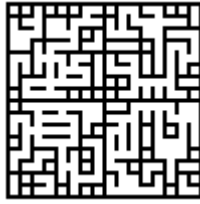


Welcome to HD Barcode



Congratulations on obtaining the HD Barcode SDK (Software Development Kit). This SDK will enable you to explore this very exciting new technology.

System requirements:

- Windows-based development computer (Windows XP or later)
- Android smartphone or tablet (iOS support coming soon)

The SDK consists of two parts:

- PrinterSDK
- ReaderSDK

The PrinterSDK is used to create HD Barcodes. In a production HD Barcode system, the creation of HD Barcodes is done in one of two ways. One method incorporates a hardware device called a code generator. The other method is a purely software-based solution. The PrinterSDK implements a software-only solution. Partial source code is provided making it possible to integrate the printing of HD Barcodes into an existing or new process.

The ReaderSDK is used to create an Android app for reading HD Barcodes. Partial source code is provided making it possible to tailor the reading app for your specific needs.

Trial Period

The PrinterSDK enforces an expiration date. If you need more time to evaluate this technology, you will need to contact us in order to purchase an updated authorization key.

The ReaderSDK does not impose an expiration date on public HD Barcodes, but Company Private HD Barcodes can only be read until the expiration date used by the PrinterSDK.

Why are HD Barcodes better?

QR Codes are immensely popular, but HD Barcodes are better because:

They can hold far more data and be read more reliably than QR codes

Codes may be rectangularly-shaped and not just perfect squares

Codes may be constructed from lines or from dots

Codes may be superimposed on pictures (embossing)

You can easily restrict who is authorized to read the codes

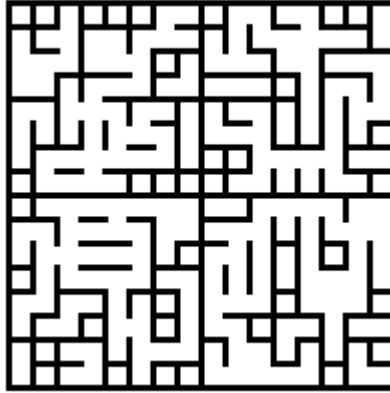
With an HD Barcode, you can embed a complete JPEG photograph (someone's face for example) and quickly and reliably read it with a cell phone. With QR Code, you must either link to an external database, or put so much data in the code that it is next to impossible to read.

When using a flat-bed scanner, HD Barcodes can be read containing an amazing 700,000 bytes of information. This is 175 times more data than the most dense QR Code.

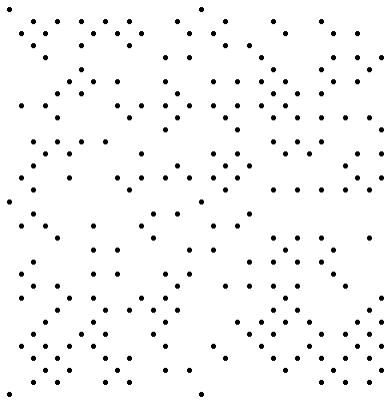
QR Codes are completely in the public domain, which means anyone can read them, and anyone can produce them. HD Barcodes provide features that make it very easy to control which codes are allowed to be read and by whom. You can have public codes which anyone can read, or you can have private codes which require authorization (via yet another HD Barcode) in order to be read.

HD Barcode Styles

HD Barcodes can be constructed from lines:



or from dots:



or dots superimposed in an image (we call this embossing):



HD Barcode Data

HD Barcodes embed data that consists of two parts:

- a text part
- a binary part

An HD Barcode may contain either one or both of these types of parts.

Because not using compression can result in an HD Barcode which is much larger than necessary, the binary part must use one of the following compressed formats:

JPG	identified by first two bytes of FF and D8
PNG	identified by first two bytes of 89 and 50
ZIP	identified by first two bytes of 50 and 4B

To use a format other than one of these, create a ZIP file containing the desired file and embed that ZIP file in the HD Barcode.

Note that even though HD Barcodes support PNG files, the included CreateHDBarcode program does not support PNG files. It only supports JPG and ZIP files.

