

Barcode 1D Encode&Decode SDK v2.5

USER MANUAL

AIPSYS Software Laboratory

<http://www.aipsys.com>

Last Updated 15th March 2010

1.	1D Barcode introduction	4
1.1.	1D Barcode types	4
1.2.	License	4
1.3.	About trial version	12
2.	Encoder SDK	12
2.1.	Static link library	12
2.1.1	Data structure and constant for barcode type	12
2.1.2.	Function or procedure	14
2.1.2.1.	_InitBarcodeContext	14
2.1.2.2.	_1DBarEncode2File	15
2.1.2.3.	_1DBarEncode2Bitmap	15
2.1.2.4.	_FreeBarcodeContext	15
2.1.3.	Example for Microsoft visual C++	16
2.2.	Dynamic link library	16
2.2.1.	Data structure and constant for barcode type	16
2.2.2.	Function or procedure	18
2.2.2.1.	InitWorkSpace	18
2.2.2.2.	Bar1DEncode2File	19
2.2.2.3.	Bar1DEncode2Bitmap	19
2.2.2.4.	FreeWorkSpace	19
2.2.3.	Example for Microsoft visual C++	20
2.2.5.	Example for Microsoft visual Basic	20
2.2.5.1.	Redeclaration of the data type and function	20
2.2.5.2.	Example	22
2.3.	ActiveX	23
2.3.1.	Barcode type id constants	23
2.3.2.	textAlign values	24
2.3.3.	Properties	28
2.3.3.1.	charSet	28
2.3.3.2.	outputOptions	28
2.3.3.3.	charSize	28
2.3.3.4.	whiteSpaceWidth	28
2.3.3.5.	rotateAngle	28
2.3.3.6.	Margin	28
2.3.3.7.	EncodeData	29
2.3.3.10.	foreColor	29
2.3.3.11.	backColor	29
2.3.2.	Methods	29
2.3.2.1.	Encode2ImageFile	29
2.3.3.	Register activeX component	29
2.3.6.	Example for Microsoft visual Basic	30
2.4.	ASP Control for server side	30
2.4.1.	Barcode type id constants	30

2.4.2. textAlign values	30
2.4.3. Properties	30
2.4.3.1. charSet	30
2.4.3.2. outputOptions	30
2.4.3.3. charSize	31
2.4.3.4. whiteSpaceWidth	31
2.4.3.5. rotateAngle	31
2.4.3.6. Margin	31
2.4.3.7. EncodeData	31
2.4.3.10. foreColor	31
2.4.3.11. backColor	32
2.4.2. Methods	32
2.4.2.1. InitWorkspace	32
2.4.2.2. FreeWorkspace	32
2.3.2.3. Encode2File	32
2.4.3. Register the ASP server component	33
2.4.4. Example for ASP	33
3. Decoder SDK	34
3.1. Static link library	34
3.1.1. Result structure for decoding	34
3.1.2. Function or procedure	34
3.1.2.1. _Barcode1DDecodeImageFile	34
3.1.2.2. _Barcode1DDecodeBitmap	35
3.1.2.3. _Barcode1DFree	35
3.1.2.4. _LoadImageEx	36
3.1.3. Examples	36
3.2. Dynamic link library	37
3.2.1. Result structure for decoding	37
3.2.2. Function or procedure	37
3.2.2.1. Barcode1DDecodeImageFile	37
3.2.2.2. Barcode1DDecodeBitmap	38
3.1.2.3. _Barcode1DFree	38
3.1.2.4. LoadImageEx	39
3.1.3. Examples	39
4. Order Information	40
5. Affiliate program	42
6. Support Information	44
7. Product Information Link	44

1. 1D Barcode introduction

1.1. 1D Barcode types

- 2 of 5 family (including IATA 2 of 5, Matrix 2 of 5 and China Postal)
- Australia Post
- Codabar
- Codablock F
- Code 128 and EAN/UCC-128
- Code 39
- Code 49
- Code 93
- DataBar (formerly RSS)
- Deutschen Post
- DUN-14
- EAN/UCC (EAN-13, EAN-8)
- ISBN
- ITF (including ITF-6, ITF-14 and Interleaved 2 of 5)
- Japan Post
- Korean Postal Authority
- MSI
- OneCode
- Plessey
- PostNet
- RM4SCC and 4-State
- GS1-DataBar
- SSCC (Serial Shipping Container Code)
- SISAC
- Telepen
- UPC (including UPC-A and UPC-E)

1.2. License

AIPSYS SOFTWARE LICENSE AGREEMENT

READ THE TERMS OF THIS SOFTWARE LICENSE AGREEMENT (HEREINAFTER THE "AGREEMENT") CAREFULLY. BY DOWNLOADING, INSTALLING, IMPLEMENTING OR USING THIS SOFTWARE PRODUCT, YOU AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE

AS ANY WRITTEN AGREEMENT NEGOTIATED AND SIGNED BY YOU AND AIPSYS.COM INCORPORATED (HEREINAFTER "**AIPSYS SOFTWARE**"). IF YOU ARE ACCESSING SOFTWARE ELECTRONICALLY, INDICATE YOUR ACCEPTANCE OF THESE TERMS BY SELECTING THE "ACCEPT" (OR EQUIVALENT) BUTTON. IF YOU DO NOT AGREE TO ALL OF THE TERMS, PROMPTLY RETURN THE UNUSED SOFTWARE TO YOUR PLACE OF PURCHASE FOR A REFUND OR, IF SOFTWARE IS ACCESSED ELECTRONICALLY, SELECT THE "DECLINE" (OR EQUIVALENT) BUTTON.

NOW, THEREFORE, IN CONSIDERATION OF THE MUTUAL PROMISES SET FORTH HEREIN, AIPSYS SOFTWARE AND YOU HEREBY AGREE AS FOLLOWS:

DEFINITIONS:

- (a) "**You**" shall mean the individual using, implementing, downloading, or installing the underlying Software. In the event You are using, implementing, downloading, or installing the underlying Software on behalf of an Organization, all liability for a breach of this agreement shall be the responsibility of said Organization.
- (b) "**Licensee**" shall mean You together with any Organization You may be representing, or any related agent, employee, or representative of You that has downloaded, used, installed, or implemented the software package on Your behalf.
- (a) "**Software**" shall mean any and all computer programs produced, created, developed, or provided by AIPSYS, including, but not limited to, applicable programs, fonts, components, hosted services, source code, modules, corresponding documentation, updates, upgrades, or modifications thereto.
- (b) "**Developer**" shall mean an individual who has a primary job function of developing software applications.
- (c) "**Server**" shall mean a computer system that multiple users access or make use of, including but not limited to, terminal servers, file servers, application servers or web servers.
- (d) "**Source Code Agreement**" shall mean a separate written instrument governing the use and rights to the underlying Software.
- (e) "**Effective Users**" shall mean the number of users that are effective for software licensing, which is determined by the following method that returns the greatest number: (1) The number of users that have access to the Software, (2) The number of computers on which the Software is installed, (3) The number of printers that are being printed to with the Software, or (4) Where the Software is used on a [Server](#) or run from a Server, the number of users per week that have access to the Software on the Server, or (5) the number of users per week that have access to programs making use of the Software on the server.
- (f) "**Affiliate Program**" shall mean the automated sales referral program described at [affiliates program](#).
- (g) "**Organization**" shall mean a single company, business unit, entity or individual. In this Agreement, each subsidiary of a company or business unit with a separate Tax ID is considered a separate Organization.

(h) "User" shall mean a single person that is making use of the Software.

