



G3 Tablet SDK Description

iDTRONIC GmbH
Donnersbergweg 1
67059 Ludwigshafen
Germany/Deutschland

Phone: +49 621 6690094-0
Fax: +49 621 6690094-9
E-Mail: info@idtronic.de
Web: idtronic.de

Issue 1.0
– 08. July 2020 –

Subject to alteration without prior notice.
© Copyright iDTRONIC GmbH 2018
Printed in Germany

Table of Contents

1	C/C++.....	5
1.1	Scanner	6
1.1.1	Scanner Functions.....	8
1.1.2	G2W_InitScanner.....	10
1.1.3	G2W_CloseScanner.....	11
1.1.4	G2W_Read	12
1.1.5	G2W_ReadCancel	14
1.1.6	G2W_RegisterWindow	15
1.1.7	G2W_GetScanDataByte	16
1.1.8	G2W_SetSymValue	17
1.1.9	G2W_GetSymValue	18
1.1.10	G2W_SetOptValue.....	19
1.1.11	G2W_GetOptValue	20
1.1.12	G2W_GetVersionInfo.....	21
1.2	Scanner Enumerations.....	22
1.2.1	Symbologies	23
1.2.2	Symbology option	27
1.2.3	module option	28
1.3	Scanner 1D Errors.....	30
2	C#.....	31
2.1	G2W_HScan_Ctrl_Net Assembly.....	32
2.2	G2W_HScan_Ctrl_Net Namespace	33
2.3	Classes.....	35
2.3.1	HScanCtrlNet Class.....	36
2.3.2	G2W_ReadCancel_net	43
2.3.3	G2WScannerDataArgs Class.....	50
2.3.4	ScanMessageWindow Class	54
2.4	Enumerations	58
2.4.1	Symbologies	59
2.4.2	Symbology option	63
2.4.3	module option	64
2.5	Delegates	66
2.5.1	G2WScannerDataDelegate	67

SDKs for IDTRONIC Devices

IDTRONIC Software Development Kit for the Microsoft® C,C++ & C#

[Revision v1.0]

Documentation Overview

This SDK documentation includes the detail description all about IDTRONIC Software Development Kit (SDK) which controls each devices such as scanner. The SDK is divided into two main sections. The one is for C/C++ and the other is for C#.

Development environment This SDK is developed and tested under VisualStudio 2015.

[SDKs for IDTRONIC Devices](#) >

1 C/C++

Overview The SDK for C/C++ provides all of the tools necessary to develop C/C++ application for IDTRONIC Devices. This SDK includes libraries for each features.

In This Section Scanner

[SDKs for IDTRONIC Devices](#) > [C/C++](#) >

1.1 Scanner

Overview This section contains the description of functions, structures and references on the scanner which reads one-dimensional barcodes.

Supported Barcode Type

CODABAR
CODE39
INT25
NEC25
CODE93
STRAIGHT25_IND
STRAIGHT25
MATRIX25
CODE11
CODE128
ISBT
GS1_128
UPCA_OP
UPCA
UPCE
EAN13
EAN8
MSI
GS1_DATABAR_OMNI
GS1_DATABAR_LIMIT
GS1_DATABAR_EXPAND
TRIOPTIC_CODE
CODABLOCK_A
CODABLOCK_F
PDF417
MICRO417
GS1_COMPOSITE_CODE
GS1_EMULATION
TCIF
QRCODE
DATABATRIX
MAXICODE
AZTEC
HANXIN

COUPON
POSTAL_2D
PLNET_CODE_CHECK
POSTNET_CHECK
AUS_INTERPRETATION
CHINA_POST
KOREA_POST

In This Section

[Scanner Functions](#) [Scanner Structures](#)

Requirements

The following table shows needed header files and libraries for Scanner APIs.

Header	G2W_HScan_Ctrl.h
Library	G2W_HScan_Ctrl.lib

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) >

1.1.1 Scanner Functions

The following table shows the functions that control 1D scanner.

Function	Description
G2W_InitScanner	This function initializes the scanner library.
G2W_CloseScanner	This function closes the scanner.
G2W_Read	This function reads a barcode data.
G2W_ReadCancel	This function cancels reading a barcode.
G2W_GetScanDataByte	This function registers the windows handle that gets notification from scanner library.
G2W_SetSymValue	This function enables or disables symbologies.
G2W_GetSymValue	This function stores retrieves whether a symbology is enabled.
G2W_SetOptValue	This function stores barcode reading options.
G2W_GetOptValue	This function stores scanner miscellaneous options.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.2 G2W_InitScanner

This function initializes scanner library.

Syntax int int G2W_InitScanner();

Parameters

None.

Return Value Returns ERROR_NONE if successful or an error number if an error occurs. In this function, an error number will be PORT_OPEN_ERROR.

Remarks When this function is called, initializing process - opening serial port and resetting all options to default - is performed internally. The serial port number 4 is assigned to the 1D scanner.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.3 G2W_CloseScanner

This function closes scanner.

Syntax `int G2W_CloseScanner();`

Parameters

None.

Return Value Returns ERROR_NONE if successful or an error number if an error occurs. In this function, an error number will be PORT_CLOSE_ERROR.

Remarks

When this functions is called, cleaning up process - closing serial port and emptying buffers used- is performed internally.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.4 G2W_Read

This function reads a barcode data.

Syntax

```
int G2W_Read();
```

Parameters

None.

Return Value Returns 1 if successful or 0 if an error occurs.

Remarks When this functions is called, the scanner emits beam and tries to read barcode data. The data structure for the barcode is configured internally. After reading, the result is notified by internal callback and window message. An appliation should recieve WM_SCAN_READ after calling G2W_Read and call the function G2W_GetScanDataByte- while processing the message - in order to get barcode type and data.

Sample Source

```
void ScanTest()
{
    .....
    nRet = G2W_Read(); // scanner will be triggered.
    .....
}

----- in the window procedure -----
....
switch(uMessage)
{
    BYTE data[512]={0,};
    BYTE codetype[128]={0,};
    ....
    case WM_SCAN_READ:
        nRet = G2W_GetScanDataByte(data, codetype); // data contains barcode data and codetype contains barcode type.
        .....
        break;
    ....
}
```

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.5 G2W_ReadCancel

This function cancels reading a barcode.

Syntax int G2W_ReadCancel();

Parameters

None.

Return Value Returns 1 if successful or 0 if an error occurs.

Remarks When this functions is called, the scanner stop giving out beam and cancel reading operation.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.6 G2W_RegisterWindow

This function registers windows handle that receives notification from scanner library.

Syntax void G2W_RegisterWindow(
 HWND hWnd
);

Parameters

hWnd [in] A window handle that receives result notification from scanner library.

Return Value None.

Remarks A result of a barcode reading operation is notified by window message. An application should register host window handle by calling this function.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.7 G2W_GetScanDataByte

This function gets barcode data and type.

Syntax void G2W_GetScanDataByte(BYTE* sBarData,
BYTE* sBarType
);

Parameters

sBarData

[in] Points the barcode data read.

sBarType

[in] Point the barcode type read.

Return Value None.

Remarks This function should be called after G2W_Read() function call.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.8 G2W_SetSymValue

This function enables or disables symbologies.

Syntax void G2W_SetSymValue(
 int nsymbology,
 int nsuboption,
 int value
);

Parameters

nsymbology [in] A symbology id. This value is one of the [ID_SYM](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_SYM_OPTION](#) enumeration option.

value [in] int that specifies whether the symbology is enabled

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.9 G2W_GetSymValue

This function retrieves whether a symbology is enabled.

Syntax void G2W_GetSymValue(int nsymbology,
 int nsuboption,
);

Parameters

nsymbology [in] A symbology id. This value is one of the [ID_SYM](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_SYM_OPTION](#) enumeration option.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.10 G2W_SetOptValue

This function sets the operation of the scan module options .

Syntax void G2W_SetOptValue(int noption,
 int nsuboption,
 int value
);

Parameters

noption

[in] A symbology id. This value is one of the [ID_MODULE](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_MODULE_OPTION](#) enumeration option.

value [in] int that specifies whether the module option value.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.11 G2W_GetOptValue

This function gets the operation of the scan module options .

Syntax void G2W_GetOptValue(

int noption,

int nsuboption

);

Parameters

noption

[in] A symbology id. This value is one of the [ID_MODULE](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_MODULE_OPTION](#) enumeration option.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Functions](#) >

1.1.12 G2W_GetVersionInfo

This function retrieves version information string.

Syntax void G2W_GetVersionInfo(
 BYTE *version);

Parameters

version

[in] Pointer to a null-terminated string that specifies the version information string.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) >

1.2 Scanner Enumerations

The following enumeration are used with Scanner functions.

Structure	Description
Symbologies	This enumeration type contains all supported symbology IDs.
Symbology option	This enumeration type contains all supported symbology option .
module option	This enumerations type contains scanner module options.