Public and Private HD Barcodes

HD Barcodes come in two distinct “flavors”:

A **Public HD Barcode** is intended to be readable by any HD Barcode reader app or device.

A **Company Private HD Barcode** is encrypted and can only be read by a reader which was specifically authorized to read codes for that company. A dot is shown in one corner as a reminder that the code is encrypted, but the existence or absence of the dot has no bearing on the ability to read the code.

(Note: Other companies using this SDK will not be able to read Company Private HD Barcodes that you produce. You will not be able to read Company Private HD Barcodes produced by other companies.)

Authorization and Encryption keys

The following pieces of information are involved in controlling who can produce and who can read HD Barcodes.

Company Name

This is the name of your company. This field is assigned by us (HD Barcode, LLC). This field is required for producing and for reading both Public and Company Private HD Barcodes.

Authorization Key

This key works in concert with the Company Name in order to authorize the creation and reading of HD Barcodes. This authorization is necessary for both public and Company Private HD Barcodes. The authorization key is an 18-digit number assigned by us and is in the following format: ~~#### #### #### #### #### ####~~. The authorization key also embeds an expiration date. If a key expires, you must purchase another key from HD Barcode, LLC.

Identification Tag

This is a phrase which is assigned by you, and allows a single company to have multiple groups of HD Barcodes and to restrict access by group. When a reader pushes the [HDSecurLink] button, it lists all of the Company Name / Identification Tags that have been authorized for that reader. Identifications Tags are paired with Secret Phrases, but while the Identification Tag is visible, the Secret Phrase is hidden.

Secret Phrase

This is a phrase which along with the Company Name is used to encrypt a Company Private HD Barcode. Using a different Secret Phrase will generate a totally different HD Barcode. The Secret Phrase is assigned by you. Without knowing the Secret Phrase, a reader is unable to decrypt and read an encrypted Company Private HD Barcode. Even we here at HD Barcode LLC. cannot read an encrypted code without knowing both the Company Name and the Secret Phrase. You may have multiple Secret Phrases, which means you could configure some readers to read specific subsets, and other readers to read all subsets. Note that you can't use a Secret Phrase of the form ~~#####~~ as that has special meaning.

Reader Serial Number

When the reader application is installed on a device, it is assigned a unique serial number. For applications where only authorized readers can read a specific type of HD Barcode, the reader serial number must be communicated in order to generate a Unique Authorization HD Barcode for that specific reader.

Sharing Company Names and Secret Phrases

The HD Barcode itself is a perfect vehicle for sharing authorization information with devices. In order to authorize a reader to be able to read certain groups of codes, it is necessary to communicate the Company Name, the Identification Tag, and the Secret Phrase. This can be done using one of three different types of authorization HD Barcodes.

A ***Universal Authorization HD Barcode*** is a Public HD Barcode, which means any reader is capable of reading this code. It contains the following four pieces of information: the Company Name, the Identification Tag, the Secret Phrase, and an expiration date. Once a reader sees this code, it is able to read matching Company Private HD Barcodes with that Secret Phrase until the expiration date.

A ***Unique Authorization HD Barcode*** is a special Company Private HD Barcode encrypted using the reader's unique serial number. Each reader will need its own unique HD Barcode in order to be authorized. Like a Universal Authorization HD Barcode, it contains the following four pieces of information: the Company Name, the Identification Tag, the Secret Phrase, and an expiration date.

A ***Cascaded Authorization HD Barcode*** is encrypted using a Company Name and one Secret Phrase. Embedded in the code is the Company Name, the Identification Tag, a different Secret Phrase, and an expiration date. The reader must already be authorized to know the first Secret Phrase in order to learn the second Secret Phrase. Please refer to the section titled Scavenger Hunt for an example of how Cascaded Authorization HD Barcodes can be used.

The SDK gives you the ability to create any of these three types of authorization codes.

PrinterSDK

The PrinterSDK includes two folders.

The Graphical Interface folder contains a Windows installation .zip file. When run, it installs a graphical Windows program which can be used to generate and display HD Barcodes. This program can also generate the authentication HD Barcodes.

The Sample Code folder contains a basic C++ program written for Microsoft Visual Studio. This program demonstrates how to create HD Barcodes programmatically.

