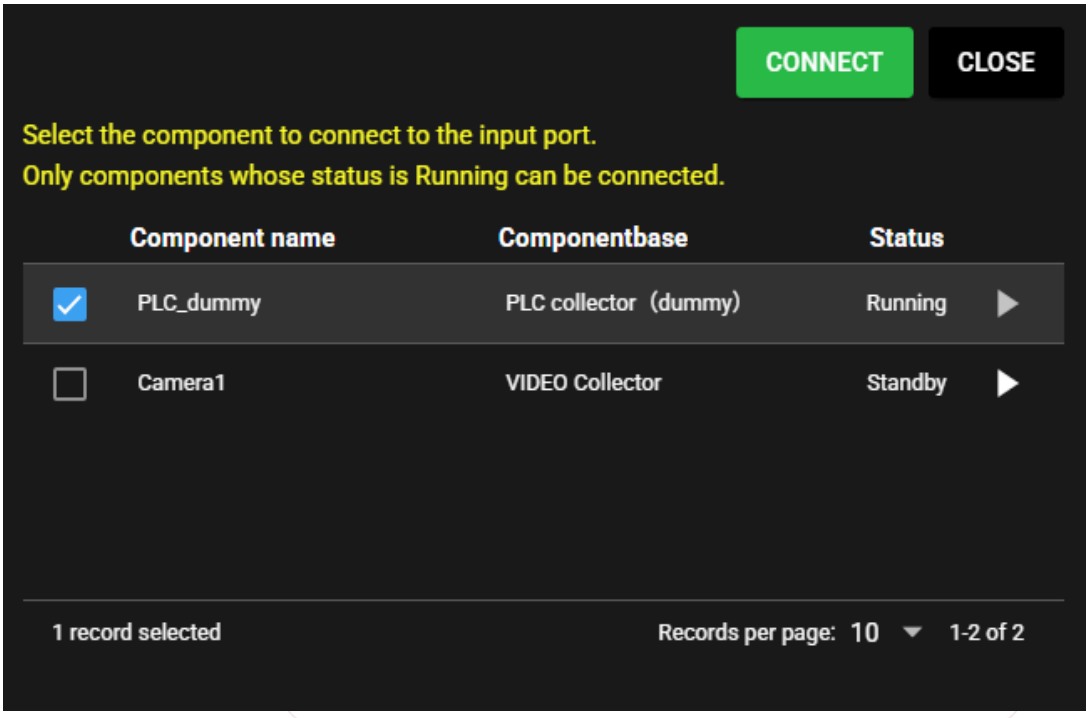


AI 번역본
-For Reference Only-

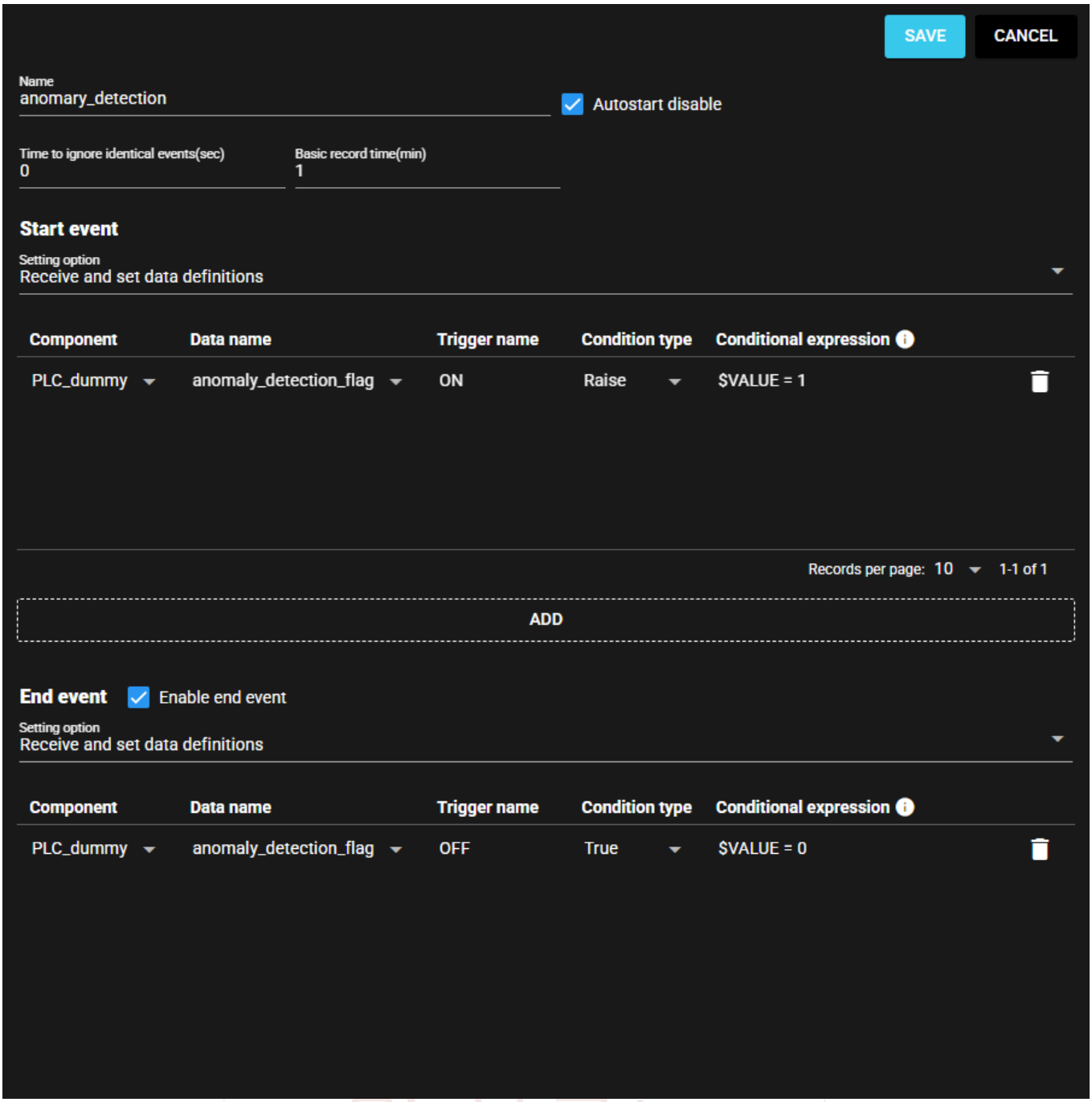
2. 플로우 에디터 좌부의 「 Logic 」의 「 이벤트 트리거 」를 에디터 중앙부로 드래그＆드롭합니다.



3. 입력 포트에 접속할 컴포넌트는, 「 PLC 더미 」에 체크하고 「 접속 」을 클릭합니다.



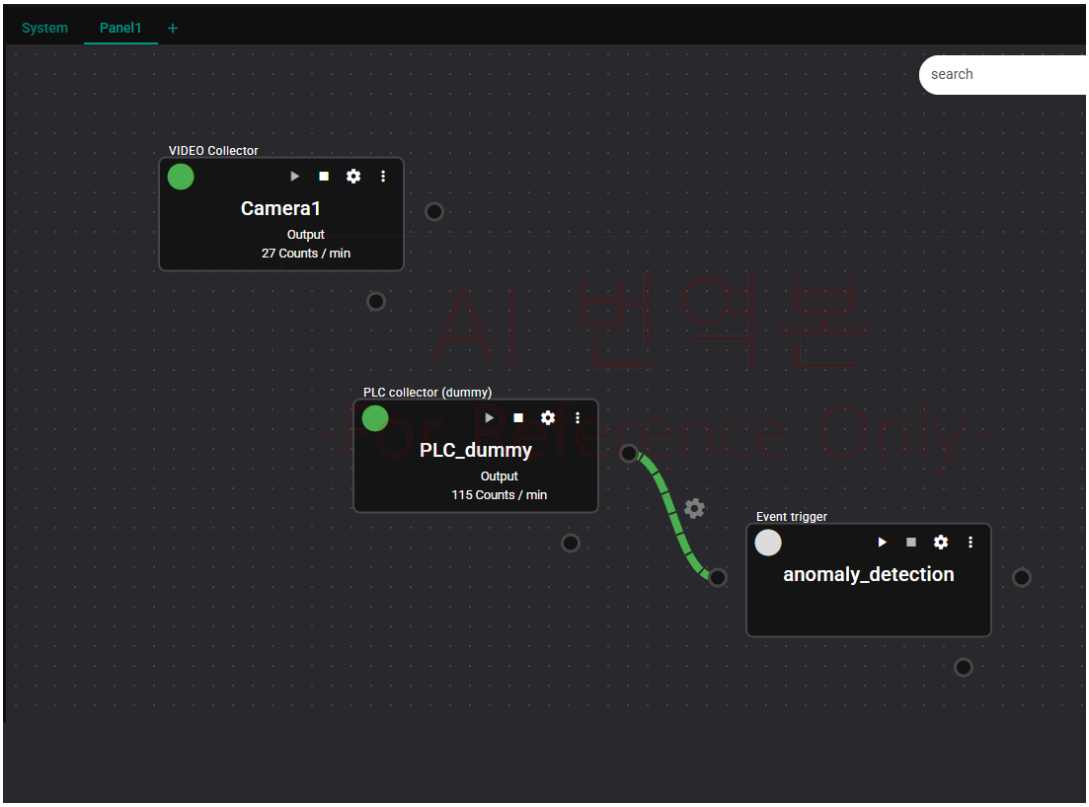
4. 명칭을「이상 검지」로 합니다. 기타 항목은, 아래를 참고로 입력하고,「저장」을 클릭합니다.



항목	값	보충
동일 이벤트를 무시하는 시간（초）	0	이번에는 무시하는 설정이 불필요하므로 0으로 한다
기본 기록 시간（분）	1	
개시 이벤트	-	-
컴포넌트	PLC 더미	

항목	값	보충
데이터명	이상 검지 플래그	
트리거명	ON	
조건 종별	Raise	
조건식	$VALUE = 1	플래그가 true가 되는 상승 조건으로 한다
종료 이벤트	-	-
컴포넌트	PLC 더미	
데이터명	이상 검지 플래그	
트리거명	OFF	
조건 종별	Raise	
조건식	$VALUE = 0	플래그가 false가 되는 하강 조건으로 한다

5. 이벤트 트리거「이상 검지」의 설정은 완료입니다.



AI 번역본
For Reference Only

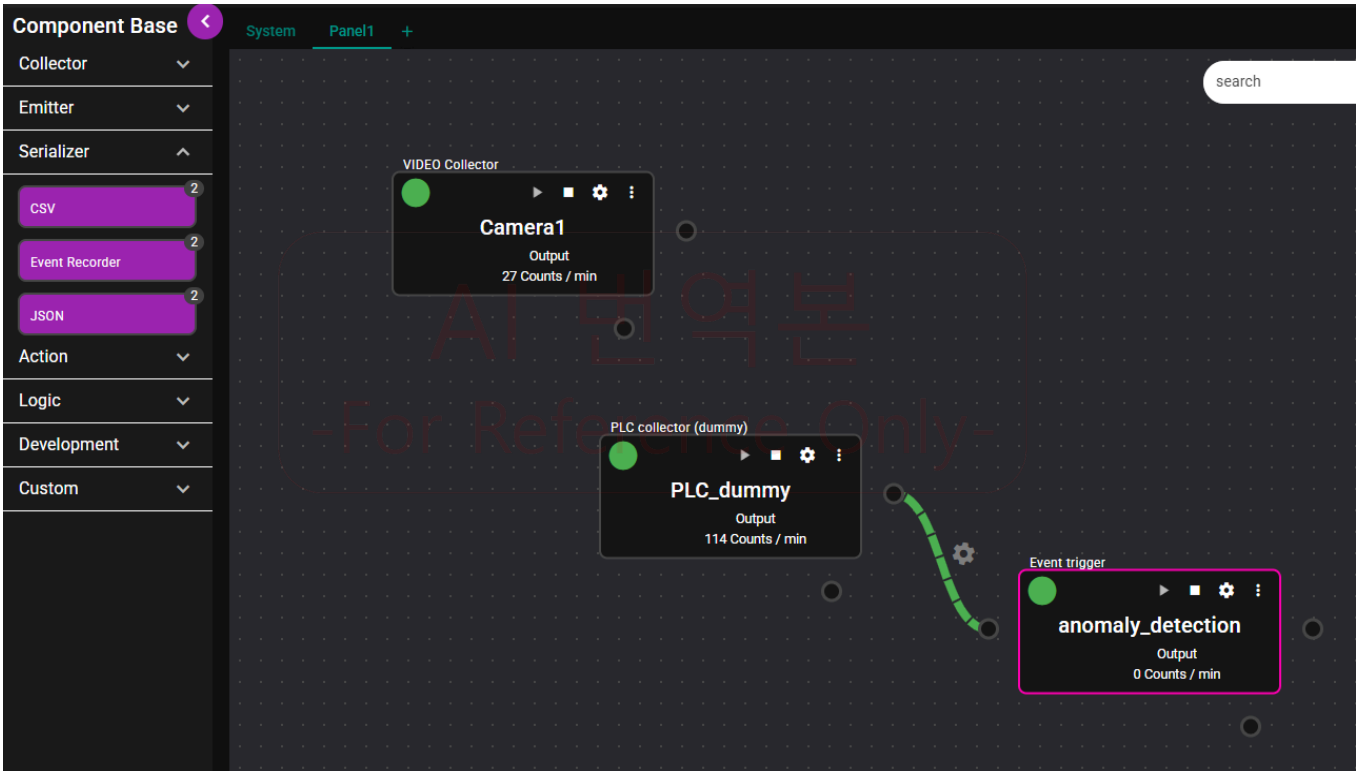
11.7.3.3 Event Recorder의 설정

마지막으로 Event Recorder의 설정을 수행합니다. 이상 발생 시의 기록 구간 등의 설정도 여기에서 수행합니다.

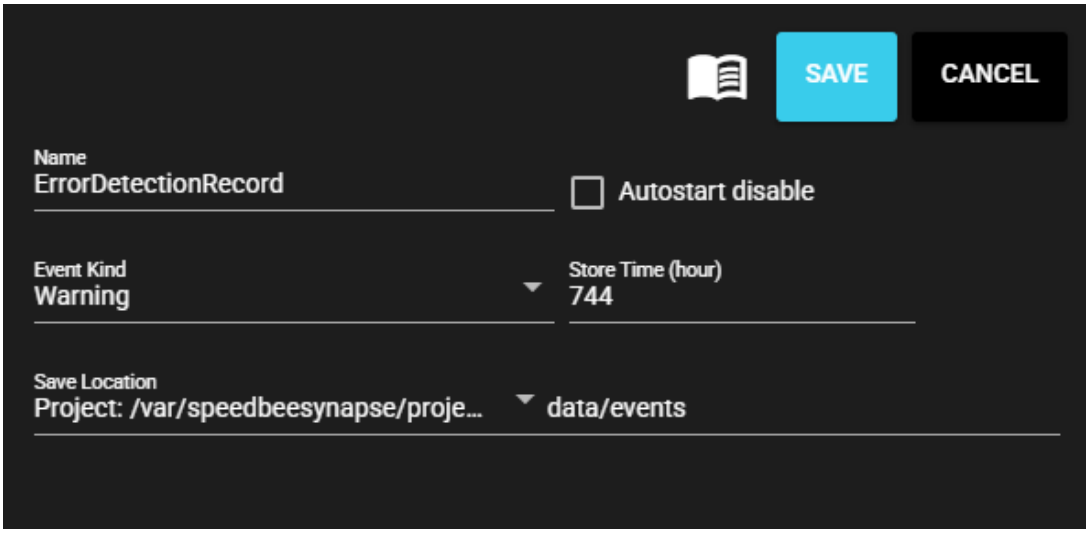
1. 이벤트 트리거「 이상 검지 」를 기동해 둡니다.「 기동 아이콘 」을 클릭합니다.



2. 플로우 에디터 좌부의「 Serializer 」의「 Event Recorder 」를 에디터 중앙부로 드래그＆드롭합니다.



3. 명칭을「이상 검지 기록」으로 설정합니다.「분류」는,「경고」를 선택합니다. 그리고,「저장」을 클릭합니다.



4.「이상 검지 기록」에,「PLC 더미」,「Camera1」,「이상 검지」의 플로우 링크를 접속합니다.



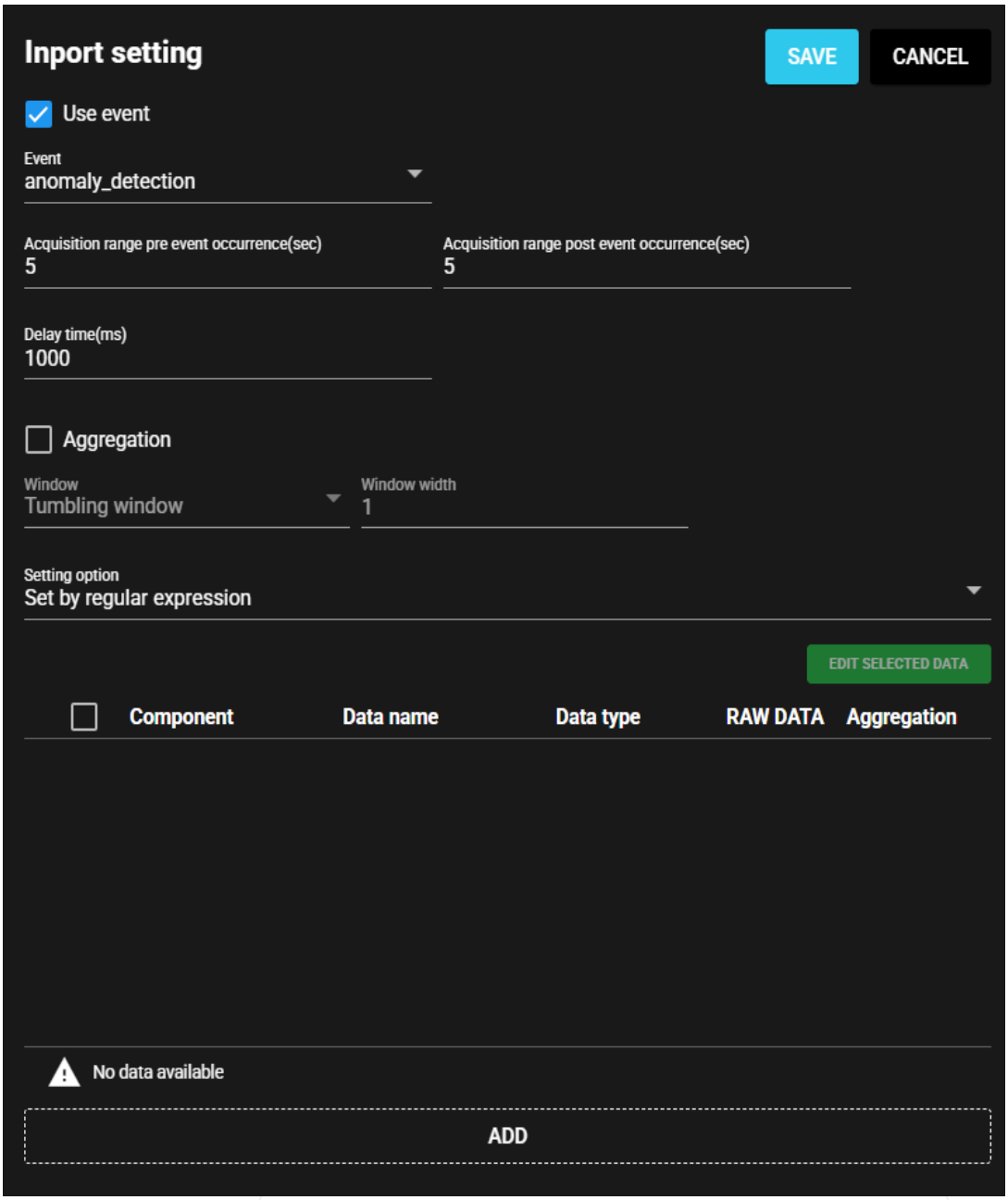
AI 번역본
-For Reference Only-

5.「 이상 검지 기록 」의 입력 포트의 설정을 엽니다.



AI 번역본
-For Reference Only-

6. 입력 포트를 설정하고,「 저장 」을 클릭합니다. 입력값은 아래를 참고로 해 주십시오.



항목	값	보충
이벤트를 사용한다	체크ON	이 설정이 없으면, Event Recorder에서 이벤트 발생 범위가 기록되지 않는다.
이벤트	이상 검지	이 설정이 없으면, Event Recorder에서 이벤트 발생 범위가 기록되지 않는다.
이벤트 발생 전의 취득 시간 （ 초 ）	5	

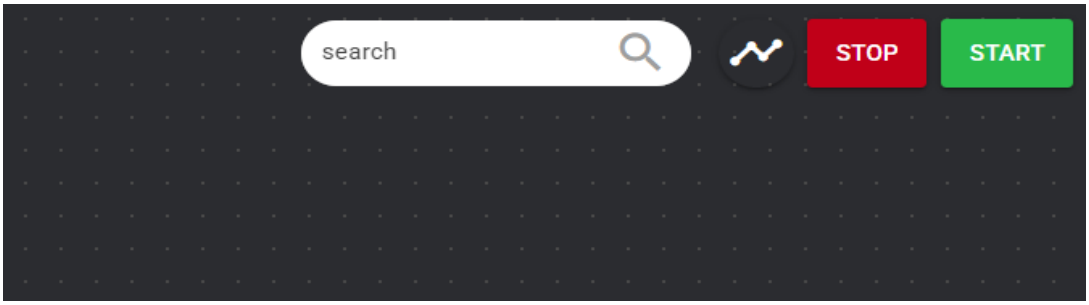
항목	값	보충
이벤트 종료 후의 취득 시간 (초)	5	

7. 이벤트의 기록의 모든 설정은 완료입니다.

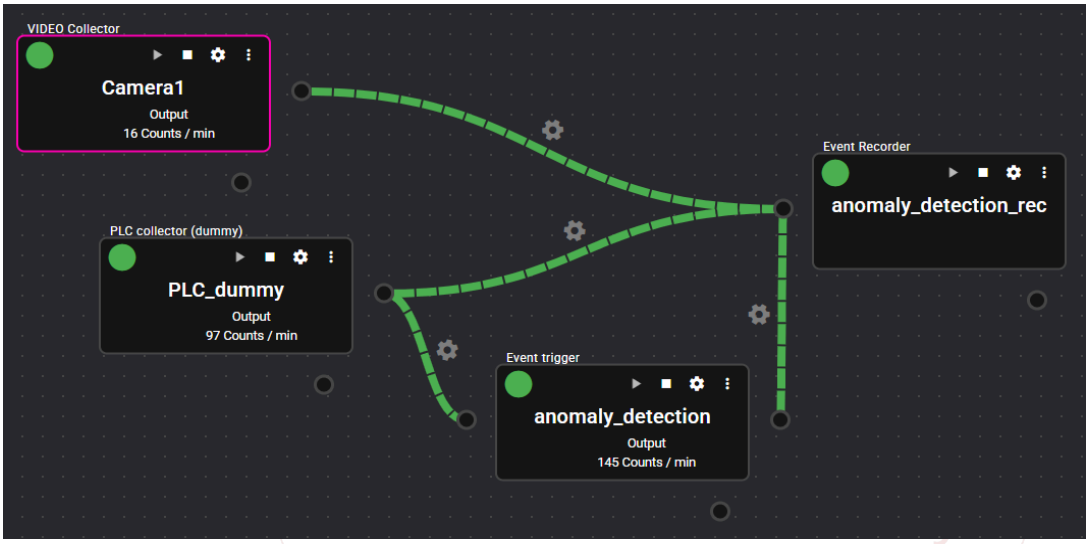
11.7.4 동작 확인

여기까지 설정한 데이터 플로우를 동작시킵니다.

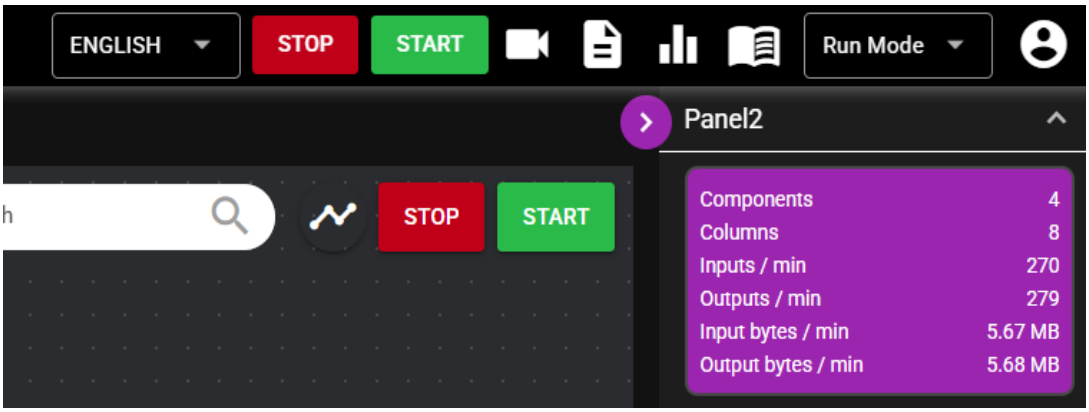
1. 설정한 컴포넌트를 기동합니다.



2. 배치한 컴포넌트가 정상적으로 기동하는 것을 확인합니다. 컴포넌트의 램프에 이상이 있으면 설정을 재검토해 주십시오.



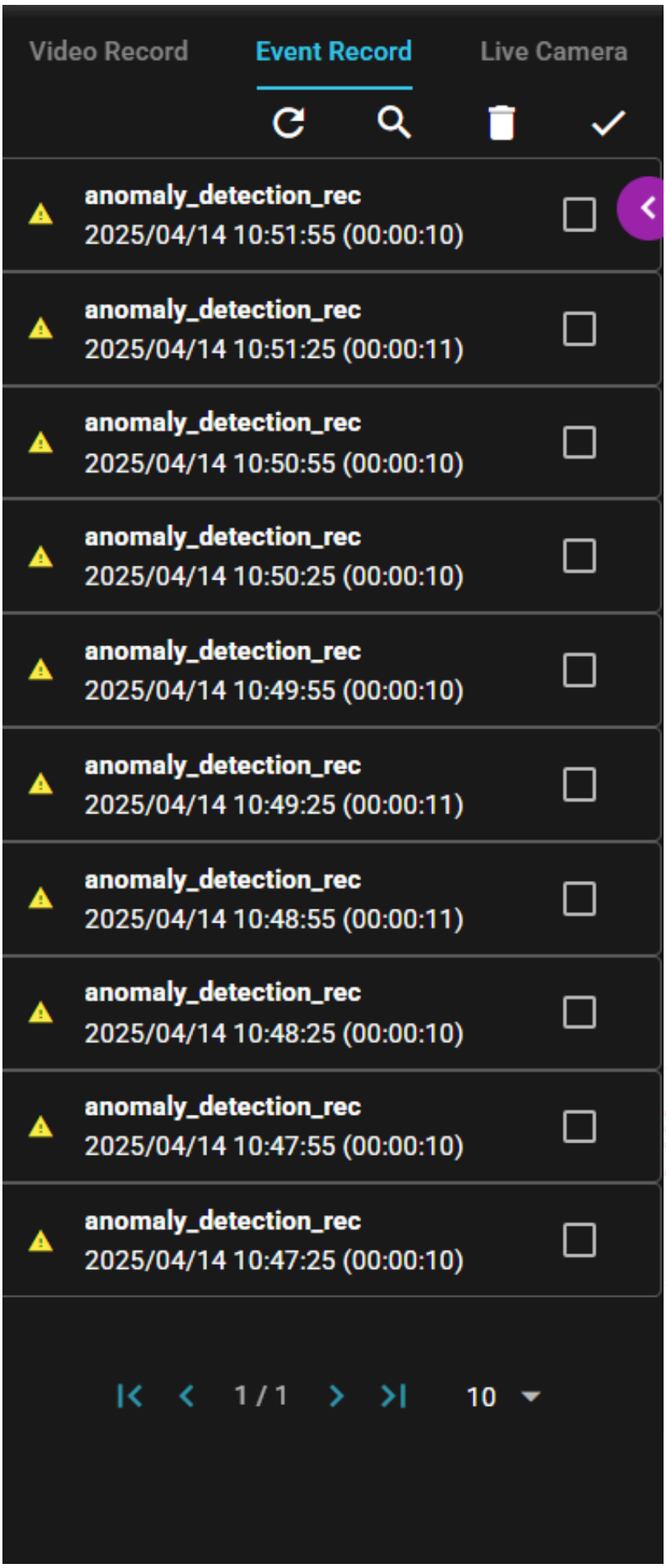
3. 수 분 이상 가동시켰으면, 뷰어를 확인합니다. 「 뷰어 아이콘 」을 클릭합니다.



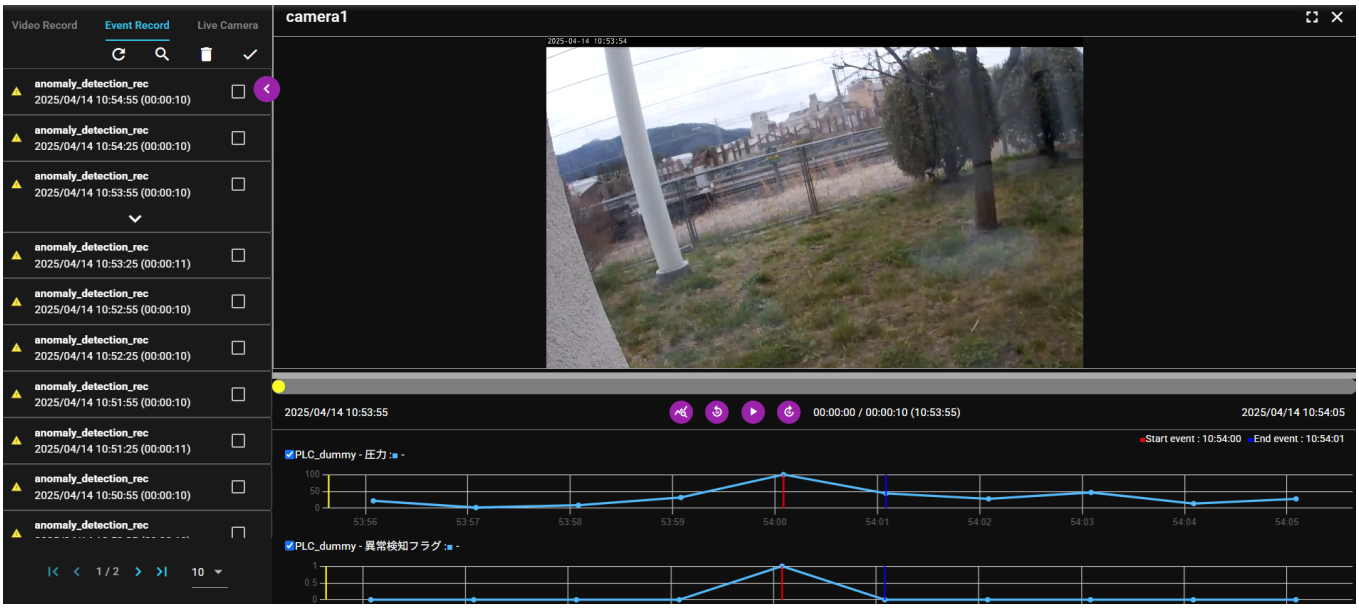
AI 번역본
-For Reference Only-

4. 뷰어 좌부에서,「 이벤트 녹화 」를 클릭합니다.「 이상 검지 기록 」의 목록이 표시되어 있는지 확인합니다.

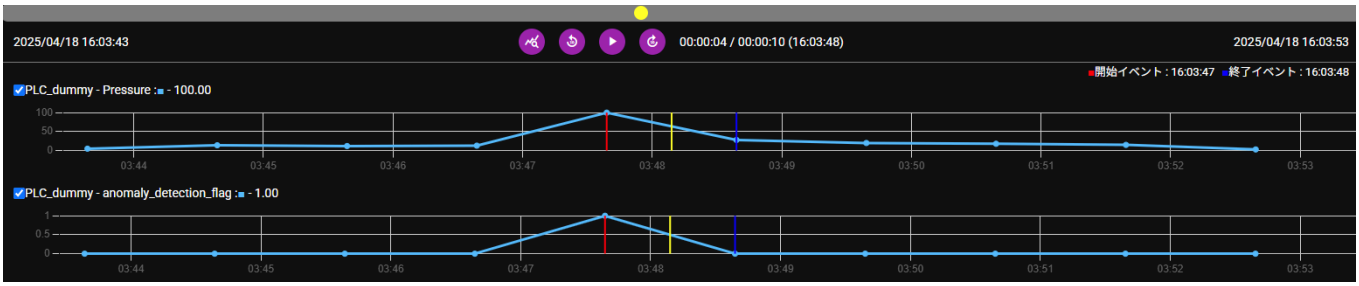
AI 번역본
-For Reference Only-


Video Record
Event Record
Live Camera
anomaly_detection_rec
2025/04/14 10:51:55 (00:00:10)
anomaly_detection_rec
2025/04/14 10:51:25 (00:00:11)
anomaly_detection_rec
2025/04/14 10:50:55 (00:00:10)
anomaly_detection_rec
2025/04/14 10:50:25 (00:00:10)
anomaly_detection_rec
2025/04/14 10:49:55 (00:00:10)
anomaly_detection_rec
2025/04/14 10:49:25 (00:00:11)
anomaly_detection_rec
2025/04/14 10:48:55 (00:00:11)
anomaly_detection_rec
2025/04/14 10:48:25 (00:00:10)
anomaly_detection_rec
2025/04/14 10:47:55 (00:00:10)
anomaly_detection_rec
2025/04/14 10:47:25 (00:00:10)
1 / 1
10

5. 임의의「 이상 검지 기록 」을 클릭하여, 뷰어 전체의 표시를 확인합니다.



6.「 再生 アイコン 」を 클릭하여, 이벤트 녹화를 재생합니다. 약 10 ~ 11초의 기록과 함께, 이상 검지가 1이 된 타이밍에, 압력도 100이 되어 있는 것을 확인할 수 있으면 정상적으로 동작하고 있습니다.



7. 튜토리얼은 여기까지 입니다.

AI 번역본
-For Reference Only-

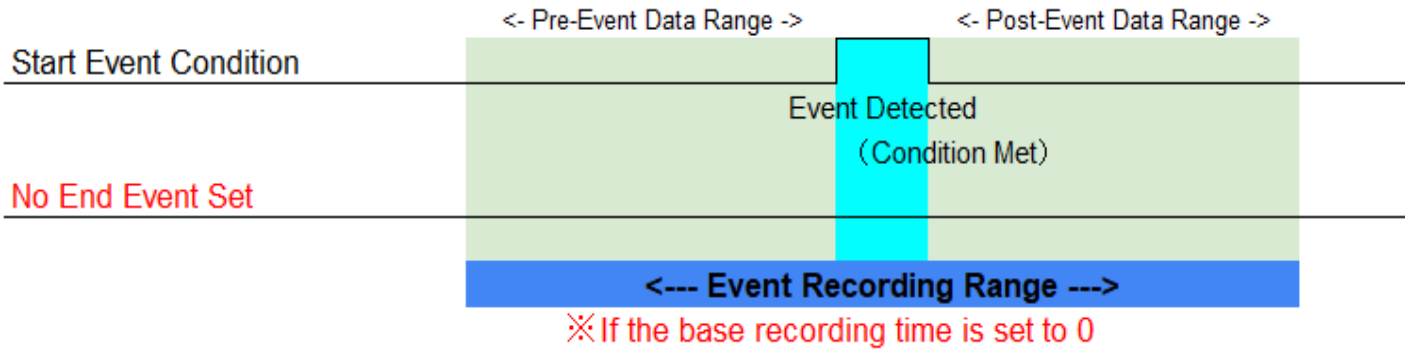
11.8 i-Trigger의 기록 사양

여기에서는, i-Trigger의 기록 사양에 대해 설명합니다. 기록 사양에 대해, 다소 복잡하므로, 자세히 확인하고 싶은 경우에 참조하십시오.

11.8.1 「 이벤트 트리거 」에서 「 개시 이벤트 」만을 설정하고 있는 경우

이벤트 트리거에서, 개시 이벤트만을 설정하고 있는 경우, 기록 구간의 사고방식은 다음과 같습니다.

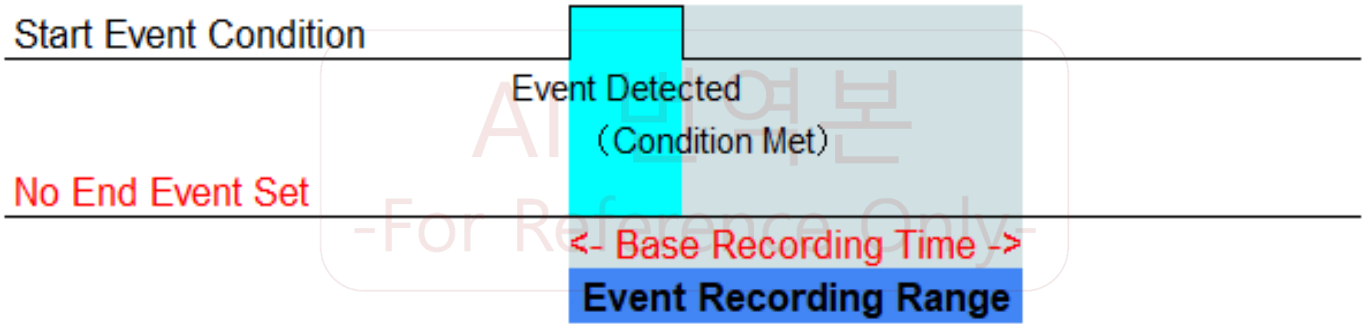
패턴1



- 기본 기록 시간을 0분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위, 이벤트 종료 후의 취득 범위에 각각 5초를 지정
- 개시 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다

이 패턴에서 기록되는 범위는, 「 이벤트의 기록 범위 (파란 배경색 부분) 」가 됩니다. 이벤트 발생 전후의 기록을 5초로 하고 있으므로, 약 10초간 이벤트가 기록됩니다.

패턴2

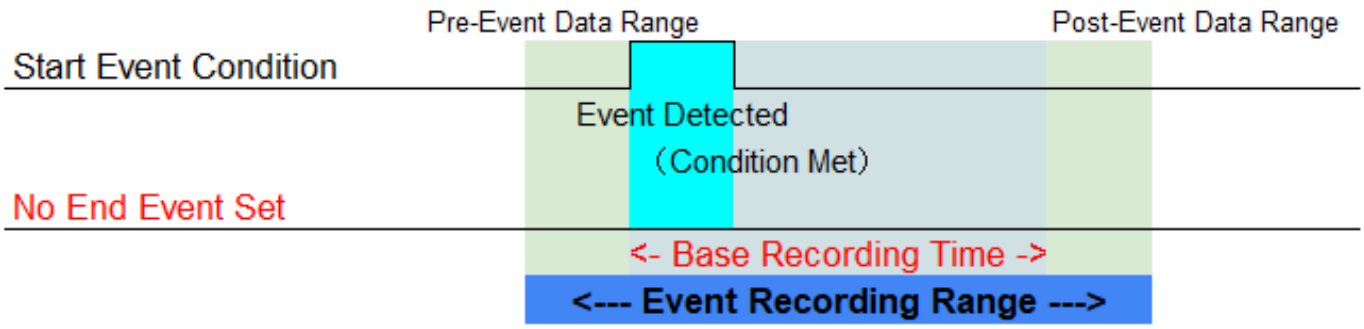


AI 번역본
-For Reference Only-

- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위, 이벤트 종료 후의 취득 범위는 각각 0초를 지정
- 개시 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다

이 패턴에서 기록되는 범위는,「이벤트의 기록 범위（파란 배경색 부분）」가 됩니다. 이벤트 발생하고 나서,
약 1분이 기록되는 범위가 됩니다.

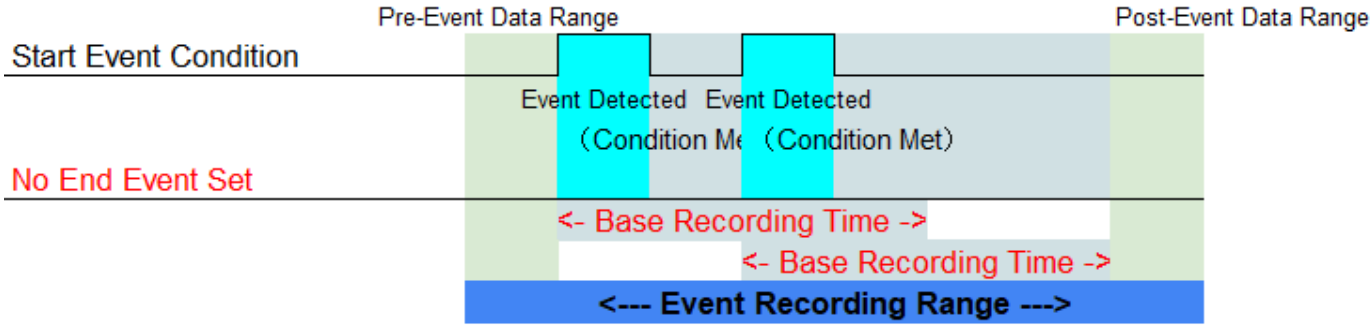
패턴3



- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위, 이벤트 종료 후의 취득 범위에 각각 5초를 지정
- 개시 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다

이 패턴에서 기록되는 범위는,「이벤트의 기록 범위（파란 배경색 부분）」가 됩니다. 이 경우는, 기본 기록 시
간 + 이벤트 전후의 범위가 기록되므로, 약 70초간 이벤트가 기록됩니다.

패턴4（이벤트의 연장）



-For Reference Only-

- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위, 이벤트 종료 후의 취득 범위에 각각 5초를 지정
- 개시 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다
- 최초의 이벤트 성립부터, 30초 후에 이벤트가 다시 성립

이 패턴에서 기록되는 범위는,「이벤트의 기록 범위（파란 배경색 부분）」가 됩니다. 기본 기록 시간의 구간 내
에서, 다시 이벤트가 성립했으므로, 이벤트가 연장됩니다. 이 경우, 최초의 이벤트 성립 구간 30초+2회째의 이벤트 성립부터의 기본
기록 시간（60초）+발생 전후 10초가 되므로, 약 100초간 이벤트가 기록됩니다.

이벤트의 연장에 대해

이벤트의 연장에 대해서는, 최대 15분의 상한이 있습니다. 15분 이상 이벤트가 연장된 경우는, 그 계속은 새로운 이벤트로서 기록됩니다.

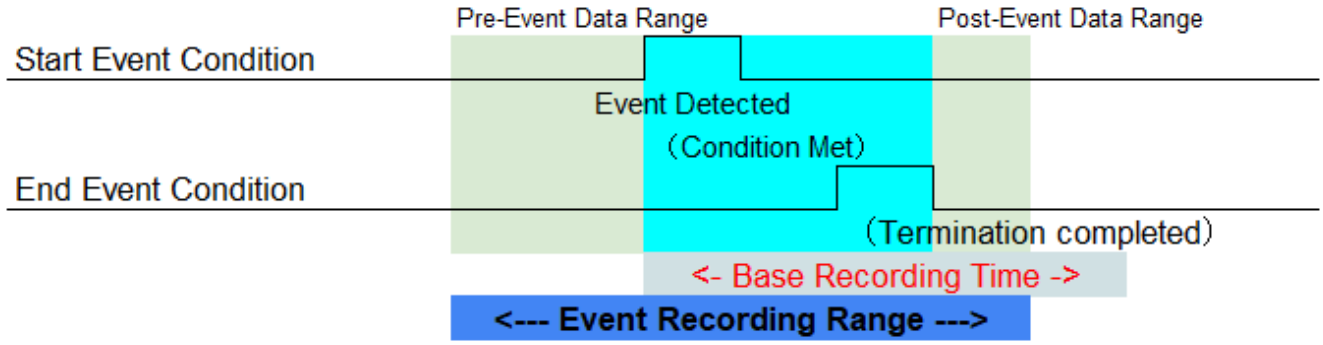
이벤트의 조건 종별에 대해

여기까지의 설명은 모두, 이벤트의 조건 종별을「Raise」로 하고 있습니다. 이 경우, 값의 상승이 조건이 되므로, 기재한 패턴의 동작이 됩니다. 조건 종별을「True」등으로 하고 있는 경우는, 조건 성립 중은 항상 이벤트가 연장되는 것에 주의하십시오.

11.8.2「이벤트 트리거」에서「개시 이벤트」,「종료 이벤트」각각을 설정하고 있는 경우

이벤트 트리거에서, 개시 이벤트, 종료 이벤트 각각을 설정하고 있는 경우, 기록 구간의 사고방식은 다음과 같습니다. 기본적인 사고방식은, 개시 이벤트만 설정한 경우와 크게는 다르지 않습니다.

패턴5

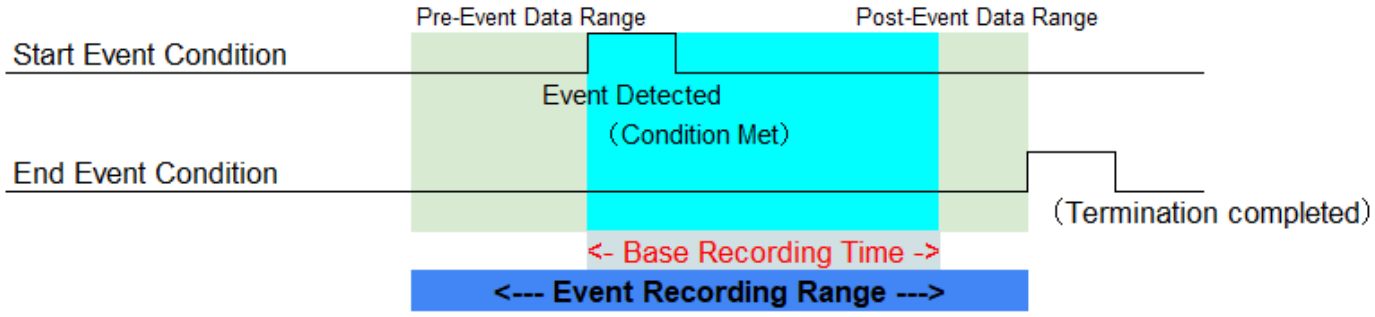


AI 번역본
-For Reference Only-

- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위를 10초, 이벤트 종료 후의 취득 범위를 5초로 지정
- 개시 이벤트 조건, 종료 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다
- 개시 이벤트 성립부터, 30초 후에 종료 이벤트가 성립

이 패턴에서 기록되는 범위는,「이벤트의 기록 범위（파란 배경색 부분）」가 됩니다. 이벤트 발생 전 10초+종료 후 5초 + 이벤트 개시부터 종료까지가 30초가 되어, 약 45초간의 이벤트가 기록됩니다. 기본 기록 시간에 도달하기 전에, 종료 이벤트가 성립했으므로, 기록 시간은 1분보다 짧아집니다.

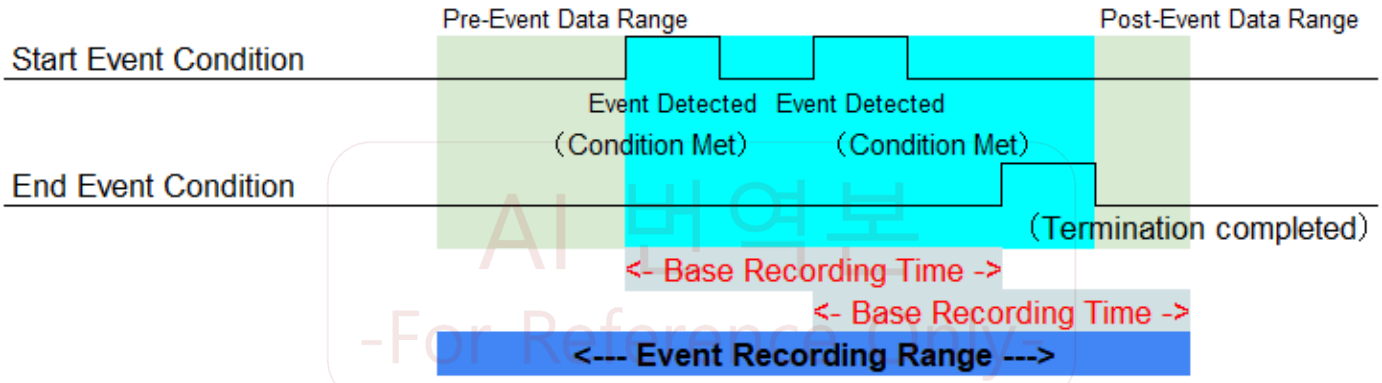
패턴6



- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위를 10초, 이벤트 종료 후의 취득 범위를 5초로 지정
- 개시 이벤트 조건, 종료 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다
- 개시 이벤트 성립부터, 1 분 경과해도 종료 이벤트가 성립하지 않았다

이 패턴에서 기록되는 범위는,「이벤트의 기록 범위（파란 배경색 부분）」가 됩니다. 이벤트 발생 전 10초+종료 후 5초 + 이벤트 개시부터 기본 기록 시간의 도달까지가 되어, 약 75초간의 이벤트가 기록됩니다. 기본 기록 시간 이내에, 종료 이벤트가 성립하지 않았으므로, 기록은 중단됩니다. 이 경우, 종료 이벤트가 기록 구간을 넘어 성립해도 무시됩니다.