TERMS:

1. License Grant

In consideration for the license fee paid, and other good and valuable consideration, AIPSYS grants to Licensee only, unless otherwise limited by the license purchased or granted, the nonexclusive, nontransferable, perpetual, world-wide right to use the Software in accordance with this Agreement and the license defined herein that Licensee purchases ("**License**"). If You are installing, accessing or using this Software for Your employer, this Agreement also includes Your employer. Licensee may only use the Software according to the License purchased or granted by AIPSYS. AIPSYS offers several license types to meet the needs of different Organizations and implementations. Particular Licenses are offered for each product depending on the intended use of the Software. AIPSYS offers some Licenses that are granted to Licensee by this Agreement and not purchased; these include the Optional Integration License, Evaluation License, Free License and the Beta License.

A. Site License - allows use of the Software for all users at a single site within a single [Organization](#). Because of the discounts associated with this license, technical support is provided to a single technical contact at Licensee's [Organization](#) instead of to each individual user.

B. Multi Site License - allows use of the Software for an unlimited number of users at an unlimited number of sites within a single [Organization](#). Because of the discounts associated with this license, technical support is provided to a single technical contact at Licensee's Organization instead of to each individual user.

C. Developer Licenses

Developer Licenses offer royalty free use of the Software internally (within the same [Organization](#)) and externally (outside the [Organization](#) bundled with an application) according to the Developer License Distribution Terms. This license type is licensed by the number of Developers that will be using or working with the Software. The following types of Developer Licenses are available:

(1). One Developer License

The One Developer License ("1DL") allows royalty-free distribution and use of the Software internally (in the same [Organization](#)) and externally (outside the [Organization](#)) for a single Developer and up to 10,000 user licenses according to [Effective Users](#), provided Licensee adheres to the [Developer License Distribution Terms](#).

(2). Five Developer License

The Five Developer License ("5DL") allows royalty-free distribution and use of the Software internally (in the same [Organization](#)) and externally (outside the [Organization](#)) for up to five Developers and up to 20,000 user licenses according to

[Effective Users](#), provided Licensee adheres to the [Developer License Distribution Terms](#). This license is also granted if two 1DLs are purchased.

(3). Unlimited Developer License

The Unlimited Developer License (“UDL”) allows complete royalty-free distribution and use of the Software internally (in the same [Organization](#)) and externally (outside the [Organization](#)) for an unlimited number of developers, servers and other user licenses, provided Licensee adheres to the [Developer License Distribution Terms](#).

(4). Small Company Developer Licenses

The Small Company Developer License (“SCDL”) grants all rights of the applicable Developer License to all [Organizations](#) with a gross annual revenue or funding of less than 2 million U.S. Dollars (or equivalent amount in a foreign currency) with a signed Small Company Agreement. All rights of the Five Developer License are granted if 2 SCDLs are purchased and all rights of the Unlimited Developer License are granted if 3 SCDLs are purchased.

D. Single User License

The Single User License ("SUL") allows use of the Software for one User in Licensee's [Organization](#) according to [Effective Users](#).

The SUL shall not be used in connection with: (1) A high speed printer that prints over 55 Pages Per Minute, or (2) A system (including all hardware, printer and software) having a cost totaling over 50,000 USD or equivalent amount in a foreign currency. Such use requires a Developer License, Site License or Multi Site License. The more single user licenses that are purchased, the more users that are allowed:

- 1 Single User License = 1 licensed user
- 2 Single User Licenses = 5 licensed users
- 3 Single User Licenses = 25 licensed users
- 4 Single User Licenses = 50 licensed users
- 5 Single User Licenses = 100 licensed users
- 6 Single User Licenses = Developer License Granted for 10,000 licensed users

E. Multiple User Licensing

Multiple user licenses grant the rights of the Single User License for a particular number of Users. For example, a 5 User License grants the rights for 5 Single User Licenses. These licenses may be combined; for example, 1 Single User License and a 10 User License = 11 licensed users.

F. Developer use with the Single User License

Developer use with the Single User License requires at least a 5 User License to be purchased unless the Developer and end User are the same person. A Developer may integrate the Software into an application if the Developer is not the end User, provided the Developer uses one License and the end User uses another License. If more than one Developer uses the Software, Licensee must purchase a Developer License for each additional Developer.

G. Single Server License

The Single Server License (“SSL”) allows use of the Software on one (1) server in Licensee's [Organization](#), where a single Server may have only 1 CPU core and up to 100 unique User accesses to the Software from the Server per day. A SSL is required for each additional Server, or CPU core. Additional SSLs may also be obtained for the same server to increase the requirements. For example, 2 SSLs allow 2 CPU cores and up to 200 unique User accesses to the Software per day on the same Server. If 4 SSLs are purchased for the same Software, the rights of the Developer License are granted. If the Software is not used on a server, the licensing options of the Single User License may be used where 1 SSL equals 1 Single User License. The SSL shall not be used in connection with: (1) A high speed printer that prints over 55 Pages Per Minute, or (2) A system (including all hardware, printer and software) having a cost totaling over 50,000 USD or equivalent in a foreign currency. Such use requires a Developer License, Site License or Multi Site License.

ANY OTHER USE REQUIRES A PURCHASE FOR THE ASSOCIATED PRODUCT LICENSE, AFTER A PERIOD OF 30 DAYS, WHICH IS GRANTED TO LICENSEE FOR EVALUATION PURPOSES ONLY.

H. Evaluation License

Software that is distributed as shareware or a demo version may only be used for testing and evaluation purposes only for a period of 30 days.

J. Beta License

Software that is distributed as a beta version may be used during the beta testing period and up to 30 days after the official release is available.

K. Developer License Distribution Terms

As used in this section, the term ("User Licenses") shall mean the number of Users that Licensee's License allows according to the definition of [Effective Users](#). The Developer License allows 10,000 [Effective Users](#), The 5 Developer License allows 20,000 [Effective Users](#) and the Unlimited Developer License allows an unlimited number of [Effective Users](#).

(a). Internal Distribution:

Allows use of the Software in Licensee's Organization, provided Licensee does not exceed its User Licenses and Licensee adheres to the following terms:

- (1) If distribution of Licensee's application exceeds its User Licenses, additional Developer Licenses are required; each Developer License purchased will allow distribution of an additional 10,000 [Effective Users](#). Royalty-free, unlimited distribution is granted after purchasing three Developer Licenses or the Unlimited Developer License.
- (2) If more than one Developer is developing with the Software, an additional Developer License is required. Up to 5 Developers may use or develop with the

Software after purchasing an additional Developer License or the 5 Developer License. Unlimited Developer use is granted after purchasing three Developer Licenses or the Unlimited Developer License.

(3) The Software may be used on any number of Servers, provided that the number of users accessing all of the servers does not exceed Licensee's User Licenses.

(b). External Distribution:

Allows Licensee to rent, lease or distribute the Software outside its Organization bundled with an application, provided Licensee does not exceed its User Licenses and Licensee adheres to the following terms:

(1) If more than one Developer is developing with the Software, an additional Developer License is required. Up to 5 developers may develop with the Software after purchasing an additional Developer License or the 5 Developer License. Unlimited Developer use is granted after purchasing three Developer Licenses or the Unlimited Developer License.

(2) Licensee may not resell, rent, lease or distribute the Software alone. The Software must be distributed as a component of an application and bundled with an application or with the application's installation files. The Software may only be used as part of, and in connection with, the bundled application. If the Unlimited Developer License is purchased, Licensee may embed the Software into Licensee's firmware, provided a copyright notice is added in the firmware or documentation as detailed in number 5 of this section.

