

Gravio Documentation - The Node Computing Platform

1 — 最終更新日時: 15 December 2023

ASTERIA Corporation

目次

1. Gravio 構成と用語の解説	8
2. Gravio 5.1での変更点	11
3. Gravio 4.xからのGravio 5.0へのアップグレード手順	12
3.1. Gravio 4.xのライセンス形式を5.0用に設定する	13
3.2. Windows版	14
3.3. macOS版	15
3.4. Gravio Hub 2	16
3.5. Linux (Ubuntu AMD64)版	17
3.6. Linux (Ubuntu ARM64/ Raspberry Pi)版	18
4. 新規インストールとセットアップ	19
4.1. Gravio Hub 2の運用方法の決定と設定手順	20
4.1.1. Gravio Hub 2の設置	21
4.1.2. Gravio HubをLANに接続する	22
4.1.3. Gravio Hub設定	24
4.1.4. 初期設定(Gravioコンフィグレーションマネージャー)	31
4.1.5. その他の設定	36
4.1.5.1. Gravio設定	37
4.1.5.2. Gravio Hubの運用の注意事項	43
4.2. Windows版のインストールと初期設定	44
4.2.1. Windows版のインストール	45
4.2.2. 初期設定(Gravioコンフィグレーションマネージャー)	48
4.3. macOS版のインストールと初期設定	54
4.3.1. macOS版のインストール	55
4.3.2. 初期設定(Gravioコンフィグレーションマネージャー)	58
4.4. Linux (Ubuntu AMD64)版のインストールと初期設定	63
4.4.1. Ubuntu(AMD64)版のインストール	64
4.4.2. 初期設定(Gravioコンフィグレーションマネージャー)	65
4.5. Linux (Ubuntu ARM64/Raspberry Pi)版のインストールと初期設定	71
4.5.1. Ubuntu(ARM64)版のインストール	72
4.5.2. 初期設定(Gravioコンフィグレーションマネージャー)	73
4.6. 過去のHubKitのインストールとバージョンアップ手順	79
4.6.1. Linux / Raspberry PiのHubKit 4.2 から4.3へのアップデート手順	80
4.6.2. Linux / Raspberry PiのHubKit 4.1から4.2へのアップデート手順	83
4.6.3. Linux / Raspberry Piのセットアップ 4.3/4.4系	85
4.6.4. Gravio 4.5でのライセンス管理	87
5. Gravio Cloudの利用	92
5.1. サーバ構築依頼	93
5.1.1. Business, Business Proをご購入の方が利用する場合	94
5.1.2. Gravio Freeの方が評価でサーバを利用する場合	97
5.2. サーバーに関する情報	100

6. Gravio運用時のアカウントの種類	101
7. Gravio Studioの利用	102
7.1. ログイン	103
7.2. Hubリストビュー	104
7.3. Hub構成ビュー	107
7.3.1. デバイスタブ	108
7.3.1.1. 物理センサー	110
7.3.1.1.1. Zigbee対応デバイスの追加	112
7.3.1.1.2. Zigbeeレシーバー(旧型)のドライバー登録(Windowsのみ)	118
7.3.1.1.3. EnOcean対応デバイスの追加	122
7.3.1.1.4. Bluetooth対応デバイスの追加	127
7.3.1.1.5. バーコードリーダーの追加	132
7.3.1.1.6. GPSデバイスの追加	137
7.3.1.2. ソフトウェアセンサー	143
7.3.1.2.1. 推論モデル	151
7.3.1.3. MQTTサブスクリバ	152
7.3.1.4. Gorilla IVAR	160
7.3.1.5. i-PRO	165
7.3.1.6. 物理デバイスリスト	169
7.3.2. トリガータブ	171
7.3.2.1. デバイストリガー	172
7.3.2.1.1. 設定例	177
7.3.2.2. タイマートリガー	181
7.3.2.3. トリガー変数とトリガープロパティ	185
7.3.3. データビューア	186
7.3.4. ログビューア	187
7.3.5. 設定タブ	189
7.4. アクション構成ビュー	198
7.4.1. アクションの作成	199
7.4.2. アクション内で利用できる変数と変数型	201
7.4.3. アクション内でObject型の変数の値を参照する方法	205
7.4.4. アクションコンポーネントの動作	207
7.4.5. Pre Mappings / Post Mappingsでの式の記述の仕方	208
7.4.6. Pre Mappings / Post Mappingsの式の中で利用できる関数	210
7.4.7. デバッグログ出力	211
7.4.8. ファイルビューア機能	212
7.4.9. アクションコンポーネント解説	220
7.4.9.1. Amazon Polly Speech	221
7.4.9.2. Aqara Smart Plug	224
7.4.9.3. AWS S3 Download	225
7.4.9.4. AWS S3 Upload	227
7.4.9.5. BrightSign	229
7.4.9.6. Camera Take Photos	230
7.4.9.7. Chatwork	232

7.4.9.8. Chromecast	235
7.4.9.9. CSV Read.....	236
7.4.9.10. CSV Write.....	238
7.4.9.11. CSV File Write	240
7.4.9.12. Exec	242
7.4.9.13. Execute Inference Model	245
7.4.9.14. File Delete	248
7.4.9.15. File List.....	249
7.4.9.16. File Read	251
7.4.9.17. File Write	252
7.4.9.18. Filter	253
7.4.9.19. GCS Upload	255
7.4.9.20. GCS Download	257
7.4.9.21. Google Analytics Send Event.....	258
7.4.9.22. GoogleDriveDownload	260
7.4.9.23. GoogleDriveUpload.....	264
7.4.9.24. GPS.....	268
7.4.9.25. Gravio Light	271
7.4.9.26. Gravio Distance	273
7.4.9.27. Gravio Matrix	275
7.4.9.28. HTTP Request.....	278
7.4.9.29. Join.....	283
7.4.9.30. LINE Notify	284
7.4.9.31. LINE WORKS (旧版)	286
7.4.9.32. LINE WORKS	288
7.4.9.33. Mail Send	292
7.4.9.34. MQTT Publish.....	299
7.4.9.35. SQL Server Exec	301
7.4.9.36. SQL Server Query	303
7.4.9.37. MSTEams Send Message	306
7.4.9.38. MySQL Exec.....	308
7.4.9.39. MySQL Query	311
7.4.9.40. OpenAICompletion.....	314
7.4.9.41. OpenAIChat.....	316
7.4.9.42. OpenAIImage.....	318
7.4.9.43. OpenAIAudio	320
7.4.9.44. PATLITE	322
7.4.9.45. Philips Hue	324
7.4.9.46. PostgreSQL Exec	327
7.4.9.47. PostgreSQL Query.....	330
7.4.9.48. Sensor Data DB.....	333
7.4.9.49. Sensor Data DB If Zero.....	336
7.4.9.50. Slack Post Message	339
7.4.9.51. Slack Files Upload	340
7.4.9.52. Sleep	342

7.4.9.53. SORACOMDataPost	343
7.4.9.54. SORACOMFileUpload	346
7.4.9.55. Sound Effect	348
7.4.9.56. Sound Play	349
7.4.9.57. Split	350
7.4.9.58. Statistics	352
7.4.9.59. Template	355
7.4.9.60. Transform	357
7.4.9.61. Twilio SMS	358
7.4.9.62. Write To Excel (365)	359
7.4.9.63. Write To Excel (Local)	361
7.4.10. 関数解説	363
7.4.10.1. 文字列	364
7.4.10.1.1. JSONPath()の使用例	369
7.4.10.1.2. Sprintfの使用	371
7.4.10.2. 型変換	372
7.4.10.3. エンコード	373
7.4.10.4. 日付	374
7.4.10.5. 環境変数	377
7.4.10.6. 正規表現	378
7.4.10.7. 算術系関数	379
8. Gravio AI オプションの利用	382
8.1. Gorilla IVAR, BAPモジュールの取得	383
8.2. Gorilla BAPの設置	384
8.2.1. Gorilla BAPのアップデート	398
8.2.2. Gorilla BAPのライセンス登録	405
8.2.3. Gorilla BAPのライセンスの登録解除	408
8.2.4. Gorilla BAPのアンインストール	409
8.2.5. Feature versionの変更	411
8.3. Gorilla IVARの設置	415
8.3.1. Gorilla IVARのアップデート	423
8.3.2. Gorilla IVARの初期設定とライセンス登録	427
8.3.3. Gorilla IVARのライセンスの登録解除	433
8.3.4. Gorilla IVARのアンインストール	434
8.4. Gravioへの接続テスト – 人数カウント	440
8.4.1. IVARのカメラ+画像認識の設定	441
8.4.2. GravioとIVAR連携の設定	450
8.4.3. Gravio IVARトリガーの設定	457
8.5. Gravioへの接続テスト – 顔認識	461
8.5.1. BAPの設定	463
8.5.2. IVARのカメラ+顔認識の設定	468
8.5.3. GravioとIVAR連携の設定	477
8.5.4. Gravio IVARトリガーの設定	484
9. より深くGravioを利用するための情報	488

9.1. 物体認識の機械学習モデル作成手順	489
9.2. 補足資料:Labellmg によるアノテーションデータ作成.....	496
9.3. Gravio Coordinatorを使う(Gravio Enterpriseサービス契約のみ)	498
10. 3rdパーティー製品との接続	499
11. Gravio Hub (Gravio Hub 2).....	500
11.1. Gravio Hub 2のインジケータランプの状態とボタン操作	501
11.2. Gravio Hub2のファクトリリセット手順.....	502
11.3. SDカードを使ったWiFi設定.....	504
11.4. SDカードを使ったGravio Hub設定	505
11.5. Gravio Hub から Gravio Hub 2への移行ガイド	506
12. Gravio Hub (Gravio Hub version 1).....	508
12.1. Gravio Hub(Gravio Hub version 1)の初期設定.....	509
12.2. Gravio Hub (Gravio Hub version 1)のインジケータランプの状態とボタン操作	511
12.3. Gravio Hubの停止方法	512
13. トラブルシューティング.....	513
13.1. 一般的な課題	514
13.2. センサー関係のトラブルシューティング.....	515
13.3. Gravio StudioからHubKitが見つからないまたは繋がらない	516
14. 補足.....	518
14.1. Gravioが利用するネットワークポート	519
14.2. HubKitのデータディレクトリ	520
14.3. カメラ画像の保存先	521
14.4. Gravio StudioからHubへのアクセス制限をかけたい場合	522
14.5. HubKitにルートCA証明書をインストールする方法	525
14.6. Gravio Studio for Windowsのサイドローディングインストール	531
14.7. Zigbee firmware のアップデート方法	538
14.8. i-PROカメラの設定方法	541
14.9. OpenAIのAPI Keyの取得方法	544
14.10. Text2SQLの使い方.....	546
14.11. Google Cloud Storage設定方法	548
14.12. AWS IoT MQTTの設定方法	553
14.13. Azure IoT Hub (MQTT)の設定方法	561
14.14. Azure アプリの登録方法 (Microsoft365)	572
14.15. LINE WORKS (旧版) のBot 登録方法	580
14.16. LINE WORKSのBot 登録方法.....	585
14.17. Gravio Hub, Gravio Zigbee レシーバーに関して	590
14.18. Gravio CO2 デバイスについて	591
14.19. OAuth2.0でのGoogle Drive設定方法	592
14.20. Gmailでのメール送信設定方法：基本認証(SMTP)	595
14.21. OAuth2.0でのメール送信設定方法：先進認証(OAuth2)	596
14.22. ファイルパスの記述方法	610
14.23. Gravioコンフィグレーションマネージャーの自動更新.....	612

14.24. ソフトウェアセンサーを利用する場合のハードウェア環境	613
14.25. HubKitの設定	614
14.26. クレジット	625
14.27. Gravio App Service API.....	627
14.28. Gravio HubでBluetoothスピーカーを利用する場合(Gravio 5.1以上).....	628
15. Security and Authentication	629

1. Gravio 構成と用語の解説

Gravioは [バージョン 5](#) からAWS, Azure, GCPなどのクラウドサービス(以下「クラウド」と記述します。)にもGravioをインストールして運用できるようになりました。

Gravio HubKitがインストールされているノードをGravio バージョン 5からは「[Nodeハブ](#)」または「[ノード](#)」と呼びます。

クラウドサービス上にGravio HubKitパッケージをインストールすることで、[このサーバをデータを集約するNodeハブサーバとして利用できます](#)。

このNodeハブサーバを「[Mainサーバ](#)」と呼びます。

注: Mainサーバを運用するには、Gravio BusinessまたはBussiness Proサービス契約で付与される「[Mainライセンス](#)」が必要です。

Gravio Business/Bussiness Proをご契約のお客様は弊社が用意するGravio Cloudのサーバもご利用いただけます。

Gravio Mainサーバで動作するモジュール

Gravio MainサーバではHubKitパッケージに同梱されている以下のモジュールがサーバとして常駐します。

モジュール名	説明
Gravio コーディネーター	Nodeハブの管理およびセンサーデータの集約データベースを包含しています。
Gravio HubKit	アクションエンジンを含むセンサーデータなどの収集・管理を行うためのモジュール群
Gravio コンフィグレーションマネージャー	HubKitの設定およびGravioコーディネーターの初期設定機能などを提供する。

Gravio Nodeハブで動作するモジュール

Gravio NodeハブではHubKitパッケージに同梱されている以下のモジュールがサーバとして常駐します。

モジュール名	説明
Gravio HubKit	アクションエンジンを含むセンサーデータなどの収集・管理を行うためのモジュール群
Gravio コンフィグレーションマネージャー	HubKitの設定およびGravioコーディネーターの初期設定機能などを提供する。

HubKitの設定やActionフローの編集ツール

Gravioでは、「[アクションフロー](#)」(または「[アクション](#)」と記述します。)と呼ばれるユーザが作成するデータフロー

プログラムをHubKit上で実行できます。

HubKitの設定やアクションフローの編集ツールとして「Gravio Studio」というGUIアプリケーションのWindows版とMac版が用意されています。

モジュール名	説明
Gravio Studio	HubKitの設定やアクションフローの編集ツール

Gravio Studioを使って以下のような内容を管理・設定できます。

- *物理的なセンサーデバイスのセットアップ。物理的なスペース内のどこに配置されているかのマッピング。
- *センサーの管理、監視とメンテナンス。
- *センサネットワークによってトリガされるアクションを作成、設定。
- *アクションフローの作成・編集。
- *生のセンサーデータのデータ表示。

Gravio.com

お客様のGravioの契約情報やライセンスの利用状況などを閲覧・管理するためのポータルサイトがあります。Gravio StudioからGravio.comにログインすると、Gravio Studioの設定パネルからgravio.comへ各種ファイルのバックアップサービスが利用できます。

備考

Gravio HubKitは複数のサーバーモジュールで構成されています。

サーバ名	説明
Gravio Control Manager	Gravio Studioとの通信を司るサーバー
Gravio Trigger Manager	様々なセンサーデータを各デバイス管理マネージャサーバ(Serial Port, Bluetooth, Videoなど)から受信し、センサーデータの保存を行うサーバー。また入って来たデータの値をトリガー設定で指定されたコンディションに当てはまるかをチェックし、当てはまる場合は、Gravio Action Managerに実行を依頼します。
Gravio Action Manager	アクション実行、センサーデータの保存を行うサーバー
Gravio Serial Port Manager	シリアルポートによる物理センサーとの接続管理と、センサーから取得したデータをGravio Trigger Managerに送信する役目を担っています。
Gravio Bluetooth Manager	Bluetoothで接続されている物理センサーとの接続管理と、センサーから取得したデータをGravio Trigger Managerに送信する役目を担っています。

Gravio Video Service	Onvif Cameraとの接続管理と、カメラからの画像/動画保存、画像推論から取得したデータをGravio Trigger Managerに送信する役目を担っています。
Gravio MQTT Service	MQTTから取得したデータをGravio Trigger Managerに送信する役目を担っています。
Gravio Image Processing Manager	IVARなどの画像処理サーバから取得したデータをGravio Trigger Managerに送信する役目を担っています。
Gravio App Service	コンポーネントを実行するための補助的な機能を持つサーバ

2. Gravio 5.1での変更点

Gravio 5.1では、Gravio 5.0までと異なり、下記のようないくつかの機能強化がされています。

Gravioコンフィグレーションマネージャー

- ・初期設定

Gravio5.1ではインターネットに接続されていない環境でもライセンス登録が出来るようになりました。
Gravio.comで予めHubKit IDを使用してライセンスファイルを作成してダウンロードしたものをインポートすることが出来ます。

詳細は Gravio Hub 2の場合は [こちら](#) を参照してください

HubKit

- ・Gravioが利用するネットワークポート

HubKit(Gravio Control Manager)は443ポートを使用するように変更となりました。
Gravio Studioも443ポートを使用してHubKitに接続します。

詳細は [こちら](#) を参照してください

- ・Gravio StudioからHubへのアクセス制限をかけたい場合

クライアント認証機能はGravio 5.1ではなくなりました。

詳細は [こちら](#) を参照してください

Gravio Studio

Gravio StudioではGravio.comにログインせずに使用出来る「ログインなし」機能が追加されました。
「ログインなし」機能によりインターネットに接続されていない環境でも一部の機能を除いて使用できます。
「ログイン」機能はコーディネーターにログインして使用します。

詳細は [こちら](#) を参照してください

Gravio Studio のHubリストビューではGravioコーディネーターへの接続状態が表示されるようになりました。
またHub 5.1以上に接続する場合と、5.0以下に接続する場合のスイッチが追加されています。

詳細は [こちら](#) を参照してください

3. Gravio 4.xからのGravio 5.0へのアップグレード手順

Gravio 4.xからのGravio 5.0へのアップグレード手順を説明します。

Gravio 5.0からはライセンスファイルを配置する方法が簡素化され、Gravioコンフィグレーションマネージャーで、ライセンスの取得が行えるようになりました。

このため、Gravio 4.xのライセンスをお持ちのお客様は、Gravio 5.0用のライセンス形式に変更する必要があります。

ライセンスの形式変更をgc1.gravio.comにログインして、ライセンス形式を5.0に変更してから、各プラットフォームでのHubKitパッケージのインストールを行ってください。

3.1. Gravio 4.xのライセンス形式を5.0用に設定する

HubKitパッケージのアップグレードをする前に、4.xのライセンス形式を5.0用に設定をしてください。

<https://gc1.gravio.com/ui/login> にログインします。

ログイン後、「アカウント」の画面で、



上記のセル内にある、緑のアイコンをクリックします。

「ライセンスバージョンの変更」のダイアログが表示されますので、「設定」を押して、5.0用に変更をしてください。

3.2. Windows版

HubKitのアップグレード

以下の手順でHubKitをアップグレードしてください。

アップグレードが正常に終了した場合には、バックアップしたデータは使用しません。

1. C:\ProgramData\HubKit をバックアップする

ProgramDataが隠しフォルダになっている場合は表示するようにしてからHubKitのデータをバックアップしてください。

2. HubKitをアンインストールする

Windowsからアンインストールします。

3. <https://www.gravio.com/jp/download> からHubKit Windows 5.0版をダウンロード

4. ダウンロードしたGravio HubKitをインストール

ダウンロードしたインストーラを実行してインストールします。

5. [初期設定](#) の手順を行う

Gravioの初期設定を Gravioコンフィグレーションマネージャーから行います。

3.3. macOS版

HubKitのアップグレード

以下の手順でHubKitをアップグレードしてください。

アップグレードが正常に終了した場合には、バックアップしたデータは使用しません。

1. ツールバーに表示されている**Gravio HubKit**のツールバーアプリで**HubKit**を停止する

2. **Gravio HubKit Toolbar**アプリを終了する

HubKitが起動している場合は先に終了してください。

3. **/Library/Application Support/HubKit** をバックアップする

HubKitのデータをバックアップしてください。

4. アプリケーションフォルダーの**Gravio HubKit4**をゴミ箱に入れる

使用しているHubKitを削除してください。

5. <https://www.gravio.com/jp/download> から**HubKit Mac**版をダウンロード

6. ダウンロードした**GravioHubKit.dmg**ファイルをダブルクリックして開く

7. **Gravio HubKit**のアイコンをアプリケーションフォルダーにドラックアンドドロップする

ダウンロードしたインストーラを実行してHubKitをコピーしてください。

8. アプリケーションフォルダーの**Gravio HubKit**をダブルクリックして起動

9. ツールバーに表示されている**Gravio HubKit**のツールバーアプリのメニューから「**Gravio初期設定**」のメニューをクリック

10. [初期設定](#) の手順を行う

Gravioの初期設定を Gravioコンフィグレーションマネージャーから行います。

3.4. Gravio Hub 2

HubKitのアップグレード

以下の手順でHubKitをアップグレードしてください。
アップグレードが正常に終了した場合には、バックアップしたデータは使用しません。

1. Update Managerにログインする

Update Managerが深夜の自動更新で最新版に更新されると、Gravio設定タブに Gravioバージョン 5へのアップグレード ボタンが表示されますので、確認してください。

Gravio設定

Gravio HubKit / Gravio コーディネーター 稼働状況

Gravio HubKit	稼働中	☒
Gravio コーディネーター	稼働中	☒

ソフトウェアアップデート

Gravioバージョン5へのアップグレード

Gravio HubKitソフトウェアのアップデート

Gravio HubKitの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio HubKit バージョン: 4.6.1

インストール

再起動

Gravio コーディネーターソフトウェアのアップデート

Gravio コーディネーターの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio コーディネーター バージョン: 4.6.1

インストール

再起動

Zigbeeソフトウェアのアップデート

バージョン: 0.10.5.1

インストール

2. HubKit / コーディネーターデータバックアップ からバックアップ

HubKitのデータをバックアップしてください。

3. Gravioバージョン 5へのアップグレード

Gravioバージョン 5へのアップグレード ボタンを押してください。確認ダイアログの表示の後、自動的に再起動されます。

再起動後にUpdate Managerは コンフィグレーションマネージャーと表示されます。

Gravioの初期設定を Gravioコンフィグレーションマネージャーから行いますので [こちら](#) を参照してください。

3.5. Linux (Ubuntu AMD64)版

HubKitのアップグレード

以下の手順でHubKitをアップグレードしてください。

アップグレードが正常に終了した場合には、バックアップしたデータは使用しません。

1. Update Managerにログインする

Update Managerが深夜の自動更新で最新版に更新されると、Gravio設定タブに Gravioバージョン 5へのアップグレード ボタンが表示されますので、確認してください。

Gravio設定

Gravio HubKit / Gravio コーディネーター 稼働状況

Gravio HubKit	稼働中	☒
Gravio コーディネーター	稼働中	☒

ソフトウェアアップデート

Gravioバージョン5へのアップグレード

Gravio HubKitソフトウェアのアップデート

Gravio HubKitの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio HubKit バージョン: 4.6.1

インストール

再起動

Gravio コーディネーターソフトウェアのアップデート

Gravio コーディネーターの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio コーディネーター バージョン: 4.6.1

インストール

再起動

Zigbeeソフトウェアのアップデート

バージョン: 0.10.5.1

インストール

2. HubKit / コーディネーターデータバックアップ からバックアップ

HubKitのデータをバックアップしてください。

3. Gravioバージョン 5へのアップグレード

Gravioバージョン 5へのアップグレード ボタンを押してください。確認ダイアログの表示の後、自動的に再起動されます。

再起動後にUpdate Managerは コンフィグレーションマネージャーと表示されます。

Gravioの初期設定を Gravioコンフィグレーションマネージャーから行いますので [こちら](#) を参照してください。

3.6. Linux (Ubuntu ARM64/ Raspberry Pi)版

HubKitのアップグレード

以下の手順でHubKitをアップグレードしてください。

アップグレードが正常に終了した場合には、バックアップしたデータは使用しません。

1. Update Managerにログインする

Update Managerが深夜の自動更新で最新版に更新されると、Gravio設定タブに Gravioバージョン 5へのアップグレード ボタンが表示されますので、確認してください。

Gravio設定

Gravio HubKit / Gravio コーディネーター 稼働状況

Gravio HubKit	稼働中	
Gravio コーディネーター	稼働中	

ソフトウェアアップデート

[Gravioバージョン5へのアップグレード](#)

Gravio HubKitソフトウェアのアップデート

Gravio HubKitの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio HubKit バージョン: 4.6.1  [インストール](#) [再起動](#)

Gravio コーディネーターソフトウェアのアップデート

Gravio コーディネーターの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio コーディネーター バージョン: 4.6.1  [インストール](#) [再起動](#)

Zigbeeソフトウェアのアップデート

バージョン: 0.10.5.1  [インストール](#)

2. HubKit / コーディネーターデータバックアップ からバックアップ

HubKitのデータをバックアップしてください。

3. Gravioバージョン 5へのアップグレード

Gravioバージョン 5へのアップグレード ボタンを押してください。確認ダイアログの表示の後、自動的に再起動されます。

再起動後にUpdate Managerは コンフィグレーションマネージャーと表示されます。

Gravioの初期設定を Gravioコンフィグレーションマネージャーから行いますので [こちら](#) を参照してください。

4. 新規インストールとセットアップ

Gravioはサーバソフトウェアである「[Gravio HubKit](#)」パッケージ と、このサーバソフトウェアを操作するためのクライアントアプリケーションとして、「[Gravio Studio](#)」というアプリケーションの2つのソフトウェアをインストールして運用します。

Gravio HubKitはLinux, Windows, Macの3つのOS上でご利用いただけます。これらは、

<https://www.gravio.com/jp/download> からダウンロードできます。

ご利用されるOS向けのインストーラをダウンロードしてください。

Gravio StudioはWindowsではMS Storeのアプリから、MacではApp Storeアプリでそれぞれ公開されていますので、そちらで「Gravio Studio」と検索していただき、アプリのインストールをお願いします。

注意：Windows版のGravio HubKitパッケージをインストールする場合は、必ず先にMS StoreからGravio Studioを先にインストールしてください。

4.1. Gravio Hub 2の運用方法の決定と設定手順

最初にGravio Hub 2をどのように運用するか運用方法を決めてください。

Gravio Hub 2を「ノード管理用サーバとして運用する」のか、または、「Nodeハブとして運用する」のかを決めてください。

設定にはGravioコンフィグレーションマネージャーの画面でブラウザから行います。

1. [Gravio Hub設定](#) 画面でGravio HubのWiFi設定を行って、Gravio Hub 2をインターネットに接続できるようにする。
2. [初期設定](#) 画面でGravioのライセンスを取得する
 - Gravio Hub 2を [ノード管理用サーバ](#) にするには [MAINライセンス](#) を取得して、[Gravioコーディネーターの初期設定する](#)
注: ノード管理用サーバと設定してもこの管理サーバもノードの1つとして、センサーデータの取得やアクションの実行を行うことができます。
 - Gravio Hub 2を [Nodeハブとして利用する場合](#) は [NODEライセンス](#) を取得して、このNodeハブを [Gravioコーディネーターの管理下に登録する](#)

これで、動作環境の設定は完了です。

4.1.1. Gravio Hub 2の設置

Gravio Hub 2の設置に関して説明していきます。

Gravio Hub 2を利用するために、必要なことは以下の3点です。

1. [Gravio HubをLANに接続する](#)
2. [Gravio Hub設定](#)
3. [初期設定](#)

Gravio Hub 2はUbuntu2004ベースのLinuxOSで動作しています。必要に応じてLinuxOSへログインできる設計になっています。このため、Linuxにログインするためのログインアカウントのパスワードをセキュリティを高めるためにGravioコンフィグレーションマネージャーの初期設定で、[パスワードの変更](#)を行います。

事前にWiFiネットワークの設定（接続するネットワーク名とパスワード）とログインアカウントの新しく設定するパスワードを準備してください。

注）再設定したログインアカウントのパスワードは大切に保存してください。もしパスワードを紛失した場合にはLinuxにログインできなくなります。

Gravio Hub 2には以下のGravio Hubを含めて以下の付属品がレンタルされています。

- Gravio Hub2 本体



- Gravio USB-Cケーブル
- 電源ボックス

Gravio Hub2にはUSB-Aコネクタ、USB-Cコネクタ、有線LANコネクタ、HDMIコネクタがそれぞれ1つあります。

4.1.2. Gravio HubをLANに接続する

有線LAN接続の場合

Gravio Hub2のLANコネクタに有線LANケーブルを繋げる。

このときDHCPが使える有線LANをご利用ください。

Gravio Hub2を起動する

Gravio Hub2のUSB-CコネクタにUSB-Cケーブルを接続して起動します。

無線LAN接続の場合

Gravio Hub2を起動する

Gravio Hub2のUSB-CコネクタにUSB-Cケーブルを接続して起動します。

Gravio Hub2にWifi設定を行います。



Gravio Hubを電源につなげて起動します。

Gravio Hubの上部のGravioマークが青く表示されたら、お持ちのPCのWiFiのSSIDからGravioHubXXXXXXXXXXという名称のSSIDにつなぎます。

WiFi パスワードは「graviohub」です。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

アカウント

パスワード

パスワード

Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

🌐 日本語

ログイン

繋いだ後に、ブラウザから「<http://192.168.100.1:8080>」と入力してGravioコンフィグレーションマネージャーを開きます。

次に、「[Gravio Hub設定](#)」に進んで、時刻設定、地域設定、Gravio Hubのインターネット設定の設定に進んでください。

4.1.3. Gravio Hub設定



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語 ▾

ログイン

GravioコンフィグレーションマネージャーはHubKitに同梱されているWebアプリケーションです。

そのため、このシステムを安全に管理するために、**Gravioコンフィグレーションマネージャーには専用のアカウントが用意されています。**

ブラウザから「http://192.168.100.1:8080」と入力してコンフィグレーションマネージャーを開きます。

ユーザ名: 「gravio」

パスワード: 「gravio」

でログインします。

Change 'gravio' account password

現在のパスワード: 

新しいパスワード: 

新しいパスワード(再入力): 

パスワードの変更

最初にログインするときにパスワードを変更するパネルが表示されますので、現在のパスワードと新しいパスワードを入力して「パスワード変更」ボタンを押します。以降は新しいパスワードを使用します。

ログインすると Gravio Hub設定 の画面が開きます。

※HubKit Windows/Mac/Linux/RaspberryPIではこのメニューは表示されません。

時刻設定

インターネットに接続しているネットワークで利用する場合は、この時刻設定は必要ありませんので、以下の時刻設定の手順を飛ばして設定を進めてください。

Gravio Hubを設置するネットワークがインターネットに接続していない場合は、マニュアル設定を選んで日時を設定してください。

時刻設定

タイムゾーン

Asia/Tokyo

セット

時刻設定

現在時刻: 2021-03-09 09:02:22

☒ NTPを使用

NTPから同期:

同期

☐ マニュアル

年月日:

セット

時刻:

☐ このGravio HubをNTPサーバーにする

（注意：NTPを利用して、タイムゾーンのみを変更する場合は、ネットワーク設定が終了したあとに、タイムゾーンの変更をしてください。）

※Gravio Hubがインターネットに接続されていない場合、マニュアル設定時は、外部との時刻同期は行われません。

地域設定

地域設定

地域

Japan

セット

ホスト名の設定

ホスト名

ホスト名: gravio

ホスト名の設定

続いて、地域を選択してホスト名の設定でホスト名を設定します。画面では「gravio」と設定するため「gravio.local」でアクセスできるようになります。

Gravio HubのWAN設定

Gravio HubのWAN設定

WAN(LTE)を利用する	☐
APN	セット
WAN接続状態	× 切断
現在のWAN側IPアドレス	IPアドレス:
	デフォルトゲートウェイ:
	デフォルトDNSサーバー:
WAN側着信ポート設定	Gravioコーディネーターポート(443) ☐
	Gravio HubKitポート(29442) ☐
	ノード間通信用MQTTポート(8883) ☐
	Configuration Managerポート(8080) ☐

Gravio Hub2ではLTEを使用した通信（LTEモード）にも対応しています。以下のキャリアのnano SIMをスロットに挿入してWAN設定を有効にすることで直接インターネットに接続することが出来ます。

APN名	キャリア
au	au
BIGLOBE	ビッグロース
DOCOMO	NTTドコモ
IIJmio	IIJ mio
Ipsim	Ipsim
povo	povo
RAKUTEN	楽天モバイル
SOFTBANK	ソフトバンク
SORACOM	ソラコム

WANの設定を有効にするには「WANを利用する」をオンにして再起動します。

再起動後はGravio Hub2のネットワークはwifiルーターモードが有効になり「192.168.96.1」となります。
またDHCPサーバも有効となりますのでDHCPでのIPアドレスの割り振りも「192.168.96.x」で有効となります。
「GravuiHub-XXXXX」にwifiで接続した場合には自動的にIPアドレスが割り振られます。
DHCPを使用しない場合にはGravio Hub2に接続する端末は「192.168.96.x/24」で固定IPアドレスを設定してください。

続いてAPNを選択してセットしてもう一度再起動します。

再起動後に自動的にWANに接続されますが、手動で「切断」「接続」ボタンを押すことで切り替えることも出来ます。WANに接続されたときには現在のWAN側IPアドレスにネットワーク情報が表示されます。

注)有線LANで接続しているネットワーク環境でDHCPを使用してGravui HubがDHCPクライアントとしてIPアドレスが割り振られていた場合、LTEモードをオンにした場合には、有線LAN側のDHCPサーバとGravio HubのDHCPサーバが競合しますので有線LANで接続しているネットワーク環境には接続しなうようにしてください。

WAN側着信ポート設定でGravio HubにWAN側からのアクセスを許可するポートを設定することも出来ます。

サービス名	ポート番号
Gravioコーディネーター	443
Gravio HubKitポート	29442
ノード間通信用MQTTポート	8883
Configuration Managerポート	8080

※Gravio 5.1ではGravio HubKitポートは443を使用するため、443を許可してください。29442を許可する必要はありません。

以上で、WAN設定が完了します。

Gravio Hubのインターネット設定

Gravio Hubのインターネット設定

WiFiネットワーク名	GravioHub-	セット
WiFiパスワード	パスワード 	セット
現在のLAN側IPアドレス	IPアドレス:	

LTEモードがオンの場合にはGravio HubのWiFiネットワーク名 (SSID) とWiFiパスワードを変更することが出来ます。

またLANに接続されている場合にはGravio HubのLAN側IPアドレスが表示されます。

Gravio Hubのインターネット設定

Current: **Access Point Mode**

無線LAN

WiFi

接続するWi-Fi: パスワード: ☒ DHCPを利用 ☐ 静的IPアドレス

静的IPアドレス

IPアドレス: デフォルトゲートウェイ: デフォルトDNSサーバー:

設定する

最後に利用するネットワークを設定をします。(固定IPでの利用を推奨します。)

※有線LAN端子にLANケーブルが接続されている場合には有線LANの設定となります。

※有線LAN端子にLANケーブルが接続されていない場合には無線LANの設定となります。

※有線LANケーブルが接続されている場合に固定IPを設定すると、有線LANが常に利用される設定になります。

(WiFiに固定IPを設定していても、有線LANの固定IP設定が解除されDHCP設定にならないと、WiFiは利用できません。)

固定IPアドレスを使用する場合には DHCPを利用をオフにして静的 IP アドレス に固定IPアドレス、デフォルトゲートウェイ、DNSサーバを入力をします。

以上で、設定が完了しますので 設定するボタンを押します。

ネットワーク設定

これで、Wifi設定が適用されますので、設定したWifiネットワークにPCを接続する必要があります。 gravio.local:8080

キャンセル

実行

無線LANならびに有線LANの設定が完了するとメッセージダイアログが表示されます。

SSHの利用

SSHの利用

SSHを利用可能にする



Gravio HubにSSHで接続する場合にSSHの利用をオンにします。(オンにした場合にはGravio Hubの再起動が必要です)

自動停止設定

自動停止設定

自動停止を設定する



停止時刻

時刻:

Gravio Hubを設定時間に自動的にシャットダウン処理を行い電源を切る設定を行います。工場などで毎日主電源を落とす運用をしている場合などに使用します。

無線LANをご使用の場合にはWiFiの接続先をセットしなおしてから、ブラウザで先ほど設定した「<Hostname>.local」と入力します。

有線LANをご使用の場合にはブラウザで先ほど設定した「<Hostname>.local」と入力します。

以上で設置は完了です。

4.1.4. 初期設定(Gravioコンフィグレーションマネージャー)

Gravio Hubにブラウザを使用してコンフィグレーションマネージャーを開きます。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語

ログイン

ブラウザで、前章で設定した「<Hostname>.local:8080」と入力します。

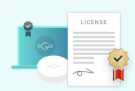


Gravioコンフィグレーションマネージャー

初期設定Gravio設定Gravio Hub設定

Gravio 

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況

未取得

ライセンスキーをインポート

インターネット経由でライセンスキーを取得

ログインすると Gravio Hub設定 の画面が開きますので初期メニューを開きます。

Copyright © 2023 ASTERIA Corporation

ページ 31 / 633

ライセンスを取得

Gravioを使用するには最初に必ず、ライセンスの取得が必要です。(Gravio Freeのサービスをお使いの方もライセンスファイルの取得をお願いします。)



ライセンスキーをインポート

Gravio.comで予めライセンスを取得する時にHubKit IDと合わせて「インターネット環境以外での使用」を選択してライセンスファイルをダウンロードした場合に、そのライセンスファイルをインポートすることでGravio Hub2がインターネットに接続していない環境でもライセンスの登録を行うことができます。

インターネット経由でライセンスキーを取得

Gravioユーザーに紐付いているライセンスをGravio.comから取得してHubKitにインストールします。

Gravio.com 接続情報

メールアドレス

Email Address

メールアドレスの認証ができませんでした

パスワード

Password

キャンセル

接続

ライセンスを取得ボタンを押すと Gravio.com への接続情報パネルが開きますので Gravioユーザー情報を入力します。

ライセンス契約の選択

Gravio Business

v5.0

Main License

Node License

Gravio Business Pro

v5.0

Main License

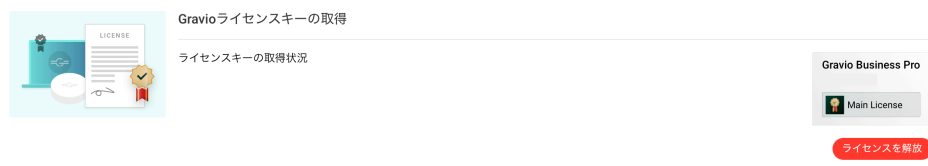
Node License

キャンセル

ライセンスを取得

紐付いているライセンスが表示されましたら選択して取得してください。

Coordinatorを使用するにはMainライセンス、HubKitを使用する場合にはNodeライセンスを取得します。



取得したライセンスが表示されます。解放するにはライセンスを解放を選択します。

Gravioノード登録



HubKitをCoordinatorに登録します。Coordinatorのサーバーアドレスを入力して登録ボタンを押します。CoordinatorでのHubKitの登録を完了するには、Coordinatorでの承認が必要となります。

Coordinator の初期設定



Coordinator アカウントを作成します。

Gravioコーディネーターの初期設定

✕

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

管理者アカウントの作成

アカウント情報を入力して管理者アカウントの作成ボタンを押すとCoordinator アカウントが作成されます。Coordinatorにはこのアカウントを使用してログインすることが出来ます。



Gravioコーディネーターの初期設定

Gravioコーディネーターアカウント設定

このマシン内のGravioコーディネーターを利用する場合は、右のボタンの「初期設定」を実行してください。

[リセット](#)

作成したアカウントを初期化する場合にはリセットボタンを押します。

リセット

✕

この操作を行うとコーディネーターのデータベースのデータ削除され、初期化されます。必要に応じてコーディネーターのバックアップをとってから実行してください。実行する場合は入力欄に「reset」と入力してください。

キャンセル

リセット

確認パネルが表示されますので確認して初期化することが出来ます。

Gravioコンフィグレーションマネージャーのパスワード変更



Gravioコンフィグレーションマネージャーのパスワード変更

パスワード

現在のパスワード:	パスワード	👁
新しいパスワード(8文字以上):	パスワード	👁
新しいパスワード(再入力):	パスワード	👁

[パスワードの変更](#)

コンフィグレーションマネージャーのパスワードを変更します。次回のログインから有効となります。

4.1.5. その他の設定

4.1.5.1. Gravio設定

Gravio Hubにブラウザを使用してコンフィグレーションマネージャーを開きます。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語 

ログイン

ブラウザで、前章で設定した「<Hostname>.local:8080」と入力します。



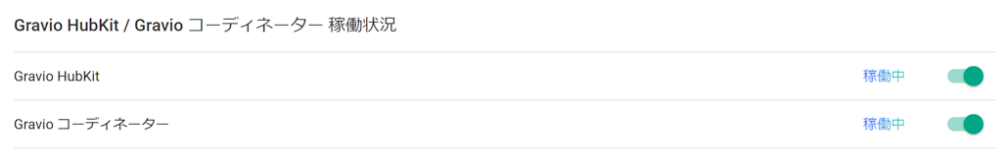
ログインすると Gravio Hub設定 の画面が開きますのでGravio設定メニューを開きます。
※HubKit Windows/Macではこのメニューは表示されません。

HubKit情報

HubKitのHubKitIDを確認することが出来ます。
コピーボタンを押すとクリップボードにコピーされますのでGravio.comでHubKitIDを入力するときに使用することが出来ます。

Gravio HubKit / Gravioコーディネーター 稼働状況

HubKitとコーディネーターのサービスの稼働状況を確認することが出来ます。



HubKitとコーディネーターのサービスをそれぞれ起動/停止する場合にスイッチを使用してください。

ソフトウェアアップデート

コーディネーターとHubKitのソフトウェアを更新することが出来ます。

ソフトウェアアップデート

Gravio HubKitソフトウェアのアップデート

Gravio HubKitの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio HubKit バージョン:

5.0.0

インストール

再起動

Gravio コーディネーターソフトウェアのアップデート

コーディネーターの新バージョンが公開されている場合は、バージョンを選択して「更新」ボタンをクリックしてください。

Gravio コーディネーター バージョン:

5.0.0

インストール

再起動

Zigbeeソフトウェアのアップデート

バージョン:

0.10.5.1

インストール

更新するHubKitとコーディネーターのソフトウェアのバージョンを選択してインストールボタンを押します。インストールが完了すると再起動ボタンを押してください。

または更新するZigbeeのファームウェアのバージョンを選択してインストールボタンを押します。インストールが完了するとそのまま使用できます。

コーディネーターの証明書を変更する

コーディネーターには標準で自己証明書がインストールされていますが、独自ドメインの証明書に変更することが出来ます。

あらかじめキーファイルと証明書ファイルを用意します。

証明書設定

SSLを利用



SSL証明書

キーファイル:

ファイルを選択してください

ファイル選択

証明書ファイル:

ファイルを選択してください

ファイル選択

リセット

アップロード

証明書設定でSSLを利用するはオンにしたままでSSL証明書にキーファイルと証明書ファイルをアップロードします。アップロードが終了すると証明書の更新が完了します。

クライアント証明書を利用してGravio Studioの接続を管理する

HubKitに接続するGravio Studioに対してクライアント証明書を設定します。

あらかじめ証明書ファイルを用意します。

クライアント証明書を利用してGravio Studioの接続を管理

クライアント認証を利用する



クライアント証明書

クライアント証明書のファイルをアップロードしてください

アップロード

プロキシ設定

プロキシ設定 (この設定はHubkitの停止時に編集することができます)		編集
HTTPSプロキシを使用する	プロキシURL:	☐
HTTPプロキシを使用する	プロキシURL:	☐

HubKitがインターネットサービスを使用する場合にProxyサーバーを経由する必要がある場合に設定します。

設定ボタンを押すと設定画面が表示されますので、HTTPSとHTTPのプロトコルでそれぞれに使用するProxyサーバーのアカウント情報（プロキシURL、アカウント、パスワード）を入力します。

プロキシ設定

HTTPSプロキシを使用する ☐

プロキシURL:

アカウント:

パスワード:

HTTPプロキシを使用する ☐

プロキシURL:

アカウント:

パスワード:

キャンセル 保存

※この項目を設定するためにはHubKitは停止している必要があります。

HubKit / コーディネーターデータバックアップ

HubKit / コーディネーターのデータをローカルPCにバックアップし、リストアすることができます。

HubKit / コーディネーターデータバックアップ	
HubKit / コーディネーターデータエクスポート	<button>バックアップ</button> <button>エクスポート</button>
HubKit / コーディネーターデータインポート	インポートファイル: ファイルを選択してください <button>ファイル選択</button> <button>インポート</button>

HubKit / コーディネーターのデータをバックアップするにはバックアップボタンを押します。しばらくするとバックアップが完了しますのでエクスポートボタンを押すことでバックアップしたファイルをローカルPCにダウンロードできます。バックアップしたデータをリストアするにはインポートファイルのファイル選択ボタンを押してファイルを選択します。インポートボタンを押すとしばらくするとファイルがアップロードされて完了します。

※インポートするファイルは必ずバックアップしたファイルを指定してください。

※Gravio HubのZigbeeファームウェアのバージョンはバックアップ元とリストア先で同じバージョンにしてからご利用ください。

Gravio HubKit サポートログ

HubKit と syslogのログファイルをzipファイルでダウンロードすることができます。

Gravio Hub サポートログ	
サポートログ出力	<button>エクスポート</button>

ログファイルをダウンロードするにはエクスポートボタンを押します。

Gravio Hub パワーマネージメント

Gravio Hubの電源をオフにします。

Gravio Hub パワーマネジメント/初期化	
電源を切る	<button>電源を切る</button>
再起動	<button>再起動</button>
初期化	<button>初期化</button>

電源を切るボタンを押すと自動的にGravio Hubの電源が切れます。
再起動ボタンを押すとGravio Hubは再起動します。

初期化ボタンを押すとGravio Hubは出荷状態に戻り、設定情報ならびにデータが削除されます。

Bluetooth設定

Gravio Bluetooth設定	
アドバタイジングモードの使用	☐
ペアリングされているデバイス	
利用可能なデバイス	<button>スキャン</button>

Gravio Hubで使用するBluetoothデバイス（例えばスピーカーなど）との接続を行います。

最初にBluetoothデバイスとのペアリングを行うためにデバイスをペアリングモードにして「スキャン」ボタンを押します。

しばらく待つと利用可能デバイスにBluetoothデバイスが表示されますので「Pair」ボタンを押します。

ペアリングが終わりましたら「Connect」ボタンでデバイスと接続してください。

アドバタイジングモードをサポートしている「おんどとり(株式会社ティアンドデイ)」のTR7Aシリーズ（TR71A、TR72A、TR72A-S、TR75A）を使用する場合には、「アドバタイジングモードを使用する」をオンにしてください。

※おんどとり（TR7Aシリーズ）はペアリングをする必要はありません。

※おんどとりをUSBで接続する場合はアドバタイジングモードは使用しません。

4.1.5.2. Gravio Hubの運用の注意事項

Gravio Hubの運用にあたっては以下のことを考慮して運用設定してください。

処理性能

Gravio Hubでソフトセンサとしてレイヤーに推論を選択した場合は、1台のGravio Hubでは利用するモデルは1つまでとしてください。

モデルファイルによってメモリとCPUにかかる負荷に差がでます。

通常のTensorflowモデルの場合、接続されるカメラは2台までとして、通常は1分間隔程度での動作を推奨します。

カメラ画像を少なくとも15-20秒の間隔を開けて動作させてください。

目安としては、SSHでGravio Hubに入ってTopコマンドで、表示されるロードアベレージが2以下になるように調整してください。

LiteのTensorflowモデルの場合、接続されるカメラは2台までとして、通常は30秒間隔程度で動作させるを推奨します。

目安としては、SSHでGravio Hubに入ってTopコマンドで、表示されるロードアベレージが2以下になるように調整してください。

これを超えるような運用を予定する場合は、処理能力が高い専用PCなどにHubKitをインストールしてご利用ください。

ディスク容量

Gravio Hubのディスク容量は、OSも含めて16GBのディスク容量となっています。

通常のワークエリアとして利用できる容量はおよそ6GBです。

推論時に画像を出力設定している場合は、容量が溢れないように注意してください。

4.2. Windows版のインストールと初期設定

HubKit Windows版はNodeハブとして運用できます。

設定にはGravioコンフィグレーションマネージャーの画面でブラウザから行います。

1. [初期設定](#) 画面でGravioのライセンスを取得する
 - このNodeハブを [Gravioコーディネーターの管理下に登録する](#)

これで、動作環境の設定は完了です。

4.2.1. Windows版のインストール

Windowsを利用してGravioを使う

Gravioの利用には、2つのソフトウェアのインストールが必要です。
Gravio Studio for WindowsとHubKitの2つです。

Gravio Studio for WindowsとHubKitのインストール

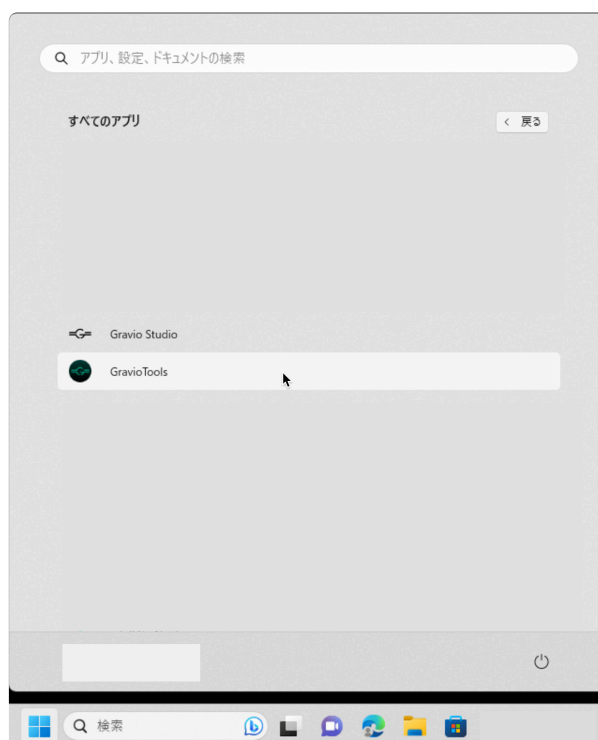
- Gravio Studioを [Microsoftストア](#) からインストールします。
- Gravio Studioを起動します。
- まだアカウントをお持ちでない場合は、このフォームに記入して利用規約に同意してアカウントを作成してください。



- 既にアカウントをお持ちの場合は、ログインしてください。
- サインアップした場合は、メールにて認証コードが送られます。電子メールを確認してください。ログイン時に認証コードが求められますので、メールに記述されている認証コードを合わせて入力してください。
- 最初、まだGravioサーバーをインストールしていない場合は、サーバーリストは空です。

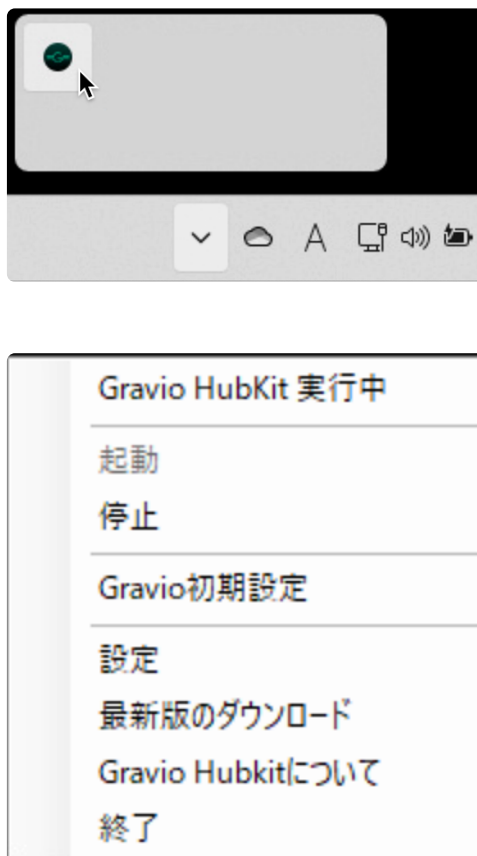


- HubKitをローカルにインストールするには、「HubKitのダウンロード」ボタンをクリックしてください。
- ブラウザが開き、ダウンロードの許可を求める画面が開きます。「GravioHubKit4.msi」をダウンロードし、インストールをしてください。
- インストールが終了するとHubKitは自動的に起動します。



- アプリからGravioToolsが起動するとタスクトレイにGravioToolsが常駐しますので、右クリックでコンテキスト

メニューが表示されます。



インストール後、必ずGravio 初期設定を選択してライセンスのインストールなどの初期設定を行ってください。
初期設定はGravioコンフィグレーションマネージャーというGravioHubに付属しているサービスをブラウザで表示して行います。

GravioコンフィグレーションマネージャーはHubKitをインストールしたPCに常駐するWebアプリケーションです。
そのため、このシステムを安全に管理するために、**Gravioコンフィグレーションマネージャーには専用のアカウントが用意されています。**

Gravio初期設定の手順は [こちら](#) を参照してください。

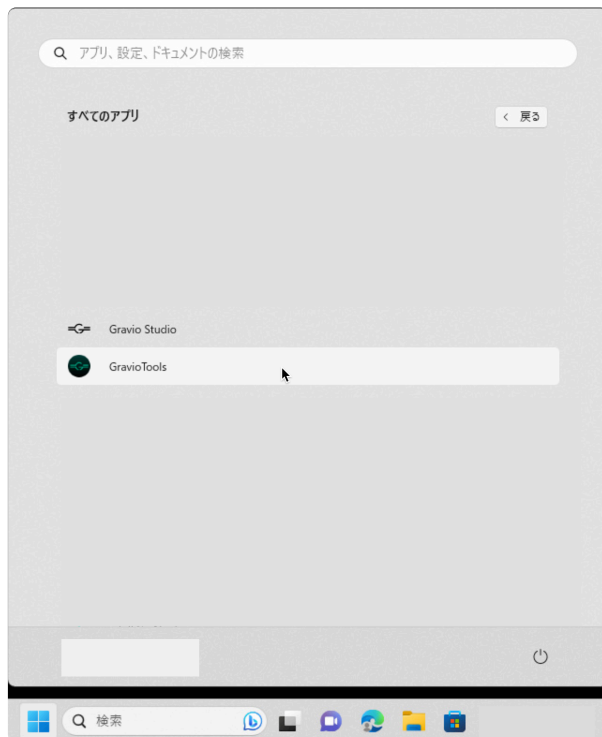
- HubKitのインストールを完了したら、再度Gravio Studioを起動します。Hubリスト上にローカルサーバーが表示されます。
注意:サーバーが異なるネットワークセグメントにある場合、Gravioは自動的にそれらを取得することはできません。この場合、IPアドレスを使用して新しいサーバーを手動で追加するには、「+」マークを使用します。

*これで、Gravio StudioとHubKitのインストールは完了です。

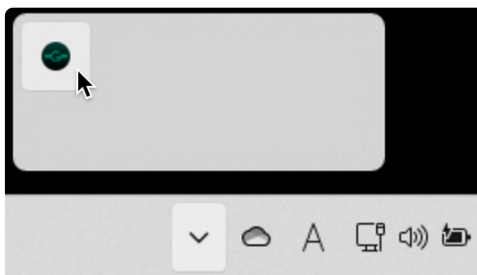
4.2.2. 初期設定(Gravioコンフィグレーションマネージャー)

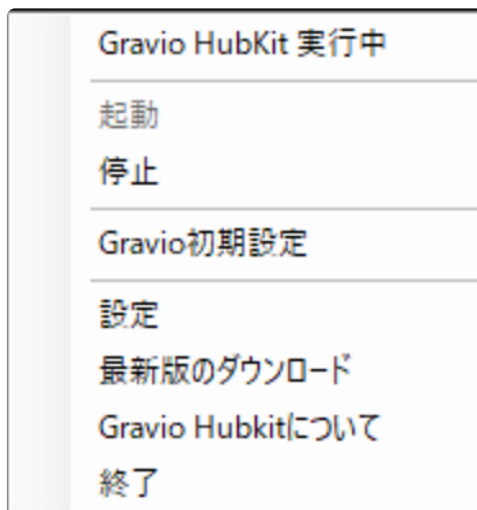
WindowsのタスクトレイからGravio Toolsを利用してGravioコンフィグレーションマネージャーの画面をブラウザで開きます。

- アプリからGravioToolsを起動します。



- GravioToolsが起動するとタスクトレイにGravioToolsが常駐しますので、右クリックでコンテキストメニューを表示します。





コンテキストメニューから Gravio 初期設定を選択します。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語 

ログイン

注: ブラウザで「<インストールしたPCのIP Address>.local:8080」をURLとして入力することでもこの開くこと

ができます。



ログインすると Gravio Hub設定 の画面が開きますので初期メニューを開きます。

ライセンスを取得

Gravioを使用するには最初に必ず、ライセンスの取得が必要です。(Gravio Freeのサービスをお使いの方もライセンスファイルの取得をお願いします。)



ライセンスキーをインポート

Gravio.comで予めライセンスを取得する時にHubKit IDと合わせて「インターネット環境以外での使用」を選択してライセンスファイルをダウンロードした場合に、そのライセンスファイルをインポートすることでGravio Hub2がインターネットに接続していない環境でもライセンスの登録を行うことができます。

インターネット経由でライセンスキーを取得

Gravioユーザーに紐付いているライセンスをGravio.comから取得してHubKitにインストールします。

Gravio.com 接続情報

メールアドレス

Email Address

メールアドレスの認証ができませんでした

パスワード

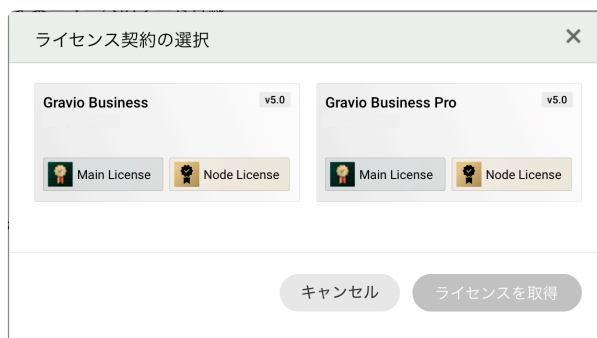
Password

キャンセル

接続

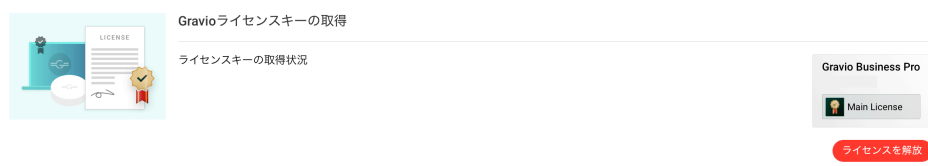
ライセンスを取得ボタンを押すと Gravio.com への接続情報パネルが開きますので **Gravioユーザーアカウントのユーザ名とパスワード** を入力します。

注: GravioユーザとはGravio Studioでサインアップした時のアカウント(もしくはGravio.comで「アカウント作成」を行ったときのアカウント)を指します。



紐付いているライセンスが表示されましたら選択して取得してください。

Windows版はNodeハブでのみ運用可能なため、HubKitを使用する場合にはNodeライセンスを取得します。



取得したライセンスが表示されます。解放するにはライセンスを解放を選択します。

Gravioノード登録



HubKitをCoordinatorに登録します。Coordinatorのサーバーアドレスを入力して登録ボタンを押します。CoordinatorでのHubKitの登録を完了するには、Coordinatorでの承認が必要となります。

Coordinator の初期設定



Coordinator アカウントを作成します。

Gravioコーディネーターの初期設定

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

管理者アカウントの作成

アカウント情報を入力して管理者アカウントの作成ボタンを押すとCoordinator アカウントが作成されます。
Coordinatorにはこのアカウントを使用してログインすることが出来ます。



作成したアカウントを初期化する場合にはリセットボタンを押します。

リセット

×

この操作を行うとコーディネーターのデータベースのデータ削除され、初期化されます。必要に応じてコーディネーターのバックアップをとってから実行してください。実行する場合は入力欄に「reset」と入力してください。

キャンセル

リセット

確認パネルが表示されますので確認して初期化することができます。

Gravioコンフィグレーションマネージャーのパスワード変更



Gravioコンフィグレーションマネージャーのパスワード変更

パスワード

現在のパスワード:

パスワード

👁

新しいパスワード(8文字以上):

パスワード

👁

新しいパスワード(再入力):

パスワード

👁

パスワードの変更

コンフィグレーションマネージャーのパスワードを変更します。次回のログインから有効となります。

Copyright © 2023 ASTERIA Corporation

ページ 53 / 633

4.3. macOS版のインストールと初期設定

HubKit Mac版はNodeハブとして運用できます。

設定にはGravioコンフィグレーションマネージャーの画面でブラウザから行います。

1. [初期設定](#) 画面でGravioのライセンスを取得する
 - このNodeハブを [Gravioコーディネーターの管理下に登録する](#)

これで、動作環境の設定は完了です。

4.3.1. macOS版のインストール

Macを利用してGravioを使う

Gravio利用には、2つのソフトウェアのインストールが必要です。
Gravio Studio for MacとHubKitの2つです。

Gravio Studio for MacとHubKitのインストール

- Gravio Studioを App Storeからインストールします。
- Gravio Studioを起動します。
- まだアカウントをお持ちでない場合は、このフォームに記入して利用規約に同意してアカウントを作成してください。

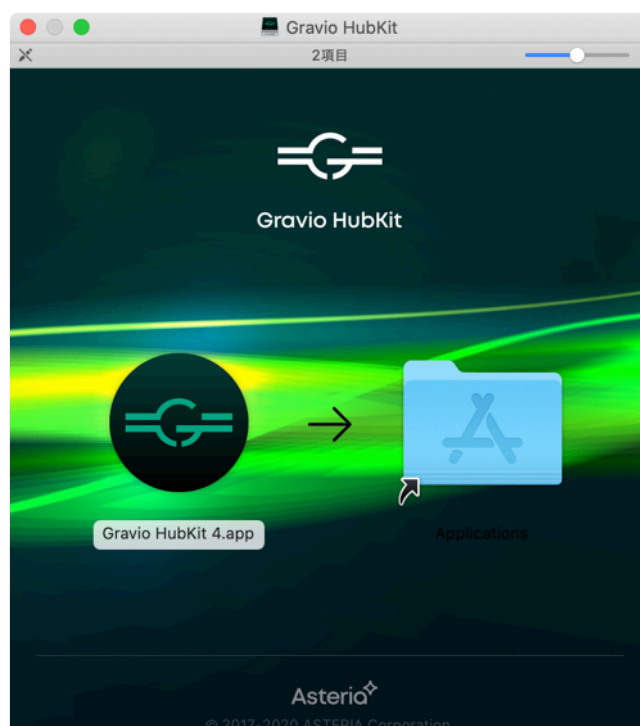


The image shows the Gravio Studio for Mac sign-up interface. On the left is a dark green window titled 'データ処理' (Data Processing) with a graphic of data being processed. On the right is a light gray sign-up form titled 'サインアップ' (Sign Up). The form includes fields for '組織名(会社名)' (Organization Name/Company Name), '姓' (Last Name), '名' (First Name), 'メールアドレス' (Email Address), and 'パスワード' (Password). There is a 'メール言語' (Email Language) dropdown set to '日本語' (Japanese). Below the fields, there is a checkbox for '「サインアップ」をクリックすることで以下に同意したものとみなされます' (By clicking 'Sign Up', you agree to the following) and links for 'Gravio利用規約' (Gravio Terms of Use) and 'プライバシーポリシー' (Privacy Policy). At the bottom are 'サインアップ' (Sign Up) and 'ログイン' (Login) buttons.

- 既にアカウントをお持ちの場合は、ログインしてください。
- サインアップした場合は、メールにて認証コードが送られます。電子メールを確認してください。ログイン時に認証コードが求められますので、メールに記述されている認証コードを合わせて入力してください。
- 最初、まだGravioサーバーをインストールしていない場合は、サーバーリストは空です。



- HubKitをローカルにインストールするには、「Gravio HubKitについて」ボタンをクリックしてください。
- ブラウザが開きますので「プロダクト」のメニューをクリックして表示されるページの下部にあるGravio HubKit for Macのダウンロードボタンをクリックします。
- ダウンロードの許可を求める画面が開きます。「GravioHubKit4.dmg」をダウンロードし、ダブルクリックで開きインストールをしてください。



- インストールが終了するとアプリケーションフォルダにあるGravio HubKit4 をダブルクリックして起動してください。管理者権限のパスワードの入力が求められるのでパスワードを入力してください。
- HubKitが起動するとMac上段メニューにアイコンが表示されます。



- HubKitのインストールを完了したら、再度Gravio Studioを起動します。Hubリスト上にローカルサーバーが表示されます。
Gravio 初期設定を選択してライセンスのインストールなどの初期設定を行ってください。Gravio初期設定の詳細は [こちら](#) を参照してください。

注意:サーバーが異なるネットワークセグメントにある場合、Gravioは自動的にそれらを取得することはできません。この場合、IPアドレスを使用して新しいサーバーを手動で追加するには、「+」マークを使用します。

*これで、Gravio StudioとHubKitのインストールは完了です。

4.3.2. 初期設定(Gravioコンフィグレーションマネージャー)

MacのツールバーからGravio Toolbarアプリを利用してGravioコンフィグレーションマネージャーの画面をブラウザで開きます。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語 

ログイン

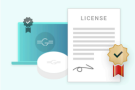
注: ブラウザで「<インストールしたPCのIP Address>.local:8080」をURLとして入力することでもこの開くことができます。

Gravioコンフィグレーションマネージャー

初期設定Gravio設定Gravio Hub設定

Gravio 

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況	未取得
ライセンスキーをインポートインターネット経由でライセンスキーを取得	

ログインすると Gravio Hub設定 の画面が開きますので初期メニューを開きます。

ライセンスを取得

Gravioを使用するには最初に必ず、ライセンスの取得が必要です。(Gravio Freeのサービスをお使いの方もライセンスファイルの取得をお願いします。)



ライセンスキーをインポート

Gravio.comで予めライセンスを取得する時にHubKit IDと合わせて「インターネット環境以外での使用」を選択してライセンスファイルをダウンロードした場合に、そのライセンスファイルをインポートすることでGravio Hub2がインターネットに接続していない環境でもライセンスの登録を行うことができます。

インターネット経由でライセンスキーを取得

Gravioユーザーに紐付いているライセンスをGravio.comから取得してHubKitにインストールします。

Gravio.com 接続情報

メールアドレス

Email Address

メールアドレスの認証ができませんでした

パスワード

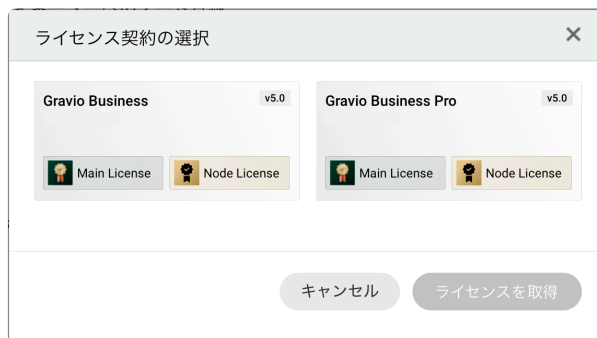
Password

キャンセル

接続

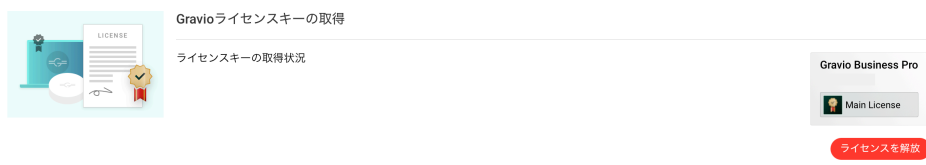
ライセンスを取得ボタンを押すと Gravio.com への接続情報パネルが開きますので [Gravioユーザーアカウントのユーザ名とパスワード](#) を入力します。

注: GravioユーザとはGravio Studioでサインアップした時のアカウント(もしくはGravio.comで「アカウント作成」を行ったときのアカウント)を指します。



紐付いているライセンスが表示されましたら選択して取得してください。

Mac版はNodeハブでのみ運用可能なため、HubKitを使用する場合にはNodeライセンスを取得します。



取得したライセンスが表示されます。解放するにはライセンスを解放を選択します。

Gravioノード登録



HubKitをCoordinatorに登録します。Coordinatorのサーバーアドレスを入力して登録ボタンを押します。CoordinatorでのHubKitの登録を完了するには、Coordinatorでの承認が必要となります。

Coordinator の初期設定



Gravioコーディネーターの初期設定

Gravioコーディネーターアカウント設定
このマシン内のGravioコーディネーターを利用する場合は、右のボタンの「初期設定」を実行してください。

[初期設定](#)

Coordinator アカウントを作成します。

Gravioコーディネーターの初期設定

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

管理者アカウントの作成

アカウント情報を入力して管理者アカウントの作成ボタンを押すとCoordinator アカウントが作成されます。
Coordinatorにはこのアカウントを使用してログインすることが出来ます。



Gravioコーディネーターの初期設定

Gravioコーディネーターアカウント設定
このマシン内のGravioコーディネーターを利用する場合は、右のボタンの「初期設定」を実行してください。

[リセット](#)

作成したアカウントを初期化する場合にはリセットボタンを押します。

リセット

×

この操作を行うとコーディネーターのデータベースのデータ削除され、初期化されます。必要に応じてコーディネーターのバックアップをとってから実行してください。実行する場合は入力欄に「reset」と入力してください。

キャンセル

リセット

確認パネルが表示されますので確認して初期化することができます。

Gravioコンフィグレーションマネージャーのパスワード変更



Gravioコンフィグレーションマネージャーのパスワード変更

パスワード

現在のパスワード:

パスワード

👁

新しいパスワード(8文字以上):

パスワード

👁

新しいパスワード(再入力):

パスワード

👁

パスワードの変更

コンフィグレーションマネージャーのパスワードを変更します。次回のログインから有効となります。

4.4. Linux (Ubuntu AMD64)版のインストールと初期設定

4.4.1. Ubuntu(AMD64)版のインストール

ご利用されるOS向けのインストーラを<https://www.gravio.com/jp/download> からダウンロードしてください。

- SSH を使用してUbuntu 20.04 Server LTSに接続します。(Desktop, Coreはサポートされておりません。Server版をご使用ください)
- ダウンロードしたdebパッケージ(AMD64は setup_-XXXX_amd64.deb)をUbuntuに転送します。
- 以下のCommandをConsoleから実行してください。

```
sudo apt update  
sudo apt upgrade
```

・次に、ダウンロードした、debパッケージをインストールします。

```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

このdebパッケージをインストールすると、以下のLinuxアカウントが作成されます。

username : gravio

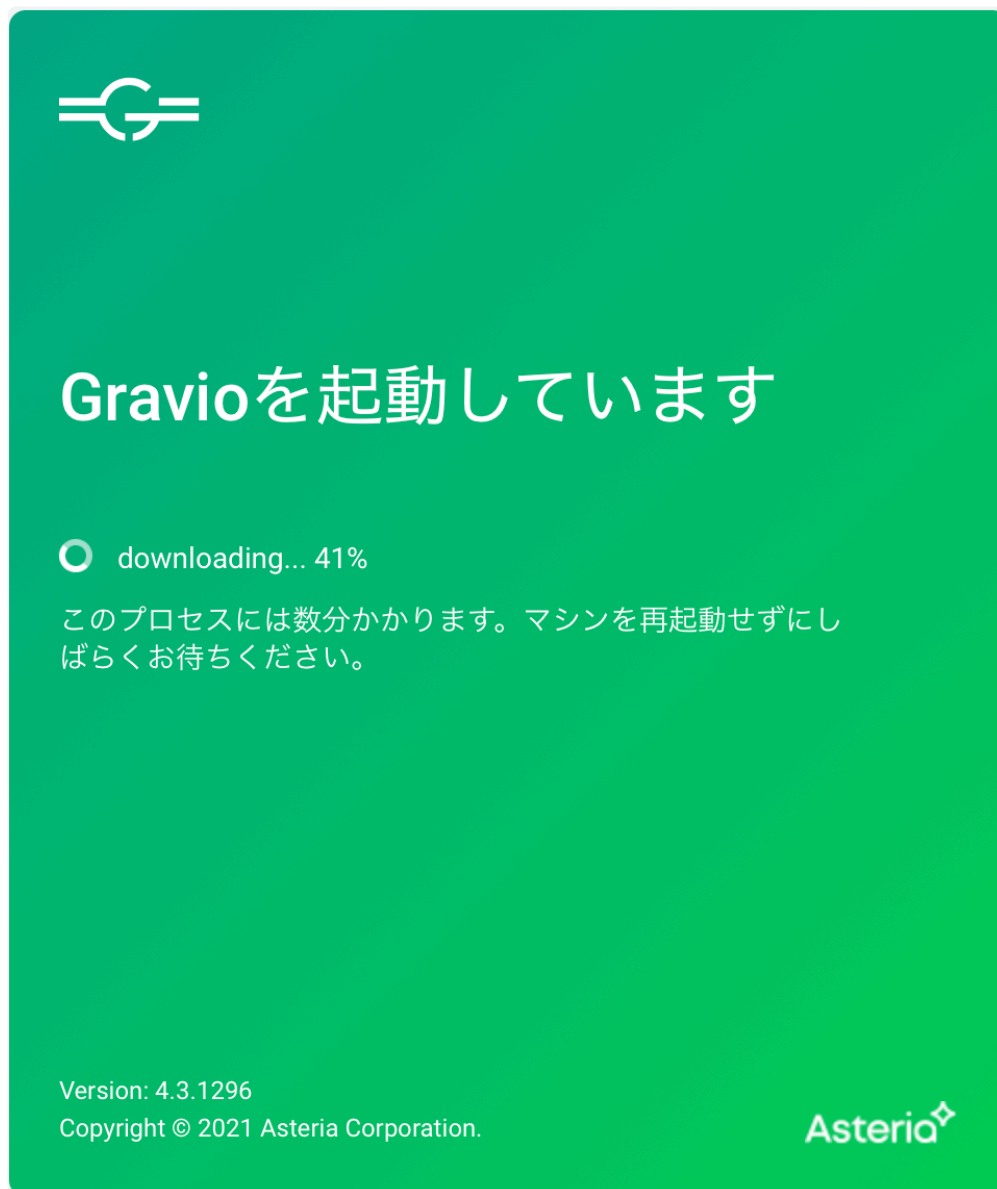
password : gravio

4.4.2. 初期設定(Gravioコンフィグレーションマネージャー)

ブラウザを開いて、Gravioコンフィグレーションマネージャーの画面でログインして、初期設定を行います。

ブラウザを開いてUbuntu 20.04 Server LTSのIPアドレスのポート8080にアクセスします。

`http://(ip address):8080/`



Coordinator, HubKitの初期化が終了するとGravioコンフィグレーションマネージャーのログイン画面が表示されます。

Gravio Hubにブラウザを使用してコンフィグレーションマネージャーを開きます。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語

ログイン

ブラウザで、前章で設定した「<Hostname>.local:8080」と入力します。

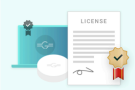


Gravioコンフィグレーションマネージャー

初期設定Gravio設定Gravio Hub設定

Gravio 

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況	未取得
--------------	-----

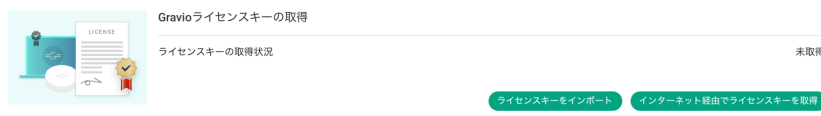
ライセンスキーをインポート

インターネット経由でライセンスキーを取得

ログインすると Gravio Hub設定 の画面が開きますので初期メニューを開きます。

ライセンスを取得

Gravioを使用するには最初に必ず、ライセンスの取得が必要です。(Gravio Freeのサービスをお使いの方もライセンスファイルの取得をお願いします。)



ライセンスキーをインポート

Gravio.comで予めライセンスを取得する時にHubKit IDと合わせて「インターネット環境以外での使用」を選択してライセンスファイルをダウンロードした場合に、そのライセンスファイルをインポートすることでGravio Hub2がインターネットに接続していない環境でもライセンスの登録を行うことができます。

インターネット経由でライセンスキーを取得

Gravioユーザーに紐付いているライセンスをGravio.comから取得してHubKitにインストールします。

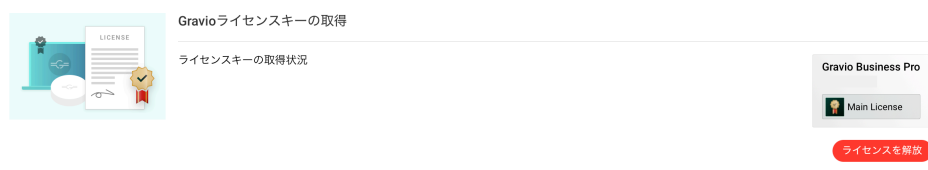
A screenshot of the 'Gravio.com 接続情報' (Gravio.com Connection Information) dialog box. It contains two input fields: 'メールアドレス' (Email Address) and 'パスワード' (Password). The 'Email Address' field has a red border and a red error message below it: 'メールアドレスの認証ができませんでした' (Email address authentication failed). At the bottom, there are two buttons: 'キャンセル' (Cancel) and '接続' (Connect).

ライセンスを取得ボタンを押すと Gravio.com への接続情報パネルが開きますので Gravioユーザー情報を入力します。

A screenshot of the 'ライセンス契約の選択' (License Agreement Selection) dialog box. It shows two license options: 'Gravio Business v5.0' and 'Gravio Business Pro v5.0'. Each option has a 'Main License' and a 'Node License' button. At the bottom, there are two buttons: 'キャンセル' (Cancel) and 'ライセンスを取得' (Acquire License).

紐付いているライセンスが表示されましたら選択して取得してください。

Coordinatorを使用するにはMainライセンス、HubKitを使用する場合にはNodeライセンスを取得します。



取得したライセンスが表示されます。解放するにはライセンスを解放を選択します。

Gravio ノード登録



HubKitをCoordinatorに登録します。Coordinatorのサーバーアドレスを入力して登録ボタンを押します。CoordinatorでのHubKitの登録を完了するには、Coordinatorでの承認が必要となります。

Coordinator の初期設定



Coordinator アカウントを作成します。

Gravioコーディネーターの初期設定

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

管理者アカウントの作成

アカウント情報を入力して管理者アカウントの作成ボタンを押すとCoordinator アカウントが作成されます。Coordinatorにはこのアカウントを使用してログインすることが出来ます。



Gravioコーディネーターの初期設定

Gravioコーディネーターアカウント設定

このマシン内のGravioコーディネーターを利用する場合は、右のボタンの「初期設定」を実行してください。

リセット

作成したアカウントを初期化する場合にはリセットボタンを押します。

リセット

この操作を行うとコーディネーターのデータベースのデータ削除され、初期化されます。必要に応じてコーディネーターのバックアップをとってから実行してください。実行する場合は入力欄に「reset」と入力してください。

キャンセル

リセット

確認パネルが表示されますので確認して初期化することが出来ます。

Gravioコンフィグレーションマネージャーのパスワード変更



Gravioコンフィグレーションマネージャーのパスワード変更

パスワード

現在のパスワード:	パスワード	👁
新しいパスワード(8文字以上):	パスワード	👁
新しいパスワード(再入力):	パスワード	👁

[パスワードの変更](#)

コンフィグレーションマネージャーのパスワードを変更します。次回のログインから有効となります。

4.5. Linux (Ubuntu ARM64/Raspberry Pi)版のインストールと初期設定

4.5.1. Ubuntu(ARM64)版のインストール

ご利用されるOS向けのインストーラを<https://www.gravio.com/jp/download> からダウンロードしてください。

- SSH を使用してUbuntu 20.04 Server LTSに接続します。(Desktop, Coreはサポートされておりません。Server版をご使用ください)
- ダウンロードしたdebパッケージ (ARM64は setup_-XXXX_arm64.deb)をUbuntuに転送します。
- 以下のCommandをConsoleから実行してください。

```
sudo apt update  
sudo apt upgrade
```

・次に、ダウンロードした、debパッケージをインストールします。

```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

このdebパッケージをインストールすると、以下のLinuxアカウントが作成されます。

username : gravio

password : gravio

4.5.2. 初期設定(Gravioコンフィグレーションマネージャー)

ブラウザを開いて、Gravioコンフィグレーションマネージャーの画面でログインして、初期設定を行います。

ブラウザを開いてUbuntu 20.04 Server LTSのIPアドレスのポート8080にアクセスします。

`http://(ip address):8080/`



Coordinator, HubKitの初期化が終了するとGravioコンフィグレーションマネージャーのログイン画面が表示されます。

Gravio Hubにブラウザを使用してコンフィグレーションマネージャーを開きます。



Gravioコンフィグレーションマネージャー

ログイン

アカウント

ユーザー名を空にすることはできません

パスワード



Gravioを利用するための環境設定を行います。

Gravioをインストールしたこの機器を安全に管理するために、Configuration Managerにはアクセス用のアカウントが用意されています。

アカウントの初期設定は以下の通りです

アカウント: gravio
パスワード: gravio

初期設定パスワードは最初のログイン時に変更するダイアログが表示されるので、変更をお願いします。

 日本語

ログイン

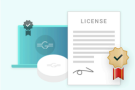
ブラウザで、前章で設定した「<Hostname>.local:8080」と入力します。

Gravioコンフィグレーションマネージャー

初期設定Gravio設定Gravio Hub設定

Gravio 

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況	未取得
--------------	-----

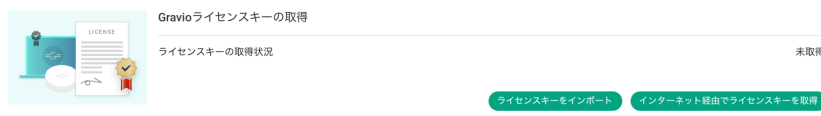
ライセンスキーをインポート

インターネット経由でライセンスキーを取得

ログインすると Gravio Hub設定 の画面が開きますので初期メニューを開きます。

ライセンスを取得

Gravioを使用するには最初に必ず、ライセンスの取得が必要です。(Gravio Freeのサービスをお使いの方もライセンスファイルの取得をお願いします。)



ライセンスキーをインポート

Gravio.comで予めライセンスを取得する時にHubKit IDと合わせて「インターネット環境以外での使用」を選択してライセンスファイルをダウンロードした場合に、そのライセンスファイルをインポートすることでGravio Hub2がインターネットに接続していない環境でもライセンスの登録を行うことができます。

インターネット経由でライセンスキーを取得

Gravioユーザーに紐付いているライセンスをGravio.comから取得してHubKitにインストールします。

Gravio.com 接続情報

メールアドレス

Email Address

メールアドレスの認証ができませんでした

パスワード

Password

キャンセル

接続

ライセンスを取得ボタンを押すと Gravio.com への接続情報パネルが開きますので Gravioユーザー情報を入力します。

ライセンス契約の選択

Gravio Business v5.0

Main License

Node License

Gravio Business Pro v5.0

Main License

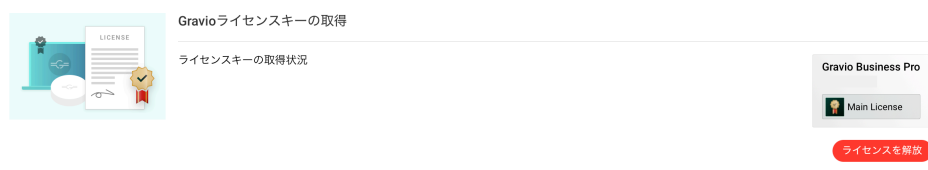
Node License

キャンセル

ライセンスを取得

紐付いているライセンスが表示されましたら選択して取得してください。

Coordinatorを使用するにはMainライセンス、HubKitを使用する場合にはNodeライセンスを取得します。



取得したライセンスが表示されます。解放するにはライセンスを解放を選択します。

Gravio ノード登録



HubKitをCoordinatorに登録します。Coordinatorのサーバーアドレスを入力して登録ボタンを押します。CoordinatorでのHubKitの登録を完了するには、Coordinatorでの承認が必要となります。

Coordinator の初期設定



Coordinator アカウントを作成します。

Gravioコーディネーターの初期設定

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

管理者アカウントの作成

アカウント情報を入力して管理者アカウントの作成ボタンを押すとCoordinator アカウントが作成されます。Coordinatorにはこのアカウントを使用してログインすることが出来ます。



Gravioコーディネーターの初期設定

Gravioコーディネーターアカウント設定

このマシン内のGravioコーディネーターを利用する場合は、右のボタンの「初期設定」を実行してください。

リセット

作成したアカウントを初期化する場合にはリセットボタンを押します。

リセット

この操作を行うとコーディネーターのデータベースのデータ削除され、初期化されます。必要に応じてコーディネーターのバックアップをとってから実行してください。実行する場合は入力欄に「reset」と入力してください。

キャンセル

リセット

確認パネルが表示されますので確認して初期化することが出来ます。

Gravioコンフィグレーションマネージャーのパスワード変更

The image shows a screenshot of the Gravio Configuration Manager password change form. It features a light blue header with a lock icon and a green checkmark. The form is titled 'Gravioコンフィグレーションマネージャーのパスワード変更' and contains three input fields for password entry, each with a toggle icon for visibility. A green button labeled 'パスワードの変更' is at the bottom right.

Gravioコンフィグレーションマネージャーのパスワード変更

パスワード

現在のパスワード:	パスワード	👁
新しいパスワード(8文字以上):	パスワード	👁
新しいパスワード(再入力):	パスワード	👁

[パスワードの変更](#)

コンフィグレーションマネージャーのパスワードを変更します。次回のログインから有効となります。

4.6. 過去のHubKitのインストールとバージョンアップ手順

4.6.1. Linux / Raspberry PiのHubKit 4.2 から4.3へのアップデート手順

HubKitやCoordinatorのソフトウェアがバージョンアップした際に、それらのソフトウェアの新しいバージョンを入れる手順は以下のようにしてください。

※このドキュメントはHubKit 4.2のバージョンを 4.3にアップデートするための手順です。4.2以前のバージョンを4.2にアップデートするには [こちら](#) を参照してください。

HubKit 4.3系とCoordinator 2.6系のアップデートインストール

該当のプラットフォームのサーバーを [ダウンロードサイト](#) からダウンロードしてください。

- SSH を使用してUbuntu 18.04/20.04 LTSに接続します。
- ダウンロードしたdebパッケージ(setup_XXXX.deb)をUbuntuに転送します。
- 以下のCommandをConsoleから実行してください。

```
sudo apt update
sudo apt upgrade
```

- ・実行しているHubKitを停止します。
(ディレクトリは運用に合わせて任意に変更してください)

```
cd hubkit
sudo docker-compose down
```

- ・作成されているgravioユーザーディレクトリを削除します
(ディレクトリを削除する前に内容を確認してください)

```
sudo rm -rf /home/gravio/
```

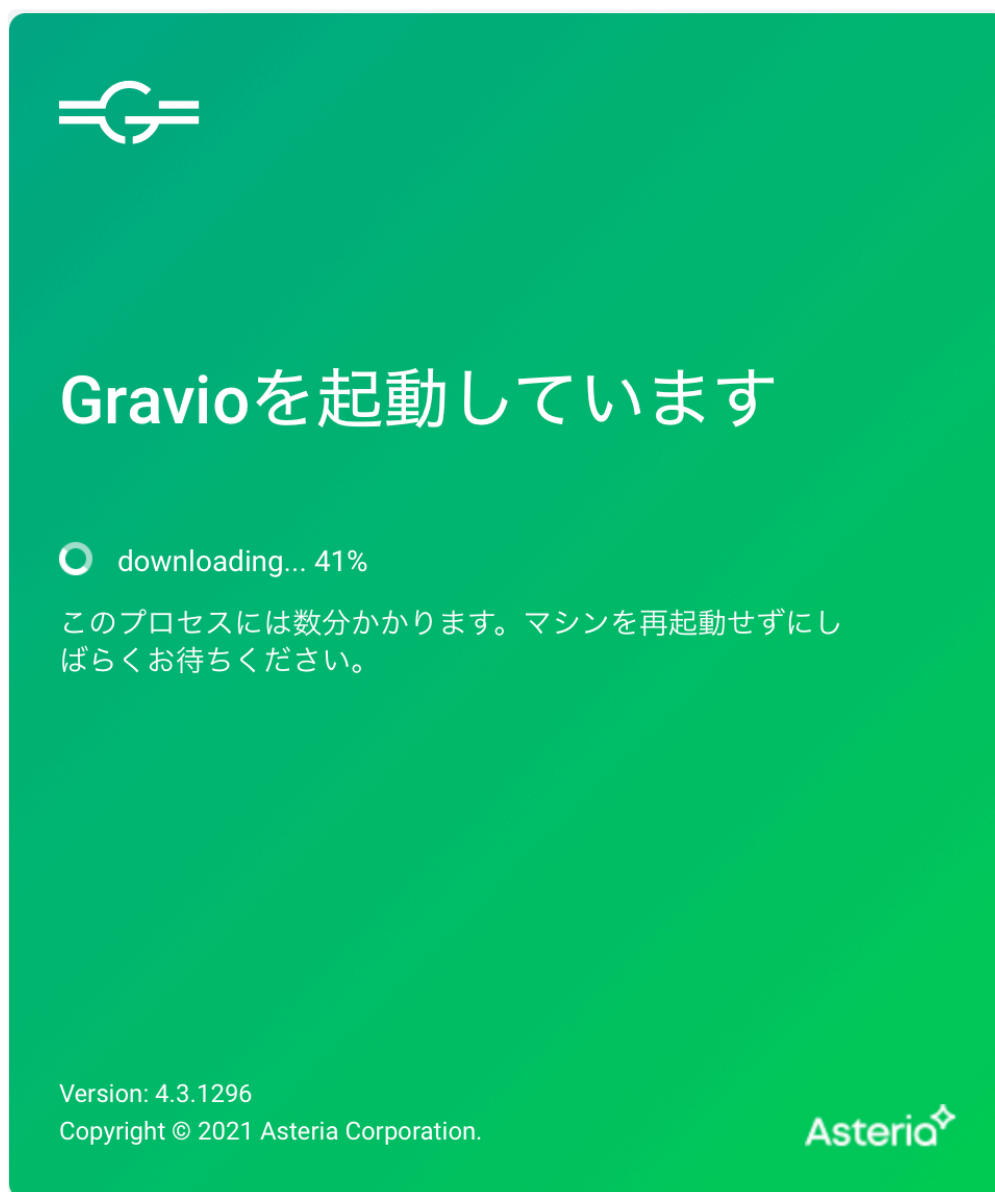
- ・Update Manager, Coordinator, HubKitのインストール

```
sudo apt install -y ./setup_XXXX.deb
```

- ・Update Managerの起動確認

ブラウザを開いてUbuntu 18.04/20.04 LTSのIPアドレスのポート8080にアクセスします。

`http://(ip address):8080/`



Update ManagerがCoordinator, HubKitの初期化が終了するとログイン画面が表示されます。
次のステップに進むのはログイン画面が出るまで待ってください。

- ・実行しているHubKitを停止して Coordinator, HubKitのデータを移行します
(新規インストールしたHubKit 4.3のディレクトリは念のためにhubkitrepo4_bakにバックアップしています)

```
cd /home/gravio
sudo docker-compose down
```

```
sudo mv hubkitrepo4 hubkitrepo4_bak  
sudo mv /home/(HubKit4.2をインストールしているユーザ名)/hubkit/hubkitrepo4 .  
sudo chown -R gravio:gravio hubkitrepo4
```

これで設定は完了です。Ubuntuを再起動してください。

```
sudo reboot
```

・Update Managerへのログイン

Ubuntuを再起動後、ブラウザを開いてUbuntu 18.04/20.04 LTSのIPアドレスのポート8080にアクセスします。

http://(ip address):8080/

Update Managerの使用方法につきましては [こちら](#) を参照してください。

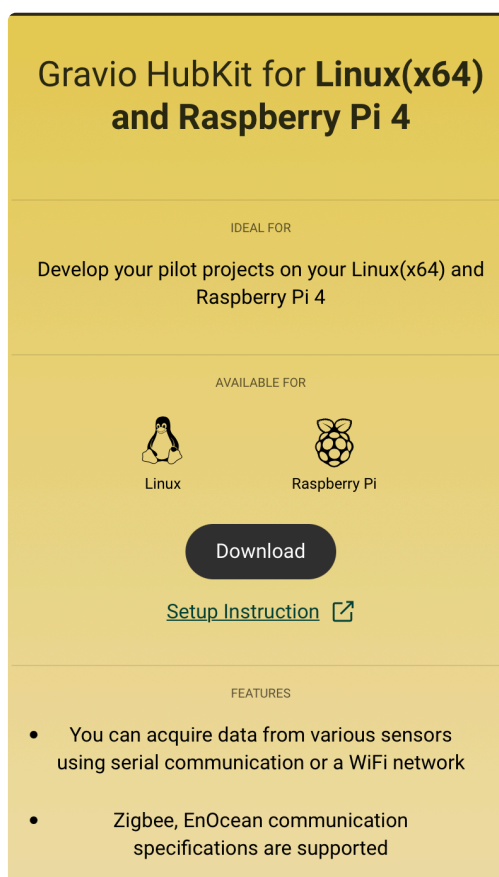
4.6.2. Linux / Raspberry PiのHubKit 4.1から4.2へのアップデート手順

HubKitやCoordinatorのソフトウェアがバージョンアップした際に、それらのソフトウェアの新しいバージョンを入れる手順は以下のようになっています。

※このドキュメントはHubKit を 4.1から4.2にアップデートするための手順です。4.2から4.3にアップデートするには[こちら](#)を参照してください。

<https://www.gravio.com/product>

のサイトからセットアップファイルをダウンロードします。



1.) docker-compose.ymlをいれているディレクトリに移動します。

2.) 以下のコマンドを実行して、HubKitを停止します。

```
sudo docker-compose down
```

3.) すでにインストールされている、docker-compose.ymlをいれているディレクトリにzipを展開して上書きします。

setupcoordinator_hubkit.sh

docker-compose.yml

のファイルが存在するか確認してください。

4.) 次に、".env" ファイルがそのディレクトリに現在まで使っていたファイルが存在します。

それをファイル名を変更するか、削除してください。

```
rm .env
```

5.) 次に、以下のコマンドを実行してください。

```
chmod 755 setupcoordinator_hubkit.sh  
./setupcoordinator_hubkit.sh
```

(新しい".env"ファイルが作成されていることを確認してください。)

これで設定は完了です。

6.) 以下のコマンドを実行して、HubKitを起動します。

```
sudo docker-compose up -d
```

4.6.3. Linux / Raspberry Piのセットアップ 4.3/4.4系

Linux / RaspberryPi

HubKit 4.3/4.4系とCoordinator 2.6/4.4系の新規インストール

該当のプラットフォームのサーバーを [ダウンロードサイト](#) からダウンロードしてください。

- SSH を使用してUbuntu 18.04/20.04 Server LTSに接続します。(Desktop, Coreはサポートされておられません。Serverをご使用ください)
- ダウンロードしたdebパッケージ(AMD64は setup_-XXXX_amd64.deb、ARM64は setup_-XXXX_arm64.deb)をUbuntuに転送します。
- 以下のCommandをConsoleから実行してください。

```
sudo apt update
sudo apt upgrade
```

・Update Manager, Coordinator, HubKitのインストール

```
sudo apt install -y ./setup_-XXXX_XXX64.deb
```

・Update Managerへのログイン

ブラウザを開いてUbuntu 18.04/20.04 Server LTSのIPアドレスのポート8080にアクセスします。

http://(ip address):8080/



Gravioを起動しています

○ downloading... 41%

このプロセスには数分かかります。マシンを再起動せずにしばらくお待ちください。

Version: 4.3.1296

Copyright © 2021 Asteria Corporation.

Asteria[✦]

Update ManagerがCoordinator, HubKitの初期化が終了するとログイン画面が表示されます。Update Managerの使用方法につきましては [こちら](#) を参照してください。

4.6.4. Gravio 4.5でのライセンス管理

Gravio 4.5からStandard契約以上のお客様ではライセンス管理機能が追加されました。
Gravio 4.5以上をご使用の場合には下記の手順でライセンスの登録を行ってください。

※画面上ではサンプルとしてGravio Enterprise AI Edition Platinumの契約内容となっております。
お客様のご契約内容によりGorillaライセンスや他の制限が異なる場合がございます。

ライセンス管理の流れ

Gravio 4.5からHub IDを Gravio.com に登録してライセンスを管理を行います。

- ・Coordinatorをインターネット接続の環境で使用している場合は、
1. Gravio.comでのHubの登録 → 3. Gravio CoordinatorでのHubライセンスの管理 の流れで設定してください。
- ・Coordinatorをオフライン接続の環境で使用している場合は、[こちら](#) を参照してGravioのライセンスファイルを使用してください。
- ・Coordinatorを使用せず、Hubをインターネット接続の環境で稼働している場合は、
1. Gravio.comでのHubの登録 → 4. Gravio StudioでのHubライセンスの管理 の流れで設定してください。
- ・Coordinatorを使用せず、Hubをオフライン接続の環境で稼働している場合は、
1. Gravio.comでのHubの登録 → 2. Hubのライセンスファイルのダウンロード → 4. Gravio StudioでのHubライセンスの管理 の流れで設定してください。

1. Gravio.comでのHubの登録

[Gravio.com](#) にログインして、Hubライセンスを開きます。

Hub IDの登録 ボタンを押して ライセンス登録するHubのHub IDを登録してください。

Hub IDの登録

Hub ID

Hub ID

コメント

Comment

キャンセル

登録

Hub IDはGravio StudioからHubを開き、設定パネルのHubKit 情報に表示されていますので、コピーして登録してください。



登録したHub IDは一覧に表示されます。



一度登録したHub IDを削除する場合には、無効化 ボタンを押してください。

2. Hubのライセンスファイルのダウンロード

Hubがインターネットに接続されている場合、ライセンスはGravio.comに定期的に確認されますが、Hubがオフラインの場合には自動的に確認することが出来ないためライセンスファイルをHubに登録する必要があります。

Hubをオフラインでご使用の場合には、該当するHub IDの LicenseFile.lic をクリックしてライセンスファイルをダウンロードしてください。

3. Gravio CoordinatorでのHubライセンスの管理

Gravio Coordinatorを使用する場合でインターネットに接続されている場合は Gravio.com からライセンスファイルを更新することができます。

 アカウント Hub Hub統計情報 カレンダー 推論モデル **ライセンス**

ライセンス

Gravioライセンス情報 ライセンスファイルのアップロード

このGravio CoordinatorサーバにはGravioエンタープライズライセンスが登録されていません。
以下の日数以内にライセンスファイルをアップロードして設定してください。残り 30 日

Gravio.com 接続情報 切断 設定

アカウントID	--
接続状況	--
Gravio Version	--

最初に Gravio.com 接続情報で設定ボタンを押してアカウントIDとパスワードを登録します。

Gravio.com 接続情報 ×

アカウントID



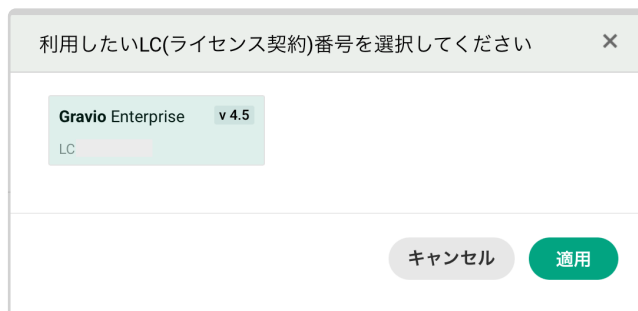
パスワード

Gravio Version



キャンセル 設定

Gravio.com 接続情報の接続状況が 接続OK となると、Gravioライセンス情報のGravio.comからライセンスを更新 ボタンが表示されますので押してください。契約しているライセンスが更表示されますので選択して適用ボタンを押して更新してください。



Gravio.comからライセンスが更新されて表示されます。

ライセンス

Gravioライセンス情報

Gravio.comからライセンスを更新

ライセンスファイルのアップロード

ライセンスをクリア

ライセンス契約ID: LC

登録されているHubID数: 2/4

HubKit登録状況	HubID	ライセンスファイル
● Yes		LicenseFile.lic
● No		LicenseFile.lic

Gravio Coordinatorを使用する場合でインターネットに接続されていないオフライン環境の場合は [こちら](#) を参照してGravioのライセンスファイルをアップロードから更新してください。

4. Gravio StudioでのHubライセンスの管理

Hubをインターネットに接続されている場合は Gravio.com からライセンスファイルを更新することが出来ます。



gravio.comからライセンスを更新 ボタンを押すとライセンスファイルが取得されて表示されます。

HubKit 情報

バックアップ/リストア

機能パッケージ

基本プロパティセット

画像推論モデル

ディスク管理

Hub ID :

HubKit バージョン: 4.5.0.6188

HubKit IP[IPv4] :

HubKit IP[IPv6] :

ライセンス : Gravio Enterprise AI Edition Platinum (有効)

クリア

gravio.comからライセンスを更新

ライセンスファイルをアップロード

Hubをオフライン接続の環境で使用されている場合は 2. Hubのライセンスファイルのダウンロード でダウンロードしたライセンスファイルを ライセンスファイルをアップロードボタンを押して指定します。ライセンスファイルが読み込まれて表示されます。

5. Gravio Cloudの利用

Gravio Business, Gravio Business Proをご契約のお客様はGravio Cloudのサーバをご利用いただけます。

このGravio Cloudサーバはアステリア社がAWSを利用してクラウドサーバをお客様のGravio契約に含まれる **MAINライセンス** を利用することでホスティングサーバをMAINサーバライセンス1台につき1台を利用できるものです。

以下の章の「サーバ構築依頼」の手順にそってサーバを作成してご利用ください。

5.1. サーバ構築依頼

Gravio Cloudをご使用頂くには

Gravio 5.0では、Business, Business Proのライセンスをお持ちの方にはクラウドでGravioを使用するMAINライセンスが付与されています。

Gravio Cloudはお客様の専用サーバーでGravio5.1を実行することが出来ます。

※すでにローカルの環境でMAINライセンスを使用されている場合はライセンスの解放を行って頂くことでクラウドサービスでご利用頂けます。

Gravio Freeのライセンスの方

Gravio Freeのライセンスの方は機能評価ライセンスを申請頂き、そのライセンスを使用してGravio Cloudのトライアルを3週間使用することが出来ます。

機能評価ライセンスはBusiness Proのライセンスを期間限定で評価頂けるライセンスとなっています。Business Proのライセンスで実行できるコンポーネントの詳細は [こちら](#) を参照してください。

Gravio Cloudのサーバーの構築

Gravio Cloudのサーバーの構築する方法につきましては、Business, Business Proのライセンスをお持ちの方は [こちら](#) を、Gravio Freeのライセンスの方は [こちら](#) を参照してください。

5.1.1. Business, Business Proをご購入の方が利用する場合

ライセンスを取得

BusinessまたはBusiness Proのライセンスを以下の手順で取得してください。

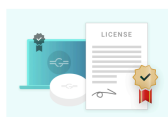
Gravio Hub2の場合はコンフィグレーションマネージャーを開き、初期設定タブからライセンスの取得を行ってください。

HubKitの場合は、それぞれのプラットフォームに合わせて初期設定メニューを開き、ライセンスの取得を行ってください。

※Gravio Hub2のコンフィグレーションマネージャーの初期設定の方法は [こちら](#) を参照してください。

※HubKitのコンフィグレーションマネージャーの初期設定の方法は [Windows](#) [macOS](#) [Linux \(AMD64\)](#) [Linux \(ARM64\)](#) のそれぞれを参照してください。

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況 未取得

[ライセンスキーをインポート](#) [インターネット経由でライセンスキーを取得](#)



Gravioコーディネーターへのノード登録

登録先Gravioコーディネーター Gravioコーディネーターサーバーアドレス: [登録](#)

Gravio.com 接続情報

メールアドレス

Email Address

メールアドレスの認証ができませんでした

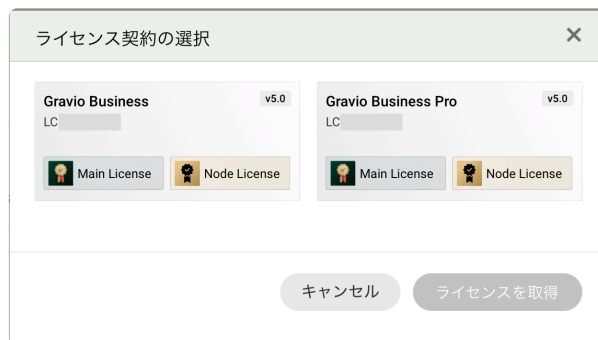
パスワード

Password

キャンセル

接続

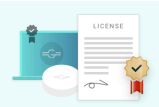
Gravio.comにサインアップしたアカウント情報を入力してください。



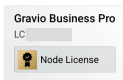
表示されるライセンスの中で Node License を選択して取得してください。

Gravio Cloudサーバーの設置を依頼

初期設定



Gravioライセンスキーの取得
ライセンスキーの取得状況


ライセンスを解放



Gravioコーディネーターへのノード登録
登録先Gravioコーディネーター

Gravioコーディネーターサーバーアドレス:

Gravio Cloudサーバーの設置を依頼する 登録

ライセンスを取得すると、一度ブラウザの画面を再読み込みしてください。そうするとGravio コーディネーターへのノード登録のセクションに Gravio Cloudサーバーの設置を依頼する ボタンが表示されます。

Gravio Cloudトライアル申請

×

サブドメイン

Subdomain

gravio.cloud

メールアドレス

Email Address

組織名

Organization Name

名

First Name

姓

Last Name

パスワード

Password

パスワード(再入力)

Password Confirm

キャンセル

申請

依頼ボタンをクリックすると、依頼に必要な情報の入力パネルが表示されますので入力して申請ボタンを押してください。

申請を受付ると登録頂いているメールアドレスに申請受付のメールが配信されます。

Gravio Cloudサーバーのご利用開始

サーバー構築の作業が完了するとサーバー情報についてのメールが配信されます。構築完了のメールを受信しましたら、依頼に必要な情報の入力パネルに登録したホスト名でアクセスすることが出来ます。

Gravio Studioからの接続

Gravio CloudサーバーにGravio Studioから接続する場合には、必ず Coordinatorへの接続方法で接続してください。

Gravio Studioを起動して「Gravio Coordinatorへログイン」からログイン画面を表示して、CoordinatorのURL、メールアドレスとパスワードを入力してログインします。

※CoordinatorのURLは、Gravio Cloudサーバーのホスト名となります。(https://{Gravio Cloudサーバーのホスト名})

※メールアドレスとパスワードは、依頼に必要な情報の入力パネルに登録したメールアドレスとパスワードになります。Gravio コーディネーターのWebUIにアクセスするときにもこちらのメールアドレスとパスワードをご使用ください。

5.1.2. Gravio Freeの方が評価でサーバを利用する場合

ライセンスを取得

Gravio Freeのライセンスを以下の手順で取得してください。

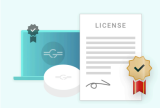
Gravio Hub2の場合はコンフィグレーションマネージャーを開き、初期設定タブからライセンスの取得を行ってください。

HubKitの場合は、それぞれのプラットフォームに合わせて初期設定メニューを開き、ライセンスの取得を行ってください。

※Gravio Hub2のコンフィグレーションマネージャーの初期設定の方法は [こちら](#) を参照してください。

※HubKitのコンフィグレーションマネージャーの初期設定の方法は [Windows](#) [macOS](#) [Linux \(AMD64\)](#) [Linux \(ARM64\)](#) のそれぞれを参照してください。

初期設定



Gravioライセンスキーの取得

ライセンスキーの取得状況

未取得

ライセンスキーをインポート

インターネット経由でライセンスキーを取得



Gravioコーディネーターへのノード登録

登録先Gravioコーディネーター

Gravioコーディネーターサーバーアドレス:

登録

Gravio.com 接続情報

✕

メールアドレス

メールアドレスの認証ができませんでした

パスワード

キャンセル

接続

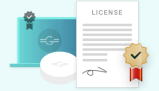
Gravio.comにサインアップしたアカウント情報を入力してください。



表示されるライセンスの Node License を選択して取得してください。

Gravio Cloudサーバーの設置を依頼

初期設定



Gravio ライセンスキーの取得
ライセンスキーの取得状況

Gravio Free
LC:

Node License

[機能評価ライセンスの申請](#) [ライセンスを解放](#)



Gravio コーディネーターへのノード登録
登録先 Gravio コーディネーター

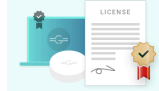
Gravio コーディネーターサーバーアドレス:

[登録](#)

ライセンスを取得した後、一度ブラウザの画面を再読み込みしてください。そうすると機能評価ライセンスの申請ボタンが表示されますので、申請してください。

※機能評価ライセンスは3週間の期間でBusiness Proの機能をご評価頂けます。

初期設定



Gravio ライセンスキーの取得
ライセンスキーの取得状況

Gravio Evaluation
LC:

Node License

[ライセンスを解放](#)

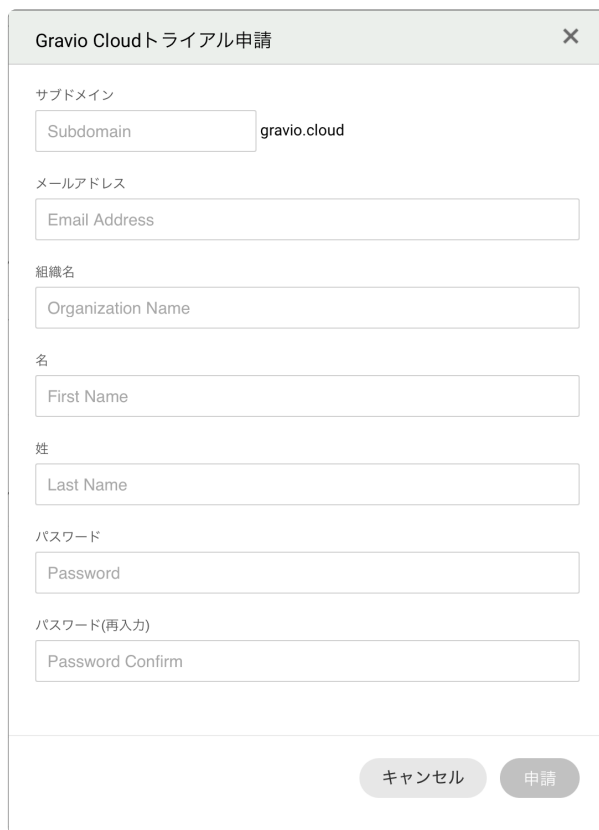


Gravio コーディネーターへのノード登録
登録先 Gravio コーディネーター

Gravio コーディネーターサーバーアドレス:

[Gravio Cloudのトライアルを申請](#) [登録](#)

機能評価ライセンスの申請が完了した後、もう一度ブラウザの画面を再読み込みしてください。Gravio コーディネーターへのノード登録のセクションに Gravio Cloudサーバーの設置を依頼する ボタンが表示されます。

A screenshot of a web form titled "Gravio Cloudトライアル申請" (Gravio Cloud Trial Application). The form contains several input fields: "サブドメイン" (Subdomain) with a text box containing "gravio.cloud", "メールアドレス" (Email Address) with a text box containing "Email Address", "組織名" (Organization Name) with a text box containing "Organization Name", "名" (First Name) with a text box containing "First Name", "姓" (Last Name) with a text box containing "Last Name", "パスワード" (Password) with a text box containing "Password", and "パスワード(再入力)" (Password Confirm) with a text box containing "Password Confirm". At the bottom right, there are two buttons: "キャンセル" (Cancel) and "申請" (Apply).

依頼ボタンをクリックすると、依頼に必要な情報の入力パネルが表示されますので入力して申請ボタンを押してください。

申請を受付ると登録頂いているメールアドレスに申請受付のメールが配信されます。

Gravio Cloudサーバーのご利用開始

サーバー構築の作業が完了するとサーバー情報についてのメールが配信されます。構築完了のメールを受信しましたら、依頼に必要な情報の入力パネルに登録したホスト名でアクセスすることが出来ます。

Gravio Studioからの接続

Gravio CloudサーバーにGravio Studioから接続する場合には、必ず Coordinatorへの接続方法で接続してください。

Gravio Studioを起動して「Gravio Coordinatorへログイン」からログイン画面を表示して、CoordinatorのURL、メールアドレスとパスワードを入力してログインします。

※CoordinatorのURLは、Gravio Cloudサーバーのホスト名となります。(https://{Gravio Cloudサーバーのホスト名})

※メールアドレスとパスワードは、依頼に必要な情報の入力パネルに登録したメールアドレスとパスワードになります。Gravio コーディネーターのWebUIにアクセスするときにもこちらのメールアドレスとパスワードをご使用ください。

5.2. サーバーに関する情報

Gravio Cloudマシンに関する情報

サーバのディスク容量：64GiB (OSシステムを含む)

バックアップ：定期日時にスナップショットによるバックアップ保存がされます。保存期間は7日分です。

6. Gravio運用時のアカウントの種類

Gravioをご使用頂くにあたり、いくつかのアカウントが必要になります。

Gravioのシステムのご利用いただく機能によって、それぞれの機能にアクセスするために幾つかのアカウントの作成が必要になります。

1. Gravio ユーザ アカウント

Gravio Studio でのサインアップ または Gravio.comより [アカウントの作成](#) で作成するアカウントです。

Gravio ユーザーをご登録頂くと、Gravio StudioでGravio HubやHub PCの設定を行うことができます。
ご購入頂いたライセンスは、このGravioユーザーに紐付けられて管理されます。

2. Gravioコンフィグレーションマネージャー アカウント

HubKitパッケージをインストールすると、GravioコンフィグレーションマネージャーというこのHubKitのPC向けの環境設定を行う専用のアカウントがインストール後に自動で用意されます。

ユーザー名は固定で **“gravio”** という名称になります。

初期値のパスワードは**“gravio”**です。最初のログイン時に初期パスワードは変更するように要求されます。

このGravioコンフィグレーションマネージャーアカウントはLinux版およびGravioHubではインストールしたOSに**“gravio”**というLinuxアカウントになります。

3. Gravioコーディネーターアカウント

Gravio コーディネーターを利用することで、Nodeハブにアクセスできるユーザの管理を行うことができます。
Gravioコーディネーターの最初のAdmin権限のアカウントはGravioコンフィグレーションマネージャーの [初期設定](#) で作成することができます。

Gravioコーディネーターは「アカウント」管理では以下のようなユーザ権限のアカウントを作成できます。

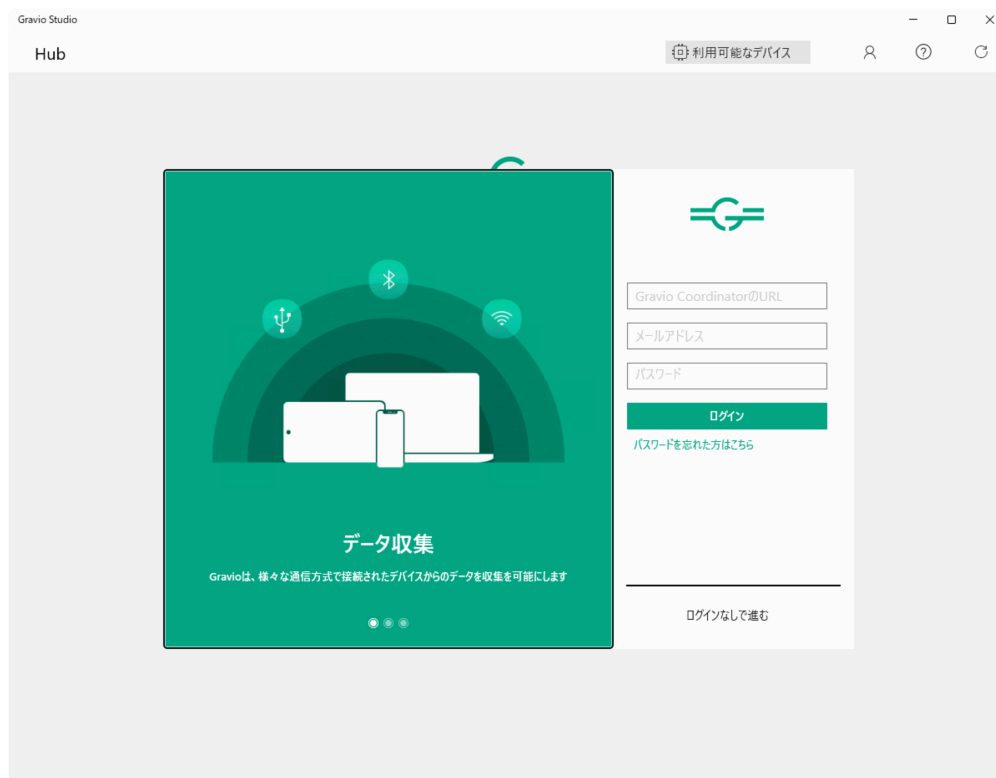
利用できる画面	Admin	Developer	Maintainer	AppUser
アカウント	○	△ (自分のみ)	△ (自分のみ)	△ (自分のみ)
Hub管理	○	○	○	X
統計情報	○	○	○	△ (閲覧のみ)
カレンダー	○	○	○	X
推論モデル	○	○	○	X

7. Gravio Studioの利用

7.1. ログイン

Gravio 5.1からは、Gravio StudioではGravio.comにログインせずに使用出来る「ログインなし」機能が追加されました。

「ログインなし」機能によりインターネットに接続されていない環境でも一部の機能を除いて使用できます。



・ログイン

ログインはコーディネーターにログインする場合に使用します。

コーディネーターのURL、メールアドレスとパスワードを入力してログインします。

・ログインなしで進む

ログインなしで進むは、Gravio.comにログインせずに使用することができます。

設定画面でGravio.comに保存している以下の機能を使用する場合のみGravio.comにログインします。

バックアップ/リストア

機能パッケージ

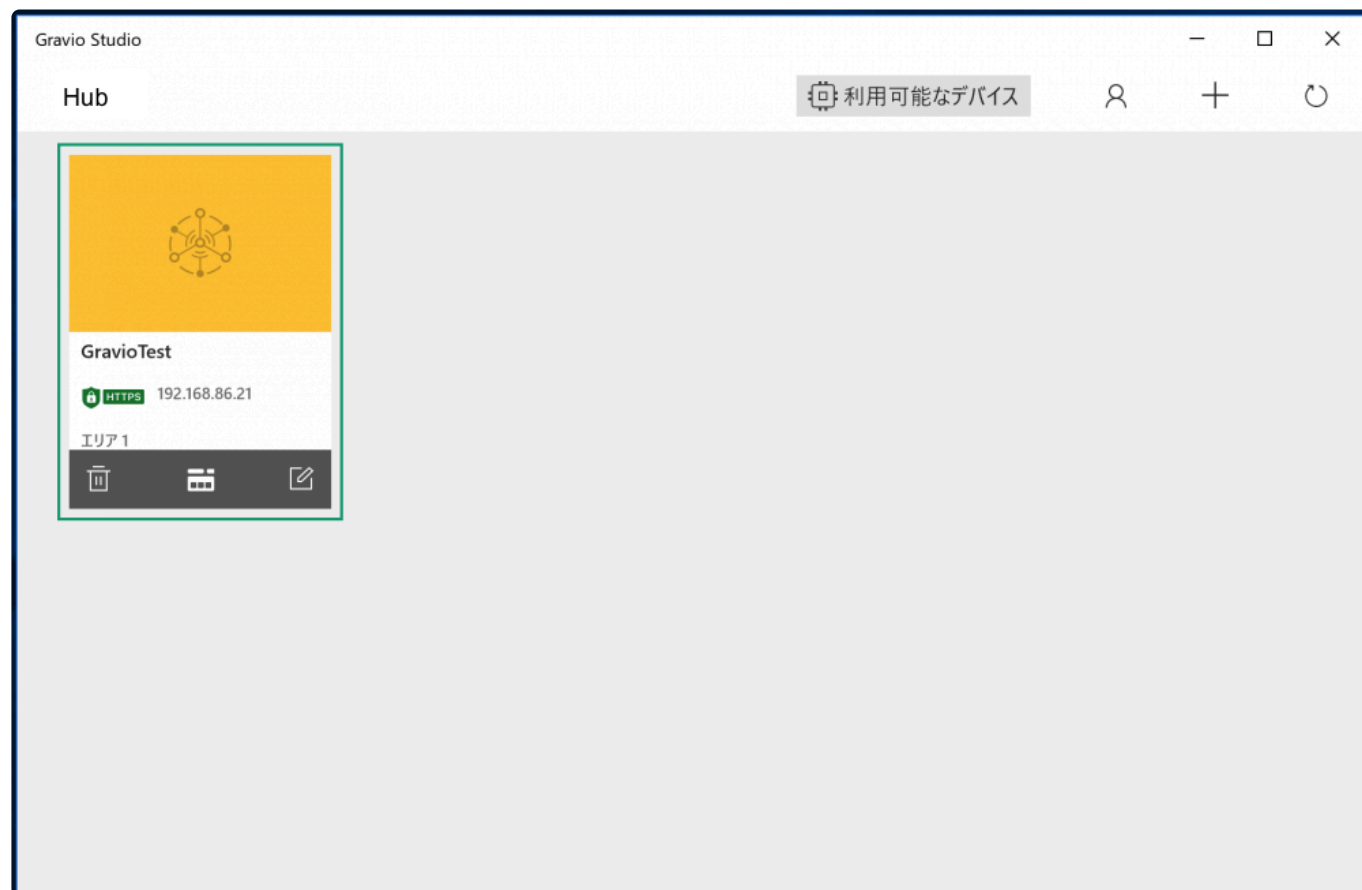
画像推論モデル

これらの情報はGravio.comに保存されているためです。

7.2. Hubリストビュー

GravioのHubリストビューでは、接続されているすべてのGravio HubKit サーバの概要を確認できます。Hubの設定方法については、[インストールとセットアップ](#) をご覧ください。

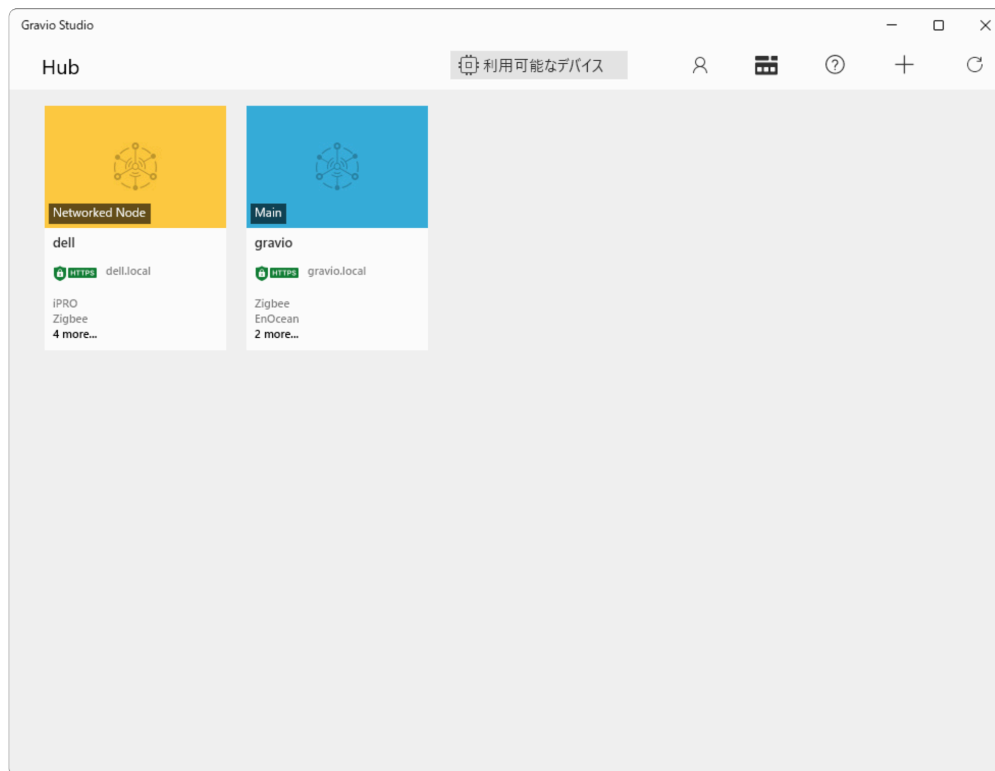
各サーバーには3つのビューがあります。



- センサーの設定と接続を行うサーバー構成ビュー
- アクション設定ビューでは、センサーによってトリガーされるアクションとステップを設定します
- 表示されるサーバー、IPアドレス、またはイメージを設定するサーバー設定ビュー
- サーバー構成ビューに入るには、上半分、つまりカードのメインイメージをダブルクリックします。
- アクション構成ビューに入るには、カードの下端にある中央のアイコンをダブルクリックします。
- サーバー設定ビューに入るには、カードの下端にある右側のアイコンをダブルクリックします。

Gravioコーディネーターへの接続状態の表示

Gravio Studio 5.1ではHubがGravioコーディネーターに接続されている状態が表示されます。



接続状態

状態	説明
Main	MAINライセンスが登録されている
Networked Node	NODEライセンスが登録されており、Gravioコーディネーターに登録されている
Standalone	NODEライセンスが登録されているが、Gravioコーディネーターに登録されていない。またはFreeライセンスが登録されている
(表示なし)	ライセンスが登録されていないHubか、Hubが電源が投入されていないなどで検出されない

Hubへの接続方法

Gravio Studio 5.1ではHubのバージョンが5.1以上と5.0以下では接続に使用するポートが異なるため、Hubにどちらのポートを使用して接続するかを設定を行います。

Hubの編集

✕

Hub 名	gravio
Hub IP	gravio.local
HubKitバージョン5.1以上	☒
ユーザー名	ユーザー名
パスワード	パスワード

カバー画像





写真を撮る

画像選択

保存

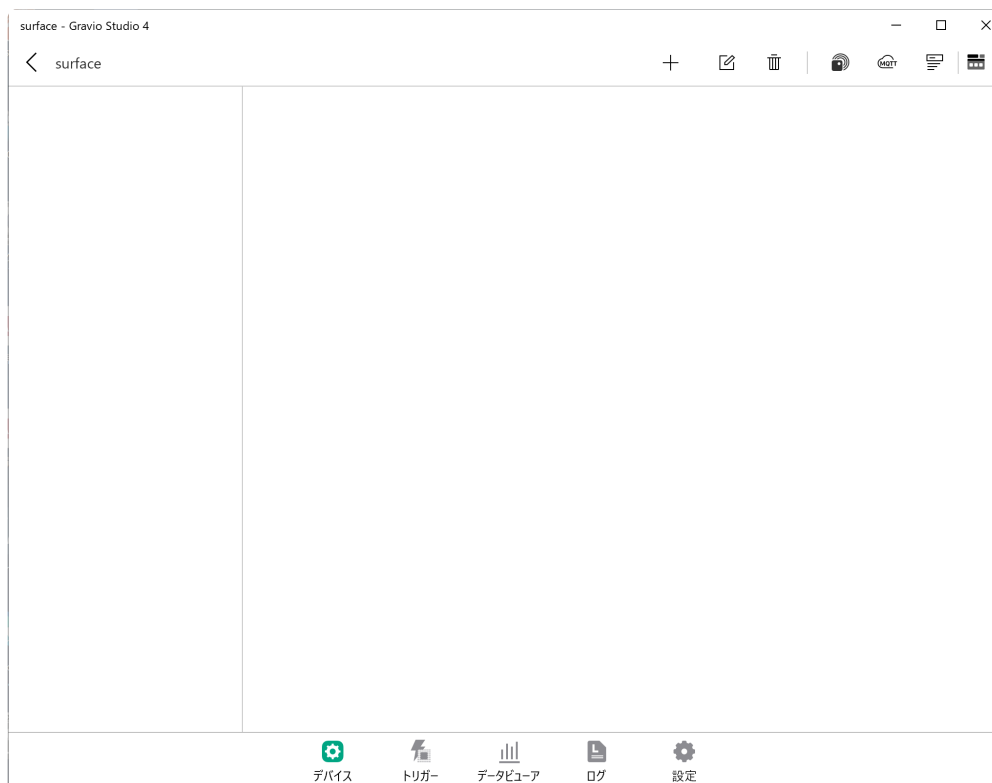
Hubの設定パネルを表示し、5.1以上として接続する場合には「HubKitバージョン5.1以上」をオンにします。5.0以下として接続する場合にはオフにして保存します。

MAINライセンスが登録されているか、またはNODEライセンスが登録されておりGravioコーディネーターに登録されているHubは「ログインなし」で使用する場合でもコーディネーターへの接続は必須となりますので、コーディネーターへログインするユーザー名とパスワードも併せて入力してください。

Hubのバージョンによる使用ポートの詳細は [こちら](#) を参照してください。

7.3. Hub構成ビュー

1つのHubをダブルクリックすると、Hub構成ビューが表示されます。



下部には5つのタブを含むタブバーがあります。

#デバイス – センサーを設定して、センサーをエリアやレイヤーに配置します。

#トリガー – アクションを実行するトリガーを設定する場所です。

#データビューア – 現在のセンサーデータを見つけることができます。センサーの値はこのタブに表示されます。

#ログ – デバッグと最適化のためのGravio Serverログを見つけることができます。

#設定 – サーバー関連の設定を見つけてバックアップを実行できます。

7.3.1. デバイスタブ

[デバイス]タブでは、物理的な場所を含めて、サーバーのデバイスを追加、削除、および構成することによって、サーバーのデバイスを管理します。

Gravioは、データストアを組織化し効率的に保つために、センサーの配置と割り当ての2つの次元を特長としています。

#エリア

#レイヤー

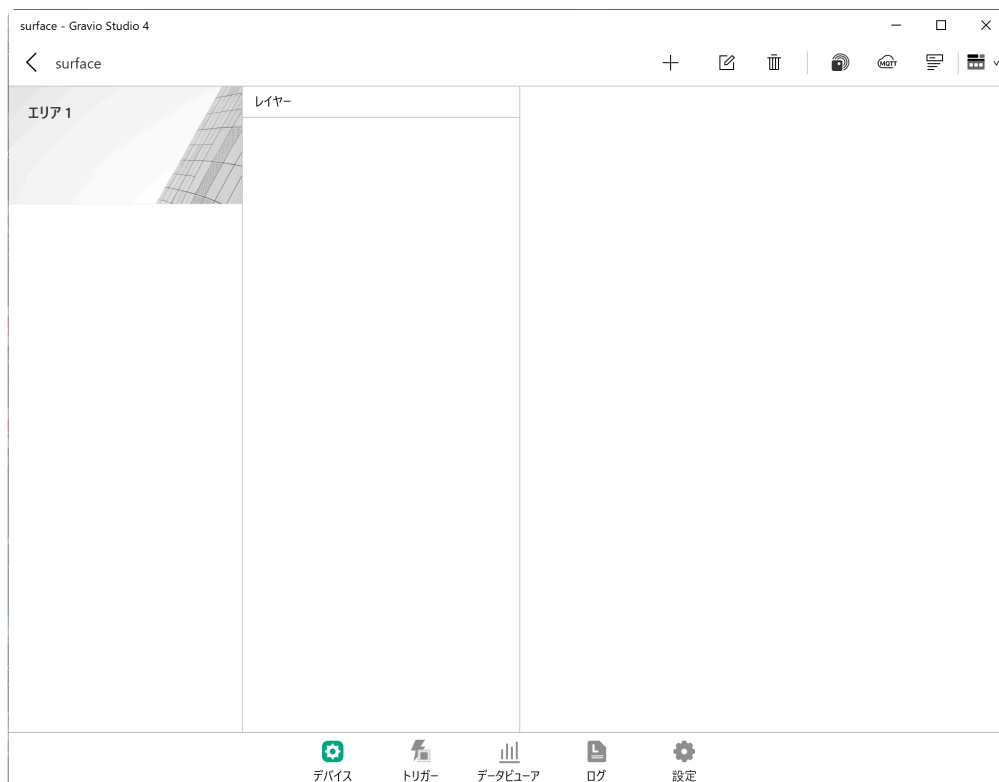
サーバーにエリアを追加する

*エリア*は、例えば「3階西」、「セミナールーム」、「入口エリア」などの物理的な場所を定義します。

エリアは、複数の*レイヤー*を有することができ、センサー情報をそれらの機能性（例えば「温度レイヤー」または「カメラデバイスレイヤー」など）に従って分類して定義されます。1つのレイヤーに1つのセンサーデバイスタイプが含まれている必要があります。

エリアを追加するのは簡単です。右上の「+」マークをクリックするだけで、新しいエリアを追加できます。

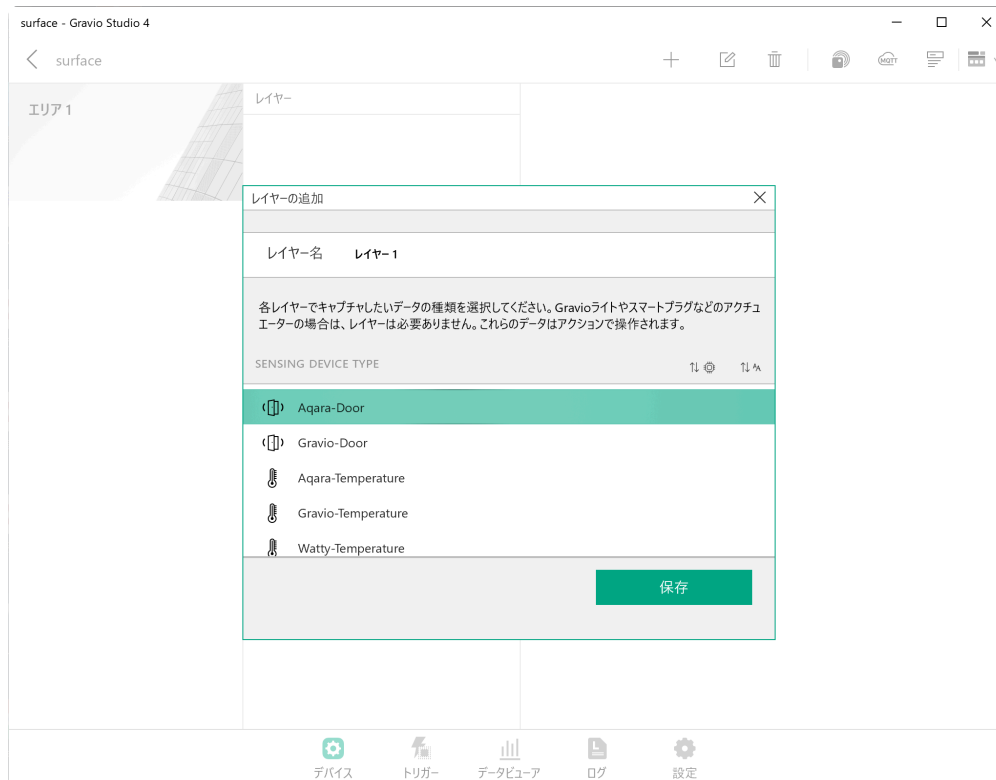
エリアを簡単に識別できるように写真を追加することをお勧めします。



エリアにレイヤーを追加する

次に、ある領域にいくつかのレイヤーを追加しましょう。層に意味のある名前をつけます。

次にこのレイヤーに入れるセンサーの種類(DataKind)を選択します。



どのデバイスを選択するかによって、オプションが異なります。Gravioは以下をサポートします：

- 「物理センサーデバイス」
- 「ソフトウェアセンサー」

7.3.1.1. 物理センサー

レイヤーにデバイスを追加する

Gravioには様々なセンサーを接続できますが、物理センサーとして、Gravio Basic購入者には [無償デバイスレンタル](#) をご利用いただけます。

2種類の無線通信型のセンサーを現在はサポートしています。

- Zigbee対応デバイス
- EnOcean対応デバイス
- Bluetooth対応デバイス

Zigbee対応デバイスのセンサーレシーバー

Zigbee対応デバイスを利用するには以下のUSBレシーバーをPCに差し込む必要があります。



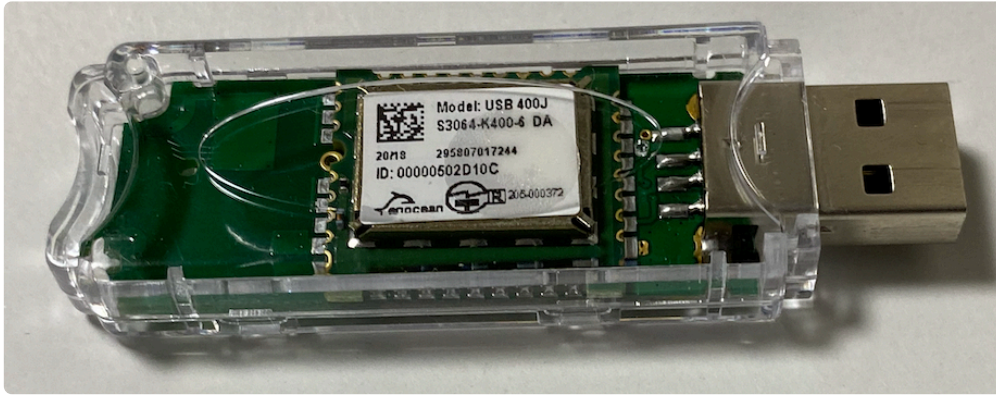
Windows / MacでZigbeeのUSBレシーバーが認識されない場合には [Silicon Lab のサイト](#) よりドライバーをインストールしてください。

Windowsは CP210x Windows Drivers を選択してダウンロードし、インストールしてください。

Macは CP210x VCP Mac OSX Driver を選択してダウンロードし、インストールしてください。

EnOcean対応デバイスのセンサーレシーバー

EnOcean対応デバイスを利用するには以下のUSBレシーバーをPCに差し込む必要があります。



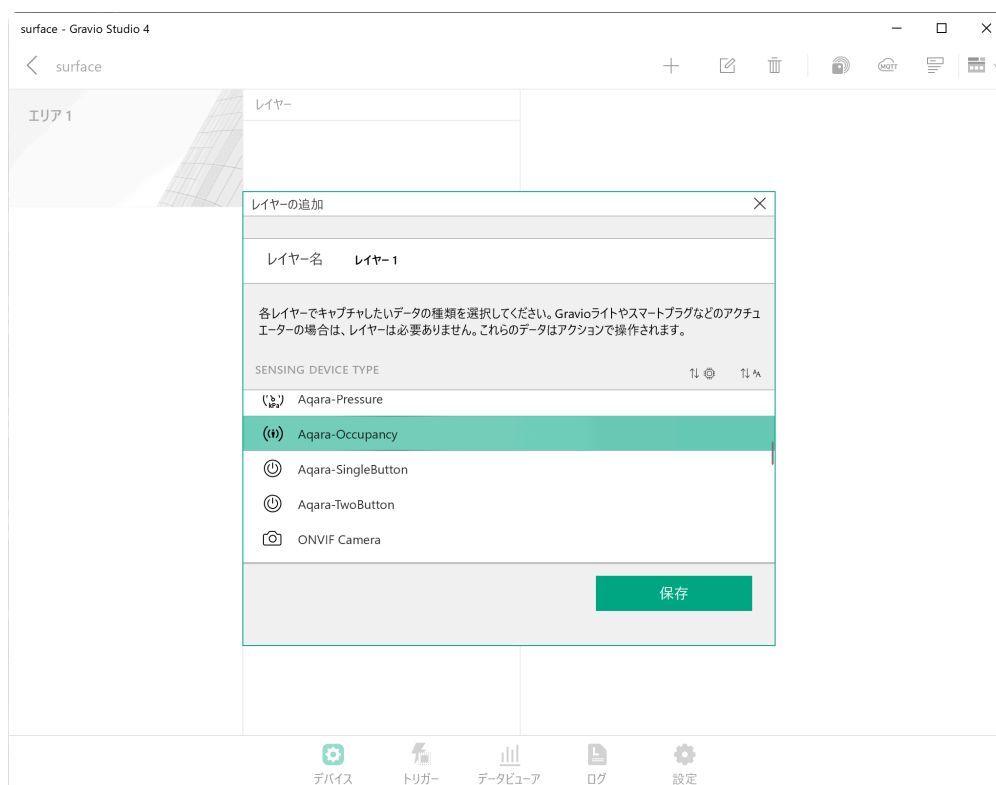
Bluetooth対応デバイスのセンサーレシーバー

Bluetooth対応デバイスを使用するにはPCに内蔵されているBluetooth機能、またはBluetooth USBレシーバをPCに差し込む必要があります。

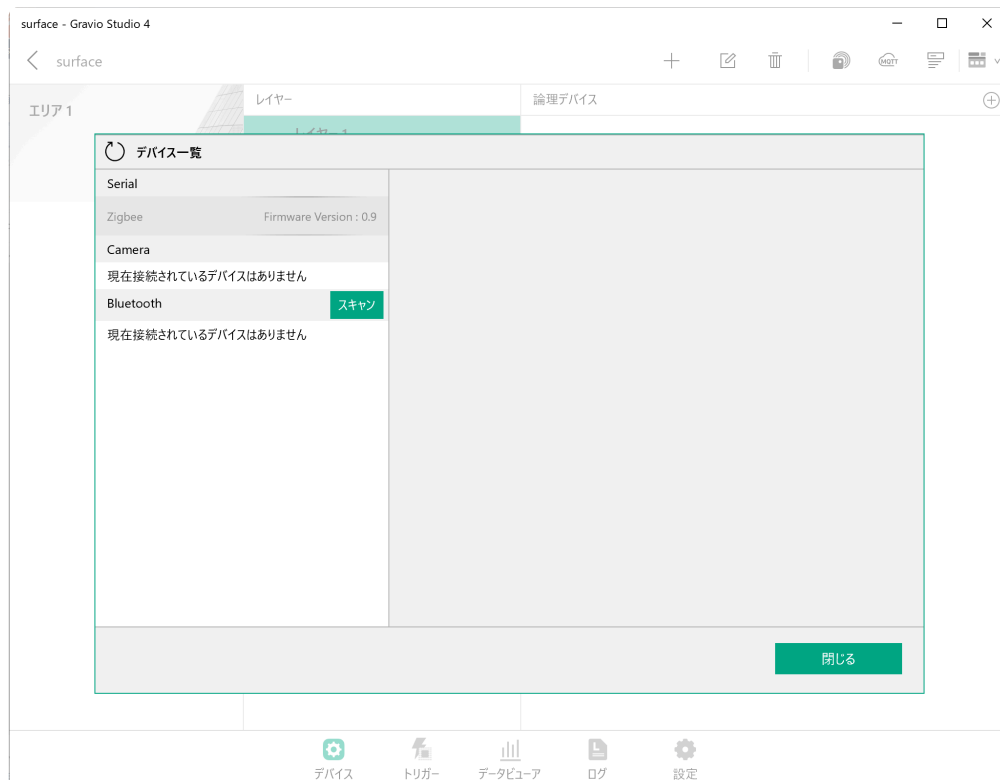
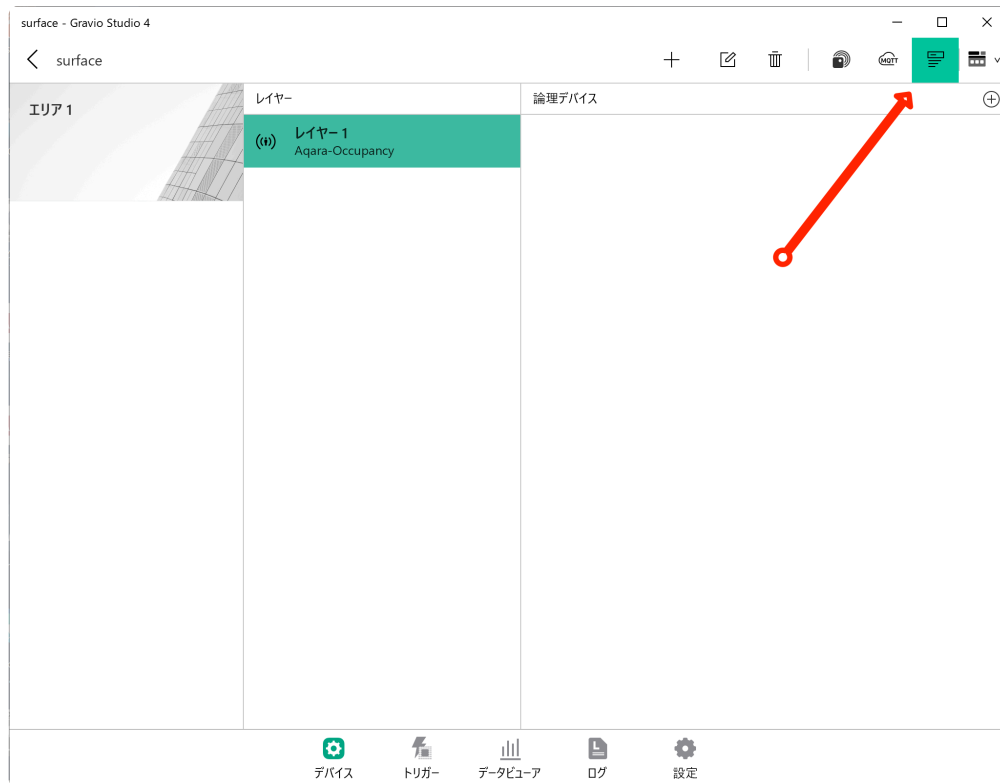
7.3.1.1.1. Zigbee対応デバイスの追加

Gravio Studioを使ってZigbee対応デバイスのセンサーレシーバーとレイヤーを紐づけます

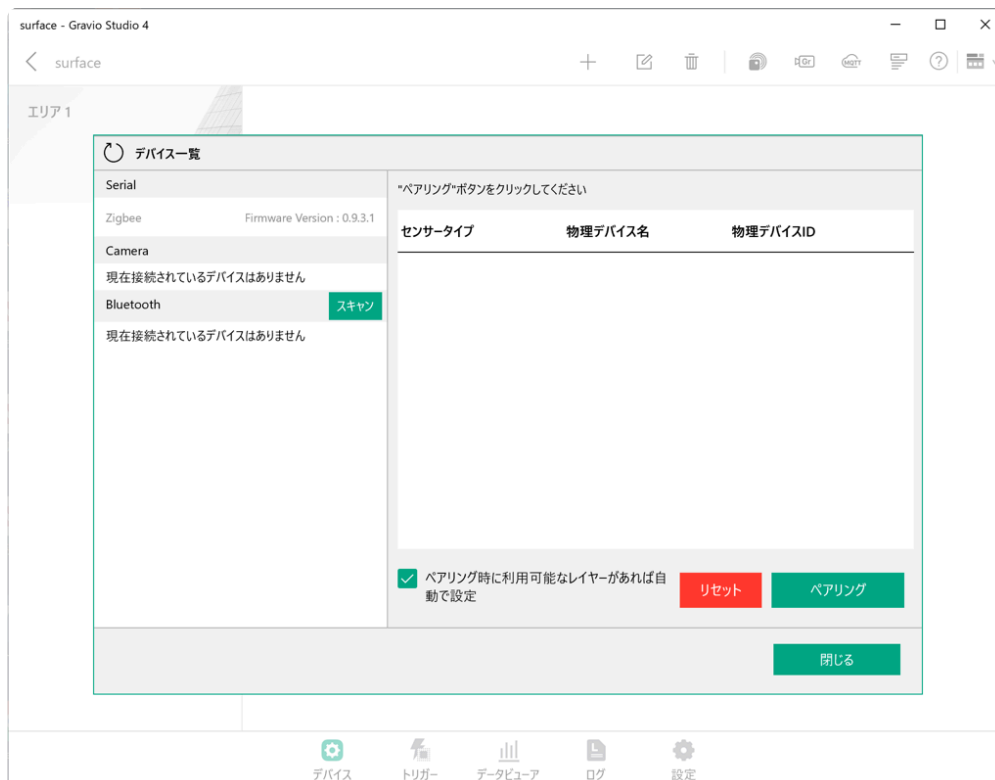
1. 「デバイス」で、使用するZigbeeセンサーの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。



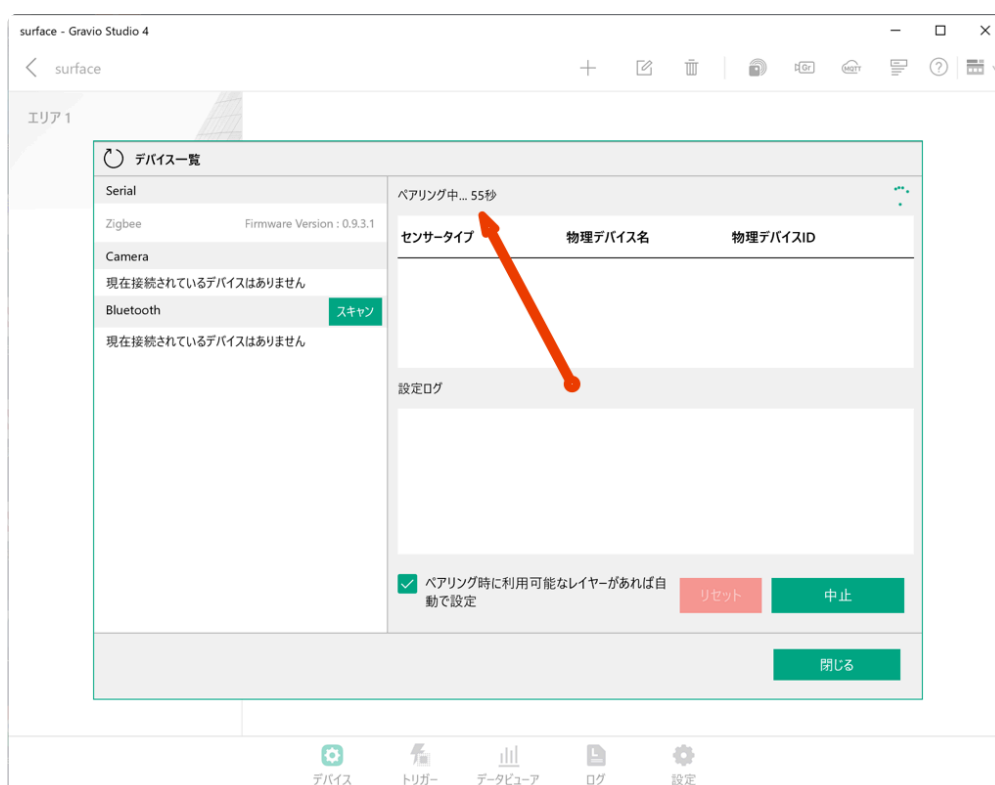
2. Zigbee対応デバイスのセンサーデータを受信するには、まず、センサーをセンサーレシーバーとペアリングする必要があります。以下のボタンを押して、シリアルポートに接続されているセンサーレシーバーの設定画面を表示します。



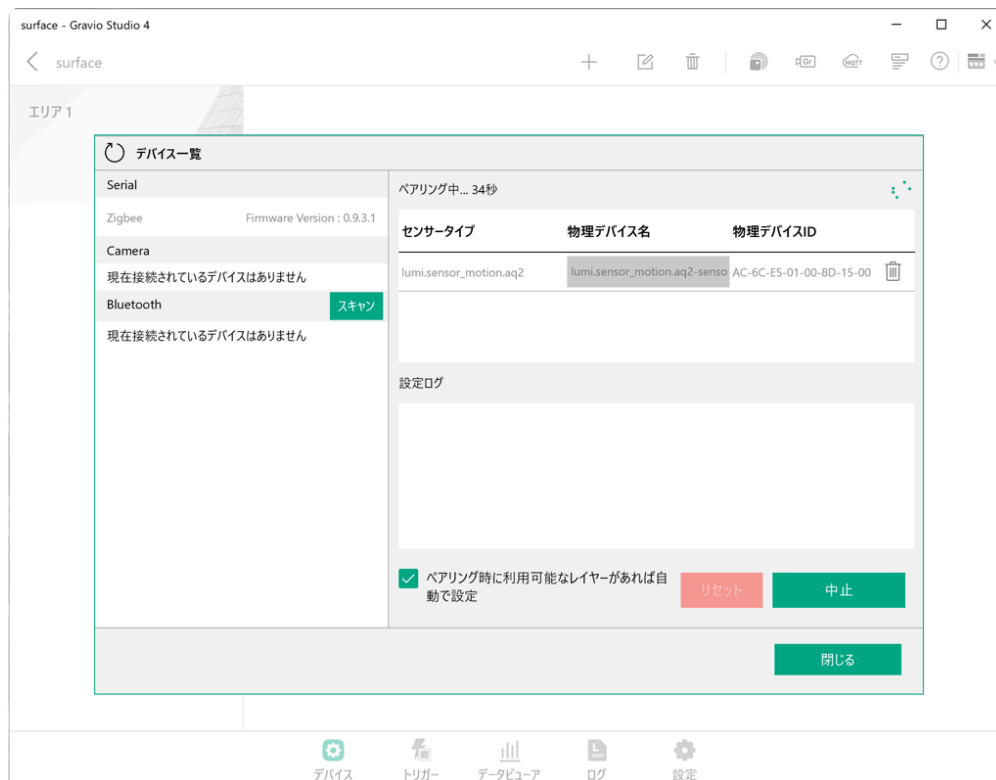
3. Zigbeeの行の右矢印をクリックして、ペアリング画面を表示します。



4. 「ペアリング」ボタンを押してレシーバーがペアリング待ちの状態にします。(ペアリングモードは1分間です。)



5. ペアリングモード中にセンサー側のボタンを5秒程度押し続けるとセンサーがペアリングされて以下のように表示されます。

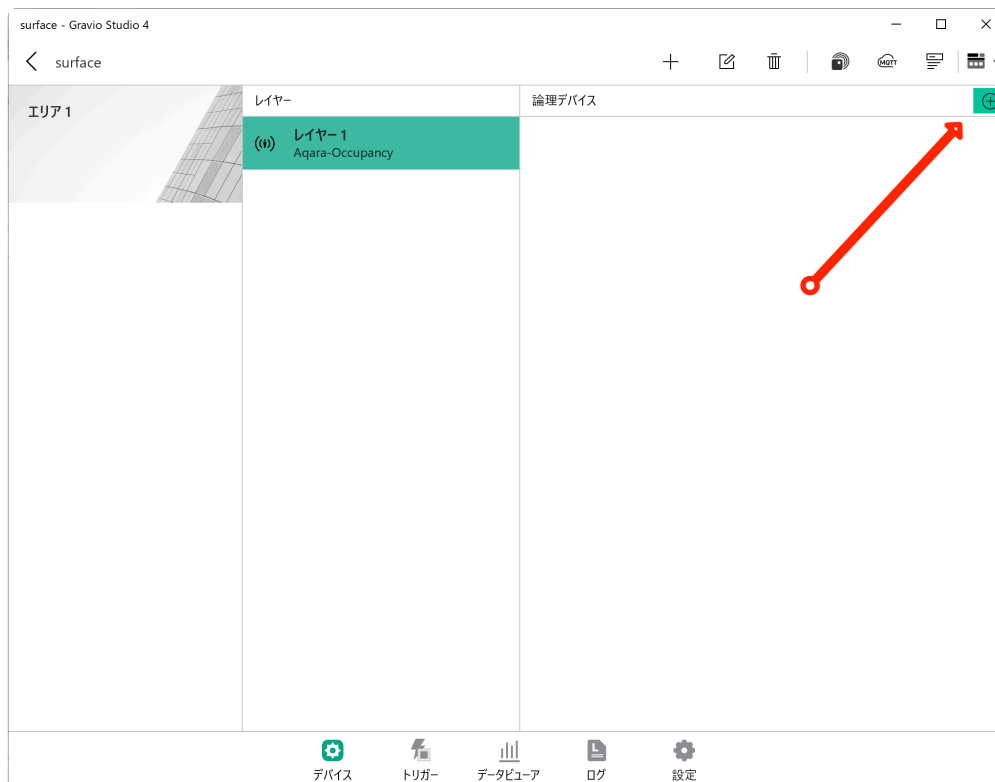


「ペアリング時に利用可能なレイヤーがあれば自動で設定」をオンにしている場合、ペアリングしたデバイスタイプを設定しているレイヤーがある場合は、自動的にそのレイヤーにデバイスが登録されて設定は有効となります。その場合は以下の3つのステップの操作は必要なくなります。

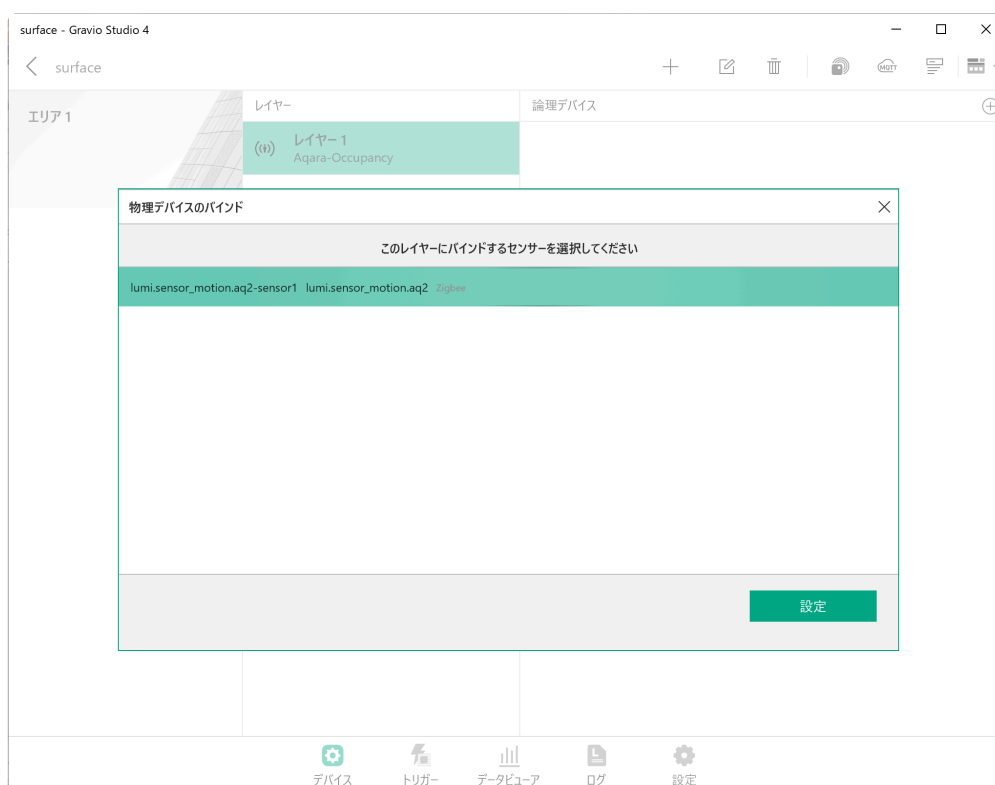
実デバイス側のリセットとペアリングの実行については、Gravio Webサイトの[センサーセットアップ](#)をご覧ください。これでセンサーのペアリングは完了です。ダイアログを閉じてください。

