国立研究開発法人　理化学研究所　御中

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

作業報告書

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](#__RefHeading___Toc6684_1389316416)

[1. はじめに 4](#__RefHeading___Toc6686_1389316416)

[2. システム概要 5](#__RefHeading___Toc6688_1389316416)

[2.1. ソフトウエア構成 5](#__RefHeading___Toc6690_1389316416)

[2.2. 動作環境とインストール作業 5](#__RefHeading___Toc6692_1389316416)

[2.2.1. 必要な動作環境 5](#__RefHeading___Toc6694_1389316416)

[2.2.2. インストール 5](#__RefHeading___Toc6696_1389316416)

[3. 操作説明 7](#__RefHeading___Toc6698_1389316416)

[3.1. 概要 7](#__RefHeading___Toc6700_1389316416)

[3.2. Temporal Buffer 7](#__RefHeading___Toc883_3236865239)

[3.2.1. 実行方法 7](#__RefHeading___Toc6702_1389316416)

[3.2.2. コマンドラインオプション 7](#__RefHeading___Toc6704_1389316416)

[3.2.3. 終了方法 8](#__RefHeading___Toc6704_13893164162)

[3.3. TB2C server 8](#__RefHeading___Toc6728_1389316416)

[3.3.1. 実行方法 8](#__RefHeading___Toc6730_1389316416)

[3.3.2. コマンドラインオプション 8](#__RefHeading___Toc6704_13893164161)

[3.3.3. 終了方法 8](#__RefHeading___Toc6704_138931641621)

[3.4. TB2C client 8](#__RefHeading___Toc6728_13893164161)

[3.4.1. 実行方法 8](#__RefHeading___Toc6730_13893164161)

[3.4.2. コマンドラインオプション 9](#__RefHeading___Toc6704_138931641611)

[3.4.3. 操作方法 9](#__RefHeading___Toc6704_1389316416111)

[3.4.4. 終了方法 11](#__RefHeading___Toc6704_1389316416211)

