Actian Zen v16

Btrieve 2 API Collection の使い方

株式会社エージーテック

2025 年 2 月 21 日

免責事項

株式会社エージーテックは本書の使用を、利用者またはその会社に対して「現状のまま」でのみ許諾するものです。株式会社エージーテックは、いかなる場合にも本書に記載された内容に関するその他の一切の保証を、明示的にも黙示的にも行いません。本書の内容は予告なく変更される場合があります。

商標

Copyright © 2025 AG-TECH Corp. All rights reserved. 本書の全文、一部に関わりなく複製、複写、配布をすることは、前もって発行者の書面による同意がない限り禁止します。
すべての Pervasive ブランド名および製品名は、Pervasive Software Inc. の米国およびその他の国における登録商標または商標です。また、すべての Actian のブランド名は、Actian Corporation の米国およびその他の国における登録商標または商標です。
文中の社名、商品名等は各社の商標または登録商標である場合があります。

Actian Zen v16 Btrieve 2 API Collection の使い方

最終更新：2025 年 2 月 21 日

Copyright © 2025 AG-TECH Corp. All rights reserved.

目次

はじめに 4
Collection クラスの概要 5
サンプルの概要 8
画像書き込みプログラム実行環境の構築 9
Python のインストール 9
OpenCV のインストール 9
Btrieve 2 API SDK のダウンロード 9
SWIG のインストール 9
Btrieve 2 API の Python 用インターフェイスを生成 10
画像検索プログラムの環境設定 11
Btrieve 2 API の C# 用インターフェイスを生成 11
Actian Zen インストールフォルダーにコピー 11
画像書き込みプログラム 12
画像検索プログラム 16
最後に 20

Copyright © 2025 AG-TECH Corp. All rights reserved.

はじめに

本資料では、Btrieve 2 API の Collection の使い方についてサンプルを用いて紹介します。
Collection を使用することで、画像の取り扱いが容易になります。
Btrieve API、SQL（ODBC、ADO、ADO.NET、JDBC）と併用可能で、用途により使用するインターフェイスを選択することで今まで以上により幅広い用途での利用に役立ちます。
本資料では、Visual Studio 2022 がインストールされている前提で記載しております。
Python と C# から Btrieve 2 API を使用するための環境設定についても記載しています。

Copyright © 2025 AG-TECH Corp. All rights reserved.

Collection クラスの概要

Collection クラスは、画像を含め BLOB データの格納に向いたクラスで、BLOB データとともに JSON 形式のデータを書き込むことで、JSON 形式のデータから簡単に検索ができるようになります。

※現行の仕様では、JSON 項目はインデックスとして登録されないため、大量のデータから検索を行う場合、読み込みに時間が掛かります。

JSON データのキーで"_label"については Varchar 型で長さが 254 のキー項目として登録されます。
検索に使用したい項目を"_label"に登録することで素早く読み込むことが可能となります。

Collection クラスで作成したドキュメント（レコード）は、次の形式のファイルとして登録されます。

データ型	長さ	備考	インデックス
Identity	4	追加順のレコード連番	Yes
Varchar	254	label データ	Yes
LONGVARCHAR		JSON 形式データ	No
LONGVARBINASRY		BLOB（画像）データ	No

次の SQL でテーブルとして登録することが可能です。

```
-- Btrieve 2 Collection
SET TRUENULLCREATE = OFF;
CREATE TABLE "Table Name" IN DICTIONARY USING 'File Path' (
    "id" IDENTITY DEFAULT '0' NOT NULL NOT MODIFIABLE,
    "label" VARCHAR(254) NOT NULL,
    "json" CLOB NOT NULL,
    "blob" BLOB NOT NULL);
SET TRUENULLCREATE = ON;
CREATE INDEX "label_index" IN DICTIONARY ON "Table Name" ( "label" );
```

※項目名は任意に変更可能です。「Table Name」および「File Path」は任意に設定して下さい。

Copyright © 2025 AG-TECH Corp. All rights reserved.

