

FIWARE Orionビルド手順書 (1.2.0版)

2024年03月01日
一般社団法人データ社会推進協議会

改版履歴

| バージョン | 改版内容 | 公開日 |
|-------|--------------------------------|------------|
| 1.0.0 | 初版 (FIWARE Orion 3.6.0 を対象に作成) | 2022/07/01 |
| 1.1.0 | FIWARE Orion 3.7.0 向けに手順書を改版 | 2023/03/01 |
| 1.2.0 | FIWARE Orion 3.10.1 向けに手順書を改版 | 2024/03/01 |

目次

| | | |
|--------|-----------------------|---|
| 1. | はじめに | 4 |
| 1-1. | 概要 | 4 |
| 1-2. | 参考資料 | 4 |
| 1-3. | 前提条件 | 4 |
| 1-4. | 表記方法 | 4 |
| 1-4-1. | コマンドの表記方法 | 4 |
| 1-4-2. | コマンド入力結果の表記方法 | 4 |
| 2. | ビルド手順 | 5 |
| 2-1. | 事前準備 | 5 |
| 2-2. | ビルド | 6 |
| 2-3. | インストール | 6 |
| 3. | 付録 | 7 |
| 3-1. | 必須パッケージインストール方法 | 7 |

1. はじめに

本書は、ブローカー（非パーソナル）として利用するソフトウェアの FIWARE Orion（正式名称 Orion Context Broker）（以下、「Orion」）をビルドする手順を記載・説明するものである。

1-1. 概要

Orion のビルド方法と、インストール方法について述べる。

1-2. 参考資料

参考にした資料は以下である。

https://fiware-orion.readthedocs.io/en/3.10.1/admin/build_source.html

1-3. 前提条件

- Debian(11.8)上で実施すること。
 - wget(1.21)がインストールされていること。
 - インターネットへアクセス可能であること。
 - sudo を使って root 権限でコマンド実行できること。
- ※パッケージ名の後ろの()内は、本書で確認したバージョンである。

1-4. 表記方法

本書での表記方法について記載する。

1-4-1. コマンドの表記方法

(例)

```
$ source ~/ENV.sh
```

コマンド入力を表す箇所については、上記のように実線で囲んでいる。

行頭の \$ はプロンプトであり、入力するのはそれ以降の青い背景色の部分である。

1-4-2. コマンド入力結果の表記方法

(例)

```
HTTP/1.1 201 Created
Date: Wed, 27 Dec 2023 07:45:51 GMT
Content-Type: application/json; charset=utf-8
:
```

コマンド入力結果を表す箇所については、上記のように破線で囲み橙色の背景色で表記している。

2. ビルド手順

本章では Orion のビルド手順について記載・説明する。

はじめにビルドマシン (Debian 11.8) へ一般ユーザでログインする。

※本書では作業ディレクトリを「~/src 配下」としているが、任意のディレクトリを作業ディレクトリとすることも可能である (その場合は作業ディレクトリを読み替えて実行すること)。

2-1. 事前準備

Orion をビルドするために必要なツールやライブラリをインストールする。

ビルドツールとライブラリをインストールする。

```
$ sudo apt-get install make cmake g++
$ sudo apt-get install libssl-dev libcurl4-openssl-dev libboost-dev libboost-regex-dev libboost-filesystem-dev libboost-thread-dev uuid-dev libgnutls28-dev libsasl2-dev libgcrypt-dev
```

Mongo Driver をインストールする。

```
$ mkdir -p ~/src/mong
$ cd ~/src/mong
$ wget https://github.com/mongodb/mongo-c-driver/releases/download/1.23.1/mongo-c-driver-1.23.1.tar.gz
$ tar xfvz mongo-c-driver-1.23.1.tar.gz
$ cd mongo-c-driver-1.23.1
$ mkdir cmake-build
$ cd cmake-build
$ cmake -DENABLE_AUTOMATIC_INIT_AND_CLEANUP=OFF ..
$ make
$ sudo make install
```

rapidjson をインストールする。

```
$ mkdir -p ~/src/rapidjson
$ cd ~/src/rapidjson
$ wget https://github.com/miloyip/rapidjson/archive/v1.1.0.tar.gz
$ tar xfvz v1.1.0.tar.gz
$ sudo mv rapidjson-1.1.0/include/rapidjson/ /usr/local/include
```

libmicrohttpd をインストールする。

※「./configure」コマンドのオプションはライブラリの最小限のフットプリントを得るための推奨ビルド設定であり、必要であれば変更することは可能である。

```
$ mkdir -p ~/src/libmicrohttpd
$ cd ~/src/libmicrohttpd
$ wget http://ftp.gnu.org/gnu/libmicrohttpd/libmicrohttpd-0.9.76.tar.gz
$ tar xvf libmicrohttpd-0.9.76.tar.gz
$ cd libmicrohttpd-0.9.76
$ ./configure --disable-messages --disable-postprocessor --disable-dauth
$ make
$ sudo make install
$ sudo ldconfig
```

mosquitto をインストールする。

※WITH_CJSON, WITH_STATIC_LIBRARIES, WITH_SHARED_LIBRARIES の設定を変更することで、「mosquitto-2.0.15/」ディレクトリ配下の「config.mk」ファイルを変更してビルドを微調整可能である。

```
$ mkdir -p ~/src/mosquitto
$ cd ~/src/mosquitto
$ wget http://mosquitto.org/files/source/mosquitto-2.0.15.tar.gz
$ tar xvf mosquitto-2.0.15.tar.gz
$ cd mosquitto-2.0.15
$ sed -i 's/WITH_CJSON:=yes/WITH_CJSON:=no/g' config.mk
$ sed -i 's/WITH_STATIC_LIBRARIES:=no/WITH_STATIC_LIBRARIES:=yes/g' config.mk
$ sed -i 's/WITH_SHARED_LIBRARIES:=yes/WITH_SHARED_LIBRARIES:=no/g' config.mk
$ make
$ sudo make install
$ sudo ldconfig
```

2-2. ビルド

Orion をビルドする。

ソースコードを取得する。

```
$ cd ~/src
$ sudo apt-get -y install git
$ git clone https://github.com/telefonicaid/fiware-orion -b 3.10.1 --depth 1
```

ビルドする。

```
$ cd ~/src/fiware-orion/
$ make
```

2-3. インストール

Orion をインストールする。

```
$ sudo make install INSTALL_DIR=/usr
$ contextBroker --version
```

Orion のバージョンを確認する。

```
3.10.1 (git version: 9a80e06abe7f690901cf1586377acec02d40e303)
```

3. 付録

本章では Orion のビルドに直接は関係しないが、間接的に必要な情報について記載・説明する。

3-1. 必須パッケージインストール方法

ビルドに必要なパッケージのインストール方法の一例を記載する。すでにインストールされている場合は本作業は不要である。また、インストール方法は複数あるため、本書の手順以外でインストールすることも可能である。

以下の手順は全て root ユーザーで実施すること。

- wget

```
# apt-get -y install wget
```