

# 2020 年度 InSitu 処理向け三次元可視化 フレームワークのプロトタイプ整備

## 利用者説明書

第 1.0 版

富士通株式会社

2021 年 1 月

## 改版履歴

リリース	版数	備考
2021/01/27	1.0	初版

# 目次

目次.....	3
1. はじめに.....	4
2. システム概要.....	5
2.1. ソフトウェア構成.....	5
2.2. 動作環境とインストール作業.....	5
2.2.1. 必要な動作環境.....	5
2.2.2. インストール.....	5
3. 操作説明.....	7
3.1. 概要.....	7
3.2. Temporal Buffer.....	7
3.2.1. 実行方法.....	7
3.2.2. コマンドラインオプション.....	7
3.2.3. 終了方法.....	8
3.3. TB2C server.....	8
3.3.1. 実行方法.....	8
3.3.2. コマンドラインオプション.....	8
3.3.3. 終了方法.....	9
3.4. TB2C client.....	9
3.4.1. 実行方法.....	9
3.4.2. コマンドラインオプション.....	9
3.4.3. 操作方法.....	9
3.4.4. 終了方法.....	12
4. ファイルフォーマット.....	13
4.1. SPH ファイル記述 JSON ファイル.....	13

# 1. はじめに

本書は、国立研究開発法人理化学研究所向け「2020 年度 InSitu 処理向け三次元可視化フレームワークの  
プロトタイプ整備」で開発した TB2C(Temporal Buffer to ChOWDER)の利用者説明書です。  
TB2C のインストール方法および使用方法について説明しています。

## 2. システム概要

### 2.1. ソフトウェア構成

本システムを構成するプログラムの構成を以下に示します。

- (1) Temporal Buffer
  - TB.py
- (2) TB2C\_server
  - TB2C\_server.py
- (3) TB2C\_client
  - TB2C\_client.py

### 2.2. 動作環境とインストール作業

#### 2.2.1. 必要な動作環境

TB2C の動作には、以下のソフトウェアがインストールされた環境が必要です。

- (1) ChOWDER

TB2C は、動作中の ChOWDER サーバーに接続して動作することを前提としています。  
ChOWDER のインストールおよび実行については、以下の URL を参照してください。

<https://github.com/digirea/ChOWDER/blob/master/UserGuide/jp/UserGuide.md>

- (2) obj23dtiles

TB2C は、動作中に obj23dtiles コマンドを使用します。

obj23dtiles は Node.js の npm パッケージとして公開されており、以下のコマンドでインストールできます。

```
sudo npm install -g obj23dtiles
```

obj23dtiles の詳細については、以下の URL を参照してください。

<https://github.com/PrincessGod/objTo3d-tiles>

- (3) Python

- Python 3.x および以下の Python モジュール
  - numpy
  - scikit-image
  - wxPython
  - PyOpenGL
  - websocket-client

#### 2.2.2. インストール

- (1) Python 3

Python 3.x のインストールは、Python の公式サイト(<https://www.python.org>)からインストーラーをダウンロードし、実行することで行います。

Windows の場合、Microsoft Store からインストールすることも可能です。Windows 10 以降のコマンドプロンプトで python コマンドを実行すると、自動的に Microsoft Store 版 Python のインストール画面が表示され、インストールできるようになっています。

macOS の場合、Homebrew を使用したインストールを行うことも可能です。以下の URL を参照してください。

[https://www.python.jp/install/macos/install\\_python.html](https://www.python.jp/install/macos/install_python.html)

[注意]

macOS で pyenv の Python3 を使用している場合、TB2C client を実行(3.4 章参照)すると以下のエラーが出力されて起動できない場合があります。

This program needs access to the screen. Please run with a Framework build, and only when you are logged in on the main display of your Mac.

これは、pyenv がインストールする Python が Framework build (“--enable-framework” オプション付きでビルドしたもの)ではないことが原因です。

pyenv の Python を使用する場合、以下の様にインストールを行ってください。

```
env PYTHON_CONFIGURE_OPTS="--enable-framework" pyenv install 3.6.7
```

Ubuntu 18.04 および 20.04 には最初から Python 3 がインストールされており、そのまま利用可能ですが、後述の Python モジュールのインストールのために、pip のインストールが必要です。pip コマンドが存在しない場合は、以下のコマンドを実行してください。

```
sudo apt install python3-pip
```

CentOS 7 では、Python 3 は EPEL リポジトリからインストールを行うことも可能です。python3 コマンドが存在しない場合は、以下のコマンドを実行してください。pip3 コマンドも併せてインストールします。

```
sudo yum install epel-release
sudo yum install python36u
sudo yum install python36u-pip
```