ペアリングしたデバイスを手動でレイヤーに設定する場合の操作方法

1. 次に、ウィンドウの右上にある丸で囲まれた「+」マークをクリックしてください。



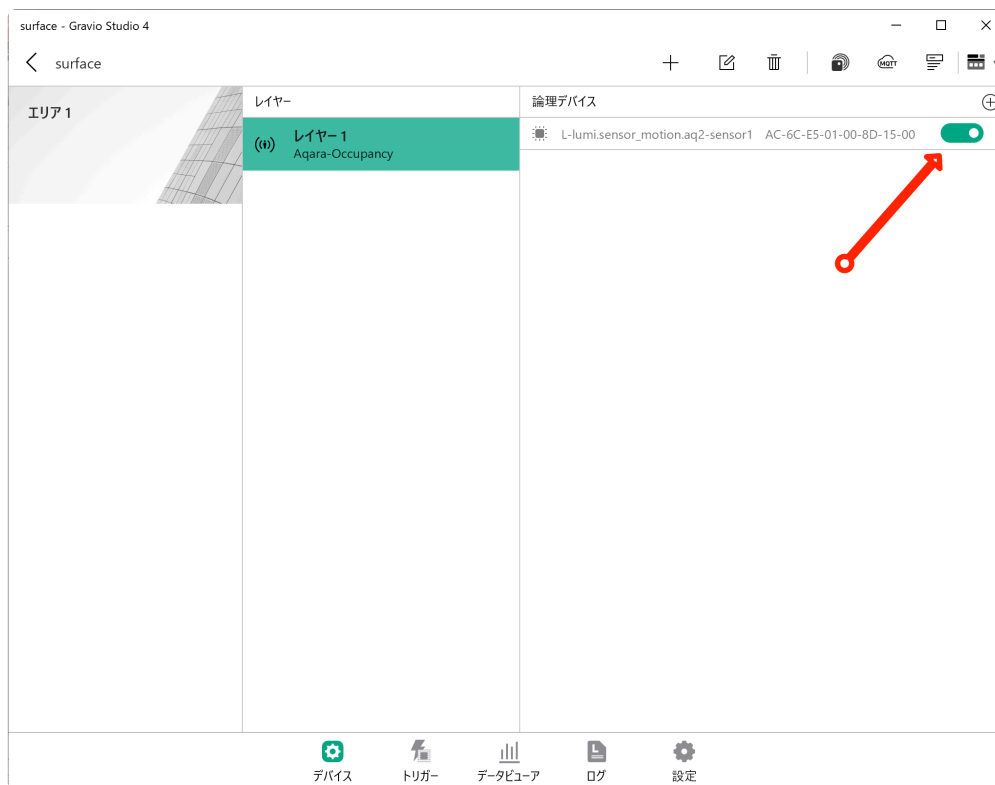
2. レイヤーに接続したいセンサーを選択して、設定します。



設定が終わったら、この画面と閉じてください。

3. 以下のような画面になります。最後にこのセンサーレシーバーからデータを受信ために、スイッチをONにして

ください。



以上でセンサーのデータの受信が開始されます。

1つの dongle で 64 個の Zigbee デバイスを接続できます。

7.3.1.1.2. Zigbeeレシーバー(旧型)のドライバー登録(Windowsのみ)

Zigbee対応USBレシーバー(旧型)のドライバーをインストール

新しいZigbee対応USBレシーバーを使用している場合は、[こちら](#)を参照してください。

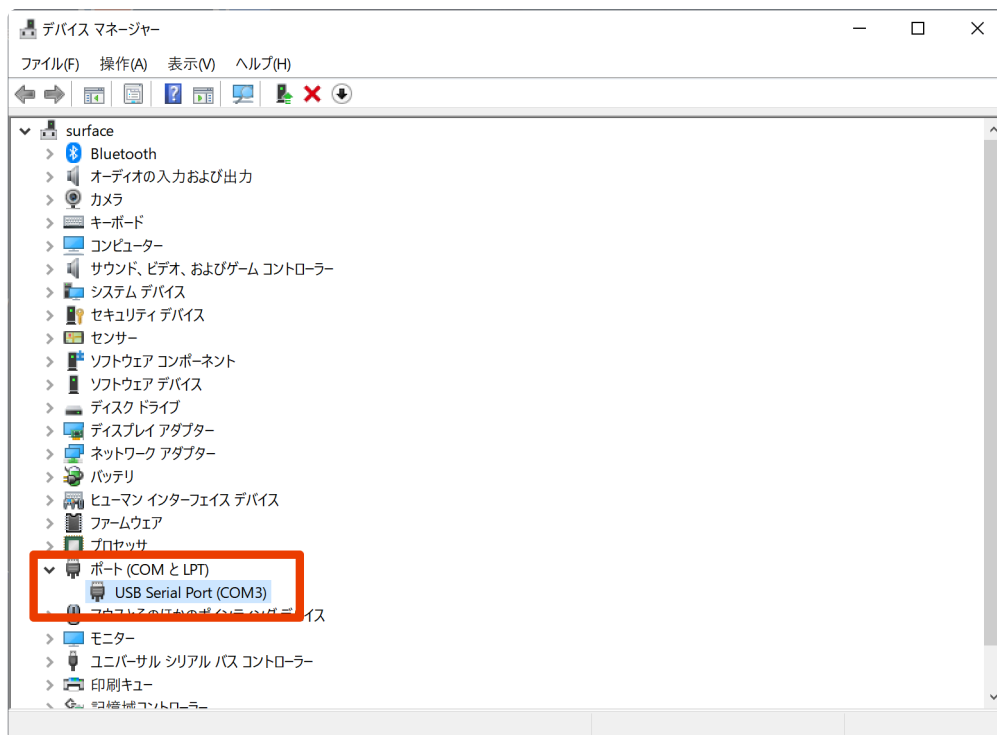
Zigbee対応USBレシーバーをPCに差し込むとWindows10が自動的にドライバーをインストールするため、そのまま使用することができます。

環境により自動的にインストールされるドライバーでは動作しない現象が報告されており、その場合には下記の手順にてドライバーをインストールします。

デバイスマネージャーでインストールされたZigbee対応USBレシーバーのドライバーを確認する

デバイスマネージャーの「ポート (COMとLPT)」を開きます。USB Serial Port (COMx)と表示されていればインストールされています。

注) COMxのxは環境により数字が変わります。



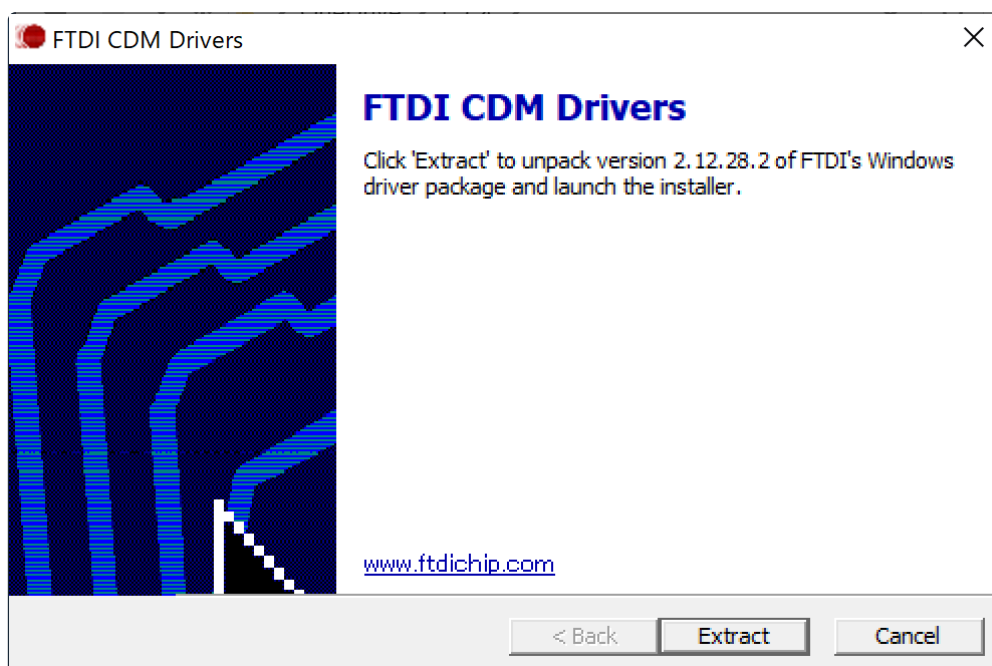
ドライバーを手動でインストールする

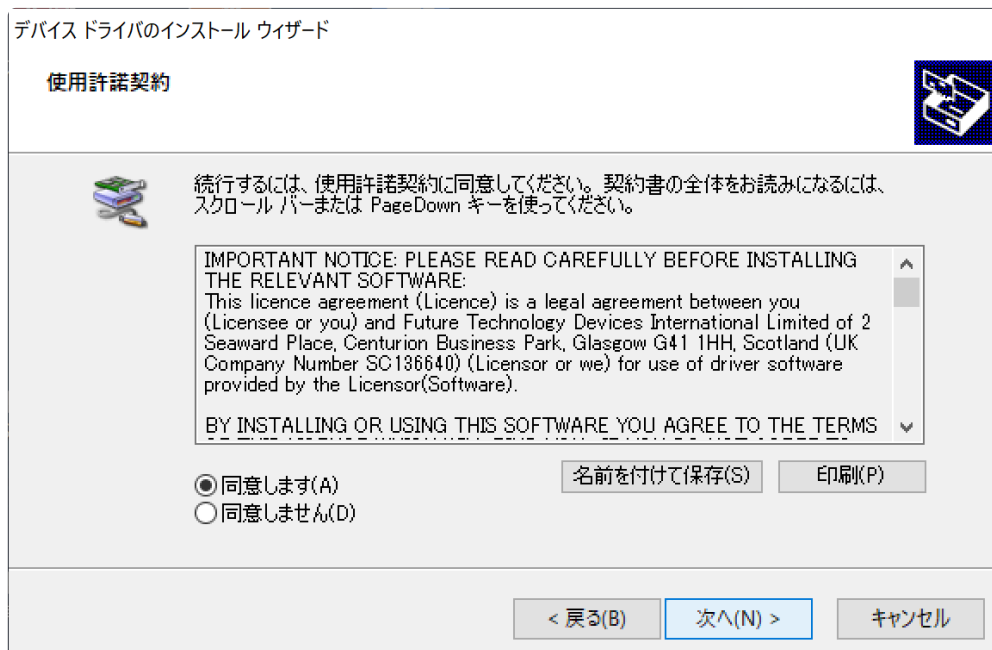
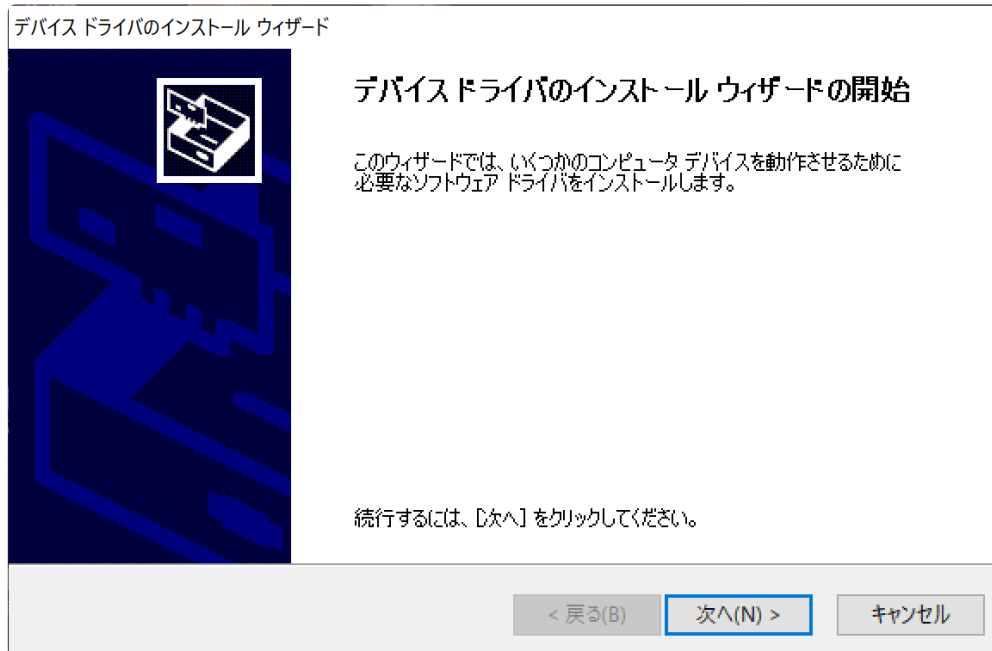
ドライバーをインストールするには、[こちらのサイト](#) からドライバーをダウンロード、実行してインストールしてください。

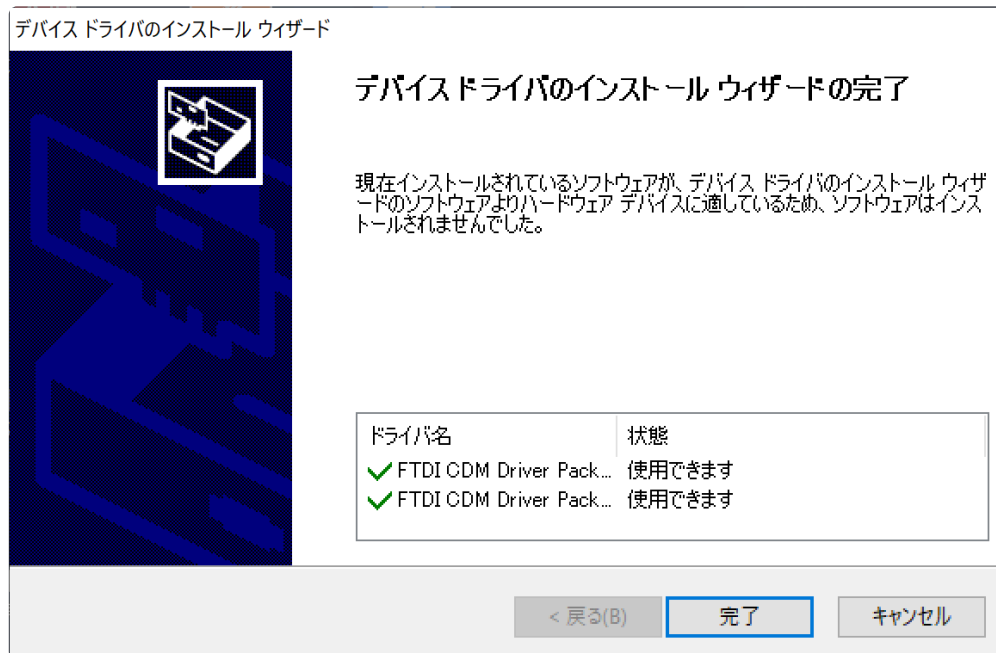
Currently Supported VCP Drivers:

Operating System	Release Date	Processor Architecture							Comments
		X86 (32-Bit)	X64 (64-Bit)	PPC	ARM	MIPSII	MIPSIV	SH4	
Windows*	2017-08-30	2.12.28	2.12.28	-	-	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a setup executable . Please read the Release Notes and Installation Guides .
Linux	-	-	-	-	-	-	-	-	All FTDI devices now supported in Ubuntu 11.10, kernel 3.0.0-19. Refer to TN-101 if you need a custom VCP VID/PID in Linux. VCP drivers are integrated into the kernel .
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-	Refer to TN-105 if you need a custom VCP VID/PID in MAC OS
Mac OS X 10.9	-	-	-	-	-	-	-	-	This driver is signed

ファイルをダウンロードすると「CDM21228_Setup.exe」が保存されますので実行してください。実行するとFTDI CDM Driversのパネルが表示されますので「Extract」ボタンを押して展開します。展開が終了するとインストーラーが表示されますので、インストールしてください。



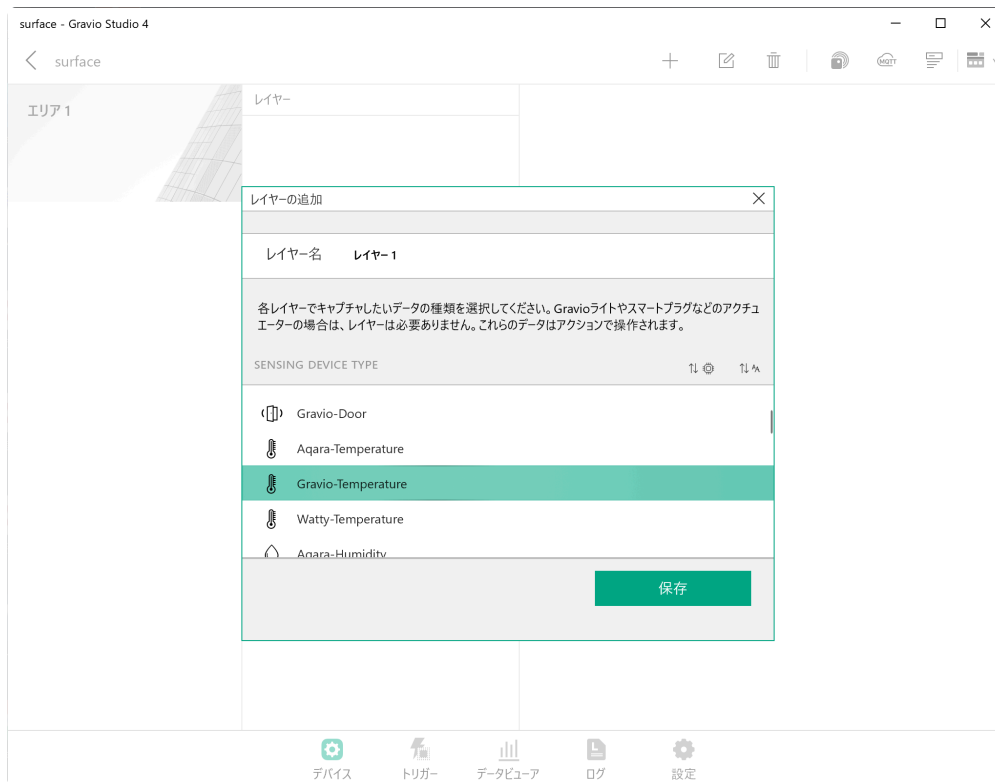




7.3.1.1.3. EnOcean対応デバイスの追加

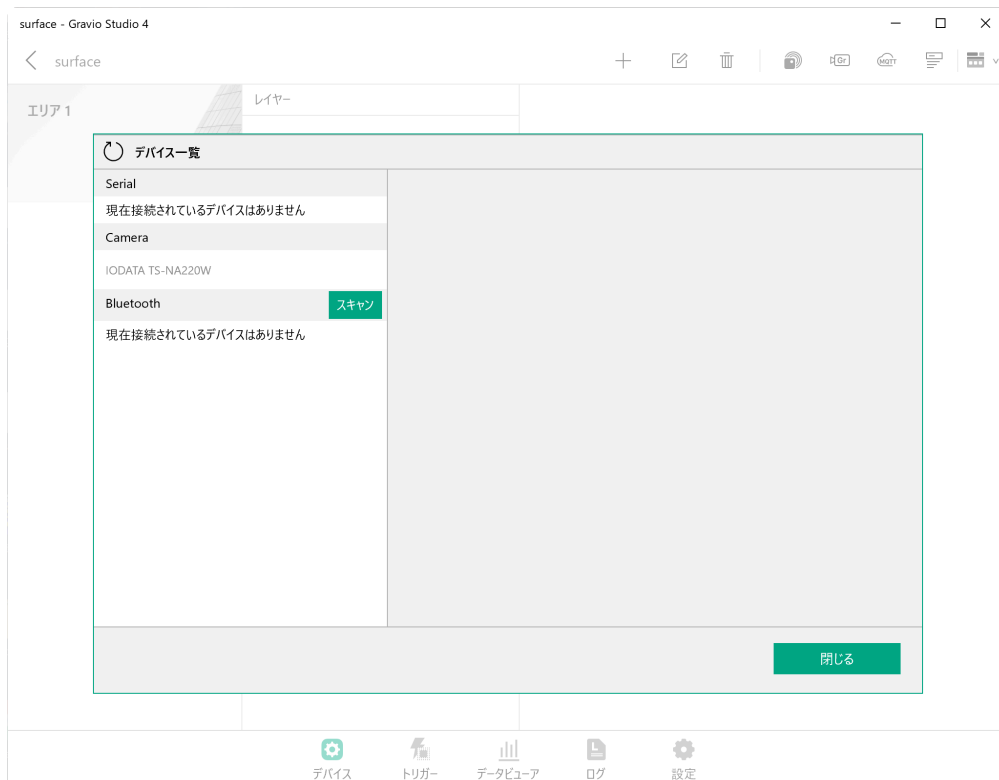
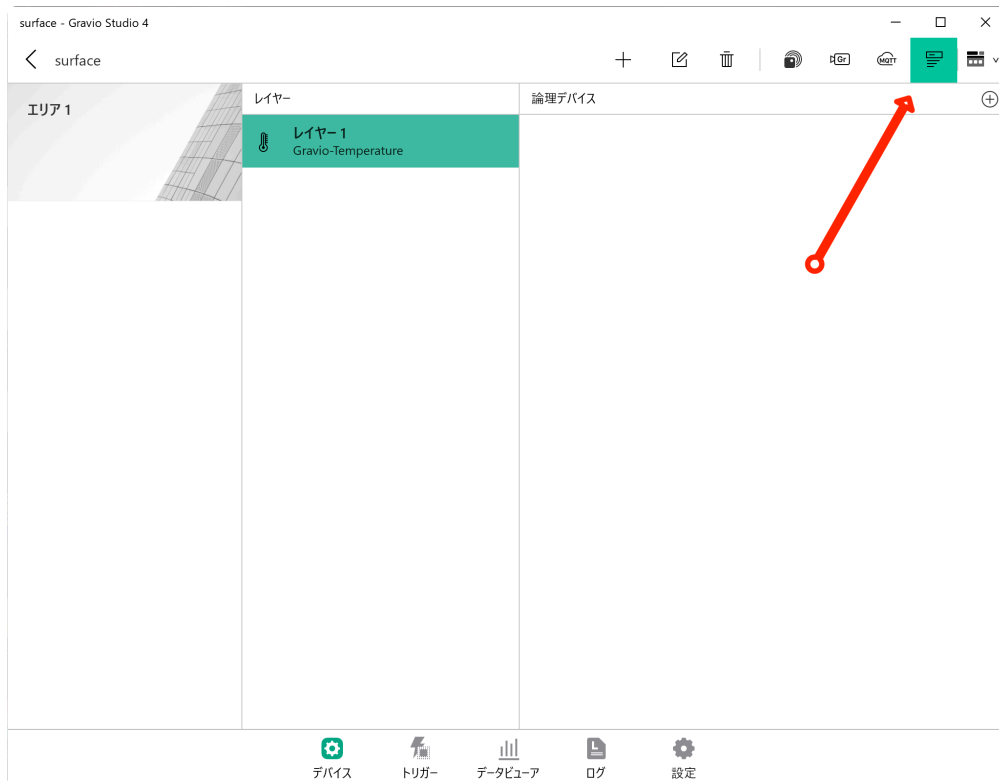
Gravio Studioを使ってEnOcean対応デバイスのセンサーレシーバーとレイヤーを紐づけます

1. 「デバイス」で、使用するEnOceanセンサーの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。

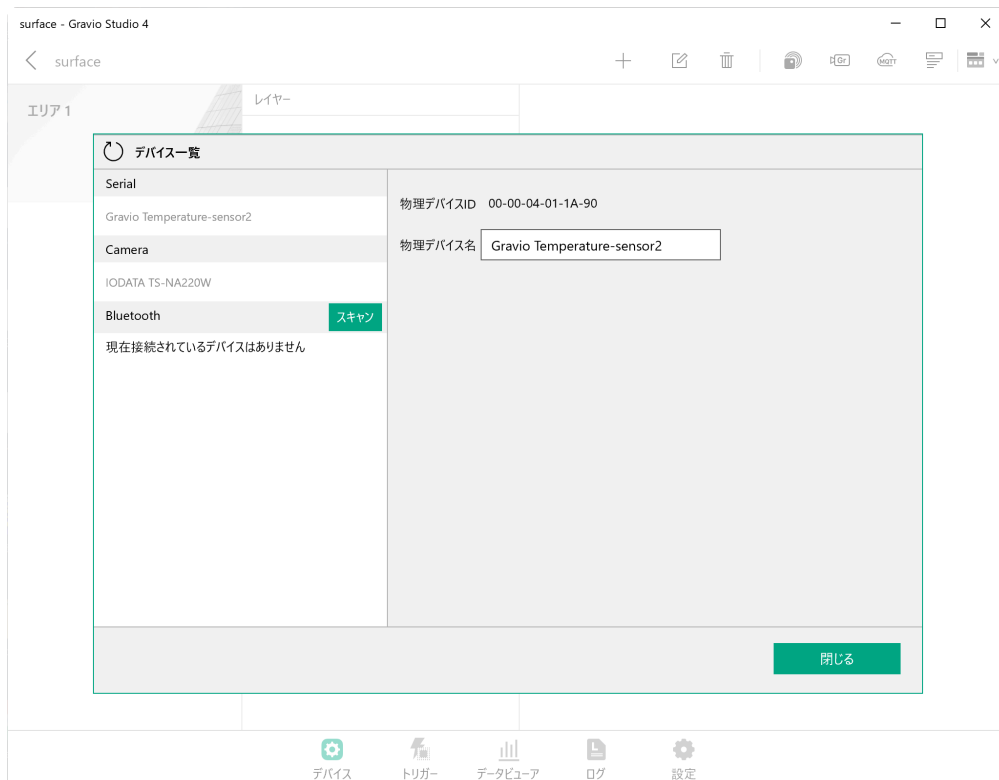
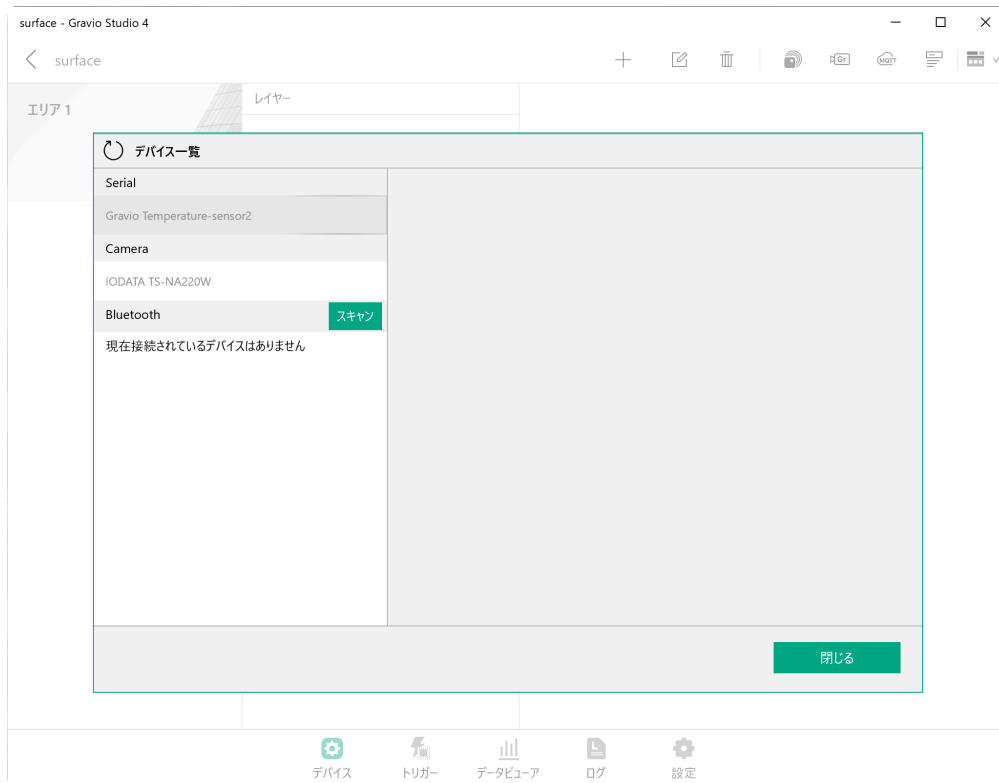


現在動作確認が取れているEnOceanセンサーは [こちら](#) を参照してください。

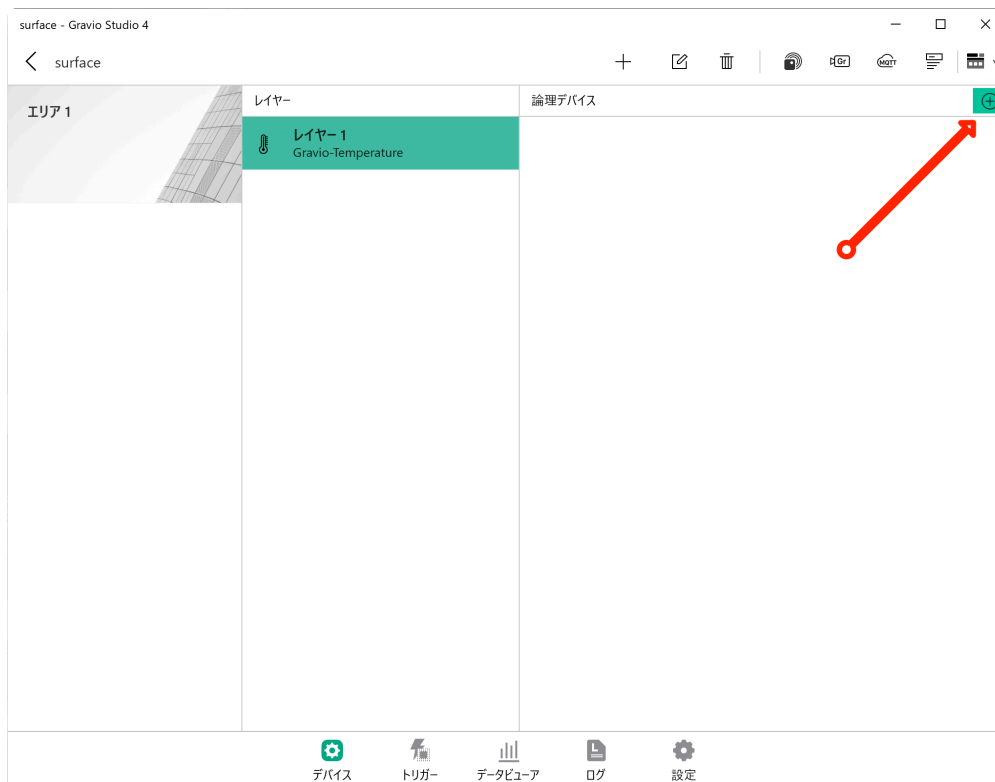
1. EnOcean対応デバイスのセンサーデータを受信するには、まず、センサーをセンサーレシーバーとペアリングする必要があります。以下のボタンを押して、シリアルポートに接続されているセンサーレシーバーの設定画面を表示します。



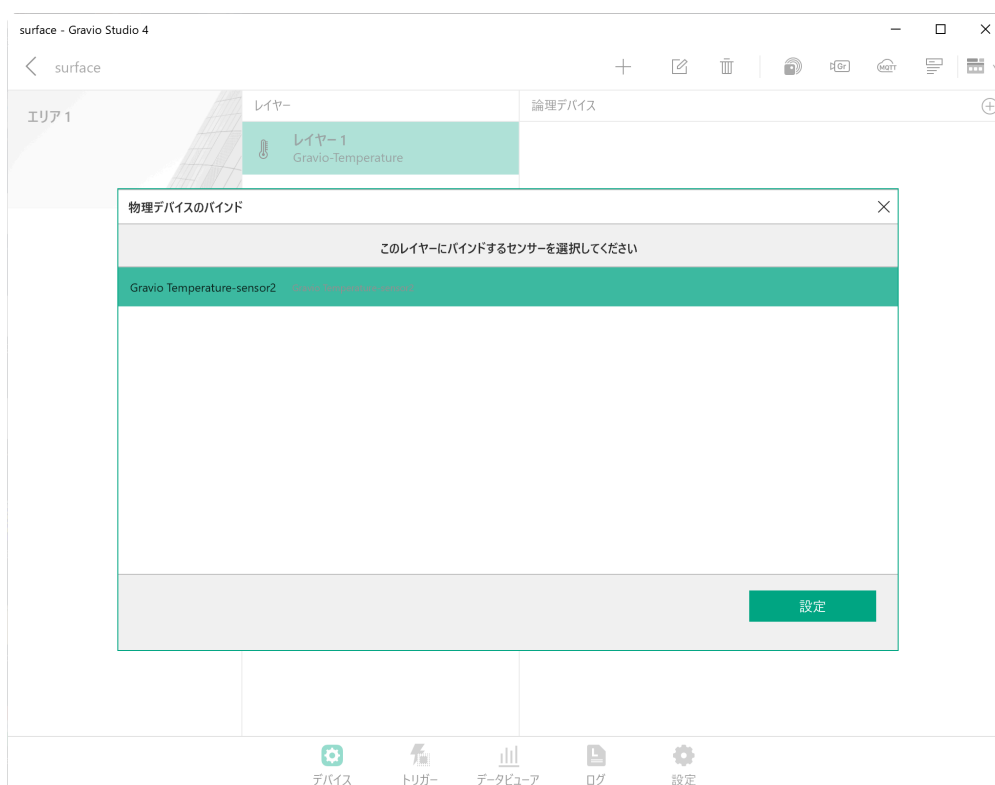
2. EnOcean対応デバイスのボタンを押します。しばらくするとそのセンサーがシリアルポートに表示されます。



3. 次に、ウィンドウの右上にある丸で囲まれた「+」マークをクリックしてください。



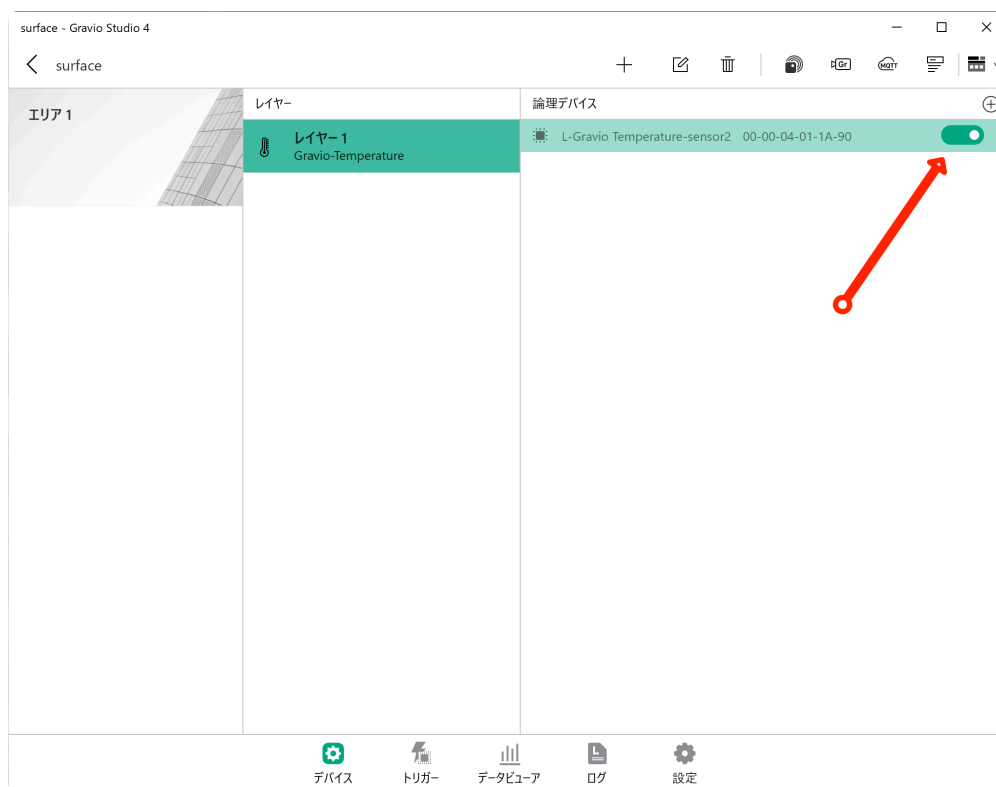
4. レイヤーに接続したいセンサーを選択して、設定します。



設定が終わったら、この画面と閉じてください。

5. 以下のような画面になります。最後にこのセンサーレシーバーからデータを受信ために、スイッチをONにして

ください。

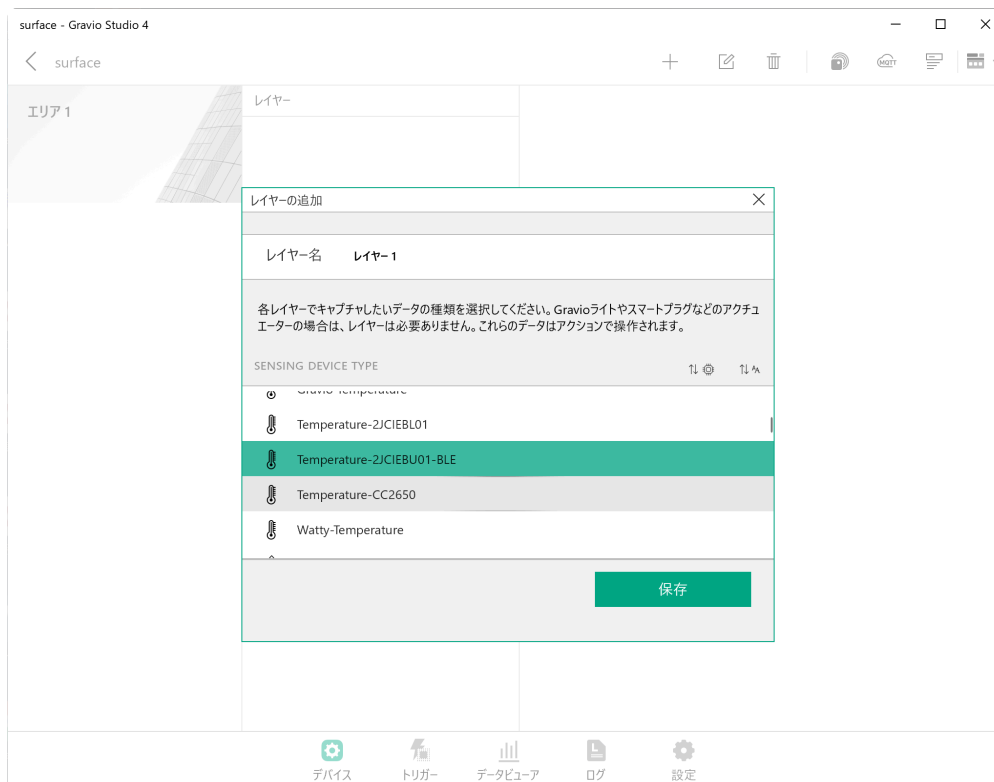


以上でセンサーのデータの受信が開始されます。

7.3.1.1.4. Bluetooth対応デバイスの追加

Gravio Studioを使ってBluetooth対応デバイスのセンサーレシーバーとレイヤーを紐づけます

1. 「デバイス」で、使用するBluetoothセンサーの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。

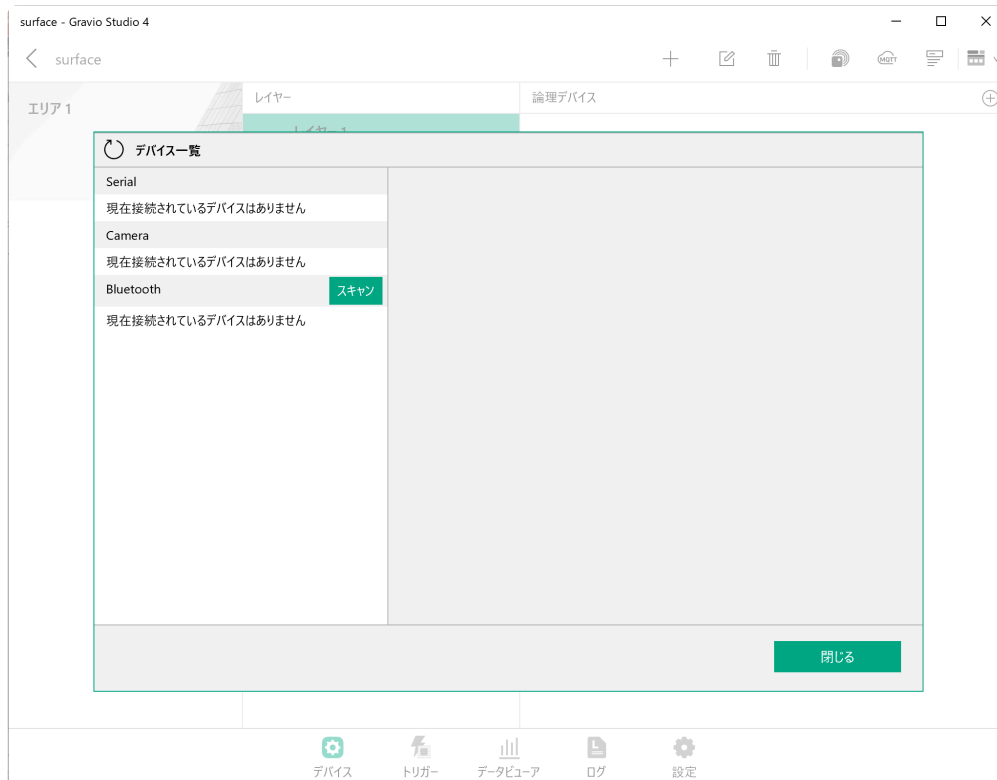
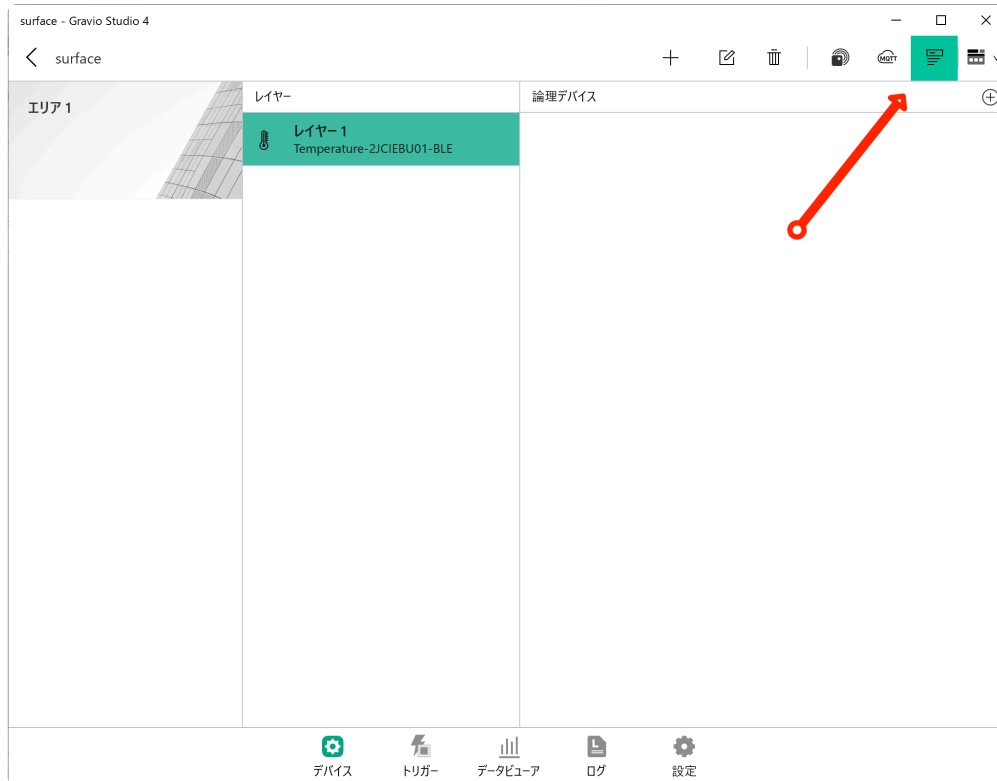


サンプルとして [オムロン製2JCIE-BU01](#) を使用して温度データを取得します。

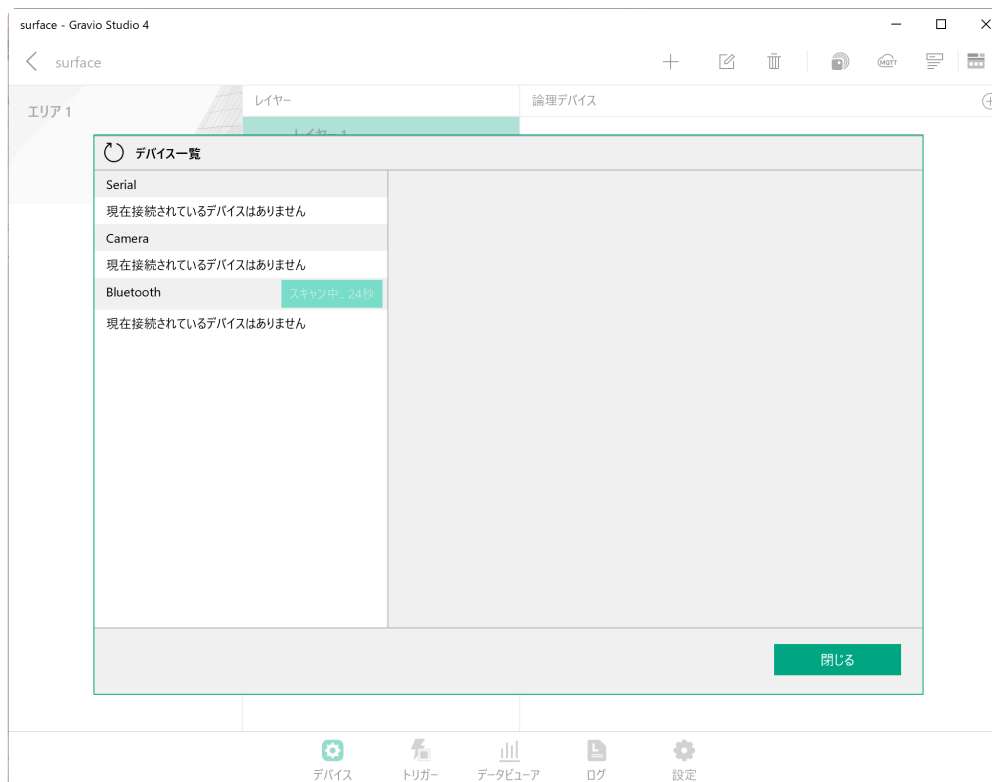
[株式会社ティアンドディ製おんどり TR7Aシリーズ](#) (TR71A、TR72A、TR72A-S、TR75A)を使用する場合には予めコンフィグレーションマネージャーのBluetooth設定で「アドバタイジングモードを使用する」をオンにしてください。

現在動作確認が取れているBluetooth対応デバイスは [こちら](#) を参照してください。

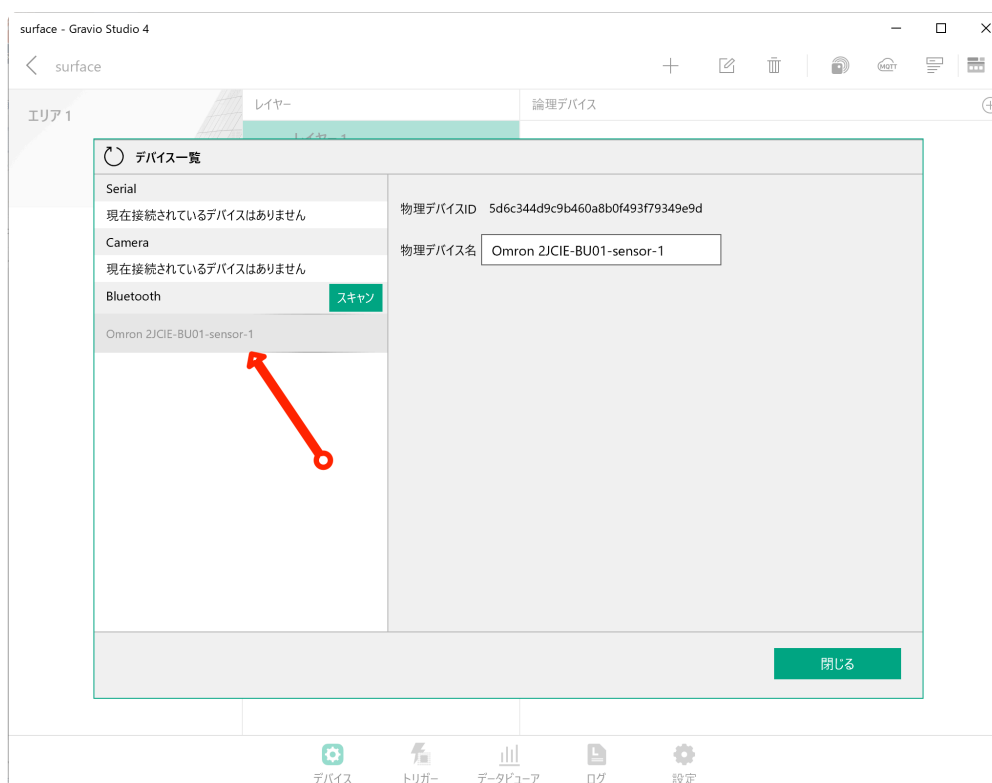
2. Bluetooth対応デバイスのセンサーデータを受信するには、まず、センサーをセンサーレシーバーとペアリングする必要があります。以下のボタンを押して、センサーレシーバーの設定画面を表示します。



3. Bluetoothの行の右矢印をクリックして、スキャンボタンを押します。(スキャンモードは1分間です。)

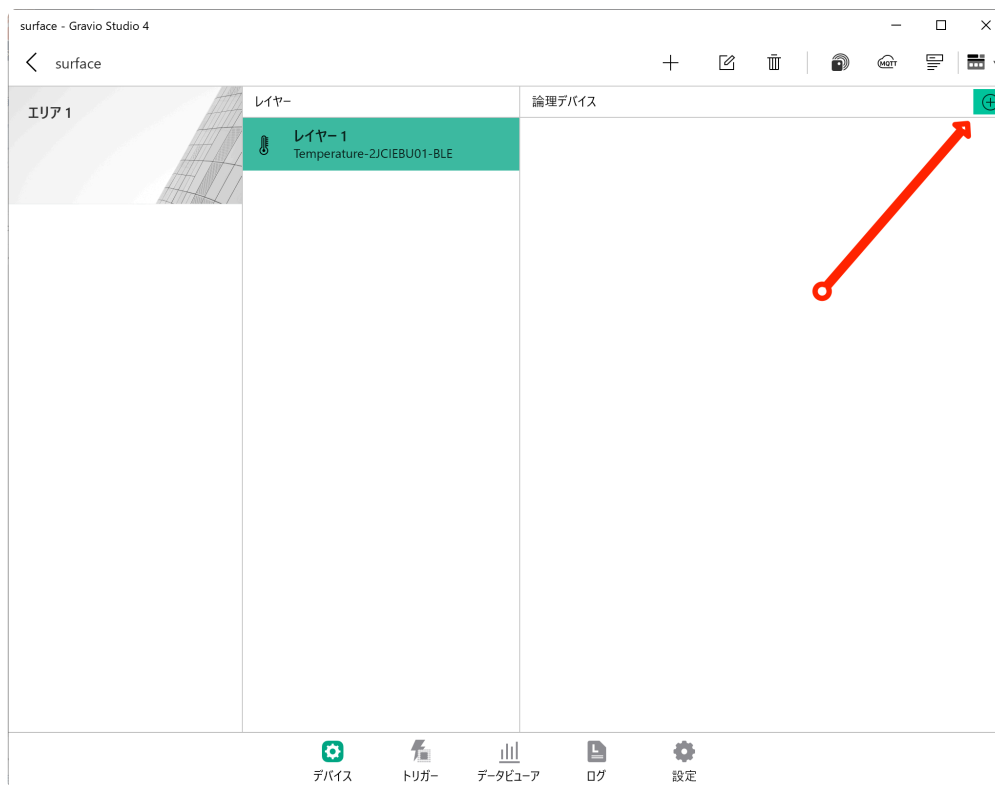


4. スキャンモード中にセンサーが発見されると以下のように表示されます。

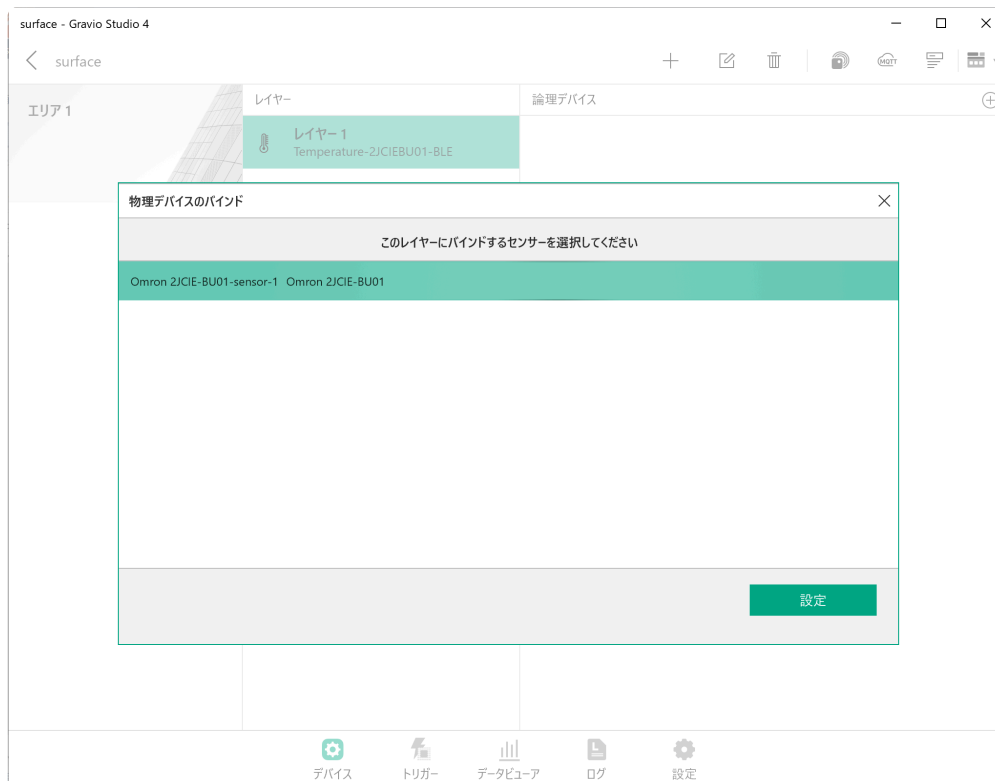


これでセンサーのペアリングは完了です。ダイアログを閉じてください。

5. 次に、ウィンドウの右上にある丸で囲まれた「+」マークをクリックしてください。

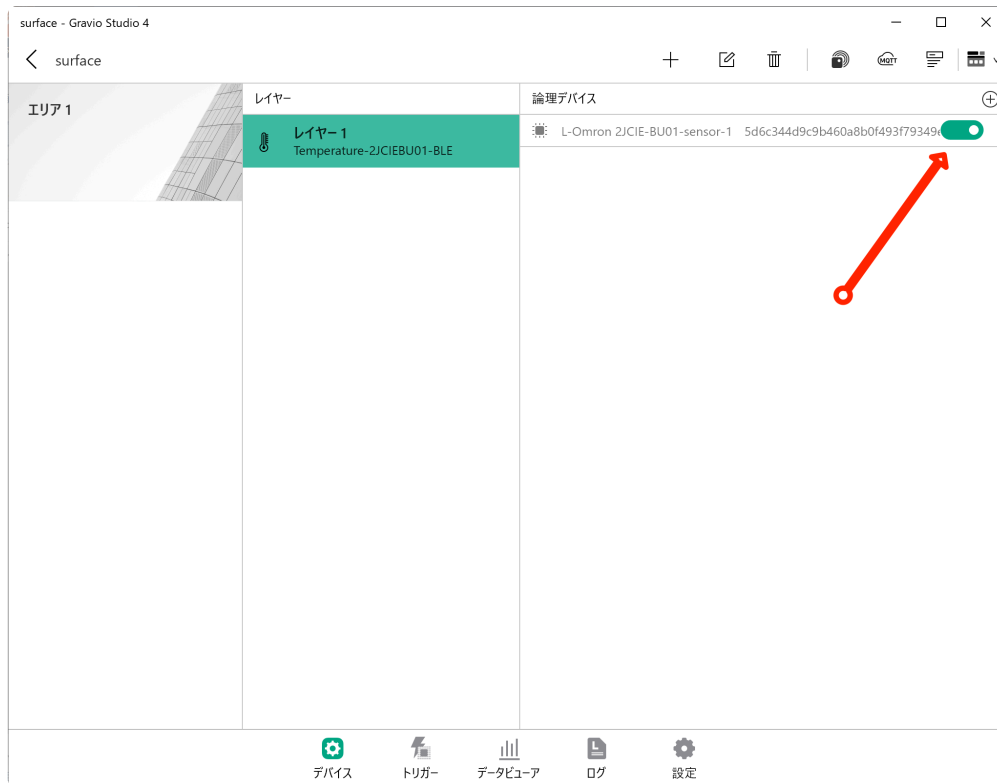


6. レイヤーに接続したいセンサーを選択して、設定します。



設定が終わったら、この画面と閉じてください。

7. 以下のような画面になります。最後にこのセンサーレシーバーからデータを受信ために、スイッチをONにしてください。



以上でセンサーのデータの受信が開始されます。

7.3.1.1.5. バーコードリーダーの追加

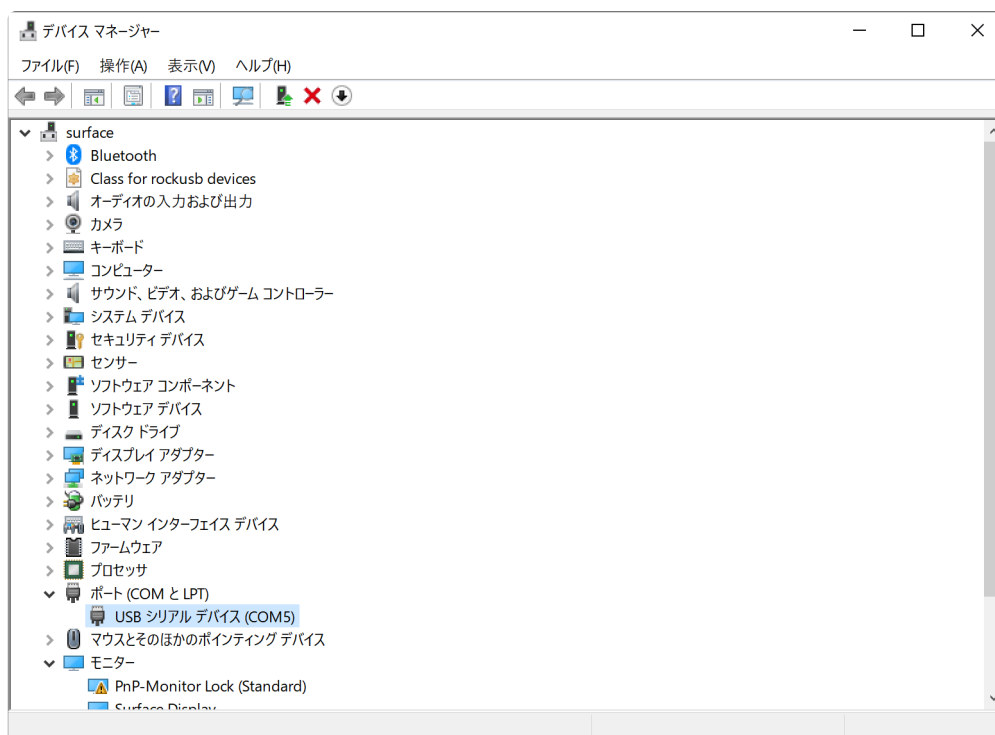
Gravio Studioを使ってバーコードリーダーとレイヤーを紐づけます

Gravioではバーコードリーダー(**USB COM ポートエミュレーション機能が利用できるバーコードリーダー**) を1つのレイヤーとしてデータ入力できるデバイスとして利用できます。

現在動作確認が取れているバーコードリーダーは [こちら](#) を参照してください。

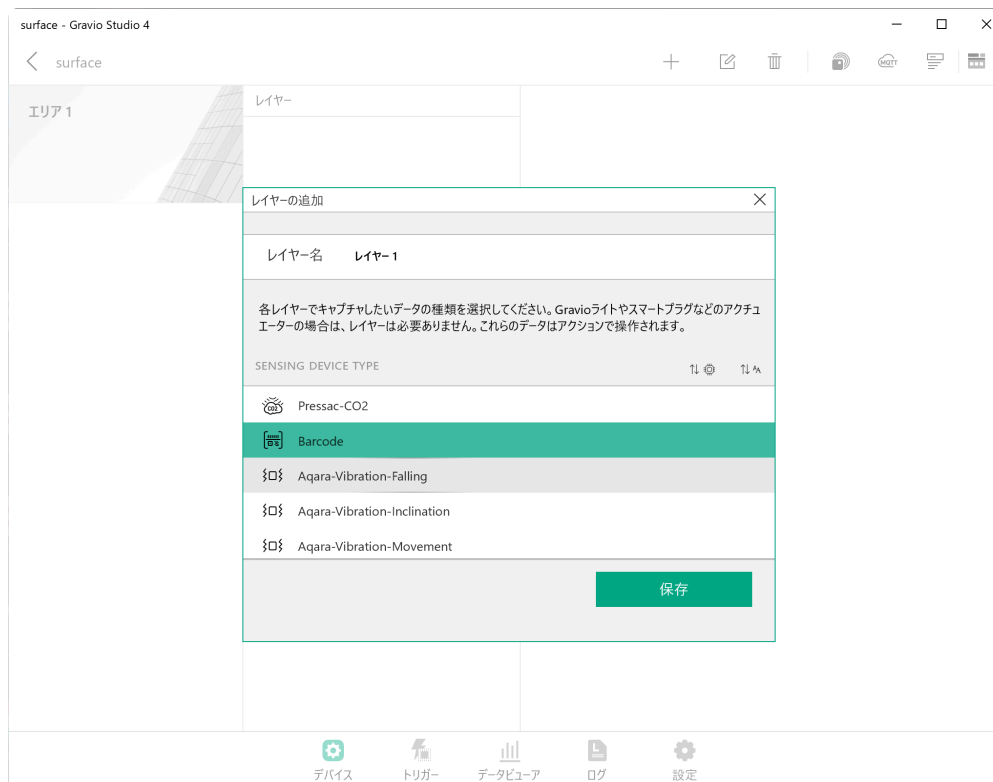
バーコードリーダーの接続確認

バーコードリーダーの接続確認はWindows10の場合、デバイスマネージャーにCOMx(この場合COM5)が表示されているかを確認してください。



バーコードリーダーの追加

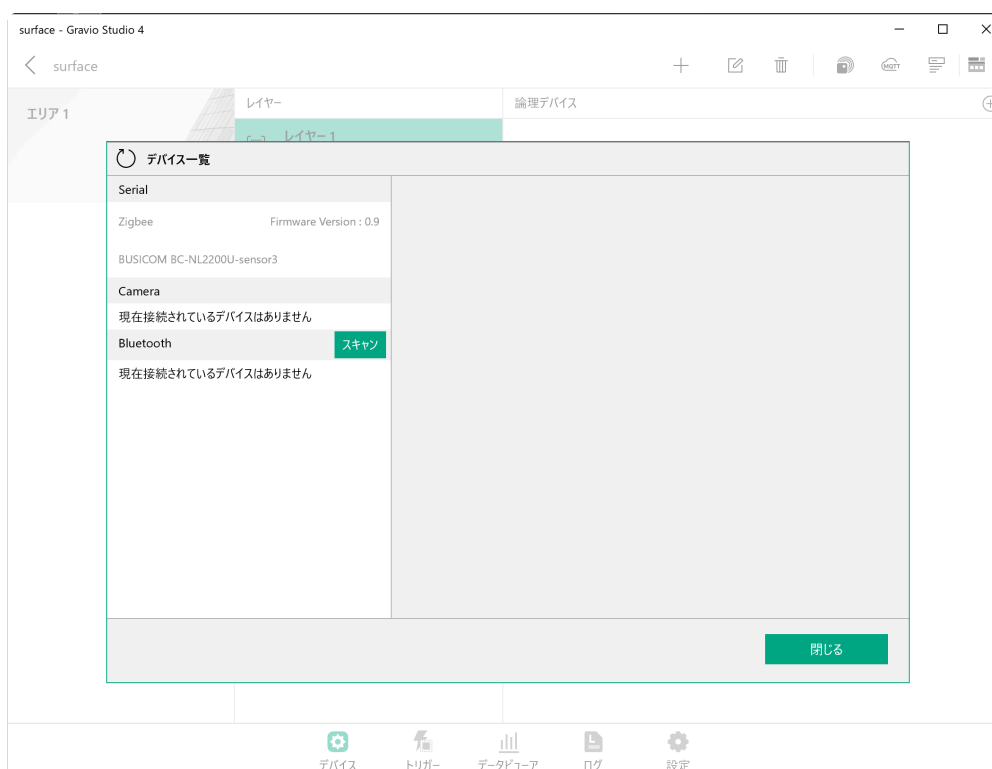
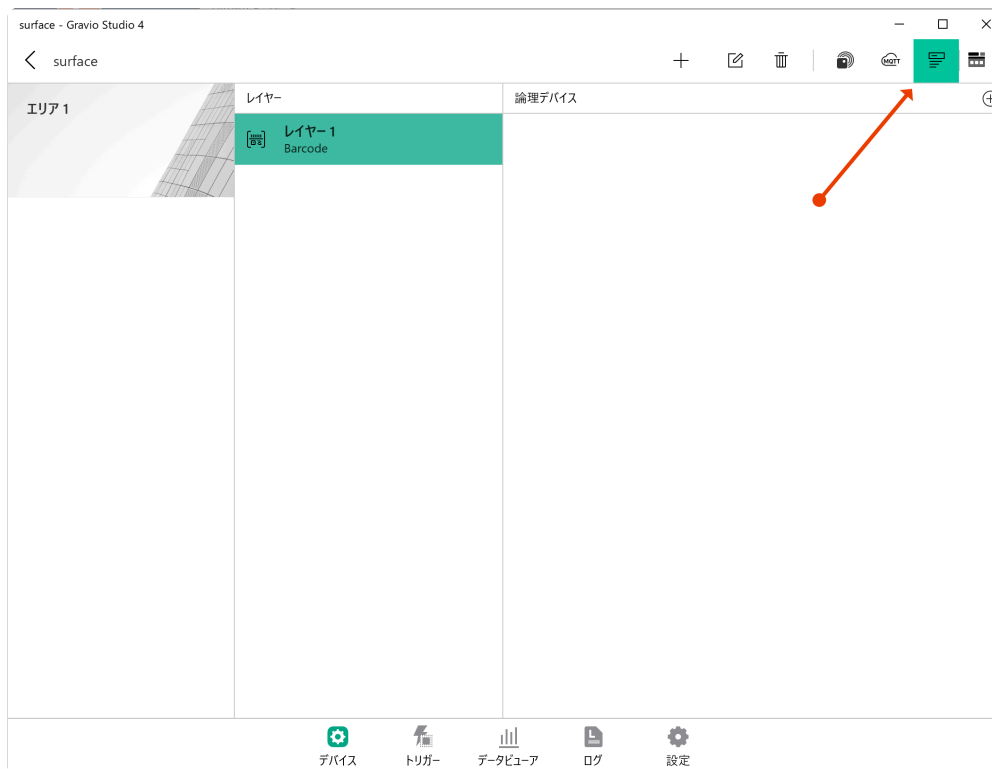
1. 「デバイス」で、使用するBarcode(DataKind)を選択して新しいエリアとレイヤーを追加します。



ここでは、バーコードリーダーは [BUSICOM製BC-NL2200U](#) を使用してバーコードのデータを取得する例として説明します。

BUSICOM社製のBC-NL1100U, BC-NL1100U II, BC-NL2200U, BC-NL2200U II, BC-NL3000U, BC-NL3000U IIIはUSB キーボード入力としてシミュレートされるUSB HID-KBW機能と、ホストがシリアルポート のようにデータを受信できるUSB COM ポートエミュレーション機能を切り替えることができます。Gravioで使用する場合には、USB COM ポートエミュレーション機能の設定にしてください。詳細はBUSICOM社のマニュアルを参照してください。

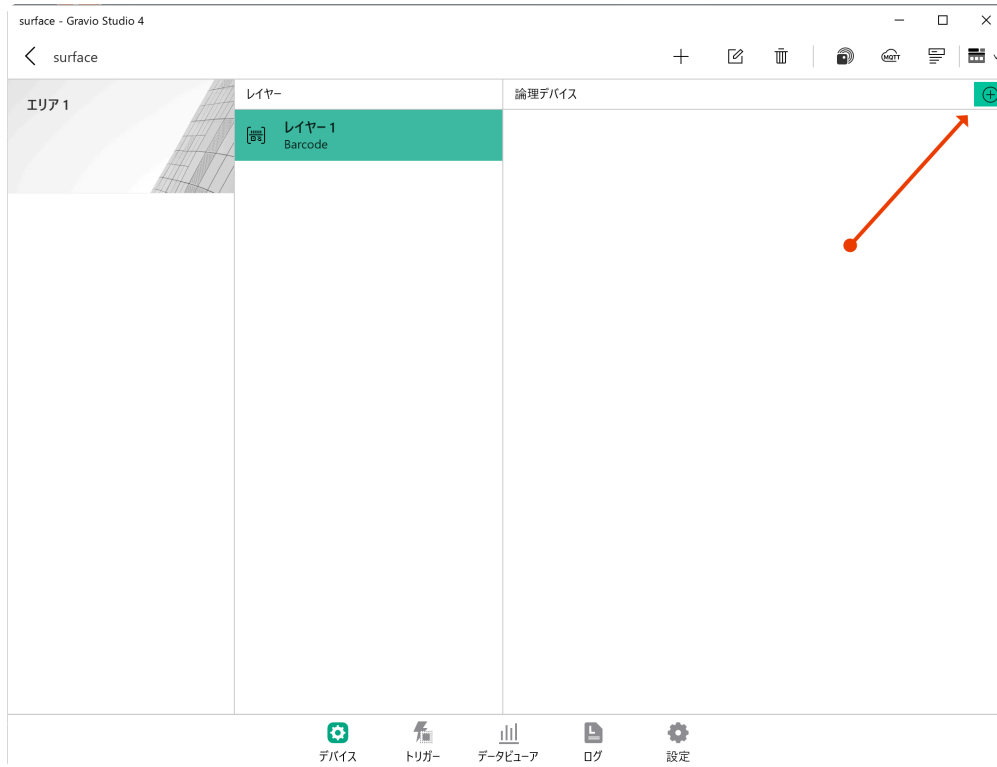
2. バーコードデータを受信するには、バーコードリーダーをUSBポートに接続してから、以下のボタンを押してバーコードの設定画面を表示します。



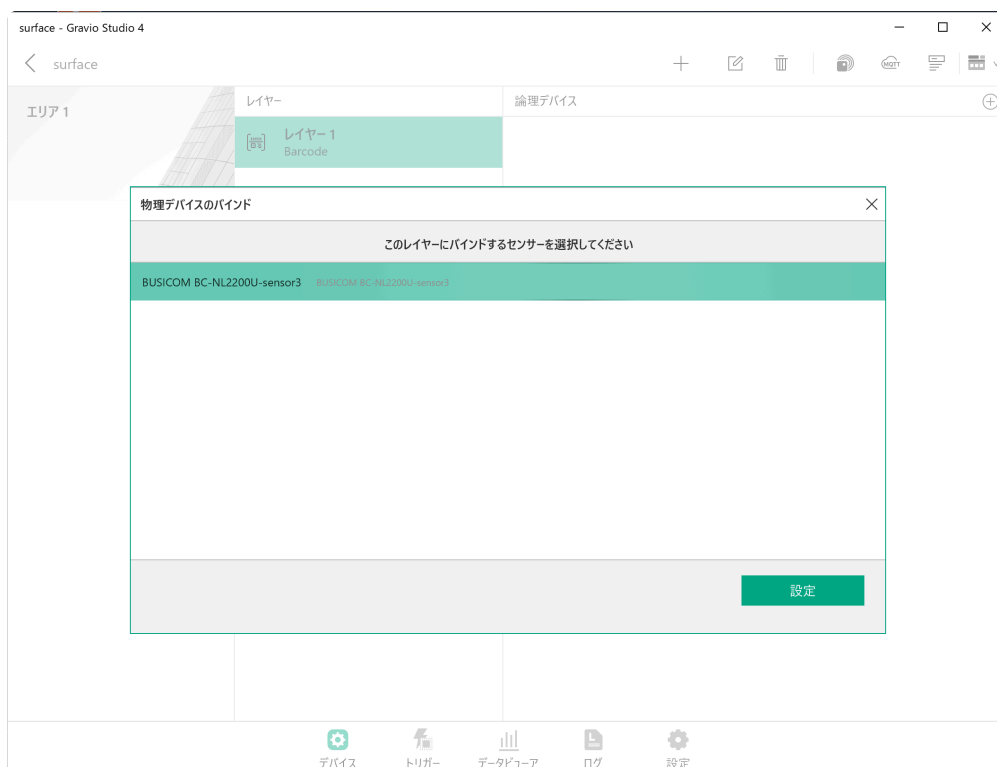
注)BUSICOM BC-NL2200Uが表示されない場合には、バーコードリーダーが正しくUSBポートに接続されているか、USB COM ポートエミュレーション機能に設定されているかを確認して、HubKitを再起動してください。

BUSICOM BC-NL2200Uが表示されている場合には完了です。ダイアログを閉じてください。

3. 次に、ウィンドウの右上にある丸で囲まれた「+」マークをクリックしてください。



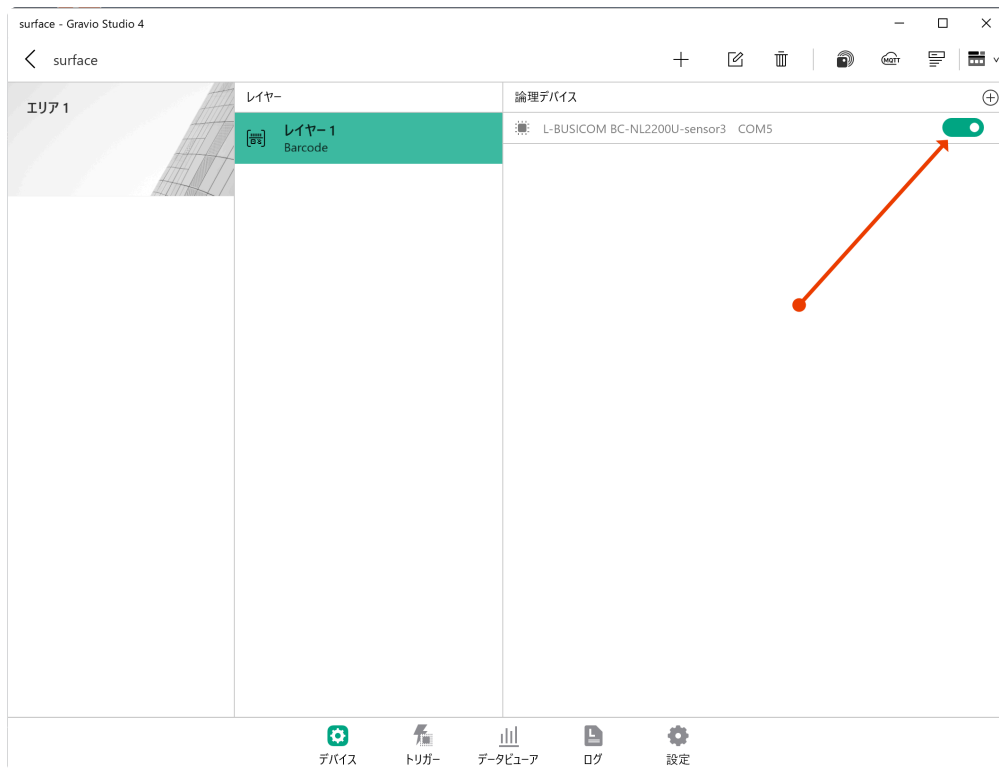
4. レイヤーに接続したいバーコードリーダーを選択して、設定します。



設定が終わったら、この画面と閉じてください。

5. 以下のような画面になります。最後にこのバーコードリーダーからデータを受信ために、スイッチをONにしてくださ

い。



以上でバーコードリーダーでバーコードを読み込むと、データの受信が開始されます。

7.3.1.1.6. GPSデバイスの追加

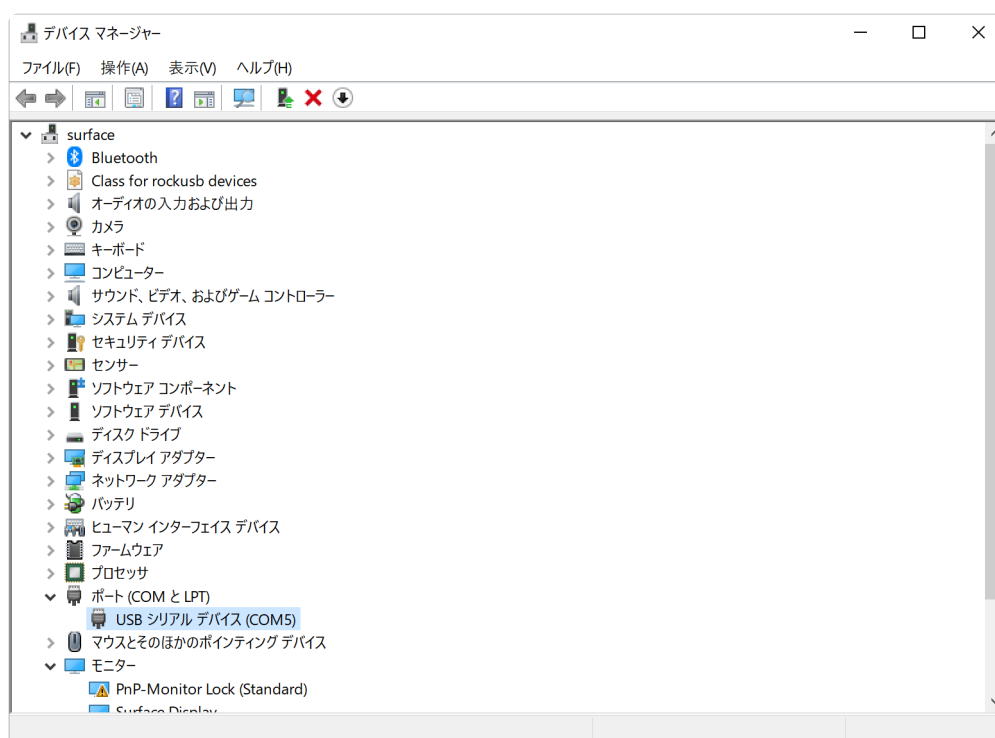
Gravio Studioを使ってGPSデバイスとレイヤーを紐づけます

GravioではGPSデバイス(**COM ポート接続で利用できるGPSデバイス**) を1つのレイヤーとしてデータ入力できるデバイスとして利用できます。

現在動作確認が取れているGPSデバイスは [こちら](#) を参照してください。

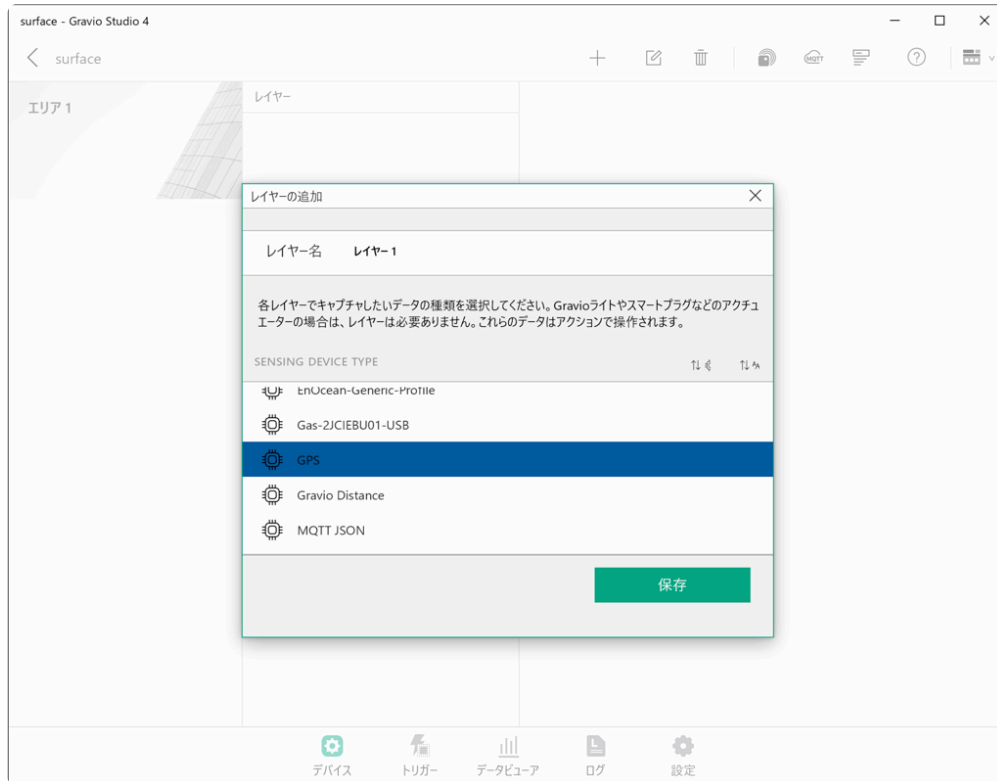
GPSデバイスの接続確認

GPSデバイスの接続確認はWindows10の場合、デバイスマネージャーにCOMx(この場合COM5)が表示されているかを確認してください。



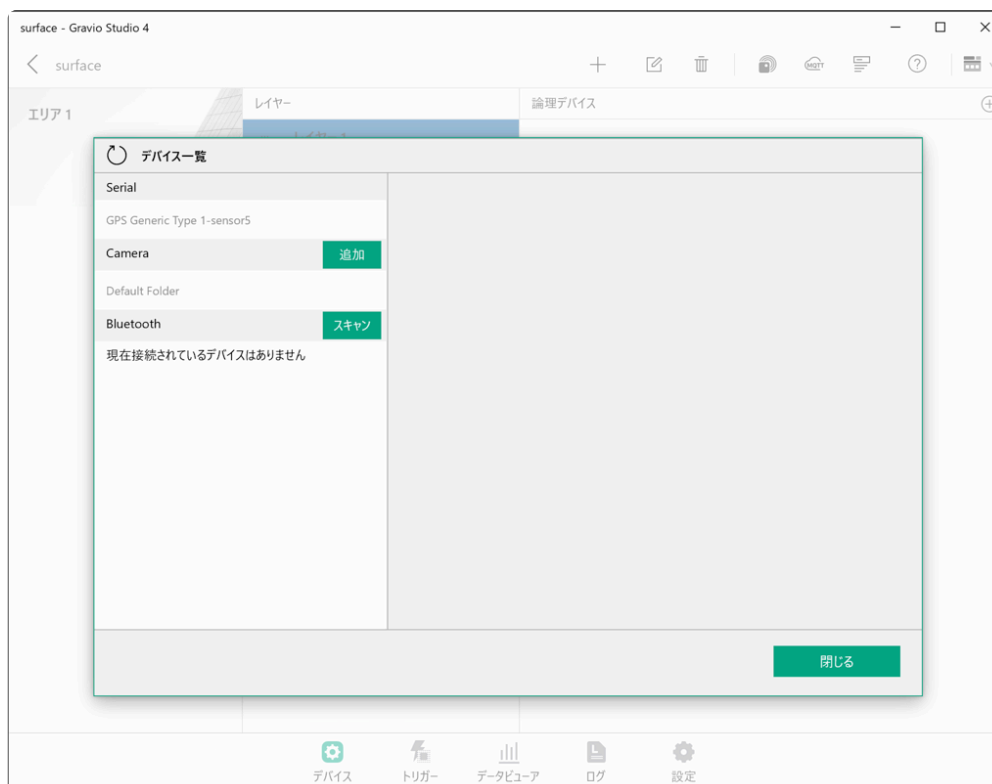
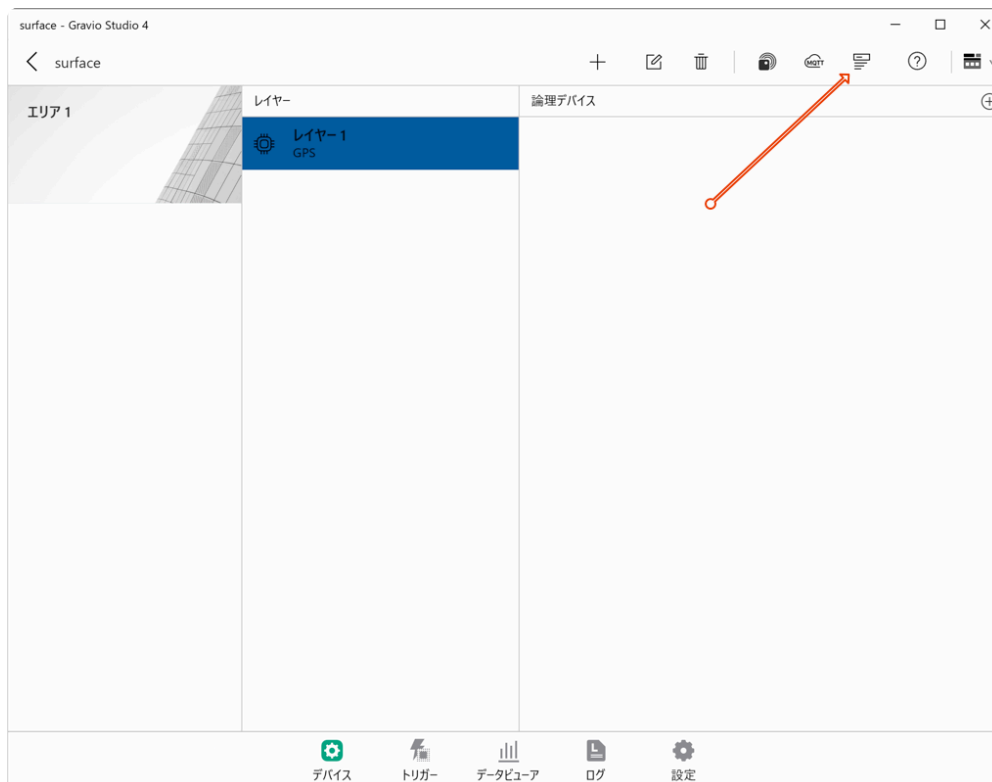
GPSデバイスの追加

1. 「デバイス」で、使用するGPS(DataKind)を選択して新しいエリアとレイヤーを追加します。



ここでは、GPSデバイスは VKLSVAN製 VK172 G-MOUSE USB を使用してGPSのデータを取得する例として説明します。

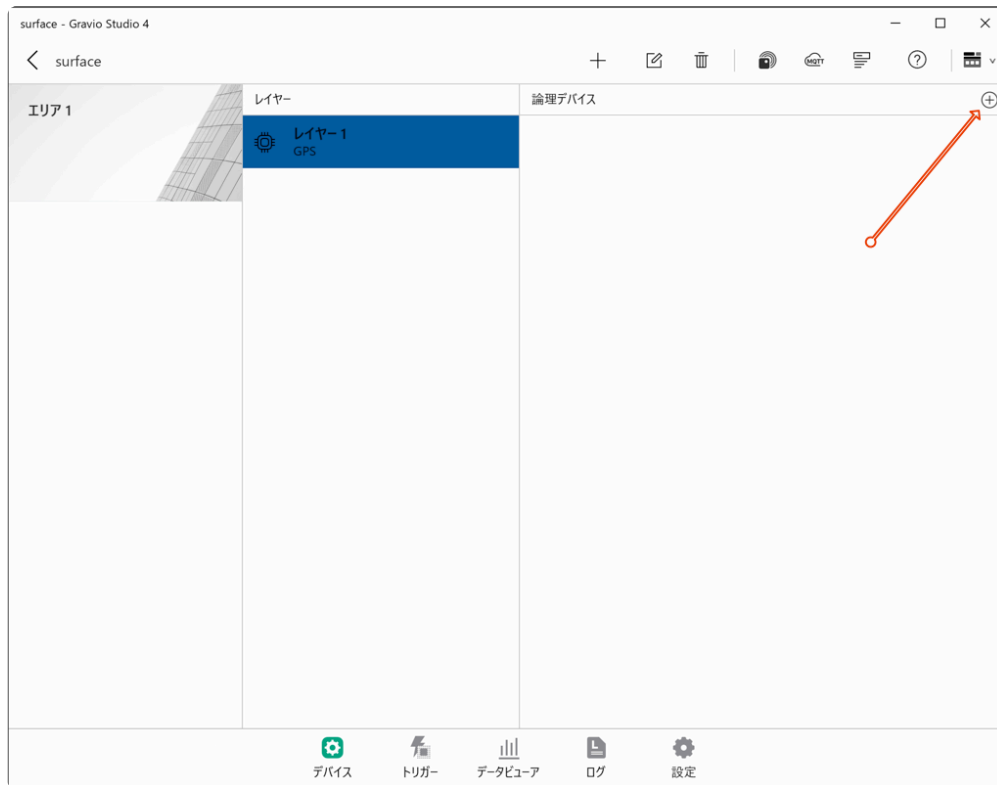
2. GPSデータを受信するには、GPSデバイスをUSBポートに接続してから、以下のボタンを押してGPSデバイスの設定画面を表示します。



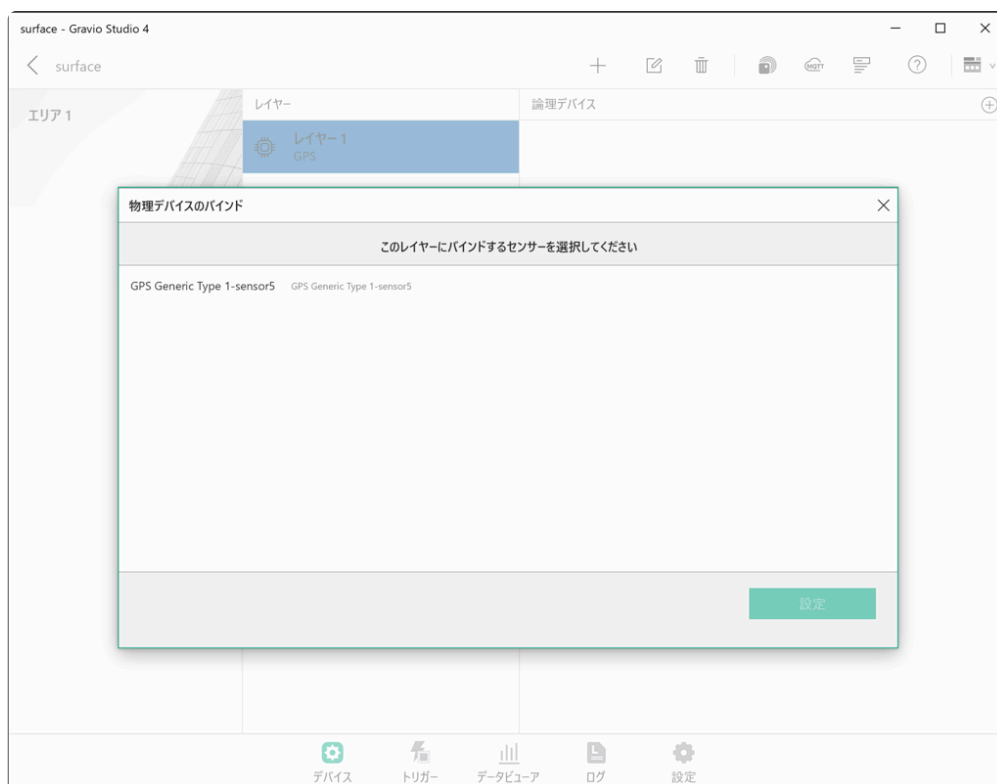
注) GPS Generic Typeが表示されない場合には、GPSデバイスが正しくUSBポートに接続されているかを確認して、HubKitを再起動してください。

GPS Generic Typeが表示されている場合には完了です。ダイアログを閉じてください。

3. 次に、ウィンドウの右上にある丸で囲まれた「+」マークをクリックしてください。

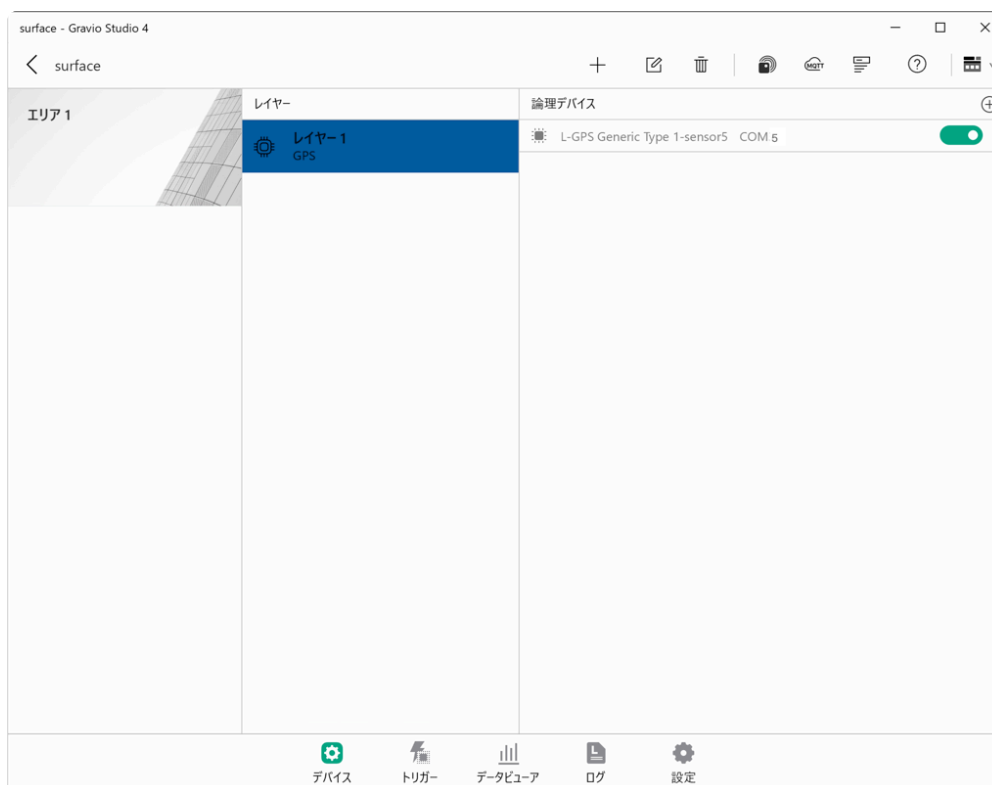


4. レイヤーに接続したいGPSデバイスを選択して、設定します。



設定が終わったら、この画面と閉じてください。

5. 以下のような画面になります。最後にこのGPSデバイスからデータを受信するために、スイッチをONにしてください。



以上でGPSデバイスから自動的にGPSデータの受信が開始されます。

6. GPSデータのフォーマットは以下の通りです。

受信データ

レイヤー設定されたGPSから受信できるデータは下記の通りです。データビューアで確認することが出来ます。

データ	説明
Course	デバイスの移動方位 (北 0度から時計回りで359.9度までの数値)-1の場合は方位は無効
SpeedKm	デバイスの移動速度 (km/h)
SpeedKn	デバイスの移動速度 (knots)
LatitudeDeg	緯度 (DEG形式)
LongitudeDeg	経度 (DEG形式)
Latitude	緯度 (DMM形式)
Longitude	経度 (DMM形式)
LatitudeDir	緯度の方位 (NまたはS)

LongitudeDir	経度の方位 (EまたはW)
--------------	---------------

用語

・DEG形式

ddd.dddのデータ形式で、ddd.dddは度数で小数点で表記されます。Googleマップで使用されている表記です。

・DMM形式

dddmm.mmmmのデータ形式で、dddは度数、mm.mmmmは分数で、60進法ですので60分で1度となります。Googleマップで使用されている ddd.dddd度表記は、 $(\text{度数} + \text{分数}/60)$ で計算することができます。

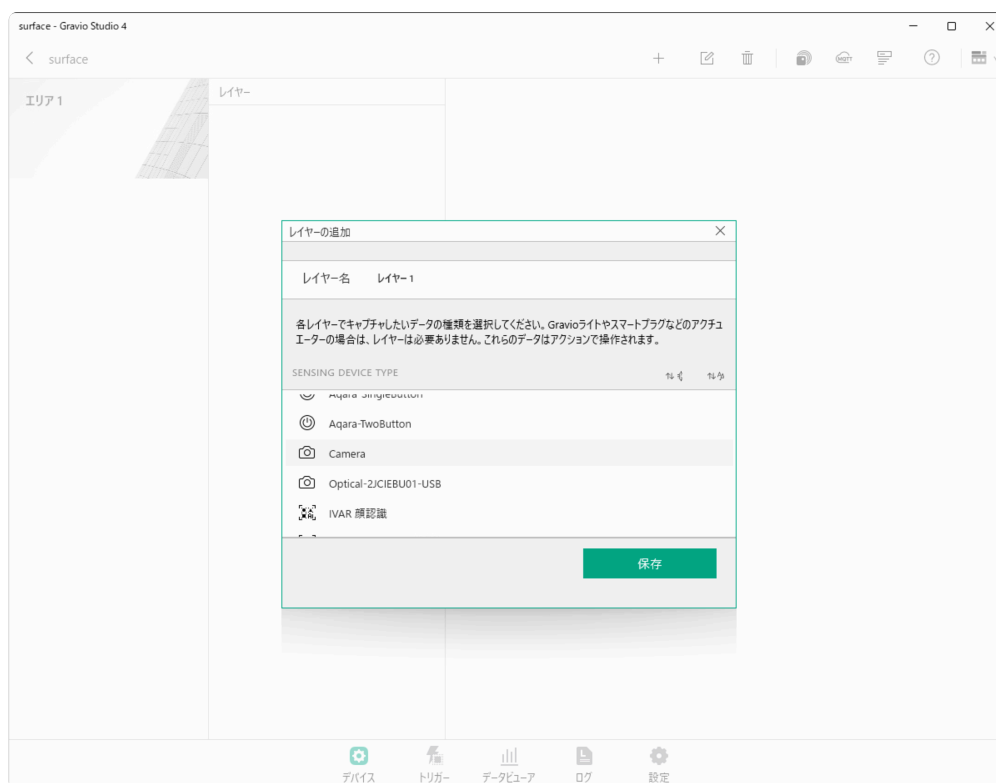
7.3.1.2. ソフトウェアセンサー

Gravioはソフトウェアセンサーもサポートしています。ソフトウェアセンサーの例は、部屋にいる人の数を検出し、特定のしきい値に達すると特定のアクションをトリガーするカメラです。これらのセンサーは、ソフトウェアおよび人工知能（推論ファイル）を使用して作成されます。

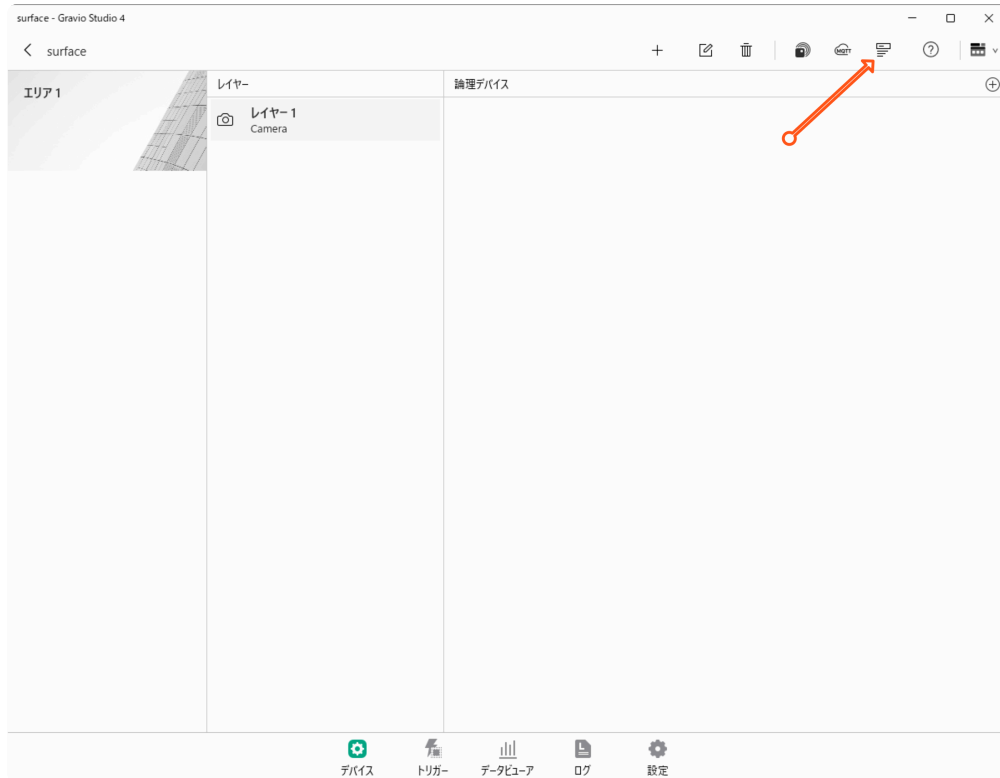
ソフトウェアセンサーの作成

Gravioで利用できるカメラデバイスはUSBのシリアル接続が可能な場合および、ONVIF (Profile T)をサポートしたカメラでデバイスになります。

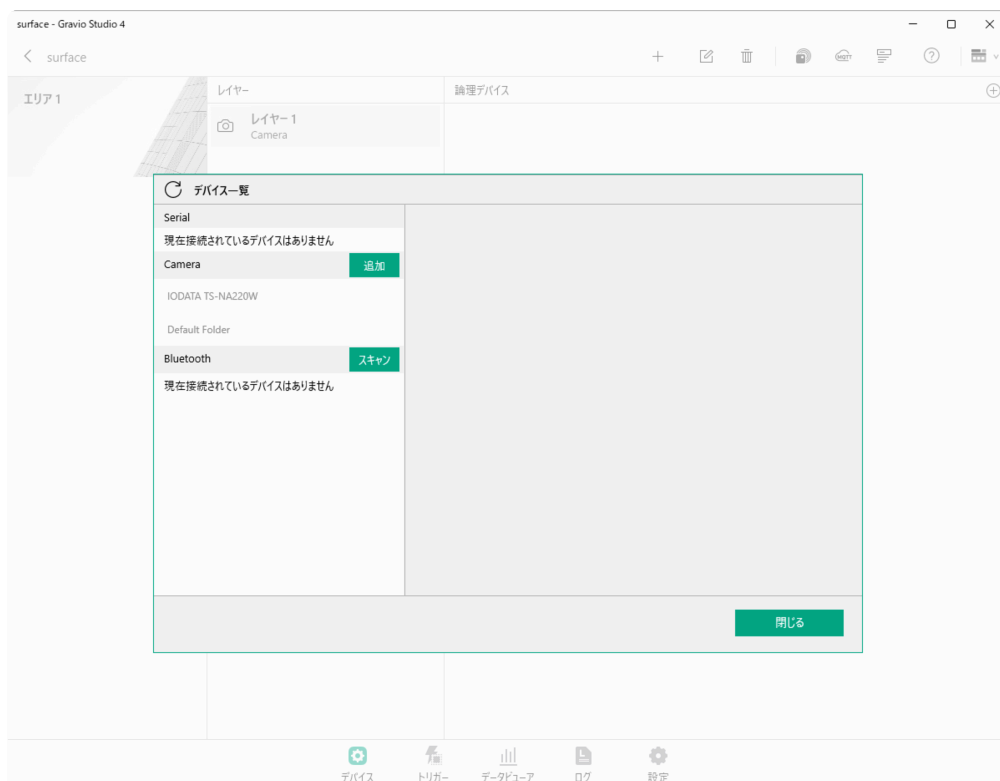
1. 「デバイス」で、カメラのセンサーの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。



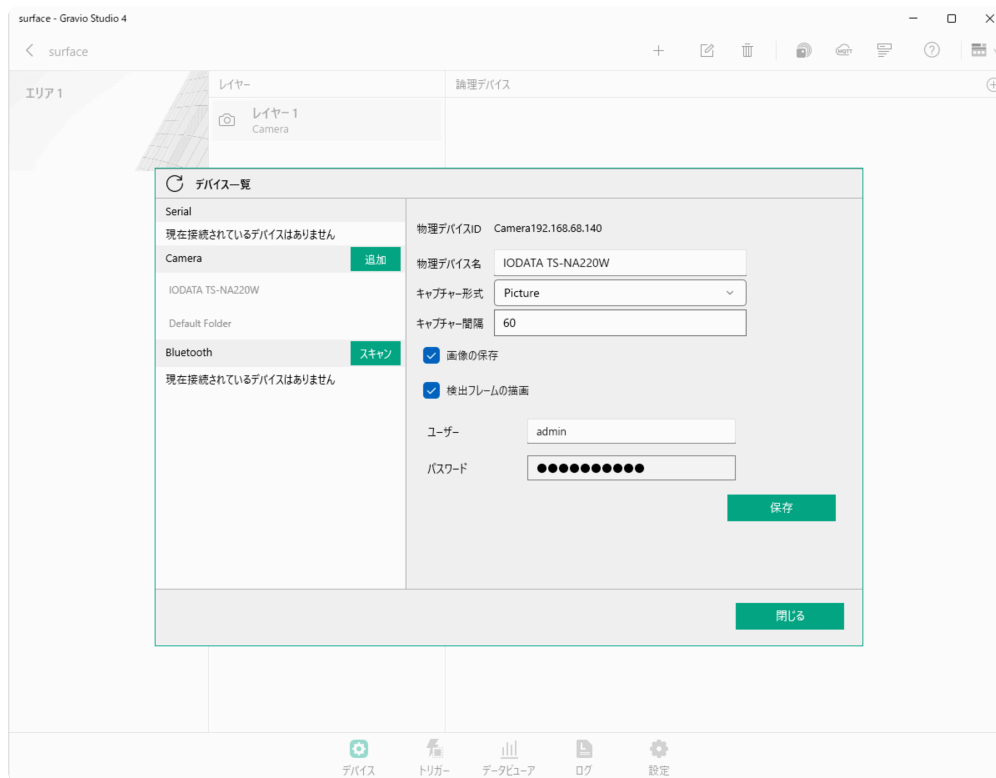
2. 以下のボタンを押して、OnVifとして認識されているカメラの設定画面を表示します。



3. 新しく追加されたカメラデバイスをクリックし、プロパティを設定します

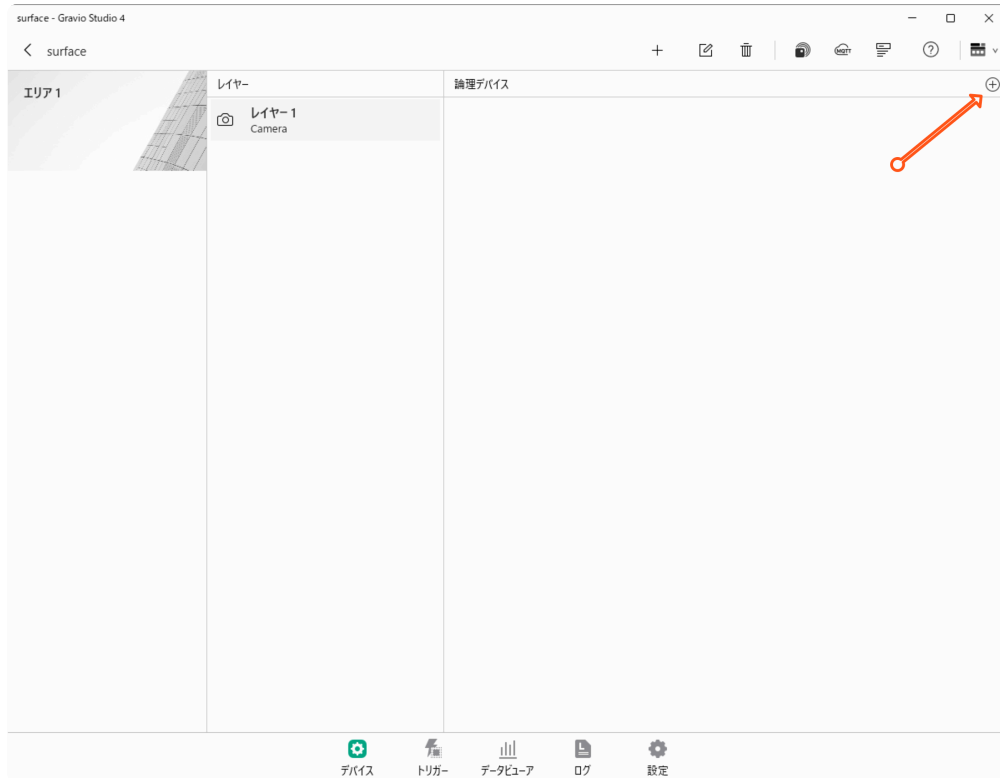


4. 以下の設定をプロパティーとして設定できます。

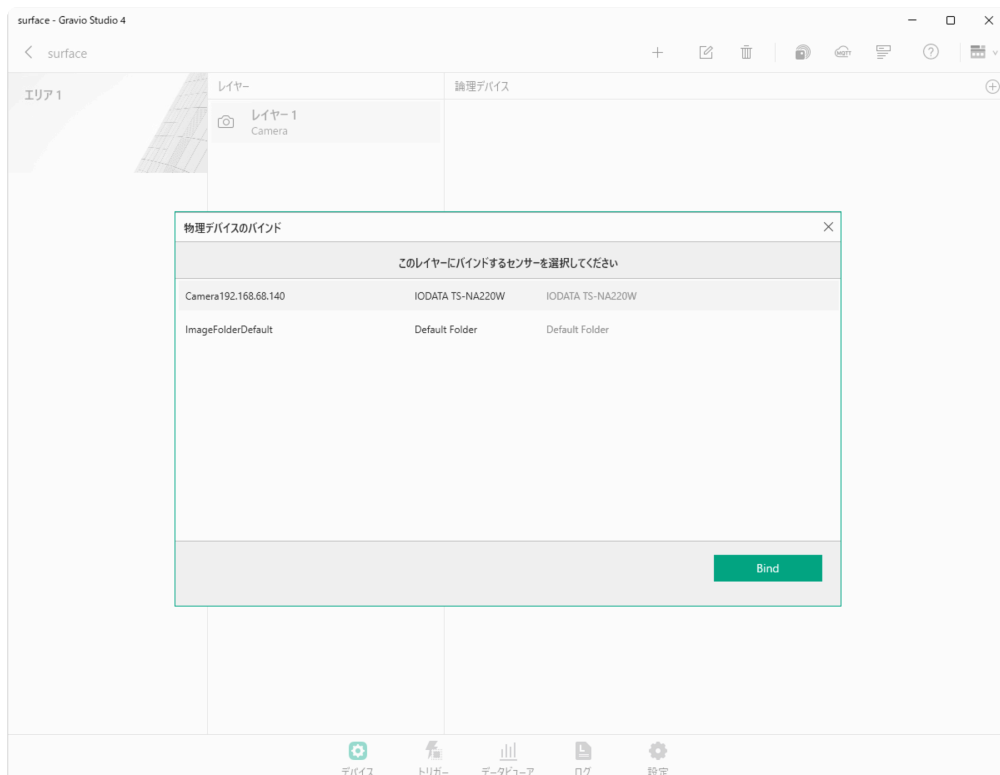


- ・カメラの種類のビデオまたは画像を選択します。画像の場合はjpeg形式となります。ビデオの場合はmp4形式となります。
- ・撮影する頻度（秒）
- ・あなたが使用したい推論ファイル、例えば “NumberOfPeople”
- ・画像を保存するか、処理した直後に破棄することができます。画像を保存する場合、Linuxなら /var/opt/hubkit/mediadata/に、Windowsなら C:\ProgramData\HubKit\mediadataに、Macなら /Library/Application Support/HubKit/mediadata に保存されます。

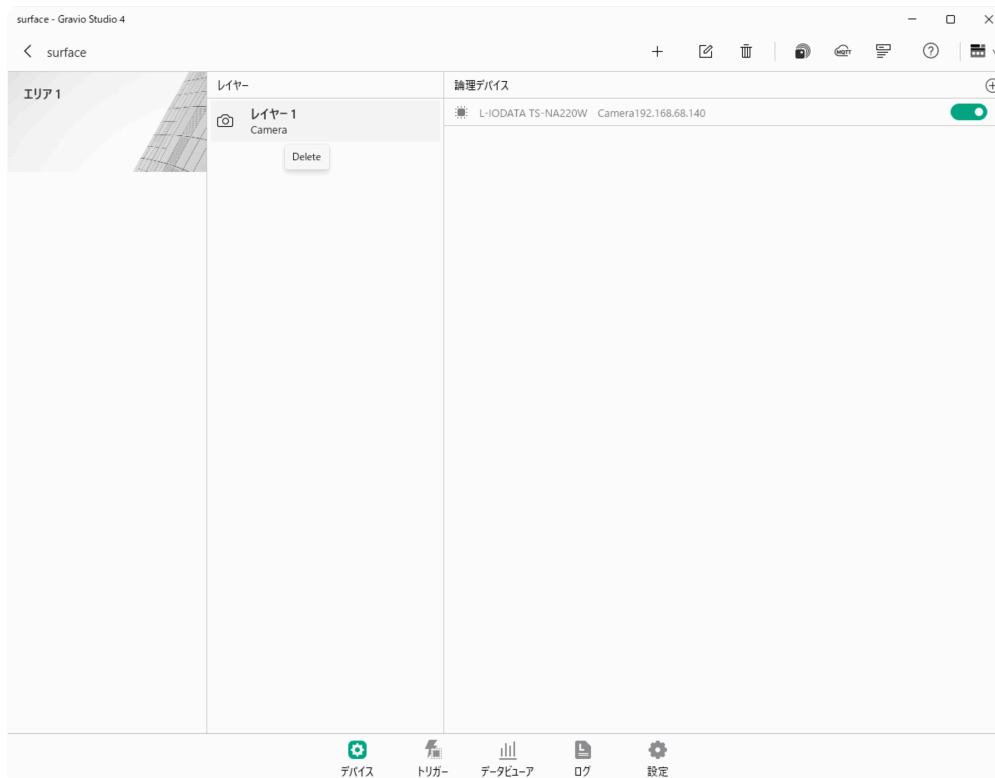
5. +ボタンをクリックしてバインド設定画面を表示します。



6. 設定したカメラをレイヤーにバインドにします。



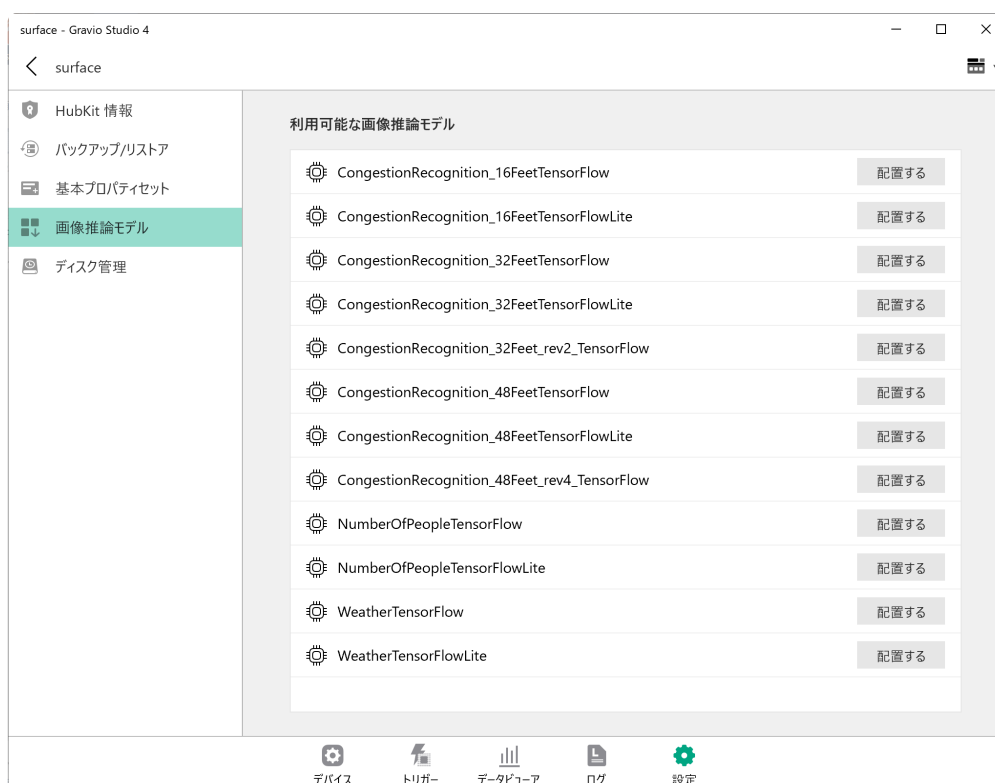
7. 最後にセンサー取得をONにします。



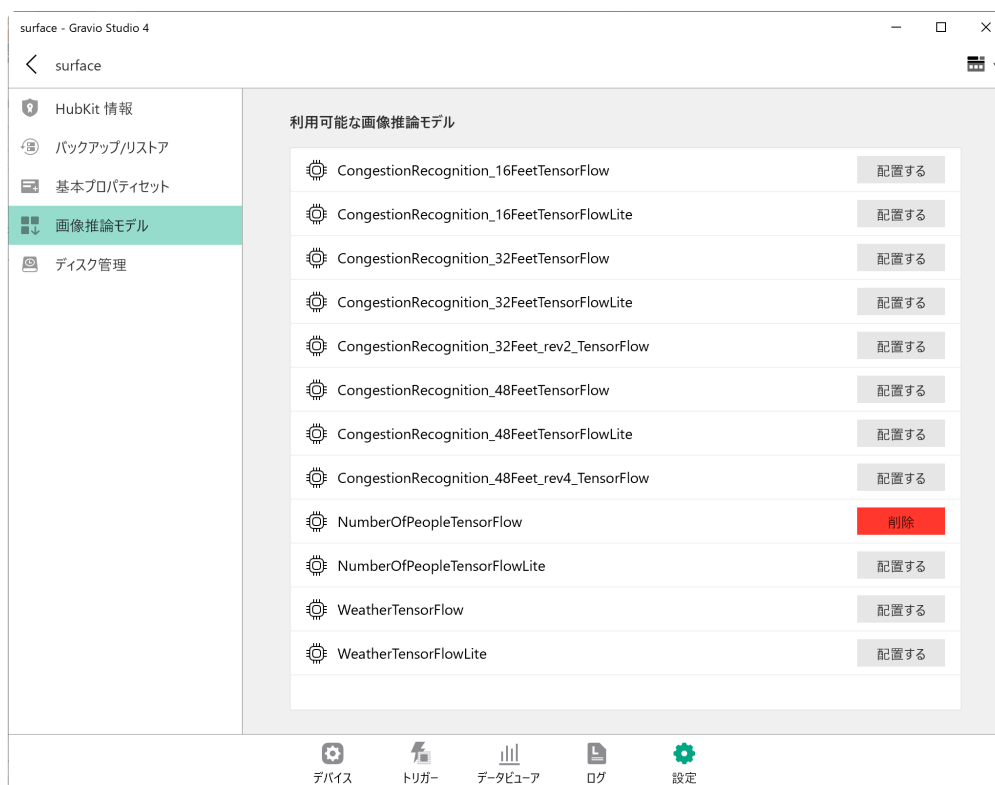
推論ファイルを使用する

Gravio Freeアカウントのサービスを契約されている方は1つの推論ファイルを、Gravio Basicアカウント以上のサービスを契約されている方は、1つ以上の推論ファイルをGravio Cloudからダウンロードしてご利用いただけます。

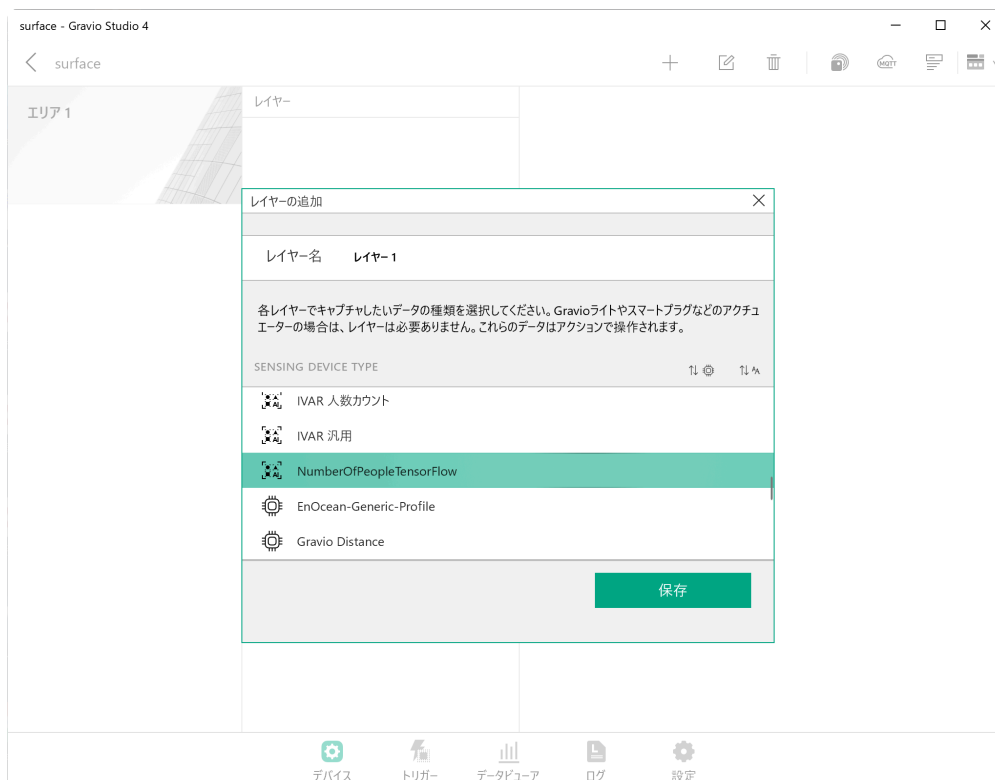
1. 設定画面の画像推論モデルを表示します。



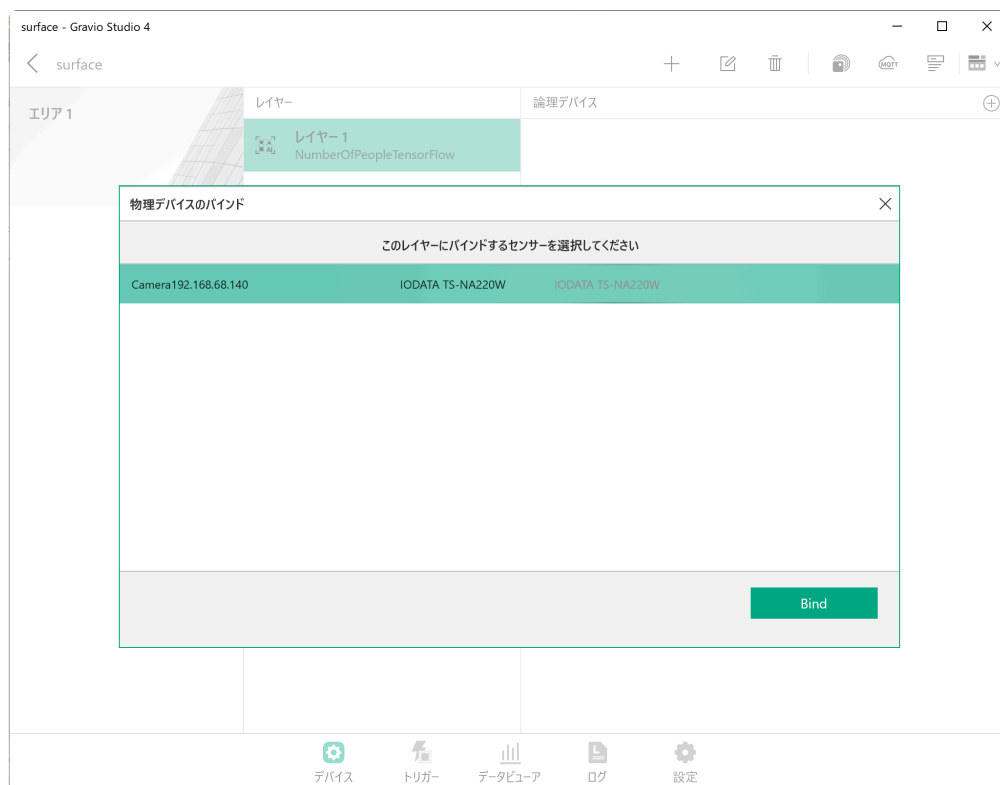
2. 推論するモデルを配置するボタンを押してダウンロードします。



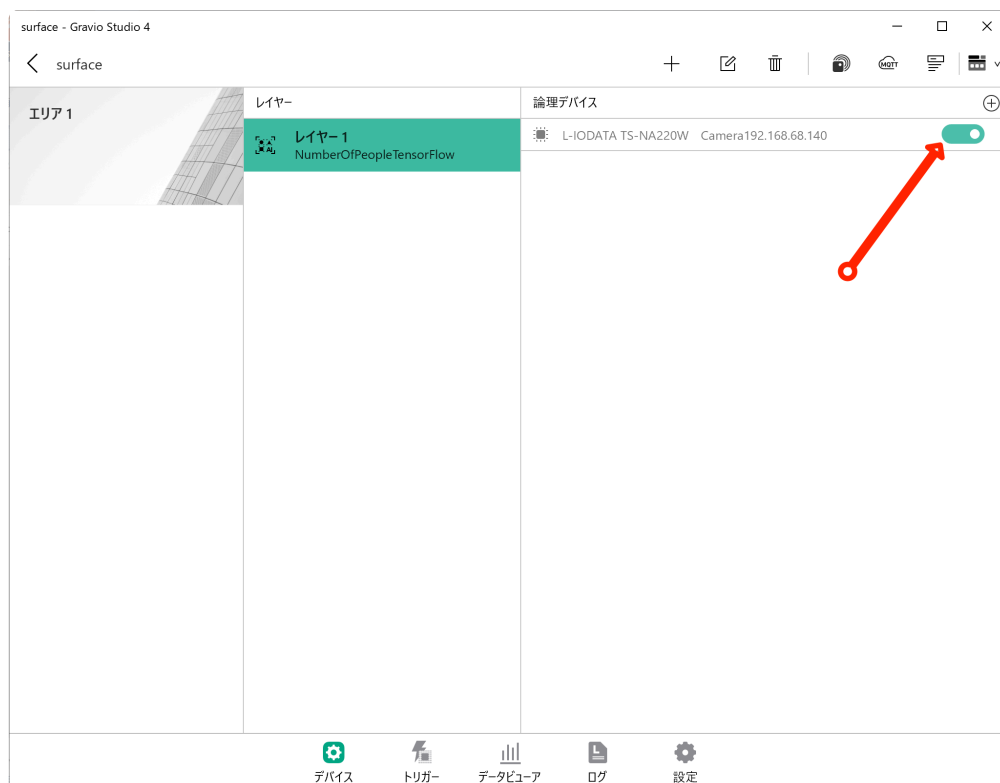
3. 推論ファイルを使用する場合にはレイヤーを作成するときにCameraではなく推論ファイルを選択します。



4. 設定した推論ファイルをレイヤーにバインドにします。



5. 最後にセンサー取得をONにします。



推論ファイルを使用する時のパフォーマンスの目安

推論ファイルを使用する場合のハードウェア環境につきましては [こちら](#) を確認してください。

7.3.1.2.1. 推論モデル

GravioではTensorFlow(テンサーフロー)を使用してカメラの画像を推論することができます。

TensorFlowのファイルは*TensorFlowと*TensorFlowLiteの2種類を用意しており、推論処理のディープラーニングの階層数の違いがあり、*TensorFlowは32階層で、*TensorFlowLiteは16階層となります。

*TensorFlowの方が推論の精度が上がりますが、処理時間と消費するメモリが大きくなりますので用途に合わせてどちらかをご使用ください。

推論ファイル

Gravioでは以下の推論ファイルを使用することができます。

・混雑具合の認識

距離の目安	推論ファイル名
16Feet	CongestionRecognition_16FeetTensorFlow
32Feet	CongestionRecognition_32FeetTensorFlow
488Feet	CongestionRecognition_48FeetTensorFlow

カメラで映るエリア(範囲)の混雑具合を認識します。

16Feet、32Feet、48Feetはカメラが認識するエリアまでの距離の目安となります。

16より48の方が広範囲での認識となりますので、遠くの人を識別するためには、その数が大きいものを使用してください。

・人数カウント

NumberOfPeopleTensorFlow

カメラが認識した人の人数をカウントします。

・天候の認識

WeatherTensorFlow

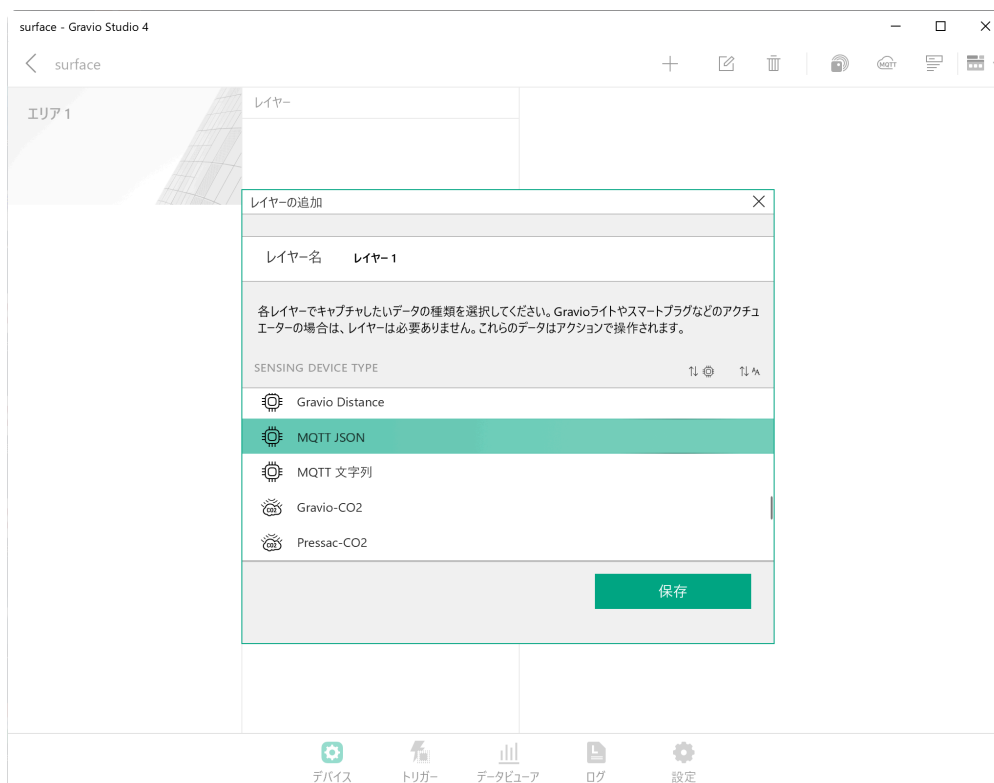
カメラが認識した空の天候を認識します。

7.3.1.3. MQTTサブスクライバー

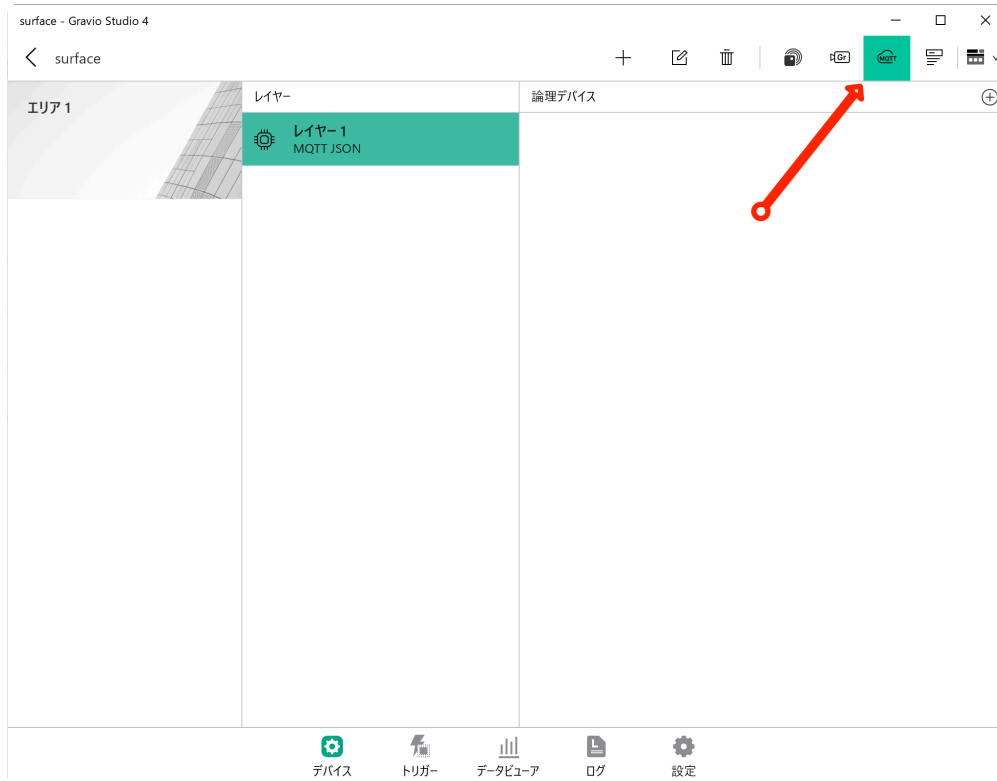
GravioはMQTTサブスクライバーもサポートしています。MQTTブローカーを使用してMQTTパブリッシャーが送信するメッセージをトピックを指定して受信します。

MQTTサブスクライバーの作成

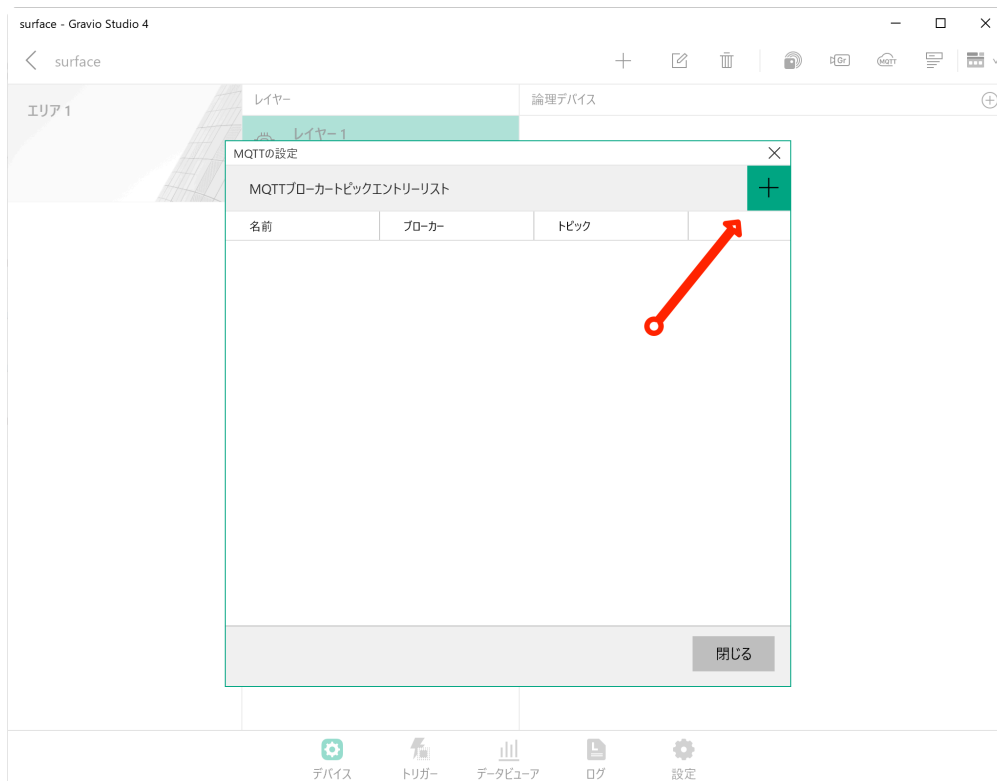
1. 「デバイス」で、MQTTの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。MQTTサブスクライバーは受信するデータの形式がJSON形式か文字列形式かを選択します。



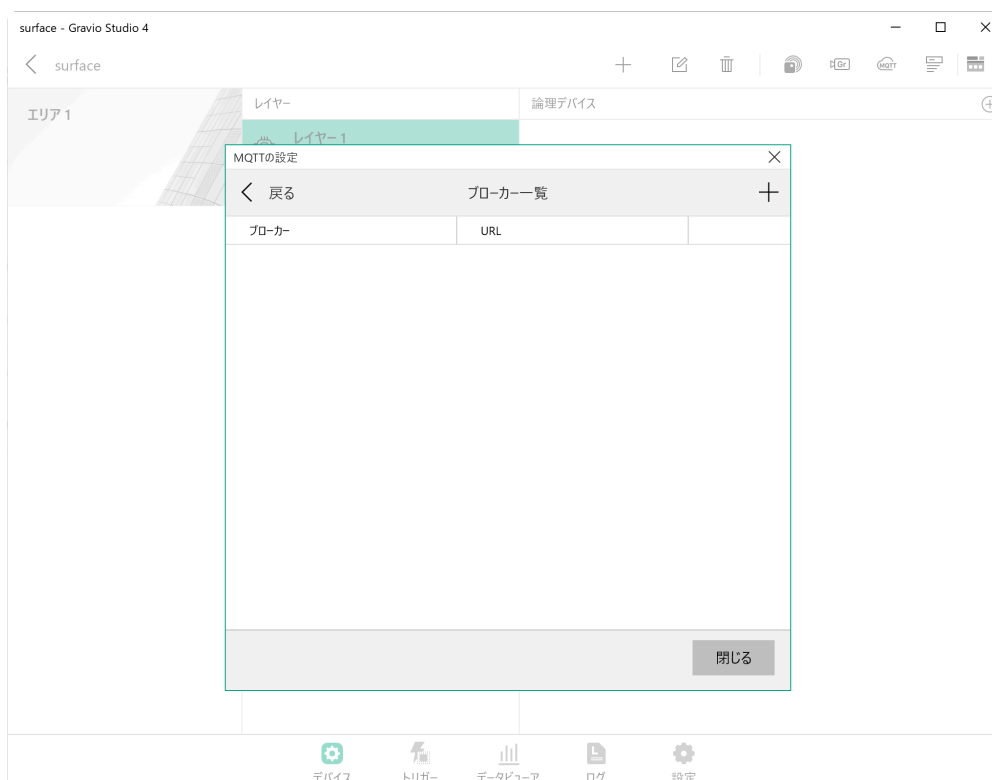
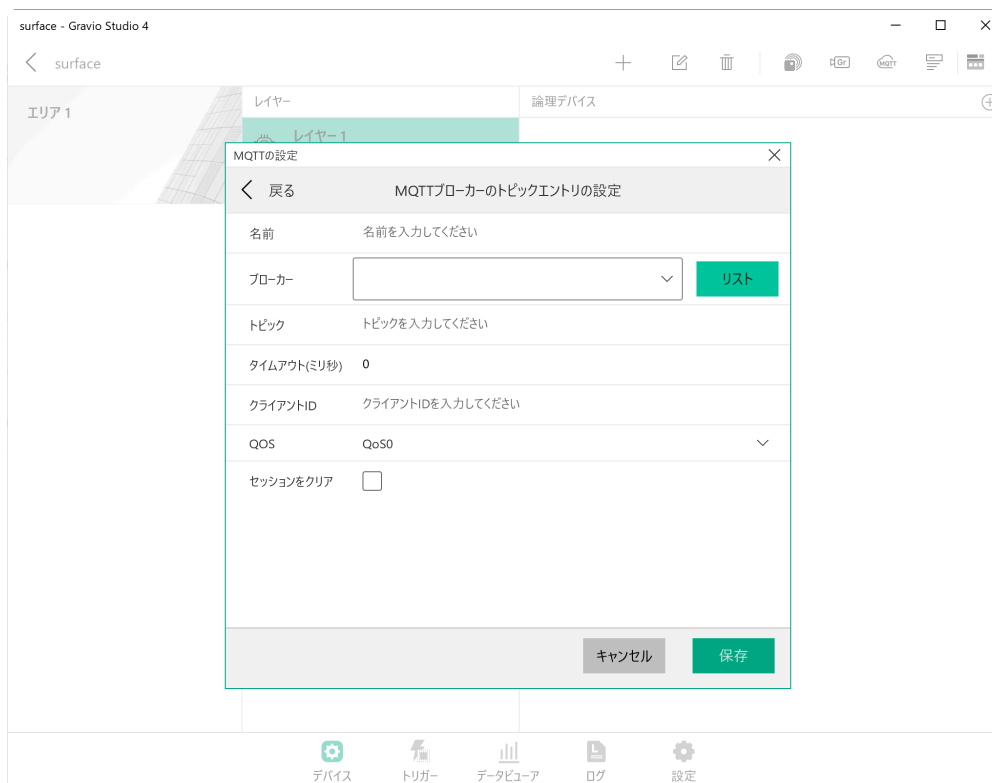
2. 以下のボタンを押して、MQTTサブスクライバー、ブローカーの設定画面を表示します。



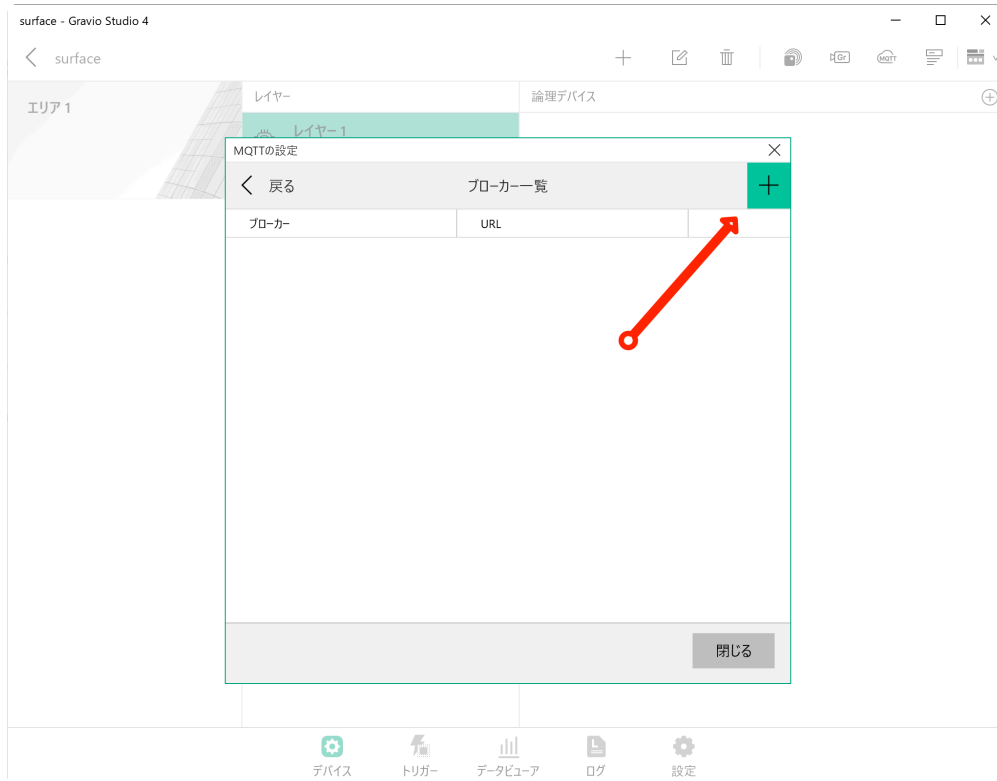
3. +ボタンをクリックしてトピックエントリの設定画面を表示します。



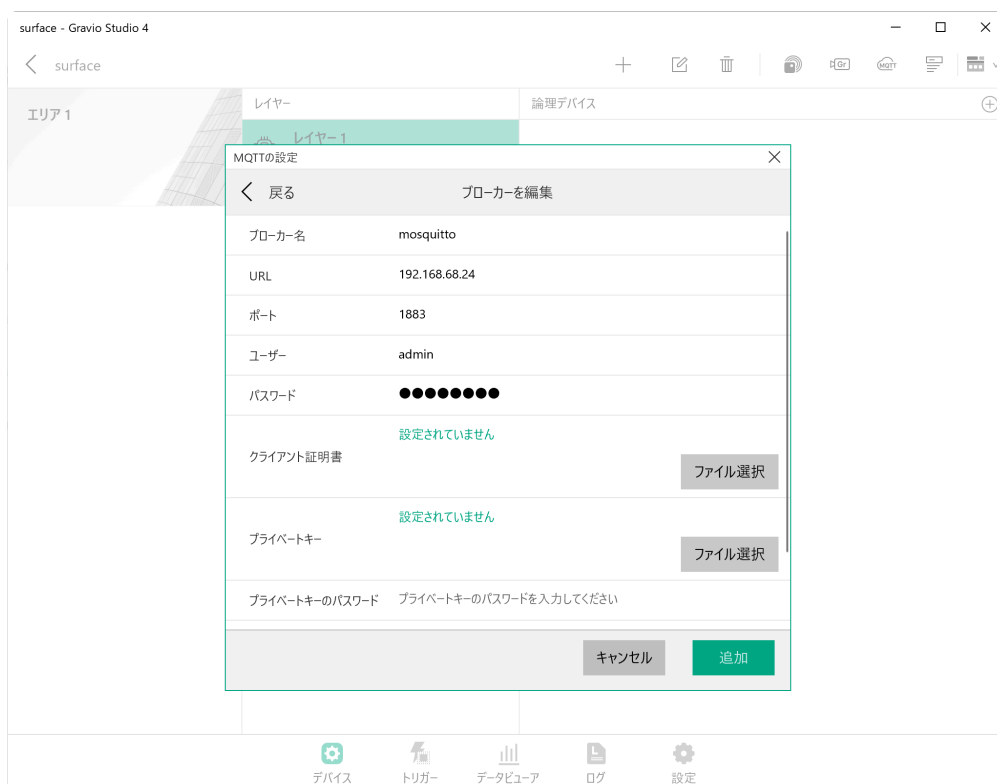
4. リストボタンをクリックしてブローカーの一覧画面を表示します。



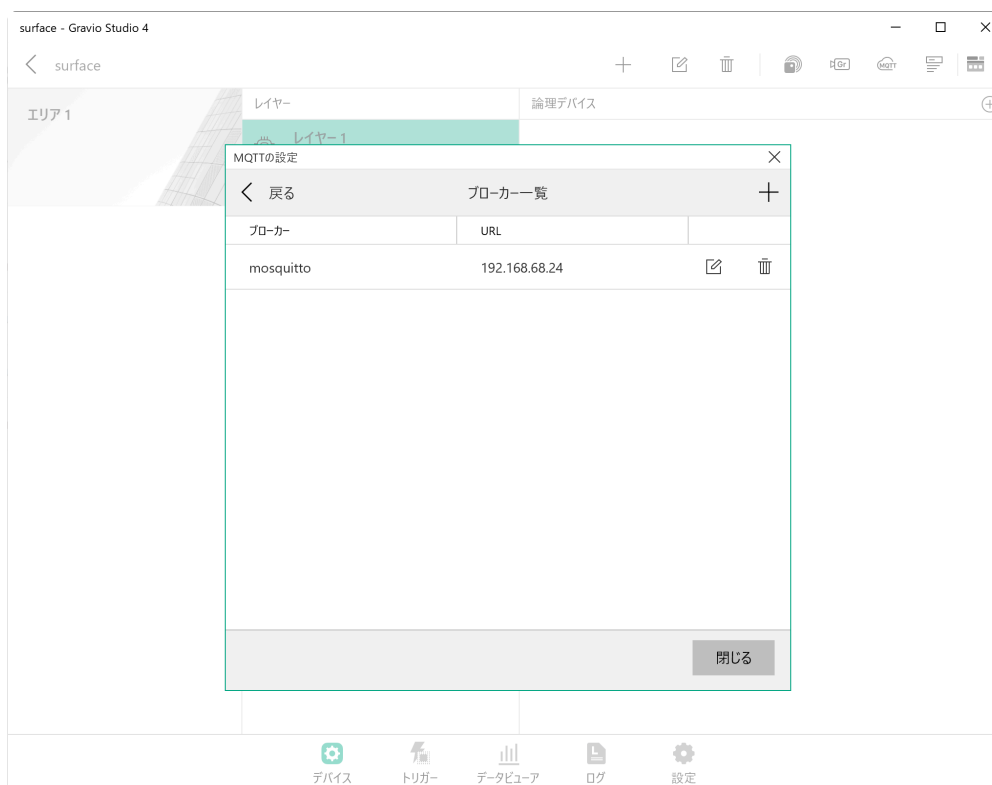
5. +ボタンをクリックしてブローカーの設定画面を表示します。



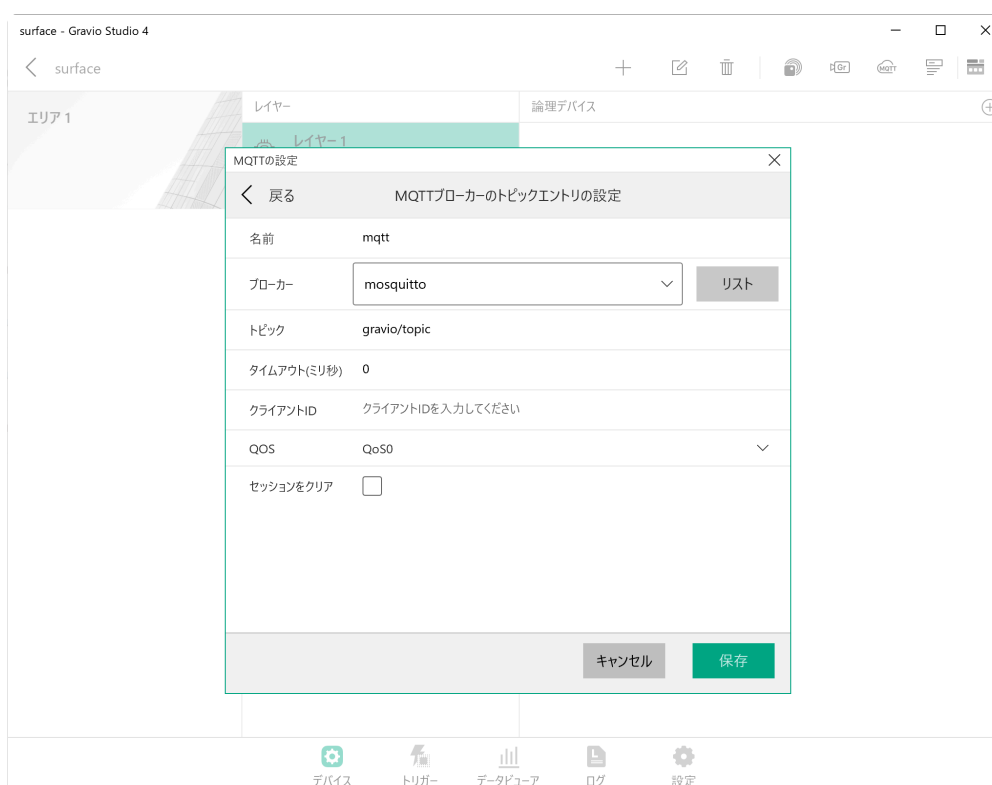
6. ブローカー情報を入力して追加ボタンを押します。



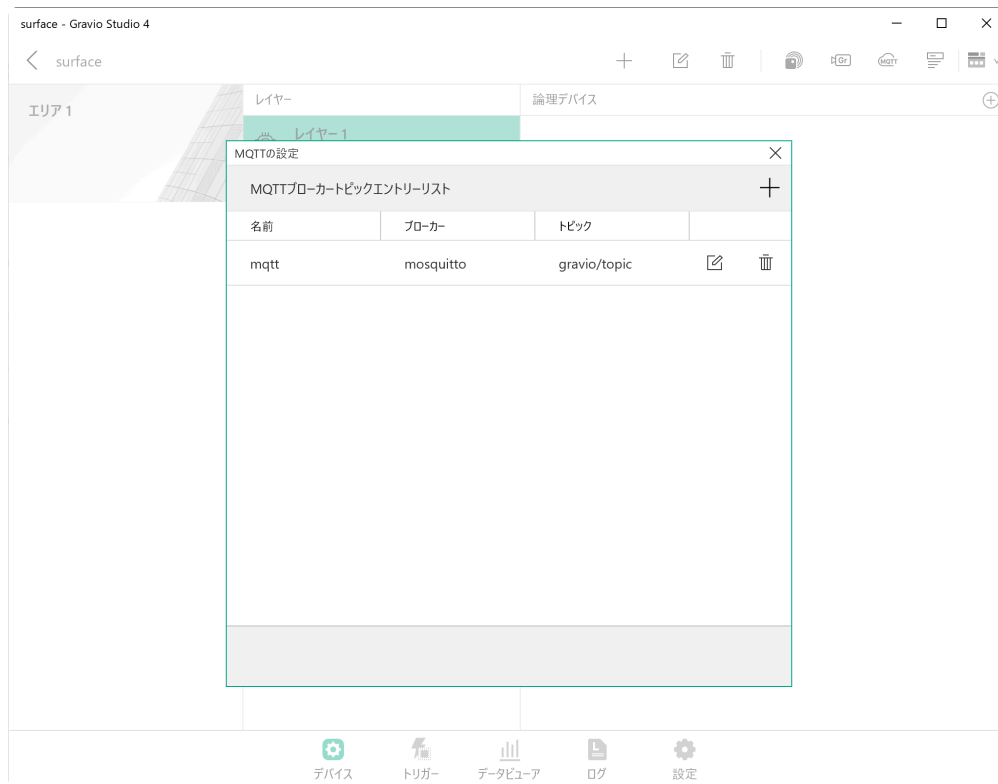
7. 追加したブローカー情報が一覧に表示されるので戻るボタンを押します。



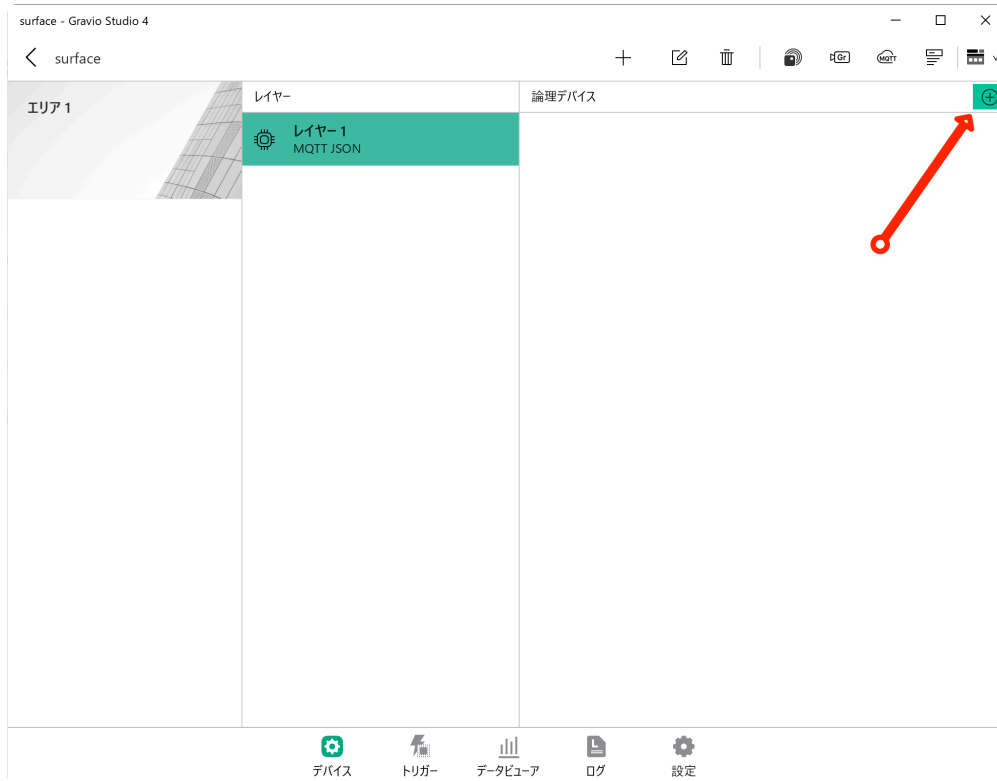
8. トピックエントリ情報を入力して保存ボタンを押します。



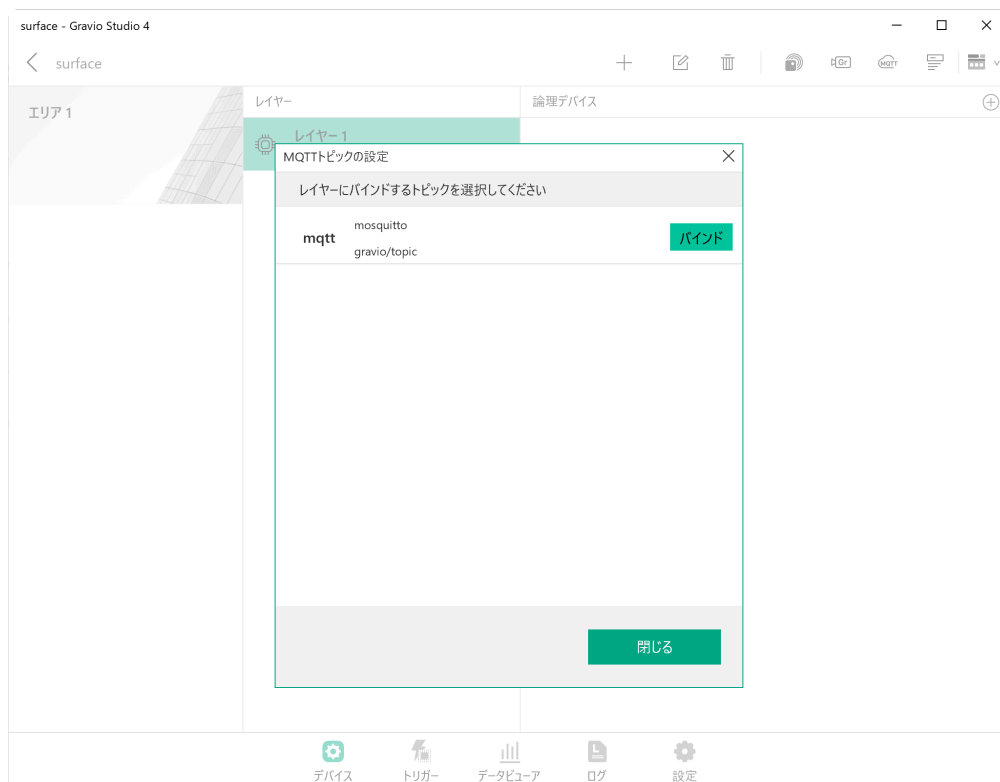
9. 保存したトピックエントリーがリスト表示されますので閉じます。



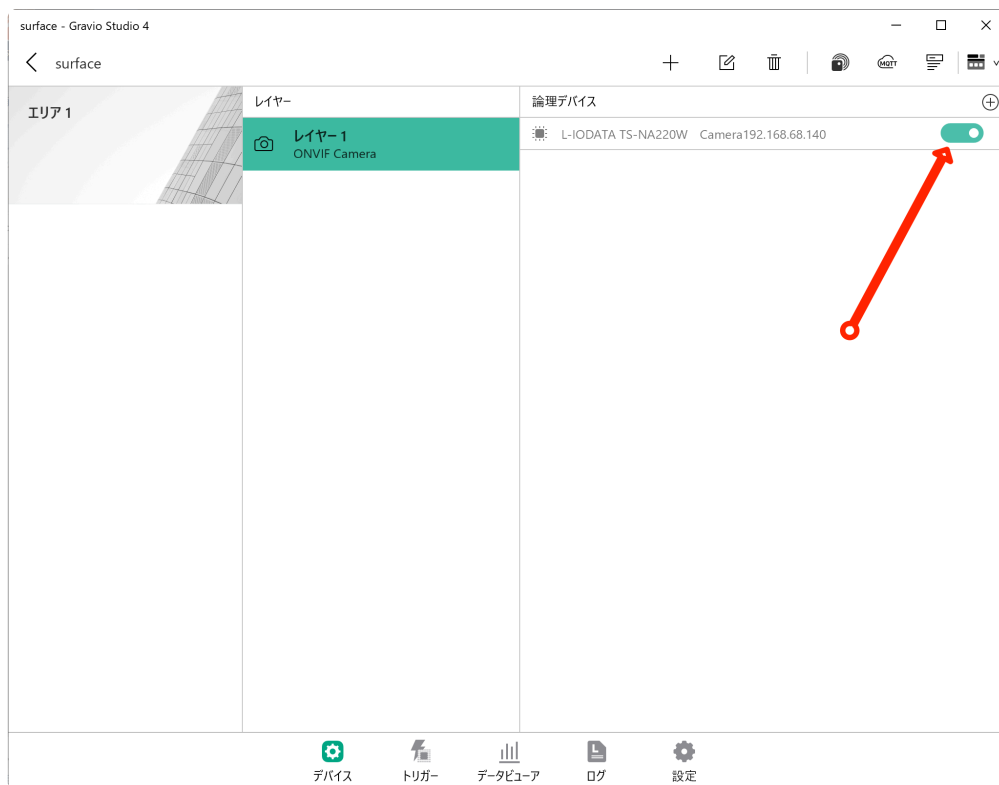
10. +ボタンをクリックしてバインド設定画面を表示します。



1. 設定したMQTTをレイヤーにバインドにします。



1. 最後にセンサー取得をONにします。



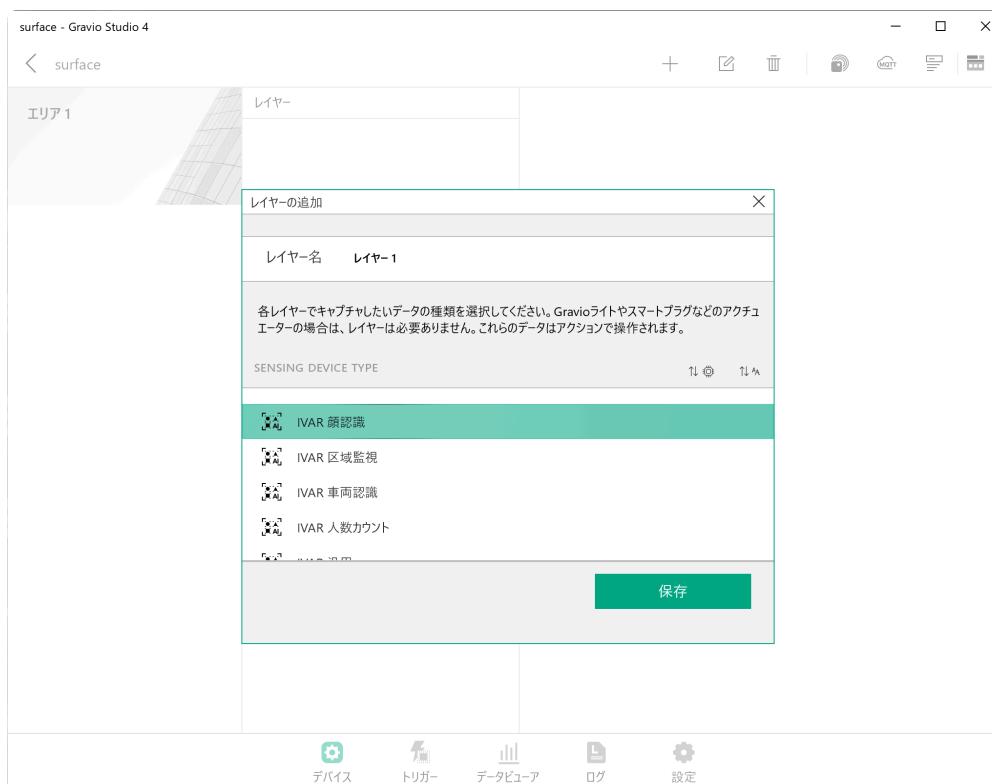
7.3.1.4. Gorilla IVAR

Gorilla IVARと連携しIVARで認識したデータを受信します。

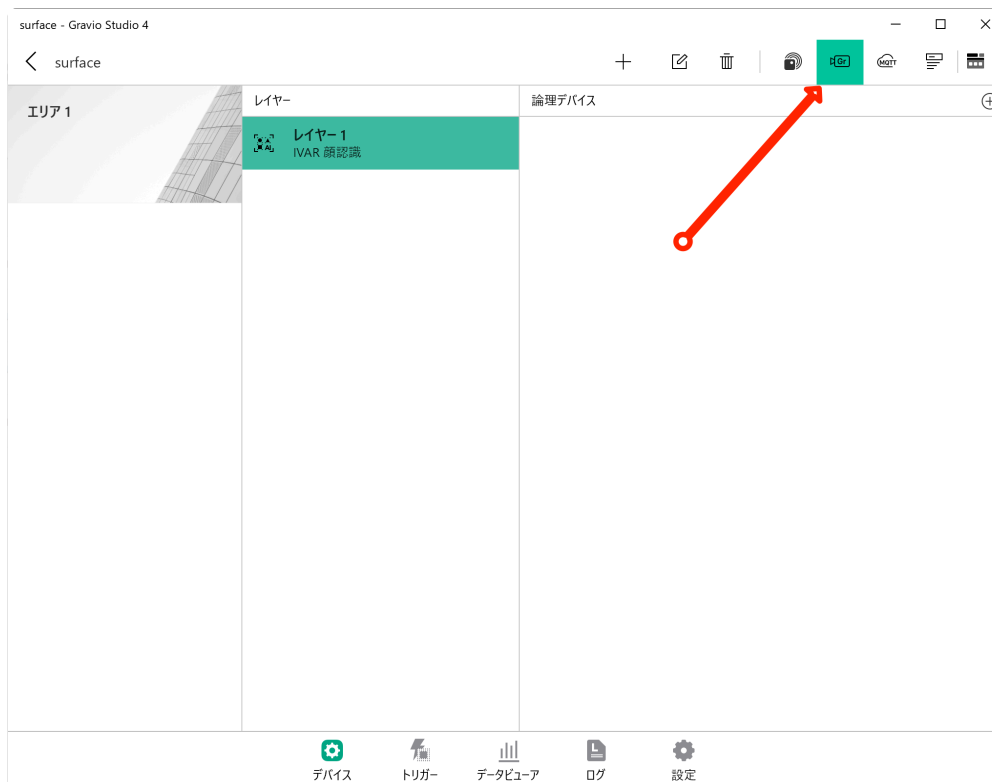
IVARの作成

1. 「デバイス」で、IVARの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。

IVAR は受信するデータの形式は顔認識、区域監視、車両認識、人数カウント、汎用から選択します。

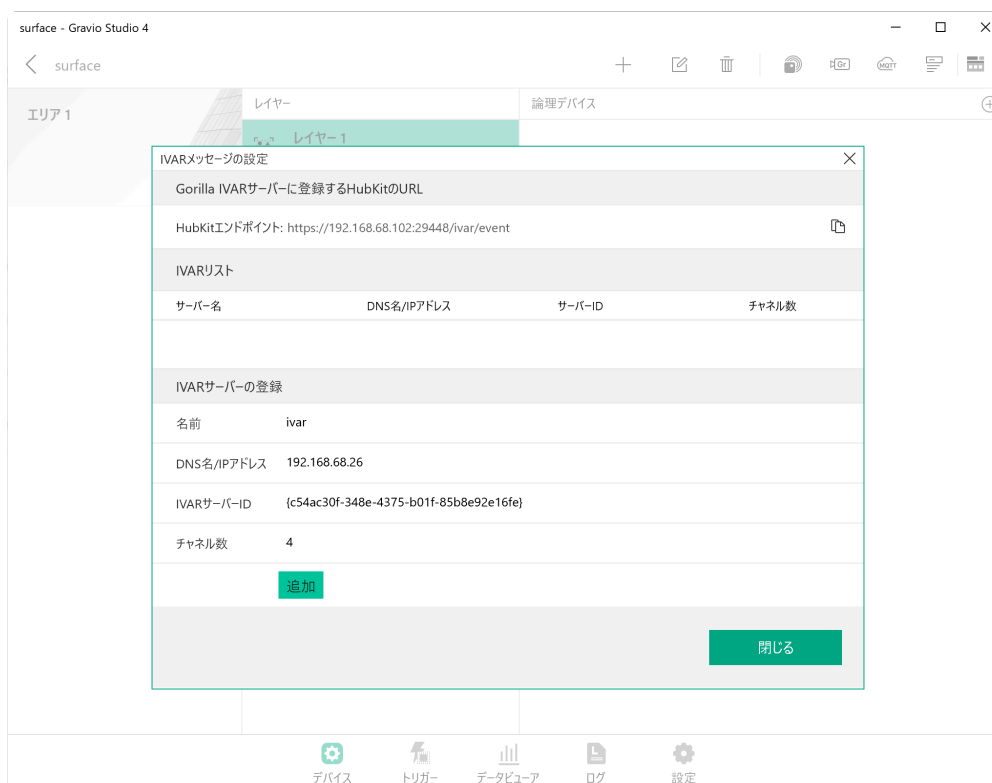


2. 以下のボタンを押して、IVARの設定画面を表示します。

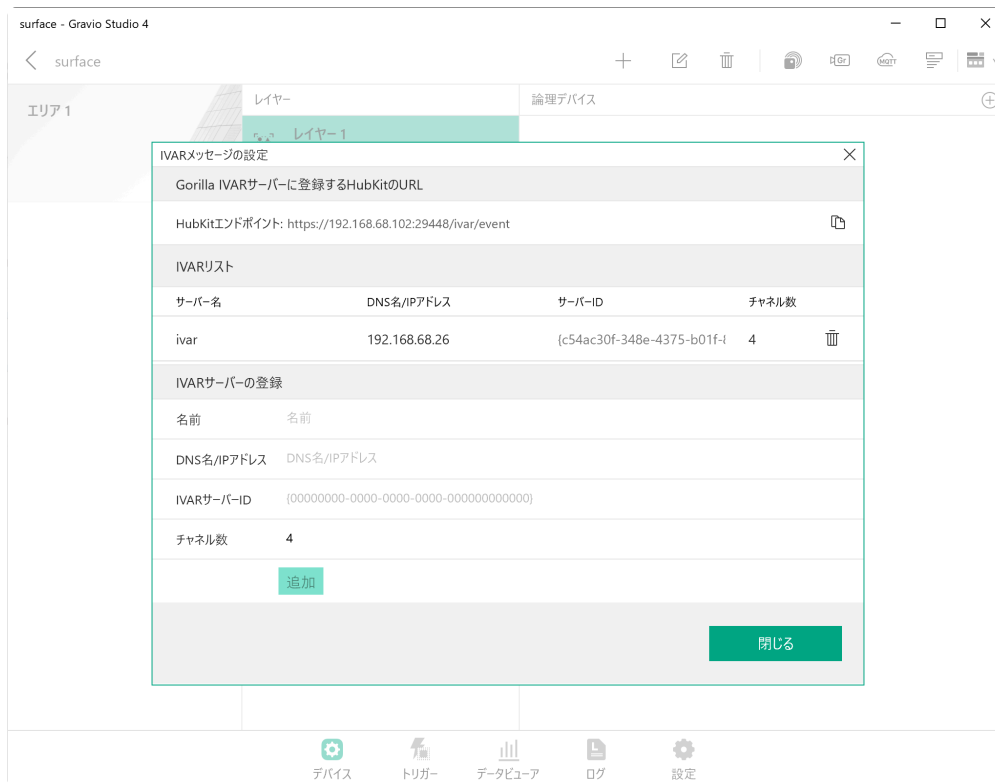


3. IVARを入力して追加ボタンを押します。Gorilla IVARサーバーに登録するHubKitのURLはクリップボードにコピーしてIVARに登録してください。

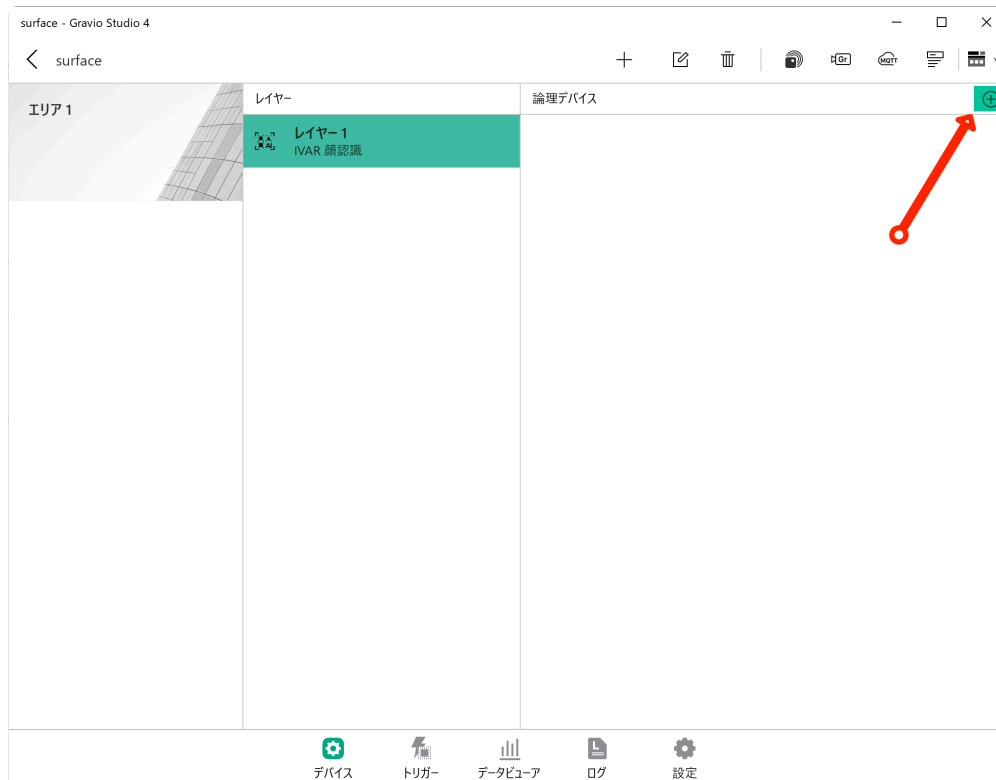
登録の方法はCoordinatorの[「GravioとIVAR連携の設定」](#)を参照してください。



