

TWAIN unit for Delphi TAcquireImage Ver 1.4 (Oct/2000)

Released as copyrighted freeware.

Programming by: Roberto Zarrelli [rzarrel@tin.it]

Comments/questions/bugs? Please write me. Public domain improvements welcome.

Please email your .pas file to Roberto, with comments or documentation to assist with the administration.

Special thanks to Achim Metzen, he developed the library (twainh.pas) that contains all the types and constants of the TWAIN protocol.

Ver 1.4 improvements for accessing specific devices by Alistair George.

History

Ver 1.0

- The component is only for Delphi 4

Ver 1.1

- Some bugs fixed.

Ver. 1.2

- Some bugs fixed.
- The component is now for Delphi 5 too
- 4 new methods added: acquire(var bmp: Tbitmap), acquireToClipboard, closeTwainSession and getSourceInfo.

Ver 1.3

- 1 new method added: acquire(var jpg: TJpegImage; compression: integer).
- 1 event added: OnTwainError.

Ver 1.4

- 2 new methods:
function GetSource(default: boolean): string,
function GetSources: TStringList;
- The component is now compatible with Delphi 3

Introduction to TWAIN

When an application uses TWAIN to acquire data, the acquisition process may be visible to the application's users in the following three areas:

The Application

The user needs to select the device from which they intend to acquire the data. They also need to signal when they are ready to have the data transferred. To allow this, TWAIN strongly recommends the application developer add two options to their File menu:

- Select Source - to select the device
- Acquire - to begin the transfer process

The Source Manager

When the user chooses the Select Source option, the application requests that the Source Manager display its Select Source dialog box. This lists all available devices and allows the user to highlight and select one device.

The Source

Every TWAIN-compliant Source provides a user interface specific to its particular device. When the application user selects the Acquire option, the Source's User Interface may be displayed.

The State-Based Protocol

The application, Source Manager, and Source must communicate to manage the acquisition of data. It is logical that this process must occur in a particular sequence. For example, the application cannot successfully request the transfer of data from a Source before the Source Manager is loaded and prepared to communicate the request.

To ensure the sequence is executed correctly, the TWAIN protocol defines seven states that exist in TWAIN sessions. A session is the period while an application is connected to a particular Source via the Source Manager. The period while the application is connected to the Source Manager is another unique session. At a given point in a session, the TWAIN elements of Source Manager and Source each occupy a particular state. Transitions to a new state are caused by operations requested by the application or Source. Transitions can be in the forward or backward direction.

The State Transition looks like this:

State 1: Pre-Session Source Manager not loaded. (Application: Load Source Manager)

State2: Source Manager Loaded. (Application: Open Source Manager)

State3: Source Manager Opened. (The user must select the source and then the application must Open the Source)

State4: Source Open. (Application: Enable Source to Acquire)

State 5: Source Enabled. (Source show user interface).

State 6: Transfer Ready.

State 7: Transferring.

The Component “AcquireImage”

This component uses the twain architecture for acquire an image from a Source to an Application.

How to use this component? It’s easy. I explain the use of this component illustrating the states of the twain architecture.

State 1: The Source Manager is not loaded.

The application may set some information with the published properties:

- *AppVersionMajorNum*: This refers to your application major revision number. (e.g. The “2” in “version 2.1”).
- *AppVersionMinorNum*: The incremental revision number of your application. (e.g. The “1” in “version 2.1”).
- *AppInfo*: General information string.
- *AppManufacturer*: The app's Manufacturer;
- *AppProductFamily*: The app's Product Family;
- *AppProductName*: The specific app Product Name;
- *Language*: The primary language for your Source or application (Appendix B);
- *Country*: The primary country where your application is intended to be distributed (Appendix C).

After this, the application must call the *function loadTwainModule*. If the result is TRUE then the Source Manager is loaded and we are in the state 2 else we remain in state 1.

State 2: Source Manager Loaded.

The application must call the *function openSourceManager*. If the result is SUCCESS then we are in the state 3 else we remain in state 2. (For the function results see the appendix A).