## (2) Python モジュール群

本ソフトウェアの動作に必要な python モジュールは pip3 コマンドでインストールを行うことができます。以下のコマンドを実行してください。

```
sudo pip3 install numpy
sudo pip3 install scikit-image
sudo pip3 install wxPython
sudo pip3 install PyOpenGL
sudo pip3 install websocket-client
```

## (3) TB2C

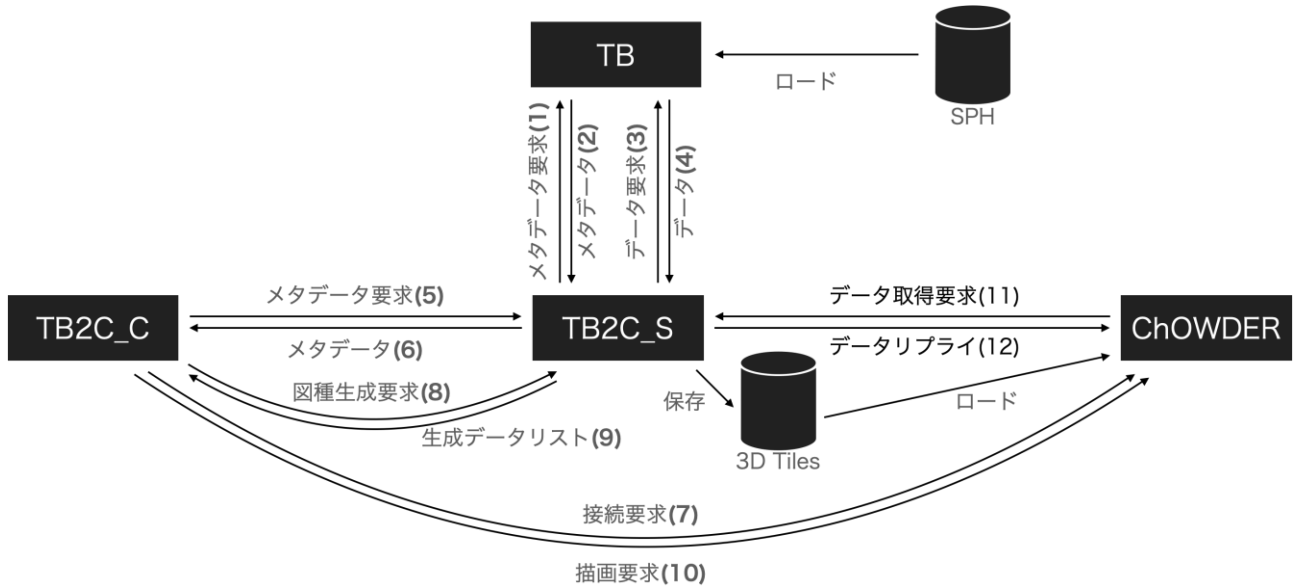
TB2C は、特別なインストール作業は必要ありません。環境を任意のディレクトリに展開すれば、そのディレクトリで使用可能です。

```
$ tar xvfz TB2C-x.y.tar.gz または TB2C-x.y.zip を展開
$ cd TB2C-x.y
```

## 3. 操作説明

### 3.1. 概要

TB2C は、Temporal Buffer (TB) と TB2C\_server (TB2C\_S) および TB2C client (TB2C\_C) から構成されるシステムであり、ChOWDER に接続して動作します。



Temporal Buffer(TB)は、SPHフォーマットで用意された時系列の数値シミュレーション結果データを読み込んでバッファリングし、外部からの要求に応じて必要な時刻スライス・物理量のデータを提供します。TB2C client(TB2C\_C)はユーザーが直接操作するGUIプログラムであり、可視化パラメータの設定や時刻スライスの指定、視点の変更等の操作を行い、ChOWDERに対する表示更新要求を行います。TB2C server(TB2C\_S)は、TB2C\_Cからの要求に応じて、TBが保持するデータを取得して可視化図種の生成を行い、ChOWDER表示用に3D-Tiles形式のファイルに出力します。TBおよびTB2C serverは、ChOWDERと同一のマシン上で動作することを前提としています。TB2C clientは別のマシン上で動作し、ChOWDERおよびTB2C serverと通信を行います。

以下の章では、TB、TB2C server および client について、実行方法・操作方法を説明します。なお、以下の説明では、TB2C 環境を展開したディレクトリを $\${TOP}$ で表します。

## 3.2. Temporal Buffer

### 3.2.1. 実行方法

ターミナル上で、 $\${TOP}/python/TB.py$  を python3 コマンドで実行します。

```
python3  $\${TOP}/python/TB.py$  [-h | --help] [-p portNo] ¥  
[-j input.json | -l file0.sph file1.sph ...]
```