패턴7（이벤트의 연장）



- 기본 기록 시간을 1분으로 하고 있습니다
- 이벤트 발생 전의 취득 범위를 10초, 이벤트 종료 후의 취득 범위를 5초로 지정

- 개시 이벤트 조건, 종료 이벤트 조건은 Raise로 하여 한 번 성립한 후, 조건이 미성립이 되지 않으면 연속으로는 조건 성립하지 않는다
- 개시 이벤트 성립부터, 30초 후에 개시 이벤트가 다시 성립, 거기에서 35초 경과하여 종료 이벤트가 성립

이 패턴에서 기록되는 범위는,「 이벤트의 기록 범위（ 파란 배경색 부분 ）」가 됩니다. 이벤트 발생 전 10초+발생 후 5초 + 최초의 이벤트 성립 구간 30초 + 다음 이벤트 성립부터 다시 35초까지가 되어, 약 80초간의 이벤트가 기록됩니다. 기본 기록 시간 이내에, 개시 이벤트가 다시 성립했으므로, 기본 기록 시간은 연장됩니다.

> **이벤트의 연장에 대해**
>
> 이벤트의 연장에 대해서는, 최대 15분의 상한이 있습니다. 15분 이상 이벤트가 연장된 경우는, 그 계속은 새로운 이벤트로서 기록됩니다.

> **이벤트의 조건 종별에 대해**
>
> 여기까지의 설명은 모두, 이벤트의 조건 종별을「 Raise 」로 하고 있습니다. 이 경우, 값의 상승이 조건이 되므로, 기재한 패턴의 동작이 됩니다. 조건 종별을「 True 」등으로 하고 있는 경우는, 조건 성립 중은 항상 이벤트가 연장되는 것에 주의하십시오.

AI 번역본
-For Reference Only-

12 릴리스 노트

본 장은, SpeeDBee Synapse의 릴리스 노트입니다.

12.1 v.4.9.9 (2025/9/12)

- BlackBear OS(v3.0.10) 대응
- 아래의 결함을 수정
  - 출력 포트의 미사용 컬럼이 무효화되지 않고 여분으로 메모리를 계속 확보하는 경우가 있다
  - 파일형 컬럼의 실파일의 삭제 처리가 동작하지 않는 경우가 있다

12.2 v.4.9.8 (2025/9/10)

- 사양 변경, 개선
  - DB 쿼리에서의 응답 데이터 건수의 상한인 1 만 건을 철폐, 및, 메모리 사용 효율의 개선
- 아래의 결함을 수정
  - 실행 라이선스용 설정 파일을 읽어 들였을 때에 일부의 프로젝트 정보를 계승하지 못해 다수의 컴포넌트가 기동할 수 없게 되는 경우가 있다

12.3 v.4.9.5 (2025/9/2)

- 사양 변경, 개선
  - https의 리버스 프록시 경유로 Web 화면에 액세스한 경우의 거동 개선
  - 전반적인 설정에서 tmpfs를 데이터 디렉터리로 설정한 경우라도 Web 화면상의 디스크 사용량 정보에 표시하는 개선
  - 매뉴얼에 Data Flow Controller 및 OMRON-SBS for Windows에 관한 보충 사항을 추가기
  - SCCDE 패키지의 커스텀 HTML로 작성된 파라미터 화면을 오퍼레이터 권한으로도 표시 가능하도록 변경
- 아래의 결함을 수정
  - 지원 파일 다운로드 시에 이벤트 녹화의 DB 파일이 포함되지 않는 경우가 있다
  - Event Recorder 실행 시에 디스크 풀이 된 경우에 이벤트 녹화 데이터에 부정합이 일어난다
  - HTML에 의한 커스텀 컴포넌트용 커스텀 UI 기능에서 파라미터 정보가 많은 경우에 커스텀 HTML로의 데이터 전달에 실패하는 경우가 있다
  - 그룹 단위의 보존 기간 설정이 OFF인 경우, 시리얼라이저에서 출력한 파일형 데이터의 보존 시간이 극단적으로 짧아진다

AI 번역본
-For Reference Only-

- SCCDE 패키지에서 커스텀 HTML을 사용하고 있는 경우, 패키지 보호를 적용하면 설정을 표시할 수 없게 되다

12.4 v.4.9.4 (2025/8/28)

- 아래의 결함을 수정
  - DB 쿼리에 개행 코드 등의 특수 문자를 포함한 SQL을 송신하면 코어가 정지/재기동되는 경우가 있다
  - 전반적인 설정에서 데이터 디렉터리를 변경하고 있는 경우에 데이터 디렉터리가 디스크 풀이 되면, 이벤트 녹화의 데이터 목록에 부정합이 생긴다

12.5 v.4.9.3 (2025/8/25)

- 아래의 결함을 수정
  - 이벤트 트리거, 이벤트 데이터, 기본 통계, 이동 평균, FFT 컴포넌트에서, 동일한 컬럼에 대해 복수의 조건 설정을 수행하면 컴포넌트 개시 시에 데드록이 발생하는 경우가 있다
- 매뉴얼의 개선
  - 설치 매뉴얼, 및 문제 해결의 설명문을 수정

12.6 v.4.9.2 (2025/8/18)

- DB 쿼리의 실행으로 대량의 데이터를 취득하는 경우에 메모리 고갈할 가능성이 있는 문제의 추가 대응

12.7 v.4.9.1 (2025/8/8)

- 사양 변경, 개선
  - 매뉴얼의 캡처 이미지에 최신 UI의 디자인을 반영
  - Linux판 설치 시의 셋업 스크립트에서 bash 의존 구문을 배제
- 아래의 결함을 수정
  - DB 쿼리의 실행으로 대량의 데이터를 취득하는 경우에 메모리 고갈할 가능성이 있다

AI 번역본
-For Reference Only-

12.8 v.4.9.0 (2025/7/31)

- 사양 변경, 개선
  - Cloud 에미터(AWS용,Azure용), MQTT 컬렉터, MQTT 에미터에서 프록시를 이용한 통신에 대응 Python 커스텀
  - 컴포넌트의 의존 모듈 설치 시에 프록시를 이용한 통신에 대응 수치 변환 기능에서 BCD 형식
  - 데이터를 통상의 수치로 변환하는 계산식의 입력에 대응

    - 출력 포트 설정에서의 영속화, 수치 변환, 등록 조건의 설정 유무의 표시 개선
    - 입력 포트 설정, 출력 포트 설정, 시스템 포트 설정, 플로우 링크 설정을 오른쪽 클릭 없이도 설정 가능하도록 하는 UI 개선
    - PLC 컬렉터, PLC 쓰기의 기종 선택에 미쓰비시전기 MELSEC iQ-R 시리즈를 추가
- 아래의 결함을 수정
    - 지원 파일의 사이즈가 거대한 경우 다운로드 시에 메모리 고갈할 가능성이 있다
    - Windows 환경에서 v4.7.2에서 덮어쓰기 설치한 경우에 일부의 컴포넌트의 데이터가 계승되지 않는다
    - v4.7.2에서 덮어쓰기 설치한 후에 발행한 액세스 토큰이 DB 쿼리에서 사용할 수 없다
    - DB 쿼리 컴포넌트로의 요청 헤더의 API 키에 여분의 문자열을 추가하면 에러가 된다

12.9 v.4.8.0 (2025/6/30)

- 사양 변경, 개선
    - 프로젝트 기능을 추가
    - 영어판 매뉴얼을 추가
    - 시계열 데이터베이스（Synapse가 수집한 데이터의 저장처）의 사이즈 상한 설정을 추가
    - 컴포넌트 개별의 로그 표시・로그 레벨 전환에 대응
    - 데이터 변환 컴포넌트를 추가
    - VIDEO 컬렉터/File 에미터/Event Recorder의 출력처 설정의 개선
    - 로그 및 임시 파일 등의 삭제 기능을 추가
    - EtherNet/IP 컬렉터에서 SHORT STRING 형 데이터의 취득에 대응
    - 각 컴포넌트의 설정 화면에 매뉴얼로의 링크 아이콘을 추가 PLC 컬렉터와
    - ModbusTCP의 수집 간격의 단위를 초에서 밀리초로 변경
    - HTML에 의한 커스텀 컴포넌트용 커스텀 UI 기능을 추가
    - DB 쿼리 컴포넌트에서 SQL의 INSERT 구문에 대응
    - 플로우 에디터에서의 터치 패널 관련 조작을 개선（iPad + Safari）
- 아래의 결함을 수정
    - 커스텀 컴포넌트의 파라미터 설정에서 첨부 파일을 이용할 수 없다（Windows판만）
    - PLC/MODBUS/EtherNetIP 컬렉터에서 수집 간격을 100ms 미만으로 하면 서버 설정의 메모리 유지 기간보다 짧은 기간밖에 유지되지 않는다
    - 입력 포트 설정, 출력 포트 설정, 필터 설정에서 대상 데이터의 형이 변경된 경우에 대상 외로 되어 버린다
    - Windows판의 EtherNet/IP 컬렉터 실행 중에 네트워크가 절단되면 복구 후에도 접속이 재개되지 않는다
    - 메일 송신 컴포넌트에서 인증 에러인 경우에도 컴포넌트가 정지하지 않고 계속 움직여 버린다

AI 번역본
-For Reference Only-

    - MQTT 에미터의 테스트 접속 시에 설정 항목의 타임아웃 값이 반영되어 있지 않다
- 내부 변경
    - 프로젝트 기능 추가에 따른 내부 데이터의 디렉터리 구성 변경
    - 대량의 컬럼을 작성하는 컴포넌트 개시 시의 성능 개선
    - 다수의 컴포넌트를 작성했을 때의 Web 화면의 동작 성능 개선

12.10 v.4.7.6 (2025/3/19)

- 사양 변경, 개선
    - EtherNet/IP 쓰기 컴포넌트에서 CIPObject로의 쓰기에 대응
    - 커스텀 컴포넌트 보호 기능의 추가
    - Synapse 커스텀 컴포넌트 개발 환경의 추가
    - Synapse가 가동하는 시스템 시각에 이상을 검출한 경우에 화면상에 통지하는 기능을 추가
    - 시뮬레이션 모드의 추가
- 아래의 결함을 수정
    - EtherNet/IP 컬렉터에서 CIPObject의 STRING형을 취득할 수 없는 경우가 있다
    - EtherNet/IP 컬렉터에서 동일 변수명을 복수 설정한 경우에 일부를 취득할 수 없다
    - MQTT 컬렉터에서 null 값을 수신한 경우에 에러가 된다
- 매뉴얼의 장 구성의 재검토
- 내부 변경
    - 과거 버전에서의 데이터 이행 처리를 개선 내부
    - 데이터의 디렉터리 구성 변경

12.11 v.4.7.5 (2025/3/13)

AI 번역본
-For Reference Only-

- 아래의 결함을 수정
    - BlackBear판에서 특정한 조건에 의해 설치가 실패한다

12.12 v.4.7.3 (2025/2/1)

- BlackBear 대응
- 사양 변경, 개선
    - 라이선스 대조에 사용하는 MAC 주소에서, 가상 인터페이스를 제외

12.13 v.4.7.2 (2025/1/31)

- v.4.7.0에서의 변경 내용을 다른 동작 환경에 반영
- 아래의 결함을 수정
  - 플로우 에디터의 컴포넌트 검색 후에, 검색란에서 선택한 컴포넌트가 화면의 중앙에 표시되지 않는다

12.14 v.4.7.0 (2024/12/20)

대상 동작 환경：Ubuntu22(arm64)

- 사양 변경, 개선
  - 컴포넌트 베이스 목록의 툴팁 표시 개선
  - 플로우 에디터 내의 컴포넌트의 복수 선택에 대응
  - 플로우 에디터 내의 컴포넌트의 컨텍스트 메뉴를 개선 전반적인 설정
  - 다이얼로그를 추가（타임존 설정만）
  - 패널의 정렬, 복제에 대응
  - 메일 송신 컴포넌트에서 사용 가능한 변수를 추가
  - PLC 컬렉터,EtherNet/IP 컬렉터, Modbus TCP 컬렉터에서의 타임스탬프의 부여 방법에 관한 설정을 추가
  - 커스텀 컬렉터의 읽기에 실패한 경우의 컴포넌트 동작을 개선
- 아래의 결함을 수정
  - FTP 에미터에서 특정한 FTP 서버로의 파일 저장에 실패한다
  - 연산 컬렉터에서 입력된 데이터가 BOOL형인 경우에 부정한 연산 결과가 출력된다

12.15 v.4.6.2 (2024/10/17)

AI 번역본
-For Reference Only-

대상 동작 환경：Ubuntu20(amd64), Ubuntu22(amd64,arm64), RaspberryPiOS(bookworm 64bit), Windows10(64bit 22H2 이후), Windows11(64bit 23H2 이후), 특수 환경（Arm32 bit）

- 사양 변경, 개선
  - 메일 송신 컴포넌트에서의 송신 실패 시의 에러 통지 개선
- 아래의 결함을 수정
  - Cloud 에미터(Azure용), Cloud 에미터(AWS용), FTP/SFTP 에미터가 특정한 조건에서 기동할 수 없게 된다
  - EtherNet/IP 쓰기 컴포넌트에서 일부의 수신 데이터가 누락되어 있는 경우에 에러가 되어 버린다
  - 입력 포트 설정의 텀블링 윈도우에서, RAW_DATA를 OFF로 하여 집약하면, 지정 폭에 의한 집약 결과를 얻을 수 없다
  - EtherNet/IP 쓰기에서 일부의 구조체의 쓰기에 실패한다

12.16 v.4.6.0 (2024/9/30)

대상 동작 환경 : Windows10(64bit 22H2 이후), Windows11(64bit 23H2 이후)

- Windows 대응
- 사양 변경, 개선
  - VIDEO 컬렉터에서, 저장처에 입력할 수 있는 문자 수 상한을 200문자로 변경
  - C 커스텀 컴포넌트 샘플의 디렉터리 구성을 변경
  - Grafana의 데이터 소스에 설정하는 URL을 Synapse의 화면에서 지정할 수 있도록 개선
  - EtherNet/IP 컬렉터에서, BOOL형 배열을 변수명만으로 지정한 경우, 배열 수가 0 또는 1이면 튜플의 1 요소째를 등록하도록 대응
- 아래의 결함을 수정
  - 연산 컬렉터에서, 복수의 입력 데이터를 트리거로 한 계산식을 설정한 경우, 컬렉터가 이상 정지한다
  - 연산 컬렉터에서, 복수의 입력 데이터를 연산할 때에 트리거를 하나만 설정하면, 연산 결과가 부정하게 되다
  - SFTP 에미터에서, 파일이 의도한 것과는 다른 디렉터리에 배치되는 경우가 있다
  - SFTP 에미터 가동 중에 SSH 접속이 계속 증가한다
  - MQTT 에미터에서, QoS가 1 또는 2인 경우라도, 네트워크 절단->재개 시에 메시지가 결손 되는 경우가 있다

12.17 v.4.5.1 (2024/9/20)

대상 동작 환경 : 특수 환경 (Arm32 bit)

- EtherNet/IP 컬렉터 배열 요소를 지정한 취득을 추가
- 아래의 결함을 수정
  - EtherNet/IP 쓰기에서 BOOL 배열의 쓰기를 할 수 없다
  - EtherNet/IP 컬렉터에서 BOOL 배열의 읽기를 할 수 없다
  - 이벤트 트리거의 설정을 변경했을 때에 경우에 따라서는 트리거가 실행되지 않는다
  - EtherNet/IP 컬렉터의 장시간 데이터 취득으로 주기가 어긋난다

AI 번역본
For Reference Only-

12.18 v.4.5.0 (2024/9/6)

대상 동작 환경 : 특수 환경 (Arm32 bit)

- PLC 컬렉터, PLC 쓰기에서 OMRON Sysmac NJ/NX를 추가
- EtherNet/IP 쓰기의 프로토타입을 추가
- 아래의 결함을 수정

    - python 컴포넌트의 입력 포트에 배열 데이터가 들어온 경우, 전부 0번째의 요소와 같은 값이 된다
- 내부 변경
    - WebAPI의 주소 설정을 127.0.0.1로 변경
    - 코어의 포트 변경 시의 내부 거동을 변경
    - system_config의 불필요하게 된 설정 항목을 삭제
    - 이벤트 트리거의 거동을 변경
    - 포트 번호의 체크를 추가

12.19 v.4.4.0 (2024/7/31)

대상 동작 환경：Ubuntu20(amd64), Ubuntu22(amd64,arm64), RaspberryPiOS(bookworm 64bit)

- 실행 라이선스 설정의 출력의 익스포트 시에 대상 패널과 자동 기동의 유무를 선택하는 기능을 추가 커스텀
- 컴포넌트용 Python-API에 최신값 취득 함수를 추가
- HTTP/HTTPS 통신의 설정 화면에 WebUI가 사용하는 주소/포트의 설정 항목을 추가
- 소프트웨어 정보 화면에 라이선스 판정용 MAC 주소의 정보를 표시
- 소프트웨어 정보 화면에서 부정한 라이선스 키를 입력했을 때의 에러 표시를 개선
- 아래의 결함을 수정
    - 스테이터스 타임라인에서 재기동 전의 데이터를 취득한다를 ON으로 했을 때에 코어가 이상 종료한다
    - 특정한 컴포넌트를 CSV/JSON 시리얼라이저에 연결하면 시리얼라이저가 에러로 정지한다
- 도큐먼트 개선
    - Python 커스텀 컴포넌트의 파일명에 대한 보충
    - Modbus의 주소 범위에 대한 보충
    - 문장 교정（오탈자의 수정, 이해하기 어려운 문장 표현의 개선 등）
- 내부 변경
    - 코어의 라이브러리 구성 변경
    - 설치 스크립트의 로그와 에러 판정의 개선
    - 영속화 데이터의 출력 간격의 디폴트 값 변경（１초→２０초）

AI 번역본
For Reference Only-

12.20 v.4.3.0 (2024/8/2)

대상 동작 환경：특수 환경（Arm32 bit）

- 4.2 이후의 일부의 수정을 반영한 후에, 특수 환경으로 이식

12.21 v.4.2.2(2024/6/28)

대상 동작 환경：Ubuntu20(amd64), Ubuntu22(amd64,arm64), RaspberryPiOS(bookworm 64bit)

- Synapse 동작 환경의 추가：RaspberryPiOS(bookworm 64bit)
- i-Trigger 관련 기능의 추가
  - VIDEO 컬렉터
  - Event Recorder
  - 동영상 재생용 뷰어
- PLC 컬렉터의 데이터 취득 최단 주기를 100ms에서 10ms로 단축
- MQTT 컬렉터에 AWS IoT Core（커스텀 인증）의 설정 항목을 추가
- MQTT 컬렉터에 의한 클라우드 서비스로부터의 데이터 수신의 순서를 매뉴얼에 추가
- 결함의 수정
  - 디스크 빈 용량이 부족한 경우의 동작 개선
  - Synapse 재설치 시에 커스텀 컴포넌트용의 Python 외부 모듈이 삭제되는 문제를 수정

12.22 v.4.1.0 (2024/6/17)

대상 동작 환경：Ubuntu22(arm64)

- 실행 라이선스 기능의 추가
- Grafana 연계에서 Grafana의 서비스 계정 기능에 대응
- Python 커스텀 컴포넌트의 실행용으로, 커스텀 컴포넌트 설정 화면에서의 whl 파일의 등록・설치에 대응

- 커스텀 UI에서 중첩된 데이터 구조의 정의에 대응
- 커스텀 UI에서 설정 화면의 사이즈 지정에 대응
- SpeeDBee Synapse 도큐먼트 교정（오탈자의 수정, 이해하기 어려운 문장 표현의 개선 등）
- 결함의 수정
  - 디스크 빈 용량이 부족한 경우의 동작 개선
  - 스테이터스 타임라인에서의 에러 정보의 표시 개선
  - 영속화된 데이터베이스가 특정한 조건에서 보존 기간을 지나도 삭제되지 않게 되는 결함을 수정
  - PLC 컬렉터 MELSEC 프로토콜에서 bit 레지스터에 0을 쓸 수 없게 되는 결함을 수정
  - admin 사용자를 무효화할 수 없는 결함을 수정
  - PLC 컬렉터 FINS 프로토콜 이용 시에 시스템이 재기동하는 결함을 수정
  - PLC 컬렉터 MELSEC(3E) 프로토콜의 ZR 레지스터 이용 시에 지정 주소와 다른 주소에 액세스해 버리는 결함을 수정

AI 번역본
-For Reference Only-

- 코어 프로세스 기동 시의 처리 개선

12.23 v.4.0.2 (2024/4/1)

대상 동작 환경：Ubuntu20(amd64), Ubuntu22(amd64,arm64)

- SpeeDBee Hive를 진화시킨 신제품「 SpeeDBee Synapse 」를 릴리스했습니다.

AI 번역본
-For Reference Only-

SpeeDBee 및 SpeeDBee Synapse는 주식회사 솔티스터의 등록상표입니다.
「 SpeeDBee Synapse 」의 로고는, 상표 등록 출원 중입니다.

Microsoft Azure, Microsoft Defender, Windows는 미국 Microsoft Corporation의 미국, 일본 및 그 밖의 국가에서의 등록상표 또는 상표입니다.

그 밖에, 기재되어 있는 회사명과 제품명 등에 대해서는, 각 사의 등록상표 또는 상표입니다.
또는,
그 밖에, 본문 중에 게재하고 있는 회사명 및 제품명은, 각각 각 사의 상표 또는 등록상표입니다.

AI 번역본
-For Reference Only-

본 제품（SpeeDBee Synapse）을 구입한 판매처에 확인하십시오.

AI 번역본
-For Reference Only-

15 부록

AI 번역본
-For Reference Only-

15.1 부록

1. 에러 코드 목록

2. 커스텀 컴포넌트 Python-API 레퍼런스

3. 커스텀 컴포넌트 C-API 레퍼런스

4. 커스텀 UI JS-API 레퍼런스

5. 이용하고 있는 오픈 소스

6. 색인

7. DB 쿼리 서비스 Web-API 사양

AI 번역본
-For Reference Only-

15.2 에러 코드 목록

스테이터스 타임라인에 표시되는 에러 식별자의 목록 및, 식별자마다 다음 항목을 게재합니다.

항목	설명
메시지	에러 식별자에 할당된 메시지
내용	에러의 자세한 내용
처치 방법	에러를 해소하기 위한 방법

15.2.1 컴포넌트 공통

15.2.1.1 NO_MEMORY

항목	설명
메시지	메모리 부족
내용	SpeeDBee Synapse가 가동하고 있는 머신의 메모리가 부족합니다.
처치 방법	사용하고 있지 않은 컴포넌트는 정지해 주십시오. 또한, 쓸데없는 데이터 수집 등이 설정되어 있지 않은가 등을 확인해 주십시오. 그래도 해소되지 않는 경우는, 시스템 관리자에게 상담하여, 메모리의 증설 또는 다른 애플리케이션과 의 조정을 수행해 주십시오.

AI 번역본

15.2.1.2 PARAMETER_PARSE_ERROR

항목	설명
메시지	파라미터 파싱 에러
내용	설정 파일이 올바른 형식으로 저장되어 있지 않습니다. 설정 내용이 파손되어 있을 가능성이 있습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.3 PARAMETER_FORMAT_ERROR

항목	설명
메시지	파라미터 에러
내용	설정 파일에 부정한 파라미터가 존재합니다.
처치 방법	연산 컬렉터에서는 변수, 식이 아무것도 설정되어 있지 않은 경우에 발생합니다. 그 밖의 컴포넌트（Azure의 접속 문자열 부정 등）에서도, 설정이 불충분한 경우에 발생하는 경우가 있습니다. 해당 컴포넌트 설정을 확인해 주십시오. 에러가 해소되지 않는 경우는, 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.4 FRAMEWORK_ERROR

항목	설명
메시지	프레임워크 API 에러
내용	해당 컴포넌트가 사용하고 있는 내부 프레임워크에서 에러가 발생했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.5 DISK_FULL

항목	설명
메시지	디스크 풀
내용	SpeeDBee Synapse가 가동하고 있는 머신의 스토리지 빈 용량이 부족합니다.
처치 방법	불필요한 파일을 삭제하는 등의 처치가 필요합니다. 시스템 관리자에게 상담하여, 스토리지 빈 용량을 확보해 주십시오.

AI 번역본
For Reference Only

15.2.1.6 IO_ERROR

항목	설명
메시지	IO 에러
내용	SpeeDBee Synapse가 가동하고 있는 머신에서, I/O 처리에 실패했습니다. (스토리지의 읽기 쓰기 실패 등)
처치 방법	시스템 관리자에게 상담하여, 스토리지의 상태 등을 확인하십시오. (고장의 가능성 등)

15.2.1.7 NETWORK_ACCESS_ERROR

항목	설명
메시지	네트워크 액세스 에러
내용	해당 컴포넌트에서, 통신의 이상이 발생했습니다.
처치 방법	(처치 방법1) SpeeDBee Synapse가 가동하고 있는 머신이, 네트워크에 접속되어 있는 것을 확인해 주십시오. (처치 방법2) 해당 컴포넌트의 통신처의 기기나 서비스가, 통신 가능한 상태인 것을 확인해 주십시오.

15.2.1.8 PROCESSING_DELAY

항목	설명
메시지	처리 지연
내용	해당 컴포넌트에서, 처리가 지연되고 있습니다.
처치 방법	(처치 방법1) 해당 컴포넌트가 송수신하고 있는 데이터 건수를 줄여 주십시오. (처치 방법2) SpeeDBee Synapse에서 가동 중인 컴포넌트 중, 불필요한 것을 정지 또는 삭제해 주십시오. (처치 방법3) SpeeDBee Synapse가 가동하고 있는 머신의 네트워크에, 지연이 발생하고 있지 않은지 확인해 주십시오.

AI 번역본
For Reference Only-

15.2.1.9 INVALID_REGEX

항목	설명
메시지	정규 표현식 에러
내용	해당 컴포넌트의 설정 항목에, 부정한 포맷의 정규 표현식이 입력되어 있습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 부정한 포맷의 정규 표현식이 입력되어 있는 설정 항목을 확인하십시오. 3. 순서2에서 확인한 설정 항목에, 올바른 포맷의 정규 표현식을 입력하여 저장해 주십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.1.10 COMPONENT_DATA_MISMATCH

항목	설명
메시지	지정한 파라미터에 일치하는 컴포넌트의 데이터가 없음 ※다음의 정보를 포함합니다. ・컴포넌트 ・데이터
내용	설정 항목에서 지정한 내용에 해당하는 데이터가 존재하지 않습니다. 이 상태에서는, 해당 설정에 의한 처리가 수행되지 않습니다.
처치 방법	1. 메시지에 포함되어 있는 컴포넌트, 데이터를 확인하십시오. 2. 해당 컴포넌트의 입력 포트를 오른쪽 클릭해 주십시오. 3. 설정 항목「 설정 방법 」에「 데이터 정의를 수신하여 설정한다 」를 지정하고, 데이터 목록을 확인해 주십시오. 4. 해당 컴포넌트의 설정 화면을 열어 주십시오. 5. 순서1에서 확인한 데이터가 지정되어 있는 설정 항목에, 순서3에서 확인한 데이터를 입력하여 저장해 주십시오. ※정규 표현식으로 데이터를 지정하는 경우는, 순서3에서 확인한 데이터를 1건 이상 포함하는 정규 표현식을 입력해 주십시오. 6. 해당 컴포넌트를 재기동해 주십시오.

샘플 번역본
-For Reference Only-

15.2.1.11 STATISTICS_INIT_ERROR

항목	설명
메시지	통계 처리 초기화 에러
내용	통계 처리의 초기화에 실패했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.12 STATISTICS_INSERT_ERROR

항목	설명
메시지	통계값 등록 에러
내용	통계값의 등록에 실패했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.13 UNAVAILABLE_DATA_TYPE

항목	설명
메시지	조건식 등에 사용할 수 없는 형이 지정되었다 ※다음의 정보를 포함합니다. ・컴포넌트 ・데이터
내용	조건식 등의 데이터에 사용할 수 없는 형이 지정되어 있습니다. 문자열이나 배열 등은 조건식에서 사용할 수 없습니다.
처치 방법	1. 메시지에 포함되어 있는 컴포넌트, 데이터를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 데이터의 형을 지정하고 있는 설정 항목에, 데이터에 적합한 형을 지정하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.1.14 UNKNOWN_ERROR

항목	설명
메시지	원인을 알 수 없는 에러
내용	원인 불명의 에러가 발생했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.1.15 INTERNAL_ERROR

항목	설명
메시지	예기치 않은 에러
내용	다른 에러 식별자에 해당하지 않는 에러가 발생했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2 리소스 컬렉터

15.2.2.1 CPUUSAGE_OPENREAD_ERROR

항목	설명
메시지	CPU 사용률 정보 오픈 에러
내용	CPU 사용률 정보를 포함하는 이하의 파일에 액세스할 수 없습니다. /proc/stat
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.2.2 CPUUSAGE_FORMAT_ERROR

항목	설명
메시지	CPU 사용률 정보 취득 에러
내용	이하의 파일에서, CPU 사용률 정보를 취득할 수 없었습니다. /proc/stat
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.3 CPUFREQ_OPENREAD_ERROR

항목	설명
메시지	CPU 주파수 정보 오픈 에러
내용	CPU 주파수 정보를 포함하는 이하의 파일에 액세스가 불가능합니다. /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
처치 방법	사용하시는 머신, 플랫폼에 따라서는, CPU 주파수를 취득할 수 없습니다. 상기의 파일이 존재하고, 이 에러가 발생하는 경우는, 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.4 CPUFREQ_FORMAT_ERROR

항목	설명
메시지	CPU 주파수 정보 취득 에러
내용	이하의 파일에서, CPU 주파수 정보를 취득할 수 없었습니다. /sys/devices/system/cpu/cpu*/cpufreq/cpuinfo_cur_freq
처치 방법	사용하시는 머신, 플랫폼에 따라서는, CPU 주파수를 취득할 수 없습니다. 상기의 파일이 존재하고, 이 에러가 발생하는 경우는, 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
For Reference Only-

15.2.2.5 CPUTEMP_OPENREAD_ERROR

항목	설명
메시지	CPU 온도 정보 오픈 에러
내용	CPU 온도 정보를 포함하는 이하의 파일에 액세스가 불가능합니다. /sys/class/thermal/thermal_zone0/temp
처치 방법	사용하시는 머신, 플랫폼에 따라서는, CPU 온도를 취득할 수 없습니다. 상기의 파일이 존재하고, 이 에러가 발생하는 경우는, 시스템 관리자에게 상담하여, 「 제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.6 CPUTEMP_FORMAT_ERROR

항목	설명
메시지	CPU 온도 정보 취득 에러
내용	이하의 파일에서, CPU 온도 정보를 취득할 수 없었습니다. /sys/class/thermal/thermal_zone0/temp
처치 방법	사용하시는 머신, 플랫폼에 따라서는, CPU 온도를 취득할 수 없습니다. 상기의 파일이 존재하고, 이 에러가 발생하는 경우는, 시스템 관리자에게 상담하여, 「 제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.7 IOSTAT_NOT_FOUND

항목	설명
메시지	IO 정보 디바이스 에러
내용	IO 정보를 포함하는 이하의 파일이 존재하지 않습니다. /dev/{설정 항목의 「 I/O 디바이스 」}
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
For Reference Only

15.2.2.8 IOSTAT_OPENREAD_ERROR

항목	설명
메시지	IO 정보 오픈 에러
내용	IO 정보를 포함하는 이하의 파일에 액세스가 불가능합니다. /dev/{설정 항목의「I/O 디바이스」}
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.9 IOSTAT_FORMAT_ERROR

항목	설명
메시지	IO 정보 취득 에러
내용	이하의 파일에서, IO 정보를 취득할 수 없었습니다. /dev/{설정 항목의「I/O 디바이스」}
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.10 IOSTAT_STORAGE_MOUNT_POINT_ERROR

항목	설명
메시지	IO 정보 취득 불가(마운트 포인트를 취득할 수 없다) ※다음의 정보를 포함합니다. ・디바이스명
내용	마운트 포인트에 관한 정보가, 설정 파일에 올바르게 정의되어 있지 않습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
For Reference Only

15.2.2.11 IOSTAT_STORAGE_INFO_ERROR

항목	설명
메시지	IO 정보 취득 불가(statvfs에서 정보를 취득할 수 없다) ※다음의 정보를 포함합니다. ・디바이스명 ・마운트 포인트 ・스테이터스
내용	설정 파일에 정의되어 있는 마운트 포인트에 대해, statvfs 커맨드를 사용하여 IO 정보를 취득할 수 없었습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.12 MEM_OPENREAD_ERROR

항목	설명
메시지	메모리 정보 오픈 에러
내용	메모리 정보를 포함하는 이하의 파일에 액세스가 불가능합니다. /proc/meminfo
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.13 MEM_FORMAT_ERROR

항목	설명
메시지	메모리 정보 취득 에러
내용	이하의 파일에서, 메모리 정보를 취득할 수 없었습니다. /proc/meminfo
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.14 NETWORK_OPENREAD_ERROR

항목	설명
메시지	네트워크 정보 오픈 에러
내용	네트워크 정보를 포함하는 이하의 파일에 액세스가 불가능합니다. /proc/net/dev
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.2.15 NETWORK_FORMAT_ERROR

항목	설명
메시지	네트워크 정보 취득 에러
내용	이하의 파일에서, 네트워크 정보를 취득할 수 없었습니다. /proc/net/dev
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여, 「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.3 PLC 컬렉터

15.2.3.1 CONNECTION_REFUSED

항목	설명
메시지	접속이 절단
내용	PLC와의 접속이 절단되었습니다.

AI 번역본
-For Reference Only-

항목	설명
처치 방법	(처치 방법1) 컴포넌트를 재기동해 주십시오. (처치 방법2) 1. PLC측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.3.2 NETWORK_TIMEOUT

항목	설명
메시지	타임아웃 ※다음의 정보를 포함합니다. ・주소 ・포트 번호 ・설정되어 있는 타임아웃값 ・조작 종별
내용	(조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」의 시간 내에 PLC에 접속할 수 없었습니다. (조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」의 시간 내에 PLC로부터 데이터를 수신할 수 없었습니다.
처치 방법	1. 메시지에 포함되어 있는 조작 종별을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. (순서1에서 확인한 조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」을 크게 하십시오. (순서1에서 확인한 조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」을 크게 하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.3.3 REGISTER_ADDRESS_ERROR

항목	설명
메시지	지정 레지스터가 부정 ※다음의 정보를 포함합니다.

항목	설명
	・레지스터명 ・레지스터 주소 ・조작 종별
내용	비트 지정할 수 없는 레지스터 주소가, 비트 지정되어 설정 항목「 주소 」에 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, 비트 지정이 가능한 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.3.4 REGISTER_ADDRESS_RANGE_ERROR

항목	설명
메시지	부정한 레지스터 주소 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 주소 」에, PLC의 프로토콜로는 다룰 수 없는 레지스터 주소가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, PLC 프로토콜로 다룰 수 있는 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오. ※이용할 수 있는 주소, 그 범위에 대해서는, 해당 기기의 매뉴얼 등을 참조하십시오.

For Reference Only

15.2.3.5 REGISTER_ADDRESS_DATA_TYPE_ERROR

항목	설명
메시지	데이터형이 부정 ※다음의 정보를 포함합니다. ・레지스터명

항목	설명
	・레지스터 주소 ・조작 종별
내용	설정 항목「 데이터형 」에, PLC측의 설정과는 다른 데이터형이 지정되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. PLC의 설정에서, 순서1에서 확인한 레지스터의 데이터형을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 데이터형 」에, 순서2에서 확인한 데이터형을 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

15.2.3.6 REGISTER_ADDRESS_SIZE_ERROR

항목	설명
메시지	연속 영역 수가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 연속 영역 수 」에, PLC측의 설정과는 다른 연속 영역 수가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. PLC의 설정에서, 순서1에서 확인한 레지스터의 연속 영역 수를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 연속 영역 수 」에, 순서2에서 확인한 연속 영역 수를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

-For Reference Only-

15.2.3.7 REGISTER_READ_ERROR

항목	설명
메시지	레지스터 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명

항목	설명
	・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터의 취득에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시하십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.3.8 GROUP_REGISTER_READ_ERROR

항목	설명
메시지	레지스터 일괄 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터의 일괄 취득에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시하십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
For Reference Only

15.2.4 Modbus TCP 컬렉터

15.2.4.1 CONNECTION_REFUSED

항목	설명
메시지	접속이 절단
내용	기기와의 접속이 절단되었습니다.
처치 방법	(처치 방법1) 컴포넌트를 재기동해 주십시오. (처치 방법2) 1. 기기측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.4.2 NETWORK_TIMEOUT

항목	설명
메시지	타임아웃 ※다음의 정보를 포함합니다. ・주소 ・포트 번호 ・설정되어 있는 타임아웃값 ・조작 종별
내용	(조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」의 시간 내에 기기에 접속할 수 없었습니다. (조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」의 시간 내에 기기로부터 데이터를 수신할 수 없었습니다.
처치 방법	1. 메시지에 포함되어 있는 조작 종별을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. (순서1에서 확인한 조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」을 크게 하십시오. (순서1에서 확인한 조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」을 크게 하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역 참조
-For Reference Only-

15.2.4.3 REGISTER_ADDRESS_ERROR

항목	설명
메시지	지정 레지스터가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	비트 지정할 수 없는 레지스터 주소가, 비트 지정되어 설정 항목「 주소 」에 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, 비트 지정 가능한 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.4.4 REGISTER_ADDRESS_RANGE_ERROR

항목	설명
메시지	부정한 레지스터 주소 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 주소 」에, 기기의 프로토콜로는 다룰 수 없는 레지스터 주소가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, 기기 프로토콜로 다룰 수 있는 레지스터 어드레스를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오. ※이용할 수 있는 주소, 그 범위에 대해서는, 해당 기기의 매뉴얼 등을 참조하십시오.

AI 번역본
For reference Only

15.2.4.5 REGISTER_ADDRESS_DATA_TYPE_ERROR

항목	설명
메시지	데이터형이 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 데이터형 」에, 기기측의 설정과는 다른 데이터형이 지정되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 기기의 설정에서, 순서1에서 확인한 레지스터의 데이터형을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 데이터형 」에, 순서2에서 확인한 데이터형을 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

15.2.4.6 REGISTER_ADDRESS_SIZE_ERROR

항목	설명
메시지	연속 영역 수가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 연속 영역 수 」에, 기기측의 설정과는 다른 연속 영역 수가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 기기의 설정에서, 순서1에서 확인한 레지스터의 연속 영역 수를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 연속 영역 수 」에, 순서2에서 확인한 연속 영역 수를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.4.7 REGISTER_READ_ERROR

항목	설명
메시지	레지스터 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터의 취득에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시하십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.4.8 GROUP_REGISTER_READ_ERROR

항목	설명
메시지	레지스터 일괄 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터의 일괄 취득에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시하십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.5 EtherNet/IP 컬렉터

15.2.5.1 TYPE_MISMATCH

항목	설명
메시지	형 불일치 ※다음의 정보를 포함합니다. ・태그명 ・설정의 형 ・읽어낸 형
내용	설정 항목「 데이터형 」에, 기기측의 설정과는 다른 데이터형이 지정되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 태그명, 설정의 형, 읽어낸 형을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 태그의 설정 항목「 데이터형 」에, 순서1에서 확인한 읽어낸 태그의 형을 지정하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.5.2 UNKNOWN_TAG

항목	설명
메시지	지정 태그 읽기 불가 ※다음의 정보를 포함합니다. ・태그명
내용	지정된 태그의 데이터를 취득할 수 없습니다. ※설정 항목「 설정 방법 」이「 변수 」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 태그명을 확인하십시오. 2. 기기측의 설정에서, 올바른 태그명을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 태그의 설정 항목「 변수명 」에, 순서2에서 확인한 태그명을 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.5.3 CIP_SETTING_ERROR

항목	설명
메시지	CIP 오브젝트 설정 에러 ※다음의 정보를 포함합니다. ・클래스 ID ・인스턴스 ID ・어트리뷰트 ID
내용	지정된 CIP 오브젝트의 데이터를 취득할 수 없습니다. ※설정 항목「 설정 방법 」이「 CIP 오브젝트 」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 클래스 ID, 인스턴스 ID, 어트리뷰트 ID를 확인하십시오. 2. 기기측의 설정에서, 올바른 클래스 ID, 인스턴스 ID, 어트리뷰트 ID를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 CIP 오브젝트의 설정 항목「 클래스 ID 」「 인스턴스 ID 」「 어트리뷰트 ID 」에, 순서2에서 확인한 정보를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. ※액세스할 수 있는 CIP 오브젝트에 대해서는, 해당 기기의 매뉴얼 등을 참조하십시오.

15.2.5.4 ARRAYSIZE_MISMATCH

항목	설명
메시지	배열 사이즈 에러 ※다음의 정보를 포함합니다. ・컬럼명
내용	수신한 데이터의 배열 수가, 설정 항목「 배열 수 」에 입력한 배열 수를 밑돌고 있습니다.
처치 방법	1. 메시지에 포함되어 있는 컬럼명을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 컬럼의 설정 항목「 배열 수 」에, 보다 작은 수를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.5.5 READ_TIME_OVER

항목	설명
메시지	읽기 시간 오버 ※다음의 정보를 포함합니다. ・읽기에 걸린 시간
내용	설정 항목「 취득 간격 」에 입력한 간격보다도, 데이터의 읽기에 시간이 걸리고 있습니다.
처치 방법	1. 메시지에 포함되어 있는, 읽기에 걸린 시간을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 취득 간격 」에, 순서1에서 확인한 시간보다 큰 시간을 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.6 연산 컬렉터

15.2.6.1 EXPRESSION_FORMAT_ERROR

항목	설명
메시지	계산식의 포맷이 부정 ※다음의 정보를 포함합니다. ・출력명
내용	설정 항목「 연산식 」에, 부정한 포맷의 연산식이 입력되어 있습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본

15.2.6.2 SOME_VARIABLE_NOT_SPECIFIED

항목	설명
메시지	계산식 내에 있는 일부 변수가 지정되어 있지 않음 ※다음의 정보를 포함합니다. ・출력명 ・변수명
내용	설정 항목「 연산식 」에, 미정의 변수가 포함되어 있습니다.

항목	설명
처치 방법	(대처 방법1) 1. 메시지에 포함되어 있는 출력명, 변수명을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 변수명 」이, 순서1에서 확인한 변수명과 같은 입력 데이터를 추가하십시오. 4. 순서1에서 확인한 출력 데이터의 설정 항목「 트리거 변수 」에, 순서3에서 추가한 변수를 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (대처 방법2) 1. 메시지에 포함되어 있는 출력명, 변수명을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 출력 데이터의 설정 항목「 연산식 」에서, 순서1에서 확인한 변수를 삭제하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.6.3 NO_TRIGGER_SPECIFIED

항목	설명
메시지	어느 변수에도 트리거가 설정되어 있지 않음 ※다음의 정보를 포함합니다. ・출력명
내용	설정 항목「 트리거 변수 」에, 변수가 등록되어 있지 않습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본

15.2.6.4 EVALUATION_ERROR

항목	설명
메시지	계산식의 평가 에러 ※다음의 정보를 포함합니다. ・출력명 ・에러 스테이터스
내용	설정 항목「 연산식 」을 평가할 수 없었습니다.

항목	설명
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.6.5 NO_VARIABLE_SPECIFIED

항목	설명
메시지	계산식 안에 변수가 하나도 지정되어 있지 않음 ※다음의 정보를 포함합니다. ・출력명
내용	설정 항목「 연산식 」에, 설정 항목「 트리거 변수 」에서 지정된 변수가 포함되어 있지 않습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.6.6 COMPUTE_ERROR

항목	설명
메시지	연산 처리에서 에러 ※다음의 정보를 포함합니다. ・출력명
내용	설정 항목「 연산식 」의 실행 시에 에러가 발생했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

-For Reference Only-

15.2.6.7 OUTCOLUMN_CREATE_ERROR

항목	설명
메시지	출력 컬럼의 작성에 실패 ※다음의 정보를 포함합니다. ・출력명

항목	설명
내용	설정 항목「 연산식 」의 실행 결과를 등록하는 출력 컬럼의 생성에 실패했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.6.8 OUTCOLUMN_INSERT_ERROR

항목	설명
메시지	출력 컬럼에의 데이터 등록에 실패 ※다음의 정보를 포함합니다. ・출력명
내용	설정 항목「 연산식 」의 실행 결과를 출력 컬럼에 등록할 수 없었습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.6.9 VARIABLE_NOT_UNIQUE_ERROR

항목	설명
메시지	변수 설정이 중복되어 있어 변수명을 특정할 수 없음 ※다음의 정보를 포함합니다. ・컴포넌트명 ・데이터명 ・통계 타입
내용	설정 항목「 입력 데이터 」에, 정의가 중복되어 있는 변수가 존재합니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.7 MQTT 컬렉터

15.2.7.1 AUTHENTICATION_FAILURE

항목	설명
메시지	인증 에러
내용	MQTT 브로커에의 인증이 실패했습니다.
처치 방법	(설정 항목「 익명 액세스 」를 OFF로 하는 경우) 1. MQTT 브로커측의 설정에서, 올바른 사용자 정보(사용자명・패스워드)를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 익명 액세스 」를 OFF로 하십시오. 4. 설정 항목「 사용자명 」「 패스워드 」에, 순서1에서 확인한 사용자명・패스워드를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (설정 항목「 익명 액세스 」를 ON으로 하는 경우) 1. MQTT 브로커측의 설정에서, 익명 액세스가 허가되어 있는 것을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 익명 액세스 」를 ON으로 하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.7.2 MESSAGE_PARSE_ERROR

항목	설명
메시지	수신 메시지 해석 에러
내용	상정과는 다른 포맷의 데이터를 수신했습니다.
처치 방법	Publish하고 있는 기기측의 설정에서, 다음 포맷의 데이터를 송신하도록 하십시오. { 'records' : [{ 'timestamp': '타임스탬프(ISO 8601형식)' },{ '데이터명': '값' }]} ※1 JSON 형식으로, records라는 키에 대해 배열 형식으로 데이터를 정의하십시오. ※2 배열 안에, { '데이터명': '값' }이라는 형식으로 데이터를 정의하십시오. ※3 타임스탬프의 설정은 임의입니다. 설정하는 경우는, { 'timestamp': '타임스탬프(ISO 8601형식)' }로 타임스탬프를 정의하십시오. 설정하지 않는 경우는, 자동적으로 타임스탬프가 부여됩니다. 이 경우, SpeeDBee Synapse의 가동하는 머신의 시스템 시각이 사용됩니다.

AI 번역본
For Reference Only

15.2.7.3 UNSUPPORTED_DATA_TYPE

항목	설명
메시지	미대응 데이터형
내용	수신한 데이터에, MQTT 컬렉터가 대응하고 있지 않은 형의 값이 포함되어 있었습니다.
처치 방법	Publish하고 있는 기기측의 설정에서, 다음 종류의 데이터를 송신하도록 하십시오. ・정수값 ・소수값 ・문자 ・진위값

15.2.7.4 TIMESTAMP_ERROR

항목	설명
메시지	타임스탬프 에러 ※다음의 정보를 포함합니다. ・전회 등록 시간 ・에러 등록 시간
내용	전회 수신한 데이터의 타임스탬프보다도, 이번에 수신한 데이터의 타임스탬프 쪽이 오래된 시각입니다.
처치 방법	1. 메시지에 포함되어 있는 전회 등록 시간을 확인하십시오. 2. Publish하고 있는 기기측의 설정에서, 순서1에서 확인한 전회 등록 시간보다도 새로운 타임스탬프의 데이터를 송신하도록 하십시오.