Btrieve ファイルは、BtrieveClient クラスの CollectionCreate メソッドを実行することで作成されます。

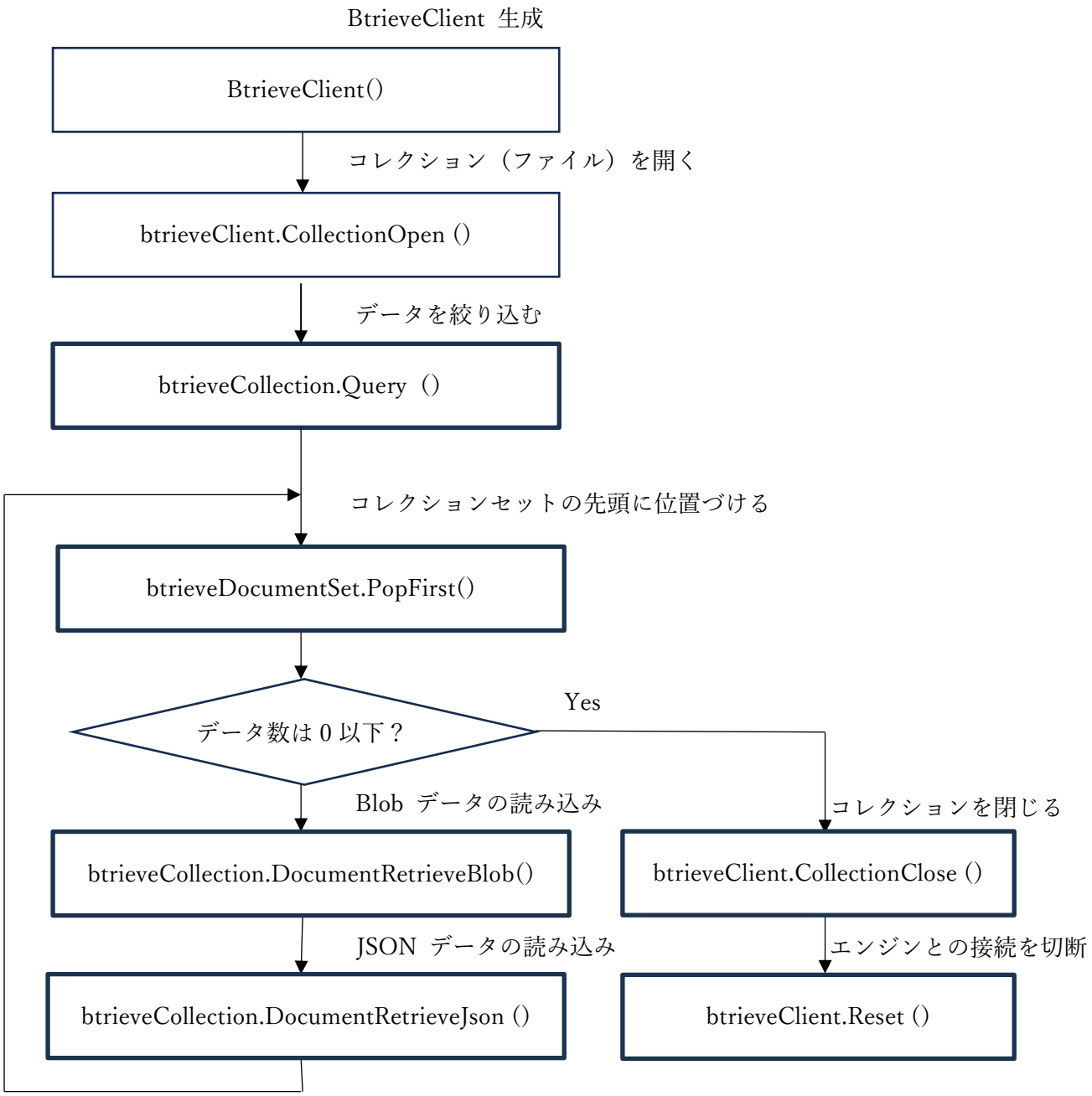
データの書き込みは、BtrieveClient クラスの CollectionOpen メソッドで Btrieve ファイルを開き、Collection クラスの DocumentCreate メソッドで BLOB（画像）データの書き込みを行います。