### 3.2.2. コマンドラインオプション

-h または --help  
簡易ヘルプメッセージを出力して終了します。

- p ポート番号  
接続を受け付けるポート番号を指定します。省略時は4001番になります。
  - j 入力 SPH ファイル記述 JSON ファイル  
入力する SPH ファイルリストを記述した JSON ファイルのパスを指定します。  
この指定か、-l による指定のいずれかを行う必要があります。
  - l 入力 SPH ファイルリスト  
入力する SPH ファイル群を時系列の順に指定します。  
この指定か、-j による指定のいずれかを行う必要があります。
- 指定例 1
- ```
python3 ${TOP}/python/TB.py -j data01/pres.json
```
- 指定例 2
- ```
python3 ${TOP}/python/TB.py -l data01/pres_0.sph data01/pres_1.sph ...
```

### 3.2.3. 終了方法

実行したターミナル上で Ctrl-C を入力します。

## 3.3. TB2C server

### 3.3.1. 実行方法

ターミナル上で、`${TOP}/python/TB2C_server.py` を `python3` コマンドで実行します。

```
python3 ${TOP}/python/TB2C_server.py [-h | --help] [--port portNo] ¥
      [--odir outDir] [--dx divX] [--dy divY] [--dz divZ]
```

### 3.3.2. コマンドラインオプション

- h または --help  
簡易ヘルプメッセージを出力して終了します。
  - port ポート番号  
接続を受け付けるポート番号を指定します。省略時は4000番になります。
  - odir 出力先ディレクトリ  
3D-Tiles 形式の等値面ポリゴンデータを出力するディレクトリを指定します。省略時はカレントディレクトリになります。  
今回の実装では、このディレクトリは ChOWDER の public ディレクトリ配下の data というディレクトリである必要があります。
  - dx X 軸方向分割数
  - dy Y 軸方向分割数
  - dz Z 軸方向分割数  
等値面を生成する際の、SPH データの各軸方向の分割数を指定します。省略時はいずれも 1 になります。
- 指定例
- ```
python3 ${TOP}/python/TB2C_server.py --odir /home/chowder/public/data ¥
      --dx 2 --dy 2 --dz 2
```



### 3.3.3. 終了方法

実行したターミナル上で Ctrl-C を入力します。

## 3.4. TB2C client

### 3.4.1. 実行方法

ターミナル上で、`${TOP}/python/TB2C_client.py` を `python3` コマンドで実行します。

```
python3 ${TOP}/python/TB2C_client.py [-h | --help] ¥  
      [-s tb2c_server_url] [-c chowder_host]
```

### 3.4.2. コマンドラインオプション

- h または --help  
簡易ヘルプメッセージを出力して終了します。
- s TB2C server の URL  
起動時に接続する TB2C server の URL を指定します。省略した場合、起動時には接続されません(メニューから接続可能です)。
- c ChOWDER の実行ホスト名  
起動時に接続する ChOWDER のホスト名(または IP アドレス)を指定します。省略した場合、起動時には接続されません(メニューから接続可能です)。  
今回の実装では、基本的に `localhost` を指定する必要があります。

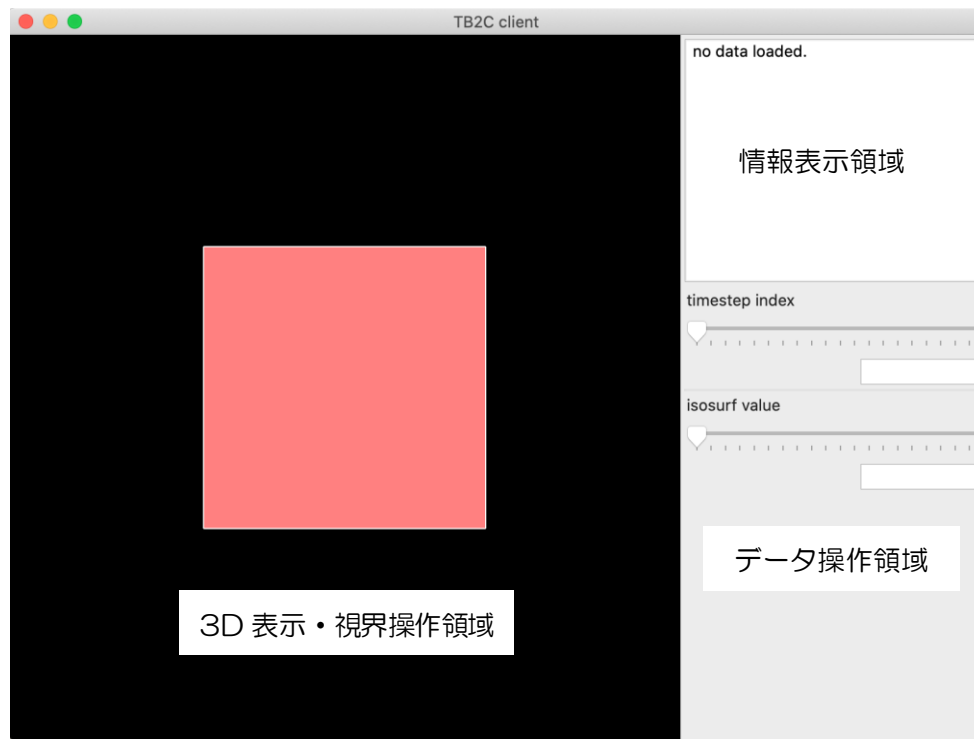
#### ■ 指定例

```
python3 ${TOP}/python/TB2C_client.py -s http://localhost:4000/ -c localhost
```

