

BANNER[®]

more sensors, more solutions

 **PresencePLUS**[®]

ビジョンセンサ
イメージエクスポート
ユーティリティ

Image Export



バナー・エンジニアリング・ジャパン

バナー・エンジニアリング・インターナショナル Inc. - ジャパン・ブランチ

〒532-0011 大阪市淀川区西中島3-23-15 セントアーバンビル305

TEL : 06-6309-0411 FAX : 06-6309-0416

E-mail : tech@bannerengineering.co.jp <http://www.bannerengineering.co.jp>

1. 概要

イメージエクスポート・ユーティリティは、ActiveXコントロールの一種であり、PresencePlusビジョンセンサのイメージエクスポート機能と連携して使用します。このユーティリティで提供するActiveXコントロールにより、ビジョンセンサで取り込んだ検査画像をTCP/IPでPCの任意のアプリケーションプログラムに送信し、画像を表示したりハードディスクへ保存したりすることができます。

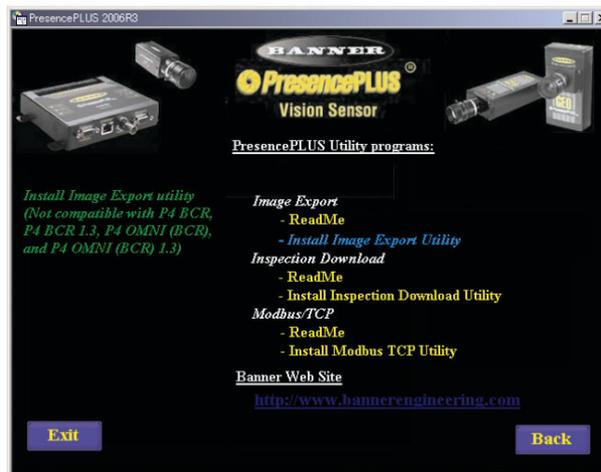
この機能は下記のコンポーネントで構成されています。

- イーサネット経由で画像データをエクスポートする処理を制御する拡張型コミュニケーションツール
- 転送画像を受信するユーザーアプリケーションで使用するActiveXコントロール
- ActiveXコントロール使用による画像処理を説明するためのサンプルアプリケーション

この機能では、PresencePlus software 2006R3かそれ以降が必要です。

イメージエクスポート・ユーティリティは、PresencePLUSのインストールCDからインストールします。

PresencePLUS Utility Program → Install Image Export Utility



このマニュアルには下記のもの組み込まれています。

- ActiveXコントロール
- ユーザーズガイド(英文)
- ヘルプファイル(英文)
- サンプルアプリケーション

本書は、ユーザーズガイドとヘルプファイルを日本語化したマニュアルです。

1.1 用語と定義

PPVS	PresencePLUS Vision Sensor (プレゼンスプラス・ビジョンセンサ)
検査ファイル	PPVSフラッシュに格納された基準イメージとツールインストラクション
イタレーション	トリガーから準備完了状態に戻るまでの1サイクル。1サイクルは、画像キャプチャ、ツールによる処理、および出力生成までです。

1.2 ActiveXの登録

Bannerイメージエクスポートコントロールをインストールするには、BannerImageExport.ocxコントロールを登録するだけです。登録するには、ocxファイルをダブルクリックし、「アプリケーションから開く」オプションを使用してWindows¥system32ディレクトリにある**regsvr32.exe**を選択します。

2. 機能

イメージエクスポート機能は下記のように動作します。

- コミュニケーションツールを追加し、「イメージ」を選択します。イーサネットTCPポートの1つを画像データ出力用に選択します。
- 任意のユーザーアプリケーションが、このTCPポートでセンサに接続します。
- コミュニケーションツールの出力フィルタとして任意のテストツールを選択した場合、このテストツールの合否基準に基づいて画像データがユーザーアプリケーションに送信されます。テストツールを何も選択しなかった場合には、検査を行うたびに画像データがユーザーアプリケーションに送信されます。
- ユーザーアプリケーションはこの画像データを受信し、表示または保存あるいはその両方を行います。