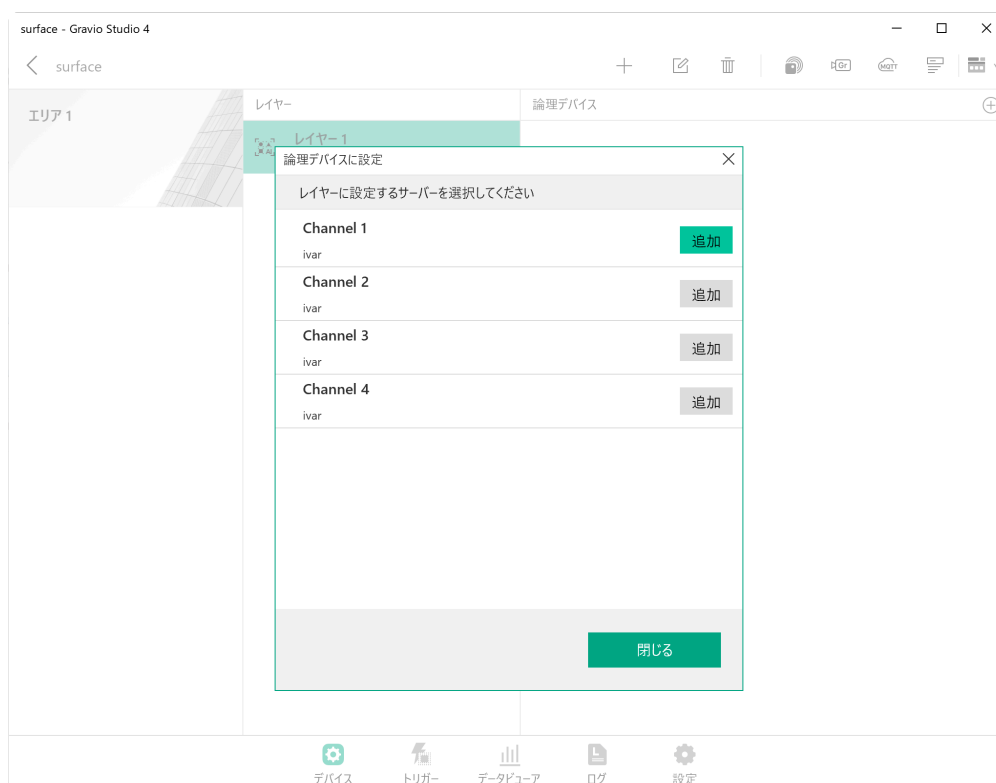
9. 保存したIVAR情報がリスト表示されますので閉じます。



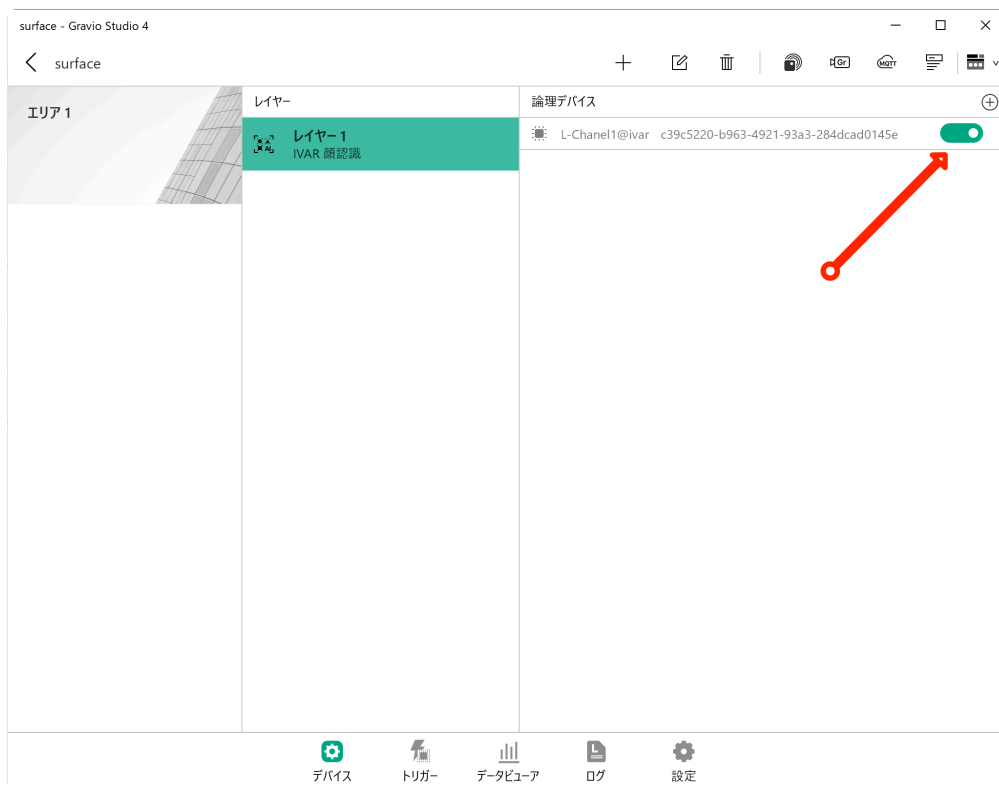
10. +ボタンをクリックしてバインド設定画面を表示します。



1. 設定したIVARのチャンネルをレイヤーにバインドにします。



1. 最後にセンサー取得をONにします。



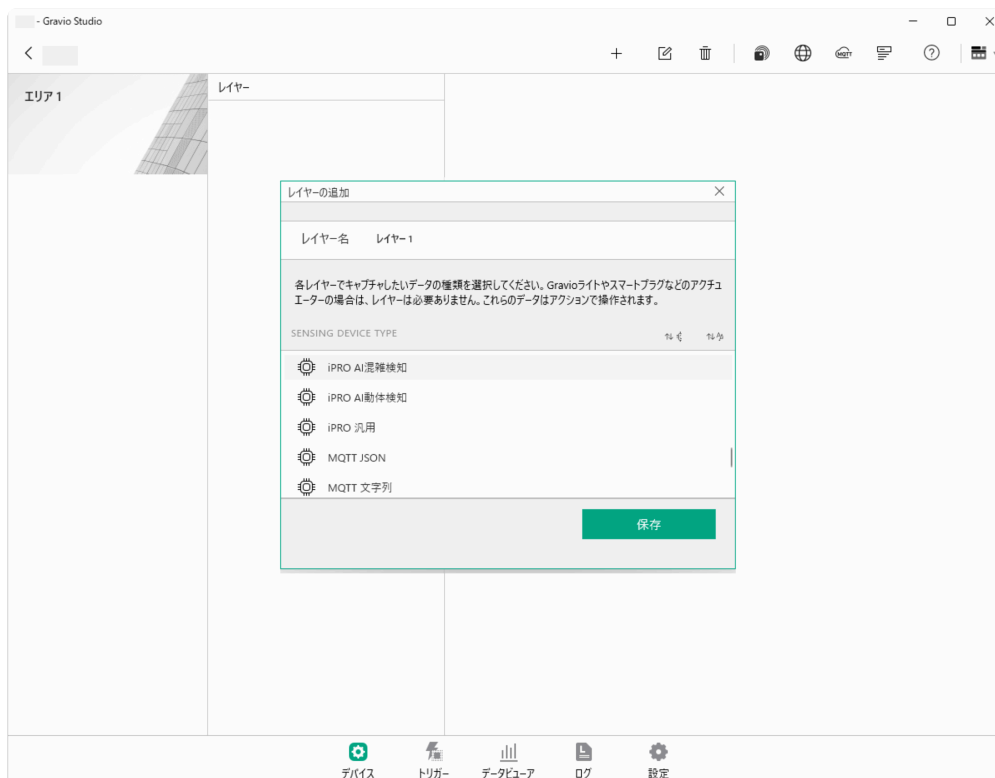
7.3.1.5. i-PRO

i-PROは [i-PRO株式会社](#) が販売している監視カメラです。Gravioでi-PROと連携しAIで認識したデータを受信します。

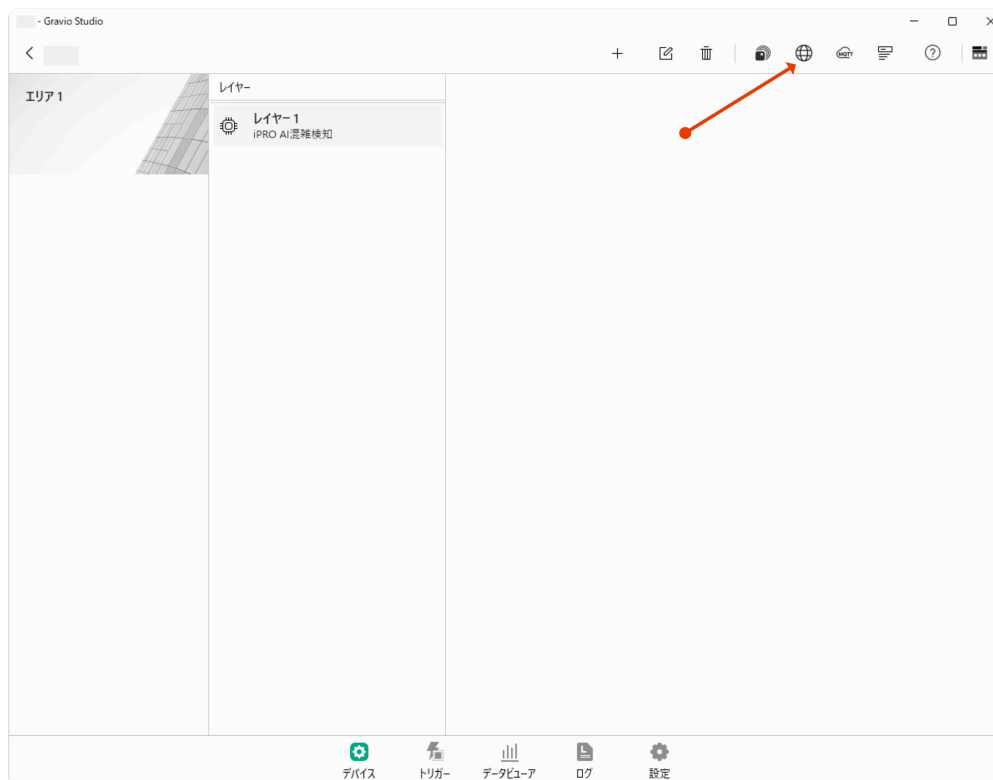
i-PROの設定

1. 「デバイス」で、i-PROの種類(DataKind)を選択して新しいエリアとレイヤーを追加します。

i-PRO は受信するデータの形式はAI混雑検知顔認識、AI動体検知、汎用から選択します。

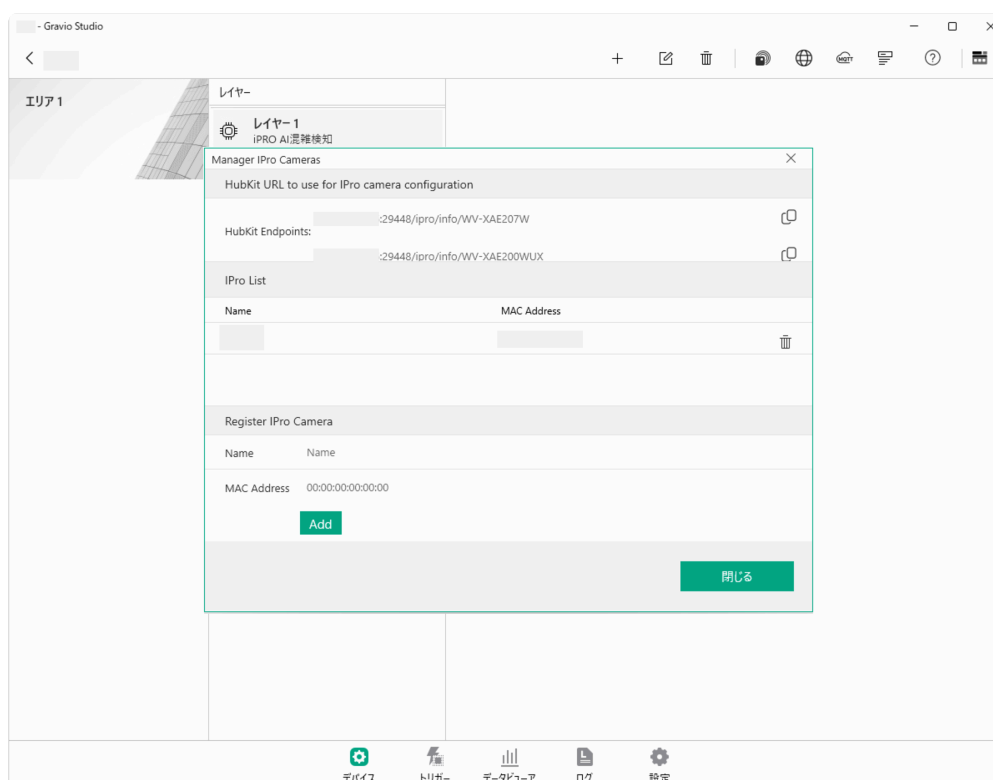


2. 以下のボタンを押して、i-PROの設定画面を表示します。

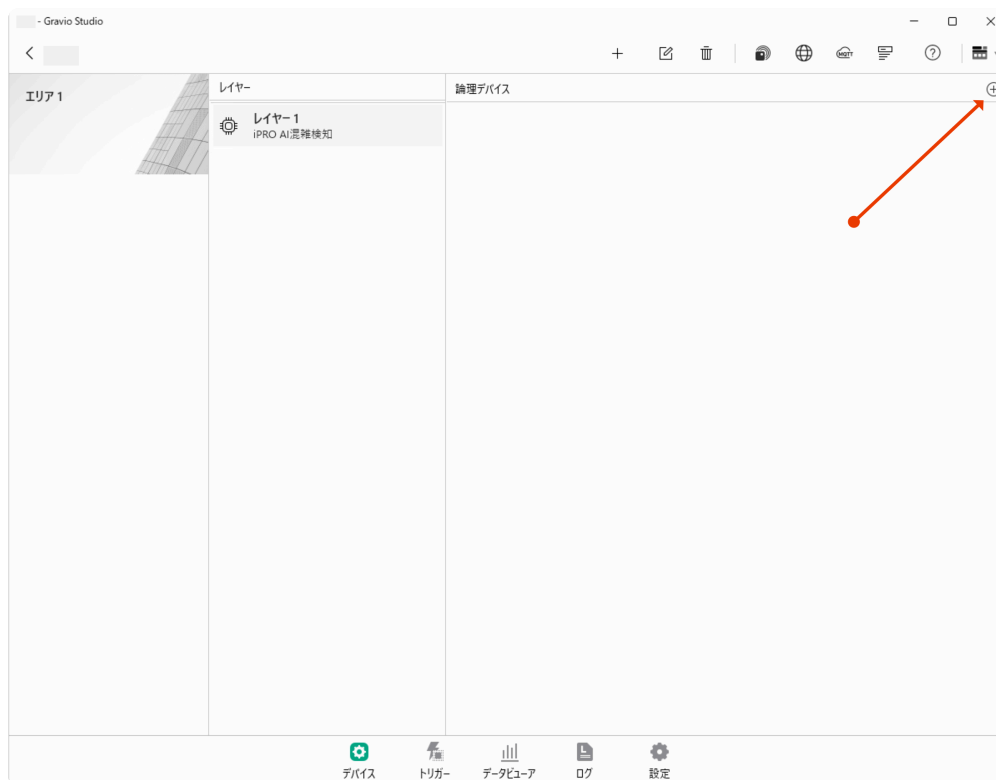


3. i-PROカメラの名前とMacアドレスを入力して追加ボタンを押します。i-PROに登録するHubKitのURLはクリップボードにコピーしてi-PROに登録してください。

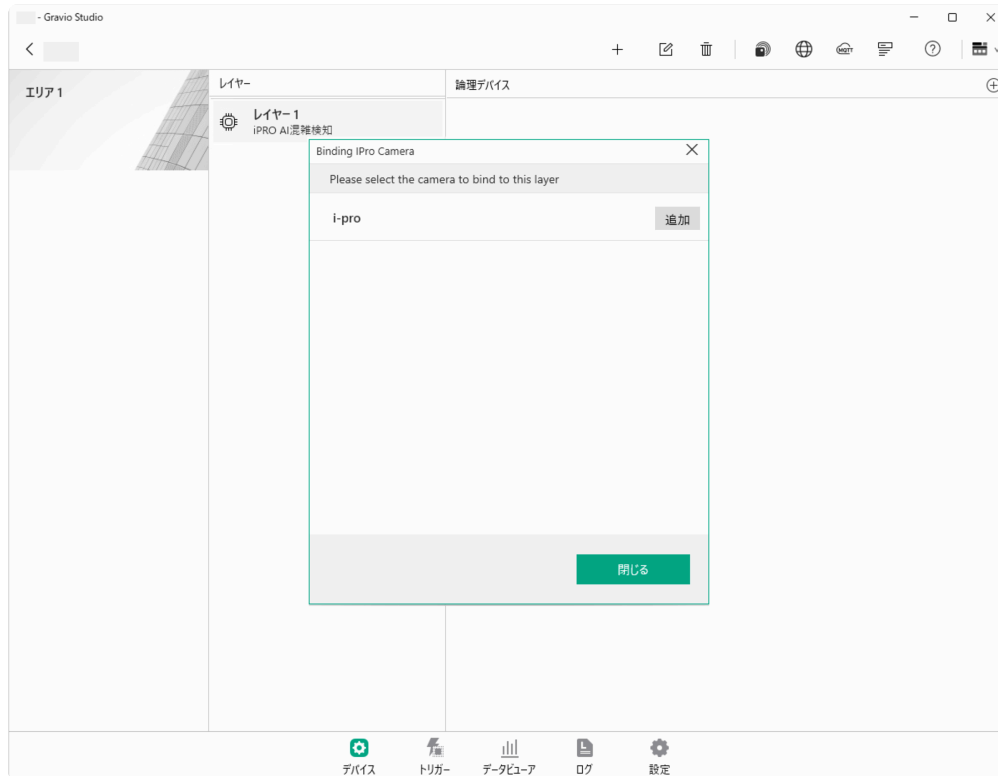
登録の方法は [i-PROカメラの設定方法](#) を参照してください。



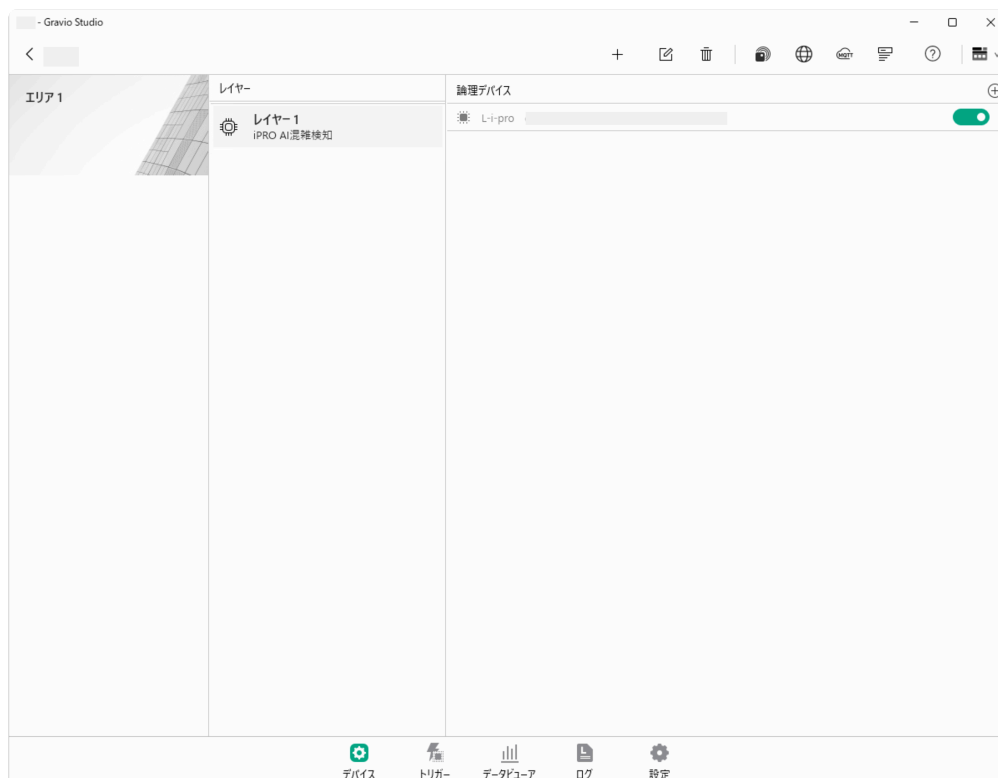
4. +ボタンをクリックしてバインド設定画面を表示します。



5. 設定したi-PROカメラをレイヤーにバインドにします。



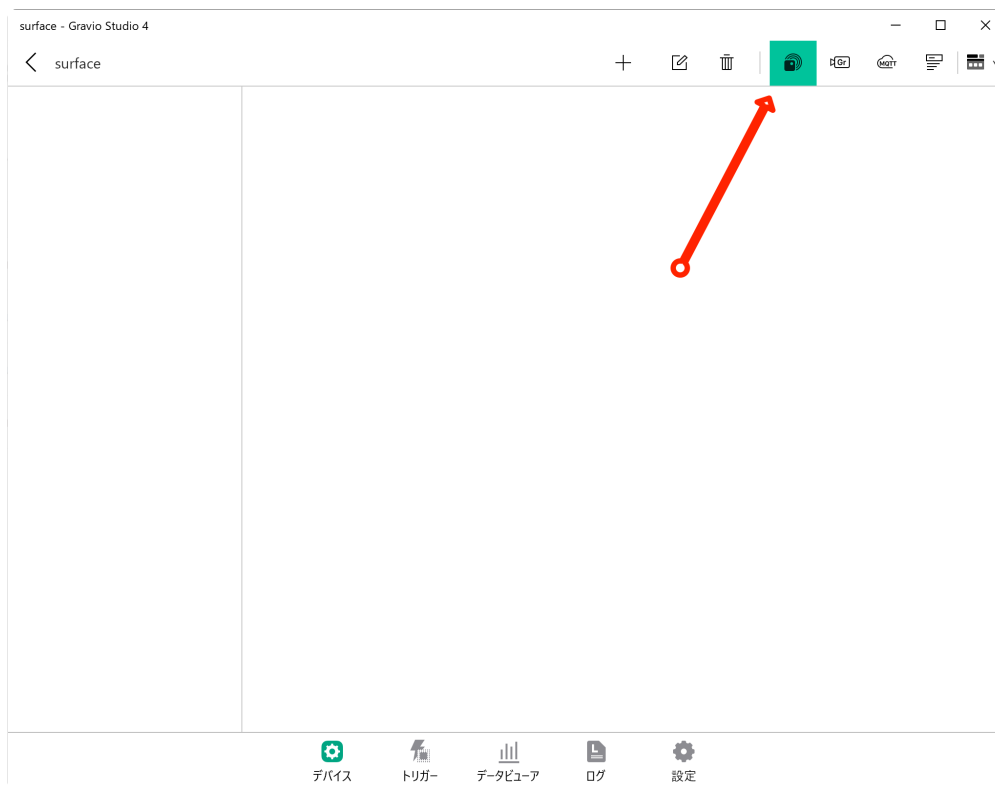
6. 最後にセンサー取得をONにします。



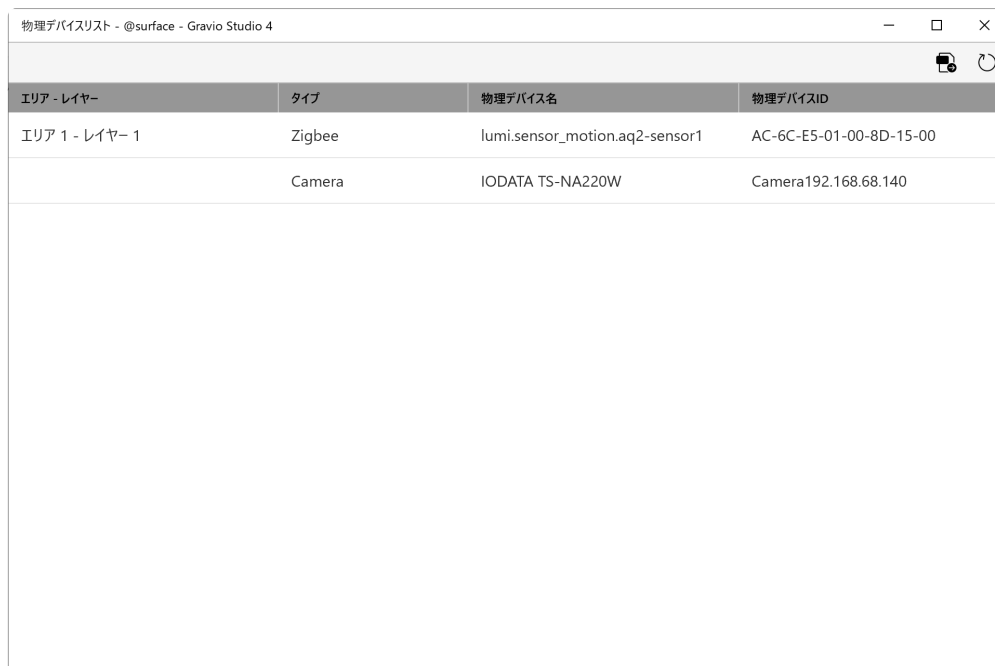
7.3.1.6. 物理デバイスリスト

Gravioに接続した物理デバイスの一覧を表示します。CSVでの一覧のダウンロードも可能です。

1. 以下のボタンを押して、物理デバイス一覧画面を表示します。

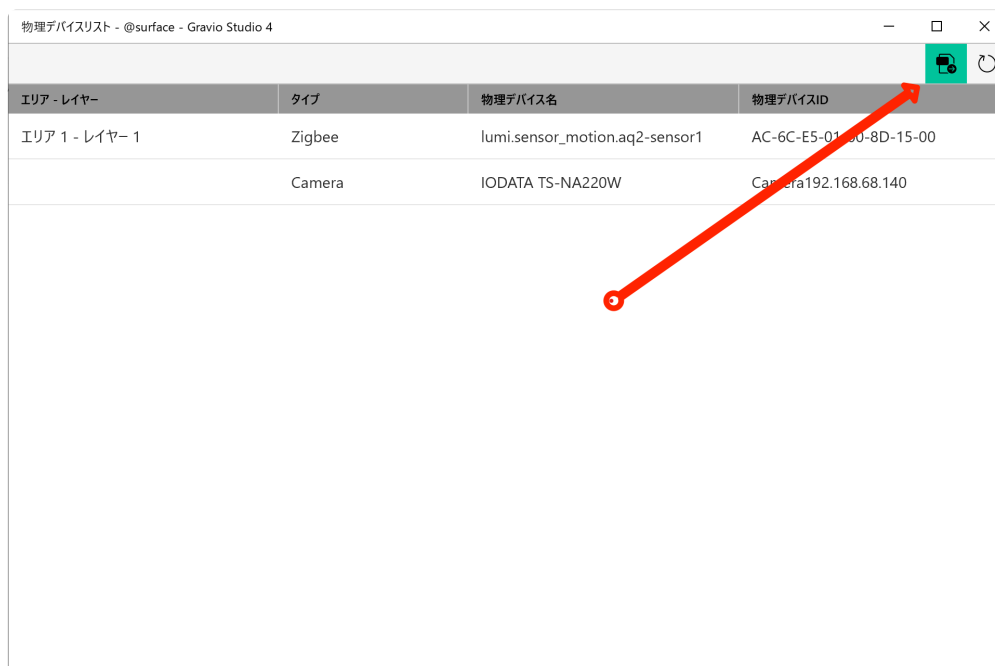


2. 物理デバイスがバインドされているエリア・レイヤー（バインドされていない場合は空）と物理デバイスのタイプ、物理デバイス名、物理デバイスIDがリストに表示されます。



エリア - レイヤー	タイプ	物理デバイス名	物理デバイスID
エリア 1 - レイヤー 1	Zigbee	lumi.sensor_motion.aq2-sensor1	AC-6C-E5-01-00-8D-15-00
	Camera	IODATA TS-NA220W	Camera192.168.68.140

3. 以下のボタンを押すと、物理デバイス一覧をCSVに出力することができます。



エリア - レイヤー	タイプ	物理デバイス名	物理デバイスID
エリア 1 - レイヤー 1	Zigbee	lumi.sensor_motion.aq2-sensor1	AC-6C-E5-01-00-8D-15-00
	Camera	IODATA TS-NA220W	Camera192.168.68.140

7.3.2. トリガータブ

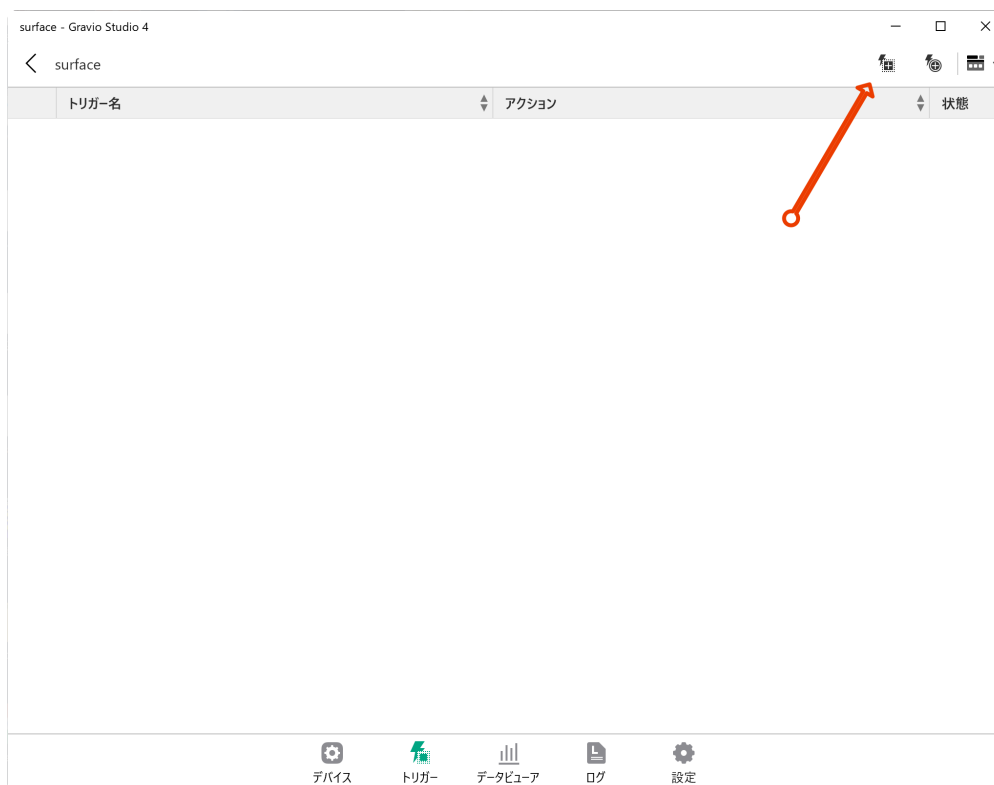
[Trigger]タブでは、センサーの値による条件で起動できるデバイストリガーとタイマーによる定期実行を行うタイマートリガーを設定します。

トリガーを作成する前に、[デバイスタブ](#) に従ってデバイスを設定し、「アクションの手順」に従って少なくとも1つのアクションを設定してください。

7.3.2.1. デバイストリガー

・デバイスとアクションを設定し、デバイストリガーを作成します。

デバイストリガーを作成するには、右上の「+」記号をクリックします。



以下のダイアログが表示されます：

トリガー名で名前を入力して、登録したデバイスのエリアを選択してください。

デバイストリガーの追加

トリガー名

新規イベントトリガー 1

エリア

時間

条件

アクション

カレンダー

なし

☐ 有効時間を指定する

開始

9 | 00

終了

17 | 00

間隔(ミリ秒)

5000

☐ 閾値によるトリガーを利用

☐ 実行中ならスキップする

キャンセル

保存

トリガーの有効時間帯の設定

- ・時間タブでは使用するカレンダーを選択します。カレンダーはCoordinatorを使用している場合に選択可能となります。
- ・有効時間を指定するにチェックを入れて開始と終了の時間を指定すると、その時間内だけこのトリガー条件は有効となります。
- ・間隔はトリガー条件を監視する間隔を指定します。
- ・閾値によるトリガーを利用するが選択されている場合は受信データが設定された閾値を超えた場合にのみ再実行されます。
- ・実行中ならスキップするを選択すると、トリガーの実行中に新しくトリガー条件が一致した場合は新しいトリガーは実行されません。

デバイストリガーの追加

×

トリガー名

新規イベントトリガー 1

エリア

時間

条件

アクション

次の条件を満たす場合に実行されます

条件数: 0/5

And

追加

キーレイヤー :

デバイスID :

キャンセル

保存

トリガーの実行条件の設定

条件タブではキーレイヤー/レイヤーとデバイスIDを選択します。センサーのタイプに応じて、ダイアログボックスの下部にさまざまなオプションが表示されます。

キーレイヤーに加えて複数(最大4つまで)のレイヤーの条件を入力する場合には、AndかOrを指定します。Andを指定して場合は「キーレイヤーの条件 And (レイヤーの条件 And レイヤーの条件 And ...)」となり、Orを指定して場合は「キーレイヤーの条件 And (レイヤーの条件 Or レイヤーの条件 Or ...)」となります。キーレイヤーの条件は必ずAndとなります。

- 数値の比較の条件式は 任意の値、=、!=、>、>=、<、<=、範囲、式
- 文字列の比較の条件式は 任意の値、=、!=、次を含む、次を含まない、次で始まる、次で始まらない、次で終わる、次で終わらない、正規表現、式
- 比較する値を入力します。

式にはトリガー内で使用できる関数を使用することが出来ます。関数は [こちら](#) を参照してください。

レイヤー :	物理デバイスID :	
		⊖
有効期間(ミリ秒)		0

キーレイヤーにさらに追加をしたレイヤーでは有効期限(ミリ秒)を指定することができます。キーレイヤーの条件を満たした場合で、レイヤーの条件を判断する受信データの有効期限となります。

デバイストリガーの追加 ×

トリガー名	新規イベントトリガー 1	
エリア		

時間	条件	アクション
アクション名	設定なし	
トリガープロパティー名	値	トリガープロパティー: 0/5
		追加

アクションの設定

アクションタブでは実行するアクションを選択します。

アクションの実行には引数としてトリガープロパティを複数(最大5つまで)指定することも出来ます。

トリガープロパティはあらかじめ定義されている下記の3つと任意のプロパティ名を設定することも出来ます。

tp.KeyAreaName : 選択されたエリア名が設定されます。

tp.KeyLayerName : 条件で指定した最初のキーレイヤー名が設定されます。

tp.TriggerName : 実行されるトリガー名が設定されます。

上記3つ以外に任意のプロパティ名と値を指定することで、その値もアクションないで参照することが出来ます。

設定例につきましては [こちら](#) を参照してください。

トリガーの有効化

スイッチを緑色に切り替えることによって、新しいトリガーを有効にすることを忘れないでください。

7.3.2.1.1. 設定例

トリガーの有効時間帯の設定

間隔の例

デバイストリガーは設定した条件に合致したデータが届いたときに動作しますが、間隔を指定することで前回の実行時刻よりこの間隔だけ経たないと次の実行が行われないように設定することができます。

例：デバイスからデータが2秒に一度届くが、トリガーは10秒ごとに実行したい場合、間隔は9000ms（10秒の10%ほど低い値）を指定します。

このように設定することで2秒毎の受信データは以下のようなタイミングでトリガーが実行されます。

受信データ1(トリガー実行)、受信データ2、受信データ3、受信データ4、受信データ5、受信データ6(トリガー実行)、受信データ7

閾値の例

このパラメータは、常にデータを送信するセンサー（温度センサー、ドア・窓開閉センサー、ワイヤレススイッチなど）に対応するために導入されました。

例1：温度センサーで閾値によるトリガーに設定がオンの場合でトリガーが $>20^{\circ}$ に設定されている場合、センサーが 20° 以上のデータを送信してトリガーが実行された後は一度 20° を下回り再度 20° 以上にならないと実行はされません。閾値によるトリガーに設定がオフの場合は、センサーが 20° 以上のデータを送信するとトリガーは常に実行されます。

21° , 22° , 18° , 23° とセンサーがデータを送信した場合、閾値によるトリガーが設定されている場合は 21° , 23° で実行されますが、設定されていない場合は 21° , 22° , 23° で実行されます。

例2：ドア・窓開閉センサーで閾値によるトリガーに設定がオンの場合でトリガーが `=close` に設定されている場合、センサーが `close` のデータを送信してトリガーが実行された後は一度 `open` にならないと実行はされません。閾値によるトリガーに設定がオフの場合は、センサーが `close` データを送信するとトリガーは常に実行されます。

`close`, `close`. `open`, `close` とセンサーがデータを送信した場合、閾値によるトリガーが設定されている場合は `close`, `close` で実行されますが、設定されていない場合は `close`, `close`. `close` で実行されます。

例3：ワイヤレススイッチで閾値によるトリガーに設定がオンの場合でトリガーが `= click` に設定されている場合、センサーが `click` のデータを送信してトリガーが実行された後は一度 `click` 以外のデータにならないと実行はされません。閾値によるトリガーに設定がオフの場合は、センサーが `click` データを送信するとトリガーは常に実行されます。
`click`, `click`. `double click`, `click` とセンサーがデータを送信した場合、閾値によるトリガーが設定されている場合は `click`, `click` で実行されますが、設定されていない場合は `click`, `click`. `click` で実行されます。

トリガーの実行条件の設定

キーレイヤーの例

キーレイヤーとは、必ず指定するレイヤーの条件となります。トリガー条件を満たしアクションが実行されるのは、このキーレイヤーのデータを受信し、かつトリガー条件を満たしている場合となります。

例1: キーレイヤーにドア・窓開閉センサーの条件が close と設定されている場合

1. ドア・窓開閉センサーのデータ open を受信
2. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
3. ドア・窓開閉センサーのデータ open を受信
4. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます

例2: キーレイヤーにドア・窓開閉センサーの条件が close と設定されており、レイヤーに温度センサーの条件が 20°以上と設定していた場合

(閾値によるトリガーを利用するを選択している場合)

1. ドア・窓開閉センサーのデータ open を受信
2. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)
3. 温度センサーのデータ 20.0° を受信
4. ドア・窓開閉センサーのデータ open を受信
5. 温度センサーのデータ 19.8° を受信
6. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)
7. ドア・窓開閉センサーのデータ open を受信
8. 温度センサーのデータ 20.1° を受信
9. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
10. 温度センサーのデータ 20.2° を受信

例3: キーレイヤーにドア・窓開閉センサーの条件が close と設定されており、レイヤーに温度センサーの条件が 20°以上と設定していた場合

(閾値によるトリガーを利用するを選択していない場合)

1. ドア・窓開閉センサーのデータ open を受信
2. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)
3. 温度センサーのデータ 20.0° を受信
4. ドア・窓開閉センサーのデータ open を受信
5. 温度センサーのデータ 20.1° を受信
6. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
7. ドア・窓開閉センサーのデータ open を受信
8. 温度センサーのデータ 20.2° を受信
9. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
10. 温度センサーのデータ 20.1° を受信

例4: キーレイヤーにドア・窓開閉センサーの条件が close と設定されており、レイヤーに温度センサーの条件が 20°以上と設定、レイヤーに湿度センサーの条件が 30%以上と設定していた場合

(Andを選択している場合)

1. ドア・窓開閉センサーのデータ open を受信
2. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)

3. 温度センサーのデータ 20.0° を受信
4. ドア・窓開閉センサーのデータ open を受信
5. 温度センサーのデータ 20.1° を受信
6. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)
7. ドア・窓開閉センサーのデータ open を受信
8. 湿度センサーのデータ 35% を受信
9. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
10. 温度センサーのデータ 20.1° を受信

例5: キーレイヤーにドア・窓開閉センサーの条件が close と設定されており、レイヤーに温度センサーの条件が 20°以上と設定、レイヤーに湿度センサーの条件が 30%以上と設定していた場合 (Orを選択している場合)

1. ドア・窓開閉センサーのデータ open を受信
2. ドア・窓開閉センサーのデータ close を受信 (トリガーは実行されません)
3. 温度センサーのデータ 20.0° を受信
4. ドア・窓開閉センサーのデータ open を受信
5. 温度センサーのデータ 20.1° を受信
6. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
7. ドア・窓開閉センサーのデータ open を受信
8. 湿度センサーのデータ 35% を受信
9. ドア・窓開閉センサーのデータ close を受信 << ここでトリガーが実行されます
10. 温度センサーのデータ 20.1° を受信

式の例

式を選択する場合は、その条件が true の場合にトリガーの実行条件が成立します。

例1: 温度センサーの温度が25度以上の場合

式 : `tv.Data >= 25`

例2: 温度センサーの温度が20度以上25度以下の場合

式 : `tv.Data >= 20 && tv.Data <= 25`

例3: IVARの顔認識でFR_UNKNOWNを受信した場合

イベントタイプ : 任意のイベント

式 : `tv.Data.common.type == "FR_UNKNOWN"`

例4: IVARの顔認識でマスクをしていないデータを受信した場合

イベントタイプ : 任意のイベント

式 : `tv.Data.iod.mask == "NONE"`

キーレイヤーと追加をしたレイヤーのトリガー条件の例

例：CO2が1000ppmを超えた場合で、かつ人感センサーで人がいるときだけトリガーを実行する場合

キーレイヤーではCO2のレイヤーの値が1000ppmを超えた時と条件を指定します。またレイヤーの人感センサーは反応したときだけの条件とします。

その際に、人感センサーは人がいないときにはデータを送信しないため、1時間前に人がいたけれどももういなくなったような場合、直近の値（1時間前のデータ）で条件比較すると、まだ人がいるということになってしまいます。

こういう場合に、レイヤーで「有効期限」に60000ミリ秒（1分）を指定しておくことで、直近1分以内に人感センサーが反応したときだけトリガーを実行することが可能になり、人がいないときにはCO2が1000ppmを超えた場合でもトリガーは実行しない、ということを実現できるようになります。

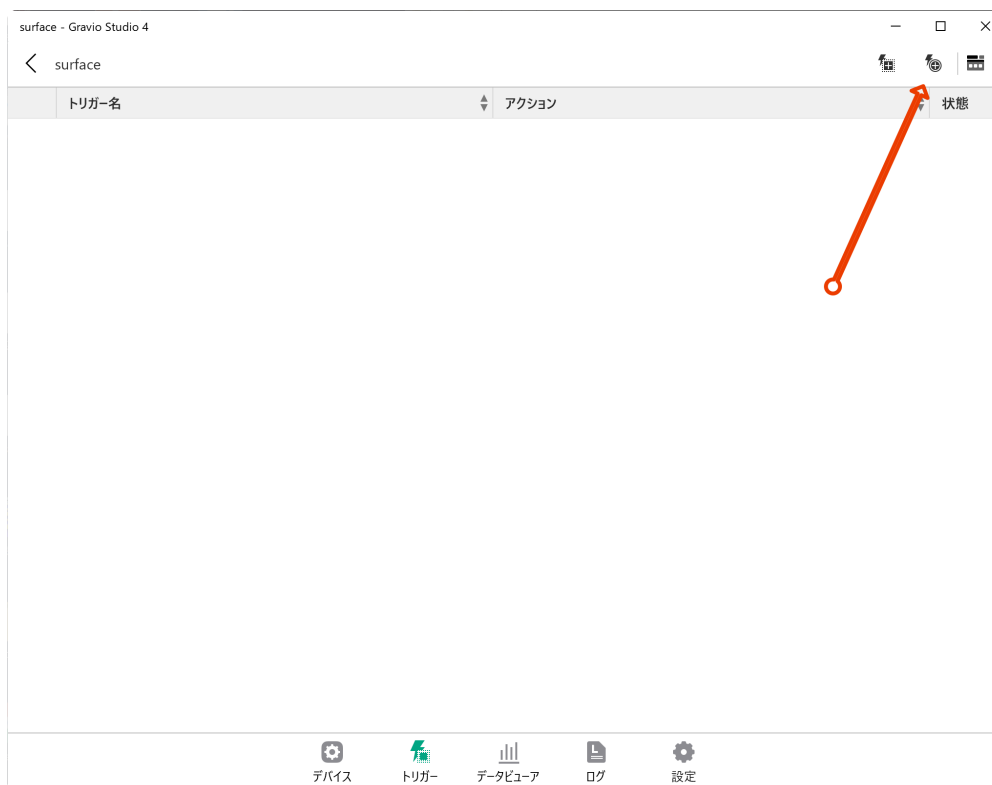
また、「有効期限」に0（ミリ秒）を指定している場合は、そのレイヤーに有効期限はなく直近の値で条件判断を行います。

7.3.2.2. タイマートリガー

・アクションを設定し、タイマートリガーを作成します。

タイマートリガーの作成

タイマートリガーを作成するには、右上の「+」記号をクリックします。



以下のダイアログが表示されます：

タイマー名で名前を入力してください。

- ・カレンダーを選択します。カレンダーはCoordinatorを使用している場合に選択可能となります。
- ・有効時間を指定するにチェックを入れて開始と終了の時間を指定すると、その時間内だけこのトリガー条件は有効となります。
- ・実行中ならスキップするを選択すると、トリガーの実行中に新しくトリガー条件が一致した場合は新しいトリガーは実行されません。

タイマーの作成

×

タイマー名

新規タイマートリガー 1

×

カレンダー

なし

▽

☐ 有効時間を指定する

開始

9

00

終了

17

00

☐ 実行中ならスキップする

スケジュール

アクション

繰り返し

設定なし

▽

キャンセル

保存

スケジュールの設定

繰り返し条件を選択します。

- ・インターバルは一定の分ごとの実行となります。
- ・1時間ごとは指定の開始時刻(分、秒)ごとの実行となります。
- ・毎日は指定の開始時刻(時、分、秒)ごとの実行となります。
- ・週次は指定の開始時刻(時、分、秒)で指定の曜日ごとの実行となります。
- ・月次は指定の開始時刻(時、分、秒)で指定の日ごとの実行となります。
- ・カスタムはcronの指定方法により詳細な実行設定が可能となります。

カスタム式は、5つのスペースで区切られたフィールドを使用して回数のセットを表します。

分	必須	許容される値	利用できる特殊文字
秒	必須	0-59	• / , -
分	必須	0-59	• / , -
時	必須	0-23	• / , -
日	必須	1-31	• / , - ?
月	必須	1-12 or JAN-DEC	• / , -
曜日	必須	0-6 or SUN-SAT	• / , - ?

例：年に1回1月1日 0:00:00 “0 0 0 1 1 *”

タイマーの作成

×

タイマー名

新規タイマートリガー 1

カレンダー

なし

▽

☐ 有効時間を指定する

開始

9 | 00

終了

17 | 00

☐ 実行中ならスキップする

スケジュール

アクション

アクション名

設定なし

▽

トリガープロパティ名

値

トリガープロパティ: 0/5

追加

キャンセル

保存

アクションの設定

アクションタブでは実行するアクションを選択します。

アクションの実行には引数としてトリガープロパティーを複数(最大5つまで)指定することも出来ます。

トリガープロパティーはあらかじめ定義されている下記の3つと任意のプロパティ名を設定することも出来ます。

tp.KeyAreaName : 選択されたエリア名が設定されます。

tp.KeyLayerName : 条件で指定した最初のキーレイヤー名が設定されます。

tp.TriggerName : 実行されるトリガー名が設定されます。

上記3つ以外に任意のプロパティー名と値を指定することで、その値もアクションないで参照することが出来ます。

トリガーの有効化

スイッチを緑色に切り替えることによって、新しいトリガーを有効にすることを忘れないでください。

7.3.2.3. トリガー変数とトリガープロパティ

デバイス/タイマートリガーではアクションを実行するときに引数としてトリガープロパティとトリガー変数を渡すことができます。

トリガー変数

トリガー変数は tv. の変数で、センサーデータの受信によりイベントで駆動されたトリガーで tv. としてアクションに渡されます。

tv.Data：受信したデータが設定されます。

トリガープロパティ

トリガープロパティ tp. はトリガー設定のダイアログで定義可能で、同一アクションを呼び出してもtp. の値による条件分岐を使用することで動作を変えることができます。

トリガープロパティはあらかじめ定義されている下記の3つと任意のプロパティ名を設定することも出来ます。

tp.KeyAreaName：選択されたエリア名が設定されます。

tp.KeyLayerName：条件で指定した最初のキーレイヤー名が設定されます。

tp.TriggerName：実行されるトリガー名が設定されます。

7.3.3. データビューア

[データビューア]タブでは、受信したセンサデータをリアルタイムで調べることができます。これは、システムをセットアップしてその機能をテストする場合に便利です。右側の「ライブ」ボタンをクリックすると、ライブの受信データが表示されます。ビューは5秒ごとに更新されます。

surface - Gravio Studio 4

< sur エリア

Any ▾

レイヤー

Any ▾

デバイス

Any ▾

開始日時

2020/10/12 📅

0 | 00

終了日時

2020/10/12 📅

23 | 59

LIVE

📄 ▾

エリア	レイヤー	種別	デバイス	送信元ID	データID	日時	値
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	b7d03c02-d89f-4be1	2020/10/12 17:43:29	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	ee7db714-16c7-466	2020/10/12 17:42:51	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	8326faed-da73-42e1	2020/10/12 17:42:39	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	b58beac8-ed0c-474	2020/10/12 17:42:32	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	afe0483c-76fd-439b	2020/10/12 16:36:08	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	50353523-2ad9-484	2020/10/12 16:36:02	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	efc89c4b-d2aa-4b5c	2020/10/12 16:35:45	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	d9f19f30-b2f5-49cb	2020/10/12 16:35:37	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	d16168b2-63d4-462	2020/10/12 16:35:31	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	3fa964cd-2455-4a87	2020/10/12 16:35:06	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	9d481158-1aa8-407	2020/10/12 16:34:57	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	eb9b8f09-c76d-43a1	2020/10/12 16:34:49	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	6239d561-a596-47d	2020/10/12 16:34:39	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	f6d01c7a-c31d-4ad5	2020/10/12 16:34:31	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	9f41f214-cc1d-48d0	2020/10/12 16:33:58	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	5888040f-bf8f-49f6	2020/10/12 16:33:52	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	2cf244cd-b6b5-4f12	2020/10/12 16:33:10	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	15e6014d-ee7f-4281	2020/10/12 16:32:46	1
エリア 1	レイヤー 1	Aqara-Occupancy	Sensor1	AC-6C-E5-01-00-8D	589d5234-74f6-41f4	2020/10/12 16:32:32	1

⚙️ デバイス

🔌 トリガー

📊 データビューア

📄 ログ

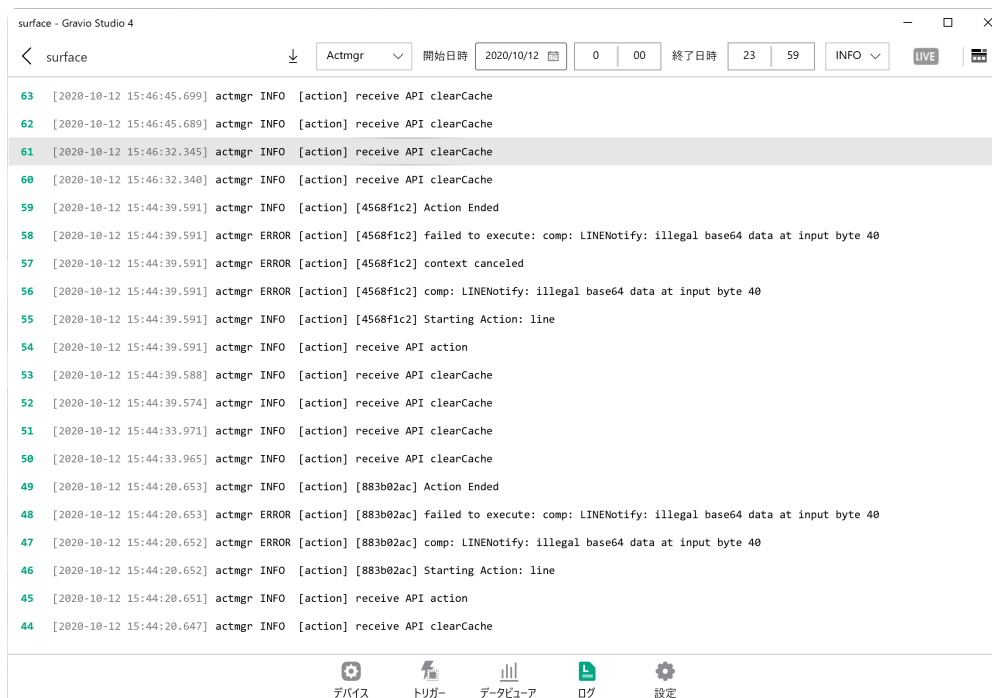
⚙️ 設定

このビューは、[Sensor ID]列でセンサのIDを識別するのに役立ちます。

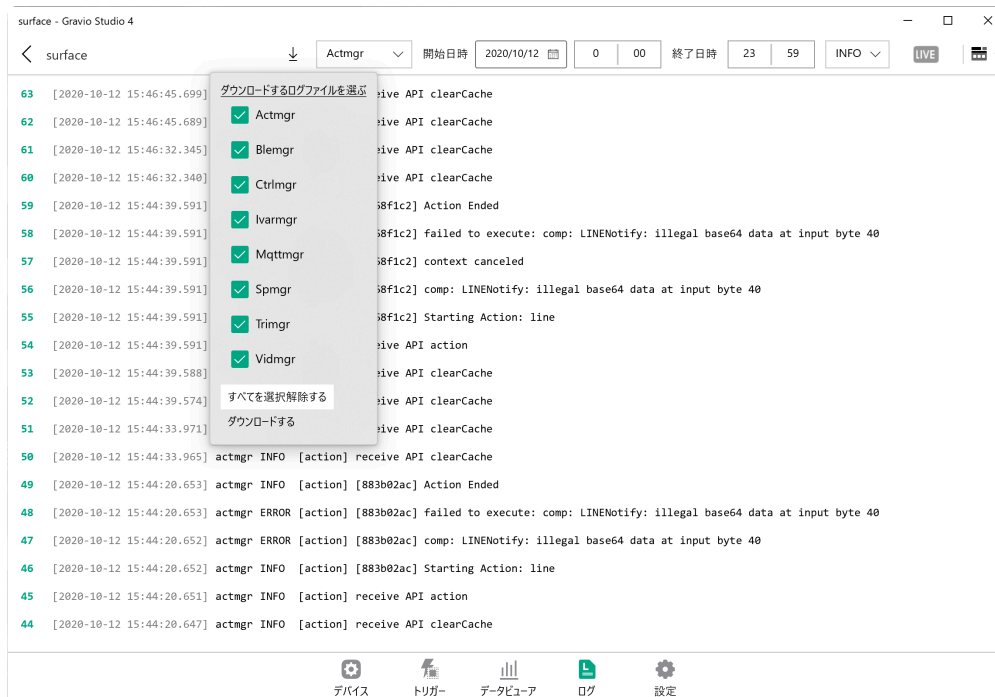
タイムフレーム、エリア、レイヤー、デバイスなど、さまざまな側面のデータをフィルタリングできます。

7.3.4. ログビューア

[ログビューア]タブでは、Gravioが生成するログファイルにアクセスできます。主にデバッグに使用されます。ログには以下が含まれます:



- **Actmgr** – アクションとステップが生成するログをすべて含みます
- **Blemgr** – Bluetoothデバイスに関するもの
- **Ctrlmgr** – 実行されたコマンド
- **Ivarmgr** – IVARに関するもの
- **Mqtmgr** – MQTTに関するもの
- **Spmgr** – Zigbeeなどシリアルデバイスに関するもの
- **Trigmgr** – タイマーやスケジューリングに関するもの
- **Vidmgr** – ビデオデバイスに関するもの



ダウンロードボタンを押すとダウンロードするログファイルを選択してzipファイルに圧縮してPCにダウンロードすることが出来ます。

ログファイルを選択してダウンロードするをクリックしてください。

7.3.5. 設定タブ

[設定]タブでは、HubKit情報と通知、バックアップ・リストア、基本プロパティ設定、画像推論モデルのダウンロードとディスク管理の機能があります。

HubKit情報

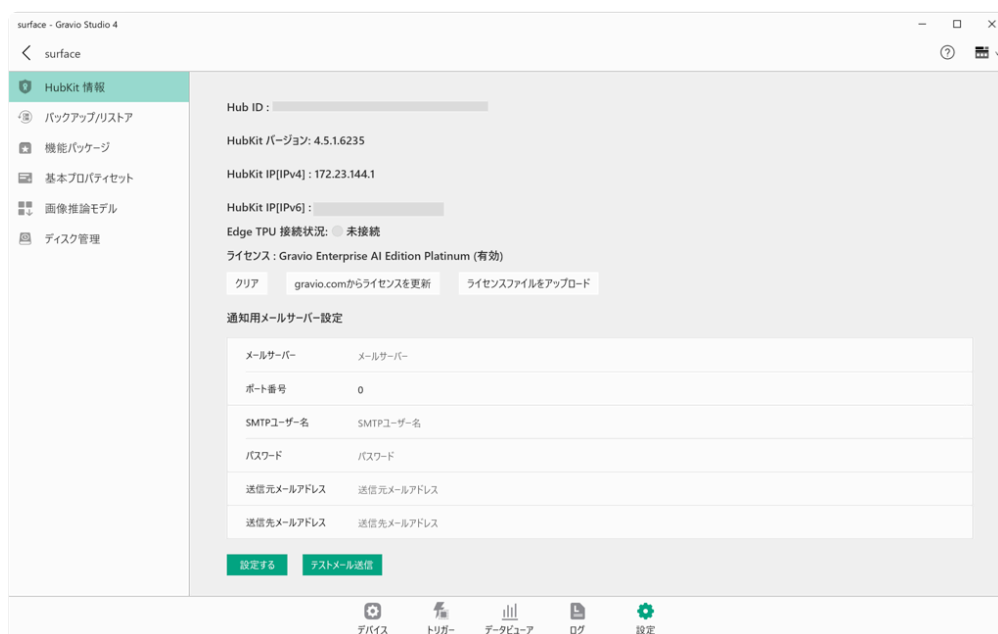
設定を行っているHubのHub ID、HubKitのバージョン番号ならびにIPアドレスの表示、通知用メールサーバの設定を行うことができます。

HubにEdge TPUが接続されており使用可能となっている場合は Edge TPU接続状況は“使用可能”と表示されます。画像推論モデルのTensorFlowLiteを使用する場合、モデルパッケージで指定したLiteEdge.tfliteファイルとEdge TPUが使用されます。Edge TPUはCoral社の [USB Accelerator](#) での動作確認を行っております。

※ Gravio 4.5からライセンス管理機能が追加されました。Gravio 4.5以上をご使用の場合には [こちら](#) の手順でライセンスの登録を行ってください。

通知機能は、Gravioがソフトウェアやハードウェアのエラー、再起動、または動作に以上が発生したした場合、メール通知を送信できます。通知用メールサーバ設定にGravioが通知に使用するサーバ情報を登録します。

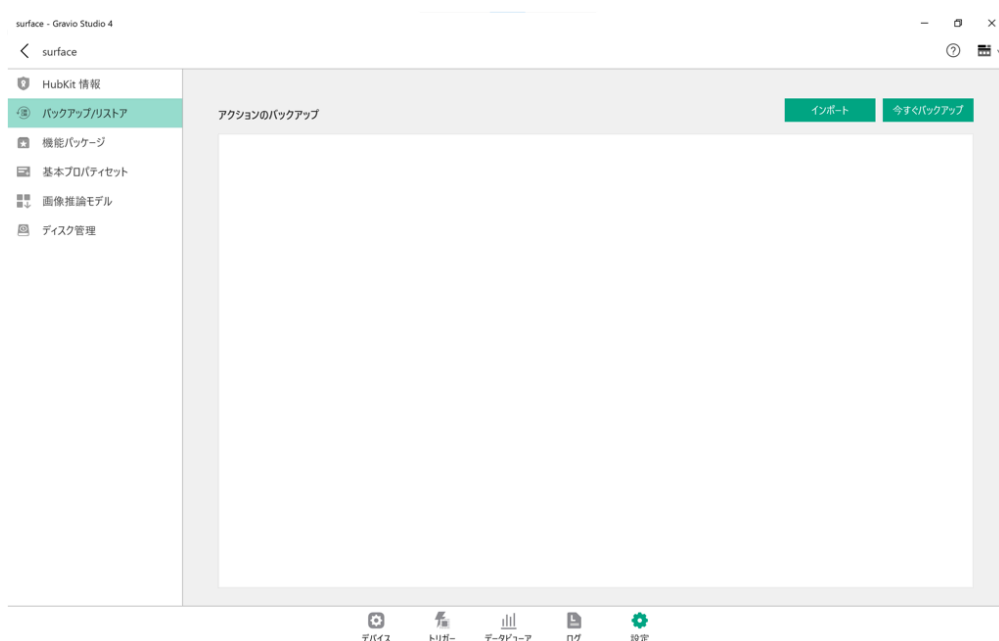
[Gravio 4.5以上] – Zigbeeデバイスのセンサーからの定期的な通信(約60分)が受信されず、その状態で一定期間データが受信できなかった場合(90分)続くと、通知設定で設定されたメールアドレスにデバイスからの通信がなくなったことを通知します。CO2は1分間信号が来ないときに発信されます。



- gmailのメールアドレスを使用してメール送信する場合の設定例は [こちら](#) を参照してください

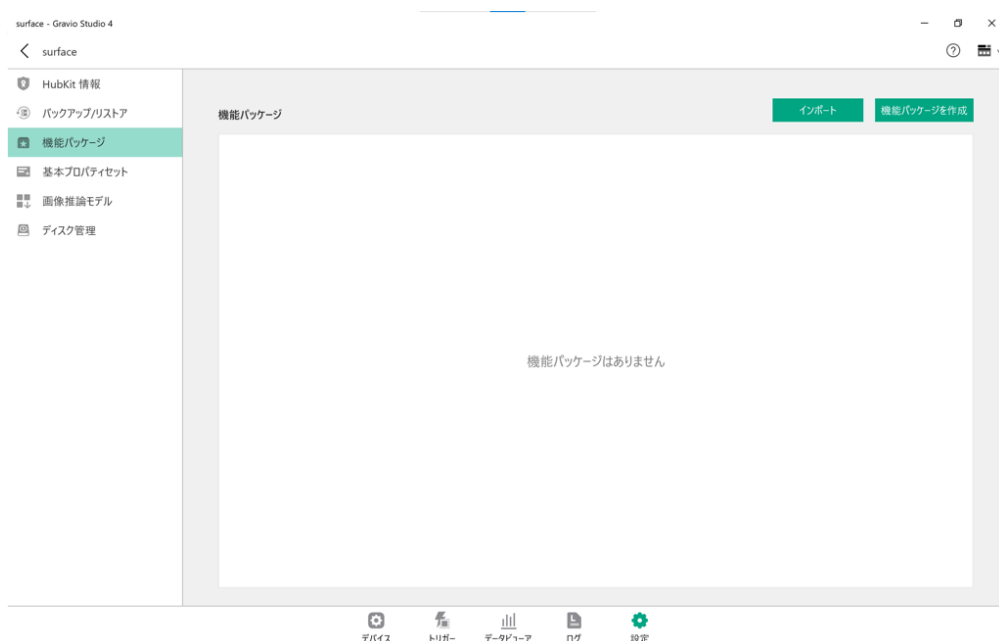
バックアップ・リストア

アクション設定をGravio Cloudアカウントにバックアップすることができます。これらの設定をGravioアカウントのホームフォルダにリストアすることができます。

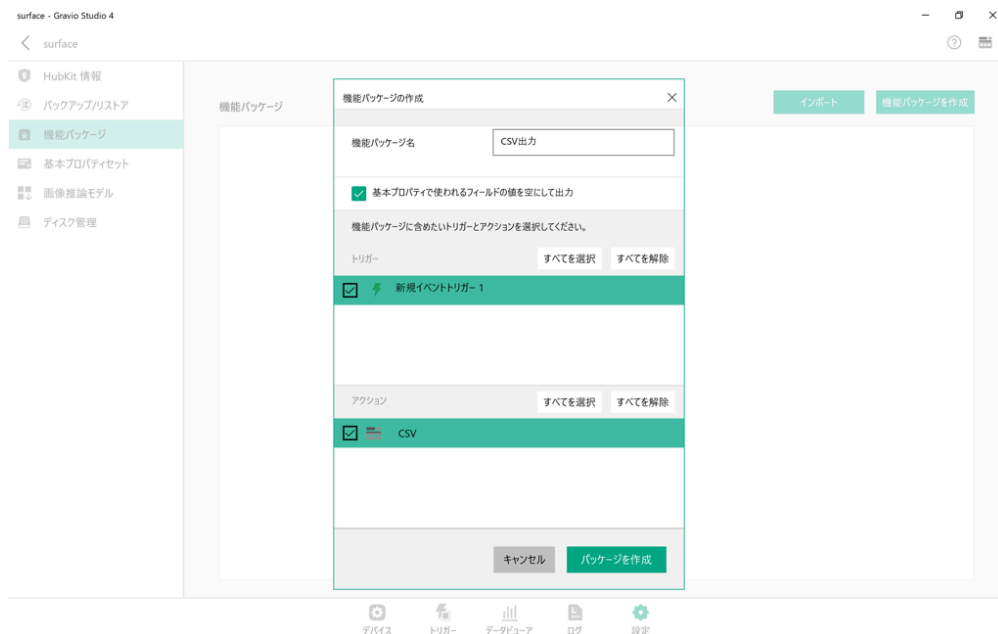


機能パッケージ

エリア、レイヤー、トリガー、アクションと基本プロパティを1つの機能パッケージとして配布することができます。機能パッケージを作成するには予めエリア、レイヤー、トリガーとアクションを作成しておきます。



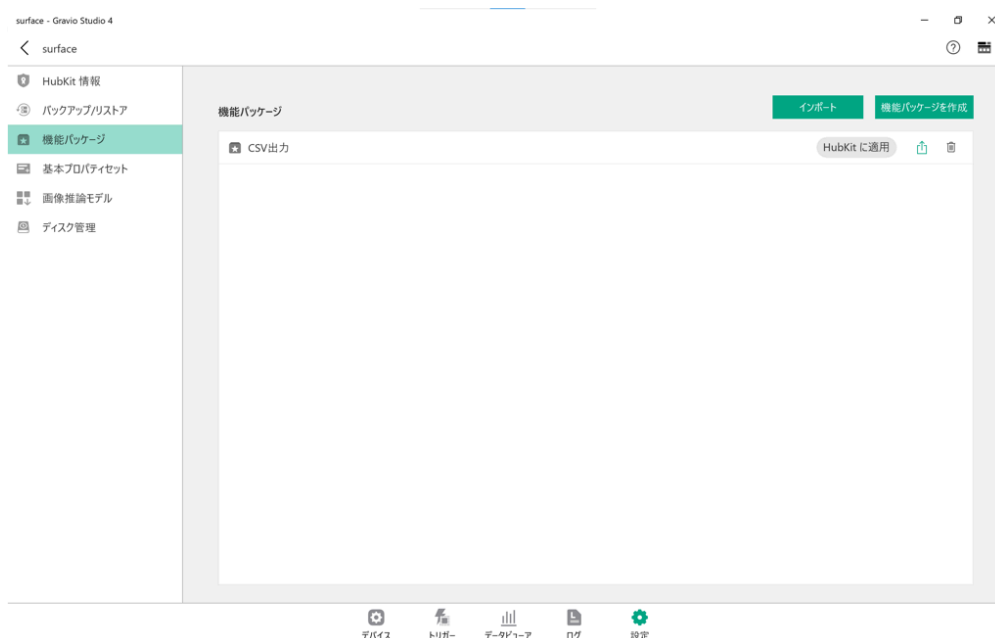
機能パッケージ一覧には現在登録されている機能パッケージがリスト表示されます。
「インポート」ボタンを押すと予め作成されている機能パッケージをインポートします。
「機能パッケージ作成」ボタンを押すと新規に機能パッケージを作成することができます。



機能パッケージを新規作成するには機能パッケージ名を入力して、パッケージに含めるトリガーとアクションを選択します。

「パッケージ作成」ボタンを押すと機能パッケージが作成されます。

「基本プロパティで使われているフィールドの値をからにして出力」をオンにすると、環境に依存するLINEのトークンやAWSのアクセスキーIDやシークレットアクセスキーなどのセキュリティ情報を削除してパッケージを作成するため安全に配布することができます。



機能パッケージが作成されるとには機能パッケージ一覧に表示されます。

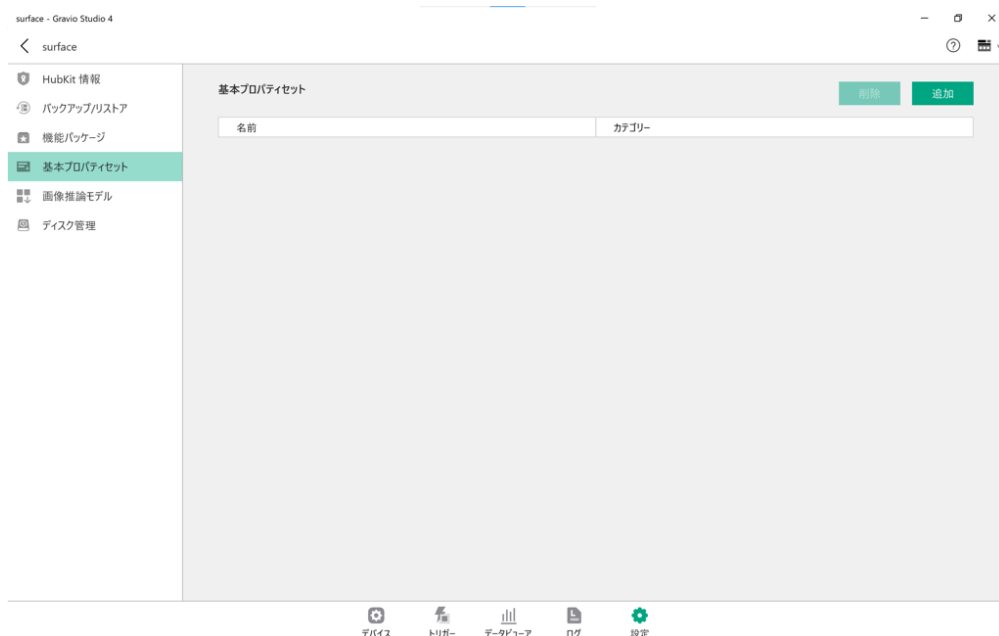
「HubKitに適用」ボタンを押すとエリア、レイヤー、トリガー、アクションと基本プロパティが自動的に登録されます。基本プロパティはセキュリティ情報はからで登録されますので、環境に合わせて値を設定します。すでに登録されているアクション名と同じアクション名が機能パッケージに含まれている場合には登録が中断されます。

「ダウンロード」ボタンを押すと機能パッケージをローカルPCにダウンロードされます。

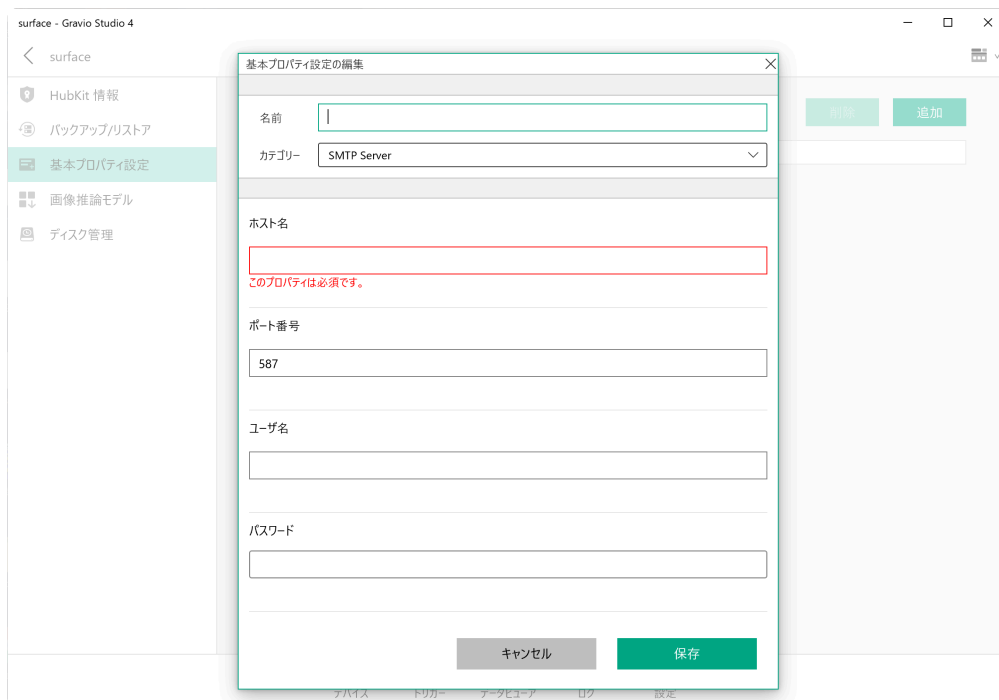
「削除」ボタンを押すとリストに登録されている機能パッケージを削除します。

基本プロパティ設定

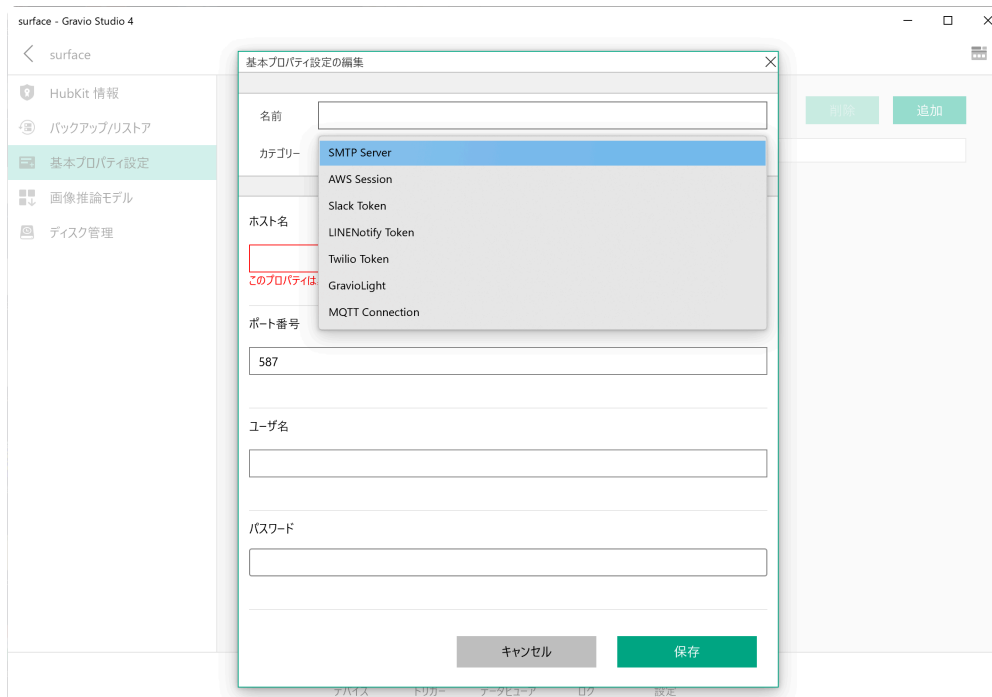
アクションを実行する時の各コンポーネントで設定する下記の環境情報を基本プロパティとして一元管理して登録することができます。



GravioライトとGravio LEDマトリックスの物理デバイスID、SMTPサーバのサーバ情報、AWSのAccess Key・Secret Key・リージョン、GCP接続の秘密鍵JSON、SlackのToken、LINE NotifyのToken、LINE WORKS Bot接続のコネクション情報、TwilioのAccountSidとToken、Chromecastのデバイス、Aqara Smartプラグの物理デバイスID、MQTTのブローカー情報、HTTP Proxyのプロキシ情報



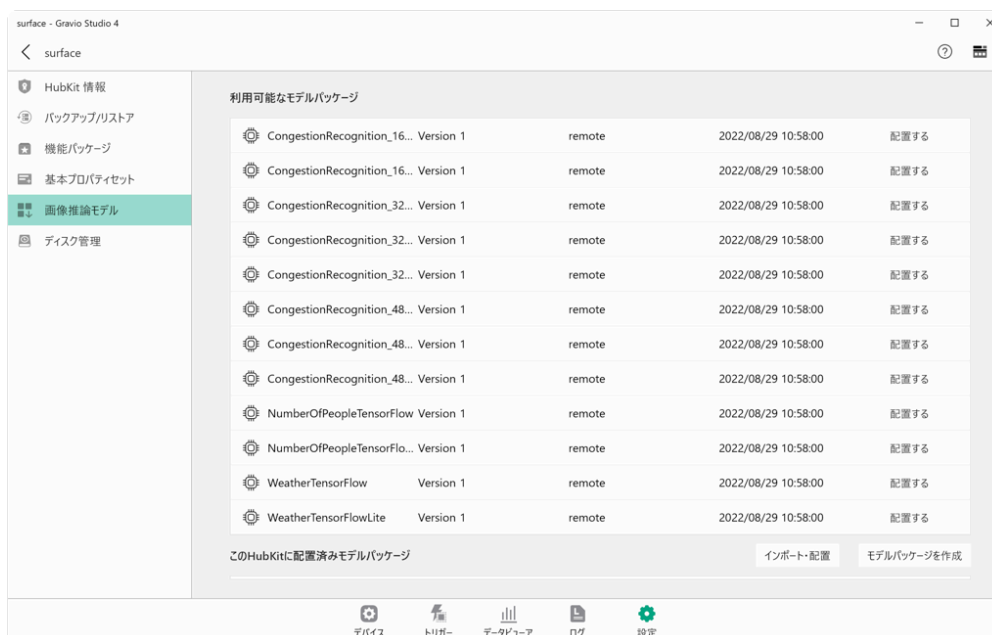
基本プロパティの名前を入力して、カテゴリを選択します。



それぞれのカテゴリで指定できる項目が表示されますので値を入力します。

画像推論モデル

リストより推論ファイルをダウンロードして利用することが可能です。(Gravio Freeのサービス購入者は1つのみ)



モデルパッケージのインポート・配置

ローカルに保存されているモデルパッケージを登録するにはインポート・配置ボタンを押してファイルを選択してからアップロードしてください。

モデルパッケージを作成

新たにTensorFlowかTensorFlowLiteのモデルパッケージを作成します。

推論モデルパッケージの作成

認識タスク

物体検出

モデルタイプ

☒ TensorFlow ☐ TensorFlow Lite

モデルパッケージ名

数文字のみ

TensorFlowメタファイル

選択

TensorFlow pbファイル

選択

メソッド

Count

出力形式

JSON

☐ 検出値を含める

信頼度の域値

0.5

HubKitへアップロード

キャンセル

認識タスクは“物体検出”か“画像分種”から選択します。“画像分種”の場合に作成できるのはTensorFlow Liteのみとなります。

TensorFlowで準備するファイル

TensorFlow モデルパッケージを作成するには以下のファイルを準備してください。

ファイル
TensorFlow メタファイル
TensorFlow pbファイル

TensorFlowで登録する項目

項目名	説明
モデルパッケージ名	英文字で設定
TensorFlow メタファイル	準備したファイルを選択

TensorFlow pbファイル	準備したファイルを選択
メソッド	数字を出力する場合はCountを、文字列を返す場合はGroupByを選択
出力形式	詳細情報を出力する場合はJSONを、値のみ出力する場合はValueを選択
検出値を含める	検出値を含める場合に選択
信頼度の閾値	0.0から1.0の範囲で設定

TensorFlow Liteで準備するファイル

TensorFlow Liteモデルパッケージを作成するには以下のファイルを準備してください。

ファイル
Lite.tfliteファイル
LiteEdge.tfliteファイル
ラベルファイル

TensorFlow Liteで登録する項目

項目名	説明
モデルパッケージ名	英文字で設定
Lite.tfliteファイル	準備したファイルを選択
LiteEdge.tfliteファイル	準備したファイルを選択
ラベルファイル	準備したファイルを選択
メソッド	数字を出力する場合はCountを、文字列を返す場合はGroupByを選択（“画像分種”を選択した場合は指定できません）
出力形式	詳細情報を出力する場合はJSONを、値のみ出力する場合はValue（“画像分種”を選択した場合はName）を選択
検出値を含める	検出値を含める場合に選択
信頼度の閾値	0.0から1.0の範囲で設定

ディスク管理

ディスク使用量の表示とメンテナンス設定で、ディスクにメディアならびにログの保存期間と処理実行開始時間の登録をすることができます。

surface - Gravio Studio 4

< surface

HubKit 情報

バックアップ/リストア

機能パッケージ

基本プロパティセット

画像推論モデル

ディスク管理

ディスク使用量

	フリー	270.92 GB	メディアデータ総量	0.00 B
	ログファイル総量	1.91 KB	センサーデータ総量	36.00 KB

メンテナンス設定

開始時間

3

00

次の実行時刻 2021年5月19日 3:00

メディアデータの保存期間

30

日分

ログの保存期間

30

日分

センサーの保存期間

30

日分

デバイス

トリガー

データビューア

ログ

設定

7.4. アクション構成ビュー

アクション構成ビューでは、トリガーによって実行されるプログラムを記述できます。
このプログラムを「アクション」と呼びます。

典型的には、センサデータはいくつかのアクションをトリガし、それらのアクションにいくつかのパラメータを渡す。

「アクション」を作成または編集するには、サーバーの概要に進み、サーバーカードの下端にある中間のアイコンをクリックしてください。

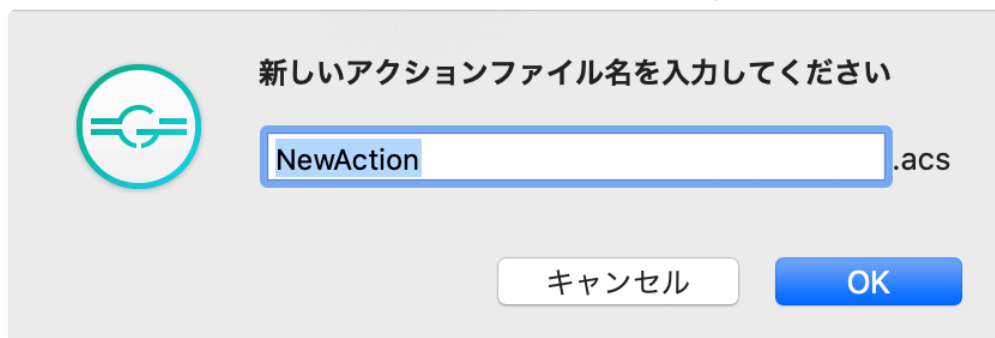
クリックすると、次のUIが表示されます。

アクションを追加するには、左上のメニューの“+”マークをクリックします。

7.4.1. アクションの作成

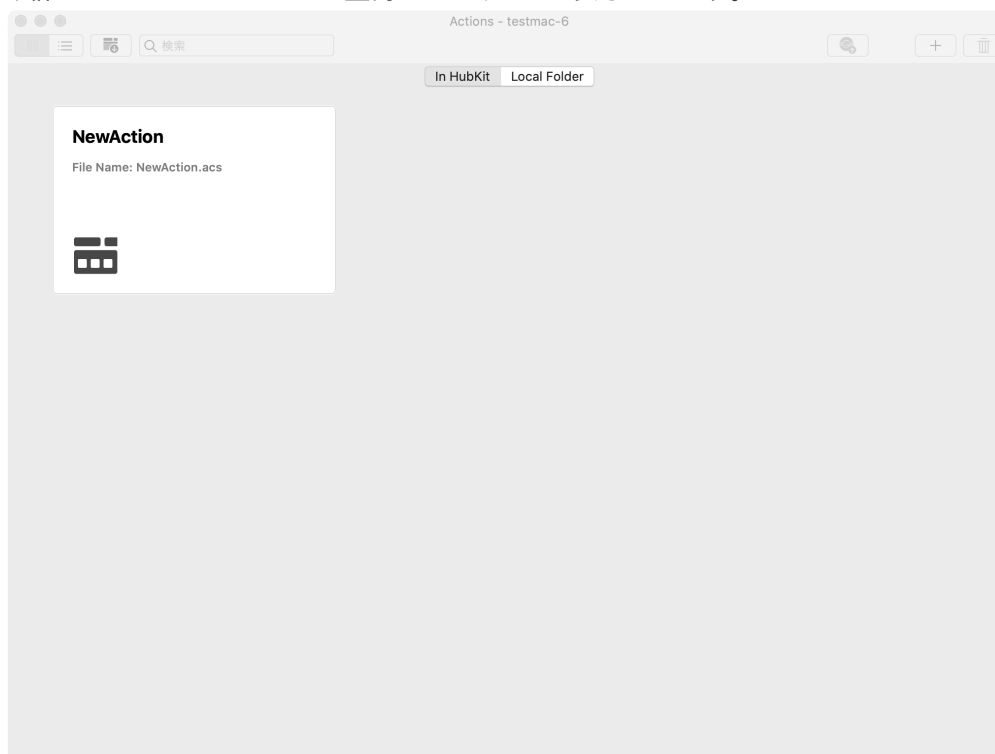
新しいアクションを作成するには、エッジサーバー内のアクション構成ビューの右上にある「+」記号をクリックします。

1. 新しいアクションを保存するアクションファイルのファイル名を入力します。



A dialog box titled "新しいアクションファイル名を入力してください" (Please enter a new action file name). It features a circular icon with a stylized 'G' on the left. In the center, there is a text input field containing "NewAction" followed by ".acs". At the bottom right, there are two buttons: "キャンセル" (Cancel) and "OK".

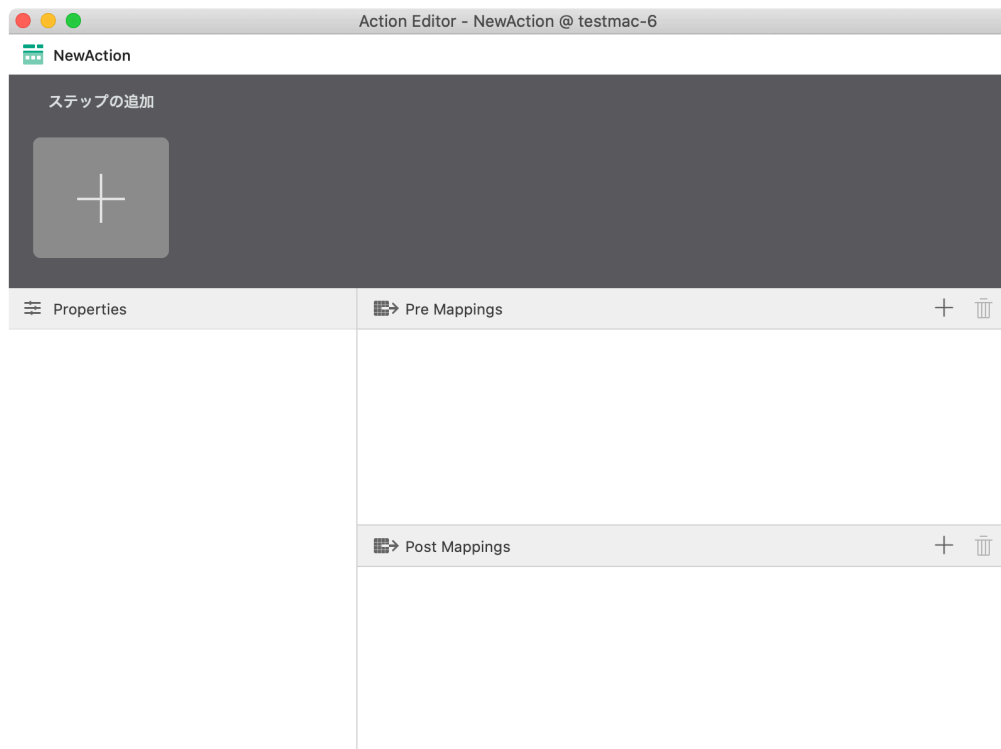
OKを押すと、新しいアクションファイルが登録されてリストに表示されます。



そのアクションファイルをダブルクリックすると、アクションエディタが開きます：

アクションは、実行されているシーケンス内で左から右に並べることができるいわゆる「ステップ」で構成されます。

新しいステップを追加するには、右上の「+」アイコンをクリックし、「アクションコンポーネントライブラリ」から新しいアクションコンポーネントを選択します。



コンポーネントを選択すると、ステップを構成してその変数にアクセスしたり、その変数进行处理したりする次のビューが表示されます。

各コンポーネントの詳細な説明については、[アクションコンポーネント](#) をご覧ください。

7.4.2. アクション内で利用できる変数と変数型

アクションプロパティ (Action Property)

“**ap.**” のプリフィックスをつけることで参照します。

アクションの実行時に外部から渡されるプロパティとして定義できます。

アクション変数 (Action Variable)

“**av.**” のプリフィックスをつけることで参照します。

アクション内で参照できる変数。

コンポーネントプロパティ (Component Property)

“**cp.**” のプリフィックスをつけることで参照します。コンポーネントの動作を決める値。それぞれのコンポーネントが持っている変数。

例えば、HTTPRequestコンポーネントならMethod(GET|POST...)/URLプロパティなど。

コンポーネント変数 (Component Variable)

“**cv.**” のプリフィックスをつけることで参照します。

コンポーネントで暗黙で宣言されている変数があります。

例) cv.StatusCode

HTTPRequestコンポーネントならHTTPステータスコードやレスポンスヘッダ類がcvとなります。

トリガー変数 (Trigger Variable)

“**tv.**” のプリフィックスをつけることで参照します。

センサーから送信されたデータは、以下のトリガー変数で、Actionプログラム内で参照できます。

トリガー変数名	説明
tv.AreaId	エリアID
tv.AreaName	エリア名
tv.LayerId	レイヤーID
tv.LayerName	レイヤー名
tv.KindId	データ種別ID
tv.KindName	データ種別名
tv.PhysicalDeviceId	センサーの物理デバイスID
tv.PhysicalDeviceName	センサーの物理デバイス名

tv.Timestamp	センサーからデータが出力されたときの時刻。フォーマットはRFC3339Nano
tv.DataId	このセンサーデータ固有ID
tv.Data	センサーデータの値。tv.Dataは最初のコンポーネントのcv.Payloadと同じになります

トリガープロパティ (Trigger Property)

イベントトリガーやタイマートリガーで設定された、トリガープロパティをアクション内で参照できます。

“tp.” のプリフィックスをつけることで参照します。

例) tp.KeyAreaName、tp.KeyLayerName、tp.TriggerName



値(型)

変数型	説明
Int	整数
Double	浮動小数点
String	文字列
Bool	true/false
DateTime	日時
バイト列	バイト列
配列	配列
Object	オブジェクト

型	例
リテラル	[1, 2, 3]
配列	["abc", "def", "hij"] ["abc", 123, cv.variable]
Object	{"key1":123, "key2":"abc", "key3":cv.variable, "key4":[1, 2, 3], "key


```
5": {"subkey1": "def"}}
```

型変換

それぞれの型からそれぞれの型への変換が可能です。

型	例
Boolean	数値系は0の時はfalse、それ以外はtrue。日時の時Unix時間で0の時はfalse、それ以外はtrue。文字列の時は“true”/“1”の時はtrue、それ以外はfalse。バイト列は文字列に変換後に文字列のルールに従う。それ以外の型のときはfalse
整数	Booleanはtrueの時は1、それ以外は0。浮動小数はGo言語のキャストのルール。日時の時はUnix時間を数値とする。文字列の時はstrconv.ParseIntのルールに従う。バイト列は文字列に変換後に文字列のルールに従う。それ以外の型は0
浮動小数	Booleanはtrueの時は1.0、それ以外は0.0。整数はGo言語のキャストのルール。日時の時はUnix時間を数値とする。文字列の時はstrconv.ParseFloatのルールに従う。バイト列は文字列に変換後に文字列のルールに従う。それ以外の型は0.0
文字列	Booleanはtrueの時は“true”、それ以外は“false” 整数はstrconv.FormatInt、浮動小数はstrconv.FormatFloatで文字列に直す。日時の時は、time.RFC3339Nanoフォーマットで文字列に直す。バイト列はutf-8バイト列とみなし文字列にする。エラーがある場合は空文字。配列、Objectはjson.Marshalをかけて文字列にする
日時	数値系とBooleanは整数に直すルールで整数にした後に、Unix時間とみなして日時に変換。文字列は、下記フォーマットを順に試してきたらそれを日時とみなし、それ以外は0(日時の0とは、1年1月1日00:00:00のこと) time.RFC3339Nano, “2006-01-02T15:04:05.999 MST”, “2006-01-02T15:04:05.999”, “20060102T150405.999 MST”, “20060102T150405.999”, バイト列は文字列に直し文字列と同じルール。それ以外は0
バイト列	Booleanと数値系と日時は文字列に直しutf-8バイト列にする。文字列はutf-8バイト列にする。配列、Objectはjson.Marshalを書けて文字列をバイト列にする。 配列、Object 数値系と、日時はその値を1つ含む配列にする。文字列は、JSON文字列とみなしjson.Unmarshalする。エラーの場合はサイズ0の配列。バイト列は文字列に直し、文字列と同じルール。配列とObjectはそのまま。それ以外の型はサイズ0の配列

演算子

使用できる演算子を以下の表で説明します。

演算子	説明
+	加算
-	減算
*	乗算
/	除算

**	べき乗
&	ビット論理積 (AND)
	ビット論理和 (OR)
^	ビット排他的論理和 (XOR)
%	剰余

比較演算子

使用できる比較演算子を以下の表で説明します。

比較演算子	説明
>	より小さい
>=	以下
<	より大きい
<=	以上
==	等価
!=	不等価
=~	正規表現マッチ
!~	正規表現非マッチ

論理演算子

使用できる論理演算子を以下の表で説明します。

論理演算子	説明
	論理 OR
&&	論理 AND

7.4.3. アクション内でObject型の変数の値を参照する方法

アクション内で利用できるObject型の変数の値を参照する場合、変数にドットで繋いだセレクトーや角括弧で要素にアクセスします。

Object型の変数の例として、IVARの顔認証機能を使用するとトリガーマネージャーでアクションを実行した場合に以下のようなJSONデータがアクションに届きます。
(一部のデータのみ抜粋しています。)

このデータはアクションの実行時に `tv.Data` としてどのコンポーネントからも参照することもできます。または最初のコンポーネントのPayloadにもなっているため、最初のコンポーネントでは `cv.Payload` としても参照可能です。

```
{
  "id": "XXXXXX",
  "source": {
    "channel": 4,
    "bacId" : "YYYYYY",
    "taskId" : "ZZZZZZ",
    "ivarId" : "000000"
  },
  "common": {
    "description": "Face Recognition",
    "presentImageTime": "2019-06-17T07:53:22.986Z",
    "time": "2019-06-17T07:53:22.986Z",
    "type": "FR"
  },
  "iod": {
    "age": "MIDDLE",
    "ageRange": [
      38,
      53
    ],
    "faceAvailable": true,
    "frSuitable": "NA",
    "gender": "MALE",
    "glasses": "REGULAR",
    "groupId": 0,
    "liveness": "NO_RESULT",
    "livenessProbability": "0.0000",
    "mask": "NA",
    "objectId": 649,
    "objectType": "FACE",
    "occlusion": "NA",
```

```
    "race": "NA"
  },
  "fr": {
    "candidates": [
      {
        "customId": "00001",
        "displayName": "Person B",
        "id": "6",
        "indentifyGroup": "Company A",
        "similiarityScore": "0.7272"
      }
    ],
    "fdrVersion": "60032084",
    "indentifyGroup": "Set2",
    "mmsReturn": "OK"
  }
}
```

例えば fr -> candidates -> displayName を参照する場合には

```
tv.Data.fr.candidates[0].displayName
```

と記述します。

IVARのJSONは単純なキー名ですのでドットで繋いだセレクトラーで参照できますが、スペースやハイフンなどがキー名に使われている場合には以下のような参照も可能です(display Nameとスペースが入っていた場合)。frやcandidatesはそのままドットで繋いで混ぜて使ってもかまいません。

```
tv.Data["fr"]["candidates"][0]["display Name"]
```

```
tv.Data.fr.candidates[0]["display Name"]
```

上記は値の参照、取得ですが、Filterコンポーネントの条件式を入れるところでも同様に使うことが可能です。

```
cv.Payload.fr.candidates[0].displayName == "Person B"
```

これらの機能はアクション以外にもトリガー条件や式を書けるところで使えます。

トリガーのIVARの条件でも式を入力することができ、男性のみアクションを起動する場合には以下の式を記述します。

```
tv.Data.iod.gender == "MALE"
```

7.4.4. アクションコンポーネントの動作

アクションコンポーネントには入力ペイロード(cv.Payload)と出力ペイロード(cv.Payload)というコンポーネント変数(cv)が用意されています。

Pre-Mappingで使う場合は入力ペイロード(cv.Payload)として、Post-Mappingで使う場合は出力ペイロード(cv.Payload)として利用することができます。

入力ペイロード

「入力ペイロード」は直前のコンポーネントの「出力ペイロード(cv.Payload)」と同じです。

アクションで一番最初に置かれたコンポーネントはトリガーから渡された [トリガー変数\(tv.Data\)](#)が [入力ペイロード\(cv.Payload\)](#) になります。

出力ペイロード

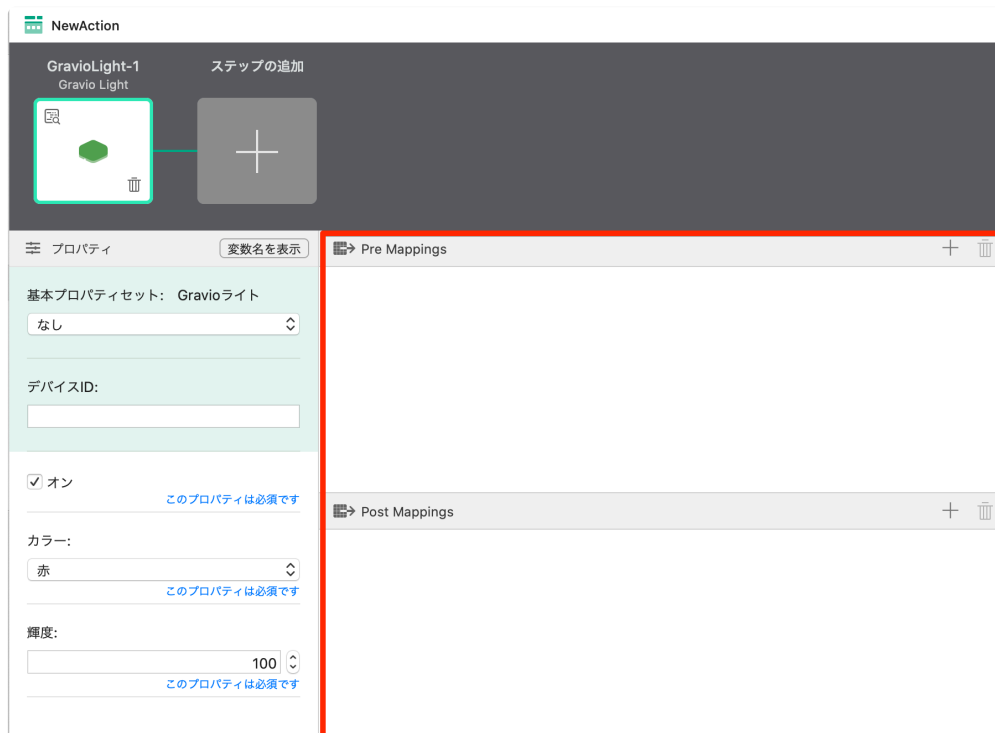
「出力ペイロード」は次のコンポーネントの「入力ペイロード」になります。

「出力ペイロード」は各コンポーネントの処理で作られるので、それぞれのコンポーネントでその内容は定義されています。

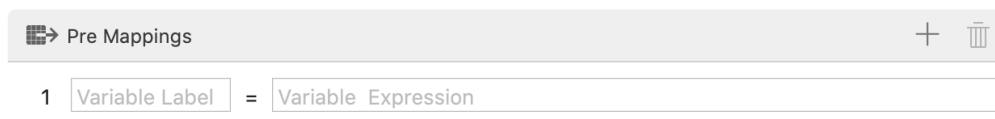
個々のアクションコンポーネントの解説でどのような「出力ペイロード」が出されるか説明されていますので、個別の [アクションコンポーネント解説](#) を参照してください。

7.4.5. Pre Mappings / Post Mappingsでの式の記述の仕方

Action Editor画面でコンポーネントを選択した時に、右側にPre Mappings / Post Mappingsの編集エリアが表示されます。

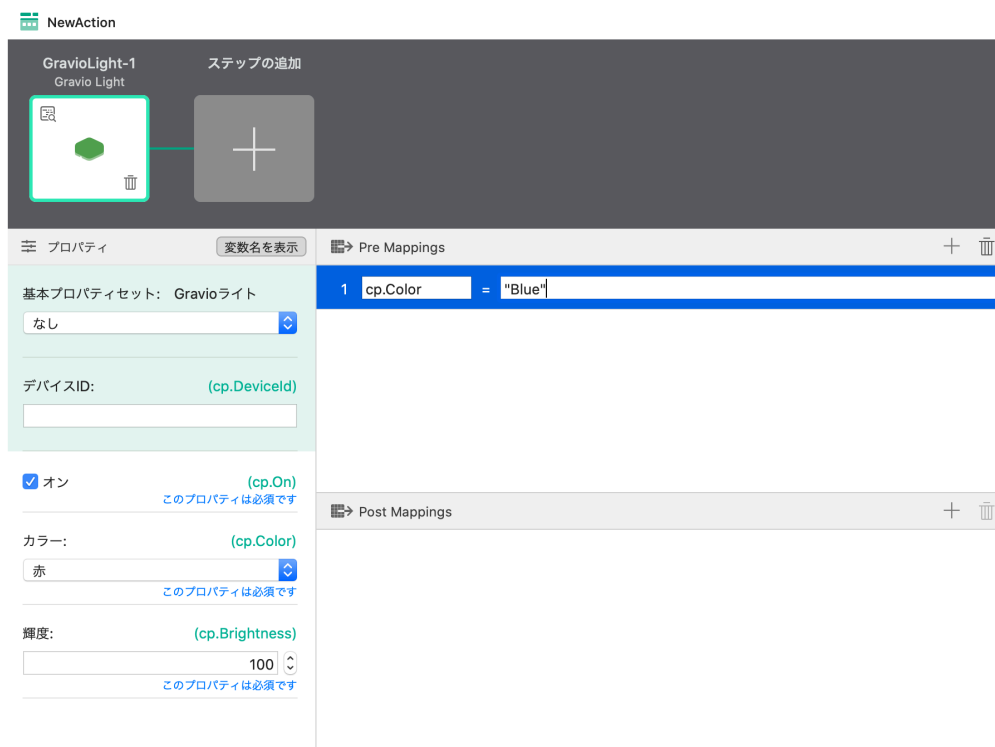


「+」ボタンを押すと入力フィールドが表示されます。



左側に変数名、右側に変数に代入する式を記述できます。

例.) cp.Color = “Blue”



式の記述

右辺の式には、四則演算(+ / - / * / /)や [関数](#) を記述できます。また、変数として、tp. / tv. / ap. / av. / cp. / cv. などのプリフィックスを持つ変数を記述できます。

右辺の式の評価結果を左辺に代入するとき、右辺の値の型(Int , Double , String , Bool , DateTime)は左辺の型に変換されます。

代入ではない式(例 : cv.Payload > 10)を記述した場合は、Bool型の値を返します。Bool型を返すには、数値や文字列などに対して比較演算をすることで返すことができます。

式例

```
cv.Payload > 10
```

```
cv.Payload == "abc"
```

```
true
```

式の評価

式の評価にはGval (Go eVALuate) を使用しています。詳細はGval のサイトを参照ください。 <https://github.com/PaesslerAG/gval>

7.4.6. Pre Mappings / Post Mappingsの式の中で利用できる関数

Pre Mappings / Post Mappingsの中で記述する式の右辺には以下の様な関数を記述して利用できます。
個別の詳しい関数の説明は、[関数解説](#)をご覧ください。

文字列関係

Len Contains RuneCount HasPrefix HasSuffix Index Join LastIndex Repeat Replace Split
ToLower ToUpper Trim TrimLeft TrimRight TrimPrefix TrimSuffix

型変換関係

ToBool.ToInt.ToFloat.ToString.ToDate.ToBinary.ToJSON

エンコード関係

URLPathEscape URLPathUnescape URLQueryEscape URLQueryUnescape MD5 SHA1 SHA256 BASE64
DecodeBASE64 BASE64URL DecodeBASE64URL

日付関係

Now Year Month Day Hour Minute Second Weekday ToLocal ToUTC ToTimezone DateFormat DateParse

環境変数

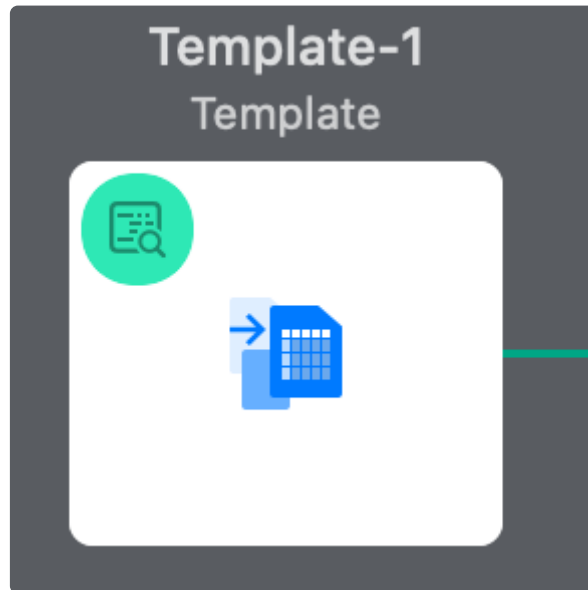
Env

正規表現

RegExpMatch RegExpFind RegExpFindSubmatch RegExpFindAll RegExpReplace UUID

7.4.7. デバッグログ出力

コンポーネントの左上のボタンを押すことで、デバックログを出力できます。



本来コンポーネントが独立して並行に動作しますが、デバックログを出力しながらアクションを実行ときは、アクションエンジン内で同期処理を行っています。

デバックログはtp/tv/ap/av/cp/cvのプロパティ、変数を順にログに出力します。

アクションプログラムは以下のような順で処理されます。

1. Payloadを含むコンポーネント変数をキューから取得する
2. 前のコンポーネントから渡されたPostMappingsの値がある場合は代入処理が行われる
3. PreMappingsの式評価、値代入が行われる
4. プロパティに応じてコンポーネントがなにか処理する
5. コンポーネントの処理結果に応じて、新たなPayloadを含むコンポーネント変数を作成する
6. PostMappingsが式評価のみが行われ結果の値は先の新たなPayloadを含むコンポーネント変数に保管される
7. 次のコンポーネントのキューに新たなPayloadを含むコンポーネント変数を入れる
8. 1に戻る

デバックログ機能をONにすると、上記コンポーネントの処理手順の中で、3のPreMappingsの式評価、代入が行われたあとの変数の内容が出力されます。

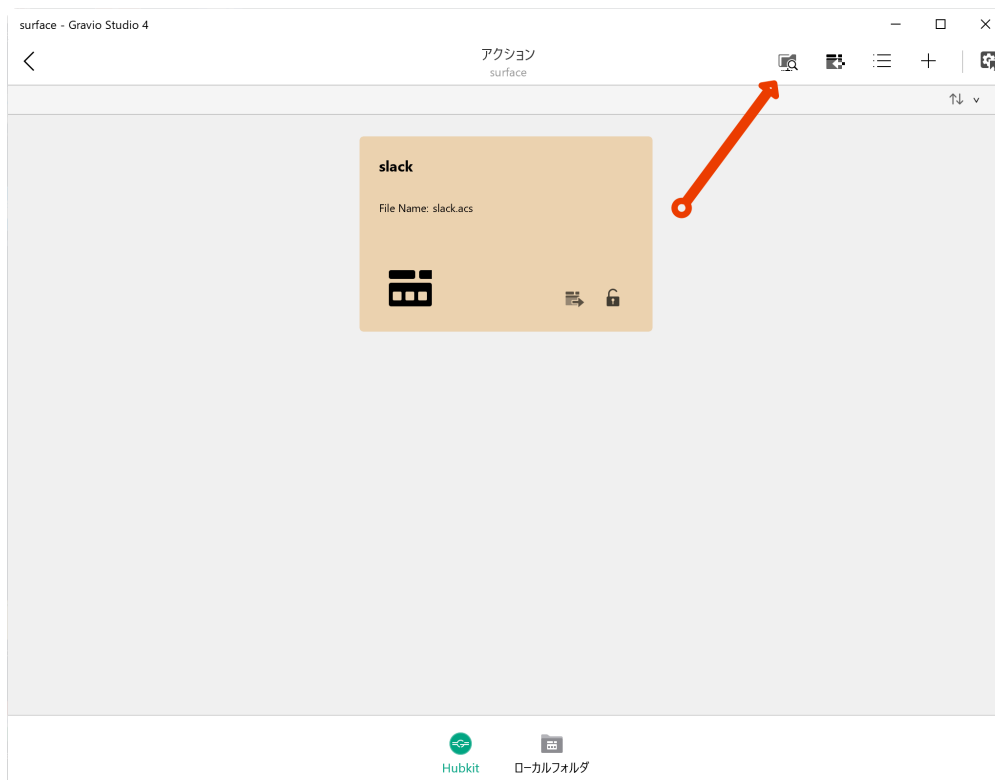
7.4.8. ファイルビューア機能

HubKitの一部のアクションコンポーネントは、actmgr/dataディレクトリのファイルを読み込んだり、そのディレクトリにファイルを保存したりすることができます。

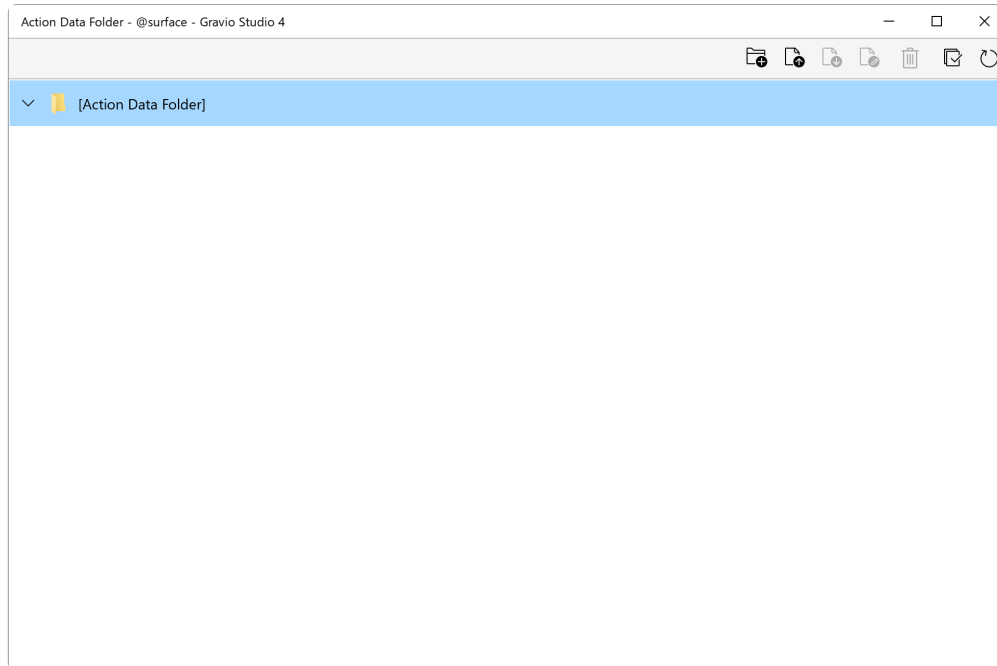
このディレクトリに保存されたファイルをGravio Studioで取り出したり、ローカルにあるファイルをGravio Studioでそのディレクトリにアップロードすることが可能です。

以下のUIをつかって操作します。

ファイルビューア機能を開く

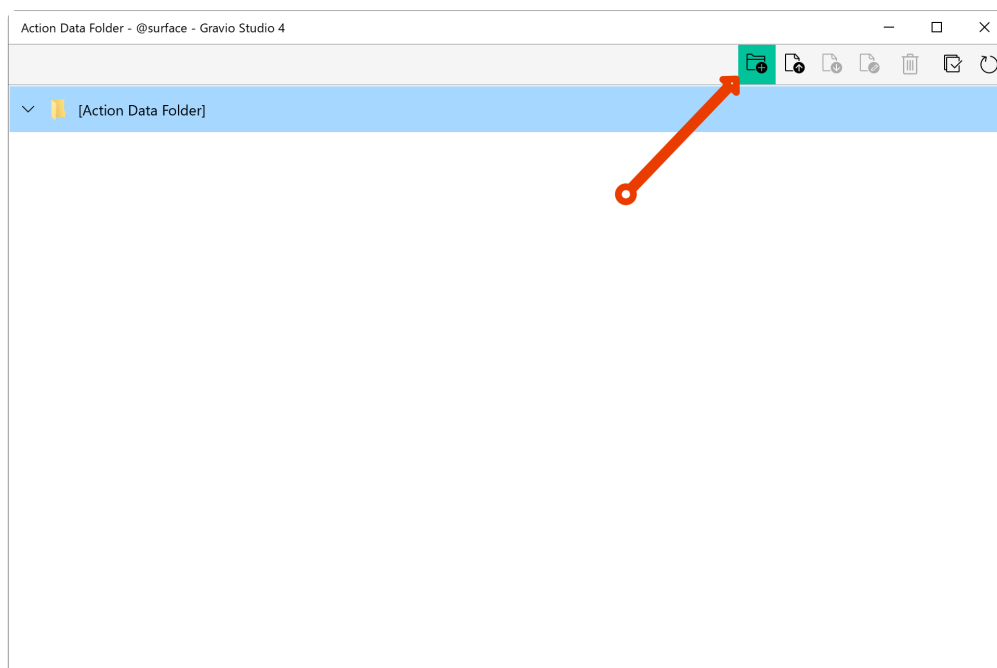


HubKitのアクションリスト画面からファイルビューアアイコンをクリックして開きます。

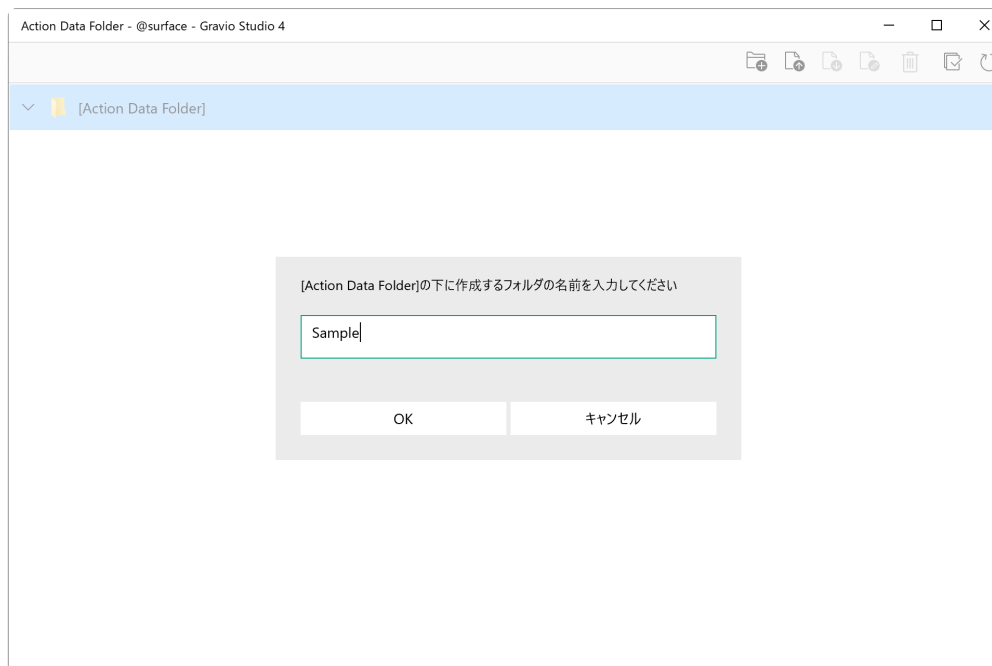


ファイルビューアではディレクトリ (actmgr/data) のファイルがツリー形式で表示されます。

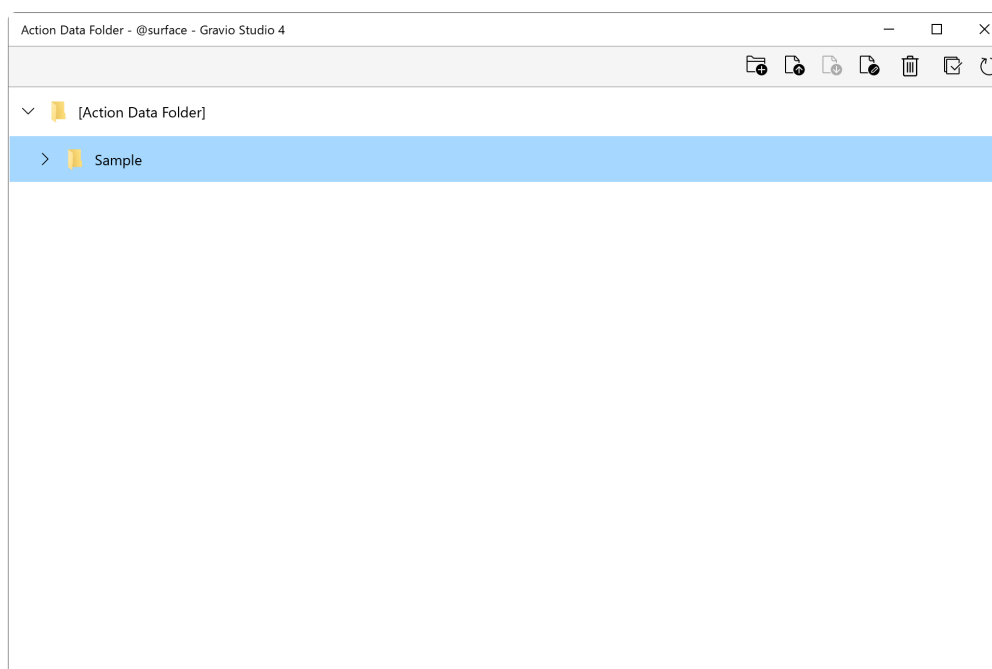
サブディレクトリを作成する



ディレクトリにサブディレクトリを作成するにはディレクトリ作成ボタンを押します。

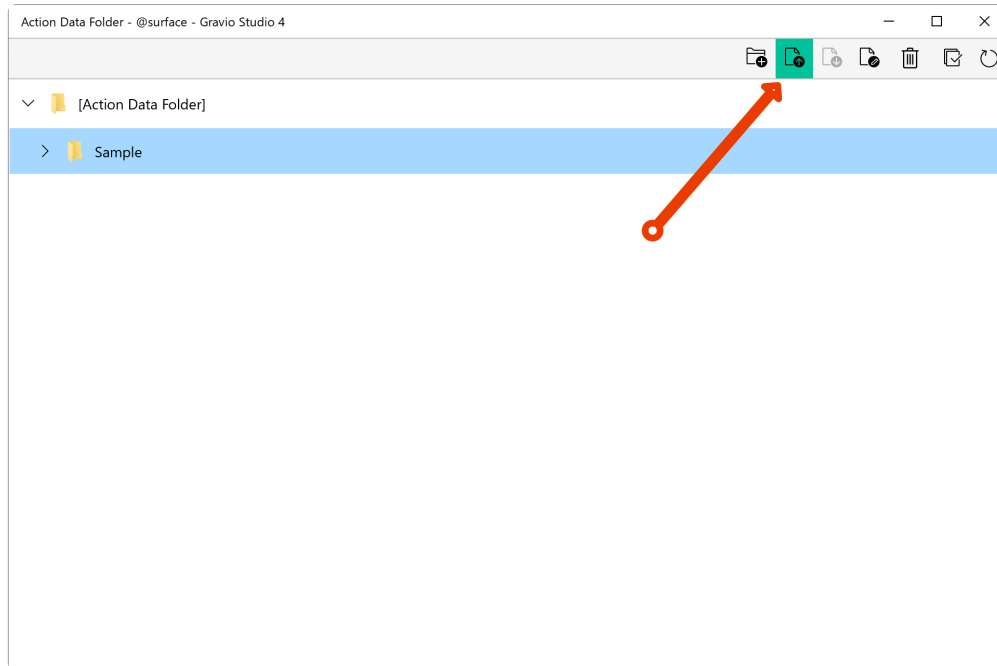


“Sample”というサブディレクトリを作成します。

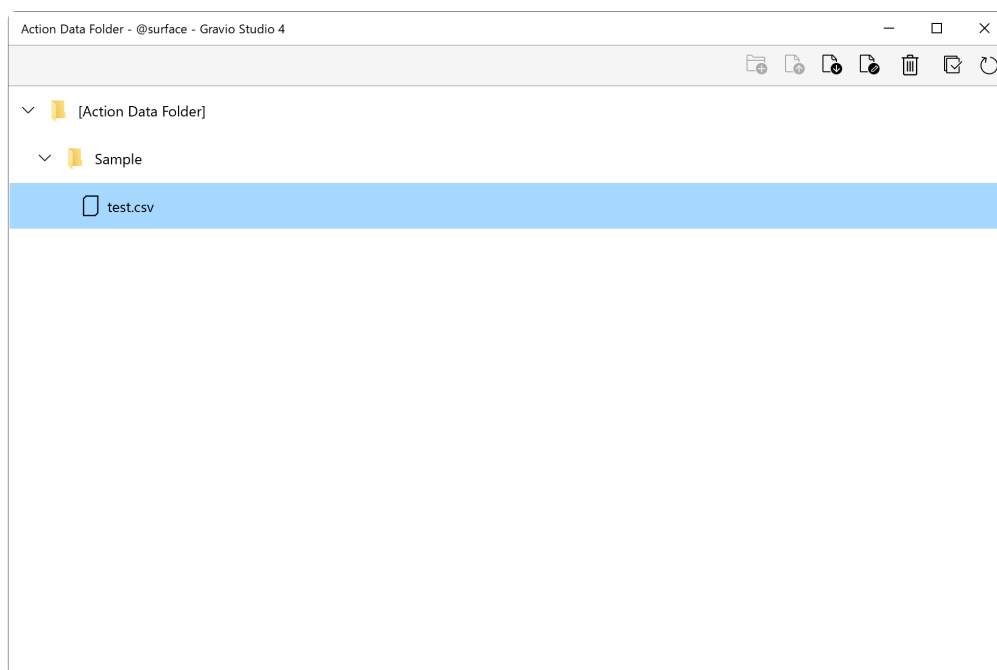


“Sample” ディレクトリが表示されます。

ファイルをアップロードする

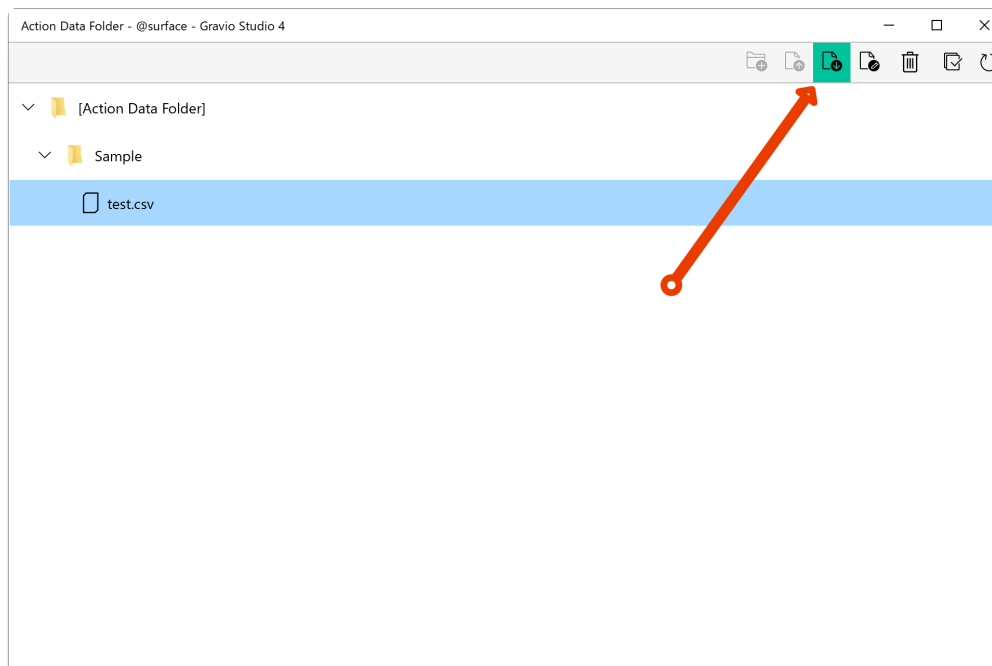


ファイルをアップロードするにはディレクトリ “Sample” を選択してアップロードボタンを押します。
ファイル選択ダイアログが表示されますのでアップロードするファイル “test.csv” を選択します。



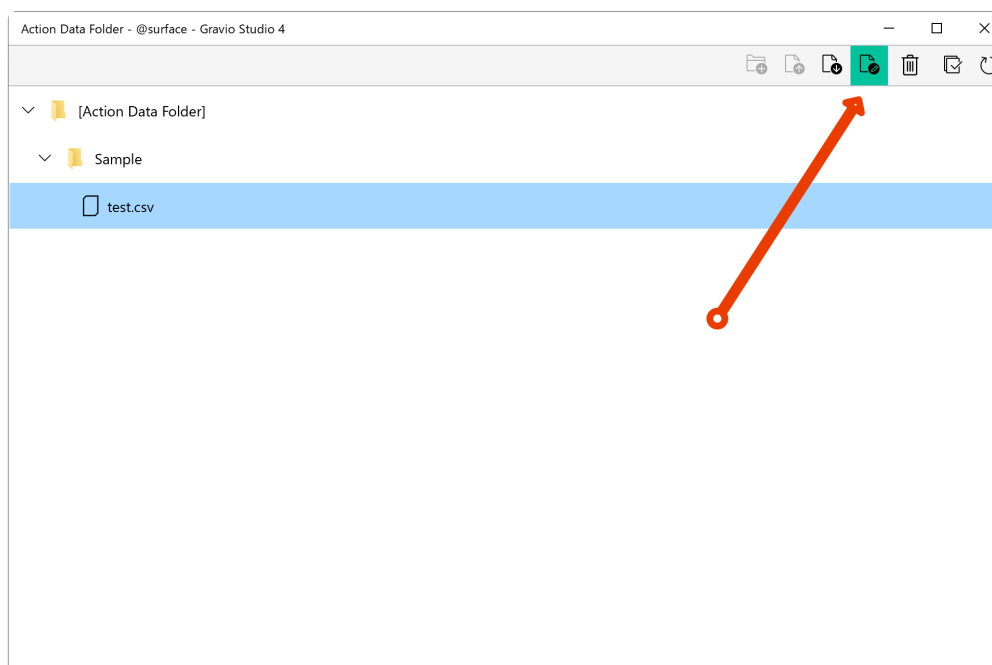
アップロードした “test.csv” が “Sample” ディレクトリに表示されます。

ファイルをダウンロードする

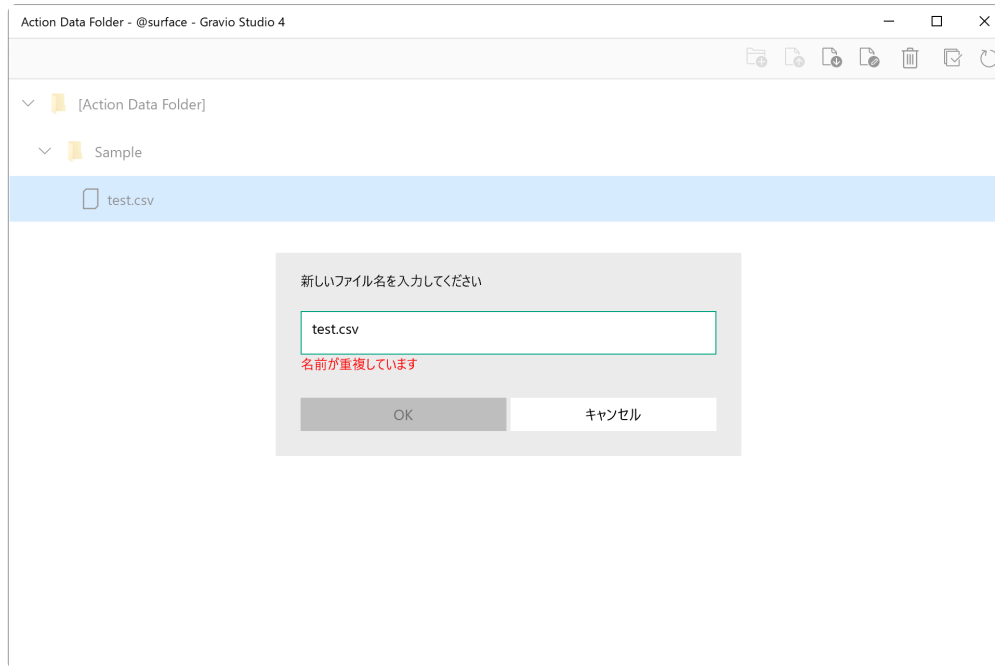


ファイルをダウンロードするにはファイルを選択してダウンロードボタンを押します。
ファイル保存ダイアログが表示されますのでダウンロードするファイル名を入力します。
ダウンロードしたファイルはダウンロードディレクトリに保存されます。

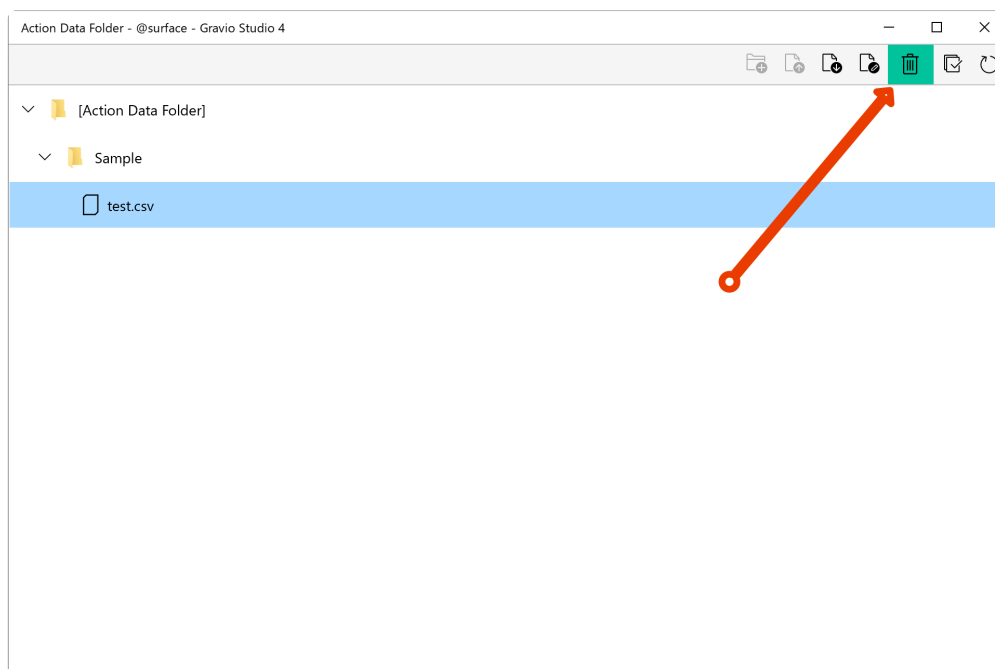
ファイル名を変更する



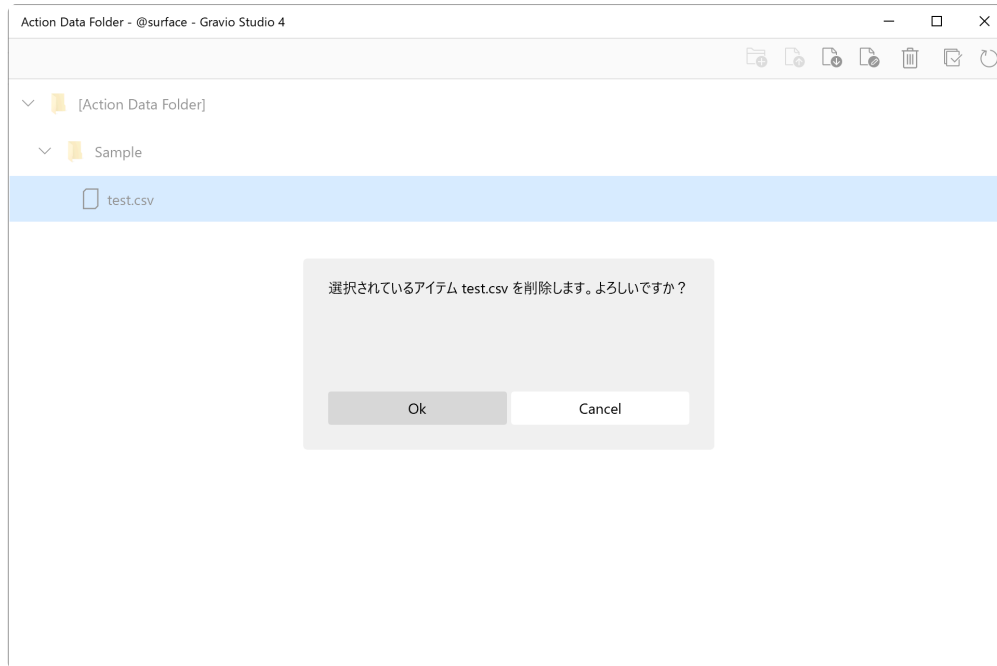
ファイル名を変更するファイルを選択して名前変更ボタンを押します。
ディレクトリ名を変更することもできます。



ファイルを削除する

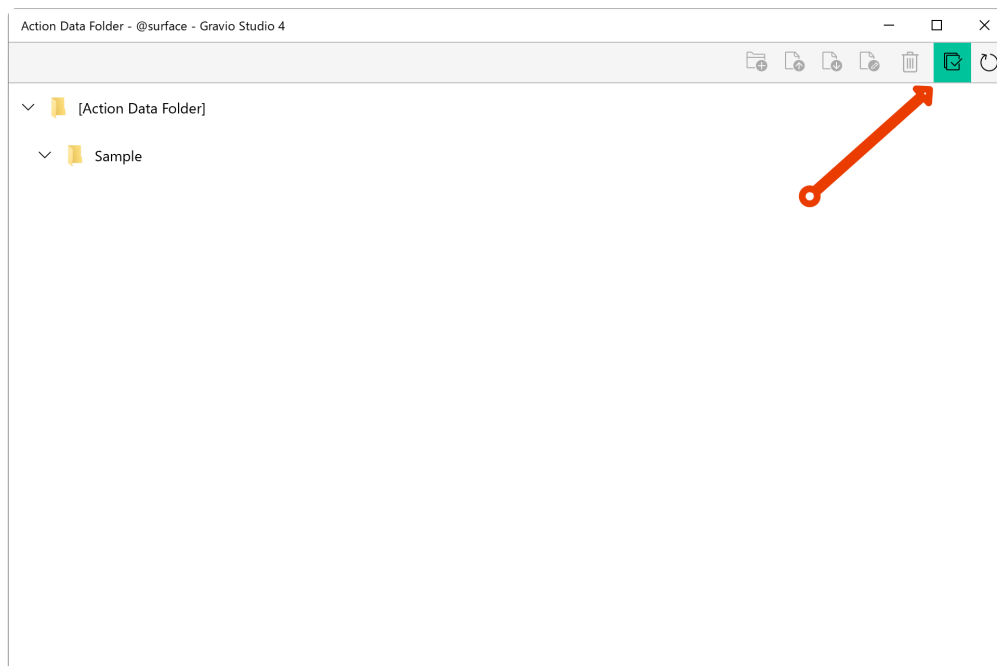


削除するファイルを選択して削除ボタンを押します。
ディレクトリを削除することもできます。



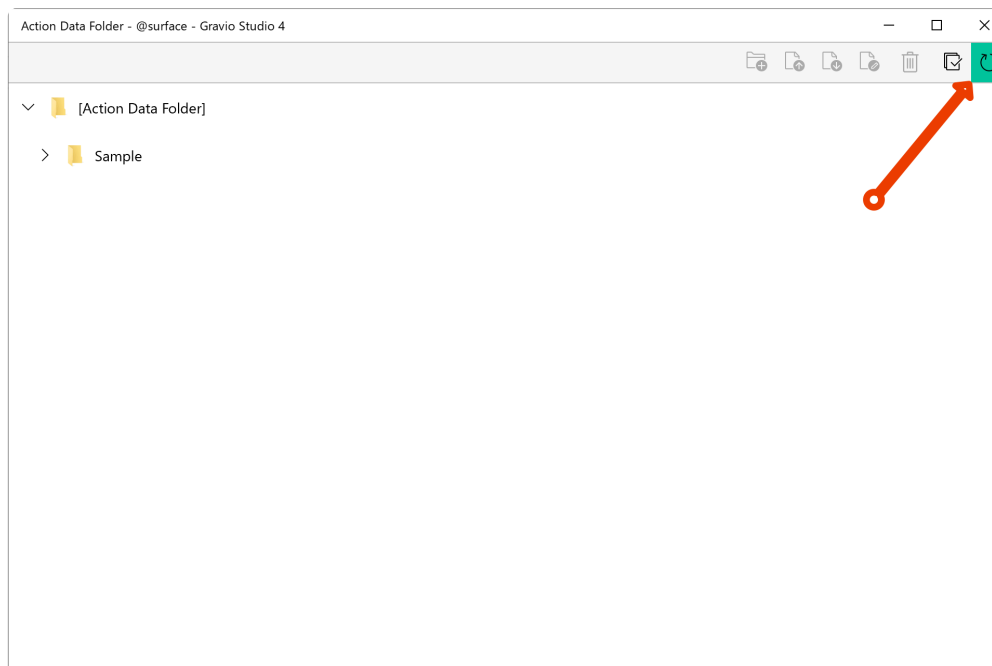
ファイル/ディレクトリを削除する確認ダイアログが表示されます。

ディレクトリのツリーを広げる



ディレクトリツリーをすべて開く場合に開くボタンを押します。

ファイル一覧の読み込み



HubKitのファイルリストを再読み込む場合に読み込みボタンを押します。

7.4.9. アクションコンポーネント解説

アクションは、1つ以上のステップで構成されます。各ステップは、アクションコンポーネントを配置して、入力、出力を定義できます。

このセクションでは、利用可能なさまざまなコンポーネントについて説明し、それらの使用方法について説明します。

アクションで使用するコンポーネントは、使用しているGravio ライセンスにより異なります。
ライセンスによっては実行できないコンポーネントがあります。

ライセンスによる実行できるコンポーネントの詳細は [こちら](#) を参照してください。

7.4.9.1. Amazon Polly Speech

Amazon Pollyは入力テキストをスピーチに変換して音声として再生します。

Amazon AWSアカウントにアクセスし、アクセスキーとシークレットアクセスキーを取得します。

[プロパティ]フィールドに入力します。

このコンポーネントを実行するとGravioが音声を再生します。

The screenshot shows the 'NewAction' dialog for the 'AWS Polly Speech' component. The 'Properties' tab is selected, showing various configuration fields. The 'Text' field is empty with a required message. The 'Voice' dropdown is set to 'Joanna English(US)-女性'. The 'Timeout' field is set to '60000'. The 'Basic Property' dropdown is set to 'AWSセッション'. The 'Access Key ID' and 'Secret Access Key' fields are empty with required messages. The 'Region' dropdown is set to '米国東部(バージニア北部) us-east-1'. The 'Pre Mappings' and 'Post Mappings' sections are also visible.

コンポーネントプロパティ

プロパティ名	説明
テキスト	喋らせたメッセージテキストを指定

音声	話者の音声のタイプを指定。テキストの言語と同じ言語の音声タイプを選択。cp.Voiceの値で設定する場合は、 "Zeina", "Zhiyu", "Naja", "Mads", "Lotte", "Ruben", "Nicole", "Russell", "Amy", "Emma", "Brian", "Aditi", "Raveena", "Ivy", "Joanna", "Kendra", "Kimberly", "Salli", "Joey", "Justin", "Kevin", "Matthew", "Geraint", "Celine", "Mathieu", "Chantal", "Marlene", "Vicki", "Hans", "Aditi", "Dora", "Karl", "Carla", "Bianca", "Giorgio", "Mizuki", "Takumi", "Seoyeon", "Liv", "Ewa", "Maja", "Jacek", "Jan", "Camila", "Vitoria", "Ricardo", "Ines", "Cristiano", "Carmen", "Tatyana", "Maxim", "Conchita", "Lucia", "Enrique", "Mia", "Lupe", "Penelope", "Miguel", "Astrid", "Filiz", "Gwyneth" のいずれか
音声エンジン	標準 または ニューラルのいずれか
音声処理	再生 または ファイル保存 のいずれか
ファイル名	音声をファイルに保存する場合のファイル名
タイムアウト	送信タイムアウト時間の指定
アクセスキーID	AWSから取得したアクセスキー
シークレット	AWSから取得したシークレットアクセスキー

アクセスキー	
リージョン	利用するAWSのリージョンを指定

注意：音声エンジンがニューラルエンジン (NTTS) に対応していない場合はValidationExceptionが発生します。

7.4.9.2. Aqara Smart Plug

このコンポーネントでは、AqaraスマートプラグをOn/Offすることができます。
このコンポーネントを利用するにはあらかじめAqaraスマートプラグをペアリングしておく必要があります。
デバイスIDのプロパティが空の場合は、全ての接続しているAqaraスマートプラグにコマンドを送ります。

出力ペイロード	説明
<code>cv.Payload</code>	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
基本プロパティ	「Aqaraスマートプラグ」カテゴリーの基本プロパティを指定します
デバイスID	On/OffするデバイスID, 何も設定されていないときは全てのAqaraスマートプラグに対してON/OFFを実行します。Aqaraスマートプラグを複数使用する場合は、デバイスIDをカンマ区切りで記述することができます
On	On/Offの指定. <code>cp.On</code> に指定する値は <code>"true"</code> または <code>"false"</code>

7.4.9.3. AWS S3 Download

AWSのS3にあるファイルをHubKit内で利用したいときに、S3からファイルをダウンロードするときに利用できるコンポーネントです。

ファイルはAction ManagerのDataディレクトリに保存されます。

The screenshot shows the 'NewAction' configuration window for the 'AWS S3 Download' action. The interface is divided into two main sections: 'Steps Addition' and 'Properties'. The 'Properties' section is further divided into 'Pre Mappings' and 'Post Mappings'. The 'Pre Mappings' section contains the following fields:

- 出力:** A dropdown menu set to 'ペイロード' (Payload). A blue link below it says 'このプロパティは必須です' (This property is required).
- ファイル名:** A text input field.
- バケット:** A text input field. A blue link below it says 'このプロパティは必須です' (This property is required).
- オブジェクトキー:** A text input field. A blue link below it says 'このプロパティは必須です' (This property is required).
- タイムアウト:** A text input field set to '60000'. A blue link below it says 'このプロパティは必須です' (This property is required).
- 基本プロパティ:** A dropdown menu set to 'AWSセッション' (AWS Session).
- アクセスキーID:** A text input field. A blue link below it says 'このプロパティは必須です' (This property is required).
- シークレットアクセスキー:** A text input field. A blue link below it says 'このプロパティは必須です' (This property is required).
- リージョン:** A dropdown menu set to '米国東部(バージニア北部) us-east-1'. A blue link below it says 'このプロパティは必須です' (This property is required).

コンポーネントプロパティ

プロパティ名	説明
出力	Actionプログラム内のPayloadに保持するか、ファイルとして保持するかを指定. cp.Outputに設定される値は "Payload", "File" のいずれか
ファイル名	ファイルとして保持する場合のファイル名
バケット	AWS S3でのバケット名
オブジェクトキー	AWS S3でのオブジェクトキー
タイムアウト	送信タイムアウト時間の指定
アクセスキー ID	AWSから取得したアクセスキー
シークレットアク	AWSから取得したシークレットアクセスキー

セスキー	
リージョン	利用するAWSのリージョンを指定

※ファイルパスを指定しないファイルの保存先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.4. AWS S3 Upload

AWS S3 UploadはAmazon S3にファイルをアップロードします。

Amazon AWSアカウントにアクセスし、アクセスキーと秘密鍵を取得します。
ファイルはアクションのデータフォルダのファイルが対象となります。

例) *.log,test.txt

この場合には拡張子がlogのファイルとtest.txtファイルがアップロードの対象となります。

The screenshot shows the 'NewAction' editor for 'AWSS3Upload-1'. The 'Properties' tab is active, displaying the following configuration fields:

- 入力:** A dropdown menu set to 'ペイロード' (Payload).
- ファイル名:** An empty text input field.
- バケット:** An empty text input field.
- オブジェクトキー:** An empty text input field.
- ACL:** A dropdown menu set to 'Private'.
- ストレージクラス:** A dropdown menu set to '標準' (Standard).
- タイムアウト:** A text input field with the value '60000'.
- 基本プロパティ:** A section containing:
 - 基本プロパティ:** A dropdown menu set to 'AWSセッション'.
 - アクセスキーID:** An empty text input field.
 - シークレットアクセスキー:** An empty text input field.
 - リージョン:** A dropdown menu set to '米国東部(バージニア北部) us-east-1'.

Each input field has a small blue icon and the text 'このプロパティは必須です' (This property is required) below it. The right side of the interface shows 'Pre Mappings' and 'Post Mappings' sections, both currently empty.