# 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」の作業報告書です。

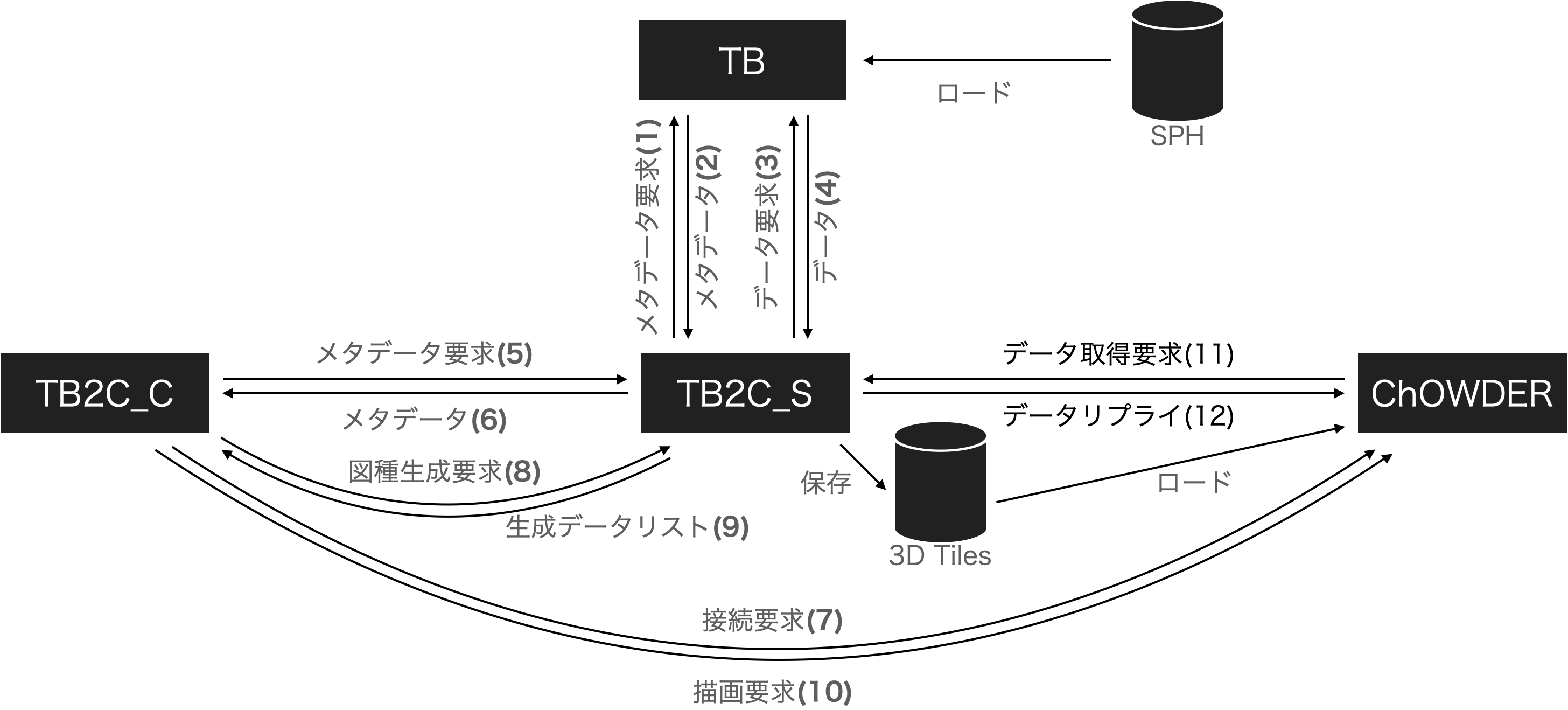
本作業で実装したTB2Cの実装報告および動作検証報告を記述しています。

# **実装報告**

# システム設計

# **システムの構成**

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(下図参照)。



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。

# **プログラム間の通信データ**

以下に、各プログラム間で通信されるデータを示します。

(1) TB2C\_serverからTBへのメタデータ要求

 TBのポートへのHTTP GET

path="/",

param:なし

(2) TBからTB2C\_serverへ返されるメタデータ

 JSON[{"id":データID,

　　　"uri":先頭SPHファイルのURI,

　　　"type":"SPH",

　　　"dims":[I,J,K],

　　　"datalen":ベクトル長,

　　　"bbox":[[x0,y0,z0],[x1,y1,z1]],

 　　　"steps":ステップ数,

　　　"timerange":[開始時刻,終了時刻],

　　　"vrange":[最小値,最大値] }, …]

TBが保持するデータがリスト化されたJSONが返される(今回の実装では１個のみ)

(3) TB2C\_serverからTBへのデータ要求(指定したタイムステップのデータを取得する)

 TBのポートへのHTTP GET

path="/data",

param:id=データID,

step=取得するステップ番号

(4) TBからTB2C\_serverへ返されるデータ

 JSON{"type":"SPH",

"step":ステップ番号,

"data":{データ本体} }

{データ本体}は、SPHクラスを base64でエンコードしてJSON化したもの

(5) TB2C\_clientからTB2C\_serverへのメタデータ要求

 TB2C\_serverのポートへのHTTP GET

path="/",

param:なし

(6) TB2C\_serverからTB2C\_clientへ返されるメタデータ

 JSON{"id":データID,

　　　"uri":先頭SPHファイルのURI,

　　　"type":"SPH",

　　　"dims":[I,J,K],

　　　"datalen":ベクトル長,

　　　"bbox":[[x0,y0,z0],[x1,y1,z1]],

 　　　"steps":ステップ数,

　　　"timerange":[開始時刻,終了時刻],

　　　"vrange":[最小値,最大値] ,

　　　"vistype":[“isosurf”]}

vistypeには、TB2C\_serverがサポートする可視化図種のリストが入る(今回の実装では“isosurf”のみ)