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Enumerations](#) >

1.2.1 Symbolologies

This enumeration type contains all supported symbology IDs

Syntax`typedef enum {`

```
ID_ALL = 0,  
ID_CODABAR,  
ID_CODE39,  
ID_INT25,  
ID_NEC25,  
ID_CODE93, //5  
ID_STRAIGHT25_IND,  
ID_STRAIGHT25,  
ID_MATRIX25,  
ID_CODE11,  
ID_CODE128, //10  
ID_ISBT,  
ID_GS1_128,  
ID_UPCA_OP,  
ID_UPCA,  
ID_UPCE, //15  
ID_EAN13,  
ID_EAN8,  
ID_MSI,  
ID_GS1_DATABAR_OMNI,  
ID_GS1_DATABAR_LIMIT, //20  
ID_GS1_DATABAR_EXPAND,  
ID_TRIOPTIC_CODE,  
ID_CODABLOCK_A,  
ID_CODABLOCK_F,  
ID_PDF417, //25  
ID_MICRO417,  
ID_GS1_COMPOSITE_CODE,  
ID_GS1_EMULATION,  
ID_TCIF,  
ID_QRCODE, //30  
ID_DATABATRIX,  
ID_MAXICODE,  
ID_AZTEC,  
ID_HANXIN,
```

```

ID_COUPON, //35
ID_POSTAL_2D,
ID_PLNET_CODE_CHECK,
ID_POSTNET_CHECK,
ID_AUS_INTERPRETATION,
ID_CHINA_POST, //40
ID_KOREA_POST
}ID_SYM;

```

Element	symbology	Element	symbology
ID_ALL	All supported symbologies	ID_MICRO417	MicroPDF417
ID_CODABAR	Codabar	ID_GS1_COMPOSITE_CODE	GS1 Composite Codes
ID_CODE39	Code 39	ID_GS1_EMULATION	GS1 Emulation
ID_INT25	Interleaved 2 of 5	ID_GS1_128	GS1 128
ID_NEC25	Nec 2 of 5	ID_TCIF	TCIF Linked Code 39
ID_CODE93	Code 93	ID_QRCODE	QR Code
ID_STRAIGHT25_IND	Straight 2 of 5 industrial	ID_DATABATRIX	Data Matrix
ID_STRAIGHT25	Straight 2 of 5	ID_MAXICODE	MaxiCode
ID_MATRIX25	Matrix 2 of5	ID_AZTEC	Aztec Code

ID_CODE11	code 11	ID_HANXIN	Chinese Sensible Code
ID_CODE128	code128	ID_COUPON	Coupon GS1
ID_ISBT	ISBT concatenation	ID_POSTAL_2D	2D Postal Codes
ID_UPCA_OP	UPCA to ean 13	ID_PLNET_CODE_CHECK	Planet Code Check Digit
ID_UPCA	UPCA	ID_POSTNET_CHECK	Postnet Check Digit
ID_UPCE	UPCE	ID_AUS_INTERPRETATION	Australian Post Interpretation
ID_EAN13	EAN13	ID_CHINA_POST	China Post
ID_EAN8	EAN 8	ID_KOREA_POST	Korea Post
ID_MSI	MSI		
ID_GS1_DATABAR_OMNI	GS1 DataBar Omnidirectional		

ID_GS1_DATABAR_LIMIT	GS1 DataBar Limited		
ID_GS1_DATABAR_EXPAND	GS1 DataBar Expanded		
ID_TRIOPTIC_CODE	Trioptic Code		
ID_CODABLOCK_A	Codablock A		
ID_CODABLOCK_F	Codablock F		
ID_PDF417	PDF417		

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Enumerations](#) >

1.2.2 Symbology option

This enumeration type contains all supported symbology option

Syntax

```
typedef enum {  
    ID_SYM_2DIGIT_ADDENDA =0,  
    ID_SYM_5DIGIT_ADDENDA,  
    ID_SYM_ADDENDA_SEPARATOR,  
    ID_SYM_ADDENDA_REQUIRED,  
    ID_SYM_APPEND,  
    ID_SYM_CODE39_FULL_ASCII,  
    ID_SYM_CODE32_PHARMACEUTICAL,  
    ID_SYM_CONCATENATION,  
    ID_SYM_CHECK_DIGIT,  
    ID_SYM_CHECK_CHAR,  
    ID_SYM_DEFAULT,  
    ID_SYM_GS1_EMULATION,  
    ID_SYM_ENABLE_N_DISABLE,  
    ID_SYM_UPCE0_ENABLE,  
    ID_SYM_UPCE1_ENABLE,  
    ID_SYM_EXPAND,  
    ID_SYM_GS1_DATBAR_OUTPUT,  
    ID_SYM_AUS_POST_INT,  
    ID_SYM_ISBN,  
    ID_SYM_MACRO_PDF,  
    ID_SYM_MINIMUM,  
    ID_SYM_MAXIMUM,  
    ID_SYM_NUMBER_SYSTEM,  
    ID_SYM_START_STOP,  
    ID_SYM_POSTAL_CODE  
}ID_SYM_OPTION;
```

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) > [Scanner Enumerations](#) >

1.2.3 module option

This enumerations type contains scanner module options.

Syntaxtypedef enum {
 ID_MO_TIMEOUT= 0,
 ID_MO_ILLUMINATION_BRIGHT,
 ID_MO_ILLUMINATION_N_AIMER_ENABLE,
 ID_MO_POOR_QULITY_1D,
 ID_MO_POOR_QULITY_PDF,
 ID_MO_VERSION_INF

}ID_MODULE;

typedef enum {
 ID_MO_OPT_TIMEOUT= 0,
 ID_MO_OPT_ILLUMINATION_BRIGHT,
 ID_MO_OPT_ILLUMINATION_ENABLE,
 ID_MO_OPT_AIMER_ENABLE,
 ID_MO_OPT_POOR_QULITY_1D,
 ID_MO_OPT_POOR_QULITY_PDF,
 ID_MO_OPT_VERSION_INF

}ID_MODULE_OPTION;

typedef enum{
 ID_IMBRIGHT_LOW = 15,
 ID_IMBRIGHT_MEDIUM = 50,
 ID_IMBRIGHT_HIGH = 150
 ID_IMMUMINATION_BRIGHT_VALUE;

typedef enum {
 ID_VALUE_OFF =0, //0
 ID_VALUE_ON
 }ID_VALUE;

[SDKs for IDTRONIC Devices](#) > [C/C++](#) > [Scanner](#) >

1.3 Scanner 1D Errors

The following table shows error codes and a description of each.

Error Name	Description	Code
ERROR_NONE	The operation completed successfully.	0
POWER_ON_ERROR	Scanner can not be turned on.	-1
POWER_OFF_ERROR	Scanner can not be turned off.	-2
PORT_OPEN_ERROR	The serial port can not be opened.	-3
PORT_CLOSE_ERROR	The serial port can not be closed.	-4
READ_ERROR	Scanned data can not be read.	-5
READ_CANCEL_ERROR	Reading operation can not be canceled.	-6

[SDKs for IDTRONIC Devices](#) >



2 C#

Overview The SDK for .NET provides all of the tools necessary to develop C# managed applications for IDTRONIC Devices. This SDK includes class libraries for each features. This SDK allows Microsoft® .NET Compact Framework developers to programmatically access the IDTRONIC value-add features of the mobile devices.

In this Section [G2W_ScanLib_Net Assembly](#)

[G2W_ImagerCtrl_Net Assembly](#)

[G2W_WlanCtrl_Net Assembly](#)

[G2W_CameraCtrl_Net Assembly](#)

[G2W_AccCtrl_Net Assembly](#)

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) >

2.1 G2W_HScan_Ctrl_Net Assembly

G2W_HScan_Ctrl_Net.dll provides a C# application with tools (classes and structures) to access and control the one-dimensional scanner.

The following table shows the namespace included in G2W_HScan_Ctrl_Net assembly.

Namespaces	Description
G2W_HScan_Ctrl_Net	This namespace provides a set of class libraries for controlling scanner module.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) >

2.2 G2W_HScan_Ctrl_Net Namespace

This namespace provides a set of class libraries for controlling 1D scanner module.

Classes

Class	Description
HScanCtrlNet	This class provides methods to initialize scanner and read scanned data as well as control scanner.
G2WScannerDataArgs	This class represents the scanned barcode data and type.
ScanMessageWindow	This class provides methods to process window messages.

Enumerations

Structure	Description
Symbologies	This enumeration type contains all supported symbology IDs.
Symbology option	This enumeration type contains all supported symbology option .

module option	This enumerations type contains scanner module options.
-------------------------------	---

Delegates

Delegate	Description
G2WScannerDataDelegate	This delegete is used to signify that a barcode is scanned and the data decoded.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net_Assembly](#) > [G2W_HScan_Ctrl_Net_Namespace](#) >

2.3 Classes

The following table shows the classes that provide barcode reading functionality.

Class	Description
G2WKScanner	This class provides methods to initialize scanner and read scanned data as well as control scanner.
G2WScannerDataArgs	This class represents the scanned barcode data and type.
ScanMessageWindow	This class provides methods to process window messages.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#)

2.3.1 HScanCtrlNet Class

This class provides methods to initialize scanner and read scanned data as well as control scanner.

Syntax public class HScanCtrlNet

Public Methods

Method	Description
G2W_GetVersionInfo_net	This method retrieves the version information string.
G2W_RegisterWindow_net	This method registers a form that receives a notification from scanner library.
G2W_CloseScanner_net	This method closes the scanner port and disposes resources used.
G2W_InitScanner_net	This method initializes the scanner library.
G2W_Read_net	This method reads a barcode data.
G2W_ReadCancel_net	This method cancels reading a barcode.
G2W_GetScanDataByte_net	This method stores barcode types to read.
G2W_SetSymValue_net	This method stores options for Code39 / I2of5 type of barcode.
G2W_GetSymValue_net	This method resets all options to the default.
G2W_SetOptValue_net	This method stores options for EAN / UPC type of barcode.

G2W_GetOptValue_net	This method stores miscellaneous options for scanner.
-------------------------------------	---

Public Events

Event	Description
G2WScannerDataEvent	This event fires when the barcode data is read.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.1.1 G2W_NetGetVersionInfo

This method retrieves the version information string.

Syntax `public string G2W_NetGetVersionInfo();`

Parameters

None.

Return Value The version information string.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.1.2 G2W_RegisterWindow_net

This method registers a form that receives a notification from scanner library.

Syntax public void G2W_RegisterWindow_net();

Parameters

None.

Return Value None.

Remarks G2WKScanner class has a [ScanMessageWindow](#) class object that is derived from MessageWindow class, which processes windows message. Actually the MessageWindow::Hwnd member is registered to class library when G2W_NetRegisterReceiveForm is called.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.1.3 G2W_CloseScanner_net

This method closes the scanner port and disposes resources used.

Syntax `public int G2W_CloseScanner_net();`

Parameters

None.

Return Value Returns ERROR_NONE if successful or an error number if an error occurs.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.1.4 G2W_InitScanner_net

This method initializes the scanner library.

Syntax

```
public int G2W_InitScanner_net();
```

Parameters

None.

Return Value Returns ERROR_NONE if successful or an error number if an error occurs.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.1.5 G2W_Read_net

This method reads a barcode data.

Syntax

```
public int G2W_Read_net();
```

Parameters

None.

Return Value Returns 1 if successful or 0 if an error occurs.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2 G2W_ReadCancel_net

This method cancels reading a barcode.

Syntax `public int G2W_ReadCancel_net();`

Parameters

None.