State 3: Source Manager Opened.

The application can call the *function GetSource(default: boolean): string*; or the *function GetSources: TStringList*.

- In the first case if default = True the default Source in the system will be selected (is the last used source), else if default = False, the user may select the Source where acquire data. The result of this function is the source name selected.
- In the second case the function *GetSources* fills a TStringList variable with all the sources available in the system (the variable doesn't need allocation).

Now we must call the *function selectSource*. Its interface is:

function selectSource(sourcenam: string): integer;

If the result is SUCCESS then we are in the state 4 else we remain in state 3. (For the function results see the appendix A).

If the application wants to know information about the source driver, I have implemented the *function getSourceInfo*.

State 4: Source Open.

The application must call the *function openSource*. If the result is SUCCESS then we are in the state 5 else we remain in state 4. (For the function results see the appendix A).

States 5 – 6 – 7: The application may call:

- the *function AcquirehDIB(var hDIB: hbitmap)*: in this case the application must pass to the function a parameter *hDIB: hbitmap*. If the result of the function is SUCCESS the image was transferred in a memory location referenced by the hDIB handle (This is an handle to a DIB (Device Independent Bitmap) image);
- the *function AcquireBMP(var bmp: TBitmap)*: in this case the application must pass to the function a parameter *bmp: Tbitmap*. If the result of the function is SUCCESS the image is in the TBitmap object;
- the *function AcquireJPG(var jpg: TjpegImage; compression: integer)*: in this case the application must pass to the function the parameter *jpg: TjpegImage* and the parameter *compression: integer* (the range for compression parameter is 1..100. Higher compression results in a poorer picture quality, but a smaller file size). If the result of the function is SUCCESS the image is in the TjpegImage object;
- *function AcquireToClipboard*: In this case, if the result of the function is SUCCESS the image was transferred in the clipboard.

If you want to scan another image you must call the *function closeTwainSession* that restart the TWAIN session to the state 2.

Closing the TWAIN session definitely: return to state 1.

If you want to exit from a TWAIN session you must:

- 1) call the *function closeTwainSession* (if SUCCESS we are in state 2)
- 2) call the *function unloadTwainModule* (if SUCCESS we are in state 1).

If the results of this functions is SUCCESS then we are in state 1.

If you want to save the image scanned into a file you can acquire the image in a TBitmap or TjpegImage object and then you may use the properties and methods of this objects (see Delphi help).

Note: Before you close the application, the image referenced by hDIB, TBitmap or TjpegImage object must be deallocated.

If there are errors when you call a function, you must see what type of error is (appendix A) and then you must return always to state 1 (see: closing TWAIN session definitely).

The *OnTwainError event*: this event has effects in the states 2..7 (see the demo program on a possible use of it).

Installation of the component

It's easy, you must install in the Delphi IDE the component `AcquireImage.pas`.

Steps:

- 1 - From Delphi menu, choose component -> install component.
- 2 - In the Unit file name, you must insert "AcquireImage.pas" with the relative path.
- 3 - You can choose to install this component into an existing package or into a new package.
- 4 - Press Ok button.
- 5 - You must first compile and after install the component.
- 6 - The component is now visible on the component's bar.

Appendix A (Error codes)

[illegible]

```
CAPSEQERROR = TWCC_CAPSEQERROR; {/ * Capability has dependency  
on other capability * / }
```