(7) TB2C\_clientからChOWDERへの接続要求

 ChOWDERへのWebSocket通信

path="ws://chowder\_host/v2",

method:"AddContent"

今回の実装では、chowder\_hostは“localhost”のみ

(8) TB2C\_clientからTB2C\_serverへの図種生成要求

 TB2C ServerのポートへのHTTP PUT

path="/visualize"

param:step=対象ステップ番号,

　　　 vistype="isosurf",

　　　 visparam={"value":等値面の値}

 SPHがベクトルデータの場合は、データの L2ノルムで等値面を生成する

TB2C Serverによる図種生成完了を待ってChOWDERへの描画要求(9)を行う

(9) TB2C\_serverからTB2C\_clientへ返される生成データリスト

JSON[{tiledLayer}, …]

TB2C\_serverで生成された3D-Tilesデータの個数分のtiledLayerのリストが返される

(10) TB2C\_clientからChOWDERへの描画要求

ChOWDERへのWebSocket通信

path="ws://chowder\_host/v2",

method:"UpdateMetaData"

タイムステップまたは等値面の値が変更された場合は、アップデートIDを変更する

視界が変更された場合は、”cameraWorldMatrix”を更新する

(11) ChOWDERからTB2C\_serverへのデータ取得要求

TB2C\_serverのポートへのHTTP GET

path="/visualized/...",

param:なし

(12) TB2C\_serverからChOWDERへのデータリプライ

要求されたファイル

今回の実装ではファイル渡し

# **Temporal Buffer**

# **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 ...]

# **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 ...

# **終了方法**

実行したターミナル上でCtrl-Cを入力します。

# **TB2C server**

# **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]

# **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

# **終了方法**

実行したターミナル上でCtrl-Cを入力します。

# **TB2C client**

# **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]

# **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

# **操作方法**

(1) ウインドウ構成

TB2C clientは、起動されると下図の様なウイ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

(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。

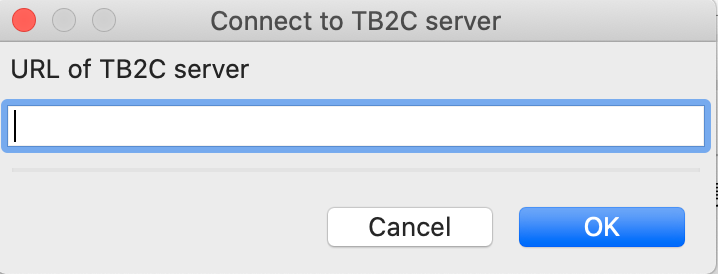
どちらもスライダーとテキスト入力欄があり、一方を変更するともう一方の値に反映されます。

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。

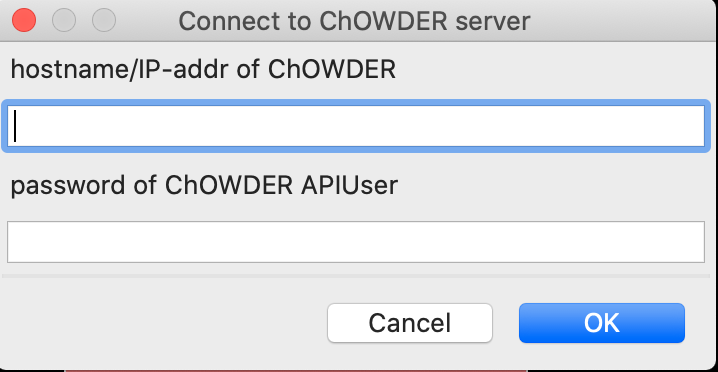
(5) メニューバーからの操作

TB2C clientのウイ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インドウが表示され、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」