Return Value Returns 1 if successful or 0 if an error occurs.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2.1 G2W_GetScanDataByte_net

This method stores barcode types to read.

Syntax public void G2W_GetScanDataByte_net(
 ref G2WBarcodeType pBCT
);

Parameters

pBCT

The [G2WBarcodeType](#) structure that specifies the barcode types.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2.2 G2W_SetSymValue_net

This function enables or disables symbologies.

Syntax void G2W_SetSymValue_net(
 ID_SYM nsymbology,
 ID_SYM_OPTION nsuboption
 int value
);

Parameters

nsymbology [in] A symbology id. This value is one of the [ID_SYM](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_SYM_OPTION](#) enumeration option.

value [in] int that specifies whether the symbology is enabled

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2.3 G2W_GetSymValue_net

This function retrieves whether a symbology is enabled.

Syntax void G2W_GetSymValue_net(ID_SYM nsymbology,
ID_SYM_OPTION nsuboption
);

Parameters

nsymbology [in] A symbology id. This value is one of the [ID_SYM](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_SYM_OPTION](#) enumeration option.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2.4 G2W_SetOptValue_net

This function sets the operation of the scan module options .

Syntax void G2W_SetOptValue_net(ID_MODULE noption,
ID_MODULE_OPTION nsuboption,
int value
);

Parameters

noption

[in] A symbology id. This value is one of the [ID_MODULE](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_MODULE_OPTION](#) enumeration option.

value [in] int that specifies whether the module option value.

Return Value None.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Methods](#) >

2.3.2.5 G2W_GetOptValue_net

This function gets the operation of the scan module options .

Syntax void G2W_GetOptValue_net(int noption,
 int nsuboption
);

Parameters

noption

[in] A symbology id. This value is one of the [ID_MODULE](#) enumeration type.

nsuboption [in] A symbology id option .This value is one of the [ID_MODULE_OPTION](#) enumeration option.

Return Value

None.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [HScanCtrlNet Class](#) > [Events](#) >

2.3.2.6 G2WScannerDataEvent

This event fires when the barcode data is read.

Syntax `public event G2WScannerDataDelegate G2WScannerDataEvent;`

Example `G2W_scan = new G2WKScanner();
G2W_scan.G2WScannerDataEvent += new G2WScannerDataDelegate(OnScanData);`

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#)
>

2.3.3 G2WScannerDataArgs Class

This class represents the scanned barcode data and type.

Syntax `public class G2WScannerDataArgs : EventArgs`

Public constructor

Constructor	Description
G2WScannerDataArgs	The default G2WScannerDataArgs constructor.

Properties

Property	Description
ScanData	Returns the barcode data read.
ScanType	Returns the barcode type read.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [G2WScannerDataArgs Class](#) >

2.3.3.1 G2WScannerDataArgs Constructor

The default G2WScannerDataArgs constructor.

Syntax `public G2WScannerDataArgs(
 string stScanData,
 string stScanType
);`

Parameters *stScanData*

Represents the barcode data read.

stScanType

Represents the barcode type read.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [G2WScannerDataArgs Class](#) > [Properties](#) >

2.3.3.2 ScanData

Returns the barcode data read.

Syntax `public string ScanData {get; }`

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [G2WScannerDataArgs Class](#) > [Properties](#) >

2.3.3.3 ScanType

Returns the barcode type read.

Syntax `public string ScanType {get; }`

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#)

2.3.4 ScanMessageWindow Class

This class provides methods to process window messages.

Syntax public class ScanMessageWindow : MessageWindow

Public constructor

Constructor	Description
ScanMessageWindow	The default ScanMessageWindow constructor.

Protected Methods

Method	Description
WndProc	This method processes some window messages differently from the default.

C#	SDK Description

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [ScanMessageWindow Class](#) >

2.3.4.1 ScanMessageWindow Constructor

The default ScanMessageWindow constructor.

Syntax
`public ScanMessageWindow(
 G2WKScanner container
);`

Parameters

container

The [G2WKScanner](#) class object that is the window message container.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Classes](#) > [ScanMessageWindow Class](#) > [Methods](#) >

2.3.4.2 WndProc

This method processes window messages.

Syntaxprotected override void WndProc(
 ref Message msg
);

Parameters *msg*

The Message class object that represents a window message.

Retrun valueNone.

Remarks

This method is overridden in order to process two window messages - WM_HOTKEY and WM_G2W_SCAN_READ.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net_Assembly](#) > [G2W_HScan_Ctrl_Net_Namespace](#) >

2.4 Enumerations

The following enumeration are used with Scanner functions.

Structure	Description
Symbologies	This enumeration type contains all supported symbology IDs.
Symbology option	This enumeration type contains all supported symbology option.
module option	This enumerations type contains scanner module options.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Enumerations](#) >

2.4.1 Symbologies

This enumeration type contains all supported symbology IDs

Syntax typedef enum {

```
ID_ALL = 0,
ID_CODABAR,
ID_CODE39,
ID_INT25,
ID_NEC25,
ID_CODE93, //5
ID_STRAIGHT25_IND,
ID_STRAIGHT25,
ID_MATRIX25,
ID_CODE11,
ID_CODE128, //10
ID_ISBT,
ID_GS1_128,
ID_UPCA_OP,
ID_UPCA,
ID_UPCE, //15
ID_EAN13,
ID_EAN8,
ID_MSI,
ID_GS1_DATABAR_OMNI,
ID_GS1_DATABAR_LIMIT, //20
ID_GS1_DATABAR_EXPAND,
ID_TRIOPTIC_CODE,
ID_CODABLOCK_A,
ID_CODABLOCK_F,
ID_PDF417, //25
ID_MICRO417,
ID_GS1_COMPOSITE_CODE,
ID_GS1_EMULATION,
ID_TCIF,
ID_QRCODE, //30
ID_DATABATRIX,
ID_MAXICODE,
ID_AZTEC,
```

```

ID_HANXIN,
ID_COUPON, //35
ID_POSTAL_2D,
ID_PLNET_CODE_CHECK,
ID_POSTNET_CHECK,
ID_AUS_INTERPRETATION,
ID_CHINA_POST, //40
ID_KOREA_POST
}ID_SYM;

```

Element	symbology	Element	symbology
ID_ALL	All supported symbologies	ID_MICRO417	MicroPDF417
ID_CODABAR	Codabar	ID_GS1_COMPOSITE_CODE	GS1 Composite Codes
ID_CODE39	Code 39	ID_GS1_EMULATION	GS1 Emulation
ID_INT25	Interleaved 2 of 5	ID_GS1_128	GS1 128
ID_NEC25	Nec 2 of 5	ID_TCIF	TCIF Linked Code 39
ID_CODE93	Code 93	ID_QRCODE	QR Code
ID_STRAIGHT25_IND	Straight 2 of 5 industrial	ID_DATABATRIX	Data Matrix
ID_STRAIGHT25	Straight 2 of 5	ID_MAXICODE	MaxiCode
ID_MATRIX25		ID_AZTEC	

	Matrix 2 of5		Aztec Code
ID_CODE11	code 11	ID_HANXIN	Chinese Sensible Code
ID_CODE128	code128	ID_COUPON	Coupon GS1
ID_ISBT	ISBT concatenation	ID_POSTAL_2D	2D Postal Codes
ID_UPCA_OP	UPCA to ean 13	ID_PLNET_CODE_CHECK	Planet Code Check Digit
ID_UPCA	UPCA	ID_POSTNET_CHECK	Postnet Check Digit
ID_UPCE	UPCE	ID_AUS_INTERPRETATION	Australian Post Interpretation
ID_EAN13	EAN13	ID_CHINA_POST	China Post
ID_EAN8	EAN 8	ID_KOREA_POST	Korea Post
ID_MSI	MSI		
ID_GS1_DATABAR_OMNI			

	GS1 DataBar Omnidirectional		
ID_GS1_DATABAR_LIMIT	GS1 DataBar Limited		
ID_GS1_DATABAR_EXPAND	GS1 DataBar Expanded		
ID_TRIOPTIC_CODE	Trioptic Code		
ID_CODABLOCK_A	Codablock A		
ID_CODABLOCK_F	Codablock F		
ID_PDF417	PDF417		

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Enumerations](#) >

2.4.2 Symbology option

This enumeration type contains all supported symbology option

Syntax	typedef	enum	{
ID_SYM_2DIGIT_ADDENDA			=0,
ID_SYM_5DIGIT_ADDENDA,			
ID_SYM_ADDENDA_SEPARATOR,			
ID_SYM_ADDENDA_REQUIRED,			
ID_SYM_APPEND,			
ID_SYM_CODE39_FULL_ASCII,			
ID_SYM_CODE32_PHARMACEUTICAL,			
ID_SYM_CONCATENATION,			
ID_SYM_CHECK_DIGIT,			
ID_SYM_CHECK_CHAR,			
ID_SYM_DEFAULT,			
ID_SYM_GS1_EMULATION,			
ID_SYM_ENABLE_N_DISABLE,			
ID_SYM_UPCE0_ENABLE,			
ID_SYM_UPCE1_ENABLE,			
ID_SYM_EXPAND,			
ID_SYM_GS1_DATBAR_OUTPUT,			
ID_SYM_AUS_POST_INT,			
ID_SYM_ISBN,			
ID_SYM_MACRO_PDF,			
ID_SYM_MINIMUM,			
ID_SYM_MAXIMUM,			
ID_SYM_NUMBER_SYSTEM,			
ID_SYM_START_STOP,			
ID_SYM_POSTAL_CODE			
}ID_SYM_OPTION;			

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Enumerations](#) >

2.4.3 module option

This enumerations type contains scanner module options.