Appendix B (Language)

```
TWLG_DAN = 0; {/ * Danish * / }  
TWLG_DUT = 1; {/ * Dutch * / }  
TWLG_ENG = 2; {/ * International English * / }  
TWLG_FCF = 3; {/ * French Canadian * / }  
TWLG_FIN = 4; {/ * Finnish * / }  
TWLG_FRN = 5; {/ * French * / }  
TWLG_GER = 6; {/ * German * / }  
TWLG_ICE = 7; {/ * Icelandic * / }  
TWLG_ITN = 8; {/ * Italian * / }  
TWLG_NOR = 9; {/ * Norwegian * / }  
TWLG_POR = 10; {/ * Portuguese * / }  
TWLG_SPA = 11; {/ * Spanish * / }  
TWLG_SWE = 12; {/ * Swedish * / }  
TWLG_USA = 13; {/ * U.S. English * / }  
TWLG_USERLOCALE = -1;  
TWLG_AFRIKAANS = 14;  
TWLG_ALBANIA = 15;  
TWLG_ARABIC = 16;  
TWLG_ARABIC_ALGERIA = 17;  
TWLG_ARABIC_BAHRAIN = 18;  
TWLG_ARABIC_EGYPT = 19;  
TWLG_ARABIC_IRAQ = 20;  
TWLG_ARABIC_JORDAN = 21;  
TWLG_ARABIC_KUWAIT = 22;  
TWLG_ARABIC_LEBANON = 23;  
TWLG_ARABIC_LIBYA = 24;  
TWLG_ARABIC_MOROCCO = 25;  
TWLG_ARABIC_OMAN = 26;  
TWLG_ARABIC_QATAR = 27;  
TWLG_ARABIC_SAUDIARABIA = 28;  
TWLG_ARABIC_SYRIA = 29;  
TWLG_ARABIC_TUNISIA = 30;  
TWLG_ARABIC_UAE = 31; {/ * United Arabic Emirates * / }  
TWLG_ARABIC_YEMEN = 32;  
TWLG_BASQUE = 33;  
TWLG_BYELORUSSIAN = 34;  
TWLG_BULGARIAN = 35;  
TWLG_CATALAN = 36;  
TWLG_CHINESE = 37;  
TWLG_CHINESE_HONGKONG = 38;  
TWLG_CHINESE_PRC = 39; {/ * People's Republic of China*/}  
TWLG_CHINESE_SINGAPORE = 40;  
TWLG_CHINESE_SIMPLIFIED = 41;  
TWLG_CHINESE_TAIWAN = 42;  
TWLG_CHINESE_TRADITIONAL = 43;  
TWLG_CROATIA = 44;  
TWLG_CZECH = 45;  
TWLG_DANISH = TWLG_DAN;  
TWLG_DUTCH = TWLG_DUT;  
TWLG_DUTCH_BELGIAN = 46;  
TWLG_ENGLISH = TWLG_ENG;  
TWLG_ENGLISH_AUSTRALIAN = 47;  
TWLG_ENGLISH_CANADIAN = 48;  
TWLG_ENGLISH_IRELAND = 49;  
TWLG_ENGLISH_NEWZEALAND = 50;  
TWLG_ENGLISH_SOUTHAFRICA = 51;
```