Both of these program interface with the file CreateHDBarcode.dll. This is a 32-bit Windows Dynamic Link Library (DLL). It is directly callable by other 32-bit Windows applications.

Note: A 64-bit Windows application must use conventional interfacing techniques (i.e. launching a secondary 32-bit process) in order to interface with this 32-bit DLL.

CreateHDBarcode

The CreateHDBarcode program is a graphical user interface for creating HD Barcodes.

To install the program:

- Navigate to the CreateHDBarcode.zip folder

- Extract all files

- Run Vcredist_x86.exe to install the appropriate C runtime library

- Run Setup.exe to install the CreateHDBarcode program

When you first run the CreateHDBarcode program, you must enter your license information. This consists of a Company Name and an Authorization Key. Once entered, you will be able to reach the other tabs on the screen.

Sample C++ Program

The TestCreateHDBarcode program is a simple C++ program which calls the DLL in order to create an HD Barcode.

Note that CreateHDBarcode.dll does not allow reverse engineering. If you call the DLL from a debugged process, the authorization will always fail and you will be unable to generate HD Barcodes.

CreateHDBarcode.dll

Authorization and creation of HD Barcodes is accomplished via a single function call to the 32-bit CreateHDBarcode.dll:

```
extern "C"
{
    int _declspec(dllexport) _stdcall CreateHDBarcode
    (
        char    *CompanyName,
        char    *AuthorizationKey,
        char    *SecretPhrase,
        char    *TextData,
        BYTE    *EmbeddedData,
        int     EmbeddedDataBytes,
        int     *BlockRows,
        int     *BlockCols,
        int     *ErrorCorrection,
        int     GridSize,
        int     Thickness,
        BYTE    *Buffer
    );
}
```

The datatype “int” is a 32-bit value. The datatype “BYTE” is an unsigned char (8-bit) value.

For all string (char *) parameters, a null pointer (0) or a pointer to a null string are treated as the same.

CompanyName is the name of your company. This field must be an exact match of the information provided to you by HD Barcode, LLC.

AuthorizationKey is the authorization code which must match CompanyName. This code must be an 18-digit number in the form: ####-####-####-####-####-####.

SecretPhrase is a pass phrase which is required in order to read a Company Private HD Barcode. Using a different Secret Phrase will generate a totally different HD Barcode. If this field is blank, then a Public HD Barcode is generated. If the field is not blank, then a Company Private HD Barcode is generated. If the Secret Phrase is of the form #####-#####-#####, then this is a request to generate a Unique Authorization HD Barcode.

TextData is a null-terminated string consisting of the text to embed in the HD Barcode. When generating authorization HD Barcodes, TextData must consist of the Identification Tag, a CR, a LF, and the Secret Phrase.

EmbeddedData is one of the following 3 choices:

- Null to indicate no embedded data
- A pointer to a JPG file data (which begins with hex FF and hex D8)
- A pointer to a ZIP file data (which begins with hex 50 and hex 4B)

EmbeddedDataBytes is the size of the binary blob used for embedded data. If embedded data is not being used, then this value should be 0. The values 1 and 2 are reserved for creating Authorization HD Barcodes.

BlockRows is the number of rows of blocks in the HD barcode. This must be an even number between 2 and 254, or 1 for a “strip code” (allowed if BlockCols is 4, 8, 12, or 16), or 0 in which case the software will automatically choose a value.

BlockCols is the number of columns of blocks in the HD barcode. This must be an even number between 2 and 254, or 1 for a “strip code” (allowed if BlockRows is 4, 8, 12, or 16), or 0 in which case the software will automatically choose a value.

ErrorCorrection is the number of bytes per group of 48 bytes to reserve for group error correction. This must be an even number between 4 and 34. It may also be set to 0 in which case the software will automatically choose a value.

GridSize is the number of pixels between lines or dots. The absolute minimum distance is 2 pixels. The value of 0 is not allowed because there is no default value.

Thickness is the number of pixels defining a line thickness (positive values between 1 and 8), or a relative size of a dot (negative values between -1 and -8). The value of 0 is not allowed because there is no default value.

Buffer is where to save the bitmap data. If null, then no information is written to the buffer. If not null, then it is assumed to be large enough to contain all of the necessary data.