エクスポートした画像は、標準のBMP形式です。画像はイーサネットで転送されます。シリアル通信はサポートされていません。画像はセンサからPCにバイナリ形式で転送されます。Bannerが定義したヘッダ(フレーム数および画像のバイト数を格納)が、BMPデータと共に送信されます。

画像の解像度は選択することができます。デシメーション処理により、送信する画像のサイズを落とすことも可能です。選択できるオプションは、1:1(デフォルト)、4:1、16:1、または64:1です。

PPVSコミュニケーションツールはイメージエクスポートをサポートするよう機能が拡張されています。イメージエクスポート用に設定されている場合、当該コミュニケーションツール(およびTCPポート)はそれ以外のデータを出力できません。

さらにコミュニケーションツールは、通常のシリアル通信/イーサネット経由で検査結果を出力するよう拡張される予定です。最初に、この「ツール」はフレーム数を出力します。

ユーザーアプリケーションの開発を支援するため、ActiveXコントロールが提供されています。このコントロールを使用して、センサへの接続、画像データの受信、およびディスクへの画像データの保存を行うことができます。

3. ActiveXコントロール

アプリケーション開発を支援するため、ActiveXコントロールが提供されています。それには、以下の機能があります。

- 設定可能なIPアドレスとTCPポート
- センサとの接続の確立
- センサからの画像データの受信
- 画像の表示
- 画像データのディスクへの保存

3.1 プロパティ

3.1.1 Visible (Boolean)

プロパティVisibleは、受信した画像を表示するかどうかを制御します。

設定： True — 受信した画像を表示します

False — 画像は表示されません。ActiveXコントロールは透明に表示されます。

例： `ppvsControl.Visible = False` ‘コントロールを透明にします。

3.1.2 ConnectTimeout (Integer)

プロパティConnectTimeoutは、センサへの接続に成功するまでコントロールが待機する時間(単位：ms)を設定します。

設定： 50～20000ms(デフォルト：5000)

例： `ppvsControl.ConnectTimeout = 1000` 'Connect()timeoutを1秒に設定します

3.2 メソッド

3.2.1 Connect (ipAddr As String, tcpPort As Integer)

Connectメソッドは、センサへのTCP接続を確立します。このメソッドは直ちに返します。呼び出し側はイベントハンドラを使用してイベントが接続の試行に成功したかどうかを返すOnConnect()イベントをキャッチする必要があります。OnConnect()イベントは接続の試行に成功したかどうかを返します。OnConnect()イベントはタイムアウト期間以内に発生します(デフォルト3秒)。このイベントは接続の試行に成功したかどうかを示します。

OnConnectイベントはタイムアウト期間(デフォルトでは5000ms)内に発生しますが、この長さはConnectionTimeoutプロパティで設定できます。

パラメータ: String — IPアドレス(例: "192.168.0.1")
Integer — tcpPort (例: 20000)

戻り値: なし

コード例: `ppvsControl.Connect "192.168.0.1", 20000`

3.2.2 Disconnect()

Disconnectメソッドはセンサからの切断を行います。このメソッドは接続状態にかかわらず、いつ呼び出しても構いません。

パラメータ: なし

戻り値: なし

コード例: `ppvsControl.Disconnect`

3.2.3 SaveImage (filename As String)

SaveImageメソッドは受信した画像をディスクに保存します。

パラメータ: String — 画像保存先のファイル名
(例: "c:\mypics\pic_1234.bmp")

戻り値: Boolean — 画像が保存された場合はTRUE、画像の受信またはディスクへの保存が行われなかった場合にはFALSE

コード例: `ppvsControl.SaveImage "c:\mypics\pic_1234.bmp"`

3.2.4 GetNetworkState()

GetNetworkStateメソッドは、TCP接続の状態をセンサに返します(下表参照)。

コード	意味
0	コントロールはセンサに接続されていません
1	コントロールはセンサへの接続のプロセス中です
2	コントロールはセンサに接続されています

パラメータ：なし

戻り値： Integer — 接続の状態

コード例：

```
Dim netState As Integer

netState = ppvsControl.GetNetworkState
If netState = 2 Then
    Debug.Print "接続しました。"
End If
```