### 3.4.3. 操作方法

#### (1) ウィンドウ構成

TB2C client は、起動されると下図の様なウィンドウを表示します。



## (2) 3D 表示・視界操作領域の操作

3D 表示・視界操作領域には、可視化対象データのバウンディングボックスを表すキューブが表示されています。キューブの各面の色は、以下の意味を表します。

- 明赤：YZ 平面と平行な+X 側の面
- 暗赤：YZ 平面と平行な-X 側の面
- 明緑：ZX 平面と平行な+Y 側の面
- 暗緑：ZX 平面と平行な-Y 側の面
- 明青：XY 平面と平行な+Z 側の面
- 暗青：XY 平面と平行な-Z 側の面

3D 表示・視界操作領域では、マウス操作によって視界変更を行うことができます。

- 視線方向の回転：マウス左ボタンドラッグ
- 視点位置の上下の移動：Shit キー + マウス左ボタンドラッグ
- 視点位置の前後の移動：Ctrl キー + マウス左ボタンドラッグ (macOS の場合は Command キー)  
またはマウスホイール回転
- 視界操作のリセット：マウス左ボタンダブルクリック

## (3) 情報表示領域の説明

情報表示領域には、TB2C server から取得したデータの情報が表示されます。

- 時間ステップ数
- シミュレーション時刻の範囲
- データ項目数
- データの値域
- バウンディングボックスの座標値

## (4) データ操作領域での操作

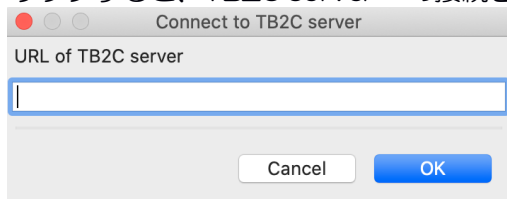
データ操作領域では、可視化対象の時間ステップ番号と、等値面の値を設定します。どちらもスライダーとテキスト入力欄があり、一方を変更するともう一方の値に反映されます。テキスト入力欄では、数値を入力した後 Enter キーを入力しないと値は反映されません。

## (5) メニューバーからの操作

TB2C client のウィンドウのメニューバーには「File」メニューがあり、ここから以下の操作を行うことができます。

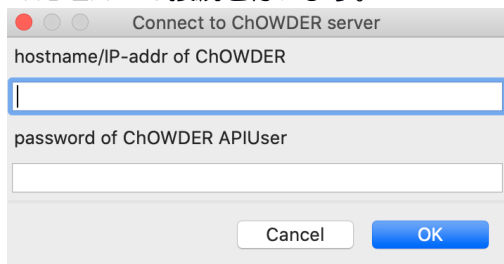
### 「Connect to TB2C server」

以下のダイアログウィンドウが表示され、ここにTB2C server のURL を入力し、「OK」ボタンをクリックすると、TB2C server への接続を行います。



### 「Connect to ChOWDER server」

以下のダイアログウィンドウが表示され、ここにChOWDERが動作するマシンのホスト名(IPアドレス)と、ChOWDERのAPI User のパスワードを入力し、「OK」ボタンをクリックすると、ChOWDER への接続を行います。