(3) Licensee may not resell, rent, lease, distribute or otherwise use the Software for the License that was purchased, in any way that would compete with AIPSYS. If it is determined by AIPSYS or Licensee that Licensee's distribution or use of the Software competes with AIPSYS, a reasonable royalty fee for Licensee's distribution or use of the Software must be negotiated and agreed to by Licensee and AIPSYS and paid to AIPSYS each quarter or another agreed upon interval of time.

(4) If Licensee uses the Software internally within its Organization, Licensee shall deduct the quantity of its User Licenses used within its Organization from the total number of its User Licenses that are distributed outside its Organization. For example, if Licensee has a One Developer License and uses 4,000 User Licenses internally, it may only distribute up to 6,000 User Licenses outside its Organization.

(5) A valid copyright notice must be provided within the user documentation, start-up screen or in the help-about section of Licensee's application that specifies AIPSYS as the provider of the Software bundled with its application, for example: "<<your application name>> contains barcode components licensed from AIPSYS.com, Inc. These products may only be used as part of and in connection with <<your application name>>."

(6) Licensee's User Licenses are counted by the number of users the Licensee's bundled product is licensed for, with the exception that if its product is licensed for more than 500 users per copy, only 500 licenses of the Developer License are used per copy distributed. For example, if Licensee distributes 1 application licensed for 25 users, then that distribution used 25 User Licenses. If Licensee distributes an

application that is licensed to be used on a server or host system in such a way that 900 users use it, then that application only uses 500 of its User Licenses.

2. Registration

If Licensee purchases the License directly from AIPSYS, registration is automatic. If Licensee purchases the License from a reseller, Licensee must register the License at www.aipsys.com/register/ before technical support or upgrades for the Software can be made available.

3. Copyright

By downloading, installing, using, or implementing this Software, Licensee acknowledges the validity and enforceability of AIPSYS's copyright in the underlying software and code. The Software and the accompanying materials are licensed, not sold, to Licensee. AIPSYS maintains ownership of all copyright interests in the Software, including any derivative works based upon the Software. Licensee may not rent, lease, display or distribute copies of the Software to others except under the conditions of this Agreement. Unauthorized copying of the Software or accompanying materials even if modified, merged, or included with other software, or of the written materials, is expressly forbidden. Licensee may be held legally responsible for any infringement of intellectual property rights that is caused or encouraged by Licensees failure to abide by the terms of this Agreement. Licensee may make copies of the Software as needed for development and use provided that the number of copies made do not exceed the number of users allowed by the License purchased. Licensee may also make a reasonable number of archival copies of the Software for backup and recovery purposes. In any case, when a copy is created, any copyright notices included in the Software must be reproduced in their entirety on the copy.

4. Software Modifications

If the Unlimited Developer License is purchased, Licensee may modify any portions of the Software as needed, provided that copyright notices are not removed, including but not limited to the height, width and tables of any fonts provided.

5. Agreement Duration and Termination

Subject to the terms and conditions of this Agreement, this Agreement begins when the Software is downloaded, installed, used or when a License for Software is purchased or granted and is perpetual unless terminated. When the Agreement begins, this Agreement shall supersede all older versions of this Agreement including any older Agreements that may be embedded in the Software. This Agreement shall inure to the benefit of and be binding upon AIPSYS and Licensee. Licensee may terminate this Agreement at any time by returning the Software to AIPSYS and destroying all copies thereof. This Agreement shall terminate upon notice from AIPSYS if Licensee fails to comply with any provision contained herein or if the funds paid for the license are refunded or are not received, and such failure or breach is not cured within thirty

(30) days of such notice. Upon termination, Licensee must destroy the Software and all copies (in part and in whole, including modified copies, if any) in its possession or control. AIPSYS reserves the right to terminate this Agreement if the use of Software by Licensee causes a loss of revenue for AIPSYS that exceeds ten (10) times the amount Licensee paid for the License. Termination of this Agreement shall not affect the Software bundled and distributed with an application under the Developer License by Licensee prior to termination, provided Licensee has purchased a Developer License for the Software, the bundled application does not compete with AIPSYS in any way, and funds for the License were received and not returned or refunded in any way. All restrictions prohibiting Licensee's use of the Software and intellectual property provisions relating to Software to the benefit of AIPSYS shall survive termination of this Agreement.

6. Warranty and Limitation of Liability

Although efforts have been made to assure that the Software is date compliant, correct, reliable, technically accurate and will perform in accordance with the documentation, the Software is licensed to Licensee as is and without warranties as to performance of merchantability, fitness for a particular purpose or use, or any other warranties whether expressed or implied. Licensee, its Organization, and all users of the Software, assume all risks when using it. To the maximum extent permitted by applicable law, in no event shall AIPSYS be liable for any consequential, incidental, indirect, punitive or special damages arising out of the use of or inability to use the Software or the provision of or failure to provide support services or hosted services, even if AIPSYS has been advised of the possibility of such damages. In any case, AIPSYS's entire liability under any provision of this Agreement shall be limited to ten (10) times the amount actually paid by Licensee for the License or \$5.00 USD if no license was purchased.

7. Technical Support and Product Upgrades

Unless otherwise indicated in the documentation of the Software, AIPSYS offers a free Priority Support and Product Upgrade Subscription for a period of thirty (30) days from the date of purchase on all licensed Software. When Licensee's Priority Support is active, Licensee may contact AIPSYS by phone, email and through the Online Priority Support Request Form. Priority Support and Product Upgrades may be provided beyond thirty (30) days if the Priority Support and Upgrade Subscription is purchased. Support may be provided to the appropriate individual that (a) ordered the License; (b) is integrating the Software; (c) a Developer; or (d) the end user if each end user has a separate License for the Software. If one Developer License is purchased, technical support is provided for only one Developer. If the 5 Developer License is purchased, technical support is provided for up to 5 Developers. If the Unlimited Developer License is purchased, technical support is provided for an unlimited number of Developers. The Developers responsibilities may be transferred to another Developer within the Organization as necessary provided no more than 2 transfers occur within any ninety (90) day period. If Licensee's Priority Support and

Product Upgrade Subscription expires, Licensee may obtain free technical support by referring to support documents at the website or by renewing the Priority Support and Product Upgrade Subscription.

Whenever any Software update, upgrade, or revision is provided to Licensee or Licensee purchases an additional License, all related Software from AIPSYS (including any Software that was acquired previously) shall be covered by the latest version of the Agreement that exists at the time the most recent update was provided to Licensee.

1.3. About trial version

With 2D barcode encoder and decoder SDK, some of the input element will be replaced with char '*' before encoding, and some of the output element will be replaced with '*' after decoding.

With 1D linear barcode encoder and decoder, some of the input element will be replaced with char '0' before encoding, and some of the output element will be replaced with '0' after decoding.

The Trial version have 30 days' evaluation time, you must remove it from your computer and your application after expiration.

We will mail the licensed version or register serial no to you after you order it.