TWLG_ENGLISH_UK = 52;
TWLG_ENGLISH_USA = TWLG_USA;
TWLG_ESTONIAN = 53;
TWLG_FAEROESE = 54;
TWLG_FARSI = 55;
TWLG_FINNISH = TWLG_FIN;
TWLG_FRENCH = TWLG_FRN;
TWLG_FRENCH_BELGIAN = 56;
TWLG_FRENCH_CANADIAN = TWLG_FCF;
TWLG_FRENCH_LUXEMBOURG = 57;
TWLG_FRENCH_SWISS = 58;
TWLG_GERMAN = TWLG_GER;
TWLG_GERMAN_AUSTRIAN = 59;
TWLG_GERMAN_LUXEMBOURG = 60;
TWLG_GERMAN_LIECHTENSTEIN = 61;
TWLG_GERMAN_SWISS = 62;
TWLG_GREEK = 63;
TWLG_HEBREW = 64;
TWLG_HUNGARIAN = 65;
TWLG_ICELANDIC = TWLG_ICE;
TWLG_INDONESIAN = 66;
TWLG_ITALIAN = TWLG_ITN;
TWLG_ITALIAN_SWISS = 67;
TWLG_JAPANESE = 68;
TWLG_KOREAN = 69;
TWLG_KOREAN_JOHAB = 70;
TWLG_LATVIAN = 71;
TWLG_LITHUANIAN = 72;
TWLG_NORWEGIAN = TWLG_NOR;
TWLG_NORWEGIAN_BOKMAL = 73;
TWLG_NORWEGIAN_NYNORSK = 74;
TWLG_POLISH = 75;
TWLG_PORTUGUESE = TWLG_POR;
TWLG_PORTUGUESE_BRAZIL = 76;
TWLG_ROMANIAN = 77;
TWLG_RUSSIAN = 78;
TWLG_SERBIAN_LATIN = 79;
TWLG_SLOVAK = 80;
TWLG_SLOVENIAN = 81;
TWLG_SPANISH = TWLG_SPA;
TWLG_SPANISH_MEXICAN = 82;
TWLG_SPANISH_MODERN = 83;
TWLG_SWEDISH = TWLG_SWE;
TWLG_THAI = 84;
TWLG_TURKISH = 85;
TWLG_UKRANIAN = 86;
TWLG_ASSAMESE = 87;
TWLG_BENGALI = 88;
TWLG_BIHARI = 89;
TWLG_BODO = 90;
TWLG_DOGRI = 91;
TWLG_GUJARATI = 92;
TWLG_HARYANVI = 93;
TWLG_HINDI = 94;
TWLG_KANNADA = 95;
TWLG_KASHMIRI = 96;
TWLG_MALAYALAM = 97;
TWLG_MARATHI = 98;
TWLG_MARWARI = 99;
TWLG_MEGHALAYAN = 100;
TWLG_MIZO = 101;
TWLG_NAGA = 102;
TWLG_ORISSI = 103;

```
TWLG_PUNJABI = 104;  
TWLG_PUSHTU = 105;  
TWLG_SERBIAN_CYRILLIC = 106;  
TWLG_SIKKIMI = 107;  
TWLG_SWEDISH_FINLAND = 108;  
TWLG_TAMIL = 109;  
TWLG_TELUGU = 110;  
TWLG_TRIPURI = 111;  
TWLG_URDU = 112;  
TWLG_VIETNAMESE = 113;
```

Appendix C (Country)

```
TWCY_AFGHANISTAN = 1001;  
TWCY_ALGERIA = 213;  
TWCY_AMERICANSAMOA = 684;  
TWCY_ANDORRA = 033;  
TWCY_ANGOLA = 1002;  
TWCY_ANGUILLA = 8090;  
TWCY_ANTIGUA = 8091;  
TWCY_ARGENTINA = 54;  
TWCY_ARUBA = 297;  
TWCY_ASCENSIONI = 247;  
TWCY_AUSTRALIA = 61;  
TWCY_AUSTRIA = 43;  
TWCY_BAHAMAS = 8092;  
TWCY_BAHRAIN = 973;  
TWCY_BANGLADESH = 880;  
TWCY BARBADOS = 8093;  
TWCY_BELGIUM = 32;  
TWCY_BELIZE = 501;  
TWCY_BENIN = 229;  
TWCY_BERMUDA = 8094;  
TWCY_BHUTAN = 1003;  
TWCY_BOLIVIA = 591;  
TWCY_BOTSWANA = 267;  
TWCY_BRITAIN = 6;  
TWCY_BRITVIRGINIS = 8095;  
TWCY_BRAZIL = 55;  
TWCY_BRUNEI = 673;  
TWCY_BULGARIA = 359;  
TWCY_BURKINAFASO = 1004;  
TWCY_BURMA = 1005;  
TWCY_BURUNDI = 1006;  
TWCY_CAMAROON = 237;  
TWCY_CANADA = 2;  
TWCY_CAPEVERDEIS = 238;  
TWCY_CAYMANIS = 8096;  
TWCY_CENTRALAFREP = 1007;  
TWCY_CHAD = 1008;  
TWCY_CHILE = 56;  
TWCY_CHINA = 86;  
TWCY_CHRISTMASIS = 1009;  
TWCY_COCOSIS = 1009;  
TWCY_COLOMBIA = 57;  
TWCY_COMOROS = 1010;  
TWCY_CONGO = 1011;  
TWCY_COOKIS = 1012;  
TWCY_COSTARICA = 506;  
TWCY_CUBA = 005;  
TWCY_CYPRUS = 357;
```