Syntax

```
typedef enum {
    ID_MO_TIMEOUT= 0,
    ID_MO_ILLUMINATION_BRIGHT,
    ID_MO_ILLUMINATION_N_AIMER_ENABLE,
    ID_MO_POOR_QULITY_1D,
    ID_MO_POOR_QULITY_PDF,
    ID_MO_VERSION_INF

}ID_MODULE;

typedef enum {
    ID_MO_OPT_TIMEOUT= 0,
    ID_MO_OPT_ILLUMINATION_BRIGHT,
    ID_MO_OPT_ILLUMINATION_ENABLE,
    ID_MO_OPT_AIMER_ENABLE,
    ID_MO_OPT_POOR_QULITY_1D,
    ID_MO_OPT_POOR_QULITY_PDF,
    ID_MO_OPT_VERSION_INF

}ID_MODULE_OPTION;

typedef enum{
    ID_IMBRIGHT_LOW = 15,
    ID_IMBRIGHT_MEDIUM = 50,
    ID_IMBRIGHT_HIGH = 150
}ID_IMMUMINATION_BRIGHT_VALUE;

typedef enum {
    ID_VALUE_OFF =0, //0
    ID_VALUE_ON
}ID_VALUE;
```

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net_Assembly](#) > [G2W_HScan_Ctrl_Net_Namespace](#) >

2.5 Delegates

The following table shows the delegate in G2W_HScan_Ctrl_Net namespace.

Delegate	Description
G2WScannerDataDelegate	This delegete is used to signify that a barcode is scanned and the data decoded.

[SDKs for IDTRONIC Devices](#) > [C#](#) > [Scanner](#) > [G2W_HScan_Ctrl_Net Assembly](#) > [G2W_HScan_Ctrl_Net Namespace](#) > [Delegates](#)

>

2.5.1 G2WScannerDataDelegate

This delegate is used to signify that a barcode is scanned and the data is decoded.

Syntax public delegate void G2WScannerDataDelegate(
 object sender,
 G2WScannerDataArgs e
);

Parameters

sender

The sender object.

e

A [G2WScannerDataArgs](#) class object.

G2W_ImagerCtrl_Net.dll assembly provides a set of classes that used to access to 2D imager type scanner and to get a barcode data.

The following table shows the namespaces included in G2W_ImagerCtrl_Net.dll assembly.

Namespaces	Description
G2W_ImagerCtrl_Net	This namespace provides a set of class libraries for controlling 2D imager type of scanner module.

G2W_WlanCtrl_Net.dll assembly provides classes and structures that used to control wlan and retrieve informations about radio.

The following table shows the namespaces included in G2W_WlanCtrl_Net.dll assembly.

Namespace	Description
G2W_WlanCtrl_Net	This namespace provides a set of class libraries for controlling WLAN.

G2W_CameraCtrl_Net.dll assembly provides a set of classes that used to access to camera of the device.

The following table shows the namespaces included in G2W_CameraCtrl_Net.dll assembly.

Namespaces	Description
G2W_CameraCtrl_Net	This namespace provides a set of class libraries for controlling camera.

Some IDTRONIC devices have 3-axis digital sensor (accelerometer) - SMB380- to get the position data of a device. Your application can communicate with the sensor through SMB380 Acceleration Sensor API.

G2W_AccCtrl_Net.dll assembly provides G2W_AccCtrl_Net namespace which include a class and structures for accessing the sensor. The public methods included in the class are the imported functions from the SMB380 Acceleration Sensor APIs.

See ***SMB380 Acceleration Sensor API Documentation*** for more information.

This structure specifies barcode type options.

Syntax public struct G2WBarcodeType

Memberspublic bool bG2W_EAN_13 Specifies whether EAN-13 type of barcode is enabled.

public bool bG2W_CODA_BAR Specifies whether Codabar type of barcode is enabled.

public bool bG2W_CODE_128 Specifies whether Code 128 type of barcode is enabled.

public bool bG2W_CODE_39 Specifies whether Code 39 type of barcode is enabled.

public bool bG2W_CODE_35 Specifies whether Code 35 type of barcode is enabled.

public bool bG2W_CODE_93 Specifies whether Code 93 type of barcode is enabled.

public bool bG2W_CODE_I25 Specifies whether Code 125 type of barcode is enabled.

public bool bG2W_EAN_8 Specifies whether EAN-8 type of barcode is enabled.

public bool bG2W_PDF417 Specifies whether PDF417 type of barcode is enabled.

public bool bG2W_UCCEAN_128 Specifies whether UCC/EAN-128 type of barcode is enabled.

public bool bG2W_UPCA Specifies whether UPC-A type of barcode is enabled.

public bool bG2W_UPCE Specifies whether UPC-E type of barcode is enabled.

public bool bG2W_ITF_14 Specifies whether ITF-14 type of barcode is enabled.

public bool bG2W_UPCE_25 Specifies whether the add-on for UPC-E is enabled.

public bool bG2W_UPCA_25 Specifies whether the add-on for UPC-A is enabled.

public bool bG2W_ISBNISSNISMN Specifies whether Bookland type of barcode is enabled.

public bool bG2W_EAN_13_25 Specifies whether the add-on for EAN-13 is enabled.

public bool bG2W_EAN_8_25 Specifies whether the add-on for EAN-8 is enabled.

public bool bG2W_CODE35_REVERSE Specifies whether Code 35 reverse type of barcode is enabled.

public bool bG2W_CODE35_REVERSE Specifies whether Code 35 reverse type of barcode is enabled.

Constructor public G2WBarCodeType(

```
bool bG2W_EAN_13,  
bool bG2W_CODA_BAR,  
bool bG2W_CODE_128,  
bool bG2W_CODE_39,  
bool bG2W_CODE_35,  
bool bG2W_CODE_93,  
bool bG2W_CODE_125,  
bool bG2W_EAN_8,  
bool bG2W_PDF417,  
bool bG2W_UCCEAN_128,  
bool bG2W_UPCA,  
bool bG2W_UPCE,  
bool bG2W_ITF_14,  
bool bG2W_UPCE_25,  
bool bG2W_UPCA_25,  
bool bG2W_ISBNISSNISMN,  
bool bG2W_EAN_13_25,  
bool bG2W_EAN_8_25,  
bool bG2W_CODE35_REVERCE  
);
```

The parameters of the constructor specifies whether the each type of barcode is enabled

This structure specifies barcode reading options.

Syntax `public struct G2WReadOption;`

Members
`public bool bG2W_WIDESCANANGLE` Specifies whether the wide scan angle option is enabled.

`public bool bG2W_RETURNCHECK` Specifies whether the checking return value option is enabled.

`public bool bG2W_ERRORCHECK` Specifies whether the checking error option is enabled.

`public bool bG2W_HIGHFILTERMODE`
Specifies whether the high filter mode is enabled.

`public bool bG2W_SCANSOUND`
Specifies whether the playing scan sound option is enabled.

`public bool bG2W_VIBRATION`
Specifies whether the vibrating at scanning option is enabled.

Constructor
`public G2WReadOption(
 bool bG2W_WIDESCANANGLE,
 bool bG2W_RETURNCHECK,
 bool bG2W_ERRORCHECK,
 bool bG2W_HIGHFILTERMODE,
 bool bG2W_SCANSOUND,
 bool bG2W_VIBRATION
);`

The parameters of the constructor specifies whether the each barcode reading options are enabled.

This structure specifies the miscellaneous scanner options.

Syntax `public struct G2WModuleOption;`

Members `public int nG2W_TimeOutSec`

Specifies the timeout value of reading in second.

`public int nG2W_MinLen`

Specifies the minimum length of barcode data.

`public int nG2W_SecurityLevel`

Specifies the security level.

Constructor `public G2WModuleOption(`

`int nG2W_TimeOutSec,`

`int nG2W_MinLen,`

`int nG2W_SecurityLevel`

`);`

The parameters of the constructor specifies the each scanner options.

This namespace provides a set of class libraries for controlling 2D imager type of scanner module.

Classes

Class	Description
G2W_Imager	This class is the base class for controlling 2D imager type of scanner.
G2W_Camera	This class provides methods to control camera functions of the imager.
G2W_Scanner	This class provides methods to read barcode.
G2W_ScannerControl	This class provides methods to get notification of reading reading barcode.
ScannerDataArgs	This class contains event data of scanning.
ScanMessageWindow	This class provides methods to process window messages.

Enumerations

Enumeration	Description
OCRDIRECTION	This enumeration type contains the direction of OCR.
OCRMode	This enumeration type contains the OCR modes.
ScanIlluminat	This enumeration type contains on/off options for the aimer and the illuminator.

SetupType	This enumeration type contains ???
SYMID	This enumeration type contains all supported symbology IDs.

Delegates

Delegate	Description
ScannerDataDelegate	This delegete is used to signify that a barcode is scanned and the data decoded.
ScannerTimeoutDelegate	This delegete is used to signify the barcode reading timeout.

This namespace provides a set of class libraries for controlling WLAN.

Classes

Class	Description
WLAN_CTRL_NET	This class retrieves the information about wifi operation and controls power of wifi module.

Structures

Structure	Description
G2W_INT_BSSID_LIST	This structure ????
G2W_WLAN_BSSID_LIST	This structure represents a BSSID list item.

This namespace provides a set of class libraries for controlling camera.

Classes

Class	Description
Cameractrl_net	This class provides methods to control camera.
CameraCtrl_net_power	This class provides methods to get a notification of changing device power status.

Enumerations

Enumeration	Description
CAMERA_MODE	This enumerations represents camera mode.
VIDEO_TYPE	This enumerations represents the video streaming format.

This class is the base class for controlling 2D imager type of scanner.

Syntax public class G2W_Imager

Public Methods

Methods	Description
G2W_AimerOn	This method turns on or off the scanner aimer.
G2W_Connect	This method initializes the platform APIs in order to utilize the camera functions.
G2W_GetInfo	This method inquires the information about the 2D imager
G2W_LightsOn	This method turns the illuminator on or off.
G2W_VibrateOn	This method enables or disables the vibration mode.
GetImageData	This method retrieves the decoded data and the length.
SHSetAppKey	This method assigns a window to receive a particular hardware button's key-press messages.

Public Structures

Structure	Description
IMAGE_VERSION_INFO	This structure represents a few versions of a imager.

This class provides methods to control camera functions of the imager.

Syntax `public class G2W_Camera : G2W_Imager`

Public Methods

Methods	Description
G2W_Capture	This method captures an image and save it into a file.
G2W_GetOption	This method retrieves camera options.
G2W_Init	This method initializes camera options.
G2W_PreviewStart	This method starts preview.
G2W_PreviewStop	This method stops preview.
G2W_SetOption	This method stores camera options.
G2W_Uninit	This method uninitializes camera options.