コンポーネントプロパティ

プロパティ名	説明
入力	Actionプログラム内のPayloadから読み込むか、ファイルから読み込むかを指定. cp.Inputの値で設定する場合は、“Payload”, “File”のいずれか
ファイル名	入力ファイル名
バケット	アップロードするAWS S3でのバケット名

オブジェクトキー	AWS S3でのオブジェクトキー
ACL	AWS S3でのACLの権限. <code>cp.ACL</code> の値で設定する場合は、 <code>"private"</code> 、 <code>"public-read"</code> 、 <code>"public-read-write"</code> 、 <code>"authenticated-read"</code> 、 <code>"aws-exec-read"</code> 、 <code>"bucket-owner-read"</code> 、 <code>"bucket-owner-full-control"</code> のいずれか
ストレージクラス	AWS S3でのストレージの種類. <code>cp.StorageClass</code> の値で設定する場合は、 <code>"STANDARD"</code> 、 <code>"REDUCED_REDUNDANCY"</code> 、 <code>"GLACIER"</code> 、 <code>"STANDARD_IA"</code> 、 <code>"ONEZONE_IA"</code> 、 <code>"INTELLIGENT_TIERING"</code> 、 <code>"DEEP_ARCHIVE"</code> のいずれか
タイムアウト	送信タイムアウト時間の指定
アクセスキー ID	AWSから取得したアクセスキー
シークレットアクセスキー	AWSから取得したシークレットアクセスキー
リージョン	利用するAWSのリージョンを指定

Access Control List (ACL) Permissionsの詳細につきましては[こちら](#)を参照してください。

Storage Classの詳細につきましては[こちら](#)を参照してください。

※ファイルパスを指定しないファイルの参照先は`actmgr/data`となりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.5. BrightSign

BrightSignコンポーネントは、BrightSignにコマンドを送信することができます。

Step 1 BrightSign ステップの追加

Properties

ホスト名 (cp.Hostname)
このプロパティは必須です。

ポート番号 (cp.Port)
5000

コマンド (cp.Command)
このプロパティは必須です。

Pre Mappings
No Pre Mappings have been added for this component

Post Mappings

出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
ホスト名	BrightSignのホスト名またはIPアドレス
ポート番号	BrightSignのUDPポート番号
コマンド	BrightSignに送信するコマンド

7.4.9.6. Camera Take Photos

接続されているカメラからアクションの実行タイミングで写真を保存することができます。

CameraTakePhotos

ステップの追加

+

Properties

物理デバイスID

(cp.Deviceld)

Camera192.168.68.140

枚数

(cp.Shots)

1

間隔(秒)

(cp.Interval)

10

保存先

(cp.SaveTo)

camera

Pre Mappings

No Pre Mappings have been adde

Post Mappings

No Post Mappings have been adde

事前にデバイスを一覧を表示して、接続されているカメラの物理デバイスIDを確認してください。

デバイス一覧

Serial

Zigbee

Firmware Version : 0.10.5.1

Camera

追加

IODATA TS-NA220W

Default Folder

Bluetooth

スキャン

現在接続されているデバイスはありません

物理デバイスID

Camera192.168.68.140

物理デバイス名

IODATA TS-NA220W

キャプチャー形式

Picture

キャプチャー間隔

10

☒ 画像の保存

☒ 検出フレームの描画

ユーザー

admin

パスワード

●●●●●●●●

保存

閉じる

出力ペイロード	説明
cv.Payload	保存した写真のファイル名を配列で出力します

コンポーネントプロパティ

プロパティ名	説明
--------	----

物理デバイスID	デバイス一覧に接続されているカメラの物理デバイスIDを指定します
枚数	コンポーネントの実行時に保存する写真の枚数を指定します（最大5枚）
間隔（秒）	写真を保存するインターバルを指定します
保存先	ファイルの保存先としてactmgr/dataのサブフォルダを指定します。写真はこのサブフォルダに保存されます

※コンポーネントの実行は枚数*間隔で指定された時間が必要となりますので、あまり長時間になる設定は避けてください。

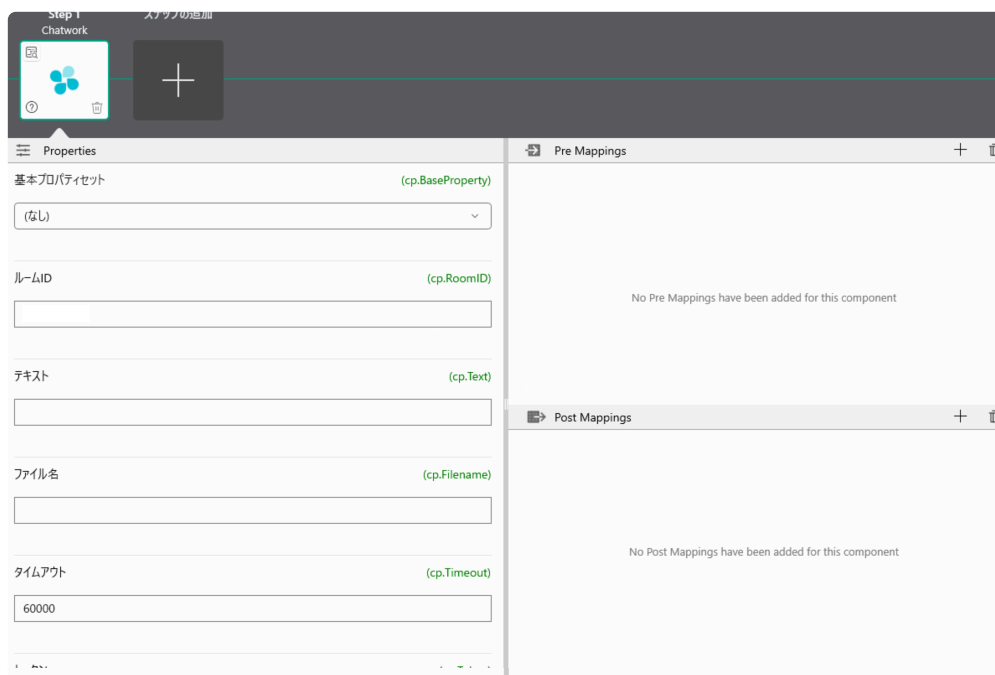
例)

枚数を5枚で、間隔を6秒とした場合、コンポーネントの実行時間の目安は $5 \times 6 = 30$ 秒となります。

7.4.9.7. Chatwork

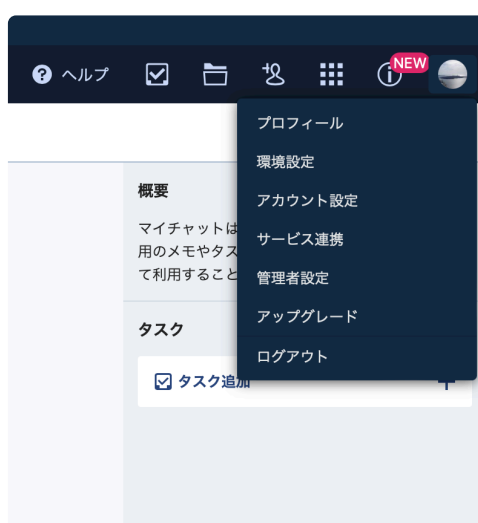
Chatworkコンポーネントは、Chatworkを使用してコンポーネントのプロパティに指定されたメッセージをポストすることができます。センサーからの情報を人に通知したりすると便利です。

※Chatworkコンポーネントを使用するには、管理者が初期設定としてサービス連携で設定し、APIトークンを取得する必要があります。



初期設定

Chatworkのサービス連携画面からAPIトークンを取得します。



アカウントのメニューからサービス連携を選択します。



APIメニューのAPIトークンの値を取得します。コンポーネントプロパティで使用します。



Chatrowkコンポーネントはグループチャットに投稿しますので、そのグループチャットを開き設定メニューからグループチャットの設定を選択します。

グループチャットの設定

チャット情報 ミュート 招待リンク 権限

チャット名：
グループチャット

概要：
このチャットの説明やメモ、関連するリンクなどを記入することができます

ルームID：

キャンセル 保存

グループチャットの設定画面に表示されているルームIDを取得します。コンポーネントプロパティで使します。

これで初期設定は完了です。

コンポーネントプロパティ

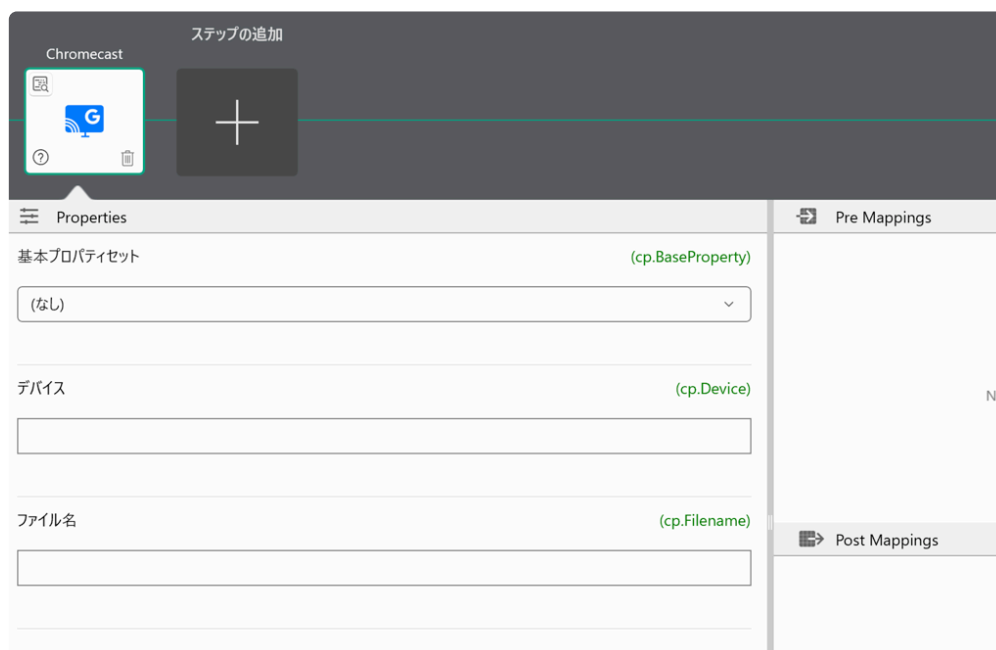
プロパティ名	説明	例
基本プロパティ	「Chatwork Token」カテゴリーの基本プロパティを指定します	
ルームID	投稿するグループのルームID(初期設定で取得したもの)	
テキスト	Chatworkに投稿するメッセージ文字列	例) "現在の気温は"+cv.Payload+"度です"
ファイル名	ファイルを送信します	
タイムアウト	APIのタイムアウト	
トークン	APIトークン(初期設定で取得したもの)	

※テキストとファイルの両方を指定した場合は、テキストとファイルが送信されます。

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.8. Chromecast

ChromecastコンポーネントはChromecast デバイスでFilenameプロパティで指定されたビデオ、音声を再生するコンポーネントです。



コンポーネントプロパティ

プロパティ名	説明
デバイス	ChromecastデバイスのIPアドレス
ファイル名	読み込むファイルのファイル名 (サポートされているファイル mp4/webmとmp3/ogg/wav形式)

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.9. CSV Read

CSV Readコンポーネントは入力としてCSVデータを取り、これを解析して、「出力形式」コンポーネントプロパティ(`cp.Output`)で指定された形式で出力ペイロードを作成します。

CSVReadには「入力」プロパティ(`cp.Input`)があり、データソースを選択できます。「ペイロード」を選ぶと直前のコンポーネントから入ってくる入力ペイロード(`cv.Payload`)をCSVの元データとし、Fileを選ぶと「ファイル名」プロパティ(`cv.Filename`)で指定されたファイルをCSVの元データとします。

入力ペイロード(`cv.Payload`)がバイト列や文字列だった場合は、そのままCSVの元データとなりますが、それ以外の型の場合は文字列に変換してからCSVの元データとして処理します。

出力ペイロード	説明
「出力形式」コンポーネントプロパティ(<code>cp.Output</code>)で、「配列」を選択した場合	CSVの各フィールドを文字列にした配列を各行毎に出力ペイロード(<code>cv.Payload</code>)にいて次のコンポーネントに渡します。
「出力形式」コンポーネントプロパティ(<code>cp.Output</code>)で「オブジェクト」を選択した場合	CSVの1行目をキーとみなしJSON Objectを生成し各業毎に出力ペイロードにいて次のコンポーネントに渡します。

CSV例

```
MailAddress,FirstName,LastName
yamada.taro@asteria.com,Taro,Yamada
suzuki.hanako@asteria.com,Hanako,Suzuki@
```

「オブジェクト」を選択したときのJSON Object例

```
{"MailAddress":"yamada.taro@asteria.com", "FirstName":"Taro", "LastName":"Yamada"}
{"MailAddress":"suzuki.hanako@asteria.com", "FirstName":"Hanako", "LastName":"Suzuki"}
```

NewAction

CSVRead-1
CSV Read

ステップの追加

プロパティ

変数名を表示

Pre Mappings

入力:

ペイロード

このプロパティは必須です

ファイル名:

エンコーディング:

utf-8

出力形式:

配列

このプロパティは必須です

開始行:

0

行数:

-1

Post Mappings

コンポーネントプロパティ

プロパティ名	説明
入力	入力を「ペイロード」もしくは「ファイル」のいずれかを指定. cp.Inputの値で設定する場合は、"Payload", "File" のいずれか
ファイル名	ファイル入力の場合のファイル名を指定
エンコーディング	ファイル出力のエンコーディングを指定 (utf-8, shift_jis, euc-jp, iso-2022-jp)
出力形式	出力形式を「JSON配列」もしくは「JSONオブジェクト」形式のいずれかを指定. cp.Outputの値で設定する場合は、"Array", "Object" のいずれか
開始行	CSVのデータとして読み込む開始行を指定
行数	何行読むかを指定。-1を指定した場合は全ての行を読み込む

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.10. CSV Write

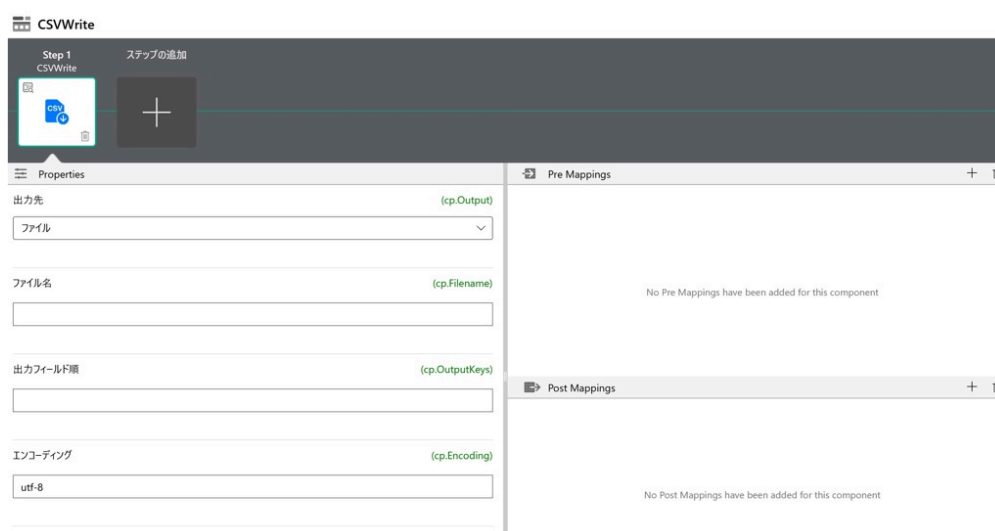
CSVWriteは入力ペイロードを入力としてCSVを生成します。

入力ペイロード(`cv.Payload`)はJSON配列の配列か、JSON Objectの配列での入力を受け付けます。それ以外の場合は文字列に変換して、1行、1カラムのCSVとして出力します。

入力ペイロード	例
JSON配列ペイロードの例	<code>[[123, "abc", "xxx"], [456, "def", "yyy"], [789, "hij", "zzz"]]</code>
JSON Objectのペイロードの例	<code>[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]</code>

「出力先」プロパティによって出力先をファイルかペイロードかを選択できます。

出力ペイロード	説明
「出力先」プロパティで「ファイル」を選んだ場合	生成したCSVをファイルに書き出し、出力ペイロードは入力ペイロードを渡します。(パススルー)
「出力先」プロパティで「ペイロード」を選んだ場合	生成したCSVのバイト列を出力ペイロードに入れます。



コンポーネントプロパティ

プロパティ名	説明
出力先	出力先を「ファイル」もしくは「ペイロード」のいずれかを指定. <code>cp.Output</code> に設定される値は <code>"Payload"</code> , <code>"File"</code> のいずれか

ファイル名	ファイル出力の場合のファイル名を指定
出力フィールド順	入力ペイロードがJSON Objectの場合、出力するフィールド名を出力させたい順に、(カンマ)で区切り列挙します
エンコーディング	ファイル出力のエンコーディングを指定 (utf-8, shift_jis, euc-jp, iso-2022-jp)

※ファイルパスを指定しないファイルの保存先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

出力フィールド順の指定方法

JSON Objectのペイロードが下記の場合に、出力フィールド順に field3, field1 と指定すると下記の出力ペイロード (CSV) が出力されます。

・JSON Objectのペイロード

```
[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]
```

・出力ペイロード (CSV)

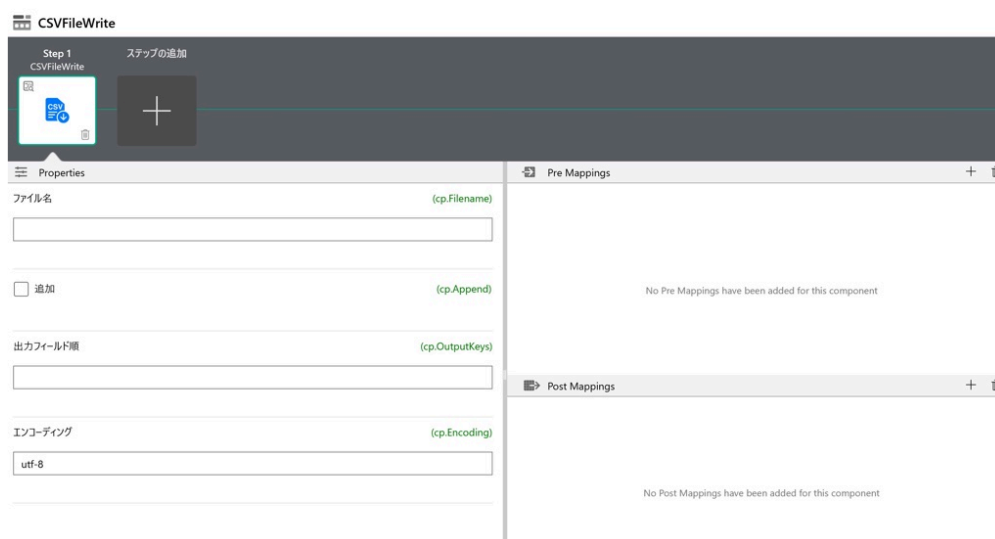
```
field3, field1
xxx, 123
yyy, 456
zzz, 789
```

7.4.9.11. CSV File Write

CSVFileWriteは入力ペイロードを入力としてCSVを生成します。

入力ペイロード(`cv.Payload`)を1レコード分と解釈して保存します。追加プロパティでアクションの実行毎に追記ができます。そのためヘッダ出力はされません。

入力ペイロード	例
JSON配列ペイロードの例	[123, "abc", "xxx"]
JSON Objectのペイロードの例	{"field1": 123, "field2": "abc", "field3": "xxx"}



コンポーネントプロパティ

プロパティ名	説明
ファイル名	ファイル出力の場合のファイル名を指定
追加	trueのときはすでにファイルがある場合はデータを追記します。falseの時はファイルを上書きします
出力フィールド順	入力ペイロードがJSON Objectの場合、出力するフィールド名を出力させたい順に、(カンマ)で区切り列挙します
エンコーディング	ファイル出力のエンコーディングを指定

※ファイルパスを指定しないファイルの保存先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

出力フィールド順の指定方法

JSON Objectのペイロードが下記の場合に、出力フィールド順に field3, field1 と指定すると下記の出力ペイロード (CSV) が出力されます。

・JSON Objectのペイロード

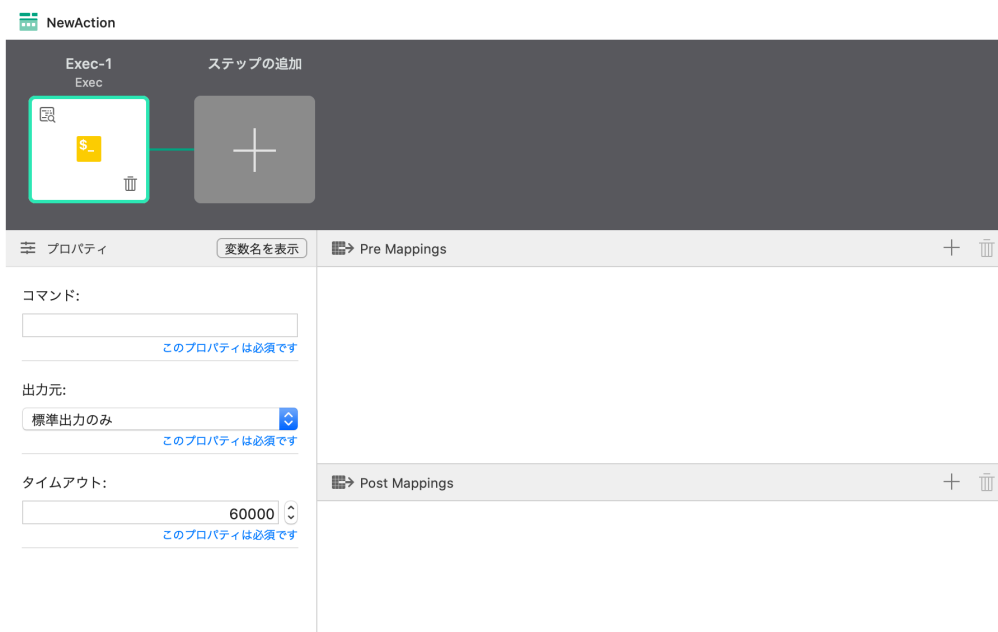
```
{"field1": 123, "field2":"abc", "field3":"xxx"}
```

・出力ペイロード(CSV)

```
xxx,123
```

7.4.9.12. Exec

Execコンポーネントを使用すると、OS標準のスクリプトを実行することができます。
HubKitがWindowsで実行されている場合はバッチファイルおよびコマンドラインで呼び出せるプログラム、Linux、Macで実行されている場合はシェルスクリプトおよびコマンドラインで呼び出せるプログラムを実行することができます。



コンポーネントプロパティ

プロパティ名	説明
コマンド	実行するプログラムを指定。ファイルはactmgr/dataに保存します
出力元	出力元を指定する。「標準出力のみ」はコマンドの実行で出力された標準出力を出力Payloadとします。「標準出力と標準エラー出力」は標準出力と標準エラー出力の両方を出力Payloadとします。cp.Outputに設定される値は "stdout" , "combine" のいずれか
タイムアウト	実行タイムアウト時間の指定
出力型	出力形式の設定. cp.OutputTypeで設定される値は "string" , "binary" のいずれか

コマンド

コマンド実行するプログラムを指定します。

・実行できるコマンド

Windows環境

バッチファイル、外部コマンドを実行します。外部コマンドはウィンドウ表示のないコンソールアプリケーションです。dirなどの内部コマンドを実行するには `C:¥WINDOWS¥system32¥cmd.exe /c dir` のようにcmd.exeのフルパスを指定してください。

バッチファイルを実行する場合はactmgr¥data にコピーして相対パスでパスを指定するか、任意のフォルダーにコピーして絶対パスを指定してください。

外部コマンドを実行する場合はフルパスを指定してください。

Mac、Linux、RaspberryPI、Gravio Hub

シェル /bin/sh を起動してシェルスクリプト、外部コマンドを実行します。

シェルスクリプトを実行する場合はactmgr/data にコピーして相対パスでパスを指定するか、任意のフォルダーにコピーして絶対パスを指定してください。

外部コマンドを実行する場合はフルパスを指定してください。

※シェルスクリプトの場合には、先頭行に `#!/bin/sh` を付加してください。

※シェルスクリプト、外部コマンドには実行権を付ける必要があります。

※Linux、RaspberryPI、Gravio HubではDocker環境で実行されているためホストOSにインストールされている外部コマンドは使用できません。

※Macではファイルをコピーしてフォルダーにアクセス権が必要です。アクセス権はシステム環境設定のセキュリティとプライバシーでプライバシーにあるフルディスクアクセスにGravio HubKit4.appを追加してください。

・実行するコマンドのパス

Windows環境

バッチファイル、外部コマンドを指定した場合には、パスにはactmgr¥dataが自動的に保管されます。パスの指定も可能です。

※フォルダー名、ファイル名にスペースが含まれている場合には"" (ダブルクォート) で囲む必要があります。

例)

```
actmgr¥data のbatchfile.bat を実行する場合は batchfile.bat をコマンドに指定します。  
actmgr¥data¥subfolder のbatchfile.bat を実行する場合は subfolder¥batchfile.bat をコマンドに指定します。  
actmgr¥data¥subfolder 2021 のbatchfile.bat を実行する場合は "subfolder 2021¥batchfile.bat" をコマンドに指定します。
```

絶対パスを指定することも可能です。

例)

c:¥ のbatchfile.bat を実行する場合は c:¥batchfile.bat をコマンドに指定します。

cmd.exeの内部コマンドを実行することも可能です。

例)

```
C:¥WINDOWS¥system32¥cmd.exe /c dir
```

Mac、Linux、RaspberryPI、Gravio Hub環境

シェルスクリプト、外部コマンドを指定した場合には、パスにはactmgr/dataが自動的に補完されます。パスの指定も可能です。

※フォルダー名、ファイル名にスペースが含まれている場合には""（ダブルクォート）で囲む必要があります。

例)

```
actmgr/data のshellscript.sh を実行する場合は shellscript.sh をコマンドに指定します。  
actmgr/data/subfolder のshellscript.sh を実行する場合は subfolder/shellscript.sh を  
コマンドに指定します。  
actmgr/data/subfolder 2021のshellscript.sh を実行する場合は "subfolder 2021/shells  
cript.sh" をコマンドに指定します。
```

絶対パスを指定することも可能です。

例)

```
/home/username のshellscript.sh を実行する場合は /home/username/shellscript.sh をコ  
マンドに指定します。
```

シェルの内部コマンドを実行することも可能です。

例)

```
ls /-lsa
```

7.4.9.13. Execute Inference Model

配置している画像推論ファイルを指定して写真に推論を実行することができます。

Step 1 InferenceModel ステップの追加

Properties

画像ファイル (cp.ImageFile)

このプロパティは必須です。

推論モデル (cp.ModelName)

このプロパティは必須です。

フィルター1 (cp.Filter1)

モデルのデフォルト

条件 (cp.Criterion1)

Pre Mappings

No Pre Mappings have been added

Post Mappings

No Post Mappings have been added

出力ペイロード	説明
cv.Payload	推論の実行結果が出力されます

コンポーネントプロパティ

プロパティ名	説明
画像ファイル	ファイルとして保存されている写真のファイルパスを指定します
推論モデル	実行する推論のモデル名を指定します
フィルター1	推論の結果をフィルターする場合に選択します
条件	フィルターの条件を指定します
フィルター2	推論の結果をフィルターする場合に選択します
条件	フィルター2の条件を指定します
変換	変換を指定します
描く	推論の結果を画像として保存する場合に指定します
コピーの保存先	描くで“コピー”を指定した場合の画像の保存先を指定します

フィルター

推論の結果から条件に合う結果のみを抽出する場合に使用します。

フィルター	説明
なし	フィルターを使用しません
モデルのデフォルト	モデルで定義されているフィルターを使用します
最も高いn個の予測	複数の予測の高いものから条件で指定したn個のデータを取得します
最も低いn個の予測	複数の予測の低いものから条件で指定したn個のデータを取得します
スコアがxを超える予測	複数の予測からスコアが条件で指定したx以上のデータを取得します
スコアがx未満の予測	複数の予測からスコアが条件で指定したx未満のデータを取得します
「○○」ラベルの付いた予測	複数の予測からラベルに条件で指定した文字列が含まれるデータを取得します

変換

推論の結果を変換する場合に使用します。

変換	説明
なし	変換を使用しません
モデルのデフォルト	モデルで定義されている変換を使用します
数	推論結果の個数を取得します
一番目のラベル	推論結果の一番最初のラベルを取得します
すべてのラベル	推論結果のすべてのラベルを取得します

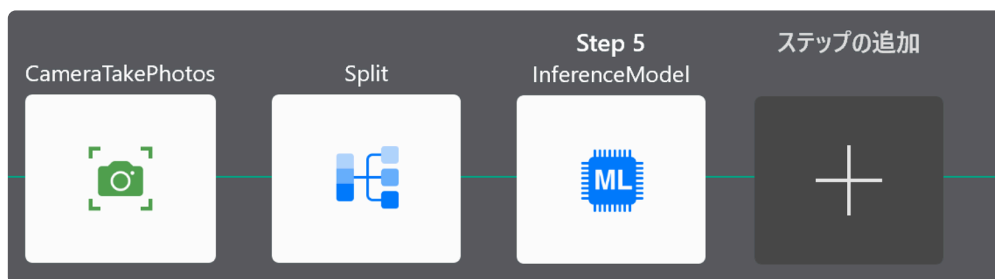
描く

推論の結果を画像として保存する場合に使用します。

描く	説明
なし	推論の結果の画像を保存しません
元画像	推論の結果を元の画像に保存します
コピー	コピーの保存先に別のファイルを作成して推論の結果を保存します

CameraTakesPhotosコンポーネントとの組み合わせでの使用

アクションの実行時にCameraTakesPhotosコンポーネントを使用して写真をファイルとして保存したものに推論を実行することが出来ます。



この例のようにCameraTakesPhotosコンポーネントとExecuteInferenceModelコンポーネントの間にSplitコンポーネントを配置します。

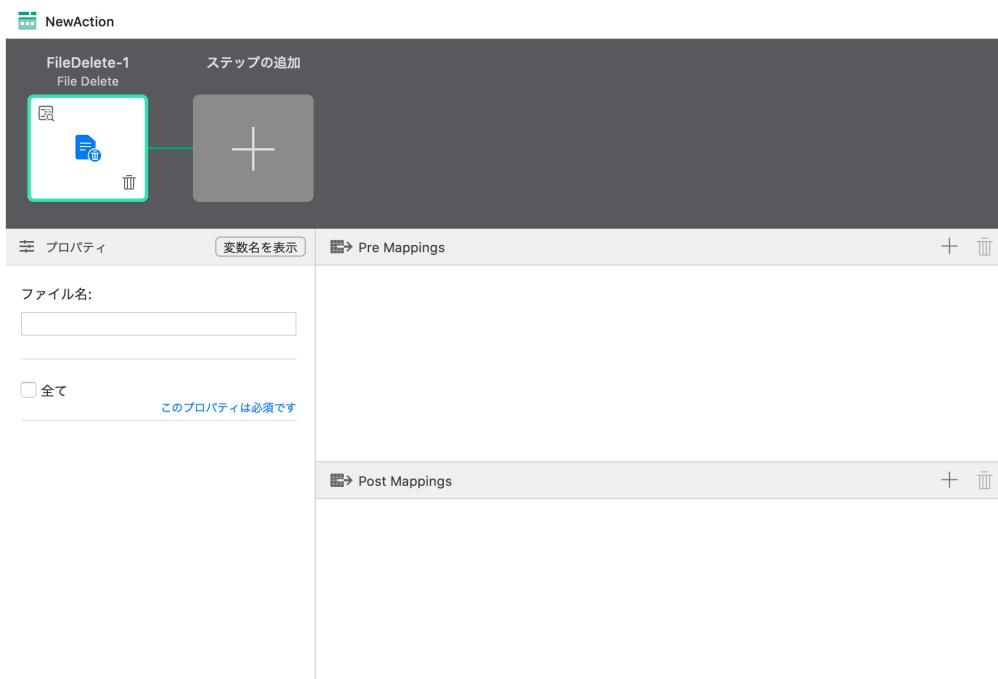
これはCameraTakesPhotosコンポーネントの出力が保存した写真のファイル名の配列ですので、その配列をSplitコンポーネントで分割して

1つずつのファイル名としてExecuteInferenceModelコンポーネントに渡しています。

そうすることでExecuteInferenceModelコンポーネントは1つずつ写真ファイルを読み込み推論を実行することが出来ます。

7.4.9.14. File Delete

File Delete コンポーネントは指定されたディレクトリにあるファイルを削除します。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

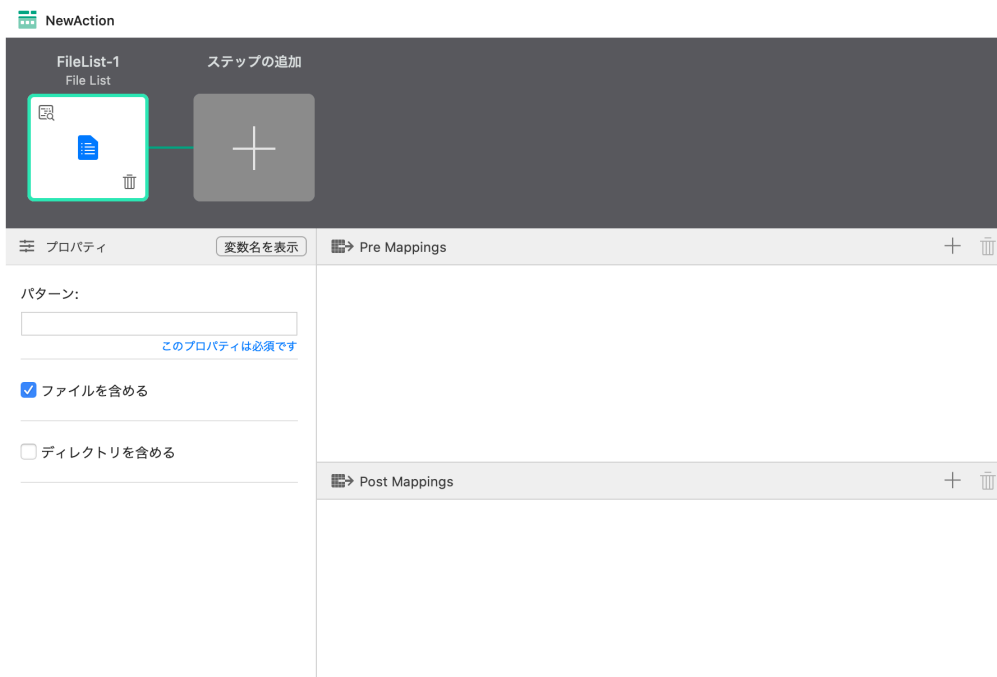
コンポーネントプロパティ

プロパティ名	説明
ファイル名	削除するファイル名を指定
全て	全て削除するかどうかの指定. cp.Allの値は "true" , "false" のいずれか

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.15. File List

File Listコンポーネントは指定したディレクトリにあるファイルの一覧を取得します。



出力ペイロード	説明
<code>cv.Payload</code>	読み込んだディレクトリの内容を出力します。

ディレクトリの内容は以下のデータを持ちます。

例)
パターンに“icons/*”を指定した場合の最初のファイルのデータ
{“IsDir”:false,“ModTime”:“2021-08-10T13:18:29.2577374+09:00”,“Name”:“arrow_down.bmp”,“Path”:“icons/arrow_down.bmp”,“ Size “:824}

データ名	説明
IsDir	ディレクトリの場合 true、ファイルの場合 false
ModTime	修正日時
Name	ファイル名
Path	ファイルパス
Size	ファイルサイズ

コンポーネントプロパティ

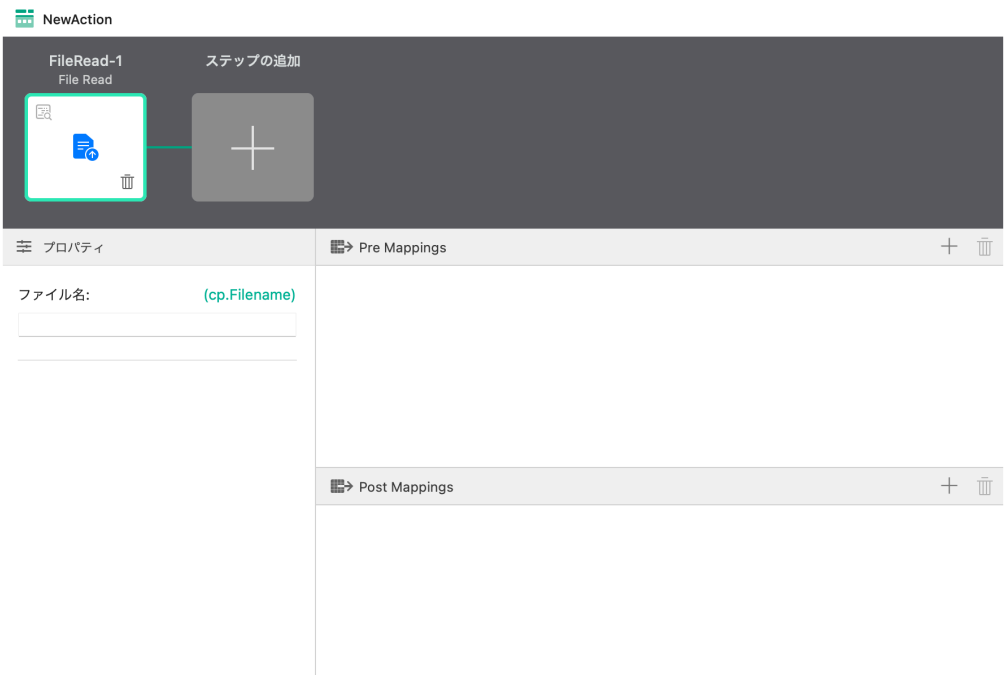
プロパティ名	説明
パターン	条件式を記述します

ファイルを含める	ファイルを含めるかどうかの指定
ディレクトリを含める	ディレクトリを含めるかどうかの指定

※ファイルパスを指定しないパターン参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.16. File Read

FileReadコンポーネントはFilenameプロパティで指定されたファイルを読み込んでストリームとして出力するコンポーネント



出力ペイロード	説明
cv.Payload	ファイルの内容

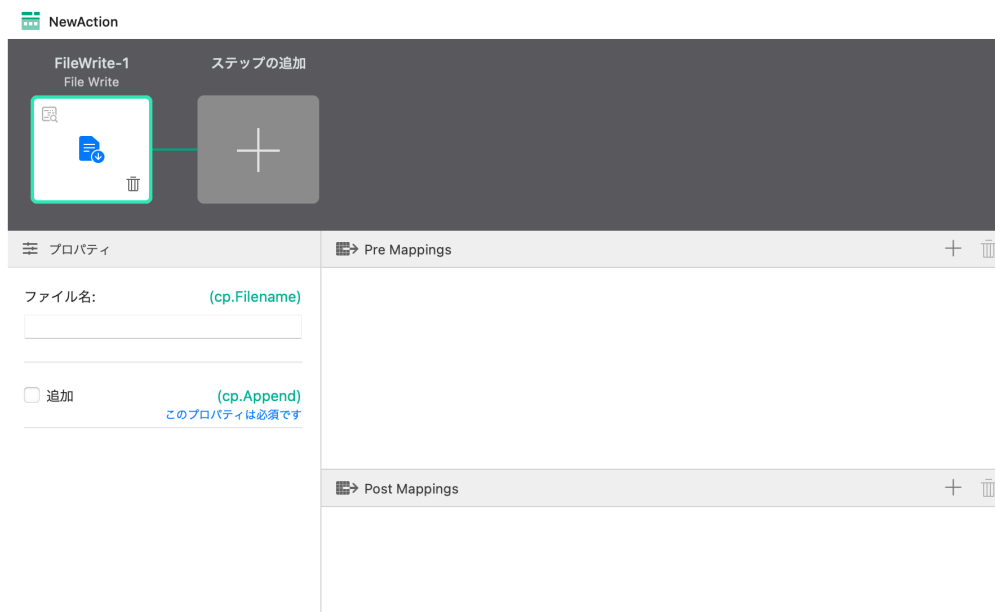
コンポーネントプロパティ

プロパティ名	説明
ファイル名	読み込むファイルのファイル名

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.17. File Write

File Writeコンポーネントは入力ペイロード(`cv.Payload`)に渡された内容をファイルに書き出します。



ファイルは以下のディレクトリに書き出されます。

Windows: C:\ProgramData\HubKit\action\actmgr\data\

Mac: /Library/Application Support/HubKit/action/scripts/actmgr/data/

Linux: /var/opt/hubkit/action/scripts/actmgr\data\

出力ペイロード	説明
<code>cv.Payload</code>	書き出したファイルの内容

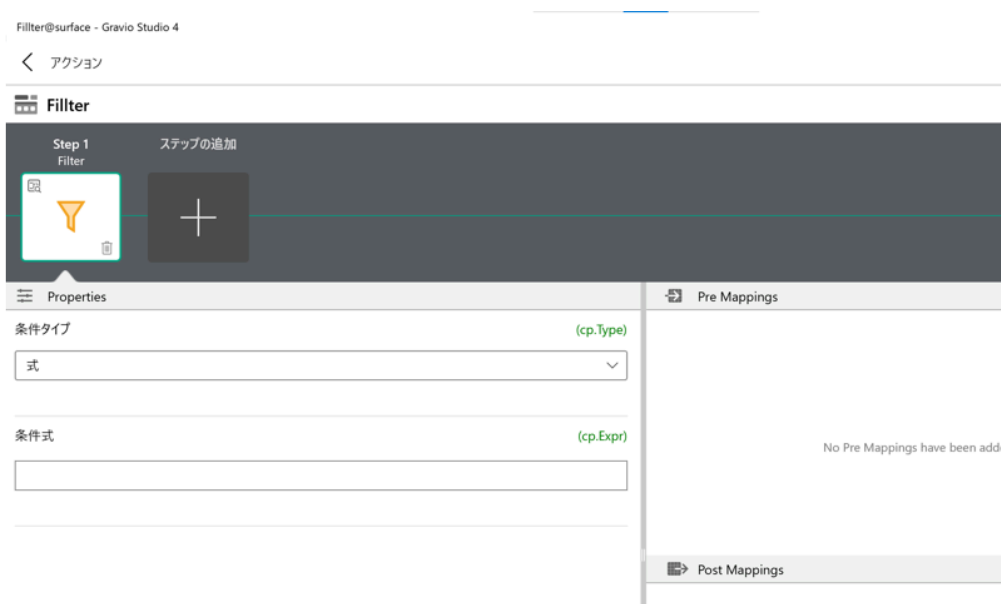
コンポーネントプロパティ

プロパティ名	説明
ファイル名	書き出すファイルのファイル名
追加	trueのときはすでにファイルがある場合はデータを追記します。falseの時はファイルを上書きします

※ファイルパスを指定しないファイルの保存先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.18. Filter

Filterコンポーネントは入力ペイロード(`cv.Payload`)の内容を次のコンポーネントに引き渡すデータをフィルタリングして出力ペイロード(`cv.Payload`)に出力するコンポーネントです。



Actionの処理を通過させるかの条件を式で入力することができます。`true`の時は通過させ、`false`の時は通過しません。通過しない場合はActionは終了します。

変数として、`tp.` / `tv.` / `ap.` / `av.` / `cp.` / `cv.` のプリフィックスを持つ変数を比較対象として条件式を記述します。

コンポーネントプロパティ

プロパティ名	説明
条件タイプ	式、最初のPayloadのみ、最後のPayloadのみ
条件式	条件タイプで“式”を選択した場合、条件式を記述

条件タイプが“式“の場合

条件例	式例
Payloadを対象としPayloadが数値系	<code>cv.Payload > 10</code>
文字列系	<code>cv.Payload =~ "^abc"</code>
JSON配列	<code>cv.Payload[1] > 10</code>
JSON Object	<code>cv.Payload.key1 == "abc"</code>

式で利用できる、コンポーネントローカル変数

コンポーネントローカル変数	説明
Index	コンポーネントに届いたPayloadの順を0を基準として表し、式を <code>Index % 2 == 0</code> とすると偶数番目に届いたPayloadのみ通過させる式表現となります
Passed	このコンポーネントを通過したPayloadの数を表し、式を <code>Passed < 30</code> とすると最初から30個までは通過し、それ以降は捨てられる式表現となります

条件タイプが“式“以外の場合

条件タイプを“最初のPayloadのみ“を選択した場合、Payloadの最初だけがフィルタリングされて出力ペイロード(`c.v.Payload`)に出力されます。

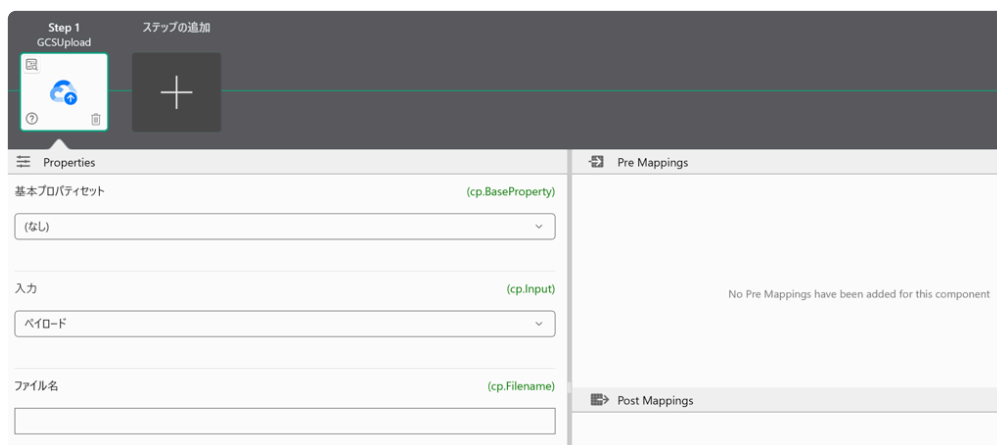
条件タイプを“最後のPayloadのみ“を選択した場合、Payloadの最後だけがフィルタリングされて出力ペイロード(`c.v.Payload`)に出力されます。

7.4.9.19. GCS Upload

GCS Uploadコンポーネントは、データをGoogle Cloud Storageにファイルとして送信することができます。このコンポーネントを使用するには、Google Cloudの契約と事前設定が必要です。

※Google Cloud Storageへアクセスするためのロールは「Storage レガシー バケット オーナー」と「Storage レガシー オブジェクト オーナー」を追加する必要があります。

事前設定は [こちら](#) を参照してください。



コンポーネントプロパティ

プロパティ名	説明
入力	ペイロードの内容をアップロードするか、ファイル名を指定してそれをアップロードするか選択します
ファイル名	ファイル名を選択した時に、アップロードするファイルをDataディレクトリからの相対パスで指定します
バケット	アップロードする先のバケット名を指定します
オブジェクト	アップロードする先のオブジェクト名を指定します。フォルダ込で指定する時はスラッシュで区切って指定してください。プロパティが空だった場合で、アップロード元がファイルの場合は、ディレクトリ名を含まないファイル名部分だけがファイル名に使用されます
ACL	アクセス制御リストを指定します*1
ストレージクラス	ストレージクラスを指定します*2
ContentType	オブジェクトのContentTypeを指定します。空の場合は自動的に付加されます
タイムアウト	APIのタイムアウト
秘密鍵JSON	Google Cloud で作成したサービスアカウントの認証に必要な秘密鍵JSONファイルを指定します

※1 ACLは、どのユーザ(IAMやその他のユーザ)にOwner/Reader権を与えるというように、リスト形式で複数のユ

一ザ一与えることで実現できます。

この形式はGravio Studioでは設定出来ませんので、[定義済のACL](#) を使用します。

ACLで(デフォルト)を選択するとACLは指定されず、バケットのデフォルトが使用されます。

また、バケットはデフォルトではACLは推奨されていないため指定できるようにはなっていません。

バケットでACLを使用する場合には、バケットを作成する時に「均一」と「きめ細かい管理」と選択する画面で、「きめ細かい管理」を選択する必要があります。

「均一」が選択されているバケットでACLを(デフォルト)以外を選択するとコンポーネントの実行時にエラーになります。

※2 (デフォルト)を選ぶと、バケットのデフォルトが使用されます。

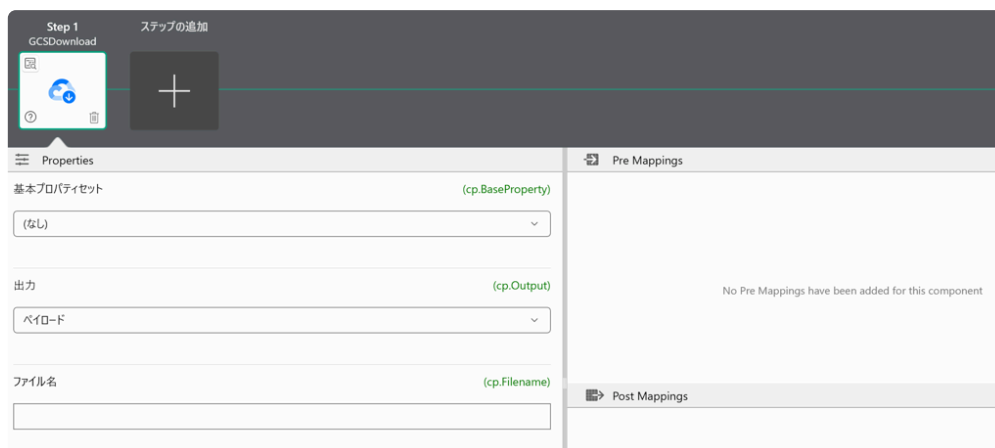
7.4.9.20. GCS Download

GCS Downloadコンポーネントは、データをGoogle Cloud Storageからファイルとして受信することができます。このコンポーネントを使用するには、Google Cloudの契約と事前設定が必要です。

※Google Cloud Storageへアクセスするためのロールは「Storage レガシー バケット読み取り」と「Storage レガシー オブジェクト 読み取り」を追加する必要があります。

※GCS Uploadコンポーネントも使用する場合には、「Storage レガシー バケット オーナー」と「Storage レガシー オブジェクト オーナー」を追加してください。

事前設定は [こちら](#) を参照してください。

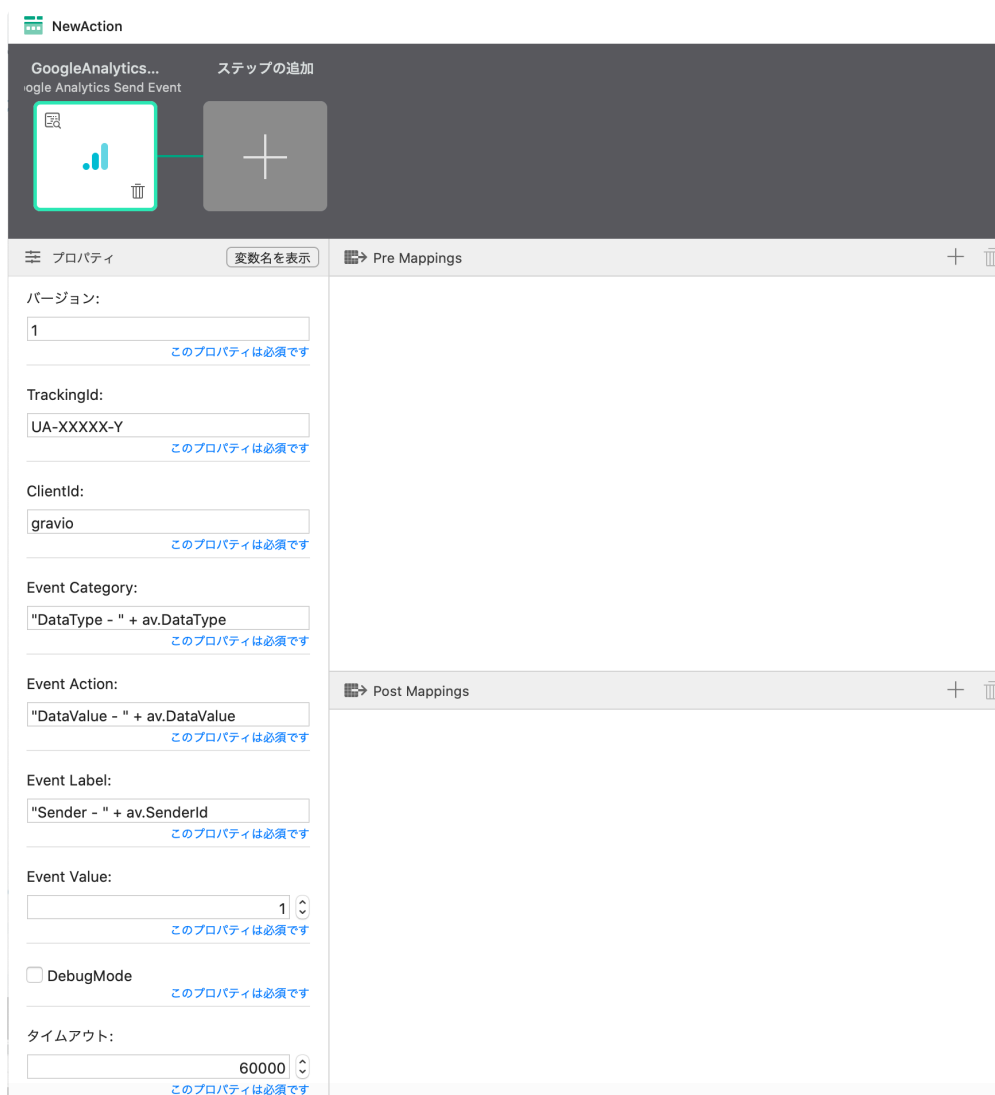


コンポーネントプロパティ

プロパティ名	説明
出力	ダウンロードしたオブジェクトをペイロードに出力するか、ファイルに保存するか指定します
ファイル名	ファイル名を選択した時に、ダウンロードするファイルをDataディレクトリからの相対パスで指定します
バケット	ダウンロードするバケット名を指定します
オブジェクト	ダウンロードするオブジェクト名を指定します
タイムアウト	APIのタイムアウト
秘密鍵JSON	Google Cloud で作成したサービスアカウントの認証に必要な秘密鍵JSONファイルを指定します

7.4.9.21. Google Analytics Send Event

GoogleAnalyticsSendEventコンポーネントを使用すると、GoogleアナリティクスのMeasurement Protocolを使用してイベントを送信します。



The screenshot shows the 'NewAction' configuration window for the 'Google Analytics Send Event' component. The interface is divided into several sections:

- Properties (プロパティ):** This section contains several required fields:
 - バージョン (Version):** Set to '1'.
 - TrackingId:** Set to 'UA-XXXXX-Y'.
 - ClientId:** Set to 'gravio'.
 - Event Category:** Set to '"DataType - " + av.DataType'.
 - Event Action:** Set to '"DataValue - " + av.DataValue'.
 - Event Label:** Set to '"Sender - " + av.SenderId'.
 - Event Value:** Set to '1'.
 - Debug Mode:** A checkbox that is currently unchecked.
 - タイムアウト (Timeout):** Set to '60000'.
- Mappings (マッピング):** This section includes 'Pre Mappings' and 'Post Mappings'.

Measurement Protocolやパラメータの詳細は[こちら](#)を参考にしてください。

Google Analyticsでプロパティを作成するには、「プロパティの作成」から「ウェブ」を選択してください。画面上に表示される「トラッキング ID」をコンポーネントの“Tracking ID”プロパティに設定してください。

“Debug Mode”をチェックするとデバッグモードになります。デバッグモードの場合、“Tracking ID”に不備がある場合などはエラーが表示されますが、Google Analyticsにはデータが送信されないためイベントとしてはカウントされません。逆にデバッグモードでない場合は、パラメータに不備があってもエラーにはなりませんので気をつけてください。

出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明	例
バージョン	Analyticsのバージョン番号	例) 1
TrackingId	AnalyticsのトラッキングID	例) UA-XXXXX-Y
ClientId	クライアントID	記録するクライアントでユニークなIDを指定します
Event Category	イベントのカテゴリ	例) "DataType - " + av.DataType
Event Action	イベントのアクション	例) "DataValue - " + av.DataValue
Event Label	イベントのラベル	例) "Sender - " + av.SenderId
Event Value	送信するデータ	例) 1
DebugMode	デバッグモード	設定内容にエラーが出るときに使用します
タイムアウト	APIのタイムアウト	通常は変更する必要はありません

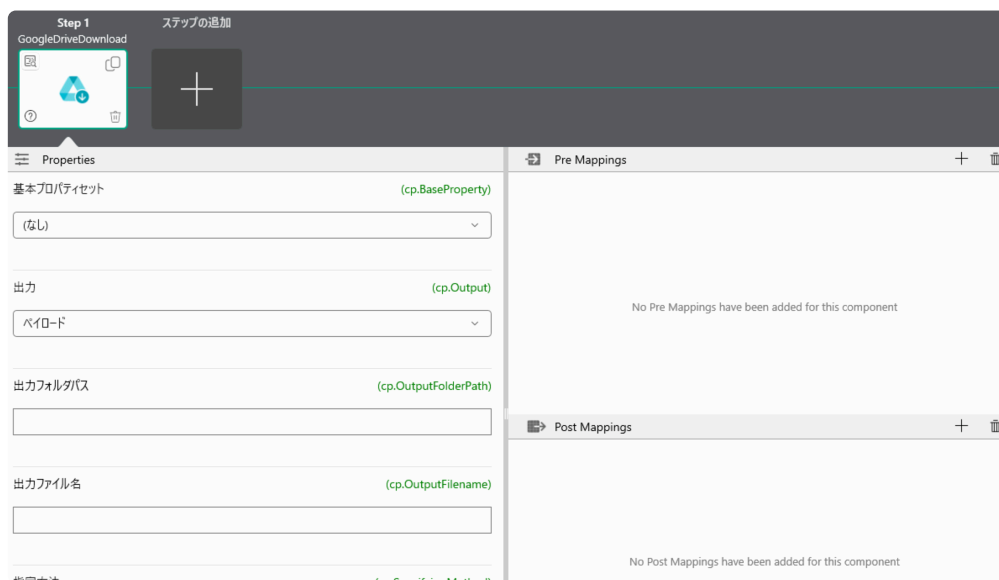
7.4.9.22. GoogleDriveDownload

Google Drive Downloadコンポーネントは、Google Driveからファイルをダウンロードすることができます。
次のようなことができます。

Google Driveの特定フォルダや特定ファイル名で保存したファイルを定期的に**Gravio**で取得して処理する

- ・特定フォルダなどに日付付きのファイル名で置かれたCSVなどの売上データや処理ログをGoogle DriveからダウンロードしてGravioで処理
フォルダ名は複数階層のフォルダにも対応しています。

- ・モバイル端末で保存した音声データをGoogle Driveにアップロードし、Gravioが定期的にそのファイルをダウンロードしてOpenAIAudioコンポーネントでテキスト化の処理
同様に画像ファイルも取得して、社内に送信することにも対応しています。



出力ペイロード	説明
cv.Payload	出力プロパティでペイロードが指定されている場合はダウンロードしたファイル。ファイル名が指定されている場合は前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
出力	ペイロードまたはファイル名
出力フォルダパス	「入力プロパティ」がファイル名の場合、保存するフォルダパス
出力ファイル名	「入力プロパティ」がファイル名の場合、保存するファイル名
指定方法	ファイル名を指定して検索、ファイルパスとファイル名を指定して検索、フォルダIDとファイル名を指定して検索、ファイルIDを指定して検索

1番新しいものだけをダウンロード	同一ファイル名で複数の日付で保存されている場合に、新しいファイルを1つだけダウンロードするか全てをダウンロードするか
取得対象	「指定方法プロパティ」にファイルID以外を指定している場合、完全一致、部分一致でDriveファイル名を指定します
フォルダID	「指定方法プロパティ」にフォルダIDを指定している場合、ダウンロード対象のフォルダID
Driveフォルダパス	「指定方法プロパティ」にフォルダパスを指定している場合、ダウンロード対象のDriveフォルダパス
Driveファイル名	「指定方法プロパティ」にファイル名を指定している場合、ダウンロードするDriveファイル名
ファイルID	「指定方法プロパティ」にファイルIDを指定している場合、ダウンロードするファイルID
この時間以降に変更されたファイルを取得	ダウンロードするファイルの更新日時の指定
タイムアウト	HTTPリクエストのレスポンスタイムアウト値
OAuth2クライアント ID	GCPからAPI 2.0のClient IDを指定します ※1
OAuth2クライアントシークレット	GCPからClient Secretを指定します ※1
OAuth2リダイレクトURL	GCPからRedirect URLを指定します ※1
OAuth2アクセストークン	トークンを取得ボタンを押して取得します。またトークンを更新ボタンを押して更新することもできます ※1 ※2

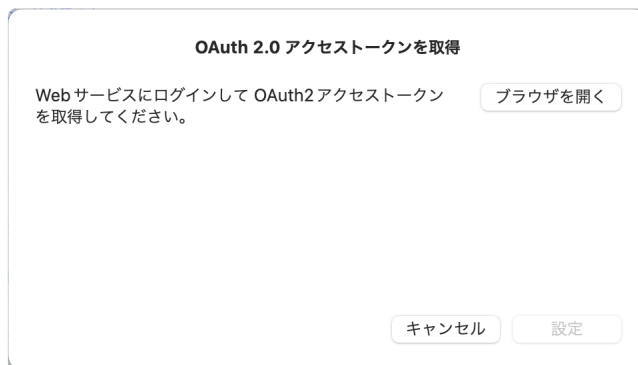
※1 GCPでのOAuth2の設定例は [こちら](#) を参照してください

コンポーネント変数

コンポーネント変数	説明
cv. FileID	ファイルID
cv. ReadLength	読込データのバイト数
cv. Filename	「出力」プロパティがファイル名の場合、ダウンロードしたファイル名

OAuth2アクセストークンの取得方法

※2 OAuth2アクセストークンを取得するボタンを押すとOAuth 2.0 アクセストークンを取得 パネルが表示されます。ブラウザを開く ボタンを押してトークンを取得してください。トークンを取得すると 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



Gravio Studio for Windowsの場合も同様にブラウザを開く ボタンを押してトークンを取得しますが、エラー画面となります。この状態でURLをアドレスバーからコピーして、そのURLをパネルにペースとして取得ボタンを押してください。トークンが取得されますので 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



ブラウザを開くボタンを押すと自動的にブラウザで認証が行われます。必要がありましたらログイン処理など画面の指示に合わせて操作してください。



認証が完了すると「このページに到達できません」とエラーが表示されるので、この時にURLをアドレスバーからコピーしてください。



コピーしたURLをパネルにペーストして取得ボタンを押すとOAuth2アクセストークンが“取得済み”となります。設定ボタンを押してパネルを閉じます。

7.4.9.23. GoogleDriveUpload

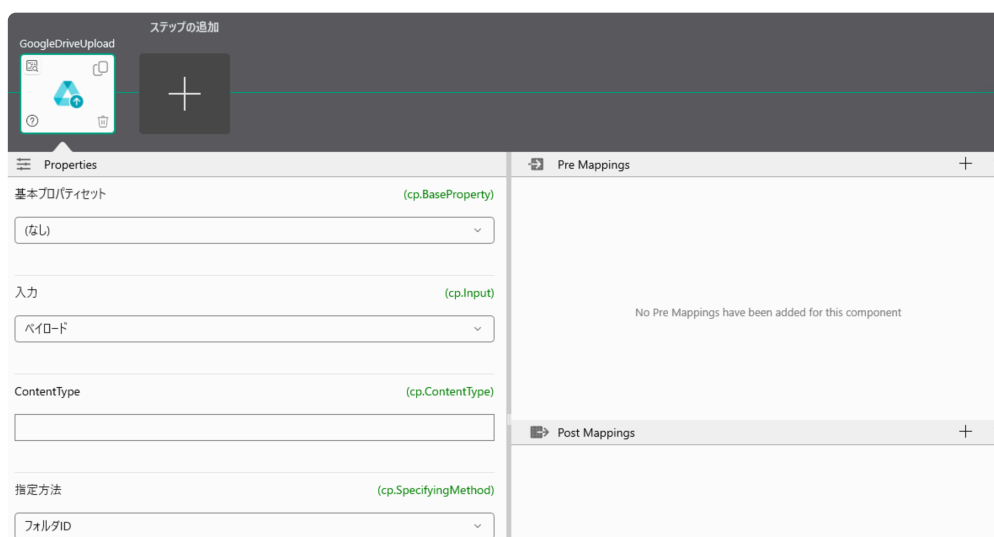
Google Drive Uploadコンポーネントは、Google Driveにファイルをアップロードすることができます。
次のようなことができます。

Gravioで処理した結果データをGoogle Driveに転送する

- ・定期的にデバイスのデータをログデータとしてGoogle Driveにアップロードして保存
- CSV、Excel形式、カメラから取得した画像データをGoogle Driveに転送にも対応しています。
- ・ボタンや人感などのセンサーをトリガーに画像を撮影して、画像データをGoogle Driveにアップロードして保存

Gravioで取得したRDBやCSVのデータをExcelなどに加工してGoogle Driveに転送する

- ・売上データやセンサーデータの定期報告用途でExcelファイルを作成して、社内で使用出来るようにGoogle Driveにアップロードして保存



出力ペイロード	説明
<code>cv.Payload</code>	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
入力	ペイロードまたはファイル名
ファイル名	「入力プロパティ」がファイル名の場合に送信するファイル名
Content-Type	HTTPヘッダのContent-Type
指定方法	フォルダパスまたはフォルダID
Driveフォルダパス	Google Driveの保存先フォルダ名
ファイルID	Google Driveの保存先フォルダID

タイムアウト	HTTPリクエストのレスポンスタイムアウト値
OAuth2クライアント ID	GCPからAPI 2.0のClient IDを指定します ※1
OAuth2クライアントシークレット	GCPからClient Secretを指定します ※1
OAuth2リダイレクトURL	GCPからRedirect URLを指定します ※1
OAuth2アクセストークン	トークンを取得ボタンを押して取得します。またトークンを更新ボタンを押して更新することもできます ※1 ※2

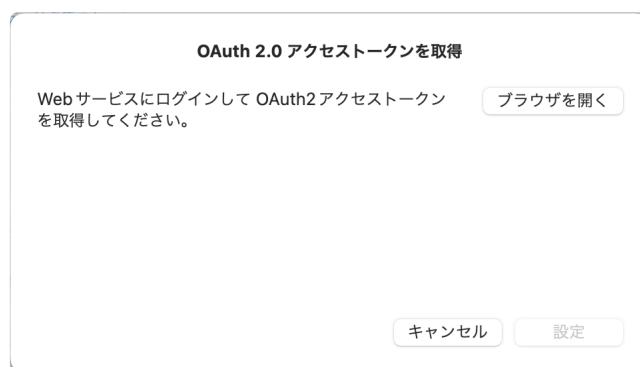
※1 GCPでのOAuth2の設定例は [こちら](#) を参照してください

コンポーネント変数

コンポーネント変数	説明
cv. ContentType	HTTPヘッダのContent-Type
cv. FileID	ファイルID
cv. FolderID	フォルダID
cv. DriveFilename	Google Driveの保存先ファイル名

OAuth2アクセストークンの取得方法

※2 OAuth2アクセストークンを取得するボタンを押すとOAuth 2.0 アクセストークンを取得 パネルが表示されます。ブラウザを開く ボタンを押してトークンを取得してください。トークンを取得すると 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



Gravio Studio for Windowsの場合も同様にブラウザを開く ボタンを押してトークンを取得しますが、エラー画面となります。この状態でURLをアドレスバーからコピーして、そのURLをパネルにペーストとして取得ボタンを押してください。トークンが取得されますので 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



ブラウザを開くボタンを押すと自動的にブラウザで認証が行われます。必要がありましたらログイン処理など画面の指示に合わせて操作してください。



認証が完了すると「このページに到達できません」とエラーが表示されるので、この時にURLをアドレスバーからコピーしてください。



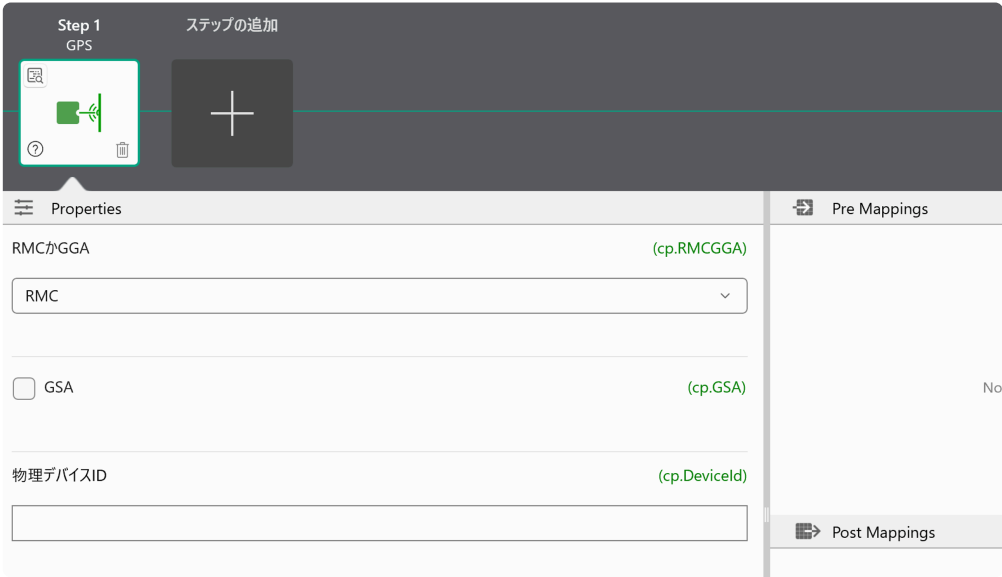
The screenshot shows a dialog box titled "OAuth 2.0のアクセストークンの取得" (Acquisition of OAuth 2.0 Access Token). It contains the following elements:

- Instruction: "OAuth2のAccess Tokenを取得するため、Webサービスにログインしてください。" (To acquire the OAuth2 Access Token, please log in to the web service.)
- Link: A button labeled "ブラウザを開く" (Open browser).
- Label: "リダイレクトされるURLをここに貼り付けてください:" (Please paste the URL you will be redirected to here:)
- Text Input: A text box containing the URL "ZeqT03xtJWmkNyw&session_state=b22da329-0afe-4bdb-80ce-34dc870f470a".
- Action: A button labeled "取得" (Acquire).
- Status: Below the "取得" button, the text "OAuth2アクセストークン:" is on the left, and "取得済み" (Acquired) is on the right.
- Buttons: At the bottom, there are two buttons: "設定" (Settings) and "キャンセル" (Cancel).

コピーしたURLをパネルにペーストして取得ボタンを押すとOAuth2アクセストークンが“取得済み”となります。設定ボタンを押してパネルを閉じます。

7.4.9.24. GPS

GPSデバイスからGPSデータを受信して出力することができます。



出力ペイロード	説明
<code>cv.Payload</code>	以下のデータが出力されます

・データがRMCの場合

データ	説明
Time	GPSデータのUTC時刻(hhmmss.ss形式)
Status	A. データは正常 V. データは無効
Latitude	緯度(DMM形式)
LatitudeDir	緯度の方位(NまたはS)
Longitude	経度(DMM形式)
LongitudeDir	経度の方位(EまたはW)
SpeedKn	デバイスの移動速度(knots)
Course	デバイスの移動方位(北 0度から時計回りで359.9度までの数値)-1の場合は方位は無効
Date	GPSデータの年月日(ddmmyy形式)
MagneticVarDeg	磁北と真北の間の角度の差(0度から359.9度)
MagneticVarDir	磁北と真北の間の角度の差の方向(EまたはW)
Mode	GPSモード(N. データなし, A. 自律方式, D. 干渉測位方式, E. 推定)

・データがGGAの場合

データ	説明
Utc	GPSデータの受信時刻(hhmmss.ss形式)
Latitude	緯度(DMM形式)
LatitudeDir	緯度の方位(NまたはS)
Longitude	経度(DMM形式)
LongitudeDir	経度の方位(EまたはW)
FixQuality	位置特定品質(0. 位置特定できない、1. 標準測位サービスモード、2. 干渉測位方式モード)
SatellitesInUse	使用衛星数
HDOP	水平精度低下率
Altitude	アンテナの海拔高さ(単位 M)
Undulation	ジオイド高さ(単位 M)
Age	最後に補正情報を受信してからの経過時間
StnID	ステーションID

・データにGSAが含まれる場合

Mode	モード(M. 手動、A. 自動)
FixType	特定タイプ(1.非サポート 2. 2D, 3. 3D)
SatellitesList	衛星番号
PDOP	位置精度低下率
HDOP	水平精度低下率
VDOP	垂直精度低下率

コンポーネントプロパティ

プロパティ名	説明
RMC か GGA	使用しているGPSデバイスのデータの出力形式をRMCかGGAで選択します
GSA	使用しているGPSデバイスの出力データにGSAが含まれている場合に指定します
デバイ	GPSデバイスID, 何も設定されていないときは全てのGPSデバイスからのデータ出力を実行します。GPS

スID	デバイスを複数使用する場合は、デバイスIDをカンマ区切りで記述することができます
-----	--

用語

・RMS、GGA、GSA

GPSから出力されるNMEAフォーマットのセンテンスの種類のことです。
GPSデバイスによってサポートされているセンテンスが異なります。
センテンスは他にもGSV、VTGがあります。

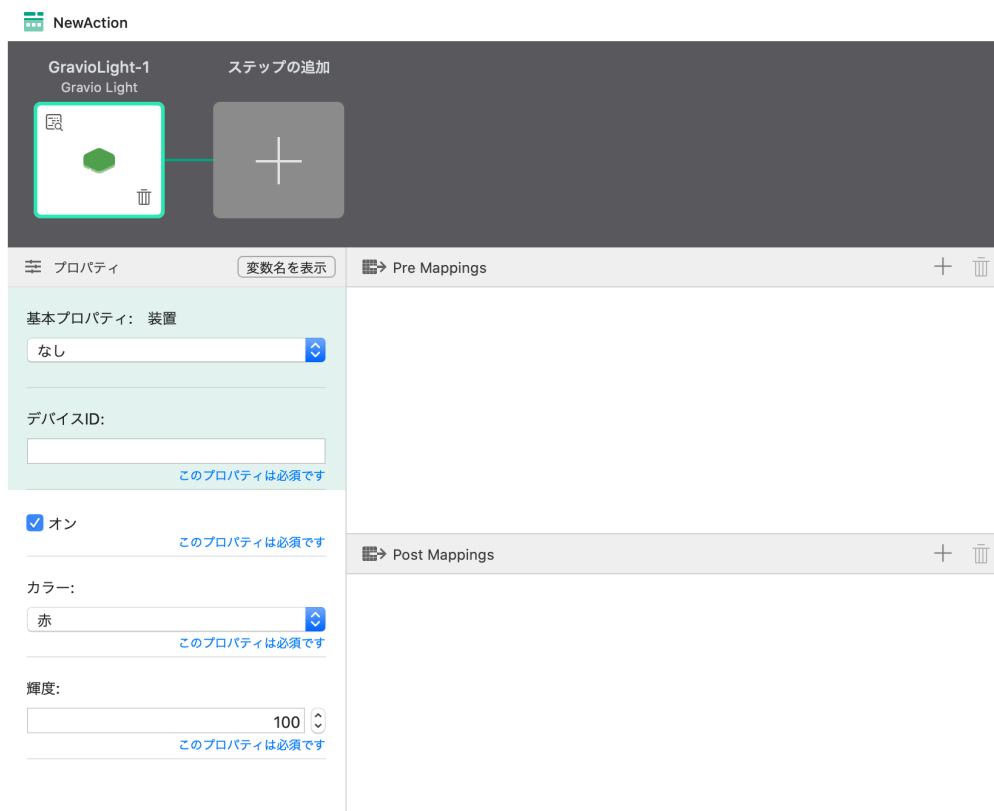
・DMM形式

dddmm.mmmmのデータ形式で、dddは度数、mm.mmmmは分数で、60進法ですので60分で1度となります。
Googleマップで使用されている ddd.dddd度表記は、(度数 + 分数/60) で計算することができます。

7.4.9.25. Gravio Light

Gravio Lightコンポーネントは、Gravio LightのOn/Offならびに色と明るさの変更ができます。
このコンポーネントを利用するにはあらかじめGravio Lightをペアリングしておく必要があります。

デバイスIDのプロパティが空の場合は、全ての接続しているGravio Lightにコマンドを送ります。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
基本プロパティ	「Gravio Light」カテゴリーの基本プロパティを指定します
デバイスID	On/OffするデバイスID, 何も設定されていないときは全てのGravio Lightに対してON/OFFを実行します。 Gravio Lightを複数使用する場合は、デバイスIDをカンマ区切りで記述することができます
オン	On/Offの指定 cp.Onに指定する値は "true" または "false"
カラー	色の指定 (赤,黄,緑,青から選択) cp.Colorに設定する場合の値は "red" , "yellow" , "green" , "blue" のいずれかの文字列を設定します

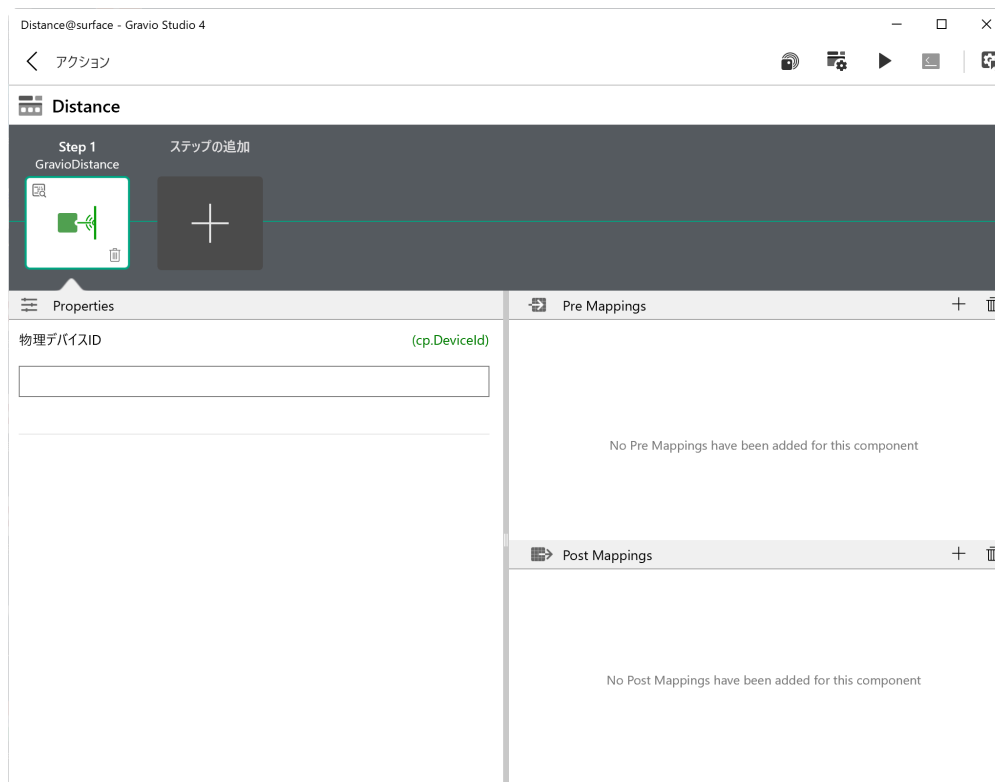
輝度

明るさの選択 (0～100から選択)

7.4.9.26. Gravio Distance

Gravio Distanceコンポーネントは、Gravio Distanceから距離を測定した値を取得できます。
このコンポーネントを利用するにはあらかじめGravio Distanceをペアリングしておく必要があります。

デバイスIDのプロパティが空の場合は、全ての接続しているGravio Distanceにコマンドを送ります。



出力ペイロード	説明
<code>cv.Payload</code>	測定した距離が出力されます。物理IDを指定した場合や対象デバイスが1台の場合は値で、複数デバイスの場合は配列となります

コンポーネントプロパティ

プロパティ名	説明
デバイスID	距離を測定するデバイスID, 何も設定されていないときは全てのGravio Distanceに対して距離の測定を実行します。Gravio Distanceを複数使用する場合は、デバイスIDをカンマ区切りで記述することができます

実行エラー

エラーコード	エラー	説明	対策
ERR—10	low power	電流不足	電源とセンサー間のUSBケーブルが正しく接続されていることを確認してください
ERR—14	calculation error	実行エラー	もう一度実行してください
ERR—15	weak signal	弱い信号	デバイスを動かさないようにして、障害物を取り除いてください
ERR—16	unstable laser transition	レーザーが不安定	デバイスとターゲットオブジェクトを静止させてください
ERR—18	ambient light too strong	周囲光が強すぎる	周囲光を減らしてください
ERR—26	out of display range	表示範囲外	もう一度実行してください

レイヤーにバインドされているときに発生するエラーについて

デバイスのレイヤーにバインドして有効設定されているDistanceセンサーをGravio Distanceコンポーネントで指定して使用する場合に下記のエラーが発生します。

✳ **ERROR 400 action: “GravioLaser”: failed to execute: component: GravioDistance: device is being used**

レイヤーにバインドして有効設定されたDistanceセンサーは5秒ごとに距離データを送信する設定となり、その送信時にGravio Distanceコンポーネントが実行されると処理が衝突するためです。

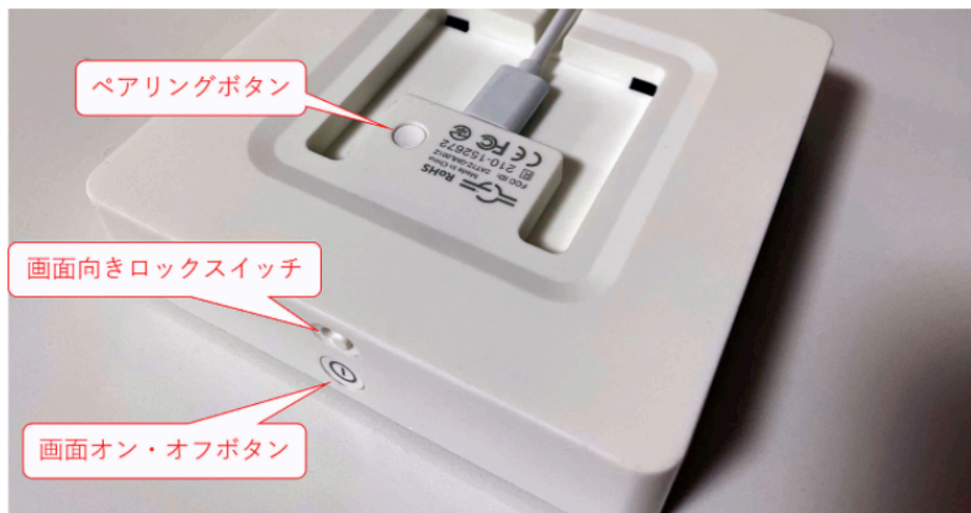
レイヤーにバインドして有効設定されたDistanceセンサーはGravio Distanceコンポーネントでは使用できませんので、どちらかの処理でのみ使用してください。

7.4.9.27. Gravio Matrix

Gravio Matrixコンポーネントは、Gravio Matrixに文字または画像を設定できます。
このコンポーネントを利用するにはあらかじめGravio Matrixをペアリングしておく必要があります。

ペアリング

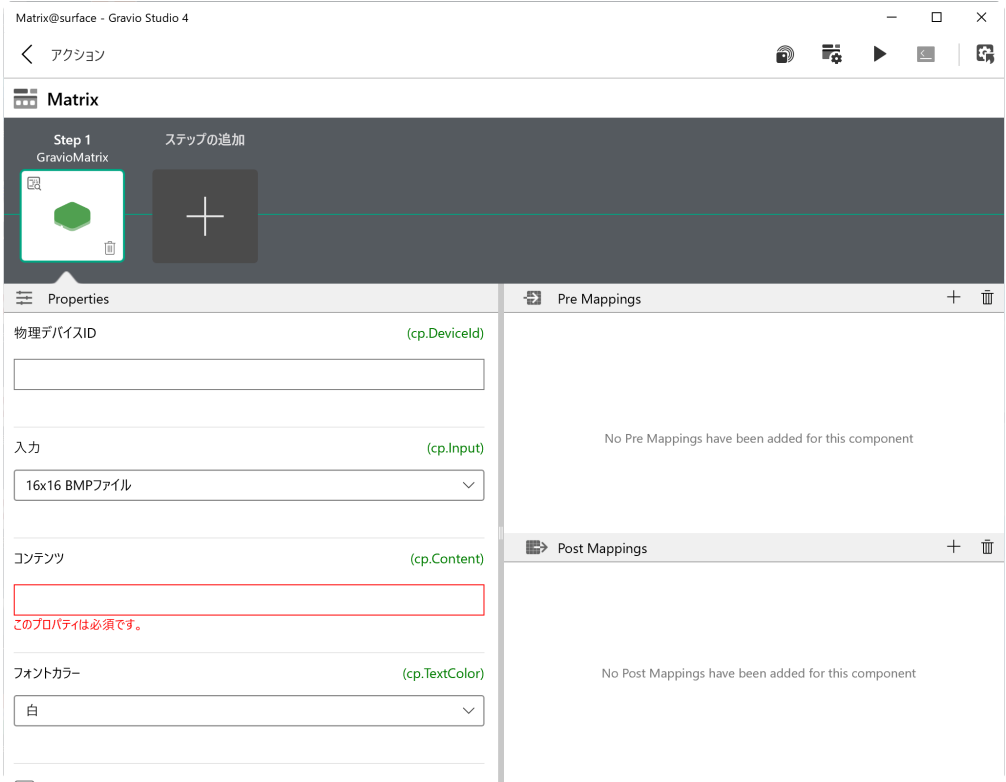
ペアリングは背面カバーを外して背面のペアリングボタンを押します。



また側面には画面回転スイッチと画面表示オン・オフのボタンがあります。画面回転スイッチをオンにするとMatrixが画面を縦向き/横向きにかかわらず固定されます。

使用方法

デバイスIDのプロパティが空の場合は、全ての接続しているGravio Matrixにコマンドを送ります。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
デバイスID	文字または画像を設定、On/OffするデバイスID, 何も設定されていないときは全てのGravio Matrixに対して実行します。Gravio Matrixを複数使用する場合は、デバイスIDをカンマ区切りで記述することができます
入力	"16×16 BMPファイル"、"ASCII" またはテンプレートを設定します。テンプレートは "温度℃", "温度°F", "湿度%", "気圧hPa", "CO2濃度", "開/閉", "日付", "時間" となります
コンテンツ	入力が“16×16 BMPファイル“の時は actmgr/data/icons にBMPファイルを保存し、「コンテンツ」プロパティには" icons/BMPファイル名 "と指定します。"ASCII" の時はアルファベットで1文字または2文字を指定します
フォントカラー	色の指定(白,黒,赤,緑,青,自動から選択)cp.Colorに設定する場合の値は "white", "black", "red", "green", "blue", "auto" のいずれかの文字列を設定します
オン	On/Offの指定 cp.Onに指定する値は "true" または "false"
輝度	明るさの選択 (0～100から選択)

テンプレートを使用した表示

コンポーネントプロパティの入力でテンプレートを選択するとアイコン画像と合わせて簡単にデータを表示することができます。

テンプレートは以下の中から選択することができ、コンテンツに値を指定するとテンプレートに合わせた画像と値がマトリクスに表示されます。

テンプレート	説明
温度℃	温度（摂氏）センサーの値を表示します。コンテンツには数値を入力します
温度°F	温度（華氏）センサーの値を表示します。コンテンツには数値を入力します
湿度%	湿度センサーの値を表示します。コンテンツには数値を入力します
気圧hPa	気圧センサーの値を表示します。コンテンツには数値を入力します
CO2濃度	CO2センサーの値を表示します。コンテンツには数値を入力します
開/閉	ドアの開閉を表示します。コンテンツには 0. 閉じる、1. 開くのどちらかを入力します
日付	現在の日付（月日）を表示します。コンテンツには PreMappingで <code>cp.Content = now()</code> と入力します
時間	現在の時間（時分）を表示します。コンテンツには PreMappingで <code>cp.Content = now()</code> と入力します

7.4.9.28. HTTP Request

HTTP Requestコンポーネントを使用すると、指定したURLへHTTPリクエストを送信することができます。

HTTP Requestコンポーネントは入力ペイロード(`cv.Payload`)をプロパティに応じてHTTPリクエストのBodyコンテンツとしてHTTPリクエストを実行し、そのHTTPレスポンスのBodyを出力ペイロード(`cv.Payload`)に書き出します。

Content-TypeプロパティによるHTTPリクエスト時のBodyの内容

Content-Type プロパティ	HTTPリクエスト時のBodyの内容
なし	入力ペイロードはBodyに設定されずにHTTPリクエストを実行します。
デフォルト	入力ペイロードの型に応じてContent-Typeを自動で選びます。
文字列型	「text/plain; charset=utf-8」でPayloadを文字列としてHTTPリクエストを実行します。
JSON配列、 JSONオブジェ クト	「application/json」として、Payloadをマーシャリングしたバイト列でHTTPリクエストを実行します。
バイト列	「application/octet-stream」にしてPayloadを文字列に変換したものをBodyとしてHTTPリクエストを実行します。
その他	「text/plain」にして文字列に変換したものをBodyとしてHTTPリクエストを実行します。
application/x- www-form- urlencoded	入力ペイロードはJSONオブジェクト型として、キーと値を「=」ではさみ、それぞれを&で区切った形でBodyにした上で、HTTPリクエストを実行します。（この場合で、JSON Object型以外のデータが入力としてきた場合はエラーとなります。）
multipart/form- data	入力ペイロードはJSON Object型として、キーと値をマルチパートで区切り、Content-Typeのmultipart/form-dataはバウンダリを自動的に付加してHTTPリクエストを実行します。（この場合、JSON Object型以外のデータが入力としてきた場合はエラーとなります。）
それ以外の任 意に入力され たContent- Type	入力されたContent-Typeで入力ペイロードをバイト列にしたものをBodyとして出力します。

HeadersプロパティによるHTTPヘッダーの上書きと追加

Headersプロパティ	説明
<code>cp.Headers</code>	リクエストのヘッダー

入力と出力のペイロード

入力ペイロード	説明
---------	----

<code>cv.Payload</code>	HTTPリクエストのBody
-------------------------	----------------

出力 ペイ ロード	説明
<code>cv.Payload</code>	HTTPレスポンスのContent-Typeヘッダが“application/json”の場合はJSON型、“text/*”でcharset (utf-8/shift_jis/euc-jp/iso-2022-jp) が指定されている場合は文字列として、その他のContent-Typeヘッダの場合はバイト列のまま格納します。
<code>cv.RawBody</code>	HTTPレスポンスのBodyをバイト列のまま格納します。

NewAction

HTTPRequest-1
HTTP Request

ステップの追加

プロパティ

変数名を表示

Pre Mappings

HTTPメソッド:

GET

このプロパティは必須です

URL:

https://

このプロパティは必須です

Content-Type:

(なし)

このプロパティは必須です

認証方式:

(なし)

このプロパティは必須です

ユーザ名:

パスワード:

トークン:

☐ 無効な証明書を許可

タイムアウト:

60000

このプロパティは必須です

基本プロパティ: HTTPプロキシ

なし

プロキシURL:

プロキシユーザ名:

プロキシパスワード:

ダンプ:

(なし)

このプロパティは必須です

Post Mappings

コンポーネントプロパティ

プロパティ名	説明
HTTPメソッド	GET,POST,HEAD,PUT,DELETE
URL	HTTPリクエストを送信するURL
Content-Type	HTTPヘッダのContent-Type
認証方式	なし, ベーシック認証, ベーシック/ダイジェスト認証, Bearer認証. cp.Authorizationの値で指定する場合は、"(none)", "prebasic", "basicdigest", "bearer" のいずれか
ユーザ名	ベーシック認証ユーザ名
パスワード	ベーシック認証パスワード
トークン	Bearerトークン
リトライ回数	サーバとのコネクションが確立できないエラー、HTTPレスポンスが受信できないエラー、HTTPレスポンスが500以上、408と429の場合にリトライをする回数
リトライ間隔	リトライする場合の次にリトライするまでの間隔値
タイムアウト	HTTPリクエストのレスポンスタイムアウト値
プロキシURL	Proxy利用時のURL
プロキシユーザ名	Proxy利用時のユーザ名
プロキシパスワード	Proxy利用時のパスワード
リクエストレスポンスダン プ	なし, Bodyなし, Bodyあり

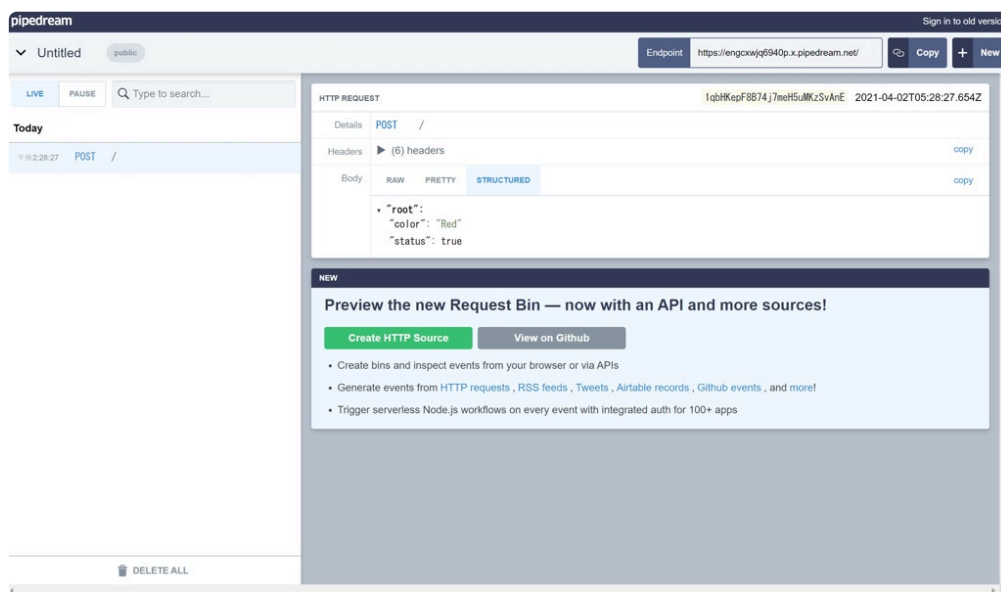
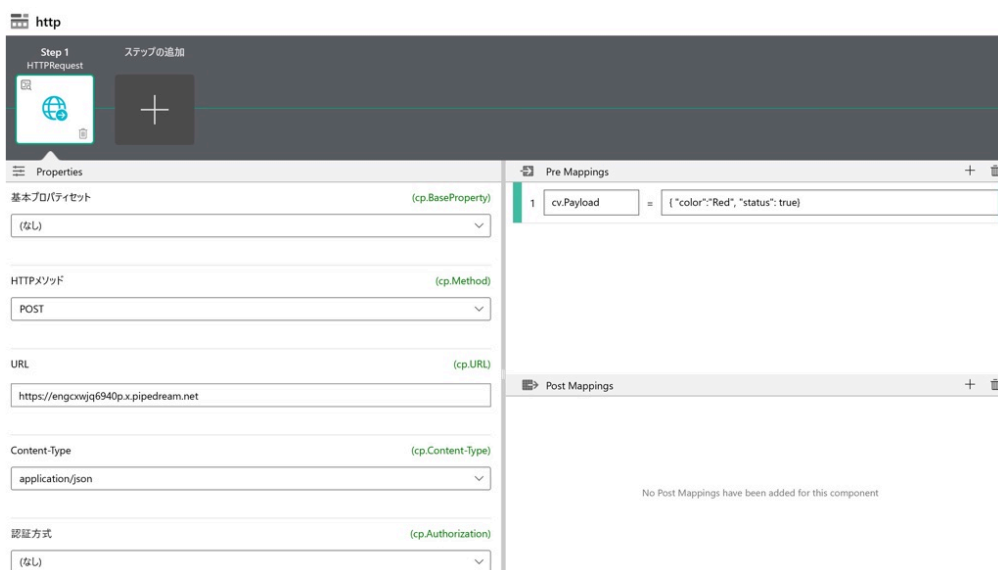
コンポーネント変数

コンポーネント変数	説明
cv.Payload	レスポンスのbody
cv.StatusCode	リクエスト結果
cv.Headers	レスポンスのヘッダー
cv.ContentLength	レスポンスの長さ

コンポーネントのテスト

テストエンドポイントを作成し、HTTPリクエストを送信するには、requestbin.comという無料のサービスをお勧めし

ます。このサービスのライブダッシュボードでは、受信リクエストを確認できます。（上記のリンクはパブリック requestbin を作成することに注意してください。プライベート requestbin の場合、アカウントが必要になります）。



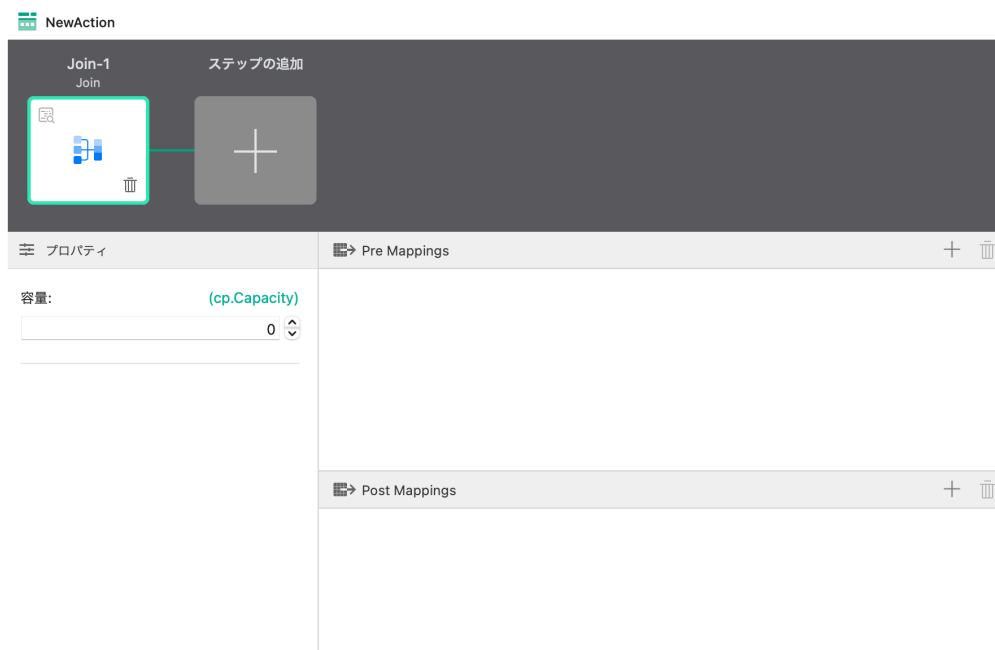
ヘッダーの上書きと追加

HTTPリクエスト Header 情報を PreMapping で JSON 形式で上書きと追加ができます。

```
cp.Headers = {"X-Header-1": "abc; def", "X-Header-2": ap.Timeout/1000, "X-Header-3": true, "X-Header-4": cp.URL, "Accept": "application/json", "User-Agent": "fugahoge browser"}
```


7.4.9.29. Join

Joinコンポーネントは、Joinコンポーネント複数の入力ペイロード(`cv.Payload`)を配列にまとめて出力する



例) CSVReadとCSVWriteの間にJoinコンポーネントを挟み、Joinのキャパシティプロパティで2を選ぶと2行ずつ配列にし、出力ペイロード(`cv.Payload`)を生成して、後続のコンポーネントに渡すことができます。

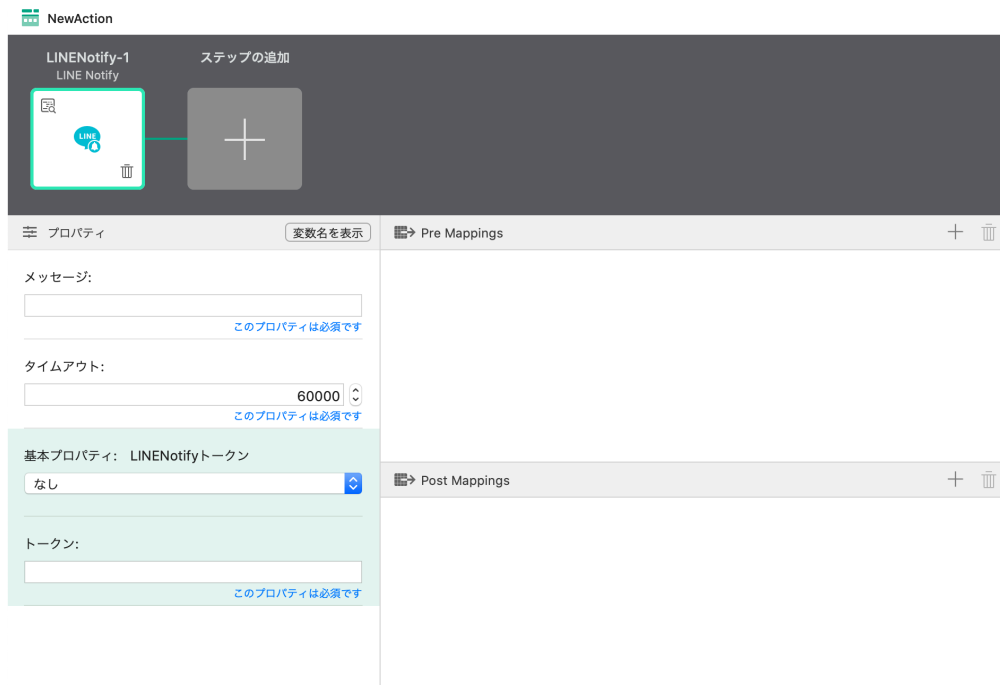
出力ペイロード	説明
<code>cv.Payload</code>	結合された配列

コンポーネントプロパティ

プロパティ名	説明
キャパシティ	出力される値をいくつの入ってきた内容でまとめて出力するかを指定する。

7.4.9.30. LINE Notify

LINE Notifyアクションコンポーネントは、LINE Notifyサービス (<https://notify-bot.line.me/>) を使用して、コンポーネントのプロパティに指定されたメッセージをポストすることができます。センサーからの情報を人に通知したりすると便利です。



コンポーネントプロパティ

プロパティ名	説明	例
メッセージ	LINEに投稿するメッセージ文字列	例) "現在の気温は"+cv.Payload+"度です"
パッケージID	メッセージと一緒にスタンプを送信します。スタンプIDと一緒に指定します *1	例) 446
スタンプID	メッセージと一緒にスタンプを送信します。パッケージIDと一緒に指定します *1	例) 1988
イメージファイル	メッセージと一緒に画像を送信します。JPEGまたはPNGファイルのファイルパスを指定します。	例) image.jpg
タイムアウト	APIのタイムアウト	
トークン	LINE Notifyサービスで発行したアクセストークン	

※1 LINEスタンプのパッケージIDとスタンプIDの指定方法は [こちら](#) を参照してください。

※ファイルパスを指定しないイメージファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル

名の書き方は [こちら](#) を参照してください。

7.4.9.31. LINE WORKS (旧版)

LINE WORKSコンポーネントは、LINE WORKSを使用してコンポーネントのプロパティに指定されたメッセージをポストすることができます。センサーからの情報を人に通知したりすると便利です。

※このLINEWORKSコンポーネントはLINE WORKS API 1.0を使用しておりました。新しいLINEWORKSコンポーネントではAPI2.0を使用し、互換性がありません。新たにLINE WORKS Developer Consoleにて設定が必要となります。LINE WORKS Developer Consoleでの設定方法につきましては [こちら](#) を参照してください。

※LINE WORKSコンポーネントを使用するには、管理者が初期設定としてLINE WORKS Developer Consoleで設定し、Botを公開設定する必要があります。LINE WORKS Developer Consoleでの設定方法につきましては [こちら](#) を参照してください。

※LINE Notifyサービス (<https://notify-bot.line.me/>)を使用する場合には、LINE Notifyアクションコンポーネントを使用してください。

コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「LINE WORKS Bot Connection」カテゴリーの基本プロパティを指定します	
メッセージ	LINE WORKSに投稿するメッセージ文字列	例) "現在の気温は"+cv.Payload+"度です"

パッケージID	メッセージと一緒にスタンプを送信します。スタンプIDと一緒に指定します ※1	例) 446
スタンプID	メッセージと一緒にスタンプを送信します。パッケージIDと一緒に指定します ※1	例) 1988
イメージファイル	メッセージと一緒に画像を送信します。JPEGまたはPNGファイルのファイルパスを指定します。	例) image.jpg
ファイル	メッセージと一緒にファイルを送信します。拡張子が bat、cmd、com、cpl、exe、scr、js、vbs、wsf、hta のものは送信できません	
送信先種類	メッセージを送信する送信先をメンバーアカウントかトークルームIDを選択します	
送信先ID	送信先種類で選択したメンバーアカウントかトークルームIDを指定します	例) メンバーアカウントの場合 aa aa@bbbb、トークルームIDの場合 12345
タイムアウト	APIのタイムアウト	
API ID	LINE WORKS Developer ConsoleからAPIのIDを指定します ※3	
Consumer Key	LINE WORKS Developer ConsoleからConsumer Keyを指定します ※3	
Server Token	LINE WORKS Developer ConsoleからServer List (固定IPタイプ)のTokenを指定します ※2 ※3	
Server ID	LINE WORKS Developer ConsoleからServer List (ID登録タイプ)のServer IDを指定します ※2 ※3	
認証キー	LINE WORKS Developer ConsoleからServer List (ID登録タイプ)の認証キーを指定します ※2 ※3	
Bot No	LINE WORKS Developer ConsoleからBot Noを指定します ※3	

※1 LINEスタンプのパッケージIDとスタンプIDの指定方法は [こちら](#) を参照してください。

※2 Server TokenとServer ID、認証キーはServer List (固定IPタイプ)とServer List (ID登録タイプ)の設定した方を使用します。

※3 Developer Consoleから指定する項目は [こちら](#) を参照してください。

※ファイルパスを指定しないイメージファイル/ファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.32. LINE WORKS

LINE WORKSコンポーネントは、LINE WORKSを使用してコンポーネントのプロパティに指定されたメッセージをポストすることができます。センサーからの情報を人に通知したりすると便利です。

※このLINEWORKSコンポーネントはLINE WORKS API 2.0を使用する新しいコンポーネントとなります。従来のLINEWORKSコンポーネントではAPI1.0を使用しており、互換性がありません。新たにLINE WORKS Developer Consoleにて設定が必要となります。LINE WORKS Developer Consoleでの設定方法につきましては [こちら](#) を参照してください。

※LINE WORKSコンポーネントを使用するには、管理者が初期設定としてLINE WORKS Developer Consoleで設定し、Botを公開設定する必要があります。LINE WORKS Developer Consoleでの設定方法につきましては [こちら](#) を参照してください。

※LINE Notifyサービス (<https://notify-bot.line.me/>)を使用する場合には、LINE Notifyアクションコンポーネントを使用してください。

コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「LINE WORKS Bot Connection」カテゴリーの基本プロパティ	

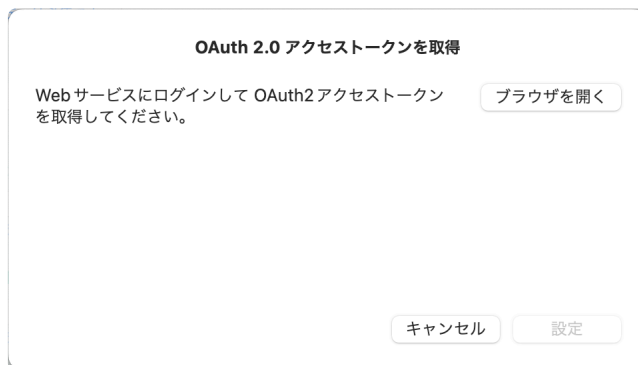
イ	を指定します	
メッセージ	LINE WORKSに投稿するメッセージ文字列	例) "現在の気温は"+cv.Payload+"度です"
パッケージID	メッセージと一緒にスタンプを送信します。スタンプIDと一緒に指定します ※1	例) 446
スタンプID	メッセージと一緒にスタンプを送信します。パッケージIDと一緒に指定します ※1	例) 1988
イメージファイル	メッセージと一緒に画像を送信します。JPEGまたはPNGファイルのファイルパスを指定します。	例) image.jpg
ファイル	メッセージと一緒にファイルを送信します。拡張子が bat、cmd、com、cpl、exe、scr、js、vbs、wsf、hta のものは送信できません	
送信先種類	メッセージを送信する送信先をメンバーアカウントかトークルームIDを選択します	
送信先ID	送信先種類で選択したメンバーアカウントかトークルームIDを指定します	例) メンバーアカウントの場合 aa aa@bbbb、トークルームIDの場合 12345
タイムアウト	APIのタイムアウト	
OAuth2クライアントID	LINE WORKS Developer ConsoleからAPI 2.0のClient IDを指定します ※2	
OAuth2クライアントシークレット	LINE WORKS Developer ConsoleからClient Secretを指定します ※2	
OAuth2リダイレクトURL	LINE WORKS Developer ConsoleからRedirect URLを指定します ※2	
OAuth2アクセストークン	トークンを取得ボタンを押して取得します。またトークンを更新ボタンを押して更新することも出来ます ※2 ※3	
Bot No	LINE WORKS Developer ConsoleからBot Noを指定します ※2	

※1 LINEスタンプのパッケージIDとスタンプIDの指定方法は [こちら](#) を参照してください。

※2 Developer Consoleから指定する項目は [こちら](#) を参照してください。

OAuth2アクセストークンの取得方法

※3 OAuth2アクセストークンを取得するボタンを押すとOAuth 2.0 アクセストークンを取得 パネルが表示されます。ブラウザを開く ボタンを押してトークンを取得してください。トークンを取得すると 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



Gravio Studio for Windowsの場合も同様にブラウザを開く ボタンを押してトークンを取得しますが、エラー画面となります。この状態でURLをアドレスバーからコピーして、そのURLをパネルにペースとして取得ボタンを押してください。トークンが取得されますので 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



ブラウザを開くボタンを押すと自動的にブラウザで認証が行われます。必要がありましたらログイン処理など画面の指示に合わせて操作してください。



認証が完了すると「このページに到達できません」とエラーが表示されるので、この時にURLをアドレスバーからコピーしてください。



コピーしたURLをパネルにペーストして取得ボタンを押すとOAuth2アクセストークンが“取得済み”となります。設定ボタンを押してパネルを閉じます。

7.4.9.33. Mail Send

Mail Sendコンポーネントは、電子メールを送信します。

Mail Sendコンポーネントは入力ペイロード(`cv.Payload`)をメール本文としてメールを送信するコンポーネントです。

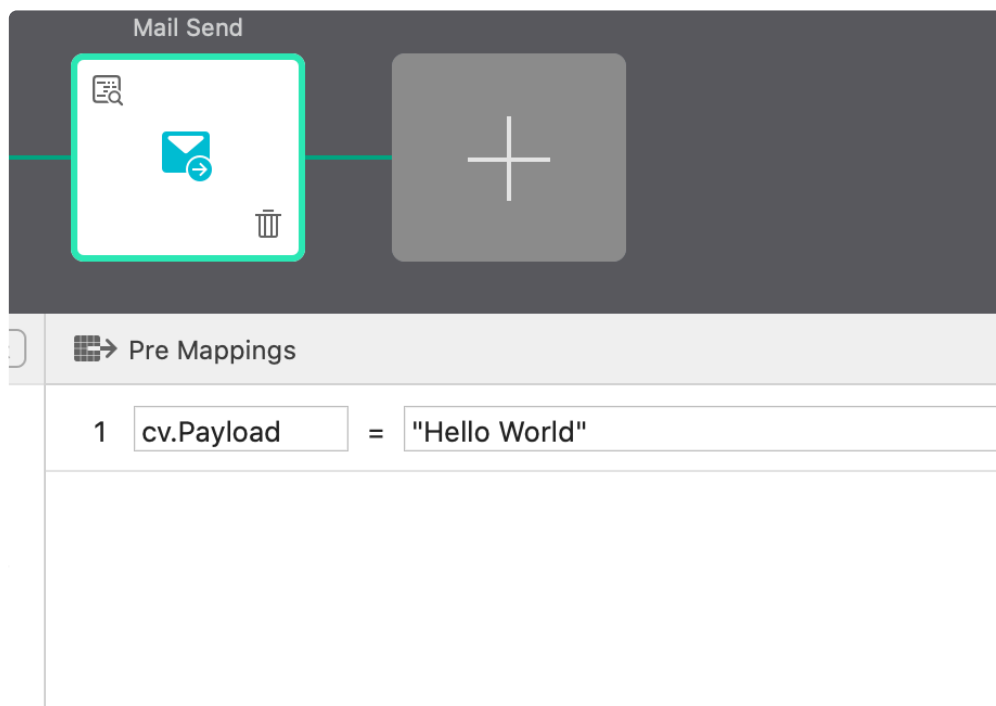
入力ペイロードのデータはいずれの型も文字列に変換してメール本文として送信します。

メール送信後は、入力ペイロード(`cv.Payload`)をそのまま出力ペイロード(`cv.Payload`)として格納します(パズスルー)

The screenshot shows the 'NewAction' configuration window for the 'MailSend-1' component. The interface is divided into two main sections: 'Pre Mappings' and 'Post Mappings'. The 'Pre Mappings' section is currently active and displays the configuration for the 'Mail Send' component. It includes fields for '宛先:' (To), 'Cc:', 'Bcc:', '件名:' (Subject), '差出人:' (From), '基本プロパティ: SMTPサーバ' (Basic Properties: SMTP Server), 'ホスト名:' (Host Name), 'ポート番号:' (Port Number), 'ユーザ名:' (Username), and 'パスワード:' (Password). Each field has a corresponding input box and a blue link below it that says 'このプロパティは必須です' (This property is required). The 'ポート番号' field has a dropdown menu showing '587'. The 'Post Mappings' section is currently empty.

電子メールを送信するにはSMTP情報が必要です。

Mail本文に内容を差し込むには、`cv.Payload`変数に文字列をPre-Mappingで書き込むことで実現できます。



コンポーネント変数

コンポーネント変数	説明
cv.Payload	Mailの本文にいれる内容をこの変数に設定する

コンポーネントプロパティ

プロパティ名	説明
宛先	送信先のメールアドレスを指定
Cc	Ccのメールアドレスを指定
Bcc	Bccのメールアドレスを指定
件名	メールの件名を指定
差出人	送信元メールアドレスを指定
添付ファイル	添付ファイルを指定
メッセージはHTML	メッセージ本文をHTMLとして送信
ホスト名	SMTPサーバホスト名
ポート番号	SMTPサーバのポート番号
認証方式	先進認証 (OAuth2) または基本認証 (SMTP)
ユーザ名	SMTPサーバへのログインユーザ名 ※1 ※2
パスワード	SMTPサーバへのログインパスワード ※1
OAuth2クライアントID	クライアントID ※2

OAuth2クライアントシークレット	クライアントシークレット ※2
OAuth2リダイレクトURL	リダイレクトURL ※2
OAuth2スコープ	スコープ ※2
OAuth2認可URL	認可URL ※2
OAuth2トークンURL	トークンURL ※2
OAuth2アクセストークン	トークンを取得ボタンを押して取得します。またトークンを更新ボタンを押して更新することも出来ます ※2 ※3

※Gmailのメールアドレスとパスワードを使用して基本認証（SMTP）でメール送信する場合の設定例は [こちら](#) を参照してください

※Gmailのメールアドレスを使用して先進認証（OAuth2）でメール送信する場合の設定例は [こちら](#) を参照してください

※1は認証方式が基本認証（SMTP）の時に使用します

※2は認証方式が先進認証（OAuth2）の時に使用します

Gmailを使用する場合の設定方法

ホスト名	smtp.gmail.com
OAuth2クライアントID	GCPで設定したクライアントID
OAuth2クライアントシークレット	GCPで設定したクライアントシークレットID
OAuth2リダイレクトURL	GCPで設定した承認済みのリダイレクトURI
OAuth2スコープ	https://mail.google.com/
OAuth2認可URL	https://accounts.google.com/o/oauth2/auth
OAuth2トークンURL	https://accounts.google.com/o/oauth2/token

Microsoft 365を使用する場合の設定方法

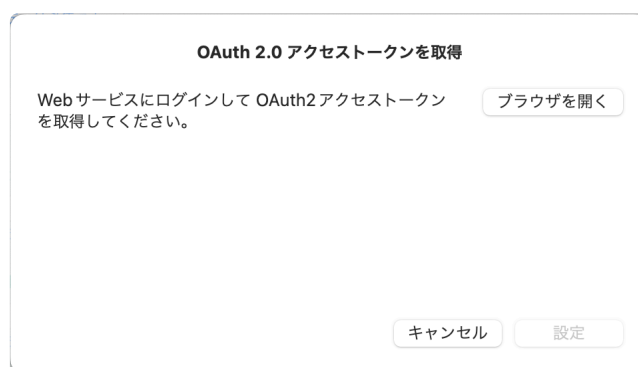
ホスト名	smtp.office365.com
OAuth2クライアントID	Azureで設定したクライアントID
OAuth2クライアントシークレット	Azureで設定したクライアントシークレットID
OAuth2リダイレクトURL	Azureで設定したリダイレクトURI
OAuth2スコープ	offline_access,https://outlook.office.com/SMTP.Send
OAuth2認可URL	https://login.microsoftonline.com/common/oauth2/v2.0/authorize
OAuth2トークンURL	https://login.microsoftonline.com/common/oauth2/v2.0/token

Microsoft 365でのエラーの対処方法

Microsoft 365を使用して MailSend: failed to send mail: 535 5.7.139 Authentication unsuccessful, SmtplibClientAuthentication is disabled for the Tenant. Visit http://aka.ms/smtp_auth_disabled for more information. このようなSmtplibClientAuthenticationが無効になっているというエラーが発生した場合には [こちら](#) を参照してMicrosoft 365 管理センターで認証済みSMTPの設定を有効にしてください。

OAuth2アクセストークンの取得方法

※3 OAuth2アクセストークンを取得するボタンを押すとOAuth 2.0 アクセストークンを取得 パネルが表示されます。ブラウザを開く ボタンを押してトークンを取得してください。トークンを取得すると 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



Gravio Studio for Windowsの場合も同様にブラウザを開く ボタンを押してトークンを取得しますが、エラー画面となります。この状態でURLをアドレスバーからコピーして、そのURLをパネルにペーストとして取得ボタンを押してください。トークンが取得されますので 設定ボタンを押すとOAuth2アクセストークン プロパティに自動的に設定されます。



ブラウザを開くボタンを押すと自動的にブラウザで認証が行われます。必要がありましたらログイン処理など画面の指

示に合わせて操作してください。



認証が完了すると「このページに到達できません」とエラーが表示されるので、この時にURLをアドレスバーからコピーしてください。



コピーしたURLをパネルにペーストして取得ボタンを押すとOAuth2アクセストークンが“取得済み”となります。設定ボタンを押してパネルを閉じます。

添付ファイル

添付ファイルは複数のファイルを指定することができます。

・添付するファイルのパス

Windows環境

ファイルはパスにはactmgr¥dataが自動的に保管されます。パスの指定も可能です。

※フォルダー名、ファイル名にスペースが含まれている場合には""(ダブルクォート)で囲む必要があります。

例)

```
actmgr¥data のimage.jpg を添付する場合は image.jpg を添付ファイルに指定します。  
actmgr¥data¥subfolder のimage.jpg を添付する場合は subfolder¥image.jpg を添付ファイル  
に指定します。  
actmgr¥data¥subfolder 2021 のimage.jpg を添付する場合は "subfolder 2021¥image.jp  
g" を添付ファイルに指定します。
```

絶対パスを指定することも可能です。

例)

```
c:¥ のimage.jpg を添付する場合は c:¥image.jpg を添付ファイルに指定します。
```

ファイルは複数のファイル、ワイルドカードを指定することも可能です。

複数のファイルのセパレーターは";"(セミコロン)を使用し、ワイルドカードは"*"(アスタリスク)を使用します。

例)

```
actmgr¥data のimage.jpgとimage2.jpg を添付する場合は image.jpg;image2.jpg を添付ファイル  
に指定します。  
actmgr¥data¥subfolder の*.jpg を添付する場合は subfolder¥*.jpg を添付ファイルに指定しま  
す。  
actmgr¥data のimage.jpgとimage2.jpgとactmgr¥data¥subfolder の*.jpg を添付する場合は  
image.jpg;image2.jpg;subfolder¥*.jpg を添付ファイルに指定します。
```

Mac、Linux、RaspberryPI、Gravio Hub環境

ファイルはパスにはactmgr/dataが自動的に保管されます。パスの指定も可能です。

※フォルダー名、ファイル名にスペースが含まれている場合には""(ダブルクォート)で囲む必要があります。

例)

```
actmgr/data のimage.jpg を添付する場合は image.jpg を添付ファイルに指定します。  
actmgr/data/subfolder のimage.jpg を添付する場合は subfolder/image.jpg を添付ファイル  
に指定します。  
actmgr/data/subfolder 2021のimage.jpg を添付する場合は "subfolder 2021/image.jpg"  
を添付ファイルに指定します。
```

絶対パスを指定することも可能です。

例)

`/home/username` の `image.jpg` を添付する場合は `/home/username/image.jpg` を添付ファイルに指定します。

ファイルは複数のファイル、ワイルドカードを指定することも可能です。

複数のファイルのセパレーターは";" (セミコロン) を使用し、ワイルドカードは"*" (アスタリスク) を使用します。

例)

`actmgr/data` の `image.jpg` と `image2.jpg` を添付する場合は `image.jpg;image2.jpg` を添付ファイルに指定します。

`actmgr/data¥subfolder` の `*.jpg` を添付する場合は `subfolder/*.jpg` を添付ファイルに指定します。

`actmgr/data` の `image.jpg` と `image2.jpg` と `actmgr/data/subfolder` の `*.jpg` を添付する場合は `image.jpg;image2.jpg;subfolder/*.jpg` を添付ファイルに指定します。

p.

7.4.9.34. MQTT Publish

MQTTパブリッシュ・アクション・コンポーネントを使用すると、MQTT PubSubシステム内のトピックにメッセージをパブリッシュできます。

MQTT

MQTTPublish-1

MQTT Publish

ステップの追加

プロパティ

変数名を表示

Pre Mappings

トピック:

このプロパティは必須です

クライアントID:

このプロパティは必須です

QoS:

QoS 0

このプロパティは必須です

☐ Retain

このプロパティは必須です

タイムアウト:

60000

このプロパティは必須です

基本プロパティ: MQTTコネクション

なし

URL:

ssl://

このプロパティは必須です

ポート:

8883

このプロパティは必須です

ユーザ名:

パスワード:

証明書:

秘密鍵:

秘密鍵パスフレーズ:

ルート証明書:

Post Mappings

出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明	例
トピック	MQTTのトピック	
クライアントID	MQTTのクライアントID	
QoS	QoS値	0,1,2から選択します
Retain	Retainフラグ	
タイムアウト	タイムアウト	通常は変更する必要はありません
URL	MQTTサーバのurl	ssl://のスキーマで指定します。sslを使用しないときはホスト名 (IPアドレス) のみを指定します
ポート	MQTTサーバのポート	
ユーザ名	MQTTサーバへの認証	認証を設定している場合に指定します
パスワード	MQTTサーバへの認証	認証を設定している場合に指定します
証明書	X.509 の certificate based authenticationの証明書	
秘密鍵	X.509 の certificate based authenticationの秘密鍵	
秘密鍵パスフレーズ	秘密鍵のパスフレーズ	
ルート証明書	証明書のルート証明書	

7.4.9.35. SQL Server Exec

SQL Server Execコンポーネントは、SQL Serverに対してSQL文を実行してデータを更新/削除することができます。

SQL Server 2005以上に対応しています。

h3. 出力ペイロード

出力ペイロード	説明
<code>cv.Payload</code>	入力ペイロードに加えて実行したSQL文によりLastInsertIdまたはRowsAffectedを渡します

LastInsertIdは、実行したSQL文がINSERT文だったときに、DBによって生成されたIDが取得できた場合に返されます。

RowsAffectedは、実行したSQL文によって影響を受けたレコード数を返します。例えば、DELETE文だったときに削除された行数が戻ります。

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
バルクモード	SQL文の実行でバルクモードを使用するかを選択
バルクサイズ	バルクモードを使用する場合にそのサイズを指定します
ホスト	SQL Serverのホストを指定します

ポート	SQL Serverのポートを指定します
ユーザ名	SQL Serverのユーザ名を指定します
パスワード	SQL Serverのパスワードを指定します
データベース	データベースを指定します
Encrypt	通信の暗号化を disable (SSL/TLS接続をしない)、true (SSL/TLS接続をする)、false (ログインのみ暗号化する) から指定します
TrustServerCertificate	SQL Serverのサーバー証明書を信頼するか選択します。オンの場合はSQL Server が送信する証明書を信頼します。オフを指定した場合にはトラストストア証明書に照らしてSQL Server 証明書を検証します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
UPDATE table1 SET name='更新名' WHERE id=?
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを更新します。

プレースホルダーは2つ以上指定することもできます。

```
UPDATE table1 SET name=? WHERE id=?
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

Text2SQL

Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.36. SQL Server Query

SQL Server Queryコンポーネントは、SQL Serverに対してクエリー文を実行してデータを取得することができます。

SQL Server 2005以上に対応しています。

出力ペイロード

出力ペイロード	説明
<code>cv.Payload</code>	条件に一致した各レコードをプロパティで指定した出力形式でレコードを出力します。

配列形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「配列」を指定すると、以下のように出力ペイロードが生成されます。

出力例

```
[1, "abc"]  
[2, "def"]  
[3, "hij"]
```

オブジェクト形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「オブジェクト形式」を指定すると、以下のように出力ペイロードが生成され

ます。

出力例

```
{ "column1": 1, "column2": "abc" }
{ "column1": 2, "column2": "def" }
{ "column1": 3, "column2": "hij" }
```

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
出力形式	配列形式かオブジェクト形式を選択
ホスト	SQL Serverのホストを指定します
ポート	SQL Serverのポートを指定します
ユーザ名	SQL Serverのユーザ名を指定します
パスワード	SQL Serverのパスワードを指定します
データベース	データベースを指定します
Encrypt	通信の暗号化を disable (SSL/TLS接続をしない)、true (SSL/TLS接続をする)、false (ログインのみ暗号化する) から指定します
TrustServerCertificate	SQL Serverのサーバー証明書を信頼するか選択します。オンの場合はSQL Server が送信する証明書を信頼します。オフを指定した場合にはトラストストア証明書に照らしてSQL Server 証明書を検証します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
SELECT * FROM table1 WHERE id=?
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを取得できます。

プレースホルダーは2つ以上指定することもできます。

```
SELECT * FROM table1 WHERE name=? AND age=?
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

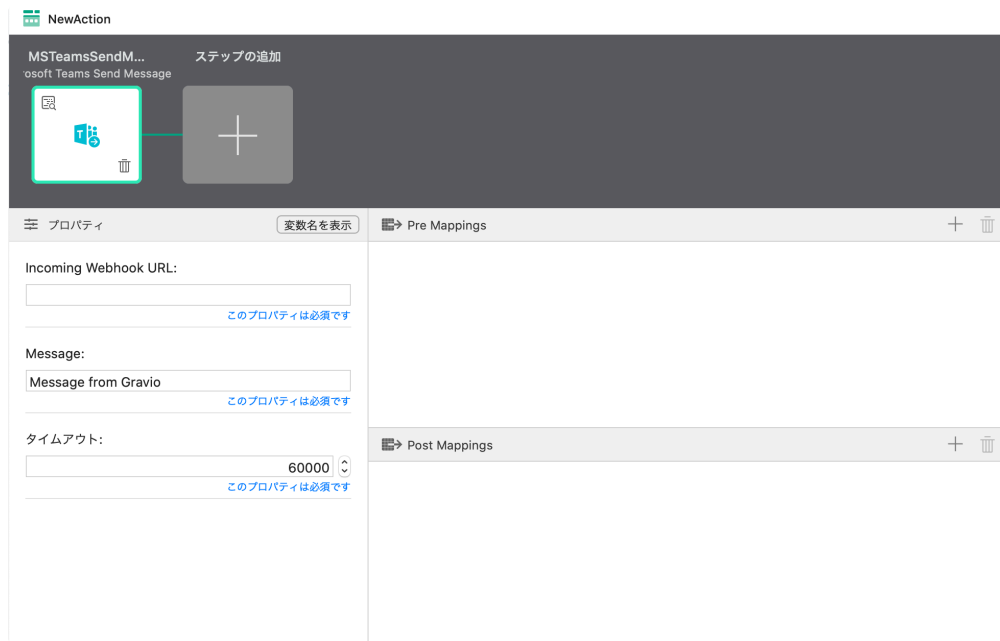
Text2SQL

Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.37. MStTeams Send Message

MStTeams Send Messageアクションコンポーネントは、マイクロソフト TeamsAPIを使用して、コンポーネントのプロパティに指定されたチャンネルにメッセージをポストすることができます。



Incoming Webhook URLの取得方法：

Teamsのメッセージをポストするチャンネルにある「その他のオプション」メニューから「コネクタ」を選択します。

コネクタの設定画面から“Incoming Webhook”を選択して「構成」ボタンを押します。

Incoming Webhookの名前を入力して「作成」ボタンを押します。

Incoming Webhookが作成されるとURLも自動的に作成されます。

詳細は [こちら](#) を参照してください。

コンポーネントのIncoming Webhook URLプロパティには、このURLを指定してください。

出力ペイロード	説明
<code>cv.Payload</code>	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

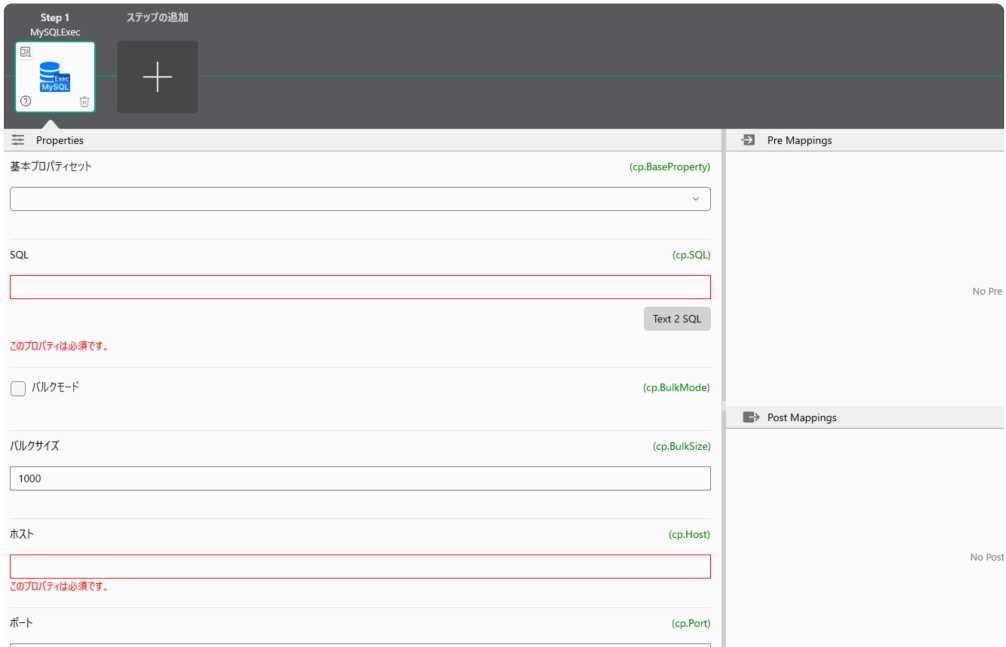
コンポーネントプロパティ

プロパティ名	説明	例
Incoming Webhook URL	TeamsAPIのurl	
Message	送信するメッセージ	例) "現在の気温は"+ <code>cv.Payload</code> +"度です"

タイムアウト	APIのタイムアウト	通常は変更する必要はありません
--------	------------	-----------------

7.4.9.38. MySQL Exec

MySQL Execcコンポーネントは、MySQLに対してSQL文を実行してデータを更新/削除することができます。
MySQL 5.6以上に対応しています。



出力ペイロード

出力ペイロード	説明
cv.Payload	入力ペイロードに加えて実行したSQL文によりLastInsertIdまたはRowsAffectedを渡します

LastInsertIdは、実行したSQL文がINSERT文だったときに、DBによって生成されたIDが取得できた場合に返されます。
RowsAffectedは、実行したSQL文によって影響を受けたレコード数を返します。例えば、DELETE文だったときに削除された行数が戻ります。

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
バル	SQL文の実行でバルクモードを使用するかを選択

クモード	
バルクサイズ	バルクモードを使用する場合にそのサイズを指定します
ホスト	MySQLのホストを指定します
ポート	MySQLのポートを指定します
ユーザ名	MySQLのユーザ名を指定します
パスワード	MySQLのパスワードを指定します
データベース	データベースを指定します
タイムゾーン	データベースのタイムゾーンを指定します
TLS	通信の暗号化を true (SSL/TLS接続をする)、false (SSL/TLS接続をしない)、skip-verify (SSL/TLS接続をするがサーバー証明書の検証をしない)、preferred (最初にSSL/TLS接続を試し接続できなかったら、次にSSL/TLS接続をしない接続を試す) から指定します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
UPDATE table1 SET name='更新名' WHERE id=?
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを更新します。

プレースホルダーは2つ以上指定することもできます。

```
UPDATE table1 SET name=? WHERE id=?
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

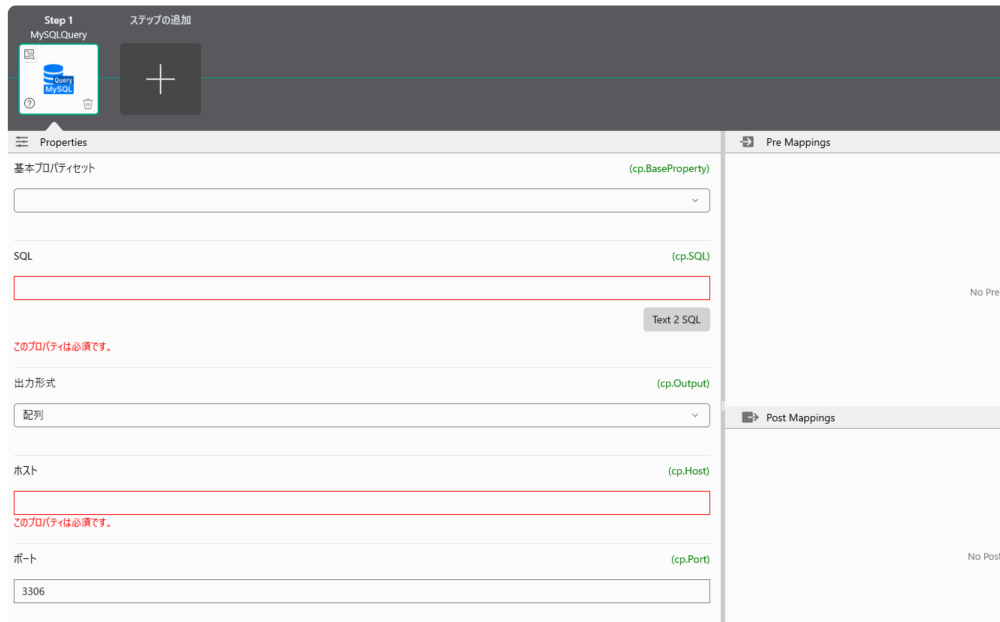
Text2SQL

Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.39. MySQL Query

MySQL Queryコンポーネントは、MySQLに対してクエリー文を実行してデータを取得することができます。MySQL 5.6以上に対応しています。



出力ペイロード

出力ペイロード	説明
<code>cv.Payload</code>	条件に一致した各レコードをプロパティで指定した出力形式でレコードを出力します。

配列形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「配列」を指定すると、以下のように出力ペイロードが生成されます。

出力例

```
[1, "abc"]  
[2, "def"]  
[3, "hij"]
```

オブジェクト形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「オブジェクト形式」を指定すると、以下のように出力ペイロードが生成されます。

出力例

```
{ "column1": 1, "column2": "abc" }  
{ "column1": 2, "column2": "def" }  
{ "column1": 3, "column2": "hij" }
```

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
出力形式	配列形式かオブジェクト形式を選択
ホスト	MySQLのホストを指定します
ポート	MySQLのポートを指定します
ユーザ名	MySQLのユーザ名を指定します
パスワード	MySQLのパスワードを指定します
データベース	データベースを指定します
タイムゾーン	データベースのタイムゾーンを指定します
TLS	通信の暗号化を true (SSL/TLS接続をする)、false (SSL/TLS接続をしない)、skip-verify (SSL/TLS接続をするがサーバー証明書の検証をしない)、preferred (最初にSSL/TLS接続を試し接続できなかったら、次にSSL/TLS接続をしない接続を試す) から指定します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
SELECT * FROM table1 WHERE id=?
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを取得できます。

プレースホルダーは2つ以上指定することもできます。

```
SELECT * FROM table1 WHERE name=? AND age=?
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

Text2SQL

Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.40. OpenAICompletion

OpenAICompletionコンポーネントは、OpenAI社が提供しているAI機能と連携するコンポーネントです。OpenAICompletionは与えられたプロンプトに対する補完を返します。チャットベースの機能は [OpenAIChat コンポーネント](#) を使用してください。

※このOpenAICompletionコンポーネントを使用するにはOpenAIのAI機能呼び出すためにAPI Keyを予め取得する必要があります。OpenAIのサイトでAPI Keyを取得する方法につきましては [こちら](#) を参照してください。

コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「OpenAI」カテゴリーの基本プロパティを指定します	
API Key	API Key *1	
Organization Id	組織ID *1	
User	ユーザー名	
Model	使用するモデル名の選択または手入力	Text Davinci 003 Text Davinci 002 Text Curie 001 Text Babbage 001 Text Ada 001

Temperature	サンプリング温度 0～1 で指定します	
Prompt	プロンプト	
Suffix	接尾辞	
Max Tokens	生成されるトークンの最大数	
Top-P	核サンプリングを 0～1 で指定します。デフォルトは1	
N	プロンプトごとに生成する回答数	
Log Probabilities	最も可能性の高いトークンのログ確率と、選択したトークンを含めます	
Echo	回答にプロンプトを追記します	
Stop	トークンの生成を停止する文字列	
Presence Penalty	既に出てきた単語をもう一度使うかどうかを-2.0から2.0で指定します。デフォルトは0	
Frequency Penalty	出てきた回数が多いほどペナルティを与えるかを-2.0から2.0で指定します。デフォルトは0	
Best Of	候補回答の数を制御します。返す数を指定しますがbest_of は N より大きい必要があります	
Logit Bias	指定されたトークンが表示される可能性を変更するパラメータ	

※ OpenAI関連の各プロパティの詳細は [OpenAI APIリファレンス](#) を参照してください。

※1 OpenAICompletionコンポーネントを使用するには、API keyを取得する必要があります。API keyの取得方法につきましては [こちら](#) を参照してください。

出力のペイロード

出力ペイロード	説明
<code>cv.Payload</code>	OpenAI Completion APIからのレスポンスをそのまま格納します

7.4.9.41. OpenAIChat

OpenAIChatコンポーネントは、OpenAI社が提供しているAI機能と連携するコンポーネントです。OpenAIChatはチャット補完に特化した言語モデルを使用しています。複数のやりとりをする会話を簡単にするように設計されているチャットベースの機能を呼び出すことができます。

※このOpenAIChatコンポーネントを使用するにはOpenAIのAI機能を呼び出すためにAPI Keyを予め取得する必要があります。OpenAIのサイトでAPI Keyを取得する方法につきましては [こちら](#) を参照してください。

コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「OpenAI」カテゴリーの基本プロパティを指定します	
API Key	API Key *1	
Organization Id	組織ID *1	
User	ユーザー名	
Model	使用するモデル名の選択または手入力。使用出来るモデルはOpenAIとの契約により異なります。	GPT 3.5 Turbo GPT 3.5 Turbo 16k GPT 4 GPT 4 32k
System Message	役割の設定	
User Message Sample Input	ユーザーからの質問の例を指定します	

Assistant Message Sample Input	回答の例を指定します	
User Message	ユーザーからの質問	
Temperature	サンプリング温度 0～1 で指定します	
Max Tokens	レスポンスの最大トークン数	
Top-P	核サンプリングを 0～1 で指定します。デフォルトは1	
N	プロンプトごとに生成する回答数	
Log Probabilities	最も可能性の高いトークンのログ確率と、選択したトークンを含めます	
Stop	トークンの生成を停止する文字列	
Presence Penalty	既に出てきた単語をもう一度使うかどうかを-2.0から2.0で指定します。デフォルトは0	
Frequency Penalty	出てきた回数が多いほどペナルティを与えるかを-2.0から2.0で指定します。デフォルトは0	
Logit Bias	指定されたトークンが表示される可能性を変更するパラメータ	

※ OpenAI関連の各プロパティの詳細は [OpenAI APIリファレンス](#) を参照してください。

※1 OpenAIChatコンポーネントを使用するには、API keyを取得する必要があります。API keyの取得方法につきましては [こちら](#) を参照してください。

出力のペイロード

出力ペイロード	説明
<code>cv.Payload</code>	OpenAI Chat APIからのレスポンスをそのまま格納します

7.4.9.42. OpenAllImage

OpenAllImageコンポーネントは、OpenAI社が提供しているAI機能と連携するコンポーネントです。OpenAllImageは入力されたプロンプトから画像を生成する機能を呼び出すことができます。

※このOpenAllImageコンポーネントを使用するにはOpenAIのAI機能を呼び出すためにAPI Keyを予め取得する必要があります。OpenAIのサイトでAPI Keyを取得する方法につきましては [こちら](#) を参照してください。

Step 1 OpenAllImage ステップの追加

Properties

基本プロパティセット (cp.BaseProperty)

API Key (cp.APIKey)

Organization Id (cp.OrganizationID)

User (cp.User)

Operation (cp.Operation)

Pre Mappings

Post Mappings

コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「OpenAI」カテゴリーの基本プロパティを指定します	
API Key	API Key *1	
Organization Id	組織ID *1	
User	ユーザー名	
Operation	処理を 画像生成 画像編集 類似画像生成から選択します	Create Image Edit Image Make Image Variation
Prompt	プロンプト(処理 画像生成か 画像編集のみ)	

Image File	画像PNGファイル（処理 画像編集か 類似画像生成のみ）	
Mask File	マスクPNGファイル 4MB未満でImage Fileと同じサイズの画像を使用します（処理 画像編集のみ）	
n	プロンプトごとに生成する回答数	
Image Size	生成する画像サイズ	1024px by 1024px 512px by 512px 256px by 256px
Response Format	生成データフォーマット ファイル URL Base64	File URL Base64
Filename	出力ファイル名（生成データフォーマット ファイルのみ）	
File Overwrite	出力ファイルの上書き（生成データフォーマット ファイルのみ）ファイルを上書きしない場合で同一ファイル名が存在すると末尾に「-N」を付けて保存します	

※ OpenAI関連の各プロパティの詳細は [OpenAI APIリファレンス](#) を参照してください。

※1 OpenAIImageコンポーネントを使用するには、API keyを取得する必要があります。API keyの取得方法につきましては [こちら](#) を参照してください。

Operationプロパティ

・Create Imageを指定する場合

Image Fileには4MB未満で正方形の画像を使用します

・Edit Imageを指定する場合

Image Fileには4MB未満で正方形の画像を使用します、Mask Fileを指定する場合はImage Fileと同じサイズの画像を使用します。

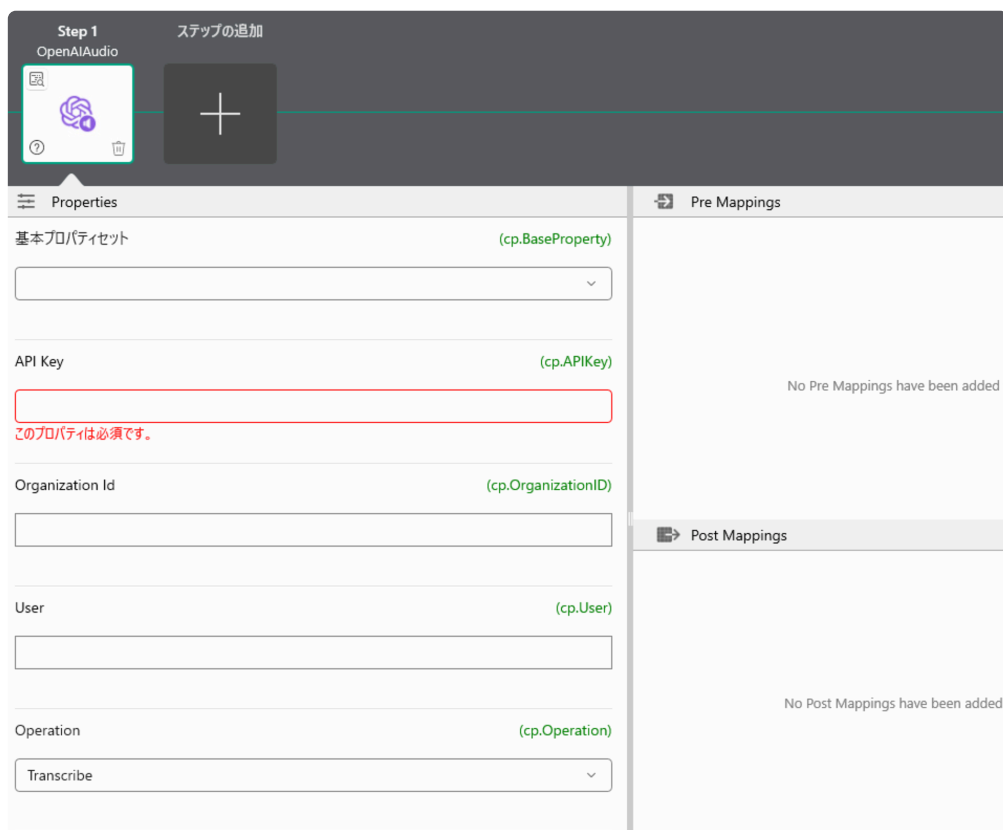
出力のペイロード

出力ペイロード	説明
cv.Payload	Response Format プロパティ ファイルの場合は書き出したファイルのファイルパス、URLの場合はURL、Base64の場合は生成してデータをBase64でエンコードしたデータを格納します

7.4.9.43. OpenAIAudio

OpenAIAudioコンポーネントは、OpenAI社が提供しているAI機能と連携するコンポーネントです。OpenAIAudioは入力された音声ファイルから文字起こしを行う機能呼び出すことができます。

※このOpenAIAudioコンポーネントを使用するにはOpenAIのAI機能呼び出すためにAPI Keyを予め取得する必要があります。OpenAIのサイトでAPI Keyを取得する方法につきましては [こちら](#) を参照してください。



コンポーネントプロパティ

プロパティ名	説明	例
基本プロパティ	「OpenAI」カテゴリーの基本プロパティを指定します	
API Key	API Key *1	
Organization Id	組織ID *1	
User	ユーザー名	
Operation	処理を 文字起こし 翻訳 から選択します	Transcribe Translation
Model	使用するモデル名を選択または手入力	Whisper-1

Audio File	音声ファイル名	
Language	音声の言語 ISO-639-1形式で指定します。デフォルトは英語です	ja en
Prompt	プロンプト	
Temperature	サンプリング温度 0～1 で指定します	
Response Format	出力データフォーマットを JSON 詳細なJSON テキスト SRTファイル VTTファイル から選択します	JSON Verbose JSON Text srt vtt

※ OpenAI関連の各プロパティの詳細は [OpenAI APIリファレンス](#) を参照してください。

※1 OpenAIAudioコンポーネントを使用するには、API keyを取得する必要があります。API keyの取得方法につきましては [こちら](#) を参照してください。

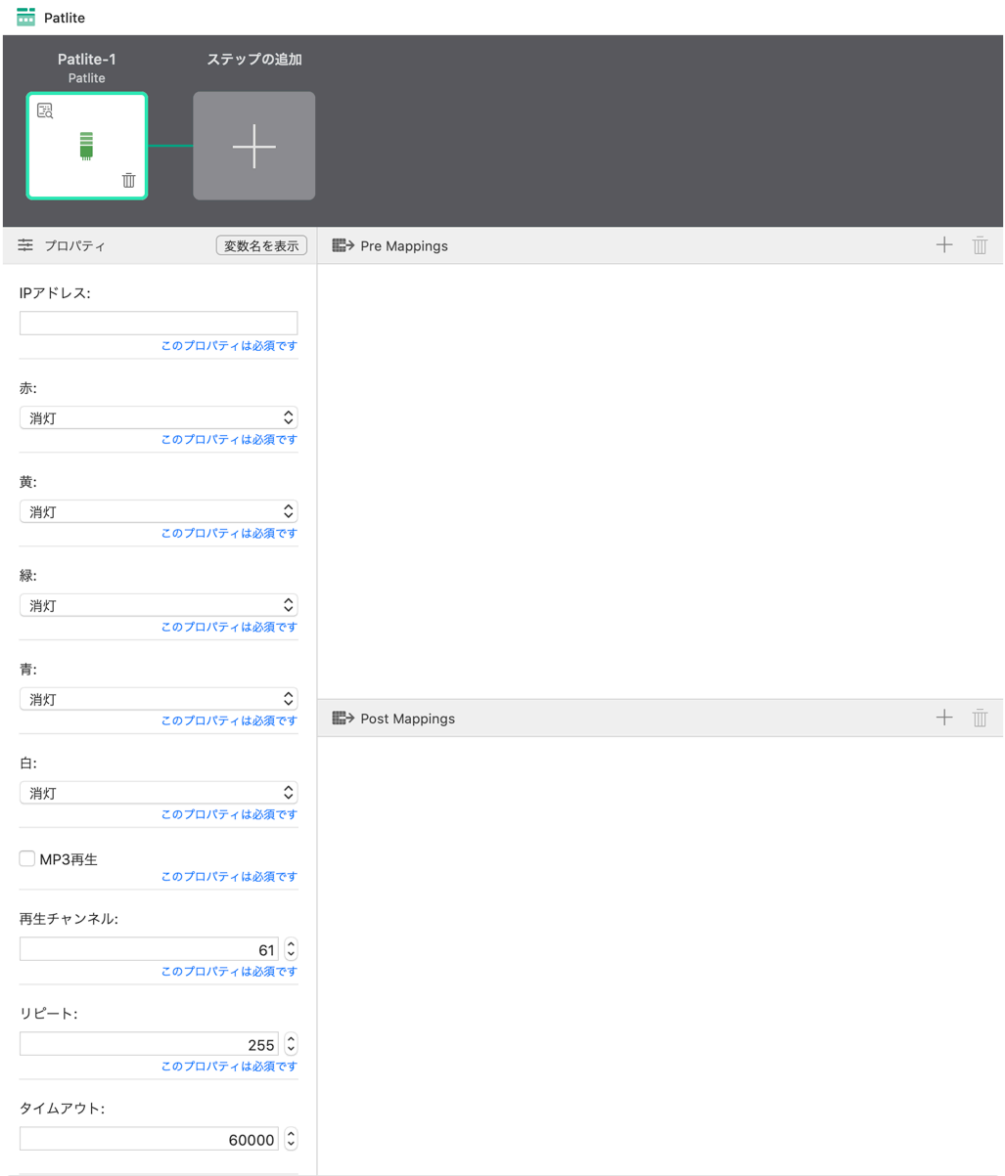
出力のペイロード

出力ペイロード	説明
cv.Payload	Response Format プロパティ 出力データフォーマットで指定した内容を格納します

7.4.9.44. PATLITE

PATLITEアクションコンポーネントは、パトライト社製のネットワーク対応監視塔のライトを制御したり、音声を再生したることができます。監視塔はHTTPコマンドに対応している必要があります。詳しくは、パトライト社のサイトをご確認ください。 <https://www.patlite.co.jp/>

センサーからの情報に応じて表示を制御すると便利です。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明	例

IP アドレス	監視塔のIPアドレス	
赤	赤のランプを制御します	消灯/点灯/点滅1/点滅2/変化なし[cp.Redに設定する値は "off", "on", "blink-1", "blink-2", "no-change"]
黄	黄色のランプを制御します	消灯/点灯/点滅1/点滅2/変化なし[cp.Amberに設定する値は "off", "on", "blink-1", "blink-2", "no-change"]
緑	緑のランプを制御します	消灯/点灯/点滅1/点滅2/変化なし[cp.Greenに設定する値は "off", "on", "blink-1", "blink-2", "no-change"]
青	青のランプを制御します	消灯/点灯/点滅1/点滅2/変化なし[cp.Blueに設定する値は "off", "on", "blink-1", "blink-2", "no-change"]
白	白のランプを制御します	消灯/点灯/点滅1/点滅2/変化なし[cp.Whiteに設定する値は "off", "on", "blink-1", "blink-2", "no-change"]
MP3 再生	チェックすると音声を再生します	
再生チャンネル	再生するチャンネルを数字で指定します。詳しくは機器のマニュアルを参照ください	例) 65
リピート	追加の繰り返し再生回数を指定します。繰り返さない場合は空白のままにしてください。	例) 1 を指定すると2回再生されます
タイムアウト	タイムアウト	通常は変更する必要はありません

7.4.9.45. Philips Hue

このコンポーネントでは、特定のPhilips Hueライトの設定を変更できます。色と不透明度を指定し、ブリッジのIPアドレス、ユーザーID、ライトIDを設定する必要があります。

The screenshot shows the 'NewAction' configuration window for a 'PhilipsHue-1' component. The interface is divided into two main sections: 'Properties' (プロパティ) on the left and 'Mappings' (マッピング) on the right. The 'Properties' section includes a 'Steps' (ステップ) area at the top with a 'Add Step' (+) button. Below this, there are several input fields: 'On' (checked), 'Color' (empty), 'Color Loop' (unchecked), 'IP Address' (empty), 'User ID' (empty), and 'Light ID' (empty). Each input field has a small blue link that says 'このプロパティは必須です' (This property is required). The 'Mappings' section is currently empty, showing only a 'Pre Mappings' header and a 'Post Mappings' header, both with a '+' button to add mappings.