TWCY_CZECHOSLOVAKIA = 42;
TWCY_DENMARK = 45;
TWCY_DJIBOUTI = 1013;
TWCY_DOMINICA = 8097;
TWCY_DOMINCANREP = 8098;
TWCY_EASTERIS = 1014;
TWCY_ECUADOR = 593;
TWCY_EGYPT = 20;
TWCY_ELSALVADOR = 503;
TWCY_EQGUINEA = 1015;
TWCY_ETHIOPIA = 251;
TWCY_FALKLANDIS = 1016;
TWCY_FAEROEIS = 298;
TWCY_FIJIISLANDS = 679;
TWCY_FINLAND = 358;
TWCY_FRANCE = 33;
TWCY_FRANTILLES = 596;
TWCY_FRGUIANA = 594;
TWCY_FRPOLYNEISA = 689;
TWCY_FUTANAIS = 1043;
TWCY_GABON = 241;
TWCY_GAMBIA = 220;
TWCY_GERMANY = 49;
TWCY_GHANA = 233;
TWCY_GIBRALTER = 350;
TWCY_GREECE = 30;
TWCY_GREENLAND = 299;
TWCY_GRENADA = 8099;
TWCY_GRENEDINES = 8015;
TWCY_GUADELOUPE = 590;
TWCY_GUAM = 671;
TWCY_GUANTANAMOBAY = 5399;
TWCY_GUATEMALA = 502;
TWCY_GUINEA = 224;
TWCY_GUINEABISSAU = 1017;
TWCY_GUYANA = 592;
TWCY_HAITI = 509;
TWCY_HONDURAS = 504;
TWCY_HONGKONG = 852;
TWCY_HUNGARY = 36;
TWCY_ICELAND = 354;
TWCY_INDIA = 91;
TWCY_INDONESIA = 62;
TWCY_IRAN = 98;
TWCY_IRAQ = 964;
TWCY_IRELAND = 353;
TWCY_ISRAEL = 972;
TWCY_ITALY = 39;
TWCY_IVORYCOAST = 225;
TWCY_JAMAICA = 8010;
TWCY_JAPAN = 81;
TWCY_JORDAN = 962;
TWCY_KENYA = 254;
TWCY_KIRIBATI = 1018;
TWCY_KOREA = 82;
TWCY_KUWAIT = 965;
TWCY_LAOS = 1019;
TWCY_LEBANON = 1020;
TWCY_LIBERIA = 231;
TWCY_LIBYA = 218;
TWCY_LIECHTENSTEIN = 41;
TWCY_LUXENBOURG = 352;
TWCY_MACAO = 853;

TWCY_MADAGASCAR = 1021;
TWCY_MALAWI = 265;
TWCY_MALAYSIA = 60;
TWCY_MALDIVES = 960;
TWCY_MALI = 1022;
TWCY_MALTA = 356;
TWCY_MARSHALLIS = 692;
TWCY_MAURITANIA = 1023;
TWCY_MAURITIUS = 230;
TWCY_MEXICO = 3;
TWCY_MICRONESIA = 691;
TWCY_MIQUELON = 508;
TWCY_MONACO = 33;
TWCY_MONGOLIA = 1024;
TWCY_MONTSEERRAT = 8011;
TWCY_MOROCCO = 212;
TWCY_MOZAMBIQUE = 1025;
TWCY_NAMIBIA = 264;
TWCY_NAURU = 1026;
TWCY_NEPAL = 977;
TWCY_NETHERLANDS = 31;
TWCY_NETHANTILLES = 599;
TWCY_NEVIS = 8012;
TWCY_NEWCALEDONIA = 687;
TWCY_NEWZEALAND = 64;
TWCY_NICARAGUA = 505;
TWCY_NIGER = 227;
TWCY_NIGERIA = 234;
TWCY_NIUE = 1027;
TWCY_NORFOLKI = 1028;
TWCY_NORWAY = 47;
TWCY_OMAN = 968;
TWCY_PAKISTAN = 92;
TWCY_PALAU = 1029;
TWCY_PANAMA = 507;
TWCY_PARAGUAY = 595;
TWCY_PERU = 51;
TWCY_PHILLIPPINES = 63;
TWCY_PITCAIRNIS = 1030;
TWCY_PNEWGUINEA = 675;
TWCY_POLAND = 48;
TWCY_PORTUGAL = 351;
TWCY_QATAR = 974;
TWCY_REUNIONI = 1031;
TWCY_ROMANIA = 40;
TWCY_RWANDA = 250;
TWCY_SAIPAN = 670;
TWCY_SANMARINO = 39;
TWCY_SAOTOME = 1033;
TWCY_SAUDIARABIA = 966;
TWCY_SENEGAL = 221;
TWCY_SEYCHELLESIS = 1034;
TWCY_SIERRALEONE = 1035;
TWCY_SINGAPORE = 65;
TWCY_SOLOMONIS = 1036;
TWCY_SOMALI = 1037;
TWCY_SOUTHAFRICA = 27;
TWCY_SPAIN = 34;
TWCY_SRILANKA = 94;
TWCY_STHELENA = 1032;
TWCY_STKITTS = 8013;
TWCY_STLUCIA = 8014;
TWCY_STPIERRE = 508;