2. Encoder SDK

2.1. Static link library

2.1.1 Data structure and constant for barcode type

The following data structure define the properties of the 1D barcode, it can be transfer into function as parameter.

```

typedef struct
{
    int    symbology;        //[IN]barcode id
    int    height;           //[OUT] barcode height
    int    whiteSpaceWidth; //[IN] size of white space
    int    borderWidth;     //[IN] size of border width
    int    outputOptions;   //[IN]
    int    rotateAngle;     //[IN]angle of rotatin
    int    rows;            //[OUT]
    int    width;           //[OUT]
    int    charSize;        //[IN]char size when print string to barcode
    int    textAlign;       //[IN]alignment mode printing string to barcode
    char *charSet;          //[IN] set the char set
    char *prefix;           //[IN] set the prefix of the string printed
    char *surfix;           //[IN] set the surfix of the string printed
    COLORREF foreColor;     //[IN]barcode color
    COLORREF backColor;     //[IN]background color
    char text[100];         //[IN]data to be encoded
    char encodedData[8][1000]; //[OUT]matrix outputed
    char errtxt[100];       //[OUT] error message
} _SYMBOL;

```

Constants for barcode type

BARCODE_CODE11	1
BARCODE_C25MATRIX	2
BARCODE_C25INTER	3
BARCODE_C25IATA	4
BARCODE_C25LOGIC	6
BARCODE_C25IND	7
BARCODE_CODE39	8
BARCODE_EXCODE39	9
BARCODE_EANX	13
BARCODE_EAN128	16
BARCODE_CODABAR	18
BARCODE_CODE128	20
BARCODE_DPLEIT	21
BARCODE_DPIDENT	22
BARCODE_CODE16K	23
BARCODE_CODE93	25
BARCODE_FLAT	28
BARCODE_TELEPEN	32
BARCODE_UPCA	34
BARCODE_UPCE	37
BARCODE_POSTNET	40

BARCODE_MSI_PLESSEY	47
BARCODE_FIM	49
BARCODE_LOGMARS	50
BARCODE_PHARMA	51
BARCODE_PZN	52
BARCODE_PHARMA_TWO	53
BARCODE_CODE128B	60
BARCODE_AUSPOST	63
BARCODE_AUSREPLY	66
BARCODE_AUSROUTE	67
BARCODE_AUSREDIRECT	68
BARCODE_ISBNX	69
BARCODE_RM4SCC	70
BARCODE_ITF14	72
BARCODE_NVE18	75
BARCODE_KOREAPOST	77
BARCODE_PLANET	82
BARCODE_ONECODE	85
BARCODE_PLESSEY	86
BARCODE_TELEPEN_NUM	100
BARCODE_MSI_10	101
BARCODE_MSI_10_10	102
BARCODE_MSI_11	103
BARCODE_MSI_11_10	104
BARCODE_CODE39_43	105
BARCODE_EXCODE39_43	106

2.1.2. Function or procedure

2.1.2.1. _InitBarcodeContext

The _InitBarcodeContext function initialize the environment of 1d Barcode encoding with default value.

```
void __stdcall __InitBarcodeContext(_SYMBOL *pSymbol);
```

Parameters

pSymbol [in] define the 1D Barcode attributes for encoding, refer structure type _SYMBOL

Return values

None

2.1.2.2. **_1DBarEncode2File**

The `_1DBarEncode2File` function encode the data inputed with the defined attributes and save the barcode to an image file

```
BOOL __stdcall _1DBarEncode2File (_SYMBOL *pSymbol, LPCTSTR  
pImageFile);
```

Parameters

pSymbol

[in] define the 1D Barcode attributes for encoding, refer structure type
`_SYMBOL`

pImageFile

[in] define the image file outputted, currently bitmap image supported

Return values

If the function succeeds, the return value is TRUE, otherwise , return FALSE.

2.1.2.3. **_1DBarEncode2Bitmap**

The `_1DBarEncode2Bitmap` function encode the data inputed with the defined attributes and return the bitmap handle of the 1D barcode image

```
HBITMAP __stdcall _1DBarEncode2Bitmap (_SYMBOL *pSymbol);
```

Parameters

pSymbol

[in] define the 1D Barcode attributes for encoding, refer structure type
`_SYMBOL`

Return values

If the function succeeds, the return value is BITMAP handle of 1D Barcode, otherwise , return NULL.

2.1.2.4. **_FreeBarcodeContext**

The `_FreePDF417Context` function free environment of the 1D encoding

```
BOOL __stdcall _FreeBarcodeContext ();
```

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.1.3. Example for Microsoft visual C++

Example1

```
#include "1DBarEncodeLib.h"

.....

_SYMBOL symbol;

_InitBarcodeContext(&symbol);
symbol.symbology = 20;
symbol.borderWidth=10;
// symbol.outputOptions;
symbol.rotateAngle = 0;
// symbol.textAlign = CT_L;
symbol.charSet = "arial";
symbol.prefix = NULL;
symbol.suffix = NULL;
symbol.foreColor = RGB(255,0,0);
symbol.backColor = RGB(255,255,255);
_1DBarEncode2File(&symbol,"c:\\1d.bmp");
_FreeBarcodeContext();
....
```

Link library

1DBarcodeLib.lib

2.2. Dynamic link library

2.2.1. Data structure and constant for barcode type

```
typedef struct tagSYMBOL
{
```



```

int    symbology;    //[IN]barcode id
int    height;       //[OUT] barcode height
int    whiteSpaceWidth; //[IN] size of white space
int    borderWidth;  //[IN] size of border width
int    outputOptions; //[IN]
int    rotateAngle;  //[IN]angle of rotatin
int    rows;         //[OUT]
int    width;        //[OUT]
int    charSize;     //[IN]char size when print string to barcode
int    textAlign;    //[IN]alignment mode printing string to barcode
char *charSet;       //[IN] set the char set
char *prefix;        //[IN] set the prefix of the string printed
char *surfix;        //[IN] set the surfix of the string printed
COLORREF foreColor;  //[IN]barcode color
COLORREF backColor;  //[IN]background color
char text[100];      //[IN]data to be encoded
char encodedData[8][1000]; //[OUT]matrix outputed
char errtxt[100];    //[OUT] error message
} SYMBOL;

```

Constants for barcode

BARCODE_CODE11	1
BARCODE_C25MATRIX	2
BARCODE_C25INTER	3
BARCODE_C25IATA	4
BARCODE_C25LOGIC	6
BARCODE_C25IND	7
BARCODE_CODE39	8
BARCODE_EXCODE39	9
BARCODE_EANX	13
BARCODE_EAN128	16
BARCODE_CODABAR	18
BARCODE_CODE128	20
BARCODE_DPLEIT	21
BARCODE_DPIDENT	22
BARCODE_CODE16K	23
BARCODE_CODE93	25
BARCODE_FLAT	28
BARCODE_TELEPEN	32
BARCODE_UPCA	34
BARCODE_UPCE	37
BARCODE_POSTNET	40
BARCODE_MSI_PLESSEY	47
BARCODE_FIM	49

BARCODE_LOGMARS	50
BARCODE_PHARMA	51
BARCODE_PZN	52
BARCODE_PHARMA_TWO	53
BARCODE_CODE128B	60
BARCODE_AUSPOST	63
BARCODE_AUSREPLY	66
BARCODE_AUSROUTE	67
BARCODE_AUSREDIRECT	68
BARCODE_ISBNX	69
BARCODE_RM4SCC	70
BARCODE_ITF14	72
BARCODE_NVE18	75
BARCODE_KOREAPOST	77
BARCODE_PLANET	82
BARCODE_ONECODE	85
BARCODE_PLESSEY	86
BARCODE_TELEPEN_NUM	100
BARCODE_MSI_10	101
BARCODE_MSI_10_10	102
BARCODE_MSI_11	103
BARCODE_MSI_11_10	104
BARCODE_CODE39_43	105
BARCODE_EXCODE39_43	106

2.2.2. Function or procedure

2.2.2.1. InitWorkSpace

The InitWorkSpace function initialize the environment of 1D barcode encoding with default value.

```
void __stdcall InitWorkSpace(SYMBOL *pSymbol);
```

Parameters

pSymbol

[in] define the 1D barcode attributes for encoding, refer structure type
SYMBOL

Return values

None

2.2.2.2. Bar1DEncode2File

The PDF417Encode2File function encode the data inputed with the defined attributes and save the barcode to an image file

```
BOOL __stdcall Bar1DEncode2File (SYMBOL *pSymbol,  
                                LPCTSTR lpImageFile);
```

Parameters

pSymbol

[in] define the 1D barcode attributes for encoding, refer structure type

SYMBOL

lpImageFile

[in] define the image file outputted, currently bitmap image supported

Return values

If the function succeeds, the return value is TRUE, otherwise , return FALSE.