データの読み込みは、BtrieveClient クラスの CollectionOpen メソッドで Btrieve ファイルを開き、Collection クラスの Query メソッドで絞り込みを行います。
Query メソッドでは、絞り込みを行ったレコードがドキュメントセットとして返されます。
BtrieveDocumentSet クラスの PopFirst または PopLast メソッドで位置付けを行い、Collection クラスの DocumentRetrieveBlob メソッドで BLOB データを、DocumentRetrieveJson メソッドで JSON データを読み込みます。

※PopFirst、PopLast メソッドを実行すると、最初（または最後）のデータはドキュメントセットから削除され、次（またはひとつ前）のデータが先頭（または最後）のデータとなります。返されたドキュメントの ID でデータへのアクセスが可能です。

Copyright © 2025 AG-TECH Corp. All rights reserved.

次のフローは、コレクションの絞り込みを行ったデータを先頭から読み込む場合の処理の流れです。



Copyright © 2025 AG-TECH Corp. All rights reserved.

サンプルの概要

サンプルは、USB 接続されたカメラから定期的に画像を読み込む Python のプログラム（画像書き込みプログラム）と、直前の画像と指定した割合よりも一致率が低い画像を抽出して表示する C# のプログラム（画像検索プログラム）で構成されます。
USB カメラからの画像読み込みは OpenCV を使用し、約 5 秒おきに読み込みを行い、一致率は OpenCV の compareHist メソッドで直前の画像と比較することで求めています。
画像検索の際、一致率が特定の画像を素早く検索するために、JSON データの _labels 項目に一致率を設定しています。
画像を抽出して表示するプログラムでは、Query メソッドで JSON データの _labels 項目に設定された一致率で絞り込みを行い、最新のデータから読み込んで表示を行います。
直前の画像と比較ができるよう、直前の画像を並べて表示しています。

Copyright © 2025 AG-TECH Corp. All rights reserved.

画像書き込みプログラム実行環境の構築

サンプルの実行には、Python と OpenCV のインストール、Python 用の Btrieve 2 API モジュールの作成が必要です。

◆ Python のインストール

Python は、次の WEB ページよりインストーラーをダウンロードしてインストールを行います。
※Python のバージョンにより、後述の画像読み込みプログラムがエラーになる可能性があります。本サンプルでは Python 3.8 を使用しています。

https://www.python.org/ftp/python/3.8.8/python-3.8.8-amd64.exe

ダウンロードしたファイルを実行してインストールを行います。
インストールの際、「Add Python 3.x to PATH」のチェックをオンにしてください。

◆ OpenCV のインストール

次のコマンドで OpenCV およびキーボード入力用のライブラリ keyboard をインストールします。

```
pip install opencv-python
pip3 install keyboard
```

◆ Btrieve 2 API SDK のダウンロード

次の Web ページから Btrieve 2 API SDK をダウンロードし、インストールします。

https://www.agtech.co.jp/actian/support/reference/sdk/access_methods/btrieve/v16/

◆ SWIG のインストール

次の WEB ページから SWIG をダウンロードして c:¥ に解凍します。
その後、フォルダー名を swigwin に変更してください。

http://www.swig.org/

Copyright © 2025 AG-TECH Corp. All rights reserved.