TWCY_STVINCENT = 8015;
TWCY_SUDAN = 1038;
TWCY_SURINAME = 597;
TWCY_SWAZILAND = 268;
TWCY_SWEDEN = 46;
TWCY_SWITZERLAND = 41;
TWCY_SYRIA = 1039;
TWCY_TAIWAN = 886;
TWCY_TANZANIA = 255;
TWCY_THAILAND = 66;
TWCY_TOBAGO = 8016;
TWCY_TOGO = 228;
TWCY_TONGAIS = 676;
TWCY_TRINIDAD = 8016;
TWCY_TUNISIA = 216;
TWCY_TURKEY = 90;
TWCY_TURKSCAICOS = 8017;
TWCY_TUVALU = 1040;
TWCY_UGANDA = 256;
TWCY_USSR = 7;
TWCY_UAEMIRATES = 971;
TWCY_UNITEDKINGDOM = 44;
TWCY_USA = 1;
TWCY_URUGUAY = 598;
TWCY_VANUATU = 1041;
TWCY_VATICANCITY = 39;
TWCY_VENEZUELA = 58;
TWCY_WAKE = 1042;
TWCY_WALLISIS = 1043;
TWCY_WESTERNSAHARA = 1044;
TWCY_WESTERNSAMOA = 1045;
TWCY_YEMEN = 1046;
TWCY_YUGOSLAVIA = 38;
TWCY_ZAIRE = 243;
TWCY_ZAMBIA = 260;
TWCY_ZIMBABWE = 263;
TWCY_ALBANIA = 355;
TWCY_ARMENIA = 374;
TWCY_AZERBAIJAN = 994;
TWCY_BELARUS = 375;
TWCY_BOSNIAHERZGO = 387;
TWCY_CAMBODIA = 855;
TWCY_CROATIA = 385;
TWCY_CZECHREPUBLIC = 420;
TWCY_DIEGOGARCIA = 246;
TWCY_ERITREA = 291;
TWCY_ESTONIA = 372;
TWCY_GEORGIA = 995;
TWCY_LATVIA = 371;
TWCY_LESOTHO = 266;
TWCY_LITHUANIA = 370;
TWCY_MACEDONIA = 389;
TWCY_MAYOTTEIS = 269;
TWCY_MOLDOVA = 373;
TWCY_MYANMAR = 95;
TWCY_NORTHKOREA = 850;
TWCY_PUERTORICO = 787;
TWCY_RUSSIA = 7;
TWCY_SERBIA = 381;
TWCY_SLOVAKIA = 421;
TWCY_SLOVENIA = 386;
TWCY_SOUTHKOREA = 82;
TWCY_UKRAINE = 380;

```
TWCY_USVIRGINIS = 340;  
TWCY_VIETNAM = 84;
```