AI 번역본

15.2.7.5 TIMESTAMP_RANGE_ERROR

항목	설명
메시지	타임스탬프 허용 범위 에러 ※다음의 정보를 포함합니다. ・타임스탬프 ・허용 최소 시각 ・허용 최대 시각을 포함
내용	수신한 데이터 내에서 정의되어 있는 타임스탬프와 실타임스탬프의 오차가, 설정 항목「타임스탬프 오차 허용 범위」의 설정값보다 큽니다.

항목	설명
처치 방법	(처치 방법1) Publish하고 있는 기기측의 설정에서, 타임스탬프를 포함하지 않는 데이터를 송신하도록 하십시오. (처치 방법2) 1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 타임스탬프 오차 허용 범위 」를 확인하십시오. 3. Publish하고 있는 기기측의 설정에서, 실제 시각과의 오차가 순서2에서 확인한 허용 범위 내에 들어가는 타임스탬프를 가진 데이터를 송신하십시오. (처치 방법3) 1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 타임스탬프 오차 허용 범위 」에, 보다 큰 값을 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.7.6 TIMESTAMP_PARSE_ERROR

항목	설명
메시지	타임스탬프 파싱 에러
내용	수신한 데이터 내에서 정의되어 있는 타임스탬프가, 상정하는 포맷과는 달랐습니다.
처치 방법	Publish하고 있는 기기측의 설정에서, ISO 8601 형식의 타임스탬프를 가진 데이터를 송신하도록 하십시오.
비고	ISO 8601 형식의 타임스탬프의 예는 다음과 같습니다. 2023-10-30T10:00:00.012345678Z

AI 번역본

15.2.7.7 RECEIVED_PARAMETER_ERROR

항목	설명
메시지	수신 데이터 에러 ※다음의 정보를 포함합니다. ・컬럼명(또는 키 명칭) ・에러 요인
내용	수신한 데이터 내에서 정의되어 있는 파라미터에 문제가 있었습니다. 에러 요인의 종류는 다음과 같습니다. ・data_type：작성된 컬럼의 형과 수신한 데이터의 형이 다름

항목	설명
	・array_size：작성된 컬럼의 배열 수와 수신한 데이터의 배열 수가 다름 ・value：수신한 데이터의 값이 컬럼의 형의 범위 외 ・key_name：설정 항목「동적 컬럼」이 ON이고, 컬렉터가 수신한 JSON 데이터를 바탕으로 컬럼을 생성할 때, 키가 컬럼명에 사용할 수 없음(부정한 문자, 문자 수 상한을 넘고 있는 등)
처치 방법	(에러 요인이 data_type인 경우) 1. 메시지에 포함되어 있는 컬럼명을 확인하십시오. 2. Publish하고 있는 기기측의 설정에서, 순서1에서 확인한 컬럼의 데이터형을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 컬럼의 설정 항목「데이터형」에, 순서2에서 확인한 데이터형을 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (에러 요인이 array_size인 경우) 1. 메시지에 포함되어 있는 컬럼명을 확인하십시오. 2. Publish하고 있는 기기측의 설정에서, 순서1에서 확인한 컬럼의 배열 수를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 컬럼의 설정 항목「데이터 수」에, 순서2에서 확인한 배열 수를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (에러 요인이 value인 경우) 1. 메시지에 포함되어 있는 컬럼명을 확인하십시오. 2. Publish하고 있는 기기측의 설정에서, 순서1에서 확인한 데이터의 값을, 데이터형의 범위 내에 들어가도록 변경하십시오. (에러 요인이 key_name인 경우) 1. 메시지에 포함되어 있는 키 명칭을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「데이터명」에 입력할 수 있는 이름을 확인하십시오. 4. Publish하고 있는 기기측의 설정에서, 순서1에서 확인한 컬럼의 이름을, 순서3에서 확인한 이름으로 변경하십시오.

-For Reference Only-

15.2.7.8 CA_FILE_ERROR

항목	설명
메시지	증명서 파일 에러
내용	설정 항목「CA 증명서」에 업로드된 증명서가 부정합니다.

항목	설명
처치 방법	1. MQTT 브로커측의 설정에서, 올바른 CA 증명서를 취득하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 CA 증명서 」에, 순서1에서 취득한 증명서를 업로드하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.8 MQTT 에미터

15.2.8.1 AUTHENTICATION_FAILURE

항목	설명
메시지	인증 에러
내용	MQTT 브로커에의 인증이 실패했습니다.
처치 방법	(설정 항목「 익명 액세스 」를 OFF로 하는 경우) 1. MQTT 브로커측의 설정에서, 올바른 사용자 정보(사용자명・패스워드)를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 익명 액세스 」를 OFF로 하십시오. 4. 설정 항목「 사용자명 」「 패스워드 」에, 순서1에서 확인한 사용자명・패스워드를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (설정 항목「 익명 액세스 」를 ON으로 하는 경우) 1. MQTT 브로커측의 설정에서, 익명 액세스가 허가되어 있는 것을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 익명 액세스 」를 ON으로 하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

-For Reference Only-

15.2.8.2 UNSUPPORTED_DATA_TYPE

항목	설명
메시지	미대응 데이터형
내용	송신하는 데이터 중에, 미대응 데이터형의 것이 포함되어 있습니다.

항목	설명
처치 방법	1. 해당 컴포넌트의 입력 포트를 오른쪽 클릭해 주십시오. 2. 설정 항목「 설정 방법 」에「 데이터 정의를 수신하여 설정한다 」를 지정해 주십시오. 3. 데이터 목록 중에서, MQTT 에미터가 대응하고 있지 않은 형의 데이터를 확인하십시오. 4. 순서3에서 확인한 데이터를 송신하고 있는 컴포넌트에 대해, 다음 중 어느 하나의 처치를 수행해 주십시오. ・플로우 링크를 삭제한다 ・플로우 링크의 필터 기능으로, 순서3에서 확인한 데이터가 통과하지 않도록 설정한다 ・컴포넌트를 정지 혹은 삭제한다 ・순서3에서 확인한 데이터를 송신하지 않도록 컴포넌트의 설정을 수행한다
비고	MQTT 에미터가 대응하고 있는 데이터형은 다음과 같습니다. ・BOOLEAN ・INT8 ・INT16 ・INT32 ・INT64 ・UINT8 ・UINT16 ・UINT32 ・UINT64 ・FLOAT ・DOUBLE ・STRING ・BINARY ・FILE

15.2.8.3 PUBLISH_ERROR

항목	설명
메시지	송신 에러
내용	데이터의 송신에 실패했습니다.
처치 방법	(대처 방법1) SpeeDBee Synapse가 가동하고 있는 머신이, 네트워크에 접속되어 있는 것을 확인하십시오. (대처 방법2) 1. MQTT 브로커측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시

항목	설명
	오. 4. 해당 컴포넌트를 재기동해 주십시오. (대처 방법3) 시스템 관리자에게 상담하여, MQTT 브로커측의 설정에 미비가 없는지 확인하십시오.

15.2.9 File 에미터

15.2.9.1 UNSUPPORTED_DATA_TYPE

항목	설명
메시지	미대응 데이터형
내용	송신하는 데이터 중에, 미대응 데이터형의 것이 포함되어 있습니다.
처치 방법	1. 해당 컴포넌트의 입력 포트를 확인하십시오. 2. 다음의 것이 아닌 컴포넌트와 플로우 링크로 이어져 있는 경우, 그 플로우 링크를 삭제해 주십시오. ・CSV 시리얼라이저 ・JSON 시리얼라이저

15.2.10 Cloud 에미터(AWS용)

15.2.10.1 AUTHENTICATION_FAILURE

항목	설명
메시지	인증 에러
내용	AWS IoT Core에의 인증이 실패했습니다.
처치 방법	1. AWS IoT Core의 설정에서, 올바른 호스트・포트를 확인하고, 올바른 CA 증명서・SSL 증명서・비밀키를 취득해 주십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 다음의 설정 항목에, 순서1에서 확인한 정보를 입력하고, 취득한 파일을 업로드하여 저장해 주십시오. ・호스트 ・포트

AI 번역본
-For Reference Only-

항목	설명
	・CA 증명서 ・SSL 증명서 ・비밀키 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.10.2 PUBLISH_ERROR

항목	설명
메시지	송신 에러
내용	AWS IoT Core에의 데이터의 송신에 실패했습니다.
처치 방법	(대처 방법1) SpeeDBee Synapse가 가동하고 있는 머신이, 네트워크에 접속되어 있는 것을 확인하십시오. (대처 방법2) 1. AWS IoT Core의 설정에서, 올바른 호스트・포트를 확인하고, 올바른 CA 증명서・SSL 증명서・비밀키를 취득해 주십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 다음의 설정 항목에, 순서1에서 확인한 정보를 입력하고, 취득한 파일을 업로드하여 저장해 주십시오. ・호스트 ・포트 ・CA 증명서 ・SSL 증명서 ・비밀키 4. 해당 컴포넌트를 재기동해 주십시오. (대처 방법3) 시스템 관리자에게 상담하여, AWS Iot Core측의 설정에 미비가 없는지 확인하십시오.

-For Reference Only-

15.2.10.3 UNSUPPORTED_DATA_TYPE

항목	설명
메시지	미대응 데이터형
내용	송신하는 데이터 중에, 미대응 데이터형의 것이 포함되어 있습니다.

항목	설명
처치 방법	1. 해당 컴포넌트의 입력 포트를 확인하십시오. 2. 다음의 것이 아닌 컴포넌트와 플로우 링크로 이어져 있는 경우, 그 플로우 링크를 삭제해 주십시오. ・CSV 시리얼라이저 ・JSON 시리얼라이저

15.2.11 Cloud 에미터(Azure용)

15.2.11.1 AUTHENTICATION_FAILURE

항목	설명
메시지	인증 에러
내용	Azure IoT Hub에의 인증이 실패했습니다.
처치 방법	1. Azure IoT Hub측의 설정에서, 올바른 접속 문자열을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 접속 문자열 」에, 순서1에서 확인한 접속 문자열을 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.11.2 PUBLISH_ERROR

항목	설명
메시지	송신 에러
내용	데이터의 송신에 실패했습니다.
처치 방법	(대처 방법1) SpeeDBee Synapse가 가동하고 있는 머신이, 네트워크에 접속되어 있는 것을 확인하십시오. (대처 방법2) 1. Azure IoT Hub측의 설정에서, 올바른 접속 문자열을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 접속 문자열 」에, 순서1에서 확인한 접속 문자열을 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

항목	설명
	(대처 방법3) 시스템 관리자에게 상담하여, Azure IoT Hub측의 설정에 미비가 없는지 확인하십시오.

15.2.11.3 UNSUPPORTED_DATA_TYPE

항목	설명
메시지	미대응 데이터형
내용	송신하는 데이터 중에, 미대응 데이터형의 것이 포함되어 있습니다.
처치 방법	1. 해당 컴포넌트의 입력 포트를 확인하십시오. 2. 다음의 것이 아닌 컴포넌트와 플로우 링크로 이어져 있는 경우, 그 플로우 링크를 삭제해 주십시오. ・CSV 시리얼라이저 ・JSON 시리얼라이저

15.2.12 FTP/FTPS/SFTP 에미터

15.2.12.1 AUTHENTICATION_FAILURE

항목	설명
메시지	인증 에러
내용	FTP/FTPS/SFTP 서버에의 인증이 실패했습니다.
처치 방법	(설정 항목「 익명 액세스 」를 OFF로 하는 경우) 1. FTP/FTPS/SFTP 서버측의 설정에서, 올바른 사용자 정보(사용자명・패스워드)를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 익명 액세스 」를 OFF로 하십시오. 4. 설정 항목「 사용자명 」「 패스워드 」에, 순서1에서 확인한 사용자명・패스워드를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오. (설정 항목「 익명 액세스 」를 ON으로 하는 경우) 1. FTP/FTPS/SFTP 서버측의 설정에서, 익명 액세스가 허가되어 있는 것을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오.

AI 번역본
-For Reference Only-

항목	설명
	3. 설정 항목「 익명 액세스 」를 ON으로 하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.12.2 SEND_ERROR

항목	설명
메시지	송신 에러
내용	데이터의 송신에 실패했습니다.
처치 방법	(대처 방법1) SpeeDBee Synapse가 가동하고 있는 머신이, 네트워크에 접속되어 있는 것을 확인하십시오. (대처 방법2) 1. FTP/FTPS/SFTP 서버측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오. (대처 방법3) 시스템 관리자에게 상담하여, FTP/FTPS/SFTP 서버측의 설정에 미비가 없는지 확인하십시오.

15.2.12.3 PERMISSION_ERROR

항목	설명
메시지	디렉터리 조작 에러
내용	FTP/FTPS/SFTP 서버 상의 디렉터리의 조작에 실패했습니다.(권한 에러 등)
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 출력처 디렉터리 」에 입력되어 있는 디렉터리를 확인하십시오. 3. 시스템 관리자에게 상담하여, FTP/FTPS/SFTP 서버 상에서, 순서3에서 확인한 디렉터리에 대하여 쓰기 권한의 부여 등을 수행해 주십시오.

AI 번역본
- For Reference Only -

15.2.13 JSON 시리얼라이저

15.2.13.1 MAX_SIZE_ERROR

항목	설명
메시지	상한 사이즈 에러
내용	파일 사이즈가 설정 항목「 상한 사이즈 」를 초과하기 때문에, 파일을 작성할 수 없습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 상한 사이즈 」에, 보다 큰 값을 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.14 CSV 시리얼라이저

15.2.14.1 MAX_SIZE_ERROR

항목	설명
메시지	상한 사이즈 에러
내용	파일 사이즈가 설정 항목「 상한 사이즈 」를 초과하기 때문에, 파일을 작성할 수 없습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 상한 사이즈 」에, 보다 큰 값을 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.15 이벤트 트리거

AI 번역본

15.2.15.1 IGNORE_INVALID_TIMESTAMP_DETECT

항목	설명
메시지	무효한 타임스탬프의 데이터로부터의 이벤트 검지를 무시했다
내용	개시 이벤트, 또는 종료 이벤트를 검지 후에, 무효한 타임스탬프의 데이터로부터 다른 트리거가 발생하여, 해당 트리거에 의한 검지를 무시한 것을 나타내는 에러입니다. 무효한 타임스탬프란, 최초의 검지 시와 같은 타임스탬프, 또는 그보다 과거의 타임스탬프의 데이터를 나타냅니다. 이벤트 트리거는, 이와 같은 타임스탬프의 데이터로부터의 검지를 무시합니다.

항목	설명
처치 방법	개시 이벤트와 종료 이벤트가 동시 성립하는 조건식을 재검토해 주십시오. 개시 이벤트와 종료 이벤트가 다른 타이밍에 발생하는 경우에, 이 에러는 발생하지 않습니다.

15.2.16 셸 커맨드

15.2.16.1 COMMAND_EXEC_ERROR

항목	설명
메시지	커맨드 실행 에러
내용	설정 항목「 커맨드 」에 입력된 커맨드의 실행에 실패했습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 커맨드 」에, 실행 가능한 커맨드를 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.17 PLC 쓰기

15.2.17.1 CONNECTION_REFUSED

항목	설명
메시지	접속이 절단
내용	PLC와의 접속이 절단되었습니다.
처치 방법	(처치 방법1) 컴포넌트를 재기동해 주십시오. (처치 방법2) 1. PLC측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.17.2 NETWORK_TIMEOUT

항목	설명
메시지	타임아웃 ※다음의 정보를 포함합니다. ・주소 ・포트 번호 ・설정되어 있는 타임아웃값 ・조작 종별
내용	(조작 종별이 connect, close인 경우) 설정 항목「접속 타임아웃」의 시간 내에 PLC에 접속할 수 없었습니다. (조작 종별이 receive, send인 경우) 설정 항목「읽기 타임아웃」의 시간 내에 PLC로부터 데이터를 수신할 수 없었습니다.
처치 방법	1. 메시지에 포함되어 있는 조작 종별을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. (순서1에서 확인한 조작 종별이 connect, close인 경우) 설정 항목「접속 타임아웃」을 크게 하십시오. (순서1에서 확인한 조작 종별이 receive, send인 경우) 설정 항목「읽기 타임아웃」을 크게 하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.17.3 REGISTER_ADDRESS_ERROR

항목	설명
메시지	지정 레지스터가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	비트 지정할 수 없는 레지스터 주소가, 비트 지정되어 설정 항목「주소」에 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「주소」에, 비트 지정하지 않는 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.17.4 REGISTER_ADDRESS_RANGE_ERROR

항목	설명
메시지	부정한 레지스터 주소 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 주소 」에, PLC의 프로토콜로는 다룰 수 없는 레지스터 주소가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, PLC 프로토콜로 다룰 수 있는 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오. ※이용할 수 있는 주소, 그 범위에 대해서는, 해당 기기의 매뉴얼 등을 참조하십시오.

15.2.17.5 REGISTER_ADDRESS_DATA_TYPE_ERROR

항목	설명
메시지	데이터형이 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 데이터형 」에, PLC측의 설정과는 다른 데이터형이 지정되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. PLC의 설정에서, 순서1에서 확인한 레지스터의 데이터형을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 데이터형 」에, 순서2에서 확인한 데이터형을 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
For Reference Only-

15.2.17.6 REGISTER_ADDRESS_SIZE_ERROR

항목	설명
메시지	연속 영역 수가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 연속 영역 수 」에, PLC측의 설정과는 다른 연속 영역 수가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. PLC의 설정에서, 순서1에서 확인한 레지스터의 연속 영역 수를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 연속 영역 수 」에, 순서2에서 확인한 연속 영역 수를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

15.2.17.7 REGISTER_WRITE_FIX_VALUE_ERROR

항목	설명
메시지	고정값이 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소
내용	설정 항목「 쓸 값 」이 부정입니다. ※설정 항목「 값 종별 」이「 고정값 」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 쓸 값 」이, 다음의 조건을 만족하고 있는 것을 확인하여 저장하십시오. ・쓸 값의 형이 설정 항목「 데이터형 」과 일치하고 있을 것 ・쓸 값이 문자인 경우, 문자 수가 설정 항목「 연속 영역 수 」 이하일 것 5. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.17.8 REGISTER_WRITE_NO_VALUE_ERROR

항목	설명
메시지	레지스터 쓰기 값의 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소
내용	설정 항목「쓸 값」에 지정된 데이터의 취득에 실패했습니다. ※설정 항목「값 종별」이「컴포넌트」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 입력 포트를 클릭하여, 데이터 모니터를 열어 주십시오. 3. 데이터가 1건 이상 등록된 컬럼을 확인하십시오. 4. 해당 컴포넌트의 설정 화면을 열어 주십시오. 5. 순서1에서 확인한 레지스터의 설정 항목「쓸 값」에, 순서3에서 확인한 컬럼을 지정하여 저장 하십시오. 6. 해당 컴포넌트를 재기동해 주십시오.

15.2.17.9 REGISTER_WRITE_ERROR

항목	설명
메시지	레지스터 쓰기 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터에의 Write 액세스에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시하십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.18 Modbus 쓰기

15.2.18.1 CONNECTION_REFUSED

항목	설명
메시지	접속이 절단
내용	기기와의 접속이 절단되었습니다.
처치 방법	(처치 방법1) 컴포넌트를 재기동해 주십시오. (처치 방법2) 1. 기기측의 설정에서, 통신 가능한 호스트・포트를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 호스트 」「 포트 」에, 순서1에서 확인한 호스트・포트를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.18.2 NETWORK_TIMEOUT

항목	설명
메시지	타임아웃 ※다음의 정보를 포함합니다. ・주소 ・포트 번호 ・설정되어 있는 타임아웃값 ・조작 종별
내용	(조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」의 시간 내에 기기에 접속할 수 없었습니다. (조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」의 시간 내에 기기로부터 데이터를 수신할 수 없었습니다.
처치 방법	1. 메시지에 포함되어 있는 조작 종별을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. (순서1에서 확인한 조작 종별이 connect, close인 경우) 설정 항목「 접속 타임아웃 」을 크게 하십시오. (순서1에서 확인한 조작 종별이 receive, send인 경우) 설정 항목「 읽기 타임아웃 」을 크게 하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역 참조
-For Reference Only-

15.2.18.3 REGISTER_ADDRESS_ERROR

항목	설명
메시지	지정 레지스터가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	비트 지정할 수 없는 레지스터 주소가, 비트 지정되어 설정 항목「 주소 」에 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, 비트 지정하지 않는 레지스터 주소를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.18.4 REGISTER_ADDRESS_RANGE_ERROR

항목	설명
메시지	부정한 레지스터 주소 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 주소 」에, 기기의 프로토콜에서는 다룰 수 없는 레지스터 주소가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「 주소 」에, 기기 프로토콜로 다룰 수 있는 레지스터 어드레스를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오. ※이용할 수 있는 주소, 그 범위에 대해서는, 해당 기기의 매뉴얼 등을 참조하십시오.

AI 번역본
For reference Only

15.2.18.5 REGISTER_ADDRESS_DATA_TYPE_ERROR

항목	설명
메시지	데이터형이 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 데이터형 」에, 기기측의 설정과는 다른 데이터형이 지정되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 기기의 설정에서, 순서1에서 확인한 레지스터의 데이터형을 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 데이터형 」에, 순서2에서 확인한 데이터형을 지정하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

15.2.18.6 REGISTER_ADDRESS_SIZE_ERROR

항목	설명
메시지	연속 영역 수가 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・조작 종별
내용	설정 항목「 연속 영역 수 」에, 기기측의 설정과는 다른 연속 영역 수가 입력되어 있습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 기기의 설정에서, 순서1에서 확인한 레지스터의 연속 영역 수를 확인하십시오. 3. 해당 컴포넌트의 설정 화면을 열어 주십시오. 4. 순서1에서 확인한 레지스터의 설정 항목「 연속 영역 수 」에, 순서2에서 확인한 연속 영역 수를 입력하여 저장하십시오. 5. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.18.7 REGISTER_WRITE_FIX_VALUE_ERROR

항목	설명
메시지	고정값이 부정 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소
내용	설정 항목「쓸 값」이 부정입니다. ※설정 항목「값 종별」이「고정값」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 레지스터의 설정 항목「쓸 값」이, 다음의 조건을 만족하고 있는 것을 확인하여 저장하십시오. ・쓸 값의 형이 설정 항목「데이터형」과 일치하고 있을 것 ・쓸 값이 문자인 경우, 문자 수가 설정 항목「연속 영역 수」 이하일 것 5. 해당 컴포넌트를 재기동해 주십시오.

15.2.18.8 REGISTER_WRITE_NO_VALUE_ERROR

항목	설명
메시지	레지스터 쓰기 값의 취득 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소
내용	설정 항목「쓸 값」에 지정된 데이터의 취득에 실패했습니다. ※설정 항목「값 종별」이「컴포넌트」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소를 확인하십시오. 2. 해당 컴포넌트의 입력 포트를 클릭하여, 데이터 모니터를 열어 주십시오. 3. 데이터가 1건 이상 등록된 컬럼을 확인하십시오. 4. 해당 컴포넌트의 설정 화면을 열어 주십시오. 5. 순서1에서 확인한 레지스터의 설정 항목「쓸 값」에, 순서3에서 확인한 컬럼을 지정하여 저장 하십시오. 6. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
-For Reference Only-

15.2.18.9 REGISTER_WRITE_ERROR

항목	설명
메시지	레지스터 쓰기 에러 ※다음의 정보를 포함합니다. ・레지스터명 ・레지스터 주소 ・프로토콜 에러 번호
내용	레지스터에의 Write 액세스에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 레지스터명, 레지스터 주소, 프로토콜 에러 번호를 확인해 주십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 프로토콜 에러 번호의 대처 방법이 존재하는지 확인하십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시해 주십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우) 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.19 EtherNet/IP 쓰기

15.2.19.1 WRITE_ERROR

항목	설명
메시지	쓰기 에러 ※다음의 정보를 포함합니다. ・에러 요인 ・변수명 or 클래스 ID ・인스턴스 ID ・어트리뷰트 ID
내용	쓰기에 실패했습니다.
처치 방법	1. 메시지에 포함되어 있는 에러 요인, 변수명, 클래스 ID, 인스턴스 ID, 어트리뷰트 ID를 확인하십시오. 2. 기기의 통신 사양의 매뉴얼에, 순서1에서 확인한 에러 요인의 대처 방법이 존재하는지 확인해 주십시오. 3. (대처 방법이 존재하는 경우) 대처 방법을 실시해 주십시오. (대처 방법이 존재하지 않는 경우, 또는 대처 방법에 관한 설명을 잘 모르는 경우)

AI 번역본
-For Reference Only-

항목	설명
	시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.19.2 UNSUPPORTED_FUNCTION_ERROR

항목	설명
메시지	지원하고 있지 않은 기능
내용	현재 상태에서 지원하고 있지 않은 기능입니다. CIPObject에서의 쓰기는 지원하고 있지 않습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 변수의 설정 항목「 설정 방법 」이, 다음의 조건을 만족하고 있는 것을 확인하여 저장하십시오. ・「 변수명 」으로 되어 있을 것 (「 CIPObject 」에서의 쓰기는 지원하고 있지 않습니다.) 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.19.3 FIX_DATA_TYPE_INCOLLECT

항목	설명
메시지	고정값 쓰기의 데이터형 불일치 ※다음의 정보를 포함합니다. ・데이터형 ・쓸 값
내용	고정값의 쓸 값이, 쓰기처의 데이터형과 일치하고 있지 않습니다. ※설정 항목「 값 종별 」이「 고정값 」인 경우에 일어나는 에러입니다.
처치 방법	1. 메시지에 포함되어 있는 데이터형, 쓸 값을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 변수의 설정 항목「 쓸 값 」이, 다음의 조건을 만족하고 있는 것을 확인하여 저장 하십시오. ・쓸 값의 형이 설정 항목「 데이터형 」과 일치하고 있을 것 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
For Reference Only-

15.2.19.4 STRUCT_ERROR

항목	설명
메시지	지정한 구조체가 존재하지 않음 ※다음의 정보를 포함합니다. ・변수명
내용	지정한 구조체가 존재하지 않습니다.
처치 방법	1. 메시지에 포함되어 있는 변수명을 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 순서1에서 확인한 변수의 설정 항목「 변수명 」이, 다음의 조건을 만족하고 있는 것을 확인하여 저장하십시오. ・지정한 변수명이 정확할 것 ・지정한 변수가 구조체일 것 4. 해당 컴포넌트를 재기동해 주십시오.

15.2.20 메일 송신

15.2.20.1 SMTP_AUTHENTICATION_ERROR

항목	설명
메시지	SMTP 인증 에러 ※다음의 정보를 포함합니다. ・사용자명 ・패스워드
내용	SMTP 서버로의 인증이 실패했습니다.
처치 방법	1. SMTP 서버측의 설정에서, 올바른 SMTP 사용자 정보(사용자명・패스워드)를 확인하십시오. 2. 해당 컴포넌트의 설정 화면을 열어 주십시오. 3. 설정 항목「 SMTP사용자명 」「 SMTP패스워드 」에, 순서 1에서 확인한 사용자명・패스워드를 입력하여 저장하십시오. 4. 해당 컴포넌트를 재기동해 주십시오.

AI 번역본
For Reference Only

15.2.21 코어 인터널

15.2.21.1 DELETE_STORAGE_GROUPS_ERROR

항목	설명
메시지	과거 데이터 삭제 에러
내용	영속화된 컬럼의 오래된 데이터 삭제에 실패했습니다. 스토리지의 이상 등의 가능성이 있습니다.
처치 방법	시스템 관리자에게 상담하여, 스토리지의 상태 등을 확인하십시오. (고장의 가능성 등)

15.2.21.2 DELETE_FILE_TYPE_OLD_DATA_ERROR

항목	설명
메시지	파일형 컬럼 데이터 삭제 에러
내용	파일형 컬럼의 오래된 데이터 삭제에 실패했습니다. 스토리지의 이상 등의 가능성이 있습니다. (파일형 컬럼이란, 시리얼라이저가 저장하는 파일을 가리킵니다.)
처치 방법	시스템 관리자에게 상담하여, 스토리지의 상태 등을 확인하십시오. (고장의 가능성 등)

15.2.21.3 ACCESS_FILE_TYPE_DATA_DIR_ERROR

항목	설명
메시지	파일형 컬럼 데이터 삭제 에러 ※다음의 정보를 포함합니다. ・파일 수
내용	디렉터리에 액세스할 수 없기 때문에, 파일형 컬럼의 오래된 데이터 삭제에 실패했습니다.
처치 방법	시스템 관리자에게 상담하여, 스토리지의 상태 등을 확인하십시오. (고장의 가능성 등)

AI 번역본
-For Reference Only-

15.2.21.4 SYNC_COLUMNS_TO_STORAGE_ERROR

항목	설명
메시지	영속화 데이터 쓰기 에러

항목	설명
내용	영속화된 컬럼의 스토리지 쓰기에 실패했습니다. 스토리지의 이상 등의 가능성이 있습니다.
처치 방법	시스템 관리자에게 상담하여, 스토리지의 상태 등을 확인하십시오. (고장의 가능성 등)

15.2.21.5 CORE_INTERRUPT

항목	설명
메시지	전회 실행 시 이상 종료 검출 ※다음의 정보를 포함합니다. ・코어 개시 시각 ・최종 상태 ・최종 상태 천이 시각
내용	SpeeDBee Synapse가 가동하고 있는 머신의 전원이 꺼지는 등으로, 정상적인 종료 절차를 거치지 않고 코어가 정지했습니다.
처치 방법	SpeeDBee Synapse에 등록되어 있는 컴포넌트가, 정상적으로 가동하는 것을 확인하십시오.

15.2.21.6 SPEEDBEE_ERROR_ACCESS

항목	설명
메시지	DB 액세스 에러 검출
내용	SpeeDBee Synapse의 시계열 데이터베이스의 파일 액세스에 실패했습니다.
처치 방법	(대처 방법1) 빈발하는 경우는,「 서버 설정 」을 참고하여 데이터베이스를 초기화하십시오. (대처 방법2) 대처 방법1을 실시해도 해소되지 않는 경우는, 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.21.7 SPEEDBEE_ERROR_CORRUPT

항목	설명
메시지	DB 파일 손상 검출
내용	SpeeDBee Synapse의 시계열 데이터베이스의 파일이 손상되어 있습니다.
처치 방법	(대처 방법1) 빈발하는 경우는,「 서버 설정 」을 참고하여 데이터베이스를 초기화하십시오. (대처 방법2) 대처 방법1을 실시해도 해소되지 않는 경우는, 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.21.8 SPEEDBEE_ERROR_INTR

항목	설명
메시지	DB 전원 차단 검출 구분
내용	SpeeDBee Synapse가 가동하고 있는 머신의 전원이 꺼지는 등으로, 시계열 데이터베이스의 조작이 중단되었습니다.
처치 방법	SpeeDBee Synapse에 등록되어 있는 컴포넌트가, 정상적으로 가동하는 것을 확인하십시오.

15.2.21.9 SPEEDBEE_ERROR_NO_REPORTS

항목	설명
메시지	DB 결손 로그 취득 에러
내용	SpeeDBee Synapse의 시계열 데이터베이스의 결손 검지 로그 취득에 실패했습니다.
처치 방법	(대처 방법1) 빈발하는 경우는,「 서버 설정 」을 참고하여 데이터베이스를 초기화하십시오. (대처 방법2) 대처 방법1을 실시해도 해소되지 않는 경우는, 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
For Reference Only

15.2.22 DB 쿼리 서비스

15.2.22.1 PARAMETER_ERROR_IPADDRESS

항목	설명
메시지	IP 주소 설정 에러
내용	설정 항목「 주소 」가 미입력입니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 주소 」에, IP 주소를 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.22.2 PARAMETER_ERROR_PORT_NUMBER

항목	설명
메시지	포트 번호 설정 에러
내용	설정 항목「 포트 」가 미입력입니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 포트 」에, 포트 번호를 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.22.3 PARAMETER_ERROR_HTTPS_CERTIFICATION

항목	설명
메시지	HTTPS 증명서 에러
내용	설정 메뉴「 HTTP/HTTPS 통신 」의 설정 항목「 SSL/TLS 증명서 」에 업로드된 HTTPS 증명서 파일이 존재하지 않습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.22.4 PARAMETER_ERROR_HTTPS_SECRETKEY

항목	설명
메시지	HTTPS 비밀키 에러
내용	설정 메뉴「HTTP/HTTPS 통신」의 설정 항목「비밀키」에 업로드된 HTTPS 비밀키 파일이 존재하지 않습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.22.5 PARAMETER_ERROR_JWS_SETTING

항목	설명
메시지	JWS 검증의 유효 무효 설정 에러
내용	설정 항목「토큰 인증」이 미지정입니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.22.6 PARAMETER_ERROR_JWS_VERIFICATION

항목	설명
메시지	JWS 검증용 파일 에러
내용	JWS 검증용 파일에 불비가 있습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「제조원 문의」의 순서에 따라 제조원에 문의해 주십시오.

15.2.22.7 DB_QUERY_HTTP_SERVER_START_ERROR

항목	설명
메시지	DB 쿼리 서비스 기동 에러

항목	설명
내용	HTTP 서버의 기동에 실패했습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 호스트 」「 포트 」를 확인하십시오. 3. 시스템 관리자에게 상담하여, 순서 2에서 확인한 호스트 · 포트가, 다른 서비스와 경합하고 있지 않은지 확인하십시오.

15.2.23 에러 매니저

15.2.23.1 PARAMETER_ERROR_INTERVAL

항목	설명
메시지	처리 간격 설정 에러
내용	설정 항목「 처리 간격 」에 부정한 값이 설정되어 있습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 처리 간격 」에, 5부터 180까지의 값을 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.23.2 PARAMETER_ERROR_CORE_PORT

항목	설명
메시지	코어 포트 번호 설정 에러
내용	코어의 포트 번호에 부정한 값이 설정되어 있습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

AI 번역본
-For Reference Only-

15.2.23.3 NETWORK_TIMEOUT_ERROR

항목	설명
메시지	통신 타임아웃 에러
내용	통신 타임아웃이 발생했습니다.
처치 방법	1. 해당 컴포넌트의 설정 화면을 열어 주십시오. 2. 설정 항목「 타임아웃 」에, 더 큰 값을 입력하여 저장하십시오. 3. 해당 컴포넌트를 재기동해 주십시오.

15.2.24 이벤트 데이터 매니저

15.2.24.1 RESTORE_DB

항목	설명
메시지	데이터베이스의 손상에 의해, 백업으로부터 데이터를 복원
내용	i-Trigger의 내부 데이터베이스 액세스 시에, 손상을 검지하여, 백업되어 있는 데이터베이스로부터 데이터를 복원한 것을 알리는 경고입니다.
처치 방법	특별한 처치는 필요하지 않습니다. 다만, 일부 데이터는 소실된 것이 됩니다.

15.2.25 VIDEO 컬렉터（i-Trigger 확장 패키지）

15.2.25.1 READ_FRAME_ERROR

항목	설명
메시지	프레임 불러오기에 실패
내용	네트워크 카메라로부터 수신한 스트리밍 데이터 불러오기에 실패했습니다.
처치 방법	통상은, 처치는 필요하지 않습니다. 다만, 이 에러가 해소되지 않는 경우는, 다음 점을 확인하십시오. 1. 접속하고 있는 카메라가, i-Trigger에 대응하고 있는 카메라인지 확인한다.（참조） 2. 카메라측의 설정을 재검토하십시오. 해상도를 낮추거나, 프레임 레이트의 숫자를 낮추는 등을 시도해 보십시오.

15.2.25.2 MAKE_CONTAINER_ERROR

항목	설명
메시지	mp4 작성 중에 에러가 발생
내용	mp4 파일 작성 중에, 내부 에러가 발생했습니다.
처치 방법	통상은 발생하지 않는 에러입니다. 증상이 계속되는 경우는, 시스템 관리자에게 상담하여,「 제조원 문의 」의 순서에 따라 제조원에 문의해 주십시오.

15.2.26 Event Recorder（ i-Trigger 확장 패키지 ）

이 컴포넌트에서 발생하는 에러는, 모두「 컴포넌트 공통 」의 것이 됩니다.

AI 번역본
-For Reference Only-

15.3 커스텀 컴포넌트 Python-API 레퍼런스

커스텀 컴포넌트를 Python 언어로 개발할 때의 API 사양입니다. 커스텀 컴포넌트의 개발 순서에 대해서는 아래를 참조하십시오.

Python 커스텀 컴포넌트 작성

15.3.1 speedbeesynapse.component.base

Hive 컴포넌트 베이스 모듈

이 모듈은 SpeeDBee Synapse의 컴포넌트를 Python으로 실장할 경우에 임포트하는 모듈입니다.

15.3.2 HiveComponentBase Objects

```
class HiveComponentBase()
```

컴포넌트용 기저 클래스

모든 컴포넌트는 이 클래스를 상속하여 정의할 필요가 있습니다.

Attributes:

- `in_portX` HiveInPort - 입력 포트 X번을 나타내는 인스턴스 변수. 실제로는 in_port1, in_port2와 같이, X에는 숫자가 들어갑니다.
- `out_portX` HiveOutPort - 출력 포트 X번을 나타내는 인스턴스 변수. 실제로는 out_port1, out_port2와 같이, X에는 숫자가 들어갑니다.

15.3.2.1 premain

```
def premain(param: str | dict) -> typing.Optional[ErrorInfoBase]
```

-For Reference Only-

컴포넌트 메인 처리 직전에 호출되는 함수

컴포넌트의 개시 시, 메인 처리의 직전에 호출됩니다. 여기서 파라미터를 읽어들여 컬럼의 작성이나 독자적인 초기화 처리 등을 수행하는 것이 가능합니다.

다만, 이 함수에서의 처리는 전 컴포넌트 내에서 배타 제어가 걸리므로 장시간 처리에 시간이 걸리면, 그 사이, 다른 컴포넌트의 premain이 동작할 수 없다는 점에 유의하십시오. 대신, 여기서 컬럼을 작성하는 경우는 메인 처리에서 작성하는 경우에 비해 고속으로 동작합니다.

정상적으로 초기화 처리 등이 종료된 경우는, 아무것도 반환하지 않고 본 메서드를 종료하십시오. 그 후 main 메서드가 콜됩니다.

어떠한 이유에 의해, 컴포넌트의 실행을 계속할 수 없는 경우는, ErrorInfoBase를 상속한 클래스의 인스턴스를 반환하십시오. 그것에 의해 화면상에도 에러의 내용이 표시되고, 컴포넌트는 정지합니다.
(`main`, `postmain` 메서드는 호출되지 않습니다)

이 메서드의 정의는 필수가 아닙니다.

Arguments:

- `param` str | dict - 컴포넌트에 대한 파라미터

Returns:

- `None` - 정상적으로 종료함
- `ErrorInfoBase` - 어떠한 이유로 컴포넌트를 개시할 수 없는 경우, ErrorInfoBase를 상속한 에러 오브젝트를 반환한다

15.3.2.2 main

```
def main(param: str | dict)
```

컴포넌트 메인 처리 함수

컴포넌트의 메인 처리를 실장하기 위한 메서드입니다. HiveComponentBase를 상속한 클래스에서는 이 메서드를 반드시 오버라이드하여, 실장하십시오.

이 함수의 종료는, 컴포넌트의 처리가 완료되었음을 의미합니다. 정상적으로 종료할 때는 아무것도 반환할 필요가 없습니다. 에러에 의해 종료하는 경우는, ErrorInfoBase를 상속한 클래스의 인스턴스를 반환하십시오. 반환된 인스턴스의 정보는 화면상에 정지 이유로서 표시됩니다.

이 메서드의 밖으로 예외가 송출된 경우, 불명한 에러로서 컴포넌트는 정지합니다. 그 거동이 바람직
하지 않은 경우는, 예외를 캐치하여 처리를 계속하거나, ErrorInfoBase를 상속한 클래스의 인스턴스를 반환하여 본 메서드를 종료하십시오.

AI 번역본
-For Reference Only-

Arguments:

- `param` str | dict - 컴포넌트에 대한 파라미터

Returns:

- `None` - 정상적으로 종료함
- `ErrorInfoBase` - 어떠한 이유로 컴포넌트를 개시할 수 없는 경우, ErrorInfoBase를 상속한 에러 오브젝트를 반환한다

15.3.2.3 postmain

```
def postmain(param: str | dict)
```

컴포넌트 메인 처리 직후에 호출되는 함수

컴포넌트의 정지 시 등, 메인 처리 함수가 종료된 직후에 호출됩니다.
메인 처리가 에러 등으로 종료된 경우에도 반드시 호출되므로, 사용한 리소스의 해제 등을 수행하는 것을 상정하고 있습니다.

이 메서드의 정의는 필수가 아닙니다.

Arguments:

- `param` str | dict - 컴포넌트에 대한 파라미터

15.3.2.4 is_runnable

```
def is_runnable()
```

컴포넌트의 실행 계속 가부 판정

이 컴포넌트의 메인 처리를 계속해도 좋은지를 판정합니다. 컴포넌트는 정기적(적어도 5초 이내)
에 이 함수를 호출하여, 메인 처리를 계속해도 좋은지 확인할 필요가 있습니다. 이 함수가 True를 반환할 때는 그대로 메인 처리를 계속할 수 있지만, False를 반환할 때는, 신속히 메인 처리를 종료할 필요가 있습니다.

Returns:

- True라면 계속 가능, Flase라면 컴포넌트의 정지 요구가 나와 있음

15.3.2.5 get_timestamp

```
def get_timestamp() -> int
```

현재 시각의 취득

나노초 단위의 UNIX 타임으로 표현된 현재 시각을 취득합니다.

AI 번역본
-For Reference Only-

Returns:

- 현재 시각의 타임스탬프.

15.3.2.6 notify_stop

```
def notify_stop()
```

컴포넌트의 정지 요구 통지

이 컴포넌트가 시스템으로부터 정지 요구를 받은 경우에 호출됩니다.
HiveComponentBase를 상속한 클래스에서는 필요에 따라 이 메서드를 오버라이드하십시오.
메서드가 콜된 경우에는, 신속히 컴포넌트의 메인 처리를 정지시킬 필요가 있습니다.

메인 처리 내에서 is_runnable() 을 정기적으로 호출하여 계속 가부를 체크하는 것이라면, 이 함수를 실장할 필요는 없습니다.

15.3.2.7 register_status

```
def register_status(status: RunningStatus)
```

스테이터스 정보의 등록

에러 정보가 추가된 스테이터스를 등록합니다

Arguments:

- status - 에러 정보가 등록된 스테이터스 오브젝트

15.3.2.8 Status

```
def Status()
```

스테이터스 정보의 생성

에러 정보를 등록하기 위한 스테이터스 오브젝트를 생성합니다.

15.3.2.9 interval_iteration

```
def interval_iteration(interval, base_time=0)
```

정기 처리용 이터레이터

컴포넌트에서 일정 시간 간격으로 어떠한 처리를 하기 위한 이터레이터입니다.
지정된 기준 시각부터, 지정한 정기 처리 간격으로 이터레이터에서 값이 yield됩니다. 다만 실제로 처리가 개시되는 타이밍은 엄밀하게 예정 시각과 일치하는 것은 아니며, OS나 디바이스의 부하나 클록 정밀도에 영향을 받습니다.

AI 번역본
For Reference Only

또한, 사용자가 실장하는 정기적인 처리의 실행에, 정기 처리 간격보다 긴 시간을 필요로 한 경우, 다음 정기 처리가 스킵되는 경우가 있습니다. 이 경우, 스킵된 정기 처리의 횟수를 yield의 제2 요소로서 받을 수 있습니다.

본 함수는 컴포넌트의 정지 요구를 받은 경우에는 이터레이션을 종료합니다.

Arguments:

- interval - 정기 처리 간격（단위：나노초）
- base_time - 이터레이션 개시 기준 시각을 나타내는 UNIX 타임（단위：나노초）

Yields:

- [0] - 그때의 이터레이션이 실행될 예정이었던 시각을 나타내는 UNIX 타임（단위：나노초）
- [1] - 직전의 이터레이션의 스킵 수

Examples:

이하와 같이 for문으로 이터레이터로서 사용할 수 있습니다.

```
for [ts, skip] in self.interval_iteration(1000000000):
    #
    # 정기적으로 실행하고 싶은 처리
    #
```

이 경우는 1초 간격으로 「 정기적으로 실행하고 싶은 처리 」의 부분이 실행됩니다.

15.3.2.10 save_running_info

```
def save_running_info(key: str, data: bytes)
```

임의의 정보를 저장한다

임의의 바이트열을 시스템에 기록합니다.
저장된 데이터는, 동일한 컴포넌트라면 시스템의 재기동 후에도 load_running_info()로 참조할 수 있습니다. 또한, 지정된 key에 의해 다른 데이터로서 저장되므로, 복수 저장하고 싶은 경우는 각각 key를 나누십시오.

Arguments:

- `key` str - 저장할 정보를 식별하기 위한 키
- `data` bytes - bytes형의 저장하고 싶은 정보

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.2.11 load_running_info

```
def load_running_info(key: str) -> typing.Optional[bytes]
```

AI 번역은
-For Reference Only-

임의의 정보를 읽어들인다

Arguments:

- `key` str - 읽어들이고 싶은 정보를 식별하기 위한 키

Returns:

- `data` - byte형의 읽어들인 데이터
- `None` - 데이터가 없다

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.2.12 get_common_variable

```
def get_common_variable(varname: str) -> typing.Optional[str]
```

컴포넌트 공통 변수 취득

Arguments:

- `varname` str - 취득하고 싶은 공통 변수의 변수명

Returns:

- 지정한 공통 변수의 값
- `None` - 지정한 변수가 존재하지 않는다

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.2.13 create_lock

```
def create_lock(key: typing.Union[str, bytes],
                interbase: bool = False,
                timeout: typing.Optional[float] = None) -> SynapseLock
```

컴포넌트 간에서의 배타 제어를 위한 락 획득 함수

본 메서드는, 인수 key 로 지정된 문자열 or 바이트열로 식별되는 리소스의 배타 제어를 위한 락을 획득합니다. 동일한 key 에서의 락은 1 개의 스레드에서만 획득할 수 있습니다. 획득한 락은, 반환되는 락 오브젝트의 unlock() 메서드로 해제할 수 있습니다. 혹은, 본 메서드를 with 구로 사용하면 스코프를 빠져나갈 때에 해제됩니다. 또한, 락을 획득한 채로 컴포넌트가 정지한 경우에도 자동으로 락이 해제됩니다.

AI 번역본
-For Reference Only-

인수 interbase 가 true인 경우, 같은 key 를 사용함으로써 다른 컴포넌트 베이스 간에서도 배타 제어가 가능해집니다. 이 경우는 C-API의 hive_lock() 을 사용한 커스텀 컴포넌트와의 배타 제어도 이용 가능합니다. 가 false인 경우, 동일한 컴포넌트 `interbase` 베이스에서만 배타 제어됩니다.

본 메서드를 호출했을 때에 이미 외부의 컴포넌트 인스턴스가 락을 획득하고 있는 경우, 메서드 내에서 실행이 블록됩니다. 블록하는 최대 시간은, 인수 timeout 에서 초 단위로 지정 가능합니다. 타임아웃 시간을 지나도 락을 획득할 수 없는 경우는 HiveTimeoutError 예외가 송출됩니다. timeout 에 0 이하의 값을 지정한 경우는, 실행이 블록되는 일은 없고, 락을 획득할 수 없으면 즉시 타임아웃이 됩니다.
`timeout` 에 None을 지정한 경우는 타임아웃하지 않고 락 획득할 수 있을 때까지 계속 기다립니다.

락 획득을 기다리고 있는 동안에 실행하고 있는 컴포넌트 인스턴스가 정지 요구를 받은 경우, 타임아웃 시간에 관계없이, HiveLockCanceled 예외를 송출합니다.

동일 스레드에서 같은 key 에 대해 이중으로 락할 수는 없습니다. 그 경우는 락 해제 대기 상태가 되므로 타임아웃하거나 정지 요구를 받을 때까지 계속 블록합니다.

Arguments:

- key str|bytes - 배타 제어 대상을 식별하기 위한 임의의 문자열 or 바이트열
- interbase bool - 다른 컴포넌트 베이스 간에서 배타 제어를 수행하는 경우는 True
- timeout float|None - 락을 즉시 획득할 수 없는 경우에 해제를 기다릴 때의 타임아웃 시간

Returns:

SynapseLock 오브젝트

Raises:

- HiveTimeoutError - 지정된 타임아웃 시간 내에 락을 획득할 수 없었다
- HiveLockCanceled - 락을 획득하기 전에 컴포넌트의 정지 요구를 받았다
- HiveApiError - Hive 프레임워크의 각종 에러

Examples:

이하와 같이 with구를 사용할 수 있습니다.

```
while self.is_runnable();
    try:
        with self.create_lock('lock-serial-device', False, 10.0):
            #
            # 배타 제어가 필요한 처리
            #
    except HiveTimeoutError as exc:
        self.log.debug('wait lock')
        continue
    except HiveLockCanceled:
        return
```

AI 번역본

15.3.3 DataType Objects

```
class DataType(IntEnum)
```

데이터형 정의 Enum 클래스

데이터형을 나타내는 Enum 클래스입니다. 컬럼 작성 시에 지정하는 DataType으로서, 이 어트리뷰트를 사용하십시오.