The DLL does NOT allocate memory. It is the responsibility of the calling program to allocate memory, make sure enough memory has been allocated, and to release the memory when it is no longer needed.

A successful call to CreateHDBarcode returns the number of bytes of memory needed. This number is equal to SizeX (the number of pixels across) times SizeY (the number of pixels up and down). The values SizeX and SizeY are not directly returned from the function, but can be easily computed using the following formulas:

```
SizeX = GridSize + (GridSize * BlockCols * 8) + abs(Thickness) + GridSize;  
SizeY = GridSize + (GridSize * BlockRows * 8) + abs(Thickness) + GridSize;
```

Error Return Codes

If a call to CreateHDBarcode is unsuccessful, it will return one of the following negative numbers which are also listed in CreateHDBarcode.h:

- 1 Parameter 1 CompanyName error
 not allowed to be blank or to contain special characters
- 2 Parameter 2 AuthorizationCode error
 not of the form ####-####-####-####-####-####, or not authorized
- 3 Parameter 3 SecretPhrase error
 contains special characters, or blank for Authorization HD Barcode
- 4 Parameter 4 TextData error
 not of form #####-#####-##### (for Authorization HD Barcodes)
- 5 Parameter 5 EmbeddedData error
 first two bytes were not FF D8 (JPG). 89 50 (PNG), or 50 4B (ZIP)
- 6 Parameter 6 EmbeddedDataBytes error
 did not match EmbeddedData
- 7 Parameter 7 BlockRows error
 was neither 0, 1, nor an even number between 2 and 254
 if BlockCols is 1, then BlockRows must be 4, 8, 12, or 16
- 8 Parameter 8 BlockCols error
 was neither 0, 1, nor an even number between 2 and 254
 if BlockRows is 1, then BlockCols must be 4, 8, 12, or 16
- 9 Parameter 9 ErrorCorrection error
 was neither 0 nor an even number between 4 and 34
- 10 Parameter 10 GridSize error
 was not a number between 2 and 128
- 11 Parameter 11 Thickness error
 was not 1 to 8 (for lines), or -1 to -8 (for dots)
- 91 Insufficient memory error
 the computer reported insufficient memory for processing
- 92 Too much data error
 the data exceeds the specified symbol size

Generating Authorization HD Barcodes

When a reader successfully scans an Authorization HD Barcode, it learns a new Company Name + Secret Phrase giving it the ability to decipher HD Barcodes that it could not read without this knowledge. When creating an Authorization HD Barcode, there are several parameters that must be set properly.

Text Data must be set to the Identification Tag. This tag is shown on the reader app when the [HDSecurLink] button is pushed to list current authorizations.

To create a Universal Authorization HD Barcode, Secret Phrase must be blank. This means that any reader will be able to read the generated Authorization HD Barcode.

To create a Unique Authorization HD Barcode, Secret Phrase must be set to the reader application's Serial Number which is of the form #####-#####-####. This number can be obtained by pushing the [HDSecurLink] button while running a reader app on the device which is to receive the authorization.

To create a Cascaded Authorization HD Barcode, Secret Phrase must be set to the phrase that the reader is expected to already know. This is different from the Secret Phrase included in EmbeddedData which is the new phrase that the reader will learn.

Set EmbeddedDataBytes to 1 to generate a permanent authorization HD Barcode. Permanent means that the reader app will retain the information until the app is uninstalled or its cache memory cleared.

Set EmbeddedDataBytes to 2 to generate a temporary authorization HD Barcode. Temporary means that the reader app will not retain the information, so this authorization HD Barcode must be rescanned prior to reading corresponding Company Private HD Barcodes.