3.2.5 GetLastError()

GetLastErrorメソッドは、発生した最新のエラーを返します(下表参照)。GetErrorString()を使用すると、エラーの内容を説明するテキスト文字列を取得できます。

コード	意味
0	エラーなし
100	受信したパケットが小さ過ぎて、有効なヘッダとは見なされません
101	受信したヘッダが指定している画像のサイズが大き過ぎます
102	ソケットからデータを読み取れませんでした
103	ヘッダが送られてくると予想していたところ、別のデータが送られてきました
104	ソケットから読み取ろうとしたところ、既に閉じられていました
1000-10100	Winsock固有エラー。Winsockエラー一覧を参照するか、またはGetErrorStringを使用してください

パラメータ：なし

戻り値： Integer — 最新のエラー

3.2.6 GetErrorString (errCode As Integer)

GetErrorStringメソッドは、発生した最新のエラーの内容を説明するテキスト文字列を返します。

パラメータ： Integer — GetLastError()が返すエラーコード

戻り値： String — 説明文

コード例：

```
Dim errCode As Integer, errStr As String
errCode = ppvsControl.GetLastError
If Not errCode = 0 Then
    errStr = ppvsControl.GetErrorString(errCode)
    Debug.Print "受信エラー - " & errStr
End If
```

3.3 イベント

3.3.1 OnImageReceived (ByVal frameCount As Long)

イベントOnImageReceivedは、センサから画像を1つ受信すると発生します。

パラメータ： Integer — 画像のフレーム数

イベントハンドラ例：

```
Private Sub ppvsControl_OnImageReceived(ByVal frameCount As Long)
    Debug.Print "イメージ受信 - フレームカウント：" & frameCount
End Sub
```

3.3.2 OnConnect (ByVal Connected As Boolean)

イベントOnConnectはConnect()メソッドへの応答として、またはセンサが不意にソケットを閉じた場合に発生します。 イベントは、ConnectTimeoutプロパティで定義された時間内に発生します。 接続に失敗した場合、GetLastErrorを使用して具体的な理由を取得可能です。

パラメータ： Boolean — センサへの接続に成功した場合にはTRUE、それ以外の場合はFALSE

イベントハンドラ例：

```
Private Sub ppvsControl_OnConnect(ByVal Connected As Boolean)
    If Connected Then
        Debug.Print "接続成功"
    Else
        Debug.Print "接続に失敗、または閉じられた"
    End If
End Sub
```

3.3.3 OnPacketError (ByVal EventType As Integer)

イベントOnPacketErrorは、センサから受信したパケットが有効でない（破損か形式外の）場合、またはその他の接続エラー発生時に発生します。

パラメータ： Integer — EventTypeは発生したエラーを指定します（下記テーブル参照）。具体的な説明を取得するには、GetErrorString()を使用します。

イベントハンドラ例：

```
Private Sub ppvsControl_OnPacketError(ByVal EventType As Integer)
    Dim errStr As String, netState As Integer

    netState = ppvsControl.GetNetworkState

    '接続中にこのイベントが発生した場合はパケットエラーです。
    If netState = 2 Then
        errStr = ppvsControl.GetErrorString(EventType)
        Debug.Print "パケットエラー：" & errStr
    Else
        Debug.Print "接続を切断しました"
    End If
End Sub
```

3.4 センサへの接続

ActiveXコントロールを使用すると、自動およびポーリングという2通りの方法でセンサに接続できます。

自動

自動接続ではイベントOnConnectを使用して、呼び出し側にセンサへの接続試行の結果を通知します。プロパティConnectTimeoutは、接続試行が成功するまでコントロールが待機する時間を設定します。