Attributes:

- NONE(0) - 어느 형에도 일치하지 않는 미정의의 형
- BOOLEAN(1) - 진위값형

- `INT8(2)` - 8bit 부호 있는 정수형
- `INT16(3)` - 16bit 부호 있는 정수형
- `INT32(4)` - 32bit 부호 있는 정수형
- `INT64(5)` - 64bit 부호 있는 정수형
- `UINT8(6)` - 8bit 부호 없는 정수형
- `UINT16(7)` - 16bit 부호 없는 정수형
- `UINT32(8)` - 32bit 부호 없는 정수형
- `UINT64(9)` - 64bit 부호 없는 정수형
- `FLOAT(10)` - 32bit 부동소수점형
- `DOUBLE(11)` - 64bit 부동소수점형
- `STRING(13)` - 문자열
- `BINARY(14)` - 바이너리
- `FILE(16)` - 파일

15.3.4 ColumnOption Objects

```
class ColumnOption()
```

컬럼 옵션

출력 포트의 컬럼에 대한 옵션 설정

Attributes:

- `column_type` ColumnType - 컬럼 타입(LOW、MIDDLE、HIGH)
- `samplingrate` float - 샘플링 레이트
- `compression_unit` int - 압축 단위
- `fixedsize_binary` int - 컬럼 데이터 사이즈
- `max_binarysize` int - LO 가변 길이 타입의 데이터 최대 길이

AI 번역본
-For Reference Only-

15.3.5 FileTypeMetaInfo Objects

```
class FileTypeMetaInfo(typing.TypedDict)
```

파일형 데이터 메타 정보

파일형 컬럼에 등록할 파일의 메타 정보를 갖는 클래스입니다. 어느 멤버도 생략 가능합니다.

Attributes:

- `media_type` ColumnType - 미디어 타입(application/json이나 text/csv 등을 설정할 수 있다)
- `filename` float - 파일명의 힌트（파일 에미터에 연결한 경우에 이 파일명이 사용된다）
- `begin_timestamp` int - 이 파일에 수록된 데이터의 시간 범위의 선두 시각（UNIX 타임、단위：나노초）
- `end_timestamp` int - 이 파일에 수록된 데이터의 시간 범위의 말미 시각（UNIX 타임、단위：나노초）

15.3.6 OutColumn Objects

```
class OutColumn()
```

출력 포트의 컬럼

출력 포트의 컬럼에 대한 데이터의 등록 등은 이 클래스의 인스턴스를 통해 수행합니다

Attributes:

- `name` str - 작성 시에 지정한 컬럼명
- `data_type` DataType - 작성 시에 지정한 컬럼의 데이터형
- `data_array` int - 작성 시에 지정한 컬럼의 요소 수 스칼라형인 경우는 0이 됩니다
- `options` ColumnOption - 작성 시에 지정한 컬럼의 옵션

15.3.6.1 insert

```
def insert(*args) -> None
```

컬럼에의 값 등록

이 컬럼에 지정한 타임스탬프 ts 의 값을 등록합니다.
타임스탬프를 생략한 경우는 현재 시각으로서 다루어집니다.
data에는, 이 컬럼 작성 시의 데이터형과 일치하는 데이터를 전달하십시오. 다른 데이터를 전달한 경우, 암묵적으로 변환 가능한 것은 변환됩니다만, 그렇지 않은 경우는 에러가 됩니다.

AI 번역본
-For Reference Only-

Arguments:

- `data` - 등록할 값
- `ts` - 이 값을 등록할 타임스탬프

Raises:

- `TypeError` - 이 컬럼의 데이터형에 일치하지 않는, 부정한 데이터가 전달되었다
- `InvalidTimestamp` - 타임스탬프가 현재 시각과 현저하게 떨어져 있다, 혹은 이전에 등록한 타임스탬프보다 과거로 되어 있다

15.3.6.2 open_file

```
@contextmanager
def open_file()
```

신규 파일의 작성

파일형으로서 작성된 컬럼에 등록할 신규 파일을 오픈하고, 그 파일 오브젝트를 반환합니다.

Returns:

- `FileObject` - 오픈된 파일 오브젝트

Raises:

- `TypeError` - FILE형이 아닌 컬럼에 대해 이 메서드가 호출되었다
- `HiveApiError` - Hive 프레임워크의 각종 에러
- `OSError` - 파일 오픈에 관련된 각종 에러

Examples:

이하와 같이 with구로 사용할 수 있습니다.

```
file_column = self.out_port1.Column("clmfile", DataType.FILE)
with file_column.open_file() as fo:
    fo.write("filedata - line 1\n".encode('utf-8))
    fo.write("filedata - line 2\n".encode('utf-8))
    fo.write("filedata - line 3\n".encode('utf-8))

    file_column.insert_file(fo, ts)
```

15.3.6.3 insert_file

```
def insert_file(fo: typing.IO[typing.AnyStr],
                ts: int = 0,
                meta: typing.Optional[FileTypeMetaInfo] = None) -> None
```

AI 번역본
For Reference Only-

파일의 DB 등록

파일을 파일형의 컬럼에 등록합니다.

Arguments:

- `file` - 오픈된 파일 오브젝트
- `ts` - 이 값을 등록할 타임스탬프
- `meta` - 등록할 파일의 메타 정보

Raises:

- `ValueError` - 등록하려고 한 파일이 이미 클로즈되어 있다

- `HiveApiError` - Hive 프레임워크의 각종 에러
- `OSError` - 파일 조작에 관련된 각종 에러

15.3.6.4 insert_file_move

```
def insert_file_move(pathlike,
                     ts: int = 0,
                     meta: typing.Optional[FileTypeMetaInfo] = None) -> None
```

기존 파일의 DB 등록 (이동)

이미 작성 완료된 파일을 파일형의 컬럼에 등록합니다. 지정된 파일은, DB의 관리 디렉터리로
이동됩니다. 인수로 지정한 경로에는 파일이 남지 않는 것에 주의하십시오. 디렉터리는 등록할 수 없습니다.

Arguments:

- `pathlike` - 등록 대상의 파일의 경로를 나타내는 path-like 오브젝트
- `ts` - 이 값을 등록할 타임스탬프
- `meta` - 등록할 파일의 메타 정보

Raises:

- `ValueError` - 지정한 경로
- `HiveApiError` - Hive 프레임워크의 각종 에러
- `OSError` - 파일 조작에 관련된 각종 에러

15.3.6.5 insert_file_copy

```
def insert_file_copy(pathlike,
                     ts: int = 0,
                     meta: typing.Optional[FileTypeMetaInfo] = None) -> None
```

AI 번역본
For Reference Only

기존 파일의 DB 등록 (복사)

이미 작성 완료된 파일을 파일형의 컬럼에 등록합니다. 지정된 파일은 DB의 관리 디렉터리로 복사되어, 그 복사처의 파일이
컬럼에 등록됩니다. 이후, 인수로 지정한 원본 파일의 경로는 참조되지 않습니다. 디렉터리는 등록할 수
없습니다.

Arguments:

- `pathlike` - 등록 대상의 파일의 경로를 나타내는 path-like 오브젝트
- `ts` - 이 값을 등록할 타임스탬프
- `meta` - 등록할 파일의 메타 정보

Raises:

- `ValueError` - 지정한 경로
- `HiveApiError` - Hive 프레임워크의 각종 에러
- `OSError` - 파일 조작에 관련된 각종 에러

15.3.6.6 insert_file_ref

```
def insert_file_ref(pathlike,
                    ts: int = 0,
                    meta: typing.Optional[FileTypeMetaInfo] = None) -> None
```

기존 파일의 DB 등록 (참조)

이미 작성 완료된 파일을 파일형의 컬럼에 등록합니다. 지정된 파일은, 그 경로 그대로 컬럼에 등록됩니다. 그 때문에 파일을 삭제해 버리면, 이 컬럼의 정보를 다른 컴포넌트가 참조할 때에도
파일을 참조할 수 없게 됩니다. 디렉터리는 등록할 수 없습니다.

Arguments:

- `pathlike` - 등록 대상의 파일의 경로를 나타내는 path-like 오브젝트
- `ts` - 이 값을 등록할 타임스탬프
- `meta` - 등록할 파일의 메타 정보

Raises:

- `ValueError` - 지정한 경로
- `HiveApiError` - Hive 프레임워크의 각종 에러
- `OSError` - 파일 조작에 관련된 각종 에러

15.3.7 AggregationType Objects

```
class AggregationType(IntEnum)
```

AI 번역본
For Reference Only-

집약 종별 정의 Enum 클래스

입력 포트에서 취득하는 컬럼 데이터의 집약 종별을 나타냅니다

Attributes:

- `NONE(0)` - 없음
- `OLDER(1)` - 가장 오래된 값
- `NEWER(2)` - 최신값
- `COUNT(3)` - 건수
- `SUM(4)` - 총합

- `SUMSQ(5)` - 2제곱합
- `SUMSQD(6)` - 편차 제곱합
- `MIN(7)` - 최소
- `MAX(8)` - 최대
- `RANGE(9)` - 범위
- `MEAN(10)` - 산술평균값
- `VAR(11)` - 분산
- `STDEV(12)` - 표준편차
- `UVAR(13)` - 불편분산
- `USTDEV(14)` - 표본 불편 표준편차
- `STDER(15)` - 표준오차
- `CV(16)` - 변동계수
- `MEDIAN(17)` - 중앙값
- `QUARTMIN(18)` - 사분위수
- `QUART1(19)` - 사분위수
- `QUART2(20)` - 사분위수
- `QUART3(21)` - 사분위수
- `QUARTMAX(22)` - 사분위수
- `MODE(23)` - 최빈값
- `MODECOUNT(24)` - 최빈값 개수
- `MDEV(25)` - 평균편차
- `HMEAN(26)` - 조화평균
- `GMEAN(27)` - 기하평균
- `KURT(28)` - 첨도
- `SKEW(29)` - 왜도

AI 번역본
For Reference Only

15.3.8 AggregationTypeSet Objects

```
class AggregationTypeSet()
```

입력 포트의 컬럼에서 데이터 취득할 때에 유효하게 되어 있는 통계 정보를 나타내는 플래그의 세트.

`AggregationType` 의 속성을 `test()` 메서드에 전달함으로써 이 통계 정보가 유효하게 되어 있는지를 확인할 수 있습니다. 또한, 이터레이터를 사용하여 유효하게 되어 있는 통계 정보의 리스트를 취득하는 것도 가능합니다.

Examples:

AggregationType 의 속성을 test() 메서드에 전달함으로써 이 통계 정보가 유효하게 되어 있는지를 확인할 수 있습니다.

```
if a_type_set.test(AggregationType.MAX):
  pring(max: ON")
else:
  pring(max: OFF")
```

또한, 이터레이터를 사용하여 유효하게 되어 있는 통계 정보의 리스트를 취득하는 것도 가능합니다.

```
for a_type in aggregation_type_set:
    print(a_type.name)
```

15.3.9 InColumn Objects

```
class InColumn()
```

입력 포트의 컬럼

이 클래스에서는 정보를 취득할 뿐입니다. 컬럼에 대해 데이터를 등록하는 등의 조작은 할 수 없습니다.

Attributes:

- source_name str - 컬럼을 생성한 컴포넌트의 이름
- data_name str - 컬럼의 이름
- data_type DataType - 컬럼의 형 정보
- array_size int - 컬럼의 배열 요소 수 (스칼라인 경우는 0)
- raw_data bool - 컬럼에서의 데이터 취득 시에 미가공 데이터가 포함되는지 여부를 나타내는 진위값
- stat_type_set AggregationTypeSet - 컬럼에서의 데이터 취득 시에 유효하게 되어 있는 통계 정보를 나타내는 플래그 세트
- output_name str - 컬럼의 출력명. 지정이 없는 경우는 None

AI 번역본

15.3.9.1 get_latest_value

```
def get_latest_value(use_storage: bool = False) -> (int, typing.Any)
```

컬럼의 최신값 취득

컬럼에 등록되어 있는 최신 데이터를 읽어들여, 그 타임스탬프와 값의 튜플을 반환합니다.

Arguments:

- use_storage bool - ※현시점에서는 미지원
  스토리지에 영속화된 데이터로부터의 불러오기를 수행하는 경우에 True를 지정하십시오.
  False인 경우, 메모리상에서만 불러오기를 수행하므로, 영속화된 데이터가 있어도 데이터 없음을 반환할 가능성이

있습니다.
True인 경우에도, 메모리상에 최신 데이터가 있으면 스토리지의 액세스는 수행하지 않습니다.

Returns:

- `(0, None)` : 컬럼에서 읽어들일 수 있는 값이 없는 경우
- `(>0, !=None)` : 읽어들여진 타임스탬프와 값

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.10 InPortReader Objects

```
class InPortReader()
```

입력 포트 리더

입력 포트의 컬럼에서, 데이터를 읽어들이는 클래스입니다.
이 클래스는 직접 인스턴스화하지 않고, `self.in_portX.ContinuousReader()` 、 혹은,
`self.in_portX.TimeRangeReader()` 함수를 사용하십시오.

15.3.10.1 read

```
def read() -> typing.Optional[HiveWindowData]
```

컬럼 데이터 불러오기

입력 포트에서 컬럼 데이터를 읽어들입니다.

Returns:

- `HiveWindowData` - 읽어들인 윈도우 데이터를 반환합니다.
- `None` - 현시점에서 읽어들일 수 없는 경우는 None이 반환됩니다.

AI 번역본
-For Reference Only-

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.11 HiveOutPort Objects

```
class HiveOutPort()
```

출력 포트를 나타내는 클래스

출력 포트에 대한 컬럼의 작성 등은 이 클래스의 인스턴스를 통해 수행합니다.
이 클래스의 인스턴스를 직접 생성할 수는 없습니다. 컴포넌트의 개시 시에, 어트리뷰트 `self.out_port1` 과 같이 처음부터 액세스할 수 있으므로 이것을 사용하십시오.

15.3.11.1 Column

```
def Column(name: str,
           data_type: DataType,
           data_array: int = 0,
           opt: typing.Optional[ColumnOption] = None) -> OutColumn
```

컬럼 생성

이 출력 포트 내에, 지정된 명칭의 컬럼을 생성합니다.

Arguments:

- `name` - 작성할 컬럼명
- `data_type` - 작성할 컬럼의 데이터형
- `data_array` - 작성할 컬럼의 요소 수(스칼라형인 경우는 0을 지정)
- `opt` - 컬럼 옵션

Returns:

- OutColumn

Raises:

- `HiveApiError` - Hive 프레임워크의 각종 에러

15.3.11.2 register_confirmed_time

```
def register_confirmed_time(ts: int = 0) -> None
```

출력 포트에의 확정 시각 등록

이 출력 포트의 컬럼에의 데이터 등록이, 지정한 시각 ts 까지는 확정된 것을 통지합니다. 이 등록에 의해, 지정한 시각 ts 보다 이전 시각의 데이터는 읽어들일 수 있다는 것을, 플로우 링크를 통해 연결되어 있는 다른 컴포넌트가 알 수 있습니다.

※이 API의 호출은 필수가 아닙니다.

Arguments:

- `ts` - 출력 포트의 컬럼에의 데이터 등록이 확정되어 있는 시각 ※생략한 경우는 현재 시각이 됩니다

AI 번역본

15.3.12 HiveInPort Objects

```
class HiveInPort()
```

입력 포트를 나타내는 클래스

입력 포트에서 데이터를 취득하는 것과 같은 처리는 이 클래스의 인스턴스를 통해 수행합니다

15.3.12.1 get_columns

```
def get_columns() -> list[InColumn]
```

컬럼 리스트의 취득

이 입력 포트에 접속되어 있는 전 컬럼의 정보를 취득합니다

Returns:

- list[InColumn]

15.3.12.2 ContinuousReader

```
def ContinuousReader(start: int) -> InPortReader
```

계속 컬럼 리더의 생성

이 입력 포트에 접속되어 있는 컬럼에서, 계속적으로 데이터를 읽어들이는 ContinuousReader를 생성합니다

Arguments:

- `start` - 입력 포트에서의 불러오기를 개시할 시각（UNIX 타임、단위：나노초）

Returns:

- ContinuousReader

AI 번역본
-For Reference Only-

Examples:

이하와 같이 with구로 reader를 취득하여 사용할 수 있습니다.

```
with self.in_port1.ContinuousReader(start=self.get_timestamp()) as reader:
    while self.is_runnable():
        window_data = reader.read()
        if not window_data:
            continue
            :
```

15.3.12.3 TimeRangeReader

```
def TimeRangeReader(start: int, end: int) -> InPortReader
```

시간 범위 데이터 리더의 생성

지정한 컬럼, 혹은 입력 포트에 접속되어 있는 전 컬럼에서, 지정한 시간 범위의 데이터를 읽어들이는 TimeRangeReader를 생성합니다

Arguments:

- `start` - 입력 포트에서의 불러오기를 개시할 시각 (UNIX 타임、단위 : 나노초)
- `end` - 입력 포트에서의 불러오기를 종료할 시각 (UNIX 타임、단위 : 나노초)
- `column` - 데이터를 취득할 컬럼(지정이 없는 경우에는 입력 포트의 전 컬럼)

Returns:

- data

Examples:

이하와 같이 with구로 reader를 취득하여 사용할 수 있습니다.

```
with self.in_port1.ContinuousReader(start=self.get_timestamp()) as reader:
    for window_data in reader:
        :
```

15.3.13 RunningStatus Objects

```
class RunningStatus()
```

15.3.13.1 add_error

```
def add_error(err: ErrorInfoBase)
```

에러 정보 등록

스테이터스 오브젝트에 에러 정보를 등록합니다. 이 오브젝트에 등록한 것만으로는 저장되지 않습니다. 최종적으로는 HiveComponentBase.register_status를 호출하여 시스템에 등록하십시오.

Arguments:

- `err` - 에러 정보

15.3.13.2 ErrorType

```
def ErrorType(error_id: str, *param_names: str)
```

AI 번역본
For Reference Only

에러 정보 클래스 생성

에러 정보를 관리하는 클래스를 생성합니다. 실제의 에러 정보는, 이 함수로 생성한 클래스에서 인스턴스를 생성하십시오.

Arguments:

- `error_id` - 에러 정보를 식별하는 문자열
- `param_names` - 에러 정보의 파라미터의 명칭

15.3.14 ErrorInfoBase Objects

```
class ErrorInfoBase(Exception)
```

스테이터스에 등록하기 위한 에러 정보

컴포넌트의 에러의 종별이나 상세를 보유하기 위한 클래스입니다.
이 클래스는, 직접 인스턴스화하거나 상속하거나 하지 않고, ErrorType 함수를 사용하여 생성한 클래스를 인스턴스화하십시오.

Attributes:

- `errorTypeIndex int` - 에러 파라미터의 타입
- `params list[ErrorParam]` - 에러 파라미터

15.3.15 HiveLogLevel Objects

```
class HiveLogLevel(IntEnum)
```

로그 레벨형 정의 Enum 클래스

로그 레벨을 나타내는 Enum 클래스입니다.

AI 번역본
-For Reference Only-

Attributes:

- `ERROR(0)` - 에러 로그
- `WARNING(1)` - 경고 로그
- `INFO(2)` - 통상 정보 로그
- `DEBUG(3)` - 디버그용 로그
- `TRACE(4)` - 트레이스 로그

15.3.16 HiveLog Objects

```
class HiveLog()
```

로그 출력 클래스

15.3.16.1 get_current_level

```
def get_current_level()
```

로그 레벨 취득

Returns:

- `HiveLogLevel` - 현재의 로그 레벨 Enum

15.3.16.2 error

```
def error(*msg)
```

에러 로그 출력

Arguments:

- `msg` - 로그 출력할 문자열

15.3.16.3 warning

```
def warning(*msg)
```

경고 로그 출력

Arguments:

- `msg` - 로그 출력할 문자열

AI 번역본
-For Reference Only-

15.3.16.4 info

```
def info(*msg)
```

통상 정보 로그 출력

Arguments:

- `msg` - 로그 출력할 문자열

15.3.16.5 debug

```
def debug(*msg)
```

디버그 로그 출력

Arguments:

- `msg` - 로그 출력할 문자열

15.3.16.6 trace

```
def trace(*msg)
```

트레이스 로그 출력

Arguments:

- `msg` - 로그 출력할 문자열

15.3.16.7 exception

```
def exception(exc: Exception)
```

예외 전용 에러 로그 출력

Arguments:

- `exc` - 로그 출력할 예외

15.3.17 HiveWindowData Objects

```
class HiveWindowData()
```

윈도우 데이터 클래스

AI 번역본
-For Reference Only-

입력 포트에서 컬럼의 데이터를 취득할 때에, 특정 시간 단위로 정리된 데이터의 덩어리입니다. 시간 단위 가 어느 정도가 되는지는 입력 포트의 설정에 따라 다릅니다.

Attributes:

- `time_range` tuple[int:2] - 이 윈도우의 시간 범위를 나타내는 2 요소의 튜플. 최초의 요소가 개시 시각, 2 번째의 요소가 종료 시각을 나타낸다. 어느 쪽도 나노초 단위.
- `window_id` int - 윈도우를 식별하는 수치. 윈도우마다 다른 값이 취득되지만, 항상 1 개씩 증가한다고는 할 수 없는 것에 주의.
- `event_id` int - 이벤트 범위를 식별하는 수치. 이벤트 정의되어 있지 않은 경우나 이벤트 범위 외에서는 0으로 고정. 이벤트 중에는 1 이상의 값이 들어간다.

- `records` list[HiveRecord] - 이 윈도우 내의 레코드 리스트
- `columns` list[InColumn] - 취득한 입력 포트에 접속된 컬럼의 목록. HiveInPort.get_columns()로 취득한 것과 같음.
- `event` HiveInPortEventSetting - 입력 포트에 설정된 이벤트 정보

15.3.18 HiveRecordType Objects

```
class HiveRecordType(IntEnum)
```

레코드 종별 Enum 클래스

레코드 종별을 나타내는 Enum 클래스입니다.

Attributes:

- `RAW_DATA(0)` - 미가공 데이터
- `STATISTICS(1)` - 집약 데이터

15.3.19 HiveRecord Objects

```
class HiveRecord()
```

레코드 클래스

입력 포트에서 컬럼의 데이터를 취득할 때에, 동일 시각의 데이터를 정리한 것.
다른 컬럼의 데이터라도 동일 시각의 것이 있으면 모두 1 개의 인스턴스로 정리됩니다. 레코드는 2 종류, 미가공 데이터와 집약 데이터가 있으며, record_type 으로 식별할 수 있습니다.

Attributes:

- `record_type` HiveRecordType - 이 윈도우 내의 레코드 리스트
- `timestamp` int - 이 레코드에 포함되는 데이터의 타임스탬프. UNIX 타임（단위 나노초）
- `data` list[HiveRecordData] - 이 레코드의 데이터의 리스트

AI 번역본
-For Reference Only-

15.3.20 HiveRecordData Objects

```
class HiveRecordData()
```

레코드 데이터 클래스

1 레코드 내의 각 컬럼의 데이터.

Attributes:

- `value` any - 실데이터. 수치, 문자열, 배열 등, 컬럼의 형에 따라 다른 형식.
- `column` InColumn - 대상 컬럼을 나타내는 오브젝트
- `stat_type` AggregationType - 이 데이터의 집약 종별

15.3.21 HiveCompError Objects

```
class HiveCompError()
```

컴포넌트 에러 정보 클래스

컴포넌트 실행 중, 혹은, 에러로 정지한 경우의 에러 정보

Attributes:

- `error_type` str - 컴포넌트의 에러의 종별을 나타내는 문자열
- `parameters` list[any] - 에러의 부가 정보

15.3.22 HiveCompStatus Objects

```
class HiveCompStatus()
```

컴포넌트 스테이터스 클래스

컴포넌트의 스테이터스와 에러 정보

Attributes:

- `status` str - 컴포넌트의 실행 상태를 나타내는 문자열
- `errors` list[HiveCompError] - 컴포넌트 실행 중, 혹은, 에러로 정지한 경우의 에러 정보

AI 번역본
For Reference Only

15.3.23 HiveInPortEventSetting Objects

```
class HiveInPortEventSetting()
```

입력 포트의 이벤트 설정 정보 클래스

입력 포트에 설정된 이벤트 정보를 보유하는 클래스 이 클래스의 속성값을 변경해도 실제의 입력 포트 설정에 는 반영되지 않습니다

Attributes:

- `source_name` str - 이벤트 판정의 대상이 되고 있는 컴포넌트명
- `data_name` str - 이벤트 판정의 대상이 되고 있는 컬럼명

- pre_time int - 이벤트 구간 전방의 확장 시간 범위
- post_time int - 이벤트 구간 후방의 확장 시간 범위

15.3.24 SynapseLock Objects

```
class SynapseLock()
```

컴포넌트 간 배타 제어를 위한 락 오브젝트

HiveComponentBase.create_lock() 으로 락을 획득했을 때에 취득할 수 있는 오브젝트입니다. 통상은 이 오브젝트를 조작할 필요는 없습니다. create_lock() 의 사용 예와 같이, with 구를 사용하고 있으면 그 스코프를 빠져나갈 때에 자동으로 락이 해제됩니다.

with 구를 사용하지 않는 경우는, 이 클래스의 메서드 acquire() , release() 로 락의 획득 해제가 가능합니다.

15.3.24.1 acquire

```
def acquire() -> None
```

락을 획득한다

이 오브젝트가 생성되었을 때의 인수 key, timeout, interbase 를 사용하여 락을 요구합니다. 락 획득 성공한 경우는 아무것도 반환하지 않고 메서드가 종료합니다.

Raises:

- HiveTimeoutError - 지정된 타임아웃 시간 내에 락을 획득할 수 없었다
- HiveLockCanceled - 락을 획득하기 전에 컴포넌트의 정지 요구를 받았다
- HiveApiError - Hive 프레임워크의 각종 에러

15.3.24.2 release

```
def release() -> None
```

락을 해제한다

이 오브젝트가 보유하고 있는 락을 해제하여, 다른 컴포넌트/스레드가 락을 획득할 수 있도록 합니다.

시 버 여 부
-For Reference Only-

15.3.25 HiveTimeoutError Objects

```
class HiveTimeoutError(Exception)
```

함수, 메서드 내에서 블로킹하는 경우의 타임아웃 시간에 도달한 경우에 송출되는 예외입니다

15.3.26 HiveLockCanceled Objects

```
class HiveLockCanceled(Exception)
```

SynapseLock 오브젝트에서, 락 획득 요구 중에 컴포넌트가 정지된 경우에 송출되는 예외입니다

AI 번역본
-For Reference Only-

15.4 커스텀 컴포넌트 C-API 레퍼런스

커스텀 컴포넌트를 C 언어로 개발할 때의 API 사양입니다. 커스텀 컴포넌트의 개발 순서에 대해서는 아래를 참조하십시오.

C 언어 커스텀 컴포넌트 작성

15.4.1 출력 포트 함수

15.4.1.1 hive_outport_create_column() ... 신규 컬럼 작성

컴포넌트가 수집한 데이터를 SpeeDBee Synapse에 등록하기 위한 컬럼의 핸들을 취득하는 함수

정의

```
HIVE_OUTCOLUMN hive_outport_create_column(HIVE_COMPONENT component, int out_port_no, const char
*name, HIVE_DATA_TYPE type, HIVE_COLUMN_OPTIONS opt)
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`int out_port_no`	컬럼을 작성할 포트 번호 OUTPORT1, OUTPORT2 등의 매크로 정수를 지정합니다. 통상은 OUTPORT1만을 사용하면 OK입니다.
3	`const char *name`	컬럼명 작성할 컬럼의 이름을 지정합니다
4	`HIVE_DATA_TYPE type`	데이터형 작성할 컬럼의 형을 지정합니다
5	`HIVE_COLUMN_OPTIONS opt`	컬럼 옵션 작성할 컬럼의 옵션을 지정합니다. 상세는 HIVE_COLUMN_OPTIONS를 참조하십시오.

AI 번역본
-For Reference Only-

반환값

- 컬럼 핸들HIVE_OUTCOLUMN : 컬럼을 다루는 핸들. 나중에 컬럼에 데이터를 등록하거나, 컬럼 설정을 변경할 때에 사용합니다

- NULL: 컬럼의 작성에 실패

설명

컴포넌트가 수집한 데이터를 SpeeDBee Synapse에 등록하기 위한 컬럼의 핸들을 취득하는 함수입니다. 컬럼에 데이터를 등록할 때에는 이 함수가 반환한 컬럼 핸들을 사용합니다.

컬럼명에는 "/"는 사용할 수 없습니다. 그 외의 문자는 사용 가능합니다만, ASCII 이외의 문자를 사용하는 경우는 UTF-8 인코딩되어 있을 필요가 있습니다. 또한, 컬럼명은 127바이트까지로 하십시오.

1 개의 컴포넌트 내에서 동일한 컬럼명의 컬럼을 취득한 경우, 마지막에 취득한 컬럼 핸들만이 유효하게 됩니다. 그 때문에, 본 함수로 같은 컬럼명의 컬럼 핸들을 취득하면, 컬럼의 설정을 변경할 수 있습니다. 다만 사전에 취득하고 있던 컬럼 핸들은 무효가 되므로 사용하지 마십시오.

컬럼의 옵션 지정에 대한 상세는 HIVE_COLUMN_OPTIONS를 참조하십시오.

15.4.1.2 hive_outport_register_confirmed_time() ... 스테이터스 정보에의 컬럼 데이터 확정 시각의 등록

출력 포트에 컬럼 데이터의 확정 시각을 등록하는 함수

정의

```
void hive_outport_register_confirmed_time(HIVE_COMPONENT component, int out_port_no, hive_time_t
ts);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`int out_port_no`	등록 대상의 출력 포트 번호
3	`hive_time_t ts`	등록할 시각

AI 번역본
For Reference Only-

반환값

없음

설명

출력 포트에 컬럼 데이터의 확정 시각을 등록합니다.

이 정보가 등록됨으로써, 이 컴포넌트는, 등록한 시각보다 이전의 데이터는 모두 등록이 완료된 것을 프레임워크에 전달할 수 있습니다.
이에 의해 데이터를 받아들이는 측의 컴포넌트의 처리가 최적화되어, 보다 적은 지연으로 데이터를 다룰 수 있게 됩니다.

데이터를 등록하는 컴포넌트에서는, 어떠한 타임스탬프의 데이터를 등록한 후에, 향후 그보다 이전의 타임스탬프의 데이터를 등록하는 일이 없다는 것이 확정된 타이밍에 그 시각을 등록하십시오.
이 API를 사용하지 않아도 각종 컴포넌트는 동작 가능합니다만, 데이터를 받아들이는 측의 컴포넌트에서 데이터를 받아들일 때에 다소의 지연이 발생합니다.

반대로, 이 API로 어떤 타임스탬프를 등록한 경우, 그 타임스탬프보다 이전의 타임스탬프의 데이터를 등록 하지 않도록 주의가 필요합니다. 등록해 버리면, 그 데이터를 받아들이는 컴포넌트에서 데이터를 놓칠 가능성이 있습니다.

15.4.2 출력 컬럼 함수

15.4.2.1 hive_outcolumn_insert() ... 데이터 등록

컬럼에 데이터를 등록하는 함수

정의

```
bool hive_outcolumn_insert(HIVE_OUTCOLUMN column, const void *data);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()에서 취득한 컬럼 핸들을 지정합니다
2	`void *data`	등록할 데이터에의 포인터

-For Reference Only-

반환값

- true: 등록 성공
- false: 등록 실패

설명

제1 인수에 지정한 컬럼에 단일의 데이터를 등록합니다. 제2 인수 data 에는, 등록할 컬럼의 데이터형에 맞춘 메모리 주소를 지정할 필요가 있습니다.

예

- uint16형의 스칼라인 경우, 2바이트의 메모리 영역의 선두 주소
- float형의 5요소 배열인 경우, 20바이트의 메모리 영역의 선두 주소

본 함수로 등록한 데이터의 타임스탬프는, 본 함수 호출 시의 현재 시각이 됩니다. 시각을 지정할 필요가 있는 경우는 hive_outcolumn_insert_ts()를 사용하십시오.

15.4.2.2 hive_outcolumn_insert_ts() … 타임스탬프 지정 데이터 등록

컬럼에 타임스탬프 지정의 데이터를 등록하는 함수

정의

```
bool hive_outcolumn_insert_ts(HIVE_OUTCOLUMN column, const void *data, hive_time_t ts);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()에서 취득한 컬럼 핸들을 지정합니다
2	`const void *data`	등록할 데이터에의 포인터
3	`hive_time_t ts`	등록할 데이터의 타임스탬프

반환값

- true: 등록 성공
- false: 등록 실패

설명

제1 인수, 제2 인수는 hive_outcolumn_insert()와 동일합니다.
제3 인수 ts 에는 타임스탬프를 지정하십시오. 현재 시각의 타임스탬프는 hive_timestamp()로 취득할 수 있습니다.

AI 번역본
For Reference Only-

본 함수는, 복수의 컬럼에 대해 동일한 타임스탬프로 데이터를 등록하는 것과 같은 장면에서의 이용을 상정하고 있습니다.

사용 예 : 같은 타임스탬프로 복수 컬럼에 데이터를 등록한다

```
 :
hive_time_t ts0 = hive_timestamp();
hive_outcolumn_insert_ts(clmA, dataA, ts0);
hive_outcolumn_insert_ts(clmB, dataB, ts0);
hive_outcolumn_insert_ts(clmC, dataC, ts0);
 :
```

hive_outcolumn_insert()을 사용한 경우, clmA, clmB, clmC 각각에서 다른 타임스탬프가 되어 버립니다. 본 시스템에서는, 동일한

컬럼에 등록하는 데이터의 타임스탬프는, 기본적으로 시계열순으로 등록할 필요가 있습니다. 그 때문에, 어떤 컬럼에 시각 T0 의 데이터를 등록한 후에 T0 보다 이전의 타임스탬프의 데이터를 같은 컬럼에 등록할 수는 없습니다.

15.4.2.3 hive_outcolumn_insert_binary() ... 바이너리형 데이터 등록

BINARY형 컬럼에 가변 길이 바이너리 데이터를 등록하는 함수

정의

```
bool hive_outcolumn_insert_binary(HIVE_OUTCOLUMN column, const void *data, size_t n);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()에서 취득한 컬럼 핸들을 지정합니다
2	`const void *data`	등록할 데이터에의 포인터
3	`size_t n`	등록할 데이터의 바이트 수

AI 번역본
-For Reference Only-

반환값

- true: 등록 성공
- false: 등록 실패

설명

BINARY형으로서 작성된 컬럼에 대해 임의 길이의 데이터를 등록하는 함수입니다.
제1 인수, 제2 인수는 hive_outcolumn_insert()와 동일합니다. 제3 인수 n 에는, 등록할 데이터의 바이트 수를 지정하십시오.

본 함수로 등록한 데이터의 타임스탬프는, 본 함수 호출 시의 현재 시각이 됩니다. 시각을 지정할 필요가 있는 경우는 hive_outcolumn_insert_binary_ts()를 사용하십시오.

15.4.2.4 hive_outcolumn_insert_binary_ts() ... 타임스탬프 지정 바이너리형 데이터 등록

BINARY형 컬럼에 가변 길이 바이너리 데이터를 타임스탬프 지정으로 등록하는 함수

정의

```
bool hive_outcolumn_insert_binary_ts(HIVE_OUTCOLUMN column, const void *data, size_t n,
hive_time_t ts);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()에서 취득한 컬럼 핸들을 지정합니다
2	`const void *data`	등록할 데이터에의 포인터
3	`size_t n`	등록할 데이터의 바이트 수
4	`hive_time_t ts`	등록할 데이터의 타임스탬프

반환값

- true: 등록 성공
- false: 등록 실패

설명

제1 인수, 제2 인수, 제3 인수는 hive_outcolumn_insert_binary()와 동일합니다.
제3 인수 `ts` 에는 타임스탬프를 지정하십시오. 타임스탬프에 대한 주의 사사항은
hive_outcolumn_insert_ts()를 참조하십시오.

AI 번역본
-For Reference Only-

15.4.2.5 hive_outcolumn_insert_multisamples() ... 복수 샘플 데이터 등록

1 개의 컬럼에 복수의 타임스탬프의 데이터를 일괄로 등록하는 함수

정의

```
bool hive_outcolumn_insert_multisamples(HIVE_OUTCOLUMN column, const void *data, hive_time_t ts,
int sample_count);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()이 반환한 컬럼 핸들을 지정합니다
2	`const void *data`	등록할 데이터에의 포인터
3	`hive_time_t ts`	등록할 데이터의 타임스탬프
4	`int sample_count`	등록할 데이터의 샘플 수

반환값

- true: 등록 성공
- false: 등록 실패

설명

본 함수는 컬럼 옵션에서 컬럼 타입 MIDDLE로 샘플링 레이트를 지정한 경우에만 사용할 수 있습니다.
제1, 제2, 제3 인수는 hive_outcolumn_insert_ts()와 동일합니다.
제4 인수 sample_count에는 등록할 데이터의 샘플 수를 지정하십시오. 등록할 복수의 데이터는 제2 인수의 data가 가리키는 메모리상에 연속하여 배치되어 있어야 합니다.

사용 예: 연속하는 복수 샘플의 데이터를 등록한다

```
  HIVE_COLUMN_OPTIONS opt = {
    .type = column_type_middle,
    .sampling_rate = 10;
  };
  HIVE_OUTCOLUMN clmA = hive_outport_create_column(compo, "clmA",
HIVE_DATA_SCALAR(HIVE_TYPE_UINT16), opt)
  uint16_t data[10] = {.}
   :
  get_data(data, 10);
  hive_time_t ts0 = hive_timestamp();
  hive_outcolumn_insert_multisamples(clmA, data, ts0, 10);
   :
```

For Reference Only

위 예의 경우, data[0]은 지정한 타임스탬프 ts0으로 등록되고, 이어지는 data[1]은 ts0 + 0.1초, data[2]는 ts0 + 0.2초로 등록됩니다.
각 타임스탬프의 간격은 opt.sampling_rate에서 산출됩니다.

15.4.2.6 hive_outcolumn_open_file() ... 파일 작성

파일형 컬럼용 파일을 작성하는 함수

정의

```
HIVE_FILE *hive_outcolumn_open_file(HIVE_OUTCOLUMN column);
```

인수

No	항목	설명
1	`HIVE_OUTCOLUMN column`	컬럼 핸들 hive_outport_create_column()이 반환한 컬럼 핸들을 지정합니다

반환값

- HIVE_FILE포인터HIVE_FILE : 작성된 파일 포인터를 갖는 HIVE_FILE의 포인터.
- NULL: 파일 작성에 실패

설명

본 함수는 파일형 컬럼에 데이터를 등록하기 위한 파일을 작성하고, 그 파일 포인터를 가진 HIVE_FILE 포인터를 반환합니다.

반환된 HIVE_FILE 내의 파일 포인터를 사용하여 파일에 쓰기를 수행하십시오. 그 후,
hive_outcolumn_insert_file()을 사용하여 컬럼에 파일을 등록할 수 있습니다.

사용 예: 파일 작성과 데이터 쓰기

```
  HIVE_OUTCOLUMN clmfile  = hive_outport_create_column(comp, OUTPORT1, "clmfile",
HIVE_DATA_SCALAR(HIVE_TYPE_FILE),   HIVE_COLUMN_OPTION_NONE);
   :
  HIVE_FILE *hf = hive_outcolumn_open_file(clmfile);
  if (hf == NULL) {
    HIVE_API_ERROR error = hive_get_api_error();
    HIVE_LOG_ERROR("hive_outcolumn_open_file error(0x%x,%d,%d)", error.code, error.system_errno,
error.db_errno);
  }
   :
  for (int i = 0; i < 10; i++) {
    fprintf(hf->fp, "data: %d\n");
  }
   :
  HIVE_FILE_METAINFO meta = {
   .filename = "a",
   .media_type = "text/plain",
   .begin_timestamp = 1,
   .end_timestamp = 2,
  };

  if (!hive_outcolumn_insert_file(hf, 0, &meta)) {
    HIVE_API_ERROR error = hive_get_api_error();
    HIVE_LOG_ERROR("hive_outcolumn_insert_file error(0x%x,%d,%d)", error.code,
error.system_errno, error.db_errno);
  }
```

AI 번역본
-For Reference Only-

본 함수가 반환한 HIVE_FILE 포인터는 반드시 hive_outcolumn_insert_file() 또는 hive_outcolumn_dispose_file()을 사용하여 리소스를 해제하십시오. 그렇게 하지 않을 경우, 리소스가 해제되지 않아 메모리 리크의 원인이 됩니다.

15.4.2.7 hive_outcolumn_insert_file() ... 파일형 데이터의 등록

파일형 컬럼에 데이터를 등록하는 함수

정의

```
bool hive_outcolumn_insert_file(HIVE_FILE *hf, hive_time_t ts, const HIVE_FILE_METAINFO *meta);
```

인수

No	항목	설명
1	`HIVE_FILE *hf`	HIVE_FILE의 포인터 hive_outcolumn_open_file()이 반환한 HIVE_FILE 포인터를 지정하니다
2	`hive_time_t ts`	등록할 데이터의 타임스탬프
3	`HIVE_FILE_METAINFO *meta`	등록할 파일의 메타 정보

반환값

- true: 등록 성공
- false: 등록 실패

설명

본 함수는 파일형 컬럼에 데이터를 등록합니다.
제1 인수 hf에는 hive_outcolumn_open_file()로 작성한 HIVE_FILE 포인터를 지정합니다. 제2 인수 ts에는 타임스탬프를 지정하십시오. 0이 지정되면 현재 시각으로 등록됩니다. 제3 인수 meta에는 HIVE_FILE_METAINFO에 등록할 파일의 메타 정보를 넣어 지정하십시오. 메타 정보는 이 파일형 컬럼의 데이터를 받는 쪽에서 참조되지만, 불필요하면 NULL 지정도 가능합니다.

사용 예는 hive_outcolumn_open_file()을 참조하십시오.

AI 번역본 -For Reference Only-

15.4.2.8 hive_outcolumn_dispose_file() ... 파일 파기

파일을 파기하는 함수

정의

```
void hive_outcolumn_dispose_file(HIVE_FILE *hf);
```

인수

No	항목	설명
1	`HIVE_FILE *hf`	HIVE_FILE의 포인터 hive_outcolumn_open_file()이 반환한 HIVE_FILE 포인터를 지정합니다

반환값

없음

설명

본 함수는 hive_outcolumn_open_file()로 작성한 파일형 컬럼용 파일과 HIVE_FILE을 파기합니다.
hive_outcolumn_open_file()로 파일을 오픈했으나 도중에 불필요해진 경우에는 본 함수를 호출하여 파일을 파기하십시오.

15.4.3 입력 컬럼 함수

15.4.3.1 hive_inport_get_column_list() ... 입력 컬럼 정보 취득

입력 포트의 컬럼 정보를 취득한다

정의

```
const HIVE_INCOLUMN_LIST *hive_inport_get_column_list(HIVE_COMPONENT component, int inPortNo);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`int in_port_no`	입력 포트 번호

반환값

- HIVE_INCOLUMN_LIST: 컬럼 정보의 리스트
- NULL: 메모리 부족 등으로 인해 컬럼 정보를 취득할 수 없는 경우

설명

본 함수는 지정한 입력 포트에 접속된 다른 컴포넌트의 컬럼 정보(컴포넌트명, 컬럼명, 형 등)를 취득합니다.

반환값으로 받은 구조체의 메모리는 시스템이 내부에서 관리하고 있으므로, 변경, 해제 등을 하지 마십시오. 또한, 다른 hive_inport_xxxx API를 호출하면 내용이 변화하는 경우가 있으므로, 저장해 둘 필요가 있는 경우는 다른 메모리 영역에 복사하는 등의 조작을 해 주십시오.