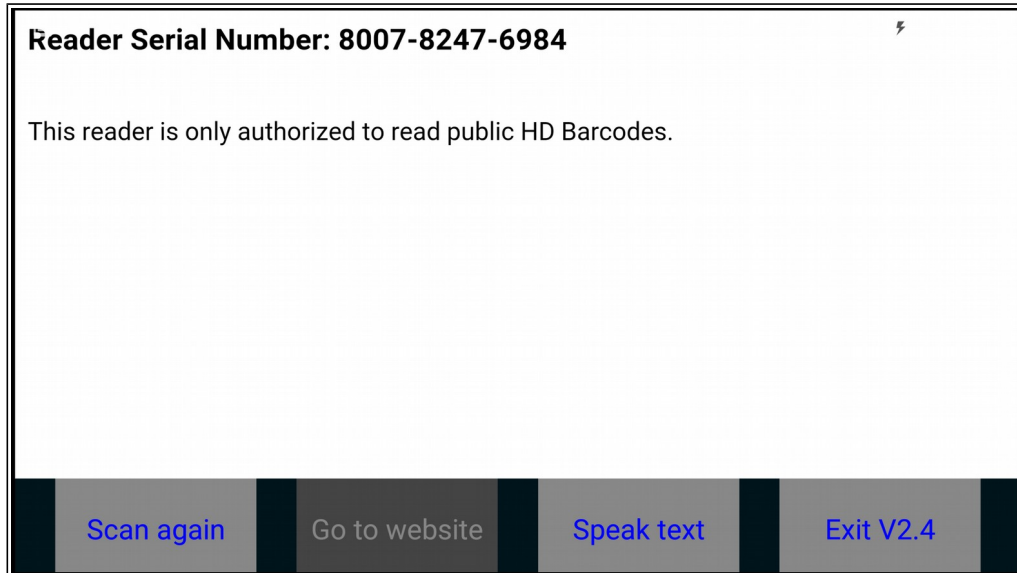
EmbeddedData must point to the new Secret Phrase which corresponds to the Identification Tag. Note that EmbeddedData is declared to be (BYTE *) instead of (char *) because it usually points to binary data, not a character string.

All Authorization HD Barcodes contain the expiration date embedded in the Authorization Key. They must be reauthorized with a newer authorization HD Barcode to work beyond the expiration date.

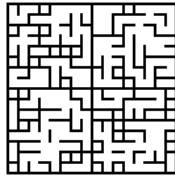
Scavenger Hunt

This example demonstrates how HD Barcodes can be used to implement a scavenger hunt. The idea is that readers must eventually find and scan all of the barcodes in a specific order. Some barcodes are encrypted so they cannot be read until all previous barcodes have been found and read.

Before starting the scavenger hunt, if the [HDSecurLink] button is pushed from the reader app, the following appears on the screen:



Here is the HD Barcode at the first station:



This is a Universal Authorization HD Barcode, which means the barcode is NOT encrypted and that any reader can read the code. It embeds the following information:

Company Name: Scavenger Hunt, Inc.
Identification Tag: 1st Station
Secret Phrase: Blue and gold

After reading this code, the reader app shows the following screen:

Reader Serial Number: 8007-8247-6984

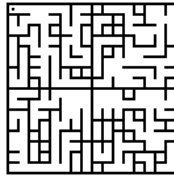
This reader is authorized to read public HD Barcodes, and also the following Company Private HD Barcodes:

Company Name	Identification Tag	Expires
Scavenger Hunt, Inc.	1st Station	05/2016

[Scan again](#)[Go to website](#)[Speak text](#)[Exit V2.4](#)

The [HDSecurLink] button on the reader can be pressed to redisplay this information. Note that the Identification Tag of 1st Station is shown, but the Secret Phrase of “Blue and gold” is not shown.

Here is the HD Barcode at the second station:



This is a Cascaded Authorization HD Barcode. Cascaded means it is an Authorization HD Barcode that is also encrypted and therefore not a public code. In order to be able to read the code, the reader must know the Company Name and the Secret Phrase used to encrypt the barcode. This HD Barcode is encrypted using the following information:

Company Name: Scavenger Hunt, Inc.
Secret Phrase: Blue and gold

Do not confuse the information used to encrypt the HD Barcode with the information actually embedded in the HD Barcode. This HD Barcode embeds the following:

Company Name: Scavenger Hunt, Inc.
Identification Tag: 2nd Station
Secret Phrase: Green and white

Until a reader learns the Secret Phrase Blue and Gold from the HD Barcode at the first station, it is not capable of reading the HD Barcode at the second station, and therefore can't learn the next Secret Phrase which is Green and white.