Public Structures

Structure	Description
CAM_OPTION	This structure specifies camera options.

This class provides methods to read barcode.

Syntax public class G2W_Scanner : G2W_Imager

Public Methods

Methods	Description
G2W_CancelIO	This method cancels a reading operation.
G2W_DefaultSymbology	This method enables a symbology as a default.
G2W_GetDecodeCenteringWindow	This method retrieves the rectangle area for the decoding.
G2W_GetEnableDisableSymbology	This method retrieves whether a symbology is enabled.
G2W_GetScanResult	This method retrieves the result of a reading operation.
G2W_MultiDecodeModeEnable	This method specifies whether the multi-decode mode is enabled.
G2W_ReadSymbologyConfig	This method reads the flags which present the options of a symbology.
G2W_ScanLed	This method turns on or off the scanner led.
G2W_ScanRead	This method triggers the reading operation.
G2W_SetDecodeCenteringWindow	This method sets a rectangle area for the decoding.

G2W_SetEnableDisableSymbology	This method enables or disables symbologies.
G2W_SetScanningLightsMode	This method sets the operation of the illuminator and the aimer.
G2W_StartIntellImgXfer	This method ??????
G2W_WriteSymbologyConfig	This method writes the flags which present the options of a symbology.

Public Structures

Structure	Description
DECODE_MSG	This structure specifies decoded message which contains data.
SymFlagsOnly	This structure specifies flags for a symbology.
SymFlagsRange	This structure specifies the flags and range limit for a symbology.
SymCodeOCR	This structure specifies the options for OCR.
IntellImg_t	This structure ???
RECT	This structure specifies a rectangle.

Public Enumerations

Enumerations	Description
EVENT_TYPE	This enumeration type contains the event types.

This class provides methods to get notification of reading barcode.

Syntax public class G2W_ScannerControl

Public Methods

Methods	Description
G2W_RegisterRecieveForm	This method registers a form that receives a notification from scanner library.
OnScanDataMsg	This method is called by 2D scanner library when the host form (window) recieves WM_SCAN_TIMEOUT window message.
OnScanTimeoutMsg	This method is called by 2D scanner library when the host form (window) recieves WM_SCAN_DATA window message.

Public Events

Event	Description
ScannerDataEvent	This event fires when the barcode data is read.
ScannerTimeoutEvent	This event fires when the barcode reading time-out interval elapsed and the barcode is not read.

This class contains event data of scanning.

Syntax `public class ScannerDataArgs : EventArgs`

This class provides methods to process window messages.

Syntax `public class ScanMessageWindow : MessageWindow`

Public constructor

Constructor	Description
ScanMessageWindow	The default ScanMessageWindow constructor.

Protected Methods

Method	Description
WndProc	This method processes some window messages differently from the default.

This enumeration type contains the direction of OCR.

Syntax

```
public enum ORCDirection{  
    OCR_DIRECTION_LeftToRight = 0,  
    OCR_DIRECTION_TopToBottom,  
    OCR_DIRECTION_RightToLeft,  
    OCR_DIRECTION_BottomToTop,  
};
```

Elements **OCR_DIRECTION_LeftToRight** Specifies the direction of OCR is from left to right.

OCR_DIRECTION_TopToBottom Specifies the direction of OCR is from top to bottom.

OCR_DIRECTION_RightToLeft Specifies the direction of OCR is from right to left.

OCR_DIRECTION_BottomToTop Specifies the direction of OCR is from bottom to top.

This enumeration type contains the OCR modes.

Syntax

```
public enum OCRMode
{
    OCR_MODE_DISABLED = 0,
    OCR_MODE_A,
    OCR_MODE_B,
    OCR_MODE_MONEY,
    OCR_MODE_MICR_UNSUPPORTED,
};
```

Elements **OCR_MODE_DISABLED** OCR mode is disabled.

OCR_MODE_A Represents that OCR mode A is enabled.

OCR_MODE_B Represents that OCR mode A is enabled.

OCR_MODE_MONEY Represents that OCR mode for money is enabled.

OCR_MODE_MICR_UNSUPPORTED Represents OCR mode unsupported currently.

This enumeration type contains on/off options for the aimer and the illuminator.

Syntax public enum ScanIlluminat

```
{  
    ILLUM_AIMER_OFF = 0,  
    ILLUM_ONLY_ON,  
    AIMER_ONLY_ON,  
    ILLUM_AIMER_ON  
};
```

Elements **ILLUM_AIMER_OFF** Specifies both the aimer and illuminator are off.

ILLUM_ONLY_ON

Specifies only the illuminator is on.

AIMER_ONLY_ON

Specifies only the aimer is on.

ILLUM_AIMER_ON Specifies both the aimer and illuminator are on.

This enumeration type contains ???

Syntax public enum SetupType
{
 SETUP_DEFAULT = 0,
 SETUP_CURRENT,
};

Elements **SETUP_DEFAULT** Represents the default setup type.

SETUP_CURRENT Represents the current setup type.

This enumeration type contains all supported symbology IDs.

Syntaxpublic enum SYMID

```
{
    ID_TYPE_AZTEC = 0,
    ID_TYPE_MESA,
    ID_TYPE_CODABAR,
    ID_TYPE_CODE11,
    ID_TYPE_CODE128,
    ID_TYPE_CODE39,
    ID_TYPE_CODE49,
    ID_TYPE_CODE93,
    ID_TYPE_COMPOSITE,
    ID_TYPE_DATAMATRIX,
    ID_TYPE_EAN8,
    ID_TYPE_EAN13,
    ID_TYPE_INT25,
    ID_TYPE_MAXICODE,
    ID_TYPE_MICROPDF,
    ID_TYPE_OCR,
    ID_TYPE_PDF417,
    ID_TYPE_POSTNET,
    ID_TYPE_QR,
    ID_TYPE_RSS,
    ID_TYPE_UPCA,
    ID_TYPE_UPCE0,
    ID_TYPE_UPCE1,
    ID_TYPE_ISBT,
    ID_TYPE_BPO,
    ID_TYPE_CANPOST,
    ID_TYPE_AUSPOST,
    ID_TYPE_IATA25,
    ID_TYPE_CODABLOCK,
    ID_TYPE_JAPOST,
    ID_TYPE_PLANET,
    ID_TYPE_DUTCHPOST,
    ID_TYPE_MSI,
    ID_TYPE_TLCODE39,
    ID_TYPE_TRIOPTIC,
    ID_TYPE_CODE32,
    ID_TYPE_STRT25,
    ID_TYPE_MATRIX25,
    ID_TYPE_PLESSEY,
    ID_TYPE_CHINAPOST,
    ID_TYPE_KOREAPOST,
    ID_TYPE_TELEPEN,
    ID_TYPE_CODE16K,
    ID_TYPE_POSICODE,
    ID_TYPE_COUPONCODE,
```

```

ID_TYPE_USPS4CB,
ID_TYPE_IDTAG,
ID_TYPE_GS1_128,
ID_TYPE_GEN_CODE128,
ID_TYPE_ALL = 100
};

```

Elements

Each elements represent barcode symbologies as the following table.

Element	symbology	Element	symbology
ID_TYPE_TYPE_AZTEC	Aztec Code	ID_TYPE_TYPE_MESA	Mesa
ID_TYPE_CODABAR	Codabar	ID_TYPE_CODE11	Code 11
ID_TYPE_CODE128	Code 128	ID_TYPE_CODE39	Code 39
ID_TYPE_CODE49	Code 49	ID_TYPE_CODE93	Code 93
ID_TYPE_COMPOSITE	EAN/UCC Composite	ID_TYPE_DATAMATRIX	Data Matrix
ID_TYPE_EAN8	EAN-8	ID_TYPE_EAN13	EAN-13
ID_TYPE_INT25	Interleaved 2 of 5	ID_TYPE_MAXICODE	MaxiCode
ID_TYPE_MICROPDF	MicroPDF417	ID_TYPE_OCR	OCR
ID_TYPE_PDF417	PDF417	ID_TYPE_POSTNET	Postnet
ID_TYPE_QR	QR Code	ID_TYPE_RSS	Reduced Space Symbology
ID_TYPE_UPCA	UPC-A	ID_TYPE_UPCE0	UPC-E0
ID_TYPE_UPCE1	UPC-E1	ID_TYPE_ISBT	ISBT 128

ID_TYPE_BPO	British Post	ID_TYPE_CANPOST	Canadian Post
ID_TYPE_AUSPOST	Australian Post	ID_TYPE_IATA25	Straight 2 of 5 IATA
ID_TYPE_CODABLOCK	Codablock F	ID_TYPE_JAPOST	Japanese Post
ID_TYPE_PLANET	Planet Code	ID_TYPE_DUTCHPOST	KIX (Netherlands) Post
ID_TYPE_MSI	MSI	ID_TYPE_TLCODE39	TCIF Linked Code 39 (TLC39)
ID_TYPE_TRIOPTIC	Trioptic Code 39	ID_TYPE_CODE32	Code 32
ID_TYPE_STRT25	Straight 2 of 5	ID_TYPE_MATRIX25	Matrix 25
ID_TYPE_PLESSEY	Plessey Code	ID_TYPE_CHINAPOST	China Post
ID_TYPE_KOREAPOST	Korean Post	ID_TYPE_TELEPEN	Telepen
ID_TYPE_CODE16K	Code 16K	ID_TYPE_POSICODE	PosiCode
ID_TYPE_COUPONCODE	UPC-A with extended Coupon Code	ID_TYPE_USPS4CB	USPS-4CB
ID_TYPE_IDTAG	ID Tag	ID_TYPE_GS1_128	GS1-128
ID_TYPE_GEN_CODE128-		ID_TYPE_ALL	All supported symbologies.

This delegate is used to signify that a barcode is scanned and the data decoded.

Syntax public delegate void ScannerDataDelegate(
 object sender,
 ScannerDataArgs e
);

Parameters

sender

The sender object.

e

A [ScannerDataArgs](#) class object.

This delegate is used to signify the barcode reading timeout.

Syntax

```
public delegate void ScannerDataDelegate(  
    object sender,  
);
```

Parameters

sender

The sender object.

This class retrieves the information about wifi operation and controls power of wifi module.