2.2.2.3. Bar1DEncode2Bitmap

The Bar1DEncode2Bitmap function encode the data inputed with the defined attributes and return the bitmap handle of the 1D barcode barcode image

```
HBITMAP __stdcall Bar1DEncode2Bitmap (SYMBOL *pSymbol);
```

Parameters

pSymbol

[in] define the 1D Barcode attributes for encoding, refer structure type

SYMBOL

Return values

If the function succeeds, the return value is BITMAP handle of PDF417 Barcode, otherwise , return NULL.

2.2.2.4. FreeWorkSpace

The FreeWorkSpace function free environment of the 1D barcode encoding

```
BOOL __stdcall FreeWorkSpace();
```

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.2.3. Example for Microsoft visual C++

Example1

Example1

```
#include "1DBarEncodeDll.h"

.....

SYMBOL symbol;

InitWorkSpace(&symbol);
symbol.symbology = 20;
symbol.borderWidth=10;
symbol.rotateAngle = 0;
symbol.charSet = "arial";
symbol.prefix = NULL;
symbol.suffix = NULL;
symbol.foreColor = RGB(255,0,0);
symbol.backColor = RGB(255,255,255);
1DBarEncode2File(&symbol,"c:\\1d.bmp");
FreeWorkSpace();
....

Link library
    1DBarcodeDll.lib
Runtime Library
    1DBarcodeDll.DLL
```

2.2.5. Example for Microsoft visual Basic

2.2.5.1. Redclaration of the data type and function

Constants declaration

```
Public Const BARCODE_CODE11 = 1
Public Const BARCODE_C25MATRIX = 2
```

```

Public Const BARCODE_C25INTER = 3
Public Const BARCODE_C25IATA = 4
Public Const BARCODE_C25LOGIC = 6
Public Const BARCODE_C25IND = 7
Public Const BARCODE_CODE39 = 8
Public Const BARCODE_EXCODE39 = 9
Public Const BARCODE_EANX = 13
Public Const BARCODE_EAN128 = 16
Public Const BARCODE_CODABAR = 18
Public Const BARCODE_CODE128 = 20
Public Const BARCODE_DPLEIT = 21
Public Const BARCODE_DPIDENT = 22
Public Const BARCODE_CODE16K = 23
Public Const BARCODE_CODE93 = 25
Public Const BARCODE_FLAT = 28
Public Const BARCODE_TELEPEN = 32
Public Const BARCODE_UPCA = 34
Public Const BARCODE_UPCE = 37
Public Const BARCODE_POSTNET = 40
Public Const BARCODE_MSI_PLESSEY = 47
Public Const BARCODE_FIM = 49
Public Const BARCODE_LOGMARS = 50
Public Const BARCODE_PHARMA = 51
Public Const BARCODE_PZN = 52
Public Const BARCODE_PHARMA_TWO = 53
Public Const BARCODE_CODE128B = 60
Public Const BARCODE_AUSPOST = 63
Public Const BARCODE_AUSREPLY = 66
Public Const BARCODE_AUSROUTE = 67
Public Const BARCODE_AUSREDIRECT = 68
Public Const BARCODE_ISBNX = 69
Public Const BARCODE_RM4SCC = 70
Public Const BARCODE_ITF14 = 72
Public Const BARCODE_NVE18 = 75
Public Const BARCODE_KOREAPOST = 77
Public Const BARCODE_PLANET = 82
Public Const BARCODE_ONECODE = 85
Public Const BARCODE_PLESSEY = 86
Public Const BARCODE_TELEPEN_NUM = 100
Public Const BARCODE_MSI_10 = 101
Public Const BARCODE_MSI_10_10 = 102
Public Const BARCODE_MSI_11 = 103
Public Const BARCODE_MSI_11_10 = 104
Public Const BARCODE_CODE39_43 = 105

```

```
Public Const BARCODE_EXCODE39_43 = 106
```

Structure declaration

```
Type symbol
    symbology As Long
    height As Long
    whiteSpaceWidth As Long
    borderWidth As Long
    outputOptions As Long
    rotateAngle As Long
    rows As Long
    width As Long
    charSize As Long
    textAlign As Long
    charSet As String * 32
    prefix As String * 16
    surfix As String * 16
    foreColor As Long
    backColor As Long
    text(100) As Byte
    encodedData(8, 1000) As Byte
    errtxt(100) As Byte
End Type
```

Function declaration

```
Public Declare Function FreeWorkspace Lib "1DBarEncodedll.dll" () As Boolean
Public Declare Function Bar1DEncode2File Lib "1DBarEncodedll.dll" (pSymbol As symbol,
    ByVal ImageFile As String) As Boolean
Public Declare Function Bar1DEncode2Bitmap Lib "1DBarEncodedll.dll" (pSymbol As symbol)
    As Long
Public Declare Sub InitWorkspace Lib "1DBarEncodedll.dll" (pSymbol As symbol)
```

2.2.5.2. Example

Example1

```
.....
Private Sub Command1_Click()
    Dim sym As symbol
    Dim a As Boolean
    Dim i, symaddr As Long
```

```

Dim s As String

symaddr = VarPtr(sym)

Call InitWorkSpace(sym)

sym.symbology = BARCODE_CODE11
sym.borderWidth = 10
sym.whiteSpaceWidth = 10
sym.rotateAngle = 45
sym.foreColor = 0

For i = 0 To 9
    sym.text(i) = 48 + i
Next

a = Bar1DEncode2File(sym, "c:\1d.bmp")
If Not a Then
    MsgBox (sym.errtxt)
End If

a = FreeWorkSpace()
End Sub
.....

```

2.3. ActiveX

2.3.1. Barcode type id constants

SYMBOLID	Value	SYMBOLID	Value
BARCODE_CODE11	1	BARCODE_C25MATRIX	2
BARCODE_C25INTER	3	BARCODE_C25IATA	4
BARCODE_C25LOGIC	6	BARCODE_C25IND	7
BARCODE_CODE39	8	BARCODE_EXCODE39	9
BARCODE_EANX	13	BARCODE_EAN128	16
BARCODE_CODABAR	18	BARCODE_CODE128	20
BARCODE_DPLEIT	21	BARCODE_DPIDENT	22
BARCODE_CODE16K	23	BARCODE_CODE93	25
BARCODE_FLAT	28	BARCODE_TELEPEN	32
BARCODE_UPCA	34	BARCODE_UPCE	37

BARCODE_POSTNET	40	BARCODE_MSI_PLESSEY	47
BARCODE_FIM	49	BARCODE_LOGMARS	50
BARCODE_PHARMA	51	BARCODE_PZN	52
BARCODE_PHARMA_TWO	53	BARCODE_CODE128B	60
BARCODE_AUSPOST	63	BARCODE_AUSREPLY	66
BARCODE_AUSROUTE	67	BARCODE_AUSREDIRECT	68
BARCODE_ISBNX	69	BARCODE_RM4SCC	70
BARCODE_ITF14	72	BARCODE_NVE18	75
BARCODE_KOREAPOST	77	BARCODE_PLANET	82
BARCODE_ONECODE	85	BARCODE_PLESSEY	86
BARCODE_TELEPEN_NUM	100	BARCODE_MSI_10	101
BARCODE_MSI_10_10	102	BARCODE_MSI_11	103
BARCODE_MSI_11_10	104	BARCODE_CODE39_43	105
BARCODE_EXCODE39_43	106		

2.3.2. textAlign values

Value	Description
DT_BOTTOM 0x00000008	Justifies the text to the bottom of the rectangle. This value is used only with the DT_SINGLELINE value.
DT_CALCRECT 0x00000400	Determines the width and height of the rectangle. If there are multiple lines of text, DrawText uses the width of the rectangle pointed to by the <i>lpRect</i> parameter and extends the base of the rectangle to bound the last line of text. If the largest word is wider than the rectangle, the width is expanded. If the text is less than the width of the rectangle, the width is reduced. If there is only one line of text, DrawText modifies the right side of the rectangle so that it bounds the last character in the line. In either case, DrawText returns the height of the formatted text but does not draw the text.
DT_CENTER 0x00000001	Centers text horizontally in the rectangle.