15.4.3.2 hive_inport_continuous_reader() ... 컬럼 데이터 계속 검색용 Reader 취득

입력 포트의 컬럼에서 계속적으로 데이터를 취득하기 위한 Reader를 취득한다

정의

```
HIVE_CONTINUOUS_READER *hive_inport_continuous_reader(HIVE_COMPONENT component, int in_port_no,
hive_time_t search_begin);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`int in_port_no`	입력 포트 번호
3	`hive_time_t search_begin`	검색 범위 개시 시각

반환값

- HIVE_CONTINUOUS_READER: 컬럼 데이터 계속 검색용 Reader

AI 번역본
-For Reference Only-

설명

본 함수는 지정한 입력 포트에 접속된 다른 컴포넌트의 컬럼 데이터를 계속하여 검색하기 위한 Reader를 취득합니다.
취득한 Reader를 hive_continuous_reader_read()에서 사용함으로써 컬럼 데이터를 검색, 취득할 수 있습니다,
반환값으로 받은 구조체의 메모리는 시스템이 내부에서 관리하고 있으므로, 변경, 해제 등을 하지 마십시오.

사용 예:

```
HIVE_CONTINUOUS_READER *creader;
creader = hive_inport_continuous_reader(comp, INPORT1, hive_timestamp());

while (hive_component_runnable(comp)) {
  const HIVE_COLUMN_READ_RESULT *read_result = hive_continuous_reader_read(creader);
  if (!read_result) { continue; }

  HIVE_RECORD_ITERATOR iter = HIVE_GET_RECORD_ITERATOR(read_result);
  const HIVE_RECORD *record;
  while ((record = HIVE_RECORD_ITERATOR_GET_NEXT(&iter)) != NULL)
  {
    for (int i=0; i < record->data_count; i++) {
      const HIVE_RECORD_DATA *rd = HIVE_RECORD_GET_DATA(record, i);
      if (!rd) { continue; }

      const char *sourcename   = read_result->incolumn_list->incolumns[i].source_component;
      const char *dataname     = read_result->incolumn_list->incolumns[i].data_name;
      HIVE_DATA_TYPE type      = read_result->incolumn_list->incolumns[i].data_type;
      hive_aggregation_stat_type_t stat = read_result->incolumn_list->incolumns[i].stat_type;
      const char *dptr         = rd->data;
      size_t size              = rd->data_size;

      process_record_data(sourcename, dataanme, type, stat_type, dptr, size);
    }
  }
}
hive_continuous_reader_release(creader);
```

15.4.3.3 hive_continuous_reader_read() ... 컬럼 데이터 계속 검색

Reader로 지정된 입력 포트에 접속되어 있는 컬럼의 데이터를 계속하여 검색, 취득한다

정의

```
const HIVE_COLUMN_READ_RESULT *hive_continuous_reader_read(HIVE_CONTINUOUS_READER *contReader);
```

인수

No	항목	설명
1	`HIVE_CONTINUOUS_READER *contReader`	hive_inport_continuous_reader()로 취득한 Reader

반환값

- HIVE_COLUMN_READ_RESULT: 컬럼 데이터 검색 결과

설명

본 함수는 hive_inport_continuous_reader()로 취득한 Reader로부터 컬럼 데이터를 취득합니다.
호출을 수행할 때마다 검색 시간이 갱신되어, 계속적으로 컬럼 데이터를 취득할 수 있습니다.
반환값으로 받은 구조체의 메모리는 시스템이 내부에서 관리하고 있으므로, 변경, 해제 등을 하지 마십시오.

15.4.3.4 hive_continuous_reader_release() ... 컬럼 데이터 계속 검색용 Reader의 해제

hive_inport_continuous_reader()로 취득한 Reader를 해제한다

정의

```
const void hive_continuous_reader_release(HIVE_CONTINUOUS_READER *contReader);
```

인수

No	항목	설명
1	`HIVE_CONTINUOUS_READER *contReader`	hive_inport_continuous_reader()로 취득한 Reader

반환값

없음

설명

본 함수는 hive_inport_continuous_reader()로 취득한 Reader를 해제합니다.
컴포넌트의 처리 중에 Reader가 불필요해지는 등의 경우에 호출하십시오.

15.4.3.5 hive_inport_timerange_reader() ... 컬럼 데이터 시간 범위 검색용 Reader 취득

입력 포트의 컬럼에서 시간 범위를 지정하여 데이터를 취득하기 위한 Reader를 취득한다

정의

```
HIVE_TIMERANGE_READER *hive_inport_timerange_reader(HIVE_COMPONENT component, const HIVE_INCOLUMN
*incolumn, hive_time_t searchBegin, hive_time_t searchEnd);
```

AI 번역본
For Reference Only

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.

No	항목	설명
2	`HIVE_INCOLUMN *incolumn`	데이터를 취득할 컬럼
3	`hive_time_t search_begin`	검색 범위 개시 시각
3	`hive_time_t search_end`	검색 범위 종료 시각

반환값

- HIVE_TIMERANGE_READER: 컬럼 데이터 시간 범위 검색용 Reader

설명

본 함수는 지정한 컬럼의 데이터를 시간 범위를 지정하여 검색하기 위한 Reader를 취득합니다.
컬럼을 지정하지 않는 경우에는 입력 포트에 접속된 다른 컴포넌트의 컬럼 모두를 검색 대상으로 합니다. 취득한 Reader를 hive_timerange_reader_read()에서 사용함으로써 컬럼 데이터를 검색, 취득할 수 있습니다.
반환값으로 받은 구조체의 메모리는 시스템이 내부에서 관리하고 있으므로, 변경, 해제 등을 하지 마십시오.

사용 예:

AI 번역본
-For Reference Only-

```
hive_time_t search_bigin_ts = hive_timestamp() - 2*HIVE_TIME_1S;

while (hive_component_runnable(comp)) {
  HIVE_TIMERANGE_READER *reader = hive_inport_timerange_reader(comp, incolumn, search_bigin_ts,
search_bigin_ts + 1*HIVE_TIME_1S);

  const HIVE_COLUMN_READ_RESULT *read_result;
  while ((read_result = hive_timerange_reader_read(reader)) != NULL) {
    if (!hive_component_runnable(comp)) { break; }

    HIVE_LOG_INFO("time range: %lu ~ %lu", begin, end);

    HIVE_RECORD_ITERATOR iter = HIVE_GET_RECORD_ITERATOR(read_result);
    const HIVE_RECORD *record;
    while ((record = HIVE_RECORD_ITERATOR_GET_NEXT(&iter)) != NULL)
    {
      Logbuff logb;
      logbuff_reset(&logb);
      logbuff_write(&logb, " %c %19lu", (record->record_type==0)?'D':'S', record->timestamp);

      for (int i=0; i < record->data_count; i++) {
        const HIVE_RECORD_DATA *rd = HIVE_RECORD_GET_DATA(record, i);
        if (!rd) {
          logbuff_write(&logb, "%10s", "(no data)");
          continue;
        }
        const char *sourcename   = read_result->incolumn_list->incolumns[i].source_component;
        const char *dataname     = read_result->incolumn_list->incolumns[i].data_name;
        HIVE_DATA_TYPE type      = read_result->incolumn_list->incolumns[i].data_type;
        hive_aggregation_stat_type_t stat = read_result->incolumn_list->incolumns[i].stat_type;
        const char *dptr         = rd->data;
        size_t size              = rd->data_size;

        process_record_data(sourcename, dataanme, type, stat_type, dptr, size);
      }
    }
  }
  hive_timerange_reader_release(reader);
  usleep(1*1000*1000);
  search_next_ts += 1*HIVE_TIME_1S;
}
```

AI 번역본
-For Reference Only-

15.4.3.6 hive_timerange_reader_read() ... 컬럼 데이터 시간 범검색

Reader로 지정된 컬럼, 또는 입력 포트에 접속되어 있는 전 컬럼의 데이터를 지정된 시간 범위에서 검색, 취득한다

정의

```
const HIVE_COLUMN_READ_RESULT *hive_timerange_reader_read(HIVE_TIMERANGE_READER *rangeReader);
```

인수

No	항목	설명
1	`HIVE_TIMERANGE_READER *rangeReader`	hive_inport_timerange_reader()로 취득한 Reader

반환값

- HIVE_COLUMN_READ_RESULT: 컬럼 데이터 검색 결과

설명

본 함수는 hive_inport_timerange_reader()로 취득한 Reader로부터 컬럼 데이터를 취득합니다.
Reader 취득 시에 지정한 시간 범위의 컬럼 데이터를 취득할 수 있습니다.
반환값으로 받은 구조체의 메모리는 시스템이 내부에서 관리하고 있으므로, 변경, 해제 등을 하지 마십시오.

15.4.3.7 hive_timerange_reader_release() ... 컬럼 데이터 시간 범위 검색용 Reader의 해제

hive_inport_timerange_reader()로 취득한 Reader를 해제한다

정의

```
const void hive_timerange_reader_release(HIVE_TIMERANGE_READER *reader);
```

인수

No	항목	설명
1	`HIVE_TIMERANGE_READER *reader`	hive_inport_timerange_reader()로 취득한 Reader

반환값

없음

설명

AI 번역본
-For Reference Only-

본 함수는 hive_inport_timerange_reader()로 취득한 Reader를 해제합니다.
컴포넌트의 처리 중에 Reader가 불필요해지는 등의 경우에 호출하십시오.

15.4.3.8 HIVE_GET_RECORD_ITERATOR() ... 레코드 이터레이터 취득

윈도우 데이터에서 레코드의 이터레이터를 취득

정의

```
HIVE_RECORD_ITERATOR HIVE_GET_RECORD_ITERATOR(const HIVE_COLUMN_READ_RESULT *data);
```

인수

No	항목	설명
1	`const HIVE_COLUMN_READ_RESULT *data`	hive_continuous_reader_read()로 취득한 윈도우 데이터

반환값

- HIVE_RECORD_ITERATOR: 레코드 이터레이터

설명

본 함수는 hive_continuous_reader_read()로 취득한 윈도우 데이터가 갖는 레코드의 이터레이터를 취득합니다.

취득한 이터레이터에 대해 HIVE_RECORD_ITERATOR_GET_NEXT()를 호출함으로써, 윈도우에 포함된 레코드를 순차적으로 취득할 수 있습니다.

사용 예:

```
while (hive_component_runnable(component)) {
  const HIVE_RECORD *record;
  const HIVE_COLUMN_READ_RESULT *read_result;

  read_result = hive_continuous_reader_read(creader);
  if (!read_result) { continue; }

  HIVE_RECORD_ITERATOR iter = HIVE_GET_RECORD_ITERATOR(read_result);
  while ((record = HIVE_RECORD_ITERATOR_GET_NEXT(&iter)) != NULL)
  {
    for (int i=0; i < record->data_count; i++) {
      const HIVE_RECORD_DATA *rd = HIVE_RECORD_GET_DATA(record, i);
    }
  }
}
```

AI 번역본
-For Reference Only-

15.4.3.9 HIVE_RECORD_ITERATOR_GET_NEXT() ... 레코드 이터레이터 취득

레코드 취득

정의

```
const HIVE_RECORD *HIVE_RECORD_ITERATOR_GET_NEXT(HIVE_RECORD_ITERATOR *iter);
```

인수

No	항목	설명
1	`HIVE_RECORD_ITERATOR *iter`	HIVE_GET_RECORD_ITERATOR()로 취득한 이터레이터로의 포인터

반환값

- `HIVE_RECORD*` : 레코드 포인터
- NULL : 모든 레코드를 취득 완료

설명

본 함수는 윈도우 데이터에 포함된 레코드를 순차적으로 취득하기 위해, HIVE_GET_RECORD_ITERATOR()와 조합하여 사용합니다.
사용 예는 HIVE_GET_RECORD_ITERATOR()를 참고하십시오.

15.4.3.10 HIVE_RECORD_GET_DATA() ... 레코드 이터레이터 취득

레코드 데이터 취득

정의

```
const HIVE_RECORD_DATA *HIVE_RECORD_GET_DATA(const HIVE_RECORD *record, size_t index);
```

인수

No	항목	설명
1	`HIVE_RECORD *record`	HIVE_RECORD_ITERATOR_GET_NEXT()로 취득한 레코드로의 포인터
2	`size_t index`	취득할 레코드 데이터의 인덱스. record->data_count 미만의 값으로 할 것

-For Reference Only-

반환값

- `HIVE_RECORD_DATA*` : 레코드 데이터 포인터

설명

본 함수는 레코드에 포함된 개개의 데이터를 취득합니다.
사용 예는 HIVE_GET_RECORD_ITERATOR()를 참고하십시오.

15.4.3.11 hive_incolumn_get_latest_value() ... 컬럼 데이터 최신값의 취득

입력 포트의 컬럼에서 최신 데이터를 취득한다

정의

```
bool hive_incolumn_get_latest_value(const HIVE_INCOLUMN *incolumn, hive_time_t *rts, char
**dataPointer, size_t *dataSize);
```

인수

No	항목	설명
1	`HIVE_INCOLUMN *incolumn`	데이터를 취득할 컬럼
2	`hive_time_t *rst`	취득한 데이터의 타임스탬프를 넣는 포인터
3	`char **dataPointer`	취득한 데이터의 선두 주소를 저장하는 포인터
4	`size_t *dataSize`	취득한 데이터의 사이즈를 저장하는 포인터

반환값

- true: 최신값 취득 성공
- false: 최신값 취득 실패

설명

본 함수는 지정한 컬럼의 최신 데이터를 취득합니다.
취득에 실패한 경우에는 반환값은 false로, 최신 데이터가 없는 등 취득이 불가능한 경우에는 취득 데이터의 타임스탬프는 0, 데이터 사이즈도 0이 됩니다. 읽어들인 데이터 영역은 HIVE_INCOLUMN 내에 유지하고 있습니다. 그로 인해 복수 스레드에서 동일한 HIVE_INCOLUMN에 대해 이 함수를 호출하는 경우, 배타 제어가 필요합니다.

AI 번역본
-For Reference Only-

15.4.4 컴포넌트 스테이터스 함수

15.4.4.1 hive_status_clear() ... 스테이터스 정보 클리어

스테이터스 정보를 클리어하여, 에러가 없는 상태로 한다

정의

```
void hive_status_clear(HIVE_STATUS status);
```

인수

No	항목	설명
1	`HIVE_STATUS status`	스테이터스 구조체로의 포인터

반환값

없음

설명

스테이터스 구조체에 등록된 에러 정보 등을 소거하여, 아무 에러도 없는 정상 상태로 되돌립니다.
이 상태의 스테이터스 구조체가 hive_component_register_status()로 등록되면, 본 컴포넌트는 정상적으로 동작하고 있는 것으로 판단됩니다.

15.4.4.2 hive_status_add_error() ... 스테이터스 정보로의 에러 등록

스테이터스 정보에 에러를 등록한다

정의

```
void hive_status_add_error(HIVE_STATUS status, HIVE_ERROR_INFO error);
```

인수

No	항목	설명
1	`HIVE_STATUS status`	스테이터스 구조체로의 포인터
2	`HIVE_ERROR_INFO error`	등록할 에러의 번호 사전에 정의된 에러 번호를 지정할 수 있습니다

AI 번역본
-For Reference Only-

반환값

없음

설명

스테이터스 구조체에 에러를 등록합니다. error에는 HIVE_ERROR() 매크로 함수로 생성한 값을 지정하십시오.

사용 예:

```
const static HIVE_ERROR_TYPE error_types[] = {
  { .name="CONNECTION REFUSED",       .parameter_names={"b","c"}},
  { .name="REGISTER READ ERROR",      .parameter_names={"f","g"}},
  { .name=NULL },
};
const int ERROR_CONNECTION_REFUSED       = 0;
const int ERROR_REGISTER_READ_ERROR      = 1;

int proc(COMPONENT *comp, HIVE_STATUS status) {
         :
  hvie_status_clear(status);
  if (detect_error1())
    hive_status_add_error(status, HIVE_ERROR(ERROR_CONNECTION_REFUSED, STR("hostname:port")));

  if (detect_error2())
    hive_status_add_error(status, HIVE_ERROR(ERROR_REGISTER_READ_ERROR, STR("REGDATA_A"),
STR("DT0001")));

         :
  hive_component_register_status(comp, status);
}
```

15.4.4.3 hive_status_set_total_xxx_xxx() ... 스테이터스 정보로의 실행 시 보조 정보 등록

스테이터스 정보에 실행 시 보조 정보를 등록한다

정의

```
void hive_status_set_total_emission_count(HIVE_STATUS status, uint64_t value);
void hive_status_set_total_emission_bytes(HIVE_STATUS status, uint64_t value);
void hive_status_set_total_received_count(HIVE_STATUS status, uint64_t value);
void hive_status_set_total_received_bytes(HIVE_STATUS status, uint64_t value);
```

인수

No	항목	설명
1	`HIVE_STATUS status`	스테이터스 구조체로의 포인터
2	`uint64_t value`	등록값

반환값

없음

설명

스테이터스 구조체에 컴포넌트 실행 시 보조 정보를 등록합니다. 현시점에서 정의되어 있는 보조 정보는 아래와 같습니다.

- 누계 송신 수(hive_status_set_total_emission_count)
  컴포넌트가 외부에 송신한 데이터의 누계 수
- 누계 송신 바이트 수(hive_status_set_total_emission_bytes)
  컴포넌트가 외부에 송신한 데이터의 누계 바이트 수
- 누계 수신 수(hive_status_set_total_received_count)
  컴포넌트가 외부에서 수신한 데이터의 누계 수
- 누계 수신 바이트 수(hive_status_set_total_received_bytes)
  컴포넌트가 외부에서 수신한 데이터의 누계 바이트 수

이 정보는 UI에서 보조적으로 표시할 목적으로 사용됩니다. 따라서 API를 호출하지 않아도 컴포넌트의 동작에 영향은 없습니다.

15.4.4.4 hive_component_register_status() ... 컴포넌트로의 스테이터스 등록

컴포넌트에 스테이터스 정보를 등록한다

정의

```
void hive_component_register_status(HIVE_COMPONENT component, const HIVE_STATUS status);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`HIVE_STATUS status`	스테이터스 구조체로의 포인터

반환값

없음

설명

AI 번역본
-For Reference Only-

컴포넌트에 스테이터스 정보를 등록합니다.
이때의 스테이터스 정보에 에러가 아무것도 등록되어 있지 않은 경우, 컴포넌트는 정상적으로 가동하고 있다고 시스템이 판단합니다.
스테이터스 정보에 에러가 등록되어 있으면, 컴포넌트에 에러가 있음이 시스템에 보고됩니다.

시스템은 마지막으로 등록된 스테이터스 정보로부터 컴포넌트의 상태를 판정합니다. 컴포넌트는 에러가 해소된 경우에는 에러가 클리어된 스테이터스 정보를 이 함수로 재등록해야 합니다.

15.4.4.5 HIVE_ERROR() ... 에러 정보의 생성

에러 정보 구조체를 생성하는 매크로 함수

정의

```
HIVE_ERROR_INFO HIVE_ERROR(int error_no, ...);
```

인수

No	항목	설명
1	`int error_no`	컴포넌트가 정의하고 있는 에러 종별 정보의 인덱스를 지정하십시오..
2	...	발생한 에러의 부속 정보를 지정하십시오. 정수, 부동소수점수, 문자열 중 하나를 지정할 수 있지만, 아래 설명대로 이용 방법이 특수합니다.

반환값

- 에러 정보 구조체 HIVE_ERROR_INFO

설명

발생한 에러의 정보를 갖는 구조체를 생성합니다. (힙 영역이 아닌 스택 영역에 확보하므로 명시적으로 해제할 필요는 없습니다)

제1 인수에는 컴포넌트에서 정의하고 있는 에러 종별 정보의 인덱스를 지정합니다. 이 정보로부터 에러 종별이 판별됩니다. 발생한 에러의 상세를 시스템에 알리기 위해 부속 정보를 인수로 지정할 수 있습니다. 그 경우 부속 정보의 형에 따라 다음과 같이 매크로를 이용하여 인수를 지정하십시오.

```
HIVE_ERROR_INFO error = HIVE_ERROR(ERROR_CONNECTION_REFUSED, STR(hostname), I64(portno));
```

AI 번역본
-For Reference Only-

부속 정보에는 형에 따라 다음 매크로를 지정할 수 있습니다.

- `I64(x)` : 64bit 부호 있는 정수
- `U64(x)` : 64bit 부호 없는 정수
- `DBL(x)` : 배정밀도 부동소수점수
- `STR(x)` : 문자열 포인터

부속 정보에 문자열을 사용하는 경우는, 생성한 HIVE_ERROR_INFO를 hive_component_register_status()로 등록할 때까지는 메모리를 해제하지 않도록 하십시오.

15.4.4.6 hive_status_alloc() ... 스테이터스 정보 구조체의 확보

스테이터스 정보 구조체를 확보한다

정의

```
HIVE_STATUS hive_status_alloc(HIVE_COMPONENT component);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.

반환값

- 스테이터스 정보 구조체 HIVE_STATUS의 포인터
- NULL: 메모리 부족

설명

스테이터스 정보를 기록하는 구조체를 확보합니다.
통상적인 컴포넌트 개발에서는 이 함수를 이용할 필요가 없습니다. 컴포넌트 개시 전에 자동으로 스테이터스 정보 HIVE_STATUS가 확보되어, mainloop의 인수로 전달됩니다.

15.4.4.7 hive_status_free() … 스테이터스 정보 구조체의 해제

스테이터스 정보 구조체를 해제한다

정의

```
void hive_status_free(HIVE_STATUS status);
```

AI 번역본

인수

No	항목	설명
1	`HIVE_STATUS status`	해제할 스테이터스 정보 구조체의 포인터

반환값

없음

설명

hive_status_alloc()로 확보한 스테이터스 정보 구조체를 해제합니다.
통상적인 컴포넌트 개발에서는 이 함수를 이용할 필요가 없습니다. 컴포넌트 개시 전에 자동으로 확보된 스테이터스 정보 HIVE_STATUS는 컴포넌트 정지 시에 자동으로 해제됩니다.

15.4.5 로그 관련 함수

15.4.5.1 HIVE_LOG_XXXX() ... 로그 출력 매크로

컴포넌트용 로그 출력 매크로 함수

정의

```
void HIVE_LOG_ERROR(const char *format, ...);
void HIVE_LOG_WARNING(const char *format, ...);
void HIVE_LOG_INFO(const char *format, ...);
void HIVE_LOG_DEBUG(const char *format, ...);
void HIVE_LOG_TRACE(const char *format, ...);
```

인수

No	인수	설명
1	`const char *format`	로그 출력 서식 지정 문자열 로그에 출력할 서식 지정 문자열을 지정합니다. 표준 C 라이브러리의 printf와 동일한 형식입니다.
2	-	서식 지정 문자열의 변환 지시자에 의해 변환되는 파라미터

반환값

없음

설명

SpeeDBee Synapse 본체의 로그 출력에 문자열을 출력하기 위한 매크로 함수입니다.
인수 format, 및, 그에 이어지는 가변 길이 인수는, 표준 C 라이브러리의 printf()와 동일하게 사용할 수 있습니다.
매크로는 5종류가 준비되어 있으며, 각각 로그 레벨에 대응하고 있습니다.

AI 번역본
-For Reference Only-

15.4.5.2 hive_current_loglevel() ... 로그 레벨 취득

시스템의 현재 로그 레벨을 취득한다

정의

```
HIVE_FW_API HIVE_LOG_LEVEL hive_get_loglevel();
```

인수 없음

반환값

- HIVE_LOG_LEVEL_ERROR(0): 에러 로그만을 로그 출력하는 상태
- HIVE_LOG_LEVEL_WARNING(1): 에러 및 경고 로그를 로그 출력하는 상태
- HIVE_LOG_LEVEL_INFO(2): 통상 레벨의 로그를 출력하는 상태
- HIVE_LOG_LEVEL_DEBUG(3): 디버그용 로그를 출력하는 상태
- HIVE_LOG_LEVEL_TRACE(4): 상세한 트레이스 로그를 출력하는 상태

설명

시스템의 현재 로그 레벨을 취득합니다. ERROR가 최저값, TRACE가 최고값으로 되어 있으며, 반환값의 값보다 작은 값의 로그는 로그 파일에 출력됩니다.

15.4.6 그 외의 함수

15.4.6.1 hive_component_interval_call() ... 정기 처리 함수

일정한 주기로 지정한 함수를 계속적으로 실행하는 함수

정의

```
bool hive_component_interval_call(HIVE_COMPONENT component, hive_interval_proc_t proc, void
*data, hive_time_t base_time, uint64_t interval_usec);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`hive_interval_proc_t proc`	정기 처리 함수 정기적으로 실행할 처리를 정의한 함수의 포인터를 지정합니다.
3	`void *data`	사용자 데이터 임의의 데이터 포인터를 지정할 수 있습니다. proc가 콜될 때 인수로 전달됩니다.
4	`hive_time_t base_time`	정기 처리 함수의 실행 개시 시각 proc 함수를 실행 개시하는 기준 시각 0을 지정하면 현재 시각이 된다

AI 번역본 -For Reference Only-

No	항목	설명
5	`uint64_t interval_usec`	정기 처리 함수 실행의 주기 proc 함수를 호출하는 주기를 usec 단위로 지정합니다.

반환값

- true: proc 함수가 한 번도 에러가 되지 않고, 정상적으로 컴포넌트가 정지한 경우
- false: proc 함수 내에서 에러가 발생한 경우

설명

일정한 주기로 어떠한 처리를 실행하는 경우에, 이 함수를 사용하십시오.

15.4.6.2 hive_component_save_running_info() ... 실행 시 정보를 저장

컴포넌트의 실행 시 정보를 저장하는 함수

정의

```
bool hive_component_save_running_info(HIVE_COMPONENT component, const char *key, const void
*data, size_t size);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`const char *key`	저장할 데이터를 특정하기 위한 정보 서로 다른 key를 지정함으로써 각각 별도의 정보를 유지할 수 있습니다.
3	`const void *data`	저장할 데이터
4	`size_t size`	저장할 데이터의 사이즈

-For Reference Only-

반환값

- true: 저장 성공
- false: 저장 실패

설명

컴포넌트 실행 시의 임의의 정보를 저장합니다.
에미터 등에서 한 번 정지한 후, 이어서 처리하고 싶은 경우를 위해 데이터를 유지해 둘 수 있습니다.

15.4.6.3 hive_component_load_running_info() ... 저장한 실행 시 정보를 읽어들이다

hive_component_save_running_info()로 저장한 데이터를 읽어들이는 함수

정의

```
const void *hive_component_load_running_info(HIVE_COMPONENT component, const char *key, size_t
*size);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`const char *key`	저장한 데이터를 특정하기 위한 정보
3	`size_t *size`	읽어들인 데이터의 사이즈가 반환된다 출력 인수

반환값

읽어들인 데이터의 선두 주소. 데이터가 존재하지 않는 경우에는 null이 반환됩니다.
이 메모리는 프레임워크 측에서 관리하고 있으므로, 사용자가 해제할 필요는 없습니다. 다만, 같은 API를 재차 호출하면 이전의 메모리 영역이 덮어써지므로, 필요한 경우는 사용자 측에서 복사하여 갖도록 하십십시오.

AI 번역본
For Reference Only

설명

hive_component_save_running_info()로 저장한 데이터를 읽어냅니다. 실행 중인 컴포넌트를 정지한 후, 다시 이어서 처리하고 싶은 경우에 저장한 데이터를 읽어들임으로써, 데이터의 도중부터 처리를 재개할 수 있습니다.

15.4.6.4 hive_component_runnable() ... 컴포넌트 계속 가부

컴포넌트의 계속 가부를 판정한다

정의

```
bool hive_component_runnable(HIVE_COMPONENT component);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.

반환값

- true: 컴포넌트의 메인 처리를 계속 가능
- false: 컴포넌트의 정지 요구가 나와 있음

설명

본 함수는, 컴포넌트의 메인 처리를 계속 가능한지를 판정하기 위해 사용합니다.
컴포넌트의 메인 처리는, 이 함수가 true를 반환하는 동안, 처리를 계속하고, false를 반환한 경우에는 신속하게 종료해야 합니다. 그 때문에 메인 처리 내에서는 정기적으로（최장이라도 5 초 간격 정도）이 함수를 호출하여 반환값을 확인하십시오.

이 함수가 false를 반환하는 상태임에도 불구하고, 컴포넌트가 소정의 시간 정지하지 않는 경우, 컴포넌트의 스레드는 시스템에 의해 강제 정지되는 경우가 있습니다.

15.4.6.5 hive_timestamp() ... 현재 시각 취득

시스템의 현재 시각을 취득한다

정의

```
hive_time_t hive_timestamp();
```

인수

없음

반환값

- 시스템의 현재 시각

AI 번역본
-For Reference Only-

설명

시스템의 현재 시각을 취득합니다.
타임스탬프의 단위에 대해서는 hive_time_t를 참조하십시오.

15.4.6.6 hive_get_api_error() ... 발생한 에러의 정보를 취득

컴포넌트의 실행 시에 발생한 에러의 정보를 취득하는 함수

정의

```
HIVE_API_ERROR hive_get_api_error();
```

인수 없음

반환값

- HIVE_API_ERROR: 에러 정보

설명

본 함수는 컴포넌트의 실행 시에 발생한 에러의 정보를 취득합니다. 에러 정보의 내용은 HIVE_API_ERROR 구조체를 참조하십시오.

사용 예 : 에러 정보의 취득

```
  :
  HIVE_OUTCOLUMN clm  = hive_outport_create_column(comp, OUTPORT1, "clm",
HIVE_DATA_SCALAR(HIVE_TYPE_INT8),   HIVE_COLUMN_OPTION_NONE);
  if (clm == nullptr) {
    HIVE_API_ERROR error = hive_get_api_error();
    HIVE_LOG_ERROR("hive_outport_create_column error(0x%x,%d,%d)", error.code,
error.system_errno, error.db_errno);
  }
   :
```

15.4.6.7 HIVE_DATA_TYPE_NAME() ... 데이터형의 이름(문자열)을 취득

SpeeDBee Synapse의 데이터형의 이름(문자열)을 취득하는 함수

정의

```
const char *HIVE_DATA_TYPE_NAME(hive_primitive_type_t primitive_type);
```

인수

샘플 버전본

No	항목	설명
1	`hive_primitive_type_t primitive_type`	데이터형

반환값

- 전달된 데이터형의 이름이 문자열로서 반환된다

설명

본 함수는 SpeeDBee Synapse의 데이터형(hive_primitive_type_t)의 이름을 문자열로서 취득합니다.

사용 예 : 데이터형의 이름(문자열)을 취득

```
  HIVE_LOG_INFO("HIVE_DATA_TYPE_NAME: %s", HIVE_DATA_TYPE_NAME(HIVE_TYPE_INT16));
  :
```

15.4.6.8 HIVE_AGGREGATION_STAT_TYPE_NAME() ... 통계 종별의 이름(문자열)을 취득

SpeeDBee Synapse의 통계 종별의 이름(문자열)을 취득하는 함수

정의

```
const char *HIVE_AGGREGATION_STAT_TYPE_NAME(hive_aggregation_stat_type_t stat_type);
```

인수

No	항목	설명
1	`hive_aggregation_stat_type_t stat_type`	통계 종별

반환값

- 전달된 통계 종별의 이름이 문자열로서 반환된다

설명

본 함수는 SpeeDBee Synapse의 통계 종별의 이름을 문자열로서 취득합니다. 사용

예 : 통계 종별의 이름(문자열)을 취득

```
  HIVE_LOG_INFO("HIVE_AGGREGATION_STAT_TYPE_NAME: %s",
HIVE_AGGREGATION_STAT_TYPE_NAME(HIVE_AGGREGATION_STAT_TYPE_BASICSUM));
  :
```

AI 번역본
-For Reference Only-

15.4.6.9 HIVE_DATA_TYPE_FROM_NAME() ... 데이터명(문자열)에서 데이터형과 배열 수를 취득

데이터명(문자열)에서 SpeeDBee Synapse의 데이터형과 배열 수를 취득하는 함수

정의

```
HIVE_DATA_TYPE HIVE_DATA_TYPE_FROM_NAME(const char *typeName);
```

인수

No	항목	설명
1	`char *typeName`	데이터명

반환값

- HIVE_DATA_TYPE: SpeeDBee Synapse의 데이터형과 배열 수의 정보

설명

본 함수는 데이터명(문자열)에서 SpeeDBee Synapse의 데이터형과 배열 수를 취득합니다. 사용

예 : 데이터형과 배열 수를 취득

```
HIVE_DATA_TYPE data_type = HIVE_DATA_TYPE_FROM_NAME("int16[3]");
HIVE_LOG_INFO("HIVE_DATA_TYPE: type: %d array: %d", data_type.type, data_type.array);
:
```

15.4.6.10 hive_get_common_variable() ... 컴포넌트 공통 변수 취득

컴포넌트 공통 변수를 취득하는 함수

정의

```
const char *hive_get_common_variable(HIVE_COMPONENT component, const char *varname);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
1	`char *varname`	취득하고 싶은 공통 변수의 변수명

For Reference Only

반환값

지정한 변수명에 설정되어 있는 값 (문자열) 의 선두 주소. 데이터가 존재하지 않는 경우에는 null이 반환됩니다.
이 메모리는 프레임워크 측에서 관리하고 있으므로, 사용자가 해제할 필요는 없습니다. 다만, 같은 API를 재차 호출하면 이전의 메모리 영역이 덮어써지므로, 필요한 경우는 사용자 측에서 복사하여 갖도록 하십시오.

설명

본 함수는 변수명을 지정하여, 설정되어 있는 공통 변수의 값을 취득합니다.

사용 예 : 데이터형과 배열 수를 취득

```
const char *project_dir = hive_get_common_variable(component, "PROJECT_DIR");
HIVE_LOG_INFO("project_dir: %s", project_dir);
:
```

15.4.6.11 hive_lock() ... 컴포넌트 간 배타 제어

컴포넌트 간에서의 배타 제어를 위한 락 획득 함수

정의

```
HIVE_LOCK_RESULT hive_lock(HIVE_COMPONENT component, const char *key, bool interbase, uint64_t
timeout);
HIVE_LOCK_RESULT synapse_lock(HIVE_COMPONENT component, const char *key, bool interbase, uint64_t
timeout);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`const char *key`	배타 제어 대상을 식별하기 위한 임의의 문자열
3	`bool interbase`	다른 컴포넌트 베이스 간에서 배타 제어를 수행하는 경우는 true
4	`uint64_t timeout`	락의 해제를 기다리는 경우의 타임아웃 시간

AI 번역본
-For Reference Only-

반환값

- HIVE_LOCK_OK: 락 획득 성공
- HIVE_LOCK_TIMEOUT: 지정된 타임아웃 시간 내에 락을 획득할 수 없었다
- HIVE_LOCK_CANCELED: 락을 획득하기 전에 컴포넌트의 정지 요구를 받았다

설명

본 함수는, 인수 `key` 로 지정된 문자열로 식별되는 리소스의 배타 제어를 위한 락을 획득합니다. 동일한 key 에서의 락은 1 개의 스레드에서만 획득할 수 있습니다. 획득한 락은, 대응하는 API 함수 hive_unlock()
에서 해제할 수 있습니다. 혹은, 락을 획득한 채로 컴포넌트의 실행이 종료된 경우에, 자동으로 락이 해제됩니다.

사용 예 : 디바이스 파일을 사용하는 로직의 배타 제어

```
HIVE_LOCK_RESULT result = hive_lock(component, "/dev/serial", false, 10000000000);
if (result == HIVE_LOCK_TIMEOUT) {
  HIVE_LOG_ERROR("could not take lock.");
  return false;
} else if (result == HIVE_LOCK_CANCELED) {
  return true;
}

  // 이 구간은 단일 컴포넌트 인스턴스만 실행 가능

hive_unlock(component, "/dev/serial", false);
```

인수 key 는 임의의 문자열을 지정 가능합니다.

인수 interbase 가 true인 경우, 다른 컴포넌트 베이스라도, 같은 key 를 사용하면 배타 제어가 가능해집니다. 이 경우는 Python-API의 HiveComponentBase::lock() 을 사용한 커스텀 컴포넌트와의 배타 제어도 이용 가능합니다. interbase 가 false인 경우, 동일한 컴포넌트 베이스에서만 배타 제어됩니다.

본 함수를 호출했을 때에 이미 외부의 컴포넌트 인스턴스가 락을 획득하고 있는 경우, 함수 내에서 실행이 블록됩니다. 블록하는 최대 시간은, 인수 timeout 에서 나노초 단위로 지정 가능합니다. (실질적인 단위는 실행 환경의 클록 정밀도에 의존합니다)

0 을 지정한 경우, 락을 획득할 수 없으면 바로 함수가 리턴합니다. 타임아웃하지 않고 락을 획득할 수 있을 때까지 기다리는 경우는, 인수 timeout 에 HIVE_LOCK_NO_TIMEOUT 을 지정하십시오.

동일 스레드라도 같은 key 에 대해 이중으로 락할 수는 없습니다. 그 경우는 락 해제 대기 상태가 되므로 타임아웃하거나 정지 요구를 받을 때까지 계속 블록합니다.

15.4.6.12 hive_unlock() ... 컴포넌트 간 배타 제어

컴포넌트 간에서의 배타 제어를 위해 획득한 락을 해제한다

정의

```
HIVE_LOCK_RESULT hive_unlock(HIVE_COMPONENT component, const char *key, bool interbase);
HIVE_LOCK_RESULT synapse_unlock(HIVE_COMPONENT component, const char *key, bool interbase);
```

인수

-For Reference Only-

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 premain/main/postmain의 함수 인수에 전달된 컴포넌트 인스턴스를 그대로 전달하십시오.
2	`const char *key`	배타 제어 대상을 식별하기 위한 임의의 문자열

No	항목	설명
3	`bool interbase`	다른 컴포넌트 베이스 간에서 배타 제어를 수행하는 경우는 true

반환값

- HIVE_LOCK_OK: 언락 성공
- HIVE_LOCK_ERROR: 에러

설명

본 함수는, hive_lock 에서 획득한 락을 해제합니다. 인수 key , interbase 에는, 락 획득 시에 지정한 값과 같은 값을 사용하십시오.

사용 예 : 디바이스 파일을 사용하는 로직의 배타 제어

```
HIVE_LOCK_RESULT result = hive_lock(component, "lock-serialdevice", false, 10000000000);
if (result == HIVE_LOCK_TIMEOUT) {
  HIVE_LOG_ERROR("could not take lock.");
  return false;
} else if (result == HIVE_LOCK_CANCELED) {
  return true;
}

  // 이 구간은 단일 컴포넌트 인스턴스만 실행 가능

hive_unlock(component, "lock-serialdevice", false);
```

15.4.7 형・구조체・함수

15.4.7.1 hive_time_t ... HIVE 시각형

시스템의 시각을 나타내는 정수형

정의

```
typedef uint64_t hive_time_t;
```

AI 번역본
For Reference Only

설명

시각을 나타내는 정수값입니다.

부호 없는 64bit 정수값으로 되어 있으며, UNIX 에포크(1970년1월1일0시0분0초(UTC))를 0으로 했을 때의 경과 나노초수로 표현되어 있습니다.

시각	값
1970년1월1일0시0분0초.0(UTC)	0
1970년1월1일9시0분0초.0(JST)	0
1970년1월1일9시0분1 초.0(JST)	1000000000
1970년1월1일9시1분0 초.0(JST)	60000000000
1970년1월1일10시0분0초.0(JST)	3600000000000
2022년1월31일13시0분0초.0(JST)	1643688000000000000
2022년1월31일13시0분1초.0(JST)	1643688001000000000
2022년1월31일13시1분0초.0(JST)	1643688060000000000
2022년1월31일14시0분0초.0(JST)	1643691600000000000

15.4.7.2 hive_primitive_type_t ... HIVE 컬럼 프리미티브형

컬럼의 프리미티브형의 열거형 정의

정의

AI 번역본
-For Reference Only-

```
typedef enum {
  HIVE_TYPE_NONE = 0,
  HIVE_TYPE_BOOLEAN,
  HIVE_TYPE_INT8,
  HIVE_TYPE_INT16,
  HIVE_TYPE_INT32,
  HIVE_TYPE_INT64,
  HIVE_TYPE_UINT8,
  HIVE_TYPE_UINT16,
  HIVE_TYPE_UINT32,
  HIVE_TYPE_UINT64,
  HIVE_TYPE_FLOAT,
  HIVE_TYPE_DOUBLE,
  HIVE_TYPE_TIMESTAMP,         // unsupported
  HIVE_TYPE_STRING,
  HIVE_TYPE_BINARY,
  HIVE_TYPE_COMPLEX,           // unsupported
  HIVE_TYPE_FILE,
  HIVE_TYPE_JSON,              // unsupported
  HIVE_TYPE_BSON,              // unsupported
  HIVE_TYPE_MESSAGEPACK,       // unsupported
  HIVE_TYPE_COMPONENT_STATUS,  // unsupported

  HIVE_TYPE_USERDEFINED = 100,// unsupported
} hive_primitive_type_t;
```

멤버

No	항목	값	설명
1	`HIVE_TYPE_NONE`	0	올바르지 않은 형식
2	`HIVE_TYPE_BOOLEAN`	1	진위값형
3	`HIVE_TYPE_INT8`	2	8bit 부호 있는 정수
4	`HIVE_TYPE_INT16`	3	16bit 부호 있는 정수
5	`HIVE_TYPE_INT32`	4	32bit 부호 있는 정수
6	`HIVE_TYPE_INT64`	5	64bit 부호 있는 정수
7	`HIVE_TYPE_UINT8`	6	8bit 부호 없는 정수
8	`HIVE_TYPE_UINT16`	7	16bit 부호 없는 정수
9	`HIVE_TYPE_UINT32`	8	32bit 부호 없는 정수

AI 번역본
-For Reference Only-

No	항목	값	설명
10	`HIVE_TYPE_UINT64`	9	64bit 부호 없는 정수
11	`HIVE_TYPE_FLOAT`	10	32bit 부동소수점수
12	`HIVE_TYPE_DOUBLE`	11	64bit 부동소수점수
13	`HIVE_TYPE_STRING`	13	문자열
14	`HIVE_TYPE_BINARY`	14	바이너리 데이터
15	`HIVE_TYPE_FILE`	16	파일 데이터

설명

출력 포트에 컬럼을 작성할 때나, 입력 포트에서 컬럼 정보를 취득할 때에, 그 컬럼의 기본형을 나타냅니다.

15.4.7.3 HIVE_DATA_TYPE ... HIVE 컬럼 데이터형

컬럼의 데이터형을 식별하는 구조체

정의

```
typedef struct _tag_HIVE_DATA_TYPE {
  uint8_t type;
  uint8_t array;
} HIVE_DATA_TYPE;
```

멤버

No	항목	설명
1	`uint8_t type`	프리미티브형
2	`uint8_t array`	배열 요소 수

설명

출력 포트에 컬럼을 작성할 때나, 입력 포트에서 컬럼 정보를 취득할 때에, 그 컬럼의 데이터형을 나타내는 구조체입니다.

컬럼의 데이터형에는, 스칼라 외에 배열도 지정 가능하므로, 배열을 지정하는 경우는 그 요소 수가 array 가 됩니다. 다만, array=0 은 스칼라형인 것을 나타냅니다.

이하의 프리미티브형은 스칼라만 대응하고 있습니다. 배열로는 할 수 없으므로 array 는 항상 0으로 하십시오.

- `HIVE_TYPE_TIMESTAMP`
- `HIVE_TYPE_STRING`
- `HIVE_TYPE_BINARY`
- `HIVE_TYPE_FILE`

15.4.7.4 HIVE_COMPONENT … 컴포넌트 인스턴스

컴포넌트의 인스턴스 정보

정의

```
typedef struct {
  void *user_data;
} *HIVE_COMPONENT;
```

멤버

No	항목	설명
1	`void *user_data`	컴포넌트 측에서 자유롭게 참조・편집할 수 있는 전용 멤버

설명

컴포넌트의 실행 정보를 유지하는 구조체에의 포인터입니다.
상기 정의 외에, 컴포넌트를 제어하는 시스템 측이 개개의 컴포넌트를 참조/식별하기 위한 것도 별도 멤버로서 정의되어 있습니다.

AI 번역본
-For Reference Only-

각종 API 호출 시에 필요하게 되므로 그 인수로서 사용하십시오.

user_data 는 사용자가 임의로 참조・편집할 수 있습니다. 컴포넌트의 콜백 함수를 넘어 유지하고 싶은 데이터가 있는 경우는 이쪽을 사용하십시오.

15.4.7.5 hive_interval_proc_t … 정기 처리 함수 시그니처

API hive_component_interval_call을 사용할 때에 지정하는 함수 인터페이스의 typedef입니다.

정의

```
typedef bool (*hive_interval_proc_t)(HIVE_COMPONENT component, void *data, hive_time_t now, int skipped, HIVE_STATUS status);
```

인수

No	항목	설명
1	`HIVE_COMPONENT component`	컴포넌트의 인스턴스 `hive_component_interval_call` 의 호출 시에 전달한 component 입니다.
2	`void *data`	사용자 데이터 `hive_component_interval_call` 에서 지정한 임의의 포인터입니다.
3	`hive_time_t now`	각 정기 처리 실행 시에, 이 함수가 호출된 시각의 타임스탬프
4	`int skipped`	정기 처리가 제때 이루어지지 못하고 스킵된 경우의 스킵 수
5	`HIVE_STATUS status`	에러 정보 등의 등록에 사용하는 status

반환값

- true: 정기 처리를 이후에도 계속하는 경우
- false: 정기 처리를 정지하는 경우

설명

API hive_component_interval_call을 사용할 때에 지정하는 함수 인터페이스의 typedef입니다.
정기 처리를 수행하는 함수는 이 선언과 일치하도록 정의하십시오.

이 함수의 반환값에 의해, 이후의 정기 처리가 계속될지, 정지할지가 정해집니다. 에러로 이후의 처리를 계속할 수 없는 것 같은 경우 이외는 true를 반환하도록 하십시오.

AI 번역본
-For Reference Only-

15.4.7.6 HIVE_COLUMN_READ_RESULT ... 컬럼 정보 불러오기 결과 (윈도우 데이터)

입력 포트에서 컬럼의 값을 읽어들인 결과를 격납하는 구조체

정의

```
typedef struct hive_HIVE_COLUMN_READ_RESULT {
  hive_time_t timerange_begin;
  hive_time_t timerange_end;
  const HIVE_INCOLUMN_LIST *incolumn_list;
  uint64_t window_id;
  uint64_t event_id;
  const struct hive_HIVE_RECORD *statistics_record;
  bool is_continued;
  const char *buffer;
  size_t buffer_size;
} HIVE_COLUMN_READ_RESULT;
```

멤버

No	항목	설명
1	`hive_time_t timerange_begin`	이 구조체에 포함되는 데이터의 시각 범위의 선두 시각
2	`hive_time_t timerange_end`	이 구조체에 포함되는 데이터의 시각 범위의 종단 시각
3	`const HIVE_INCOLUMN_LIST *incolumn_list`	입력 포트에 접속된 컬럼의 목록
6	`uint64_t window_id`	이 데이터의 윈도우ID
6	`uint64_t event_id`	이 데이터의 이벤트ID
6	`HIVE_RECORD *statistics_record`	집약 레코드
6	`bool is_continued`	윈도우가 다음 READ_RESULT에 계속하는지 여부
4	`const char *buffer`	시스템이 내부에서 사용하는 영역. 액세스하지 않을 것
5	`size_t buffer_size`	시스템이 내부에서 사용하는 영역. 액세스하지 않을 것

AI 번역본
-For Reference Only-

설명

API hive_continuous_reader_read()를 사용했을 때에 반환값으로서 받아들이는 데이터형입니다.
이 1 개의 구조체로 1 윈도우분의 데이터를 유지하고 있습니다. 윈도우 범위는 입력 포트의 설정에 의해 변동합니다.

이 구조체에서 데이터를 취득하는 경우, HIVE_GET_RECORD_ITERATOR()나 HIVE_RECORD_ITERATOR_GET_NEXT()를 사용하십시오.

15.4.7.7 HIVE_RECORD ... 컬럼 데이터 레코드

윈도우 데이터에 포함되는 동일 타임스탬프의 각 컬럼의 데이터를 정리한 구조체

정의

```
typedef struct hive_HIVE_RECORD {
  uint8_t record_type;
  uint8_t unused;
  uint16_t data_count;
  uint32_t data_size;
  uint64_t timestamp;
  uint32_t offset[0];
} HIVE_RECORD;
```

멤버

No	항목	설명
1	`uint8_t record_type`	레코드 종별
2	`uint8_t unused`	미사용 영역
3	`uint16_t data_count`	레코드 내의 데이터 수
4	`uint32_t data_size`	레코드 내의 전 데이터의 바이트 수
5	`uint64_t timestamp`	레코드의 타임스탬프
6	`uint32_t offset[0]`	시스템이 내부에서 사용하는 영역. 액세스하지 않을 것

-For Reference Only-

설명

입력 포트에서 취득한 윈도우 데이터 내의 동일 타임스탬프의 데이터 세트입니다.
HIVE_RECORD_ITERATOR_GET_NEXT()를 사용하여 취득할 수 있습니다.
이 레코드에 포함되는 데이터 수는 data_count 에서 취득할 수 있지만, 윈도우 데이터의 incolumn_list의 컬럼 수와 같아집니다.

레코드 내의 각 데이터에는 HIVE_RECORD_GET_DATA(record, i) 로 액세스할 수 있습니다.

15.4.7.8 HIVE_RECORD_DATA ... 레코드 데이터

레코드에 포함되는 개개의 데이터를 갖는 구조체

정의

```
typedef struct hive_HIVE_RECORD_DATA {
  uint16_t index;
  uint16_t stat_type;
  uint32_t data_size;
  uint64_t timestamp;
  char data[0];
} HIVE_RECORD_DATA;
```

멤버

No	항목	설명
1	`uint16_t index`	미사용 영역
2	`uint16_t stat_type`	본 데이터의 집약 종별
3	`uint32_t data_size`	데이터 사이즈
4	`uint64_t timestamp`	타임스탬프
5	`char data[0]`	데이터

설명

레코드 내의 개개의 데이터를 유지하는 구조체입니다.
멤버 data 는, 0요소의 char 배열로 정의되어 있지만, 실제로는 data_size 분의 메모리 영역이 확보되어 있습니다. 다만,

data_size 가 0인 경우, 이 타임스탬프의 데이터가 없는 것을 나타냅니다. 그 경우도 data 의 포인터는 NULL 이 되지 않는 것에 주의하십시오.

AI 번역본
-For Reference Only-

15.4.7.9 HIVE_COLUMN_OPTIONS ... 컬럼 옵션

컬럼 작성 시의 옵션 정보를 격납하는 구조체

정의

```
typedef struct _tag_HIVE_COLUMN_OPTIONS {
  hive_column_type_t type;
  double sampling_rate;
  int compression_unit;
  int fixed_size_binary;
  int max_binary_size;
} HIVE_COLUMN_OPTIONS;

#define HIVE_COLUMN_OPTION_NONE ((HIVE_COLUMN_OPTIONS){column_type_low,0,0,0,0})
#define HIVE_COLUMN_OPTION_MIDDLE(r) ((HIVE_COLUMN_OPTIONS){column_type_middle,r,0,0,0})
```

멤버

No	항목	설명
1	`hive_column_type_t type`	컬럼 종별 (column_type_low or column_type_middle)
2	`double sampling_rate`	샘플링 레이트
3	`int compression_unit`	압축 단위
4	`int fixed_size_binary`	바이너리형 고정 사이즈
5	`int max_binary_size`	바이너리형 최대 사이즈

설명

hive_outport_create_column()에서 컬럼을 작성할 때에, 제 5 인수에 지정하는 옵션 정보입니다. 필수는 아닙니다.

매크로

매크로 HIVE_COLUMN_OPTION_NONE 은 디폴트의 옵션 정의입니다. 통상은 이것을 사용하십시오.
매크로 HIVE_COLUMN_OPTION_MIDDLE(r) 은 미들 종별의 컬럼을 정의하기 위한 옵션 정보입니다. 미들 컬럼인 경우, 샘플링 레이트의 지정이 필수가 됩니다.

AI 번역본
For Reference Only-

15.4.7.10 HIVE_API_ERROR ... API 에러

API 에러 정보 구조체

정의

```
typedef struct hive_HIVE_API_ERROR {
  uint32_t code;
  int system_errno;
  int db_errno;
} HIVE_API_ERROR;
```

멤버

No	항목	설명
1	`uint32_t code`	에러의 종별을 나타내는 수치
2	`int system_errno`	시스템의 에러 번호
3	`int db_errno`	데이터베이스의 에러 번호

설명

API 에러 발생 시에, 그 에러의 종별이나 상세 정보를 유지하는 구조체입니다.
에러가 발생한 경우, hive_get_api_error() 함수에 의해 이 정보를 취득할 수 있습니다.

code 에는, 발생한 에러를 식별하는 수치가 기록되어 있습니다.

API 에러가 시스템 에러인 경우, 그 에러 번호가 system_errno 에 격납됩니다. 시스템 에러가 아닌 경우는 system_errno 는 0입니다.

데이터베이스 시스템의 에러가 기인이 되고 있었던 경우, 그 에러 번호가 db_errno 에 격납됩니다. 데이터베이스 시스템 기인이 아닌 경우는 db_errno 는 0입니다.

AI 번역본
-For Reference Only-

15.5 커스텀 UI JS-API 레퍼런스

커스텀 UI를 HTML로 작성할 때의 JavaScript의 API 사양입니다. 커스텀 UI를 HTML로 작성하는 순서에 대해서는 아래를 참조하십시오.

커스텀 UI 작성 HTML을 이용한 정의 파일의 작성

15.5.1 콜백 함수

파라미터 설정 다이얼로그가 표시될 때, 및, 설정을 저장하여 닫을 때에, SpeeDBee Synapse의 UI 화면 본체와 파라미터 정보를 주고받기 위한 콜백 함수가 필요하게 됩니다. 아래에 설명하는 2 개의 함수 `onLoad` , `onSave` 는, 커스텀 UI(HTML) 내의 JavaScript로 정의하십시오. SpeeDBee Synapse의 UI 화면 본체에서 호출됩니다.

15.5.1.1 onLoad ... 커스텀 UI 불러오기 훅

커스텀 컴포넌트의 설정 다이얼로그가 사용자에게 표시되었을 때에 호출되는 훅 함수

정의

```
onLoad?: (editable: boolean, parameter: {}, newComponent: boolean, componentId?: string) =>
Promise<void>;
```

인수

- editable: 현재 표시하고 있는 설정 다이얼로그를 편집 가능으로 하는 경우는 true, 편집 불가인 경우는 false
- parameter: 저장 완료된 파라미터 오브젝트
- newComponent: 컴포넌트의 신규 작성 시에 true, 갱신 시는 false
- componentId: 대상 컴포넌트 인스턴스의 ID

AI 번역본
-For Reference Only-

반환값

없음

설명

커스텀 컴포넌트의 설정 다이얼로그가 사용자에게 표시되었을 때에 호출되는 함수를 등록하는 프로퍼티입니다. 저장 완료된 파라미터를 취득하기 위해 필요하므로, 이하와 같이 하여 반드시 등록하십시오.