◆ Btrieve 2 API の Python 用インターフェイスを生成

次のファイルを c:¥work フォルダーにコピーします。

```
C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥swig¥ btrievePython.swig
C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥swig¥ btrieveSwig.swig
C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥win64¥x86_64¥ btrieveCpp.lib
C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥include¥ btrieveCpp.h
C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥include¥ btrieveC.h
```

スタートメニューの"Visual Studio 2022"にある"x64 Native Tools Command Prompt for VS 2022"のコマンドプロンプトを「管理者として実行」で起動し、c:¥work フォルダーをカレントとして次のコマンドを実行することで、Btrieve 2 の Python インターフェイスモジュールをビルドします。

```
c:¥swigwin¥swig -cppext cpp -outcurrentdir -python -c++ -Iinclude -o btrievePython.cpp
btrievePython.swig
cl -EHsc -c -Iinclude -IC:¥Users¥user1¥AppData¥Local¥Programs¥Python¥Python38¥include b
trievePython.cpp
link -dll btrieveCpp.lib -libpath:C:¥Users¥user1¥AppData¥Local¥Programs¥Python¥Python38
¥libs -out:_btrievePython.pyd btrievePython.obj
```

※ User1 は、OS にログインしているアカウントに変更してください。

出来上がった、Btrieve 2 の Python インターフェイスモジュールを Zen v16 のインストールフォルダー内にコピーします。

```
copy btrievePython.py "c:¥Program Files¥Actian¥Zen¥bin"
copy _btrievePython.pyd "c:¥Program Files¥Actian¥Zen¥bin"
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

画像検索プログラムの環境設定

◆ Btrieve 2 API の C# 用インターフェイスを生成

スタートメニューの"Visual Studio 2022"にある"x64 Native Tools Command Prompt for VS 2022"のコマンドプロンプトを「管理者として実行」で起動し、C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API フォルダーをカレントとして次のコマンドを実行することで、Btrieve 2 API の C# インターフェイスモジュールをビルドします。

```
c:¥swigwin¥swig.exe -cppext cpp -outcurrentdir -csharp -namespace Actian.Data.Zen -
outfile btrieveCSharp.cs -c++ -Iinclude -o btrieveCSharp.cpp swig¥btrieveCSharp.swig
cl -EHsc -c -Iinclude btrieveCSharp.cpp
link -dll win64¥x86_64¥btrieveCpp.lib btrieveCSharp.obj
```

◆ Actian Zen インストールフォルダーにコピー

ビルドした Btrieve 2 API の C# インターフェイスモジュールを Actian Zen インストールフォルダーにコピーします。

```
copy btrieveCSharp.cs "c:¥Program Files¥Actian¥Zen¥bin"
copy btrieveCSharp.dll "c:¥Program Files¥Actian¥Zen¥bin"
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

画像書き込みプログラム

画像書き込みプログラムは、USB 接続されたカメラから定期的に画像を読み込む処理を Python のプログラムで実行しています。
USB カメラからの画像読み込みは OpenCV を使用し、約 5 秒おきに読み込みを行い、一致率は OpenCV の compareHist メソッドで直前の画像と比較することで求めています。
画像検索の際に特定の画像を素早く見つけるため、JSON データの labels 項目に一致率を設定しています。
一致率は、検索処理を簡易とするため、10% 単位（0 ~ 9 の値）で記録しています。

```
import cv2
import os
import sys
import struct
import time
import keyboard
from datetime import datetime
from pytz import timezone

sys.path.append("C:¥¥Program Files¥¥Actian¥¥Zen¥¥bin")
sys.path.append("C:¥¥Program Files (x86)¥¥Actian¥¥Zen¥¥bin")
import btrievePython as btrv

btrfile='C:¥¥test¥Zen16¥OpenCV¥db¥CVdata.mkd'
fflg = 0

cap = cv2.VideoCapture(0)
ret, image = cap.read()
imagebak = image
print('終了するには "q" を入力')
while True:
    img_size = (100, 100)
    fflg = 1
# Create a session:
    btrieveClient = btrv.BtrieveClient(0x4232, 0) #B2
    Bcollection = btrv.BtrieveCollection()
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
# Collection open
    status = btrieveClient.CollectionOpen(Bcollection, btrfile)
# Collection が無ければ作成後オープン
    if (status == btrv.Btrieve.STATUS_CODE_FILE_NOT_FOUND):
        rc=btrieveClient.CollectionCreate(btrfile)
        status = btrieveClient.CollectionOpen(Bcollection, btrfile)
    if (status != btrv.Btrieve.STATUS_CODE_NO_ERROR):
        print('File ', btrfile, ' not open; ', status, ': ',
btrv.Btrieve.StatusCodeToString(status))

    imageSize = int(image.size)
    imageShape = image.shape
    if (fflg == 0):
        matchr = '"matchrate" : ' + str(000)
    else:
        matchr = '"matchrate" : ' + str(000)
        image1 = cv2.resize(image, img_size)
        image2 = cv2.resize(imagebak, img_size)
        image1_hist = cv2.calcHist([image1], [0], None, [256], [0, 256])
        image2_hist = cv2.calcHist([image2], [0], None, [256], [0, 256])
        matchi = cv2.compareHist(image1_hist, image2_hist, 0)
        matchr = '"matchrate" : ' + str(matchi * 100)
    imagebak = image
    if (matchi >= 0.9):
        matchl = '"matchlevel" : ' + '"9"'
        labels = '"9"'
    if (matchi < 0.9 and matchi >= 0.8):
        matchl = '"matchlevel" : ' + '"8"'
        labels = '"8"'
    if (matchi < 0.8 and matchi >= 0.7):
        matchl = '"matchlevel" : ' + '"7"'
        labels = '"7"'
    if (matchi < 0.7 and matchi >= 0.6):
        matchl = '"matchlevel" : ' + '"6"'
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
        labels = '"6"'
    if (matchi < 0.6 and matchi >= 0.5):
        matchl =  '"matchlevel" : ' + '"5"'
        labels = '"5"'
    if (matchi < 0.5 and matchi >= 0.4):
        matchl =  '"matchlevel" : ' + '"4"'
        labels = '"4"'
    if (matchi < 0.4 and matchi >= 0.3):
        matchl =  '"matchlevel" : ' + '"3"'
        labels = '"3"'
    if (matchi < 0.3 and matchi >= 0.2):
        matchl =  '"matchlevel" : ' + '"2"'
        labels = '"2"'
    if (matchi < 0.2 and matchi >= 0.1):
        matchl =  '"matchlevel" : ' + '"1"'
        labels = '"1"'
    if (matchi < 0.1 and matchi >= 0):
        matchl =  '"matchlevel" : ' + '"0"'
        labels = '"0"'
    utc_now = datetime.now(timezone('UTC'))
    jst_now = utc_now.astimezone(timezone('Asia/Tokyo'))
    ts = '"timestamp" : "' + jst_now.strftime("%Y-%m-%d %H:%M:%S") + '"'

    jsonString = '{'
    jsonString += '"imagesize" : ' + str(imageSize) + ','
    jsonString += '"imageshape" : "' + str(imageShape) + '",'
    jsonString += '"_label" : ' + labels + ', '
    jsonString += ts + ','
    jsonString += matchr + ','
    jsonString += matchl
    jsonString += '}'

    tmpfile = "image.png"
    cv2.imwrite(tmpfile, image)
    blobFile = open(tmpfile, mode="rb")
    documentBlob = blobFile.read()
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

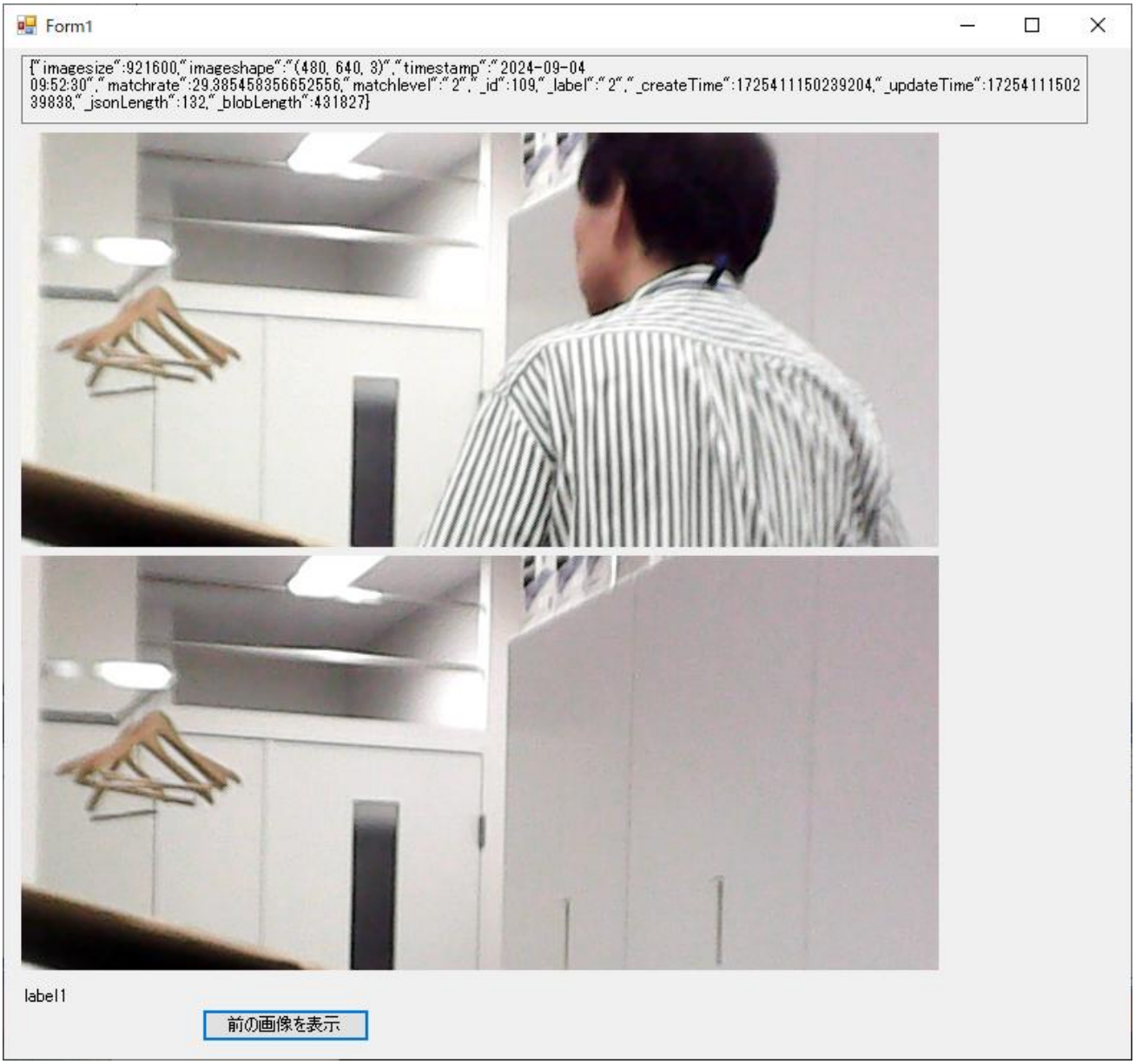
```
blobFile.close()
print(ts + '      ' + matchr)

documentId = Bcollection.DocumentCreate(jsonString,documentBlob)
wcnt = 0
while wcnt < 50:
    if keyboard.is_pressed('q'):  # 'q'キーが押された場合終了
        exit()
    wcnt = wcnt + 1
    time.sleep(0.1)
cap = cv2.VideoCapture(0)
ret, image = cap.read()
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

画像検索プログラム

画像を抽出して表示するプログラムでは、Query メソッドで JSON データの _labels 項目に設定された一致率が 9 でないデータで絞り込みを行い、最新のデータから読み込んで表示を行います。
直前の画像と比較ができるよう、下に画像の不一致を検出した際の画像を、上に直前の画像を並べて表示しています。
テキストボックスには、JSON で返されるデータを表示しています。
この中には、レコードの項目に設定したデータの他、ID やレコードの作成日時、更新日時等も含まれます。



Copyright © 2025 AG-TECH Corp. All rights reserved.

フォームには、JSON データを表示する TextBox と画像を表示する PictureBox を 2 つ、Label と Button 配置し、コードには次の内容を記述します。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Actian.Data.Zen;

namespace cView
{
    public partial class Form1 : Form
    {
        BtrieveClient btrieveClient = new BtrieveClient();
        BtrieveCollection btrieveCollection = new BtrieveCollection();
        Btrieve.StatusCode statusCode;
        BtrieveDocumentSet btrieveDocumentSet = new BtrieveDocumentSet();
        int lid;
        byte[] blobp = new byte[700000];
        byte[] blobp2 = new byte[700000];
        int bsize = 700000;
        string query;
        string json;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
        {

            statusCode          =          btrieveClient.CollectionOpen(btrieveCollection,
"C:¥¥test¥¥Zen16¥¥OpenCV¥¥db¥¥CVdata.mkd");
            //            query = "{¥"$not¥" : {¥"_label¥" : ¥"9¥"}}";
            //query = "{¥"_label¥" : ¥"9¥"}";
            query = "{¥"_label¥" : {¥"$ne¥" : ¥"9¥"}}";
            statusCode = btrieveCollection.Query(btrieveDocumentSet, query);
            lid = btrieveDocumentSet.PopLast();
            if (lid <= 1)
            {
                label1.Text =  "対象データなし";
            }
            else
                {
               // 下のピクチャーボックスに不一致を検出した画像を表示
                statusCode = btrieveCollection.DocumentRetrieveBlob(lid, blobp, bsize);
                json = btrieveCollection.DocumentRetrieveJson(lid);
                textBox1.Text = json;
                ImageConverter imgconv = new ImageConverter();
                Image img = (Image)imgconv.ConvertFrom(blobp);
                pictureBox1.Image = img;
               // 上のピクチャーボックスに直前の画像を表示
                statusCode = btrieveCollection.DocumentRetrieveBlob(lid - 1, blobp2, bsize);
                ImageConverter imgconv2 = new ImageConverter();
                Image img2 = (Image)imgconv.ConvertFrom(blobp2);
                pictureBox2.Image = img2;

            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            lid = btrieveDocumentSet.PopLast();
            if (lid <= 1)
```

Copyright © 2025 AG-TECH Corp. All rights reserved.

```
            {
                label1.Text = "対象データなし";
            }
            else
            {
               // 下のピクチャーボックスに不一致を検出した画像を表示
                statusCode = btrieveCollection.DocumentRetrieveBlob(lid, blobp, bsize);
                json = btrieveCollection.DocumentRetrieveJson(lid);
                textBox1.Text = json;
                ImageConverter imgconv = new ImageConverter();
                Image img = (Image)imgconv.ConvertFrom(blobp);
                pictureBox1.Image = img;
               // 上のピクチャーボックスに直前の画像を表示
                statusCode = btrieveCollection.DocumentRetrieveBlob(lid - 1, blobp2, bsize);
                ImageConverter imgconv2 = new ImageConverter();
                Image img2 = (Image)imgconv.ConvertFrom(blobp2);
                pictureBox2.Image = img2;

            }

        }
    }
}
```

プロジェクトのフォルダーに C:¥Program Files (x86)¥Actian¥Zen¥SDK¥Btrieve2API¥btrieveCSharp.cs をコピーして、プロジェクトに追加します。

Copyright © 2025 AG-TECH Corp. All rights reserved.

最後に

今後、画像データを扱うシステムが増えてくると予想されますが、Btrieve 2 API の Collection クラスをご使用いただくことで、お手軽にデータベースに格納することが可能となります。
画像ファイルそのままでは誰でも簡単に参照できてしまいますが、データベースに登録することで、簡単に参照できなくすることも可能です。
JSON データを同時に管理できることもメリットといえます。

Copyright © 2025 AG-TECH Corp. All rights reserved.