DT_EDITCONTROL 0x00002000	Duplicates the text-displaying characteristics of a multiline edit control. Specifically, the average character width is calculated in the same manner as for an edit control, and the function does not display a partially visible last line.
DT_END_ELLIPSIS x00008000	For displayed text, if the end of a string does not fit in the rectangle, it is truncated and ellipses are added. If a word that is not at the end of the string goes beyond the limits of the rectangle, it is truncated without ellipses. The string is not modified unless the DT_MODIFYSTRING flag is specified. Compare with DT_PATH_ELLIPSIS and DT_WORD_ELLIPSIS.
DT_EXPANDTABS 0x00000040	Expands tab characters. The default number of characters per tab is eight. The DT_WORD_ELLIPSIS, DT_PATH_ELLIPSIS, and DT_END_ELLIPSIS values cannot be used with the DT_EXPANDTABS value.
DT_EXTERNALLEADING 0x00000200	Includes the font external leading in line height. Normally, external leading is not included in the height of a line of text.
DT_HIDEPREFIX 0x00100000	Windows 2000/XP: Ignores the ampersand (&) prefix character in the text. The letter that follows will not be underlined, but other mnemonic-prefix characters are still processed. For example: input string: "A&bc&&d" normal: "A b c&d" DT_HIDEPREFIX: "A b c&d" Compare with DT_NOPREFIX and DT_PREFIXONLY.
DT_INTERNAL 0x00001000	Uses the system font to calculate text metrics.
DT_LEFT 0x00000000	Aligns text to the left.

DT_MODIFYSTRING 0x00010000	Modifies the specified string to match the displayed text. This value has no effect unless DT_END_ELLIPSIS or DT_PATH_ELLIPSIS is specified.
DT_NOCLIP 0x00000100	Draws without clipping. DrawText is somewhat faster when DT_NOCLIP is used.
DT_NOFULLWIDTHCHARBREAK 0x00080000	Windows 98/Me, Windows 2000/XP: Prevents a line break at a DBCS (double-wide character string), so that the line breaking rule is equivalent to SBCS strings. For example, this can be used in Korean windows, for more readability of icon labels. This value has no effect unless DT_WORDBREAK is specified.
DT_NOPREFIX 0x00000800	Turns off processing of prefix characters. Normally, DrawText interprets the mnemonic-prefix character & as a directive to underscore the character that follows, and the mnemonic-prefix characters && as a directive to print a single &. By specifying DT_NOPREFIX, this processing is turned off. For example, input string: "A&bc&&d" normal: "A ^u bc&d" DT_NOPREFIX: "A&bc&&d" Compare with DT_HIDEPREFIX and DT_PREFIXONLY.
DT_PATH_ELLIPSIS 0x00004000	For displayed text, replaces characters in the middle of the string with ellipses so that the result fits in the specified rectangle. If the string contains backslash (\) characters, DT_PATH_ELLIPSIS preserves as much as possible of the text after the last backslash. The string is not modified unless the DT_MODIFYSTRING flag is specified. Compare with DT_END_ELLIPSIS and DT_WORD_ELLIPSIS.
DT_PREFIXONLY 0x00200000	Windows 2000/XP: Draws only an underline at the position of the character following the ampersand (&) prefix character. Does not draw any other characters in the string. For example,

	input string: "A&bc&&d" normal: "Abc&d" DT_PREFIXONLY: " _ " Compare with DT_HIDEPREFIX and DT_NOPREFIX.
DT_RIGHT 0x00000002	Aligns text to the right.
DT_RTLREADING 0x00020000	Layout in right-to-left reading order for bi-directional text when the font selected into the <i>hdc</i> is a Hebrew or Arabic font. The default reading order for all text is left-to-right.
DT_SINGLELINE 0x00000020	Displays text on a single line only. Carriage returns and line feeds do not break the line.
DT_TABSTOP 0x00000080	Sets tab stops. Bits 15-8 (high-order byte of the low-order word) of the <i>uFormat</i> parameter specify the number of characters for each tab. The default number of characters per tab is eight. The DT_CALCRECT, DT_EXTERNALLEADING, DT_INTERNAL, DT_NOCLIP, and DT_NOPREFIX values cannot be used with the DT_TABSTOP value.
DT_TOP 0x00000000	Justifies the text to the top of the rectangle.
DT_VCENTER 0x00000004	Centers text vertically. This value is used only with the DT_SINGLELINE value.
DT_WORDBREAK 0x00000010	Breaks words. Lines are automatically broken between words if a word would extend past the edge of the rectangle specified by the <i>lpRect</i> parameter. A carriage return-line feed sequence also breaks the line. If this is not specified, output is on one line.
DT_WORD_ELLIPSIS 0x00040000	Truncates any word that does not fit in the rectangle and adds ellipses. Compare with DT_END_ELLIPSIS and DT_PATH_ELLIPSIS.

2.3.3. Properties

2.3.3.1. charSet

FontName such as 'arial'

String charSet

2.3.3.2. outputOptions

BARCODE_BOX or BARCODE_BIND

short outputOptions

2.3.3.3. charSize

if not equal 0, then print encoded text on the barcode , the position is controled align attribute

short charSize

2.3.3.4. whiteSpaceWidth

size of white space

short whiteSpaceWidth

2.3.3.5. rotateAngle

angle of rotation

short rotateAngle

2.3.3.6. Margin

The property set the margin of PDF417 barcode

short Margin

2.3.3.7. EncodeData

data to be encoded

string EncodeData

2.3.3.10. foreColor

The property set the Foreground color of barcode

OLE_COLOR ForeColor

2.3.3.11. backColor

The property set the Background color of barcode

OLE_COLOR backColor

2.3.2. Methods

2.3.2.1. Encode2ImageFile

The method Encode2ImageFile encode the data inputed and save the barcode image to file.

boolean Encode2ImageFile(BSTR lpFile);

Parameters

lpFile

[in] specify the barcode image file to be saved

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.3.3. Register activeX component

Regsvr32 Bar1DEncodeOcx.OCX

2.3.6. Example for Microsoft visual Basic

```
Private Sub Command1_Click()  
    Bar1DEncodeOcx1.CodedText = "1234567"  
  
    Bar1DEncodeOcx1.Symbolology = 1  
    Bar1DEncodeOcx1.RotateAngle = 45  
  
    Bar1DEncodeOcx1.ForeColor = &HFF00FF  
    Bar1DEncodeOcx1.EncodeMode = 1  
    Bar1DEncodeOcx1.BorderWidth = 10  
    Bar1DEncodeOcx1.ConfigType = 1  
  
End Sub
```

2.4. ASP Control for server side

2.4.1. Barcode type id constants

Refer 2.3.1.