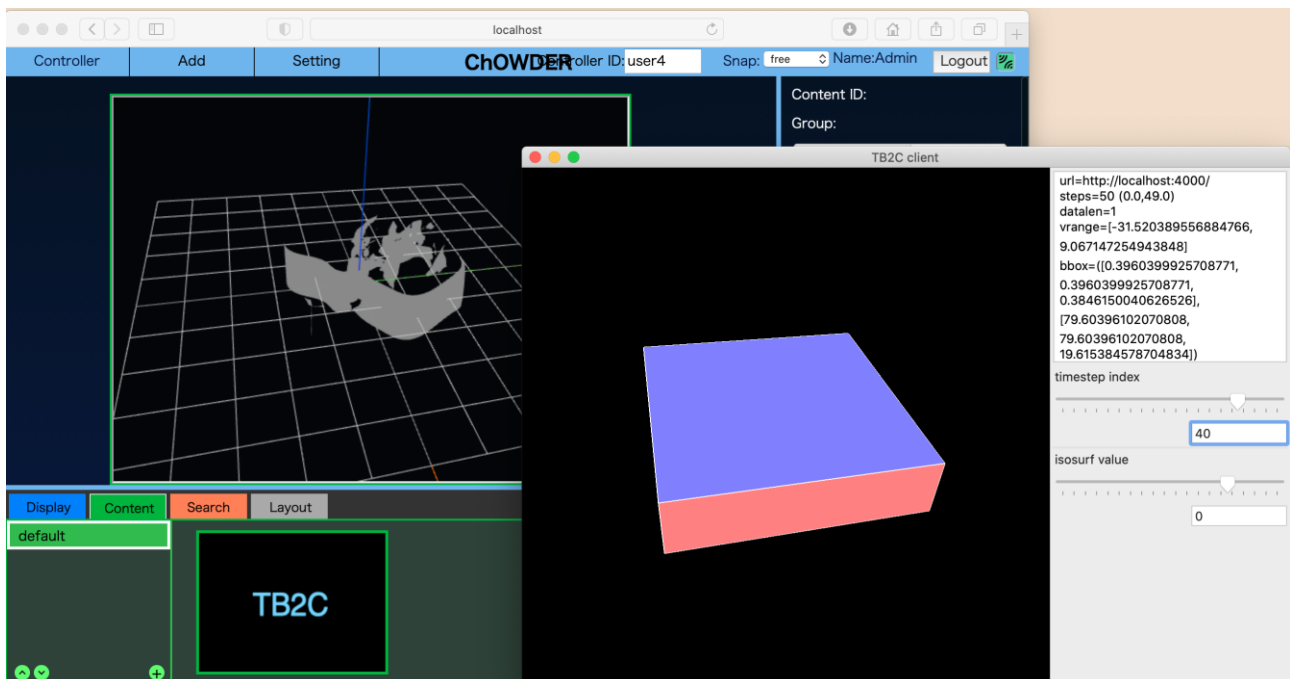
### 「Quit TB2C client」

TB2C client を終了します。

なお、macOS では「File」メニューではなく「Python」メニューに配置されています。

### (6) ChOWDER と連携した動作

TB2C server および ChOWDER と接続した状態で、3D 表示・視界操作領域、データ操作領域での操作を行うと、ChOWDER 上の TB2C コンテンツ(Content ID: tb2c\_3dtile) の表示が更新されます(下図参照)。



#### 3.4.4. 終了方法

実行したターミナル上で `ctrl-c` を入力するか、「File」メニューの「Quit TB2C client」を選択します。尚、TB2C client を終了しても、ChOWDER 上の TB2C コンテンツは削除されません。

## 4. ファイルフォーマット

### 4.1. SPH ファイル記述 JSON ファイル

SPH ファイル記述 JSON ファイルは、Temporal Buffer に入力する時系列 SPH ファイル群を指定するためのファイルです。以下の形式で記述します。

```
{
  "basedir": "data_p",
  "filelist": [
    {"file": "p_100.sph", "step": "100", "time": "0.1"},
    {"file": "p_200.sph", "step": "200", "time": "0.2"},
    ...
  ]
}
```

#### ■ 説明

- **basedir :**  
SPH ファイルが存在するディレクトリを、この JSON ファイルがあるディレクトリからの相対パスまたは絶対パスで指定します。この記述を省略した場合は"./"とみなされます。
- **filelist :**  
SPH ファイルエントリーの配列を記述します。SPH ファイルエントリーは、file, step, time の各項目を記述することができます。
- **file :**  
SPH ファイルのパス名を記述します。SPH ファイルエントリーではこの指定は省略できません。
- **step :**  
この SPH ファイルのタイムステップ番号を記述します。この記述を省略した場合、ファイルの記述順に 0 起点で番号が割り振られます。
- **time :**  
この SPH ファイルのシミュレーション時刻を記述します。この記述を省略した場合、タイムステップ番号を実数化した値が割り振られます。