```
window.customUiLib.onLoad = function (editable, parameter, newComponent, componentId) {
  :
};
```

이 함수의 인수로서, 이전에 저장한 파라미터 오브젝트를 받아들일 수 있습니다. 이 오브젝트를 기초로, HTML 내의 입력 상자 등에 초기값을 설정하십시오.

Run 모드인 경우는 editable 이 false 로 되어 있습니다. 이 경우는 사용자가 설정을 변경할 수 없도록, 입력 항목 등은 비활성으로 하는 등의 대응을 하십시오. 비활성으로 하지 않는 경우 사용자가 화면상에서 값을 변경할 수 있게 됩니다. 다만, 그 경우에도 파라미터의 저장은 할 수 없습니다.

제4 인수의 componentId 는, window.customUiLib.api 오브젝트 내의 API를 사용할 때의 인수로서 사용할 수 있습니다.
다만 컴포넌트의 신규 작성 시에는, 아직 컴포넌트가 작성되어 있지 않으므로 componentId 는 인수로서 전달되지 않습니다.

15.5.1.2 onSave ... 파라미터 저장 훅

커스텀 컴포넌트의 설정 다이얼로그에서, 사용자가 저장 버튼을 눌러 파라미터를 저장할 때에 자동으로 호출되는 훅 함수

정의

```
onSave?: () => Promise<{} | null>;
```

인수

없음

반환값

- null: 파라미터의 저장을 캔슬하는 경우에 null을 반환합니다.
- {}: 저장하고 싶은 파라미터를 반환합니다.

설명

사용자가 저장 버튼을 눌렀을 때에 자동으로 실행되는 함수입니다. 저장 시에 수행하고 싶은 처리를 하나의 함수로서 실장 하고, 아래와 같이 onSave 에 등록하십시오.

```
window.customUiLib.onSave = function () {
  :
  return ..;
};
```

이 함수가 호출되면, 저장할 파라미터 정보를 오브젝트로서 반환할 필요가 있습니다. 오브젝트는 JSON화 가능한 데이터로 하십시오. 반환된 오브젝트가 해당 컴포넌트의 파라미터로서 저장되고, 설정 다이얼로그는 닫힙니다.

AI 자동화
-For Reference Only-

null을 반환한 경우, 파라미터의 저장은 캔슬됩니다. 이 경우는 다이얼로그도 닫히지 않습니다. 커스텀 UI 화면에서 사용자가 입력한 데이터 등의 포맷을 체크하고, 입력 내용에 문제가 있는 경우 등에, 이 방법으로 저장을 캔슬할 수 있습니다.
다만 캔슬된 것은 자동으로는 사용자에게 통지되지 않으므로, 어떤
요인으로 저장을 캔슬했는지에 대해서는 커스텀 UI의 HTML 내에서 에러 표시 등을 수행하도록 하십시
오.

15.5.2 SpeeDBee Synapse 설정 취득 함수

window.customUiLib.api 오브젝트에는, SpeeDBee Synapse의 화면상에서 설정한 다양한 정보를 취득할 수 있는 유틸리티 함수가 정의되어 있습니다. 이들 함수의 사용은 필수는 아니지만, 적절히 이용함으로써 입력 보조 등의 용도로 사용자의 편리성, 조작성을 향상시킬 수 있습니다.

15.5.2.1 getComponentbases ... 컴포넌트 베이스 목록 취득

컴포넌트 베이스 정보의 목록을 취득하는 함수

정의

```
window.customUiLib.api.getComponentbases: () => Promise<ComponentbaseResponse[]>;
```

인수

없음

반환값

- ComponentbaseResponse[]: 컴포넌트 베이스 정보의 목록

설명

본 함수는, 컴포넌트 베이스 정보를 목록으로 취득합니다.

15.5.2.2 getPanels ... 패널 목록 취득

패널 목록을 취득하는 함수

정의

```
window.customUiLib.api.getPanels: () => Promise<PanelResponse[]>;
```

인수

없음

반환값

- PanelResponse[]: 패널 정보의 목록

AI 번역본
For Reference Only-

설명

본 함수는, 전 패널의 정보를 목록으로 취득합니다.

15.5.2.3 getPanel ... 패널 정보 취득

패널 정보를 취득하는 함수

정의

```
window.customUiLib.api.getPanel: (id: number) => Promise<PanelResponse>;
```

인수

No	항목	설명
1	`id` *number*	패널ID

반환값

- PanelResponse: 패널 정보

설명

본 함수는, 지정한 패널ID의 패널 정보를 취득합니다.

15.5.2.4 getComponents ... 컴포넌트 목록 취득

컴포넌트 목록을 취득하는 함수

정의

```
window.customUiLib.api.getComponents: (panelId?: number) => Promise<ComponentResponse[]>;
```

인수

No	항목	설명
1	`panelId` *number | undefined*	취득할 패널의 ID (미지정인 경우는 전 패널)

반환값

- ComponentResponse[]: 컴포넌트 목록

AI 번역본
-For Reference Only-

설명

본 함수는, 컴포넌트 정보를 목록으로 취득합니다.
패널ID를 지정하면, 그 패널에 배치하고 있는 컴포넌트만을 취득할 수 있습니다.

15.5.2.5 getComponent ... 컴포넌트 정보 취득

컴포넌트 정보를 취득하는 함수

정의

```
window.customUiLib.api.getComponent: (id: string) => Promise<ComponentResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- ComponentResponse: 컴포넌트 정보

설명

본 함수는, 지정한 컴포넌트ID의 컴포넌트 정보를 취득합니다.

15.5.2.6 getInPort … 입력 포트 설정 취득

입력 포트 설정을 취득하는 함수

정의

```
window.customUiLib.api.getInPort: (id: string) => Promise<InportResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- InportResponse: 컴포넌트의 입력 포트 설정

AI 번역본
-For Reference Only-

설명

본 함수는, 컴포넌트의 입력 포트 설정을 취득합니다.
입력 포트 설정에는, 보내져 오는 데이터에 대한 설정이나 집약, 지연 시간, 이벤트에 대한 설정이 포함되어 있습니다.

15.5.2.7 getInPortData … 입력 포트 정보 취득

입력 포트에서 수신하고 있는 데이터에 대한 정보를 취득하는 함수

정의

```
window.customUiLib.api.getInPortData: (id: string) => Promise<PortDataResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- PortDataResponse: 컴포넌트의 입력 포트에서 수신하고 있는 데이터의 정보

설명

본 함수는, 컴포넌트의 입력 포트에서 수신하고 있는 데이터에 대한 정보를 취득합니다.
입력 포트 정보에는, 보내져 오는 데이터가 어떤 형인지, 배열인지 여부 등의 정보가 포함되어 있습니다.

15.5.2.8 getInPortDataValue ... 입력 포트 데이터 취득

입력 포트에서 수신하고 있는 데이터를 취득하는 함수

정의

```
window.customUiLib.api.getInPortDataValue: (id: string) => Promise<PortDataValueResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

AI 번역본
-For Reference Only-

반환값

- PortDataValueResponse: 컴포넌트의 입력 포트에서 수신하고 있는 데이터

설명

본 함수는, 컴포넌트의 입력 포트에서 수신하고 있는 데이터나 타임스탬프를 취득합니다. 취득할 수 있는 값은
각 데이터마다 최신 1건이 됩니다.

15.5.2.9 getSystemPort ... 시스템 포트 설정 취득

시스템 포트 설정을 취득하는 함수

정의

```
window.customUiLib.api.getSystemPort: (id: string) => Promise<OutportResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- OutportResponse: 컴포넌트의 시스템 포트 설정

설명

본 함수는, 컴포넌트의 시스템 포트 설정을 취득합니다.
시스템 포트 설정에는, 송신할 데이터의 영속화, 수치 변환, 등록 조건이나 부여한 코멘트의 설정이 포함되어 있습니다.

15.5.2.10 getSystemPortData … 시스템 포트 정보 취득

시스템 포트에서 송신하고 있는 데이터에 대한 정보를 취득하는 함수

정의

```
window.customUiLib.api.getSystemPortData: (id: string) => Promise<PortDataResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

AI 번역본
-For Reference Only-

반환값

- PortDataResponse: 컴포넌트의 시스템 포트에서 송신하고 있는 데이터의 정보

설명

본 함수는, 컴포넌트의 시스템 포트에서 송신하고 있는 데이터에 대해 취득합니다.
시스템 포트 정보에는, 보내는 데이터는 어떤 형인지, 배열인지 여부 등의 정보가 포함되어 있습니다.

15.5.2.11 getSystemPortDataValue … 시스템 포트 데이터 취득

시스템 포트에서 송신하고 있는 최신 데이터를 취득하는 함수

정의

```
window.customUiLib.api.getSystemPortDataValue: (id: string) => Promise<PortDataValueResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- PortDataValueResponse: 컴포넌트의 시스템 포트에서 송신하고 있는 데이터

설명

본 함수는, 컴포넌트의 시스템 포트에서 송신하고 있는 데이터, 타임스탬프를 취득합니다. 취득하는 값은 각 데이터마다 최신 1건이 됩니다.

15.5.2.12 getOutPort ... 출력 포트 설정 취득

출력 포트 설정을 취득하는 함수

정의

```
window.customUiLib.api.getOutPort: (id: string) => Promise<OutportResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

AI 번역본
-For Reference Only-

반환값

- OutportResponse: 컴포넌트의 출력 포트 설정

설명

본 함수는, 컴포넌트의 출력 포트 설정을 취득합니다.
출력 포트 설정에는, 송신할 데이터의 영속화, 수치 변환, 등록 조건이나 부여한 코멘트의 설정이 포함되어 있습니다.

15.5.2.13 getOutPortData ... 출력 포트 정보 취득

출력 포트에서 송신하고 있는 데이터에 대한 정보를 취득하는 함수

정의

```
window.customUiLib.api.getOutPortData: (id: string) => Promise<PortDataResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

반환값

- PortDataResponse: 컴포넌트의 출력 포트에서 송신하고 있는 데이터의 정보

설명

본 함수는, 컴포넌트의 출력 포트에서 송신하고 있는 데이터에 대해 취득합니다.
출력 포트 정보에는, 보내는 데이터가 어떤 형인지, 배열인지 여부 등의 정보가 포함되어 있습니다.

15.5.2.14 getOutPortDataValue ... 출력 포트 데이터 취득

출력 포트에서 송신하고 있는 최신 데이터를 취득하는 함수

정의

```
window.customUiLib.api.getOutPortDataValue: (id: string) => Promise<PortDataValueResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 컴포넌트의 ID

-For Reference Only-

반환값

- PortDataValueResponse: 컴포넌트의 출력 포트에서 송신하고 있는 데이터

설명

본 함수는, 컴포넌트의 출력 포트에서 송신하고 있는 데이터, 타임스탬프를 취득합니다. 취득하는 값은 각 데이터마다 최신 1건이 됩니다.

15.5.2.15 runMethod ... 컴포넌트 인스턴스 메서드 실행

컴포넌트 인스턴스 메서드를 실행하는 함수입니다.

> ⚠ 주의
>
> 메서드를 실행할 수 있는 것은 관리자 이상입니다.

정의

```
runMethod: (id: string, methodName: string, parameter: string | object | null) =>
Promise<ComponentMethodResponse>;
```

인수

No	항목	설명
1	`id` *string*	실행할 컴포넌트의 ID
2	`methodName` *string*	취득할 메서드의 명칭
3	`parameter` *string* | *object* | *null*	실행할 메서드에 전달할 파라미터

반환값

- ComponentMethodResponse: 실행한 인스턴스 메서드의 결과와 반환값

설명

본 함수는, 컴포넌트에 실장되어 있는 인스턴스 메서드를 실행합니다. 인스턴스 메서드에 대해서는 을 참조하십시오.

AI 번역본
-For Reference Only-

15.5.2.16 getFlowlinks ... 플로우 링크 목록 취득

플로우 링크 목록을 취득하는 함수

정의

```
window.customUiLib.api.getFlowlinks: (componentId?: string) => Promise<FlowlinkResponse[]>;
```

인수

No	항목	설명
1	`componentId` *string*	취득할 컴포넌트의 ID (미지정인 경우는 전 플로우 링크)

반환값

- FlowlinkResponse[]: 플로우 링크 목록

설명

본 함수는, 플로우 링크 정보를 취득합니다.
컴포넌트 ID를 지정한 경우는 그 컴포넌트에 접속되어 있는 플로우 링크를, 미지정인 경우는 전 플로우 링크를 취득합니다.

15.5.2.17 getFlowlink ... 플로우 링크 정보 취득

플로우 링크 정보를 취득하는 함수

정의

```
window.customUiLib.api.getFlowlink: (id: string) => Promise<FlowlinkResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 플로우 링크의 ID

반환값

- FlowlinkResponse: 플로우 링크 정보

설명

본 함수는, 플로우 링크 정보를 ID로 검색하여 취득합니다.

AI 번역본
-For Reference Only-

15.5.2.18 getFlowlinkFilter ... 플로우 링크 필터 정보 취득

플로우 링크에 설정되어 있는 필터 정보를 취득하는 함수

정의

```
window.customUiLib.api.getFlowlinkFilter: (id: string) => Promise<FlowlinkFiltersResponse>;
```

인수

No	항목	설명
1	`id` *string*	취득할 플로우 링크의 ID

반환값

- FlowlinkFiltersResponse: 플로우 링크 정보

설명

본 함수는, 플로우 링크에 설정되어 있는 필터 정보를 ID로 검색하여 취득합니다.

15.5.3 형

15.5.3.1 ComponentbaseResponse

SpeeDBee Synapse에 등록되어 있는 컴포넌트 베이스 정보 정의

```
type ComponentbaseResponse = {
    id: string,
    name: string,
    tag: string,
    inports: number,
    outports: number,
    required_point: number,
    error_types: {
        id: string,
        parameter_names: string[],
    }[],
}
```

구성

No	항목	설명
1	`id`	컴포넌트 베이스 ID
2	`name`	컴포넌트 베이스명
3	`tag`	태그
4	`inports`	입력 포트 수
5	`outports`	출력 포트 수

No	항목	설명
6	`required_point`	기동 시에 필요한 포인트 수
7	`error_types`	에러 종별 오브젝트의 목록

> *error_types*
>
> | No | 항목 | 설명 |
> | --- | --- | --- |
> | 1 | `id` | 에러 ID |
> | 2 | `parameter_names` | 파라미터명 |

설명

각 컴포넌트의 베이스 정보의 데이터형입니다.
getComponentbases를 사용했을 때 반환값으로서 목록을 취득할 수 있습니다.

15.5.3.2 PanelResponse

패널 정보

정의

```
type PanelResponse = {
    id: number,
    order: number,
    name: string,
    type: string,
    options: any,
    components: any[],
};
```

구성

-For Reference Only-

No	항목	설명
1	`id`	패널 ID
2	`order`	표시순
3	`name`	명칭

No	항목	설명
4	`type`	종별
5	`options`	옵션
6	`components`	컴포넌트 정보

설명

등록되어 있는 각 패널의 데이터형입니다.
getPanels, getPanel을 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.3 ComponentResponse

컴포넌트 정보

정의

```
type ComponentResponse = {
    id: string,
    baseid: string,
    name: string,
    enable: boolean,
    parameter: any | undefined,
    options: any | undefined,
};
```

구성

No	항목	설명
1	`id`	컴포넌트 ID
2	`baseid`	컴포넌트 베이스 ID
3	`name`	명칭
4	`enable`	유효/무효
5	`parameter`	파라미터
6	`options`	옵션

AI 번역본
-For Reference Only-

설명

사용자가 설정한 각 컴포넌트를 취득할 때의 데이터형입니다.
getComponents, getComponent를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.4 InportResponse

입력 포트 설정

정의

```
type InportResponse = {
    setting_option: 'reception' | 'regexp',
    column_data: {
        target: {
            source_name: string,
            data_name: string,
            data_types?: string[],
        }[],
        raw_data: boolean,
        typical_value?: string,
        basic_statistics: string[],
    }[],
    window: {
        window_type: 'tumbling' | 'value_change',
        time_span?: number,
        columns?: {
            source_name: string,
            data_name: string,
            data_types: string[]
        },
        max_duration?: number
    },
    delaytime: number,
    event?: {
        source_name: string,
        data_name: string,
        pre_event_time: number,
        post_event_time: number
    },
    column_order_output?: {
        specify_columns: {
            source_name: string,
            data_name: string,
            outputname?: string,
        }[],
        order_type: string
    }
};
```

AI 번역본
For Reference Only-

구성

No	항목	설명
1	`setting_option`	필터를 정규 표현식으로 수행할지, 데이터마다 수행할지 여부
2	`column_data`	입력 데이터 정보
3	`window`	윈도우 설정
4	`delaytime`	지연 시간(나노초)
5	`event`	입력 포트로의 데이터를 외부 이벤트에 의해 ON/OFF하는 설정
6	`column_order_output`	입력 포트에 들어온 데이터를 정렬하는 설정과 별칭 설정

AI 번역본
-For Reference Only-

column_data

No	항목	설명
1	`target`	집약 대상이 되는 소스와 데이터의 패턴
2	`raw_data`	미가공 데이터의 이용 유무
3	`typical_value`	대표값의 설정. 최신값(newer) 또는 가장 오래된 값(older)
4	`basic_statistics`	기본 통계 데이터

target

No	항목	설명
1	`source_name`	대상으로 하는 컴포넌트명의 패턴을 나타내는 정규 표현식
2	`data_name`	대상으로 하는 데이터명의 패턴을 나타내는 정규 표현식
3	`data_types`	대상으로 하는 데이터형의 리스트(생략/null은 임의의 형에 매칭)

window

No	항목	설명
1	`window_type`	윈도우 종별
2	`time_span`	검색 시간 폭(나노초)
3	`columns`	대상으로 하는 데이터 정보의 목록
4	`max_duration`	윈도우의 최대 시간 폭(나노초)

AI 번역본
-For Reference Only-

columns

No	항목	설명
1	`source_name`	대상으로 하는 컴포넌트명의 패턴을 나타내는 정규 표현식
2	`data_name`	대상으로 하는 데이터명의 패턴을 나타내는 정규 표현식
3	`data_types`	대상으로 하는 데이터형의 리스트(생략/null은 임의의 형에 매칭)

event

No	항목	설명
1	`source_name`	외부 이벤트의 컬럼을 생성하는 컴포넌트명
2	`data_name`	외부 이벤트의 컬럼명(bool형만 유효)
3	`pre_event_time`	프리 이벤트 시간(나노초)
4	`post_event_time`	포스트 이벤트 시간(나노초)

column_order_output

No	항목	설명
1	`specify_columns`	정렬 순서와 출력명
2	`order_type`	지정 외 데이터의 정렬 순서

AI 번역본
For Reference Only

specify_columns

No	항목	설명
1	`source_name`	컴포넌트명
2	`data_name`	데이터명
3	`outputname`	컬럼에 부여할 출력명

설명

컴포넌트의 입력 포트 설정의 데이터형입니다.
getInPort를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.5 OutportResponse

시스템, 출력 포트 설정

정의

```
type OutportResponse = {
    setting_option: 'reception' | 'regexp';
    pattern_settings: {
        target: {
            source_name?: string,
            data_name: string,
            data_types?: string[],
        }[],
        storage_type: string,
        file_index: boolean,
        conversion?: {
            data_type?: string,
            expression?: string,
        },
        value_filter?: {
            expression: string,
        },
        comment?: string,
    }[];
};
```

구성

No	항목	설명
1	`target`	대상 데이터를 지정하기 위한 패턴 정의
2	`storage_type`	영속화
3	`file_index`	검색용 인덱스의 작성 플래그
4	`conversion`	수치 변환
5	`value_filter`	등록 조건
6	`comment`	코멘트

AI 번역본
-For Reference Only-

target

No	항목	설명
1	`source_name`	대상으로 하는 컴포넌트명의 패턴을 나타내는 정규 표현식
2	`data_name`	대상으로 하는 데이터명의 패턴을 나타내는 정규 표현식
3	`data_types`	대상으로 하는 데이터형의 리스트(생략/null은 임의의 형에 매칭)

conversion

No	항목	설명
1	`data_type`	변환 후의 데이터형
2	`expression`	변환식

value_filter

No	항목	설명
1	`expression`	조건식

컴포넌트의 출력 포트 설정의 데이터형입니다.
getSystemPort, getOutPort를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.6 PortDataResponse

입출력 포트가 수신, 송신하고 있는 데이터에 대한 정보

정의

AI 번역본
-For Reference Only-

```
type PortDataResponse = {
    id: string;
    name: string;
    data?: {
        name: string;
        source?: string;
        column_type: string,
        sampling_rate?: number,
        data_type: string;
        data_array?: number;
    }[]
};
```

구성

No	항목	설명
1	`id`	컴포넌트 ID
2	`name`	명칭
3	`data`	입력 포트에 수신하고 있는 데이터 정보

> *data*
>
> | No | 항목 | 설명 |
> |---|---|---|
> | 1 | `name` | 데이터명 |
> | 2 | `source` | 컴포넌트명 |
> | 3 | `column_type` | 컬럼 종별 |
> | 4 | `sampling_rate` | 샘플링 레이트 |
> | 5 | `data_type` | 데이터형 |
> | 6 | `data_array` | 배열 수 |

AI 번역본
-For Reference Only-

설명

컴포넌트의 입출력 포트가 수신, 송신하고 있는 데이터에 대한 데이터형입니다.
getInPortData, getSystemPortData, getOutPortData를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.7 PortDataValueResponse

입출력 포트가 수신, 송신하고 있는 데이터

정의

```
type PortDataValueResponse = {
    id: string;
    name: string;
    data?: {
        value: any | any[];
        timestamp?: number;
        name: string;
        source?: string;
        data_type: string;
        data_array?: number;
    }[]
};
```

구성

No	항목	설명
1	`id`	컴포넌트 ID
2	`name`	명칭
3	`data`	입력 포트에 수신하고 있는 데이터

> *data*
>
> | No | 항목 | 설명 |
> | --- | --- | --- |
> | 3 | `value` | 값 |
> | 4 | `timestamp` | 타임스탬프 (나노초) |
> | 1 | `name` | 데이터명 |
> | 2 | `source` | 컴포넌트명 |
> | 5 | `data_type` | 데이터형 |
> | 6 | `data_array` | 배열 수 |

AI 번역본
-For Reference Only-

설명

컴포넌트의 입출력 포트가 수신, 송신하고 있는 데이터의 데이터형입니다.
getInPortDataValue, getSystemPortDataValue, getOutPortDataValue를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.8 ComponentMethodResponse

컴포넌트의 인스턴스 메서드를 실행한 후의 결과

정의

```
type ComponentMethodResponse = {
    method_name: string,
    result: string | object | null
};
```

구성

No	항목	설명
1	`method_name`	실행한 함수명
2	`result`	실행한 함수의 반환값

설명

컴포넌트의 인스턴스 메서드를 실행한 후에 반환되는 데이터의 데이터형입니다.
`result`의 내용은 커스텀 컴포넌트 측의 구현에 따라 달라지므로, 이후에 수행할 처리에 맞춰 커스텀 컴포넌트 측의 반환값을 구현해 주십시오.

15.5.3.9 FlowlinkResponse

플로우 링크 정보

AI 번역본

정의

```
type FlowlinkResponse = {
    id: string,
    source?: {
        id: string,
        port: number,
    },
    destination?: {
        id: string,
        port: number,
    }
};
```

구성

No	항목	설명
1	`id`	플로우 링크 ID
2	`source`	접속원의 정보
3	`destination`	접속처의 정보

> *source, destination*
>
> | No | 항목 | 설명 |
> | --- | --- | --- |
> | 1 | `id` | 접속처의 포트 ID |
> | 2 | `port` | 포트 번호 |

설명

컴포넌트가 접속하고 있는 플로우 링크의 데이터형입니다.
getFlowlinks, getFlowlink를 사용했을 때 반환값으로서 취득할 수 있습니다.

15.5.3.10 FlowlinkFiltersResponse

플로우 링크의 필터 정보

정의

```
type FlowlinkFiltersResponse = {
    setting_option: 'reception' | 'regexp',
    pattern_settings: {
        target: {
            data_name: string,
            data_types?: string[]
        },
        pass: boolean,
    }[],
};
```

AI 번역본
-For Reference Only-

구성

No	항목	설명
1	`setting_option`	필터를 정규 표현식으로 수행할지, 데이터마다 수행할지 여부
2	`pattern_settings`	필터 설정 오브젝트의 목록

pattern_settins

No	항목	설명
1	`target`	대상 데이터를 지정하기 위한 패턴 정의
2	`pass`	값을 통과시킬지 여부의 판정용 플래그

target

No	항목	설명
1	`data_name`	대상으로 하는 데이터명의 패턴을 나타내는 정규 표현식
2	`data_types`	대상으로 하는 데이터형의 리스트(생략/null은 임의의 형에 매칭)

설명

컴포넌트가 접속하고 있는 플로우 링크에 설정된 필터의 데이터형입니다. getFlowlinkFilter를 사용했을 때 반환값으로서 취득할 수 있습니다.

AI 번역본
-For Reference Only-

SpeeDBee Synapse는, 오픈 소스를 이용하고 있습니다. 이용하고 있는 오픈 소스에 대해서는, 아래를 참조하십시오.

- cpp-httplib

```
The MIT License (MIT)

Copyright (c) 2017 yhirose

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- cppcodec

```
Copyright (c) 2015 Topology Inc.
Copyright (c) 2018 Jakob Petsovits
Copyright (c) various other contributors, see individual files

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

AI 번역본
For Reference Only

- ghc

```
Copyright (c) 2018, Steffen Schümann <s.schuemann@pobox.com>

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- martinmoene-optional-lite

```
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```

- martinmoene-variant-lite

```
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```

- nlohmann-json

```
MIT License

Copyright (c) 2013-2022 Niels Lohmann

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- tiny-AES-c

All Rights Reserved
For Reference Only

```
This is free and unencumbered software released into the public domain.

Anyone is free to copy, modify, publish, use, compile, sell, or
distribute this software, either in source code form or as a compiled
binary, for any purpose, commercial or non-commercial, and by any
means.

In jurisdictions that recognize copyright laws, the author or authors
of this software dedicate any and all copyright interest in the
software to the public domain. We make this dedication for the benefit
of the public at large and to the detriment of our heirs and
successors. We intend this dedication to be an overt act of
relinquishment in perpetuity of all present and future rights to this
software under copyright law.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.

For more information, please refer to <http://unlicense.org/>
```

- PicoSHA2

```
MIT License

Copyright (c) 2017 okdshin

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- takamin/win-c

AI 번역본
For Reference Only

```
The MIT License (MIT)

Copyright (c) 2015 Koji Takami

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- tronkko/dirent

```
The MIT License (MIT)

Copyright (c) 1998-2019 Toni Ronkko

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- libuuid

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
1. Redistributions of source code must retain the above copyright
   notice, and the entire permission notice in its entirety,
   including the disclaimer of warranties.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote
   products derived from this software without specific prior
   written permission.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESS OR IMPLIED
WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ALL OF
WHICH ARE HEREBY DISCLAIMED.  IN NO EVENT SHALL THE AUTHOR BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT
OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE
USE OF THIS SOFTWARE, EVEN IF NOT ADVISED OF THE POSSIBILITY OF SUCH
DAMAGE.
```

- libpython

AI 번역본
-For Reference Only-

1. This LICENSE AGREEMENT is between the Python Software Foundation
("PSF"), and the Individual or Organization ("Licensee") accessing and
otherwise using this software ("Python") in source or binary form and
its associated documentation.

2. Subject to the terms and conditions of this License Agreement, PSF
hereby grants Licensee a nonexclusive, royalty-free, world-wide
license to reproduce, analyze, test, perform and/or display publicly,
prepare derivative works, distribute, and otherwise use Python
alone or in any derivative version, provided, however, that PSF's
License Agreement and PSF's notice of copyright, i.e., "Copyright (c)
2001, 2002, 2003, 2004, 2005, 2006 Python Software Foundation; All Rights
Reserved" are retained in Python alone or in any derivative version
prepared by Licensee.

3. In the event Licensee prepares a derivative work that is based on
or incorporates Python or any part thereof, and wants to make
the derivative work available to others as provided herein, then
Licensee hereby agrees to include in any such work a brief summary of
the changes made to Python.

4. PSF is making Python available to Licensee on an "AS IS"
basis. PSF MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, PSF MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

5. PSF SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF PYTHON
FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS
A RESULT OF MODIFYING, DISTRIBUTING, OR OTHERWISE USING PYTHON,
OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

7. Nothing in this License Agreement shall be deemed to create any
relationship of agency, partnership, or joint venture between PSF and
Licensee. This License Agreement does not grant permission to use PSF
trademarks or trade name in a trademark sense to endorse or promote
products or services of Licensee, or any third party.

8. By copying, installing or otherwise using Python, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

AI 번역본
-For Reference Only-

BEOPEN.COM LICENSE AGREEMENT FOR PYTHON 2.0
_____________-

BEOPEN PYTHON OPEN SOURCE LICENSE AGREEMENT VERSION 1

1. This LICENSE AGREEMENT is between BeOpen.com ("BeOpen"), having an
office at 160 Saratoga Avenue, Santa Clara, CA 95051, and the
Individual or Organization ("Licensee") accessing and otherwise using
this software in source or binary form and its associated
documentation ("the Software").

2. Subject to the terms and conditions of this BeOpen Python License

Agreement, BeOpen hereby grants Licensee a non-exclusive,
royalty-free, world-wide license to reproduce, analyze, test, perform
and/or display publicly, prepare derivative works, distribute, and
otherwise use the Software alone or in any derivative version,
provided, however, that the BeOpen Python License is retained in the
Software, alone or in any derivative version prepared by Licensee.

3. BeOpen is making the Software available to Licensee on an “AS IS”
basis. BEOPEN MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR
IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, BEOPEN MAKES NO AND
DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS
FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE WILL NOT
INFRINGE ANY THIRD PARTY RIGHTS.

4. BEOPEN SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE
SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS
AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE, OR ANY
DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

5. This License Agreement will automatically terminate upon a material
breach of its terms and conditions.

6. This License Agreement shall be governed by and interpreted in all
respects by the law of the State of California, excluding conflict of
law provisions. Nothing in this License Agreement shall be deemed to
create any relationship of agency, partnership, or joint venture
between BeOpen and Licensee. This License Agreement does not grant
permission to use BeOpen trademarks or trade names in a trademark
sense to endorse or promote products or services of Licensee, or any
third party. As an exception, the “BeOpen Python” logos available at
http://www.pythonlabs.com/logos.html may be used according to the
permissions granted on that web page.

7. By copying, installing or otherwise using the software, Licensee
agrees to be bound by the terms and conditions of this License
Agreement.

CNRI OPEN SOURCE LICENSE AGREEMENT (for Python 1.6b1)
——————————————

IMPORTANT: PLEASE READ THE FOLLOWING AGREEMENT CAREFULLY.

BY CLICKING ON “ACCEPT” WHERE INDICATED BELOW, OR BY COPYING,
INSTALLING OR OTHERWISE USING PYTHON 1.6, beta 1 SOFTWARE, YOU ARE
DEEMED TO HAVE AGREED TO THE TERMS AND CONDITIONS OF THIS LICENSE
AGREEMENT.

1. This LICENSE AGREEMENT is between the Corporation for National
Research Initiatives, having an office at 1895 Preston White Drive,
Reston, VA 20191 (“CNRI”), and the Individual or Organization
(“Licensee”) accessing and otherwise using Python 1.6, beta 1
software in source or binary form and its associated documentation,
as released at the www.python.org Internet site on August 4, 2000
(“Python 1.6b1”).

2. Subject to the terms and conditions of this License Agreement, CNRI
hereby grants Licensee a non-exclusive, royalty-free, world-wide
license to reproduce, analyze, test, perform and/or display
publicly, prepare derivative works, distribute, and otherwise use

AI 번역 결과
-For Reference Only-

Python 1.6b1 alone or in any derivative version, provided, however, that CNRIs License Agreement is retained in Python 1.6b1, alone or in any derivative version prepared by Licensee.

Alternately, in lieu of CNRIs License Agreement, Licensee may substitute the following text (omitting the quotes): “Python 1.6, beta 1, is made available subject to the terms and conditions in CNRIs License Agreement. This Agreement may be located on the Internet using the following unique, persistent identifier (known as a handle): 1895.22/1011. This Agreement may also be obtained from a proxy server on the Internet using the URL:http://hdl.handle.net/1895.22/1011”.

3. In the event Licensee prepares a derivative work that is based on or incorporates Python 1.6b1 or any part thereof, and wants to make the derivative work available to the public as provided herein, then Licensee hereby agrees to indicate in any such work the nature of the modifications made to Python 1.6b1.

4. CNRI is making Python 1.6b1 available to Licensee on an “AS IS” basis. CNRI MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED. BY WAY OF EXAMPLE, BUT NOT LIMITATION, CNRI MAKES NO AND DISCLAIMS ANY REPRESENTATION OR WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF PYTHON 1.6b1 WILL NOT INFRINGE ANY THIRD PARTY RIGHTS.

5. CNRI SHALL NOT BE LIABLE TO LICENSEE OR ANY OTHER USERS OF THE SOFTWARE FOR ANY INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES OR LOSS AS A RESULT OF USING, MODIFYING OR DISTRIBUTING PYTHON 1.6b1, OR ANY DERIVATIVE THEREOF, EVEN IF ADVISED OF THE POSSIBILITY THEREOF.

6. This License Agreement will automatically terminate upon a material breach of its terms and conditions.

7. This License Agreement shall be governed by and interpreted in all respects by the law of the State of Virginia, excluding conflict of law provisions. Nothing in this License Agreement shall be deemed to create any relationship of agency, partnership, or joint venture between CNRI and Licensee. This License Agreement does not grant permission to use CNRI trademarks or trade name in a trademark sense to endorse or promote products or services of Licensee, or any third party.

8. By clicking on the “ACCEPT” button where indicated, or by copying, installing or otherwise using Python 1.6b1, Licensee agrees to be bound by the terms and conditions of this License Agreement.

ACCEPT

CWI LICENSE AGREEMENT FOR PYTHON 0.9.0 THROUGH 1.2
——————————————

Copyright (c) 1991 – 1995, Stichting Mathematisch Centrum Amsterdam, The Netherlands. All rights reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that

AI 번역본
For Reference Only

```
both that copyright notice and this permission notice appear in
supporting documentation, and that the name of Stichting Mathematisch
Centrum or CWI not be used in advertising or publicity pertaining to
distribution of the software without specific, written prior
permission.

STICHTING MATHEMATISCH CENTRUM DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND
FITNESS, IN NO EVENT SHALL STICHTING MATHEMATISCH CENTRUM BE LIABLE
FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT
OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
```

- pycomm3

```
The MIT License (MIT)

Copyright (c) 2020 Ian Ottoway
Copyright (c) 2014 Agostino Ruscito

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- paramiko

AI 번역본
-For Reference Only-

```
          GNU LESSER GENERAL PUBLIC LICENSE
               Version 2.1, February 1999

 Copyright (C) 1991, 1999 Free Software Foundation, Inc.
     51 Franklin Street, Suite 500, Boston, MA  02110-1335  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL.  It also counts
 as the successor of the GNU Library Public License, version 2, hence
 the version number 2.1.]

                Preamble

  The licenses for most software are designed to take away your
freedom to share and change it.  By contrast, the GNU General Public
Licenses are intended to guarantee your freedom to share and change
free software--to make sure the software is free for all its users.

  This license, the Lesser General Public License, applies to some
specially designated software packages--typically libraries--of the
Free Software Foundation and other authors who decide to use it.  You
can use it too, but we suggest you first think carefully about whether
this license or the ordinary General Public License is the better
strategy to use in any particular case, based on the explanations below.

  When we speak of free software, we are referring to freedom of use,
not price.  Our General Public Licenses are designed to make sure that
you have the freedom to distribute copies of free software (and charge
for this service if you wish); that you receive source code or can get
it if you want it; that you can change the software and use pieces of
it in new free programs; and that you are informed that you can do
these things.

  To protect your rights, we need to make restrictions that forbid
distributors to deny you these rights or to ask you to surrender these
rights.  These restrictions translate to certain responsibilities for
you if you distribute copies of the library or if you modify it.

  For example, if you distribute copies of the library, whether gratis
or for a fee, you must give the recipients all the rights that we gave
you.  You must make sure that they, too, receive or can get the source
code.  If you link other code with the library, you must provide
complete object files to the recipients, so that they can relink them
with the library after making changes to the library and recompiling
it.  And you must show them these terms so they know their rights.

  We protect your rights with a two-step method: (1) we copyright the
library, and (2) we offer you this license, which gives you legal
permission to copy, distribute and/or modify the library.

  To protect each distributor, we want to make it very clear that
there is no warranty for the free library.  Also, if the library is
modified by someone else and passed on, the recipients should know
that what they have is not the original version, so that the original
author's reputation will not be affected by problems that might be
introduced by others.
```

-For Reference Only-

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

AI 번역본
For Reference Only

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

AI 번역 결과
-For Reference Only-

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

    d) If a facility in the modified Library refers to a function or a
    table of data to be supplied by an application program that uses
    the facility, other than as an argument passed when the facility
    is invoked, then you must make a good faith effort to ensure that,
    in the event an application does not supply such function or
    table, the facility still operates, and performs whatever part of
    its purpose remains meaningful.

    (For example, a function in a library to compute square roots has
    a purpose that is entirely well-defined independent of the
    application. Therefore, Subsection 2d requires that any
    application-supplied function or table used by this function must
    be optional: if the application does not supply it, the square
    root function must still compute square roots.)

These requirements apply to the modified work as a whole. If
identifiable sections of that work are not derived from the Library,
and can be reasonably considered independent and separate works in
themselves, then this License, and its terms, do not apply to those
sections when you distribute them as separate works. But when you
distribute the same sections as part of a whole which is a work based
on the Library, the distribution of the whole must be on the terms of
this License, whose permissions for other licensees extend to the
entire whole, and thus to each and every part regardless of who wrote
it.

Thus, it is not the intent of this section to claim rights or contest
your rights to work written entirely by you; rather, the intent is to
exercise the right to control the distribution of derivative or
collective works based on the Library.

In addition, mere aggregation of another work not based on the Library
with the Library (or with a work based on the Library) on a volume of
a storage or distribution medium does not bring the other work under
the scope of this License.

  3. You may opt to apply the terms of the ordinary GNU General Public
License instead of this License to a given copy of the Library. To do
this, you must alter all the notices that refer to this License, so
that they refer to the ordinary GNU General Public License, version 2,
instead of to this License. (If a newer version than version 2 of the
ordinary GNU General Public License has appeared, then you can specify
that version instead if you wish.) Do not make any other change in
these notices.

AI 번역본
-For Reference Only-

  Once this change is made in a given copy, it is irreversible for
that copy, so the ordinary GNU General Public License applies to all
subsequent copies and derivative works made from that copy.

  This option is useful when you wish to copy part of the code of
the Library into a program that is not a library.

  4. You may copy and distribute the Library (or a portion or
derivative of it, under Section 2) in object code or executable form
under the terms of Sections 1 and 2 above provided that you accompany
it with the complete corresponding machine-readable source code, which
must be distributed under the terms of Sections 1 and 2 above on a

medium customarily used for software interchange.

  If distribution of object code is made by offering access to copy
from a designated place, then offering equivalent access to copy the
source code from the same place satisfies the requirement to
distribute the source code, even though third parties are not
compelled to copy the source along with the object code.

  5. A program that contains no derivative of any portion of the
Library, but is designed to work with the Library by being compiled or
linked with it, is called a "work that uses the Library". Such a
work, in isolation, is not a derivative work of the Library, and
therefore falls outside the scope of this License.

  However, linking a "work that uses the Library" with the Library
creates an executable that is a derivative of the Library (because it
contains portions of the Library), rather than a "work that uses the
library". The executable is therefore covered by this License.
Section 6 states terms for distribution of such executables.

  When a "work that uses the Library" uses material from a header file
that is part of the Library, the object code for the work may be a
derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be
linked without the Library, or if the work is itself a library. The
threshold for this to be true is not precisely defined by law.

  If such an object file uses only numerical parameters, data
structure layouts and accessors, and small macros and small inline
functions (ten lines or less in length), then the use of the object
file is unrestricted, regardless of whether it is legally a derivative
work. (Executables containing this object code plus portions of the
Library will still fall under Section 6.)

  Otherwise, if the work is a derivative of the Library, you may
distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6,
whether or not they are linked directly with the Library itself.

  6. As an exception to the Sections above, you may also combine or
link a "work that uses the Library" with the Library to produce a
work containing portions of the Library, and distribute that work
under terms of your choice, provided that the terms permit
modification of the work for the customer's own use and reverse
engineering for debugging such modifications.

AI 번역본
-For Reference Only-

  You must give prominent notice with each copy of the work that the
Library is used in it and that the Library and its use are covered by
this License. You must supply a copy of this License. If the work
during execution displays copyright notices, you must include the
copyright notice for the Library among them, as well as a reference
directing the user to the copy of this License. Also, you must do one
of these things:

    a) Accompany the work with the complete corresponding
    machine-readable source code for the Library including whatever
    changes were used in the work (which must be distributed under
    Sections 1 and 2 above); and, if the work is an executable linked

with the Library, with the complete machine-readable "work that
uses the Library", as object code and/or source code, so that the
user can modify the Library and then relink to produce a modified
executable containing the modified Library. (It is understood
that the user who changes the contents of definitions files in the
Library will not necessarily be able to recompile the application
to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the
Library. A suitable mechanism is one that (1) uses at run time a
copy of the library already present on the user's computer system,
rather than copying library functions into the executable, and (2)
will operate properly with a modified version of the library, if
the user installs one, as long as the modified version is
interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at
least three years, to give the same user the materials
specified in Subsection 6a, above, for a charge no more
than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy
from a designated place, offer equivalent access to copy the above
specified materials from the same place.

e) Verify that the user has already received a copy of these
materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the
Library" must include any data and utility programs needed for
reproducing the executable from it. However, as a special exception,
the materials to be distributed need not include anything that is
normally distributed (in either source or binary form) with the major
components (compiler, kernel, and so on) of the operating system on
which the executable runs, unless that component itself accompanies
the executable.

It may happen that this requirement contradicts the license
restrictions of other proprietary libraries that do not normally
accompany the operating system. Such a contradiction means you cannot
use both them and the Library together in an executable that you
distribute.

AI 번역본
For Reference Only

7. You may place library facilities that are a work based on the
Library side-by-side in a single library together with other library
facilities not covered by this License, and distribute such a combined
library, provided that the separate distribution of the work based on
the Library and of the other library facilities is otherwise
permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work
based on the Library, uncombined with any other library
facilities. This must be distributed under the terms of the
Sections above.

b) Give prominent notice with the combined library of the fact
that part of it is a work based on the Library, and explaining
where to find the accompanying uncombined form of the same work.

  8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

  9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

  10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

  11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

  12. If the distribution and/or use of the Library is restricted in

AI 번역 결과
-For Reference Only-

certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest

AI HUB
For Reference Only

```
possible use to the public, we recommend making it free software that
everyone can redistribute and change.  You can do so by permitting
redistribution under these terms (or, alternatively, under the terms of the
ordinary General Public License).

  To apply these terms, attach the following notices to the library.  It is
safest to attach them to the start of each source file to most effectively
convey the exclusion of warranty; and each file should have at least the
"copyright" line and a pointer to where the full notice is found.

    <one line to give the library's name and a brief idea of what it does.>
    Copyright (C) <year>  <name of author>

    This library is free software; you can redistribute it and/or
    modify it under the terms of the GNU Lesser General Public
    License as published by the Free Software Foundation; either
    version 2.1 of the License, or (at your option) any later version.

    This library is distributed in the hope that it will be useful,
    but WITHOUT ANY WARRANTY; without even the implied warranty of
    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
    Lesser General Public License for more details.

    You should have received a copy of the GNU Lesser General Public
    License along with this library; if not, write to the Free Software
    Foundation, Inc., 51 Franklin Street, Suite 500, Boston, MA  02110-1335  USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your
school, if any, to sign a "copyright disclaimer" for the library, if
necessary.  Here is a sample; alter the names:

  Yoyodyne, Inc., hereby disclaims all copyright interest in the
  library `Frob' (a library for tweaking knobs) written by James Random Hacker.

  <signature of Ty Coon>, 1 April 1990
  Ty Coon, President of Vice

That's all there is to it!
```

- wheel

AI 번역본
-For Reference Only-

```
MIT License

Copyright (c) 2012 Daniel Holth <dholth@fastmail.fm> and contributors

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL
THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
OTHER DEALINGS IN THE SOFTWARE.
```

- paho.MQTT

```
This project is dual licensed under the Eclipse Public License 2.0 and the
Eclipse Distribution License 1.0 as described in the epl-v20 and edl-v10 files.
```

- pahoMQTT

```
Eclipse Public License - v 2.0

This program and the accompanying materials
are made available under the terms of the Eclipse Public License v2.0
and Eclipse Distribution License v1.0 which accompany this distribution.

The Eclipse Public License is available at
  https://www.eclipse.org/legal/epl-2.0/
and the Eclipse Distribution License is available at
  http://www.eclipse.org/org/documents/edl-v10.php.

For an explanation of what dual-licensing means to you, see:
https://www.eclipse.org/legal/eplfaq.php#DUALLIC
```

-For Reference Only-

- OpenSSH

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

-For Reference Only-

```
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
```

-For Reference Only-

notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

AI Use
-For Reference Only-

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

```
   END OF TERMS AND CONDITIONS
```

- json-c

```
Copyright (c) 2009-2012 Eric Haszlakiewicz

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.