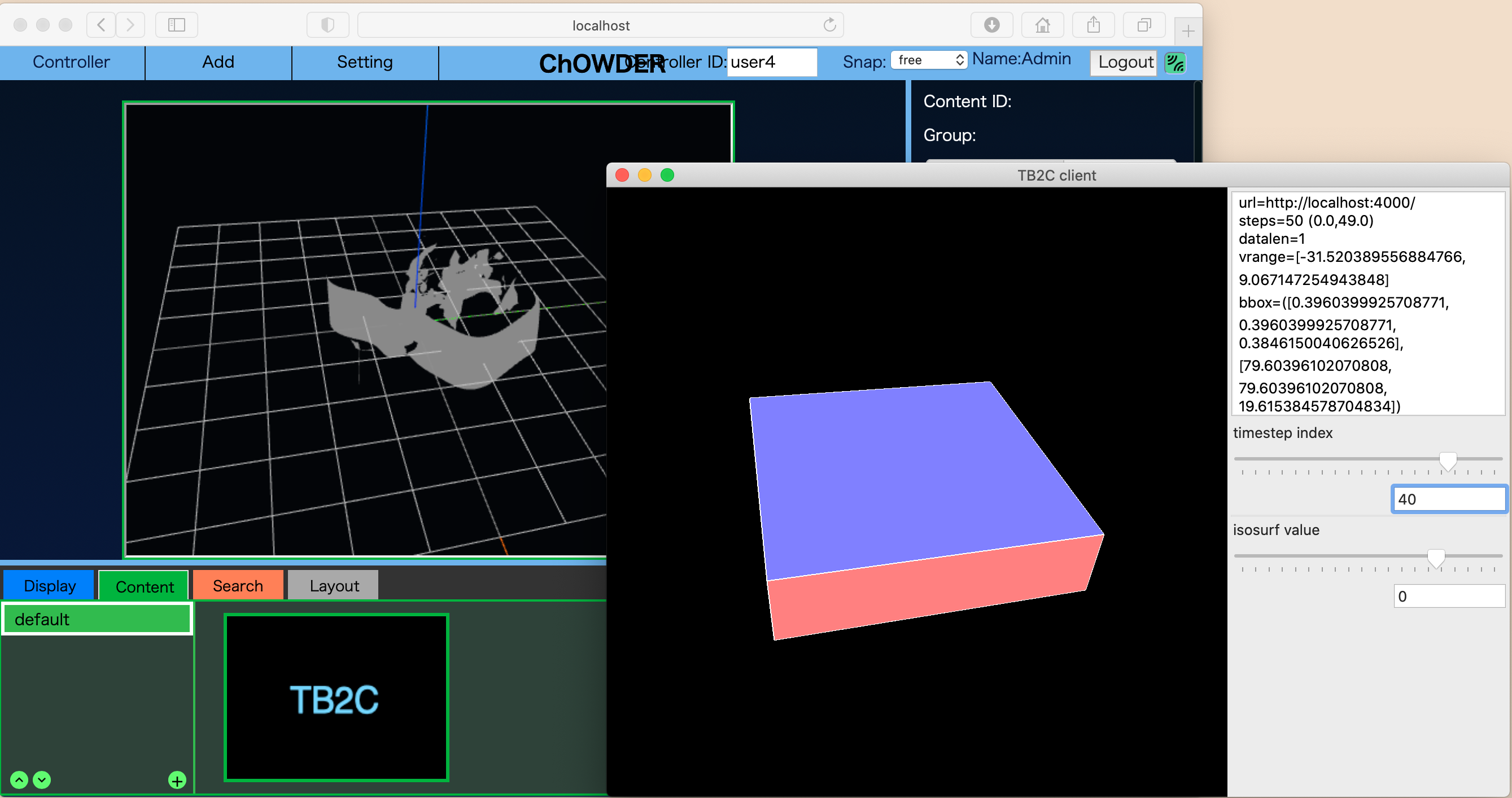
以下のダイアログウインドウが表示され、ここにChOWDERが動作するマシンのホスト名(IPアドレス)と、ChOWDERのAPI Userのパスワードを入力し、「OK」ボタンをクリックすると、ChOWDERへの接続を行います。  
　　

　「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)。



# **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。