2.4.2. textAlign values

Refer 2.3.2

2.4.3. Properties

2.4.3.1. charSet

FontName such as 'arial'

String charSet

2.4.3.2. outputOptions

BARCODE_BOX or BARCODE_BIND

short outputOptions

2.4.3.3. charSize

if not equal 0, then print encoded text on the barcode , the position is controled align attribute

short charSize

2.4.3.4. whiteSpaceWidth

size of white space

short whiteSpaceWidth

2.4.3.5. rotateAngle

angle of rotation

short rotateAngle

2.4.3.6. Margin

The property set the margin of PDF417 barcode

short Margin

2.4.3.7. EncodeData

data to be encoded

string EncodeData

2.4.3.10. foreColor

The property set the Foreground color of barcode

OLE_COLOR ForeColor

2.4.3.11. backColor

The property set the Background color of barcode

OLE_COLOR backColor

2.4.2. Methods

2.4.2.1. InitWorkspace

The method InitWorkspace initialize the working environment

BOOL InitWorkspace().

Parameters

none

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.4.2.2. FreeWorkspace

The method FreeWorkspace destroy the working environment

BOOL FreeWorkspace().

Parameters

none

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.3.2.3. Encode2File

The method Encode2File encode the data inputed and save the barcode image to file.

boolean Encode2File(BSTR strImageFile);

Parameters

strImageFile

[in] specify the barcode image file to be saved

Return values

If the function succeeds, the return TRUE, otherwise , return FALSE.

2.4.3. Register the ASP server component

Regsvr32 Bar1DEncodeASP.DLL

2.4.4. Example for ASP

```
<%  
    set symbol = Server.CreateObject("AztecEncodeCOM.EncodeService")  
  
    symbol.InitWorkspace()  
  
    symbol.Symbology = 2;  
  
    symbol.BorderWidth=10;  
  
    symbol.OutputOptions;  
  
    symbol.RotateAngle = 0;  
  
    symbol.CharSet = "arial";  
  
    obj.CodedText= "Http://www.aipsys.com"  
  
    obj.Encode2File("C:\1.gif")  
  
    obj.FreeWorkspace()  
  
    response.Write("<img src='1.gif'>")  
  
    response.Write("<br>")  
  
    response.Write("Trial Version randomly change ")  
  
%>
```

3. Decoder SDK

3.1. Static link library

3.1.1. Result structure for decoding

```
typedef struct tagResult
{
    char sBarcodeType[32];    //barcode name
    BYTE *pData;             //barcode data decoded
    int nSize;               //barcode data size
    int xPos;                //barcode topleft position on image
    int yPos;                //barcode topleft position on image
}Result;
```

3.1.2. Function or procedure

3.1.2.1. _Barcode1DDecodeImageFile

The _Bar1DDecodeImageFile function read 1D barcode figure from image and decode it to text or binary data.

```
Result __stdcall _DataMatrixDecodeImageFile(char *lpImageFile, int *pCount);
```

Parameters

lpImageFile LPCTSTR

[in] the image file containing 1D barcode figure, it can be BMP,GIF,PNG,JPG or TIF formats.

pCount int

[out] the count of the barcode found. The pointer need be allocate memory before call function.

Return values

Result when decode success, the decoded data written in Result buffer, you can refer the pCount to get result one by one

NULL decode failure.

3.1.2.2. **_Barcode1DDecodeBitmap**

The `_Barcode1DDecodeBitmap` function read 1D barcode figure from image and decode it to text or binary data.

```
Result __stdcall _Barcode1DDecodeBitmap (HBITMAP hImage, int *pCount);
```

Parameters

`hImage` (HBITMAP

[in] the bitmap handle of image containing 1D barcode figure,.

`pCount` int

[out] the count of the barcode found. The pointer need be allocate memory before call function.

Return values

Result when decode success, the decoded data written in Result buffer, you can refer the `pCount` to get result one by one

NULL decode failure.

3.1.2.3. **_Barcode1DFree**

The `_Barcode1DFree` function free the returned result buffer after you get data from it.

```
void __stdcall _Barcode1DFree(Result *pResult, int nCount);
```

Parameters

`pResult` Result

[in] the result buffer after decoding,.

`nCount` int

[in] the barcodes decoded

Return values

void

3.1.2.4. **_LoadImageEx**

The `_LoadImageEx` function load image such as JPG,GIF,BMP,PNG file and return image handle of the image.

HBITMAP __stdcall _LoadImageEx(char *lpImageFile);

Parameters

`lpImageFile` **char***

[in] the file name including path.

Return values

HBITMAP the image handle if success

NULL if failed

3.1.3. **Examples**

```
#include "stdafx.h"
```

```
#include "../barcode1ddeco.lib.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    Result *p = NULL;
```

```
    int nSize;
```

```
    p = _Barcode1DDDecodeImageFile("G: \\code39-1\\4.gif",&nSize);
```

```
    if (p)
```

```
    {
```

```
        char buf[8192];
```

```
        memset(buf,0,8192);
```

```
        memcpy(buf,p->pData,p->nSize);
```

```

        MessageBox(buf);

        _Barcode1DFree(p,nSize);

    }

    return 0;

}

```

Linked library

Bar1dDecodeLib.lib

3.2. Dynamic link library

3.2.1. Result structure for decoding

```

typedef struct tagResult
{
    char sBarcodeType[32];    //barcode name
    BYTE *pData;             //barcode data decoded
    int nSize;               //barcode data size
    int xPos;               //barcode topleft position on image
    int yPos;               //barcode topleft position on image
}Result;

```

3.2.2. Function or procedure

3.2.2.1. Barcode1DDecodeImageFile

The Bar1DDecodeImageFile function read 1D barcode figure from image and decode it to text or binary data.

```
Result __stdcall Bar1DDecodeImageFile(char *lpImageFile, int *pCount);
```

Parameters

lpImageFile LPCTSTR

[in] the image file containing 1D barcode figure, it can be BMP,GIF,PNG,JPG or TIF formats.

pCount int

[out] the count of the barcode found. The pointer need be allocate memory before call function.

Return values

Result when decode success, the decoded data written in Result buffer, you can refer the pCount to get result one by one

NULL decode failure.

3.2.2.2. Barcode1DDecodeBitmap

The Barcode1DDecodeBitmap function read 1D barcode figure from image and decode it to text or binary data.

Result __stdcall _Barcode1DDecodeBitmap (HBITMAP hImage, int *pCount);

Parameters

hImage HBITMAP

[in] the bitmap handle of image containing 1D barcode figure,.

pCount int

[out] the count of the barcode found. The pointer need be allocate memory before call function.

Return values

Result when decode success, the decoded data written in Result buffer, you can refer the pCount to get result one by one

NULL decode failure.

3.1.2.3. _Barcode1DFree

The Barcode1DFree function free the returned result buffer after you get data from it.

void __stdcall Barcode1DFree(Result *pResult, int nCount);

Parameters

pResult Result

[in] the result buffer after decoding,.

nCount int

[in] the barcodes decoded

Return values

void

3.1.2.4. LoadImageEx

The LoadImageEx function load image such as JPG,GIF,BMP,PNG file and return image handle of the image.

HBITMAP __stdcall LoadImageEx(char *lpImageFile);

Parameters

lpImageFile **char***

[in] the file name including path.