IPアドレスは <https://discovery.meethue.com/> を参照してください。

この情報があれば、デバッガーAPIシミュレーターを `http://<your_ip>/debug/clip.html` を開きます。

JSONを使用してAPIに `POST` コマンドを送信する前に、ブリッジボタンを1回押してデバイスを“ペアにする”場合にのみ取得できるユーザーIDを取得する必要があります。{"devicetype": "my_hue_app#gravio"}

CLIP API Debugger

URL:

GET

PUT

POST

DELETE

Message Body:

```
{"devicetype":"my_hue_app#gravio"}
```

Command Response:

```
[
  {
    "success": {
      "username":
"ntyIC69JHC0h1SLMaS2IB0oJJay54ANoL1NfsI71"
    }
  }
]
```

デバッガの使い方の詳細な手順については、こちらをご覧ください: <https://developers.meethue.com/documentation/getting-started>

出力ペイロード	説明
<code>cv.Payload</code>	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明	例
On	On/Offの指定. <code>cp.On</code> に指定する値は "true" または "false"	
カラー	R,G,B,不透明度の順に00からFFまでの16進数でそれぞれの値を指定します	例.) FFFFFFF
カラーループ	色を一定時間ごとに切り替える	

IPアドレス	Philips HueのIPアドレス	
ユーザーID	Philips HueのユーザID	
ライトID	Philips HueのライトIDを指定します。Light ID の値は「1」から設定される番号になります。	

7.4.9.46. PostgreSQL Exec

PostgreSQL Execcコンポーネントは、PostgreSQLに対してSQL文を実行してデータを更新/削除することができます。

PostgreSQL 11以上に対応しています。

出力ペイロード

出力ペイロード	説明
<code>cv.Payload</code>	入力ペイロードに加えて実行したSQL文によりLastInsertIdまたはRowsAffectedを渡します

LastInsertIdは、実行したSQL文がINSERT文だったときに、DBによって生成されたIDが取得できた場合に返されます。

RowsAffectedは、実行したSQL文によって影響を受けたレコード数を返します。例えば、DELETE文だったときに削除された行数が戻ります。

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
バルクモード	SQL文の実行でバルクモードを使用するかを選択
バルクサイズ	バルクモードを使用する場合にそのサイズを指定します

ズ	
ホスト	PostgreSQLのホストを指定します
ポート	PostgreSQLのポートを指定します
ユーザ名	PostgreSQLのユーザ名を指定します
パスワード	PostgreSQLのパスワードを指定します
データベース	データベースを指定します
タイムゾーン	データベースのタイムゾーンを指定します
SSLMode	通信の暗号化を disable (SSL/TLS接続をしない)、allow (最初にSSL/TLS接続をしない接続を試し接続できなかったら、次にSSL/TLS接続を試す)、prefer (最初にSSL/TLS接続を試し接続できなかったら、次にSSL/TLS接続をしない接続を試す)、require (SSL/TLS接続をする。サーバー証明書は指定されたら検証するが、指定されなかったらしない)、verify-ca (SSL/TLS接続をし、サーバー証明書も検証する)、verify-full (SSL/TLS接続とサーバー証明書を検証し、かつサーバー名の検証も行う) から指定します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
UPDATE table1 SET name='更新名' WHERE id=$1
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを更新します。

プレースホルダーは2つ以上指定することもできます。

```
UPDATE table1 SET name=$1 WHERE id=$2
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

Text2SQL

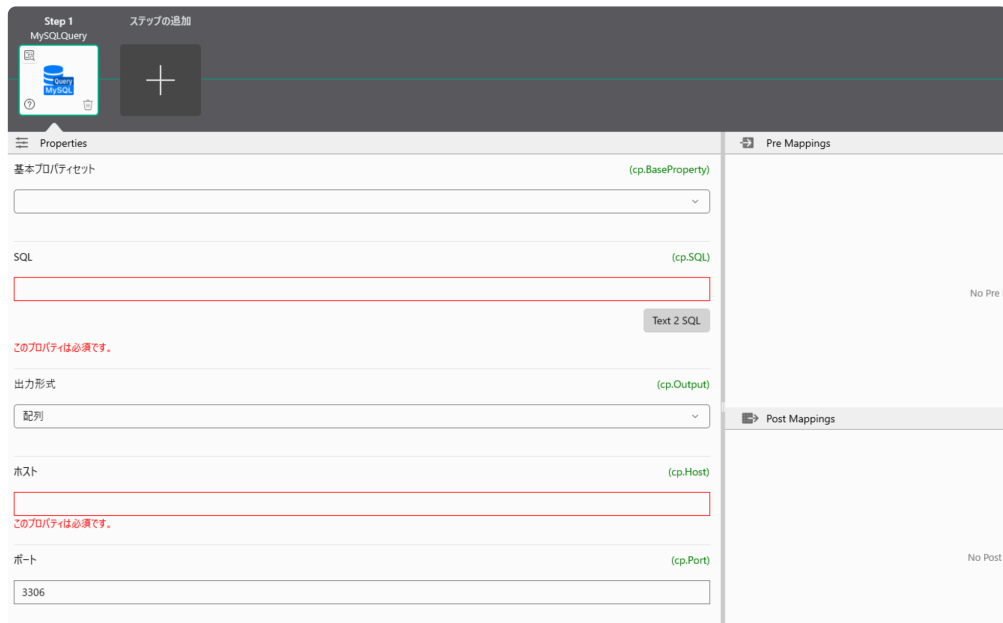
Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.47. PostgreSQL Query

PostgreSQL Queryコンポーネントは、PostgreSQLに対してクエリー文を実行してデータを取得することができます。

PostgreSQL 11以上に対応しています。



出力ペイロード

出力ペイロード	説明
<code>cv.Payload</code>	条件に一致した各レコードをプロパティで指定した出力形式でレコードを出力します。

配列形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「配列」を指定すると、以下のように出力ペイロードが生成されます。

出力例

```
[1, "abc"]  
[2, "def"]  
[3, "hij"]
```

オブジェクト形式の出力例

table1テーブルに、column1(整数型)、column2(文字列型)がある時に、SQLプロパティに `SELECT \* FROM table1` と指定し、出力形式プロパティに「オブジェクト形式」を指定すると、以下のように出力ペイロードが生成され

ます。

出力例

```
{ "column1": 1, "column2": "abc" }
{ "column1": 2, "column2": "def" }
{ "column1": 3, "column2": "hij" }
```

コンポーネントプロパティ

プロパティ名	説明
SQL	実行するSQL文を指定
出力形式	配列形式かオブジェクト形式を選択
ホスト	PostgreSQLのホストを指定します
ポート	PostgreSQLのポートを指定します
ユーザ名	PostgreSQLのユーザ名を指定します
パスワード	PostgreSQLのパスワードを指定します
データベース	データベースを指定します
タイムゾーン	データベースのタイムゾーンを指定します
SSLMode	通信の暗号化を disable (SSL/TLS接続をしない)、allow (最初にSSL/TLS接続をしない接続を試し接続できなかったら、次にSSL/TLS接続を試す)、prefer (最初にSSL/TLS接続を試し接続できなかったら、次にSSL/TLS接続をしない接続を試す)、require (SSL/TLS接続をする。サーバー証明書は指定されたら検証するが、指定されなかったらしない)、verify-ca (SSL/TLS接続をし、サーバー証明書も検証する)、verify-full (SSL/TLS接続とサーバー証明書を検証し、かつサーバー名の検証も行う) から指定します
ルート証明書	使用するルート証明書を指定します。AWSのAuroraまたはRDSを使用する場合にはAWSが発行したサーバー証明書を使用する必要があります。そのサーバー証明書のルート証明書となります

プレースホルダー

実行するSQL文にプレースホルダーを使用してcv.Payloadの値をSQL文に使用することができます。

```
SELECT * FROM table1 WHERE id=$1
```

? で指定したプレースホルダーにはcv.Payloadの値が使われます。cv.Payloadに1と入っていた時に、idが1のレコードを取得できます。

プレースホルダーは2つ以上指定することもできます。

```
SELECT * FROM table1 WHERE name=$1 AND age=$2
```

2つ以上のプレースホルダーに値を指定するためにcv.Payloadに配列で値を指定してください。

```
["First Last", 20]
```

Text2SQL

Text2SQLはOpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。

使い方は [こちら](#) を参照してください。

7.4.9.48. Sensor Data DB

SensorDataDBコンポーネントは、CoordinatorまたはHubKitに格納されているセンサデータを取得することができます。

CoordinatorまたはHubKitに格納されているSensorDataDBからデータをクエリーし、各レコードを1行毎に次のコンポーネントに渡します。

プロパティで指定された内容で、センサーDBを検索しマッチしたレコード毎にフィルター式を実行します。マッチしたレコードは“Value”変数で参照することが可能です。式の実行結果がtrueの時は、そのレコードは通過し出力ペイロード(cv.Payload)となります。

出力ペイロード	説明
cv.Payload	条件に一致した各レコードをJSON Object型をレコード毎に出力します。

出力例{

```

    "AreaId": "c753b341-c856-42fc-9170-69edaa5d8246",
    "AreaName": "オフィス",
    "Data": 460,
    "DataId": "f969129db906491e876cef43afb4fcb8",
    "DataType": "double",
    "KindId": "7e87a819-135e-40d3-9d5f-c0330f38ec4e",
    "KindName": "Gravio-CO2",
    "LayerId": "1efe01a1-41c7-4d21-bd98-d6bc87af74bd",
    "LayerName": "CO2",
    "PhysicalDeviceId": "DE-8D-06-FE-FF-57-B5-15",
    "PhysicalDeviceName": "オフィスCO2",
    "Timestamp": "2021-09-13T00:26:42.594348934Z",
    "VirtualDeviceId": "2ce70be6-257b-4854-9680-af1f02d575b4"
  
```

}

TimestampはJSON Object型の中では日時型として出力される。

各レコード毎に、コンポーネント変数(`cv.Id`)にレコードID、コンポーネント変数(`cv.Count`)に一致したレコードの最初のレコーとを1して、IDを順にインクリメントした番号が付与されます。

フィルター式例

```
Value.Data > 123
```

```
Value.Data.key1 == "abc" && Value.Data.key2 > 123
```

センサデータのフィールド

フィールド名	説明
HubId	HubId (Coordinatorからデータを取得する場合)
AreaId	エリアID
AreaName	エリア名
LayerId	レイヤーID
LayerName	レイヤー名
KindId	種別ID
KindName	種別名
TimeStamp	データの受信日時
Data	データ
DataType	データタイプ
DataId	データID
VirtualDeviceId	論理デバイスID
PhysicalDeviceId	センサーの物理デバイスID
PhysicalDeviceName	センサーの物理デバイス名

コンポーネントプロパティ

プロパティ名	説明
Coordinator	Coordinatorのデータを取得する場合に指定
HubId	HubId (Coordinatorからデータを取得する場合)を指定
エリア	エリア名を指定
レイヤー	レイヤー名を指定
デバイスID	デバイスIDを指定
From	取得する数(例 From Previousが10でタイムスケールがレコードの場合は最新の10件を取得)

Previous	
タイムスケール	取得する最新データの単位(レコード, ミリ秒, 秒, 分, 時, 日, 週, 月, 年から選択). <code>cp.Timescale</code> の値としては "Records", "Milliseconds", "Seconds", "Minutes", "Hours", "Days", "Weeks", "Months", "Years" のいずれか
全てのレコード	From Previousとタイムスケールの条件を無効にして全てのレコードを取得
時間を丸める	データを丸めて取得(下記を参照)タイムスケールがレコードの時は無効
フィルター式	Filterコンポーネントのように式でペイロードを出力させるかを選ぶようにします

時間を丸める

コンポーネントが実行された日時を指定されたタイムスケール(取得する最新データの単位)で丸め、From Previous(取得する数)で指定された値分遡ってデータを取得します。

例)

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 分、From Previousが 5の場合には 2021-01-20 12:15:00より5分遡ったデータ(2021-01-20 12:10:00～2021-01-20 12:15:00未満のデータ)を取得

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 時、From Previousが 24の場合には 2021-01-20 12:00:00より24時間遡ったデータ(2021-01-19 12:00:00～2021-01-20 12:00:00未満のデータ)を取得

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 日、From Previousが 1の場合には 2021-01-20 00:00:00より1日遡ったデータ(2021-01-19 00:00:00～2021-01-20 00:00:00未満のデータ)を取得

タイムスケールがレコードの時は時間を丸めるは無効となりレコード件数で取得されます。

例)

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが レコード、From Previousが 10の場合には 2021-01-20 12:15:15より10件遡ったデータを取得

7.4.9.49. Sensor Data DB If Zero

SensorDataDBIfZeroコンポーネントは1行も検索されなかった場合に入力ペイロード(`cv.Payload`)を出力ペイロード(`cv.Payload`)として出力します。1行でもレコードがあった場合は、何も出力しません。

プロパティで指定された内容で、センサーDBを検索しマッチしたレコード毎にフィルター式を実行します。マッチしたレコードは“Value”変数で参照することが可能です。式を実行した結果がtrueだった場合は通過しますが、1行でもレコードがあった場合に該当するので、出力ペイロード(`cv.Payload`)はありません。DBからの検索で0件でもSensorDataDBIfZeroは入力ペイロードを出力しますが、DBからの検索でなにかレコードがあっても式の実行結果が全てfalseでも入力ペイロード(`cv.Payload`)を出力します。

SensorDataDBIfZero (Sensor Data DB If Zero)

プロパティ

- エリアID: `(cp.AreaId)`
すべて
- レイヤーID: `(cp.LayerId)`
すべて
- デバイスID: `(cp.PhysicalDeviceId)`
- From Previous: `(cp.FromPrevious)`
1
このプロパティは必須です
- タイムスケール: `(cp.Timescale)`
レコード
このプロパティは必須です
- ☐ 全てのレコード `(cp.AllRecords)`
このプロパティは必須です
- ☐ 時間を丸める `(cp.TimeRounding)`
このプロパティは必須です
- フィルター式: `(cp.Filter)`

Pre Mappings

Post Mappings

出力ペイロード	説明
<code>cv.Payload</code>	1行も検索されなかった場合に入力ペイロードを出力ペイロードとして出力します

出力例{

```

{
  "AreaId": "c753b341-c856-42fc-9170-69edaa5d8246",
  "AreaName": "オフィス",
  "Data": 460,
  "DataId": "f969129db906491e876cef43afb4fcb8",
  "DataType": "double",
  "KindId": "7e87a819-135e-40d3-9d5f-c0330f38ec4e",
  "KindName": "Gravio-CO2",

```

```
{
  "LayerId": "1efe01a1-41c7-4d21-bd98-d6bc87af74bd",
  "LayerName": "CO2",
  "PhysicalDeviceId": "DE-8D-06-FE-FF-57-B5-15",
  "PhysicalDeviceName": "オフィスCO2",
  "Timestamp": "2021-09-13T00:26:42.594348934Z",
  "VirtualDeviceId": "2ce70be6-257b-4854-9680-af1f02d575b4"
}
```

TimestampはJSON Object型の中では日時型として出力されます。

各レコード毎に、コンポーネント変数(`cv.Id`)にレコードID、コンポーネント変数(`cv.Count`)に一致したレコードの最初のレコーとを1して、IDを順にインクリメントした番号が付与されます。

フィルター式例

```
Value.Data > 123
```

```
Value.Data.key1 == "abc" && Value.Data.key2 > 123
```

センサデータのフィールド

プロパティ名	説明
Areald	エリアID
AreaName	エリア名
LayerId	レイヤーID
LayerName	レイヤー名
KindId	種別ID
KindName	種別名
TimeStamp	データの受信日時
Data	データ
DataType	データタイプ
DataId	データID
VirtualDeviceId	論理デバイスID
PhysicalDeviceId	センサーの物理デバイスID
PhysicalDeviceName	センサーの物理デバイス名

コンポーネントプロパティ

プロパティ名	説明
エリア	エリア名を指定

レイヤー	レイヤー名を指定
デバイスID	デバイスIDを指定
From Previous	取得する数(例 From Previousが10でタイムスケールがレコードの場合は最新の10件を取得)
タイムスケール	取得する最新データの単位(レコード, ミリ秒, 秒, 分, 時, 日, 週, 月, 年から選択). <code>cp.Timescale</code> の値としては <code>"Records"</code> , <code>"Milliseconds"</code> , <code>"Seconds"</code> , <code>"Minutes"</code> , <code>"Hours"</code> , <code>"Days"</code> , <code>"Weeks"</code> , <code>"Months"</code> , <code>"Years"</code> のいずれか
全てのレコード	From PreviousとTimescaleの条件を無効にして全てのレコードを取得
時間を丸める	直近の秒, 分, 時, 日, 週, 月, 年で丸めた時刻からコンポーネントが実行される直前までのデータを取得
時間を丸める	データを丸めて取得(下記を参照)タイムスケールがレコードの時は無効
フィルター式	Filterコンポーネントのように式でペイロードを出力させるかを選べるようにします

時間を丸める

コンポーネントが実行された日時を指定されたタイムスケール(取得する最新データの単位)で丸め、From Previous(取得する数)で指定された値分遡ってデータを取得します。

例)

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 分、From Previousが 5の場合には 2021-01-20 12:15:00より5分遡ったデータ(2021-01-20 12:10:00~2021-01-20 12:15:00未満のデータ)を取得

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 時、From Previousが 24の場合には 2021-01-20 12:00:00より24時間遡ったデータ(2021-01-19 12:00:00~2021-01-20 12:00:00未満のデータ)を取得

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが 日、From Previousが 1の場合には 2021-01-20 00:00:00より1日遡ったデータ(2021-01-19 00:00:00~2021-01-20 00:00:00未満のデータ)を取得

タイムスケールがレコードの時は時間を丸めるは無効となりレコード件数で取得されます。

例)

コンポーネントが実行された日時が 2021-01-20 12:15:15、タイムスケールが レコード、From Previousが 10の場合には 2021-01-20 12:15:15より10件遡ったデータを取得

7.4.9.50. Slack Post Message

Slack Post Messageアクションコンポーネントは、SlackAPIを使用して、コンポーネントのプロパティにしてされたSlackチャンネルにメッセージをポストすることができます。アクションをデバッグしたり、人に通知したりすると便利です。#channelsと @people にメッセージを送信するためにも使用できます。

Slack APIのサイトからappを新規作成し、Featuresメニューの "OAuth & Permissions" にある "Scopes" に "chat:write:bot" を追加してください。

追加すると "OAuth Tokens & Redirect URLs" に "Install App to Workspace" ボタンが押せるようになりますので、Appをインストール（パーミッションの許可）を行ってください。"許可" すると "OAuth Access Token" のトークンが使用できるようになります。

コンポーネントのTokenプロパティには、このトークンを指定してください。

出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
チャンネル	Slackのチャンネルを指定
テキスト	送りたいメッセージテキストを指定
タイムアウト	送信タイムアウト時間の指定
トークン	Slackで設定されたOAuth Access Token

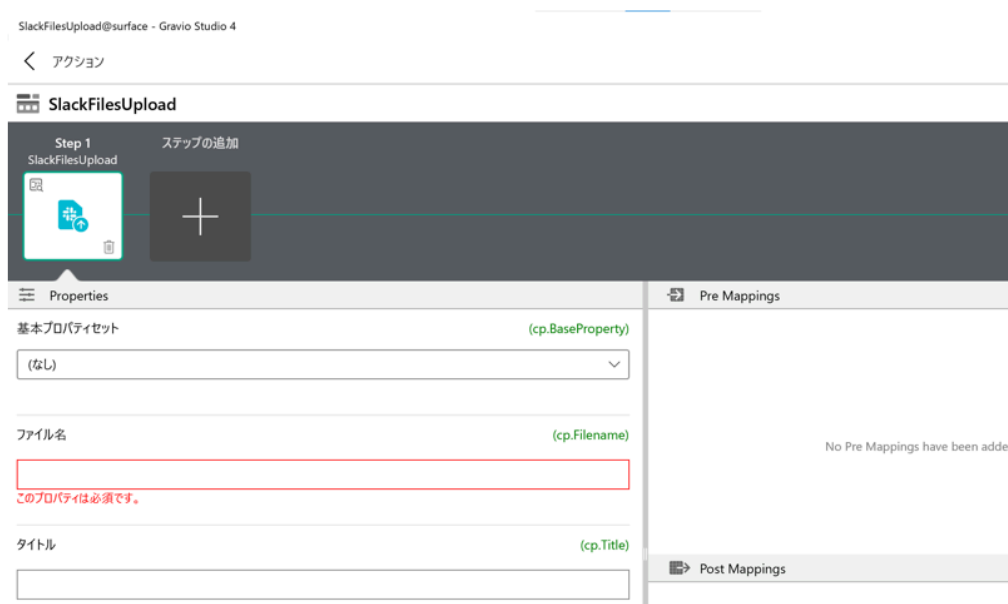
7.4.9.51. Slack Files Upload

Slack Files Uploadアクションコンポーネントは、SlackAPIを使用して、コンポーネントのプロパティにしてされたSlackチャンネルにメッセージとファイルをポストすることができます。アクションをデバッグしたり、人に通知したりすると便利です。`#channels`と `@people` にメッセージを送信するためにも使用できます。

Slack APIのサイトからappを新規作成し、Featuresメニューの "OAuth & Permissions" にある "Scopes" に "chat:write:bot" と "files:write:bot" を追加してください。

追加すると "OAuth Tokens & Redirect URLs" に "Install App to Workspace" ボタンが押せるようになりますので、Appをインストール（パーミッションの許可）を行ってください。"許可" すると "OAuth Access Token" のトークンが使用できるようになります。

コンポーネントのTokenプロパティには、このトークンを指定してください。



出力ペイロード	説明
<code>cv.Payload</code>	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

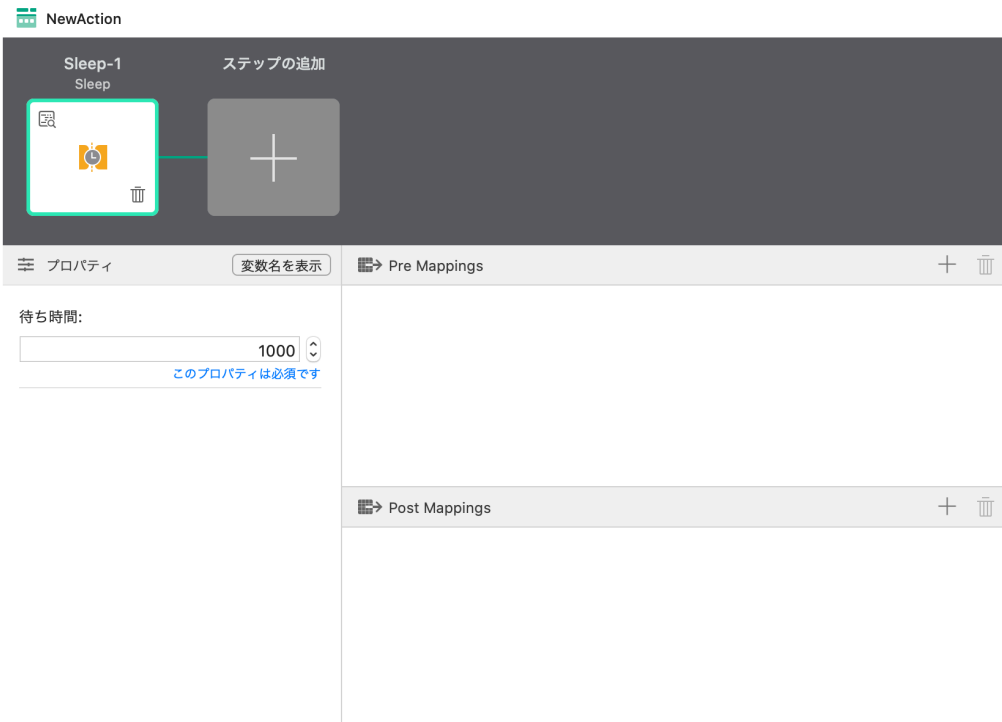
プロパティ名	説明
ファイル名	メッセージと一緒に送信するファイルを指定
タイトル	ファイルのタイトルを指定
チャンネル	Slackのチャンネルを指定
コメント	送りたいメッセージテキストを指定

タイムアウト	送信タイムアウト時間の指定
トークン	Slackで設定されたOAuth Access Token

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.52. Sleep

Sleepアクションコンポーネントでは、アクションの2つのステップの間にポーズを挿入できます。入力された数字は、アクションが一時停止する時間(ミリ秒)です。



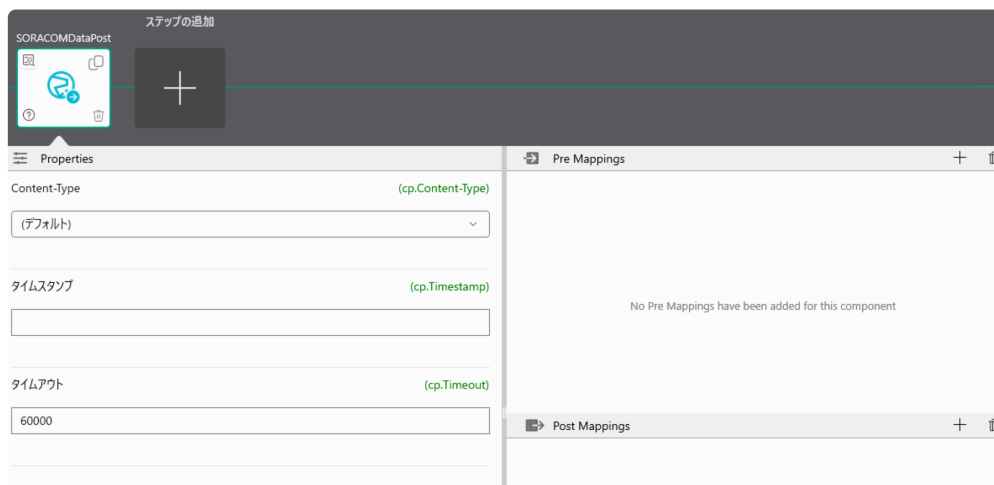
出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
待ち時間	ポーズする時間をミリ秒で指定

7.4.9.53. SORACOMDataPost

SORACOM Data Postコンポーネントは、SORACOMのHarvestにデータを送信することができます。



SORACOM側の事前設定

SORACOMの管理画面で対象のSIMカードをSIMグループに所属させ、その設定でHarvest Dataをオンにしてください。

※SORACOM AirのSIMカード、もしくは SORACOM Arcを利用してGravioからインターネットに接続する必要があります。

※SORACOMの管理画面の操作につきましてはSORACOMにお問い合わせください。

Content-TypeプロパティによるHTTPリクエスト時のBodyの内容

Content-Type プロパティ	HTTPリクエスト時のBodyの内容
デフォルト	入力ペイロードの型に応じてContent-Typeを自動で選びます。
文字列型	「text/plain; charset=utf-8」でPayloadを文字列としてHTTPリクエストを実行します。
JSON配列、 JSONオブジェクト	「application/json」として、Payloadをマーシャリングしたバイト列でHTTPリクエストを実行します。
バイト列	「application/octet-stream」にしてPayloadを文字列に変換したものをBodyとしてHTTPリクエストを実行します。
その他	「text/plain」にして文字列に変換したものをBodyとしてHTTPリクエストを実行します。
application/x-www-form-urlencoded	入力ペイロードはJSONオブジェクト型として、キーと値を「=」ではさみ、それぞれを&で区切った形でBodyにした上で、HTTPリクエストを実行します。（この場合で、JSON Object型以外のデータが入力としてきた場合はエラーとなります。）
multipart/form-	入力ペイロードはJSON Object型として、キーと値をマルチパートで区切り、Content-Typeの

data	multipart/form-dataはバウンダリを自動的に付加してHTTPリクエストを実行します。(この場合、JSON Object型以外のデータが入力としてきた場合はエラーとなります。)
それ以外の任意に入力されたContent-Type	入力されたContent-Typeで入力ペイロードをバイト列にしたものをBodyとして出力します。

入力と出力のペイロード

入力ペイロード	説明
cv.Payload	送信するデータ Harvestで利用できる形式はHarvestのドキュメントを参照ください

出力ペイロード	説明
cv.Payload	HTTPレスポンスのContent-Typeヘッダが“application/json”の場合はJSON型、“text/*”でcharset(utf-8/shift_jis/euc-jp/iso-2022-jp)が指定されている場合は文字列として、その他のContent-Typeヘッダの場合はバイト列のまま格納します。

コンポーネントプロパティ

プロパティ名	説明
Content-Type	HTTPヘッダのContent-Type
タイムスタンプ	データの日付 *1
タイムアウト	HTTPリクエストのレスポンスタイムアウト値

*1 サポートしている日付のフォーマットは下記のサポートしている日時フォーマットを参照

コンポーネント変数

コンポーネント変数	説明
cv.Payload	レスポンスのbody
cv.StatusCode	リクエスト結果
cv.Headers	レスポンスのヘッダー
cv.ContentLength	レスポンスの長さ

サポートしている日時フォーマット

Timestampは以下のフォーマットをサポートしています。

time.RFC3339形式

“2006-01-02T15:04:05.9999999Z”

“2006-01-02T15:04:05Z”

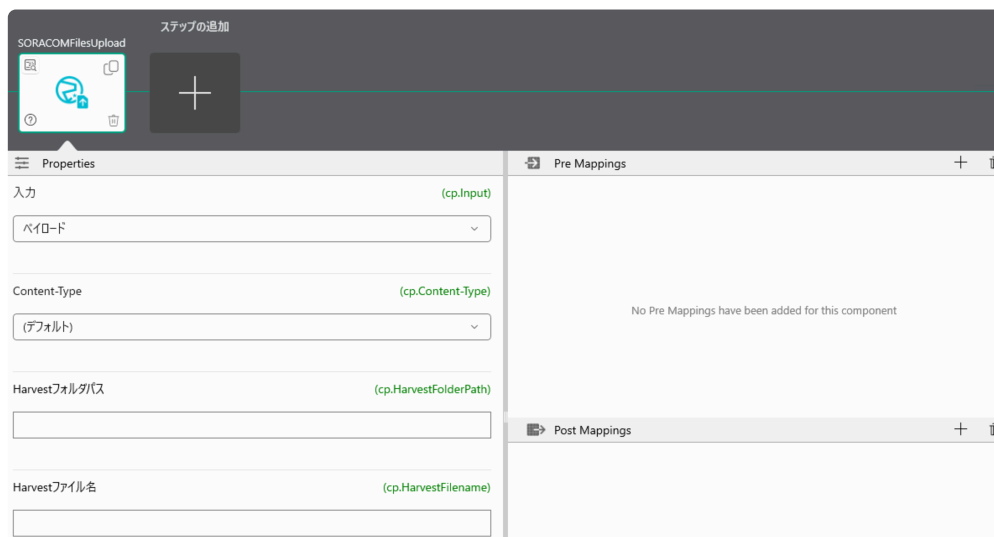
“2006-01-02 15:04:05.9999”

“2006-01-02 15:04:05”

“2006-01-02”

7.4.9.54. SORACOMFileUpload

SORACOM File Uploadコンポーネントは、SORACOMのHarvestにファイルを送信することができます。



SORACOM側の事前設定

SORACOMの管理画面で対象のSIMカードをSIMグループに所属させ、その設定でHarvest Filesをオンにしてください。

※SORACOM AirのSIMカード、もしくは SORACOM Arcを利用してGravioからインターネットに接続する必要があります。

※SORACOMの管理画面の操作につきましてはSORACOMにお問い合わせください。

入力と出力のペイロード

入力ペイロード	説明
<code>cv.Payload</code>	入力プロパティがペイロードの場合はデータ Harvestで利用できる形式はHarvestのドキュメントを参照ください

出力ペイロード	説明
<code>cv.Payload</code>	HTTPレスポンスのContent-Typeヘッダが“application/json“の場合はJSON型、“text/*“でcharset (utf-8/shift_jis/euc-jp/iso-2022-jp) が指定されている場合は文字列として、その他のContent-Typeヘッダの場合はバイト列のまま格納します。

コンポーネントプロパティ

プロパティ名	説明
入力	ペイロードまたはファイル名
ファイル名	入力プロパティがファイル名の場合に送信するファイル名
Content-Type	HTTPヘッダのContent-Type
Harvestフォルダ名	Harvest側の保存先フォルダ名。空の場合はrootフォルダで保存されます
Harvestファイル名	Harvest側の保存先ファイル名
タイムアウト	HTTPリクエストのレスポンスタイムアウト値

コンポーネント変数

コンポーネント変数	説明
cv.Payload	レスポンスのbody
cv.StatusCode	リクエスト結果
cv.Headers	レスポンスのヘッダー
cv.ContentLength	レスポンスの長さ

Harvestフォルダ名で使用出来るプレースホルダー

Harvestフォルダ名には以下のプレースホルダーが使用出来ます。

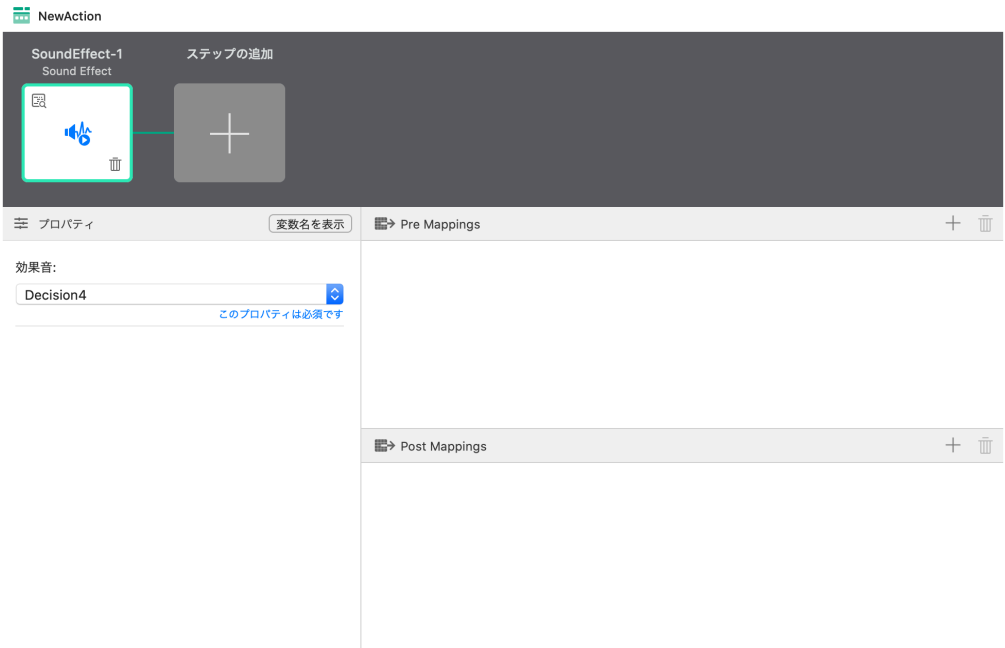
プレースホルダー	説明
:imsi:	IoT SIM の IMSI
:time:	ファイルを送信した UNIX 時間 (ミリ秒)

プレースホルダーの使い方に関して以下のリンクを参考にしてください。

<https://users.soracom.io/ja-jp/docs/harvest/enable-files/>

7.4.9.55. Sound Effect

効果音を鳴らすことができます。



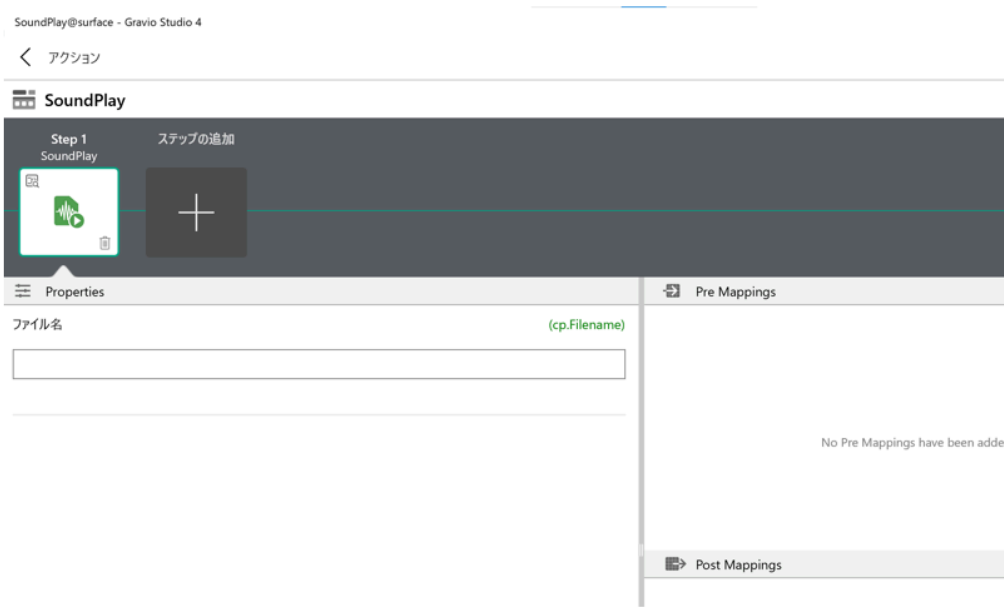
出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティ名	説明
効果音	効果音を指定. cp.Typeの値で設定する場合は "warning1", "warning2", "decision3", "decision4" のいずれか

7.4.9.56. Sound Play

指定したmp3ファイルを再生することができます。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

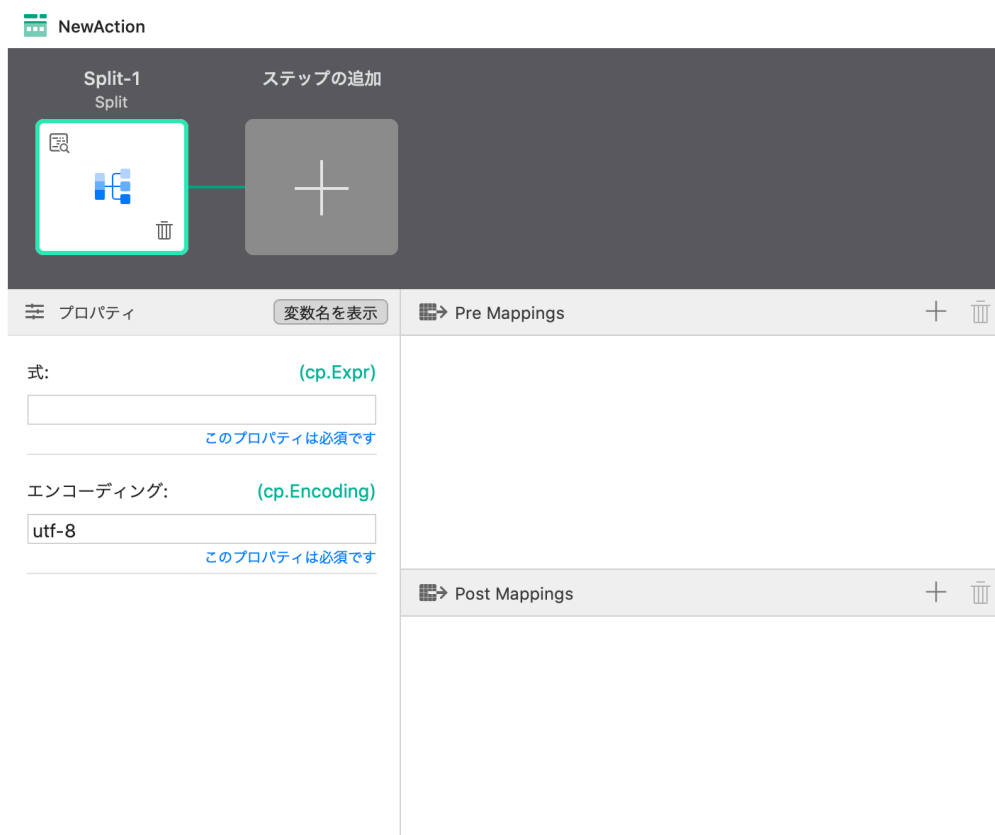
コンポーネントプロパティ

プロパティ名	説明
ファイル名	ファイル入力の場合のファイル名を指定

※ファイルパスを指定しないファイルの参照先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

7.4.9.57. Split

Splitコンポーネントは入力ペイロード(`cv.Payload`)が文字列なら改行毎に、JSON配列なら要素毎に分割して次のコンポーネントに渡すコンポーネントです。



Splitコンポーネントでは分割後の出力データを次にコンポーネントに渡すかどうかのフィルタ機能があります。条件を式で入力することができます。以下の2つの変数がこのコンポーネントの条件式内で利用できます。

変数名	変数の意味
Value	分割後の出力データ
Index	何番目の出力データかを表す

条件例	式例
abcで始まる行 (JSON配列の場合は要素) だけを次のコンポーネントに渡す	<code>Value =~ "^abc"</code>
偶数番目の行だけを渡す	<code>Index % 2 == 0</code>

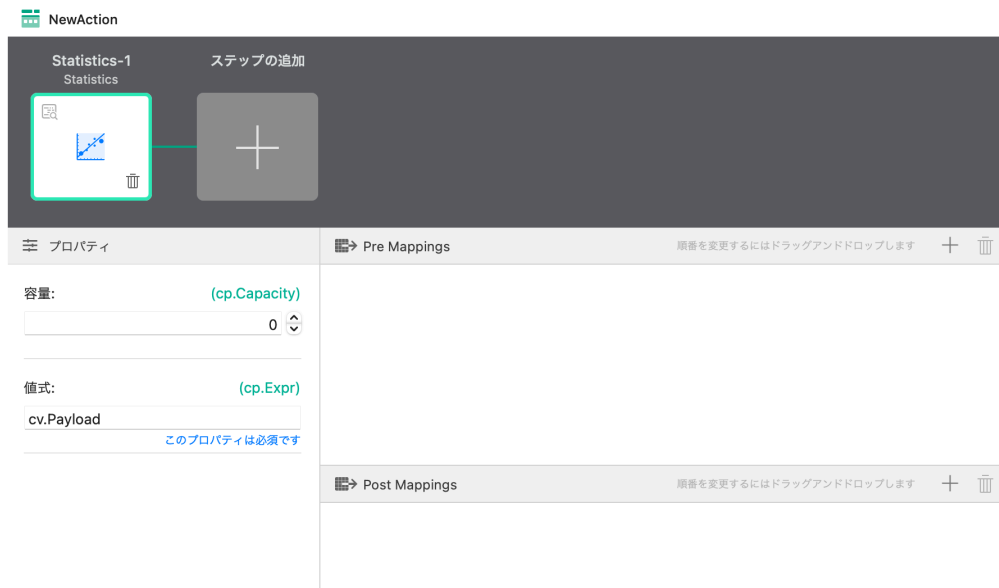
コンポーネントプロパティ

プロパティ名	説明
条件式	条件式を記述 (条件は正規表現の記述も可能)

エンコーディング	文字エンコーディングを指定
----------	---------------

7.4.9.58. Statistics

Statisticsコンポーネントは、指定された値で統計(最低値(Min)、最高値(Max)、平均(Mean/Average)、中央値(Median)、標準偏差(StandardDeviation)、合計(Sum)、最頻値(Mode))をコンポーネント変数に付加します。



指定された値が数値として不適正な値だった場合には、その値は統計から除外されます。除外されずに統計に入った値の数はCountとしてコンポーネント変数に出力されます。

例1

テストの点数がペイロードとして流れてくる場合、Exprに「cv.Payload」で容量に「0」を指定したとします。10個の値が流れてきて終了した時には、Statisticsはペイロードとして値を配列にして出力にします。同時に、Statisticsコンポーネント変数内に統計データを出力します。

```
cv.Payload [100 90 85 80 75 70 65 30 10 0]
cv.Statistics {"Average":60.5, "Count":10, "Max":100, "Mean":60.5, "Median":72.5, "Min":0, "Mode":[], "StandardDeviation":32.973474187595095, "Sum":605}
```

値を指定する値式は、数値計算式を記述することができます。

上記の例で、「cv.Payload * 100」と指定した場合にはそれぞれの値が100倍された値として統計がとられます。1つの値から複数の統計を取りたい場合や、JSON Object型でその中から複数の値を抽出し統計を取りたい場合など、一度に複数の値で統計を取りたい場合には値式を配列で値を指定します。

```
[cv.Payload, cv.Payload*100]
{"val1":123, "val2":456}というJSON Object型が流れてくるとして
[cv.Payload.val1, cv.Payload.val2]
という指定も可能です。
```

値式に配列を指定した場合は、コンポーネント変数Statisticsも配列になり、それぞれの統計がとられます。

SensorDataDBコンポーネントでセンサー値で統計をとるには、値式プロパティに「cv.Payload.Data」と指定することで、センサーデータの値をもとに統計データを作成できます。

容量プロパティの数を超えてデータが送られてくる場合は、容量に到達したら、そこまでのデータはクリアされま

す。
例えば、容量を 10 に設定して、データが12個来た場合、1回目は1～10までのデータで統計を取り、そのあとこれら10個のデータは捨てられ、11-12個目のデータで計算されます。

例2

SensorDataDBコンポーネントからデータを抽出して Statisticsコンポーネントで統計データを作成する場合

例ではSensorDataDBコンポーネントで3行のレコードがマッチしたとします。

SensorDataDBコンポーネントは3つの出力Payloadを分けて、順に次のStatisticsコンポーネントに渡します。

```
1. {Data:123, Layer:"abc"}
2. {Data:456, Layer:"abc"}
3. {Data:789, Layer:"abc"}
```

容量が0で、値式に `cv.Payload.Data` と指定すると `[123, 456, 789]` で様々な統計が計算されますが、出力Payloadは `[{Data:123, Layer:"abc"}, {Data:456, Layer:"abc"}, {Data:789, Layer:"abc"}]` となります。

容量が2の場合は、1度目は `[123, 456]` で統計が計算され、出力Payloadは `[{Data:123, Layer:"abc"}, {Data:456, Layer:"abc"}]` となり、2度目は `[789]` で統計が計算され、出力Payloadは `[{Data:789, Layer:"abc"}]` と2回の出力Payloadを次のコンポーネントに渡します。

出力ペイロード・変数	説明
<code>cv.Payload</code>	容量=0の場合は、前のコンポーネントの出力ペイロードが配列として出力されます。容量=0以外場合は、前のコンポーネントの出力ペイロードが容量分がまとめられて配列として出力されます。
<code>cv.Statistics.Average</code>	平均
<code>cv.Statistics.Count</code>	除外されずに統計に入った値の数
<code>cv.Statistics.Max</code>	最高値
<code>cv.Statistics.Mean</code>	平均
<code>cv.Statistics.Median</code>	中央値
<code>cv.Statistics.Min</code>	最低値
<code>cv.Statistics.Mode</code>	最頻値

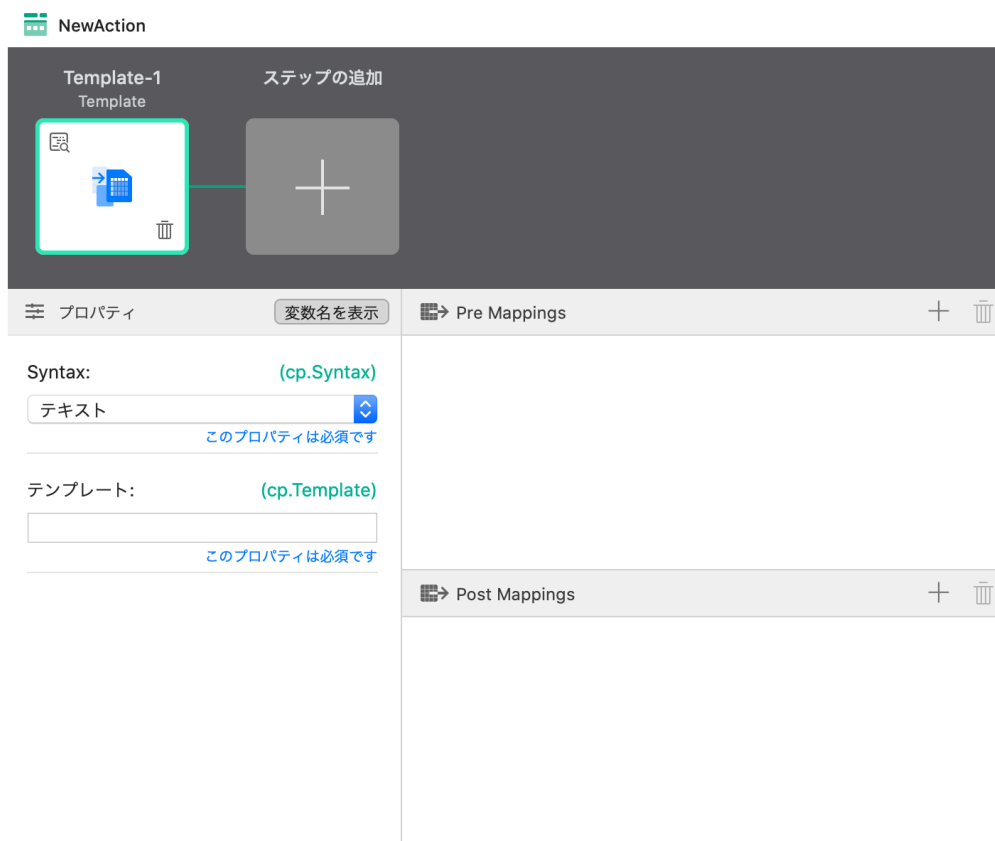
<code>cv.Statistics.StandardDeviation</code>	標準偏差
<code>cv.Statistics.Sum</code>	合計

コンポーネントプロパティ

プロパティ名	説明
容量	指定された数までためて統計を出力する
値式	統計を取る値の式

7.4.9.59. Template

Templateコンポーネントは、指定されたテンプレートから文字列(テキスト)を生成するコンポーネントです。



テンプレート構文としては <https://golang.org/pkg/text/template/> で説明されている構文が利用できます。
`tp. / tv. / ap. / av. / cp. / cv.` は全てテンプレートから参照可能なので、テンプレート内でこれらの変数が参照されていれば、その変数の値で置換されます。
 またTemplateコンポーネント内で各変数への参照は先頭に「.」を追加します。

例

テンプレート	説明
<code>{{.cv.Payload.LastName}} {{.cv.Payload.FirstName}}</code>	それぞれの変数の値で中括弧ないが置換される

出力ペイロード	説明
<code>cv.Payload</code>	テンプレート文字列で変数部分が変数の値で置換された内容で生成された文字列(テキスト)

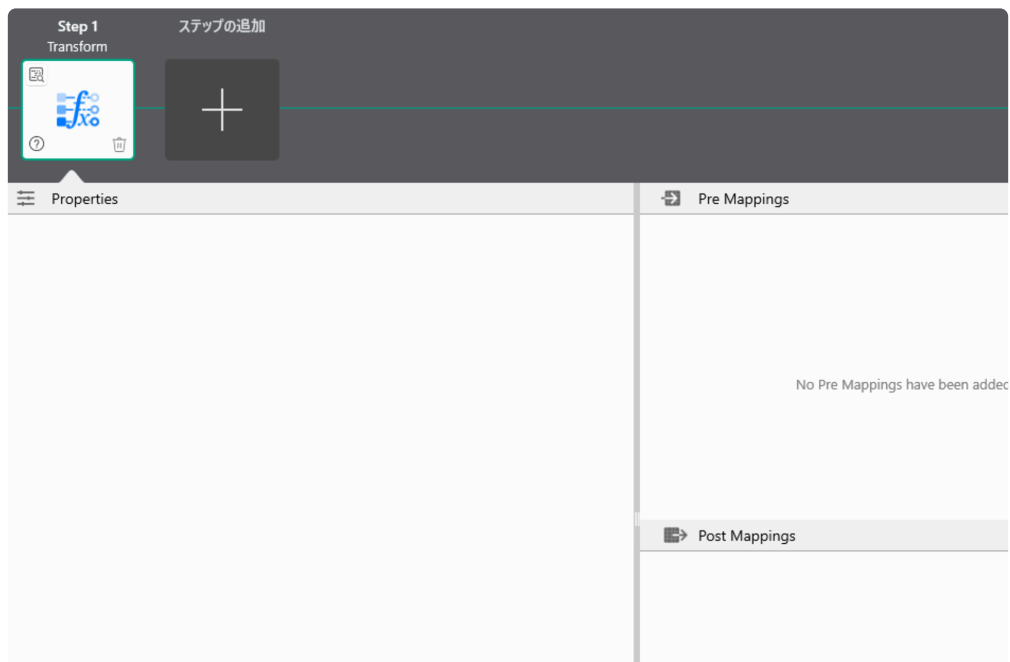
コンポーネントプロパティ

プロパティ名	説明
Syntax	テキストまたはHTMLを指定. <code>cp.Syntax</code> の値としては <code>"text"</code> , <code>"html"</code> のいずれか

テンプレート	置換される変数を含む文字列
--------	---------------

7.4.9.60. Transform

TransformコンポーネントはPre MappingとPost Mappingを処理するコンポーネントです。
入力と出力のデータを変換する時に使用する専用のコンポーネントです。



出力ペイロード	説明
cv.Payload	前のコンポーネントの出力ペイロードをそのまま出力します。(パススルー)

コンポーネントプロパティ

プロパティはありません。

7.4.9.61. Twilio SMS

Twilio SMSアクションコンポーネントは、Twilioサービス (<https://www.twilio.com/>) を使用して、コンポーネントのプロパティに指定されたメッセージを携帯電話のSMSにポストすることができます。センサーからの情報を人に通知したりすると便利です。

コンポーネントプロパティ

プロパティ名	説明	例
To	送信先の電話番号。国番号から始めます (日本の場合 81)	例) 8108012345678
From	発信元の電話番号。国番号から始めます (アメリカの場合 1)	日本の番号ではSMSを送ることはできません
メッセージ	SMSに投稿するメッセージ文字列	例) "現在の気温は" + <code>cv.Payload</code> + "°Cです"
Timeout	APIのタイムアウト	
アカウントSID	TwilioサービスのアカウントSID	
AuthToken	TwilioサービスのAUTHTOKEN	

7.4.9.62. Write To Excel (365)

WriteToExcel365コンポーネントは、データをExcel形式でMicrofost365に出力することができます。
このコンポーネントを使用するには、Microfost365 for Businessの契約と事前設定が必要です。

事前設定は [こちら](#) を参照してください。

Step 1 WriteToExcel365

ステップの追加

Properties

File Name (cp.FileName)

このプロパティは必須です。

Sheet Name (cp.SheetName)

Sheet1

Write Mode (cp.WriteMode)

Append

☒ Output Header (cp.OutputHeader)

出力フィールド順 (cp.OutputKeys)

Pre Mappings

Post Mappings

コンポーネントプロパティ

ファイル名	出力するExcelのファイル名を指定
シート名	データを出力するシート名を指定
書込モード	ファイルを上書き(Replace)するか、データを追記(Append)するかを指定. cp.WriteModeの値としては、"Append", "Overwrite" のいずれか
出力ヘッダー	データを出力するときにヘッダも出力するかを選択
出	ステップ出力のカラムのうち、Excelのどの列に出力するかを指定します。複数指定可能です。(例: 出力ペイロ

カ フ ィ ー ル ド 順	ードのaをエクセルのA列、cをエクセルのB列に出力する場合は「Payload.a:A,Payload.c:C」とステップ出力のフィールドとエクセルの列は ":" で指定して、複数指定する場合には "," カンマで区切ります)
ス タ ー ト 列	出力する最初の行を指定します。
ク ラ イ ア ン ト Id	アプリケーションIDを指定
テ ナ ン ト Id	ディレクトリ(テナント)IDを指定
イ ン ス タ ン ス	(変更しません)
ユ ー ザ ー 名	Microsoft365 Businessのアカウントを指定
パ ス ワ ー ド	Microsoft365 Businessのアカウントのパスワードを指定

7.4.9.63. Write To Excel (Local)

WriteToExcelLocalコンポーネントは、データをExcel形式でxlsxファイルに出力することができます。

入力ペイロード(`cv.Payload`)はJSON配列の配列か、JSON Objectの配列での入力を受け付けます。それ以外の場合は文字列に変換して、1行、1カラムのExcelファイルとして出力します。

入力ペイロード	例
JSON配列ペイロードの例	<pre>[[123, "abc", "xxx"], [456, "def", "yyy"], [789, "hij", "zzz"]]</pre>
JSON Objectのペイロードの例	<pre>[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]</pre>

Step 1 WriteToExcelLocal ステップの追加

Properties

ファイル名 (cp.Filename)

☐ 追加 (cp.Append)

Start Column (cp.StartColumn)

A

Start Row (cp.StartRow)

1

出力フィールド順 (cp.OutputKeys)

Pre Mappings

Post Mappings

コンポーネントプロパティ

プロパティ名	説明
ファイル名	ファイル出力の場合のファイル名を指定
追加	ファイルを上書き(Replace)するか、データを追記(Append)するかを指定. <code>cp.WriteMode</code> の値としては、"Append", "Overwrite" のいずれか

シート名	データを出力するシート名を指定
スタート列	出力する最初の列を指定します。
スタート行	出力する最初の行を指定します。
出力ヘッダー	データを出力するときにヘッダも出力するかを選択
出力フィールド順	入力ペイロードがJSON Objectの場合、出力するフィールド名を出力させたい順に、(カンマ)で区切り列挙します

※ファイルパスを指定しないファイルの保存先はactmgr/dataとなりますが、ファイルパスを指定するファイル名の書き方は [こちら](#) を参照してください。

出力フィールド順の指定方法

JSON Objectのペイロードが下記の場合に、出力フィールド順に field3, field1 と指定するとExcelファイルが出力されます。

・JSON Objectのペイロード

```
[{"field1": 123, "field2": "abc", "field3": "xxx"}, {"field1": 456, "field2": "def", "field3": "yyy"}, {"field1": 789, "field2": "hij", "field3": "zzz"}]
```

・出力ペイロード(Excelファイル)

```
field3, field1
xxx, 123
yyy, 456
zzz, 789
```


7.4.10. 関数解説

Pre Mappings / Post Mappingsの中で記述する式の右辺とトリガーの式で利用できる関数を個別に解説します。

7.4.10.1. 文字列

以下の関数が利用できます。

関数名	引数	説明	例
Len	Len(s)	文字列の長さ(バイト数)を返す。sが配列やObjectの場合は要素数を返す。	Len("abc012漢字") -> 11
Contains	Contains(s, substr)	sの中にsubstrが含まれているかBoolで返す	Contains("abc012漢字", "漢字") -> true
RuneCount	RuneCount(s)	sの文字列の文字数を返す	RuneCount("abc012漢字") -> 8
HasPrefix	HasPrefix(s, prefix)	sがprefixで始まるかBoolで返す	HasPrefix("https://www.asteria.com/", "https://") -> true
HasSuffix	HasSuffix(s, suffix)	sがsuffixで終わるかBoolで返す	

f i x			
F i r s t I n d e x	FirstIndex(s, substr)	sの中で最初に現れるsubstrの位置を返す。ない場合は-1	FirstIndex(“abc012漢字”, “漢字”) -> 6
J o i n	Join(a, sep)	aの配列要素をsepを挟んで1つの文字列にする。	Join([“abc”, “def”], “-”) -> “abc-def”
L a s t I n d e x	LastIndex(s, substr)	sの中で最後に現れるsubstrの位置を返す。ない場合は-1	
R e p e a t	Repeat(s, count)	sをcount回繰り返した文字列を返す	Repeat(“abc012漢字”, 3) -> “abc012漢字abc012漢字abc012漢字”
R e p l a c e	Replace(s, old, new[, n])	sの中でold文字列をnew文字列にn回目まで置換する。nが0より小さい時は全て(デフォルトは-1)。	
S p l i	Split(s, sep[, n])	sをsepで最大n個の要素に区切り配列で返す。nが0より小さい時は全て(デフォルトは-1)	Split(“a,b,c”, “,”) -> [“a”, “b”, “c”]

i t			
T o L o w e r	ToLower(s)	sを小文字に変換する	
T o U p p e r	ToUpper(s)	sを大文字に変換する	
T r i m	Trim(s[, cutset])	sの頭と最後のcutsetに含まれる文字を除いた文字列を返す。cutsetはデフォルトはスペース文字	Trim(" abc ") -> "abc"
T r i m L e f t	TrimLeft(s[, cutset])	sの頭cutsetに含まれる文字を除いた文字列を返す。cutsetはデフォルトはスペース文字	
T r i m R i g h t	TrimRight(s[, cutset])	sの最後のcutsetに含まれる文字を除いた文字列を返す。cutsetはデフォルトはスペース文字	
T r i m P r e f i x	TrimPrefix(s, prefix)	sがprefixではじまっていたら、それを取り除いた文字列を返す。始まっていない場合はsをそのまま返す	TrimPrefix("https://www.asteria.com", "https://") -> "www.asteria.com"

r e f i x			
T r i m S u f f i x	TrimSuffix(s, suffix)	sがsuffixでおわっていたら、それを取り除いた文字列を返す。終わっていない場合はsをそのまま返す	
J S O N P a t h	JSONPath(path, s)	sからpathで指定して値を文字列で返す。	JSONPath("\$.store.book[*].author", cv.Payload)
S p r i n t f	Sprintf(format, s)	書式付の文字列を整形し文字列を返します。書式は%に続いて変換指定子とオプションでフラグ、幅、精度で指定できます	Sprintf("メッセージ件数: %d", ToInt(cv.Payload))
I F S	IFS (条件1, 値1[, 条件2, 値2, ...])	条件を1から順に評価し合致した値を返す。条件にtrueと指定すると必ず合致するので、最後の条件にtrueを指定すると、それ以外すべてという条件を指定できます。	IFS (ToFloat(cv.Payload) >= 10, "10以上です", true, "10未満です")
L e f t	Left(s, len)	sの先頭からlen文字数を取り出す。lenは漢字などの文字も1文字と数える	
R i g h t	Right(s, len)	sの末尾からlen文字数を取り出す。lenは漢字などの文字も1文字と数える	

t			
M i d	Mid(s, start, len)	sのstart位置からlen文字数を取り出す。startは先頭の位置を1とし、漢字などの文字も1文字とみなし位置を指定します。lenは漢字などの文字も1文字と数える	
S u b s t r i n g	Substring(s, start[, end])	sのstart位置からend位置までの文字列を取り出す。start/endは先頭の位置を0とし、漢字などの文字も1文字とみなし位置を指定する	
S l i c e	Slice(s[, start[, end]])	sのstart位置からend位置までの文字列を取り出す。start/endは先頭の位置を0とし、漢字などの文字も1文字とみなし位置を指定する。start/endに負の値を指定した場合は末尾から位置指定とみなす	

7.4.10.1.1. JSONPath()の使用例

JSONPath()の使用例を解説します。

センサーデータ(@cv.Payload@)やトリガー条件で実行するアクションの引数(@tv.Data@)がJSON形式の場合にJSONPath()を使用して値を参照することができます。

サンプルデータ

```
{ "store": {  
  "book": [  
    { "category": "reference",  
      "author": "Nigel Rees",  
      "title": "Sayings of the Century",  
      "price": 8.95  
    },  
    { "category": "fiction",  
      "author": "Evelyn Waugh",  
      "title": "Sword of Honour",  
      "price": 12.99  
    },  
    { "category": "fiction",  
      "author": "Herman Melville",  
      "title": "Moby Dick",  
      "isbn": "0-553-21311-3",  
      "price": 8.99  
    },  
    { "category": "fiction",  
      "author": "J. R. R. Tolkien",  
      "title": "The Lord of the Rings",  
      "isbn": "0-395-19395-8",  
      "price": 22.99  
    }  
  ],  
  "bicycle": {  
    "color": "red",  
    "price": 19.95  
  }  
}
```

例えば store -> bicycle -> color を参照する場合には

```
cv.Payload.store.bicycle.color
```

と記述することができますがJSONPath()を使用した場合には

```
JSONPath("$.store.bicycle.color", cv.Payload)
```

と記述することもできます。

JSONPathの第一引数 path

path	説明
\$	ルート オブジェクト/エレメント
@	カレント オブジェクト/エレメント
. or []	子オペレーター
..	再帰下降
*	ワイルドカード
[]	添え字演算子（ネイティブの配列演算子）
[,]	ユニオン演算子（代替名または配列インデックスをセットとして使用できます）
[start:end:step]	配列スライス演算子
?()	フィルタ（スクリプト）式
()	基盤となるスクリプトエンジンを使用したスクリプト式

JSONPathの第一引数 path の例

例	説明
\$.store.book[*].author	store内のbook配列のすべての author
\$..author	すべてのauthor
\$.store.*	store内のbook配列とbicycle
\$.store..price	store内のすべてのprice
\$..book[2]	3番目のbook
\$..book[(@.length-1)]\$..book[-1:]	最後のbook
\$..book[0,1]\$..book[:2]	book配列の最初の2つのbook
\$..book[?(@.isbn)]	book配列のすべてのisbn
\$..book[?(@.price<10)]	book配列のpriceが10より小さいもの
\$..*	すべてのノード/エレメント

JSONPathの詳細は引用元の [こちら](#) を参照してください。

7.4.10.1.2. Sprintfの使用方法

Sprintf

書式付の文字列を整形し文字列を返します。書式は%に続いて変換指定子とオプションでフラグ、幅、精度で指定できます。

変換指定子	説明
%b	整数2進数
%d	整数10進数
%x %X	16進数
%e %E	浮動小数、科学的表記法
%f %F	浮動小数、指数のない表記
%g %G	浮動小数、指数が大きい場合は%e、それ以外が%f
%s	文字列
%%	%そのものの文字

フラグ	説明
+	符号つき
-	左寄せ
0	幅未満のときはスペースではなく0を埋める

幅、精度	説明
%f	デフォルトの幅と精度
%9f	幅9
%.2f	精度2
%9.2f	幅9、精度2

例

```
Sprintf("メッセージ件数: %d",.ToInt(cv.Payload))
```

```
Sprintf("浮動小数の小数点第二位まで: %.2f", cv.Payload)
```

7.4.10.2. 型変換

以下の関数が利用できます。

関数名	引数	説明
ToBool	ToBool(a)	Bool型へ変換
ToInt	ToInt(a)	整数型へ変換
ToFloat	ToFloat(a)	浮動小数型へ変換
ToString	ToString(a)	文字列型へ変換
ToDate	ToDate(a)	日時型へ変換
ToBinary	ToBinary(a)	バイト列へ変換
ToJSON	ToJSON(a)	JSON配列型またはJSON Object型へ変換

7.4.10.3. エンコード

以下の関数が利用できます。

関数名	引数	説明	例
URLPathEscape	URLPathEscape(s)	sをURLパスセグメントに安全に配置できる文字列にエスケープする	URLPathEscape("test 1?test2 test3") -> "test1%3Ftest2%20test3"
URLPathUnescape	URLPathUnescape(s)	URLPathEscapeでエスケープされた文字列をもとに戻す	
URLQueryEscape	URLQueryEscape(s)	sをURLクエリに安全に配置できる文字列にエスケープする	URLQueryEscape("test 1?test2 test3") -> "test1%3Ftest2+test3"
URLQueryUnescape	URLQueryUnescape(s)	URLQueryEscapeでエスケープされた文字列をもとに戻す	
MD5	<u>MD5</u>	バイト列bのMD5を返す	
SHA1	<u>SHA1</u>	バイト列bのSHA1を返す	
SHA256	<u>SHA256</u>	バイト列bのSHA256を返す	
BASE64	<u>BASE64</u>	バイト列bのBASE64を返す。padでtrueを指定すると=パディングが行われる(デフォルトはtrue)	
DecodeBASE64	DecodeBASE64(s[, pad])	BASE64sをデコードする。padでtrueを指定すると=パディングが行われたBASE64と解釈する(デフォルトはtrue)	
BASE64URL	<u>BASE64URL</u>	バイト列bをURLまたはファイル名で使用されるBASE64を返す。padでtrueを指定すると=パディングが行われる(デフォルトはtrue)	
DecodeBASE64URL	DecodeBASE64URL(s[, pad])	BASE64sをデコードする。padでtrueを指定すると=パディングが行われたBASE64と解釈する(デフォルトはtrue)	

7.4.10.4. 日付

以下の関数が利用できます。

関数名	引数	説明	例
Now	Now([timezone])	今現在の日時の日時型を返す	
Year	Year(t[, timezone])	日時型tの年を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Month	Month(t[, timezone])	日時型tの月を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Day	Day(t[, timezone])	日時型tの日を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Hour	Hour(t[, timezone])	日時型tの時を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Minute	Minute(t[, timezone])	日時型tの分を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Second	Second(t[, timezone])	日時型tの秒を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。	
Weekday	Weekday(t[, timezone])	日時型tの曜日を返す。timezoneが指定されたら、そのタイムゾーンとして解釈する。(0は日曜日)	
ToLocal	ToLocal(t)	日時型tをシステムタイムゾーンに変換する	
ToUTC	ToUTC(t)	日時型tをUTCに変換する	
ToTimezone	ToTimezone(t, timezone)	日時型tを指定されたタイムゾーンに変換する	ToTimezone(Now(), "America/New_York")

ne			
Date Format	DateFormat(t, layout)	日時型tを指定されたレイアウトで文字列に変換する	<code>DateFormat (Now(), "02 Jan 06 15:04 JST")</code>
Date Parse	DateParse(layout, s[, timezone])	sをlayoutに沿って解釈し日時型を返す	
Add Duration	AddDuration(t, s)	日時型tに指定されたsを加算する。sが整数の場合にはミリ秒とし、文字列の場合は整数に単位を指定する。単位は時“h”、分“m”、秒“s”、ミリ秒“ms”、マイクロ秒“us”、ナノ秒“ns”が指定できます	<code>AddDuration (Now(), 100) AddDuration (Now(), "1 h")</code>
Add Date	AddDate(t, y, m, d)	日時型tに指定されたy(年)、m(月)、d(日)を加算する	<code>AddDate (Now(), 0, 0, 7)</code>

DateFormat, DateParseで指定するレイアウトの例

形式	レイアウト
Layout	“01/02 03:04:05PM '06 +0900”
ANSIC	“Mon Jan _2 15:04:05 2006”
UnixDate	“Mon Jan _2 15:04:05 JST 2006”
RubyDate	“Mon Jan 02 15:04:05 +0900 2006”
RFC822	“02 Jan 06 15:04 JST”
RFC822Z	“02 Jan 06 15:04 +0900”
RFC850	“Monday, 02-Jan-06 15:04:05 JST”
RFC1123	“Mon, 02 Jan 2006 15:04:05 JST”
RFC1123Z	“Mon, 02 Jan 2006 15:04:05 +0900”
RFC3339	“2006-01-02T15:04:05Z07:00”
RFC3339Nano	“2006-01-02T15:04:05.999999999Z07:00”
Generic	“20060102”

詳細は [こちら](#) を参照してください。

・使用例

データを保存するファイル名を現在の7日前の日付で作成する

```
DateFormat(AddDate(Now("Asis/Tokyo"),0,0,-7),"20060102")+".CSV"
```

7.4.10.5. 環境変数

以下の関数が利用できます。

関数名	引数	説明
<code>Env</code>	<code>Env(s)</code>	環境変数を返す

7.4.10.6. 正規表現

以下の関数が利用できます。

関数名	引数	説明	例
RegExpMatch	RegexMatch(re, s)	正規表現reとsがマッチするかBoolで返す	RegexMatch("G.*o", "Gravio HubKit") -> true
RegExpFind	RegexFind(re, s)	正規表現reとsがマッチした最初の文字列を返す	RegexFind("i.", "Gravio HubKit") -> "io"
RegExpFindSubmatch	RegexFindSubmatch(re, s)	正規表現reとsがマッチした最初の文字列と括弧で囲まれた部分式があれば配列で返す	RegexFindSubmatch("(G.*o)(H.*t)", "Gravio HubKit") -> ["Gravio HubKit", "Gravio", "HubKit"]
RegExpFindAll	RegexFindAll(re, s[, n])	正規表現reとsがマッチした全ての文字列を配列で返す	RegexFindAll("i.", "Gravio HubKit") -> ["io", "it"]
RegExpReplace	RegexReplace(re, s, repl)	正規表現reとsがマッチしたreplに置換した文字列を返す。replでは\$1のような部分式を使用できる。	
UUID	UUID	UUID文字列を返す。sepはセパレータ(デフォルトは空文字列)	

7.4.10.7. 算術系関数

以下の関数が利用できます。

関数名	引数	説明	例
Abs	Abs(v)	絶対値を返します	
Acos	Acos(v)	逆余弦を返します	
Acosh	Acosh(v)	逆双曲線余弦を返します	
Asin	Asin(v)	逆正弦を返します	
Asinh	Asinh(v)	逆双曲線正弦を返します	
Atan	Atan(v)	逆正接を返します	
Atan2	Atan2(y, x)	y/xの逆正接を返します	
Atanh	Atanh(v)	逆双曲線正接を返します	
Cbrt	Cbrt(v)	立方根を返します	
Ceil	Ceil(v)	小数点以下を切り上げた値を返します	
Copysign	Copysign(x, y)	大きさxで符号yの値を返します	
Cos	Cos(v)	余弦を返します	
Cosh	Cosh(v)	双曲線余弦を返します	
Erf	Erf(v)	誤差関数を返します	
Erfc	Erfc(v)	相補誤差関数を返します	
Exp	Exp(x)	自然対数eのx乗を返します	
Exp	Exp2(x)	2のx乗を返します	

2			
Exp m1	Expm1(x)	自然対数eのx乗から1を引いた値を返します	
Flo or	Floor(v)	小数点以下を切り捨てた値を返します	
Hyp ot	Hypot(x, y)	Sqrt(x*x + y*y)を返します	
Inf	Inf(sign)	sign>=0の時、正の無限大を返し、sign<0の時、負の無限大を返します	
IsI nf	IsInf(v, sign)	signの符号に応じて無限大であるかを返します	
IsN aN	IsNaN(v)	正規の値ではないNot a Numberであるかを返します	
Log	Log(v)	自然対数を返します	
Log 10	Log10(v)	10を底とした対数を返します	
Log 1p	Log1p(v)	+1した自然対数を返します	
Log 2	Log2(v)	2を底とした対数を返します	
Max	Max(v1[, v2, v3, ...])	指定された値で最大の値を返します	
Min	Min(v1[, v2, v3, ...])	指定された値で最小の値を返します	
NaN	NaN()	正規の値ではないNot a Numberを返します	
PI	PI()	円周率を返します	
Pow	Pow(x, y)	xのy乗を返します	
Rou nd	Round(v[, n])	小数点以下を四捨五入した値を返します。nを指定すると小数点第n位で四捨五入します	
Sig nbi t	Signbit(v)	値が負であるかを返します	
Sin	Sin(v)	正弦を返します	
Sin h	Sinh(v)	双曲線正弦を返します	
Sqr	Sqrt(v)	平方根を返します	

t			
Random	Random([max])	乱数を返します。引数がない場合は、0以上1未満の浮動小数を返し、引数を指定すると、指定された値を整数の最大値とみなし、0以上指定最大値未満の整数を返します	
Tan	Tan(v)	正接を返します	
Tanh	Tanh(v)	双曲線正接を返します	
Trunc	Trunc(v[, n])	小数部を切り捨て整数値を返します。nを指定すると小数点第n位で切り捨てます	
Degrees	Degrees(v)	ラジアンを度に変換します	
Radians	Radians(v)	度をラジアンに変換します	
FtoC	FtoC(v)	華氏を摂氏に変換します	
CtoF	CtoF(v)	摂氏を華氏に変換します	
GtoLBM	GtoLBM(v)	グラムをポンドに変換します	
LBMtoG	LBMtoG(v)	ポンドをグラムに変換します	
GtoOZM	GtoOZM(v)	グラムをオンスに変換します	
OZMtoG	OZMtoG(v)	オンスをグラムに変換します	
MtoIN	MtoIN(v)	メートルをインチに変換します	
INtoM	INtoM(v)	インチをメートルに変換します	
MtoFT	MtoFT(v)	メートルをフィートに変換します	
FTtoM	FTtoM(v)	フィートをメートルに変換します	

8. Gravio AI オプションの利用

ここでは、Gravio Enterprise AI Edition Gold / Platinumで利用するGorilla社提供のGorilla IVARとGorilla BAPの設置と利用法について説明します。

8.1. Gorilla IVAR, BAPモジュールの取得

Gorilla IVAR, BAPインストーラ及びドキュメントの入手

Gorilla IVAR, BAPインストーラ及びドキュメントは [ここ](#) をクリックしログインして入手します。
ログイン後「ダウンロード」タブを選択すると、

製品	
Gravio	
Gravio Coordinator Setup	Download
Gorilla BAP	Download
Gorilla IVAR	Download

のようにそれぞれのダウンロードボタンが表示されます。
IVAR, BAPともzipファイル形式のファイルがダウンロードできます。
それぞれのzipファイルにインストーラ及びドキュメントが含まれています。

インストールは[BAP](#)、[IVAR](#) の順にインストールしてください。

なお、Gorilla IVAR/BAPの使い方の詳細は、ダウンロードしたファイルの中のマニュアル(PDF)を参照してください

8.2. Gorilla BAPの設置

事前に準備するもの

- ・BAPのアカウント Root, Administrator, Application ユーザのパスワード
- ・BAPのライセンスキー

BAPをアップデートする場合には [こちら](#) を参照してください。

インストールする環境

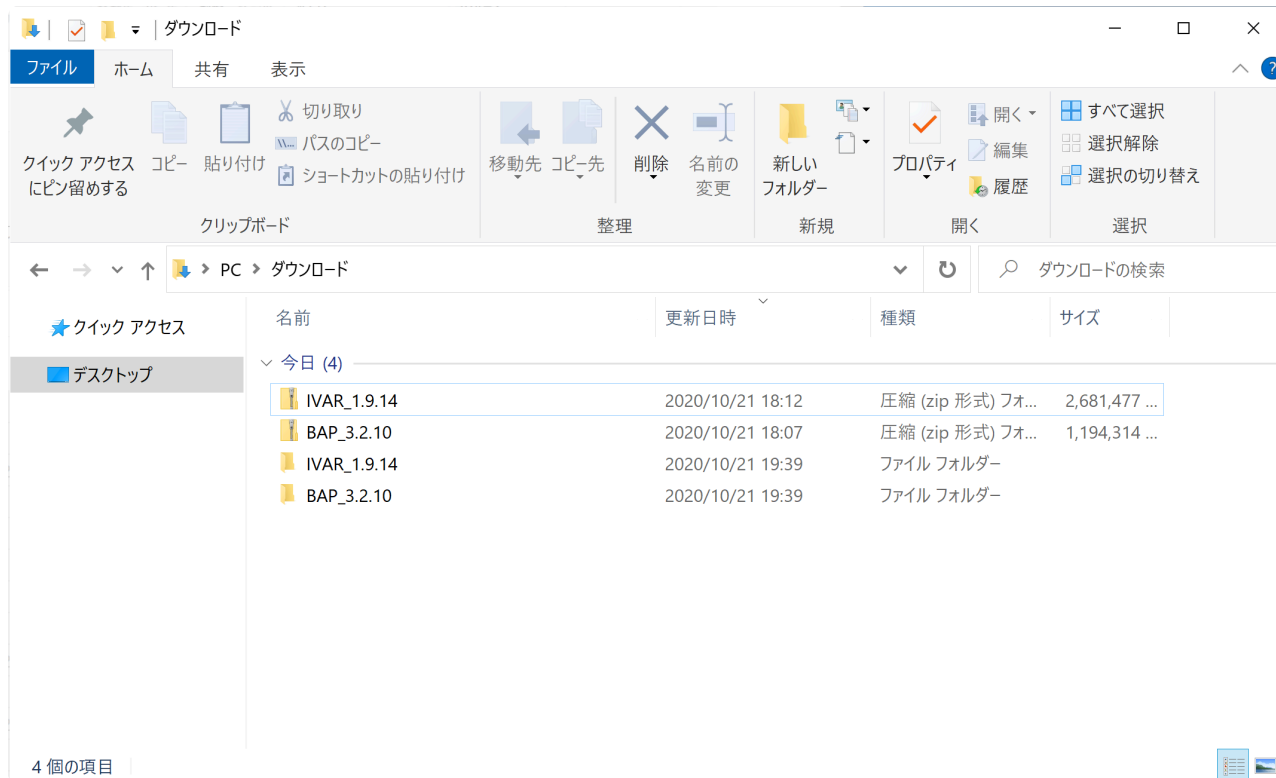
BAPをインストールするWindows10 PCとChromeブラウザを用意してください。
またライセンス登録をしますのでインターネットに接続されている必要があります。

インストールする前のご注意

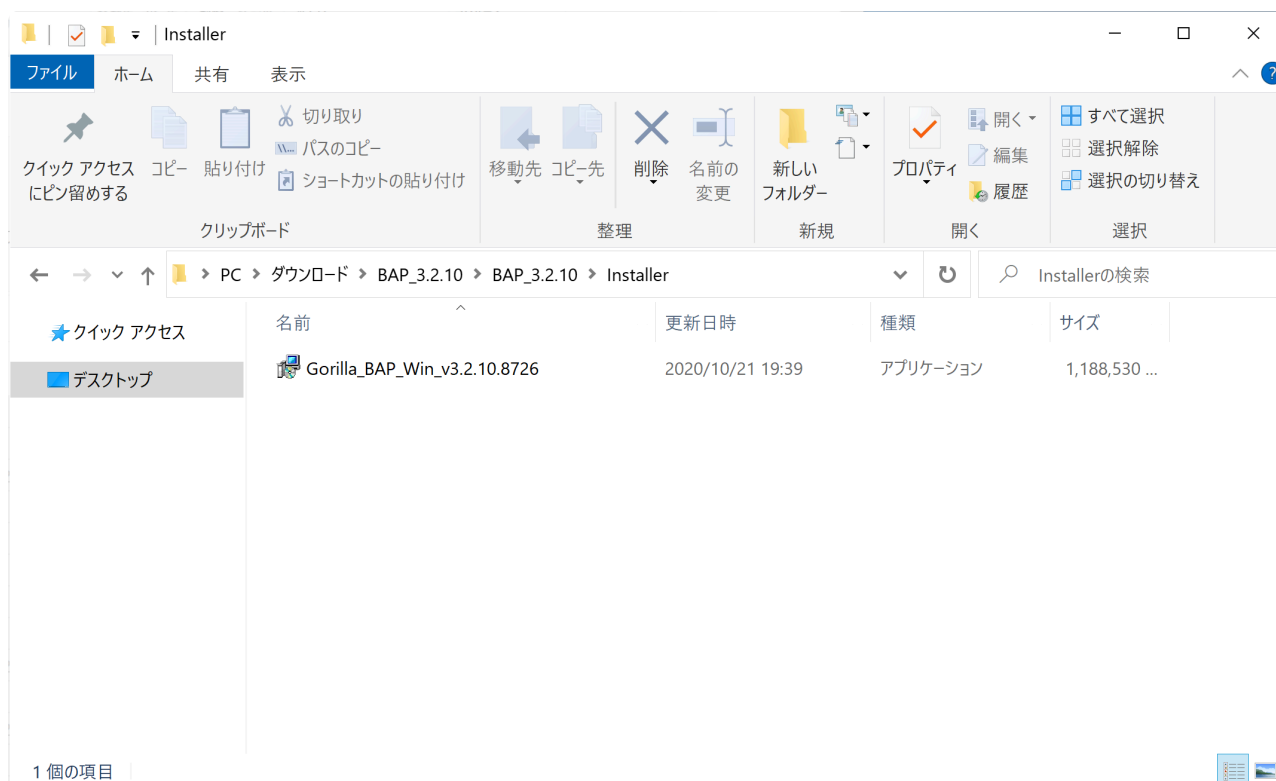
BAPはPostgreSQLを使用します。すでにPostgreSQLがインストールされているとバージョン違いなどの競合により正しくインストールできないことがあるため、PostgreSQLはアンインストールをお願いします。またデータフォルダも削除してください。

また、Windows10はWindows Updateを使用して最新バージョンへのアップデートをお願いします。

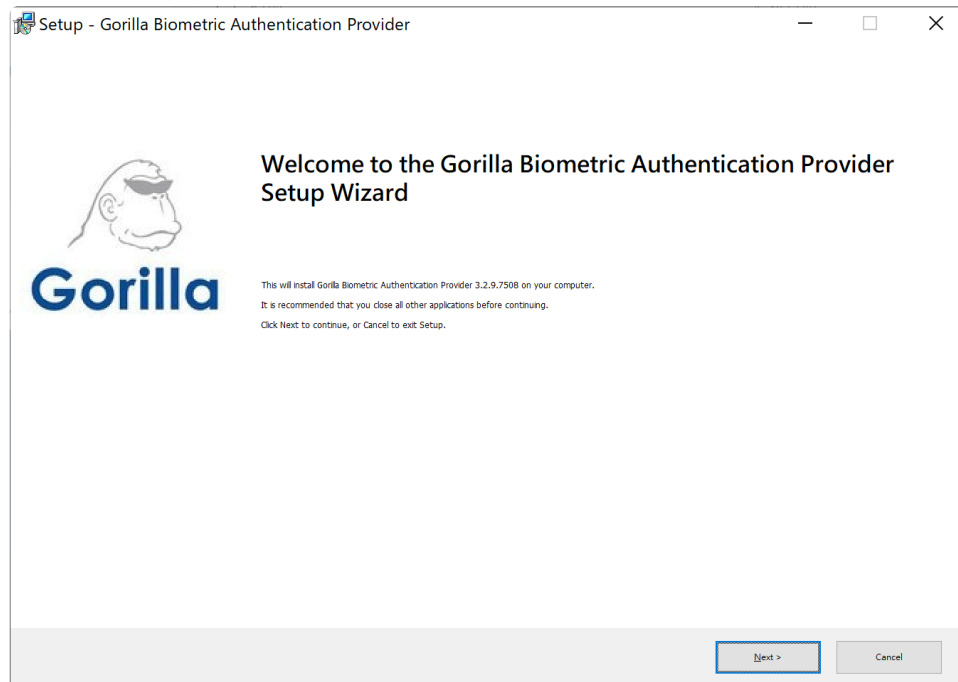
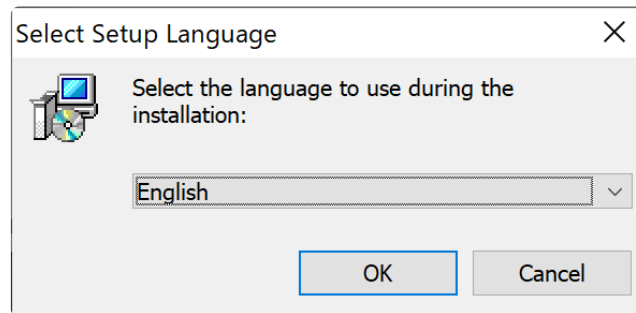
インストール手順



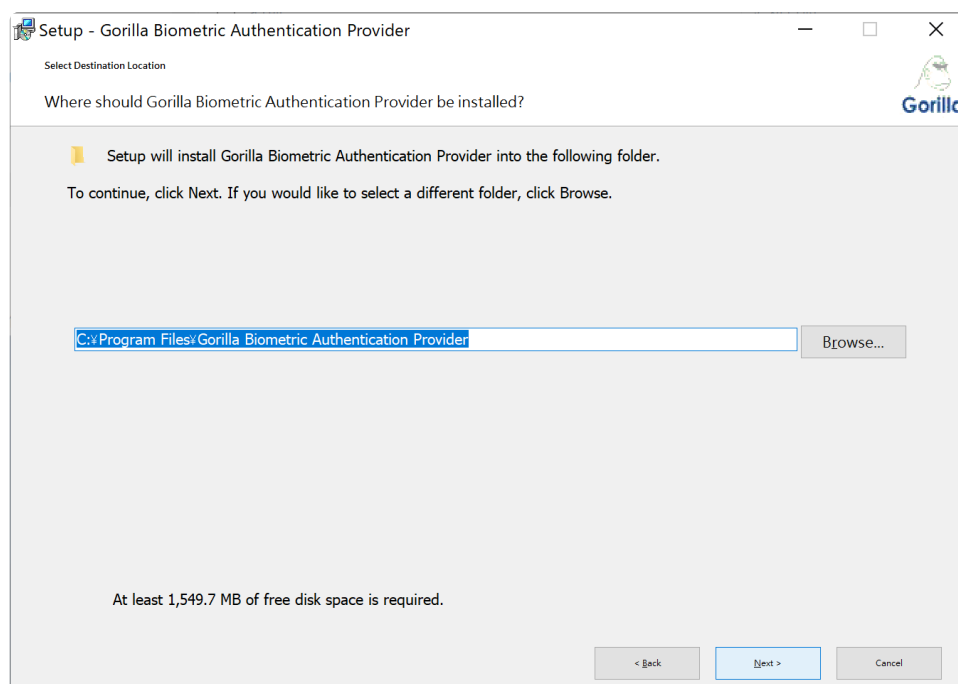
ダウンロードしたzipを展開してください。
展開するとBAPフォルダが出来ますので中を開いてください。



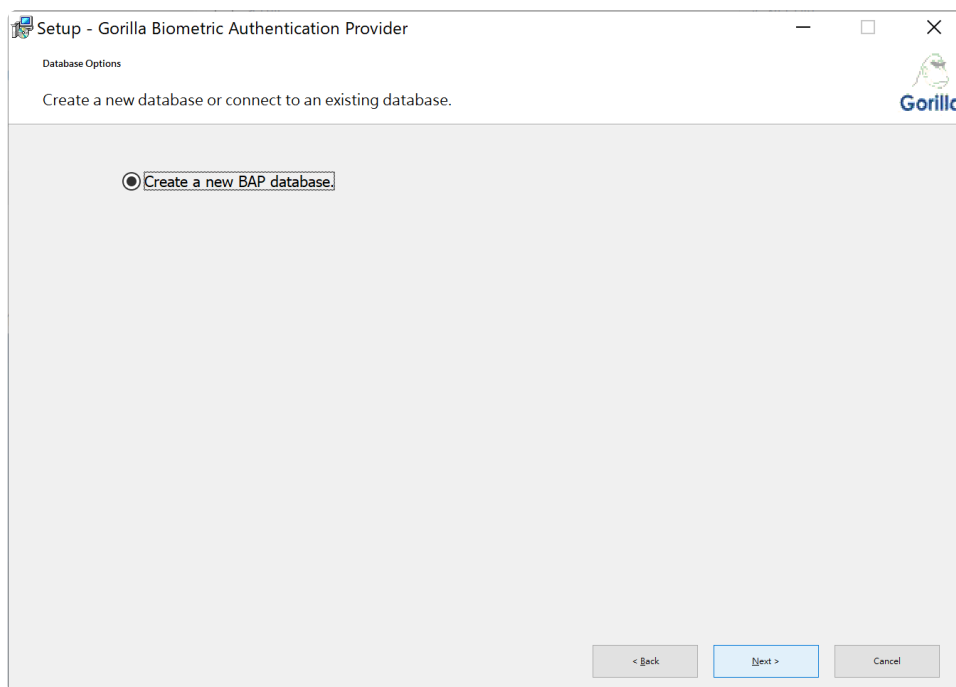
Gorilla_BAP_Win_<バージョン番号>.Setup アプリケーションがありますので、これを実行してください。



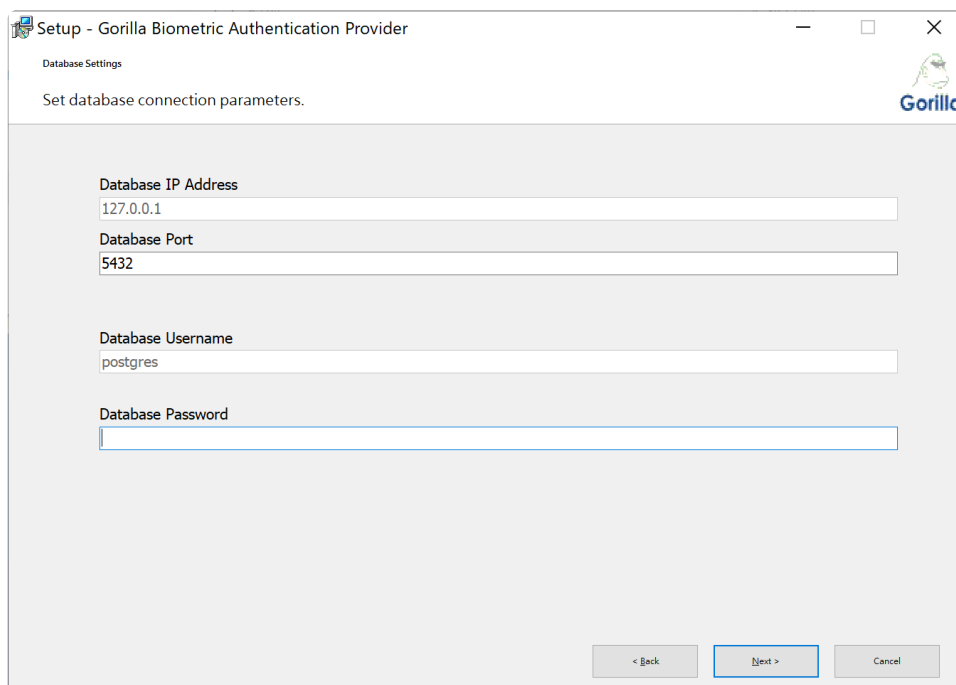
Next ボタンを押してインストールを進めてください。



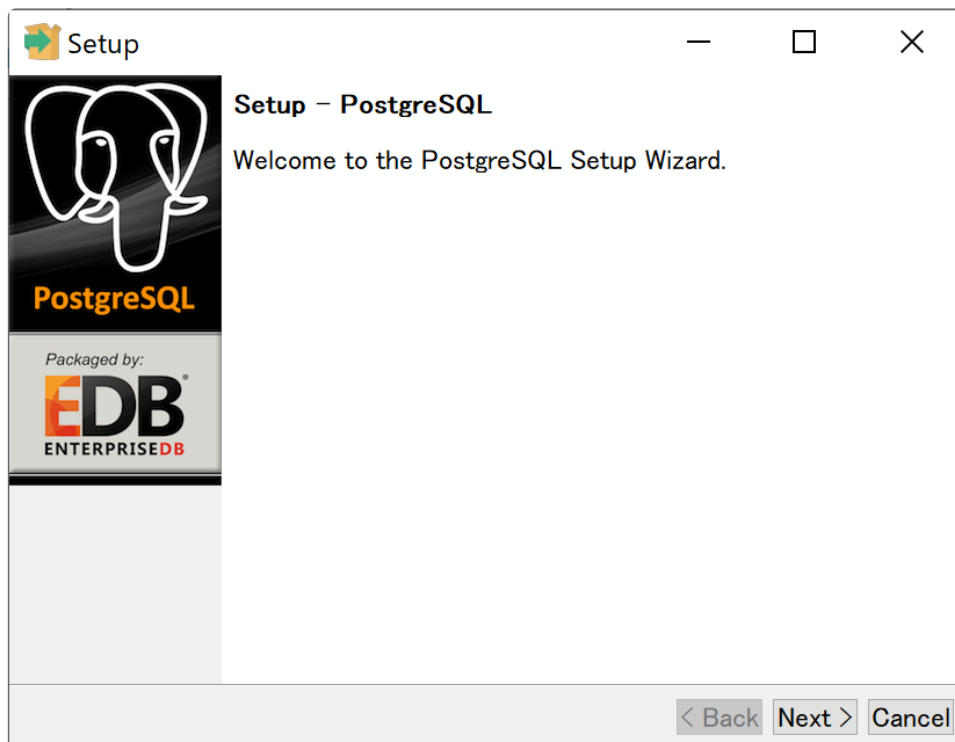
標準のフォルダにインストールするためそのまま Next ボタンを押してください。



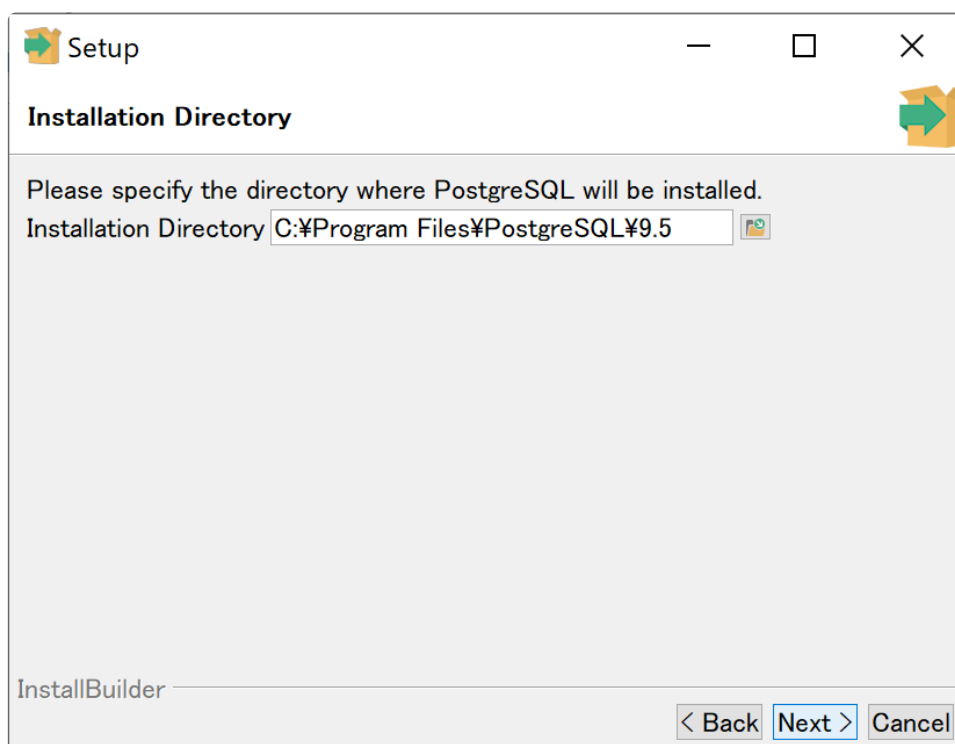
Next ボタンを押してください。



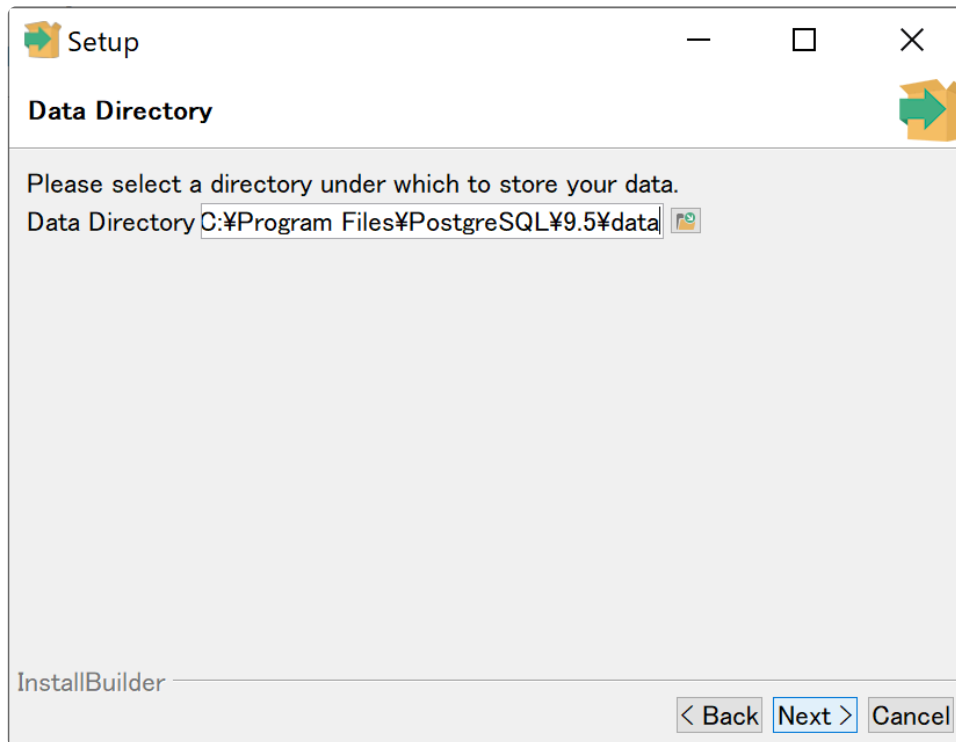
Database Passwordには後で設定するPostgreSQLのパスワードを設定してください。
Next ボタンを押すとPostgreSQLのインストーラが起動します。



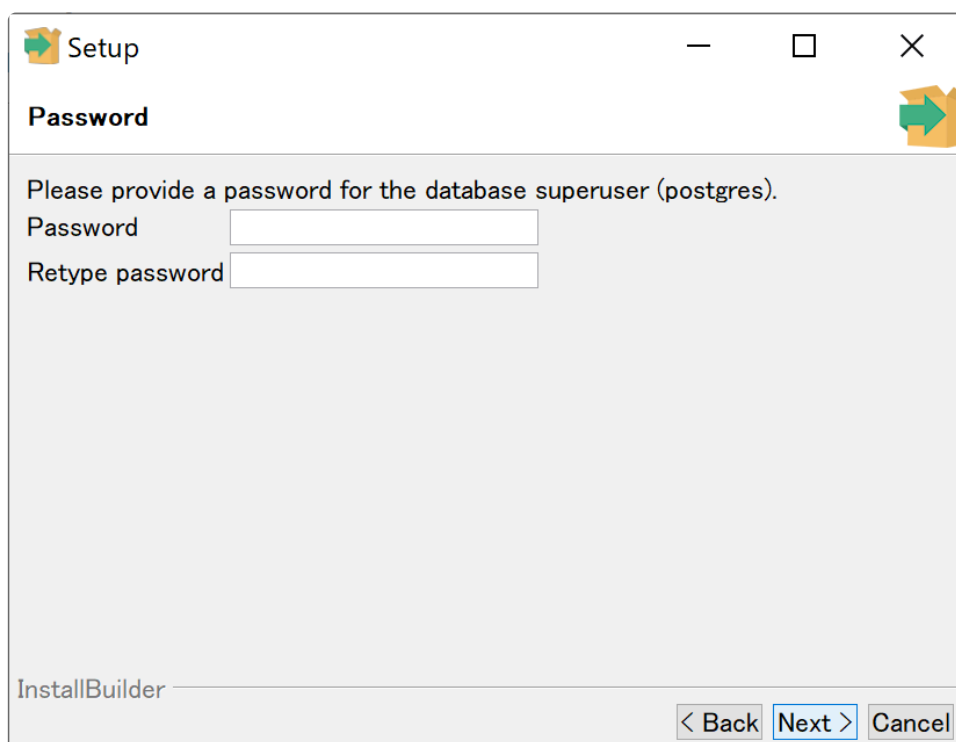
Next ボタンを押してください。



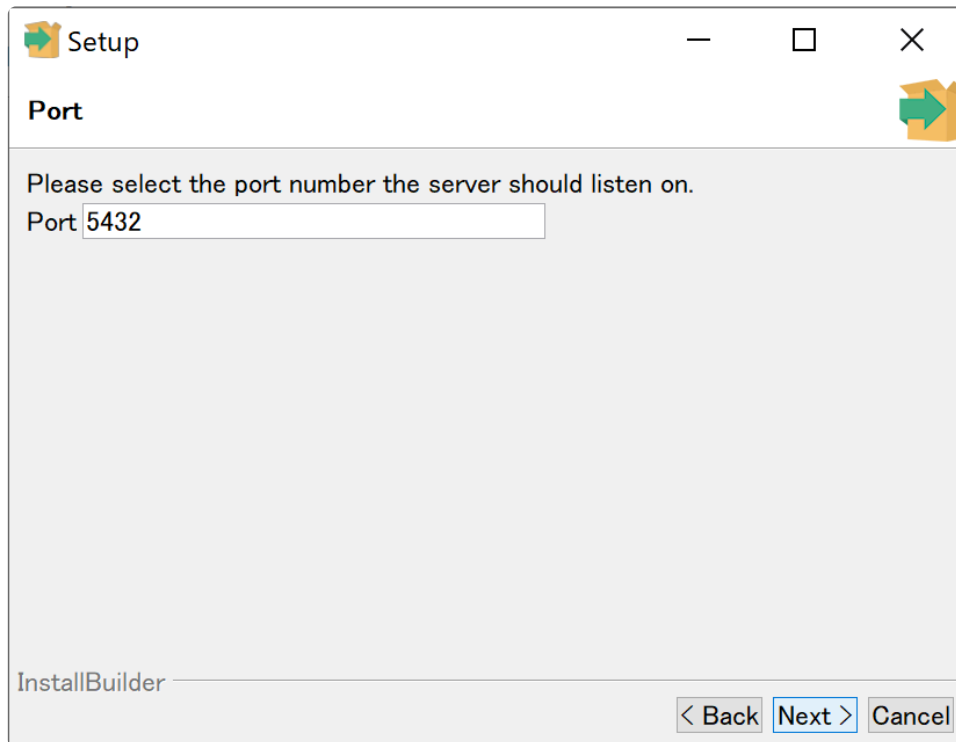
標準のフォルダにインストールするためそのまま Next ボタンを押してください。



標準のフォルダにインストールするためそのまま Next ボタンを押してください。

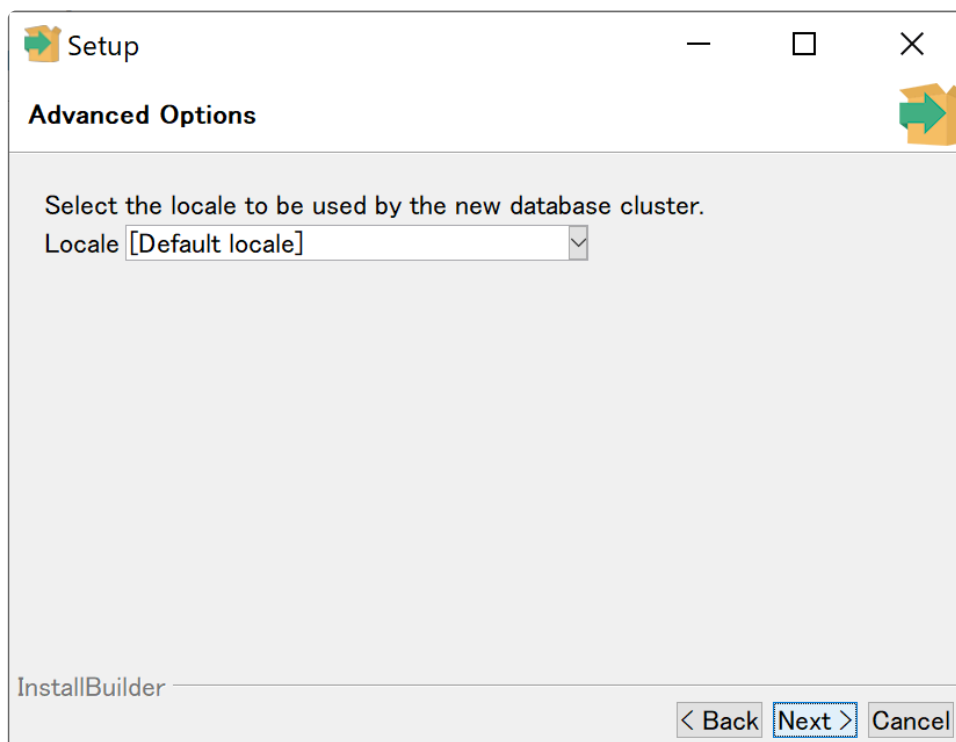


先ほど登録したPostgreSQLのパスワードを登録して Next ボタンを押してください。



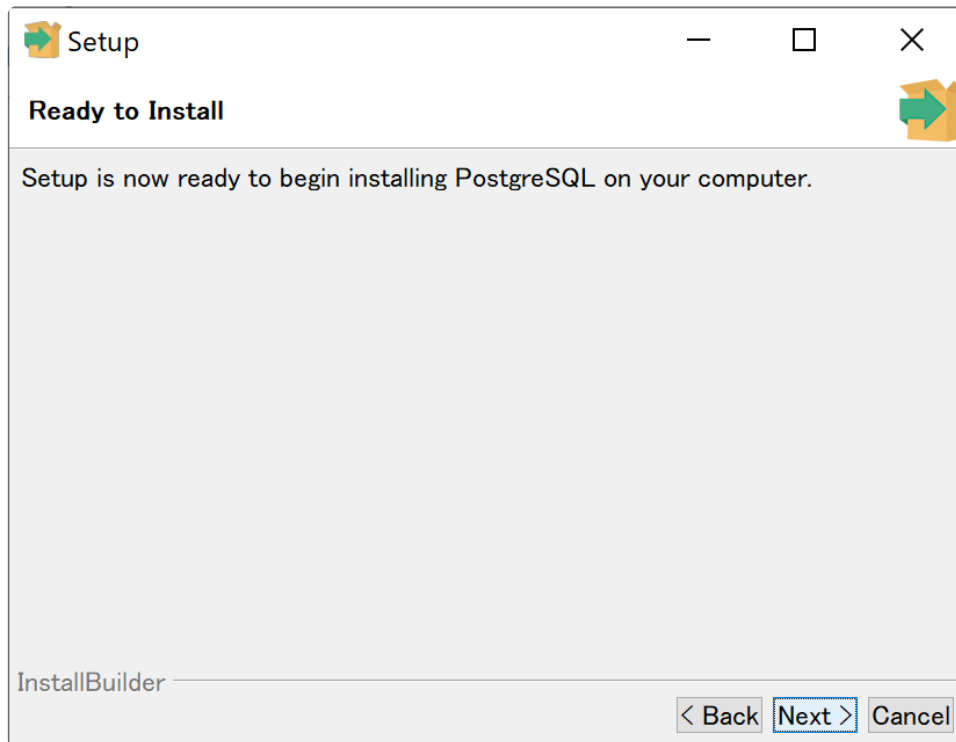
The image shows a 'Setup' window with a title bar containing a green arrow icon, the text 'Setup', and standard window controls (minimize, maximize, close). The window has a light gray background. At the top left, the word 'Port' is displayed. Below it, the text 'Please select the port number the server should listen on.' is shown. A text input field contains the value '5432'. At the bottom left, the text 'InstallBuilder' is visible. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'. A green arrow icon is also present in the top right corner of the window content area.

標準のポートを使用するためそのまま Next ボタンを押してください。

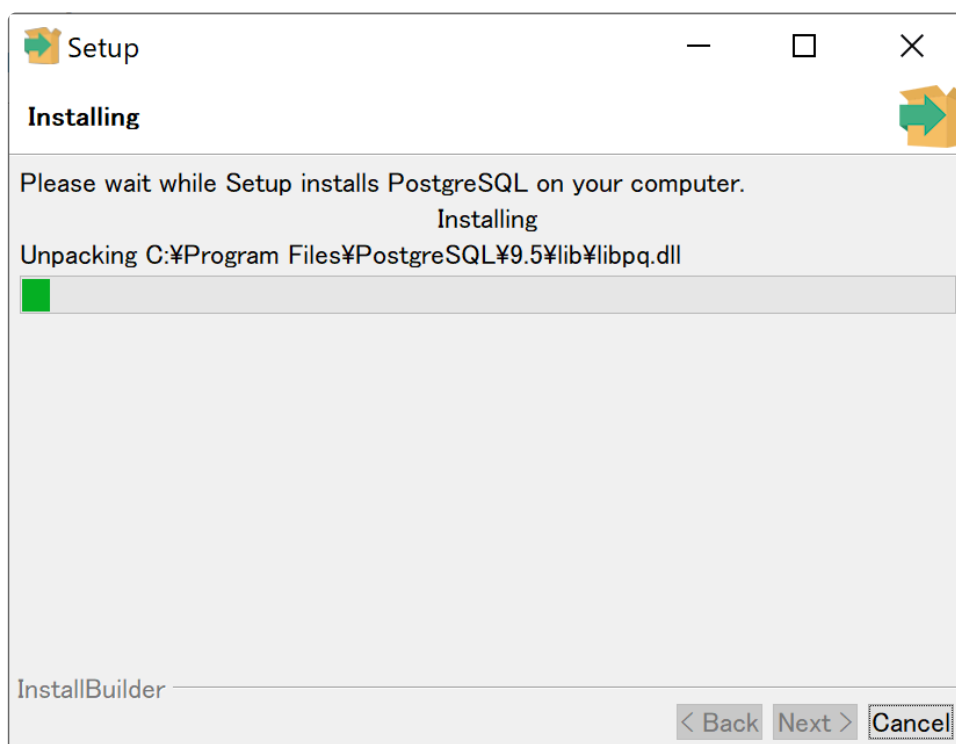


The image shows a 'Setup' window with a title bar containing a green arrow icon, the text 'Setup', and standard window controls (minimize, maximize, close). The window has a light gray background. At the top left, the text 'Advanced Options' is displayed. Below it, the text 'Select the locale to be used by the new database cluster.' is shown. A dropdown menu is set to 'Default locale'. At the bottom left, the text 'InstallBuilder' is visible. At the bottom right, there are three buttons: '< Back', 'Next >' (highlighted with a blue border), and 'Cancel'. A green arrow icon is also present in the top right corner of the window content area.

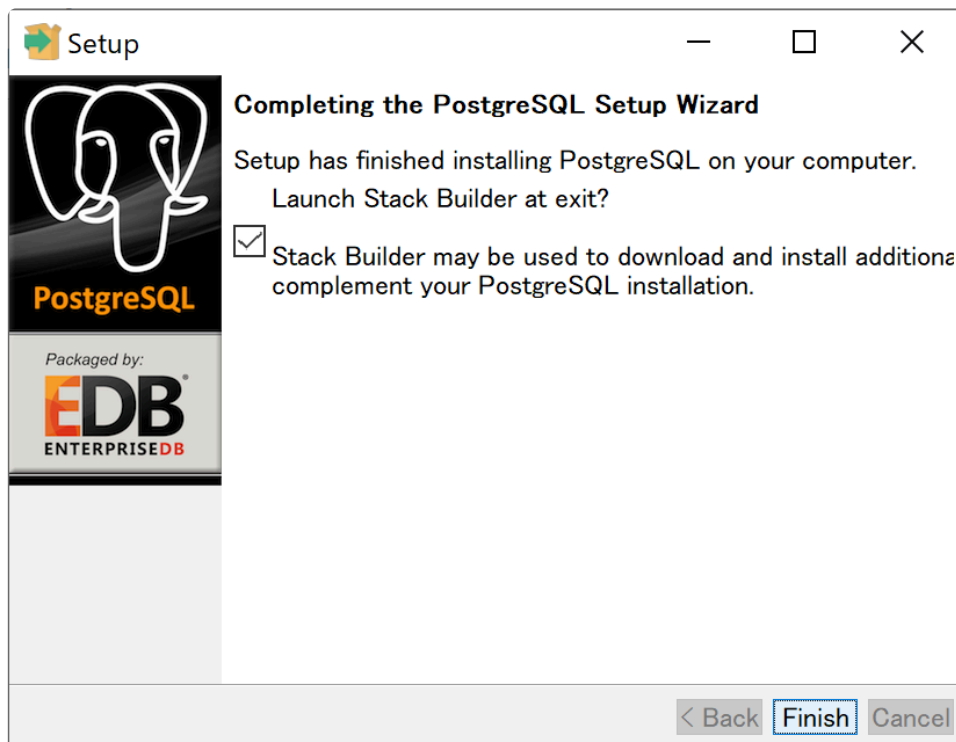
変更せずに Next ボタンを押してください。



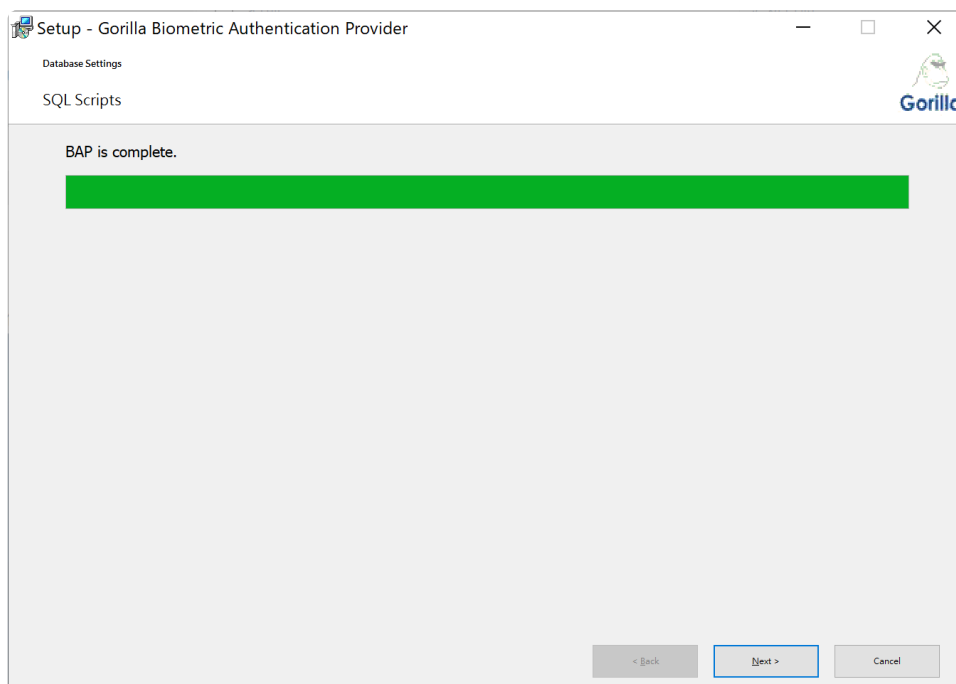
Next ボタンを押してください。



ファイルのコピーが開始します。

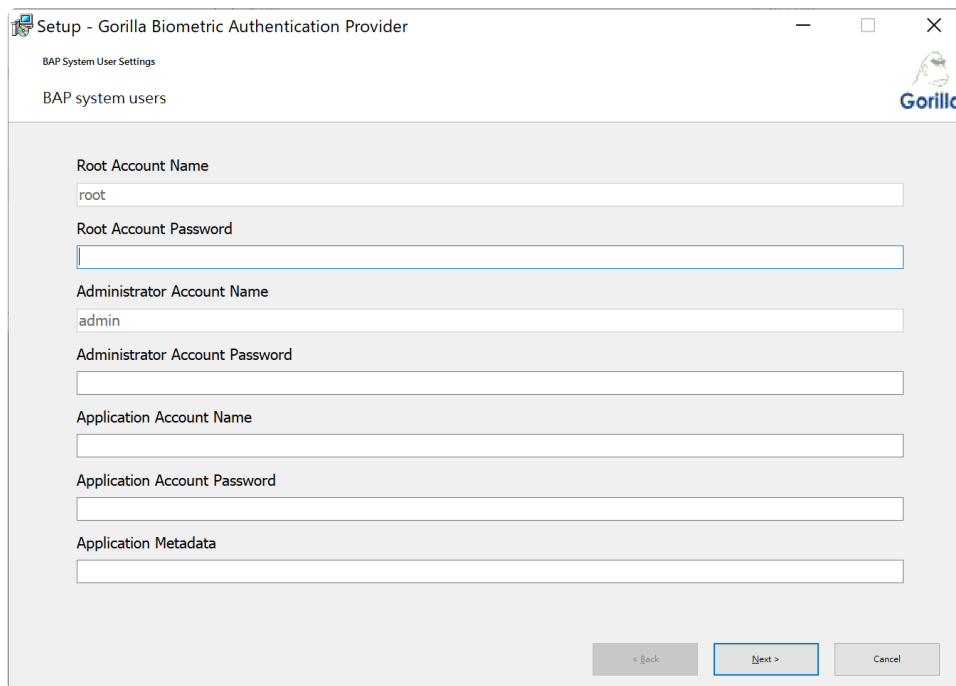


インストールが終了したら Stack Builder may be... のチェックを外して Finish ボタンを押してください。



BAPのインストールが終わりましたら続いて初期設定を行いますので Next ボタンを押してください。

初期設定



Setup - Gorilla Biometric Authentication Provider

BAP System User Settings

BAP system users

Root Account Name

root

Root Account Password

Administrator Account Name

admin

Administrator Account Password

Application Account Name

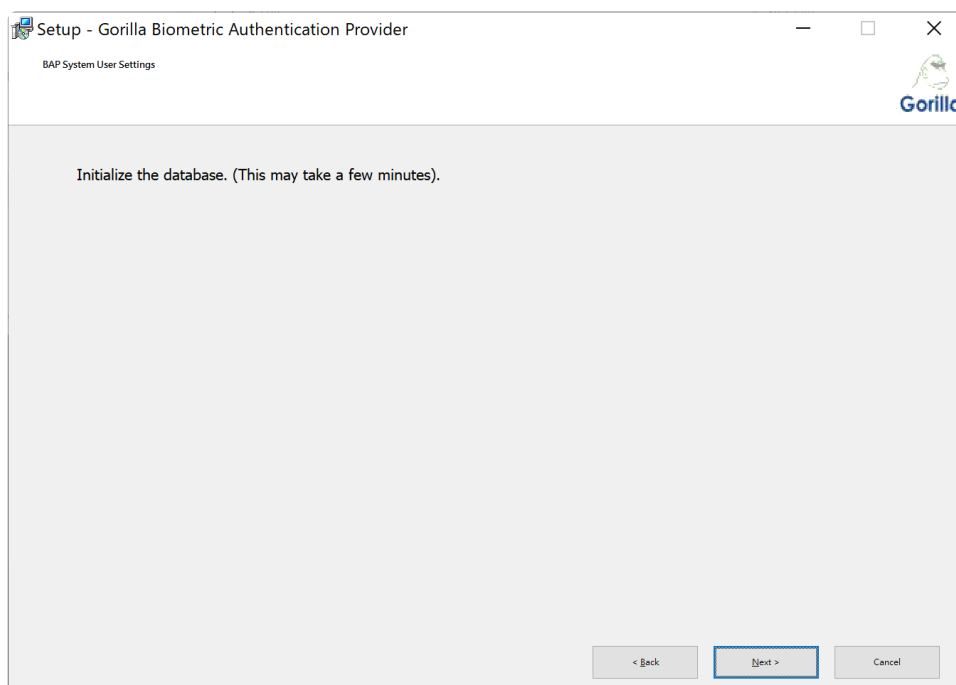
Application Account Password

Application Metadata

< Back Next > Cancel

BAPには Root, Administrator, Applicationの3つのユーザを作成します。
運用で使用するのは Admin アカウントとなります。

Root Account Password, Administrator Account Password, Application Account Name, Application Account Passwordにそれぞれ登録して Next ボタンを押してください。



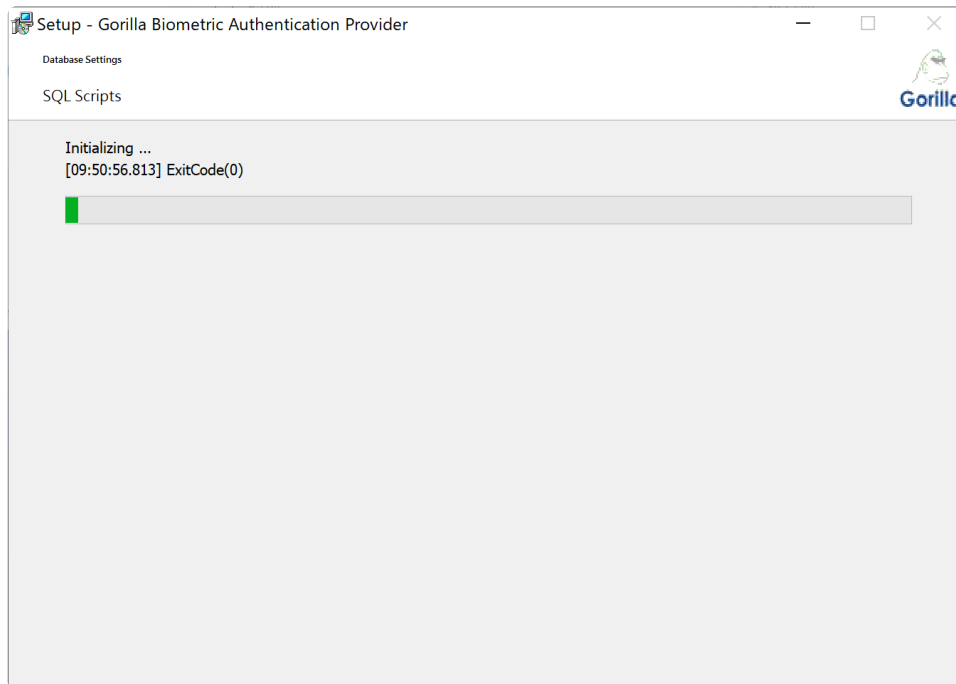
Setup - Gorilla Biometric Authentication Provider

BAP System User Settings

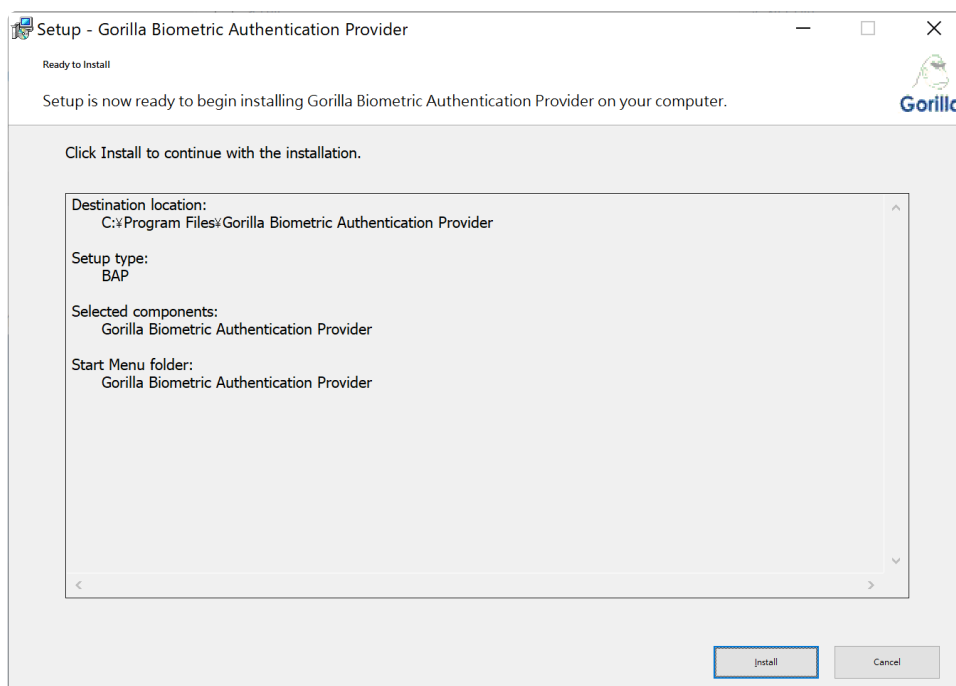
Initialize the database. (This may take a few minutes).

< Back Next > Cancel

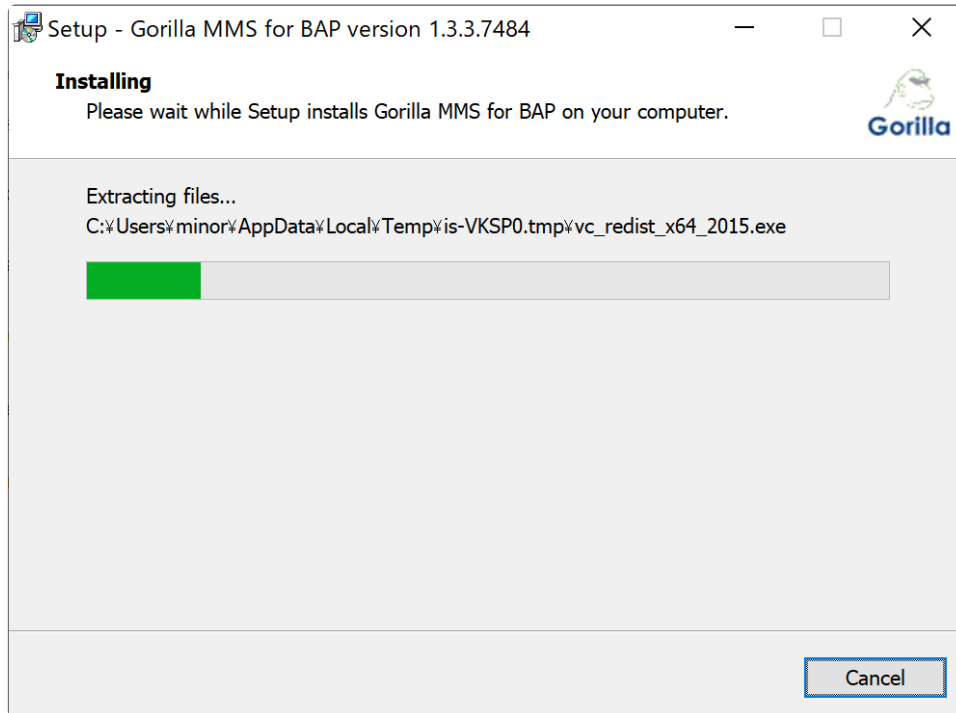
Next ボタンを押してください。



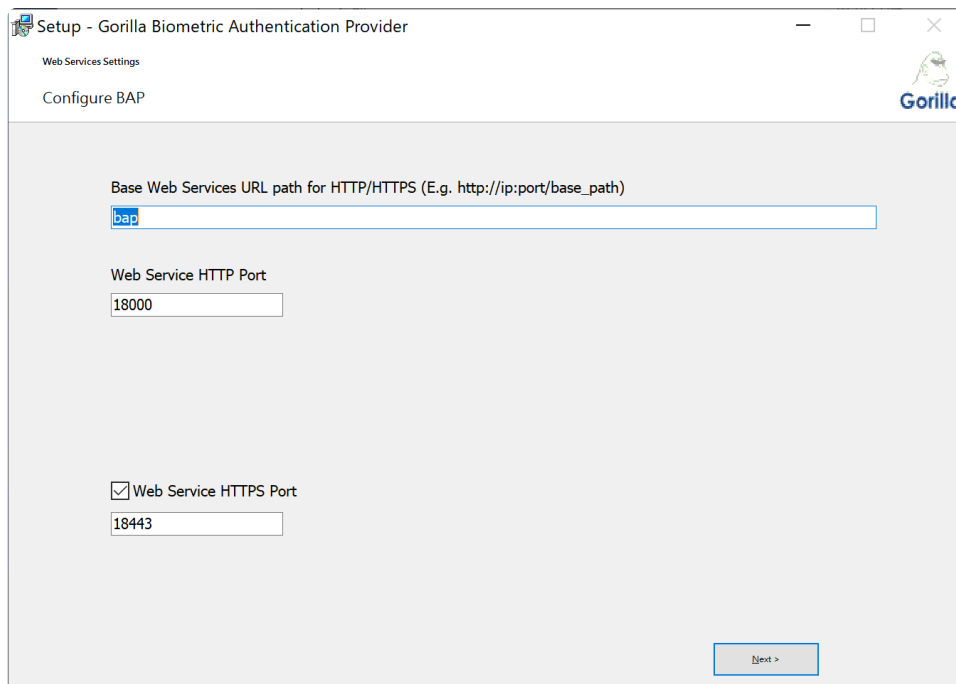
ファイルのコピーが開始します。



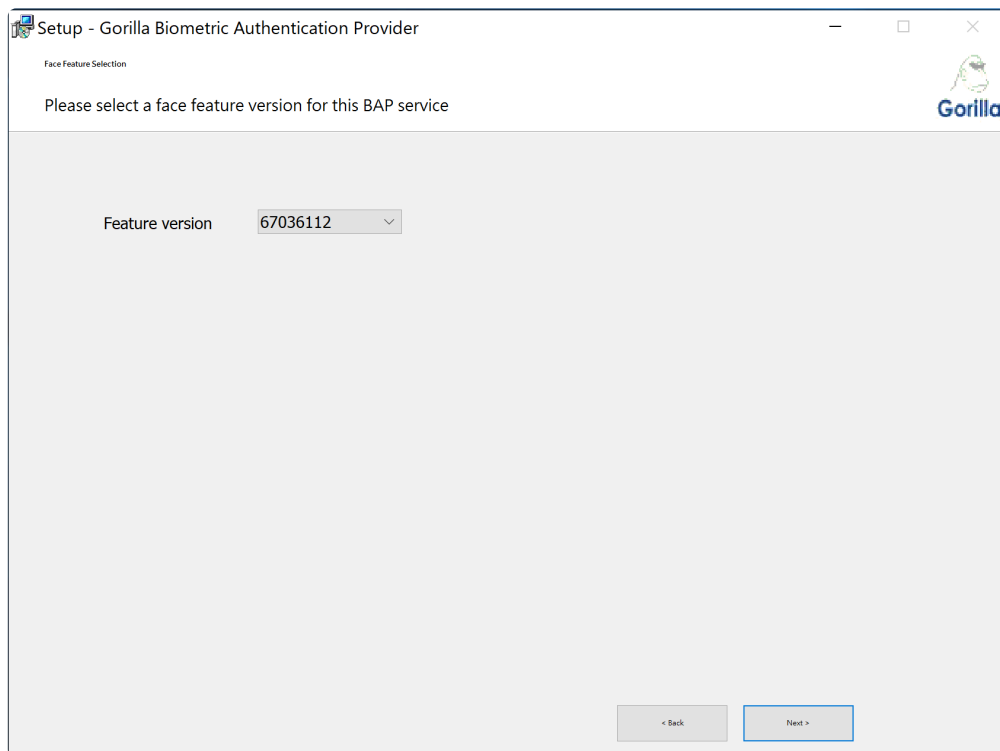
Install ボタンを押してください。MMSのインストールが開始します。



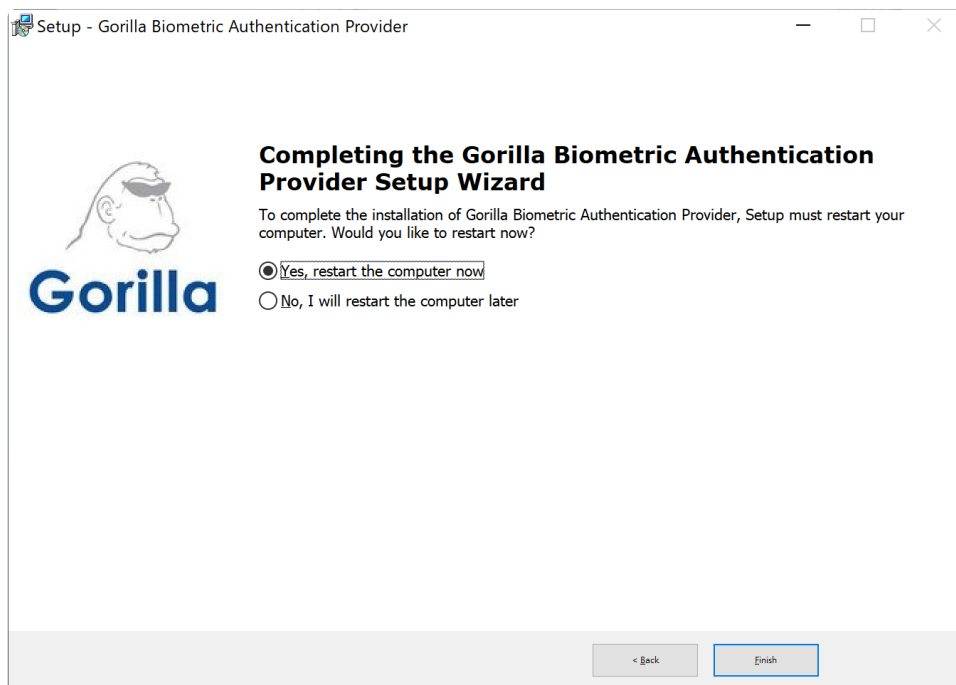
ファイルのコピーが開始します。



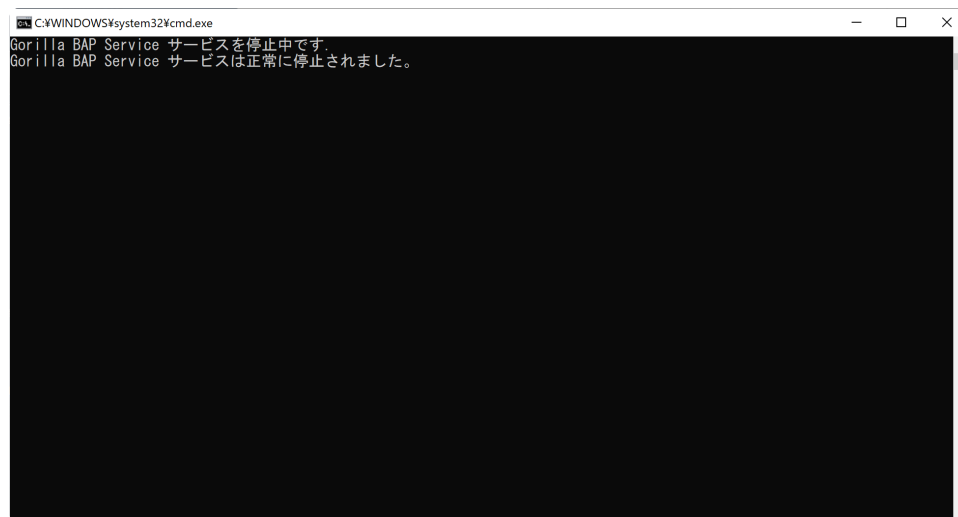
標準の設定のためそのまま Next ボタンを押してください。



Feature versionはIVAR 1.10を使用する場合には 70038900 を選択します。IAVR1.9以下を使用する場合にはデフォルトの 67036112 を選択してください。



インストールが終わりましたらPCを再起動しますので そのまま Finish ボタンを押してください。



インストールしているファイルが表示され、不定期にプロンプトが表示されますのでインストール中は画面操作をせずお待ちください。

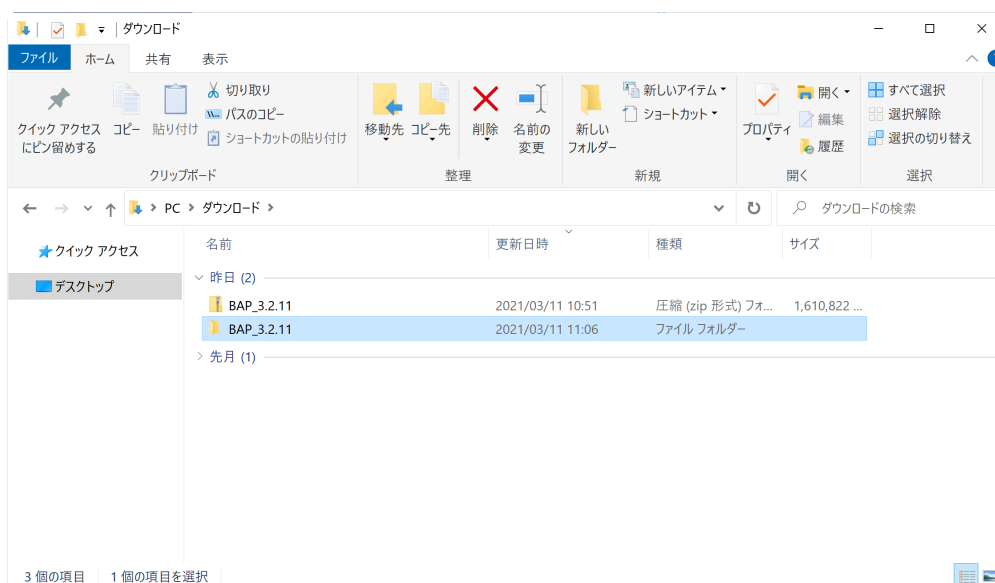
8.2.1. Gorilla BAPのアップデート

事前に準備するもの

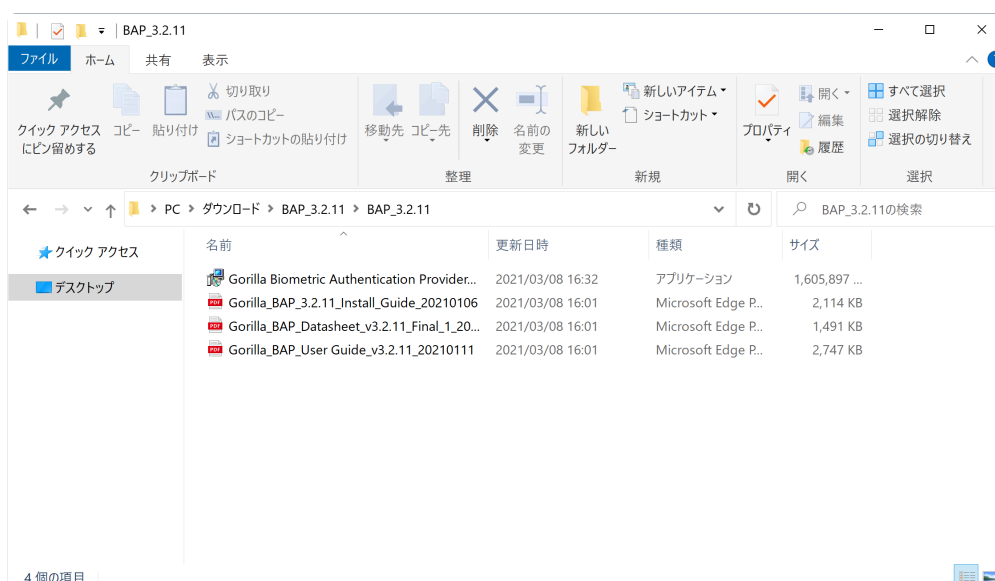
- ・PostgreSQLのパスワード

BAPを新規インストールする場合には [こちら](#) を参照してください。

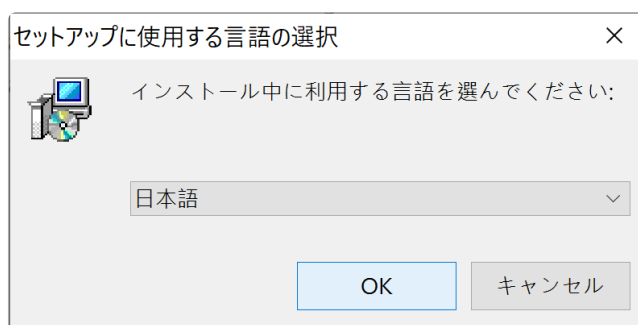
アップデート手順



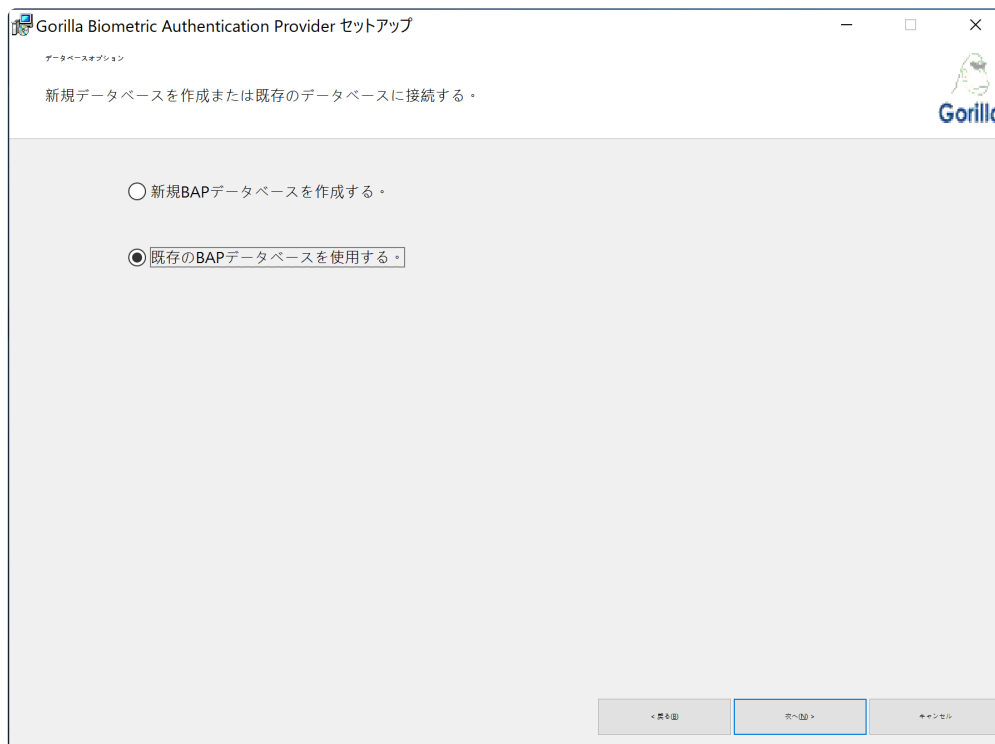
ダウンロードしたzipを展開してください。
展開するとBAPフォルダが出来ますので中を開いてください。



Gorilla Biometric Authentication Provider アプリケーションがありますので、これを実行してください。
詳細はInstall GuideやUser GuideのPDFを参照してください。



次へ ボタンを押してインストールを進めてください。



Gorilla Biometric Authentication Provider セットアップ

データベースオプション

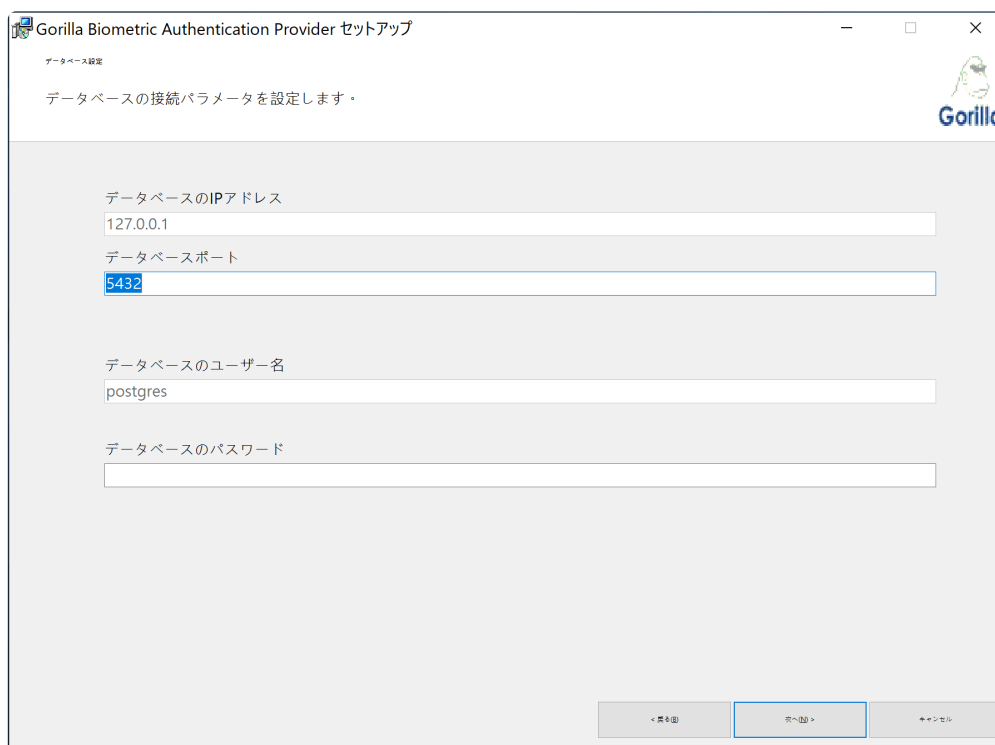
新規データベースを作成または既存のデータベースに接続する。

☐ 新規BAPデータベースを作成する。

☒ 既存のBAPデータベースを使用する。

戻る > 次へ > キャンセル

アップデートインストールするため、既存のデータベースを使用するが選択されたまま 次へ ボタンを押してください。



Gorilla Biometric Authentication Provider セットアップ

データベース設定

データベースの接続パラメータを設定します。

データベースのIPアドレス
127.0.0.1

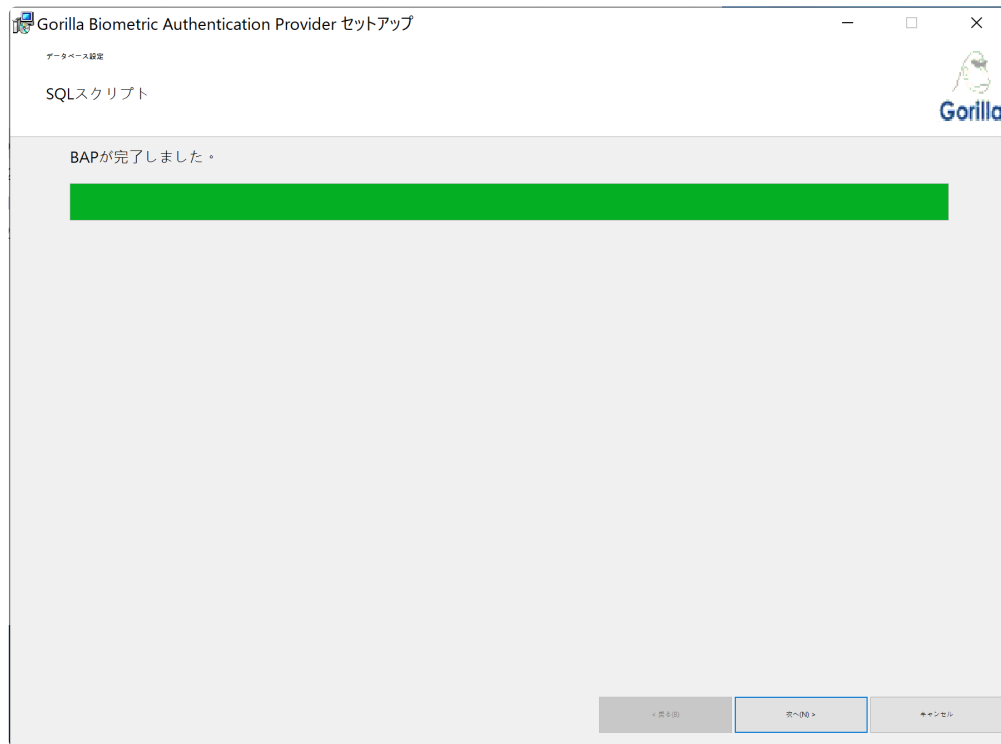
データベースポート
5432

データベースのユーザー名
postgres

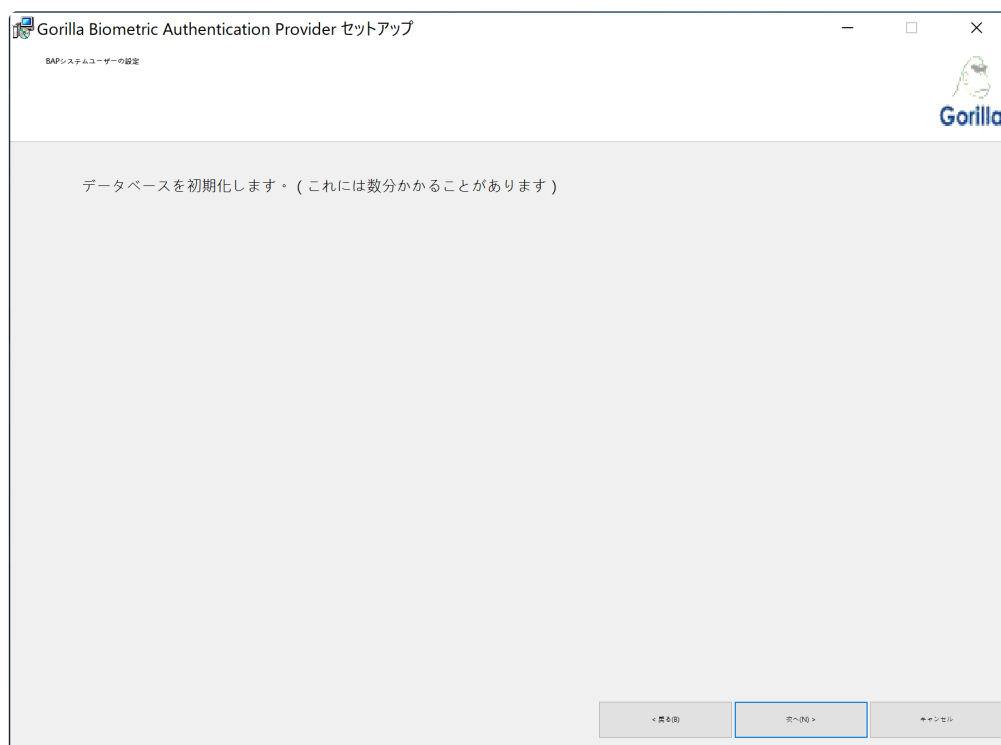
データベースのパスワード
[]

戻る > 次へ > キャンセル

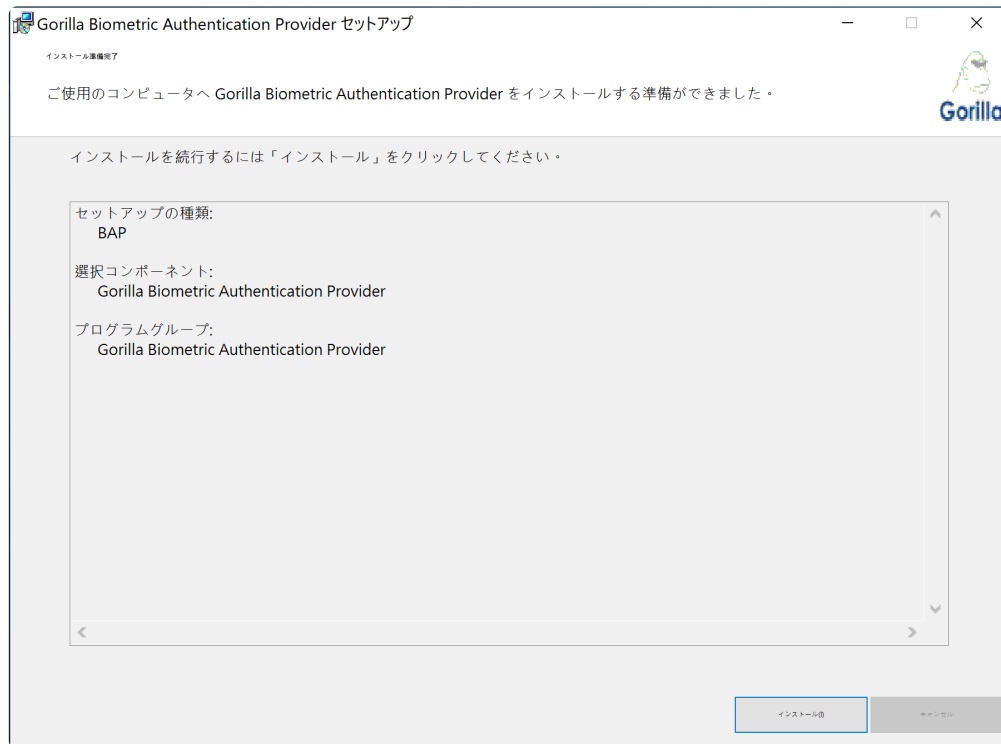
PostgreSQLのデータベースのパスワードを入力して次へ ボタンを押してください。



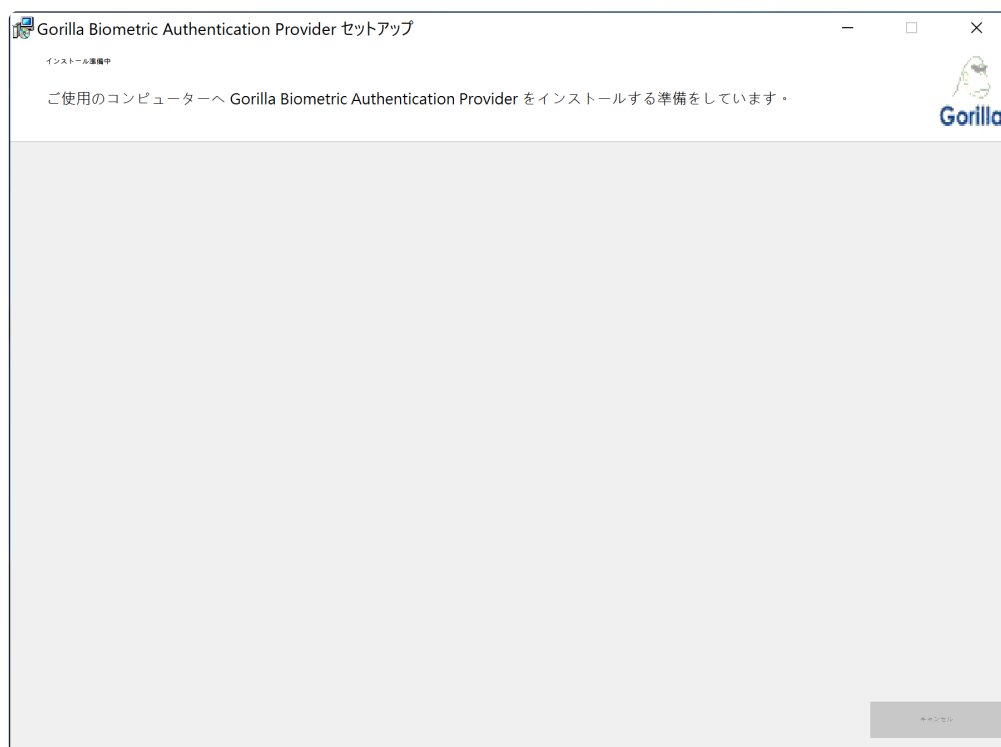
BAPへの接続が完了するまで待ってから次へ ボタンを押してください。



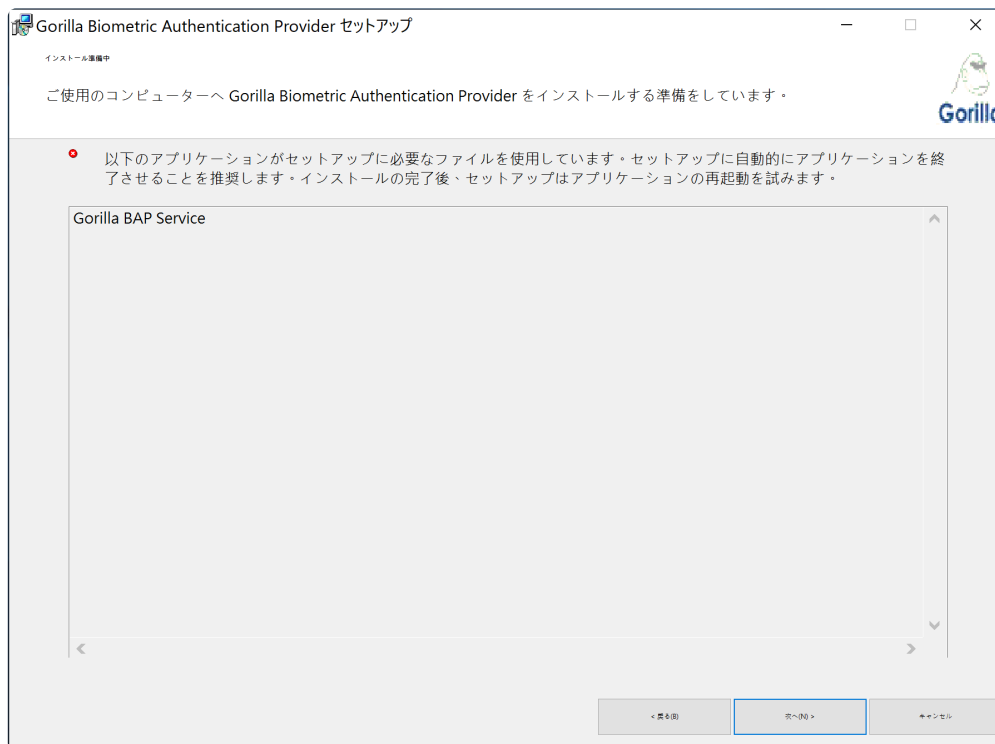
続いてデータベースを初期化しますので次へ ボタンを押してください。



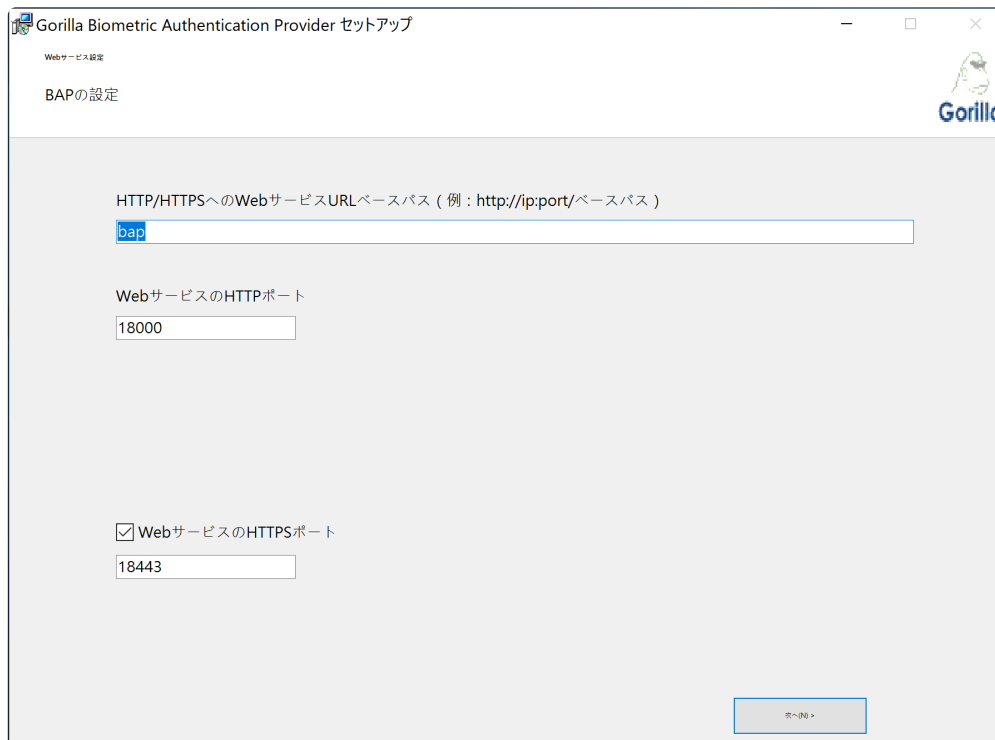
インストールの準備が完了しましたので インストール ボタンを押してください。



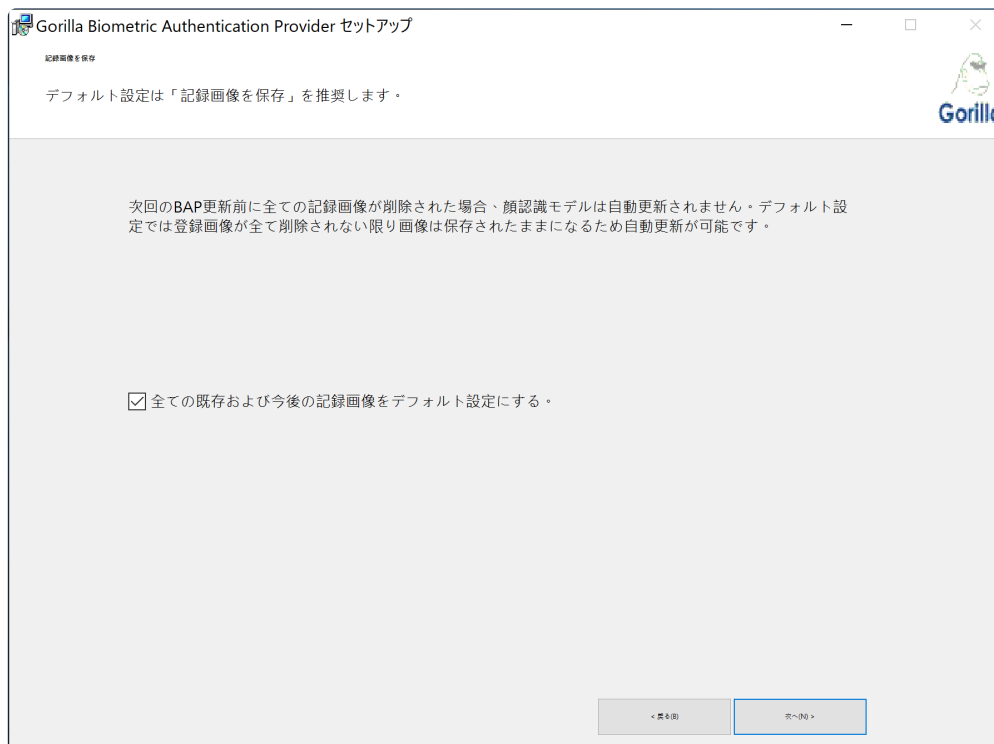
インストールの準備が進みますのでそのまま待ちます。



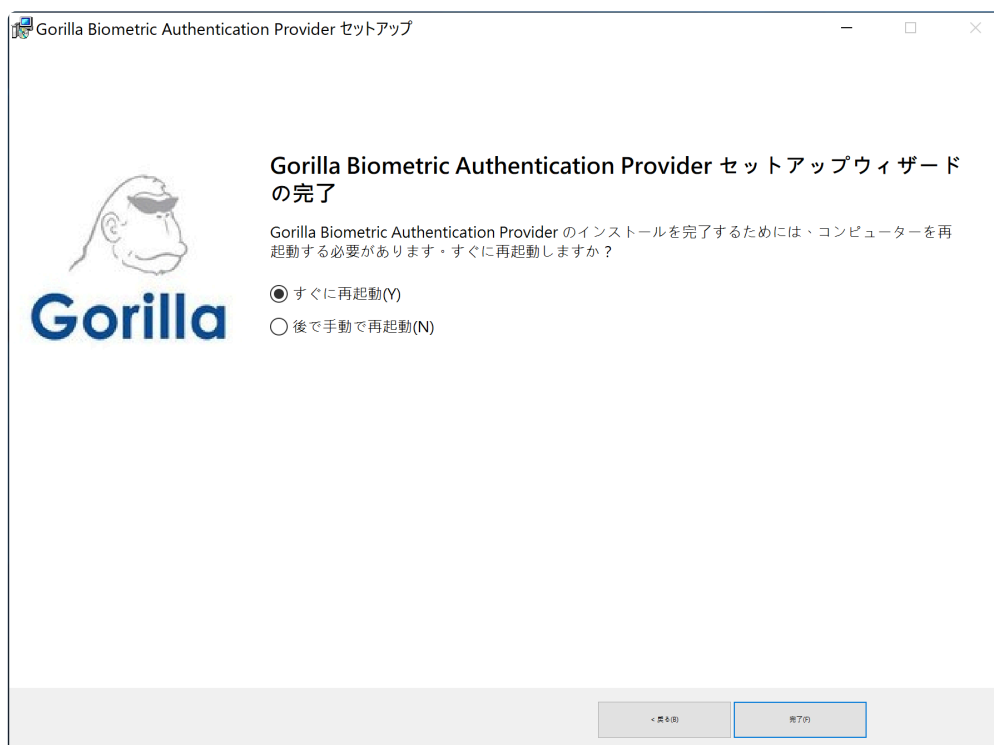
アップデートインストールのため、既存のBAPサービスが起動しているため警告が表示されますがそのまま 次へ ボタンを押してください。



BAPの設定は変更せずにそのまま 次へ ボタンを押してください。



そのまま 次へ ボタンを押してください。



Gorilla Biometric Authentication Provider のアップデートが完了しましたのでPCを再起動する必要があります。すぐに再起動を選択して 完了 ボタンを押してください。

8.2.2. Gorilla BAPのライセンス登録

ライセンスの登録

BAPのインストールが終わるとChromeを使用して下記のアドレスを開いてください。

注) BAPをインストールしたPCでChromeを操作することを想定しています。

異なるPCで操作する場合には“localhost”を環境に合わせて変更してください。

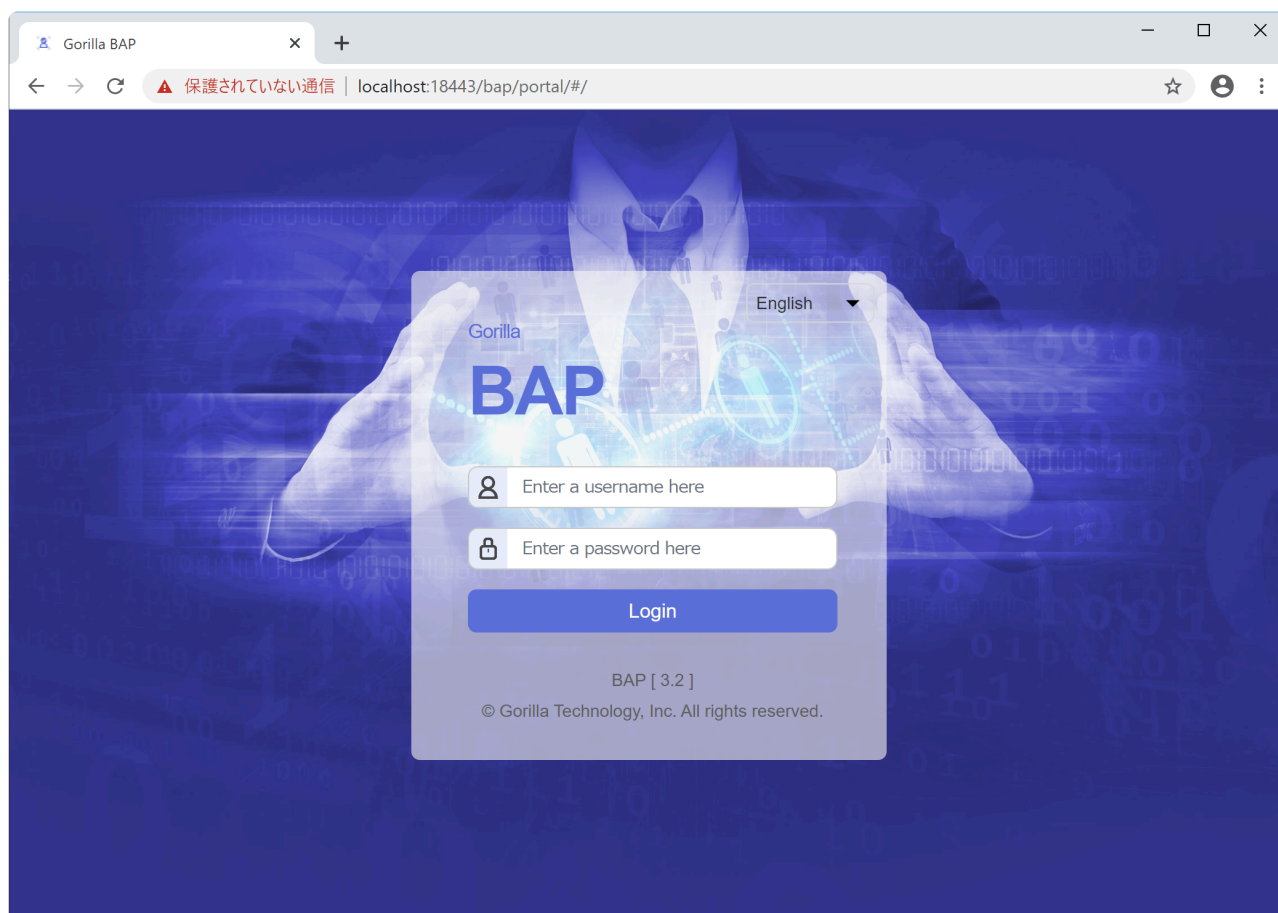
アドレス : <https://localhost:18443/bap/portal>



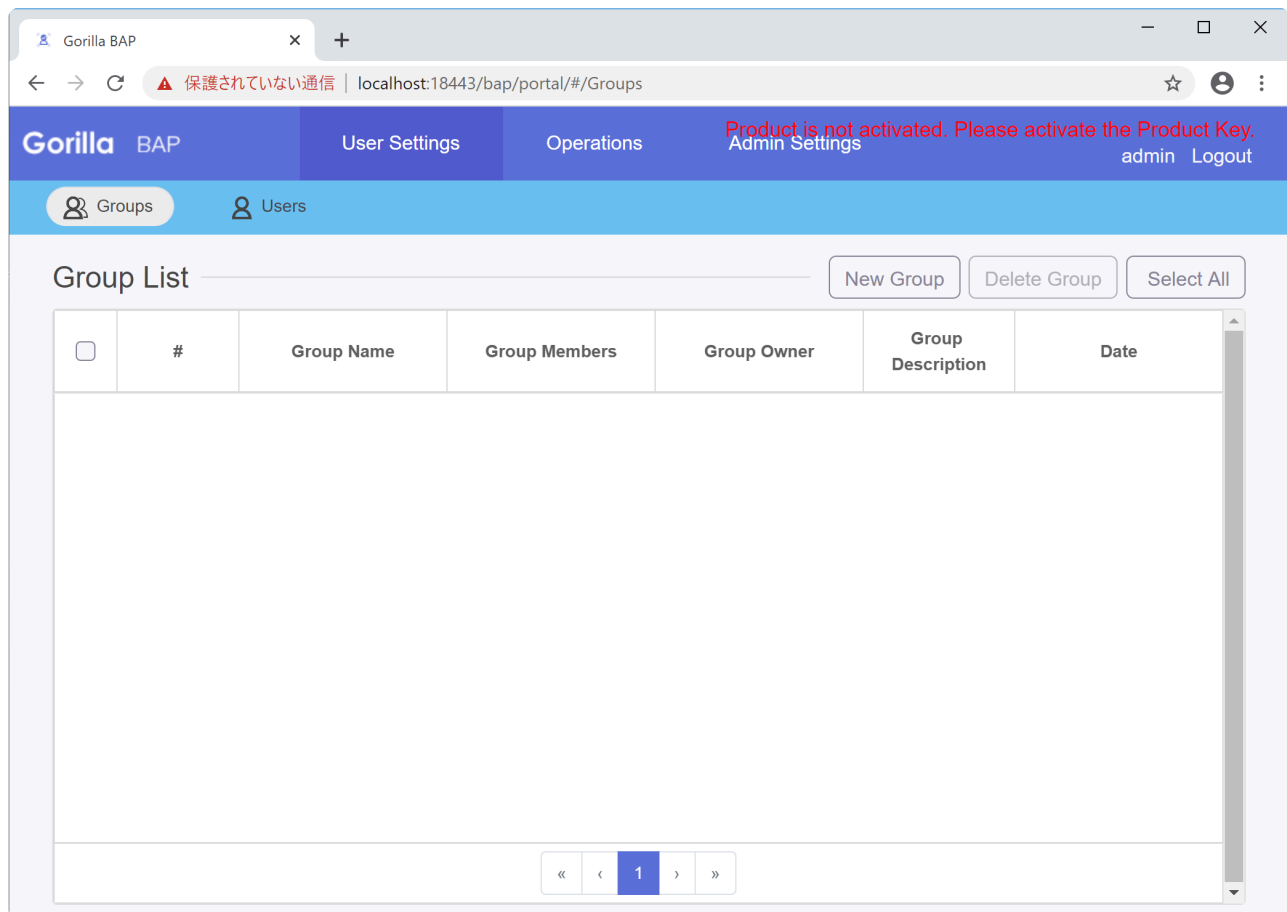
この接続ではプライバシーが保護されません

注) httpsでセルフサインの証明書を使用しているため最初に警告が表示されます。

この警告が表示された場合には 詳細設定 ボタンを押してlocalhostにアクセスする(安全ではありません) リンクをクリックして開いてください。



BAPのAdministratorの admin とパスワードを登録して Login ボタンを押してください。



ログインするとUser Setting 画面が表示されます。

The screenshot shows a web browser window with the URL `localhost:18443/bap/portal/#/SystemInformation`. The page has a blue header with the 'Gorilla BAP' logo and navigation tabs: 'User Settings', 'Operations', 'Admin Settings' (selected), and 'Logout'. Below the header is a sub-header with 'Accounts' and 'System Information' (selected). The main content area is divided into three columns: 'BAP Information', 'System Information', and 'License'. The 'BAP Information' column contains a table with BAP Version, FDRCmdLib Version, FID Feature Version(s), FDR-Core Version, Software Edition, and Web Portal Version. The 'System Information' column contains a table with Operating System, Processor, Graphics Processor, Memory, and Time Zone. The 'License' column contains a table with Product Key, Expiry Date, Installation Path, Identity limitation, User identity limitation, and Proxy Support. A red warning message at the top right states 'Product is not activated. Please activate the Product Key.' and an 'Activate' button is located at the bottom right.

BAP Information	
BAP Version	3.2.9.7508
FDRCmdLib Version	6.9.2001.1209
FID Feature Version(s)	[60032084]
FDR-Core Version	6.9.2.38041
Software Edition	N/A
Web Portal Version	3.2.8.6800

System Information	
Operating System	Windows 10 Pro
Processor	Intel(R) Core(TM) i7-6650U CPU @ 2.20GHz
Graphics Processor	Intel(R) Iris(R) Graphics 540
Memory	15.9 GB
Time Zone	(UTC+09:00) Osaka, Sapporo, Tokyo

License	
Product Key	
Expiry Date	
Installation Path	
Identity limitation	
User identity limitation	
Proxy Support	

[Activate](#)

Admin Setting メニューから System Information を開いてください。
続いて取得したBAPのライセンスキーを登録して Activate ボタンを押すと認証が行われます。
認証を行うにはPCがインターネットに接続されている必要があります。

注) ライセンスの登録解除をせずにBAPをアンインストールするとライセンスは登録状態のままとなり、再インストールや別のPCでこのライセンスは使用できなくなります。

ライセンスの登録解除をせずにBAPをアンインストールしてライセンスが使用できなくなった場合には、サポートまでお問い合わせください。

8.2.3. Gorilla BAPのライセンスの登録解除

ライセンスの登録解除

BAPの運用を他のPCに移動させる場合には、必ずインストールしているライセンスを登録解除してください。

Admin Setting -> System Information -> License にある Deactivate ボタンを押して解除されたことを確認してください。

注) ライセンスの登録解除をせずにBAPをアンインストールするとライセンスは登録状態のままとなり、別のPCでこのライセンスは使用できなくなります。

登録解除を行うにはPCがインターネットに接続されている必要があります。

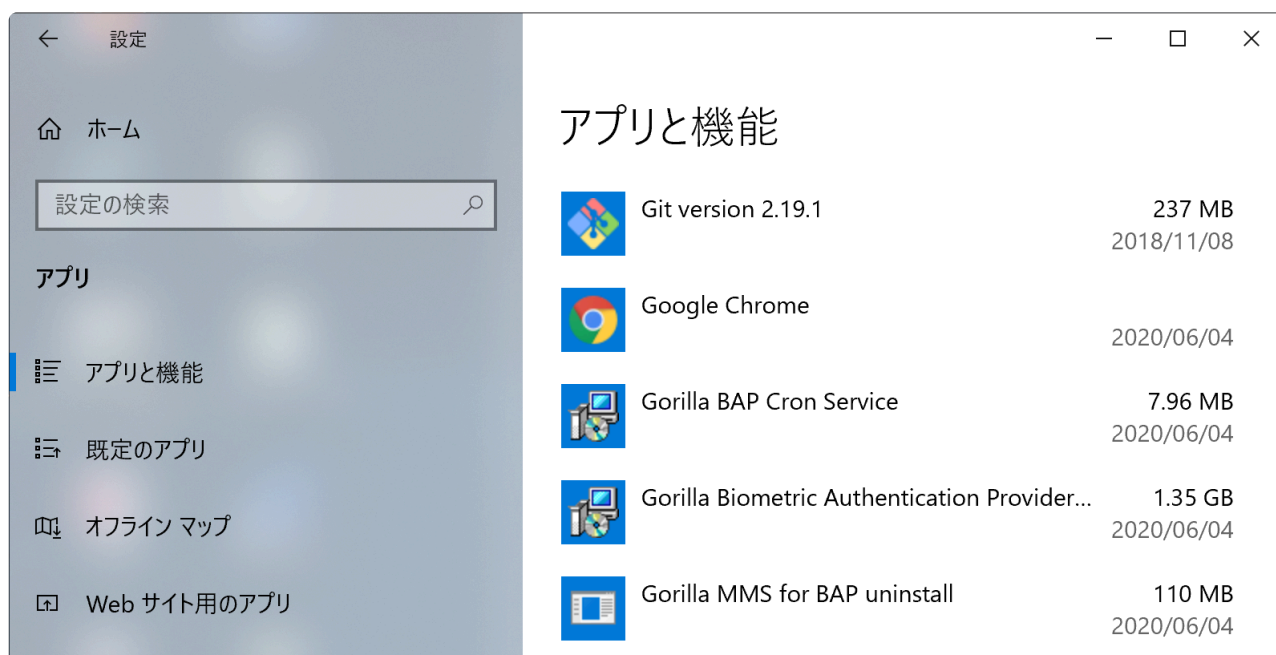
ライセンスの登録解除をせずにBAPをアンインストールしてライセンスが使用できなくなった場合には、サポートまでお問い合わせください。

8.2.4. Gorilla BAPのアンインストール

アンインストール

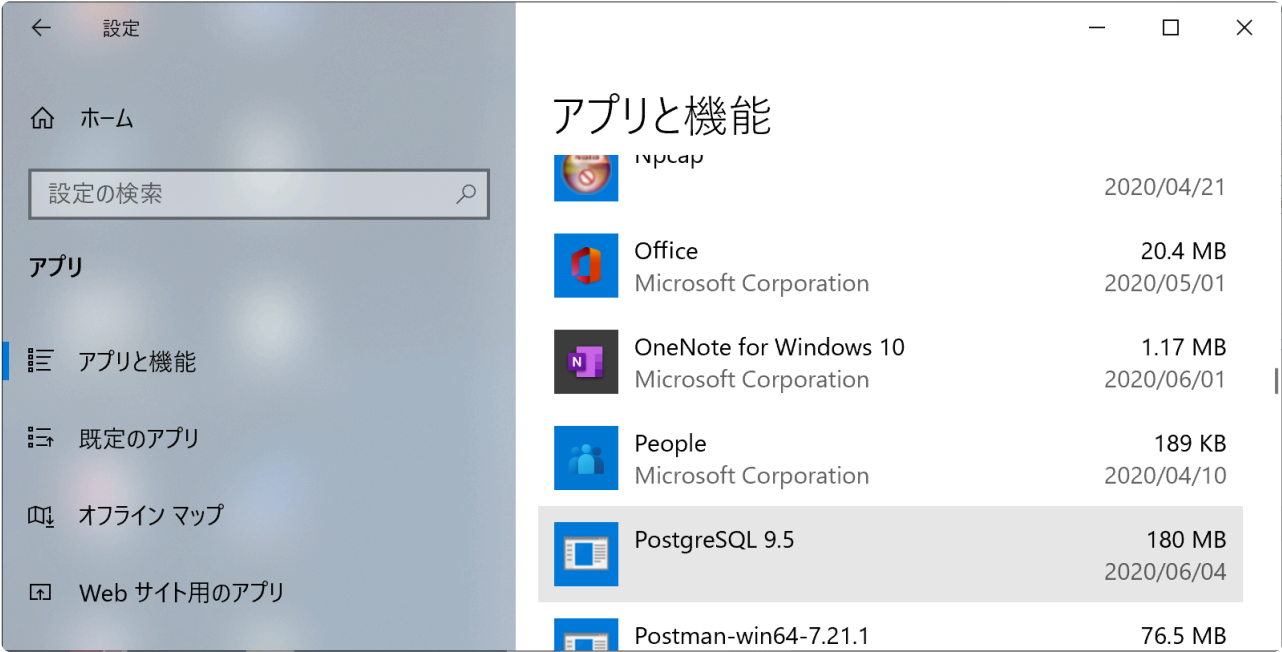
注) ライセンスの登録解除をせずにBAPをアンインストールするとライセンスは登録状態のままとなり、再インストールや別のPCでこのライセンスは使用できなくなります。アンインストールする前にライセンスを登録解除したかもう一度確認してください。

Windowsのスタートメニューから設定を開き、アプリと機能を開いてください。



Gorilla Bap Cron Service をアンインストールしてください

続いて Gorilla Biometric Authentication Provider, Gorilla MMS for BAP uninstall をそれぞれアンインストールしてください。



続いて PostgreSQL をアンインストールしてください。

データフォルダの削除

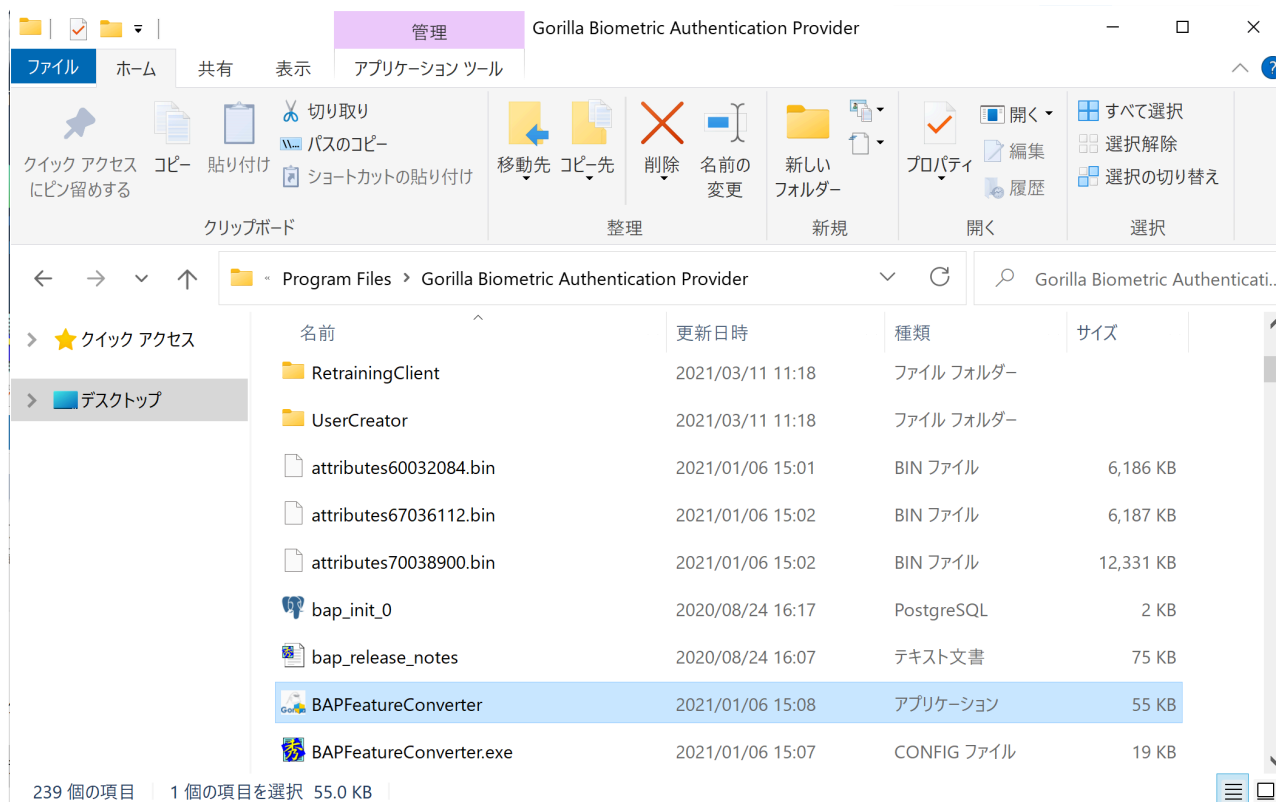
BAPが使用していたデータ書きのフォルダも削除してください。

PostgreSQL	c:\Program Files\PostgreSQL
データ	c:\ProgramData\Gorilla Biometric Authentication Provider
Gorillaデータ	c:\Program Files\Gorilla Biometric Authentication Provider, c:\Program Files\Gorilla Technology Group, Inc.

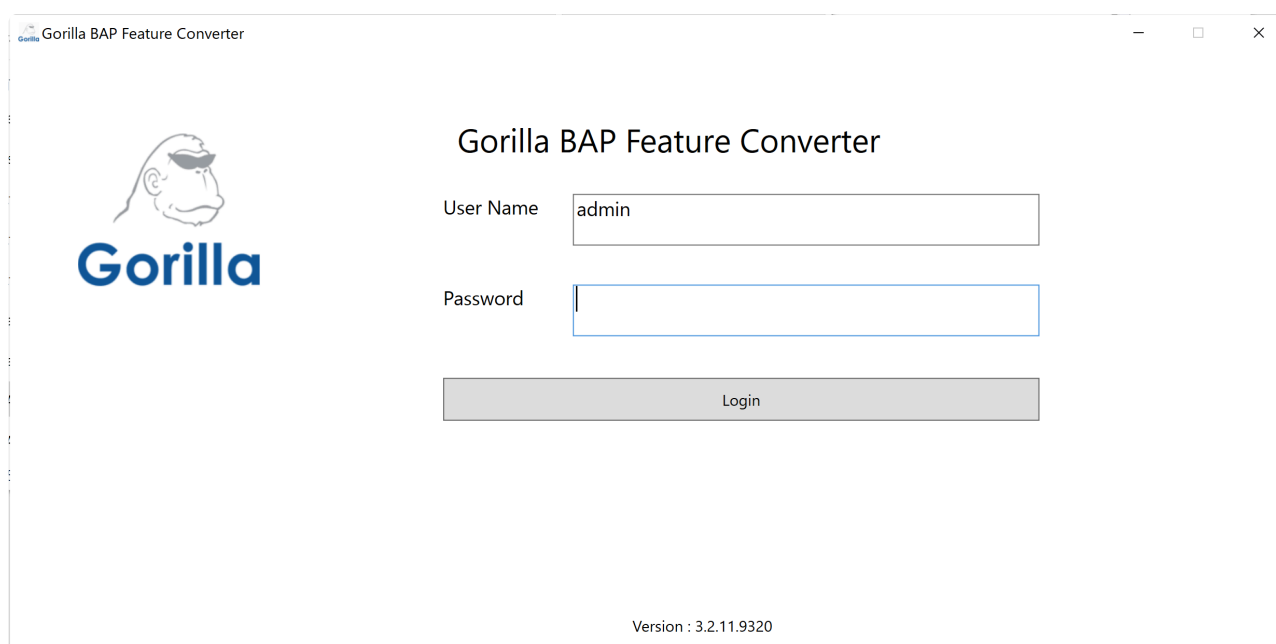
8.2.5. Feature versionの変更

BAPをインストール後にFeature versionを変更する方法

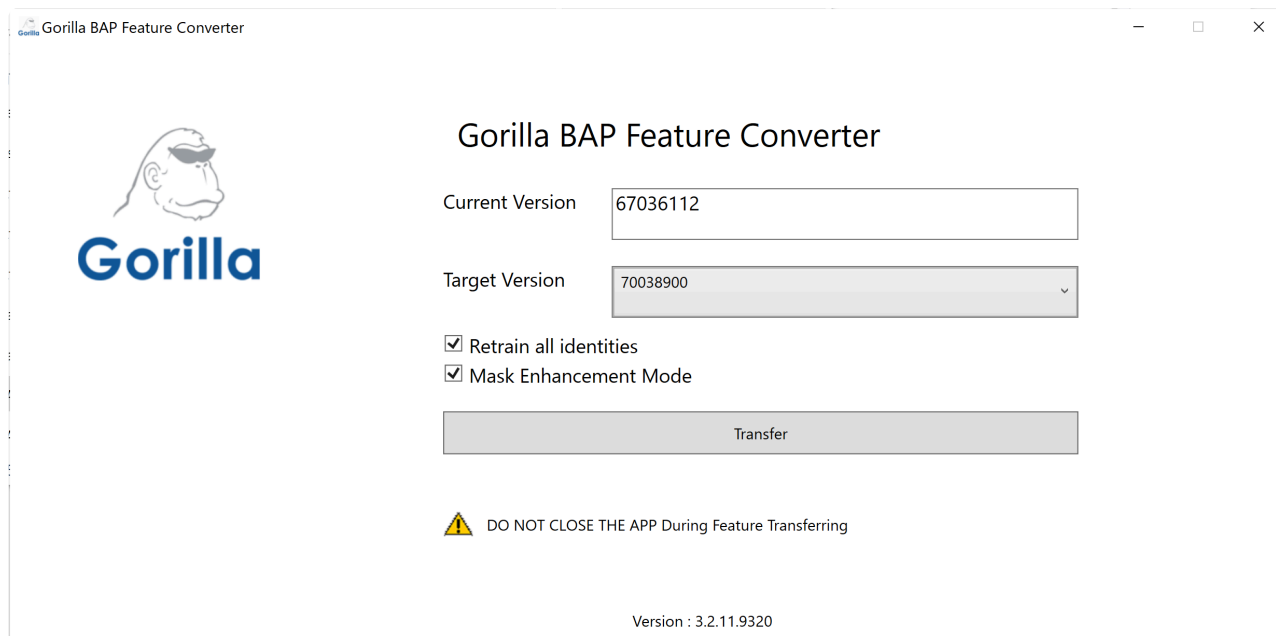
C:¥Program Files¥Gorilla Biometric Authentication Providerをエクスプローラーで開きます。



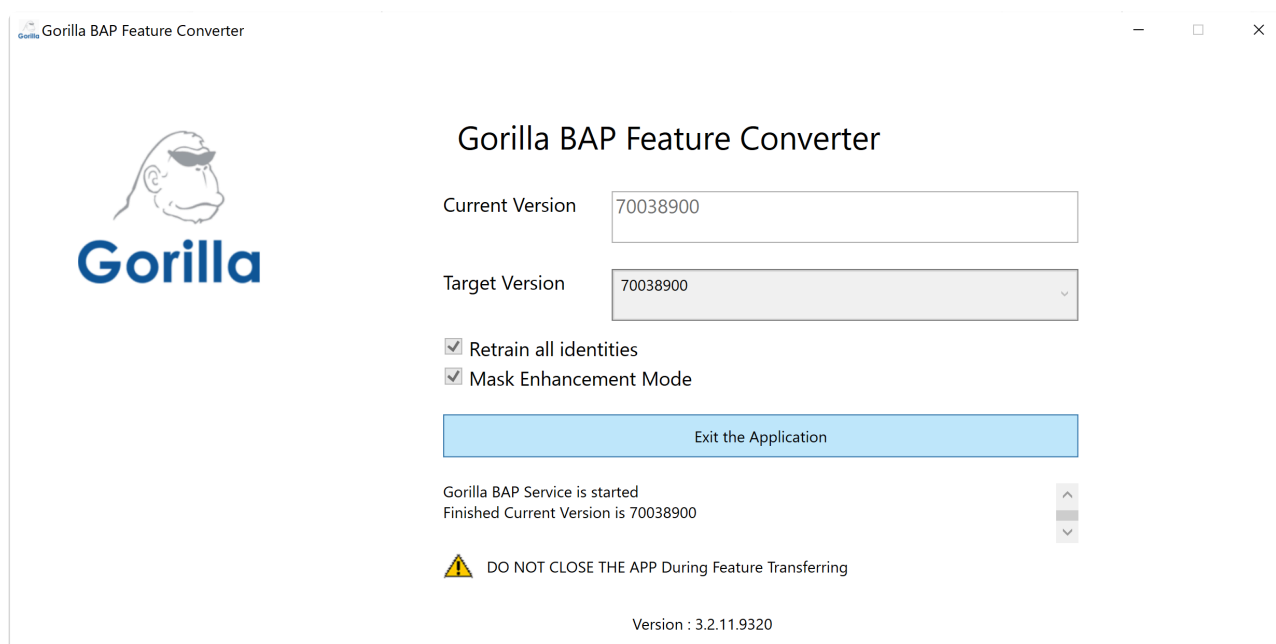
BAPFeatureConverter.exe を実行します。



BAPの管理者アカウントでログインします。



Target Version を 70038900 に選択して Transfer ボタンを押して変換を実行します。



変換が終わると “Finish Current Version is 70038900” と表示されますので Exit the Application ボタンを押して終了します。

Feature versionを確認する方法

Feature versionが更新されていることを確認するには BAPの管理画面にログインして Admin Settings メニューから BAP Information の FID Feature Version(S) を確認します。

Gorilla

BAP

User Settings

Oper

Accounts

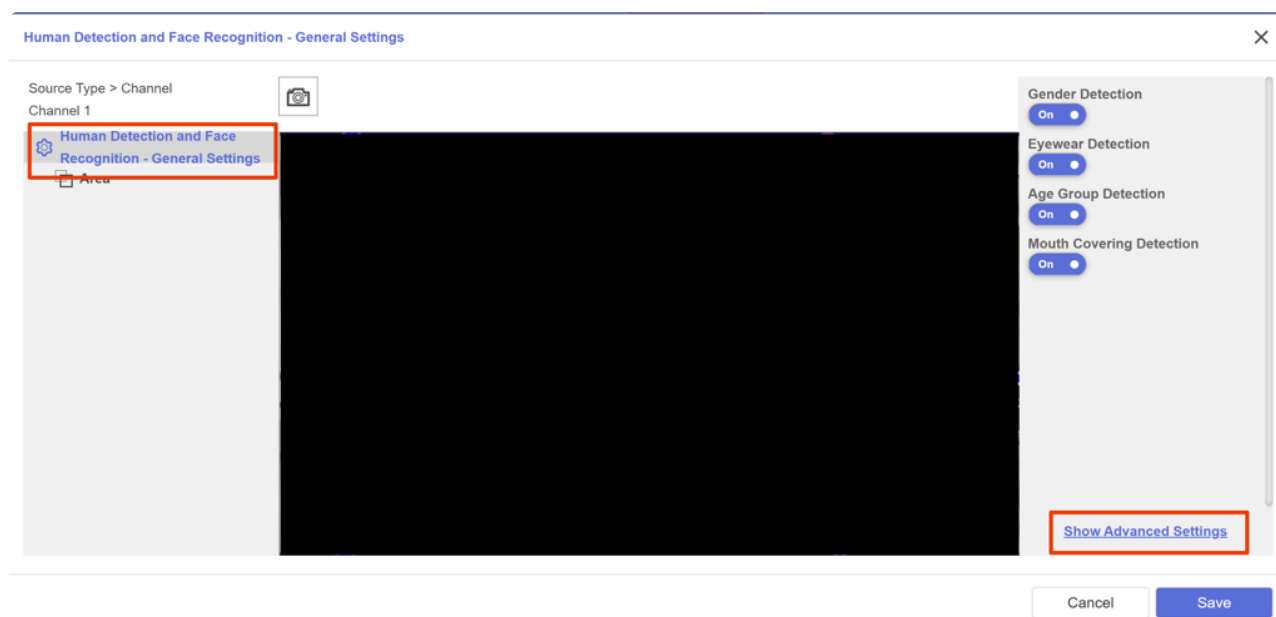
System Information

BAP Information

BAP Version	3.2.11.9320
FDRCmdLib Version	6.10.5003.1320
FID Feature Version(s)	[70038900]
FDR-Core Version	6.10.5.39975
Software Edition	Professional
Web Portal Version	3.2.11.9316

IVARでのFeature versionの変更方法

IVARの管理画面からIVAR Serupメニューを選択して、チャンネルに設定しているプロファイルの、既に設定してある“Human Detection and Face Recognition – General Settings”を開きます。
右下の“Show Advances Settings”を選択すると詳細設定が開きます。



この Feature Version も 70038900 に変更して保存します。

Human Detection and Face Recognition - General Settings

Source Type > Channel
Channel 1

Human Detection and Face Recognition - General Settings

Area

Feature Version

67036112

60032084

67036112

70038900

Eyewear Detection

On

Age Group Detection

On

Mouth Covering Detection

On

Recognizable Factors

Face

Minimum Object Width (px)

130

Maximum Object Width (px)

[Hide Advanced Settings](#)

Cancel

Save

8.3. Gorilla IVARの設置

事前に準備するもの

- ・IVARの運用アカウント Administrator のパスワード
- ・IVARのライセンスキー

インストールする環境

IVARをインストールするWindows10 PCとChromeブラウザを用意してください。
またライセンス登録をしますのでインターネットに接続されている必要があります。

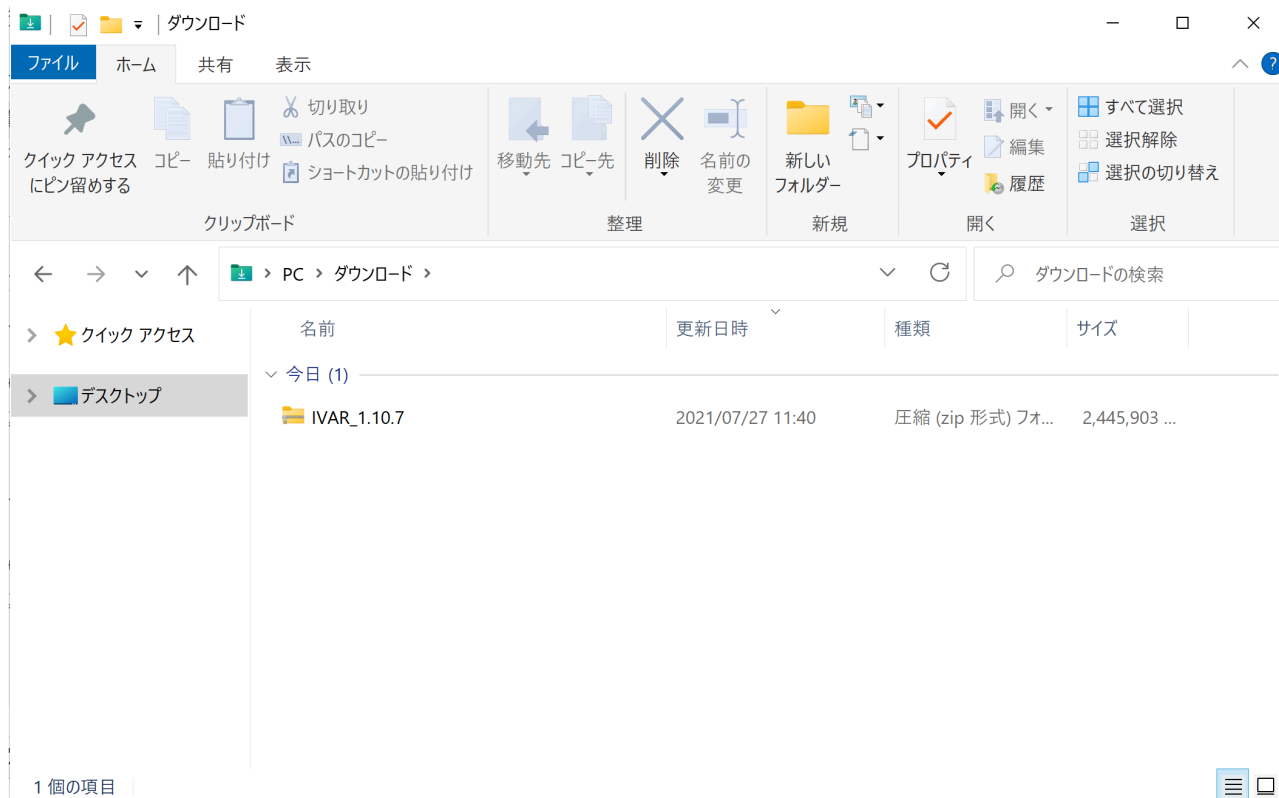
インストールする前のご注意

IVARはMongoDBを使用します。すでにMongoDBがインストールされているとバージョン違いなどの競合により正しくインストールできないことがあるため、MongoDBはアンインストールをお願いします。またデータフォルダも削除してください。

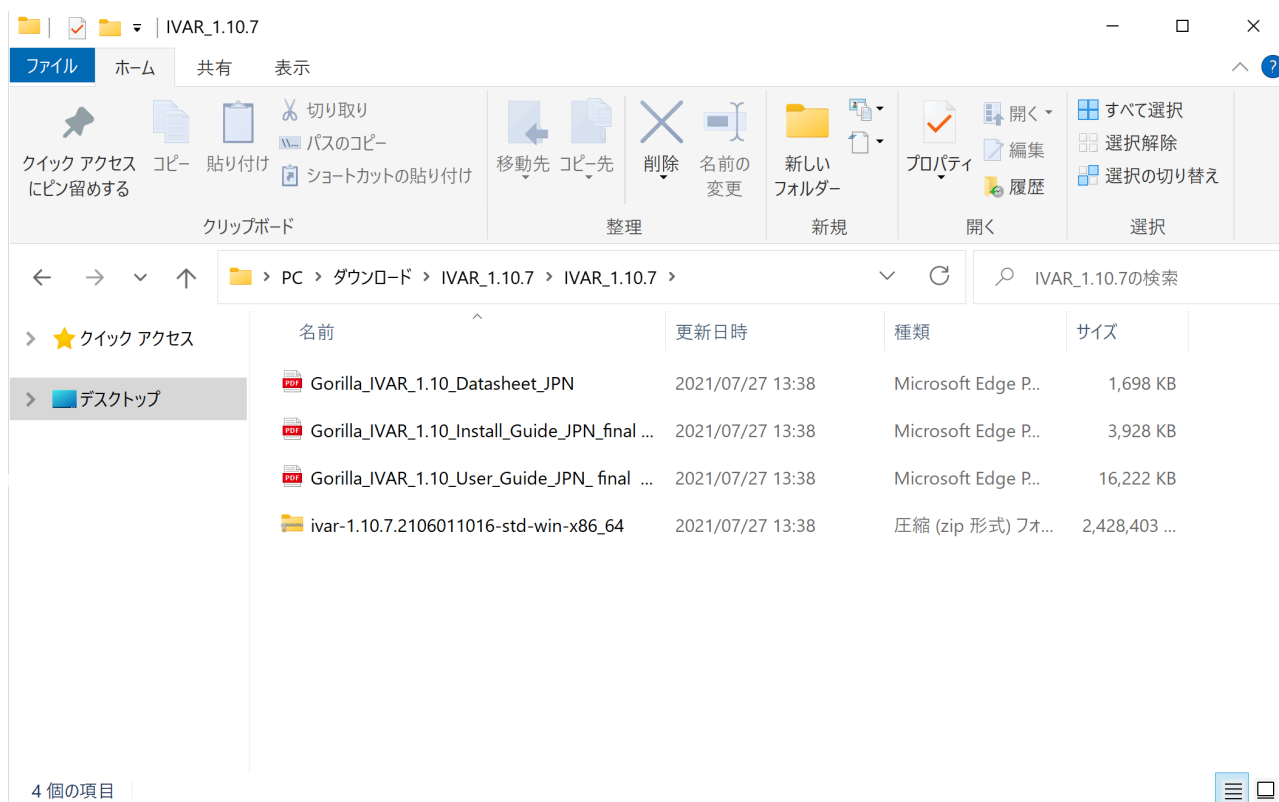
また、Windows10はWindows Updateを使用して最新バージョンへのアップデートをお願いします。

インストール手順

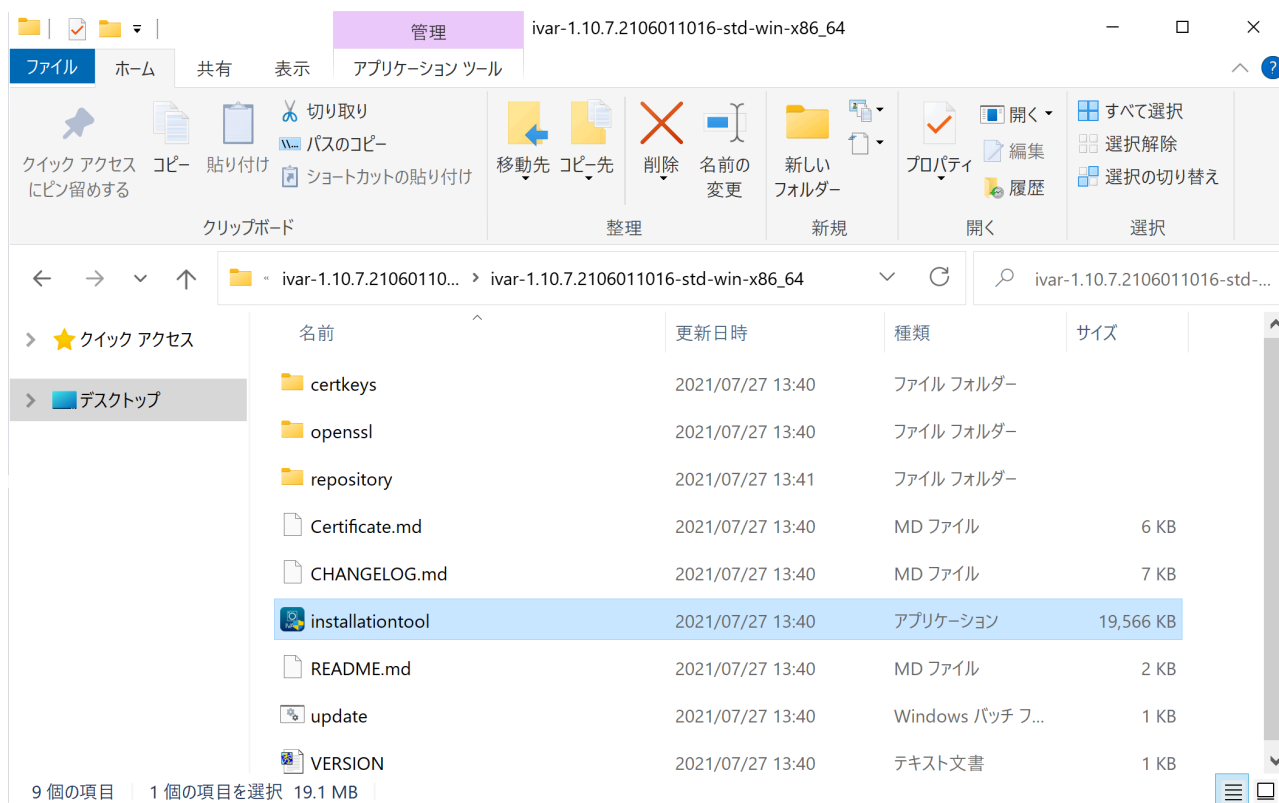
ダウンロードしたzipを展開してください。



展開するとIVARフォルダが出来ますので中を開いてください。



さらにzipファイルがありますので展開して中を開いてください。



installationtool アプリケーションがありますので、これを「管理者として実行」で実行してください。

※管理者権限のないユーザでインストールした場合には正しくインストールが出来ません。



Gorilla IVAR Setup

Setup - Gorilla IVAR

Welcome to the Gorilla IVAR Setup Wizard.

Settings

Next

Quit

起動するとGollira IVARインストーラが表示されますので Nextボタンを押してください。



← Gorilla IVAR Setup

Select Components

Please select the components you want to install.

Default	Select All	Deselect All																					
<table><tr><td>Base</td><td></td><td>^</td></tr><tr><td>Microsoft Visual C++ 2015 Redistrib...</td><td></td><td></td></tr><tr><td>Gorilla MMS</td><td></td><td></td></tr><tr><td>proxy service</td><td></td><td></td></tr><tr><td>Service Manager</td><td></td><td></td></tr><tr><td>Account Manager</td><td></td><td></td></tr><tr><td>Comm Server</td><td></td><td>v</td></tr></table>			Base		^	Microsoft Visual C++ 2015 Redistrib...			Gorilla MMS			proxy service			Service Manager			Account Manager			Comm Server		v
Base		^																					
Microsoft Visual C++ 2015 Redistrib...																							
Gorilla MMS																							
proxy service																							
Service Manager																							
Account Manager																							
Comm Server		v																					

This component will occupy
approximately 136.59 MB on your
hard disk drive.

Next

Cancel

Select Components にはIVARに必要なモジュールが表示されます。
そのまま Next ボタン押してください。



Gorilla IVAR Setup

Ready to Install

Setup is now ready to begin installing Gorilla IVAR on your computer. Installation will use 4.44 GB of disk space.

Install

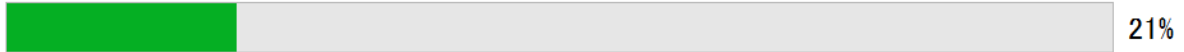
Cancel

インストールの準備ができましたので Install ボタンを押すとインストールが開始されます。



Gorilla IVAR Setup

Installing Gorilla IVAR



Downloading archive "1.10.7.2106011016ivar-core.7z" for component IVAR Core.

Download finished.

[Show Details](#)[Install](#)[Cancel](#)

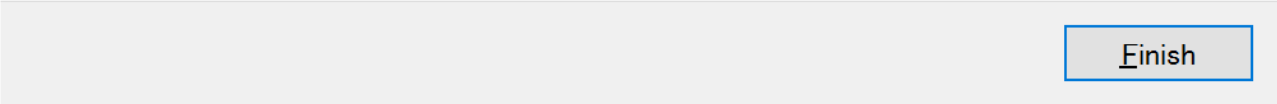
インストールしているファイルが表示され、不定期にプロンプトが表示されますのでインストール中は画面操作をせずにお待ちください。



Gorilla IVAR Setup

Completing the Gorilla IVAR Wizard

Click Finish to exit the Gorilla IVAR Wizard.



Finish

インストールが終わる Finish ボタンを押してください。

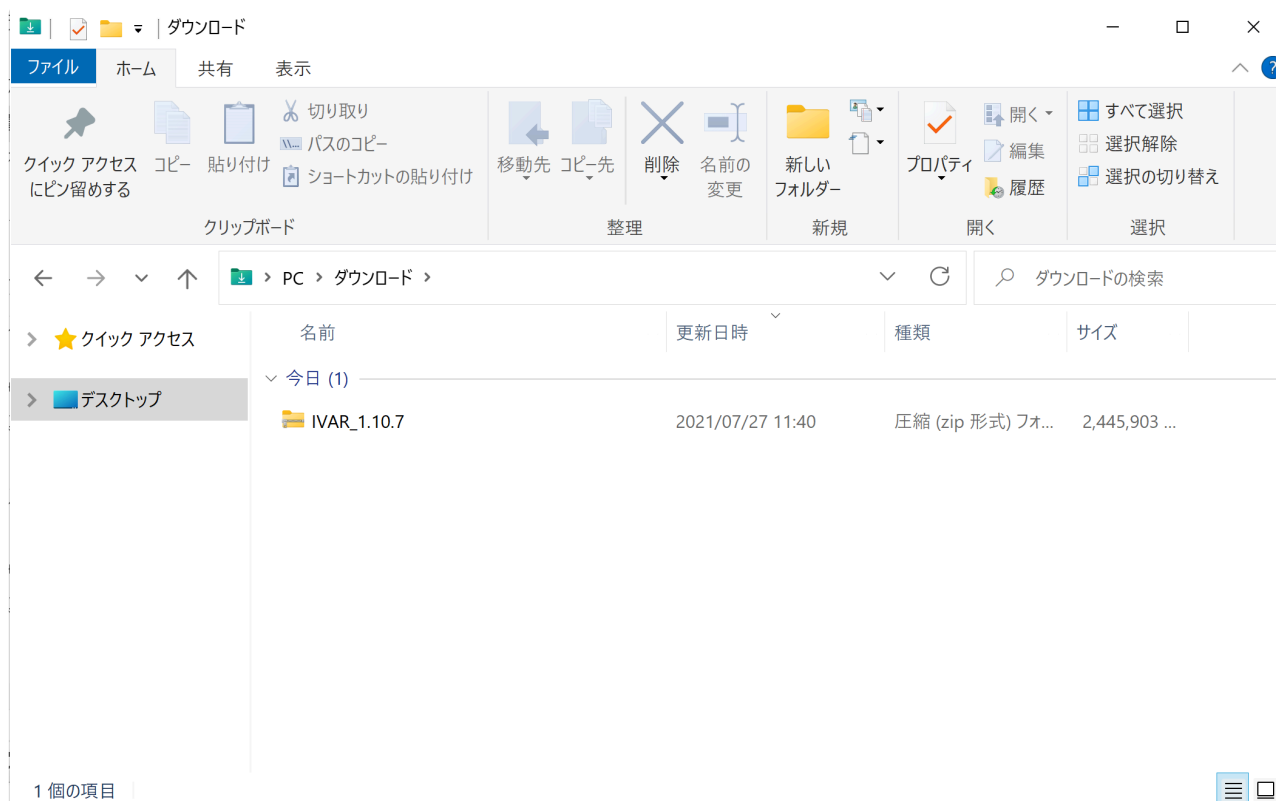
8.3.1. Gorilla IVARのアップデート

IVARを新規インストールする場合には [こちら](#) を参照してください。

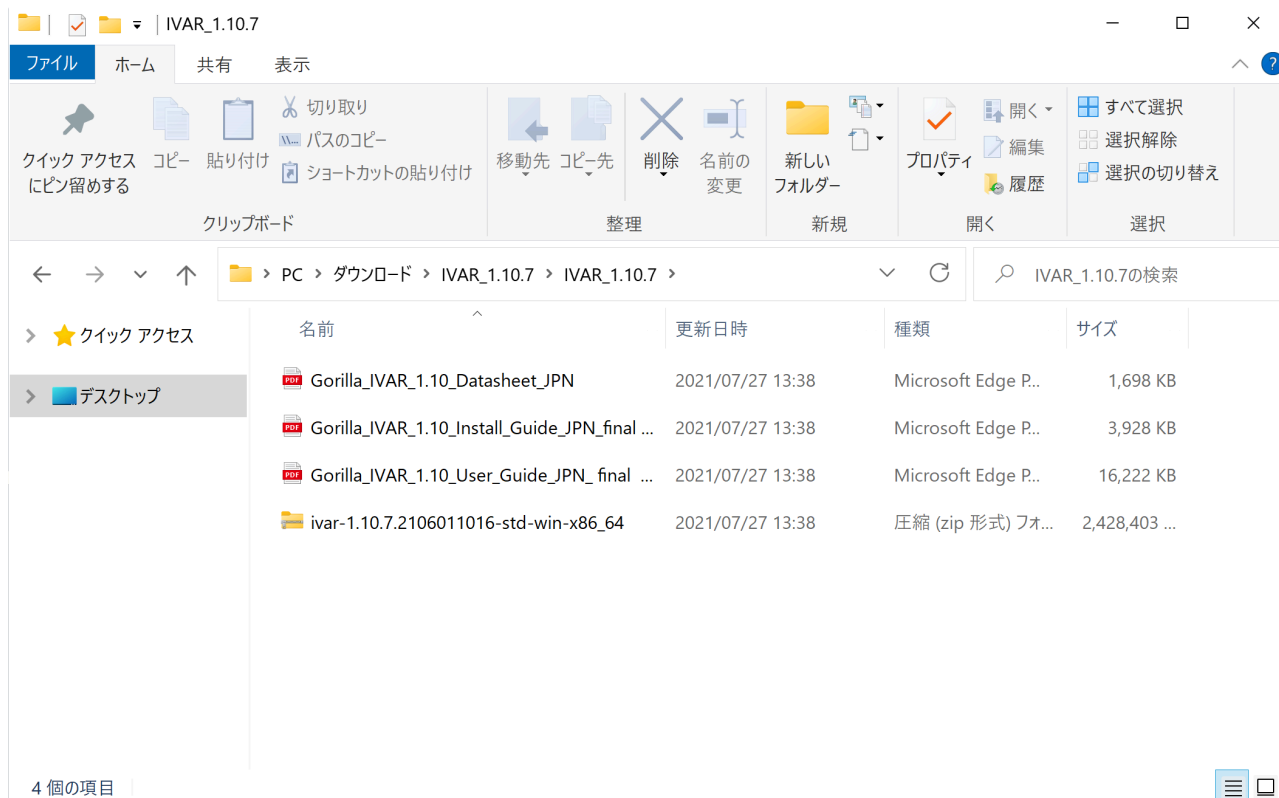
アップデート手順

ダウンロードしたzipを展開してください。

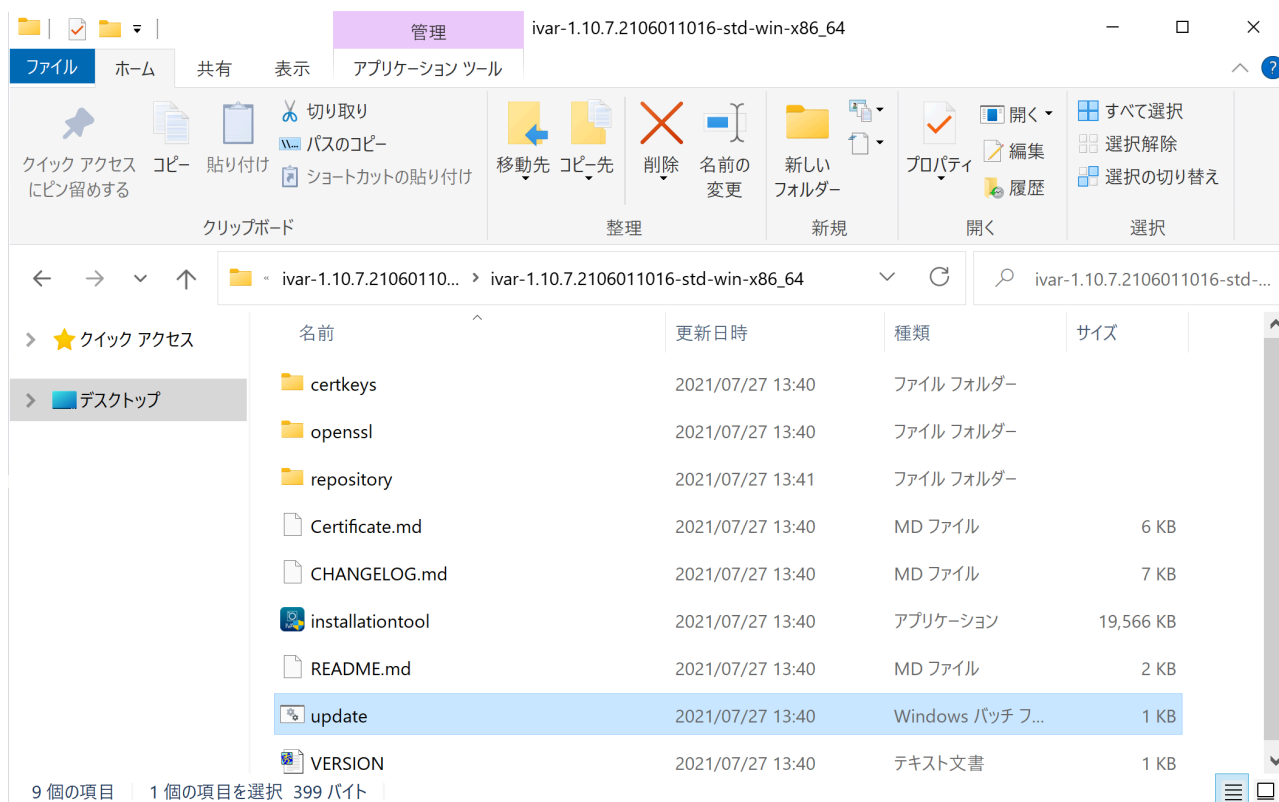
展開するとIVARフォルダが出来ますので中を開いてください。



さらにzipファイルがありますので展開して中を開いてください。



update.bat Windows バッチファイルがありますので、これを実行してください。



実行するとコンソール画面が開きIVARがアップデートされます。

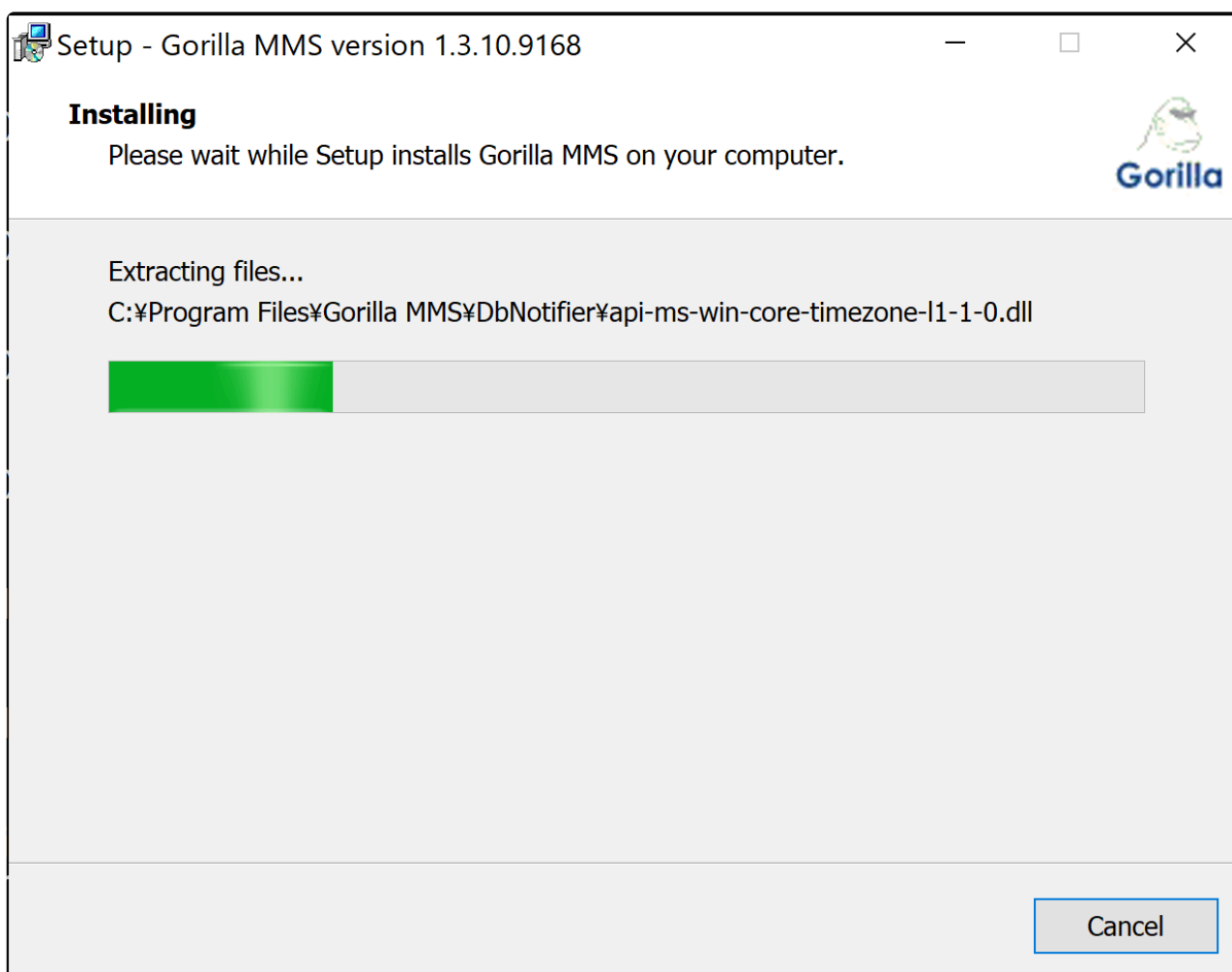
```