----------------------------------------------------------------

Copyright (c) 2004, 2005 Metaparadigm Pte Ltd

Permission is hereby granted, free of charge, to any person obtaining a
copy of this software and associated documentation files (the "Software"),
to deal in the Software without restriction, including without limitation
the rights to use, copy, modify, merge, publish, distribute, sublicense,
and/or sell copies of the Software, and to permit persons to whom the
Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included
in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- Protocol Buffers

```
Copyright 2008 Google Inc.  All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * Redistributions in binary form must reproduce the above
copyright notice, this list of conditions and the following disclaimer
in the documentation and/or other materials provided with the
distribution.
    * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
(INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Code generated by the Protocol Buffer compiler is owned by the owner
of the input file used when generating it.  This code is not
standalone and requires a support library to be linked with it.  This
support library is itself covered by the above license.
```

- libcares(c-ares)

AI 번역본
-For Reference Only-

```
MIT License

Copyright (c) 1998 Massachusetts Institute of Technology
Copyright (c) 2007 - 2023 Daniel Stenberg with many contributors, see AUTHORS
file.

Permission is hereby granted, free of charge, to any person obtaining a copy of
this software and associated documentation files (the "Software"), to deal in
the Software without restriction, including without limitation the rights to
use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of
the Software, and to permit persons to whom the Software is furnished to do so,
subject to the following conditions:

The above copyright notice and this permission notice (including the next
paragraph) shall be included in all copies or substantial portions of the
Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

- grpc

AI 번역본
-For Reference Only-

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

-For Reference Only-

excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution

AI Store
-For Reference Only-

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

AI Use
-For Reference Only-

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

```
   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.

-----------------------------------------------------------

BSD 3-Clause License

Copyright 2016, Google Inc.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
this list of conditions and the following disclaimer in the documentation
and/or other materials provided with the distribution.

3. Neither the name of the copyright holder nor the names of its
contributors may be used to endorse or promote products derived from this
software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
THE POSSIBILITY OF SUCH DAMAGE.

-----------------------------------------------------------
```

AI 번역본
For Reference Only

```
Mozilla Public License Version 2.0
==================================

1. Definitions
--------------

1.1. "Contributor"
    means each individual or legal entity that creates, contributes to
    the creation of, or owns Covered Software.

1.2. "Contributor Version"
    means the combination of the Contributions of others (if any) used
    by a Contributor and that particular Contributor's Contribution.

1.3. "Contribution"
    means Covered Software of a particular Contributor.

1.4. "Covered Software"
    means Source Code Form to which the initial Contributor has attached
    the notice in Exhibit A, the Executable Form of such Source Code
    Form, and Modifications of such Source Code Form, in each case
    including portions thereof.

1.5. "Incompatible With Secondary Licenses"
    means

    (a) that the initial Contributor has attached the notice described
        in Exhibit B to the Covered Software; or

    (b) that the Covered Software was made available under the terms of
        version 1.1 or earlier of the License, but not also under the
        terms of a Secondary License.

1.6. "Executable Form"
    means any form of the work other than Source Code Form.

1.7. "Larger Work"
    means a work that combines Covered Software with other material, in
    a separate file or files, that is not Covered Software.

1.8. "License"
    means this document.

1.9. "Licensable"
    means having the right to grant, to the maximum extent possible,
    whether at the time of the initial grant or subsequently, any and
    all of the rights conveyed by this License.

1.10. "Modifications"
    means any of the following:

    (a) any file in Source Code Form that results from an addition to,
        deletion from, or modification of the contents of Covered
        Software; or

    (b) any new file in Source Code Form that contains any Covered
        Software.
```

AI 번역본
For Reference Only

1.11. "Patent Claims" of a Contributor
    means any patent claim(s), including without limitation, method,
    process, and apparatus claims, in any patent Licensable by such
    Contributor that would be infringed, but for the grant of the
    License, by the making, using, selling, offering for sale, having
    made, import, or transfer of either its Contributions or its
    Contributor Version.

1.12. "Secondary License"
    means either the GNU General Public License, Version 2.0, the GNU
    Lesser General Public License, Version 2.1, the GNU Affero General
    Public License, Version 3.0, or any later versions of those
    licenses.

1.13. "Source Code Form"
    means the form of the work preferred for making modifications.

1.14. "You" (or "Your")
    means an individual or a legal entity exercising rights under this
    License. For legal entities, "You" includes any entity that
    controls, is controlled by, or is under common control with You. For
    purposes of this definition, "control" means (a) the power, direct
    or indirect, to cause the direction or management of such entity,
    whether by contract or otherwise, or (b) ownership of more than
    fifty percent (50%) of the outstanding shares or beneficial
    ownership of such entity.

2. License Grants and Conditions

2.1. Grants

Each Contributor hereby grants You a world-wide, royalty-free,
non-exclusive license:

(a) under intellectual property rights (other than patent or trademark)
    Licensable by such Contributor to use, reproduce, make available,
    modify, display, perform, distribute, and otherwise exploit its
    Contributions, either on an unmodified basis, with Modifications, or
    as part of a Larger Work; and

(b) under Patent Claims of such Contributor to make, use, sell, offer
    for sale, have made, import, and otherwise transfer either its
    Contributions or its Contributor Version.

AI 번역 결과
-For Reference Only-

2.2. Effective Date

The licenses granted in Section 2.1 with respect to any Contribution
become effective for each Contribution on the date the Contributor first
distributes such Contribution.

2.3. Limitations on Grant Scope

The licenses granted in this Section 2 are the only rights granted under
this License. No additional rights or licenses will be implied from the
distribution or licensing of Covered Software under this License.
Notwithstanding Section 2.1(b) above, no patent license is granted by a
Contributor:

(a) for any code that a Contributor has removed from Covered Software;
    or

(b) for infringements caused by: (i) Your and any other third party's
    modifications of Covered Software, or (ii) the combination of its
    Contributions with other software (except as part of its Contributor
    Version); or

(c) under Patent Claims infringed by Covered Software in the absence of
    its Contributions.

This License does not grant any rights in the trademarks, service marks,
or logos of any Contributor (except as may be necessary to comply with
the notice requirements in Section 3.4).

2.4. Subsequent Licenses

No Contributor makes additional grants as a result of Your choice to
distribute the Covered Software under a subsequent version of this
License (see Section 10.2) or under the terms of a Secondary License (if
permitted under the terms of Section 3.3).

2.5. Representation

Each Contributor represents that the Contributor believes its
Contributions are its original creation(s) or it has sufficient rights
to grant the rights to its Contributions conveyed by this License.

2.6. Fair Use

This License is not intended to limit any rights You have under
applicable copyright doctrines of fair use, fair dealing, or other
equivalents.

2.7. Conditions

Sections 3.1, 3.2, 3.3, and 3.4 are conditions of the licenses granted
in Section 2.1.

3. Responsibilities

3.1. Distribution of Source Form

All distribution of Covered Software in Source Code Form, including any
Modifications that You create or to which You contribute, must be under
the terms of this License. You must inform recipients that the Source
Code Form of the Covered Software is governed by the terms of this
License, and how they can obtain a copy of this License. You may not
attempt to alter or restrict the recipients' rights in the Source Code
Form.

3.2. Distribution of Executable Form

If You distribute Covered Software in Executable Form then:

(a) such Covered Software must also be made available in Source Code
    Form, as described in Section 3.1, and You must inform recipients of
    the Executable Form how they can obtain a copy of such Source Code

AI 번역본
-for reference only-

Form by reasonable means in a timely manner, at a charge no more
than the cost of distribution to the recipient; and

(b) You may distribute such Executable Form under the terms of this
License, or sublicense it under different terms, provided that the
license for the Executable Form does not attempt to limit or alter
the recipients' rights in the Source Code Form under this License.

3.3. Distribution of a Larger Work

You may create and distribute a Larger Work under terms of Your choice,
provided that You also comply with the requirements of this License for
the Covered Software. If the Larger Work is a combination of Covered
Software with a work governed by one or more Secondary Licenses, and the
Covered Software is not Incompatible With Secondary Licenses, this
License permits You to additionally distribute such Covered Software
under the terms of such Secondary License(s), so that the recipient of
the Larger Work may, at their option, further distribute the Covered
Software under the terms of either this License or such Secondary
License(s).

3.4. Notices

You may not remove or alter the substance of any license notices
(including copyright notices, patent notices, disclaimers of warranty,
or limitations of liability) contained within the Source Code Form of
the Covered Software, except that You may alter any license notices to
the extent required to remedy known factual inaccuracies.

3.5. Application of Additional Terms

You may choose to offer, and to charge a fee for, warranty, support,
indemnity or liability obligations to one or more recipients of Covered
Software. However, You may do so only on Your own behalf, and not on
behalf of any Contributor. You must make it absolutely clear that any
such warranty, support, indemnity, or liability obligation is offered by
You alone, and You hereby agree to indemnify every Contributor for any
liability incurred by such Contributor as a result of warranty, support,
indemnity or liability terms You offer. You may include additional
disclaimers of warranty and limitations of liability specific to any
jurisdiction.

4. Inability to Comply Due to Statute or Regulation

If it is impossible for You to comply with any of the terms of this
License with respect to some or all of the Covered Software due to
statute, judicial order, or regulation then You must: (a) comply with
the terms of this License to the maximum extent possible; and (b)
describe the limitations and the code they affect. Such description must
be placed in a text file included with all distributions of the Covered
Software under this License. Except to the extent prohibited by statute
or regulation, such description must be sufficiently detailed for a
recipient of ordinary skill to be able to understand it.

5. Termination

5.1. The rights granted under this License will terminate automatically

AI 번역본
For Reference Only

```
if You fail to comply with any of its terms. However, if You become
compliant, then the rights granted under this License from a particular
Contributor are reinstated (a) provisionally, unless and until such
Contributor explicitly and finally terminates Your grants, and (b) on an
ongoing basis, if such Contributor fails to notify You of the
non-compliance by some reasonable means prior to 60 days after You have
come back into compliance. Moreover, Your grants from a particular
Contributor are reinstated on an ongoing basis if such Contributor
notifies You of the non-compliance by some reasonable means, this is the
first time You have received notice of non-compliance with this License
from such Contributor, and You become compliant prior to 30 days after
Your receipt of the notice.

5.2. If You initiate litigation against any entity by asserting a patent
infringement claim (excluding declaratory judgment actions,
counter-claims, and cross-claims) alleging that a Contributor Version
directly or indirectly infringes any patent, then the rights granted to
You by any and all Contributors for the Covered Software under Section
2.1 of this License shall terminate.

5.3. In the event of termination under Sections 5.1 or 5.2 above, all
end user license agreements (excluding distributors and resellers) which
have been validly granted by You or Your distributors under this License
prior to termination shall survive termination.

************************************************************************
*                                                                      *
*  6. Disclaimer of Warranty                                           *
*  -------------------------                                           *
*                                                                      *
*  Covered Software is provided under this License on an "as is"       *
*  basis, without warranty of any kind, either expressed, implied, or  *
*  statutory, including, without limitation, warranties that the       *
*  Covered Software is free of defects, merchantable, fit for a        *
*  particular purpose or non-infringing. The entire risk as to the     *
*  quality and performance of the Covered Software is with You.        *
*  Should any Covered Software prove defective in any respect, You     *
*  (not any Contributor) assume the cost of any necessary servicing,   *
*  repair, or correction. This disclaimer of warranty constitutes an   *
*  essential part of this License. No use of any Covered Software is   *
*  authorized under this License except under this disclaimer.         *
*                                                                      *
************************************************************************

************************************************************************
*                                                                      *
*  7. Limitation of Liability                                          *
*  --------------------------                                          *
*                                                                      *
*  Under no circumstances and under no legal theory, whether tort      *
*  (including negligence), contract, or otherwise, shall any           *
*  Contributor, or anyone who distributes Covered Software as          *
*  permitted above, be liable to You for any direct, indirect,         *
*  special, incidental, or consequential damages of any character      *
*  including, without limitation, damages for lost profits, loss of    *
*  goodwill, work stoppage, computer failure or malfunction, or any    *
*  and all other commercial damages or losses, even if such party      *
*  shall have been informed of the possibility of such damages. This   *
*  limitation of liability shall not apply to liability for death or   *
```

AI 번역본
For Reference Only

```
*  personal injury resulting from such party's negligence to the       *
*  extent applicable law prohibits such limitation. Some               *
*  jurisdictions do not allow the exclusion or limitation of           *
*  incidental or consequential damages, so this exclusion and          *
*  limitation may not apply to You.                                    *
*                                                                      *
************************************************************************
```

8. Litigation

Any litigation relating to this License may be brought only in the courts of a jurisdiction where the defendant maintains its principal place of business and such litigation shall be governed by laws of that jurisdiction, without reference to its conflict-of-law provisions. Nothing in this Section shall prevent a party's ability to bring cross-claims or counter-claims.

9. Miscellaneous

This License represents the complete agreement concerning the subject matter hereof. If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable. Any law or regulation which provides that the language of a contract shall be construed against the drafter shall not be used to construe this License against a Contributor.

10. Versions of the License

10.1. New Versions

Mozilla Foundation is the license steward. Except as provided in Section 10.3, no one other than the license steward has the right to modify or publish new versions of this License. Each version will be given a distinguishing version number.

10.2. Effect of New Versions

You may distribute the Covered Software under the terms of the version of the License under which You originally received the Covered Software, or under the terms of any subsequent version published by the license steward.

AI 번역본
-For Reference Only-

10.3. Modified Versions

If you create software not governed by this License, and you want to create a new license for such software, you may create and use a modified version of this License if you rename the license and remove any references to the name of the license steward (except to note that such modified license differs from this License).

10.4. Distributing Source Code Form that is Incompatible With Secondary Licenses

If You choose to distribute Source Code Form that is Incompatible With Secondary Licenses under the terms of this version of the License, the notice described in Exhibit B of this License must be attached.

```
Exhibit A - Source Code Form License Notice
-------------------------------------------

  This Source Code Form is subject to the terms of the Mozilla Public
  License, v. 2.0. If a copy of the MPL was not distributed with this
  file, You can obtain one at http://mozilla.org/MPL/2.0/.

If it is not possible or desirable to put the notice in a particular
file, then You may include the notice in a location (such as a LICENSE
file in a relevant directory) where a recipient would be likely to look
for such a notice.

You may add additional accurate notices of copyright ownership.

Exhibit B - "Incompatible With Secondary Licenses" Notice
---------------------------------------------------------

  This Source Code Form is "Incompatible With Secondary Licenses", as
  defined by the Mozilla Public License, v. 2.0.
```

- re2

```
// Copyright (c) 2009 The RE2 Authors. All rights reserved.
//
// Redistribution and use in source and binary forms, with or without
// modification, are permitted provided that the following conditions are
// met:
//
//    * Redistributions of source code must retain the above copyright
// notice, this list of conditions and the following disclaimer.
//    * Redistributions in binary form must reproduce the above
// copyright notice, this list of conditions and the following disclaimer
// in the documentation and/or other materials provided with the
// distribution.
//    * Neither the name of Google Inc. nor the names of its
// contributors may be used to endorse or promote products derived from
// this software without specific prior written permission.
//
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
// "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
// LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
// OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
// LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
// DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
// THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
// (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
// OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

- abseil

ARCHIVE
For Reference Only

Apache License
Version 2.0, January 2004
https://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but

-For Reference Only-

```
   excluding communication that is conspicuously marked or otherwise
   designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity
   on behalf of whom a Contribution has been received by Licensor and
   subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   copyright license to reproduce, prepare Derivative Works of,
   publicly display, publicly perform, sublicense, and distribute the
   Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
```

AI Store
-For Reference Only-

notices within Derivative Works that You distribute, alongside
or as an addendum to the NOTICE text from the Work, provided
that such additional attribution notices cannot be construed
as modifying the License.

You may add Your own copyright statement to Your modifications and
may provide additional or different license terms and conditions
for use, reproduction, or distribution of Your modifications, or
for any such Derivative Works as a whole, provided Your use,
reproduction, and distribution of the Work otherwise complies with
the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise,
any Contribution intentionally submitted for inclusion in the Work
by You to the Licensor shall be under the terms and conditions of
this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify
the terms of any separate license agreement you may have executed
with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade
names, trademarks, service marks, or product names of the Licensor,
except as required for reasonable and customary use in describing the
origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or
agreed to in writing, Licensor provides the Work (and each
Contributor provides its Contributions) on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied, including, without limitation, any warranties or conditions
of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A
PARTICULAR PURPOSE. You are solely responsible for determining the
appropriateness of using or redistributing the Work and assume any
risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory,
whether in tort (including negligence), contract, or otherwise,
unless required by applicable law (such as deliberate and grossly
negligent acts) or agreed to in writing, shall any Contributor be
liable to You for damages, including any direct, indirect, special,
incidental, or consequential damages of any character arising as a
result of this License or out of the use or inability to use the
Work (including but not limited to damages for loss of goodwill,
work stoppage, computer failure or malfunction, or any and all
other commercial damages or losses), even if such Contributor
has been advised of the possibility of such damages.

AI Use
-For Reference Only-

9. Accepting Warranty or Additional Liability. While redistributing
the Work or Derivative Works thereof, You may choose to offer,
and charge a fee for, acceptance of support, warranty, indemnity,
or other liability obligations and/or rights consistent with this
License. However, in accepting such obligations, You may act only
on Your own behalf and on Your sole responsibility, not on behalf
of any other Contributor, and only if You agree to indemnify,
defend, and hold each Contributor harmless for any liability
incurred by, or claims asserted against, such Contributor by reason
of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

```
   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       https://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

- 그 밖에 WEBUI에서 이용하고 있는 오픈 소스

AI 번역본
-For Reference Only-

```
Name                          Version         License
 PyYAML                        6.0.2           MIT License
 aiofiles                      23.2.1          Apache Software License
 aiohappyeyeballs              2.4.4           Other/Proprietary License; Python Software Foundation
License
 aiohttp                       3.10.11         Apache Software License
 aiohttp-sse-client            0.2.1           Apache Software License
 aiorwlock                     1.3.0           Apache Software License
 aiosignal                     1.3.1           Apache Software License
 annotated-types               0.7.0           MIT License
 anyio                         4.5.2           MIT License
 async-timeout                 5.0.1           Apache Software License
 attrs                         25.3.0          MIT License
 bcrypt                        4.3.0           Apache Software License
 certifi                       2025.4.26       Mozilla Public License 2.0 (MPL 2.0)
 cffi                          1.17.1          MIT License
 click                         8.1.8           BSD License
 cryptography                  45.0.3          Apache-2.0 OR BSD-3-Clause
 ecdsa                         0.19.1          MIT
 exceptiongroup                1.3.0           MIT License
 execnet                       2.1.1           MIT License
 fastapi                       0.115.8         MIT License
 frozenlist                    1.5.0           Apache Software License
 grafana-api-sdk               0.3.1           Apache Software License
 h11                           0.14.0          MIT License
 httpcore                      0.17.3          BSD License
 httptools                     0.6.4           MIT License
 httpx                         0.24.1          BSD License
 idna                          3.10            BSD License
 importlib_resources           6.4.5           Apache Software License
 iniconfig                     2.1.0           MIT License
 itsdangerous                  2.1.2           BSD License
 jsonschema-specifications     2023.12.1       MIT License
 multidict                     6.1.0           Apache Software License
 mypy_extensions               1.1.0           MIT License
 packaging                     25.0            Apache Software License; BSD License
 passlib                       1.7.4           BSD
 pkgutil_resolve_name          1.3.10          MIT License
 pluggy                        1.5.0           MIT License
 propcache                     0.2.0           Apache Software License
 pyasn1                        0.6.1           BSD License
 pycodestyle                   2.12.1          MIT License
 pycparser                     2.22            BSD License
 pycryptodomex                 3.23.0          BSD License; Public Domain
 pydantic                      2.10.6          MIT License
 pydantic_core                 2.27.2          MIT License
 python-dotenv                 1.0.1           BSD License
 python-jose                   3.3.0           MIT License
 python-multipart              0.0.6           Apache Software License
 pyzipper                      0.3.6           MIT License
 referencing                   0.35.1          MIT License
 rpds-py                       0.20.1          MIT License
 rsa                           4.9.1           Apache Software License
 six                           1.17.0          MIT License
 sniffio                       1.3.1           Apache Software License; MIT License
 sse-starlette                 1.6.5           BSD License
 starlette                     0.44.0          BSD License
 types-pyasn1                  0.6.0.20240913  Apache Software License
```

AI 번역본
-For Reference Only-

```
 typing_extensions          4.13.2          Python Software Foundation License
 uvicorn                    0.23.2          BSD License
 uvloop                     0.21.0          Apache Software License; MIT License
 watchfiles                 0.24.0          MIT License
 websockets                 13.1            BSD License
 yarl                       1.15.2          Apache Software License
 zipp                       3.20.2          MIT License

--------------------------------------------------------------------------------
```


```
{
  "@babel/helper-string-parser@7.25.9": {
    "licenses": "MIT",
    "repository": "https://github.com/babel/babel",
    "publisher": "The Babel Team",
    "url": "https://babel.dev/team"
  },
  "@babel/helper-validator-identifier@7.25.9": {
    "licenses": "MIT",
    "repository": "https://github.com/babel/babel",
    "publisher": "The Babel Team",
    "url": "https://babel.dev/team"
  },
  "@babel/parser@7.26.8": {
    "licenses": "MIT",
    "repository": "https://github.com/babel/babel",
    "publisher": "The Babel Team",
    "url": "https://babel.dev/team"
  },
  "@babel/types@7.26.8": {
    "licenses": "MIT",
    "repository": "https://github.com/babel/babel",
    "publisher": "The Babel Team",
    "url": "https://babel.dev/team"
  },
  "@jridgewell/sourcemap-codec@1.5.0": {
    "licenses": "MIT",
    "repository": "https://github.com/jridgewell/sourcemap-codec",
    "publisher": "Rich Harris"
  },
  "@kurkle/color@0.3.4": {
    "licenses": "MIT",
    "repository": "https://github.com/kurkle/color",
    "publisher": "Jukka Kurkela"
  },
  "@quasar/extras@1.16.17": {
    "licenses": "MIT",
    "repository": "https://github.com/quasarframework/quasar",
    "publisher": "Razvan Stoenescu"
  },
  "@vue/compiler-core@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/compiler-dom@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
```


AI 번역본 -For Reference Only-

```
  },
  "@vue/compiler-sfc@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/compiler-ssr@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/reactivity@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/runtime-core@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/runtime-dom@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/server-renderer@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/shared@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "chart.js@4.4.7": {
    "licenses": "MIT",
    "repository": "https://github.com/chartjs/Chart.js"
  },
  "cross-env@7.0.3": {
    "licenses": "MIT",
    "repository": "https://github.com/kentcdodds/cross-env",
    "publisher": "Kent C. Dodds",
    "email": "me@kentcdodds.com",
    "url": "https://kentcdodds.com"
  },
  "cross-spawn@7.0.6": {
    "licenses": "MIT",
    "repository": "https://github.com/moxystudio/node-cross-spawn",
    "publisher": "André Cruz",
    "email": "andre@moxy.studio"
  },
  "csstype@3.1.3": {
    "licenses": "MIT",
    "repository": "https://github.com/frenic/csstype",
    "publisher": "Fredrik Nicol",
    "email": "fredrik.nicol@gmail.com"
  },
```

AI 번역본
-For Reference Only-

```
  "drawflow@0.0.59": {
    "licenses": "MIT",
    "repository": "https://github.com/jerosoler/Drawflow",
    "publisher": "Jero soler"
  },
  "emittery@1.1.0": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/emittery",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "https://sindresorhus.com"
  },
  "entities@4.5.0": {
    "licenses": "BSD-2-Clause",
    "repository": "https://github.com/fb55/entities",
    "publisher": "Felix Boehm",
    "email": "me@feedic.com"
  },
  "estree-walker@2.0.2": {
    "licenses": "MIT",
    "repository": "https://github.com/Rich-Harris/estree-walker",
    "publisher": "Rich Harris"
  },
  "hls.js@1.5.20": {
    "licenses": "Apache-2.0",
    "repository": "https://github.com/video-dev/hls.js"
  },
  "isexe@2.0.0": {
    "licenses": "ISC",
    "repository": "https://github.com/isaacs/isexe",
    "publisher": "Isaac Z. Schlueter",
    "email": "i@izs.me",
    "url": "http://blog.izs.me/"
  },
  "magic-string@0.30.17": {
    "licenses": "MIT",
    "repository": "https://github.com/rich-harris/magic-string",
    "publisher": "Rich Harris"
  },
  "maska@2.1.11": {
    "licenses": "MIT",
    "repository": "https://github.com/beholdr/maska",
    "publisher": "Alexander Shabunevich"
  },
  "nanoid@3.3.8": {
    "licenses": "MIT",
    "repository": "https://github.com/ai/nanoid",
    "publisher": "Andrey Sitnik",
    "email": "andrey@sitnik.ru"
  },
  "path-key@3.1.1": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/path-key",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "sindresorhus.com"
  },
  "picocolors@1.1.1": {
    "licenses": "ISC",
```


AI 번역본
For Reference Only-

```
    "repository": "https://github.com/alexeyraspopov/picocolors",
    "publisher": "Alexey Raspopov"
  },
  "postcss@8.5.2": {
    "licenses": "MIT",
    "repository": "https://github.com/postcss/postcss",
    "publisher": "Andrey Sitnik",
    "email": "andrey@sitnik.ru"
  },
  "quasar@2.17.7": {
    "licenses": "MIT",
    "repository": "https://github.com/quasarframework/quasar",
    "publisher": "Razvan Stoenescu",
    "email": "razvan.stoenescu@gmail.com",
    "url": "https://quasar.dev"
  },
  "shebang-command@2.0.0": {
    "licenses": "MIT",
    "repository": "https://github.com/kevva/shebang-command",
    "publisher": "Kevin Mårtensson",
    "email": "kevinmartensson@gmail.com",
    "url": "github.com/kevva"
  },
  "shebang-regex@3.0.0": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/shebang-regex",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "sindresorhus.com"
  },
  "sortablejs@1.15.6": {
    "licenses": "MIT",
    "repository": "https://github.com/SortableJS/Sortable"
  },
  "source-map-js@1.2.1": {
    "licenses": "BSD-3-Clause",
    "repository": "https://github.com/7rulnik/source-map-js",
    "publisher": "Valentin 7rulnik Semirulnik",
    "email": "v7rulnik@gmail.com"
  },
  "typescript@5.7.3": {
    "licenses": "Apache-2.0",
    "repository": "https://github.com/microsoft/TypeScript",
    "publisher": "Microsoft Corp."
  },
  "vue-chartjs@5.3.2": {
    "licenses": "MIT",
    "repository": "https://github.com/apertureless/vue-chartjs",
    "publisher": "Jakub Juszczak",
    "email": "jakub@posteo.de"
  },
  "vue@3.5.13": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "when-json-met-bigint@0.27.0": {
    "licenses": "MIT",
    "repository": "https://github.com/haoadoresorange/when-json-met-bigint",
```


AI 번역본
-For Reference Only-

```
    "publisher": "haoadoresorange"
  },
  "which@2.0.2": {
    "licenses": "ISC",
    "repository": "https://github.com/isaacs/node-which",
    "publisher": "Isaac Z. Schlueter",
    "email": "i@izs.me",
    "url": "http://blog.izs.me"
  }
}
```

i-Trigger에서 이용하고 있는 오픈 소스는, 아래를 참조하십시오.

- FFmpeg

  This software uses code of FFmpeg licensed under the LGPLv2.1

  its source can be downloaded here

AI 번역본
-For Reference Only-

```
License
Most files in FFmpeg are under the GNU Lesser General Public License version 2.1 or later
(LGPL v2.1+). Read the file COPYING.LGPLv2.1 for details. Some other files have MIT/X11/BSD-
style licenses. In combination the LGPL v2.1+ applies to FFmpeg.

Some optional parts of FFmpeg are licensed under the GNU General Public License version 2 or
later (GPL v2+). See the file COPYING.GPLv2 for details. None of these parts are used by
default, you have to explicitly pass --enable-gpl to configure to activate them. In this case,
FFmpeg's license changes to GPL v2+.

Specifically, the GPL parts of FFmpeg are:

libpostproc
optional x86 optimization in the files
libavcodec/x86/flac_dsp_gpl.asm
libavcodec/x86/idct_mmx.c
libavfilter/x86/vf_removegrain.asm
the following building and testing tools
compat/solaris/make_sunver.pl
doc/t2h.pm
doc/texi2pod.pl
libswresample/tests/swresample.c
tests/checkasm/*
tests/tiny_ssim.c
the following filters in libavfilter:
signature_lookup.c
vf_blackframe.c
vf_boxblur.c
vf_colormatrix.c
vf_cover_rect.c
vf_cropdetect.c
vf_delogo.c
vf_eq.c
vf_find_rect.c
vf_fspp.c
vf_histeq.c
vf_hqdn3d.c
vf_kerndeint.c
vf_lensfun.c (GPL version 3 or later)
vf_mcdeint.c
vf_mpdecimate.c
vf_nnedi.c
vf_owdenoise.c
vf_perspective.c
vf_phase.c
vf_pp.c
vf_pp7.c
vf_pullup.c
vf_repeatfields.c
vf_sab.c
vf_signature.c
vf_smartblur.c
vf_spp.c
vf_stereo3d.c
vf_super2xsai.c
vf_tinterlace.c
vf_uspp.c
vf_vaguedenoiser.c
```

AI 번역본
-For Reference Only-

vsrc_mptestsrc.c
Should you, for whatever reason, prefer to use version 3 of the (L)GPL, then the configure parameter --enable-version3 will activate this licensing option for you. Read the file COPYING.LGPLv3 or, if you have enabled GPL parts, COPYING.GPLv3 to learn the exact legal terms that apply in this case.

There are a handful of files under other licensing terms, namely:

The files libavcodec/jfdctfst.c, libavcodec/jfdctint_template.c and libavcodec/jrevdct.c are taken from libjpeg, see the top of the files for licensing details. Specifically note that you must credit the IJG in the documentation accompanying your program if you only distribute executables. You must also indicate any changes including additions and deletions to those three files in the documentation.
tests/reference.pnm is under the expat license.
External libraries
FFmpeg can be combined with a number of external libraries, which sometimes affect the licensing of binaries resulting from the combination.

Compatible libraries
The following libraries are under GPL version 2:

avisynth
frei0r
libcdio
libdavs2
librubberband
libvidstab
libx264
libx265
libxavs
libxavs2
libxvid
When combining them with FFmpeg, FFmpeg needs to be licensed as GPL as well by passing --enable-gpl to configure.

The following libraries are under LGPL version 3:

gmp
libaribb24
liblensfun
When combining them with FFmpeg, use the configure option --enable-version3 to upgrade FFmpeg to the LGPL v3.

The VMAF, mbedTLS, RK MPI, OpenCORE and VisualOn libraries are under the Apache License 2.0. That license is incompatible with the LGPL v2.1 and the GPL v2, but not with version 3 of those licenses. So to combine these libraries with FFmpeg, the license version needs to be upgraded by passing --enable-version3 to configure.

The smbclient library is under the GPL v3, to combine it with FFmpeg, the options --enable-gpl and --enable-version3 have to be passed to configure to upgrade FFmpeg to the GPL v3.

Incompatible libraries
There are certain libraries you can combine with FFmpeg whose licenses are not compatible with the GPL and/or the LGPL. If you wish to enable these libraries, even in circumstances that their license may be incompatible, pass --enable-nonfree to configure. This will cause the resulting binary to be unredistributable.

AI 번역본
-For Reference Only-

```
The Fraunhofer FDK AAC and OpenSSL libraries are under licenses which are incompatible with
the GPLv2 and v3. To the best of our knowledge, they are compatible with the LGPL.
```

- 그 밖에 WEBUI에서 이용하고 있는 오픈 소스(i-Trigger)

AI 번역본
-For Reference Only-

```
{
  "@babel/parser@7.24.8": {
    "licenses": "MIT",
    "repository": "https://github.com/babel/babel",
    "publisher": "The Babel Team",
    "url": "https://babel.dev/team"
  },
  "@jridgewell/sourcemap-codec@1.4.15": {
    "licenses": "MIT",
    "repository": "https://github.com/jridgewell/sourcemap-codec",
    "publisher": "Rich Harris"
  },
  "@kurkle/color@0.3.2": {
    "licenses": "MIT",
    "repository": "https://github.com/kurkle/color",
    "publisher": "Jukka Kurkela"
  },
  "@quasar/extras@1.16.12": {
    "licenses": "MIT",
    "repository": "https://github.com/quasarframework/quasar",
    "publisher": "Razvan Stoenescu"
  },
  "@vue/compiler-core@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/compiler-dom@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/compiler-sfc@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/compiler-ssr@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/reactivity@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/runtime-core@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/runtime-dom@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/server-renderer@3.4.33": {
```


AI 번역본
-For Reference Only-

```
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "@vue/shared@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "chart.js@4.4.2": {
    "licenses": "MIT",
    "repository": "https://github.com/chartjs/Chart.js"
  },
  "chartjs-adapter-moment@1.0.1": {
    "licenses": "MIT",
    "repository": "https://github.com/chartjs/chartjs-adapter-moment"
  },
  "chartjs-plugin-annotation@3.0.1": {
    "licenses": "MIT",
    "repository": "https://github.com/chartjs/chartjs-plugin-annotation",
    "publisher": "Evert Timberg",
    "email": "evert.timberg@gmail.com"
  },
  "cross-env@7.0.3": {
    "licenses": "MIT",
    "repository": "https://github.com/kentcdodds/cross-env",
    "publisher": "Kent C. Dodds",
    "email": "me@kentcdodds.com",
    "url": "https://kentcdodds.com"
  },
  "cross-spawn@7.0.3": {
    "licenses": "MIT",
    "repository": "https://github.com/moxystudio/node-cross-spawn",
    "publisher": "André Cruz",
    "email": "andre@moxy.studio"
  },
  "csstype@3.1.3": {
    "licenses": "MIT",
    "repository": "https://github.com/frenic/csstype",
    "publisher": "Fredrik Nicol",
    "email": "fredrik.nicol@gmail.com"
  },
  "drawflow@0.0.59": {
    "licenses": "MIT",
    "repository": "https://github.com/jerosoler/Drawflow",
    "publisher": "Jero soler"
  },
  "emittery@1.0.1": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/emittery",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "https://sindresorhus.com"
  },
  "entities@4.5.0": {
    "licenses": "BSD-2-Clause",
    "repository": "https://github.com/fb55/entities",
    "publisher": "Felix Boehm",
    "email": "me@feedic.com"
```


AI 번역본
-For Reference Only-

```
  },
  "estree-walker@2.0.2": {
    "licenses": "MIT",
    "repository": "https://github.com/Rich-Harris/estree-walker",
    "publisher": "Rich Harris"
  },
  "hls.js@1.5.8": {
    "licenses": "Apache-2.0",
    "repository": "https://github.com/video-dev/hls.js"
  },
  "isexe@2.0.0": {
    "licenses": "ISC",
    "repository": "https://github.com/isaacs/isexe",
    "publisher": "Isaac Z. Schlueter",
    "email": "i@izs.me",
    "url": "http://blog.izs.me/"
  },
  "magic-string@0.30.10": {
    "licenses": "MIT",
    "repository": "https://github.com/rich-harris/magic-string",
    "publisher": "Rich Harris"
  },
  "maska@2.1.9": {
    "licenses": "MIT",
    "repository": "https://github.com/beholdr/maska",
    "publisher": "Alexander Shabunevich"
  },
  "moment@2.29.4": {
    "licenses": "MIT",
    "repository": "https://github.com/moment/moment",
    "publisher": "Iskren Ivov Chernev",
    "email": "iskren.chernev@gmail.com",
    "url": "https://github.com/ichernev"
  },
  "nanoid@3.3.7": {
    "licenses": "MIT",
    "repository": "https://github.com/ai/nanoid",
    "publisher": "Andrey Sitnik",
    "email": "andrey@sitnik.ru"
  },
  "path-key@3.1.1": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/path-key",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "sindresorhus.com"
  },
  "picocolors@1.0.1": {
    "licenses": "ISC",
    "repository": "https://github.com/alexeyraspopov/picocolors",
    "publisher": "Alexey Raspopov"
  },
  "postcss@8.4.41": {
    "licenses": "MIT",
    "repository": "https://github.com/postcss/postcss",
    "publisher": "Andrey Sitnik",
    "email": "andrey@sitnik.ru"
  },
  "quasar@2.16.6": {
```

AI 번역본
-For Reference Only-

```
    "licenses": "MIT",
    "repository": "https://github.com/quasarframework/quasar",
    "publisher": "Razvan Stoenescu",
    "email": "razvan.stoenescu@gmail.com",
    "url": "https://quasar.dev"
  },
  "shebang-command@2.0.0": {
    "licenses": "MIT",
    "repository": "https://github.com/kevva/shebang-command",
    "publisher": "Kevin Mårtensson",
    "email": "kevinmartensson@gmail.com",
    "url": "github.com/kevva"
  },
  "shebang-regex@3.0.0": {
    "licenses": "MIT",
    "repository": "https://github.com/sindresorhus/shebang-regex",
    "publisher": "Sindre Sorhus",
    "email": "sindresorhus@gmail.com",
    "url": "sindresorhus.com"
  },
  "sortablejs@1.15.0": {
    "licenses": "MIT",
    "repository": "https://github.com/SortableJS/Sortable"
  },
  "source-map-js@1.2.0": {
    "licenses": "BSD-3-Clause",
    "repository": "https://github.com/7rulnik/source-map-js",
    "publisher": "Valentin 7rulnik Semirulnik",
    "email": "v7rulnik@gmail.com"
  },
  "typescript@5.0.4": {
    "licenses": "Apache-2.0",
    "repository": "https://github.com/Microsoft/TypeScript",
    "publisher": "Microsoft Corp."
  },
  "vue-chartjs@5.3.0": {
    "licenses": "MIT",
    "repository": "https://github.com/apertureless/vue-chartjs",
    "publisher": "Jakub Juszczak",
    "email": "jakub@posteo.de"
  },
  "vue@3.4.33": {
    "licenses": "MIT",
    "repository": "https://github.com/vuejs/core",
    "publisher": "Evan You"
  },
  "when-json-met-bigint@0.27.0": {
    "licenses": "MIT",
    "repository": "https://github.com/haoadoresorange/when-json-met-bigint",
    "publisher": "haoadoresorange"
  },
  "which@2.0.2": {
    "licenses": "ISC",
    "repository": "https://github.com/isaacs/node-which",
    "publisher": "Isaac Z. Schlueter",
    "email": "i@izs.me",
    "url": "http://blog.izs.me"
```


AI 번역본
For Reference Only-