Syntax `public class WLAN_CTRL_NET;`

Public Methods

Methods	Description
G2WNet_AssociatedSsid_DATA	This method sets the information about authentication and associates.
G2WNet_DelBSListcf	This method deletes all bssid list information queried.
G2WNet_Get_AdapterName	This method retrieves the names of adapters currently instantiated by NDIS.
G2WNet_Get_Bssid	This method retrieves the MAC address of the associated access point.
G2WNet_Get_Configuration	This method retrieves the configuration of a wifi radio.
G2WNet_Get_Connect_Status	This method retrieves the connection status.
G2WNet_Get_LinkSpeed	This method retrieves the maximum speed of the NIC in kbps.
G2WNet_Get_NICMACAddress	This method retrieves the MAC address of the NIC.
G2WNet_Get_RssiValue	This method retrieves the signal strength of wlan.
G2WNet_Get_SsidName	This method retrieves the SSID name with which current NIC is associated.
G2WNet_Get_WlanPower_Status	This method inquiries power status of a wlan.
G2WNet_ResetPreferred	This method resets preferred networks.
G2WNet_Set_BssidList	This method retrieves a list containing all basic service set identifiers(BSSIDs) and their attributes.

G2WNet_Set_WlanPower_OnOff	This method turns on / off a wlan.
--	------------------------------------

This structure represents a BSSID list item.

Syntax `public struct G2W_INT_BSSID_LIST;`

Members
`public int Privacy`

Specifies a WEP encryption requirement.

`public int Rssi`

Specifies the signal strength.

`public int BeaconPeriod`

Specifies the interval between beacon message transmissions.

`public int ATIMWindow`

Specifies the announcement traffic information message (ATIM) window in Kμsec (1024 μsec).

`public int DSConfig`

Specifies the frequency of the selected channel in kHz.

`public int InfrastructureMode`

Specifies the infrastructure mode.

This structure represents a BSSID list item.

Syntax `public struct G2W_WLAN_BSSID_LIST;`

Members

public int Privacy

Specifies a WEP encryption requirement.

public int Rssi

Specifies the signal strength.

public int BeaconPeriod

Specifies the interval between beacon message transmissions.

public int ATIMWindow

Specifies the announcement traffic information message (ATIM) window in Kμsec (1024 μsec).

DSConfig

Specifies the frequency of the selected channel in kHz.

InfrastructureMode

Specifies the infrastructure mode.

public string MAXADDRESS Specifies the MAC address.

public string SSID

Specifies the SSID.

This class provides methods to control camera.

Syntax `public class Cameractrl_net;`

Public Methods

Method	Description
G2W_Capture	This method captures a still image into a file.
G2W_Close	This method closes camera and clean up any resource used.
G2W_Flash_On	This method turns on or off camera flash.
G2W_Get_Brightness	This method retrieves the index that represents the brightness.
G2W_Get_Info	This method retrieves the camera dll version string.
G2W_Get_Quality	This method retrieves the index that represents the image quality.
G2W_Get_Resolution	This method retrieves the index that represents the current resolution.
G2W_Get_WhiteBalance	This method retrieves the index that represents the white balance.
G2W_Open	This method opens and initialize camera device.
G2W_Preview_Start	This method starts preview.
G2W_Preview_Stop	This method stops preview.
G2W_Set_Brightness	This method sets the index that represents the brightness.
G2W_Set_Quality	This method sets the index that represents the image quality.
G2W_Set_Resolution	This method sets the index that represents the resolution.

G2W_Set_WhiteBalance	This method sets the index that represents the white balance.
G2W_SetAutoFocus	This method triggers the auto focus.
G2W_SetPosition	This method sets the position of the video window in device coordinates.
G2W_SHSetAppKey	This method assigns a window to receive a particular hardware button's key-press messages.
G2W_Video_Start	This method starts capturing video.
G2W_Video_Stop	This method stops capturing video.

This class provides methods to get a notification of changing device power status.

Syntax `public class Cameractrl_net;`

Public constructor

Constructor	Description
CameraCtrl_net_power	The default CameraCtrl_net_power constructor.

Public Methods

Method	Description
Start	This method starts the thread for monitoring power state of device.
Stop	This method stops the thread for monitoring power state of device.

This enumerations represents camera mode.

Syntaxpublic enum CAMERA_MODE{
 STILL_MODE = 0,
 VIDEO_MODE
};

Elements

STILL_MODE Specifies the camera runs in the mode of capturing still images.

VIDEO_MODE

Specifies the camera runs in the mode of capturing video streaming.

This enumerations represents the video streaming format.

Syntax

```
public enum VIDEO_TYPE{  
    VIDEO_ASF = 0,  
    VIDEO_WMV  
};
```

ElementsVIDEO_ASF

Specifies the video streaming format is *.asf.

VIDEO_WMV

Specifies the video streaming format is *.wmv.

This method turns on or off the scanner aimer.

Syntax public bool G2W_AimerOn(
 bool On
);

Parameters

On Boolean that specifies whether the aimer is turned on.

Return Value Returns true if successful or false if an error occurs.

Remarks The 2D scanner emits two kinds of beams. The one is aimer and the other is illuminator. This function controls the aimer.

This method initializes the platform APIs in order to utilize the camera functions.

Syntax public bool G2W_Connect(
 bool On
);

Parameters *On*

Boolean that specifies whether the connecting operation is performed.

Return Value Returns true if successful or false if an error occurs.

Remarks The main purpose of this function is initializing a dll which implements functions controlling the 2D scanner (imager). After calling this function successfully, all other APIs handling the 2D scanner (imager) works properly.

This method inquiries the information about the 2D imager.

Syntax public bool G2W_GetInfo(
 ref IMAGER_VERSION_INFO verinfo
);

Parameters *verinfo*

The [IMAGER_VERSION_INFO](#) structure that contains informations.

Return Value Returns true if successful or false if an error occurs.

This method turns the illuminator on or off.

Syntax public bool G2W_LightsOn(
 bool On
);

Parameters *On*

Boolean that specifies whether the illuminator is turned on.

Return Value Returns true if successful or false if an error occurs.

This method enables or disables the vibration mode.

Syntax public bool G2W_VibrateOn(
 bool On
);

Parameters *On*

Boolean that specifies whether the vibration is on.

Return Value Returns true if successful or false if an error occurs.

This method retrieves the decoded data and the length.

Syntax public bool GetImageData(
int nNumberToRead,
byte[] pBuffer,
ref int nNumBytesRead
);

Parameters *dwNumberToRead* The number of bytes to read.

pBuffer A buffer that contains data.

pNumBytesRead
The number of bytes read.

Return Value Returns true if successful or false if an error occurs.

This method assigns a window to receive a particular hardware button's key-press messages.

Syntax public bool SHSetAppKey(
byte bVk,
IntPtr hMainWnd
);

Parameters *bVk*

Key code for the hardware button.

hMainWnd
Handle to the window that will receive the messages.

Return Value Returns true if successful or false if an error occurs.

This structure represents the information about the 2D imager.

Syntax `public struct IMAGER_VERSION_INFO;`

Members `[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 128)]`

`public string tcAPIRev` A string that contains SDK API version string.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 128)]`

`public string tcDecoderRev` A string that contains decoder revision.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 128)]`

`public string tcScanDriverRev` A string that scan driver revision.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 1000)]`

`public string tcEtcInfo` A data buffer that contains etc information.

`public int dwFirmwareVersion`

Specifies the imager firmware version.

`public int dwFirmwareCksum`

Specifies the imager firmware version.

`public int dwEngineId`

Specifies the engine id of imager.

This method captures an image and save it into a file.

Syntax public bool G2W_Capture(
 string FileFullName
);

Parameters

FileFullName

[in] The string which specifies the full path of the file.

Return Value Returns true if successful or false if an error occurs.

This method retrieves camera options.

Syntax public bool G2W_GetOption(
 ref CAM_OPTION option);

Parameters

option

The [CAM_OPTION](#) structure that contains the camera options.

Return Value Returns true if successful or false if an error occurs.

This method initializes camera options.

Syntax public bool G2W_CamInit(
 IntPtr hMainWnd,
 IntPtr hPictureWnd
);

Parameters

hMainWnd

[in] Handle to the host window. The host window communicates with library.

hPictureWnd

[in] Handle to the window on which a preview image displays.

Return Value Returns true if successful or false if an error occurs.

This method starts preview.

Syntax `public bool G2W_PreviewStart();`

Parameters

None.

Return Value Returns true if successful or false if an error occurs.

This method stops preview.

Syntax

`public bool G2W_PreviewStop();`

Parameters

None.

Return Value Returns true if successful or false if an error occurs.

This method stores the camera options.

Syntax `public bool G2W_SetOption(
CAM_OPTION option);`

Parameters

Option

The [CAM_OPTION](#) structure which specifies the camera options.

Return Value Returns true if successful or false if an error occurs.

This method uninitializes camera options.

Syntax `public bool G2W_Uninit();`

Parameters

None.

Return Value Returns `true` if successful or `false` if an error occurs.

This structure specifies camera options.

Syntax `public struct CAM_OPTION;`

Members `public int nTop` Specifies the y-coordinate of the upper-left corner of the image rectangle.

public int nLeft

Specifies the x-coordinate of the upper-left corner of the image rectangle.

public int nRight

Specifies the x-coordinate of the lower-right corner of the image rectangle.

public int nBottom

Specifies the y-coordinate of the lower-right corner of the image rectangle.

public short nSaveFormat

Specifies the image file format. 0 represents bitmap and 1 represents jpeg.

public int nJpegQuality

Specifies the jpeg quality. The range of this value is ?????

public int nWhiteValue

Specifies the white balance indexes.

public int nResolution

Specifies the resolution of image.

public bool bInvert

Specifies whether an image is inverted.

This method cancels a reading operation.

Syntax `public bool G2W_CancelIO() ;`

Parameters

None.

Return Value Returns `true` if successful or `false` if an error occurs.

This method enables a symbology as a default.

Syntax `public bool G2W_DefaultSymbology(
 SYMID nSymId
);`

Parameters

nSymId

A [SYMID](#) enumeration type that represents a symbology.

Return Value Returns `true` if successful or `false` if an error occurs.

This method retrieves the rectangle area for the decoding.

Syntax `public bool G2W_GetDecodeCenteringWindow(
 SetupType SetupType,
 ref bool pbEnabled,
 ref RECT pIntersectRect
);`

Parameters

SetupType

[SetupType](#) enumeration type that specifies the ????

pbEnabled

A boolean that represents whether the area is enabled.

pIntersectRect A [RECT](#) structure that stores the decode rectangle area.

Return Value Returns `true` if successful or `false` if an error occurs.

This method retrieves whether a symbology is enabled.

Syntax public bool G2W_GetEnableDisableSymbology(SetupType SetType,
SYMID nSymId,
ref bool Enable
);

Parameters

SetupType

[SetupType](#) enumeration type that specifies the ????

nSymId An [SYMID](#) enumeration type that represents a symbology.

Enable A a boolean that represents whether the symbology is enabled.

Return Value Returns true if successful or false if an error occurs.

This method retireves the result of a reading operation.

Syntax public bool G2W_GetScanResult(ref EVENT_TYPE pEventType,
ref DECODE_MSG msg
);

Parameters

pEventType

An [EVENT_TYPE](#) enumeration type that represents the type of events.

msg

A [DECODE_MSG](#) structure that contains the result of scan.

Return Value Returns true if successful or false if an error occurs.

This method specifies whether the multi-decode mode is enabled.

Syntax `public bool G2W_MultiDecodeModeEnable(bool bEnable
);`

Parameters

bEnable A boolean that specifies whether the multi decode mode is enabled.

Return Value Returns `true` if successful or `false` if an error occurs.

This method reads the flags which present the options of a symbology.

Syntax `public bool G2W_ReadSymbologyConfig(`

```
    SetupType SetupType,  
    SYMID nSymbol,  
    ref SymFlagsOnly config);
```

```
public bool G2W_ReadSymbologyConfig(
```

```
    SetupType SetupType,  
    SYMID nSymbol,  
    ref SymFlagsRange config
```

```
);
```

```
public bool G2W_ReadSymbologyConfig(
```

```
    SetupType SetupType,  
    ref SymCodeOCR config
```

```
);
```

Parameters

SetupType

A SetupType enumeration type that specifies ????

nSymbol

An SYMID enumeration type that represents a symbology.

config

A [SymFlagsOnly](#) structure. Acutally this structure has only a int member that stores bit field of flags.

A [SymFlagsRange](#) structure.

A [SymCodeOCR](#) structure.

Return Value Returns TRUE if successful or FALSE if an error occurs.

This method turns on or off the scanner led.

Syntax public bool G2W_ScanLed(
 bool on);

Parameters

on A boolean that specifies whether the led is turned on.

Return Value Returns true if successful or false if an error occurs.

This method triggers the reading operation.

Syntax public bool G2W_ScanRead(int dwTimeout,
 ref DECODE_MSG pmsg
);

Parameters

dwTimeout

Specifies the time-out interval in seconds.

pmsg

A [DECODE_MSG](#) structure that contains the result of reading operation.

Return Value Returns true if successful or false if an error occurs.

This method sets a rectangle area for the decoding.

Syntax `BOOL G2W_SetDecodeCenteringWindow(bool bEnable,
 ref RECT pIntersectRect
);`

Parameters

bEnable A boolean that specifies whether the center area is used for the decoding. If *bEnable* sets to false *pIntersectRect* parameter is ignored.

pIntersectRect

A [RECT](#) structure represent the decoding area.

Return Value Returns `true` if successful or `false` if an error occurs.

xThis method enables or disables symbologies.

Syntax `public bool G2W_SetEnableDisableSymbology(SYMID nSymId,
 bool bEnable
);`

Parameters

nSymId

An [SYMID](#) enumeration type that represents a symbology.

bEnable

A boolean that specifies whether the symbology with *nSymId* is enabled.

Return Value Returns `true` if successful or `false` if an error occurs.

This method sets the operation of the illuminator and the aimer.

Syntax public bool G2W_SetScanningLightsMode(ScanIlluminat nIllumMode
);

Parameters

nIllumMode

[in] A [ScanIlluminat](#) enumeration type.

Return Value Returns true if successful or false if an error occurs.

This method starts an IQ Image transfer.

Syntax BOOL G2W_StartIntellImgXfer(ref IntellImg_t ImgDesc
);

Parameters

ImgDesc [in] A [IntellImg_t](#) sturcture.

Return Value Returns true if successful or false if an error occurs.

Remarks

The IQ Image is an image relative to the center of a supported decoded bar code. Supported bar codes are: PDF417, Code 128, Code 39, and Aztec. The user specifies the width, height, and center of the image to be retrieved. This image is independent of any rotation of the bar code relative to the Imager. Thus, if the bar code is decoded with the code itself upside down to the Imager, the IQ Image will still be right side up. Note, however, if the specified image is outside the field of view, FALSE will be returned.

This method writes the flags which present the options of a symbology or OCR.

Syntax `public bool G2W_WriteSymbologyConfig(SYMID nSymbol,
 SymFlagsOnly config
);`

`public bool G2W_WriteSymbologyConfig(SYMID nSymbol,
 SymFlagsRange config
);`

`public bool G2W_WriteSymbologyConfig(
 SymCodeOCR config
);`

Parameters

nSymbol

An SYMID enumeration type that represents a symbology.

config A [SymFlagsOnly](#) structure. Actually this structure has only a int member that stores bit field of flags.

A [SymFlagsRange](#) structure.

A [SymCodeOCR](#) structure.

Return Value Returns true if successful or false if an error occurs.

This structure specifies decoded message which contains data.

Syntax public struct DECODE_MSG;

Members public int dwStructSize

Specifies the size of decode structure.

[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 4096)]

public string pchMessage

Specifies the decoded message data.

public ushort chCodeID

Specifies the AIM Id of symbology.

public ushort chSymLetter

Specifies the HHP Id of symbology.

public ushort chSymModifier

Specifies the modifier characters.

public int nLength

Specifies the length of the decoded message.

This structure specifies flags for a symbology.

Syntax public struct SymFlagsOnly;

Members

public int nFlags

Specifies the ORed of valid flags for the given symbology.

This structure specifies the flags and range limit for a symbology.

Syntax `public struct SymFlagsRange;`

Members`public int nFlags`

Specifies the ORed of valid flags for the given symbology.

`public int nMinLen`

Specifies the minimum length for valid barcode string for this symbology.

`public int nMaxLen`

Specifies maximum length for valid bar code string for this symbology.

This structure specifies the options for OCR.

Syntax `public struct SymCodeOCR;`

Members`public OCRMode ocrMode`

An [OCRMode](#) enumeration type that enables or specifies OCR modes.

`public OCRDirection ocrDirection`

An [OCRDirection](#) enumeration type that specifies the direction of OCR.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]`

`public string tcTemplate`

The string that represents the template for decoded data ('d' - decimal, 'a' - ASCII, 'l' - letter, 'e' - extended).

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]`

`public string tcGroupG`

The string that contains the group G character string.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 256)]`

`public string tcGroupH`

The string that contains the group H character string.

`[MarshalAs(UnmanagedType.ByValTStr, SizeConst = 64)]`

`public string tcCheckChar`

The string that contains the check character string.

This structure sets up intelligent imaging settings.

Syntax `public struct IntellImg_t;`

Members`public int AspectRatio`

Represents the ratio of barcode height to narrow elem width.

`public int OffsetX`

Represents the offset in X direction, relative to barcode center.

`public int OffsetY`

Represents the offset in Y direction.

`public uint width`

Represents the width of image in IntellBarcodeUnits.

`public uint height`

Represents the height of image.

This structure specifies a rectangle.

Syntax `public struct RECT;`

Members`public int left`

Specifies the x-coordinate of the upper-left corner of the rectangle.

`public int top` Specifies the y-coordinate of the upper-left corner of the rectangle.

`public int right`

Specifies the x-coordinate of the lower-right corner of the rectangle.

`public int bottom`

Specifies the y-coordinate of the lower-right corner of the rectangle.

This enumeration type contains the event types.

Syntax public enum EVENT_TYPE

```
{  
    BARCODE_EVENT = 0,  
    IMAGE_EVENT,  
    TEXT_MSG_EVENT,  
    INTELIMG_BARCODE_EVENT,  
    INTELIMG_IMAGE_EVENT,  
    TRIGGER_EVENT  
};
```

Elements

BARCODE_EVENT Represents that the event is related to the decoding.

IMAGE_EVENT

Represents that the event is related to processing image.

TEXT_MSG_EVENT

Represents that the event is related to the text message.

INTELIMG_BARCODE_EVENT Represents that the event is related to Intel Image barcode unit.

INTELIMG_IMAGE_EVENT

Represents that the event is related to Intel Image.

TRIGGER_EVENT

Represents that the event is related to the hardware trigger.

This method registers a form that receives a notification from scanner library.

Syntax public void G2W_RegisterRecieveForm();

Parameters

None.

Return Value None.

Remarks G2W_ScannerControl class has a [ScanMessageWindow class](#) object that is deviced from MessageWindow class, which processes windows message. Actually the MessageWindow::Hwnd member is registered to class library when G2W_RegisterRecieveForm is called.

This method is called by 2D scanner library when the host form (window) recieves WM_SCAN_TIMEOUT window message.

Syntax public void OnScanDataMsg(
 int nBardata,
 int nBarType
);

Parameters

nBardata

Specifies the barcode data read.

nBarType

Specifies the barcode type read.

Return Value

None.

This method is called by 2D scanner library when the host form (window) receives WM_SCAN_DATA window message.

Syntax `public void OnScanTimeoutMsg();`

Parameters

None.

Return Value None.

This event fires when the barcode data is read.

Syntax `public event ScannerDataDelegate ScannerDataEvent;`

Remarks An application should add the eventhandler that receives this event.

Example `m_g2ScanCtrl = new G2W_ScannerControl();m_g2ScanCtrl.ScannerDataEvent += new
ScannerDataDelegate(OnImager_Scan_Data);`

This event fires when the barcode reading time-out interval elapsed and the barcode is not read.

Syntax `public event ScannerTimeoutDelegate ScannerTimeoutEvent;`

Remarks

An application should add the eventhandler that receives this event.

Example `m_g2ScanCtrl = new G2W_ScannerControl();m_g2ScanCtrl.ScannerTimeoutEvent += new
ScannerTimeoutDelegate(OnImager_Scan_Timeout);`

This method processes window messages.

Syntaxprotected override void WndProc(
 ref Message msg
);

Parameters *msg*

The Message class object that represents a window message.

Retrun valueNone.

RemarksThis method is overridden in order to process two window messages - WM_SCAN_DATA and WM_SCAN_TIMEOUT.

This method sets the information about authentication and associates.

Syntax public int G2WNet_AssociatedSsid_DATA(
 string wszWepKe,
 string wszBssid,
 int infrastructuremode,
 int authenticationmode,
 int privacy,
 int eaptype,
 int webkeytype
);

wszWepKey

A string that specifies a web key string.

wszBssid

A string that specifies SSID string.

infrastructuremode

Specifies the infrastructure mode as defined in the NDIS_802_11_NETWORK_INFRASTRUCTURE enumeration.

authenticationmode

Specifies the 802.11 authentication mode value.

privacy

Specifies a WEP encryption requirement.

eaptype

Specifies the EAP extension type to be used.

webkeytype

Specifies a type of WEB key value.

Return Value Returns ERROR_NONE if successful.

This method deletes all bssid list information queried.

Syntax `public int G2W_DelBSListcf();`

Parameters None.

Return Value

Returns ERROR_NONE if successful.

This method retrieves the names of adapters currently instantiated by NDIS.

Syntax `public int G2W_Get_AdapterName(
 char[] szAdapter
);`

Parameters *szAdapter*

A buffer that contains adapter name.

Return Value Returns ERROR_NONE if successful.

This method retrieves the MAC address of the associated access point.

Syntax `public int G2W_Get_Bssid(
 byte[] uszBssid
);`

Parameters *uszBssid*

A buffer that contains the MAC address of the associated access point.

Return Value Returns ERROR_NONE if successful.

This method retrieves the configuration of a wifi radio.

Syntax public int G2W_Get_Configuration(
 int nSelect
);

Parameters nSelect

Specifies the index of configuration item. The valid values are one of the FN_CONFIG_ATIM , FN_CONFIG_CHANNEL and FN_CONFIG_BEACON.

Return Value Returns the value of configuration item specified by nSelect.

This method retrieves the connection status.

Syntax public int G2W_Get_Connect_Status();

Parameters

None.

Return Value

Returns connection state value. The following table shows the valid values.

State	Value
MEDIA_STATE_CONNECTED	0
MEDIA_STATE_DISCONNECTED	1
MEDIA_STATE_UNKNOWN	-1

Remarks

This function calls DeviceIoControl function with IOCTL_NDISUIO_NIC_STATISTICS and retrieves the pointer to NIC_STATISTICS structure internally. And returns the value of MediaState member of the structure. See the *Network Drivers sections of Windows Mobile 6.5 Documentation* for more information.

This method retrieves the maximum speed of the NIC in kbps.

Syntax public int G2W_Get_LinkSpeed();

Parameters None.

Return Value Returns ERROR_NONE if successful.

Remarks The unit of measurement is 100 bps, so a value of 100,000 represents a hardware bit rate of 10 Mbps.

This method retrieves the MAC address of the NIC.

Syntax int G2W_Get_NICMACAddress(
 byte[] usbBssid
);

Parameters

usbBssid

A buffer that contains the MAC address of the NIC.

Return Value Returns ERROR_NONE if successful.

This method retrieves the signal strength of wlan.

Syntax public int G2W_Get_RssiValue();

Parameters None.

Return Value The signal strength value. The range of the value is -200 to -10.

This method retrieves the SSID name with which current NIC is associated.

Syntax int G2W_Get_SsidName(
char[] szSsid
);

Parameters *szSsid*

A buffer that contains a ssid name.

Return Value Returns ERROR_NONE if successful.

This method inquiries power status of a wlan.

Syntax public int G2W_Get_WlanPower_Status();

Parameters

None.

Return Value Returns 1 if the wlan is on or 0 if the wlan is off.

This method resets preferred networks.

Syntax public int G2W_ResetPreferred();

Parameters None.

Return Value

Returns ERROR_NONE if successful or an error number if an error occurs

This method retrieves a list containing all basic service set identifiers(BSSIDs) and their attributes.

Syntax int G2W_Set_BssidList(
 out G2W_WLAN_BSSID_LIST[] wlist,
 out int nCount
);

Parameters *wlist*

A [G2W_WLAN_BSSID_LIST](#) structure that contains bssid list.

nCnt

Contains the number of list item.

Return Value Returns ERROR_NONE if successful.

This method turns on / off a wlan.

Syntax public bool G2W_Set_WlanPower_OnNoff(
 bool onNoff
);

Parameters *onNoff*

A Boolean that specifies whether the wlan is on.

Return Value Returns true if successful or false if an error occurs.

This method captures a still image into a file.

Syntax public string G2W_Capture(
 string FileFullName
);

```
public void G2W_Capture(  
    string FileFullName,  
    StringBuilder info  
);
```

Parameters *FileFullName*

A string that specifies the image file path. If this is null, the image file name is set to date string such as "yyyymmddhhmmss.jpg, and is saved under "\\My Documents\\My Pictures\\".

info

A StringBuilder object that contains the image file path saved.

Return Value A string that contains the image file path saved.

This method closes camera and clean up any resource used.

Syntax public static void G2W_Close();

Parameters None.

Return Value None.

This method turns on or off camera flash.

Syntax public bool G2W_Flash_On(
 bool on
);

Parameters on A boolean that specifies whether the camera flash is on.

Return Value Returns true if successful or false if an error occurs.

This method retrieves the index that represents the brightness.

Syntax public int G2W_Get_Brightness();

Parameters None.

Return Value

Returns the index of the brightness. The range of the index is 0 to 6.

This method retrieves the camera dll version string.

Syntax public string G2W_Get_Info();

Parameters None.

Return Value Returns a string that contains the version string.

This method retrieves the index that represents the image quality.

Syntax `public int G2W_Get_Quality();`

Parameters None.

Return Value Returns the index of the quality. The range of the index is 0 to 2.

This method retrieves the index that represents the current resolution.

Syntax `public int G2W_Get_Resolution(
 int PinType
);`

Parameters *PinType*

Specifies the value that represents the type of pin. The type of pin is still if this value is set to 1, the type of pin is capture if this value is set to 2.

Return Value Returns the index of the resolution. The range of the index is 0 to 6.

This method retrieves the index that represents the white balance.

Syntax `public int G2W_Get_WhiteBalance();`

Parameters

None.

Return Value Returns the index of the white balance. The range of the index is 0 to 4.

This method opens and initialize camera device.

Syntax `public bool G2W_Open(
 IntPtr hMainWnd,
 CAMERA_MODE cameramode,
 VIDEO_TYPE videotype
);`

Parameters

hMainWnd

Handle to a window that displays preview and receives event notificaitons.

cameramode

[in] A [CAMERA_MODE](#) enumeration type that specifies the camera mode.

videotype [in] A [VIDEO_TYPE](#) enumeration type that specifies the video mode.

Return Value Returns true if successful or false if an error occurs.

This method starts preview.

Syntax `public bool G2W_Preview_Start();`

Parameters None.

Return Value

Returns true if successful or false if an error occurs.

This method stops preview.

Syntax `public bool G2W_Preview_Stop();`

Parameters None.

Return Value Returns `true` if successful or `false` if an error occurs.

This method sets the index that represents the brightness.

Syntax `public bool G2W_Set_Brightness(
 long value
);`

Parameters *value*

Specifies the index that represents the brightness. The range of the index is 0 to 6.

Return Value Returns `true` if successful or `false` if an error occurs.

This method sets the index that represents the image quality.

Syntax `public void G2W_Set_Quality(
 int value
);`

Parameters

value

Specifies the index that represents the quality. The range of the index is 0 to 2.

Return Value None.

This method sets the index that represents the resolution.

Syntax public bool G2W_SetResolution(
 int PinType,
 int nResolution
);

Parameters

PinType Specifies the value that represents the type of pin. The type of pin is still if this value is set to 1, the type of pin is capture if this value is set to 2.

nResolution

Specifies the index that represents the resolution. The range of the index if 0 to 6.

Return Value Returns true if successful or false if an error occurs.

This method sets the index that represents the white balance.

Syntax public bool G2W_Set_WhiteBalance(
 long value
);

Parameters

value

Specifies the index that represents the white balance. The range of the index is 0 to 4.

Return Value Returns true if successful or false if an error occurs.

This method triggers the auto focus.

Syntax `public bool G2W_SetAutoFocus();`

Parameters None.

Return Value Returns true if successful or false if an error occurs

Remarks An function call on the platform without auto-focus ability is ignored.

This method sets the position of the video window in device coordinates.

Syntax `public bool G2W_SetPosition(
 long left,
 long top,
 long width,
 long height
);`

Parameters

left

Specifies the left side of the window.

top

Specifies the top side of the window.

width

Specifies the width of the window.

height

Specifies the heght of the window.

Return Value Returns true if successful or false if an error occurs.

This method assigns a window to receive a particular hardware button's key-press messages.

Syntax `public bool G2W_SHSetAppKey(
 byte bVk,
 IntPtr hMainWnd
);`

Parameters

bVk

Key code for the hardware button.

hMainWnd

Handle to the window that will receive the messages.

Return Value Returns `true` if successful or `false` if an error occurs.

This method starts capturing video.

Syntax

```
public bool G2W_Video_Start(  
    string FileFullName  
);
```

Parameters *FileFullName*

A string that specifies the video file path. If this is null, the video file name is set to date string such as "yyyymmddhhmmss.asf", is saved under "\\My Documents\\My Pictures\\".

Return Value Returns `true` if successful or `false` if an error occurs.

This method stops capturing video.

Syntax `public bool G2W_Video_Stop();`

Parameters None.

Return Value Returns true if successful or false if an error occurs.

This method starts the thread for monitoring power state of device.

Syntax `public void Start();`

Parameters None.

Return Value None.

This method stops the thread for monitoring power state of device.

Syntax `public void Stop();`

Parameters None.

Return Value None.