C:\WINDOWS\system32\cmd.exe
[3026] [EVENT].installationStarted,update
[3029] perform operation: Execute
[3032] - arguments: taskkill, /f, /im, DbNotifier.exe
[3051] taskkill" started, arguments: "/f /im DbNotifier.exe
[3260] Warning: ?G???[: ?v???Z?X "DbNotifier.exe" ????????????????B

[3261] Done
[3266] undo operation: Execute
[3269] - arguments: taskkill, /f, /im, DbNotifier.exe
[3270] Done
[3277] [UPDATING]mms
[3279] [UPDATING]proxy-service
[3286] perform operation: Execute
[3286] - arguments: C:\Program Files\Gorilla\IVAR\proxy-service\ServiceWrapper.exe, /stop
[3294] C:\Program Files\Gorilla\IVAR\proxy-service\ServiceWrapper.exe" started, arguments: "/stop
[5432] ¥r¥nGorilla proxy-service ¥x83T¥x81[¥x83r¥x83X¥x82¥xCD¥x90¥xB3¥x8F¥xED¥x82¥xC9¥x92¥xE2¥x8E~¥x82¥xB3¥x82¥xEA¥x82¥x
DC¥x82¥xB5¥x82¥xBD¥x81"B¥r¥n¥r¥n
[5434] Done
[5438] [UPDATING]account-manager
[5441] perform operation: Execute
[5442] - arguments: C:\Program Files\Gorilla\IVAR\account-manager\ServiceWrapper.exe, /stop
[5455] C:\Program Files\Gorilla\IVAR\account-manager\ServiceWrapper.exe" started, arguments: "/stop
[7602] ¥r¥nGorilla Account Manager ¥x83T¥x81[¥x83r¥x83X¥x82¥xC9¥x90¥xB3¥x8F¥xED¥x82¥xC9¥x92¥xE2¥x8E~¥x82¥xB3¥x82¥xEA¥x82
¥xDC¥x82¥xB5¥x82¥xBD¥x81"B¥r¥n¥r¥n
[7602] Done
[7606] [UPDATING]comm-server
[7611] perform operation: Execute
[7612] - arguments: C:\Program Files\Gorilla\IVAR\comm-server\ServiceWrapper.exe, /stop
[7624] C:\Program Files\Gorilla\IVAR\comm-server\ServiceWrapper.exe" started, arguments: "/stop

```

アップデート中にMMSのインストーラも起動します。



アップデートが終わるとコンソール画面が閉じます。

IVARを 1.10にアップデートした場合、BAPのFeature versionも 70038900 に変更する必要があります。
Feature versionの変更方法につきましては [こちら](#) を参照してください。

IVARを1.10にアップデートして、BAPのFeature versionを 70038900 に変更するとアップデートは完了です。

8.3.2. Gorilla IVARの初期設定とライセンス登録

初期設定

IVARのインストールが終わるとChromeを使用して下記のアドレスを開いてください。

注) IVARをインストールしたPCでChromeを操作することを想定しています。
異なるPCで操作する場合には“localhost”を環境に合わせて変更してください。

アドレス : <https://localhost:8002/>

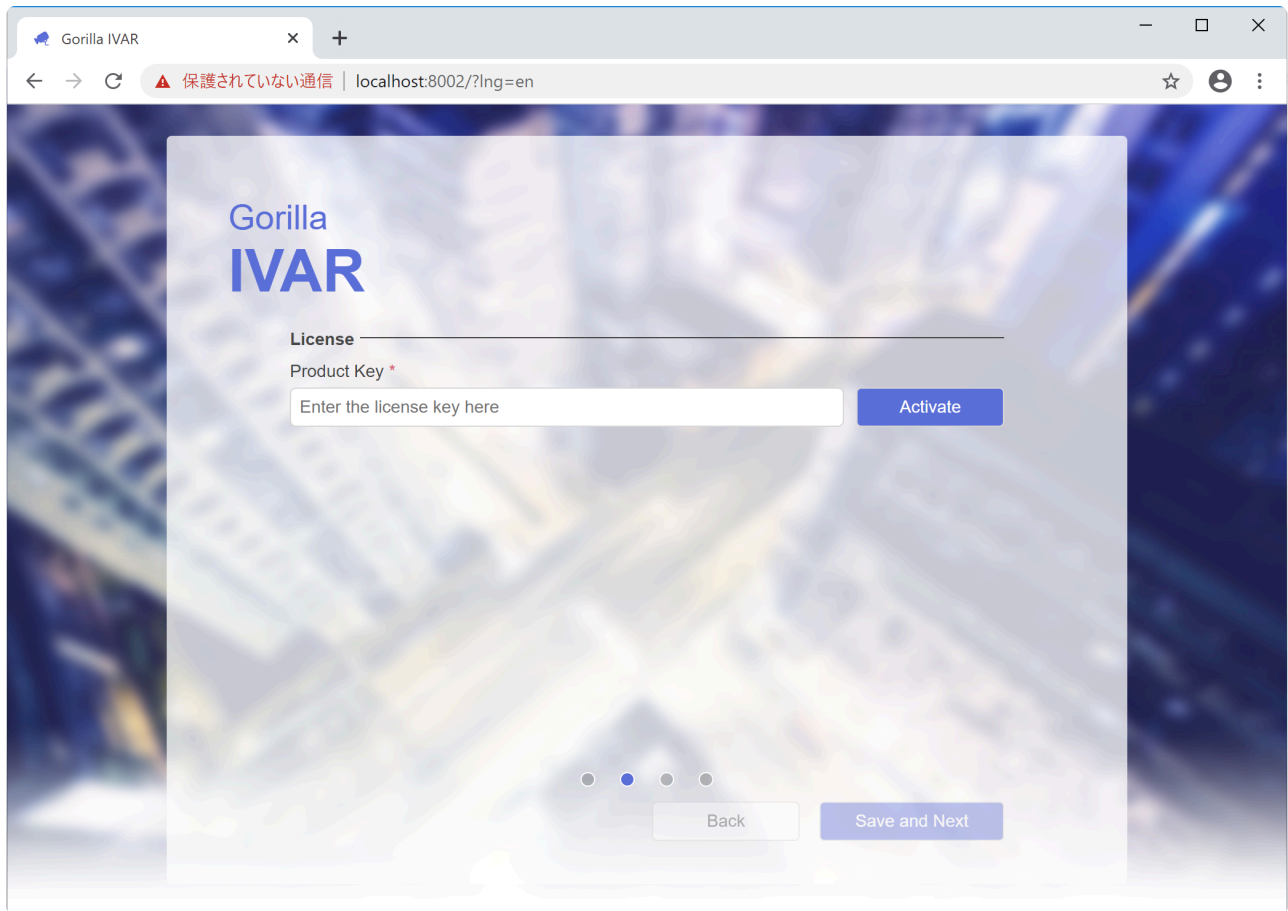


この接続ではプライバシーが保護されません

注) httpsでセルフサインの証明書を使用しているため最初に警告が表示されます。
この警告が表示された場合には 詳細設定 ボタンを押してlocalhostにアクセスする(安全ではありません) リンクをクリックして開いてください。

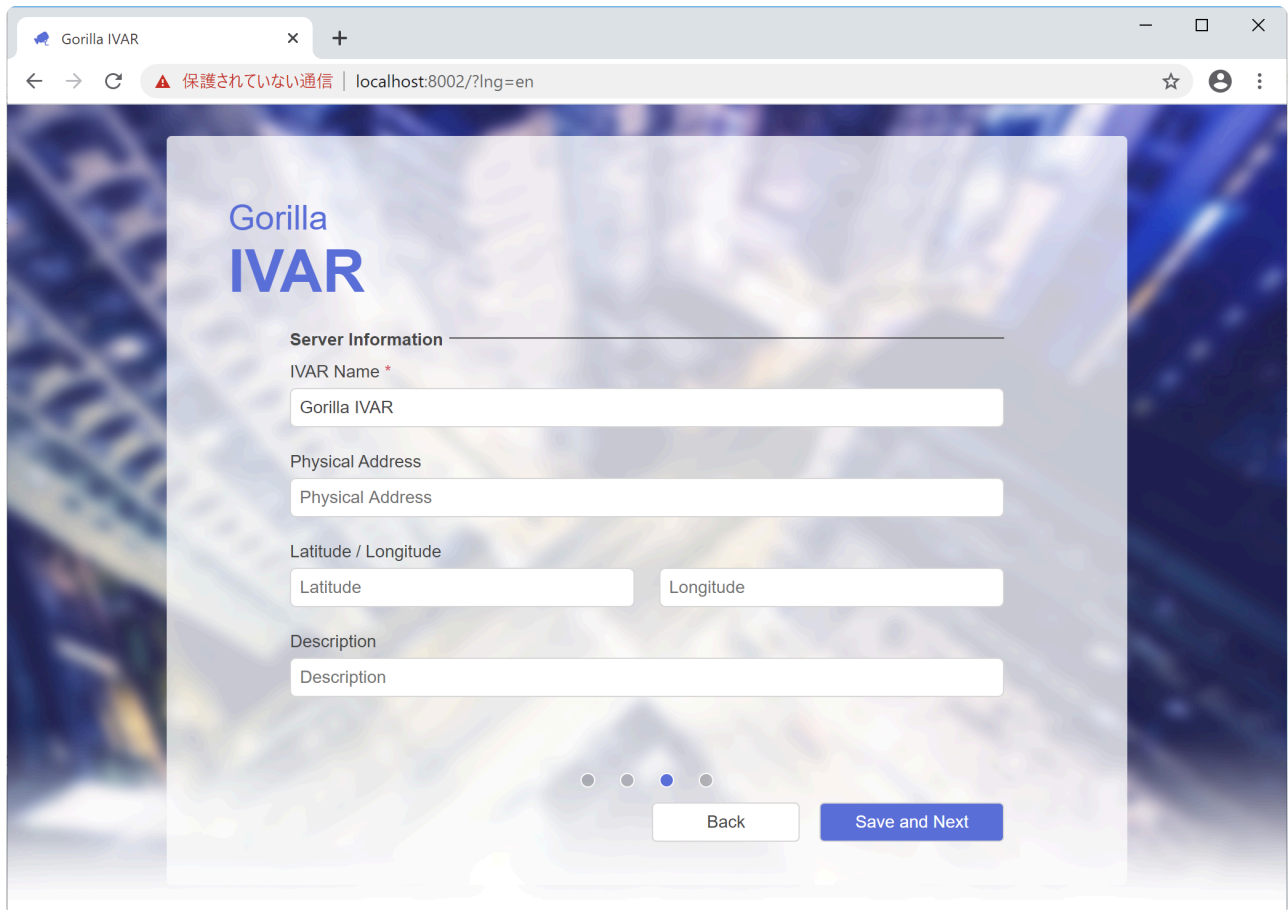
IVARのAdministratorのパスワードを登録して Create ボタンを押してください。

ライセンスの登録



続いて取得したIVARのライセンスキーを登録して **Activate** ボタンを押すと認証が行われます。
認証を行うにはPCがインターネットに接続されている必要があります。

認証が終わりますと **Save and Next** ボタンを押してください。



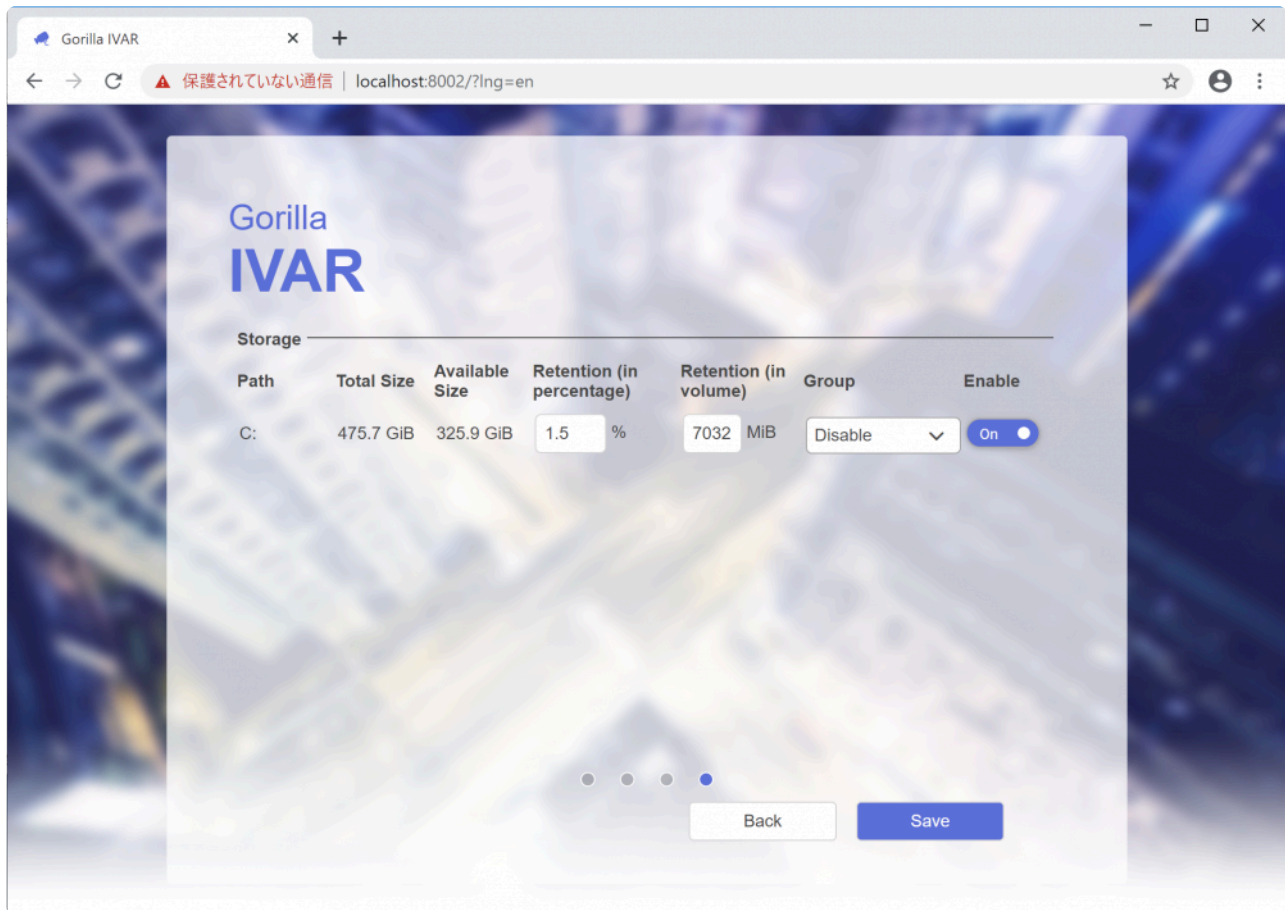
The screenshot shows a web browser window with the title 'Gorilla IVAR'. The address bar shows 'localhost:8002/?lng=en'. The main content area has a blue header with the 'Gorilla IVAR' logo. Below the logo is a section titled 'Server Information' with a horizontal line. The form contains the following fields:

- IVAR Name ***: A text input field containing 'Gorilla IVAR'.
- Physical Address**: A text input field containing 'Physical Address'.
- Latitude / Longitude**: Two text input fields. The first is labeled 'Latitude' and the second is labeled 'Longitude'.
- Description**: A text input field containing 'Description'.

At the bottom of the form, there are four small circular progress indicators, with the third one (under 'Description') being blue. Below the indicators are two buttons: 'Back' and 'Save and Next'.

続いて Server Information の登録画面となりますが特に初期設定のままで Save and Next ボタンを押してください。

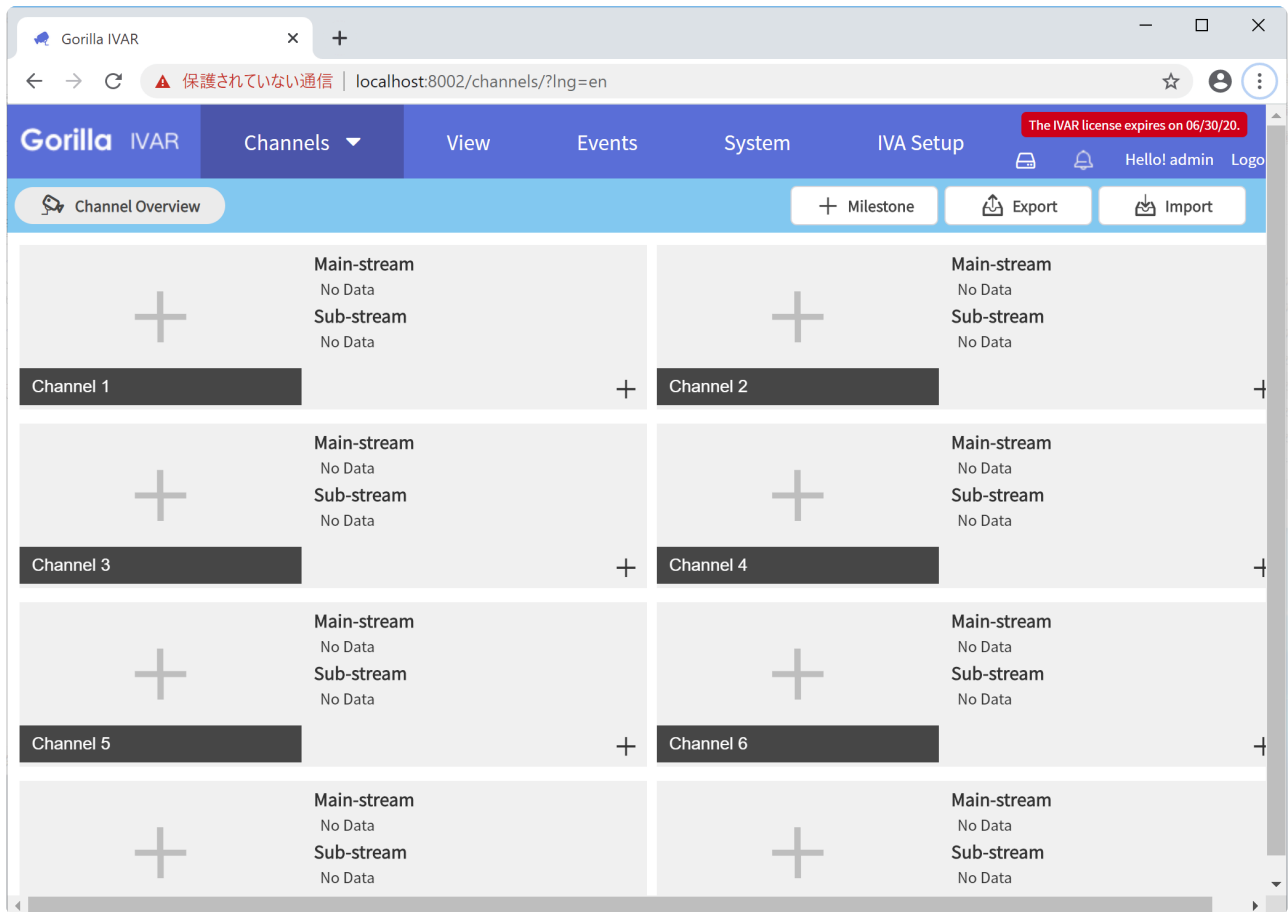
注) 情報を登録するにはインストール後の設定画面で行うことができます。



続いて Storage の登録画面となります。ビデオデータを保存するドライブを指定してください。

設定が終わりますと Save ボタンを押してください。

これで初期設定が完了してチャンネルの画面に移行します



ライセンスの確認

登録したライセンスを確認するには、System メニューから About を開いてください。

IVAR Information

ID	{00000000-0000-1000-1000-100000000000}
Version	ivar-1.8.16.42905-std-win-x86_64
Total Channels	8
Maximum Analytic Channels	2
Analytic Core Channels	All : 1 Business Intelligence : 0 Human Detection : 1 Behavior Analysis : 0 License Plate : 0
Analytic Channel Expiry Date	2099-01-01 21:00:00
Analytic Core Version	BI-CORE v2.3.3 FDR-CORE v6.7.0 IPM-CORE v3.3.0 ODC-CORE v1.5.0 LPR-CORE v5.8.1 MMS/MMSC v1.3.1

General Settings

IVAR Name	Gorilla IVAR
Physical Address	
Latitude	
Longitude	
Description	

[Revert](#) [Save](#)

License

Product Key	
Details	License Check Verified
Expiry Date	

[Deactivate](#) [Resync](#)

[System Information](#) ⓘ

License に登録したライセンスが表示されます。

注) ライセンスの登録解除をせずにIVARをアンインストールするとライセンスは登録状態のままとなり、再インストールや別のPCでこのライセンスは使用できなくなります。

ライセンスの登録解除をせずにIVARをアンインストールしてライセンスが使用できなくなった場合には、サポートまでお問い合わせください。

8.3.3. Gorilla IVARのライセンスの登録解除

ライセンスの登録解除

IVARの運用を他のPCに移動させる場合には、必ずインストールしているライセンスを登録解除してください。

System -> About -> License にある Deactivate ボタンを押して解除されたことを確認してください。

注) ライセンスの登録解除をせずにIVARをアンインストールするとライセンスは登録状態のままとなり、再インストールや別のPCでこのライセンスは使用できなくなります。

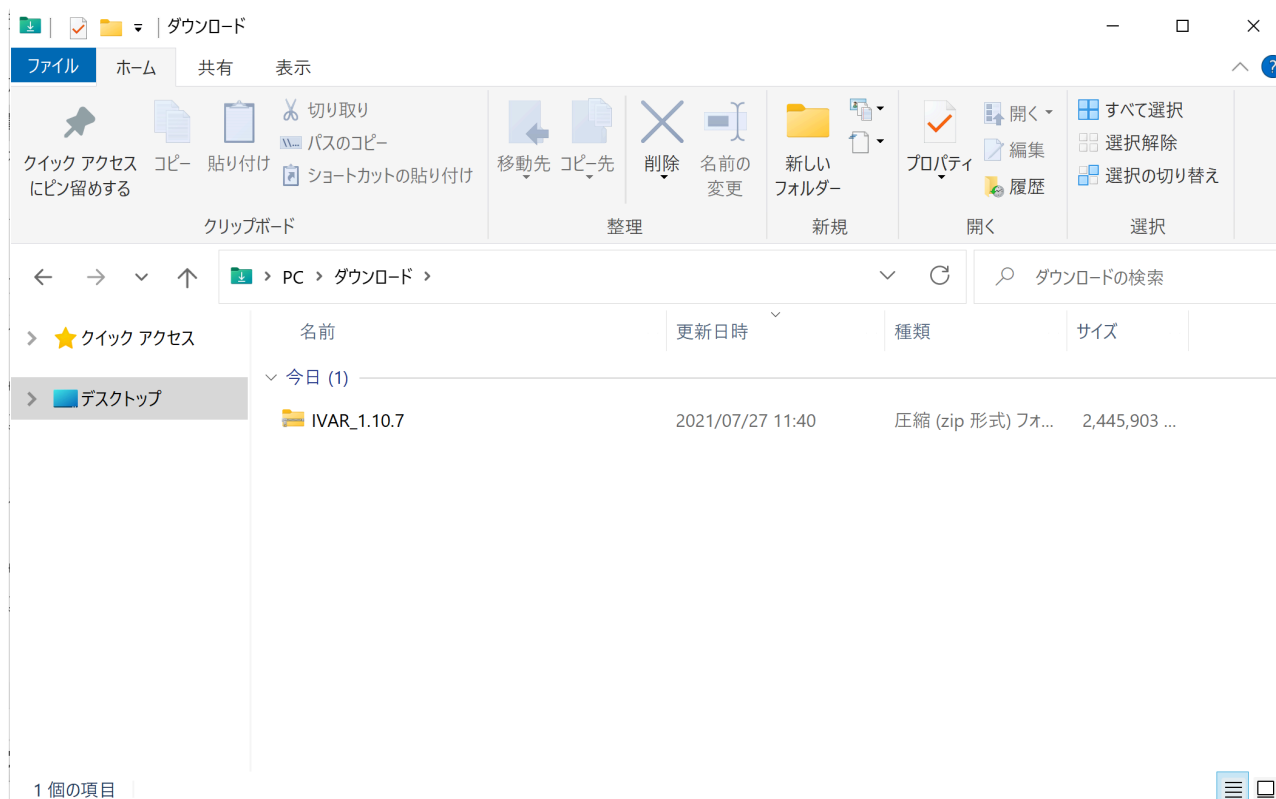
登録解除を行うにはPCがインターネットに接続されている必要があります。

ライセンスの登録解除をせずにIVARをアンインストールしてライセンスが使用できなくなった場合には、サポートまでお問い合わせください。

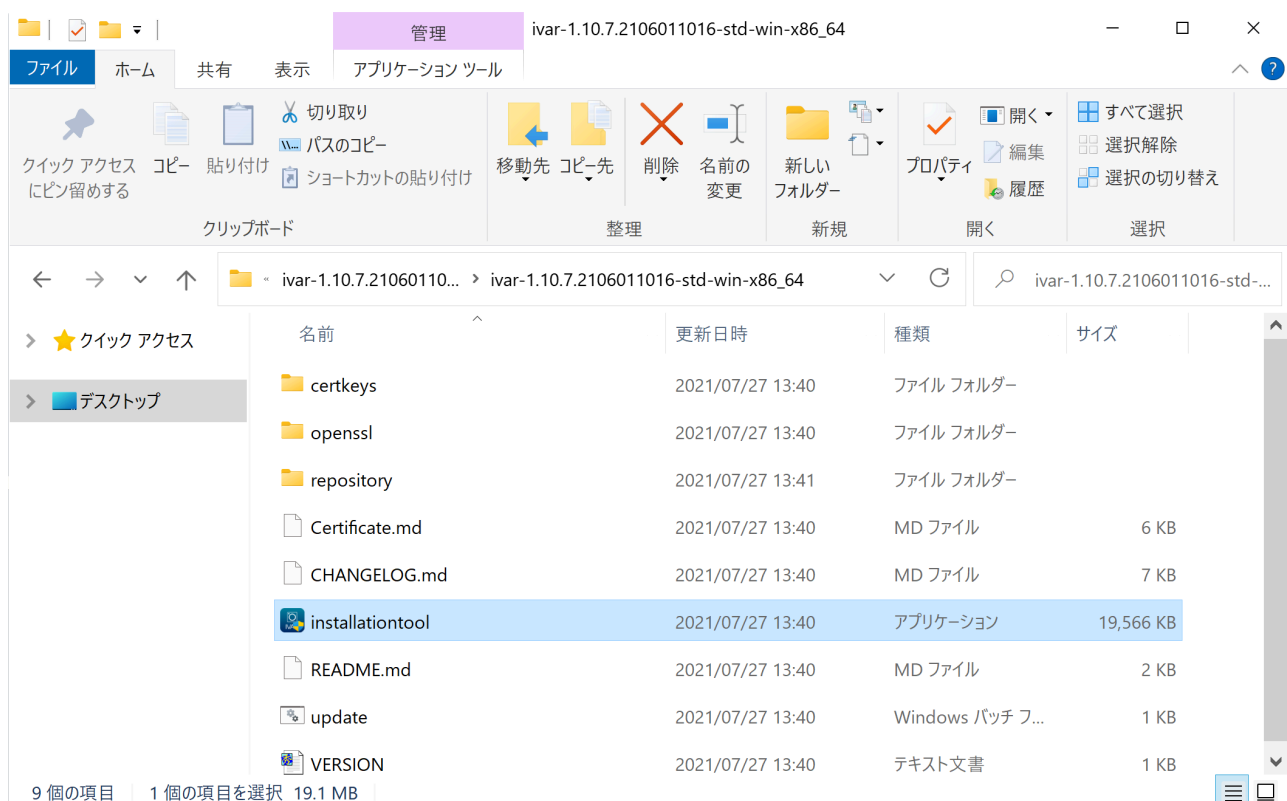
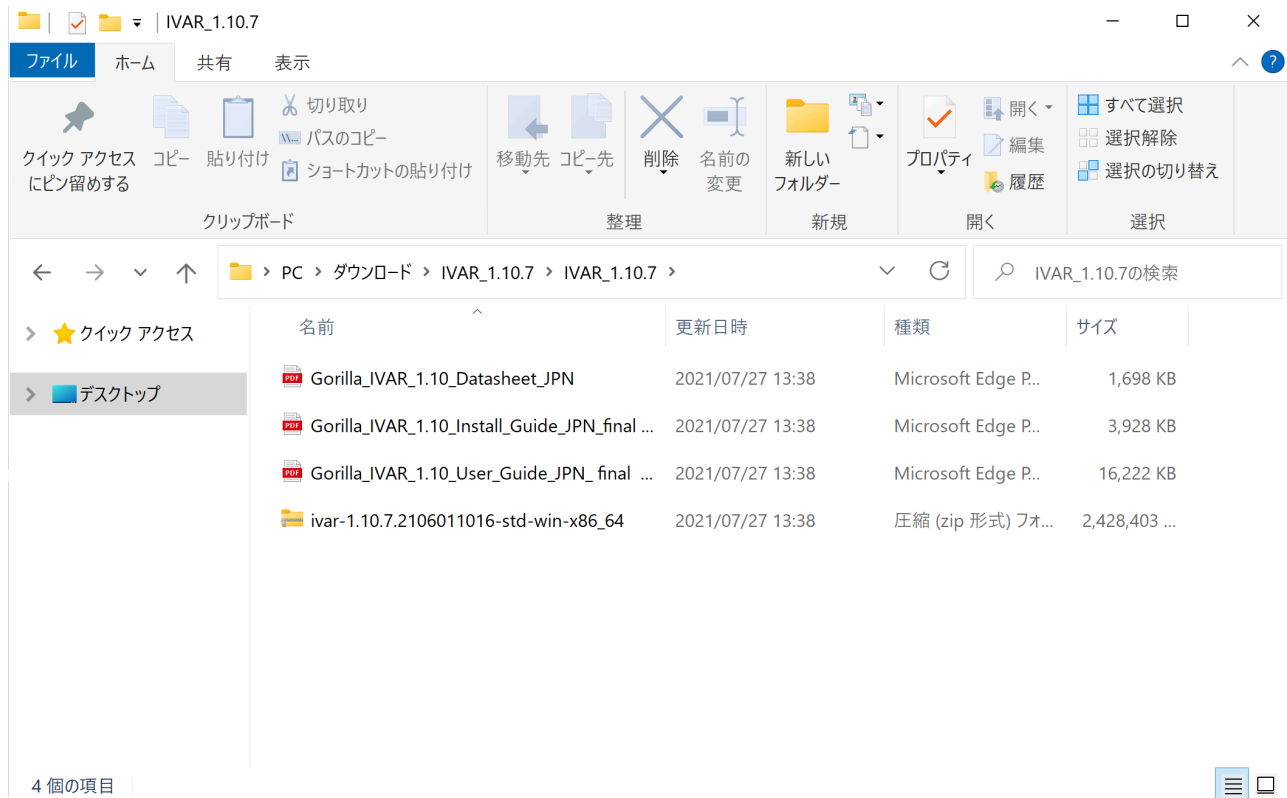
8.3.4. Gorilla IVARのアンインストール

アンインストール

注) ライセンスの登録解除をせずにIVARをアンインストールするとライセンスは登録状態のままとなり、再インストールや別のPCでこのライセンスは使用できなくなります。アンインストールする前にライセンスを登録解除したかもう一度確認してください。



インストール時に取得したzipを展開し、さらにzipファイルがありますので展開して中を開いてください。



installationtool アプリケーションがありますので、これを実行してください。



Gorilla IVAR Setup

Setup - Gorilla IVAR

Welcome to the Gorilla IVAR Setup Wizard.

- ☐ Add or remove components
- ☐ Update components
- ☒ Remove all components

Next

Quit

起動して Setup – Gollira IVAR から Remove all components を選択してください。
そして Next ボタンを押してください。

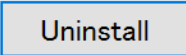
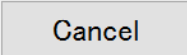


Gorilla IVAR Setup

Ready to Uninstall

Setup is now ready to begin removing Gorilla IVAR from your computer.

The program directory C:¥Program Files¥GorillaIVAR will be deleted completely, including all content in that directory!

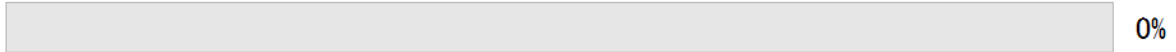
UninstallCancel

さらに Uninstall ボタンを押すとアンインストールが開始されます。

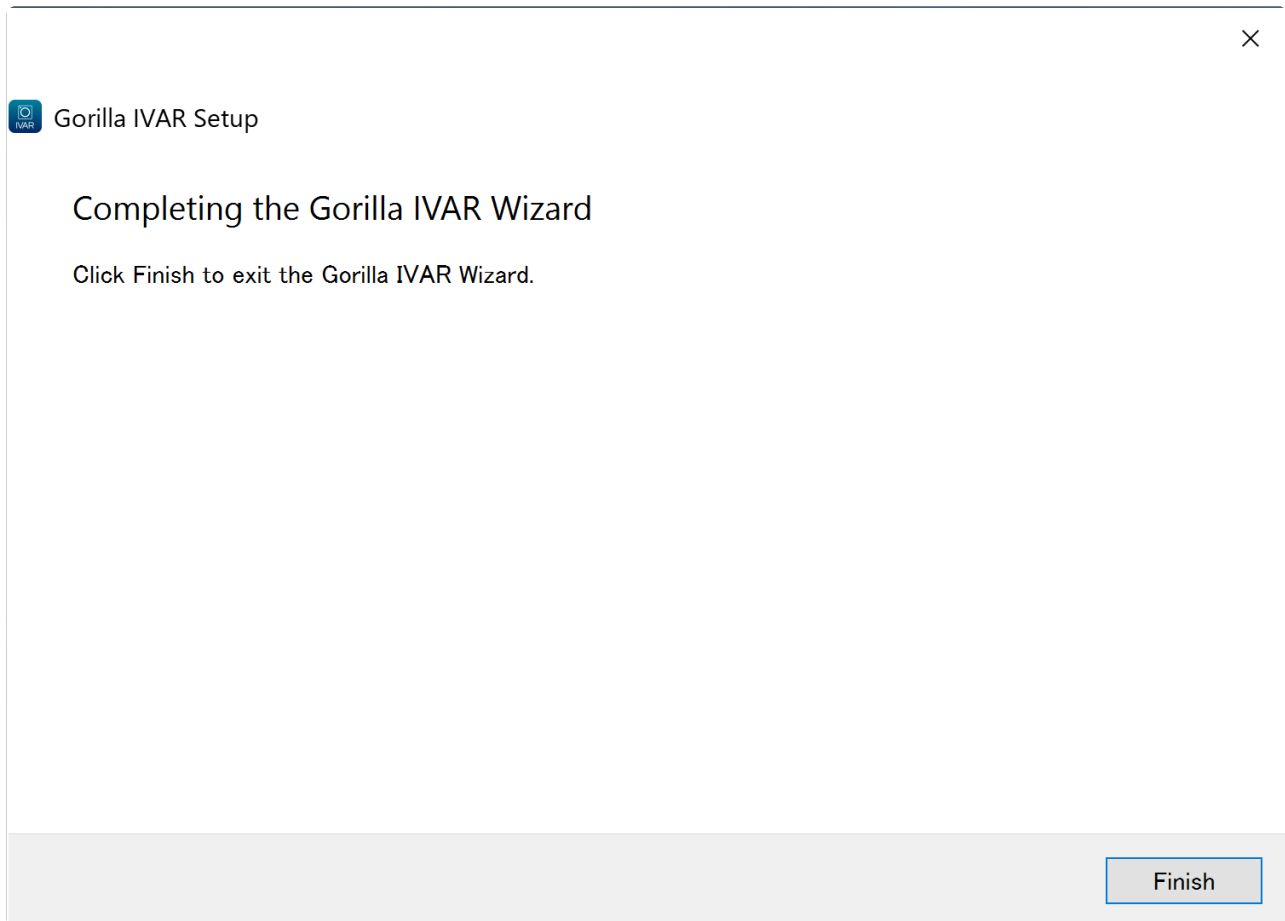


Gorilla IVAR Setup

Uninstalling Gorilla IVAR

[Show Details](#)[Uninstall](#)[Cancel](#)

ファイルがアンインストールされます。



アンインストールが完了すると、Finish ボタンを押してください。

データフォルダの削除

IVARが使用していたデータ書きのフォルダも削除してください。

ビデオデータ	c:¥records
Gorillaデータ	c:¥Program Files¥Gorilla MMS,c:¥Program Files¥GorillaIVAR

8.4. Gravioへの接続テスト – 人数カウント

Gorilla と Gravioの連携

この章では、Gorillaのカメラ映像を画像認識処理で人数をカウントし、それをトリガとしてGravioに処理データを送信します。

Gravioでは IVARトリガを指定して任意のアクションを起動します。

準備するもの

Gorilla IVAR/BAPをインストールしているPCと、onvifカメラ1台、Gravio GS/HubKitをインストールしているPCとChromeブラウザを準備します。

GravioはEnterpriseのライセンスが必要です。また **HubKit**は**Coordinator**に事前に登録が必要です。

ここではonvifカメラはIO Data社のTS-NA220Wを使用します

設定の流れ

IVARのカメラ+画像認識の設定

1. IVARでonvifカメラを登録
2. 画像認識の設定
3. Liveでの画像認識の確認

GravioとIVAR連携の設定

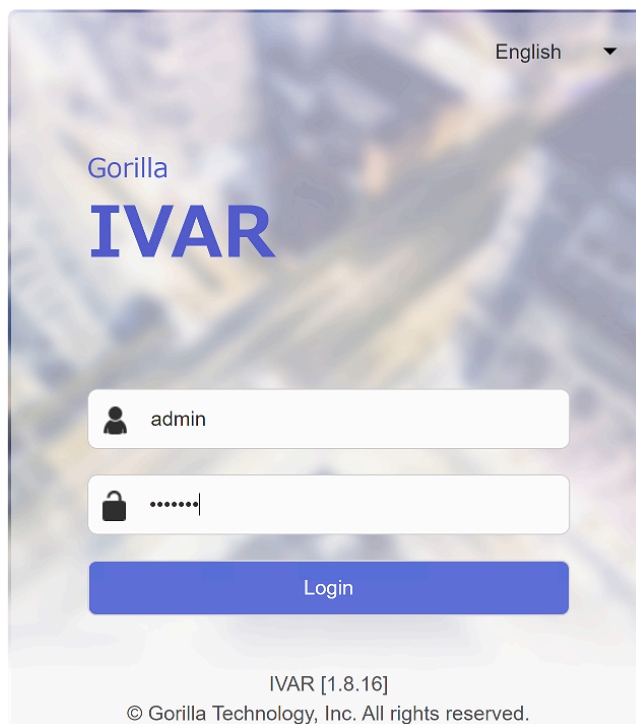
4. Gravio Studio での IVARの設定
5. イベントの登録
6. Gravio Studioでの データビューアでのイベント受信の確認

Gravio IVARトリガーの設定

7. 呼び出し用アクションの作成
8. IVARトリガーの設定
9. IVARトリガーでのアクションの実行の確認

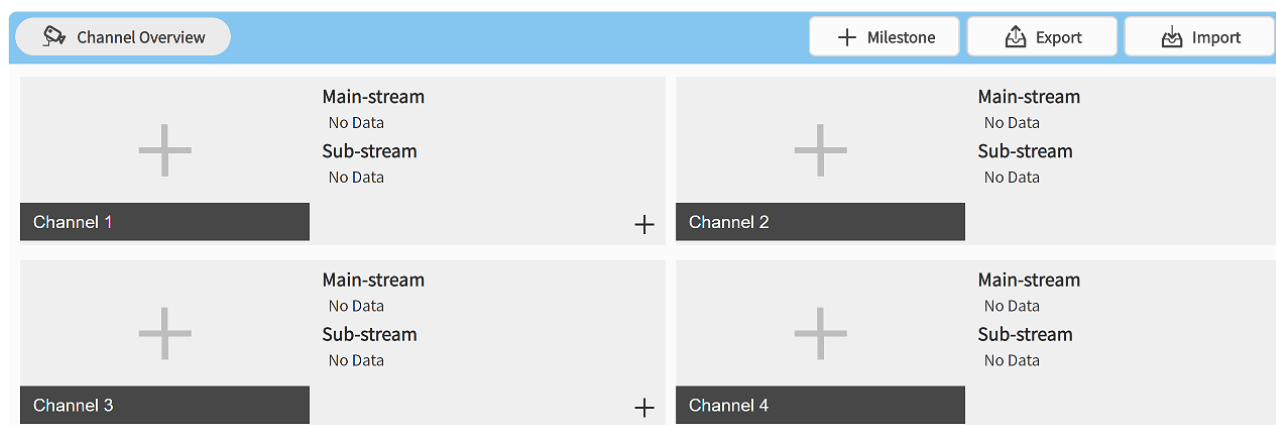
8.4.1. IVARのカメラ+画像認識の設定

IVARのカメラ+画像認識の設定



IVARにadminアカウントを使ってログインします。

1. IVARでonvifカメラを登録します



ログインするとチャンネル画面が表示されます。
+ボタンを押してチャンネル1を登録します。

The screenshot shows the 'Device' tab in the Gravio interface. The 'Device Type' section has four radio button options: 'RTSP' (Manually add an RTSP streaming camera), 'ONVIF' (Automatically search ONVIF Profile S IP cameras), 'Video File' (Upload a video clip), and 'Capture Card/USB Camera' (Select a local streaming device). The 'ONVIF' option is selected. A 'Next' button is located at the bottom right.

使用するカメラはONVIFカメラですのでONVIFを選択して Next ボタンを押します。

The screenshot shows the 'ONVIF Device List' screen. It features a table with columns 'Item', 'IP Address', and 'Device Name'. One device is listed: Item 1, IP Address 192.168.68.140:80, Device Name IODATA%20TS-NA220W. Below the table, there is a section 'Log into the Device : Unselected' with input fields for 'IP Address', 'Username', and 'Password'. 'Back' and 'Next' buttons are at the bottom.

Item ▲	IP Address ▲	Device Name ▲
☐ 1	192.168.68.140:80	IODATA%20TS-NA220W

Log into the Device : Unselected

IP Address:

Username:

Password:

Back Next

同一ネットワークに設置されているカメラが一覧に表示されます。

This screenshot is similar to the previous one, but the first device in the 'ONVIF Device List' table is now selected, indicated by a checkmark in the 'Item' column. The 'Log into the Device' section now shows 'Log into the Device : IODATA%20TS-NA220W'. The input fields are pre-filled with the device's details: IP Address 192.168.68.140, Username admin, and Password (masked with dots).

Item ▲	IP Address ▲	Device Name ▲
☒ 1	192.168.68.140:80	IODATA%20TS-NA220W

Log into the Device : IODATA%20TS-NA220W

IP Address:

Username:

Password:

Back Next

今回はIO DATAのTS-NA220Wを使用しますので、チェックボックスをオンにしてUsernameとPasswordを入力し

て Next ボタンを押します。

Item ▲	Profile Name ▲	Codec ▲	Resolution ▲
1	profile1	H264	1920 x 1080
2	profile2	H264	1280 x 720

Choose an ONVIF Profile

Main-stream: profile2 Sub-stream: Please select a profile

Main-stream: RTSP over TCP On Sub-stream: RTSP over TCP On

Enable Sub-stream Off

Back Apply

カメラで使用できるプロファイルが一覧に表示されますのでMain-streamで選択します。
顔認識の場合は、解像度が高い方が望ましいためフルHD(1920×1080)を選択します。
今回はテストですので profile2 を選択して Apply ボタンを押します。

Device Information

Type ONVIF

IP Address 192.168.68.140

Camera ID {177b22e7-e3bf-da55-177b-22e7e3bfda55}

MAC 34:76:C5:B6:A5:AE

Model TS-NA220W

Vendor IODATA

Version 1.03.09

File Time

Main-stream Profile profile2

Main-stream Path rtsp://192.168.68.140:20930/ipcam_h264s1.sdp

RTSP over TCP Yes

Remove Device

Video Stream Profile Settings

profile1		profile2	
Encoding	h264	Encoding	h264
H.264 Profile	Main	H.264 Profile	Main
Resolution	1920x1080	Resolution	1280x720
Frame Rate	15	Frame Rate	15

カメラが登録されました、削除する場合には Remove ボタンを押します。

View

Device

Settings

Channel:Channel

Description

Location

Longitude Coordinates

Latitude Coordinates

Maximum Retention (in days)

Recording Path

Revert

Save

Revert

Save

Recording Schedule

Recording Type Selection

All

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mon																								
Tue																								
Wed																								
Thu																								
Fri																								
Sat																								
Sun																								

All Main-stream Sub-stream Erase

Reset

Revert

Save

カメラの設定が完了すると自動的に映像を保存しますので、ハードディスクを圧迫しないようにテストでは映像を保存しないように設定を変更します。

Settingボタンを押して、下にある Recording Schedule を開きます。

Recording Schedule

Recording Type Selection

Erase

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mon																								
Tue																								
Wed																								
Thu																								
Fri																								
Sat																								
Sun																								

All Main-stream Sub-stream Erase

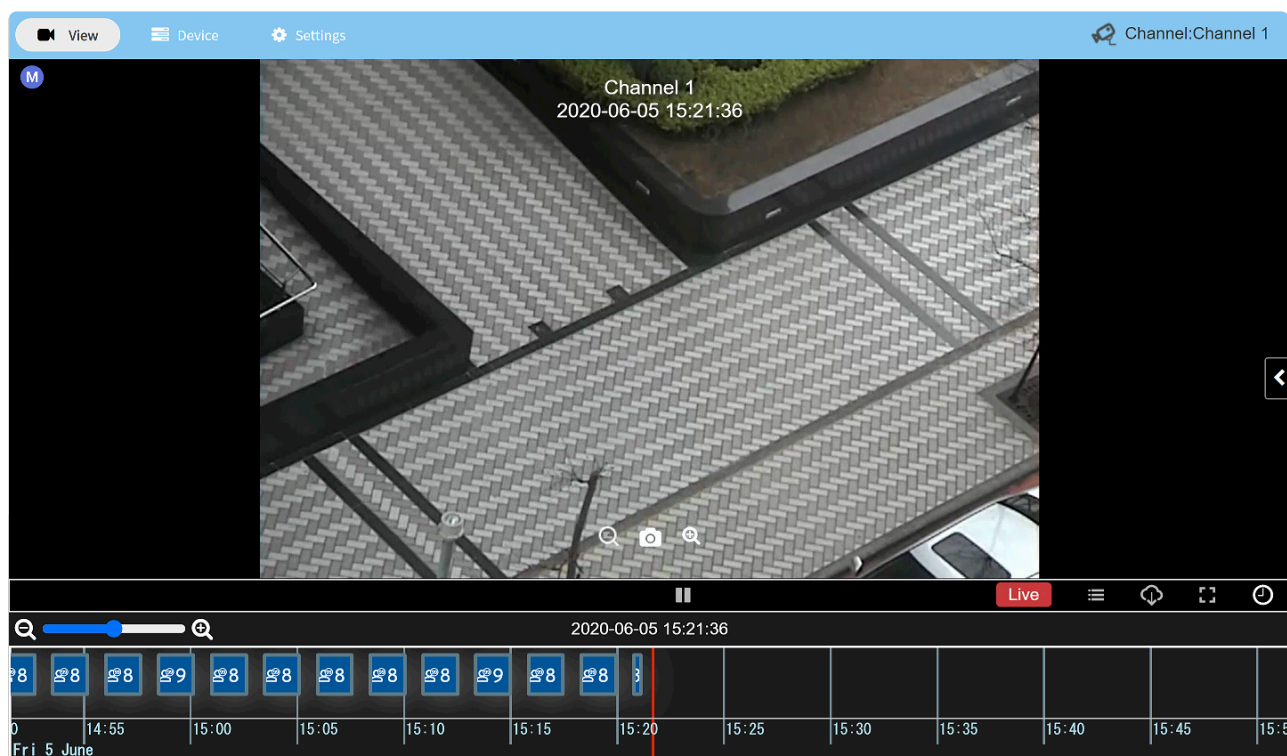
Reset

Revert

Save

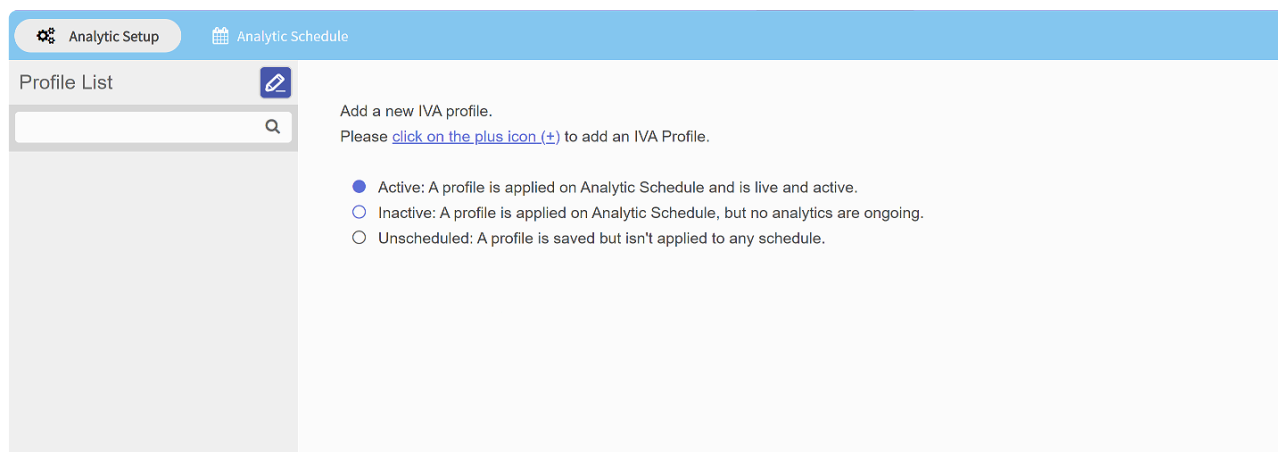
Recording Type Selectionで Erase を選択してからMain-stream(黄色)が録画されているのを範囲を指定して解除(白色)にします。

Save ボタンを押すと設定が変更されます。



View ボタンを押してライブ映像に切り替えます。

2. 画像認識の設定

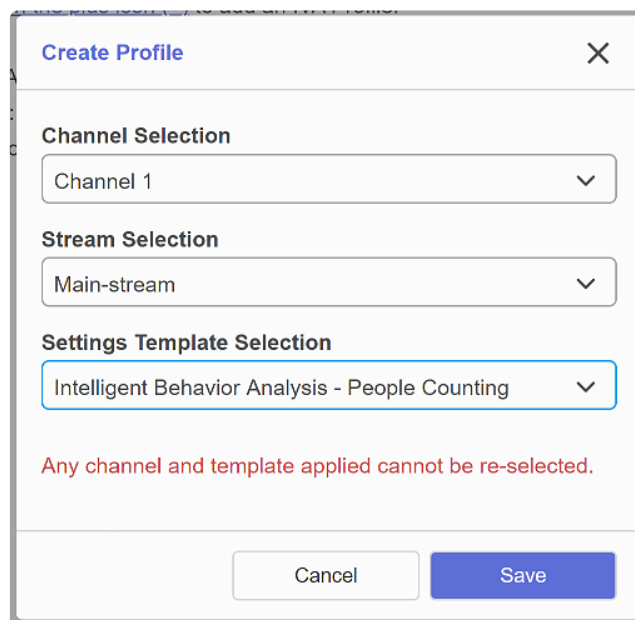


続いて画像認識の設定を行います。今回はテストとしてカメラに写る人の数を数える People Count の設定を行います。

他にもユーザを事前に登録しておく顔認識やユーザの行動遷移を認識する機能もあります。

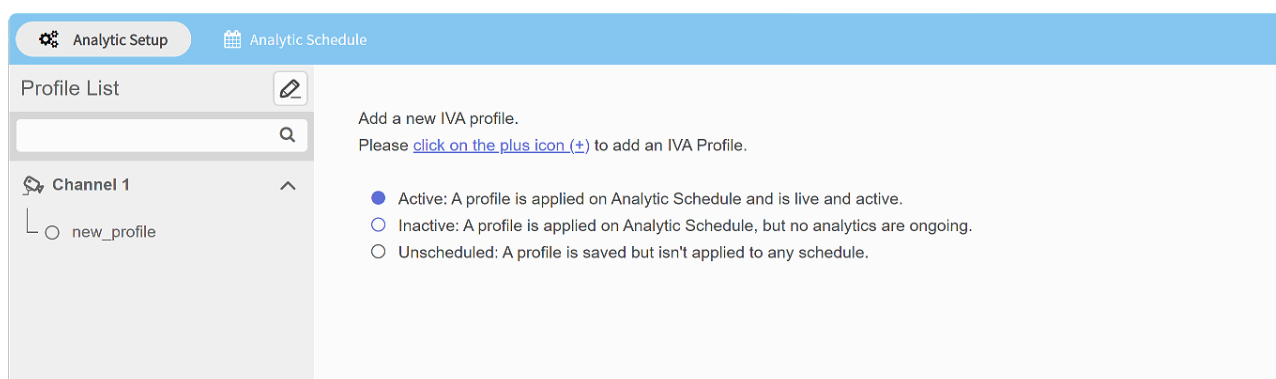
IVA Setup メニューから Analytic Setup を開きます。

左下の+ボタンか click on the plus icon (+) リンクをクリックしてください。



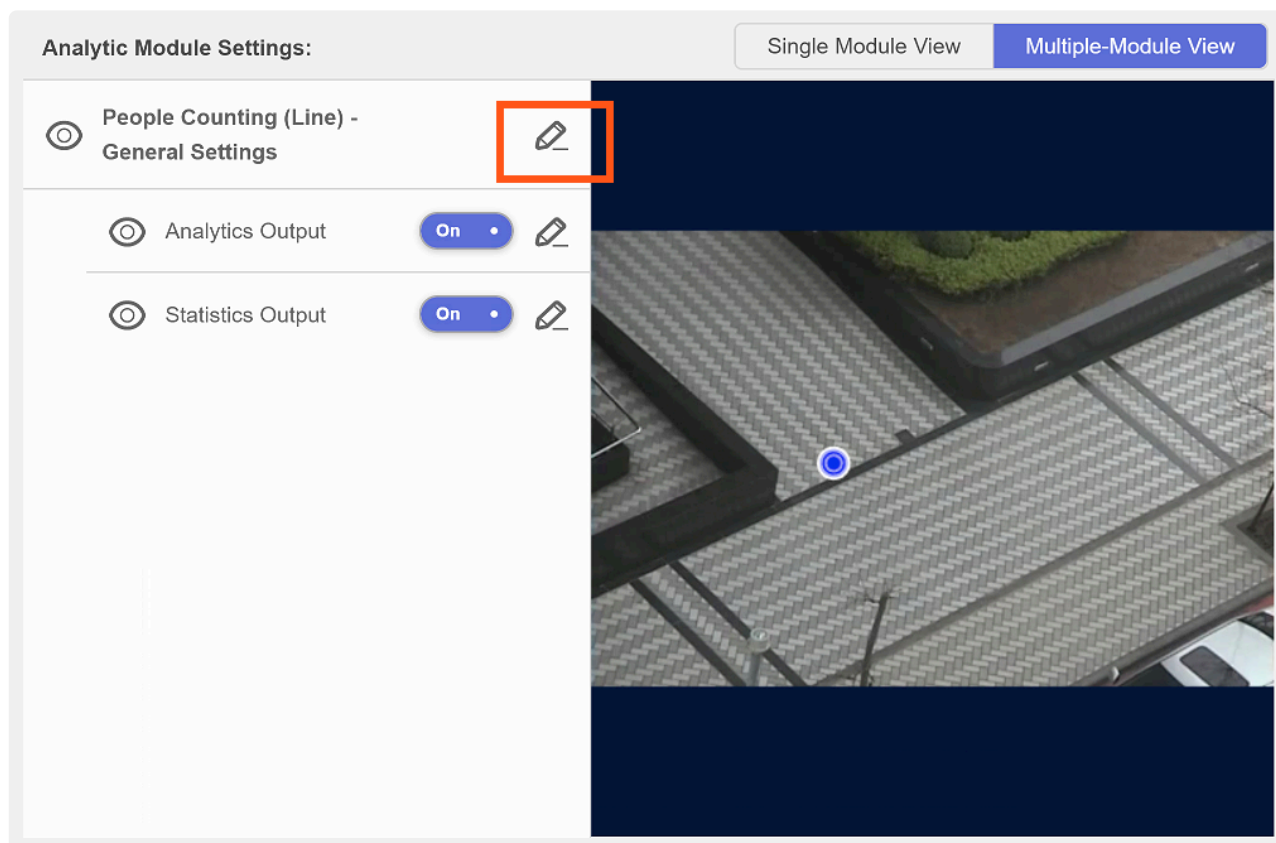
The 'Create Profile' dialog box contains three dropdown menus: 'Channel Selection' with 'Channel 1' selected, 'Stream Selection' with 'Main-stream' selected, and 'Settings Template Selection' with 'Intelligent Behavior Analysis - People Counting' selected. A red error message states: 'Any channel and template applied cannot be re-selected.' At the bottom are 'Cancel' and 'Save' buttons.

Create Profile で Channel Selection は Channel 1 を、Settings Template Selection は Intelligent Behavior Analysis – People Count を選択して Save ボタンを押して保存します。



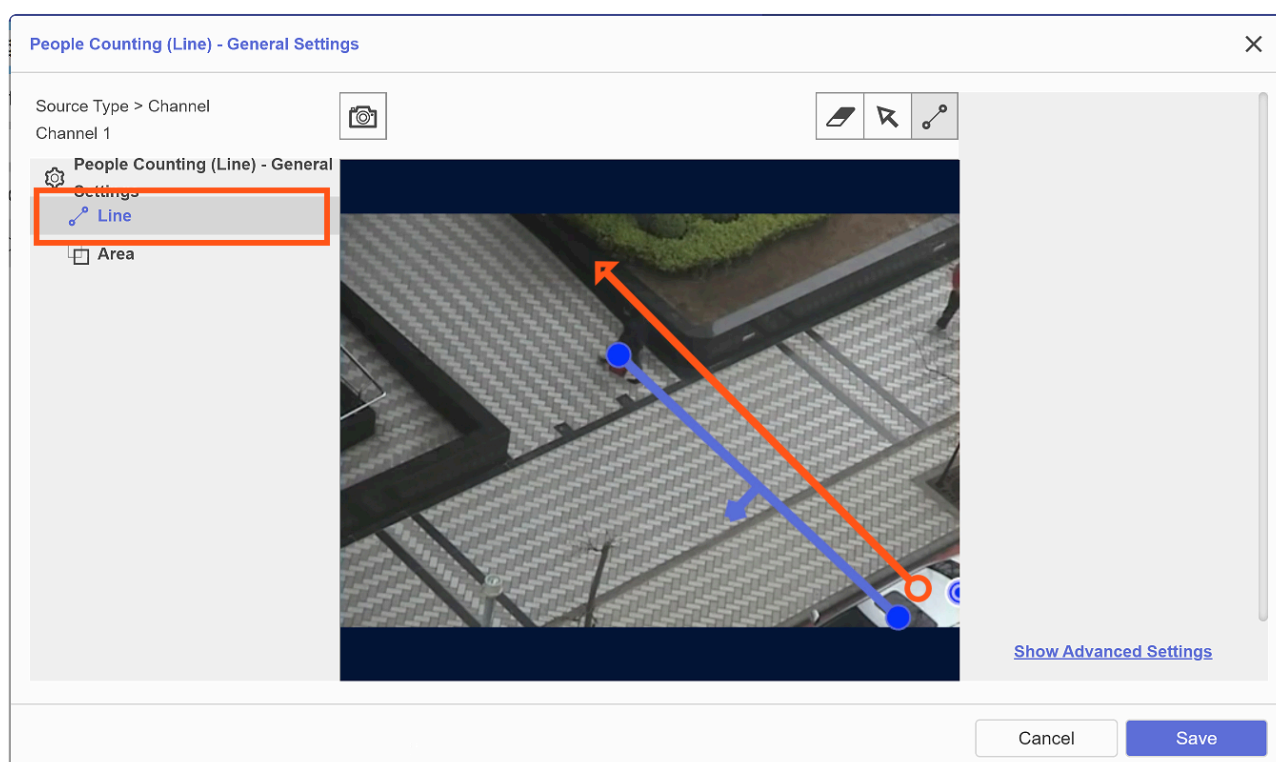
The 'Analytic Setup' interface shows a 'Profile List' on the left with a search bar and a list containing 'Channel 1' and 'new_profile'. The main area has a header 'Add a new IVA profile. Please [click on the plus icon \(+\)](#) to add an IVA Profile.' and a legend for profile states: Active (selected), Inactive, and Unscheduled.

保存すると左の Channel 1 に new_profile が表示されますので選択します。

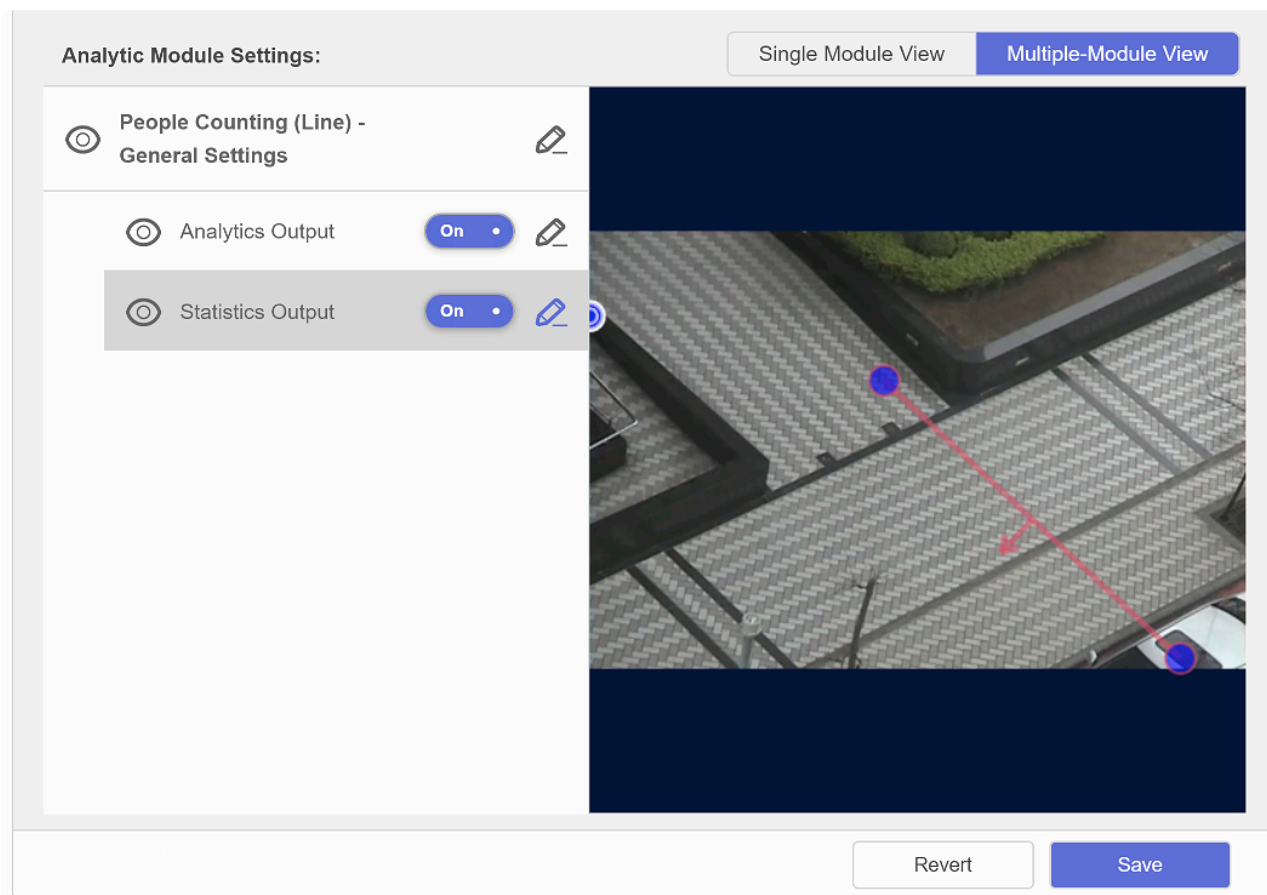


People Count が有効となる条件を設定します。テストでは認識するラインを設定して、そのラインを超えて移動する人の数を数えます。

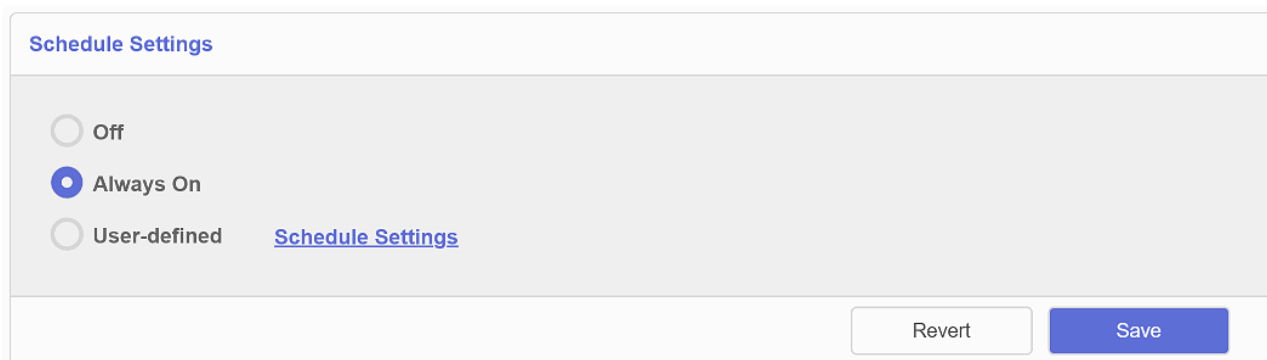
鉛筆のアイコンをクリックします。



LINE を選択して、映像の認識する箇所にマウスで線を引きます。確定はダブルクリックです。
線を引く向きによって線を左から右に越えたときか、右から左に超えたときかの指定が出来ます。
テストでは線を右から左に超えたときの設定です。
Save ボタンを押して保存します。



認識する設定条件を確認して Save ボタンを押して保存します。



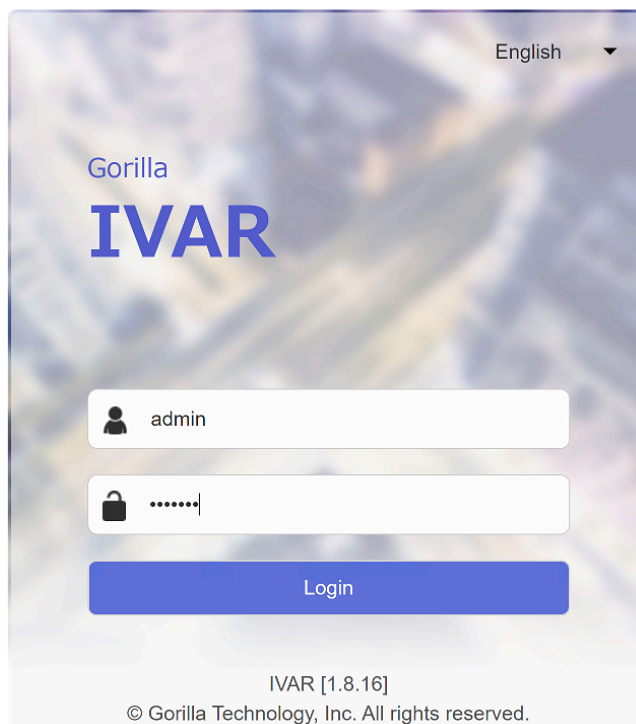
設定を有効にするために Schedule Settings は Always On を選択して Save ボタンを押します。

3. Liveでの画像認識の確認

Channels メニューから Channel 1 を開いて View を選択するとライブ映像が表示されます。
右側にカメラの映像を認識してPurple Count のイベントが発生すると表示されます。

8.4.2. GravioとIVAR連携の設定

GravioとIVAR連携の設定

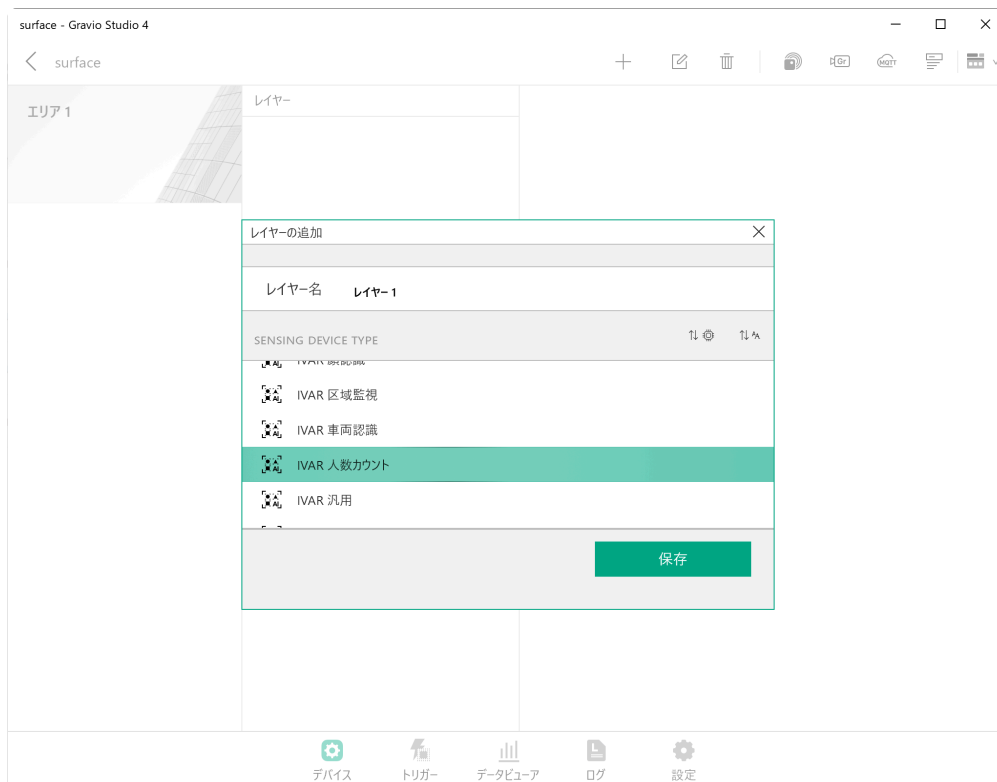


IVARにadminアカウントを使ってログインします。

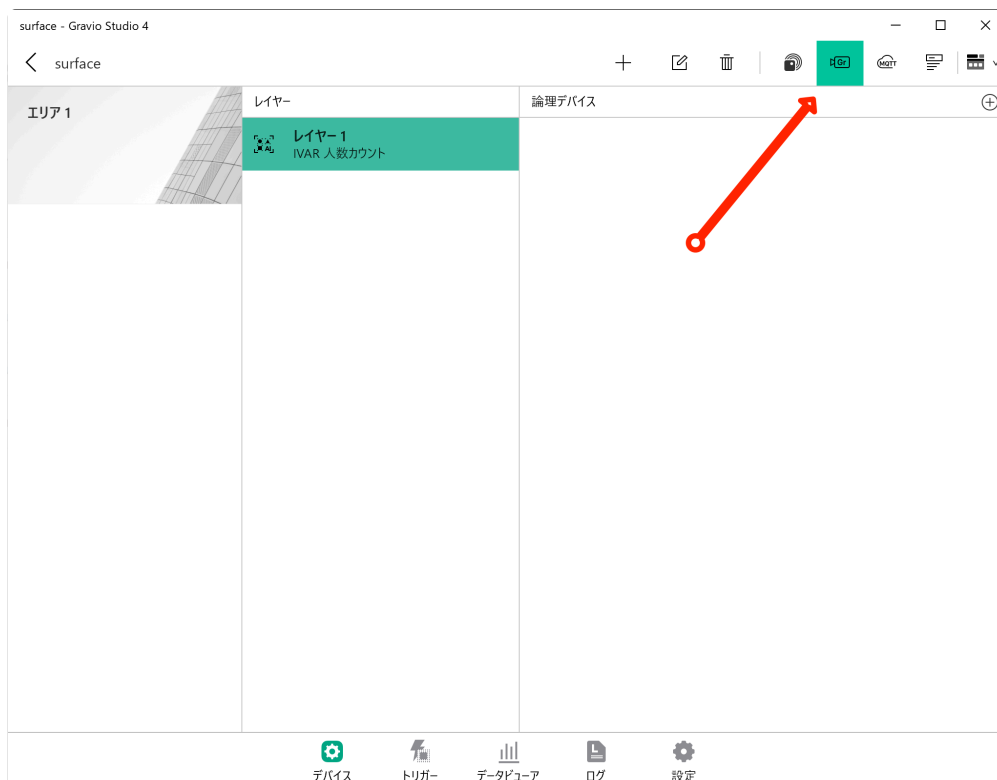
4. Gravio Studio での IVARの設定

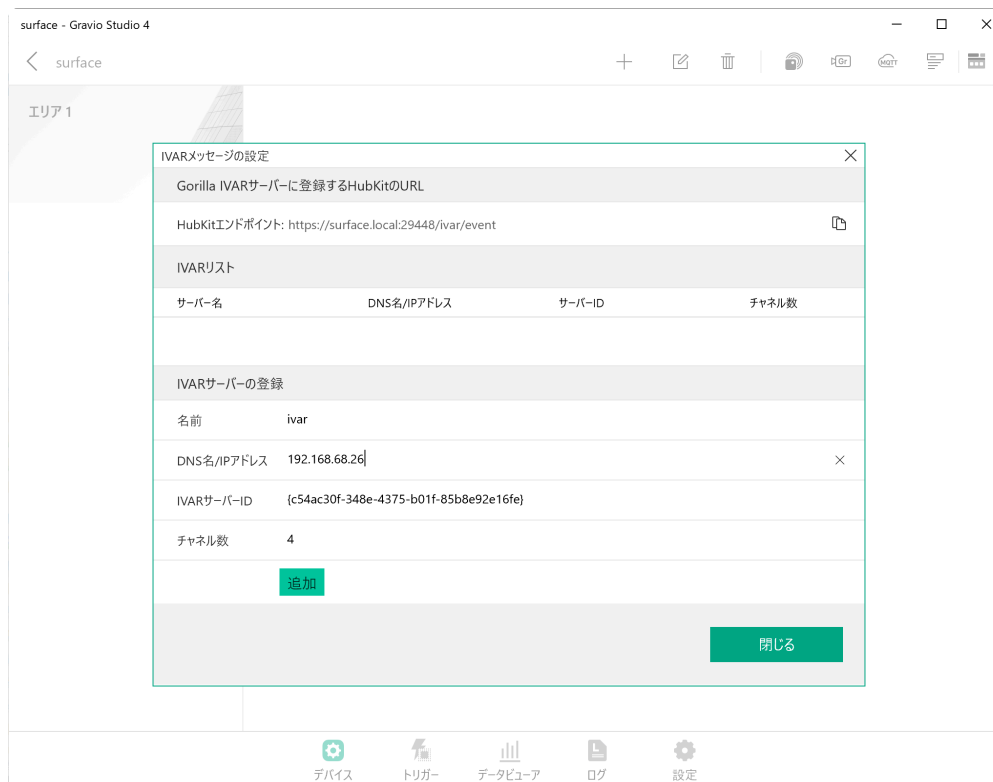
i About Basic Accounts Maintenance Service Connections	
IVAR InformationGeneral Settings	
ID	{c54ac30f-348e-4375-b01f-85b8e92e16fe}
Version	ivar-1.8.16.42905-std-win-x86_64
Total Channels	8
Maximum Analytic Channels	2
Analytic Core Channels	All : 1 Business Intelligence : 0 Human Detection : 1 Behavior Analysis : 0 License Plate : 0
Analytic Channel Expiry Date	2099-01-01 21:00:00
Analytic Core Version	BI-CORE v2.3.3 FDR-CORE v6.7.0 IPM-CORE v3.3.0 ODC-CORE v1.5.0 LPR-CORE v5.8.1 MMS/MMSC v1.3.1
IVAR NameGorilla IVAR Physical Address Latitude Longitude Description RevertSave	
License Product Key DetailsLicense Check Verified Expiry Date DeactivateResync	
System Information ⓘ	

System メニューを選択して About メニューを開きます。
IVAR Information の ID を {} を含めてコピーします。



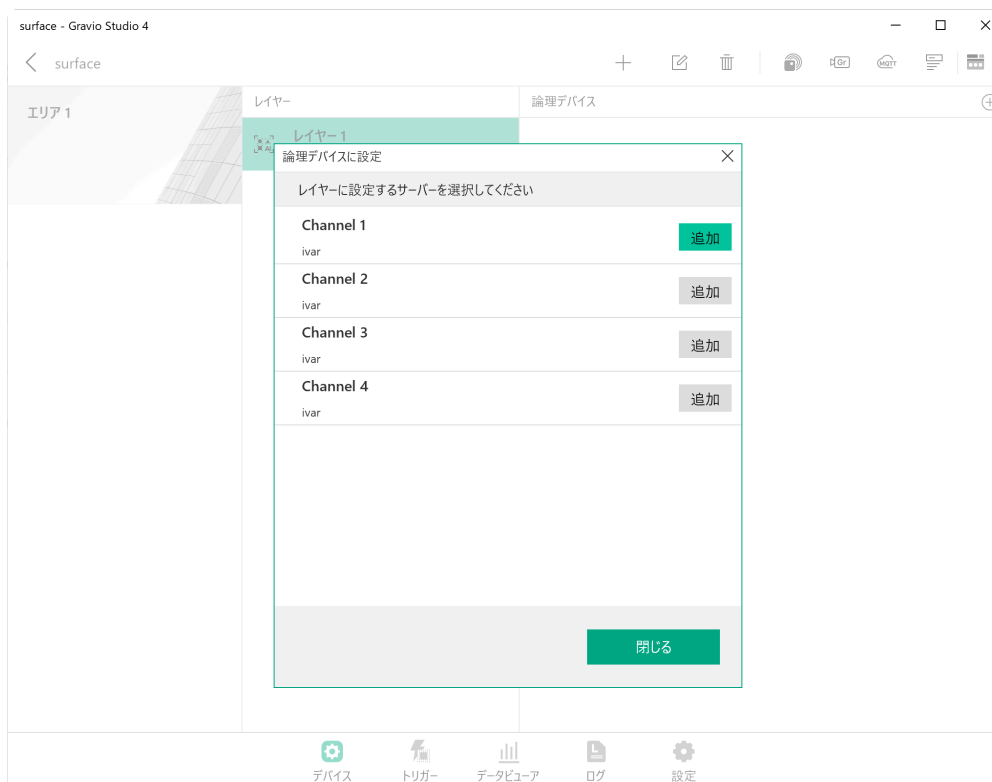
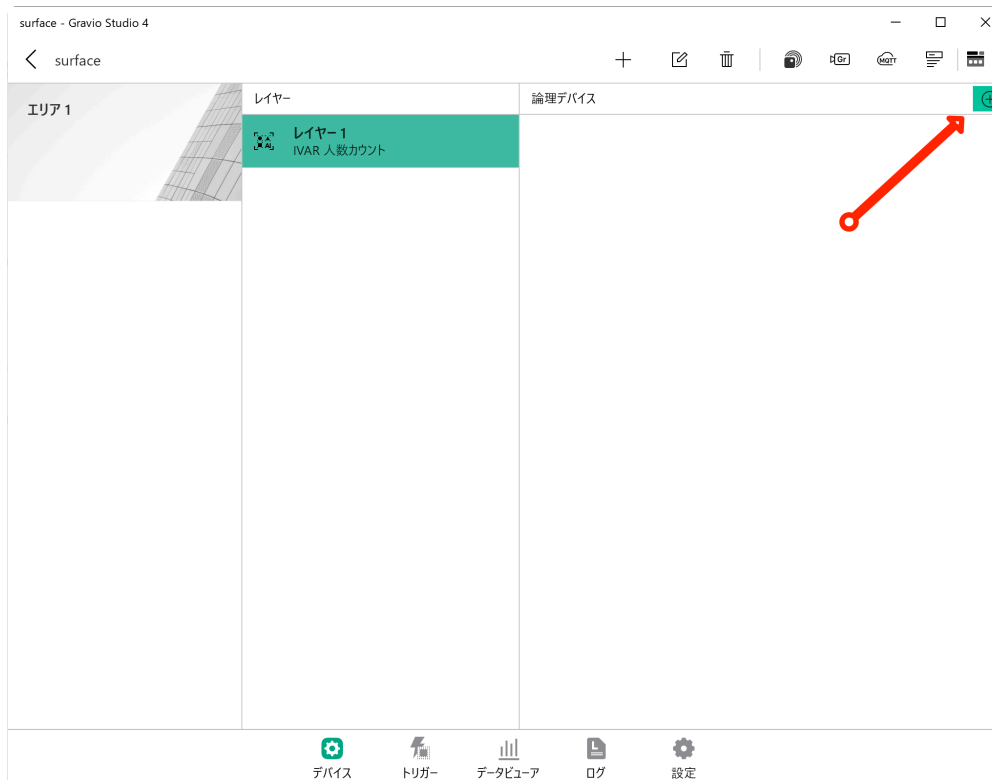
Gravio Studio のデバイスタブを選択してエリアとレイヤーを追加します。
レイヤーは IVAR 人数カウントを選択して保存ボタンを押します。

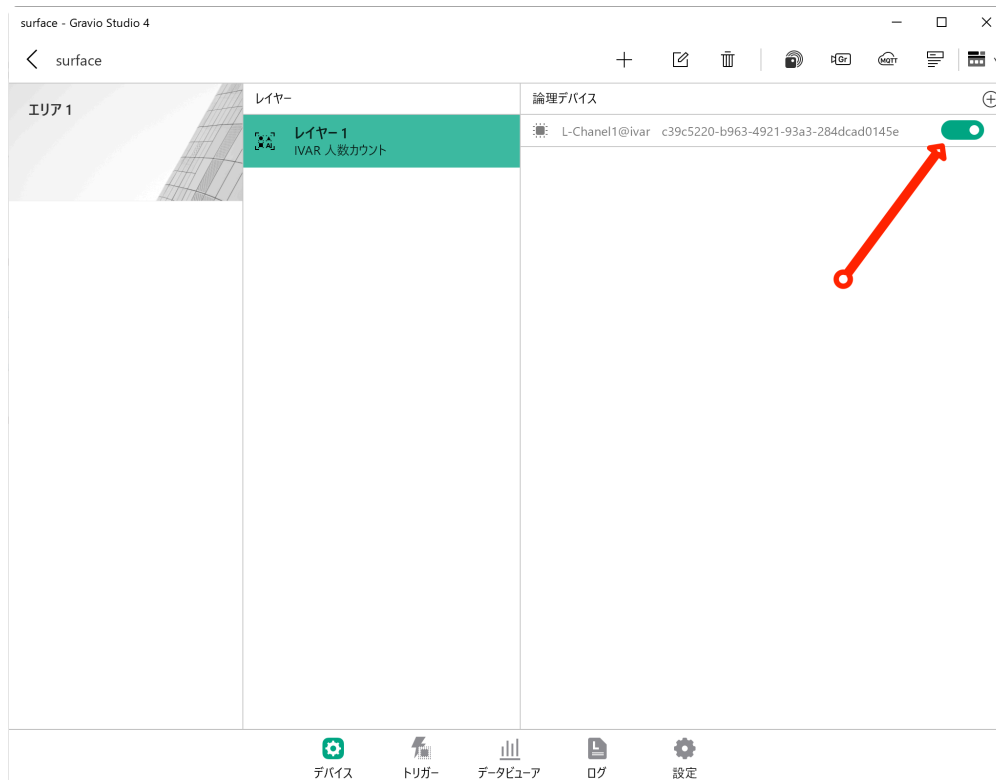




Gravio Studio から HubKit を選択してデバイスパネルから IVAR メッセージの設定画面を開きます。DNS名前/IPアドレスを追加には IVAR/BAP がインストールされている PC のアドレスを入力します。IVARサーバーIDを追加には、先ほどコピーした ID をペーストします。追加ボタンを押すと登録されます。続いて HubKit エンドポイント を右側のコピーボタンをクリックしてコピーします。

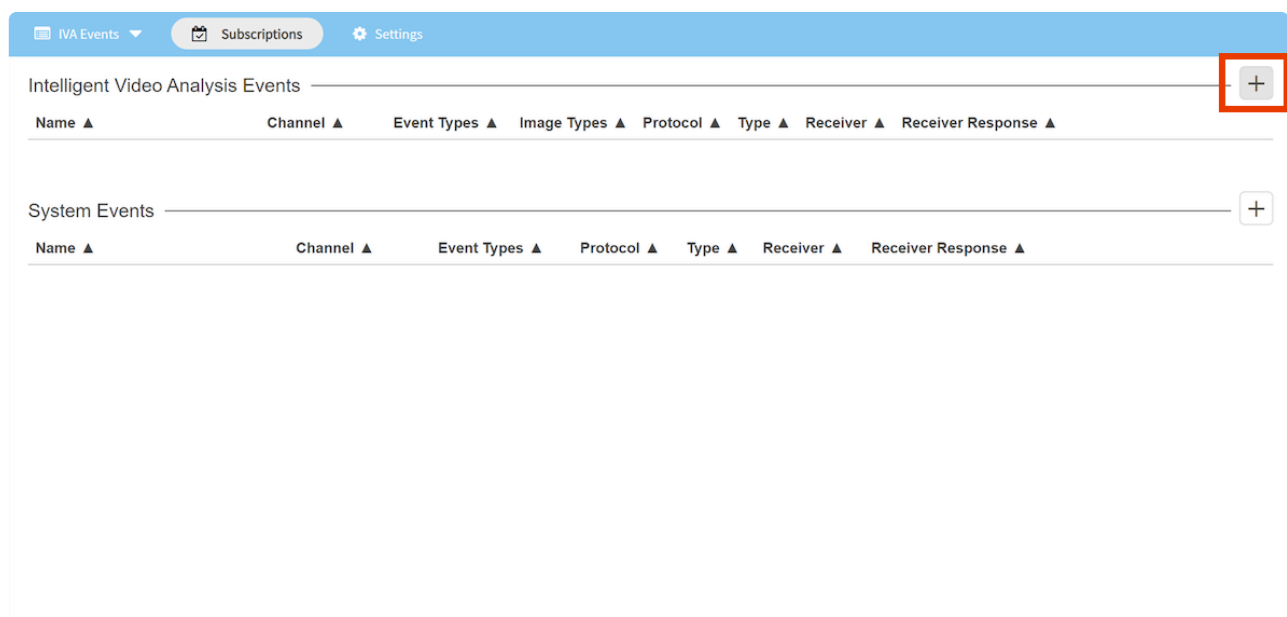
5. デバイスの登録





レイヤーにチャンネルをバインドして、保存された IVAR のレイヤーを有効にします。
GravioでのIVARの設定につきましては[こちら](#) もあわせて参照してください。

6. Gorilla IVARでのGravioのSubscriptionの登録



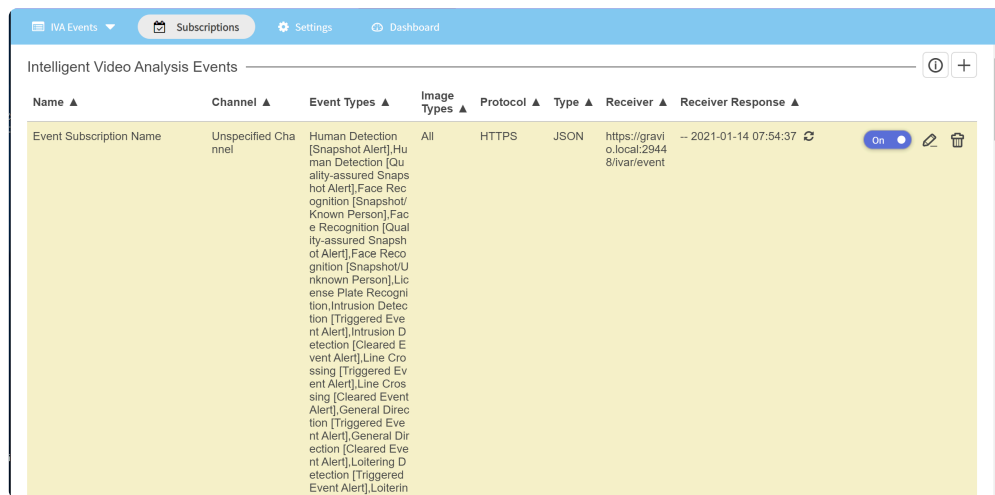
IVARの Events を選択して Subscriptions を開きます
+ボタンを押します。

The screenshot shows the 'Intelligent Video Analysis' configuration window. The 'Event Types' section is selected, and the 'Event Type Selection' dialog is open, showing a list of event types with checkboxes. The 'All Events' checkbox is checked. The 'Save' button is visible at the bottom right.

Enable にして Event Types Selection を開き All Events を選択します。

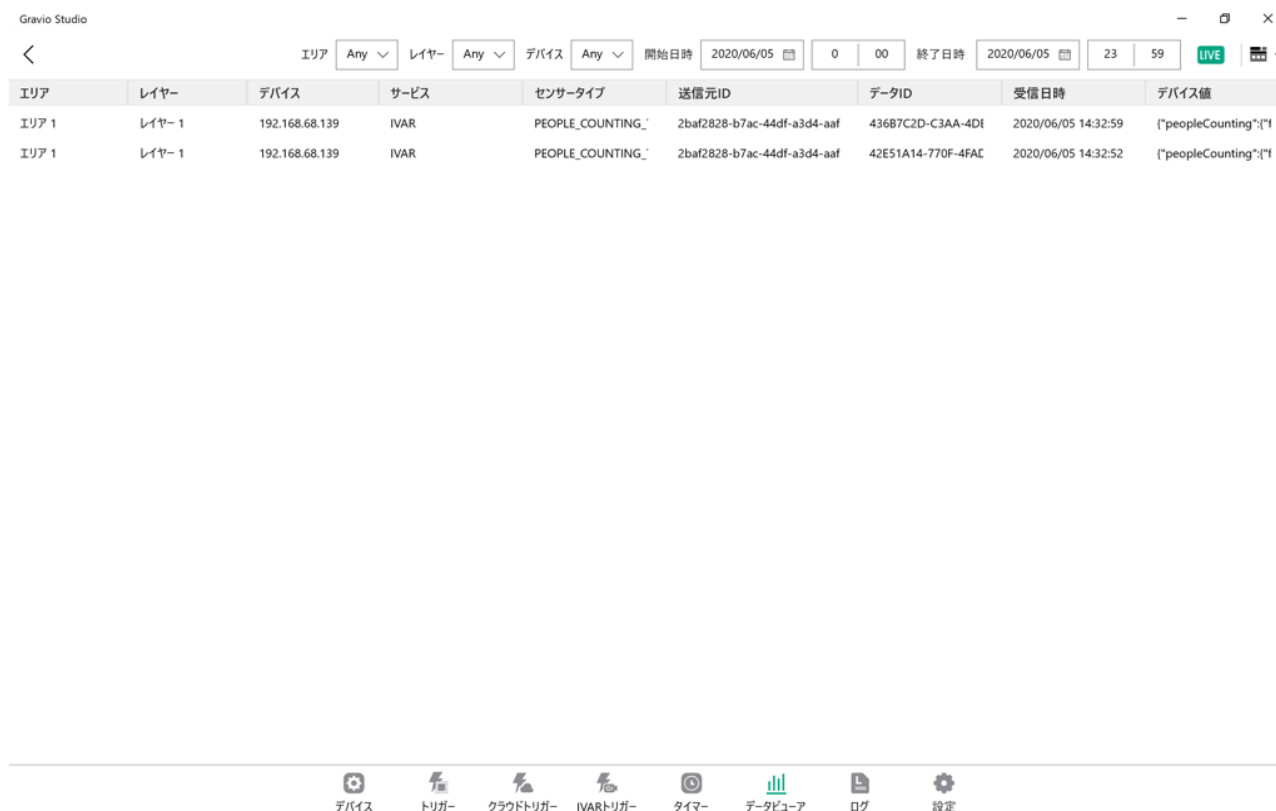
The screenshot shows the 'Intelligent Video Analysis' configuration window. The 'Event Types' section is selected, and the 'Event Type Selection' dialog is open, showing a list of event types with checkboxes. The 'All Events' checkbox is checked. The 'Save' button is visible at the bottom right.

Protocol を HTTPS に変更して、URL に先ほどコピーした HubKit Endpoint をペースとします。
Save ボタンを押して保存します。



保存されるとイベント一覧に表示されます。
削除する場合にはゴミ箱のアイコンをクリックします。

7. Gravio Studioでの データビューアでのイベント受信の確認



データビューアを開き IVAR からイベントが送信されていることを確認します。

8.4.3. Gravio IVARトリガーの設定

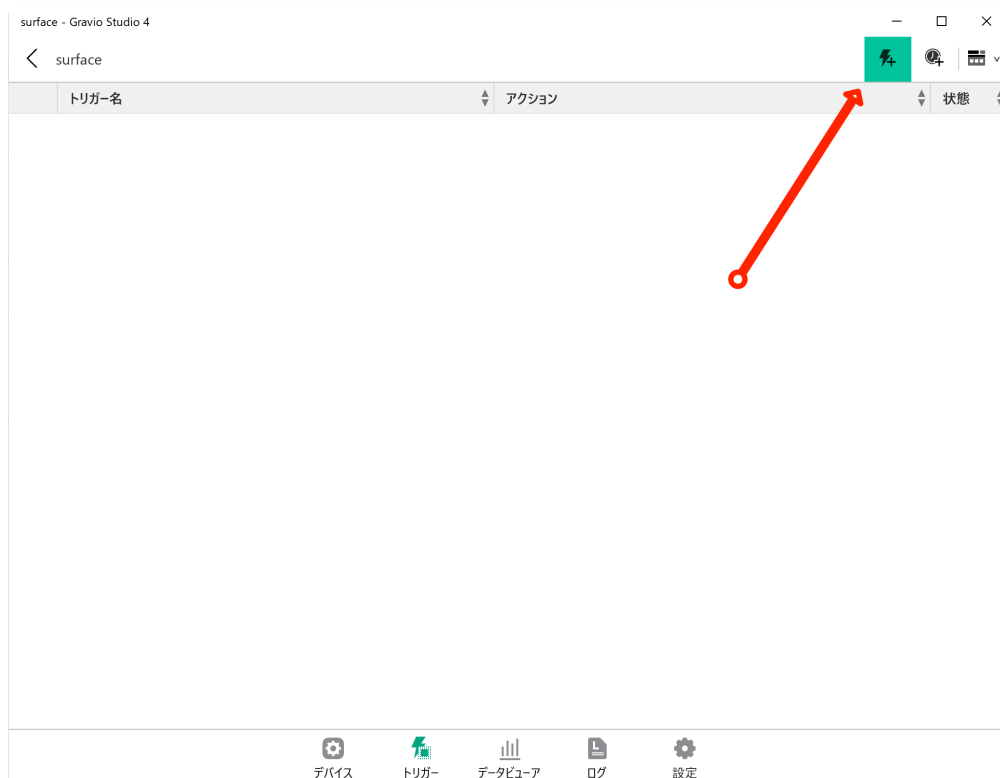
Gravio IVARトリガーの設定

7. 呼び出し用アクションの作成

IVARトリガーで呼び出すアクションを作成します。

テストでは Slack コンポーネントを配置したAlarm アクションを作成します。

8. IVARトリガーの設定



surface - Gravio Studio 4

< surface

トリガー名 アクション 状態

デバイストリガーの追加

トリガー名 新規イベントトリガー 1

エリア エリア 1

時間

条件

アクション

カレンダー なし

☐ 有効時間を指定する

開始 9 00 終了 17 00

間隔(ミリ秒) 5000

☐ 閾値によるトリガーを利用

☐ 実行中ならスキップする

キャンセル 保存

デバイス トリガー データビューア ログ 設定

surface - Gravio Studio 4

< surface

トリガー名 アクション 状態

デバイストリガーの追加

トリガー名 新規イベントトリガー 1

エリア エリア 1

時間

条件

アクション

次の条件を満たす場合に実行されます

条件数: 0/5

And

追加

キレレイヤー: レイヤー 1

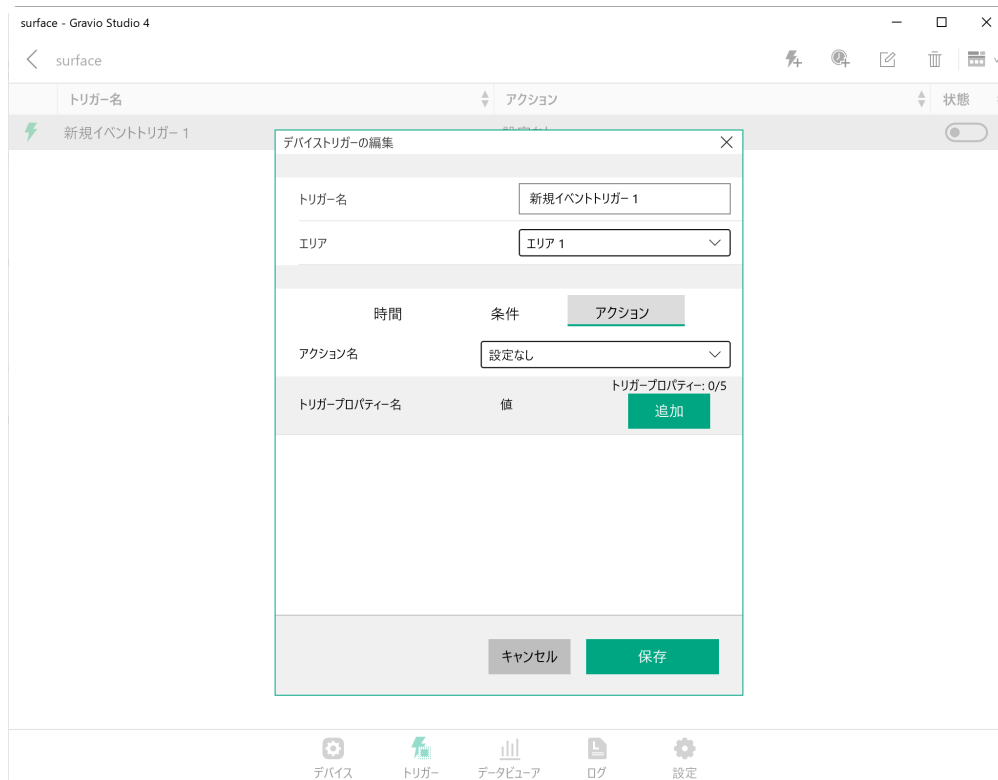
物理デバイスID:

イベントタイプ 任意のイベント

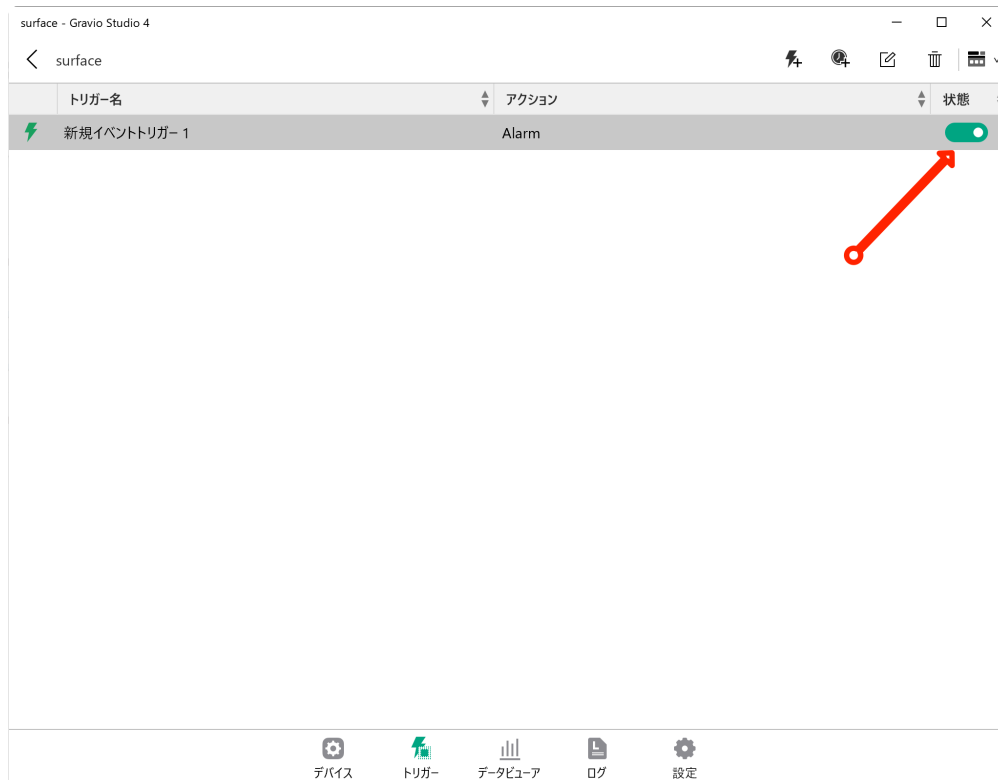
式

キャンセル 保存

デバイス トリガー データビューア ログ 設定



トリガータブを選択して+ボタンを押して新しいIVARTリガーを追加を開きます。
エリア、レイヤーを設定値に合わせて選択して、イベントタイプは 任意のイベント を選択します。
アクションは作成している Alarmアクションを選択します。
保存ボタンを押して保存します。
Gravioでのトリガーの設定につきましては[こちら](#) もあわせて参照してください。



保存した IVARトリガーを有効にします。

9. IVARトリガーでのアクションの実行の確認

有効にして IVARトリガー条件にマッチするイベントが IVARから送信されたタイミングで指定のアクションが実行されます。

テストでは AlarmアクションにはSlack コンポーネントを配置していますので、Slack に通知が届きます。

8.5. Gravioへの接続テスト – 顔認識

Gorilla と Gravioの連携

この章では、Gorillaのカメラ映像を画像認識処理で人数をカウントし、それをトリガとしてGravioに処理データを送信します。

Gravioでは IVARトリガを指定して任意のアクションを起動します。

準備するもの

Gorilla IVAR/BAPをインストールしているPCと、onvifカメラ1台、Gravio GS/HubKitをインストールしているPCとChromeブラウザを準備します。

GravioはEnterpriseのライセンスが必要です。また **HubKit**は**Coordinator**に事前に登録が必要です。

ここではonvifカメラはIO Data社のTS-NA220Wを使用します

設定の流れ

BAPの設定

1. グループを作成
2. ユーザを作成

IVARのカメラ+顔認識の設定

3. IVARでonvifカメラを登録
4. 顔認識の設定
5. Liveでの顔認識の確認

GravioとIVAR連携の設定

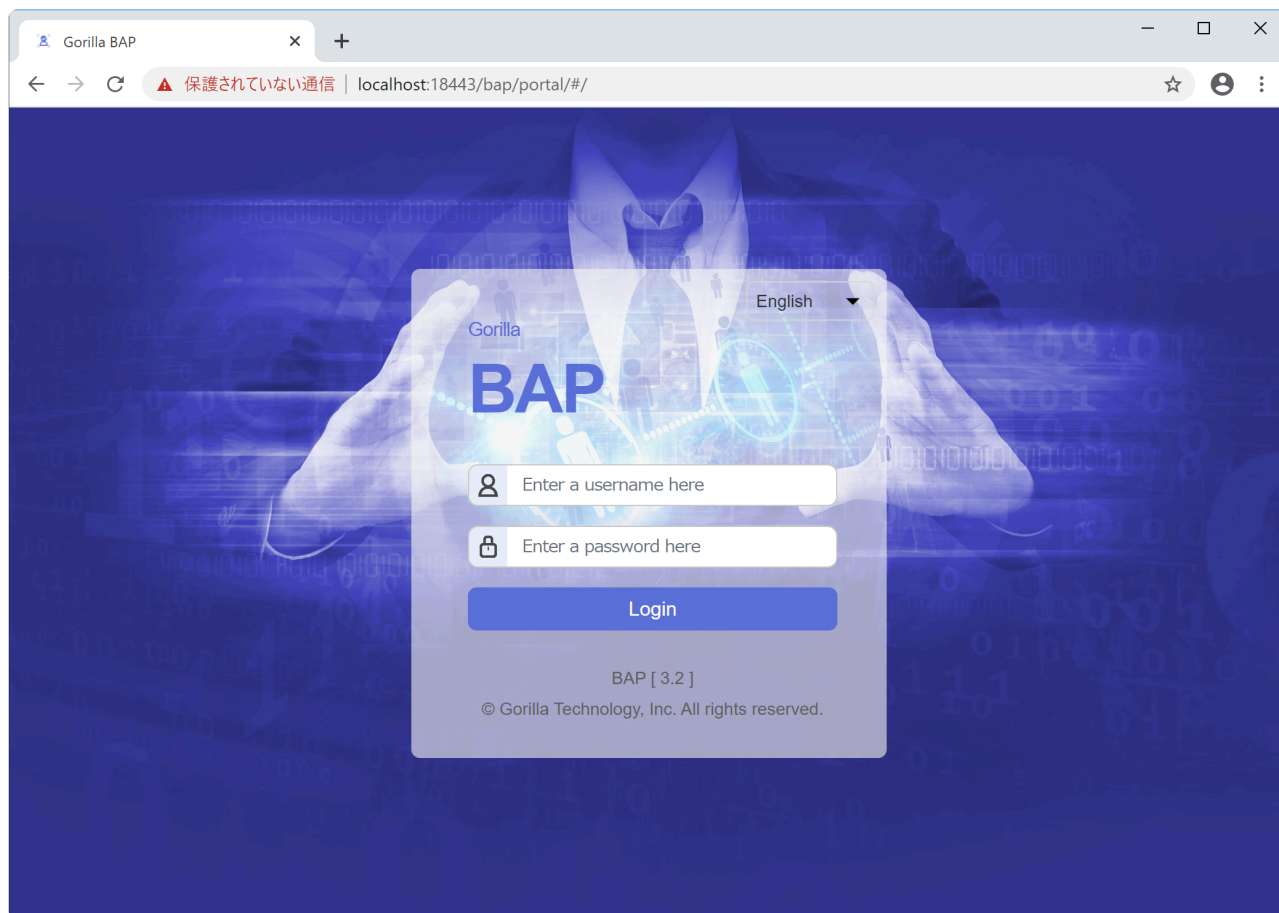
6. Gravio Studio での IVARの設定
7. イベントの登録
8. Gravio Studioでの データビューアでのイベント受信の確認

Gravio IVARトリガーの設定

9. 呼び出し用アクションの作成
10. IVARトリガーの設定
11. IVARトリガーでのアクションの実行の確認

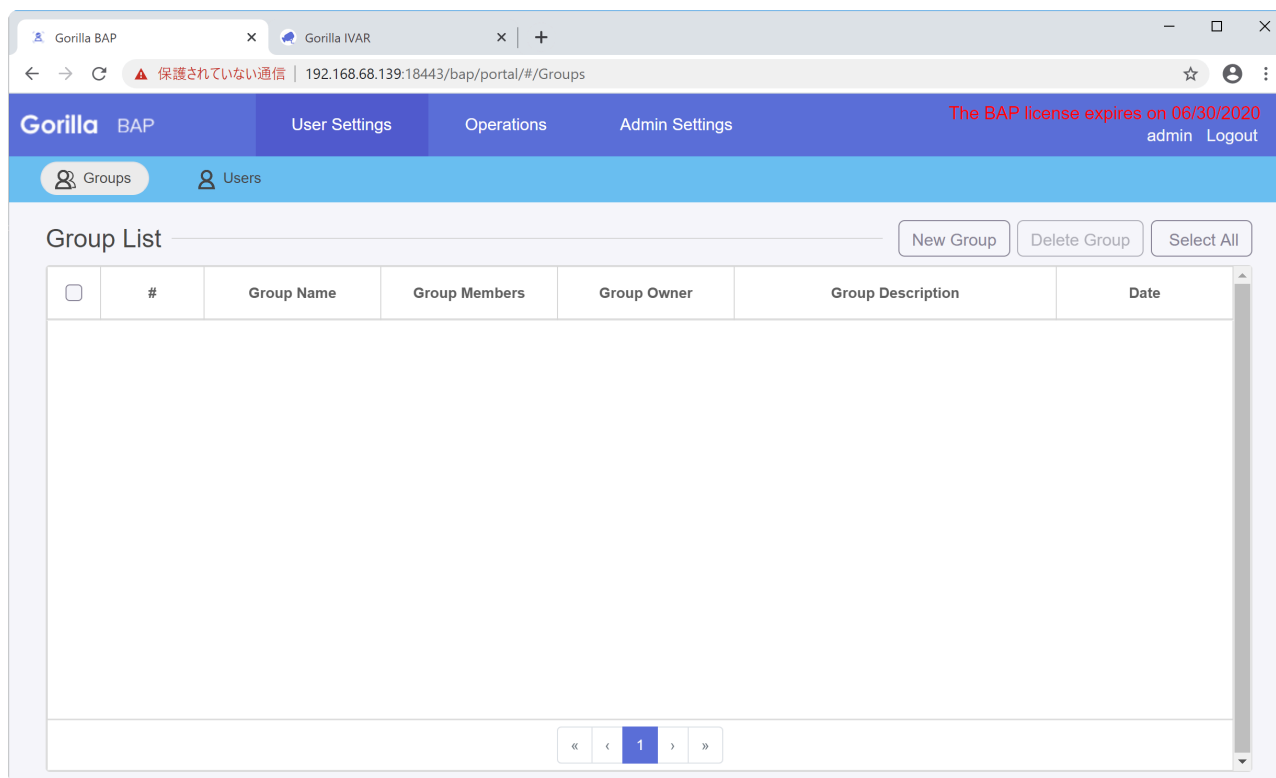
8.5.1. BAPの設定

BAPの設定

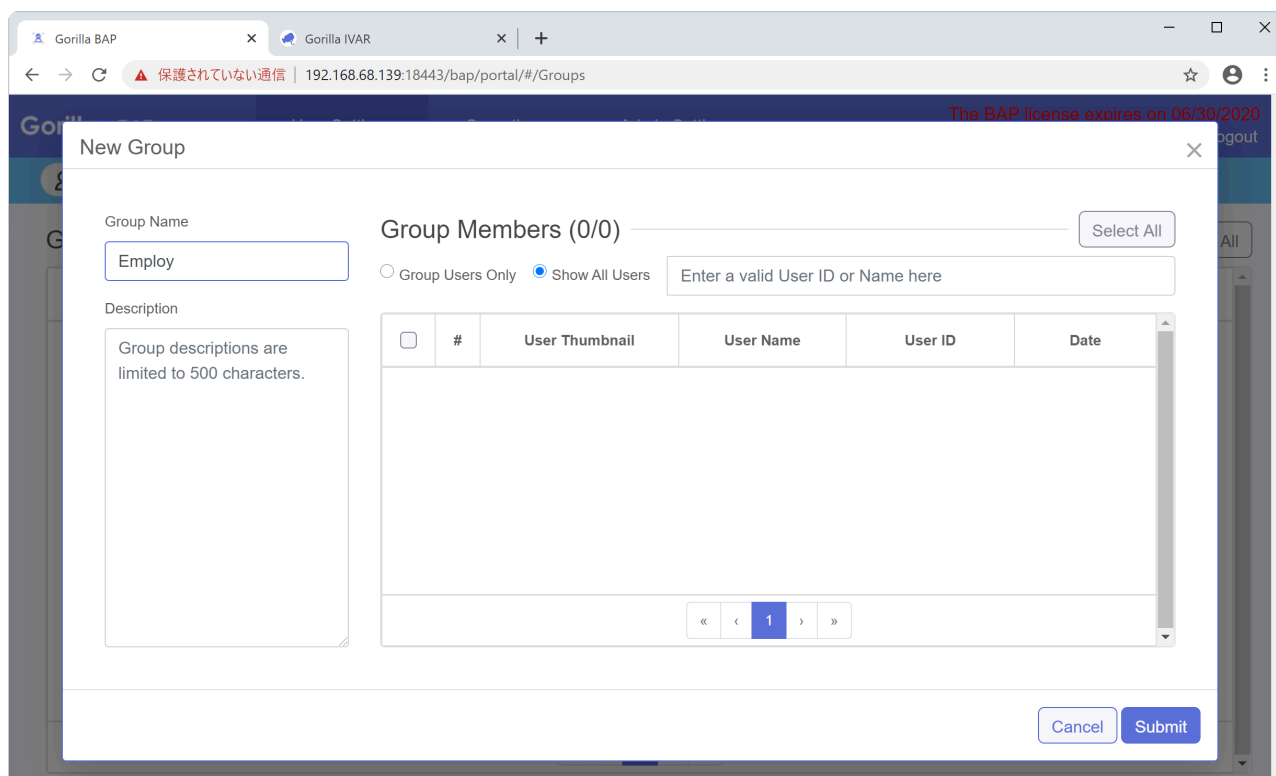


BAPにadminアカウントを使ってログインします。

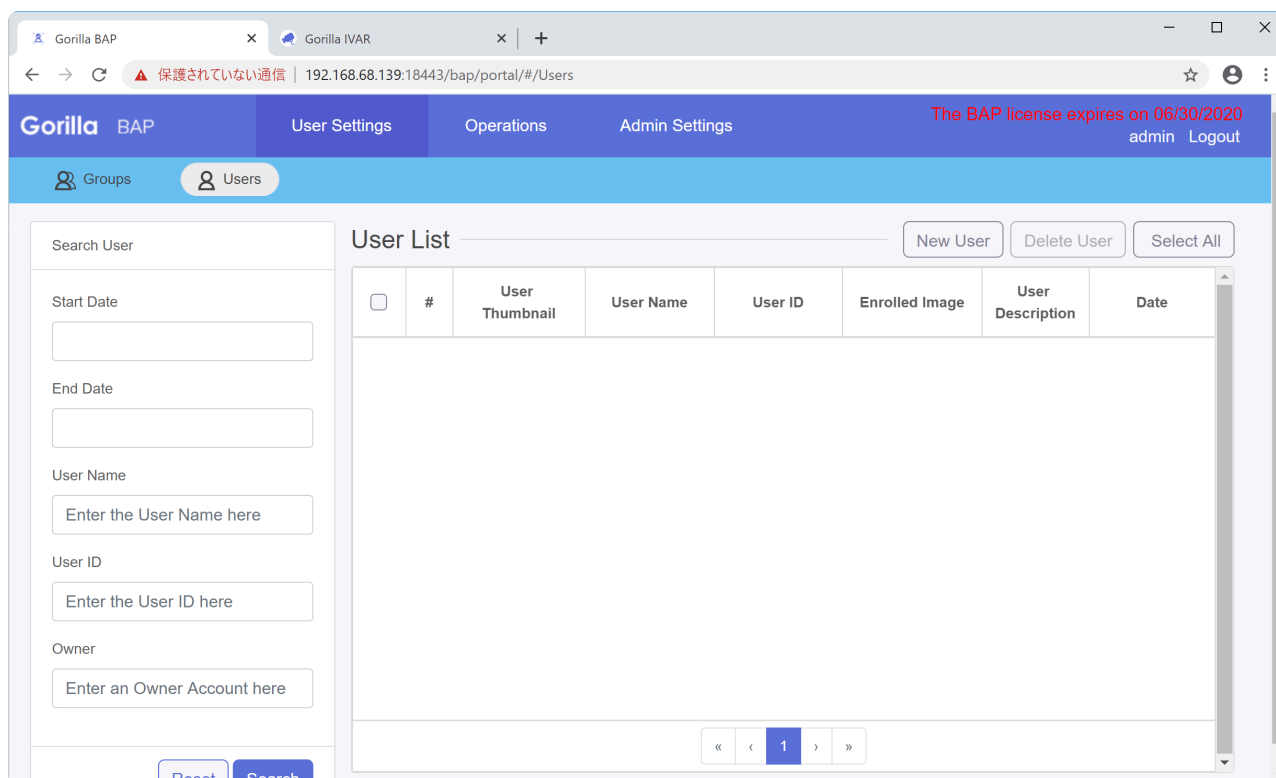
1. BAPにグループを登録します



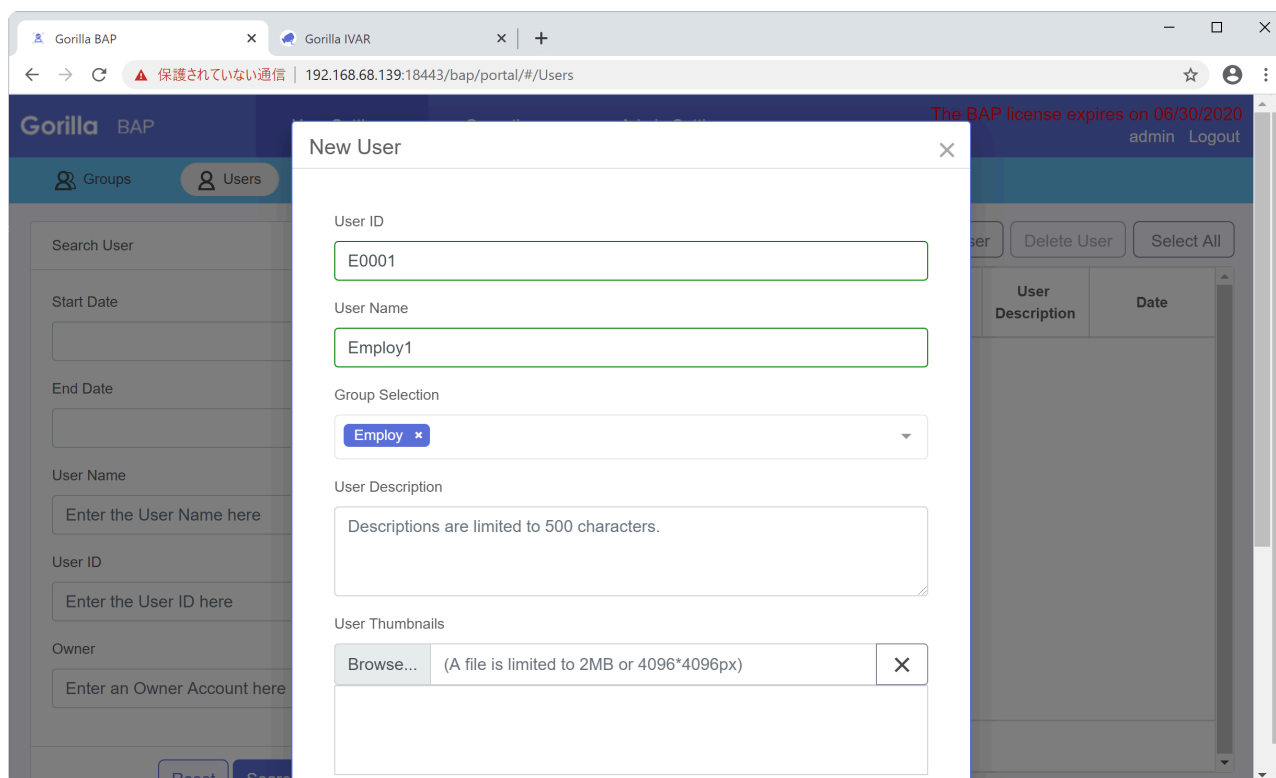
User Settings メニューから Groups を選択します。
New Group ボタンを押してグループを登録します。



Group Name にグループ名を入力します。
テストでは Employ グループを作成します。



続いて User Settings メニューから Users を選択します。
New User ボタンを押してグループを登録します。



User IDとUser Name を入力します。

テストでは E0001 と Employ1 としました。

Group Selection は先ほど作成した Employ グループを選択します。

The screenshot shows the Gorilla BAP web interface. A modal window is open for editing a user. The modal contains the following sections:

- User Description:** A text area with a note: "Descriptions are limited to 500 characters."
- User Thumbnails:** A section with a "Browse..." button and a note: "(A file is limited to 2MB or 4096*4096px)".
- Enroll Images:** A section with a "Browse..." button and a note: "(A file is limited to 2MB or 4096*4096px)". Below this, there is a cropped face image of a man. A red warning message is displayed: "Upload multiple related cropped face images of a subject. Avoid multiple faces in one image. Image size is up to 2MB, or resolution is up to 4096*4096 px."

At the bottom of the modal are "Cancel" and "Update" buttons. The background interface shows a sidebar with "Groups" and "Users" tabs, and a table of users with columns for "User Description" and "Date".

画面をスクロールして Enroll Images にこのユーザの顔写真をアップロードします。

Update ボタンを押して保存します。

The screenshot displays the Gorilla BAP web interface. The top navigation bar includes 'Gorilla BAP', 'User Settings', 'Operations', and 'Admin Settings'. A red notification states 'The BAP license expires on 06/30/2020' with 'admin' and 'Logout' links. The sidebar has 'Groups' and 'Users' tabs. The main content area is titled 'User List' and features a search form on the left and a table of users on the right.

Search Form:

- Search User:
- Start Date:
- End Date:
- User Name:
- User ID:
- Owner:
- Buttons:

User List Table:

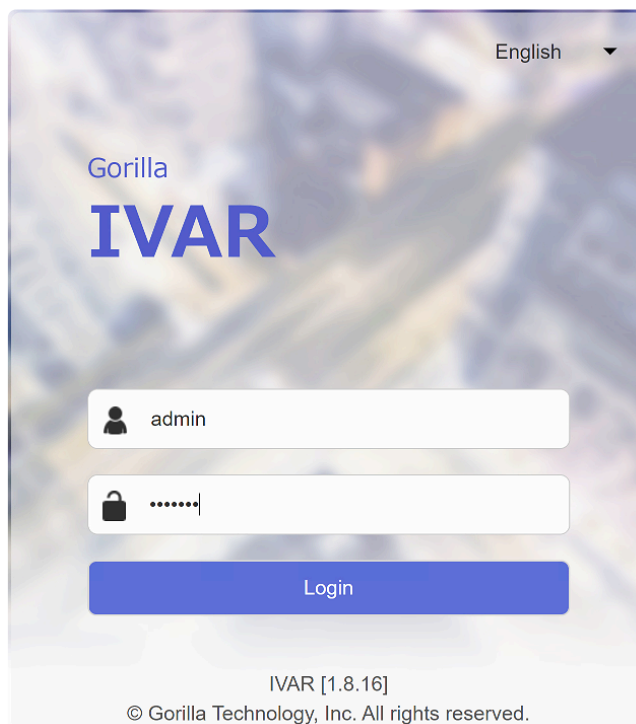
☐	#	User Thumbnail	User Name	User ID	Enrolled Image	User Description	Date
☐	1		Employ1	E0001	False		2020-6-19

Page navigation: « ‹ 1 › »

ユーザが登録されると User List に表示されます。

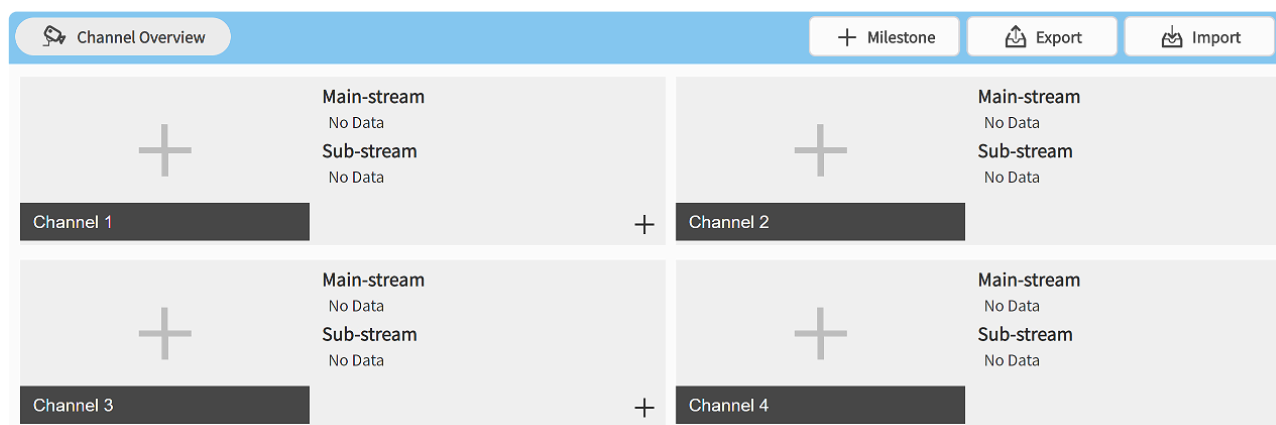
8.5.2. IVARのカメラ+顔認識の設定

IVARのカメラ+顔認識の設定



IVARにadminアカウントを使ってログインします。

1. IVARでonvifカメラを登録します



ログインするとチャンネル画面が表示されます。
+ボタンを押してチャンネル2を登録します。

The screenshot shows the 'Device' tab in the Asteria interface. The 'Device Type' section has four radio button options: 'RTSP' (Manually add an RTSP streaming camera), 'ONVIF' (Automatically search ONVIF Profile S IP cameras), 'Video File' (Upload a video clip), and 'Capture Card/USB Camera' (Select a local streaming device). The 'ONVIF' option is selected. A 'Next' button is located at the bottom right.

使用するカメラはONVIFカメラですのでONVIFを選択して Next ボタンを押します。

The screenshot shows the 'ONVIF Device List' screen. It features a table with columns 'Item', 'IP Address', and 'Device Name'. One item is listed with IP address 192.168.68.140:80 and device name IODATA%20TS-NA220W. Below the table, there is a section 'Log into the Device : Unselected' with input fields for 'IP Address', 'Username', and 'Password'. 'Back' and 'Next' buttons are at the bottom.

Item ▲	IP Address ▲	Device Name ▲
☐ 1	192.168.68.140:80	IODATA%20TS-NA220W

Log into the Device : Unselected

IP Address:

Username:

Password:

同一ネットワークに設置されているカメラが一覧に表示されます。

This screenshot is similar to the previous one, but the first device in the 'ONVIF Device List' table is now selected, indicated by a checkmark in the 'Item' column. The 'Log into the Device' section now shows 'Log into the Device : IODATA%20TS-NA220W'. The input fields are pre-filled with the device's IP address (192.168.68.140), username (admin), and a masked password (.....).

Item ▲	IP Address ▲	Device Name ▲
☒ 1	192.168.68.140:80	IODATA%20TS-NA220W

Log into the Device : IODATA%20TS-NA220W

IP Address:

Username:

Password:

今回はIO DATAのTS-NA220Wを使用しますので、チェックボックスをオンにしてUsernameとPasswordを入力し

て Next ボタンを押します。

The screenshot shows a web interface for configuring a device. At the top, there are tabs for 'View', 'Device', and 'Settings'. The 'Device' tab is active. Below the tabs, there is a 'Device Profile List' table with the following data:

Item ▲	Profile Name ▲	Codec ▲	Resolution ▲
1	profile1	H264	1920 x 1080
2	profile2	H264	1280 x 720

Below the table, there is a section titled 'Choose an ONVIF Profile'. It contains two dropdown menus for 'Main-stream' and 'Sub-stream', both currently showing 'Please select a profile'. There are also two toggle switches for 'Main-stream: RTSP over TCP' and 'Sub-stream: RTSP over TCP', both set to 'On'. An 'Enable Sub-stream' toggle switch is set to 'Off'. At the bottom right, there are 'Back' and 'Apply' buttons.

カメラで使用できるプロファイルが一覧に表示されますのでMain-streamで選択します。
顔認識の場合は、解像度が高い方が望ましいためフルHD(1920×1080)を選択します。
今回は顔認識ですので profile1 を選択して Apply ボタンを押します。

This screenshot is similar to the previous one, but the 'Main-stream' dropdown menu is now set to 'profile1'. The 'Sub-stream' dropdown menu remains at 'Please select a profile'. The 'Main-stream: RTSP over TCP' toggle switch is now set to 'On', and the 'Sub-stream: RTSP over TCP' toggle switch is also set to 'On'. The 'Enable Sub-stream' toggle switch remains set to 'Off'. The 'Back' and 'Apply' buttons are still present at the bottom right.

カメラが登録されました、削除する場合には Remove ボタンを押します。

View

Device

Settings

Channel:Channel

Description

Location

Longitude Coordinates

Latitude Coordinates

Maximum Retention (in days)

Recording Path

Revert

Save

Revert

Save

Recording Schedule

Recording Type Selection

All

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mon																								
Tue																								
Wed																								
Thu																								
Fri																								
Sat																								
Sun																								

AllMain-streamSub-streamErase

ResetRevertSave

カメラの設定が完了すると自動的に映像を保存しますので、ハードディスクを圧迫しないようにテストでは映像を保存しないように設定を変更します。

Settingボタンを押して、下にある Recording Schedule を開きます。

Recording Schedule

Recording Type Selection

Erase

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Mon																								
Tue																								
Wed																								
Thu																								
Fri																								
Sat																								
Sun																								

AllMain-streamSub-streamErase

ResetRevertSave

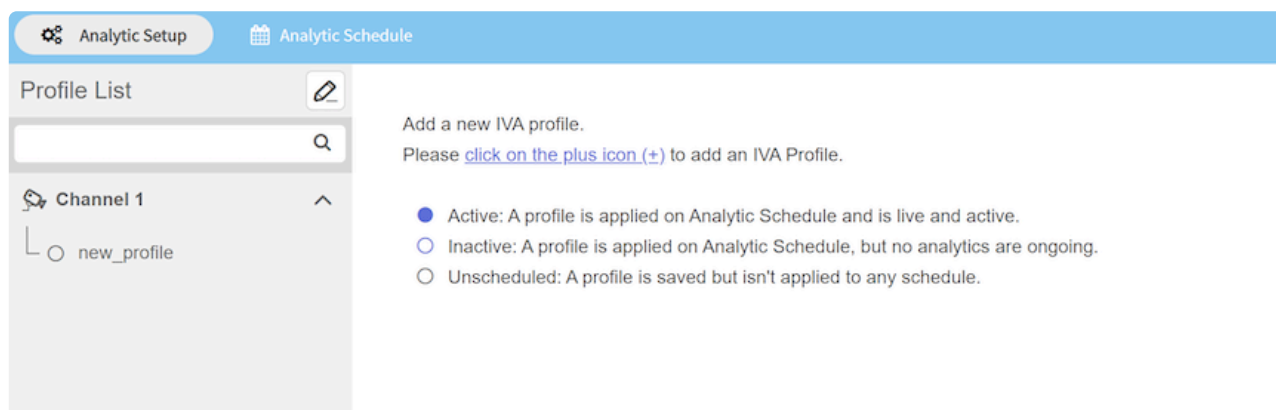
Recording Type Selectionで Erase を選択してからMain-stream（黄色）が録画されているのを範囲を指定して解除（白色）にします。

Save ボタンを押すと設定が変更されます。



View ボタンを押してライブ映像に切り替えます。

2. 顔認識の設定

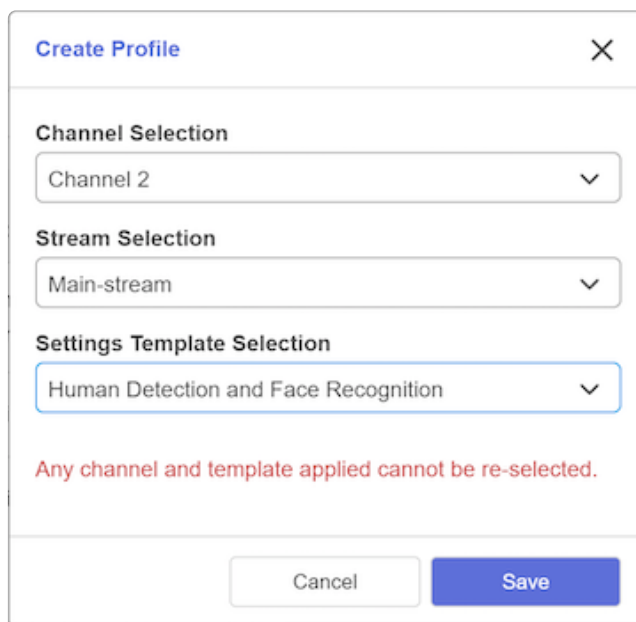


続いて顔認識の設定を行います。今回はテストとしてカメラに写る人の顔を認識する Face Recognition の設定を行います。

他にも人数を数えたりユーザの行動遷移を認識する機能もあります。

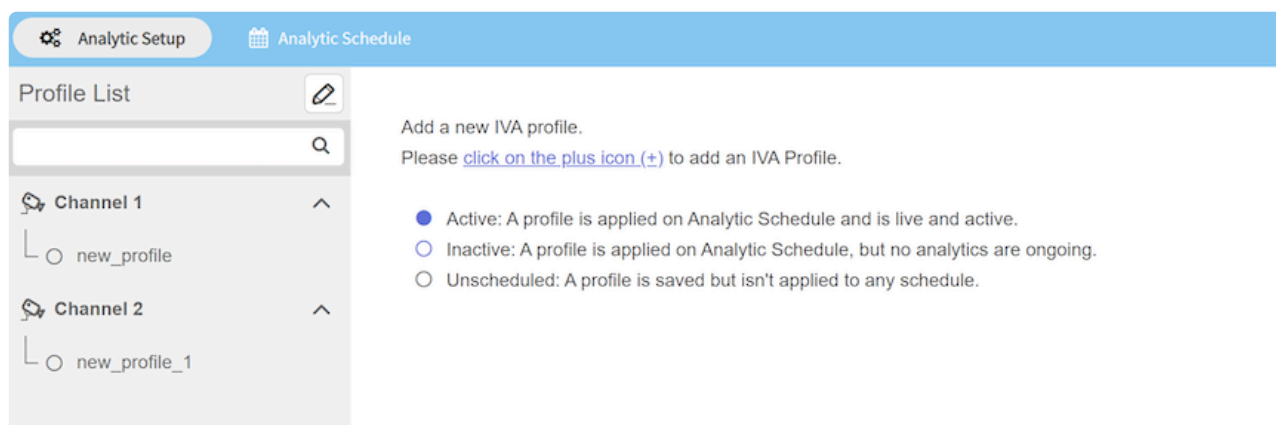
IVA Setup メニューから Analytic Setup を開きます。

左下の+ボタンか [click on the plus icon \(+\)](#) リンクをクリックしてください。



The 'Create Profile' dialog box contains three dropdown menus: 'Channel Selection' with 'Channel 2' selected, 'Stream Selection' with 'Main-stream' selected, and 'Settings Template Selection' with 'Human Detection and Face Recognition' selected. A red error message states 'Any channel and template applied cannot be re-selected.' At the bottom are 'Cancel' and 'Save' buttons.

Create Profile で Channel Selection は Channel 2 を、Settings Template Selection は Human Detection and Face Recognition を選択して Save ボタンを押して保存します。



The 'Analytic Setup' interface shows a 'Profile List' on the left with a search bar and a list of profiles under 'Channel 1' and 'Channel 2'. Under 'Channel 2', 'new_profile' is listed. On the right, instructions for adding a new IVA profile are provided, along with a legend for profile states: Active (blue dot), Inactive (blue circle), and Unscheduled (grey circle).

保存すると左の Channel 2 に new_profile が表示されますので選択します。

Profile Information

Profile Description

Profile Name:
new_profile_1

Channel: Channel 2 Main-stream
Settings Template: Human Detection and Face Recognition

Analytic Module Settings: Single Module View Multiple-Module View

☒ **Human Detection and Face Recognition - General Settings** 

☒ Human Detection Off 

☒ Face Recognition On 



Human Detection and Face Recognition が有効となる条件を設定します。テストでは認識する範囲を設定します。

鉛筆のアイコンをクリックします。

Human Detection and Face Recognition - General Settings ×

Source Type > Channel
Channel 2 

 **Human Detection and Face Recognition - General Settings**
 Area



Gender Detection
☒ On

Eyewear Detection
☒ On

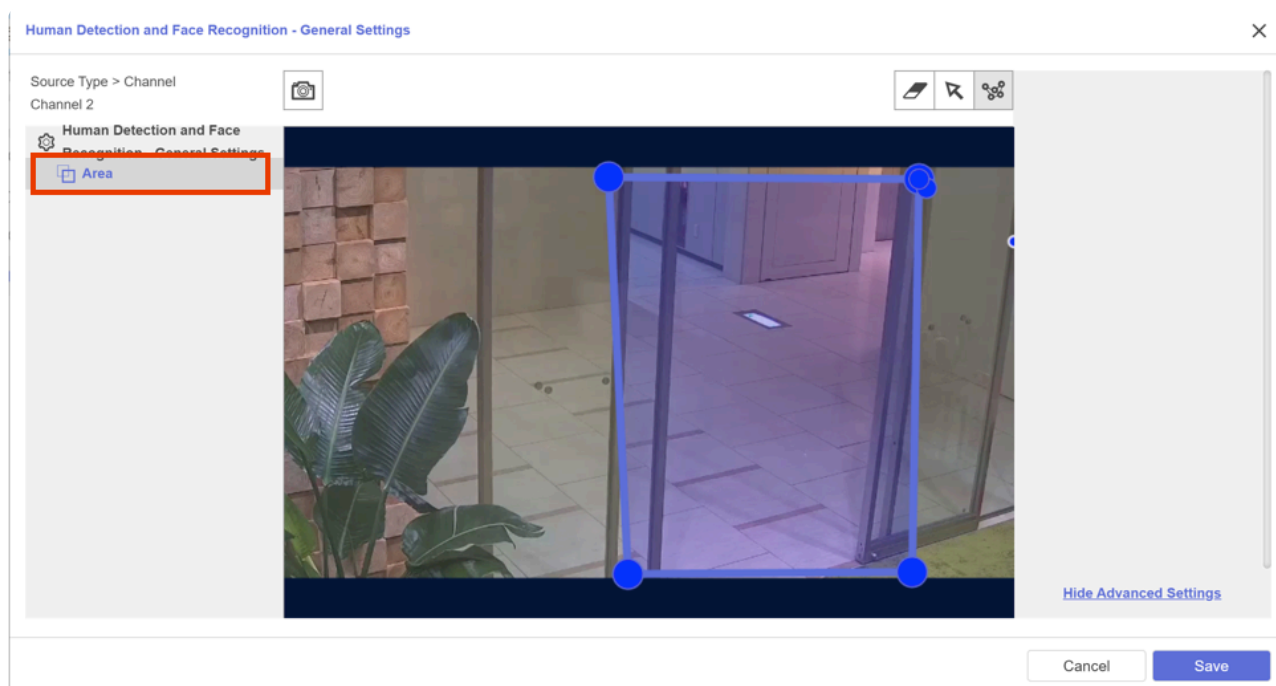
Age Group Detection
☒ On

Mouth Covering Detection
☒ On

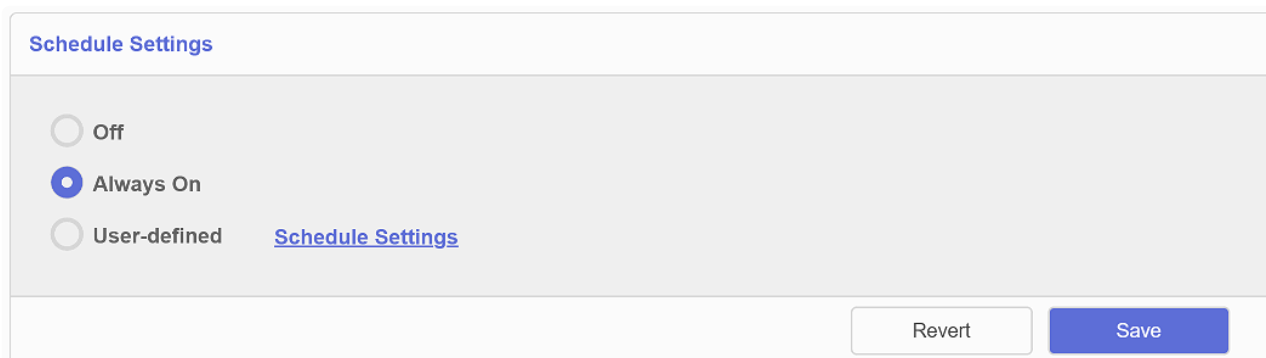
[Show Advanced Settings](#)

Cancel Save

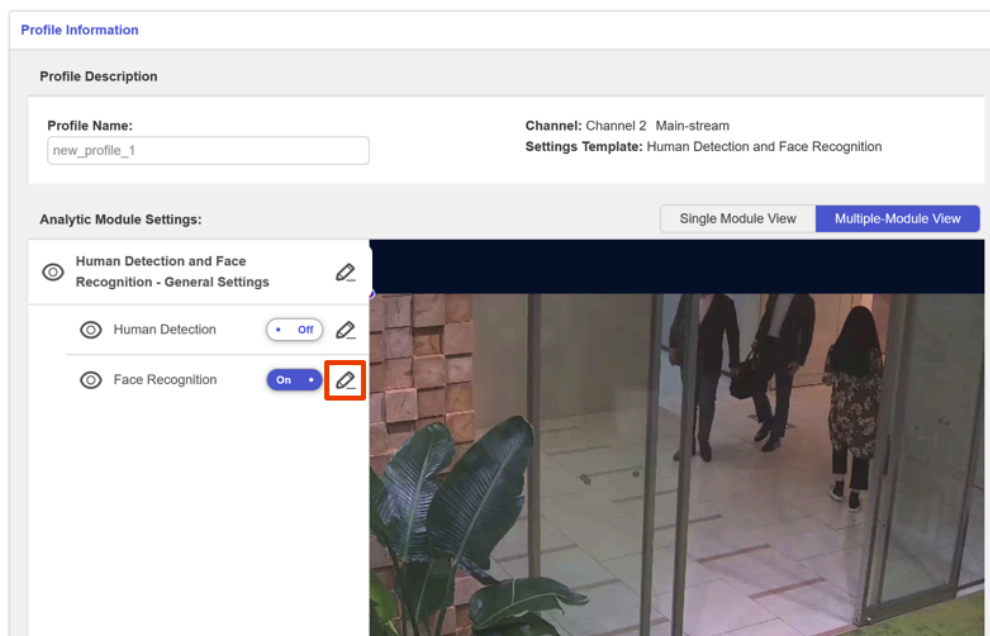
顔認識では性別推論、眼鏡対応推論、年代推論とマスク対応推論を個別に有効/無効を切り替えることが出来ます。デフォルトでは全てが有効となっています。



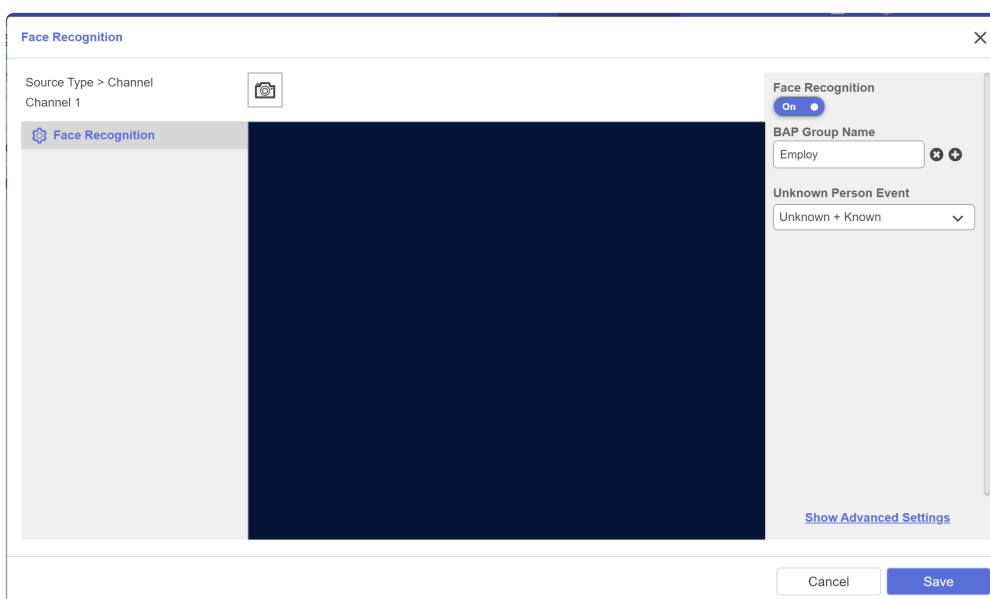
LINE を選択して、映像の認識する範囲をマウスで線を引いて囲みます。確定はダブルクリックです。
テストでは入り口のドアが開閉する範囲を指定しています。
Save ボタンを押して保存します。



設定を有効にするために Schedule Settings は Always On を選択して Save ボタンを押します。



続いてFace Recognition の鉛筆のアイコンをクリックします。



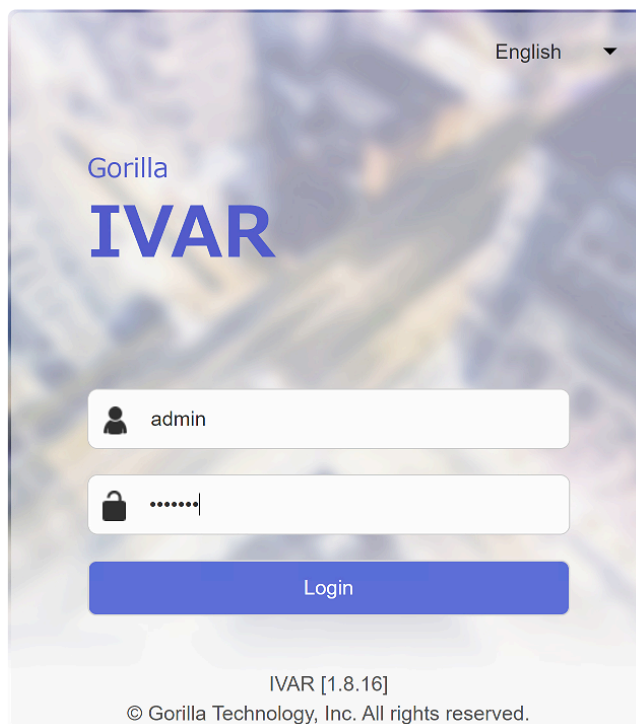
BAP Group Name にBAPで設定したグループ名を入力して Saveボタンで保存します。

3. Liveでの画像認識の確認

Channels メニューから Channel 2 を開いて View を選択するとライブ映像が表示されます。
右側にカメラの映像を認識してHuman Detection and Face Recognition のイベントが発生すると表示されます。

8.5.3. GravioとIVAR連携の設定

GravioとIVAR連携の設定

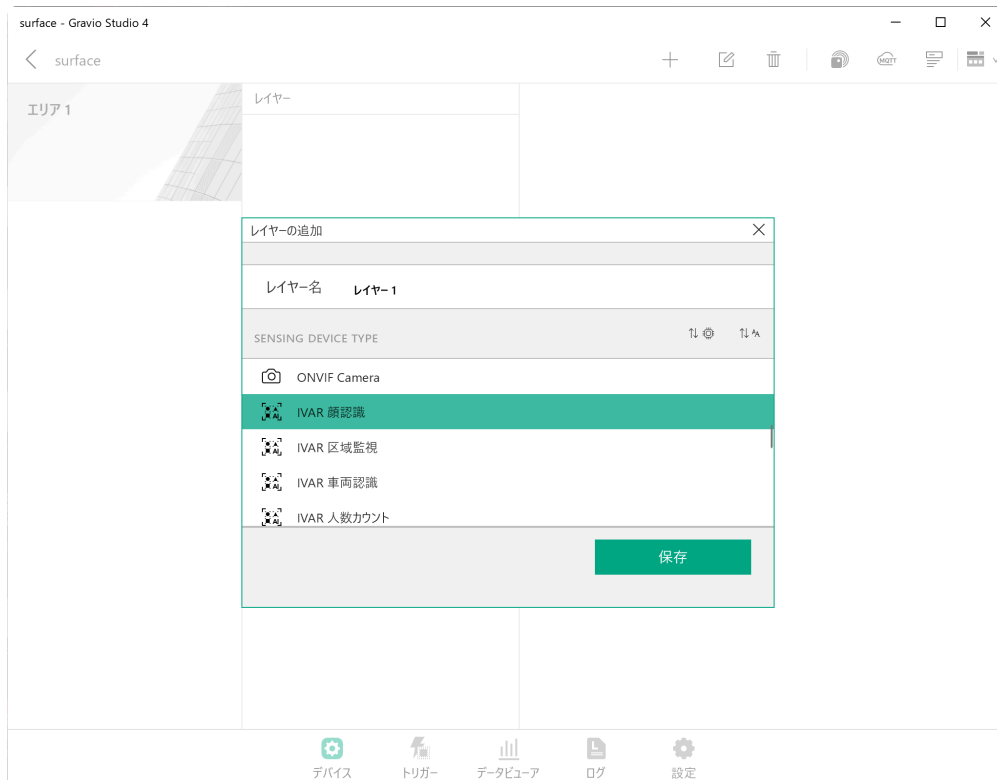


IVARにadminアカウントを使ってログインします。

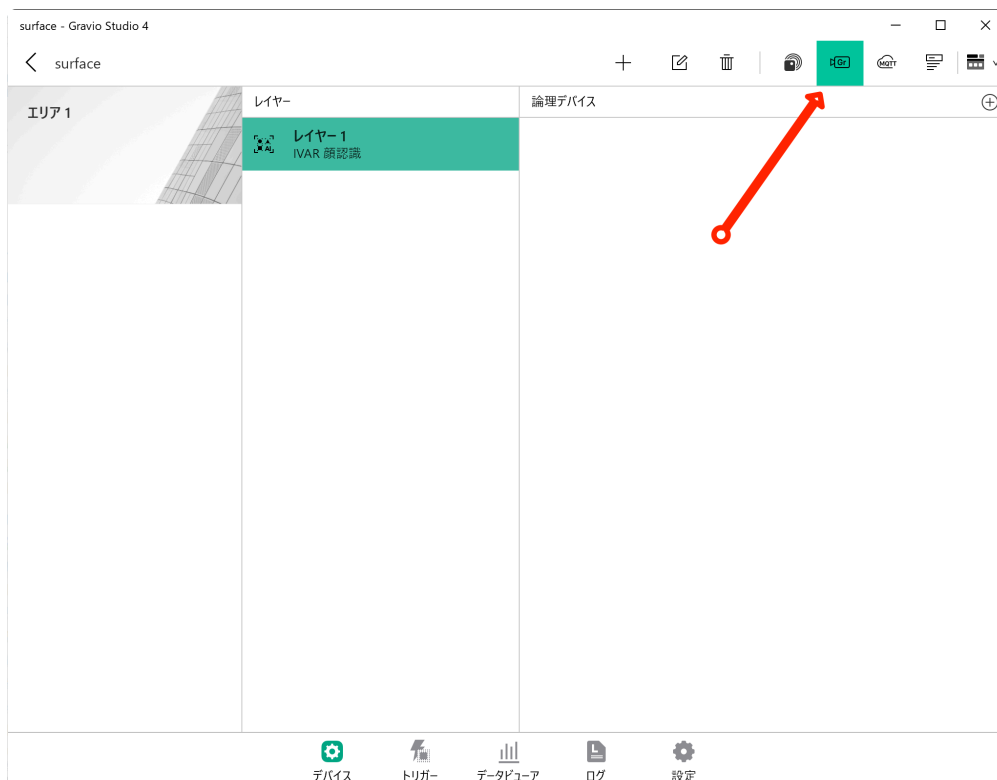
4. Gravio Studio での IVARの設定

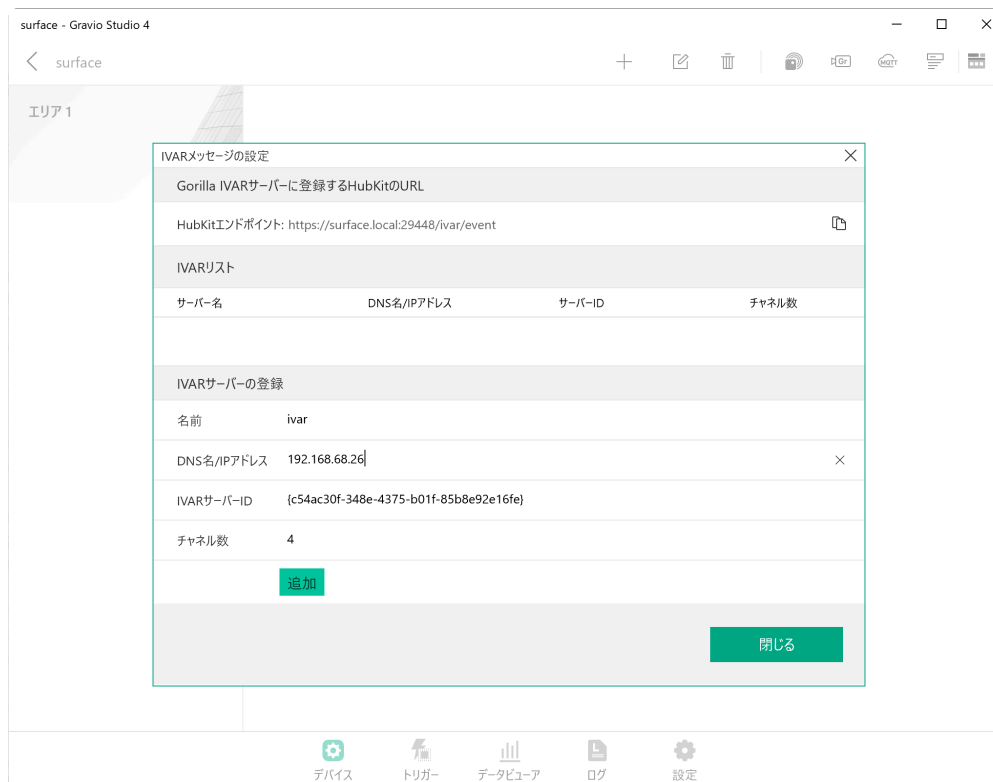
i About Basic Accounts Maintenance Service Connections	
IVAR InformationGeneral Settings	
ID	{c54ac30f-348e-4375-b01f-85b8e92e16fe}
Version	ivar-1.8.16.42905-std-win-x86_64
Total Channels	8
Maximum Analytic Channels	2
Analytic Core Channels	All : 1 Business Intelligence : 0 Human Detection : 1 Behavior Analysis : 0 License Plate : 0
Analytic Channel Expiry Date	2099-01-01 21:00:00
Analytic Core Version	BI-CORE v2.3.3 FDR-CORE v6.7.0 IPM-CORE v3.3.0 ODC-CORE v1.5.0 LPR-CORE v5.8.1 MMS/MMSC v1.3.1
IVAR NameGorilla IVAR Physical Address Latitude Longitude Description RevertSave	
License Product Key DetailsLicense Check Verified Expiry Date DeactivateResync	
System Information ⓘ	

System メニューを選択して About メニューを開きます。
IVAR Information の ID を {} を含めてコピーします。



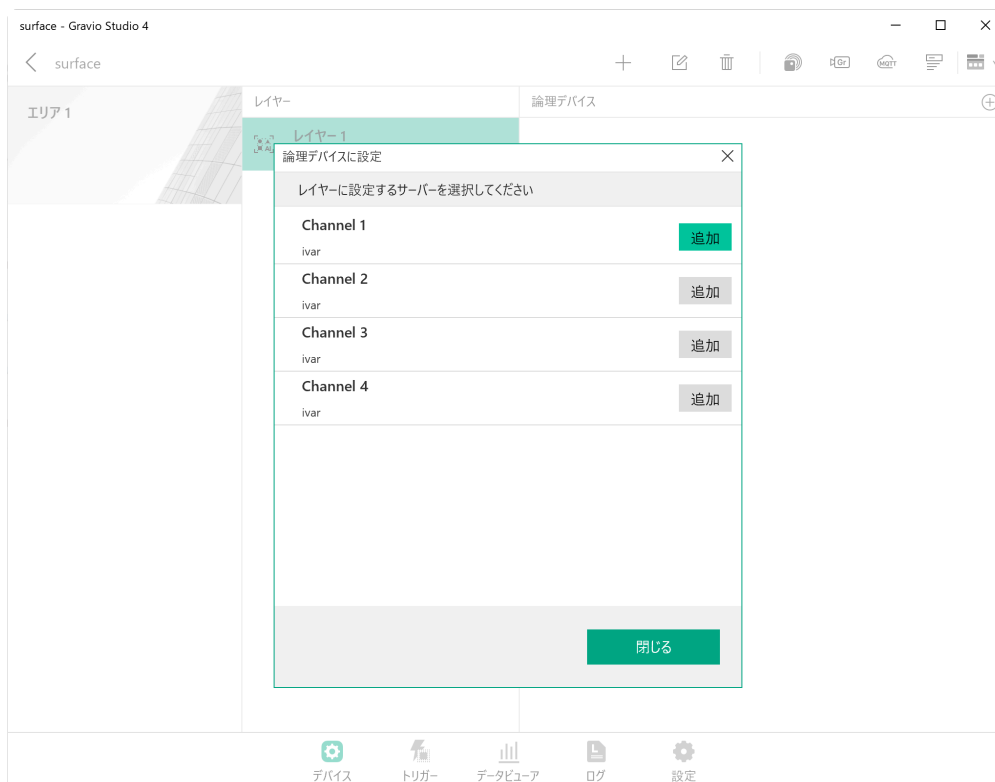
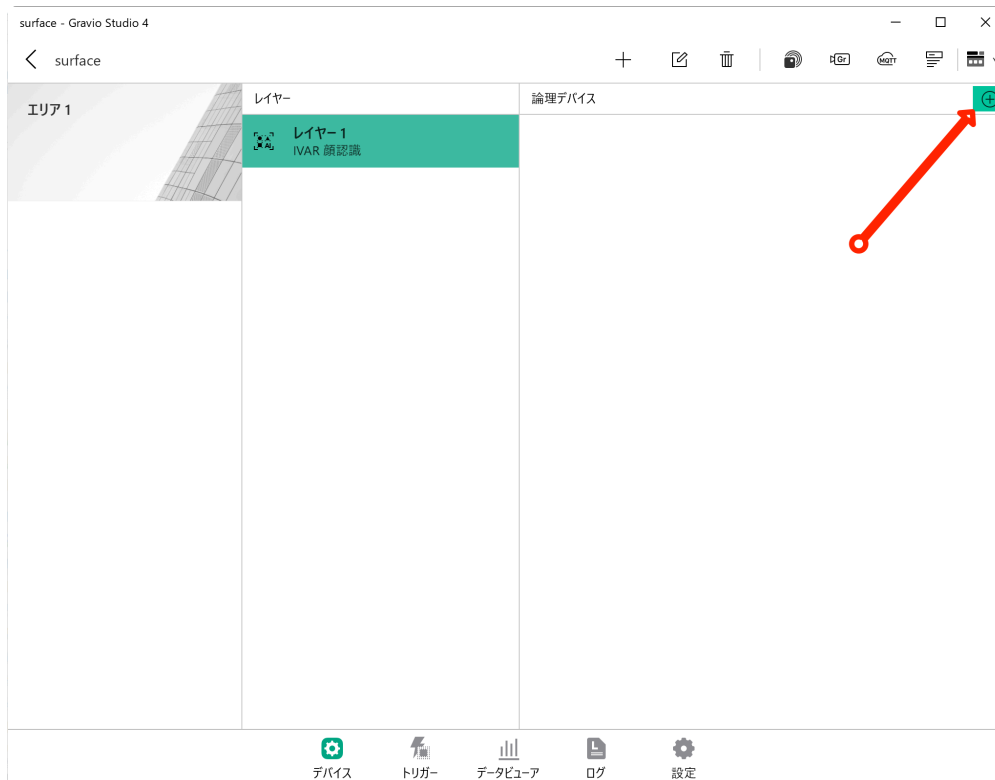
Gravio Studio のデバイスタブを選択してエリアとレイヤーを追加します。
レイヤーは IVAR 顔認識を選択して保存ボタンを押します。

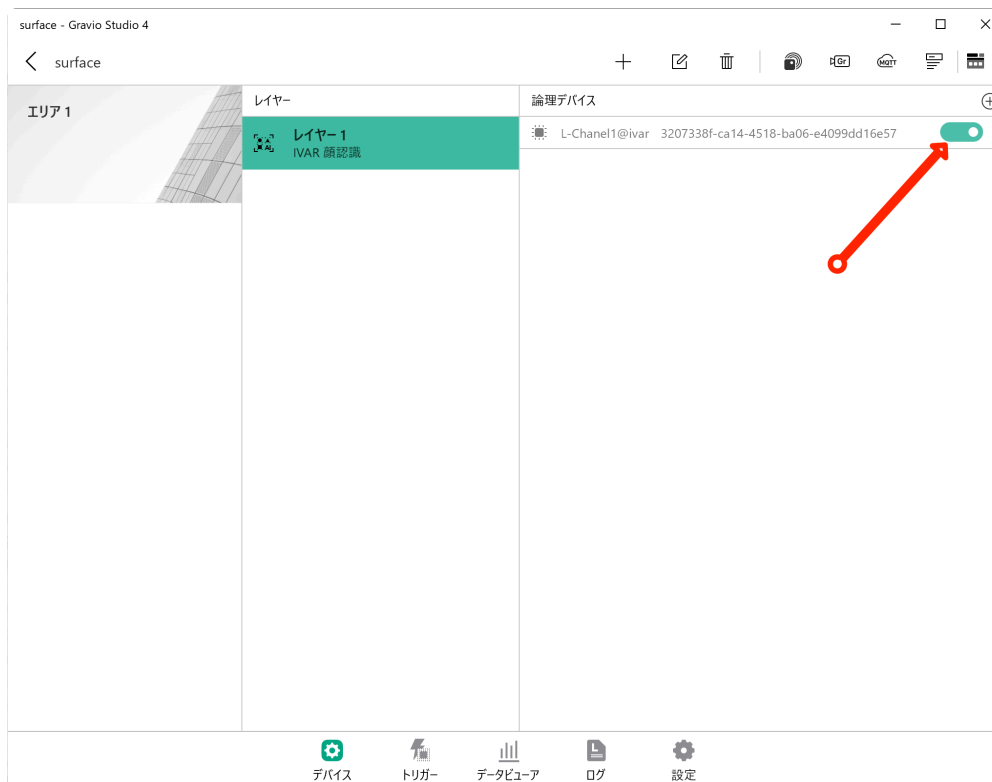




Gravio Studio から HubKit を選択してデバイスパネルから IVAR メッセージの設定画面を開きます。
DNS名前/IPアドレスを追加には IVAR/BAP がインストールされている PC のアドレスを入力します。
IVARサーバーIDを追加には、先ほどコピーした ID をペーストします。
追加ボタンを押すと登録されます。
続いて HubKit エンドポイント を右側のコピーボタンをクリックしてコピーします。

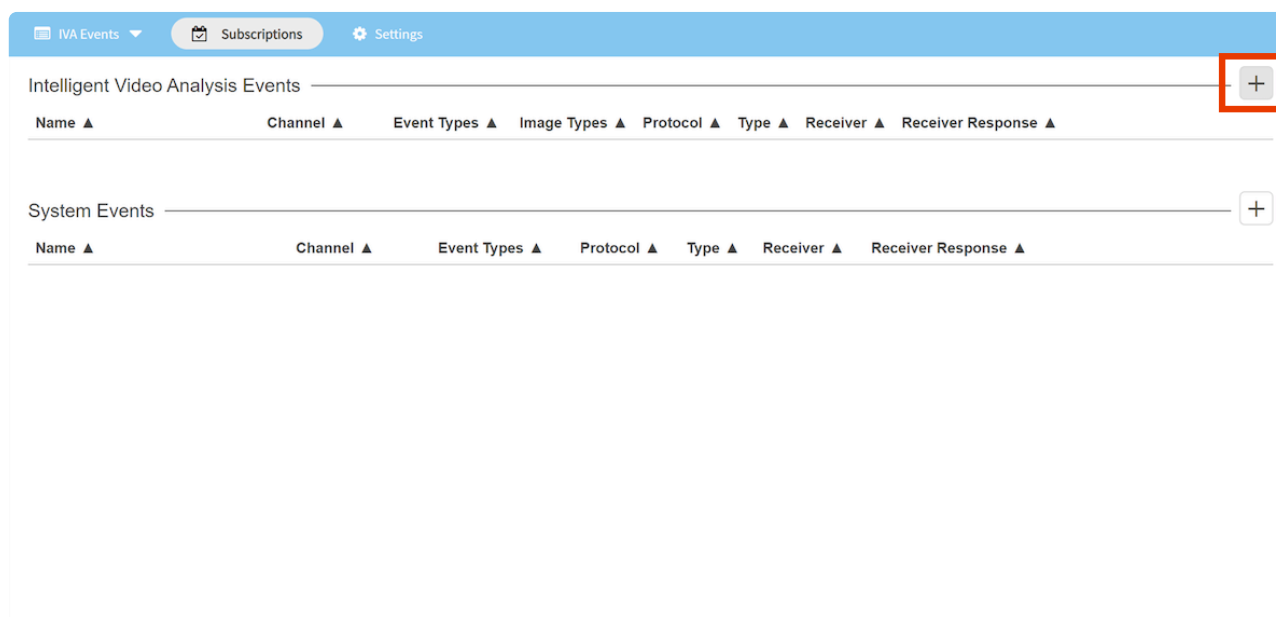
5. デバイスの登録





レイヤーにチャンネルをバインドして、保存された IVAR のレイヤーを有効にします。
GravioでのIVARの設定につきましては[こちら](#) もあわせて参照してください。

6. Gorilla IVARでのGravioのSubscriptionの登録



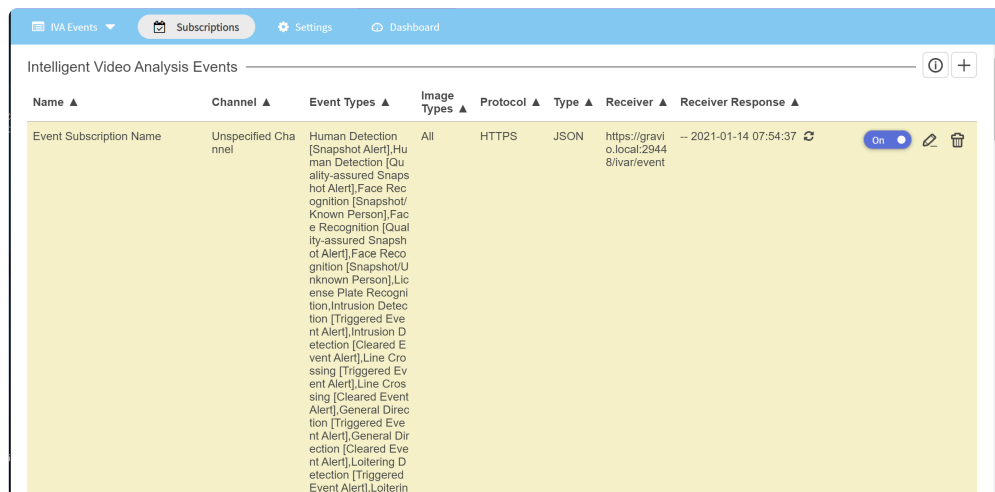
IVARの Events を選択して Subscriptions を開きます
+ボタンを押します。

The screenshot shows the 'Intelligent Video Analysis' configuration window. The 'Event Types' section is selected, and the 'Event Type Selection' dialog is open, showing a list of event types with checkboxes. The 'All Events' checkbox is checked. The 'Save' button is visible at the bottom right.

Enable にして Event Types Selection を開き All Events を選択します。

The screenshot shows the 'Intelligent Video Analysis' configuration window. The 'Event Types' section is selected, and the 'Event Type Selection' dialog is open, showing a list of event types with checkboxes. The 'All Events' checkbox is checked. The 'Save' button is visible at the bottom right.

Protocol を HTTPS に変更して、URL に先ほどコピーした HubKit Endpoint をペースとします。
Save ボタンを押して保存します。



保存されるとイベント一覧に表示されます。
削除する場合にはゴミ箱のアイコンをクリックします。

7. Gratio Studioでの データビューアでのイベント受信の確認

エリア	レイヤー	デバイス	サービス	センサータイプ	送信元ID	データID	受信日時	デバイス値
エリア 1	レイヤー 1	192.168.68.139	IVAR	FR_UNKNOWN	49cf51b6-1f1e-4440-98b7	3670d867-1c57-4c	2020/06/22 08:14	["fr":{"mmsReturn"
エリア 1	レイヤー 1	192.168.68.139	IVAR	FR_UNKNOWN	49cf51b6-1f1e-4440-98b7	fef46dcb-ad38-41	2020/06/22 08:14	["fr":{"mmsReturn"

データビューアを開き IVAR からイベントが送信されていることを確認します。

8.5.4. Gravio IVARトリガーの設定

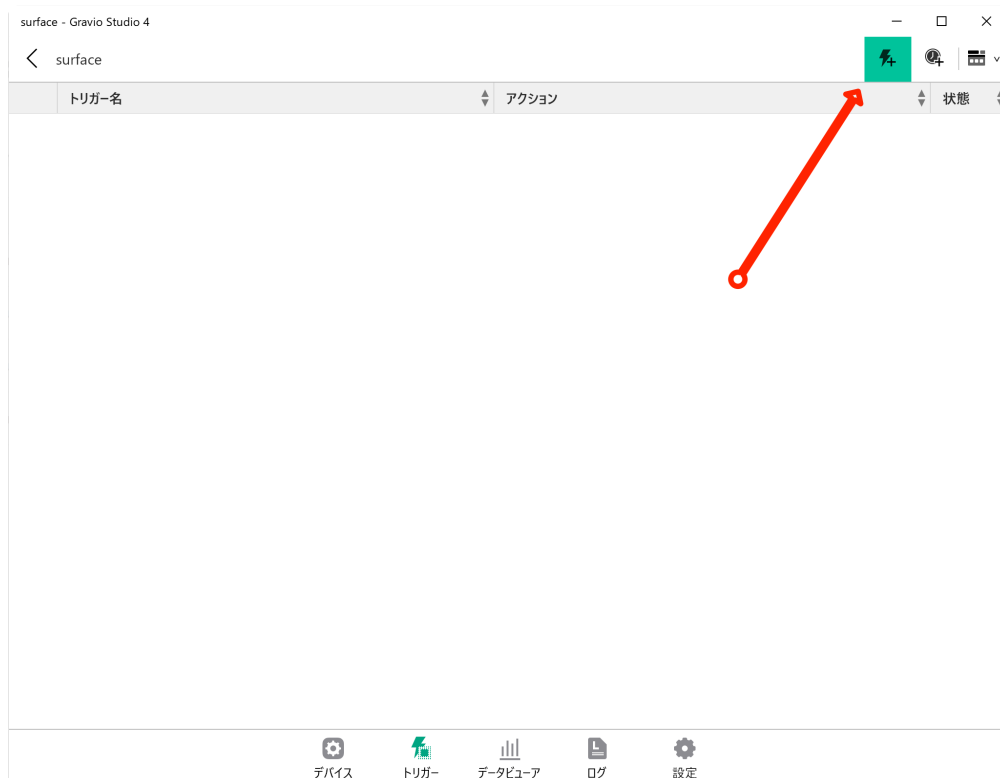
Gravio IVARトリガーの設定

7. 呼び出し用アクションの作成

IVARトリガーで呼び出すアクションを作成します。

テストでは Slack コンポーネントを配置したアクションを作成します。

8. IVARトリガーの設定



surface - Gravio Studio 4

< surface

トリガー名 アクション 状態

デバイストリガーの追加

トリガー名 新規イベントトリガー 1

エリア エリア 1

時間

条件

アクション

カレンダー なし

☐ 有効時間を指定する

開始 9 00 終了 17 00

間隔(ミリ秒) 5000

☐ 閾値によるトリガーを利用

☐ 実行中ならスキップする

キャンセル 保存

デバイス トリガー データビューア ログ 設定

surface - Gravio Studio 4

< surface

トリガー名 アクション 状態

デバイストリガーの追加

トリガー名 新規イベントトリガー 1

エリア エリア 1

時間

条件

アクション

次の条件を満たす場合に実行されます

条件数: 0/5

And

追加

キーレイヤー: レイヤー 1

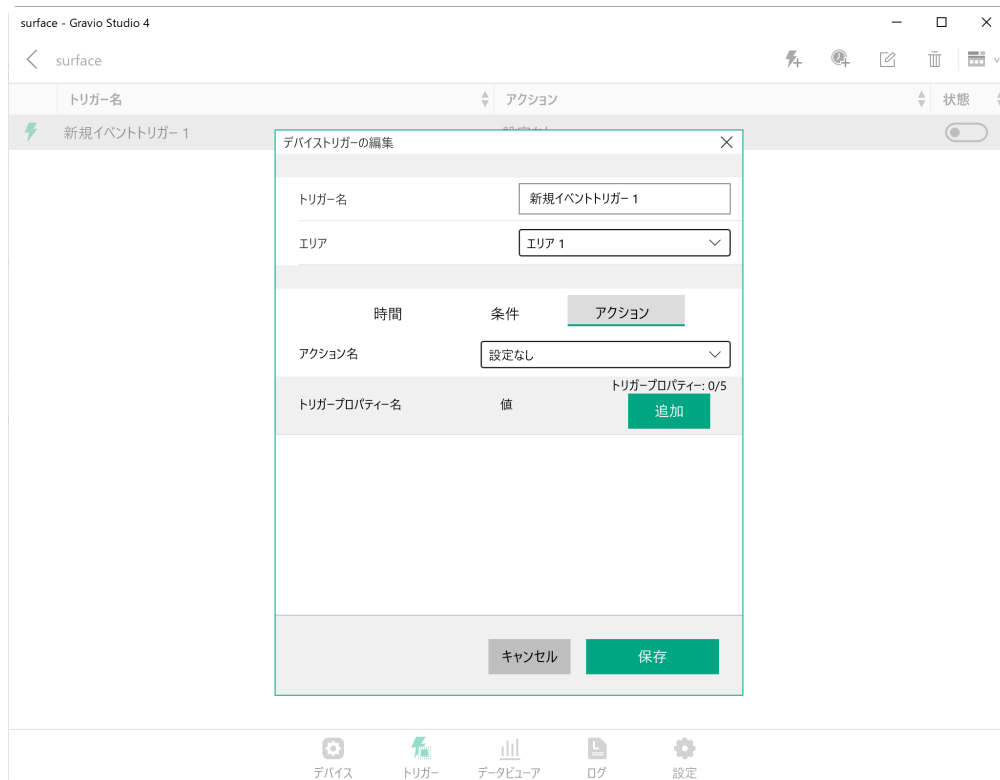
物理デバイスID:

イベントタイプ 任意のイベント

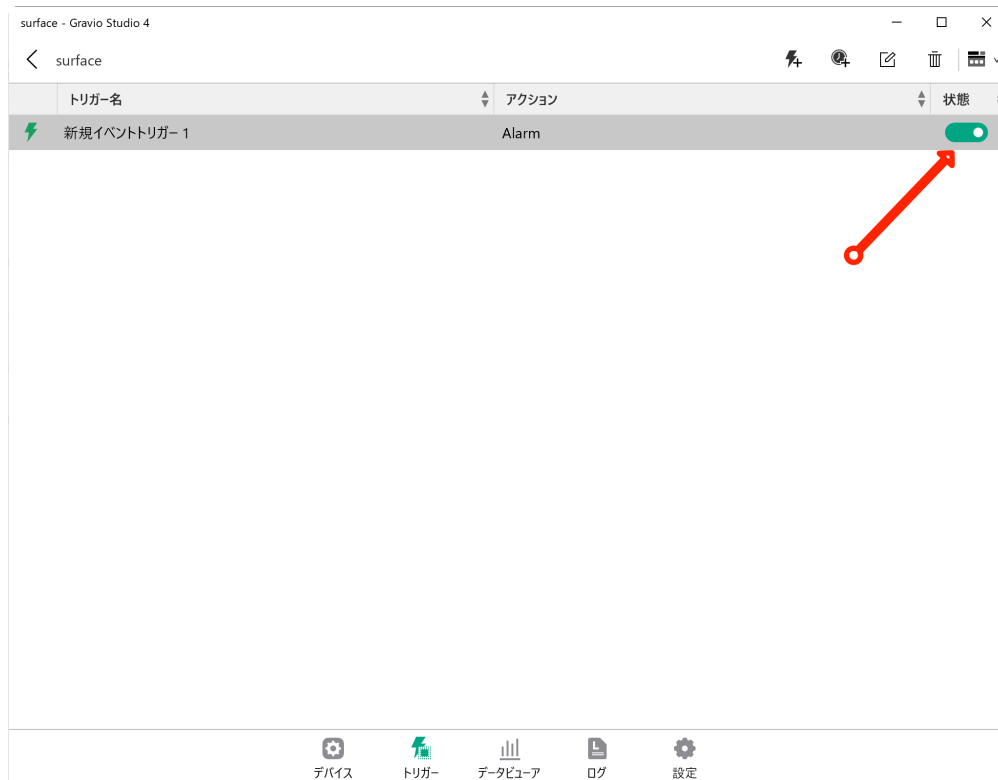
式

キャンセル 保存

デバイス トリガー データビューア ログ 設定



トリガータブを選択して+ボタンを押して新しいIVARTリガーを追加を開きます。
エリア、レイヤーを設定値に合わせて選択して、イベントタイプは 任意のイベント を選択します。
アクションは作成している Alarmアクションを選択します。
保存ボタンを押して保存します。
Gravioでのトリガーの設定につきましては[こちら](#) もあわせて参照してください。



保存した IVARトリガーを有効にします。

9. IVARトリガーでのアクションの実行の確認

有効にして IVARトリガー条件にマッチするイベントが IVARから送信されたタイミングで指定のアクションが実行されます。

テストでは AlarmアクションにはSlack コンポーネントを配置していますので、Slack に通知が届きます。

9. より深くGravioを利用するための情報

9.1. 物体認識の機械学習モデル作成手順

ここでは、物体認識の機械学習モデル (darknet YOLO v2形式) で作成し、TensorFlowで利用する形式に変換するやり方での作成方法を、以下の手順で説明します。

(Gravio HubKit Version 4.0 以降はTensorFlowモデル形式のみ動作可能です。)

1. 画像データの収集
2. 学習データの作成
3. 機械学習モデルの作成
4. TensorFlowで用いる形式への機械学習モデルの変換

[補足資料: Labellmg によるアノテーションデータ作成](#)

1. 画像データの収集

物体認識の機械学習データに用いる、画像データを以下の手順で準備します。

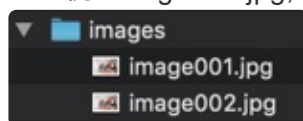
1. PC上の作業フォルダ内に、画像格納用のフォルダを新規作成しimagesと名付けます。

2. 認識する物体が含まれる画像データを準備し、imagesフォルダの配下に置きます。

その際、以下の点にご注意ください。

- ・画像ファイルフォーマットはJPEGにし、拡張子を「.jpg」に統一してください。
(.jpegや.pngなどの異なる拡張子が混在すると、学習時にエラーが発生します)
- ・画像サイズは416×416ピクセル以上が望ましいです。
- ・画像枚数は認識する物体ごとに、数百枚～数千枚程度が目安です。
- ・1画像あたりのファイルサイズを300～500KB程度に調整すると、処理がスムーズに進みます。
- ・ファイル名を「英数表記+連番」にすることで、管理が便利になります。

例: image001.jpg, image002.jpg, ...



2. 学習データの作成

学習データは、(1) 準備した画像データ、(2) クラス情報データ (認識する物体の名称リスト)、および (3) 各画像に対するアノテーションデータ (認識する物体の画像内の矩形領域を示した注釈情報) の3つで構成されます。クラス情報データとアノテーションデータは、以下の手順で追加します。

1. クラス情報データは、認識する物体の名称リストです。テキストエディタでファイルを新規作成し、認識する項目を各行に1つずつ記載します。例えば、person, dog, cat の3つを認識する学習モデルを構築する場合、各行にそのラベルを記載します。この記載順で、認識する物体の識別子は、先頭行を0として、通し番号で割り当てられます。
(person=0, dog=1, cat=2となります)