A reader which has seen the HD Barcode at station 1 is able to read this barcode. Once read, the following screen appears:

Reader Serial Number: 8007-8247-6984

This reader is authorized to read public HD Barcodes, and also the following Company Private HD Barcodes:

Company Name	Identification Tag	Expires
Scavenger Hunt, Inc.	1st Station	05/2016
Scavenger Hunt, Inc.	2nd Station	05/2016

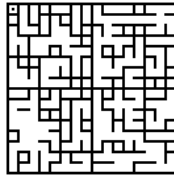
Scan again

Go to website

Speak text

Exit V2.4

Here is the HD Barcode at the third station:

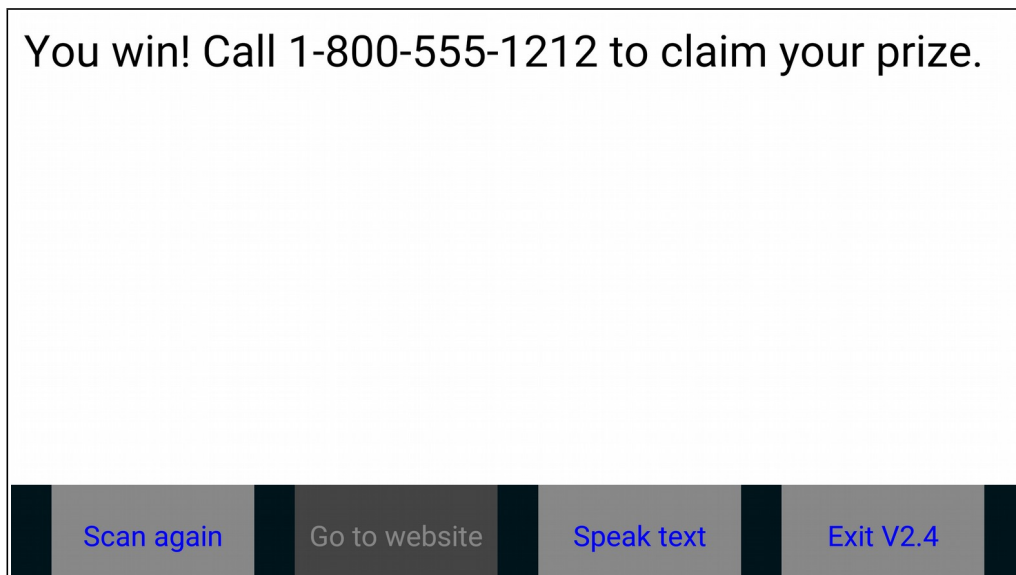


This a Company Private HD Barcode. It is encrypted using the following information:

Company Name: Scavenger Hunt, Inc.
Secret Phrase: Green and white

Unlike the other two station HD Barcodes, this is not an Authorization HD Barcode. Reading it produces text (or it could produce a picture).

If your reader has not learned the Secret Phrase “Green and white”, it is not capable of reading the code. A reader which has read all of the prior HD Barcodes gets the following screen:



ReaderSDK

The ReaderSDK uses Android Studio from Google:

<http://developer.android.com/sdk/index.html>

In the ReaderSDK folder, you'll find:

HD Barcode Reader Vx.x.apk	a ready-to-install-and-run HD Barcode Reader
HDBarcodeReaderSDK	a folder containing the source code

Abbreviations Using Tilde (~)

The tilde (~) character may be for “abbreviations”, which are references to built-in images. The advantage of this technique is that large complex images may be preloaded onto the device, and that way the images do not need to be embedded in the HD Barcode itself. This results in an HD Barcode that is much smaller and easier to read.

This technique is only useful in those applications where there is a relatively small number of fixed images to be used, and these images are known in advance.

The preloaded images may be located in any combination of the following two locations:
the assets folder in the src directory of the reader application
a folder named “HD Barcode assets” on the external storage drive (/sdcard)

For example, if the embedded text in an HD Barcode is “face~Display this text”, then the reader program will look for a built-in asset or for an external store file which begins with “face~”. If a matching file is found, then the HD Barcode is treated as if the text portion were “Display this text”, but also included a binary image which matches the file that was already loaded onto the device.

As an example, the reader application includes the single asset face~example.jpg, which is a smiley face.