Return values

HBITMAP the image handle if success

NULL if failed

3.1.3. Examples

```
#include "stdafx.h"
```

```
#include "../barcode1dddecodedll.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    Result *p = NULL;
```

```
    int nSize;
```

```
    p =Barcode1DDecodeImageFile("G: \\code39-1\\4.gif",&nSize);
```

```
    if (p)
```

```
    {
```

```

        char buf[8192];

        memset(buf,0,8192);

        memcpy(buf,p->pData,p->nSize);

        MessageBox(buf);

        _Barcode1DFree(p,nSize);

    }

    return 0;

}

```

Linked library

Bar1dDecodeDll.lib

Runtime Library


























































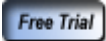

























Bar1dDecodeDll.dll

4. Order Information

Our sales department was outsourced to Shareit Inc. Ordering online is secure, safe, and guaranteed. We provide first-rate, global service to you.

We accept Visa, MasterCard, American Express, JCB, Diners Club, Switch and Solo. Credit card payments are processed within seconds, and clients receive their product or licensing information without delay.

- The *Developer License* allows one developer royalty-free distribution up to 10,000 user licenses.
- The *5 Developer License* grants the rights of the Developer License for up to five (5) developers and 20,000 user licenses.
- The *Unlimited Developer License* grants the rights of the Developer License for an unlimited number of developers and an unlimited number of user licenses.
- The *Small Company Developer License* grants the rights of the Developer License to organizations which a gross annual revenue or funding of less than 2 million U.S. Dollars.
- The *Single User License* allows use of the Software for one (1) user in your organization

Packages	Trial Dwonload	Single User	Small Company Developer	1 Developer	5 Developer	Unlimited Developer	Version
1D Barcode Encode SDK							
Static Library			 \$495	 \$990	 \$2,180	 \$3,099	1.2
Dynamic Library		 \$125	 \$179	 \$379	 \$1,090	 \$2,199	1.2
ActiveX		 \$125	 \$179	 \$379	 \$1,090	 \$2,199	1.2
ASP Component			 \$179	 \$379	 \$1,090	 \$2,199	1.2
QRCode Encode SDK							
Static Library			 \$495	 \$990	 \$2,180	 \$3,099	1.2
Dynamic Library		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ActiveX		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ASP Component			 \$179	 \$379	 \$949	 \$2,199	1.2
DataMatrix Encode SDK							
Static Library			 \$495	 \$990	 \$2,180	 \$3,099	1.2
Dynamic Library		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ActiveX		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ASP Component			 \$179	 \$379	 \$949	 \$2,199	1.2
PDF417 Encode SDK							
Static Library			 \$495	 \$990	 \$2,180	 \$3,099	1.2
Dynamic Library		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ActiveX		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ASP Component			 \$179	 \$379	 \$949	 \$2,199	1.2
Aztec Encode SDK							

Static Library			 \$495	 \$990	 \$2,180	 \$3,099	1.2
Dynamic Library		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ActiveX		 \$125	 \$179	 \$379	 \$949	 \$2,199	1.2
ASP Component			 \$179	 \$379	 \$949	 \$2,199	1.2

5. Affiliate program

You will receive a 10%~40% commission on all orders placed by referrals from your website. More sales higher commissions!

AIPSYS affiliate program allows publishers, resellers, and web site owners to advertise AIPSYS.com products to their users and visitors, and earn 30%. Signup is simple and free, and you can spread the word about these fine products and earn commissions in the process.

What You Can Earn

AIPSYS products range in price from \$49.95 to \$4000. Many users buy multiple products at a time, and since many users also order additional voices at the time of purchase, the average order size is over \$100 per order. You will earn 30% on each order you enable through your advertising of AIPSYS.com Products. Our experience suggests that focused target audiences lead to higher sales, with many customers buying up front, and 1% to 3% of downloaders will purchase.

Step by step instruction

Becoming a AIPSYS.com Affiliate is quick and easy. Our affiliate program is managed by Shareit, the established leader in software affiliate programs.

✧To Get Started, simply click  Sign Up as a Shareit affiliate.

✧Complete the form, read the agreement and FAQ;

✧Share*it will check your entry, send us confirmation of your application and, subject to final review, we will activate your affiliate account

✧We'll send you an e-mail containing your affiliate ID – the ID is used on your site to inform Share*it that the order is coming from your affiliate account, e.g.<http://www.shareit.com/product.html?cart=1&productid=YYYYYY&languageid=1&affiliateid=X> XXXX

(XXXXXX should be changed to your ID,YYYYYY should be product id you affiliate,you can select from the table below);

✧You can combine this link on your site with the logo from our site

✧Please note that you or your customers can choose from 1, 2 or even 3 years support contracts, and up

to 99 licenses can be purchased online (prices are available after clicking on "Display volume discount prices" button).

When the affiliate relationship is established, you will get an affiliate link. Any sales originating from your link will be credited to you, and you will receive the 10~40% commission. Please note currently we will grant 20% commission rate for new affiliates at ShareIt in order to avoid fraud and chargeback. But we are more than happy to increase your commission rate. Please contact us via sales@aipsys.com directly.

PRODUCT ID TABLE						
Packages	Single User	Small Company Developer	1 Developer	5 Developer	Unlimited Developer	Version
1D Barcode Encode SDK						
Static Library		300222295	300222296	300222297	300222298	1.2
Dynamic Library	300222332	300222333	300222334	300222335	300222336	1.2
ActiveX	300222367	300222368	300222369	300222370	300222371	1.2
ASP Component		300222388	300222389	300222390	300222391	1.2
QRCode Encode SDK						
Static Library		300222413	300222284	300222285	300222286	1.2
Dynamic Library	300222309	300222312	300222314	300222317	300222320	1.2
ActiveX	300222343	300222344	300222347	300222350	300222355	1.2
ASP Component		300222376	300222377	300222377	300222379	1.2
DataMatrix Encode SDK						
Static Library		300222291	300222292	300222293	300222294	1.2
Dynamic Library	300222321	300222322	300222324	300222325	300222326	1.2
ActiveX	300222357	300222358	300222359	300222360	300222361	1.2
ASP Component		300222380	300222381	300222382	300222383	1.2
PDF417 Encode SDK						
Static Library		300222280	300222281	300222282	300222283	1.2
Dynamic Library	300222299	300222300	300222301	300222303	300222305	1.2
ActiveX	300222337	300222338	300222339	300222340	300222341	1.2
ASP Component		300222372	300222373	300222374	300222375	1.2
Aztec Encode SDK						
Static Library		300222288	300222414	300222289	300222290	1.2
Dynamic Library	300222323	300222327	300222329	300222330	300222331	1.2
ActiveX	300222362	300222363	300222364	300222365	300222366	1.2
ASP Component		300222384	300222385	300222386	300222387	1.2

6. Support Information

Sales information

If you purchase online please check the Current versions list to ensure you have the latest version. The latest versions are those available on the downloads menu above.

Before you buy, you can [download it for evaluation](#), To buy it please refer [price list](#) and place an order, price in other currency shown in order form.

Having other question or requirement about sale, please contact [sales service](#).

Having some technical question or new requirement, please contact our [technical support service](#).

Barcode resources reference information

Introduction to [common barcode types](#)

[RSS barcodes renamed GS1-DataBar](#)

[Recommended sizes for barcodes](#)

Barcode specifications & Standards

- [American National Standards Institute](#)
- [Automatic Identification Manufacturer's Association](#)
- [Automotive Industry Action Group](#)
- [British Standards Institution](#) (BSI)
- [GS1](#) (formerly EAN International)
- [GS1 UK](#) (formerly the e-Centre)
- [GS1 US](#) (formerly UCC - Uniform Code Council)
- [Health Industry Barcode Standards](#)
- [ISO - International Standards Organisation](#)

7. Product Information Link

. [QRCode encoder SDK](#)

. [PDF417 encoder SDK](#)

. [DataMatrix encoder SDK](#)

- . [Aztec encoder SDK](#)
- . [Linear 1D barcode encoder SDK](#)