```
person
dog
cat
```

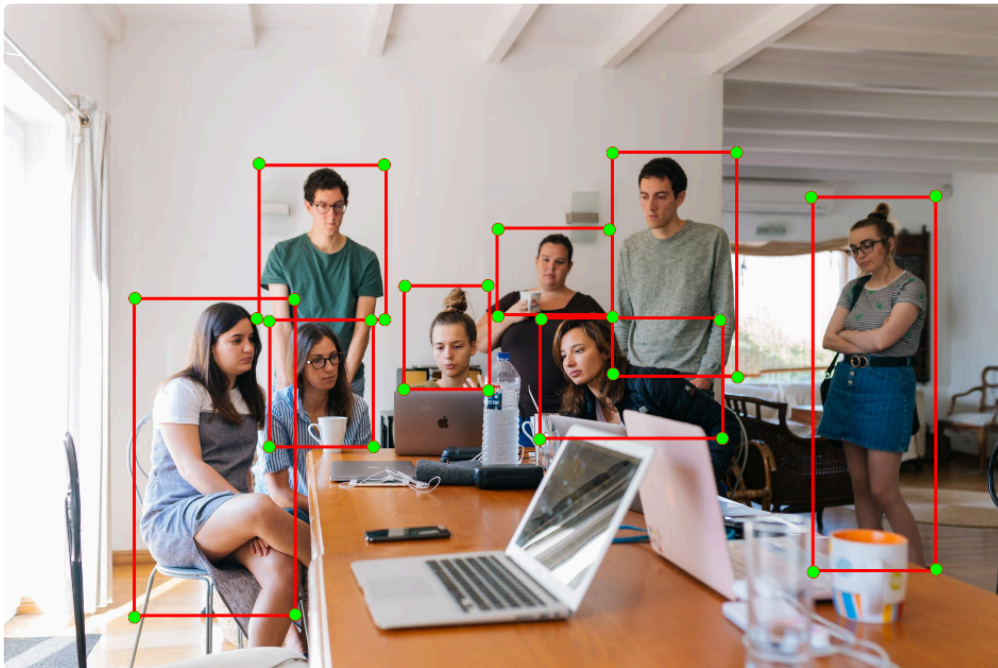
2. クラス情報データのファイル編集を完了し保存します。ファイル名称は任意に決めることができますが、ここでは、imagesフォルダと同じPC上の作業フォルダ配下に、obj.names という名称にします。

```
▶ images
  obj.names
```

3. 次に、PC上の作業フォルダ内に、アノテーションデータファイル格納用のフォルダを新規作成し、labelsと名付けます。作業フォルダ直下は以下のようになります。

```
▶ images
▶ labels
  obj.names
```

4. images フォルダ配下の各画像に含まれる物体に対して、アノテーションを付加します。各物体のアノテーションは、矩形領域で表現します。以下の写真1では「person」のアノテーションを図示しています。



実際には、アノテーションデータはlabelsフォルダ配下に、テキストファイル形式で保存します。その際、対応する画像データのファイル名の拡張子.jpgを.txtに変換した名称で保存する必要があります。

```
▼ images
  image001.jpg
  image002.jpg
▼ labels
  image001.txt
  image002.txt
  obj.names
```


アノテーションデータファイルには、各行に1つの物体についてのアノテーションデータとして、物体の識別子(数字)、矩形領域の中心点の座標(x,y)、およびサイズ(w,h)の情報を記載します。

図1の1行目は、識別子0(=person)、(x,y)=(0.221543, 0.712291)、(w,h)=(0.166489, 0.532322)を表現しています。認識する物体の矩形領域の座標(x,y)、およびサイズ(w,h)は、画像全体との比で表現し、値は0以上1以下の範囲となります。

```
0 0.221543 0.712291 0.166489 0.532322
0 0.314362 0.618915 0.112766 0.278532
0 0.318617 0.416600 0.125532 0.346369
0 0.542021 0.486433 0.140426 0.281724
0 0.644149 0.591381 0.189362 0.255387
```

GUIで、アノテーションを画像に付加するツールとして、以下のものが公開されています(英語)。詳細は本ドキュメント最後に記載しています、「補足資料:Labellmgによるアノテーションデータ作成」をご参照ください。

名称 :Labellmg

概要 :Widows, Mac等マルチプラットフォームで動作するアノテーションデータ作成ツール。

入手先 :<https://github.com/tzutalin/labellmg>

3. 機械学習モデルの作成

学習モデルを以下のステップで作成します。

1. darknet の準備

学習モデルを作成するコンピュータのターミナル環境で、以下の手順により、darknet環境を構築します。

```
$ git clone https://github.com/pjreddie/darknet
$ cd darknet
$ make
```

GPU環境(CUDA環境)で学習環境を構築する場合は、最後のmakeコマンドを実行する前に darknet/Makefile を編集し、先頭行を GPU=1 と設定します。

2. 初回時の学習用に、学習済の重みデータを、以下の場所から取得します。

```
$ wget https://pjreddie.com/media/files/darknet19_448.conv.23
```

3. 学習データを設置します。

```
$ cd data
$ mkdir learning_data (任意のフォルダ)
```

「1. 画像データの収集」、「2. 学習データの作成」で準備した画像データ、アノテーションデータ、クラス情報データを以下のように配置します。

```
darknet/data/learning_data/images/*.jpg
darknet/data/learning_data/labels/*.txt
darknet/data/learning_data/obj.names
```

4. 学習用データリスト、検証用データリストを準備します。

```
darknet/data/learning_data/train.txt
darknet/data/learning_data/test.txt
```

各々、学習するデータ、検証するデータの一覧を各行に記載します。train.txt と test.txt に、ファイル名の重複がないように、以下のように data フォルダ以下のファイルパスを記載します。学習時には train.txt に記載の画像ファイルのみが使用されます。

```
data/learning_data /images/img001.jpg
data/learning_data /images/img002.jpg
data/learning_data /images/img005.jpg
...
```

5. 学習設定ファイルを準備します。

darknet/cfg/obj.data を新規作成し、以下のように記載します。classes の値は、認識させたい物体の数を記載します。

```
classes= 3
train = data/learning_data/train.txt
valid = data/learning_data/test.txt
names = data/learning_data/obj.names
backup = backup
```

6. モデル設定ファイルを準備します。ここではすでに用意されている darknet/cfg/yolov2-voc.cfg の設定ファイルをコピーし、内容を書き換えます。

```
$ cp darknet/cfg/yolov2-voc.cfg darknet/cfg/obj.cfg
```

新しくできた obj.cfg を、以下のように調整します。Filters の値は、クラス数が 3 の場合は、40 となります。 $((3+5)*5 = 40)$

3行目: batch=64 4行目: subdivisions=8 237行目: filters=(実際に認識させるクラス数 + 5) * 5 242行目: anchors
(※) 244行目: classes=3 (実際に認識させるクラス数)

(※) anchors については、そのままの値でも学習を進めることができますが、以下のようなツールにより、anchors の値を再計算することで、精度向上する場合があります。

参考: anchors 値の再計算ツール

https://github.com/Jumabek/darknet_scripts

7. 学習開始

darknet フォルダで、以下を実行すると、darknet/backup/フォルダの下に、学習モデルデータ (*.weights) が生成されます。

```
$ ./darknet detector train cfg/obj.data cfg/obj.cfg darknet19_448.conv.23
```

学習が開始すると、以下のような出力が繰り返し出力されます。

```
...
Region Avg IOU: 0.905463, Class: 0.626457, Obj: 0.836158, No Obj: 0.006927, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.837368, Class: 0.976554, Obj: 0.842887, No Obj: 0.006556, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.893971, Class: 0.960798, Obj: 0.876742, No Obj: 0.007555, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.872773, Class: 0.711470, Obj: 0.888209, No Obj: 0.006718, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.894234, Class: 0.986793, Obj: 0.884681, No Obj: 0.006898, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.847557, Class: 0.994500, Obj: 0.849292, No Obj: 0.006747, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.802273, Class: 0.650419, Obj: 0.794552, No Obj: 0.006926, Avg Recall: 1.000000,
count: 8
Region Avg IOU: 0.866342, Class: 0.840821, Obj: 0.860806, No Obj: 0.006149, Avg Recall: 1.000000,
count: 8
9531: 0.661146, 0.704077 avg, 0.001000 rate, 2.704480 seconds, 609984 images
...
```

最後の行の先頭の数字 (赤字で示しています) は、ステップ数を表しています。学習が進むと誤差の平均値 (青字で示しています) が小さな値になります。

学習は、任意のタイミングで停止できますが、精度が高い認識モデル作成を行うためには、誤差の平均値は1未満となることが目安になります。また、数万ステップ以上が必要となる場合もあり、動作環境によっては、完了までに数日から数週間が必要となる場合があります。

8. 学習の再開

学習の再開は、以下のように行います。backupフォルダの配下には、拡張子が .backup もしくは .weights という名称のファイルが多数生成されます。いずれかのファイルで、最新のもの (通常は、.backupの拡張子のもの) を指定すると、最後に保存された学習モデルを使用して、学習を再開できます。

```
$ ./darknet detector train cfg/obj.data cfg/obj.cfg backup/(backupフォルダ配下の最新のファイル名)
```

以上の手順の結果、学習完了後には、以下の3つのファイルを利用することができます。

darknet/backup/*.weights (*印部分は、**backup**フォルダ配下に生成されるの最新のファイル名を指定)
darknet/cfg/obj.cfg,
darknet/cfg/obj.names

4. TensorFlowで用いる形式への機械学習モデルの変換

「3. 機械学習モデルの作成」で作成した3つのファイル (*.weights, obj.cfg, obj.names)をTensorFlow (<https://www.tensorflow.org/>) で記述したプログラムで利用できる、protocol buffers 形式(*.pb)に変換する手順として、変換ツール DarkFlow (<https://github.com/thtrieu/darkflow>) を用いた例を示します。

1. TensorFlow 環境のインストール

TensorFlow公式のインストール手順に従い、作業を行うコンピュータ環境に、Tensorflowをインストールします。
(Python 3 での例を示します)

<https://www.tensorflow.org/install>

```
$ pip3 install --upgrade pip  
$ pip3 install tensorflow==1.15
```

(tensorflow のバージョンは、変換後のモデル使用時の環境と一致させる必要があります)

2. Python 用に OpenCV をインストールします。

```
$ pip3 install opencv-python
```

3. DarkFlow を以下の手順で設定します。

```
$ git clone https://github.com/thtrieu/darkflow.git  
$ cd darkflow  
$ sed -i -e 's/self.offset = 16/self.offset = 20/g' darkflow/utils/loader.py  
$ python3 setup.py build_ext --inplace  
$ pip3 install -e .  
$ pip3 install .
```

4. 「3. 機械学習モデルの作成」で作成した3つのファイルを格納する models フォルダを作成します。格納する各ファイルの名称は、仮に obj.weights, obj.cfg, obj.names とします。

```
$mkdir models
```

(models フォルダの下に obj.weights, obj.cfg, obj.names の3つを格納)

5. DarkFlow のフォルダ配下で、以下のコマンドにより、各ファイルを変換します。

```
$ flow --model models/obj.cfg --load models/obj.weights --labels models/obj.names --savepb
```

6. 以上の手順により、built_graph フォルダの配下に、変換後の2つのファイルが生成されます。

obj.pb ... 機械学習のニューラルネットワークモデルファイル

obj.meta ... メタ情報ファイル

9.2. 補足資料:Labellmg によるアノテーションデータ作成

「2. 学習データの作成」の最後では、アノテーションデータ作成ツールLabellmgを紹介しました。このツールによる、アノテーションデータ作成は、以下の手順で行います。

1. 入手先 (<https://github.com/tzutalin/labelImg>) のページ記載の「Installation」の項目から、使用するPCプラットフォーム (Windows, Mac, Linuxなど) と現在のpython バージョンを確認し、手順に従って、インストールを行います。

2. 「Usage」の項目にある「Steps(YOLO)」の項目に従い進めていきます。

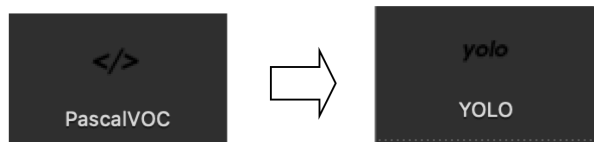
(1) Labellmgをインストールしたフォルダ配下の data/predefined_classes.txt に認識させたいクラスの名称を記載します。「2. 学習データの作成」の手順1で作成した obj.names の内容と同じものを記載します。例えば、person, dog, cat の認識の場合は以下の内容になります。

```
person
dog
cat
```

3. 「Installation」に記載の方法で、python コマンドで、Labellmg を起動します。例えば、python3 の環境では、以下のコマンドをターミナル環境で入力し、起動します。

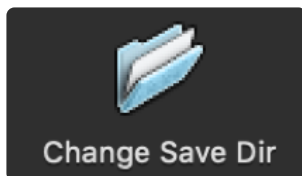
```
$ python3 labelimg.py
```

4. Labellmg の画面左のツールバーにある「PascalVOC」のボタンをクリックし、表示を「YOLO」に切り替えます。(起動する際に、必ずYOLOに設定を切り替えてください。)



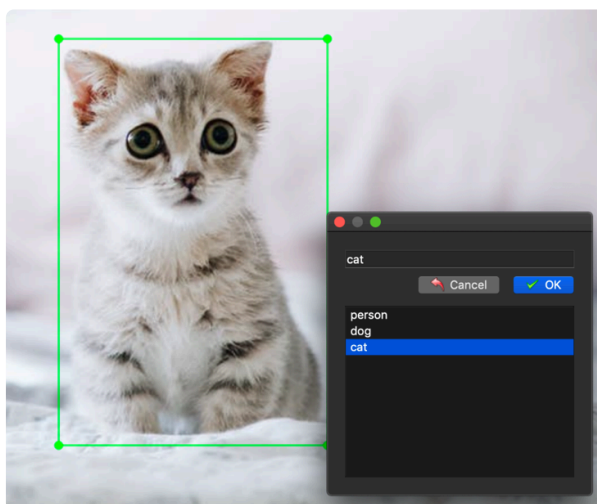
5. Labellmg の画面左のツールバーにある「Open Dir」ボタンをクリックし、PC上の作業フォルダ内に、画像格納用のフォルダとして作成したimages を選択します。すると、Labellmgの画面中央に、images フォルダ内の最初の画像が表示されます。

6. 作業後のアノテーションデータファイルの保存先を指定します。画面左のツールバーにある「Change Save Dir」ボタンをクリックし、「2. 学習データの作成」のステップ3で作成した、PC上の作業フォルダ内「labels」フォルダを選択します。

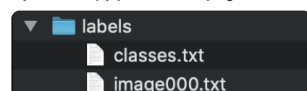


7. 画面が表示されている状態でキーボードの「w」を押すと、アノテーションを画像に付加することができるようになります。画像の上でマウスをドラッグして、矩形領域を決定します。

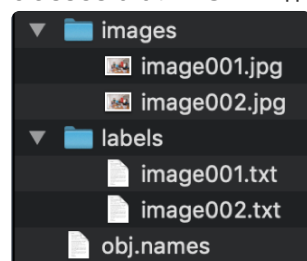
8. マウスで矩形領域を決定すると、data/predefined_classes.txt記載の物体の名称から、該当するものを選択する画面が表示されますので、該当するものを選択し、OKボタンを押します。以下の例では cat を選択しています。



9. 1つの画像内のすべての物体に対して、アノテーションの矩形領域を決定完了した後に、画面左のツールバーにある「Save」ボタンを押します。上のステップ6で指定した「labels」フォルダ配下に、画像名に対応する、～.txt という名称のアノテーションデータファイルが生成されます。同時に classes.txt という名称のファイルも labels フォルダ配下に生成されます。



10. images フォルダ内のすべての画像に対して、アノテーション付加の作業を完了しましたら、labels 内の classes.txt はPC上の作業フォルダ内に、移動し、「obj.names」とファイル名に変更します。



以上の作業で、「2. 学習データの作成」の作業が完了します。

9.3. Gravio Coordinatorを使う(Gravio Enterpriseサービス契約のみ)

Gravio は HubKit または Gravio Hub 自体が動作するコンピュータ(Mac, Windows, Linux)を管理サーバで管理したい場合は、Gravio コーディネーターを使うことができます。(Gravio Enterpriseサービス契約のみ)

Gravio コーディネーターについての詳細は、[Gravio コーディネータのドキュメント](#) をご覧ください。

10. 3rdパーティー製品との接続

11. Gravio Hub (Gravio Hub 2)

Gravio Standard 以上のサービスメニューをご契約の方は、Gravio Hubをレンタルすることが可能です。

Gravio Hub 2はGravio Hub version 1の後継機です。以後「Gravio Hub」という用語は「Gravio Hub 2」を指します。



Gravio Hubを使えば、1台で最大64個のセンサーのデータを処理できます。

11.1. Gravio Hub 2のインジケータランプの状態とボタン操作

インジケータランプの状態

Gravio Hub 2の側面にあるLEDは、さまざまな色やパターンで点灯し、下記の内容を確認することが出来ます。

LED	状態	コメント
青色点灯	箱から出した最初の起動時	標準のアクセスポイントモード
白色点灯	Wifiセットアップ後または有線LAN接続時	通常運用モード
緑色点滅	LTEモード	WAN未接続
緑色点灯	LTEモード	LTEモード + WAN接続済み または wifiルータモード (HubKit起動)
赤色2回点滅・白色(または緑色)2回点滅の繰り返し	時刻設定エラー	システムのクロック設定時にエラーが発生した場合
赤色4回点滅・白色(または緑色)4回点滅の繰り返し	初期設定エラー	システムの初期設定時にエラーが発生した場合
青LED がグラデーション付き点滅	コマンドモード	ボタン操作を参照

ボタン操作

Gravio Hub 2ではボタンをダブルクリックすることで下記のコマンドを実行するコマンドモードに入ります。

APモード

ボタンを1回押すとGravio Hub 2はAPモードとなります。LEDは青LED点灯になります。

注)LTEモードの時はAPモードには移行しません。

電源停止

ボタンを押し続けるとGravio Hub 2は電源を停止します。停止処理中はLEDは赤LED点灯になり、電源が消えるとLEDは消灯します。

コマンドモードの終了

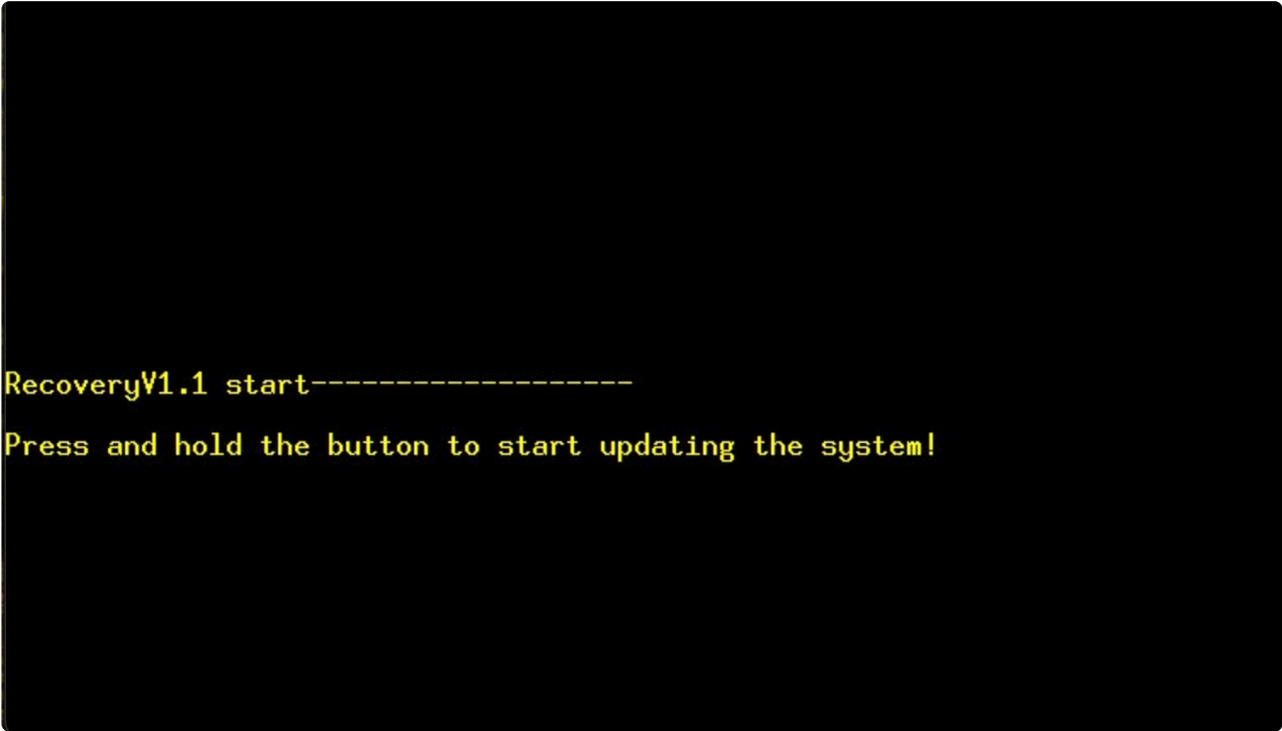
ダブルクリックでコマンドモードを抜けます。LEDは白LED点灯に戻ります。

11.2. Gravio Hub2のファクトリーリセット手順

Gravio 5.0.2のリリースからGravio Hub2を工場出荷状態に戻すためのSDカードが同梱されています。
(SDカードが同梱されていない場合は、弊社サポートまでご連絡ください。個別に送付させていただきます。)

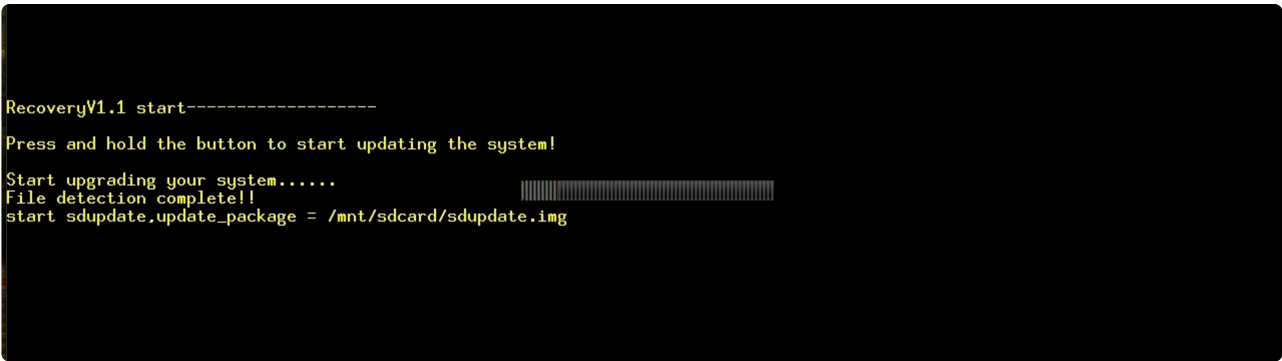
初期化の手順

1. Gravio Hub2のSDカードスロットにブルーのラベルのSDカードを挿入する
2. Gravio Hub2のHDMIポートにディスプレイを繋ぐ。
3. Gravio Hub2に電源ケーブルを繋いで起動する (これでSDカードからGravio Hub2が起動します。)
4. 起動すると、下記の画面が表示されます。



```
RecoveryV1.1 start-----
Press and hold the button to start updating the system!
```

5. Gravio Hub2の側面にあるボタンを **長押し** します。
6. 下記の画面のように初期化が始まります。



```
RecoveryV1.1 start-----
Press and hold the button to start updating the system!

Start upgrading your system.....
File detection complete!!
start sdupdate.update_package = /mnt/sdcard/sdupdate.img
```

完了するとGravio Hubのランプが **緑の点滅** になっていれば、SDカードからの書き込みは完了しています。
「Please remove SD CARD!!!. wait for reboot.」というメッセージが表示されますので、**SDカードのみ抜いてください。**(電源ケーブルはそのまま繋いでおいてください。)

SDカードを抜くと、自動でリブートされて最終初期化処理にはいります。

これはGravio Hubの内部の初期化処理を行います。

青の点灯 から一度 白のランプ になり、その後、緑と青の交互点滅 になります。(この状態が内部初期化処理です。)

これが完了するとしばらくして、最後に 青ランプの点灯 状態になります。

電源を抜いてください。

これで初期化は完了です。

11.3. SDカードを使ったWiFi設定

Gravio 5.0.2のバージョンから、SDカードを使ってGravio Hub2のWifiの設定をすることができます。

設定手順

下記の3行を記述して、「wificonfig.txt」というファイル名で保存してください。
青色の部分をご自身の環境に合わせて書き換えてください。

——— wificonfig.txt ———

```
ssid Your WiFi Access Point SSID  
password Your WiFi Access Point's password  
keep-password true or false
```

ssid	接続したいWifiのアクセスポイント名
password	接続したいWifiのアクセスポイントのパスワード
keep-password	true – 設定後にこのファイルに記述されているpasswordの内容を保持する。false – 設定後にこのファイルに記述されているpasswordを消去する

このファイルをSDカードにコピーして、SDカードを刺したままGravio Hubの電源をいれる。

1度起動して読み込まれれば、このWiFiの設定はGravio HubのWifi設定に保存されるので、SDカードを抜いた後でも、設定されたWifiにGravio Hubが接続されて起動します。

11.4. SDカードを使ったGravio Hub設定

設定手順

下記の4行を記述して、「ghubconfig.txt」というファイル名で保存してください。
青色の部分をご自身の環境に合わせて書き換えてください。

```
----- ghubconfig.txt -----  
enable-ssh true または false  
restore-backup バックアップファイル名.zip  
set-hostname Host名  
restore-hubid true または false  
-----
```

enable-ssh	true – GravioHubのSSHの利用を許可する。false – GravioHubのSSHの利用を許可しない。
restore-backup	Gravio Hubにリストアしたいバックアップファイルを指定する。
set-hostname	記述された文字列をHost名として設定する
restore-hubid	true – バックアップファイル中のHubIDを使う。false – 新しくHubIDを作成する

このファイルの改行コードはLFのみを現在はサポートしています。

このファイルをSDカードにコピーして、SDカードを刺したままGravio Hubの電源をいれる。

SDカードにwificonfig.txtも含めて利用した場合、最初に、wificonfig.txtでネットワークを設定してのち、Gravio Hub設定が実行される。

バックアップファイルにzigbeeのペアリング設定も合わせてリストアされます。

11.5. Gravio Hub から Gravio Hub 2への移行ガイド

Gravio Hub（以下、Hub1）をご使用のお客様がGravio Hub 2（以下、Hub2）へ移行する手順について説明していきます。

事前準備

現在運用しているHub1のデータをバックアップして、Hub2にリストアして運用することを前提とします。

事前にHub1を起動し、Hub2もUpdateManagerを使用してネットワーク設定を行ってください。
Hub2はHub1と同一ネットワークに接続するため、移行手順を行う間はWiFiのアクセスポイントの設定か有線LANの接続やDHCP/静的アドレスにて仮IPアドレスを設定します。

Hub1とHub2でUpdateManagerを起動してログインした状態にして移行手順を進めてください。

移行手順

Hub2への移行手順は、以下の7点となります。

1. Coordinatorのバージョンを更新する
2. HubKitのバージョンを更新する
3. Zigbeeのバージョンを更新する
4. Hub1でデータをバックアップします
5. Hub1でバックアップファイルをダウンロードします
6. Hub2でリストアする
7. Hub2のネットワークを変更する

※ UpdateManagerの使用方法につきましては [こちら](#) を参照してください。

1. Coordinatorのバージョンを更新する

Hub1とHub2のCoordinatorソフトウェアのバージョンを同一にします。

Hub1でGravio設定タブにある、Gravio コーディネーターソフトウェアのアップデートから最新バージョンに更新してください。

Hub2でも同じくGravio設定タブにある、Gravio コーディネーターソフトウェアのアップデートから最新バージョンに更新してください。

2. HubKitのバージョンを更新する

Hub1とHub2のHubKitソフトウェアのバージョンを同一にします。

Hub1でGravio設定タブにある、Gravio HubKitソフトウェアのアップデートから最新バージョンに更新してください。
Hub2でも同じくGravio設定タブにある、Gravio HubKitソフトウェアのアップデートから最新バージョンに更新してく

ださい。

3. Zigbeeのバージョンを更新する

Hub1とHub2のZigbeeソフトウェアのバージョンを同一にします。

Hub1でGravio設定タブにある、Zigbeeソフトウェアのアップデートから最新バージョンに更新してください。
Hub2でも同じくGravio設定タブにある、Zigbeeソフトウェアのアップデートから最新バージョンに更新してください。

4. Hub1でデータをバックアップします

Hub1でGravio設定タブにある、HubKit / コーディネーターデータバックアップでバックアップボタンを押してデータをバックアップを行います。

バックアップボタンを押すとHub1で保存しているデータをダウンロード出来るように処理するあいだ少し時間がかかります。

もし処理中にエラーが発生した場合は、保存しているデータが多いのが原因の可能性がありますのでメディアファイルを削除するなど不必要なデータを削除してからもう一度試してください。

メディアファイルの保存先は [こちら](#) を参照してください。

5. Hub1でバックアップファイルをダウンロードします

Hub1でGravio設定タブにある、HubKit / コーディネーターデータバックアップでエクスポートボタンを押してバックアップファイルをローカルPCにダウンロードします。

6. Hub2でリストアする

Hub2でGravio設定タブにある、HubKit / コーディネーターデータバックアップでローカルPCに保存したバックアップファイルをインポートします。

Hub2にHub1のバックアップファイルをインポートすることでHub1のデータがHub2に移行されました。

7. Hub2のネットワークを変更する

Hub2をHub1とは別の新しいホスト名やIPアドレスとして運用する場合にはここまでで移行は終わりです。

ネットワーク設定情報はバックアップファイルをインポートしても引き継がれませんので、Hub1と同じホスト名やIPアドレスを使用したい場合は、Hub1を停止し、Hub2のネットワーク設定で変更してください。

12. Gravio Hub (Gravio Hub version 1)

Gravio Standard 以上のサービスメニューをご契約の方は、Gravio Hubをレンタルすることが可能です。

Gravio Hub version 1は四角い筐体の機種です。



Gravio Hub version 1をお使いの方はGravio Hub 2機材と交換できますので、お問合せください。

12.1. Gravio Hub(Gravio Hub version 1)の初期設定

ここでは、Gravio Hubの初期設定に関して説明していきます。

Gravio Hubを利用するために、必要なことは以下の2点です。

1. Gravio Hubをネットワークに接続するために [WiFiネットワークの設定](#) を行う。
2. Gravio HubはUbuntu1804ベースのLinuxOSで動作しています。必要に応じてLinuxOSへログインできる設計になっています。このため、Linuxにログインするためのログインアカウントのパスワードをセキュリティを高めるために利用時の初期設定で、[パスワードの変更](#) をお願いしております。

事前にWiFiネットワークの設定（接続するネットワーク名とパスワード）とログインアカウントの新しく設定するパスワードを準備してください。

注）再設定したログインアカウントのパスワードは大切に保存してください。もしパスワードを紛失した場合にはLinuxにログインできなくなります。

これらの設定を行うために、iOSもしくはAndroid端末から設定ができる機能、およびアプリを用意しています。以下の手順に従って、設定を行ってください。

Gravio Hubには以下のGravio Hubを含めて以下の付属品がレンタルされています。

- Gravio Hub 本体



- Gravio USB-Cアダプター



- Gravio USB-Cケーブル



- 電源ボックス



Gravio Hubには1つのUSB-Cコネクタがあり、そこにUSB-CアダプターもしくはUSB-Cケーブルを接続して利用します。

USB-Cケーブルを電源ケーブルのみを繋ぐ場合

以下のように接続して、電源ボックスを電源につないでください。



Gravio HubをLAN内に設置する

Gravio Hubを電源につなげて起動します。

12.2. Gravio Hub (Gravio Hub version 1)のインジケータランプの状態とボタン操作

インジケータランプの状態

Gravio HubのGravioロゴは、さまざまな色やパターンで点灯し、下記の内容を確認することが出来ます。

LED	状態	コメント
青色点灯	箱から出した最初の起動時	標準のアクセスポイントモード
白色点灯	Wifiセットアップ後または有線LAN接続時	通常運用モード
緑色点滅	Docker 初期設定時	このプロセスには約2分かかります
5秒間緑色点滅	Docker 設定完了時	
5秒間赤色点滅	Docker 初期設定エラー時	
青色	アクセスポイントモードに戻る	ボタンを3秒間押し続けるとこの状態になります
黄色	電源停止準備	ボタンを5-6秒間押し続けるとGravio Hubは停止準備になります
赤色	電源停止	ボタンを7秒以上押し続けるとGravio Hubは電源を停止します

ボタン操作

Wifi情報を初期化してアクセスポイントモードに戻る

ボタンを3秒間押し続けるとWifi情報を初期化してアクセスポイントモードに戻る状態になります

電源停止準備

ボタンを5-6秒間押し続けるとGravio Hubは停止準備になります

電源停止

ボタンを7秒以上押し続けるとGravio Hubは電源を停止します

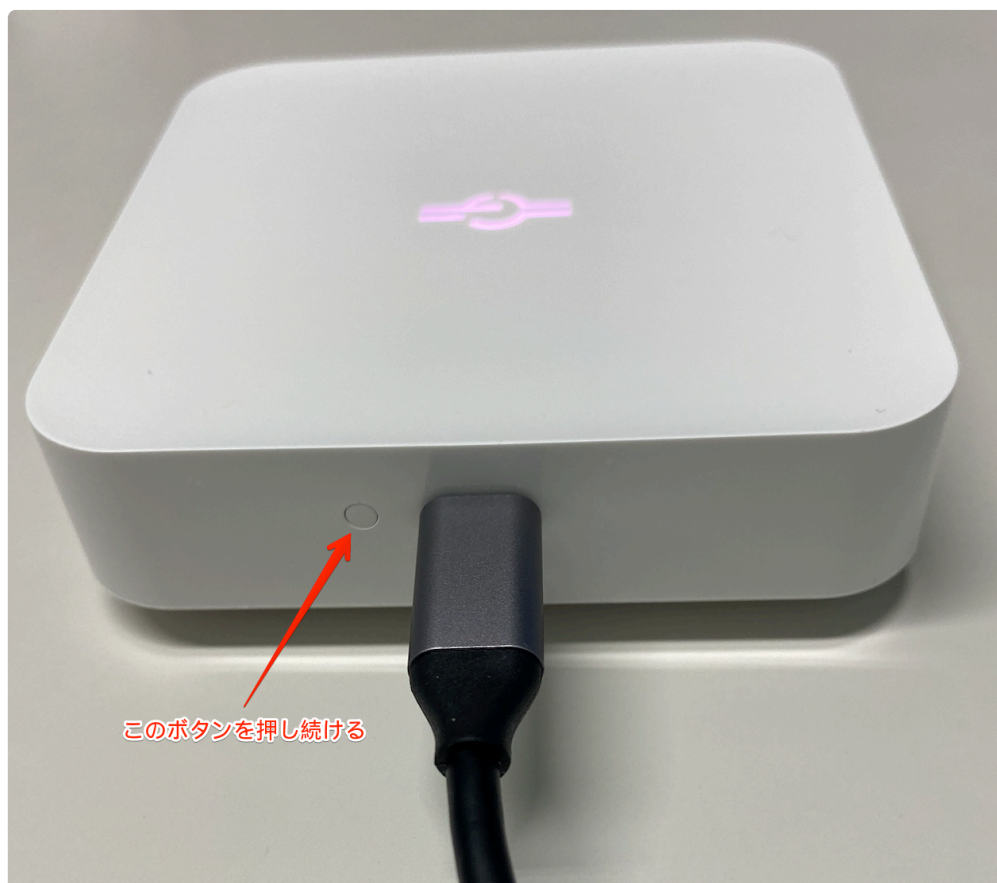
12.3. Gravio Hubの停止方法

Gravio Hub (version 1)での停止方法

Gravio Hubを停止させる場合は、必ず、以下の手順で停止してください。

以下の示すボタンを長押しすると、上面のGravioマークが「青」→「黄色」→「赤」の順に変化して、最後にライトが消灯します。

これでGravio Hubの電源が切れます。



13. トラブルシューティング

このセクションでは、Gravioでのプログラミングなどの開発時に発生しやすい問題を列挙して、修正する方法について説明します。

13.1. 一般的な課題

症状	説明	対処法	コメント
Gravio Studioが期待どおりに動作しない	HubKitの接続が不安定であるか、HubKitが動作していません	Gravio StudioとHubKitの間に安定した接続が確立されていることを確認してください。	
一部のセンサーは情報を送信していません	範囲外であるか、電源が切れている	センサーの太陽電池バッテリーが不足しているをしている場合、それらを太陽や他の自然光に数時間入れて、問題が解決するかどうか確認してください。また、ドングルレシーバーの近くに移動します。	
Gravio StudioがGravio HubKitに正しく接続していません	関連するネットワークポートが開いていない可能性があります	「 Gravioが利用するネットワークポート 」のポート要件を確認してください：	
Gravio Studioが正しく接続されていません	サーバーが停止している可能性があります	GravioTools (Windows)、メニューバーアプリケーション (macOS)、Linuxの場合は“docker-compose down”を実行後、“docker-compose up -d”のコマンドを使用してサーバーを再起動できます。	

13.2. センサー関係のトラブルシューティング

症状	説明	対処法	コメント
太陽電池パネル型デバイスドア/接触センサーが反応しない	センサーのバッテリーが切れている	センサーを直射日光に数時間あててください。また、磁石をセンサーの隣に置くと、 dongle でライトが点滅してセンサーが dongle に接続されているかどうかを確認することができます。	

13.3. Gravio StudioからHubKitが見つからないまたは繋がらない

Gravio StudioからHubKitが見つからない場合

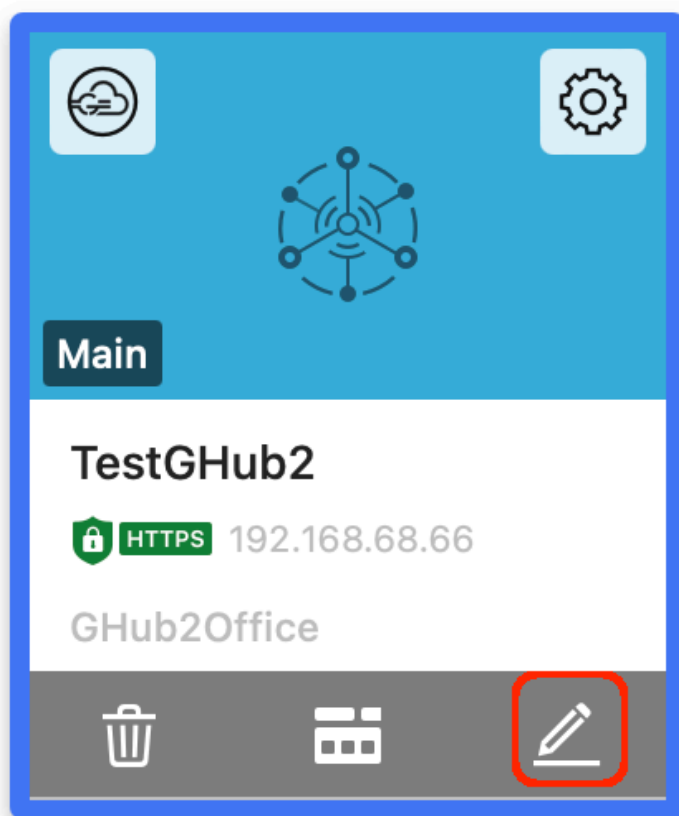
Gravio Hubの場合

1. HubKitのIPアドレスをHubKitのコンソールでログインして、ifconfigを利用して無線の場合はwlan0のIP Addressを確認する
2. Gravio StudioのPCからpingを実行して当該Gravio Hubに繋がるか確認する
3. 繋がる場合はGravio StudioのHubリストの画面「+」のアイコンを押して、そのIPアドレスでHubの追加を試みる。

Windowsの場合

1. Gravio Studioのインストールより先にHubKitを入れている場合、同一マシン内のGravio StudioからHubKitは見えません。(Windows OS のSandbox制限のため。) この場合は、一度HubKitをアンインストールして、Gravio Studioをインストールした後に、HubKitのインストールをする
2. 上記でない場合、当該PCのIPアドレスを確認し、そのIPアドレスで、Gravio StudioのHubリストの画面「+」のアイコンを押して、そのIPアドレスでHubの追加を試みる。

Gravio StudioでHubリストに出ているHubをクリックしても接続エラーになる



Gravio Hubの場合

1. HubKitのIPアドレスをHubKitのコンソールでログインして、ifconfigを利用して無線の場合はwlan0のIP Addressを確認する
2. Gravio StudioのPCからpingを実行して当該Gravio Hubに繋がるか確認する
3. Gravio StudioのHubKitのカードで編集ボタンを押して、“Hub IP”のフィールドにIPアドレスの直値を入力して保存して、再度カードをクリックしてみる。

Windowsの場合

1. 当該PCのIPアドレスを確認し、そのIPアドレスで、Gravio StudioのHubKitのカードで編集ボタンを押して、“Hub IP”のフィールドにIPアドレスの直値を入力して保存して、再度カードをクリックしてみる。

14. 補足

14.1. Gravioが利用するネットワークポート

GravioのHubKitの各サーバーモジュール、Gravioコンフィグレーションマネージャー、Coordinatorは以下のポートを利用しています。

・インバウンド (Gravio 5.1)

サービス名/プロセス名	プロトコル	ポート
Gravioコンフィグレーションマネージャー	HTTP	8080
Coordinator / Gravio Control Manager	HTTPS	443
Gravio App Service	HTTPS	29443
IVAR, i-PRO用エンドポイント	HTTPS	29448

HubKitとChrome castを使用する場合は、これらのポート番号うち、「Gravio App Service –HTTPS:29443」をHubKitのインストールしているPCでポートを開ける必要があります。

・インバウンド (Gravio 5.0)

サービス名/プロセス名	プロトコル	ポート
Gravioコンフィグレーションマネージャー	HTTP	8080
Coordinator	HTTPS	443
Gravio Control Manager	HTTPS	29442
Gravio App Service	HTTPS	29443
IVAR, i-PRO用エンドポイント	HTTPS	29448

HubKitとGravio Studioを別のPCにインストールする場合は、これらのポート番号うち、「Gravio Control Manager –HTTPS:29442」をファイアウォールでポートを開ける必要があります。

HubKitとChrome castを使用する場合は、これらのポート番号うち、「Gravio App Service –HTTPS:29443」をHubKitのインストールしているPCでポートを開ける必要があります。

・アウトバウンド (Gravio 5.1/5.0共通)

サービス名	プロトコル	ポート
DNS	UDP/TCP	53
NTP	UDP	123
HTTPSサーバ	HTTPS	443

14.2. HubKitのデータディレクトリ

Hubはインストールされると設定ファイルの保存とデータを保存するためにデータディレクトリを作成します。

Windows10にインストールした場合

C:¥ProgramData¥HubKit

macOSにインストールした場合

/Library/Application Support/HubKit/

Linux/Raspberryにインストールした場合

Linux/RaspberryではDockerで動作しているため、以下のディレクトリがボリュームとしてマウントされています。

/home/gravio/hubkitrepo4/data

14.3. カメラ画像の保存先

ONVIFなどでカメラから取得した画像は以下のディレクトリにカメラデバイス別にディレクトリが作成され、日付ディレクトリごとに保存されます。

The screenshot shows a web interface for configuring a camera. On the left, a sidebar lists device types: Zigbee, Serial, Bluetooth, and Camera. The 'Camera' section is active, displaying a list of devices. The 'Sony' device is selected. The main configuration area for the 'Sony' device (ID: Camera192.168.16.5) includes fields for '表示名' (Display Name) set to 'Sony', 'キャプチャー形式' (Capture Format) set to '画像' (Image), and 'キャプチャー間隔(秒)' (Capture Interval) set to '1'. Two checkboxes are present: '画像の保存' (Save Image), which is checked and indicated by a red arrow, and '検出フレームの描画' (Extract Frame Drawing), which is unchecked. Below these are fields for 'ユーザー名' (Username) and 'パスワード' (Password). At the bottom right are '保存' (Save) and '閉じる' (Close) buttons.

Windows10

C:\ProgramData\HubKit\mediadata

macOS

/Library/Application Support/HubKit/mediadata

Linux/Raspberry

Linux/RaspberryではDockerで動作しているため、Host側のディレクトリで見た場合は、以下のディレクトリがボリュームとしてマウントされています。

/home/gravio/hubkitrepo4/data/mediadata

Docker内部で見た時のディレクトリは、

/var/opt/hubkit/mediadata

のディレクトリとなります。

14.4. Gravio StudioからHubへのアクセス制限をかけたい場合

HubにアクセスするGravio Studioを制限する

Gravio 5.0までの機能です。Gravio 5.1ではログインによる認証を行うためクライアント認証の機能はなくなりました。

HubにアクセスするGravio Studioを制限するには、Hubに使用するGravio Studioのクライアント証明書をインストールします。

*この機能はGravio Basic, Gravio Standard, Gravio Enterprise契約のみで使用できます。

Gravio StudioでGravio Cloudにログインしてからメニューの「クライアント証明書」を選択します。



「以下のキーで証明書作成」の右にキーを入力して、「証明書を生成」ボタンを押して証明書を作成します。



「証明書をダウンロード」を選択して、証明書をPCにダウンロードします。ダウンロードした証明書ファイルはGravio Hubのデータディレクトリ(`gravio/clientcert`)にコピーしてください。

MacOSの場合

以下のディレクトリにダウンロードした証明書を入れる。
`/Library/Application\ Support/HubKit/clientcert`

Gravio HubKitのToolbarアプリで、「設定…」メニューを開き、「接続」タブを選択して、「クライアント認証を利用する」をチェックする。
その後、HubKitを起動する。

Gravio Hub / Linuxの場合

SSHでloginし、以下のディレクトリにダウンロードした証明書を入れる。
`/home/gravio/hubkitrepo4/etc/clientcert`

「`/home/gravio/hubkitrepo4/etc/ctrlmgr/conf`」のディレクトリに「`config.json`」というファイル名で以下の内容を含んだファイルを作成する。

```
{
  "Client" : {
    "UseClientCertificates" : true
  }
}
```

その後、Gravio Hubを再起動する

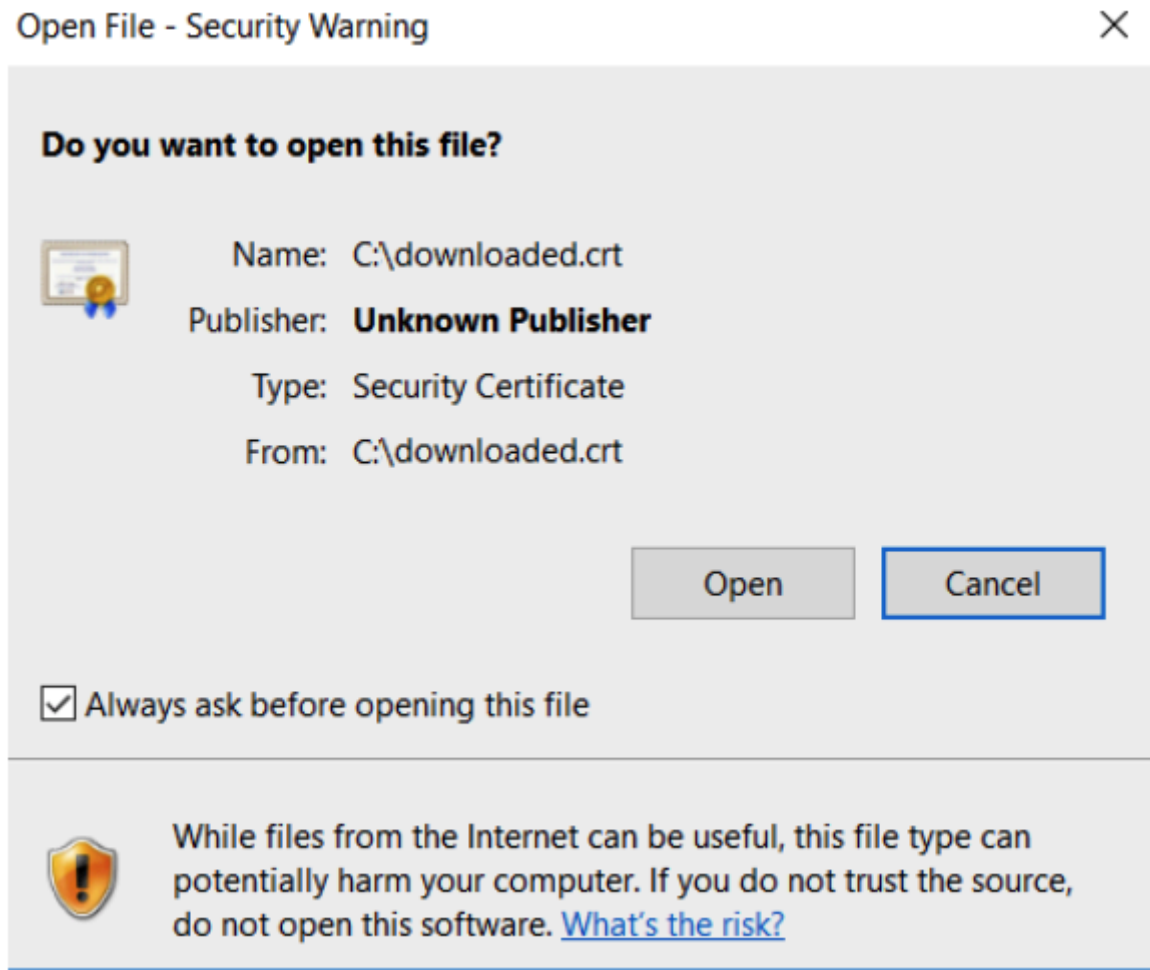
証明書のHubへの登録した後はGravio HubKitを再起動してください。再起動によりクライアント認証が有効となり

ます。

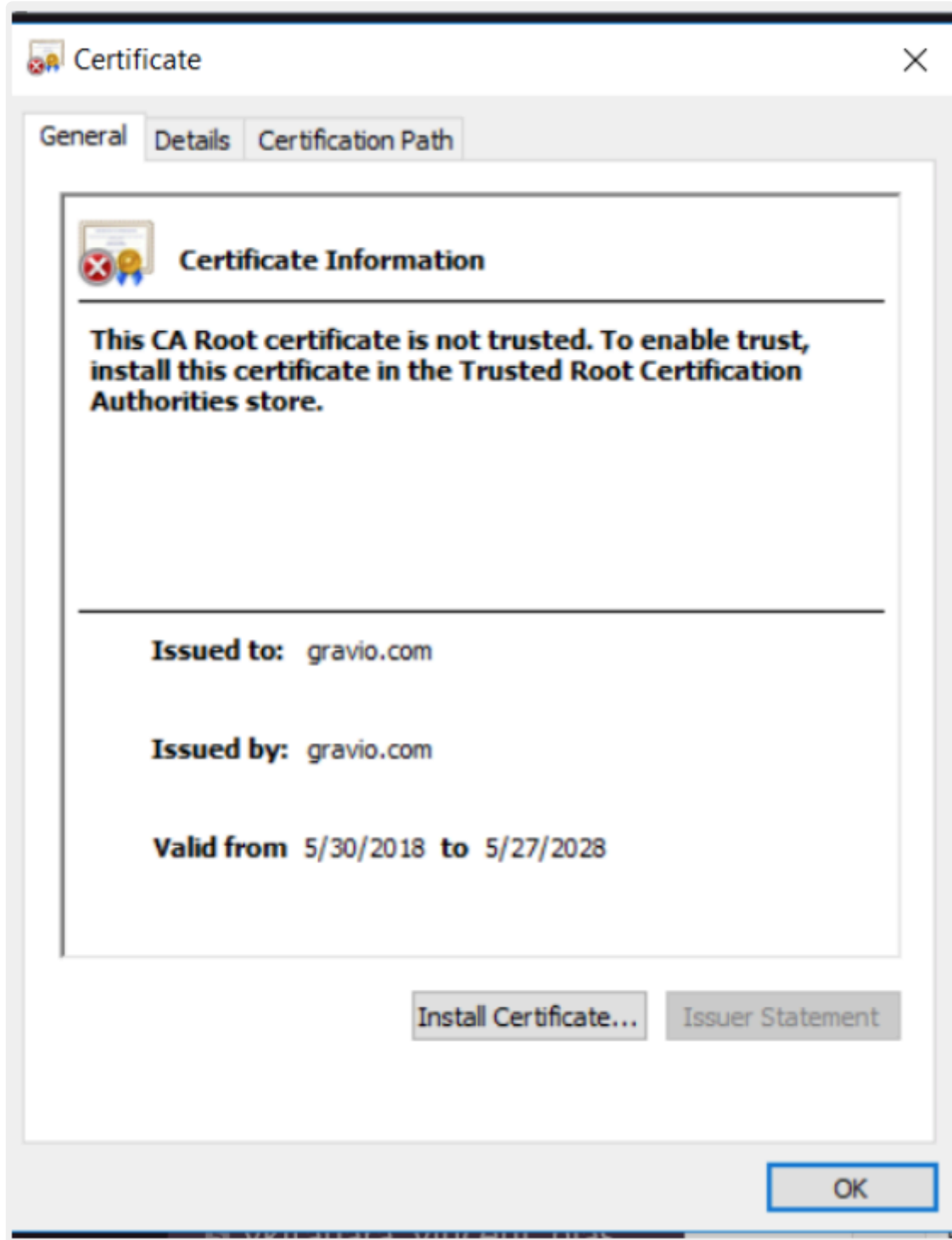
14.5. HubKitにルートCA証明書をインストールする方法

Windows 10

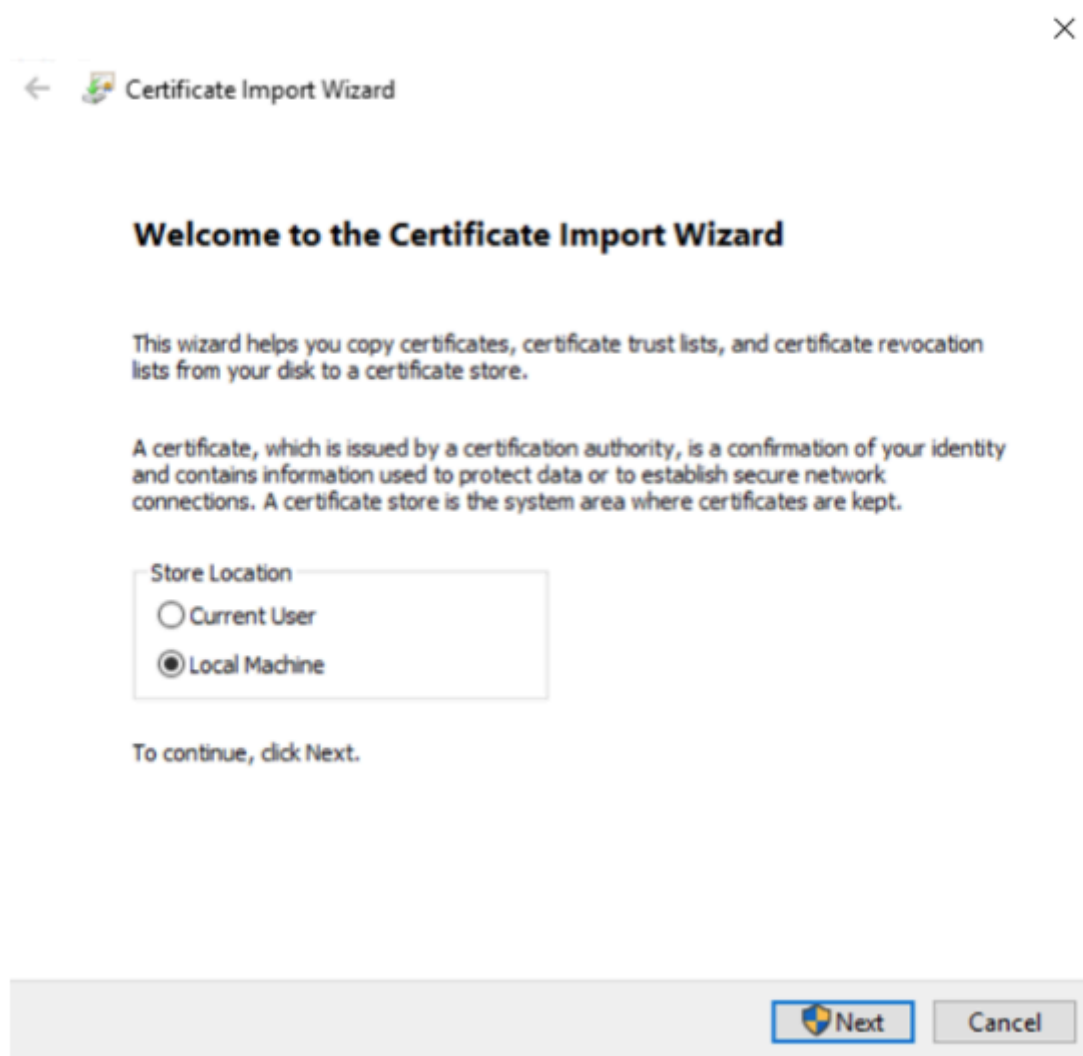
エクスプローラから証明書ファイルをダブルクリックします。



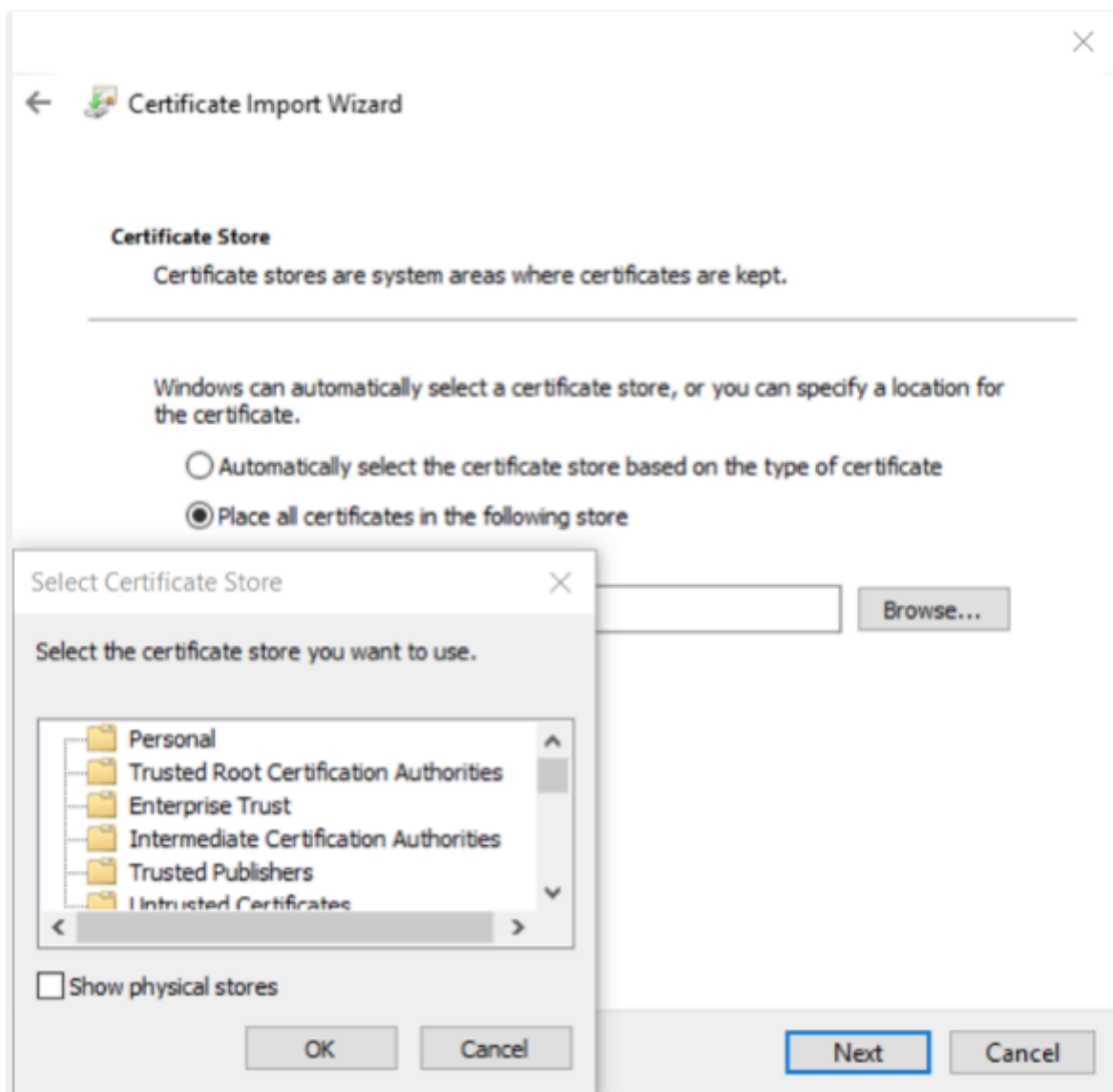
証明書ウィンドウで、[証明書のインストール]をクリックします。



インポートウィザードで、[ローカルマシン]を選択し、[次へ]をクリックします。

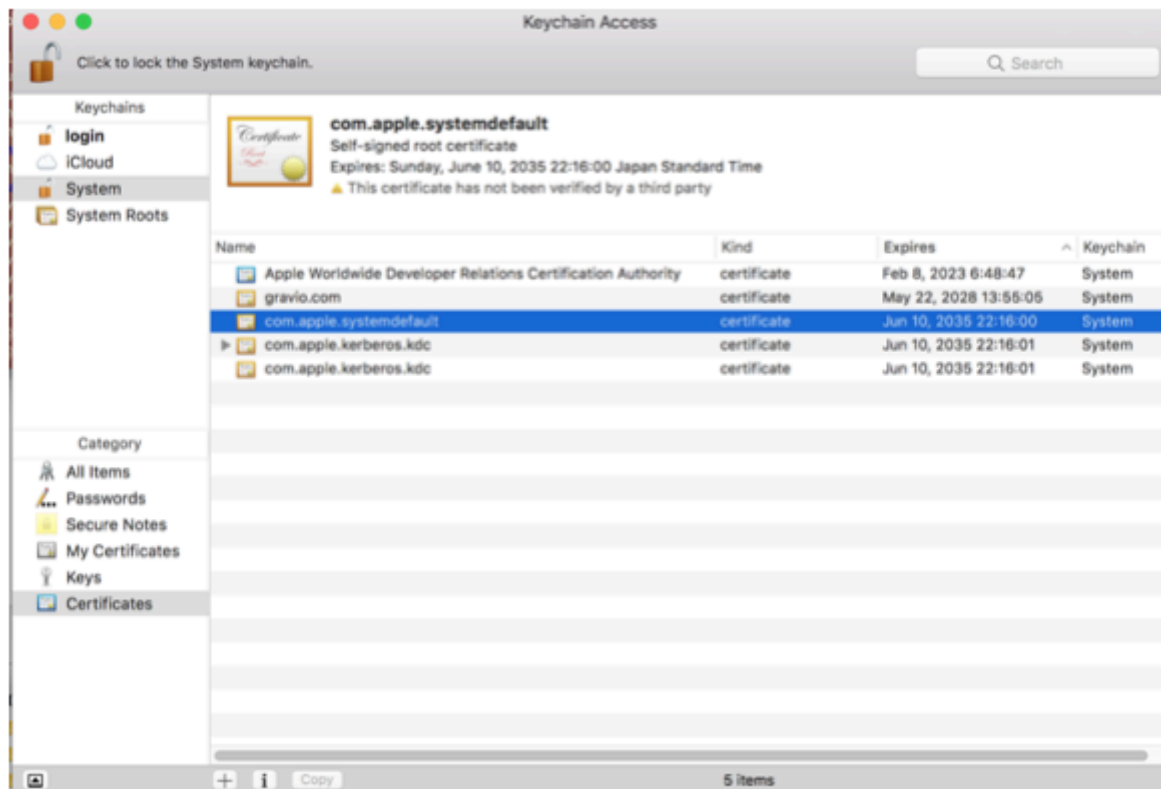


次の画面で、[すべての証明書を次のストアに配置する]ラジオボタンを選択し、[信頼されたルート証明機関]を選択します。「次へ」を押して、「終了」

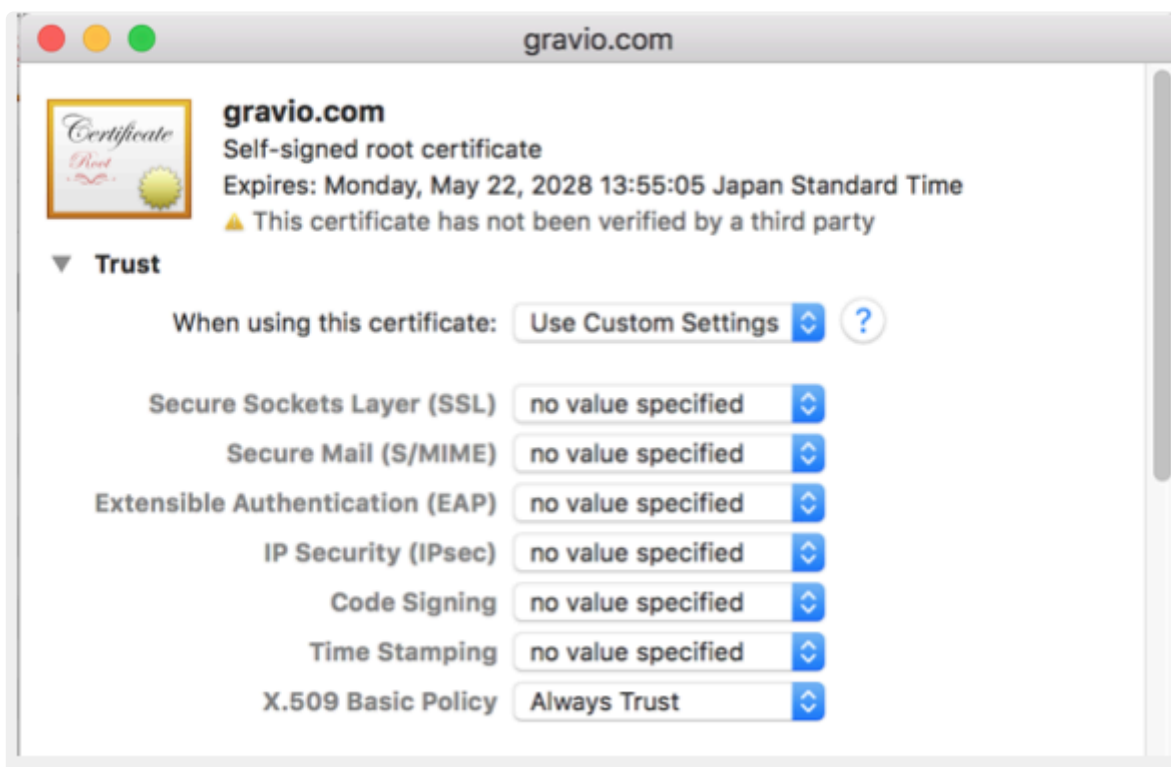


macOS

Spotlight Searchを使用して「キーチェーンアクセス」を開きます。「システム」キーチェーンカテゴリを選択し、Finderまたはデスクトップからキーをドラッグアンドドロップします。パスワードの入力が求められます。



インストールされた証明書 (gravio.com) を右クリックし、「情報を取得」を選択します。「Trust」の横にある下矢印をクリックし、X.509基本ポリシーの横にあるドロップダウンで「Always Trust」を選択します。ウィンドウを閉じると、パスワードの入力が求められます。証明書がインストールされています。



Linux (Ubuntu)

1. “sudo apt-get install -y ca-certificates”を使ってca-certificatesプログラムをインストールします。

2. Gravio証明書ファイルを “/usr/local/share/ca-certificates”フォルダに移動します。
3. “sudo update-ca-certificates”を使って証明書を更新します。（注：証明書を確認するには、Gravioサービスを再起動する必要があります）

14.6. Gravio Studio for Windowsのサイドローディングインストール

Gravio Studio for Windowsのサイドローディングの手順

サイドローディングとは、アプリをStoreからではなくインストーラからWindowsにインストールする方法です。

※サイドローディングでインストールする場合には、既にインストールされているGravio Studio for Windowsは先にアンインストールしてください。

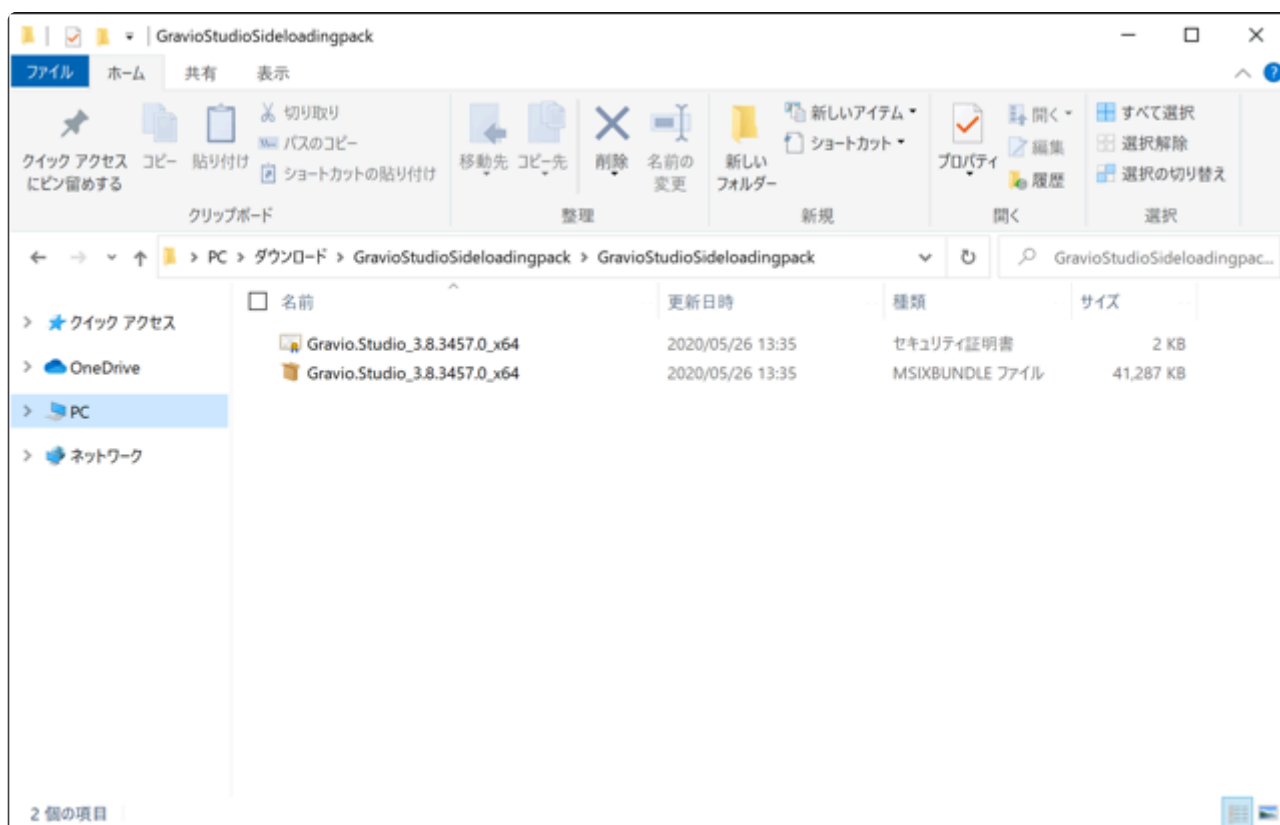
1. zipファイルのダウンロード

ダウンロードは [ここから](#) 行ってください。

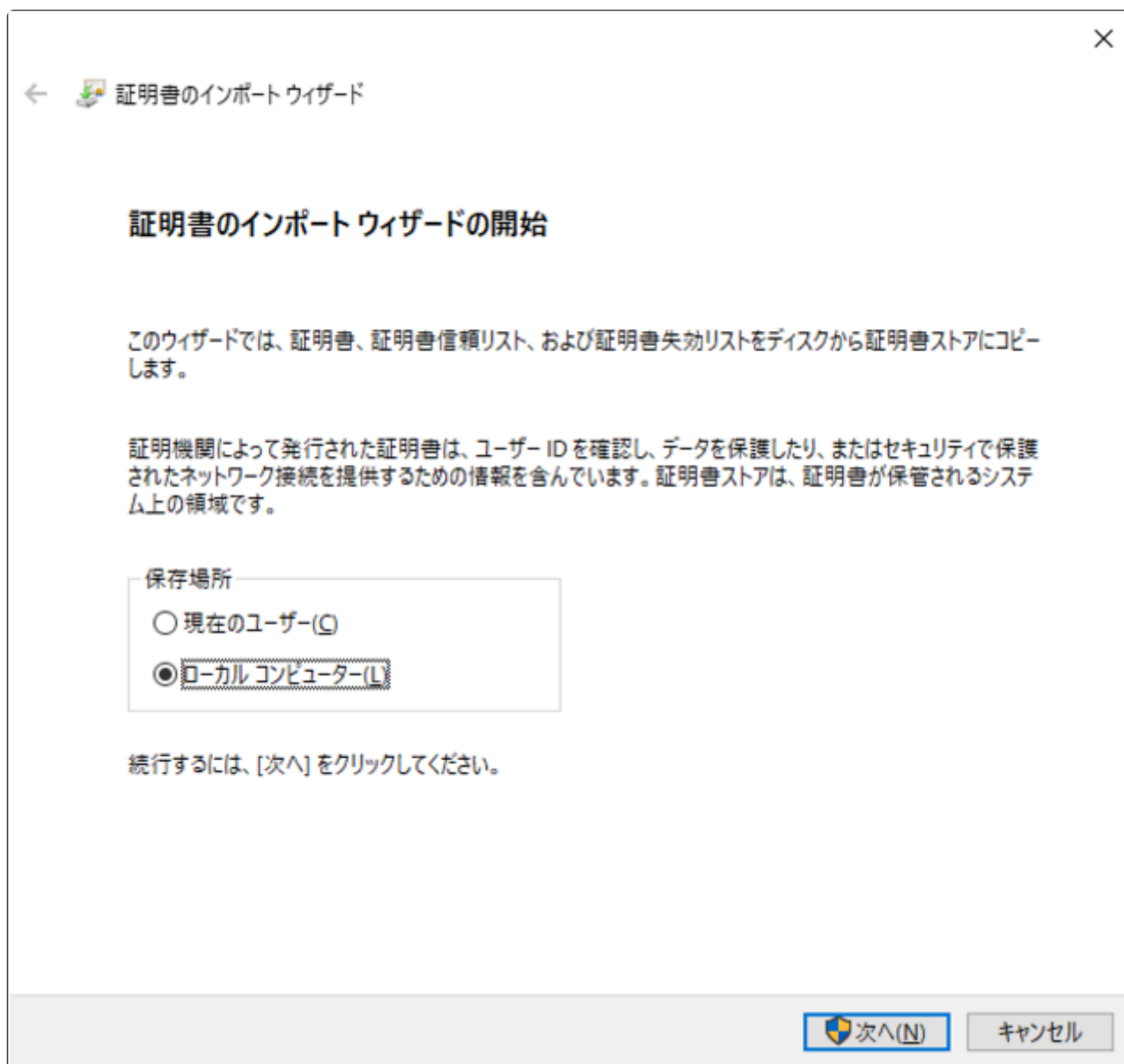
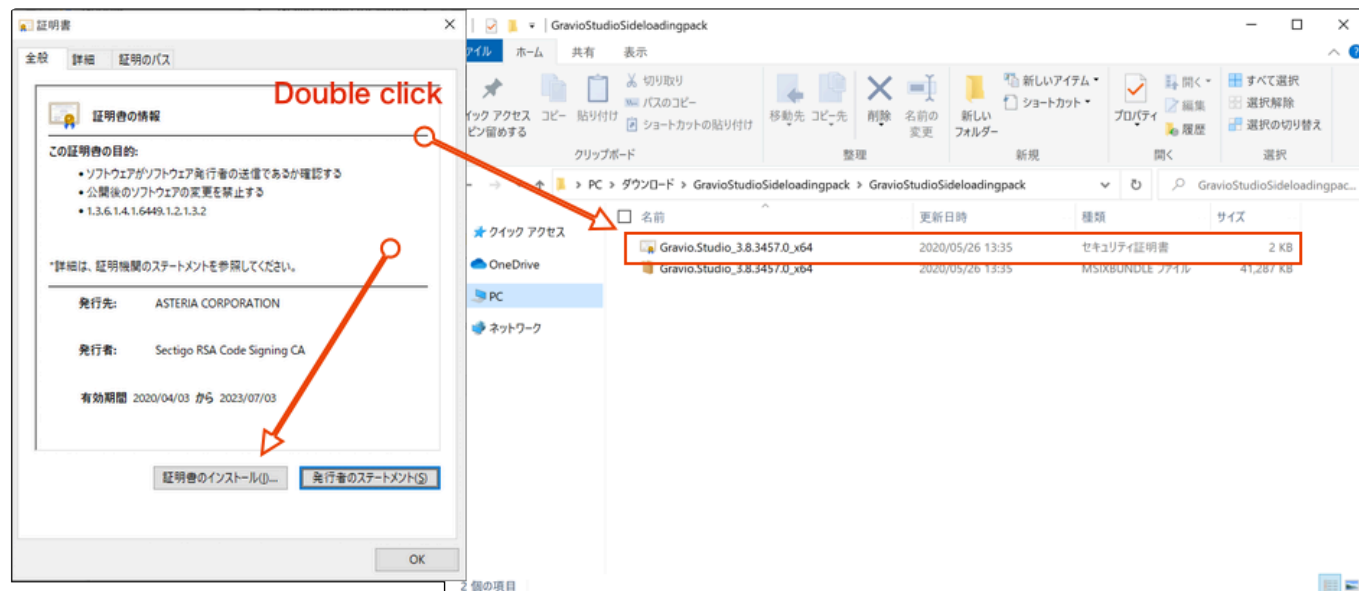
zipファイルを展開してください。

2. 証明書のインストール

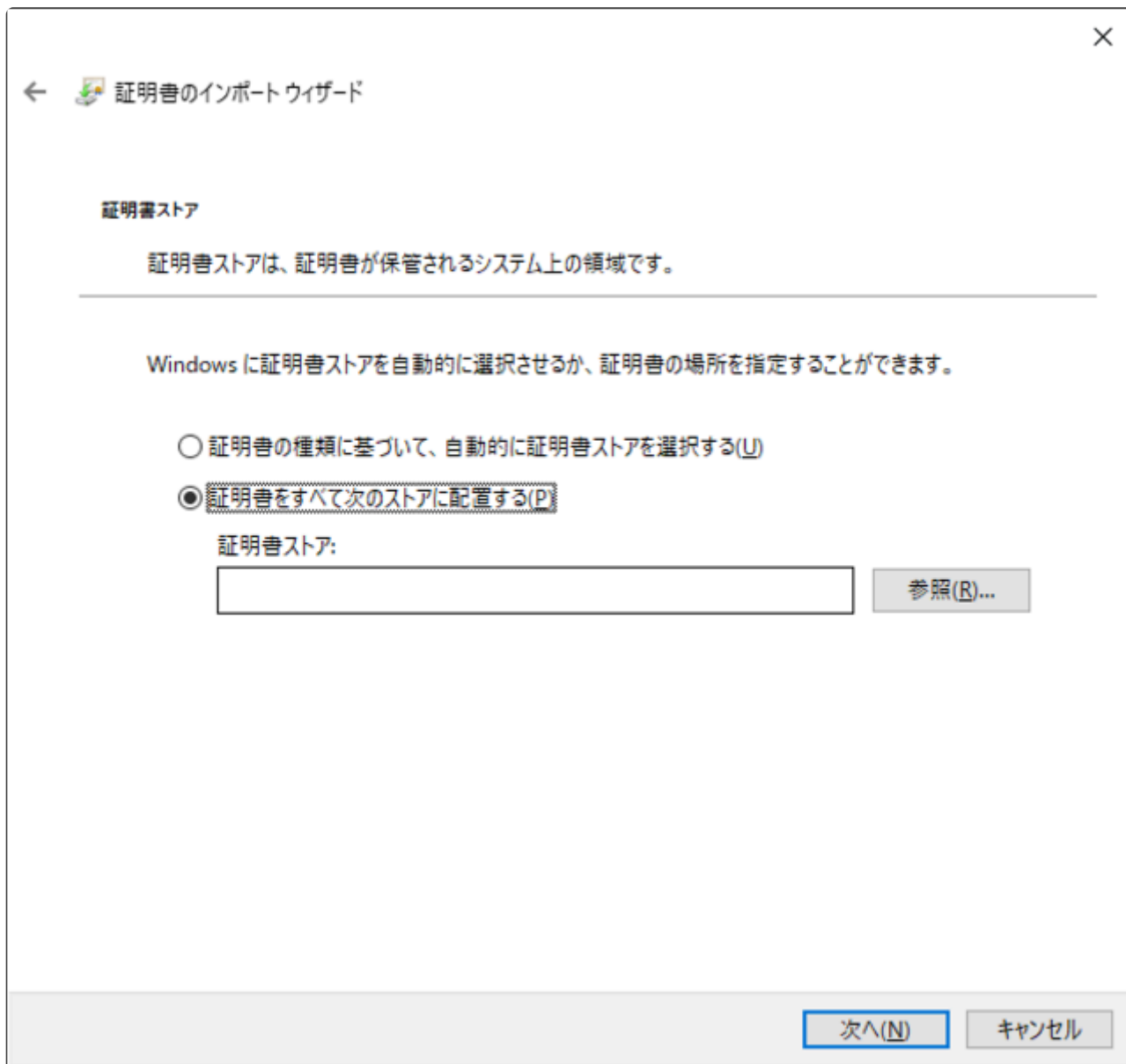
画面にある証明書を選択してダブルクリックします。



証明書のインストールをクリック

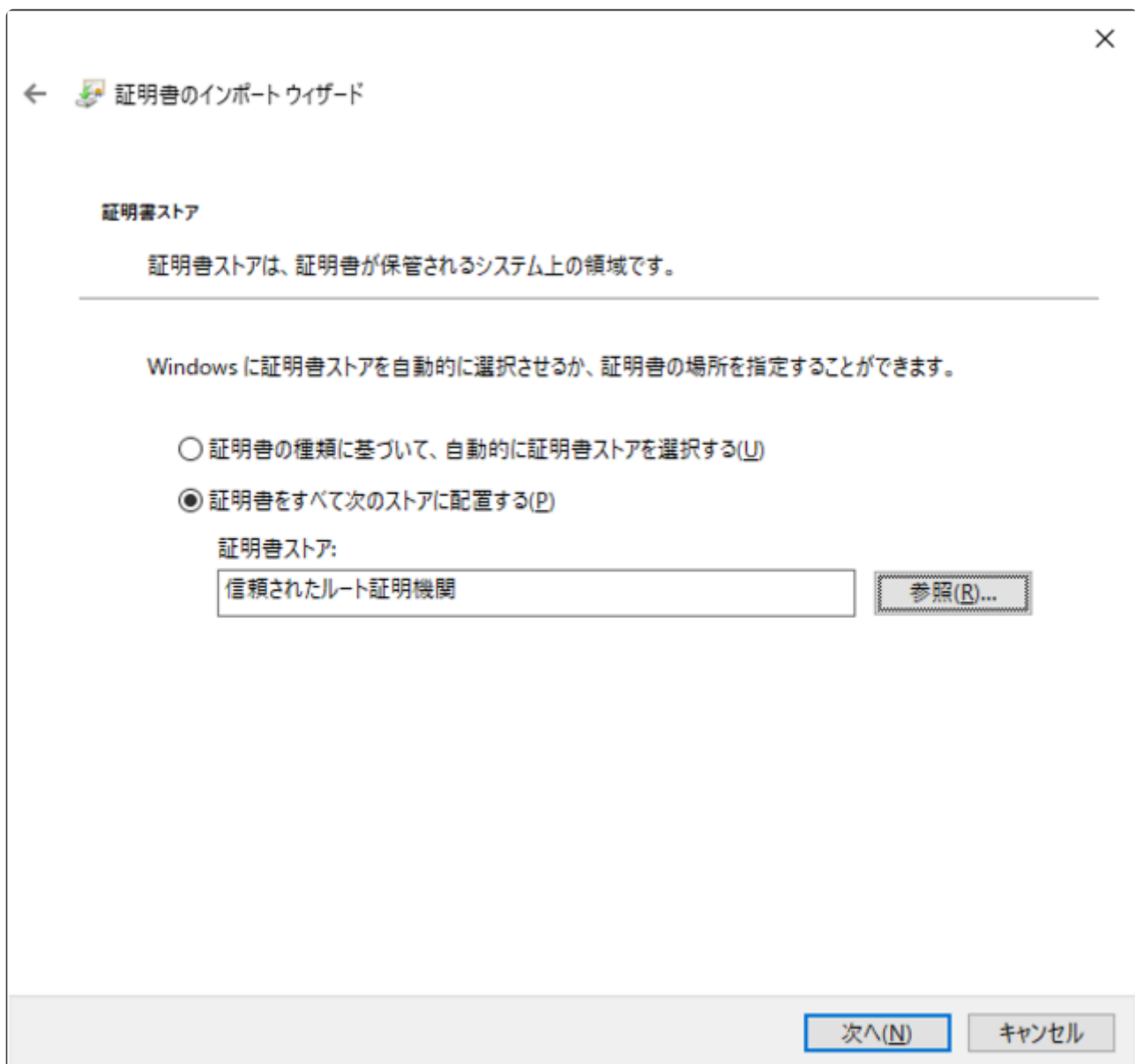
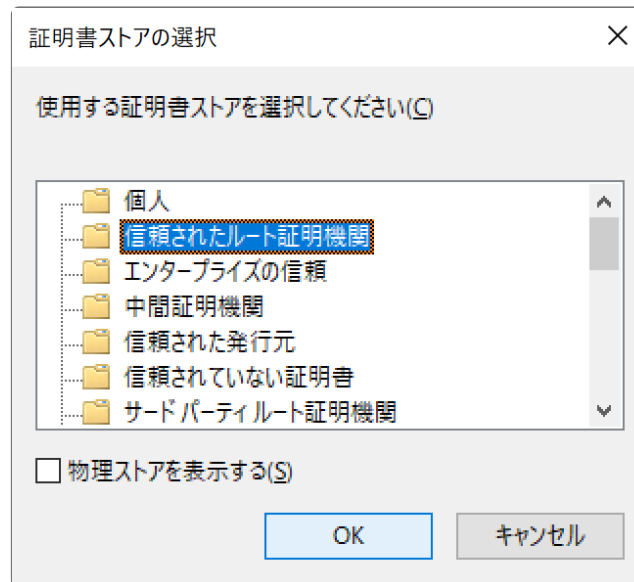


「ローカルコンピューター」を選択して「次へ」

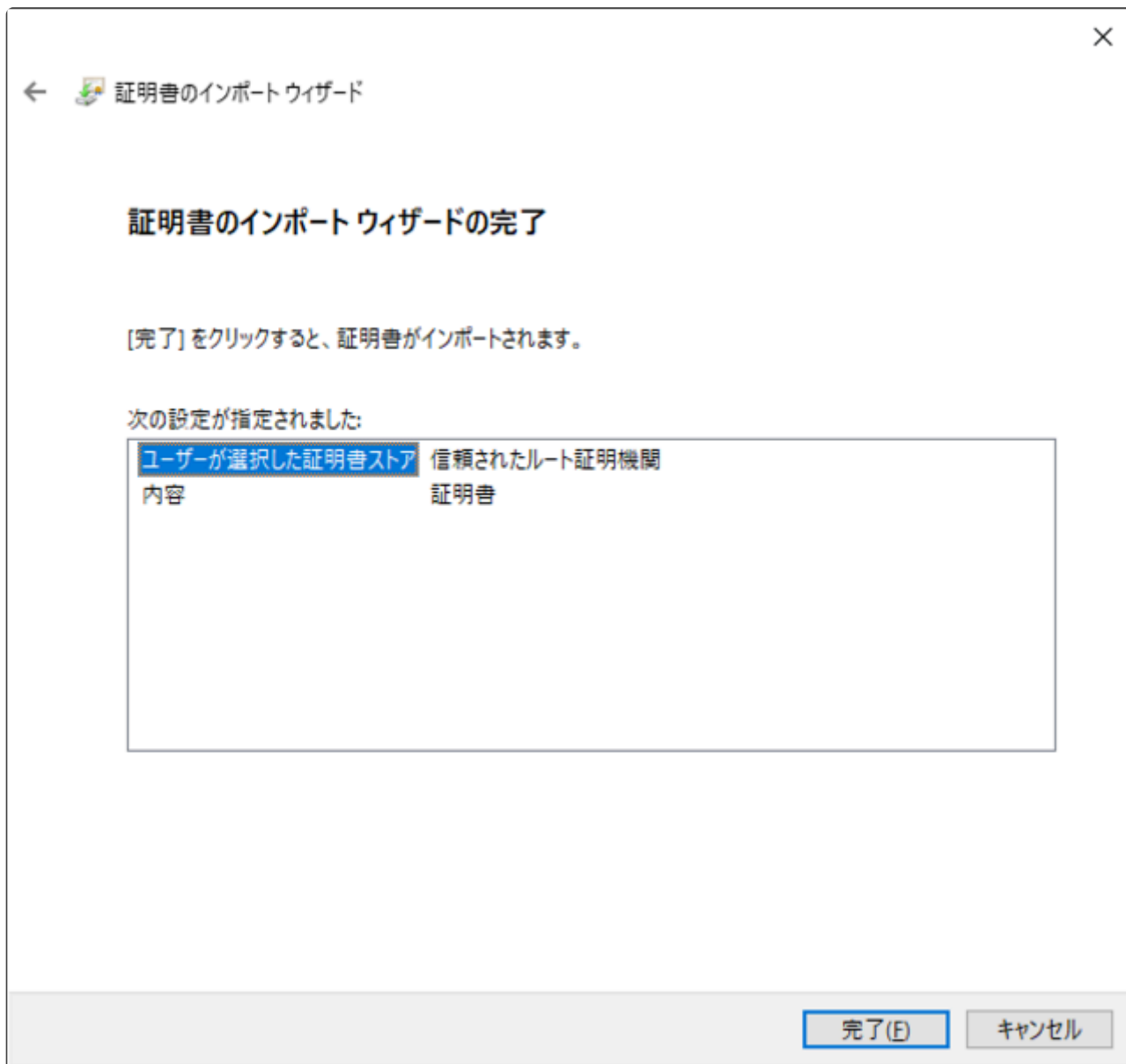


「証明書をすべて次のストアに配置する」を選択して、「参照」ボタンを押す。

「信頼されたルート証明機関」を選択します。



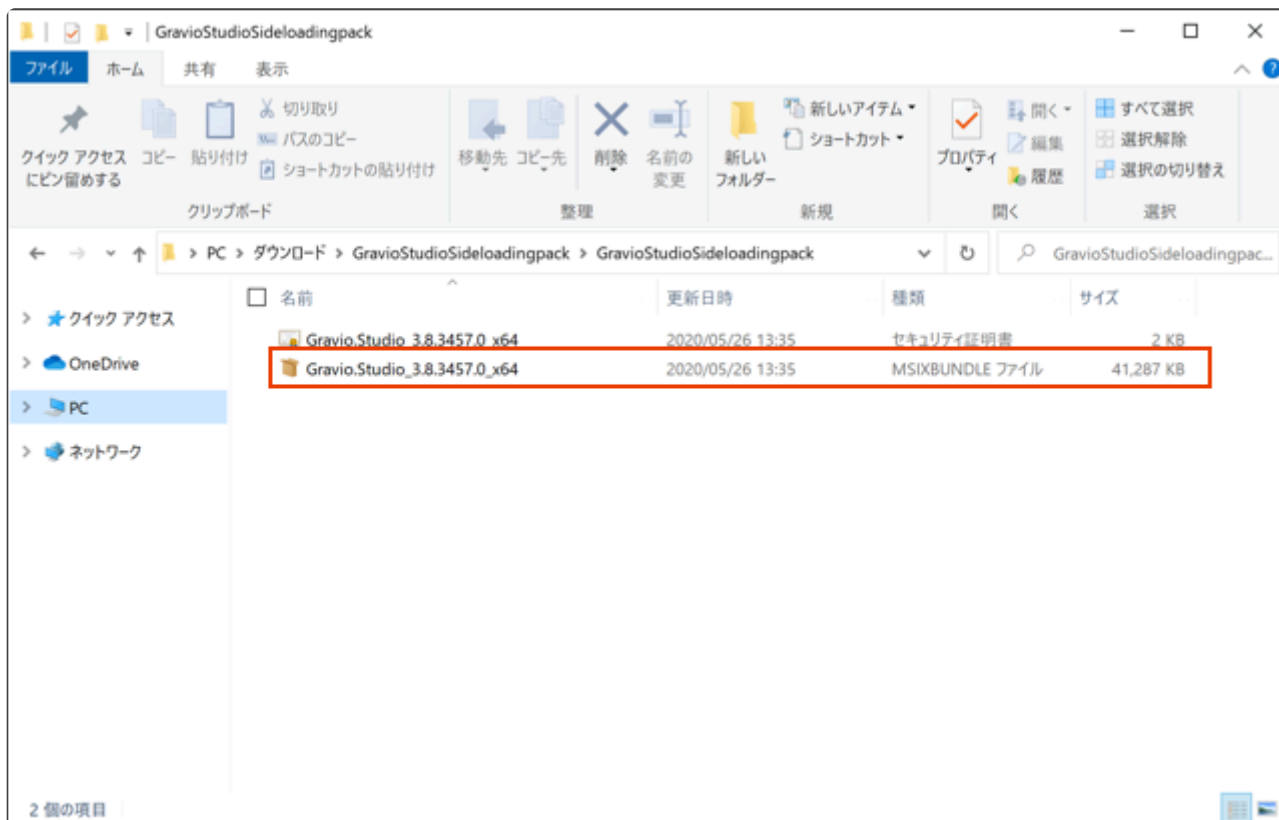
「次へ」を押して、証明書をインポートします。



これで証明書のインストールは完了です。

3. Gravio Studioのインストール

インストールパッケージを選択してダブルクリックします。



「インストール」ボタンを押してインストールします。



以上でサイドローディングでのインストールは完了です。

Windows 10 2004 (19041)、20H2 (19042)、21H1 (19043)、21H2 (19044) を使用して

いる場合でインストールのエラーが発生する場合の対処方法

インストールでエラーが発生する場合には、以下の手順をご確認ください。

(1) ダウンロードしたZip内の以下のディレクトリにある5つのファイルを上から順にダブルクリックしてインストールできるか確認してください。

場所: graviostudio_sideloading\Dependencies\x64

- ・Microsoft.NET.Native.Framework.2.2.appx
- ・Microsoft.NET.Native.Runtime.2.2.appx
- ・Microsoft.UI.Xaml.2.8.appx
- ・Microsoft.VCLibs.x64.14.00.appx
- ・Microsoft.VCLibs.x64.14.00.Desktop.appx

(2) 5つともインストールができましたら、再度、Gravio Studioのインスールを試してください。

14.7. Zigbee firmware のアップデート方法

Zigbeeドングルのファームウェアアップデート方法に示します。

1. Macを使用する場合

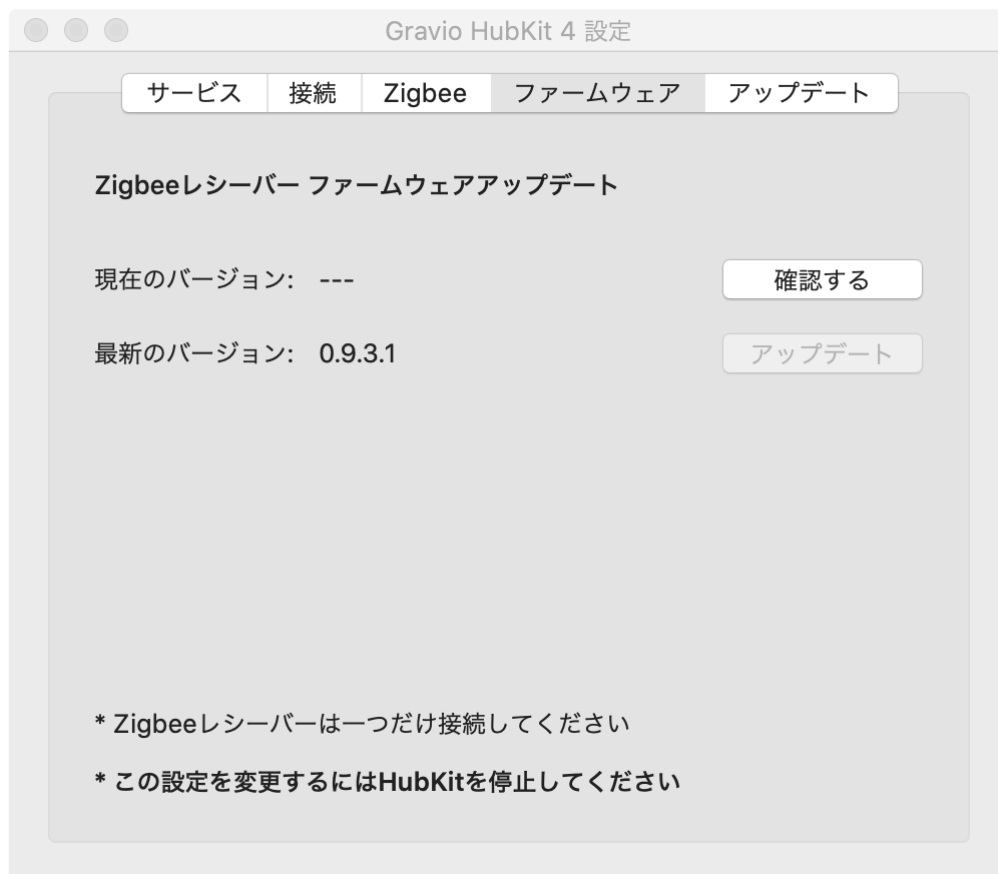
Gravio HubKit4がインストールされているMacを用意してください。

※HubKitが実行されている場合には停止してください。

ZigbeeドングルをMacのUSBポートに差し込んでください。
ドングルはGravioのロゴを上にして差し込みます。

macOSの上段のメニューバーのHubKitアイコンから「設定」メニューを選択します。

設定パネルから「ファームウェア」タブを選択します。



「確認」ボタンを押してファームウェアの現在のバージョンを確認します。

現在のバージョンが最新のバージョンより古い場合に「アップデート」ボタンを押してファームウェアを更新します。

アップデートが終了しましたらファームウェアが更新されますので使用してください。また既にペアリングしているセンサーはそのまま使用することが出来ます(ペアリングは維持されています)。

2. Windowsを使用する場合

Gravio HubKit4がインストールされているWindowsを用意してください。

※HubKitが実行されている場合には停止してください。

Zigbee dongleをWindowsのUSBポートに差し込んでください。

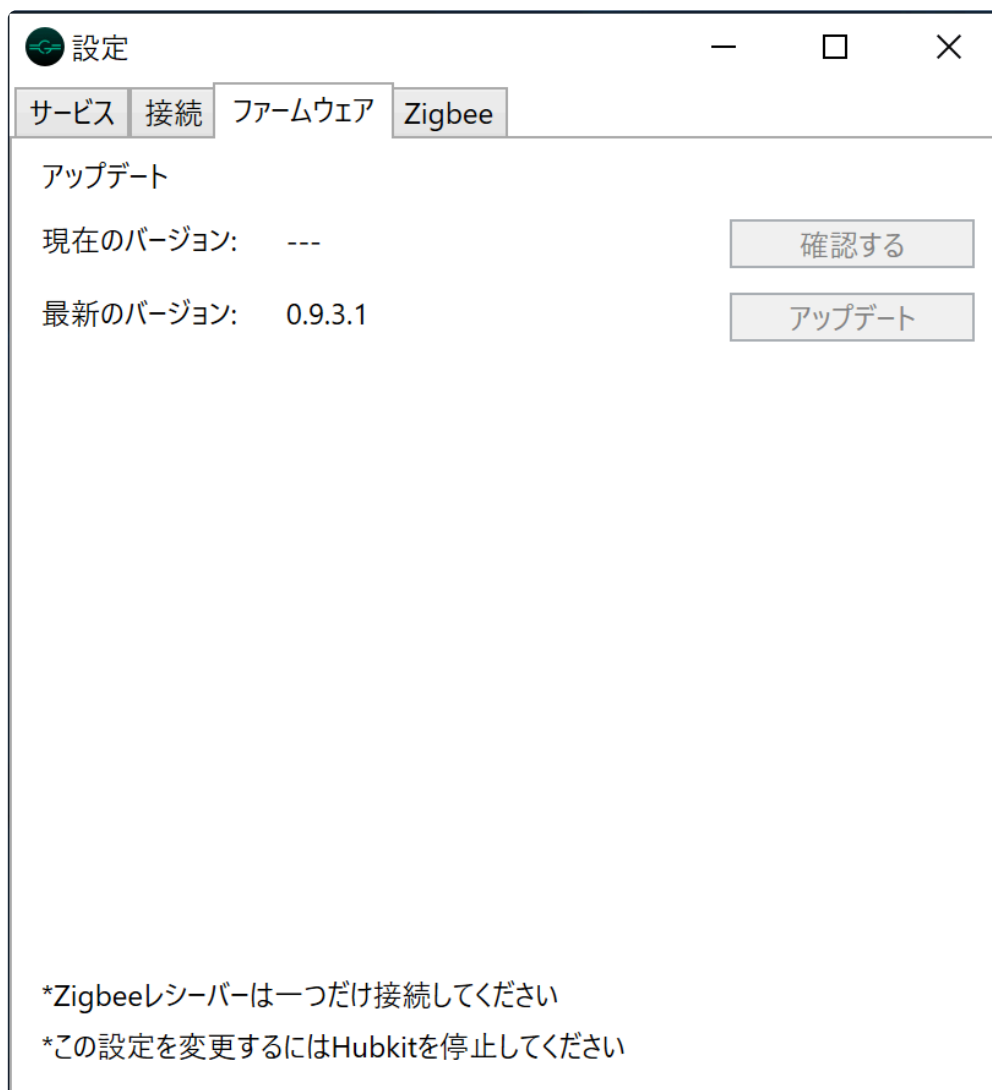
dongleはGravioのロゴを上にして差し込みます。

Windows10のスタートメニューにある「GravioHubKit」メニューの「GravioTools4」を実行します。

GravioTools4が起動するとタスクバーにアイコンが表示されますので、右クリックしてメニューを表示して「設定」メニューを選択します。



設定パネルから「ファームウェア」タブを選択します。



「確認」ボタンを押してファームウェアの現在のバージョンを確認します。

現在のバージョンが最新のバージョンより古い場合に「アップデート」ボタンを押してファームウェアを更新します。

アップデートが終了しましたらファームウェアが更新されますので使用してください。また既にペアリングしているセンサーはそのまま使用することが出来ます（ペアリングは維持されています）。

3. Linuxの場合

Linux版ではZigbeeのファームウェアのアップデートは行えませんが、WindowsもしくはMacをご使用ください。

4. Gravio Hubの場合

Gravio Hubの場合は、[こちら](#) を参照してUpdateManagerよりアップデートしてください。

14.8. i-PROカメラの設定方法

i-PROカメラの設定方法についてカメラはWV-S1135Vを使用します。

WV-S1135VカメラのWebページにアクセスして設定を開きます。WV-S1135Vの使用方法、Webページの操作方法につきましてはi-PROの説明書を参照してください。



AI動体感知

AI動体感知を使用するためにはAI-VMD (WV-XAE200WUX) をカメラにインストールする必要があります。バージョンは3.2以上を使用してください。

AI認識する動作を検知するAI動体検知設定、カウント機能設定、詳細設定などのi-PRO側の設定方法につきましてはi-PROのマニュアルを参照してください。

Gravioとの連携を行うためには、カウント機能設定画面を開きます。

HTTP送信にGravioのi-PRO設定画面のURLを登録してください。

AI-VMD設定画面 | ①モード選択 > ②AI動体検知設定 > ③カウント機能設定 > ④詳細設定 > ⑤スケジュール設定 > ⑥アラーム設定

検知ラインとマスクエリアは、それぞれ最大8か所まで設定できます。
また、対象物が検知ラインを通過する前後各1秒以上映るように画角調整してください。

2023/07/03 14:07:10

In Out

0 ← 0

ライン設定

HTTP送信

MQTT送信

その他

カウント情報をHTTPで外部サーバーに送信することができます。
HTTPの送信情報は、外部インターフェイス仕様書をご参照ください。

送信先1

☒ 送信

送信先アドレス

送信先パス名

SSL

送信先ポート番号

ユーザー名

パスワード

送信間隔

送信先2

☐ 送信

送信先アドレス

送信先パス名

SSL

送信先ポート番号

ユーザー名

パスワード

送信間隔

設定

AI混雑感知

AI混雑感知を使用するためにはWV-XAE207Wをカメラにインストールする必要があります。
バージョンは1.6以上を使用してください。

AI認識する動作を検知する基本設定、MQTT送信設定、詳細設定などのi-PRO側の設定方法につきましてはi-PROのマニュアルを参照してください。
Gravioとの連携を行うためには、HTTP送信画面を開きます。

送信先にGravioのi-PRO設定画面のURLを登録してください。

基本

HTTP送信

MQTT送信

詳細

カウント情報をHTTPで外部サーバーに送信することができます。
HTTPの送信情報は、外部インターフェイス仕様書をご参照ください。

送信先1

☒ 送信

送信先アドレス

送信先パス名

SSL

送信先ポート番号

ユーザー名

パスワード

送信間隔

送信先2

☐ 送信

送信先アドレス

送信先パス名

SSL

送信先ポート番号

ユーザー名

パスワード

送信間隔

アラーム

アラームは、メニューよりアラームの動作検知エリアを設定して通知を開きます。
通知のHTTPアラーム通知にGravioのi-PRO設定画面のURLを登録してください。

Network Camera
WV-S1135V

ライブ画

設定

機能拡張ソフトウェア

ソフトウェア管理

動作スケジュール

管理ログ

AI-VMD

AI異常検知

+かんたん設定

+基本

+映像/音声

+マルチスクリーン

-アラーム

アラーム

動作検知エリア

妨害検知エリア

音検知

通知

+ユーザー管理

+ネットワーク

+スケジュール

+メンテナンス

+サポート

アラーム

動作検知エリア

妨害検知エリア

音検知

通知

独自アラーム通知 | HTTPアラーム通知

HTTPアラーム通知

アラーム

通知先 1	☒	https:// :29448	削除
ユーザー名			
パスワード			
通知データ		/ipro/alarm?ip=%ip&mac=%mac&time=%stime&gmt=%s	
通知先 2	☒	http://	削除
ユーザー名			
パスワード			
通知データ		/cgi-bin/comalarm.cgi?CMD=01	
通知先 3	☐	http://	削除
ユーザー名			
パスワード			
通知データ		/cgi-bin/comalarm.cgi?CMD=01	
通知先 4	☐	http://	削除
ユーザー名			
パスワード			
通知データ		/cgi-bin/comalarm.cgi?CMD=01	
通知先 5	☐	http://	削除
ユーザー名			
パスワード			
通知データ		/cgi-bin/comalarm.cgi?CMD=01	

設定

14.9. OpenAIのAPI Keyの取得方法

OpenAICompletion、OpenAIChat、OpenAIImage、OpenAIAudioコンポーネントを使用するにはOpenAI社のアカウントを作成してAPI Keyを取得してください。

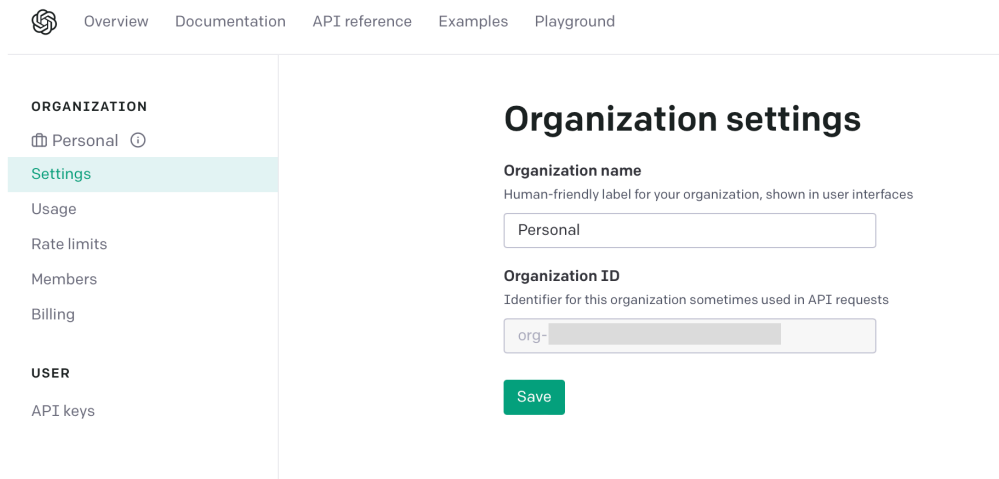
1. アカウントの作成

[こちら](#) のOpenAI社のサイトの右上にある Sign Up からサインアップします。

2. Organization Idの確認

OpenAI社のサイトでサインアップしたアカウントでログインします。

自分のアカウントの Manage Account メニューを開きます。



The screenshot shows the OpenAI Organization settings page. On the left, there is a sidebar with a menu. Under the 'ORGANIZATION' section, 'Settings' is highlighted. The main content area is titled 'Organization settings'. It contains two sections: 'Organization name' with a text input field containing 'Personal', and 'Organization ID' with a text input field showing 'org-' followed by a masked area. A green 'Save' button is at the bottom.

左のメニューからSettingを選択すると Organization settings 画面にOrganization IDが表示されます。

各コンポーネントの Organization Id プロパティで使用します。

3. API Keyの取得

左のメニューからAPI keysを選択するとAPI keys 画面が表示されます。

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically rotate any API key that we've found has leaked publicly.

NAME	KEY	CREATED	LAST USED ⓘ
	sk-...		 
	sk-...		 
	sk-...		 
+ Create new secret key			

Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

Personal 

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

Create new secret key ボタンを押して Nameを指定して作成します。

KEYは各コンポーネントのAPI Key プロパティで使します。

これでOpenAIのAPI Keysの取得ができました。

14.10. Text2SQLの使い方

データベース関係のコンポーネントで提供しているTextSQL機能は、OpenAIのChatGPTの機能を使用して取得したいデータをプロンプトに入力した日本語からSQL文を生成できる機能です。TextSQL機能を使用するには事前にOpenAIのAPI KeyとOrganizationIdを取得する必要があります。取得する方法は [こちら](#) を確認してください。

1. 事前準備

OpenAIのAPI KeyとOrganizationIdを取得してください。そして各データベース関係のコンポーネントでデータベースに正しく接続できるようにプロパティを設定してください。

1. Text2SQLでのテーブルの選択

Text2SQLボタンを押すとText2SQLパネルが開きます。

Text 2 SQL

Which tables would you like to include?
The more tables you select, the worse performance may be.

Select All

Deselect All

Selected	Table Name	Column 0	Column 1	Column 2	Column 3
☐	Customers	customer_id	first_name	last_name	email
☐	users	id	name	email	active

Cancel

Next

1. プロンプトからSQL文の生成

Text2SQLパネルが開くと、正しくデータベースに接続できている場合はテーブルが一覧で表示されますので、使用するテーブルを選択します。

Nextボタンを押すとプロンプトの入力パネルが表示されます。

Text 2 SQL

ENTER PROMPT

Usersテーブルから最新の1行を取得する

Generate

GENERATED SQL

SELECT * FROM users ORDER BY id DESC LIMIT 1

Back Test Query Accept

プロンプトに取得したいデータを日本語で入力します。Generateボタンを押すとGENERATED SQLに生成されたSQL文が表示されます。

Test Queryボタンを押すとSQLを実行してデータを取得するパネルが表示されます。

Acceptボタンを押すと生成したSQL文をコンポーネントのSQLプロパティにコピーされます。

14.11. Google Cloud Storage設定方法

GCS Upload/Downloadコンポーネントを使用してGoogle Cloud Storageにファイルを送受信する方法の設定例を次に示します。

設定方法の流れ

GCPで以下の手順で設定を行います。

1. IAMでサービスアカウントを作成して鍵ファイルを作成してダウンロードします。
2. Google Cloud Storageでバケットに対して権限として1.で作成したサービスアカウントを紐付けて、ロールに「Storage レガシーバケットオーナー」と「Storage レガシーオブジェクトオーナー」を指定します。
GCSDownloadコンポーネントの場合にはロールは「Storage レガシーバケット読み取り」と「Storage レガシーオブジェクト読み取り」を指定します。
3. GCS Upload/Downloadコンポーネントでは2.で指定したバケットに1.の鍵ファイルを指定して使用します。

事前準備

Google CloudのGoogle Cloud Storageにバケットを作成します。

Google Cloud コンソールでサービス アカウントの作成

Google CloudコンソールからIAMと管理に移動してサービス アカウントを作成します。



サービス アカウント名とサービス アカウントIDを設定します。

サービス アカウントの作成

1 サービス アカウントの詳細

サービス アカウント名
gravio-service

このサービス アカウントの表示名
サービス アカウント ID
gravio-service

メールアドレス
gravio-service@gravio-edge.iam.gserviceaccount.com

サービス アカウントの説明
このサービス アカウントで行うことを説明します

作成して続行

2 このサービス アカウントにプロジェクトへのアクセスを許可する (省略可)

3 ユーザーにこのサービス アカウントへのアクセスを許可 (省略可)

完了 キャンセル

「このサービスアカウントにプロジェクトへのアクセスを許可する」と「ユーザーにこのサービスアカウントへのアクセスを許可」は省略可能ですので何も指定せずに続行して「完了」ボタンを押して保存すると一覧に表示されます。

サービス アカウント

プロジェクト「gravio-edge」のサービス アカウント

サービス アカウントは Google Cloud サービス ID (Compute Engine VM、App Engine アプリ、Google 以外で実行されているシステムなどで実行されているコード) を表します。 [サービス アカウントの詳細をご覧ください。](#)

組織のポリシーを使用してサービス アカウントを保護できます。IAM ロールの自動付与、鍵の作成やアップロード、サービス アカウントの完全な作成など、リスクのあるサービス アカウント機能をブロックすることが可能です。 [サービス アカウントの組織のポリシーの詳細をご覧ください。](#)

フィルタ プロパティ名または値を入力

メール	ステータス	名前	説明	キー ID	キーの作成日	OAuth 2 クライアント ID	操作
gravio-service@gravio-edge.iam.gserviceaccount.com	アクティブ	gravio-service		キーがありません		105499089080261199567	詳細を管理 権限を管理 鍵を管理 指標を表示 ログを表示 無効にする 削除

操作から「鍵を管理」を選択します。

サービス アカウント

プロジェクト「gravio-edge」のサービス アカウント

サービス アカウントは Google Cloud サービス ID (Compute Engine VM、App Engine アプリ、Google 以外で実行されているシステムなどで実行されているコード) を表します。 [サービス アカウントの詳細をご覧ください。](#)

組織のポリシーを使用してサービス アカウントを保護できます。IAM ロールの自動付与、鍵の作成やアップロード、サービス アカウントの完全な作成など、リスクのあるサービス アカウント機能をブロックすることが可能です。 [サービス アカウントの組織のポリシーの詳細をご覧ください。](#)

フィルタ プロパティ名または値を入力

メール	ステータス	名前	説明	キー ID	キーの作成日	OAuth 2 クライアント ID	操作
gravio-service@gravio-edge.iam.gserviceaccount.com	アクティブ	gravio-service		キーがありません		105499089080261199567	詳細を管理 権限を管理 鍵を管理 指標を表示 ログを表示 無効にする 削除

「鍵を追加」を選択します。



キーのタイプは「JSON」で作成します。



作成するとPCに鍵ファイルがダウンロードされます。



Google Cloud Storageでサービス アカウントをプリンシバルに設定

IAMと管理で作成したサービス アカウントをGoogle Cloud Storageのプリンシバルに設定します。

Google Cloud Storageのブラウザメニューから読み書きするバケットを選択して「権限」からプリンシバルの「作成」を選択します。



「新しいプリンシバル」に作成したサービス アカウントを入力します。

「ロール」に「Storage レガシー バケット オーナー」と「Storage レガシー オブジェクト オーナー」を追加して保存します。



新しいプリンシバルを保存すると一覧に表示されます。

Cloud Storage

ブラウザ

モニタリング

設定

バケットの詳細

更新

学ぶ

gravio-edge

ロケーション: asia-northeast1 (東京)

ストレージクラス: Standard

公開アクセス: 非公開

保護: なし

オブジェクト

設定

権限

保護

ライフサイクル

公開アクセス

非公開

このバケットは一般公開されていません。公共のインターネット上にオブジェクトを公開すべきでないことが明白な場合は、このバケットの公開アクセスも禁止する必要があります。[詳細](#)

公開アクセスの防止

Access control

均一: 有効なオブジェクト レベルの ACL がない

この設定を変更できる残り日数: 88 日

すべてのオブジェクト アクセスはバケットの権限により制限されます。オブジェクトが独自のアクセス制御リスト (ACL) を持つことはできません。オブジェクト単位のアクセスを許可するには、90 日以内にきめ細かいアクセスに切り替えます。[詳細](#)

きめ細かい管理に切り替える

権限

追加

削除

表示: プリンシパル

ロール

フィルタ

プロパティ名または値を入力

?

||

タイプ	プリンシパル	名前	ロール	継承
☐	gravio-service@gravio-edge.iam.gserviceaccount.com	gravio-service	Storage レガシー オブジェクト オーナー	
☐	プロジェクト gravio-edge の所有者		Storage レガシー バケット オーナー	
☐	プロジェクト gravio-edge の所有者		Storage レガシー オブジェクト オーナー	
☐	プロジェクト gravio-edge の閲覧者		Storage レガシー バケット オーナー	
☐	プロジェクト gravio-edge の閲覧者		Storage レガシー オブジェクト 読み取り	
☐	プロジェクト gravio-edge の編集者		Storage レガシー バケット 読み取り	
☐	プロジェクト gravio-edge の編集者		Storage レガシー オブジェクト オーナー	
☐	プロジェクト gravio-edge の編集者		Storage レガシー バケット オーナー	

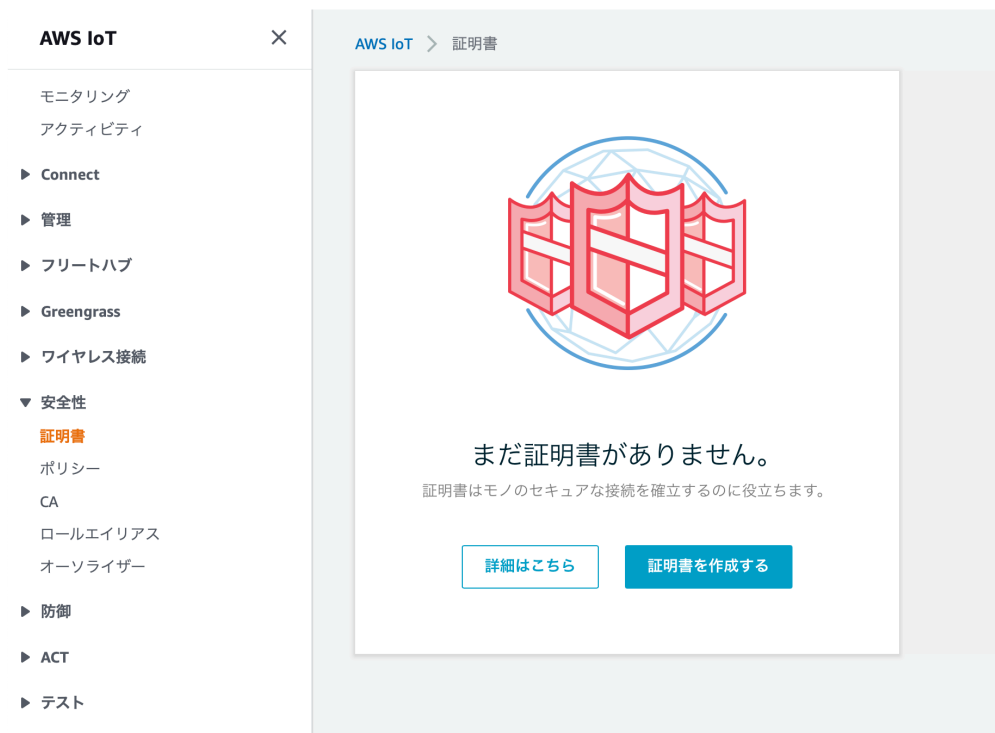
Google Cloud コンソールでの設定は終わりました。

14.12. AWS IoT MQTTの設定方法

AWS IoT MQTTシステムにメッセージを公開する方法の例を次に示します。

証明書の作成とダウンロード

AWS IoTを開いて左のメニュー「安全性」にある「証明書」を選択して、「証明書を作成する」をクリックします。



1-Click 証明書作成(推奨)を選択します。

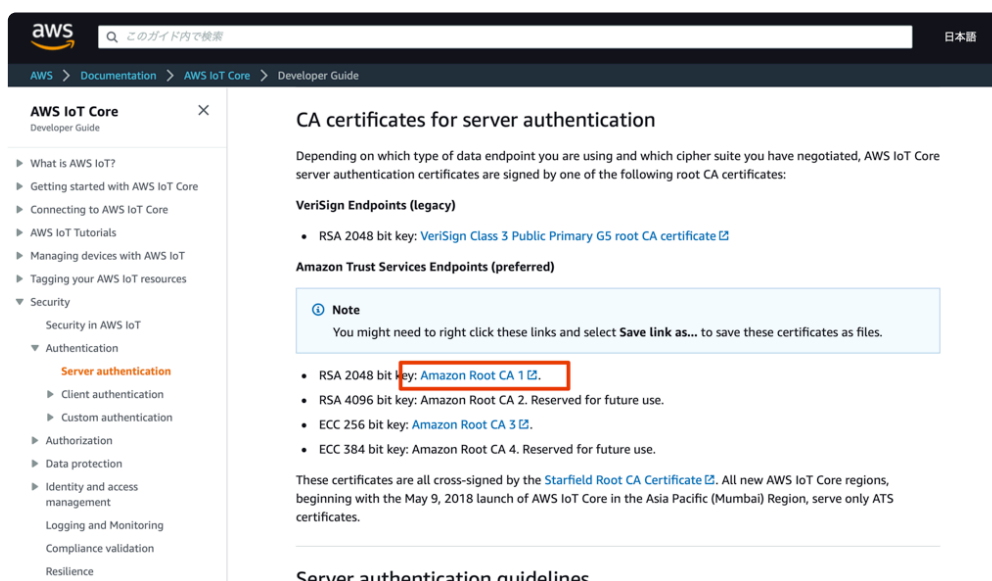


証明書が作成されると3つの証明書がダウンロードできますので、全てダウンロードします。
(パブリックキーは使用しません)

Amazon IoTのルート CAの証明書をダウンロードするために「ダウンロード」を選択します。



RSA 2048 bit key : Amazon Root CA 1のリンクをクリックすると証明書が表示されるのでテキストをコピー&ペーストでエディタに保存します。