```
    }
}
```

AI 번역본
-For Reference Only-

15.7 색인

15.7.1 C

Cloud 에미터(AWS용)

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정 각종 기능
설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
보안 이용 가이드/암호화 통신의 이용
부록/에러 코드 목록/Cloud 에미터(AWS용)

Cloud 에미터(Azure용)

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정 각종 기능
설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/Cloud 에미터(Azure용)
보안 이용 가이드/암호화 통신의 이용
부록/에러 코드 목록/Cloud 에미터(Azure용)

CSV 시리얼라이저

도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정 도입
플로우/컴포넌트의 설정/File 에미터의 설정
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/로직（이벤트）/이벤트 트리거
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/CSV 시리얼라이저

-For Reference Only-

15.7.2 D

DB 쿼리 서비스

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능 설명/플로우
에디터/화면 조작/설정 메뉴/액세스 토큰 발행
각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 개요
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/Web-API 사양
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리 서비스 이용 시의 주의점
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/데이터의 등록
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/DB 쿼리 서비스의 기동 응용적인 이용
방법/Grafana와의 연계/Grafana 데이터 소스의 수동 설정
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항
부록/에러 코드 목록/DB 쿼리 서비스

15.7.3 E

AI 번역본
-For Reference Only-

Edit Mode

도입 플로우/초기 설정/관리자 패스워드의 변경
각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴
각종 기능 설명/플로우 에디터/화면 조작/패널/추가
각종 기능 설명/플로우 에디터/화면 조작/패널/이름의 변경 각종 기능
설명/플로우 에디터/화면 조작/패널/정렬
각종 기능 설명/플로우 에디터/화면 조작/패널/삭제
각종 기능 설명/플로우 에디터/화면 조작/패널/복제
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/추가
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/편집
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/삭제
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/복사&붙여넣기
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/복수 선택
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크
각종 기능 설명/각종 로그/컴포넌트 개별 로그/개별 로그 레벨 설정
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/편집 제한의 제어

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 테스트 실행
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/등록/갱신
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/메일 송신의 설정
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 가동 환경의 셋업

AI 번역본
-For Reference Only-

EtherNet/IP 컬렉터

각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/컬렉터/EtherNet/IP 컬렉터
부록/에러 코드 목록/EtherNet/IP 컬렉터

EtherNet/IP 쓰기

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/액션/EtherNet/IP 쓰기
부록/에러 코드 목록/EtherNet/IP 쓰기

Event Recorder

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정 각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생
i-Trigger（확장 패키지）/관련 컴포넌트/Event Recorder
부록/에러 코드 목록/Event Recorder（i-Trigger 확장 패키지）

15.7.4 F

FFT

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/FFT

File 에미터

SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/컴포넌트의 설정/File 에미터의 설정
도입 플로우/수집 서비스의 실행/동작 확인
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/컴포넌트 공통
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정
각종 기능 설명/에미터/File 에미터
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축부록/에러 코드 목록/File 에미터

FTP/FTPS/SFTP 에미터

각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능
설명/에미터/FTP/FTPS/SFTP 에미터
보안 이용 가이드/암호화 통신의 이용
부록/에러 코드 목록/FTP/FTPS/SFTP 에미터

15.7.5 G

Grafana

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/Grafana 셋업
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/액세스 토큰 발행
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/영속화의 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트/영속화의 설정 각종 기능
설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/Grafana와의 연계/Grafana 연계의 개요
응용적인 이용 방법/Grafana와의 연계/Grafana의 설치
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
응용적인 이용 방법/Grafana와의 연계/그래프의 작성
응용적인 이용 방법/Grafana와의 연계/그래프의 열람
응용적인 이용 방법/Grafana와의 연계/Grafana 데이터 소스의 수동 설정 실행
라이선스/실행 라이선스 이용 환경의 동작 사양

15.7.6 I

AI 번역본

i-Trigger

도입 플로우/설치（Windows）/설치 및 업데이트
각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
i-Trigger（확장 패키지）/설치/설치 및 업데이트
i-Trigger（확장 패키지）/설치/제거
부록/에러 코드 목록/이벤트 데이터 매니저
부록/에러 코드 목록/VIDEO 컬렉터（i-Trigger 확장 패키지）
부록/에러 코드 목록/Event Recorder（i-Trigger 확장 패키지）

15.7.7 J

JSON 시리얼라이저

각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/컴포넌트 공통
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/JSON 시리얼라이저

15.7.8 M

Modbus TCP 컬렉터

각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/컬렉터/Modbus TCP 컬렉터
부록/에러 코드 목록/Modbus TCP 컬렉터

Modbus 쓰기

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/액션/Modbus 쓰기
부록/에러 코드 목록/Modbus 쓰기

AI 번역본
-For Reference Only-

MQTT 컬렉터

각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/컬렉터/MQTT 컬렉터
보안 이용 가이드/암호화 통신의 이용
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 가동 환경의 셋업
부록/에러 코드 목록/MQTT 컬렉터

MQTT 에미터

각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/포인트에 대해
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/MQTT 에미터
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축부록/에러 코드
목록/MQTT 에미터

MQTT 프로토콜

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/MQTT 에미터

15.7.9 P

PLC 컬렉터

각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 설정・전환
각종 기능 설명/컬렉터/PLC 컬렉터
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
트러블슈트/자주 있는 트러블과 그 대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼
부록/에러 코드 목록/PLC 컬렉터

AI 번역본
-For Reference Only-

PLC 쓰기

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능
설명/액션/PLC 쓰기
부록/에러 코드 목록/PLC 쓰기

15.7.10 R

Run Mode

도입 플로우/초기 설정/사용자의 추가
도입 플로우/수집 서비스의 실행/동작 확인
각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정 각종 기능
설명/플로우 에디터/화면 조작/사용자 관리/추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/편집
각종 기능 설명/플로우 에디터/화면 조작/패널/전체 패널
각종 기능 설명/플로우 에디터/화면 조작/패널/패널 스테이터스
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/설정 내용의 확인
각종 기능 설명/플로우 에디터/화면 조작/시스템 리소스
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/파라미터
다이얼로그가 표시되었을 때에 실행되는 함수
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/편집 제한의 제어

실행 라이선스/실행 라이선스 이용 환경의 동작 사양

15.7.11 S

SQL

각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/데이터베이스
각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/영속화의 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트/집약의 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트/영속화의 설정 각종 기능
설명/DB 쿼리（SQL）/DB 쿼리의 개요
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SET
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요

15.7.12 V

VIDEO 컬렉터

각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정
i-Trigger（확장 패키지）/설치/제거
i-Trigger（확장 패키지）/동영상 재생용 뷰어/상시 녹화 재생
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생/재생
i-Trigger（확장 패키지）/동영상 재생용 뷰어/라이브 재생
i-Trigger（확장 패키지）/관련 컴포넌트/VIDEO 컬렉터
i-Trigger（확장 패키지）/관련 컴포넌트/Event Recorder
부록/에러 코드 목록/VIDEO 컬렉터（i-Trigger 확장 패키지）

15.7.13 W

WEBUI

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정
각종 기능 설명/각종 로그/로그의 종별
각종 기능 설명/각종 로그/다운로드
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/실행

WebAPI

각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI의 개요
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI의 실행 방법
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양

-For Reference Only-

15.7.14 액

액션

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 항목/Edit Mode

액세스 토큰

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/액세스 토큰 발행 각종 기능 설명/시스템
컴포넌트/DB 쿼리 서비스
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리 서비스 이용 시의 주의점
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/공통 사항
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/전 컴포넌트의 실행 상태 취득 응용적인 이용
방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/개별 컴포넌트의 실행 상태 취득 응용적인 이용 방법/제어
프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 개시
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 정지
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/데이터의 등록
응용적인 이용 방법/Grafana와의 연계/Grafana 데이터 소스의 수동 설정
보안 이용 가이드/토큰의 엄중 관리
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항

HTTP/HTTPS 통신

도입 플로우/설치（Linux）/설치 및 업데이트
도입 플로우/설치（Windows）/설치 및 업데이트 각종 기능 설명/플로우 에디터/화면
조작/설정 메뉴/HTTP/HTTPS 통신 보안 이용 가이드/HTTPS의 이용

부록/에러 코드 목록/DB 쿼리 서비스

이동 평균

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/이동 평균
각종 기능 설명/로직（실시간 분석/데이터 조작）/데이터 변환

이벤트 데이터

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/로직 (이벤트) /이벤트 데이터
각종 기능 설명/로직 (이벤트) /이벤트 트리거
각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
응용적인 이용 방법/글로벌 변수/설정 예/이벤트 데이터의 조건 판정에 사용하는 임계값을 변화시킨다
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/메일 송신의 설정

i-Trigger (확장 패키지) /관련 컴포넌트/Event Recorder/설정 항목
i-Trigger (확장 패키지) /관련 컴포넌트/이벤트 데이터 매니저
부록/에러 코드 목록/이벤트 데이터 매니저

이벤트 트리거

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/로직 (이벤트) /이벤트 트리거
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트/이벤트의 설정
i-Trigger (확장 패키지) /관련 컴포넌트/Event Recorder
i-Trigger (확장 패키지) /관련 컴포넌트/이벤트 트리거
부록/에러 코드 목록/이벤트 트리거

AI 번역본
-For Reference Only-

영속화

각종 기능 설명/엔진 기능의 전체상/기능 목록/데이터베이스
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능
설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트
각종 기능 설명/연산식・조건식/연산식과 조건식（공통）
응용적인 이용 방법/글로벌 변수/글로벌 변수의 구문
부록/에러 코드 목록/코어 인터널
부록/커스텀 컴포넌트 Python-API 레퍼런스/InColumn Objects/get_latest_value
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPort ... 시스템 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPort ... 출력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse

AI 번역본
-For Reference Only-

에미터

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/컴포넌트의 설정/File 에미터의 설정
도입 플로우/수집 서비스의 실행/동작 확인
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/플로우
에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/컴포넌트 공통
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/포인트에 대해
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/File 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/개발용 컴포넌트/더미 에미터
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요
보안 이용 가이드/암호화 통신의 이용
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축
부록/에러 코드 목록/MQTT 에미터
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/FTP/FTPS/SFTP 에미터
부록/커스텀 컴포넌트 Python-API 레퍼런스/FileTypeMetaInfo Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_save_running_info() ... 실행 시의 정보를 저장

에러 매니저

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/시스템 컴포넌트/에러 매니저
부록/에러 코드 목록/에러 매니저

AI 번역본
-For Reference Only-

연산 컬렉터

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/연산식 · 조건식/연산식 · 조건식의 개요
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/연산 컬렉터

연산식

각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/연산식 · 조건식/연산식 · 조건식의 개요
각종 기능 설명/연산식 · 조건식/연산식과 조건식 (공통)
각종 기능 설명/연산식 · 조건식/연산식
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요부록/에러
코드 목록/연산 컬렉터

15.7.15 커

AI 번역본
-For Reference Only-

커스텀 컴포넌트

AI 번역본
-For Reference Only-

머리말/대상 독자
SpeeDBee Synapse의 개요/운용까지의 흐름
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/커스텀(Python)
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/커스텀(C)
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/로직（실시간 분석/데이터 조작）/FFT
각종 기능 설명/스테이터스 타임라인/스테이터스 목록
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
각종 기능 설명/설정 정보의 관리(프로젝트)/임포트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/커스텀 컴포넌트란
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트 베이스와 컴포넌트 인스턴스
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트 인스턴스의 라이프사이클
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트의 입력 포트와 출력 포트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/전제 지식
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트용 API
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트용 API
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 JSON 파일의 작성/작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/HTML 형식의 상세
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/파라미터 다이얼로그가 표시되었을 때에 실행되는 함수
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/파라미터 다이얼로그의 저장 시에 실행되는 함수

AI 번역본
-For Reference Only-

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Python 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:python}

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/C언어 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:C}

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트 관리 디렉터리의 초기화
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 추가
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 테스트 실행
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/패키징

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제
후방 호환에 관하여/후방 호환의 대상
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 설정의 출력
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects/create_lock
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_lock() ... 컴포넌트 간 배타 제어
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onLoad ... 커스텀 UI 불러오기 훅
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onSave ... 파라미터 저장 훅
부록/커스텀 UI JS-API 레퍼런스/형/ComponentMethodResponse

기본 통계

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/기본 통계
부록/커스텀 UI JS-API 레퍼런스/형/InportResponse

그래프

AI 번역본
-For Reference Only-

SpeeDBee Synapse의 개요/운용까지의 흐름
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/데이터 모니터/분포
응용적인 이용 방법/Grafana와의 연계/Grafana 연계의 개요
응용적인 이용 방법/Grafana와의 연계/그래프의 작성
응용적인 이용 방법/Grafana와의 연계/그래프의 열람
응용적인 이용 방법/Grafana와의 연계/그래프의 데이터 건수 상한의 변경
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생/재생

글로벌 변수

각종 기능 설명/컬렉터/EtherNet/IP 컬렉터
각종 기능 설명/액션/EtherNet/IP 쓰기
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SET
각종 기능 설명/연산식・조건식/연산식과 조건식（공통）
각종 기능 설명/연산식・조건식/연산식
각종 기능 설명/연산식・조건식/조건식
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 정보 정의
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요
응용적인 이용 방법/글로벌 변수/글로벌 변수의 구문
응용적인 이용 방법/글로벌 변수/설정 예

코어

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/엔진 기능의 전체상/기능 목록/데이터베이스
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/정상 정지 시
각종 기능 설명/엔진 기능의 전체상/동작 사양/이상 정지 시(전원 차단 등)
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정 각종 기능
설명/컬렉터/리소스 컬렉터
각종 기능 설명/시스템 컴포넌트/코어 인터널
각종 기능 설명/각종 로그/로그의 종별
각종 기능 설명/각종 로그/다운로드
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/실행

응용적인 이용 방법/Grafana와의 연계/그래프의 작성/DB 쿼리 서비스의 기동
응용적인 이용 방법/글로벌 변수/글로벌 변수의 구문
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법

부록/에러 코드 목록/코어 인터널
부록/에러 코드 목록/에러 매니저

코어 인터널

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/이상 정지 시(전원 차단 등)
각종 기능 설명/시스템 컴포넌트/코어 인터널
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법

부록/에러 코드 목록/코어 인터널

AI 번역본
-For Reference Only-

컬렉터

AI 번역본
-For Reference Only-

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정
도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정
도입 플로우/수집 서비스의 실행/동작 확인
도입 플로우/수집 서비스의 실행/설정 튜닝
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 설정・전환
각종 기능 설명/컬렉터/리소스 컬렉터
각종 기능 설명/컬렉터/PLC 컬렉터
각종 기능 설명/컬렉터/Modbus TCP 컬렉터
각종 기능 설명/컬렉터/EtherNet/IP 컬렉터
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/개발용 컴포넌트/더미 컬렉터
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/수치 변환의 설정 각종 기능
설명/컴포넌트 공통의 설정 항목/출력 포트/등록 조건의 설정 각종 기능 설명/DB
쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/INSERT
각종 기능 설명/연산식・조건식/연산식・조건식의 개요
각종 기능 설명/연산식・조건식/연산식과 조건식（공통）
각종 기능 설명/연산식・조건식/연산식
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요트러블슈트/자주 있는 트러블과 그
대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

보안 이용 가이드/암호화 통신의 이용
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼
i-Trigger（확장 패키지）/설치/제거
i-Trigger（확장 패키지）/동영상 재생용 뷰어/상시 녹화 재생
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생/재생
i-Trigger（확장 패키지）/동영상 재생용 뷰어/라이브 재생
i-Trigger（확장 패키지）/관련 컴포넌트/VIDEO 컬렉터
i-Trigger（확장 패키지）/관련 컴포넌트/Event Recorder
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/리소스 컬렉터
부록/에러 코드 목록/PLC 컬렉터

AI 번역본
For Reference Only-

부록/에러 코드 목록/Modbus TCP 컬렉터
부록/에러 코드 목록/EtherNet/IP 컬렉터
부록/에러 코드 목록/연산 컬렉터
부록/에러 코드 목록/MQTT 컬렉터
부록/에러 코드 목록/VIDEO 컬렉터（i-Trigger 확장 패키지）

AI 번역본
-For Reference Only-

컴포넌트

AI 번역본
-For Reference Only-

머리말/대상 독자
SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/초기 설정/사용자의 추가
도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정 도입 플로우/컴포넌트의
설정/CSV 시리얼라이저의 설정 도입 플로우/컴포넌트의 설정/File
에미터의 설정
도입 플로우/수집 서비스의 실행/수집 서비스의 개시/정지
도입 플로우/수집 서비스의 실행/설정 튜닝
도입 플로우/수집 서비스의 실행/수집 서비스의 종료
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/플로우
에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/전반적인 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/컴포넌트 공통
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/데이터 디렉터리 설정 각종 기능 설명/플로우
에디터/화면 조작/설정 메뉴/커스텀(Python)
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/커스텀(C)
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/포인트에 대해
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/편집
각종 기능 설명/플로우 에디터/화면 조작/패널/컴포넌트 검색
각종 기능 설명/플로우 에디터/화면 조작/패널/표시 배율의 조정
각종 기능 설명/플로우 에디터/화면 조작/패널/전체 패널
각종 기능 설명/플로우 에디터/화면 조작/패널/패널 스테이터스
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크/추가
각종 기능 설명/플로우 에디터/화면 조작/데이터 모니터
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 개요
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 설정 · 전환
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 동작
각종 기능 설명/컬렉터/리소스 컬렉터
각종 기능 설명/컬렉터/PLC 컬렉터
각종 기능 설명/컬렉터/Modbus TCP 컬렉터
각종 기능 설명/컬렉터/EtherNet/IP 컬렉터
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저 각종
기능 설명/시리얼라이저/CSV 시리얼라이저 각종 기능
설명/에미터/MQTT 에미터

AI 번역본
-For Reference Only-

각종 기능 설명/에미터/File 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/기본 통계 각종 기능
설명/로직（실시간 분석/데이터 조작）/FFT
각종 기능 설명/로직（실시간 분석/데이터 조작）/이동 평균
각종 기능 설명/로직（실시간 분석/데이터 조작）/데이터 변환 각종 기능
설명/액션/셸 커맨드
각종 기능 설명/액션/메일 송신
각종 기능 설명/액션/PLC 쓰기
각종 기능 설명/액션/Modbus 쓰기
각종 기능 설명/액션/EtherNet/IP 쓰기
각종 기능 설명/개발용 컴포넌트/더미 컬렉터
각종 기능 설명/개발용 컴포넌트/더미 에미터
각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
각종 기능 설명/시스템 컴포넌트/코어 인터널
각종 기능 설명/시스템 컴포넌트/에러 매니저
각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트
각종 기능 설명/스테이터스 타임라인/스테이터스 목록
각종 기능 설명/스테이터스 타임라인/스테이터스 타임라인
각종 기능 설명/각종 로그/로그의 종별
각종 기능 설명/각종 로그/컴포넌트 개별 로그
각종 기능 설명/각종 로그/다운로드
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/INSERT
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
각종 기능 설명/설정 정보의 관리(프로젝트)/임포트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/커스텀 컴포넌트란

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트 베이스와 컴포넌트 인스턴스

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트 인스턴스의 라이프사이클

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트의 입력 포트와 출력 포트

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/전제 지식
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트용 API

AI 번역본
For Reference Only

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트용 API

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/커스텀 UI의 작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 JSON 파일의 작성/작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/HTML 형식의 상세
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/파라미터 다이얼로그가 표시되었을 때에 실행되는 함수
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/파라미터 다이얼로그의 저장 시에 실행되는 함수
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/편집 제한의 제어
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/Python 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:python}

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 UI 작성/C언어 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:C}

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트 관리 디렉터리의 초기화
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 추가
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 테스트 실행
응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/패키징

응용적인 이용 방법/커스텀 컴포넌트의 작성 · 이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/공통 사항
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/전 컴포넌트의 실행 상태 취득
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/개별 컴포넌트의 실행 상태 취득
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 개시
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 정지
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/데이터의 등록
응용적인 이용 방법/Grafana와의 연계/그래프의 열람
응용적인 이용 방법/Grafana와의 연계/Grafana 데이터 소스의 수동 설정
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요

AI 번역본
For Reference Only

트러블슈트/이상 발생 시 공통 순서
트러블슈트/자주 있는 트러블과 그 대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법
트러블슈트/이상이 발생한 경우에 검지하는 방법
보안 이용 가이드/HTTPS의 이용
보안 이용 가이드/암호화 통신의 이용
후방 호환에 관하여/후방 호환의 대상
실행 라이선스/라이선스의 종류
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법/운용까지의 흐름
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 설정의 출력
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 가동 환경의 셋업 실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/동작 확인
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생/재생
i-Trigger（확장 패키지）/관련 컴포넌트/VIDEO 컬렉터
i-Trigger（확장 패키지）/관련 컴포넌트/Event Recorder
i-Trigger（확장 패키지）/관련 컴포넌트/이벤트 트리거
i-Trigger（확장 패키지）/관련 컴포넌트/이벤트 데이터 매니저
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/PLC 컬렉터
부록/에러 코드 목록/Modbus TCP 컬렉터
부록/에러 코드 목록/EtherNet/IP 컬렉터
부록/에러 코드 목록/연산 컬렉터
부록/에러 코드 목록/MQTT 컬렉터
부록/에러 코드 목록/MQTT 에미터
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/FTP/FTPS/SFTP 에미터
부록/에러 코드 목록/JSON 시리얼라이저
부록/에러 코드 목록/CSV 시리얼라이저
부록/에러 코드 목록/셸 커맨드
부록/에러 코드 목록/PLC 쓰기
부록/에러 코드 목록/Modbus 쓰기
부록/에러 코드 목록/EtherNet/IP 쓰기
부록/에러 코드 목록/메일 송신
부록/에러 코드 목록/코어 인터널
부록/에러 코드 목록/DB 쿼리 서비스
부록/에러 코드 목록/에러 매니저
부록/에러 코드 목록/Event Recorder（i-Trigger 확장 패키지）
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/DataType Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/OutColumn Objects/insert_file_ref
부록/커스텀 컴포넌트 Python-API 레퍼런스/InColumn Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveOutPort Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/ErrorInfoBase Objects

AI 번역본
-For Reference Only-

부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveCompError Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveCompStatus Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveInPortEventSetting Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/SynapseLock Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveLockCanceled Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/출력 포트 함수/hive_outport_create_column() ... 신규 컬럼 작성
부록/커스텀 컴포넌트 C-API 레퍼런스/출력 포트 함수/hive_outport_register_confirmed_time() ... 스테이터스 정보에의 컬럼 데이터 확정 시각의 등록
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_get_column_list() ... 입력 컬럼 정보 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_continuous_reader() ... 컬럼 데이터 계속 검색용의 Reader 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_continuous_reader_release() ... 컬럼 데이터 계속 검색용의 Reader의 해제
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_timerange_reader() ... 컬럼 데이터 시간 범위 검색용의 Reader 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_timerange_reader_release() ... 컬럼 데이터 시간 범위 검색용의 Reader의 해제
부록/커스텀 컴포넌트 C-API 레퍼런스/컴포넌트 스테이터스 함수
부록/커스텀 컴포넌트 C-API 레퍼런스/로그 관련 함수/HIVE_LOG_XXXX() ... 로그 출력 매크로
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_interval_call() ... 정기 처리 함수
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_save_running_info() ... 실행 시의 정보를 저장

부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_load_running_info() ... 저장한 실행 시의 정보를 읽어들인다
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_runnable() ... 컴포넌트 계속 가부

부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_get_api_error() ... 발생한 에러의 정보를 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_get_common_variable() ... 컴포넌트 공통 변수 취득

부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_lock() ... 컴포넌트 간 배타 제어
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_unlock() ... 컴포넌트 간 배타 제어
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_COMPONENT ... 컴포넌트 인스턴스

부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/hive_interval_proc_t ... 정기 처리 함수 시그니처
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onLoad ... 커스텀 UI 불러오기 훅
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onSave ... 파라미터 저장 훅
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getComponentbases ... 컴포넌트 베이스 목록 취득

부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getComponents ... 컴포넌트 목록 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getComponent ... 컴포넌트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPort ... 입력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPortData ... 입력 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPortDataValue ... 입력 포트 데이터 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPort ... 시스템 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPortData ... 시스템 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPortDataValue ... 시스템 포트 데이터 취득

AI 번역본
For Reference Only

부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPort ... 출력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPortData ... 출력 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPortDataValue ... 출력 포트 데이터 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/runMethod ... 컴포넌트 인스턴스 메서드 실행

부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getFlowlinks ... 플로우 링크 목록 취득
부록/커스텀 UI JS-API 레퍼런스/형/ComponentbaseResponse
부록/커스텀 UI JS-API 레퍼런스/형/PanelResponse
부록/커스텀 UI JS-API 레퍼런스/형/ComponentResponse
부록/커스텀 UI JS-API 레퍼런스/형/InportResponse
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataValueResponse
부록/커스텀 UI JS-API 레퍼런스/형/ComponentMethodResponse
부록/커스텀 UI JS-API 레퍼런스/형/FlowlinkResponse
부록/커스텀 UI JS-API 레퍼런스/형/FlowlinkFiltersResponse

컴포넌트 베이스

각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트 베이스와 컴포넌트 인스턴스

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/실행

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 클래스 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/ HiveComponent 클래스의 라이프사이클
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/컴포넌트 파라미터

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 정보 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 테스트 실행
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects/create_lock
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_lock() ... 컴포넌트 간 배타 제어
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_unlock() ... 컴포넌트 간 배타 제어
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getComponentbases ... 컴포넌트 베이스 목록 취득

부록/커스텀 UI JS-API 레퍼런스/형/ComponentbaseResponse
부록/커스텀 UI JS-API 레퍼런스/형/ComponentResponse

AI 번역본
-For Reference Only-

15.7.16 셸

셸 커맨드

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/액션/셸 커맨드
부록/에러 코드 목록/셸 커맨드

시계열 데이터베이스

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/데이터베이스
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/영속화의 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트/영속화의 설정 각종 기능 설명/DB
쿼리（SQL）/DB 쿼리의 개요
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SHOW
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/INSERT
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법

부록/에러 코드 목록/코어 인터널

시스템 컴포넌트

각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트 각종 기능
설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode

시스템 패널

응용적인 이용 방법/Grafana와의 연계/그래프의 작성/DB 쿼리 서비스의 기동
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/메일 송신의 설정

시스템 포트

각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크/추가
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법

부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPort ... 시스템 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPortData ... 시스템 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPortDataValue ... 시스템 포트 데이터 취득

시뮬레이션 모드

각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드

수집 서비스

도입 플로우/수집 서비스의 실행/수집 서비스의 개시/정지
도입 플로우/수집 서비스의 실행/설정 튜닝
도입 플로우/수집 서비스의 실행/수집 서비스의 종료

집약

각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요부록/커스텀 컴포넌트 Python-API 레퍼런스/AggregationType Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveRecordType Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveRecord Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveRecordData Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_COLUMN_READ_RESULT ... 컬럼 정보 읽어들이기 결과（윈도우 데이터）
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_RECORD_DATA ... 레코드 데이터
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPort ... 입력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/형/InportResponse

AI 번역본
-For Reference Only-

출력 포트

AI 번역본
-For Reference Only-

도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정
도입 플로우/컴포넌트의 설정/File 에미터의 설정
도입 플로우/수집 서비스의 실행/동작 확인
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크/추가
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/기본 통계 각종 기능
설명/로직（실시간 분석/데이터 조작）/FFT
각종 기능 설명/로직（실시간 분석/데이터 조작）/이동 평균
각종 기능 설명/로직（실시간 분석/데이터 조작）/데이터 변환 각종 기능
설명/액션/PLC 쓰기
각종 기능 설명/액션/Modbus 쓰기
각종 기능 설명/액션/EtherNet/IP 쓰기
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/연산식・조건식/연산식・조건식의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트의 입력 포트와 출력 포트

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/샘플의 내용

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/실행

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 클래스 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/출력 포트의 컬럼 작성/데이터 등록(sin_and_random)
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/샘플의 내용

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/실행

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 정보 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/출력 포트의 컬럼 작성/데이터 등록（count_and_random ）
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/ColumnOption Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/OutColumn Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveOutPort Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/출력 포트 함수
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/hive_primitive_type_t ... HIVE 컬럼 프리미티브 형
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_DATA_TYPE ... HIVE 컬럼 데이터형
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPort ... 출력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPortData ... 출력 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPortDataValue ... 출력 포트 데이터 취득

AI 번역본
-For Reference Only-

부록/커스텀 UI JS-API 레퍼런스/형/ComponentbaseResponse
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataValueResponse

조건식

각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/등록 조건의 설정 각종 기능
설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
각종 기능 설명/연산식・조건식/연산식・조건식의 개요
각종 기능 설명/연산식・조건식/연산식과 조건식（공통）
각종 기능 설명/연산식・조건식/연산식
각종 기능 설명/연산식・조건식/조건식
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요
응용적인 이용 방법/글로벌 변수/글로벌 변수의 구문
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/이벤트 트리거
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse

AI 번역본
-For Reference Only-

시리얼라이저

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정
도입 플로우/컴포넌트의 설정/File 에미터의 설정
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/컴포넌트 공통
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/File 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/로직（이벤트）/이벤트 트리거
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 개요
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/JSON 시리얼라이저
부록/에러 코드 목록/CSV 시리얼라이저
부록/에러 코드 목록/코어 인터널

수치 변환

각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/연산식・조건식/연산식・조건식의 개요
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요
응용적인 이용 방법/글로벌 변수/글로벌 변수의 구문
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPort ... 시스템 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPort ... 출력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse

AI 번역본
-For Reference Only-

스테이터스 타임라인

각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/이상 정지 시(전원 차단 등)
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/스테이터스 타임라인/스테이터스 타임라인
트러블슈트/이상 발생 시 공통 순서
트러블슈트/자주 있는 트러블과 그 대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법 실행
라이선스/실행 라이선스 이용 환경의 동작 사양

AI 번역본
-For Reference Only-

설정 메뉴

AI 번역본
-For Reference Only-

도입 플로우/초기 설정/관리자 패스워드의 변경
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴
각종 기능 설명/설정 정보의 관리(프로젝트)/신규 작성
각종 기능 설명/설정 정보의 관리(프로젝트)/복제
각종 기능 설명/설정 정보의 관리(프로젝트)/임포트
각종 기능 설명/설정 정보의 관리(프로젝트)/익스포트
각종 기능 설명/설정 정보의 관리(프로젝트)/전환
각종 기능 설명/설정 정보의 관리(프로젝트)/정보 갱신
각종 기능 설명/설정 정보의 관리(프로젝트)/삭제
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/등록

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/삭제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/표시

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/다운로드

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/보호의 부여

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/보호의 해제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/외부 라이브러리의 추가

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/외부 라이브러리의 삭제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/등록

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/삭제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/다운로드

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/보호의 부여

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/보호의 해제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Python 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:python}/등록

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Python 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:python}/삭제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/C언어 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:C}/등록

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/C언어 커스텀 컴포넌트에의 커스텀 UI(JSON)의 적용 {:C}/삭제

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의
SCCPKG 파일의 등록, 및, 삭제/등록/갱신

AI 번역본
-For Reference Only-

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/다운로드
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/삭제
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/보호의 부여
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/보호의 해제
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/데이터의 등록
트러블슈트/자주 있는 트러블과 그 대처 방법/시각 설정에 관한 문제와 대처 방법/SpeeDBee Synapse가 Windows에서 동작하고 있는 경우
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 설정의 출력
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 가동 환경의 셋업
부록/에러 코드 목록/DB 쿼리 서비스

전체 패널

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정
각종 기능 설명/플로우 에디터/화면 조작/패널/전체 패널

조작 로그

각종 기능 설명/플로우 에디터/화면 항목/Edit Mode 각종
기능 설명/플로우 에디터/화면 항목/Run Mode 각종 기능
설명/각종 로그/로그의 종별
각종 기능 설명/각종 로그/조작 로그
각종 기능 설명/각종 로그/다운로드
보안 이용 가이드/조작 로그의 확인

-For Reference Only-

15.7.17 데

데이터 수집 플로우

각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/기동/정지
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/편집 제한의 제어

데이터 변환

각종 기능 설명/로직（실시간 분석/데이터 조작）/데이터 변환

등록 조건

각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/연산식・조건식/연산식・조건식의 개요
응용적인 이용 방법/글로벌 변수/글로벌 변수의 개요
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getSystemPort … 시스템 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getOutPort … 출력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/형/OutportResponse

15.7.18 입

AI 번역본
-For Reference Only-

입력 포트

AI 번역본
-For Reference Only-

도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정
도입 플로우/컴포넌트의 설정/File 에미터의 설정
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능
설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/추가
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크/추가
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/File 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/로직（실시간 분석/데이터 조작）/기본 통계 각종 기능
설명/로직（실시간 분석/데이터 조작）/FFT
각종 기능 설명/로직（실시간 분석/데이터 조작）/이동 평균
각종 기능 설명/로직（실시간 분석/데이터 조작）/데이터 변환 각종 기능
설명/액션/셸 커맨드
각종 기능 설명/액션/메일 송신
각종 기능 설명/액션/PLC 쓰기
각종 기능 설명/액션/Modbus 쓰기
각종 기능 설명/액션/EtherNet/IP 쓰기
각종 기능 설명/개발용 컴포넌트/더미 에미터
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/컴포넌트의 입력 포트와 출력 포트

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/샘플의 내용

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 클래스 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/입력 포트로부터의 데이터 불러오기（read_and_log）
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/샘플 커스텀 컴포넌트/샘플의 내용

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/커스텀 컴포넌트 정보 정의
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/입력 포트로부터의 데이터 불러오기（read_and_log ）
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

AI 번역본
For Reference Only

트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
i-Trigger（확장 패키지）/관련 컴포넌트/Event Recorder
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/MQTT 에미터
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/PLC 쓰기
부록/에러 코드 목록/Modbus 쓰기
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/AggregationType Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/AggregationTypeSet Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/InColumn Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/InPortReader Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveInPort Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveWindowData Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveRecord Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveInPortEventSetting Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_get_column_list() ... 입력 컬럼 정보 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_continuous_reader() ... 컬럼 데이터 계속 검색용의 Reader 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_continuous_reader_read() ... 컬럼 데이터 계속 검색

부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_inport_timerange_reader() ... 컬럼 데이터 시간 범위 검색용의 Reader 취득
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_timerange_reader_read() ... 컬럼 데이터 시간 범검색
부록/커스텀 컴포넌트 C-API 레퍼런스/입력 컬럼 함수/hive_incolumn_get_latest_value() ... 컬럼 데이터 최신값의 취득

부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/hive_primitive_type_t ... HIVE 컬럼 프리미티브 형
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_DATA_TYPE ... HIVE 컬럼 데이터형
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_COLUMN_READ_RESULT ... 컬럼 정보 읽어들이기 결과（윈도우 데이터）
부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_RECORD ... 컬럼 데이터 레코드
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPort ... 입력 포트 설정 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPortData ... 입력 포트 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getInPortDataValue ... 입력 포트 데이터 취득
부록/커스텀 UI JS-API 레퍼런스/형/ComponentbaseResponse
부록/커스텀 UI JS-API 레퍼런스/형/InportResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataResponse
부록/커스텀 UI JS-API 레퍼런스/형/PortDataValueResponse

AI 번역본
For Reference Only

네트워크

도입 플로우/설치（Linux）/설치 및 업데이트
도입 플로우/설치（Windows）/설치 및 업데이트
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/컬렉터/리소스 컬렉터
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
트러블슈트/자주 있는 트러블과 그 대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법/IP 주소의 확인
트러블슈트/자주 있는 트러블과 그 대처 방법/PLC 컬렉터에 접속할 수 없는 경우의 대처 방법/포트의 확인
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼
i-Trigger（확장 패키지）/관련 컴포넌트/VIDEO 컬렉터
부록/에러 코드 목록/컴포넌트 공통
부록/에러 코드 목록/리소스 컬렉터
부록/에러 코드 목록/MQTT 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/에러 코드 목록/FTP/FTPS/SFTP 에미터
부록/에러 코드 목록/VIDEO 컬렉터（i-Trigger 확장 패키지）

15.7.19 패

AI 번역본
-For Reference Only-

패널

AI 번역본
-For Reference Only-

도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정 도입
플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정 도입
플로우/컴포넌트의 설정/File 에미터의 설정
도입 플로우/수집 서비스의 실행/수집 서비스의 개시/정지
도입 플로우/수집 서비스의 실행/동작 확인
도입 플로우/수집 서비스의 실행/설정 튜닝
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/서버 설정 각종 기능
설명/플로우 에디터/화면 조작/사용자 관리/추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/편집
각종 기능 설명/플로우 에디터/화면 조작/패널
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/추가
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/복사&붙여넣기
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/복수 선택
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/기동/정지
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/공통 사항
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/전 컴포넌트의 실행 상태 취득 응용적인 이용
방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/개별 컴포넌트의 실행 상태 취득
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 개시
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI 사양/컴포넌트 일괄 실행 정지
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/DB 쿼리 서비스의 기동
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/메일 송신의 설정
후방 호환에 관하여/후방 호환의 대상
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/실행 라이선스 설정의 출력
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getPanels ... 패널 목록 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getPanel ... 패널 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getComponents ... 컴포넌트 목록 취득
부록/커스텀 UI JS-API 레퍼런스/형/PanelResponse

AI 번역본
For Reference Only

뷰어

각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
i-Trigger (확장 패키지) /동영상 재생용 뷰어/뷰어에의 액세스
i-Trigger (확장 패키지) /관련 컴포넌트/VIDEO 컬렉터
i-Trigger (확장 패키지) /관련 컴포넌트/Event Recorder

필터

각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해 각종 기능 설명/플로우 에디터/화면 조작/플로우 링크/필터
각종 기능 설명/컴포넌트 공통의 설정 항목/필터
각종 기능 설명/컴포넌트 공통의 설정 항목/플로우 링크 아이콘
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
부록/에러 코드 목록/MQTT 에미터
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getFlowlinkFilter ... 플로우 링크 필터 정보 취득

부록/커스텀 UI JS-API 레퍼런스/형/InportResponse
부록/커스텀 UI JS-API 레퍼런스/형/FlowlinkFiltersResponse

플로우 에디터

각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 항목
각종 기능 설명/플로우 에디터/화면 조작
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트의 테스트 실행
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축

AI 번역본
-For Reference Only-

플로우 링크

도입 플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정
도입 플로우/컴포넌트의 설정/File 에미터의 설정
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/데이터 수집에 관련된 동작 사양에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/플로우 링크
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/시뮬레이션 모드의 동작 각종 기능
설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트
각종 기능 설명/컴포넌트 공통의 설정 항목/필터
각종 기능 설명/컴포넌트 공통의 설정 항목/플로우 링크 아이콘
각종 기능 설명/컴포넌트 공통의 설정 항목/입력 포트
각종 기능 설명/컴포넌트 공통의 설정 항목/시스템 포트
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축부록/에러 코드
목록/MQTT 에미터
부록/에러 코드 목록/File 에미터
부록/에러 코드 목록/Cloud 에미터(AWS용)
부록/에러 코드 목록/Cloud 에미터(Azure용)
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveOutPort Objects/register_confirmed_time
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getFlowlinks ... 플로우 링크 목록 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getFlowlink ... 플로우 링크 정보 취득
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수/getFlowlinkFilter ... 플로우 링크 필터 정보 취득

부록/커스텀 UI JS-API 레퍼런스/형/FlowlinkResponse
부록/커스텀 UI JS-API 레퍼런스/형/FlowlinkFiltersResponse

프로필

도입 플로우/초기 설정/사용자의 추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/프로필

헤더

도입 플로우/초기 설정/관리자 패스워드의 변경
도입 플로우/초기 설정/프로젝트명의 설정
도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정
도입 플로우/수집 서비스의 실행/수집 서비스의 개시/정지/SpeeDBee Synapse상의 전 컴포넌트의 기동/정지
도입 플로우/수집 서비스의 실행/동작 확인
도입 플로우/수집 서비스의 실행/설정 튜닝
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/기동/정지
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드/운전 파일의 포맷
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/각종 로그/조작 로그/열람
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/인클루드 헤더

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의
SCCPKG 파일의 등록, 및, 삭제/등록/갱신
응용적인 이용 방법/제어 프로그램과의 연계（WebAPI）/WebAPI의 실행 방법
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
응용적인 이용 방법/Grafana와의 연계/그래프의 열람
응용적인 이용 방법/Grafana와의 연계/Grafana 데이터 소스의 수동 설정
i-Trigger（확장 패키지）/동영상 재생용 뷰어/뷰어에의 액세스
i-Trigger（확장 패키지）/동영상 재생용 뷰어/이벤트 녹화 재생/삭제

AI 번역본
-For Reference Only-

포인트

각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/소프트웨어 정보 각종 기능
설명/플로우 에디터/화면 조작/설정 메뉴/포인트에 대해 각종 기능 설명/컬렉터/리소스 컬렉터
각종 기능 설명/컬렉터/PLC 컬렉터
각종 기능 설명/컬렉터/Modbus TCP 컬렉터
각종 기능 설명/컬렉터/EtherNet/IP 컬렉터
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/컬렉터/연산 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/File 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/로직 (이벤트) /이벤트 데이터
각종 기능 설명/로직 (이벤트) /이벤트 트리거
각종 기능 설명/로직 (실시간 분석/데이터 조작) /기본 통계 각종 기능
설명/로직 (실시간 분석/데이터 조작) /FFT
각종 기능 설명/로직 (실시간 분석/데이터 조작) /이동 평균
각종 기능 설명/로직 (실시간 분석/데이터 조작) /데이터 변환 각종 기능
설명/액션/셸 커맨드
각종 기능 설명/액션/메일 송신
각종 기능 설명/액션/PLC 쓰기
각종 기능 설명/액션/Modbus 쓰기
각종 기능 설명/액션/EtherNet/IP 쓰기
각종 기능 설명/개발용 컴포넌트/더미 컬렉터/기본 정보 각종 기능
설명/개발용 컴포넌트/더미 에미터/기본 정보
각종 기능 설명/시스템 컴포넌트/DB 쿼리 서비스
각종 기능 설명/시스템 컴포넌트/코어 인터널
각종 기능 설명/시스템 컴포넌트/에러 매니저
각종 기능 설명/시스템 컴포넌트/이벤트 데이터 매니저
응용적인 이용 방법/제어 프로그램과의 연계 (WebAPI) /WebAPI 사양/컴포넌트 일괄 실행 개시
i-Trigger (확장 패키지) /관련 컴포넌트/VIDEO 컬렉터/기본 정보
i-Trigger (확장 패키지) /관련 컴포넌트/Event Recorder/기본 정보
부록/에러 코드 목록/리소스 컬렉터
부록/커스텀 UI JS-API 레퍼런스/형/ComponentbaseResponse

AI 번역본
-For Reference Only-

15.7.20 메

메일 송신

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해
각종 기능 설명/액션/메일 송신
트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/특정 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정

트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/이벤트 데이터의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/Synapse 내의 어느 한 컴포넌트에서 이상이 발생한 경우에 검지하는 방법/메일 송신의 설정
트러블슈트/이상이 발생한 경우에 검지하는 방법/전원 차단 등에 의해 시계열 데이터베이스의 데이터 결손을 검지하는 방법/메일 송신의 설정
보안 이용 가이드/암호화 통신의 이용
부록/에러 코드 목록/메일 송신

모드

도입 플로우/초기 설정/관리자 패스워드의 변경
도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정 도입
플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정 도입
플로우/컴포넌트의 설정/File 에미터의 설정
도입 플로우/수집 서비스의 실행/동작 확인
도입 플로우/수집 서비스의 실행/설정 튜닝
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/엔진 기능의 전체상/동작 사양/네트워크 이상, 복구 시의 동작 사양에 대해 각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/시뮬레이션 모드
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/Cloud 에미터(AWS용)
각종 기능 설명/에미터/Cloud 에미터(Azure용)
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/Synapse에의 SCCPKG 파일의 등록, 및, 삭제/등록/갱신
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
응용적인 이용 방법/Grafana와의 연계/그래프의 작성/데이터의 등록
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onLoad ... 커스텀 UI 불러오기 훅

AI 번역본
-For Reference Only-

15.7.21 사

AI 번역본
-For Reference Only-

사용자

AI 번역본
-For Reference Only-

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
도입 플로우/설치（Linux）/설치 및 업데이트 도입 플로우/초기 설정/관리자 패스워드의
변경
도입 플로우/초기 설정/사용자의 추가
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/엔진 기능의 전체상/기능 목록/WEBUI
각종 기능 설명/엔진 기능의 전체상/기능 목록/데이터베이스
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/플로우 에디터/모드에 대해
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/HTTP/HTTPS 통신 각종 기능
설명/플로우 에디터/화면 조작/설정 메뉴/프록시 설정
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리
각종 기능 설명/플로우 에디터/화면 조작/패널
각종 기능 설명/플로우 에디터/화면 조작/컴포넌트/기동/정지
각종 기능 설명/컬렉터/MQTT 컬렉터
각종 기능 설명/에미터/MQTT 에미터
각종 기능 설명/에미터/FTP/FTPS/SFTP 에미터
각종 기능 설명/액션/메일 송신
각종 기능 설명/시스템 컴포넌트/코어 인터널
각종 기능 설명/각종 로그/로그의 종별
각종 기능 설명/각종 로그/조작 로그
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트의 개요/커스텀 컴포넌트란

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/설정 화면

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트를 사용한다/설정 화면

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/커스텀 UI의 작성 방법
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 UI 작성/Synapse 커스텀 UI용 HTML 작성, 적용 방법/편집 제한의 제어

응용적인 이용 방법/Grafana와의 연계/Grafana 연계의 개요
응용적인 이용 방법/Grafana와의 연계/Grafana의 설치
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
보안 이용 가이드/사용자의 관리
후방 호환에 관하여/후방 호환의 대상
후방 호환에 관하여/후방 호환
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항
부록/에러 코드 목록/MQTT 컬렉터
부록/에러 코드 목록/MQTT 에미터
부록/에러 코드 목록/FTP/FTPS/SFTP 에미터
부록/에러 코드 목록/메일 송신

AI 번역본
-For Reference Only-

부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveComponentBase Objects/interval_iteration
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_interval_call() ... 정기 처리 함수
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_component_load_running_info() ... 저장한 실행 시의 정보를 읽어들인다

부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_get_common_variable() ... 컴포넌트 공통 변수 취득

부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/HIVE_COMPONENT ... 컴포넌트 인스턴스

부록/커스텀 컴포넌트 C-API 레퍼런스/형・구조체・함수/hive_interval_proc_t ... 정기 처리 함수 시그니처
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onLoad ... 커스텀 UI 불러오기 훅
부록/커스텀 UI JS-API 레퍼런스/콜백 함수/onSave ... 파라미터 저장 훅
부록/커스텀 UI JS-API 레퍼런스/SpeeDBee Synapse 설정 취득 함수
부록/커스텀 UI JS-API 레퍼런스/형/ComponentResponse

15.7.22 라

라이선스

머리말/동의 사항
도입 플로우/초기 설정/관리자 패스워드의 변경
각종 기능 설명/엔진 기능의 전체상/기능 목록/코어
각종 기능 설명/엔진 기능의 전체상/동작 사양/기본 동작
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/실행 라이선스 설정 다운로드
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/소프트웨어 정보
각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/포인트에 대해
각종 기능 설명/설정 정보의 관리(프로젝트)/프로젝트의 개요
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/커스텀 컴포넌트 개발 환경(SCCDE)/SCCDE의 사용법과 패키징/커스텀 컴포넌트 관리 디렉터리의 초기화
실행 라이선스/라이선스의 종류
실행 라이선스/실행 라이선스 이용 환경의 동작 사양
실행 라이선스/실행 라이선스의 운용 방법
실행 라이선스/실행 라이선스 가동 환경에 관한 주의사항

-For Reference Only-

실시간 분석

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
SpeeDBee Synapse의 개요/운용까지의 흐름
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트 각종 기능 설명/플로우 에디터/화면 항목/Edit Mode

리소스 컬렉터

도입 플로우/컴포넌트의 설정/리소스 컬렉터의 설정 도입
플로우/컴포넌트의 설정/CSV 시리얼라이저의 설정 도입 플로우/수집
서비스의 실행/동작 확인
도입 플로우/수집 서비스의 실행/설정 튜닝
각종 기능 설명/컬렉터/리소스 컬렉터
각종 기능 설명/시리얼라이저/JSON 시리얼라이저
각종 기능 설명/시리얼라이저/CSV 시리얼라이저
각종 기능 설명/로직（이벤트）/이벤트 데이터
각종 기능 설명/로직（이벤트）/이벤트 트리거
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/수치 변환의 설정
각종 기능 설명/컴포넌트 공통의 설정 항목/출력 포트/등록 조건의 설정
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/SELECT
각종 기능 설명/DB 쿼리（SQL）/DB 쿼리의 상세/구문 사양/INSERT
실행 라이선스/실행 라이선스의 운용 방법/간이 튜토리얼/개발 라이선스를 사용한 시스템 구축부록/에러 코드
목록/리소스 컬렉터

로그아웃

도입 플로우/초기 설정/사용자의 추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/로그아웃

로그인

도입 플로우/설치（Linux）/설치 및 업데이트 도입
플로우/설치（Linux）/SpeeDBee Synapse의 제어
도입 플로우/설치（Linux）/제거
도입 플로우/초기 설정/관리자 패스워드의 변경
도입 플로우/초기 설정/사용자의 추가
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/플로우 에디터/화면 항목/Run Mode
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/추가
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/편집
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/프로필
각종 기능 설명/플로우 에디터/화면 조작/사용자 관리/로그인
응용적인 이용 방법/Grafana와의 연계/Grafana의 셋업
i-Trigger（확장 패키지）/설치/설치 및 업데이트
i-Trigger（확장 패키지）/설치/제거

AI 번역본
-For Reference Only-

로그 출력

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/로그 설정
각종 기능 설명/각종 로그/컴포넌트 개별 로그
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/Python 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/로그 출력

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/라이프사이클 콜백 정의

응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/입력 포트로부터의 데이터 불러오기（read_and_log ）
응용적인 이용 방법/커스텀 컴포넌트의 작성・이용/C언어 커스텀 컴포넌트 작성/커스텀 컴포넌트 실장 상세/로그 출력

부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveLog Objects
부록/커스텀 컴포넌트 C-API 레퍼런스/로그 관련 함수/HIVE_LOG_XXXX() ... 로그 출력 매크로
부록/커스텀 컴포넌트 C-API 레퍼런스/로그 관련 함수/hive_current_loglevel() ... 로그 레벨 취득

로그 레벨

각종 기능 설명/플로우 에디터/화면 조작/설정 메뉴/로그 설정
각종 기능 설명/각종 로그/컴포넌트 개별 로그
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveLogLevel Objects
부록/커스텀 컴포넌트 Python-API 레퍼런스/HiveLog Objects/get_current_level
부록/커스텀 컴포넌트 C-API 레퍼런스/로그 관련 함수/HIVE_LOG_XXXX() ... 로그 출력 매크로
부록/커스텀 컴포넌트 C-API 레퍼런스/로그 관련 함수/hive_current_loglevel() ... 로그 레벨 취득

로직

SpeeDBee Synapse의 개요/SpeeDBee Synapse의 구성
각종 기능 설명/엔진 기능의 전체상/기능 목록/컴포넌트
각종 기능 설명/플로우 에디터/화면 항목/Edit Mode
각종 기능 설명/로직（이벤트）/이벤트 트리거
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_lock() ... 컴포넌트 간 배타 제어
부록/커스텀 컴포넌트 C-API 레퍼런스/기타 함수/hive_unlock() ... 컴포넌트 간 배타 제어

-For Reference Only-

15.8 DB 쿼리 서비스 Web-API 사양

이 장에서는, DB 쿼리 서비스의 Web-API 사양에 대해 설명합니다.

AI 번역본
-For Reference Only-

Core Interface (0.1.0)

Download OpenAPI specification: Download

코어 DB 쿼리 전용 API

DB 쿼리 실행

DB 쿼리를 실행하고 그 실행 결과를 반환한다. 쿼리를 실행한 후 그 내용이 에러가 되는 경우는, 스테이터스 코드200 에서 result-type:error 의 응답을 반환한다.
쿼리의 실행 자체를 할 수 없는 경우는, 200 이외의 스테이터스 코드를 반환한다.

HEADER PARAMETERS

- `X-hive-api-key`
  string
  토큰 인증을 활성화 한 경우에 지정하는 토큰 문자열.

REQUEST BODY SCHEMA: application/json
required

DB 쿼리

- `query`
  required
  string
  쿼리 문자열
- `timeout`
  int [1 .. 180]
  Default: 30
  쿼리 처리의 타임아웃(초), 지정 시간을 지나면 타임아웃이 된다
- `timestamp_unit`
  string
  Default: "nanosecond"
  Enum: "nanosecond" "microsecond" "millisecond" "second"
  반환 결과의 타임스탬프의 단위
- `omit_option`
  string
  Default: 0
  Enum: 0 1 2
  데이터 없음, NULL값의 데이터를 생략할지 지정한다(0=NULL과 데이터 없음 생략한다,1 or 2=생략하지 않는다)

AI 번역본
-For Reference Only-

Responses

200 DB 쿼리 결과

RESPONSE SCHEMA: application/json

- **query** (required)
  string
  쿼리 문자열
- **timestamp_unit** (required)
  string
  타임스탬프의 단위 지정
- **success** (required)
  boolean
  쿼리의 실행이 성공했음을 나타낸다
- **cutoff**
  boolean
  쿼리 결과가 잘렸는지 여부를 나타낸다
- **error**
  object (DbQueryError)
  쿼리의 실행 에러 정보를 유지한 오브젝트
  - **code**
    string
    에러 식별자
  - **message**
    string
    에러 메시지
- **result**
  object (DbQueryResult)
  쿼리의 실행 결과가 있는 경우에 설정되는 오브젝트
  - **columns**
    Array of strings (HiveColumnListResult) [items]
    컬럼 목록 취득 요구 시에만 설정되는 오브젝트
  - **record_columns**
    Array of objects (DbQueryResultColumnMeta)
    컬럼의 메타 정보 목록

    Array [
    - **name** (required)
      string
    - **data_type** (required)
      string (DbQueryDataType)
      Enum: `"bool"` `"int8"` `"int16"` `"int32"` `"int64"` `"uint8"` `"uint16"` `"uint32"` `"uint64"` `"float"` `"double"` `"timestamp"` `"string"` `"binary"`

AI 번역본
-For Reference Only-

data_array
required

integer (DbQueryDataArray) >= 1

]

records

Array of objects (DbQueryResultRecord)

Array [

timestamp
required

int
레코드의 타임스탬프

window_id
required

int
참조 시의 윈도우를 나타내는 ID

record_type
required

string (DbQueryResultRecordType)
Enum: "D" "S"

data

Array of objects

Array [

name
required

string

value
required

any

data_type
required

string (DbQueryDataType)
Enum: "bool" "int8" "int16" "int32" "int64" "uint8" "uint16" "uint32" "uint64" "float" "double" "timestamp" "string" "binary"

data_array

integer (DbQueryDataArray) >= 1

]

]

-For Reference Only-

500 본 기능은 현재 무효화되어 있다

RESPONSE SCHEMA: application/json

id
required

string

code required	integer	
codeHex required	string `0x[0-9a-f]{8}`	
message required	string	
parameters	Array of any	
extendedInfo	any	

POST /dbquery

Request samples

Payload

Content type
application/json

Copy

```
{
    "query": "SELECT ¥"comp1:data1¥", ¥"comp2:dataA¥" FROM SPDB;",
    "timeout": 15,
    "timestamp_unit": "nanosecond"
}
```

AI 번역본
-For Reference Only-

Response samples

200 500

Content type
application/json

Example
DB 쿼리 결과

Copy Expand all Collapse all

```
{
    "query": "SELECT ¥"comp1:data1¥", ¥"comp2:dataA¥" FROM SPDB;",
    "timestamp_unit": "nanosecond",
    "success": true,
```

```
  - "result": {
      + "record_columns": [ … ],
      + "records": [ … ]
    }
}
```


DB 쿼리 실행 (스트림 형식)

DB 쿼리를 실행하고 그 실행 결과를 스트림 형식으로 반환한다. 본 API에서는 SELECT 등의 서버 상태에 변화를 주지 않는 쿼리만 지원한다. 현시점에서는 비지원. (장래적으로 추가할 가능성 있음)

REQUEST BODY SCHEMA: application/json
required

DB 쿼리

- `query` (required)
  string
  쿼리 문자열
- `timeout`
  int [1 .. 180]
  Default: `30`
  쿼리 처리의 타임아웃(초), 지정 시간을 지나면 타임아웃이 된다
- `timestamp_unit`
  string
  Default: `"nanosecond"`
  Enum: `"nanosecond"` `"microsecond"` `"millisecond"` `"second"`
  반환 결과의 타임스탬프의 단위
- `omit_option`
  string
  Default: `0`
  Enum: `0` `1` `2`
  데이터 없음, NULL값의 데이터를 생략할지 지정한다(0=NULL과 데이터 없음 생략한다,1 or 2=생략하지 않는다)

AI 번역본
-For Reference Only-

Responses

200 DB 쿼리 결과

RESPONSE SCHEMA: application/jsonlines

```
Array [
```

- timestamp (required): int
  레코드의 타임스탬프
- window_id (required): int
  참조 시의 윈도우를 나타내는 ID
- record_type (required): string (DbQueryResultRecordType)
  Enum: `"D"` `"S"`
- data: Array of objects

```
Array [
  name          string                                   required
  value         any                                      required
  data_type     string (DbQueryDataType)                 required
                Enum: "bool" "int8" "int16" "int32" "int64"
                "uint8" "uint16" "uint32" "uint64" "float"
                "double" "timestamp" "string" "binary"
  data_array    integer (DbQueryDataArray) >= 1
]
```

]

500 본 기능은 현재 무효화되어 있다

RESPONSE SCHEMA: application/json

AI 번역본
-For Reference Only-

- id (required): string
- code (required): integer
- codeHex (required): string `0x[0-9a-f]{8}`
- message (required): string
- parameters: Array of any
- extendedInfo: any

GET /dbquery

Request samples

Payload

Content type
application/json

Copy

```
{
    "query": "SELECT ¥"comp1:data1¥", ¥"comp2:dataA¥" FROM SPDB;",
    "timeout": 15,
    "timestamp_unit": "nanosecond"
}
```

Response samples

200 500

Content type
application/jsonlines

Copy Expand all Collapse all

```
[
 - {
        "timestamp": null,
        "window_id": null,
        "record_type": "D",
     + "data": [ … ]
    }
]
```

AI 번역본
-For Reference Only-

AI 번역본
-For Reference Only-

⚠ 안내 (For Reference Only)

본 문서는 AI 번역으로 작성된 참고용 국문자료 입니다.
일부 표현은 원문과 다소 다를 수 있으며,
중요한 내용은 원문과 함께 참고해 주시기 바랍니다.

Industrial Web ▶ http://www.ia.omron.co.kr

한국 오므론 제어기기 주식회사
서울특별시 서초구 강남대로 465, B동 18층(서초동, 교보타워)
TEL: 1522-8994 FAX: 02-3483-7788

매뉴얼 번호 SBSB-540A 2025년 9월 현재

©OMRON Corporation 2025 All Rights Reserved.
예고없이 사양 등을 변경하는 경우가 있으므로 양해 부탁드립니다.