ステップ：

- Connect()の呼び出し
- OnConnectイベントハンドラで、接続試行に成功したかどうかを確認します。

下記のVisual Basicのコードは、自動接続を行うコードです：

```
Private Sub button_Click()  
    Debug.Print "センサへ接続... "  
    ppvsControl.Connect "192.168.0.1", 20000  
End Sub  
  
Private Sub ppvsControl_OnConnect(ByVal Connected As Boolean)  
    If Connected Then  
        Debug.Print "接続成功"  
    Else  
        Debug.Print "接続失敗"  
    End If  
End Sub
```

ポーリング

ソケット接続の状態は、Connect()を呼び出した後にGetNetworkState()を繰り返し呼び出すことによりポーリングできます。アプリケーションで設定されたタイムアウト期間内に接続を確立できなかった場合、Disconnect()を呼び出してConnect()要求をキャンセル可能です。

3.5 画像データの保存

センサから受信した画像はディスクに保存できます。 イベント OnImageRecieved() をキャッチした後、このイベントハンドラ内部から SaveImage() を呼び出す方法を推奨します。

下記の Visual Basic のコードは、その一例です。

```
Private Sub ppvsControl_OnImageReceived (ByVal frameCount As Long)
    ppvsControl.SaveImage "C:\mypics\pic_" & frameCount & ".bmp"
End Sub
```

3.6 エラーコードと通信状態コード

エラーコード

コード	意味
0	ERROR_NO_ERROR (エラーなし)
100	ERROR_HEADER_TOO_SMALL (ヘッダーが小さすぎる)
101	ERROR_PAYLOAD_TOO_LARGE (ペイロードが大きすぎ)
102	ERROR_READING_FROM_SOCKET (ソケットから読み込み)
103	ERROR_INVALID_HEADER (ヘッダーが無効)
104	ERROR_SOCKET_CLOSED (ソケットが閉じた)
105	ERROR_INVALID_PACKET_VERSION (パケットバージョンが無効)

上記のエラー以外に、Winsock が生成するエラーも使用できます。たとえば、接続に失敗した場合には、GetLastError() および GetLastErrorString() を使用してその理由を取得できます。

通信状態コード

コード	意味
0	NETWORK_STATE_DISCONNECTED (ネットワークが切断された)
1	NETWORK_STATE_CONNECTING (ネットワークに接続中)
2	NETWORK_STATE_CONNECTED (ネットワークに接続)

4. イメージエクスポートフォーマット

イメージエクスポートデータは、big-endian(ビッグエンディアン)※のバイナリストリーム的一种であり、ヘッダおよびそれに続くペイロードで構成されます。ヘッダは下記に示すとおり32バイトであり、ペイロードサイズ(バイト単位)、フレーム数、ヘッダID(“PPVS”)、および予約バイトで構成されます。ペイロードはBMP形式の画像ファイルです。画像はグレイスケールで、ピクセル当たり8ビットです。

※big-endian：下位アドレスにデータの上位バイトを置くこと

サンプルアプリケーションの疑似コードは、次のとおりです。

- TCPクライアントとしてセンサへのソケットを開く
- 32バイト(ヘッダ)を読み取る
- ヘッダIDを見つけて、これがヘッダであることを確認する
- ヘッダからフレーム数とペイロードサイズを取得する
- 〈ペイロードサイズ〉バイトを読み取る
- ペイロードをBMPファイルとして保存または表示する

イメージエクスポートヘッダ

データタイプ	変数名	用法
32ビット符号なし整数	PayloadSize	バイトで表現されるペイロードのサイズ
32ビット符号なし整数	FrameCount	検査反復フレームカウント
8ビット符号なしの文字	HeaderIdentifier[4]	4文字 — “PPVS”
32ビット符号なし整数	Version	パケット形式のバージョン番号で、最初の値は1
32ビット符号なし整数	Reserved[4]	予約済みの、4つの符号なし整数



保証：製品保証期間は1年と致します。当社の責任により不具合が発生した場合、保証期間内にご返却頂きました製品については無償で修理または代替致します。ただし、お客様によりダメージを受けた場合や、アプリケーションが適切でなく製品動作が不安定な場合等は、保証範囲外とさせていただきます。

ご注意：本製品および本書の内容については、改良のため予告なく変更することがあります。