```
-----BEGIN CERTIFICATE-----
MIIDQTCCAimgAwIBAgITBmyfz5m/jAo54vB4ikPmljZbyjANBgkqhkiG9w0BAQSF
ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6
b24gUm9vdCBDQSAxMB4XDTE1MDUyNjAwMDAwMFoXDTE1MDUyNjAwMDAwMFowOTEL
MAkGA1UEBhMCVVMxMDZANBgNVBAoTBkFtYXpvcjEzMBCGA1UEAxMQQW1hem9uIFJv
b3QgQ0EgMTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALJ4gHHKeNXj
ca9HgFB0fW7Y14h29Jlo9lghYPl0hAEvRAiThtOgQ3pOsqTQNroBvo3bSMgHFzZM
906II8c+6zf1tRn4SWiw3te5djgdYZ6k/oI2peVKVuRF4fn9tBb6dNqcmzU5L/qw
IFAGbHrQgLKm+a/sRxmPUDgH3KKHOVj4utWp+UhnMJbulHheb4mjUcAwhmahRWa6
VOujw5H5SNz/0egwLX0tdHA114gk957EWW67c4cX8jJGKLhD+rcdqsq08p8kDi1L
93FcXmn/6pUCyziKrlA4b9v7LWIbxcceVOF34GfID5yHI9Y/QCB/IIDEgEw+OyQm
jgSubJrIqg0CAwEAaANCMEAwDwYDVR0TAQH/BAUwAwEB/zAOBgNVHQ8BAf8EBAMC
AYYwHQYDVR0OBBYEFIQYzIU07LwMlJQuCFmcx7IQTgoIMA0GCSqGSIb3DQEBCwUA
A4IBAQC8Y8jdaQZChGsV2USggNiMOruYou6r4lK5IpDB/G/wkjUu0yKGX9rbxenDI
U5PMCCjjmCXPI6T53iHTfIUJrU6adTrCC2qJeHZERxhlbI1BjJt/msv0tadQ1wUs
N+gDS63pYaACbvXy8MWy7Vu33PqUXHeeE6V/Uq2V8viTO96LXFvKW1JbYK8U90vv
o/ufQJVtMVT8QtPHRh8jrdkPSHca2XV4cdFyQzR1bldZwgJcJmApzyMZFo6IQ6XU
5MsI+yMRQ+hDKXJioaldXgjUkK642M4UwtBV8ob2xJNDd2ZhwLnoQdeXeGADbkpy
rQXRfboQnoZsG4q5WTP468SQvvG5
-----END CERTIFICATE-----
```

ポリシーの設定

続いてポリシーを設定します。AWS IoTを開いて安全性にある「ポリシー」を選択して、「ポリシーの作成」をクリックします。



ポリシーの名前を入力し、アクションは「iot:*」、リソースARNは「*」と入力して効果の「許可」にチェックを入れて「作成」をクリックしてください。

ポリシーの作成

ポリシーを作成して、認可アクションのセットを定義します。1 つ以上のリソース (モノ、トピック、トピックフィルター) のアクションを承認できます。IoT ポリシーの詳細については、「[AWS IoT ポリシーのドキュメントページ](#)」を参照してください。

名前

ステートメントを追加

ポリシー構文は、リソースで実行できるアクションの種類を定義します。

アドバンスドモード

アクション

リソース ARN

効果

☒ 許可 ☐ 拒否

削除

ステートメントを追加

作成

ポリシーが作成されました。

AWS IoT > 証明書 > b2a8fa28df81a1e752709cafc0b7a6a8d0c91d8563a155086b9edfdd3b352b61

証明書

b2a8fa28df81a1e752709cafc0b7a6a8d0c91d8563a155086b9edfdd3b352b61

アクティブ

アクション

詳細

ポリシー

ポリシー

モノ

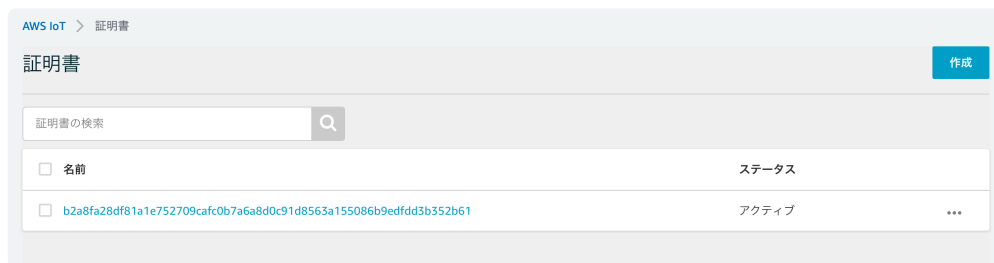
コンプライアンス違反

gravio_mqtt

...

証明書にポリシーの関連付け

安全性にある「証明書」を選択して、作成した証明書ををクリックします。



アクションを選択して「ポリシーのアタッチ」をクリックします。



先ほど作成したポリシーをチェックして「アタッチ」をクリックします。



エントリーポイントの確認

MQTT パブリッシュ、サブスクライブするエントリーポイントを確認します。

左のメニュー「設定」をクリックします。エンドポイントが表示されますのでこちらを使用します。



エンドポイントまでの設定確認が終わりますとAWS IoTでの設定は終わりました。

Gravio StudioでのMQTTPublisherの設定

Gravio StudioでMQTTPublisherのプロパティには以下のように設定します。

トピックは 任意の値を指定します。

クライアントIDは ユニークな値を指定します。

Step 1
MQTTPublish

ステップの追加

+

Properties

基本プロパティセット

(cp.BaseProperty)

(なし)

トピック

(cp.Topic)

graviotopic

クライアントID

(cp.ClientId)

gravio1

QoS

(cp.QOS)

QoS 0

☐ Retain

(cp.Retain)

URLは“ssl://”に続けてエントリポイントを入力します。
ポートは「8883」でユーザ名とパスワードは空欄にします。

Properties	
URL	(cp.URL)
ssl://	
ポート	(cp.Port)
8883	
ユーザ名	(cp.Username)
パスワード	(cp.Password)

証明書、秘密鍵、ルート証明書はダウンロードしたファイルを選択します。秘密鍵パスフレーズは空欄にします。

Properties	
証明書	(cp.Certificate)
ファイルを選択する	
秘密鍵	(cp.PrivateKey)
ファイルを選択する	
秘密鍵パスフレーズ	(cp.PrivateKeyPassword)
ルート証明書	(cp.RootCA)
ファイルを選択する	

これでMQTTPublisherからAWS IoTのMQTTにデータを送信する設定が整いました。

14.13. Azure IoT Hub (MQTT)の設定方法

Azure IoT Hub (MQTT)にメッセージを送信する方法の例を次に示します。

事前準備

証明書を作成するためにLinuxかMacを準備してください。

ツールのダウンロード

Azure IoT Hubで証明書を使用するため [こちら](#) からコマンドラインツールをLinuxかMacにダウンロードしてください。

Windowsの場合はPowerShellからbashコマンドを実行して、bashを使用してください。

次のコマンドを入力して、フィールドに貼り付けることができる証明書を作成します。

ダウンロードしたzipファイルを展開してtoolsディレクトリに移動します。

```
cd <path>/azure-iot-sdk-c-main/tools
```

ツールに実行権限をつけます。

```
chmod +x ./certGen.sh
```

CA証明書の作成

ツールを実行してCA証明書を作成します。

```
./certGen.sh create_root_and_intermediate
```

デバイス証明書の作成

ツールを実行してデバイスの証明書を作成します。

```
./certGen.sh create_device_certificate <device id>
```

※はデバイスIDを指定します

デバイスの公開鍵を作成します。

```
openssl pkcs12 -in certs/new-device.cert.pfx -clcerts -nokeys -out <device id>.crt
```

※パスワードの入力が求められたら画面に表示されている設定されているパスワードを入力します。

デバイスの秘密鍵を作成します。

```
openssl pkcs12 -in certs/new-device.cert.pfx -nocerts -nodes -out <device id>.key
```

※パスワードの入力が求められたら画面に表示されている設定されているパスワードを入力します。

.crtと.keyはGravio Studioで使います。

複数のデバイスを使用するには、デバイスの分だけ作成してください。

Azure でIoT Hubの作成

AzureにログインしてIoT Hubを表示して作成します:

[ホーム](#) >

IoT Hub

✧ ...

×

[+ 作成](#) [⚙️ ビューの管理](#) [🔄 更新](#) [↓ CSV にエクスポート](#) [🔗 クエリを開く](#) | [🏷️ タグの割り当て](#)

任意のフィールドのフィル...

サブスクリプション 次の値と等しい **すべて**

リソース グループ 次の値と等しい **すべて** ✕

場所 次の値と等しい **すべて** ✕

[+ フィルターを追加](#)

グループ化なし

≡ リスト ビュー

名前 ↑↓

種類 ↑↓

リソース グループ ↑↓

場所 ↑↓

サブスクリプション ↑↓



表示する IoT ハブ がありません

何十億にも及ぶ IoT アセットに接続し、監視および管理できるように、IoT ハブを作成します。

[IoT ハブ の作成](#)

[IoT Hub に関する詳細情報の表示](#)

[クイックスタート: デバイスからテレメトリを送信する](#)

[クイックスタート: デバイスからテレメトリを送信する](#)

リソースグループは新規作成してします。サンプルでは“iothub“としました。

IoT ハブ ...

Microsoft

[基本](#) [ネットワーク](#) [管理](#) [タグ](#) [確認および作成](#)

何十億にも及ぶ IoT アセットに接続し、監視および管理できるように、IoT ハブを作成します。 [詳細情報](#)

プロジェクトの詳細

デプロイとコストを管理するために使用するサブスクリプションを選択します。フォルダーなどのリソース グループを使用して、リソースの整理と管理を行うことができます。

サブスクリプション * ⓘ

Azure subscription 1



リソース グループ * ⓘ

[新規作成](#)

インスタンスの詳細

IoT Hub 名 * ⓘ

領域 * ⓘ

リソース グループは、Azure のソリューションに関連するリソースを保持するコンテナです。

名前 *

iothub



OK

キャンセル

IoT Hub名を入力します。サンプルでは“gravio-mqtt”としました。

IoT ハブ ...

Microsoft

[基本](#) [ネットワーク](#) [管理](#) [タグ](#) [確認および作成](#)

何十億にも及ぶ IoT アセットに接続し、監視および管理できるように、IoT ハブを作成します。 [詳細情報](#)

プロジェクトの詳細

デプロイとコストを管理するために使用するサブスクリプションを選択します。フォルダーなどのリソース グループを使用して、リソースの整理と管理を行うことができます。

サブスクリプション * ⓘ

Azure subscription 1

リソース グループ * ⓘ

(新規) iothub

[新規作成](#)

インスタンスの詳細

IoT Hub 名 * ⓘ

gravio-mqtt

領域 * ⓘ

Japan East

ネットワークを指定します。

IoT ハブ ...

Microsoft

[基本](#) [ネットワーク](#) [管理](#) [タグ](#) [確認および作成](#)

IoT ハブには、パブリック ホスト名経由でパブリックに接続することも、プライベート エンドポイントを使用してプライベートに接続することもできます。 [詳細情報](#)

接続の構成 *

☒ パブリック アクセス☐ プライベート アクセス (推奨)

i このリソースが作成された後に、この設定を変更したり、別の接続方法を構成したりすることができます。 [詳細情報](#)

管理でスケーリングレベルとユニットなどパフォーマンスを指定します。サンプルでは“F1:Free レベル”を選択しました。

IoT ハブ ...

Microsoft

[基本](#) [ネットワーク](#) [管理](#) [タグ](#) [確認および作成](#)

それぞれの IoT ハブには、特定レベルの特定数のユニットがプロビジョニングされます。レベルとユニット数によって、1 日に送信できるメッセージの最大クォータが決まります。 [詳細](#)

スケーリング レベルとユニット

価格とスケールティア * ⓘ

F1: Free レベル

[ソリューションに合った正しい IoT ハブのレベルを選択する方法について](#)

IoT ハブ F1 のユニット数 ⓘ



1

IoT ハブをスケーリングできる度合いを決定します。ニーズが増大した場合は、後でこれを変更できます。

Defender for IoT



オフ

Microsoft [Defender for IoT](#) は、Azure IoT Hub、IoT Edge、デバイスに脅威からの保護のための追加のレイヤーを付加する個別のサービスです。このサービスに対しては個別に課金されます。Defender for IoT ではデータを処理した後、IoT Hub とは異なる地理的な場所に格納する場合があります。 [詳細情報](#)

価格とスケールティア ⓘ	F1	device-to-cloud メッセージ ⓘ	有効
1日あたりのメッセージ ⓘ	8,000	メッセージ ルーティング ⓘ	有効
月あたりのコスト	0.00 USD	cloud-to-device コマンド ⓘ	有効
Defender for IoT ⓘ	無効	IoT Edge ⓘ	有効
		デバイス管理 ⓘ	有効

ロールベースのアクセス制御

アクセス許可モデルを Azure ロールベースのアクセス制御 (RBAC) のみに、または共有アクセス ポリシーと RBAC の組み合わせに変更します。 [詳細情報](#)

☐ RBAC のみ☒ 共有アクセス ポリシー + RBAC

インスタンス内の要素を管理するには、IoT Hub データ API に対するアクセスが必要です。以下の推奨されるロールを選択して、API へのフル アクセスを自分自身に許可してください。アクセスの制御 (IAM) を使用して、後で適切なロールを選択することもできます。 [詳細情報](#)

☐ IoT ハブ データ共同作成者ロールに自分を割り当てる ⓘ

詳細設定

スケーリング

Device-to-cloud パーティション ⓘ



2

確認及び作成で設定内容を確認して作成します。

IoT ハブ ...

Microsoft

 検証に成功しました。

基本 ネットワーク 管理 タグ 確認および作成

基本

サブスクリプション	Azure subscription 1
リソース グループ	iothub
領域	Japan East
IoT Hub 名	gravio-mqtt
ディザスター リカバリーが有効	はい

ネットワーク

接続の構成	パブリック アクセス
プライベート エンドポイント接続	なし
公衆ネットワーク アクセスを許可する	有効

管理

価格とスケールティア	F1
IoT ハブ F1 のユニット数	1
1 日あたりのメッセージ	8,000
Device-to-cloud パーティション	2
月あたりのコスト	0.00 USD
Defender for IoT	無効

タグ

デプロイが完了すると“gravio-mqtt”が作成されます。

gravio-mqtt-97105242 | 概要

デプロイ

検索 (Cmd+ /)

削除

キャンセル

再デプロイ

ダウンロード

最新の情報に更新

概要

入力

出力

テンプレート

デプロイが完了しました

デプロイ名: gravio-mqtt-97105242

サブスクリプション: Azure subscription 1

リソース グループ: iothub

開始時刻: 2022/9/7 10:52:44

相関 ID: e88c3b91-186b-4b67-8a00-b56bf4cc0599

展開の詳細

リソース	種類	状態	操作の詳細
gravio-mqtt	Microsoft.Devices/IoTH...	OK	操作の詳細

次の手順

IoT デバイスの追加と構成

デバイス メッセージングのルーティング規則の構成

リソースに移動

“gravio-mqtt”を選択して証明書を開いて作成します。

ホーム > IoT Hub > gravio-mqtt

gravio-mqtt | 証明書

検索 (Cmd+ /)

更新プログラム

クエリ

ハブ設定

組み込みのエンドポイント

メッセージ ルーティング

ファイルのアップロード

フェールオーバー

プロパティ

ロック

セキュリティ設定

ID

共有アクセス ポリシー

ネットワーク

証明書

こちらでデバイス認証の証明書をアップロードして管理します。詳細情報

新しい証明書をアップロードすると、自動的に証明書を確認できます (既存の証明書の状態は変更されないことに注意してください)。詳細情報

追加

最新の情報に更新

名前	作成日時	期限切れ	サブジェクト	サムプリント	状態
結果がありません。					

証明書の追加で証明書名を入力して事前準備で作成したCA証明書を指定します。
“アップロード時に証明書の状態を確認済みに設定する”をオンにします。

ホーム > IoT Hub > gravio-mqtt

gravio-mqtt | 証明書

IoT Hub

検索 (Cmd+/)

更新プログラム

クエリ

ハブ設定

組み込みのエンドポイント

メッセージ ルーティング

ファイルのアップロード

フェールオーバー

プロパティ

ロック

セキュリティ設定

ID

共有アクセス ポリシー

ネットワーク

証明書

こちらでデバイス認証の証明書をアップロードして管理します。[詳細情報](#)

新しい証明書をアップロードすると、自動的に証明書を確認できます (既存の証明書の状態は変更されません)

追加

最新の情報に更新

名前	作成日時	期限切れ	サブジェクト
結果がありません。			

証明書の追加

証明書名 ⓘ

gravio-mqtt-certs ✓

.pem または .cer の証明書ファイルです。 ⓘ

"azure-iot-test-only.root.ca.cert.pem" 📎

✓ アップロード時に証明書の状態を確認済みに設定する ⓘ

ⓘ この証明書は、手動で確認する必要があるため、自動的に検証 詳細情報
されます。

証明書を保存すると一覧に表示されます。

ホーム > IoT Hub > gravio-mqtt

gravio-mqtt | 証明書

IoT Hub

検索 (Cmd+/)

更新プログラム

クエリ

ハブ設定

組み込みのエンドポイント

メッセージ ルーティング

ファイルのアップロード

フェールオーバー

プロパティ

ロック

セキュリティ設定

ID

共有アクセス ポリシー

ネットワーク

証明書

こちらでデバイス認証の証明書をアップロードして管理します。[詳細情報](#)

新しい証明書をアップロードすると、自動的に証明書を確認できます (既存の証明書の状態は変更されないことに注意してください)。[詳細情報](#)

追加

最新の情報に更新

名前	作成日時	期限切れ	サブジェクト	サムプリント	状態
gravio-mqtt-certs	2001/1/1	2022/10/7	Azure IoT Hub CA Cert T...	412747FCB3C98FB2CE0...	✓ 確認済み ***

デバイスを開いて作成します。

Copyright © 2023 ASTERIA Corporation

ページ 568 / 633

ホーム > gravio-mqtt-97105242 | 概要 > gravio-mqtt

gravio-mqtt | デバイス ✳ ... ×

IoT Hub

🔍 検索 (Cmd+/) << IoT Hub 内のデバイスを表示、作成、削除、更新します。

🔗 概要

📄 アクティビティ ログ

🔑 アクセス制御 (IAM)

🏷 タグ

🔧 問題の診断と解決

⚡ イベント

💰 価格とスケール

📱 デバイス管理

📱 **デバイス**

☁ IoT Edge

🏗 構成

🔄 更新プログラム

📄 クエリ

デバイス名

デバイス ID の入力

🔍 デバイスの検索 </> クエリを使用して検索

+ デバイスの追加 🔄 最新の情報に更新 🗑 削除

デバイス ID	状態	前回の状態の更...	認証の...	クラウ...
表示する IoT デバイスがありません。				

デバイスIDはデバイスの証明書で指定したIDを入力します。
認証の種類は“X.509CA署名済み”を選択します。

ホーム > IoT Hub > gravio-mqtt | デバイス >

🔗 デバイスの作成 ... ×

📘 Azure IoT 用に認定されたデバイスをデバイス カタログで検索します

デバイス ID * ①

d001 ✓

認証の種類 ①

対称キー X.509 自己署名済み **X.509 CA 署名済み**

このデバイスを IoT ハブに接続する ①

有効化 無効化

親デバイス ①

親デバイスがありません

[親デバイスの設定](#)

デバイスを保存すると一覧に表示されます。

ホーム > IoT Hub > gravio-mqtt

gravio-mqtt | デバイス

IoT Hub

検索 (Cmd+/)

IoT Hub 内のデバイスを表示、作成、削除、更新します。

デバイス名

デバイス ID の入力

デバイスの検索

クエリを使用して検索

+ デバイスの追加 最新の情報に更新 削除

デバイス ID	状態	前回の状態の更新	認証の種類	クラウ...
d001	Enabled	--	CertificateAuthority	0

デバイス管理

- デバイス
- IoT Edge
- 構成
- 更新プログラム
- クエリ

概要を開いてホスト名を記録します。ホスト名はAzure IoT Hubのエントリポイントとなります。

gravio-mqtt

IoT Hub

検索 (Cmd+/)

移動 削除 最新の情報に更新 フィードバック

概要

アクティビティ ログ

アクセス制御 (IAM)

タグ

問題の診断と解決

イベント

価格とスケール

デバイス管理

デバイス

IoT Edge

へ 基本

JSON ビュー

リソース グループ (移動) : [iothub](#)

ホスト名 : gravio-mqtt.azure-devices.net

価格とスケールティア : F1 - 無料

IoT Hub のユニット数 : 1

状態 : Active

現在の場所 : Japan East

サブスクリプション (移動) : [Azure subscription 1](#)

サブスクリプション ID : fc1ebad9-8266-456d-bf0e-02512c9d0546

タグ (編集) : [タグを追加するにはここをクリック](#)

Gravio StudioでのMQTTPublisherの設定

Gravio StudioでMQTTPublisherのプロパティには以下のように設定します。

プロパティ名	説明
トピック	“devices/” <u>デバイスID</u> “/messages/events/”を指定します。例 “devices/d002/messages/events/”
クライアントID	デバイスIDを指定します。例 “d001”
QoS	(デフォルトのまま)
Retain	(デフォルトのまま)

タイムアウト	(デフォルトのまま)
URL	“ssl://”に続けてホスト名を指定します。例 “ssl://gravio-mqtt.azure-devices.net”
ポート	8883を指定します
ユーザ名	ホスト名+”/”+デバイスIDを指定します。例 “gravio-mqtt.azure-devices.net/d001”
パスワード	(指定しない)
証明書	事前準備で作成したデバイスの公開鍵を指定します
秘密鍵	事前準備で作成したデバイスの秘密鍵を指定します
秘密鍵パスフレーズ	(指定しない)
ルート証明書	(指定しない)

これでMQTTPublisherからAzure IoT Hub (MQTT) にデータを送信する設定が整いました。

14.14. Azure アプリの登録方法 (Microsoft365)

WriteToExcel365コンポーネントを使用してMicrosoft365にExcelファイルをポストする方法を次に示します。
このコンポーネントを使用するには、Microsoft365 for BusinessとOneDrive for Businessの契約と事前設定が必要です。

Microsoft365 for BusinessとOneDrive for Businessの事前設定

Microsoft365のアカウントをご準備いただき、[Azure Portal](#) にログインしてください。
契約者ページより「アプリの登録」を選択してアプリを登録してください。

The screenshot shows the 'App Registrations' page in the Azure Portal for an application named 'gravioexcel'. The 'Name' field is highlighted with a red box. Below it, the 'Supported account types' section is also highlighted with a red box, showing the selected option: 'Accounts in this organizational directory only (Azure AD - Single tenant)'. The 'Redirect URI' field is set to 'https://example.com/auth'.

アプリを作成してから詳細画面を開き、表示されるアプリケーションIDとディレクトリ(テナント)IDをメモします。
これらの値はMicrosoft365のアカウントのメールアドレスとパスワードとあわせて、WriteToExcel365コンポーネントのプロパティに設定します。

The screenshot shows the 'Overview' tab of the 'App Registrations' page for 'gravioexcel'. The 'Application ID' and 'Directory ID' fields are highlighted with red boxes. The 'Application ID' is 'a1b2c3d4-e5f6-g7h8-i9j0-k1l2m3n4o5p6' and the 'Directory ID' is '12345678-90ab-cdef-ghij-klmnopqrstuv'.

続いて認証メニューから、サポートであるアカウントの種類が この組織ディレクトリのみに含まれるアカウントであることを確認して、パブリック クライアント フローを許可するを はい に設定して保存します。

すべてのサービス > アプリの登録 > gravioexcel

gravioexcel | 認証

検索 (Cmd+F) < 保存 破棄 フィードバックがある場合

お時間があれば、フィードバックをお寄せください。 →

プラットフォーム構成

このアプリケーションが対象としているプラットフォームまたはデバイスによっては、リダイレクト URL、特定の認証設定、プラットフォームに特有のフィールドなど追加構成が必要となる場合があります。

+ プラットフォームを追加

サポートされているアカウントの種類

このアプリケーションを使用したりこの API にアクセスしたりできるのはだれですか？

☒ この組織ディレクトリのみに含まれるアカウント (Astoria Corporation のみ - シングル テナント)

☐ 任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント)

判断に役立つヘルプの表示...

サポートされている機能が一時的に異なるため、既存の登録に関して個人用 Microsoft アカウントを有効にしないでください。個人アカウントを有効にする必要がある場合、マニフェスト エディターを使用して有効にできます。 [これらの制限に関する詳細情報。](#)

詳細設定

パブリック クライアント フローを許可する ①

次のモバイルとデスクトップのフローを有効にする:

☒ はい ☐ いいえ

- アプリによってプレーンテキスト パスワードを収集する (リソース所有者のパスワード資格情報フロー) [詳細情報](#)
- キーボードなし (デバイス コード フロー) [詳細情報](#)
- ドメイン参加済みの Windows の SSO (Windows 統合認証フロー) [詳細情報](#)

プラットフォームの追加を選択してプラットフォームの構成は モバイルアプリケーションとデスクトップアプリケーションを選択します。

すべてのサービス > アプリの登録 > gravioexcel

gravioexcel | 認証

検索 (Cmd+F) < 保存 破棄 フィードバックがある場合

プラットフォーム構成

このアプリケーションが対象としているプラットフォームまたはデバイスによっては、リダイレクト URL、特定の認証設定、プラットフォームに特有のフィールドなど追加構成が必要となる場合があります。

+ プラットフォームを追加

サポートされているアカウントの種類

このアプリケーションを使用したりこの API にアクセスしたりできるのはだれですか？

☒ この組織ディレクトリのみに含まれるアカウント (Astoria Corporation のみ - シングル テナント)

☐ 任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント)

判断に役立つヘルプの表示...

サポートされている機能が一時的に異なるため、既存の登録に関して個人用 Microsoft アカウントを有効にしないでください。個人アカウントを有効にする必要がある場合、マニフェスト エディターを使用して有効にできます。 [これらの制限に関する詳細情報。](#)

詳細設定

パブリック クライアント フローを許可する ①

次のモバイルとデスクトップのフローを有効にする:

☒ はい ☐ いいえ

- アプリによってプレーンテキスト パスワードを収集する (リソース所有者のパスワード資格情報フロー) [詳細情報](#)
- キーボードなし (デバイス コード フロー) [詳細情報](#)
- ドメイン参加済みの Windows の SSO (Windows 統合認証フロー) [詳細情報](#)

プラットフォームの構成



Web アプリケーション



Web

Web サーバー アプリケーションをビルド、ホスト、デプロイします。.NET、Java、Python



シングルページ アプリケーション

ブラウザー クライアント アプリケーションとプログレッシブ Web アプリケーションを構成します。Javascript。

モバイル アプリケーションとデスクトップ アプリケーション



iOS または macOS

Objective-C、Swift、Xamarin



Android

Java、Kotlin、Xamarin



モバイル アプリケーションとデスクトップ アプリケーション

Windows、UWP、コンソール、IoT & 入力制限デバイス、クラシック iOS と Android

リダイレクトURIの最初の項目にチェックをして、カスタムリダイレクトURIに `https://localhost` と入力して構成ボタンを押して保存します。

デスクトップとデバイスの構成

✕

[すべてのプラットフォーム](#)[クイック スタート](#)[ドキュメント](#)

リダイレクト URI

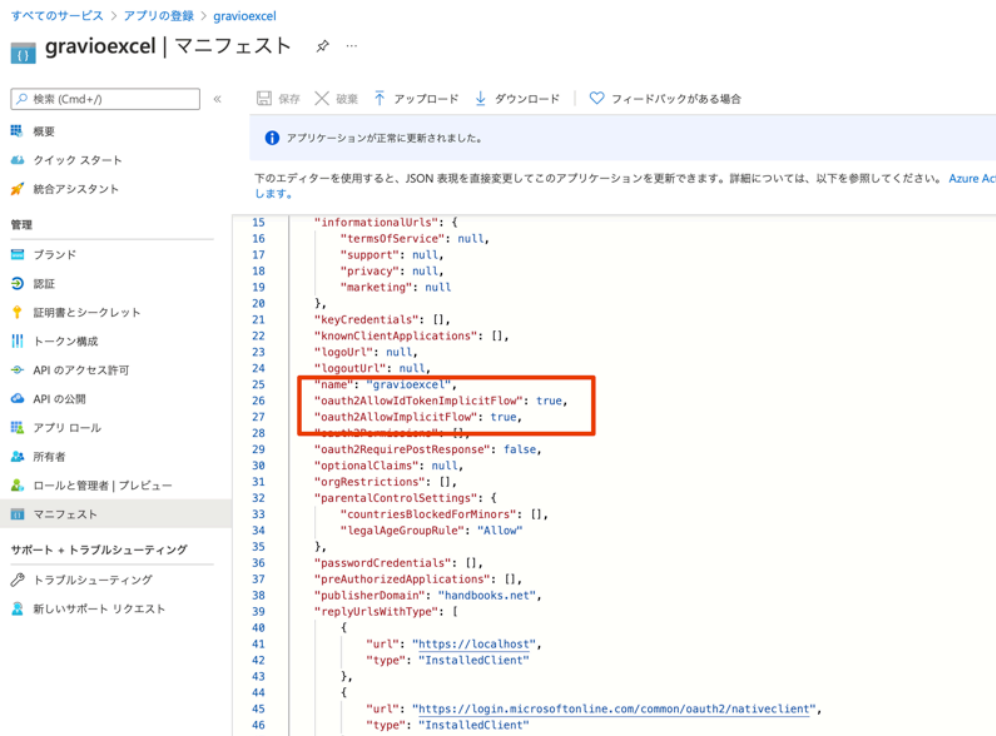
ユーザー認証に成功した後に認証応答 (トークン) を返すときに宛先として受け入れる URI。応答 URL とも呼ばれます。 [リダイレクト URI と制限の詳細情報](#)

- ☒ <https://login.microsoftonline.com/common/oauth2/nativeclient>
- ☐ https://login.live.com/oauth20_desktop.srf (LiveSDK)
- ☐ msal2c72660a2-791e-4378-b289-a2d23c212564://auth (MSAL のみ)

カスタム リダイレクト URI

✓

続いてマニフェストメニューから、`oauth2AllowIdTokenImplicitFlow` と `oauth2AllowImplicitFlow` を `true` に設定して保存します。



続いてAPIアクセスの許可メニューから、アクセス許可の追加をクリックしま、よく使用するMicrosoft APIは Microsoft Graph を選択します。

すべてのサービス > アプリの登録 > gravioexcel

gravioexcel | API のアクセス許可

検索 (Cmd+/) << 最新の情報に更新 フィードバックがある場合

概要

クイック スタート

統合アシスタント

管理

ブランド

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

アプリ ロール

所有者

ロールと管理者 | プレビュー

1 "管理者の同意が必要" 列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。組織の値が反映されていない場合があります。 [詳細情報](#)

構成されたアクセス許可

アプリケーションは、同意のプロセスの一種としてユーザーが管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成は、アプリケーションに必要なすべてのアクセス許可を含める必要があります。 [アクセス許可と同意に関する詳細情報](#)

+ アクセス許可の追加 ✓ Asteria Corporation に管理者の同意を与えます

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Microsoft Graph (1)				
User.Read	委任済み	Sign in and read user profile	いいえ	

アクセス許可とユーザーの同意を表示および管理するために、[エンタープライズ アプリケーション](#)をお試しください。

API アクセス許可の要求

✕

API を選択します

Microsoft API 所属する組織で使用している API 自分の API

よく使用される Microsoft API



Microsoft Graph

Office 365、Enterprise Mobility + Security、Windows 10 の大量のデータを活用しましょう。Azure AD、Excel、Intune、Outlook/Exchange、OneDrive、OneNote、SharePoint、Planner などに単一エンドポイント経由でアクセスできます。



Azure Rights Management Services

検証済みのユーザーに、保護されたコンテンツの読み取りと書き込みを許可します



Azure Service Management

Azure portal で利用できる機能の大部分へのプログラムによるアクセス



Data Export Service for Microsoft Dynamics 365

Microsoft Dynamics CRM 組織から外部宛先にデータをエクスポートします



Dynamics 365 Business Central

Dynamics 365 Business Central のデータと機能へのプログラムによるアクセス



Dynamics CRM

CRM ビジネス ソフトウェアと ERP システムの機能にアクセスします



Flow Service

フロー テンプレートの埋め込みとフローの管理

APIアクセス許可の要求は 委任されたアクセス許可 を選択し、アクセス許可を選択するには File と入力するとアクセス許可に Filesが表示されるので展開して Files.ReadWrite にチェックして構成ボタンを押します。

API アクセス許可の要求

[すべての API](#)

Microsoft Graph

<https://graph.microsoft.com/> [ドキュメント](#)

アプリケーションに必要なアクセス許可の種類

委任されたアクセス許可

アプリケーションは、サインインしたユーザーとして API にアクセスする必要があります。

アプリケーションの許可

アプリケーションは、サインインしたユーザーなしで、バックグラウンドサービスまたはデーモンとして実行されます。

アクセス許可を選択する

[すべて展開](#)

“管理者の同意が必要”列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この列には、ご自分の組織や、このアプリが使用される組織の値が反映されていない場合があります。 [詳細情報](#)

アクセス許可	管理者の同意が必要
▼ OpenId アクセス許可	
☐ profile ⓘ View users' basic profile	いいえ
▼ Files (1)	
☐ Files.Read ⓘ Read user files	いいえ
☐ Files.Read.All ⓘ Read all files that user can access	いいえ
☐ Files.Read.Selected ⓘ Read files that the user selects (preview)	いいえ
☒ Files.ReadWrite ⓘ Have full access to user files	いいえ
☐ Files.ReadWrite.All ⓘ Have full access to all files user can access	いいえ
☐ Files.ReadWrite.AppFolder ⓘ Have full access to the application's folder (nreview)	いいえ

API / アクセス許可の名前に `Files.ReadWrite` が表示されていることを確認して 管理者の同意を与えます をクリックします。確認のダイアログが表示されますので はい をクリックします。

[すべてのサービス](#) > [アプリの登録](#) > gravioexcel

gravioexcel | API のアクセス許可

[最新の情報に更新](#) | [フィードバックがある場合](#)

概要



クイックスタート



統合アシスタント



管理



ブランド



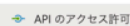
認証



証明書とシークレット



トークン構成



API のアクセス許可



API の公開



アプリ ロール



所有者



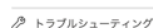
ロールと管理者 | プレビュー



マニフェスト



サポート + トラブルシューティング



トラブルシューティング

要求されたアクセス許可の管理者の同意が正常に付与されました。

“管理者の同意が必要”列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この組織の値が反映されていない場合があります。 [詳細情報](#)

構成されたアクセス許可

アプリケーションは、同意のプロセスの一環としてユーザーが管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成は、アプリケーションに必要なすべてのアクセス許可を含める必要があります。 [アクセス許可と同意に関する詳細情報](#)+ アクセス許可の追加 ☒ Asteria Corporation に管理者の同意を与えます

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
▼ Microsoft Graph (2)				
Files.ReadWrite	委任済み	Have full access to user files	いいえ	✓
User.Read	委任済み	Sign in and read user profile	いいえ	✓

アクセス許可とユーザーの同意を表示および管理するために、[エンタープライズアプリケーション](#)をお試しください。

すべてのサービス > アプリの登録 > gravioexcel

gravioexcel | API のアクセス許可

検索 (Cmd+/) < 最新の情報に更新 | フィードバックがある場合

概要

クイック スタート

統合アシスタント

管理

ブランド

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

アプリ ロール

所有者

ロールと管理者 | プレビュー

マニフェスト

サポート + トラブルシューティング

管理者の同意の確認を与えます。

Asteria Corporation のすべてのアカウントについて、要求されたアクセス許可に対する同意を付与しますか? この操作により、このアプリケーションが、以下の一覧の内容に一致するよう更新されます。

はい いいえ

構成されたアクセス許可

アプリケーションは、同意のプロセスの一環としてユーザーが管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成は、アプリケーションに必要なすべてのアクセス許可を含める必要があります。 [アクセス許可と同意に関する詳細情報](#)

+ アクセス許可の追加 ✓ Asteria Corporation に管理者の同意を与えます

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Microsoft Graph (2)				
Files.ReadWrite	委任済み	Have full access to user files	いいえ	
User.Read	委任済み	Sign in and read user profile	いいえ	

アクセス許可とユーザーの同意を表示および管理するために、[エンタープライズアプリケーション](#)をお試しください。

エンタープライズアプリケーション をクリックして、アクセス許可の 管理者に権限を与えます をクリックします。要求されているアクセス許可の組織のレビューが開きますので 承諾 をクリックします。

すべてのサービス > アプリの登録 > gravioexcel

gravioexcel | API のアクセス許可

検索 (Cmd+/) < 最新の情報に更新 | フィードバックがある場合

概要

クイック スタート

統合アシスタント

管理

ブランド

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

アプリ ロール

所有者

ロールと管理者 | プレビュー

マニフェスト

サポート + トラブルシューティング

アプリケーションに対するアクセス許可を編集しています。ユーザーは、既に同意したことがある場合でも同意が必要になります。

“管理者の同意が必要” 列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この組織の値が反映されていない場合があります。 [詳細情報](#)

構成されたアクセス許可

アプリケーションは、同意のプロセスの一環としてユーザーが管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成は、アプリケーションに必要なすべてのアクセス許可を含める必要があります。 [アクセス許可と同意に関する詳細情報](#)

+ アクセス許可の追加 ✓ Asteria Corporation に管理者の同意を与えます

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Microsoft Graph (2)				
Files.ReadWrite	委任済み	Have full access to user files	いいえ	
User.Read	委任済み	Sign in and read user profile	いいえ	

アクセス許可とユーザーの同意を表示および管理するために、[エンタープライズアプリケーション](#)をお試しください。

すべてのサービス > アプリの登録 > gravioexcel > gravioexcel

gravioexcel | アクセス許可

エンタープライズアプリケーション

概要 | デプロイ計画 | 管理

管理

- プロパティ
- 所有者
- ルールと管理者 (プレビュー)
- ユーザーとグループ
- シングル サインオン
- プロビジョニング
- アプリケーション プロキシ
- セルフサービス

セキュリティ

- 条件付きアクセス
- アクセス許可
- トークンの暗号化

アクティビティ

アクセス許可

組織とそのデータに対するアクセス許可をアプリケーションに付与するには、「管理者がすべてのユーザーのためにアプリケーションに同意する」、「ユーザーがアプリケーションを統合して、セルフサービス アクセスを有効にするかアプリケーションに直接ユーザーを割り当てる」の 3 つの方法を使用できます。[詳細について](#)

管理者はこのテナント内のすべてのユーザーに代わって同意を付与できます。これにより、エンド ユーザーはアプリケーションを使用するときに同意を求めらるには、下のボタンをクリックします。

Asteria Corporation に管理者の同意を与えます

管理者の同意 ユーザーの同意

🔍 アクセス許可の検索

API 名	↑↓ 権限	↑↓ 種類	↑↓
Microsoft Graph			
Microsoft Graph	Sign in and read user profile	Delegated	
Microsoft Graph	Have full access to user files	Delegated	

 Microsoft

admin@handbooks.net

要求されているアクセス許可 組織のレビュー

gravioexcel
handbooks.net

このアプリケーションは Microsoft によって公開されたものではありません。

このアプリに必要なアクセス許可:

- ✓ Sign in and read user profile
- ✓ Have full access to user files

同意すると、このアプリは組織内のすべてのユーザーの指定のリソースにアクセスできるようになります。これらのアクセス許可の確認を求めるメッセージは、他のユーザーには表示されません。

これらのアクセス許可を受け入れることは、サービス利用規約とプライバシーに関する声明で指定されているとおりこのアプリがデータを使用することを許可することを意味します。これらのアクセス許可は <https://myapps.microsoft.com> で変更できます。 [詳細の表示](#)

このアプリは疑わしいと思われますか? [こちらでご報告ください](#)

キャンセル **承諾**

これで事前準備は完了です。

14.15. LINE WORKS (旧版) のBot 登録方法

LINE WORKSコンポーネントを使用するためにLINE WORKSの管理者はBotの初期設定を行ってください。

LINE WORKS Developers

1. LINE WORKS Developers アカウントの作成

[こちら](#) からアカウントを作成して LINE WORKSのBotを作成します。

2. APIの設定

LINE WORKSコンポーネントはAPIを使用しますので、APIメニューを選択します。API 2.0 (beta)は使用しません。

3. API IDの発行

API IDの発行ボタンを押して作成します。作成されたAPI IDはLINE WORKSコンポーネントプロパティで使用します。

4. Server API Consumer Keyの発行

Server API Consumer Keyの発行ボタンを押して作成します。作成されたServer API Consumer KeyはLINE WORKSコンポーネントプロパティで使用します。

サーバAPIの利用範囲は追加/照会を選択します。Tokenの有効期間と自動延長は運用ルールに合わせて設定してください。テスト環境の場合には365日で自動更新に設定してください。

5. Server Listの追加

Server Listは固定IPタイプかID登録タイプのどちらかを使用します。

固定IPタイプを使用するにはGravio Hubkitがインターネットに接続され、グローバル固定IPアドレスがある場合に使用できます。

通常はID登録タイプを使用してください。ID登録タイプはIDと認証キーが発行されますので、LINE WORKSコンポーネントプロパティで使用します。

認証キーはダウンロードボタンを押してkeyファイルをダウンロードします。

6. Botの作成



BotメニューからBotの登録ボタンを押します。

Bot登録

Bot API の概要

* は必須項目

Botリストへ

Bot多国語対応

「プロフィール画像」

参照

* 画像の推奨サイズは640 x 640pxです。10MB未満のファイルのみ登録できます。
* ファイル名に特殊文字は使用できません。

* Bot名

100文字以内で入力してください。

0/100

* 「説明」

100文字以内で入力してください。

0/100

Bot No. (Bot Id)

Botの登録完了時に発行されます。

API Interface

☒ 従来のAPI ☐ API 2.0 (beta)

固定メニュー

[固定メニュー登録API](#) で設定します。

Callback URL

☐ On ☒ Off

Botポリシー

トークルームへの招待

☐ 複数人のトークルームに招待可
* 無効にするとBotとの1:1トークのみ可能になります。ただし、Botとのトークを開始するメンバーは権限が必要です。権限設定は管理者画面で行います。
* 有効にすると、複数人のトークルームにBotを招待できます。管理者画面での設定時には、全メンバーがこのBotを利用可能になります。特定メンバーのみに権限を付与することはできません。

* 「管理者」

* 主担当

名前またはメールアドレスを検索

副担当

名前またはメールアドレスを検索

保存

キャンセル

Bot名、説明を入力し、API Interfaceは従来のAPIを選択します。Botポリシーは複数人のトークルームに招待可にチェックを入れて、管理者の主担当を設定します。

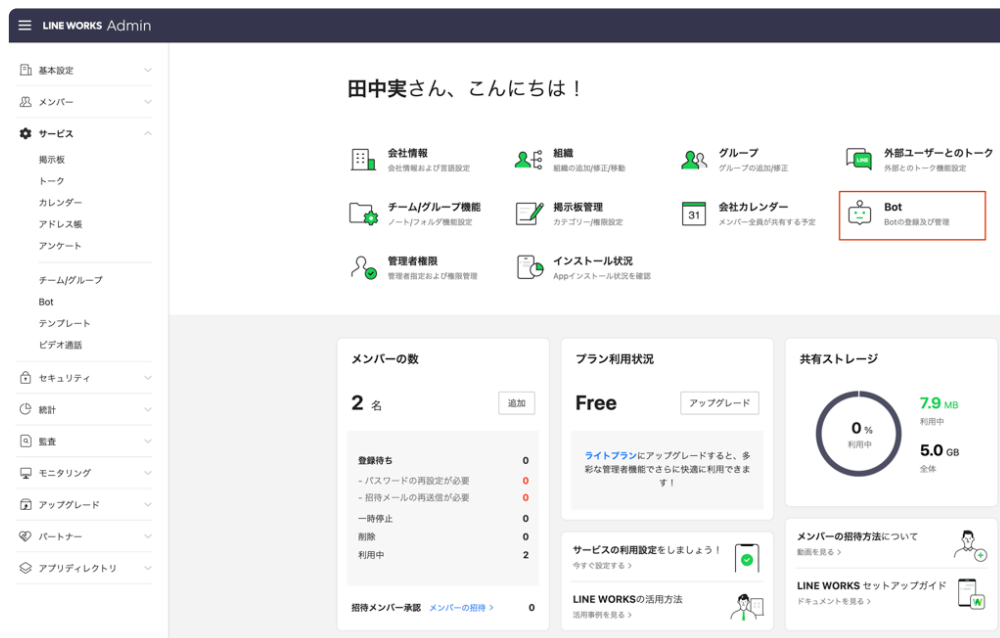
登録が完了するとBot No:が表示されますので、LINE WORKSコンポーネントプロパティで使用します。

LINE WORKS Admin

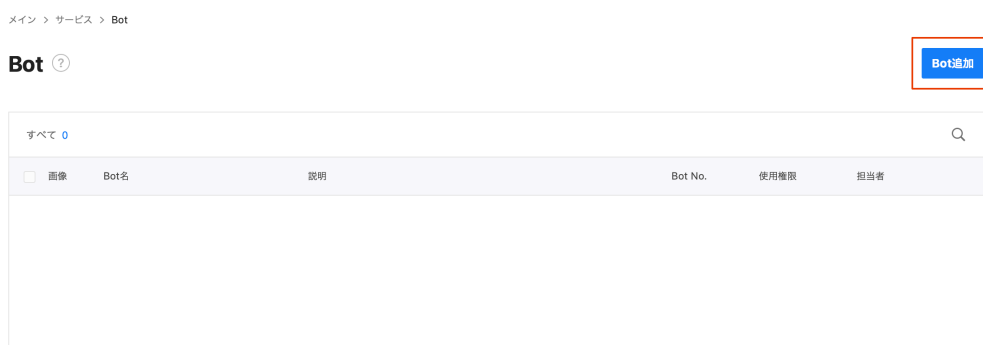
1. LINE WORKS 管理画面 へのログイン

管理者は [こちら](#) からログインして LINE WORKS管理画面を表示します。

2. Botの設定



サービスメニューよりBotを選択してBot追加ボタンを押します。



LINE WORKS Developers で作成したBotが表示されますので追加ボタンを押します。



Bot画面を再表示すると追加されたBotが表示されます。

メイン > サービス > Bot

Bot ?

Bot追加

すべて 1						🔍
☐ 画像	Bot名	説明	Bot No.	使用権限	担当者	
☐ 	gravio-bot	bot	141	すべて		

3. Botのトークルームへの追加

Botをトークルーム内で使用するには、新規にトークルームを作成するか、既存のトークルームに追加してください。

4. トークルームIDの確認

基本設定

メンバー

サービス

セキュリティ

統計

監査

管理者画面

掲示板

Drive

カレンダー

アドレス帳

タスク

アンケート

画面共有

ノート

トーク

通話

テンプレート

Developer Console

メイン > 監査 > トーク

トーク ?

ダウンロード

2021.11.24 - 2021.12.07 検索 Tokyo (GMT +09:00) 詳細

トーク	送信者	受信者	日時	トークルームID
				102343745
				102343745
				102343644
				102343644
				102343763
				102343763
				102343763
				102343763

LINE WORKSアプリやWebサイトでトークルームを作成した場合でもトークルームIDは表示されません。そこで、Botを使用するトークルームでメンバーで何かメッセージを送信してください。

しばらくして監査メニューよりトークを選択して検索するとトークルームIDが表示されますので、LINE WORKSコンポーネントプロパティで使用します。

これで事前準備は完了です。

14.16. LINE WORKSのBot 登録方法

LINE WORKSコンポーネントを使用するためにLINE WORKSの管理者はBotの初期設定を行ってください。

LINE WORKS Developers

1. LINE WORKS Developers アカウントの作成

[こちら](#) からアカウントを作成して LINE WORKSのBotを作成します。

2. アプリの作成

LINE WORKS2コンポーネントはAPI2.0を使用しますので、API2.0メニューからアプリを新規作成します。



アプリを作成するとClient ID、Client Secretが生成されますので、LINE WORKS2コンポーネントプロパティで使します。

Redirect URLは “http://127.0.0.1:48888/callback” を指定してください。LINE WORKS2コンポーネントプロパティで使します。

gravio

API ガイド

Client ID

コピー

Client Secret

コピー

Redirect URL

http://127.0.0.1:48888/callback

Service Account ?

発行

Service Accountは管理者と同じ権限が付与され、選択したScopeのすべてのリソースにアクセスできます。 Service Accountを発行すると、管理者にサービス通知が送信されます。

OAuth Scopes ?

管理

bot

- ・発行した Service Account は、このアプリのみで使用します。
- ・ Private Keyは、Service Account認証でAccess Token取得時に使用する重要なキーです。情報の管理に十分ご注意ください。
- ・ Private Key は 1 つのみ有効です。再発行の際、既存のPrivate Keyは利用できません。
- ・ Access Tokenの有効期間は発行から1日、Refresh Tokenの有効期限は発行から90日です。

変更

リストに戻る

削除

OAuth Scopesは 管理ボタンを押して、一覧からbotを選択してください。

6. Botの作成



BotメニューからBotの登録ボタンを押します。

Bot 修正

Bot API の概要

は必須項目

Botリストへ

Bot多国語対応

「プロフィール画像」	 参照 * 画像の推奨サイズは640 x 640pxです。10MB未満のファイルのみ登録できます。 * ファイル名に特殊文字は使用できません。
* Bot名	gravio-bot 10/100
* 「説明」	bot 3/100
Bot ID	
API Interface	☐ API 1.0 ☒ API 2.0 * API 1.0は、2023年4月30日をもってサービスの提供を終了する予定です。 API Interfaceおよび利用中のAPIをAPI 2.0に切り替えてください。 アップグレードガイド
固定メニュー	未設定
Callback URL	☐ On ☒ Off
Botポリシー	トークルームへの招待 ☒ 複数人のトークルームに招待可 * 無効にするとBotとの1:1トークのみ可能になります。ただし、Botとのトークを開始するメンバーは権限が必要です。権限設定は管理者画面で行います。 * 有効にすると、複数人のトークルームにBotを招待できます。管理者画面での設定時には、全メンバーがこのBotを利用可能になります。特定メンバーのみに権限を付与することはできません。
* 「管理者」	* 主担当 名前またはメールアドレスを検索
	副担当 名前またはメールアドレスを検索

保存

キャンセル

Bot削除

Bot名、説明を入力し、API InterfaceはAPI 2.0を選択します。Botポリシーは複数人のトークルームに招待可にチェックを入れて、管理者の主担当を設定します。

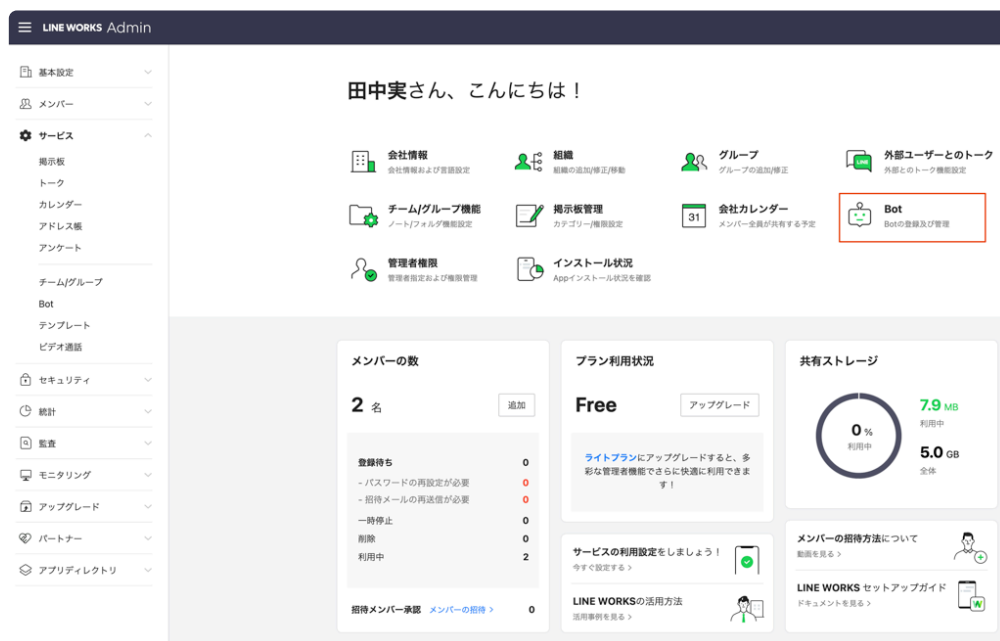
登録が完了するとBot No:が表示されますので、LINE WORKSコンポーネントプロパティで使用します。

LINE WORKS Admin

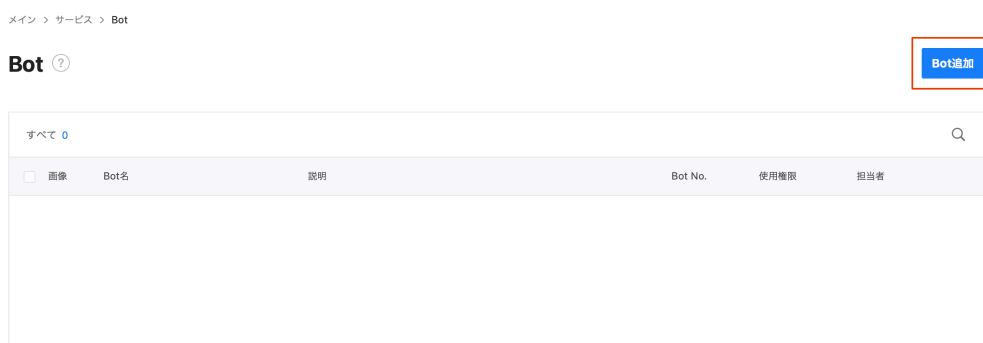
1. LINE WORKS 管理画面 へのログイン

管理者は [こちら](#) からログインして LINE WORKS管理画面を表示します。

2. Botの設定



サービスメニューよりBotを選択してBot追加ボタンを押します。



LINE WORKS Developers で作成したBotが表示されますので追加ボタンを押します。



Bot画面を再表示すると追加されたBotが表示されます。

メイン > サービス > Bot

Bot ?

Bot追加

すべて 1						🔍
☐ 画像	Bot名	説明	Bot No.	使用権限	担当者	
☐ 	gravio-bot	bot	141	すべて		

3. Botのトークルームへの追加

Botをトークルーム内で使用するには、新規にトークルームを作成するか、既存のトークルームに追加してください。

4. トークルームIDの確認

基本設定

メンバー

サービス

セキュリティ

統計

監査

管理者画面

掲示板

Drive

カレンダー

アドレス帳

タスク

アンケート

画面共有

ノート

トーク

通話

テンプレート

Developer Console

メイン > 監査 > トーク

トーク ?

ダウンロード

2021. 11. 24 - 2021. 12. 07

検索

Tokyo (GMT +09:00)

詳細

トーク	送信者	受信者	日時	トークルームID
				102343745
				102343745
				102343644
				102343644
				102343763
				102343763
				102343763
				102343763

LINE WORKSアプリやWebサイトでトークルームを作成した場合でもトークルームIDは表示されません。そこで、Botを使用するトークルームでメンバーで何かメッセージを送信してください。

しばらくして監査メニューよりトークを選択して検索するとトークルームIDが表示されますので、LINE WORKS2コンポーネントプロパティで使用します。

これで事前準備は完了です。

14.17. Gravio Hub, Gravio Zigbee レシーバー に関して

Gravio Hub、Gravio Zigbee レシーバーに接続できるデバイスは最大64個になります。

14.18. Gravio CO2 デバイスについて

Gravio CO2でCO2の値が当該環境で安定的な値を取得できるまでにデータを受信し出してから20分程度の時間がかかります。

CO2センサーの値が不自然に長時間高い値を出す場合

不自然に3000以上の値が継続して表示される場合は、外気がある場所でリセットを行なってみてください。（基準値の設定に戻るまで数週間かかることがあります）

リセット方法：18秒間ペアリング用のボタンを押し続ける。その後はなすと、成功すれば、白いLEDが約2秒間表示され、その後消灯します。

これでリセットされ、“CO2 の基準値 “に戻ります。

14.19. OAuth2.0でのGoogle Drive設定方法

Google Drive Upload/DownloadコンポーネントでのGoogle DriveへのOAuth2による認証についての設定方法を説明します。

OAuth2の設定を行うにはGoogle DriveではGCPを使用します。
設定を正しく行うにはGCPの正しい知識が必要となります。
詳細な設定方法がご不明な場合にはGCPはGoogle社までお問い合わせください。

1. Google Driveへのアクセス設定方法

設定方法の流れ

GCPで以下の手順で設定を行います。

1-1. APIサービスのライブラリからGoogle Drive APIを有効にします。

APIサービスメニューからライブラリを選択します。

Google Drive APIを選択して有効にしてください。

1-2. OAuth同意画面メニューから「外部」を選択して作成ボタンを押します。

API	API とサービス	OAuth 同意画面
🔗	有効な API とサービス	
📖	ライブラリ	
🔑	認証情報	
👤	OAuth 同意画面	アプリをどのように構成および登録するか（ターゲット ユーザーを含む）を選択します。プロジェクトに関連付けることができるアプリは 1 つだけです。 User Type ☐ 内部 ⓘ 組織内のユーザーのみが使用できます。確認を受けるためにアプリを送信する必要はありません。 ユーザーの種類の詳細 ☒ 外部 ⓘ Google アカウントを持つすべてのテストユーザーが使用できます。アプリはテストモードで起動し、アプリを使用できるのは、テストユーザーのリストに追加されたユーザーに限られます。アプリを本番環境に移す準備ができれば、アプリの確認が必要となる場合があります。 ユーザーの種類の詳細 作成
📄	ページの使用に関する契約	

1-3. テストユーザーではGoogle アカウントのメールアドレスを登録してください。

1-4. 概要で登録した内容を確認します。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

アプリ登録の編集

OAuth 同意画面

スコープ

テストユーザー

概要

OAuth 同意画面

編集

ユーザーの種類

外部

アプリ名

gravio

サポートメール

アプリのロゴ

指定されていません

[アプリケーション ホームページ] リンク

指定されていません

[アプリケーション プライバシー ポリシー] リンク

指定されていません

[アプリケーション利用規約] リンク

指定されていません

承認済みドメイン

指定されていません

連絡先メールアドレス

1-5.認証情報メニューから認証情報を作成をクリックしてOAuth 2.0クライアントIDを選択します。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

認証情報

+ 認証情報を作成

削除

削除した認証情報を復元

有効な API にアクセスするための認証情報を作成します。 [詳細](#)

API キー

名前

作成日 ↓

制限

表示する API キーがありません

OAuth 2.0 クライアント ID

名前

作成日 ↓

種類

表示する OAuth クライアントがありません

1-7.アプリケーションの種類は“Webアプリケーション“を選択して名前をつけてください。認証済みのリダイレクトURIには“http://127.0.0.1:48888/callback“を追加して保存してください。

API APIとサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

← OAuth クライアント ID の作成

アプリケーションの種類 *

ウェブ アプリケーション

名前 *

ウェブクライアント 1

下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#)として自動で追加されます。

承認済みの JavaScript 生成元

ブラウザからのリクエストに使用します

+ URI を追加

承認済みのリダイレクト URI

ウェブサーバーからのリクエストに使用します

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

作成

キャンセル

1-8. 認証情報メニューからOAuth2.0クライアントID2保存したデータを確認するとクライアントID、クライアントシークレットIDと承認済みのリダイレクトURIが表示されますので、コンポーネントで使用するためメモしてください。

API APIとサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

← ウェブ アプリケーション のクライアント ID

名前 *

ウェブクライアント 1

下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#)として自動で追加されます。

承認済みの JavaScript 生成元

ブラウザからのリクエストに使用します

+ URI を追加

承認済みのリダイレクト URI

ウェブサーバーからのリクエストに使用します

URI 1 *

http://127.0.0.1:48898/callback

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

保存

キャンセル

クライアント ID

作成日

クライアント シークレット

クライアント シークレット

作成日

ステータス

+ ADD SECRET

※上記設定では公開ステータスは“テスト”となり、テストユーザーにのみ使用できるモードとなっています。このモードを“公開”することでテストユーザー以外からも使用できるようになります。詳細な設定方法がご不明な場合にはGoogle社までお問い合わせください。個別のGCPの設定方法についてはサポート出来ない場合がございます。

これでOAuth 2.0に必要なクライアントID、クライアントシークレットIDと承認済みのリダイレクトURIがそろい事前準備は完了です。

Copyright © 2023 ASTERIA Corporation

ページ 594 / 633

14.20. Gmailでのメール送信設定方法：基本認証（SMTP）

gmailのメールアドレスを使用してメール送信する場合の設定例

項目	説明
メールサーバー	smtp.gmail.com
ポート番号	587
送信メールアドレス	gmailメールアドレス
SMTPユーザ名	ユーザ名（gmailメールアドレスの@より左側）
パスワード	アカウントのパスワード
送信先メールアドレス	メール通知を受信するメールアドレス

※ gmailの設定で以下の設定を行う必要があります。

2段階認証プロセスを“オン”にして“アプリ パスワード”を作成して、そのパスワードを指定します。

<https://support.google.com/accounts/answer/185833?hl=ja&authuser=1>

← アプリ パスワード

アプリ パスワードを使用すると、2段階認証プロセスに対応していないデバイス上のアプリから Google アカウントにログインできるようになります。このパスワードは一度入力すれば、以降は覚えておく必要はありません。 [詳細](#)

アプリ パスワードがありません。

アプリ パスワードを生成するアプリとデバイスを選択してください。

Gravio

生成

- アプリの選択は「メール」を選び、デバイス名は“Gravio”などと任意の値を入力してください。

14.21. OAuth2.0でのメール送信設定方法：先進認証（OAuth2）

Mailコンポーネントでのメール送信における先進認証（OAuth2）による認証方式について設定方法を説明します。

先進認証（OAuth2）による認証方式で動作確認しているのはGmailとOutlookとなります。

OAuth2の設定を行うにはGmailではGCPを使用し、OutlookではAzureを使用します。

設定を正しく行うにはGCPまたはAzureの正しい知識が必要となります。

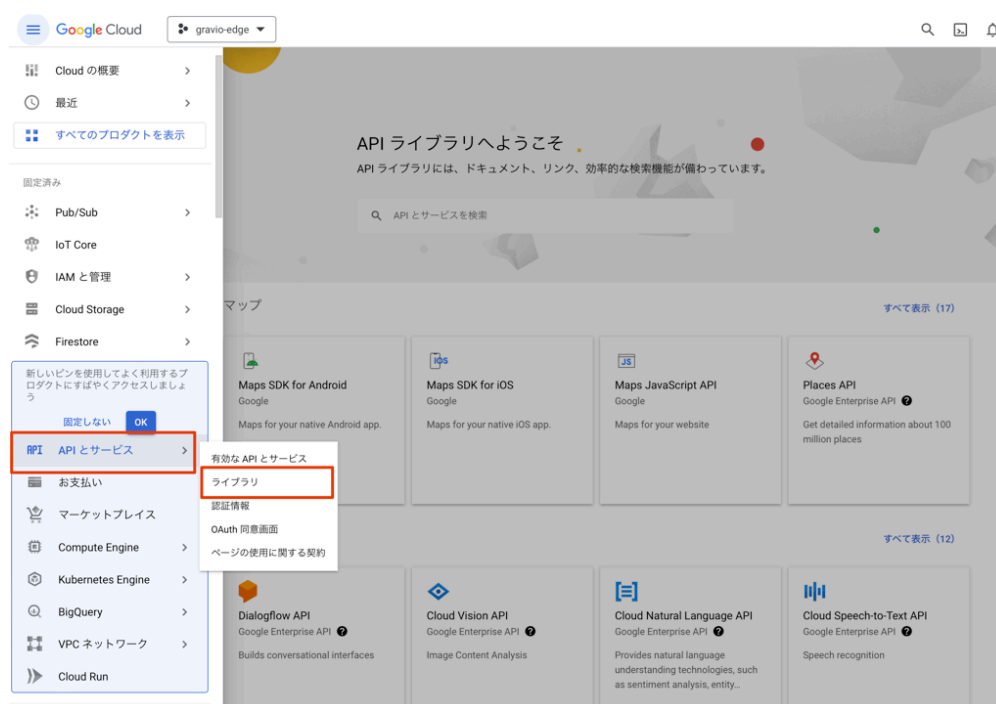
詳細な設定方法がご不明な場合にはGCPはGoogle社まで、AzureはMincrosoft社までお問い合わせください。

1. Gmailでのメール送信設定方法

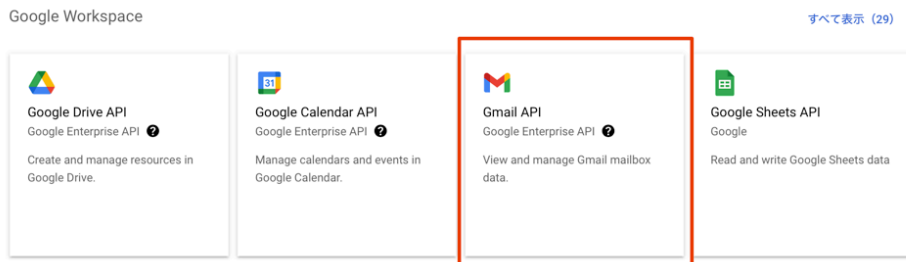
設定方法の流れ

GCPで以下の手順で設定を行います。

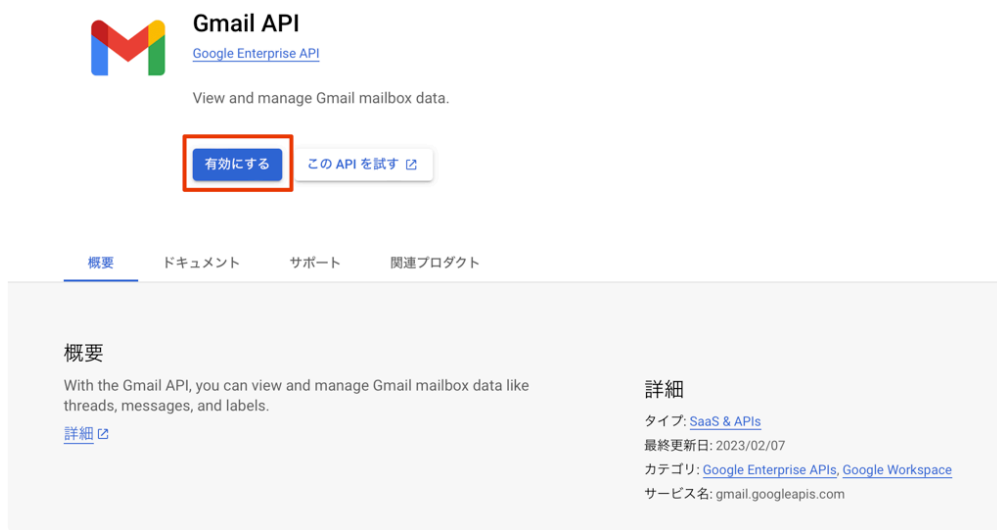
1-1. APIサービスのライブラリからGMail APIを有効にします。



APIサービスメニューからライブラリを選択します。



Gmail APIを選択します。



Gmail APIを有効にします。

1-2. OAuth同意画面メニューから「外部」を選択して作成ボタンを押します。



1-3. スコープではGMailのスコープを登録します。

API	API とサービス	アプリ登録の編集
🔗	有効な API とサービス	✓ OAuth 同意画面 — 2 スコープ — 3 テストユーザー — 4 概要
📦	ライブラリ	
🔑	認証情報	
🔗	OAuth 同意画面	
📄	ページの使用に関する契約	

スコープとは、アプリのユーザーに許可を求める権限を表します。スコープを定めることで、プロジェクトからユーザーの Google アカウントにある特定の種類のプライベートなユーザーデータへのアクセスが可能になります。 [詳細](#)

[スコープを追加または削除](#)

非機密のスコープ

API ↑	範囲	ユーザー向けの説明
表示する行がありません		

🔒 機密性の高いスコープ

機密性の高いスコープとは、プライベート ユーザーデータへのアクセスをリクエストするスコープです。

API ↑	範囲	ユーザー向けの説明
表示する行がありません		

🔒 制限付きのスコープ

制限付きのスコープとは、機密性の高いユーザーデータへのアクセスをリクエストするスコープです。

API ↑	範囲	ユーザー向けの説明
表示する行がありません		

[保存して次へ](#) [キャンセル](#)

スコープはGmail “Gmail のすべてのメールの閲覧、作成、送信、完全な削除”など必要な権限を指定してください。

1-4. テストユーザーではGMailのメール送信を行うGmailのメールアドレスを登録します。

1-5. 概要で登録した内容を確認します。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

アプリ登録の編集

OAuth 同意画面

スコープ

テストユーザー

概要

OAuth 同意画面

編集

ユーザーの種類

外部

アプリ名

gravio

サポートメール

アプリのロゴ

指定されていません

[アプリケーション ホームページ] リンク

指定されていません

[アプリケーション プライバシー ポリシー] リンク

指定されていません

[アプリケーション利用規約] リンク

指定されていません

承認済みドメイン

指定されていません

連絡先メールアドレス

1-6.認証情報メニューから認証情報を作成をクリックして**OAuth 2.0**クライアントIDを選択します。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

認証情報

+ 認証情報を作成

削除

削除した認証情報を復元

有効な API にアクセスするための認証情報を作成します。 [詳細](#)

API キー

名前作成日 ↓制限

表示する API キーがありません

OAuth 2.0 クライアント ID

名前作成日 ↓種類

表示する OAuth クライアントがありません

1-7.アプリケーションの種類は“**Webアプリケーション**“を選択して名前をつけてください。認証済みのリダイレクトURIには“**http://127.0.0.1:48888/callback**“を追加して保存してください。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

← OAuth クライアント ID の作成

クライアント ID は、Google の OAuth サーバーで個々のアプリを識別するために使用します。アプリが複数のプラットフォームで実行される場合、それぞれに独自のクライアント ID が必要です。詳しくは、[OAuth 2.0 の設定](#) をご覧ください。OAuth クライアントの種類の[詳細](#)

アプリケーションの種類 *
ウェブ アプリケーション

名前 *
ウェブ クライアント 1

OAuth 2.0 クライアントの名前。この名前はコンソールでクライアントを識別するためにのみ使用され、エンドユーザーには表示されません。

下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#) として自動で追加されます。

承認済みの JavaScript 生成元 ⓘ
ブラウザからのリクエストに使用します

+ URI を追加

承認済みのリダイレクト URI ⓘ
ウェブサーバーからのリクエストに使用します

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

作成 キャンセル

1-8. 認証情報メニューからOAuth2.0クライアントID2保存したデータを確認するとクライアントID、クライアントシークレットIDと承認済みのリダイレクトURIが表示されますので、コンポーネントで使用するためメモしてください。

API API とサービス

有効な API とサービス

ライブラリ

認証情報

OAuth 同意画面

ページの使用に関する契約

← ウェブ アプリケーション のクライアント ID 削除

名前 *
ウェブ クライアント 1

OAuth 2.0 クライアントの名前。この名前はコンソールでクライアントを識別するためにのみ使用され、エンドユーザーには表示されません。

下で追加する URI のドメインは、[OAuth 同意画面に承認済みドメイン](#) として自動で追加されます。

承認済みの JavaScript 生成元 ⓘ
ブラウザからのリクエストに使用します

+ URI を追加

承認済みのリダイレクト URI ⓘ
ウェブサーバーからのリクエストに使用します

URI 1 *
<http://127.0.0.1:48898/callback>

+ URI を追加

注: 設定が有効になるまで 5 分から数時間かかることがあります

保存 キャンセル

クライアント ID

作成日

クライアント シークレット

クライアント シークレットの変更処理中である場合は、ダウンタイムを発生させずに手動でシークレットをローテーションできます。[詳細](#)

クライアント シークレット

作成日

ステータス
有効化済み

+ ADD SECRET

※上記設定では公開ステータスは“テスト”となり、テストユーザーにのみメールが送信できるモードとなっています。このモードを“公開”することでテストユーザー以外のメールアドレスからもメールが送信できるようになります。詳細な設定方法がご不明な場合にはGoogle社までお問い合わせください。個別のGCPの設定方法についてはサポート出来ない場合がございます。

これでOAuth 2.0に必要なクライアントID、クライアントシークレットIDと承認済みのリダイレクトURIがそろい事前準備は完了です。

2. Outlookでのメール送信設定方法

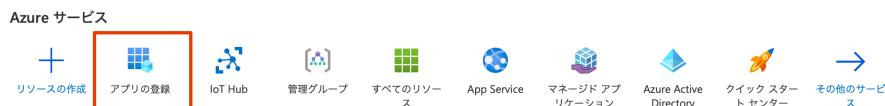
OutlookでOAuth2.0を使用するには、Microsoft365 for Businessの契約が必要です。

設定方法の流れ

Azureで以下の手順で設定を行います。

メール送信時に MailSend: failed to send mail: 535 5.7.139 Authentication unsuccessful, SmtplibClientAuthentication is disabled for the Tenant. Visit https://aka.ms/smtplib_auth_disabled for more information. このようなエラーが発生する場合には、2-9以降で Microsoft 365 管理センターで認証済みSMTPの設定を有効にしてください。

2-1. Azureサービスでアプリの登録をします。



アプリの作成ボタンをクリックします。



新規登録ボタンをクリックします。

ホーム > アプリの登録 >

アプリケーションの登録

* 名前

このアプリケーションのユーザー向け表示名 (後で変更できます)。

gravio-outlook

サポートされているアカウントの種類

このアプリケーションを使用したりこの API にアクセスしたりできるのはどれですか?

☐ この組織ディレクトリのみに含まれるアカウント (Synon,Inc. のみ - シングル テナント)

☐ 任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント)

☒ 任意の組織ディレクトリ内のアカウント (任意の Azure AD ディレクトリ - マルチテナント) と個人の Microsoft アカウント (Skype、Xbox など)

☐ 個人用 Microsoft アカウントのみ

選択に関する詳細...

リダイレクト URI (省略可能)

ユーザー認証が成功すると、この URI に認証応答を返します。この時点での指定は省略可能で、後ほど変更できますが、ほとんどの認証シナリオで値が必要となります。

Web

http://localhost:48888/callback

アプリケーションの名前、サポートされているアカウントの情報とリダイレクトURIを指定します。リダイレクトURIには“http://localhost:48888/callback“を指定します。

2-2. アプリが登録されると概要に基本情報が表示されます。

ホーム > アプリの登録 >

gravio-outlook

検索

削除

エンドポイント

プレビュー機能

概要

クイック スタート

統合アシスタント

管理

ブランド化とプロパティ

認証

へ 基本

表示名 : gravio-outlook

アプリケーション (クライアント) :

オブジェクト ID :

ディレクトリ (テナント) ID :

サポートされているアカウ... : すべての Microsoft アカウント ユーザー

クライアントの資格情報 : 証明書またはシークレットの追加

リダイレクト URI : 1 個の Web、0 個の SPA、0 個のパブリック クライアント

アプリケーション ID の URI : アプリケーション ID URI の追加

ローカル ディレクトリで... : gravio-outlook

2-3. APIのアクセス許可メニューからアクセス許可の追加をクリックします。

ホーム > gravio-outlook

gravio-outlook | API のアクセス許可

検索

最新の情報に更新

フィードバックがある場合

概要

クイック スタート

統合アシスタント

管理

ブランド化とプロパティ

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

構成されたアクセス許可

アプリケーションは、同意のプロセスの一環としてユーザーや管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成されたアクセス許可の一覧には、アプリケーションに必要なすべてのアクセス許可を含める必要があります。アクセス許可と同意に関する詳細情報

+ アクセス許可の追加

に管理者の同意を与えます

API / アクセス許可の名前	種類	説明	管理者の同意が必要	状態
Microsoft Graph (1)				
User.Read	委任済み	Sign in and read user profile	いいえ	...

個々のアプリに関する同意済みのアクセス許可とテナントの同意設定を表示および管理するには、エンタープライズ アプリケーションお試しください。

2-4. Microsoft APIのMicrosoft Graphを選択して、Mail.Sendとoffline_accessにチェックをして追加します。

API アクセス許可の要求

✕

API を選択します

Microsoft API 所属する組織で使用している API 自分の API

よく使用される Microsoft API

**Microsoft Graph**
Office 365、Enterprise Mobility + Security、Windows 10 の大量のデータを活用しましょう。Azure AD、Excel、Intune、Outlook/Exchange、OneDrive、OneNote、SharePoint、Planner などに単一エンドポイント経由でアクセスできます。

**Azure Service Management**
Azure portal で利用できる機能の大部分へのプログラムによるアクセス

**Dynamics 365 Business Central**
Dynamics 365 Business Central のデータと機能へのプログラムによるアクセス

**Office 365 Management APIs**
Office 365 と Azure AD のアクティビティ ログからユーザー、管理者、システム、ポリシーのアクションとイベントに関する情報を取得します

**OneNote**
OneNote ノートブックでノート、リスト、画像、ファイルなどを作成して管理します

**Power BI Service**
Power BI のデータセット、テーブル、行などのダッシュボード リソースへのプログラムによるアクセス

**SharePoint**
SharePoint データとリモートで対話します

API アクセス許可の要求

[すべての API](#)

Microsoft Graph

<https://graph.microsoft.com/> [ドキュメント](#)

アプリケーションに必要なアクセス許可の種類

委任されたアクセス許可

アプリケーションは、サインインしたユーザーとして API にアクセスする必要があります。

アプリケーションの許可

アプリケーションは、サインインしたユーザーなしで、バックグラウンドサービスまたはデーモンとして実行されます。

アクセス許可を選択する

[すべて展開](#)

"管理者の同意が必要"列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この列には、ご自分の組織や、このアプリが使用される組織の値が反映されていない場合があります。 [詳細情報](#)

アクセス許可	管理者の同意が必要
OpenId アクセス許可	
☐ email ^① View users' email address	いいえ
MailboxSettings	
Mail (1)	
☐ Mail.Read ^① Read user mail	いいえ
☐ Mail.Read.Shared ^① Read user and shared mail	いいえ
☐ Mail.ReadBasic ^① Read user basic mail	いいえ
☐ Mail.ReadBasic.Shared ^① Read user and shared basic mail	いいえ
☐ Mail.ReadWrite ^① Read and write access to user mail	いいえ
☐ Mail.ReadWrite.Shared ^① Read and write user and shared mail	いいえ
☒ Mail.Send ^① Send mail as a user	いいえ
☐ Mail.Send.Shared ^① Send mail on behalf of others	いいえ

API アクセス許可の要求

すべての API

Microsoft Graph

https://graph.microsoft.com/ [ドキュメント](#)

アプリケーションに必要なアクセス許可の種類

委任されたアクセス許可

アプリケーションは、サインインしたユーザーとして API にアクセスする必要があります。

アプリケーションの許可

アプリケーションは、サインインしたユーザーなしで、バックグラウンドサービスまたはデーモンとして実行されます。

アクセス許可を選択する

すべて展開

off

“管理者の同意が必要”列には、組織の既定値が表示されます。ただし、ユーザーの同意は、アクセス許可、ユーザー、アプリごとにカスタマイズできます。この列には、ご自分の組織や、このアプリが使用される組織の値が反映されていない場合があります。 [詳細情報](#)

アクセス許可

管理者の同意が必要

OpenId アクセス許可 (1)

offline_access

Maintain access to data you have given it access to

いいえ

2-5. 保存されたら構成されたアクセス許可に表示されます。組織に管理者の同意を与えますをクリックして同意を付与します。

ホーム > アプリの登録 > gravio-outlook

gravio-outlook | API のアクセス許可

検索

最新の情報に更新

フィードバックがある場合

概要

クイックスタート

統合アシスタント

管理

ブランド化とプロパティ

認証

証明書とシークレット

トークン構成

API のアクセス許可

API の公開

アプリ ロール

所有者

構成されたアクセス許可

アプリケーションは、同意のプロセスの一環としてユーザーが管理者からアクセス許可が付与されている場合、API を呼び出すことが承認されます。構成されたアクセス許可の一覧には、アプリケーションに必要なすべてのアクセス許可を含める必要があります。 [アクセス許可と同意に関する詳細情報](#)

+

アクセス許可の追加

に管理者の同意を与えます

API / アクセス許可の名前

種類

説明

管理者の同意が必要

詳細

Microsoft Graph (3)

Mail.Send

委任済み

Send mail as a user

いいえ

に付与されま...

offline_access

委任済み

Maintain access to data you have given it access to

いいえ

に付与されま...

User.Read

委任済み

Sign in and read user profile

いいえ

に付与されま...

個々のアプリに関する同意済みのアクセス許可とテナントの同意設定を表示および管理するには、エンタープライズアプリケーションお試しください。

2-6. クライアントIDとリダイレクトURIをメモします。

Copyright © 2023 ASTERIA Corporation

ページ 605 / 633



2-7. 証明書とシークレットメニューから新しいクライアントシークレットをクリックして作成します。



クライアントシークレットを作成します。



クライアントシークレットには有効期限が設定出来ますので指定します。

2-8. クライアントシークレットを表示します。

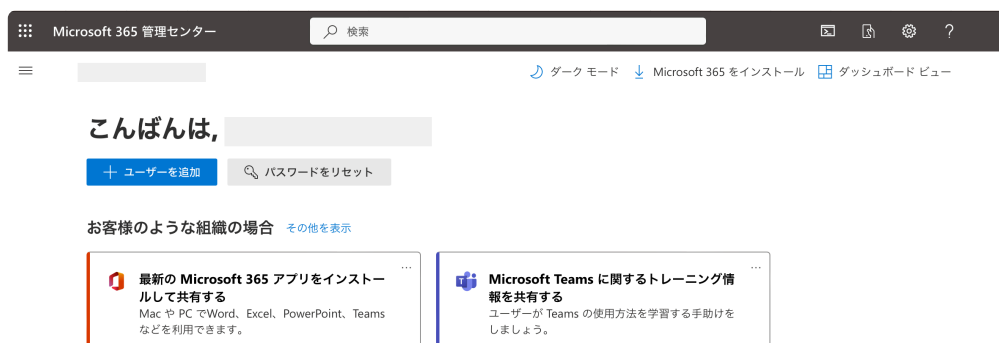


値がクライアントシークレットとなりますのでメモします。

これでOAuth 2.0に必要なクライアントID、クライアントシークレットとリダイレクトURIがそろい事前準備は完了です。

2-9. Microsoft 365 管理センターで認証済みSMTPの設定を有効にする。

Microsoft 365 管理センター



左のメニューから アクティブユーザーを選択します。



目的のユーザーを選択すると、右側にパネルが開きます。



メールタブにある メールアプリを管理する を選択します。



認証済みSMTP を有効にして保存してください。設定した内容はすぐには反映されないためメール送信で引き続き同じエラーが発生する場合にはしばらく待ってから再度実行してください。引き続きエラーが発生するようでしたらもう一度設定して内容を確認してください。

14.22. ファイルパスの記述方法

コンポーネントプロパティに指定するファイルは以下のような記述でファイルパスを指定することができます。

ファイルパスを指定しないファイルの参照/保存先

ファイルパスを指定しないファイルの参照/保存先は以下となります。

Windows: C:\ProgramData\HubKit\action\actmgr\data\

Mac: /Library/Application Support/HubKit/action/scripts/actmgr/data/

Linux: /var/opt/hubkit/action/scripts/actmgr/data/

1. dataフォルダ内のファイル

・Windows以外の場合

ファイルパス	説明
sample.txt	actmgr/dataフォルダー内の sample.txt を指定しています
./sample.txt	actmgr/dataフォルダー内の sample.txt を指定しています

・Windowsの場合

ファイルパス	説明
sample.txt	actmgr/dataフォルダー内の sample.txt を指定しています
¥¥sample.txt	actmgr/dataフォルダー内の sample.txt を指定しています

ファイルパスを指定するファイルの参照/保存先

2. dataフォルダからの相対パスのファイル

・Windows以外の場合

ファイルパス	説明
../../../../gravio/sample.txt	actmgr/dataフォルダーから4階層上の gravio/sample.txt を指定しています

・Windowsの場合

ファイルパス	説明
..¥¥../../../../gravio/sample.txt	actmgr/dataフォルダーから4階層上の gravio¥¥sample.txt を指定しています

3. 絶対パスのファイル

・Windows以外の場合

ファイルパス	説明
/home/gravio/sample.txt	ルート下の home/gravio/sample.txt を指定しています

・Windowsの場合

ファイルパス	説明
c:¥¥temp¥¥sample.txt	c:ドライブ下の temp¥¥sample.txt を指定しています

※例では sample.txt ファイルを読み込むことを想定しております。

14.23. Gravioコンフィグレーションマネージャーの自動更新

Gravio HubにインストールされているGravioコンフィグレーションマネージャー、Linux版、ARM64版(Rapsberry)のGravioコンフィグレーションマネージャーは毎日AM02:00に自動更新設定がされており、更新作業は必要ありません。またGravio HubにインストールされているGravioコンフィグレーションマネージャーはGravio Hubを再起動するタイミングでも自動更新されます。

14.24. ソフトウェアセンサーを利用する場合のハードウェア環境

推論ファイルを使用する時のパフォーマンスの目安

Intel i7のCPUを使用しているPCの場合、2台のONVIFカメラを使用して10秒以上の間隔で画像保存を設定して動作することを検証しています。

メモリの搭載量や外部記憶装置がSSDの場合やI/O処理によりご使用頂く環境で必ず動作することを保証するものではありませんが目安としてください。

また、ATOMチップセットを使用しているPCの場合は1台のカメラで20秒以上の間隔で画像保存を設定して動作することを検証しています。

ONVIF対応カメラは **ONVIF Profile T** をサポートしているデバイスに限ります。

CPU/チップセット	ONVIFカメラで推論ファイルを使用した場合の画像保存の設定(目安)
i7	2台のカメラで10秒以上の間隔
ATOM	1台のカメラで20秒以上の間隔

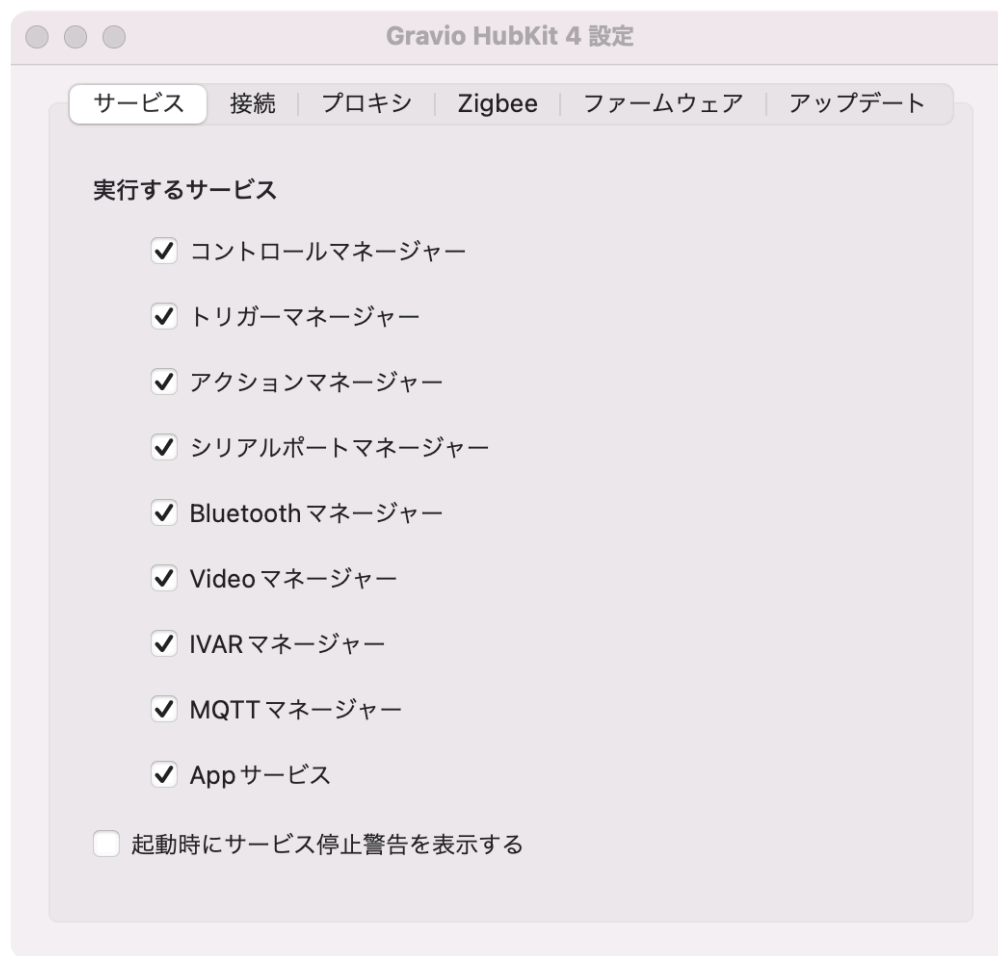
14.25. HubKitの設定

HubKitの設定方法を示します。

1. Macを使用する場合

macOSの上段のメニューバーのHubKitアイコンから「設定」メニューを選択します。

1. サービス



HubKitを構成するサービスを個別に起動・停止する場合に使用します。

HubKitを起動するときにサービスが停止している場合に警告を表示するには「起動時にサービス停止警告を表示する」を選択します。

2. 接続



HubKitに接続するGravio Studioを制限する場合には「クライアント認証を利用する」を選択します。
また証明書のインストール方法は [Gravio StudioからHubへのアクセス制限をかけたい場合](#) を参照してください。

※HubKitが実行されている場合には停止してください。

3. プロキシ

The image shows a screenshot of the 'Gravio HubKit 4 設定' (Gravio HubKit 4 Settings) window. The window has a title bar with three window control buttons on the left and the title 'Gravio HubKit 4 設定' in the center. Below the title bar is a tabbed interface with five tabs: 'サービス' (Service), '接続' (Connection), 'プロキシ' (Proxy), 'Zigbee', 'ファームウェア' (Firmware), and 'アップデート' (Update). The 'プロキシ' (Proxy) tab is currently selected and highlighted. Inside this tab, there are two sections. The first section is titled 'HTTPS プロキシを使用する' (Use HTTPS Proxy) and has an unchecked checkbox. Below this checkbox is a form with three input fields: 'プロキシURL:' (Proxy URL), 'ユーザー名:' (Username), and 'パスワード:' (Password). The second section is titled 'HTTP プロキシを使用する' (Use HTTP Proxy) and also has an unchecked checkbox. Below this checkbox is another form with three input fields: 'プロキシURL:', 'ユーザー名:', and 'パスワード:'. At the bottom of the window, there is a note: '* この設定を変更するにはHubKitを停止してください' (To change this setting, please stop HubKit). To the right of this note is a '保存' (Save) button.

HubKitがインターネットサービスを使用する場合にProxyサーバーを経由する必要がある場合に設定します。HTTPSとHTTPのプロトコルでそれぞれに使用するProxyサーバーのアカウント情報（プロキシURL、ユーザー名、パスワード）を入力します。

※この項目を設定するためにはHubKitは停止している必要があります。

4. Zigbee

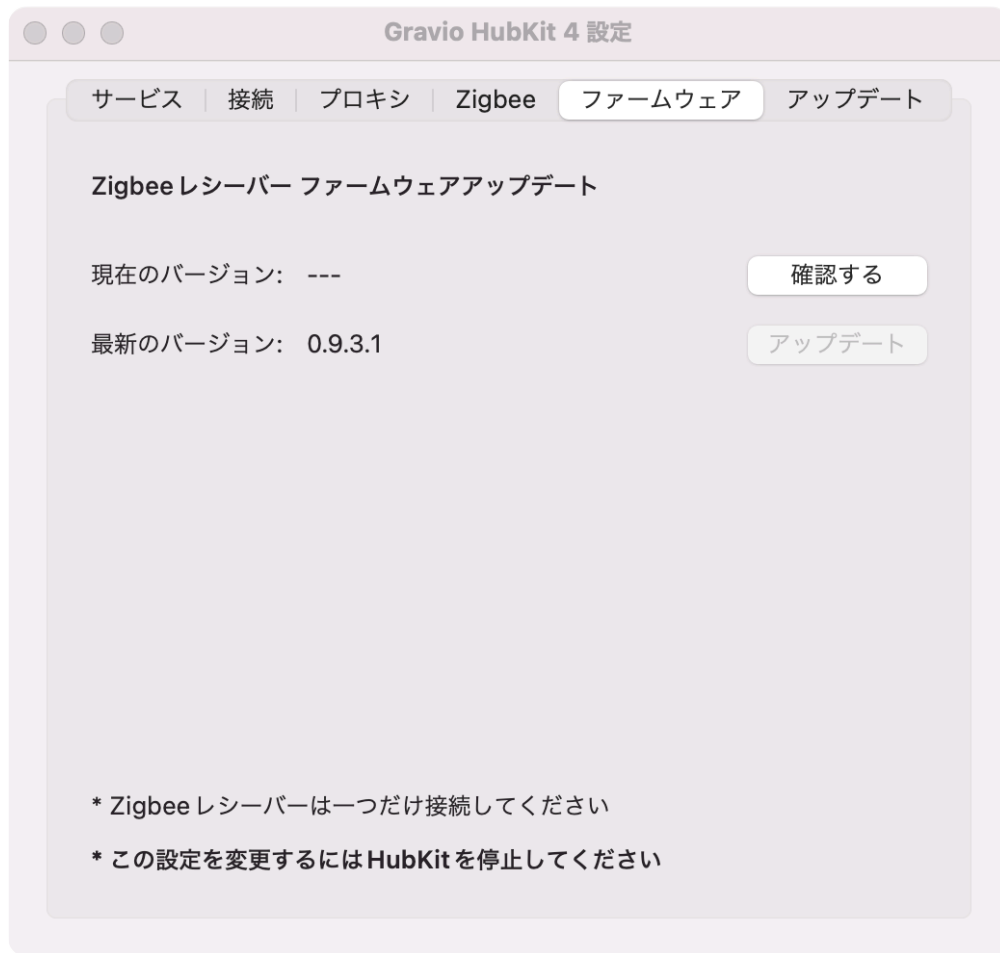


Zigbeeの dongle のセンサーとのペアリング情報をバックアップする場合や別の dongle に引き継ぐ場合に「今すぐバックアップ」ボタンを押してバックアップします。

またバックアップしたペアリング情報をリストアする場合には、バックアップした内容を選択して「リストア」ボタンを押します。

※HubKitが実行されている場合には停止してください。

5. ファームウェア



ファームウェアのアップデートは [Zigbee firmware のアップデート方法](#) を参照してください

6. アップデート



HubKitの最新バージョンがあるかを定期的に確認するタイミングを設定します。
「今すぐ確認」ボタンを押すとすぐに最新バージョンがあるかを確認します。

最新バージョンがある場合には更新するかの確認ダイアログが表示されます。更新する場合にはメニューに沿ってインストールしてください。

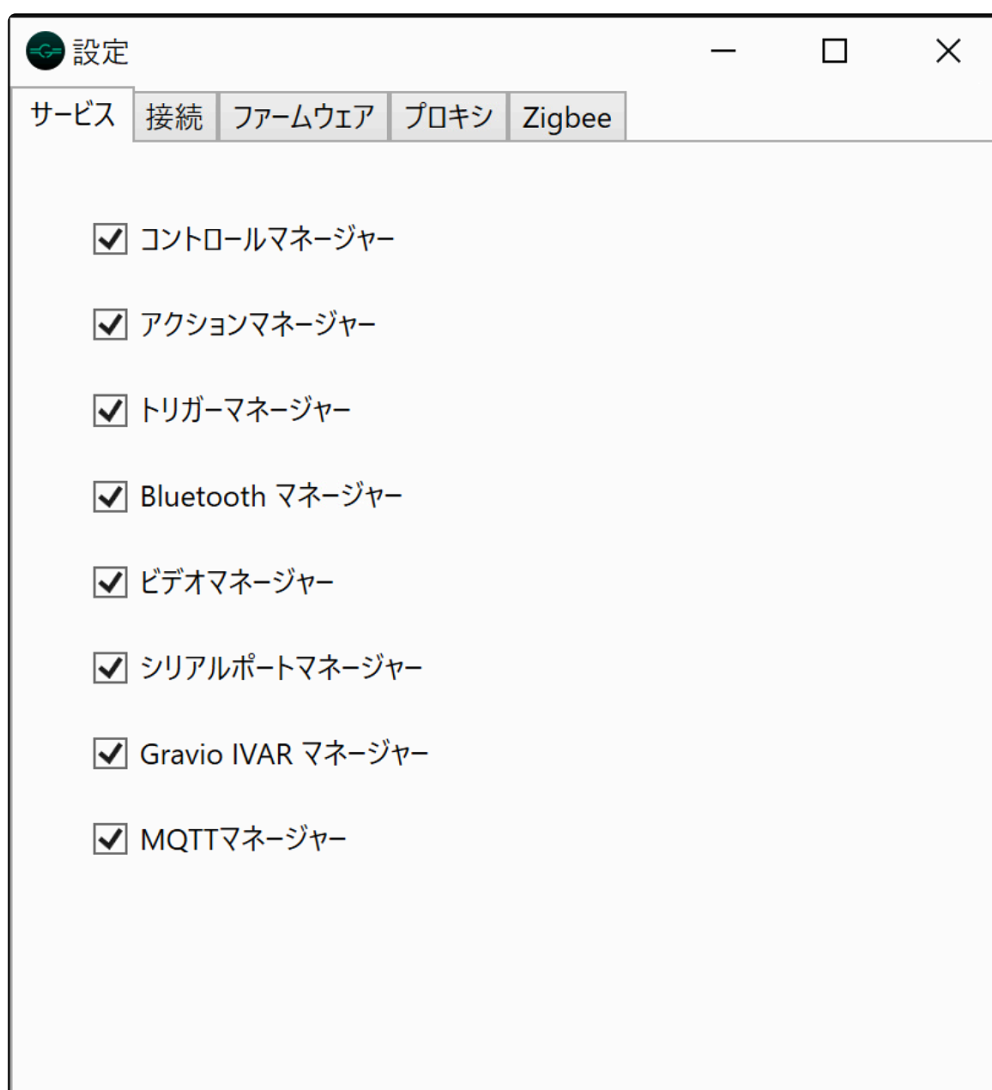
2. Windowsを使用する場合

Windows10のスタートメニューにある「GravioHubKit」メニューの「GravioTools4」を実行します。

GravioTools4が起動するとタスクバーにアイコンが表示されますので、右クリックしてメニューを表示して「設定」メニューを選択します。



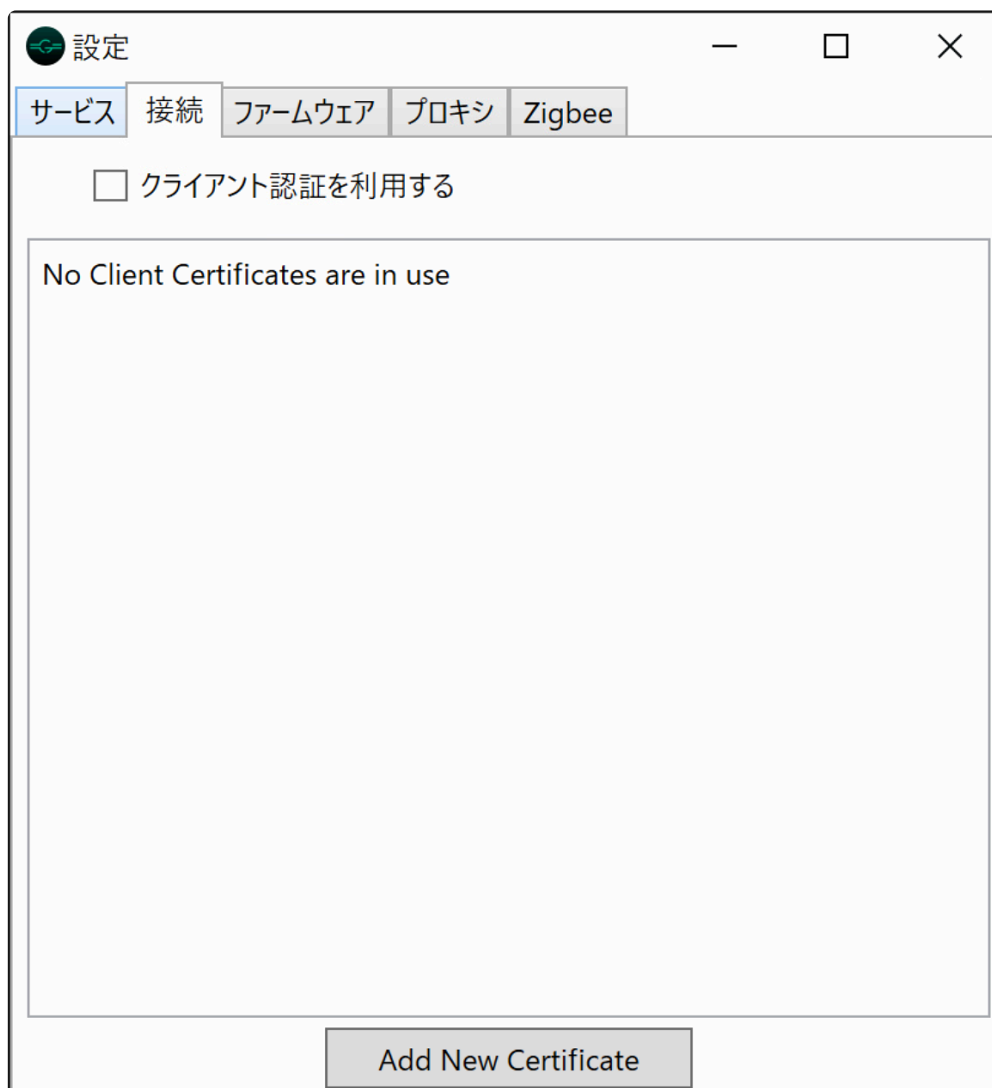
1. サービス



HubKitを構成するサービスを個別に起動・停止する場合に使用します。

HubKitを起動するときにサービスが停止している場合に警告を表示するには「起動時にサービス停止警告を表示する」を選択します。

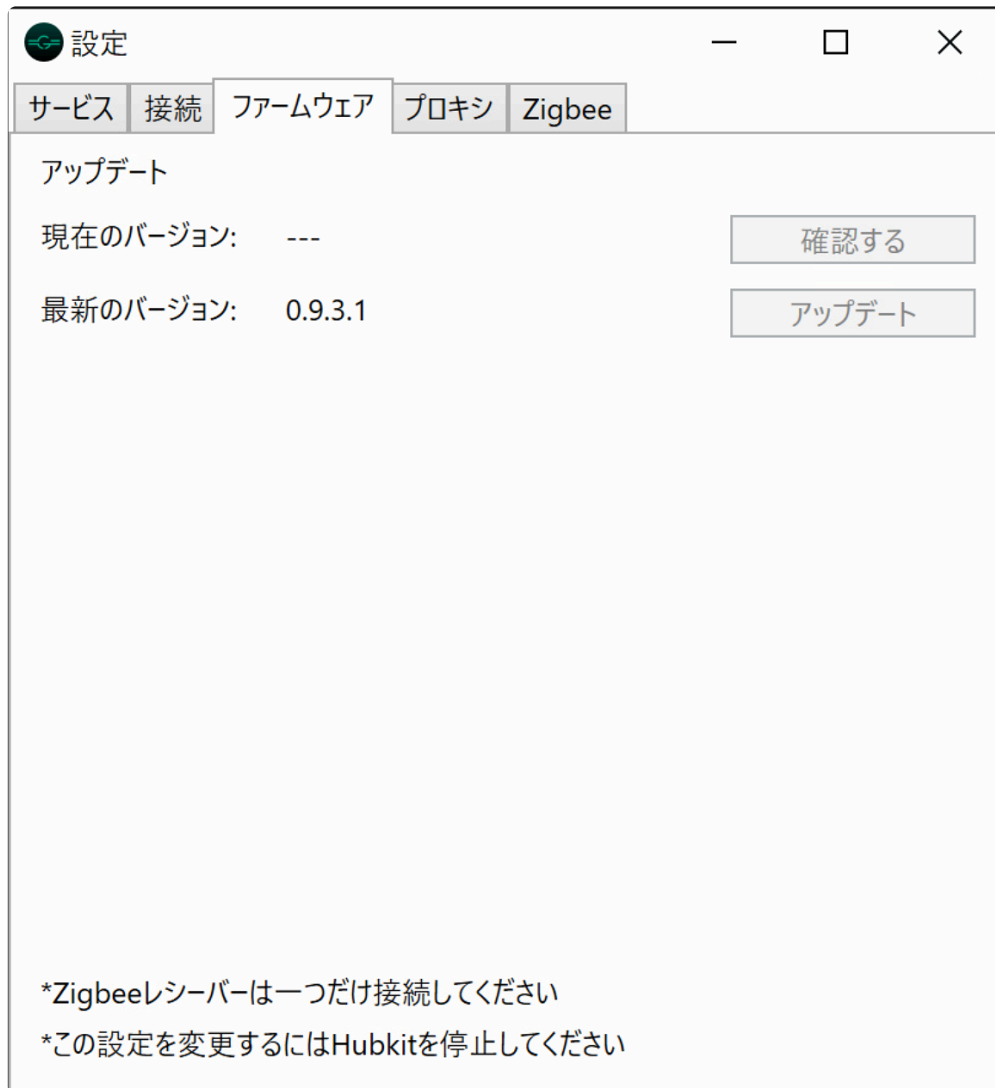
2. 接続



HubKitに接続するGravio Studioを制限する場合には「クライアント認証を利用する」を選択します。
また証明書のインストール方法は [Gravio StudioからHubへのアクセス制限をかけたい場合](#) を参照してください。

※HubKitが実行されている場合には停止してください。

3. ファームウェア



ファームウェアのアップデートは [Zigbee firmware のアップデート方法](#) を参照してください

4. プロキシ

 設定

サービス 接続 ファームウェア プロキシ Zigbee

☐ HTTPSプロキシを使用する

プロキシURL:

ユーザー名:

パスワード:

☐ HTTPプロキシを使用する

プロキシURL:

ユーザー名:

パスワード:

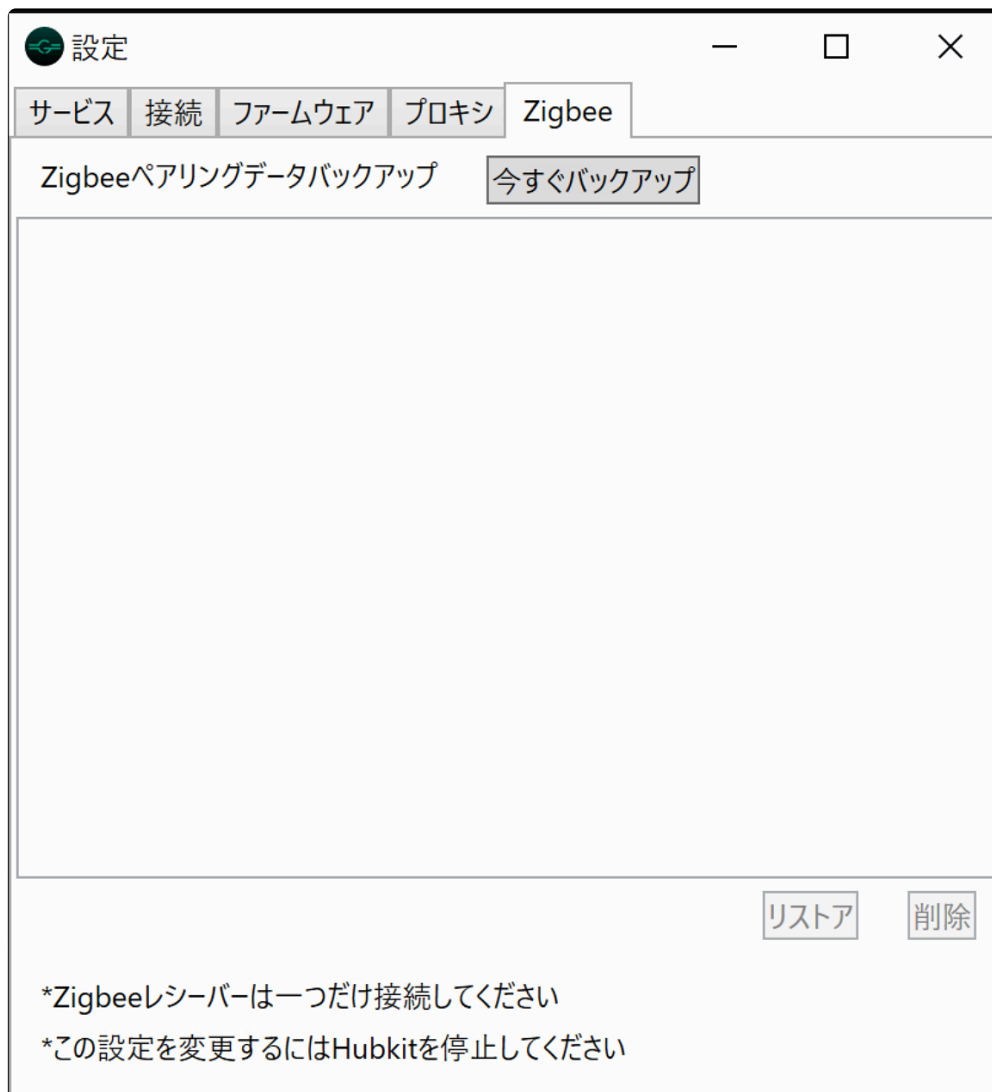
保存

*この設定を変更するにはHubkitを停止してください

HubKitがインターネットサービスを使用する場合にProxyサーバーを経由する必要がある場合に設定します。HTTPSとHTTPのプロトコルでそれぞれに使用するProxyサーバーのアカウント情報（プロキシURL、ユーザー名、パスワード）を入力します。

※この項目を設定するためにはHubKitは停止している必要があります。

5. Zigbee



Zigbeeの dongle のセンサーとのペアリング情報をバックアップする場合や別の dongle に引き継ぐ場合に「今すぐバックアップ」ボタンを押してバックアップします。

またバックアップしたペアリング情報をリストアする場合には、バックアップした内容を選択して「リストア」ボタンを押します。

※HubKitが実行されている場合には停止してください。

6. アップデート

GravioTools4が起動するとタスクバーにアイコンが表示されますので、右クリックしてメニューを表示して「最新版のダウンロード」メニューを選択します。

最新バージョンがある場合には更新するかの確認ダイアログが表示されます。更新する場合にはメニューに沿ってインストールしてください。

14.26. クレジット

Software Libraries Used

Under BSD-2-Clause License

- github.com/godbus/dbus
- github.com/pkg/errors

Under BSD-3-Clause License

- pkg.go.dev/golang.org/x/text
- pkg.go.dev/golang.org/x/net
- pkg.go.dev/golang.org/x/sys
- pkg.go.dev/golang.org/x/crypto
- pkg.go.dev/golang.org/x/image
- pkg.go.dev/golang.org/x/oauth2
- pkg.go.dev/golang.org/x/net
- github.com/muka/ble
- github.com/bugst/go-serial
- github.com/PaesslerAG/gval
- github.com/PaesslerAG/jsonpath
- github.com/generikvault/gvalstrings
- github.com/gorilla/mux

Under Apache 2.0 License

- github.com/coreos/go-systemd
- github.com/aws/aws-sdk-go
- github.com/hajimehoshi/go-mp3
- github.com/hajimehoshi/oto
- github.com/spf13/cobra
- github.com/tensorflow/tensorflow

Under MIT License

- github.com/fatih/structs
- github.com/mattn/go-colorable
- github.com/mattn/go-isatty
- github.com/mgutz/ansi
- github.com/mgutz/logxi
- github.com/sigurn/crc16
- github.com/sigurn/utls
- github.com/StackExchange/wmi
- github.com/go-ole/go-ole

- github.com/hashicorp/mdns
- github.com/rs/cors
- github.com/dgrijalva/jwt-go
- github.com/matttn/go-sqlite3
- github.com/montanaflynn/stats
- github.com/robfig/cron
- github.com/spf13/viper
- github.com/stretchr/testify
- github.com/jmoiron/sqlx
- github.com/matttn/go-pointer
- github.com/matttn/go-tflite
- github.com/satori/go.uuid
- github.com/sirupsen/logrus

Under Other Licenses

- [LICENSE](#): github.com/clbanning/mxj
- [LICENSE](#): github.com/eclipse/paho.mqtt.golang
- [LICENSE](#): gopkg.in/check.v1
- [LICENSE](#): gopkg.in/yaml.v3
- [LICENSE](#): github.com/golang-migrate/migrate/
- [LICENSE](#): github.com/golang/freetype
- [LICENSE](#): github.com/shirou/gopsutil
- [ISC LICENSE](#): github.com/nfnt/resize
- [ISC LICENSE](#): github.com/davecgh/go-spew
- [MPL-2.0 License](#): github.com/go-sql-driver/mysql
- [Zlib License](#): github.com/kardianos/service

Credits for icons, images and other sources.

- Icons made by [Freepik](#) from www.flaticon.com is licensed by [CC 3.0 BY](#)

14.27. Gravio App Service API

14.28. Gravio HubでBluetoothスピーカーを利用する場合(Gravio 5.1以上)

Gravio HubでBluetoothスピーカーを利用する場合(Gravio 5.1以上)は、以下の2行をdocker-compose.ymlに追加で記述してください。

```
– /var/run/pulse/native:/tmp/pulse-socket – /var/run/pulse/.config/pulse/cookie:/root/.config/pulse/cookie
```

手順

1. sshでGravio Hubに接続。
2. "docker-compose down" コマンドを実行
3. docker-compose.ymlの「hubkit-action-manager」のvolumesセクションに以下の様に2行を追加してください。
4. "docker-compose up -d" コマンドを実行

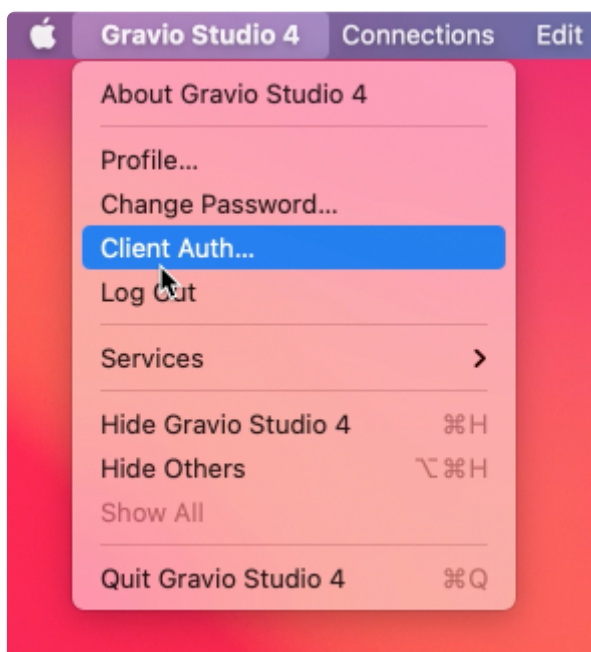
```
hubkit-action-manager:
  container_name: ${HUBKIT_ACTION_MANAGER_CONTAINER}
  image: ${HUBKIT_IMAGE}
  privileged: true
  command: /opt/hubkit/GravioActionManager
  environment:
    PULSE_SERVER: unix:/tmp/pulse-socket
  depends_on:
    - hubkit-control-manager
  volumes:
    - /etc/timezone:/etc/timezone:ro
    - /etc/localtime:/etc/localtime:ro
    - /etc/graviohub.txt:/etc/graviohub.txt:ro
    - hubkit-data4:/var/opt/hubkit
    - hubkit-etc4:/etc/hubkit
    - hubkit-log4:/var/log/hubkit
    - /usr/local/bin:/usr/local/bin
    - /var/run/pulse/native:/tmp/pulse-socket
    - /var/run/pulse/.config/pulse/cookie:/root/.config/pulse/cookie
  stdin_open: true
  tty: true
```

- 5.

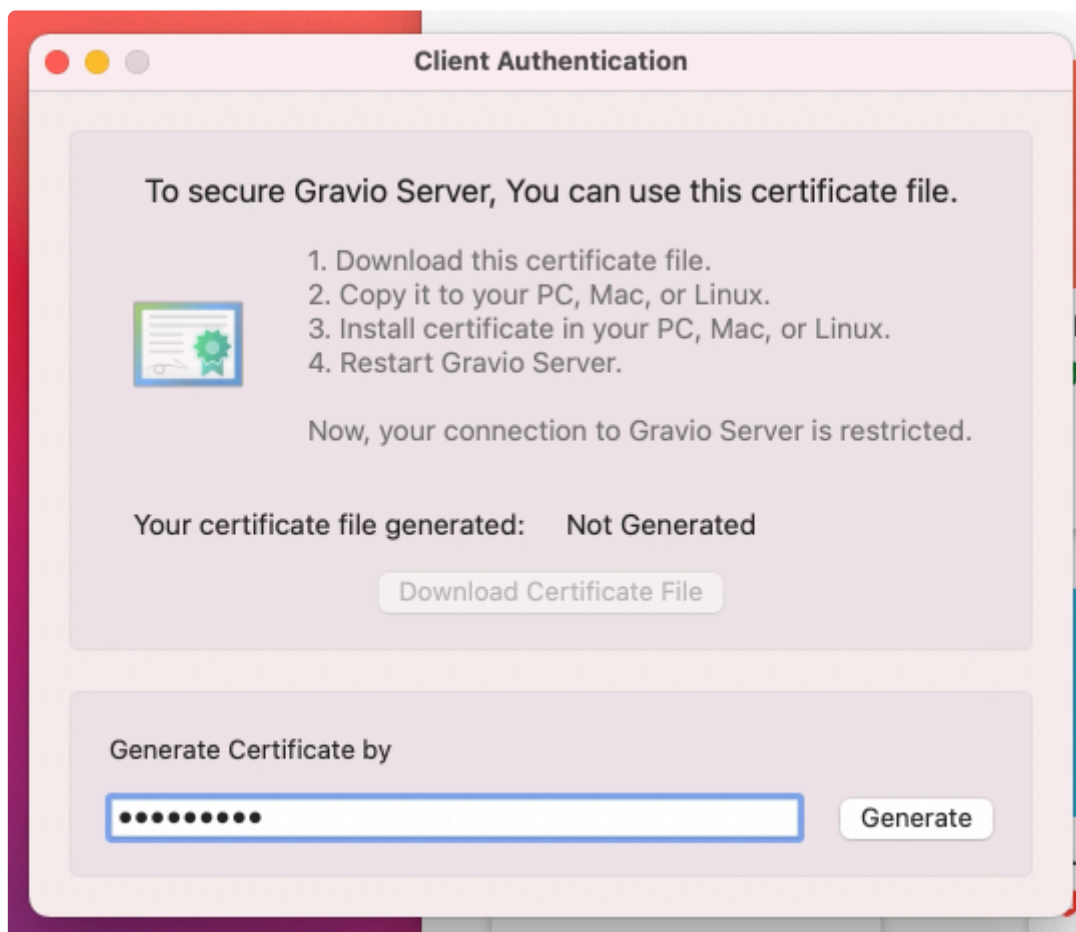
15. Security and Authentication

If you have Gravio Standard or better, you can secure your Gravio HubKit against your Gravio Studio machine using a Client Authentication method. This means you generate an authentication certificate from your Gravio Studio machine, which will be deployed on the Gravio HubKit server. Any Gravio HubKit server that has the client certificate authentication setting enabled will ensure that only Gravio Studios with the respective counterparts are allowed to access it.

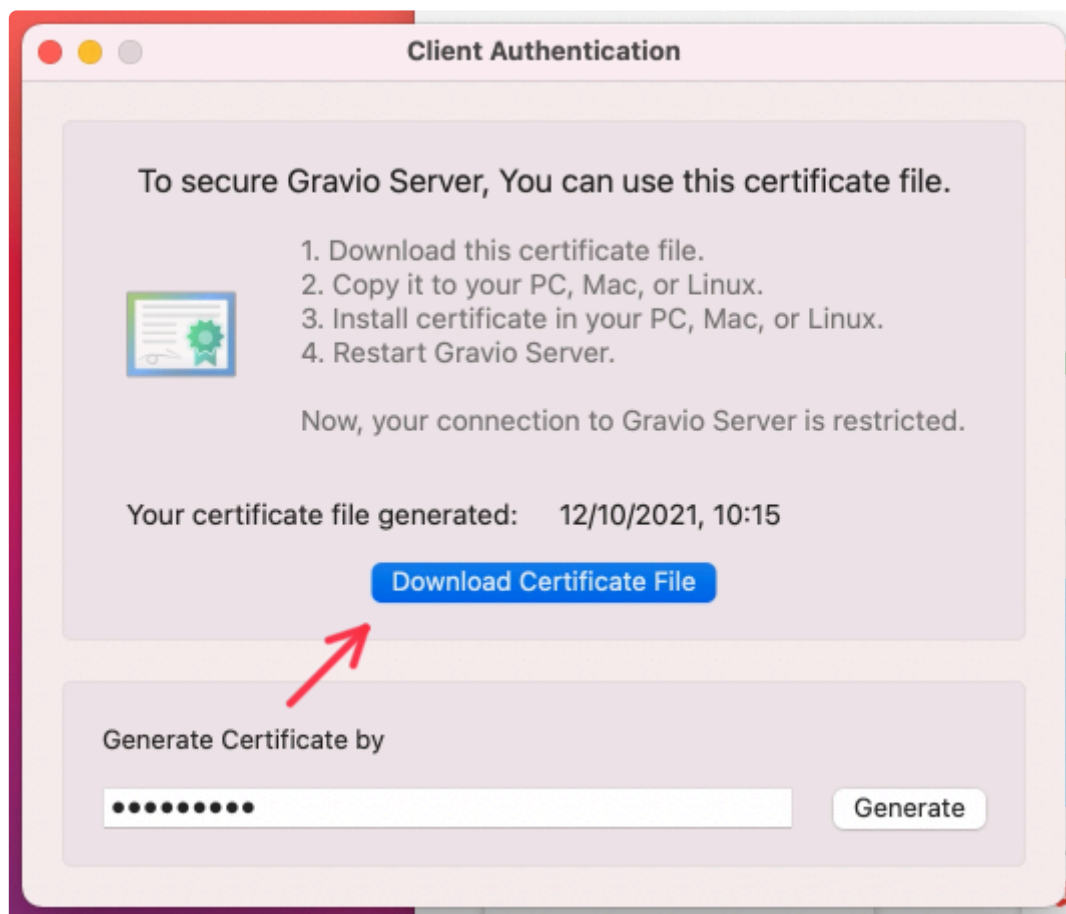
To create the certificate, open your Gravio Studio and generate the from the main menu:



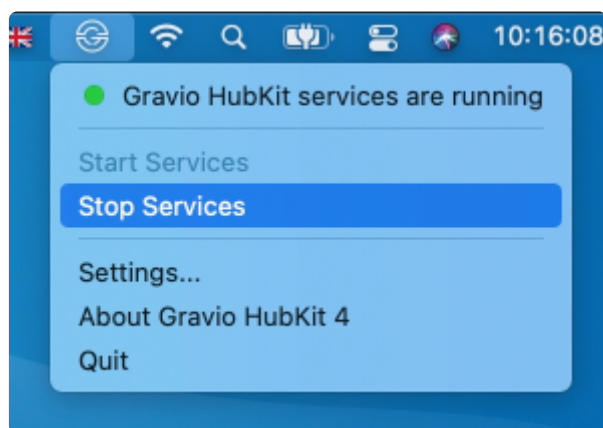
You will have to enter a key, based on which the certificate will be issued.



Click the “Generate” button. At this point, the Certificate matching this Gravio Studio is generated. You can download it to your computer, so you can deploy it on the respective Gravio HubKits.



In the “Settings” > “Connection” menu of HubKit, there is an option to upload the certificates. Note that you will have to stop the server in order to do so.



Click on “Settings...” again and enable the certificate authentication in the “Connections” tab. Upload the just downloaded certificate.



Your connection is now authenticated.

Note that if another Gravio Studio tries to access the HubKit now, you will get an error message similar to the following:

