

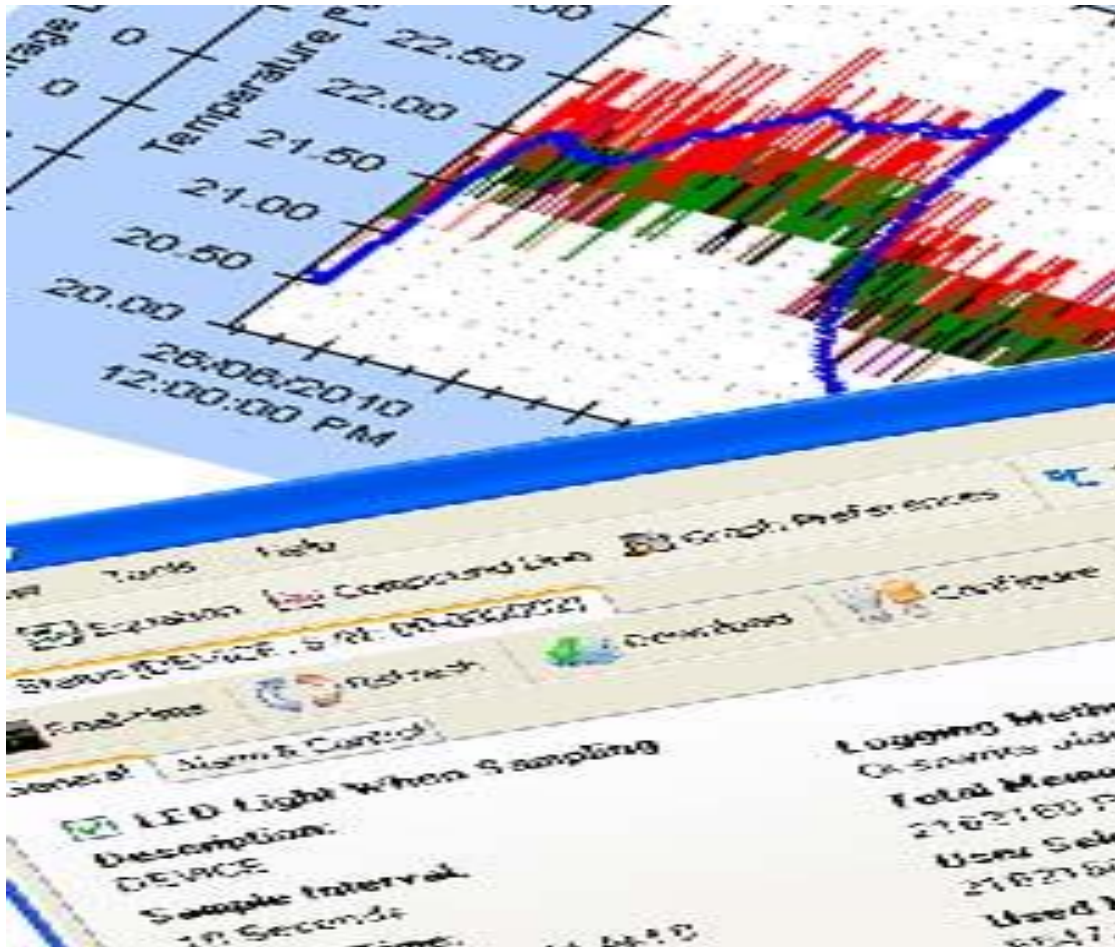


404 – 1688 152nd Street  
Surrey, BC  
Canada, V4A 4N2  
Phone: 604.424.9092

Toll Free: 1.877.352.9158  
Fax: 1.877.453.0658  
Email: info@microedgeinstruments.com  
Web: www.microedgeinstruments.com

# SiteView SDK

*Software Development Kit*



## User's Manual

Microedge Instruments Inc.  
404 – 1688 152nd Street Surrey, BC  
Canada, V4A 4N2  
Toll Free: 1.877.352.9158  
www.microedgeinstruments.com

## **ABOUT THIS MANUAL**

This manual describes how to get started using SiteView SDK (Software Development Kit) to develop application programs for communications with data loggers from Microedge Instruments Inc (MEI).

Copyright © 2010-2016, Microedge Instruments Inc.

### **Limits of Liability and Disclaimer of Warranty:**

---

The manual contained in this document is furnished for informational use only and is subject to change without notice.

Microedge Instruments Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

### **Copyright and Trademarks**

---

This manual contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without prior written consent of Microedge Instruments Inc.

Microedge Instruments Inc., Site-Log, SiteView, SiteView SDK are the trademarks of Microedge Instruments Inc.

All other trademarks and registered trademarks are the property of their respective owners.

### **End User License Agreement**

---

End User License Agreement is available for download on our website.

PLEASE DOWNLOAD AND READ THE SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE DOWNLOADING OR USING THE SOFTWARE.

#### **Microedge Instruments Inc.**

404 – 1688 152nd Street

Surrey, BC Canada, V4A 4N2

Toll Free: 1.877.352.9158

Web-Site: [www.microedgeinstruments.com](http://www.microedgeinstruments.com)

Email: [info@microedgeinstruments.com](mailto:info@microedgeinstruments.com)

Revision 1.4, 2018-10 for SiteView SDK 2.1

# Table of Contents

<b>ABOUT THIS MANUAL</b> .....	<b>2</b>
LIMITS OF LIABILITY AND DISCLAIMER OF WARRANTY: .....	2
COPYRIGHT AND TRADEMARKS .....	2
END USER LICENSE AGREEMENT .....	2
<b>CHAPTER 1 - INTRODUCTION</b> .....	<b>5</b>
INTENDED AUDIENCE.....	5
WHAT YOU CAN LEARN FROM THIS MANUAL .....	5
SYSTEM REQUIREMENTS .....	5
<b>CHAPTER 2 - INSTALL SITEVIEW SDK</b> .....	<b>6</b>
INSTALL SITEVIEW SDK .....	6
CONNECT DATA LOGGER.....	10
ACTIVATE SITEVIEW SDK .....	14
<b>CHAPTER 3 – ENTITIES AND FUNCTIONS</b> .....	<b>15</b>
SITEVIEW SDK SYSTEM ARCHITECTURE .....	15
CUSB.....	15
<i>Methods</i> .....	15
FindDataLoggers.....	15
CUSBDEVICE SERVER.....	16
<i>Methods</i> .....	16
FindDataLoggers.....	16
FindDataLoggers.....	17
GetDeviceServerEntries.....	17
GetDeviceServerEntry .....	18
DeleteDeviceServerEntry .....	18
SetDeviceServerEntry .....	18
CSERIALPORT .....	18
<i>Methods</i> .....	19
FindDataLoggers.....	19
FindDataLogger .....	19
FindDataLogger .....	20
CSERIALPORTDEVICE SERVER .....	20
<i>Methods</i> .....	20
FindDataLogger .....	20
FindDataLogger .....	21
GetDeviceServerEntries.....	21
GetDeviceServerEntry .....	22
DeleteDeviceServerEntry .....	22
SetDeviceServerEntry .....	22
CDATA LOGGER.....	23
<i>Methods</i> .....	23
UpdateStatus .....	23

GetCurrentMeasurements .....	23
StartNewSession .....	24
StartNewSession .....	24
StartDownloadData.....	25
DownloadData .....	27
DownloadData .....	28
<i>Properties</i> .....	29
CDATALOGGERCHANNEL .....	30
<i>Methods</i> .....	30
GetAvailableChannelTypes .....	30
SetChannelType.....	30
GetAvailableEquations .....	31
SetEquation .....	31
<i>Properties</i> .....	31
CSITEVIEWSDKSETTINGS .....	32
<i>Functions</i> .....	32
GetAvailableChannelTypes .....	32
DisplayUnitEditor .....	32
DisplayUnitEditor .....	33
DisplayPlotPreferences .....	33
DisplayPlotPreferences .....	33
DisplayEquationEditor.....	33
DisplayEquationEditor.....	34
About.....	34
CSITEVIEWSDKPLOTCTRL .....	34
<i>Methods</i> .....	34
OpenFiles .....	34
OpenFile.....	35
CDEVICESERVERENTRYPROPERTY .....	35
<i>Properties</i> .....	35
ENUMERATIONS .....	35
<b>CHAPTER 4 – MORE EXAMPLES .....</b>	<b>38</b>
VISUAL BASIC 6 USER .....	38
<i>Reference SiteView SDK</i> .....	38
<i>Find Data Loggers in USB Ports</i> .....	39
<i>Open SiteView File</i> .....	40
<i>Display About Dialog</i> .....	40

# CHAPTER 1 - INTRODUCTION

Congratulations on using SiteView SDK - developer' software development kit (SDK) working with MEI data loggers for device configuration, data downloading, plotting and analyzing.

Note that nothing in this SDK pertains to SiteView software. If you purchased SiteView software, then you already have everything you need to interact with data loggers (ignore this manual and refer back to SiteView User's Manual). You do not need to install this software.

If, however you intend to write your own programs to work with MEI data loggers, then this manual contains the information you need.

## Intended Audience

---

This document is intended for engineers, scientists, technicians, OEMs, system integrators, or others responsible for developing application programs that can interact with .Net Assembly in order to perform data acquisition operations with MEI data loggers.

## What You Can Learn From This Manual

---

This manual provides installation instructions for Windows XP, Vista, Windows 7, 8 and 10; summarizes the functionalities provided by SiteView SDK; and describes how to use the functions to develop your own data logging program.

Using this manual, you should be able to successfully install SiteView SDK and get started writing an application program for data acquisition and data plotting.

## System Requirements

---

### Computer:

- CPU: 1.0 GHZ or above
- Memory: 256M or above
- Port: 1 USB port or 1 COM port
- Hard Drive: 1GB or above

### Operating System:

- Window XP with SP2 or later, Window Vista, Window 7, 8, 10

### Developing Tools:

- Any language that has the ability to instantiate .Net Assembly directly or indirectly.

## CHAPTER 2 - INSTALL SITEVIEW SDK

**NOTE:** Before the installation, please make sure the product key is available.

### Install SiteView SDK

1. Configuration for Windows 8, 10 User

For detailed configuration please visit:

<http://www.microedgeinstruments.com/Win8Configurations.pdf>

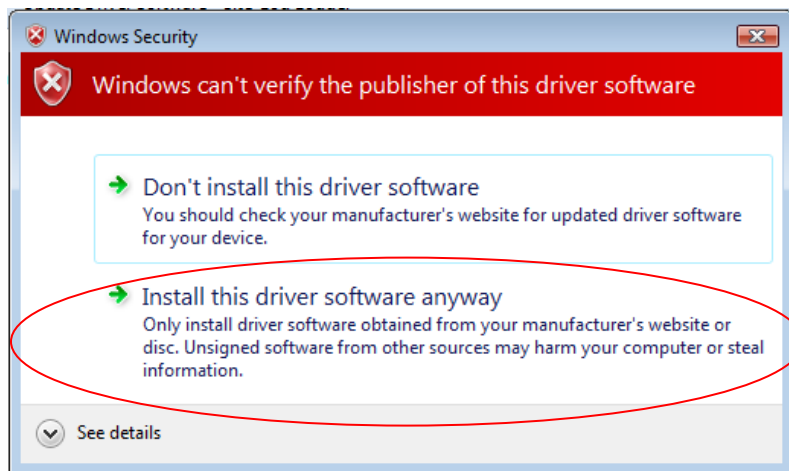
and

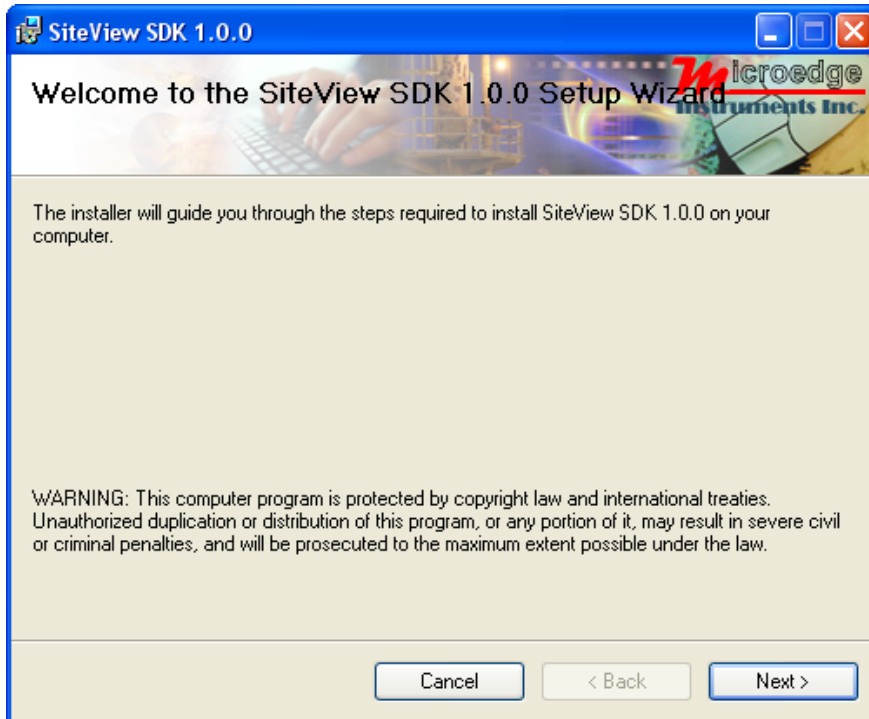
<http://www.microedgeinstruments.com/Win10Configurations.pdf>

2. Install SiteView SDK and USB Drivers

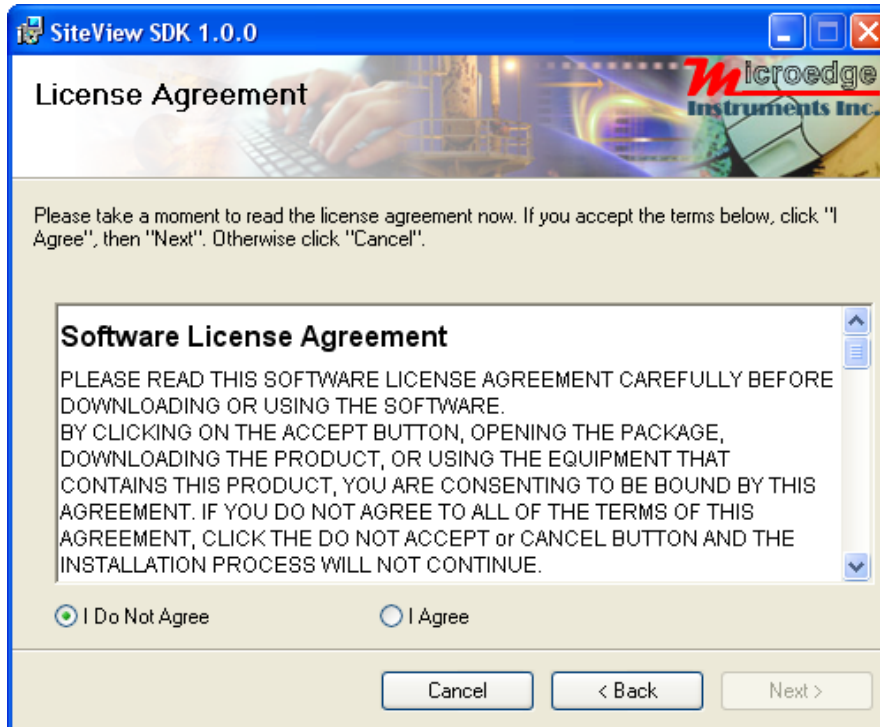
Insert the included CD to the CD Drive. The installation should start to run automatically. Follow the on-screen instructions to complete the installation.

Depending on the operating system, you may see the dialog similar to the one below displayed. Please select “**Continue Anyway**” or “**Install this driver software anyway**” to allow the software and the driver to be installed.

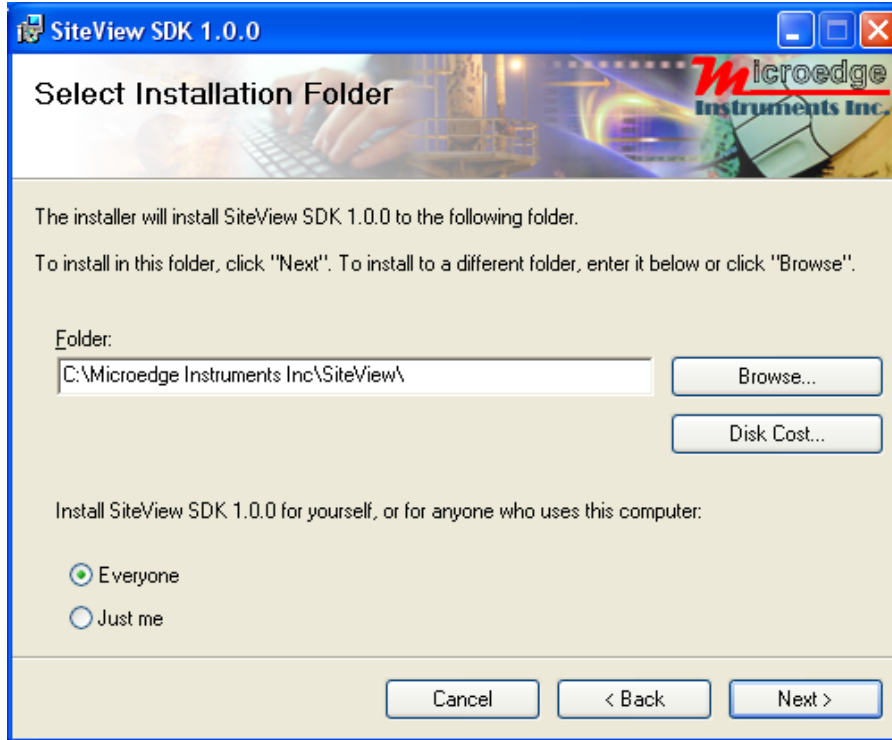




Click “Next >” button to proceed to the next page.



Please read the License Agreement carefully. If you accept the terms click “I Agree”, then click “Next >” button. Otherwise click “Cancel” to cancel the installation.



In this dialog select a destination folder where SiteView SDK will be installed. We recommend you keep the default folder.

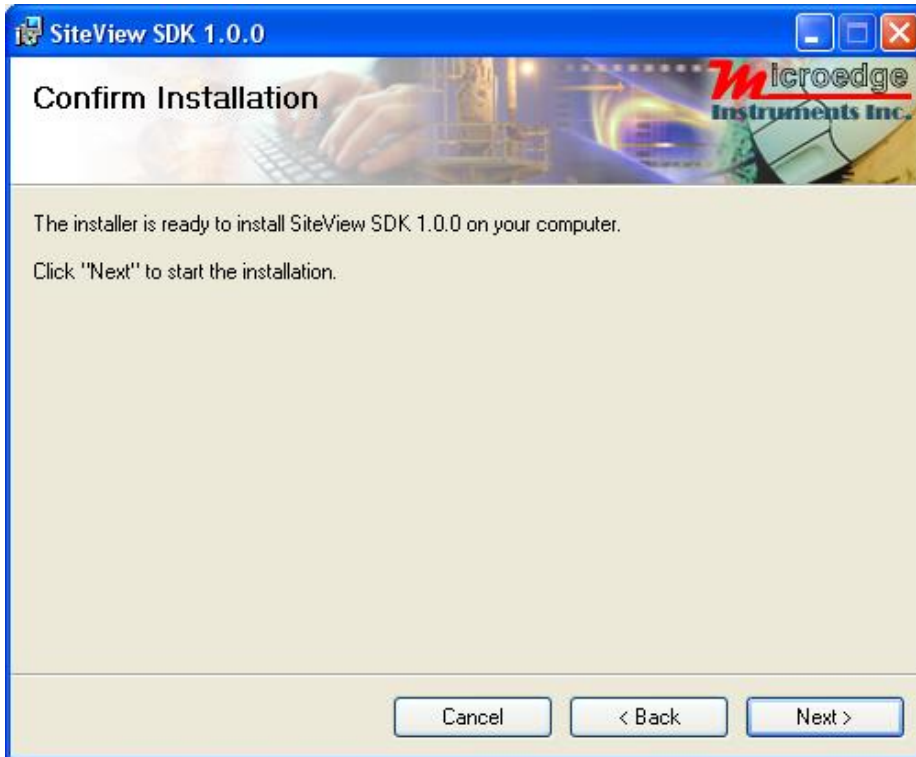
---

**NOTE:** Since SiteView and SiteView SDK share the same settings like units, equations and other application level properties, please make sure they are installed in the same directory.

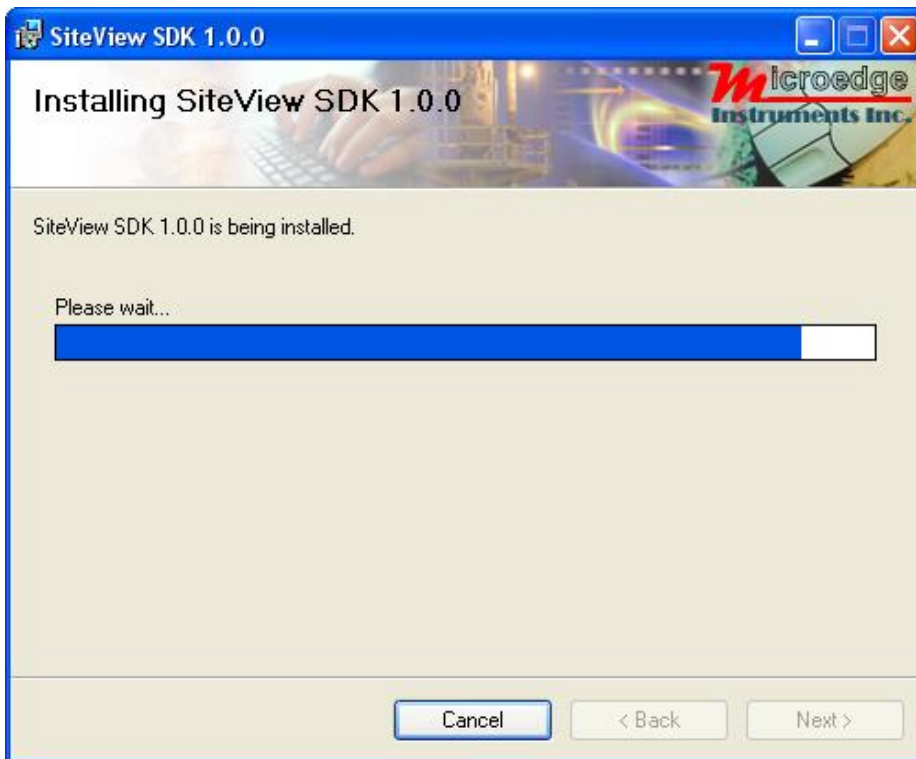
---

Once you are ready, click “Next >” button to proceed to the next page.

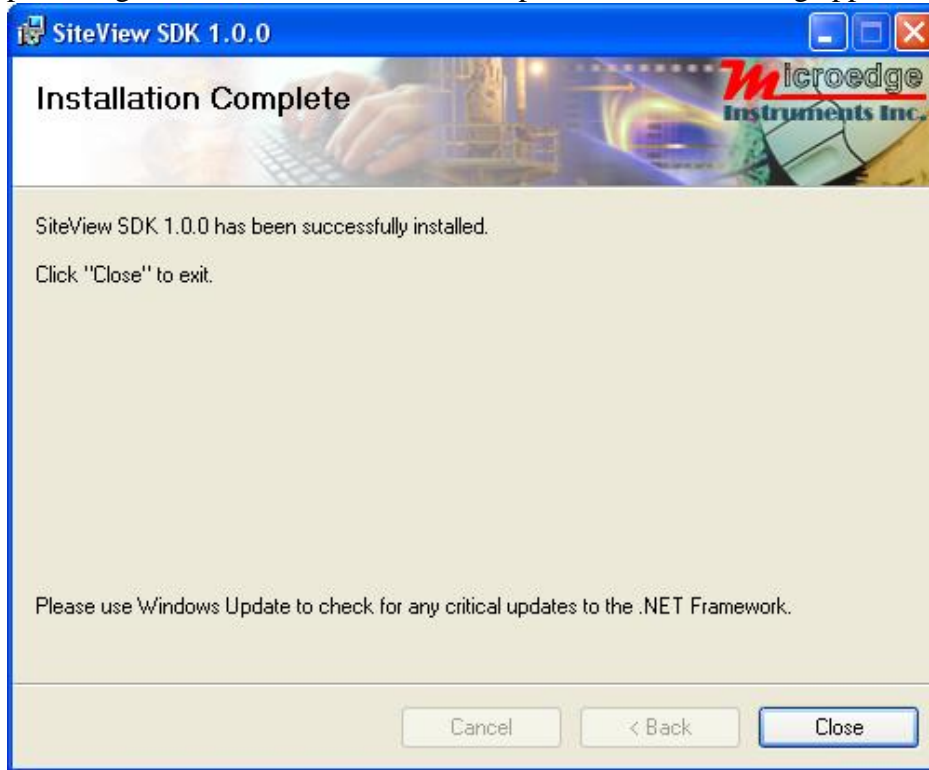




This confirmation page gives you the chance to modify previously selected options. Click “Next >” button to start the installation.



As SiteView SDK is being installed the above dialog shows the installation progress by percentage. Once the installation is complete, the below dialog appears:



Click "Close" button to finish the installation and close the dialog.

## **Connect Data Logger**

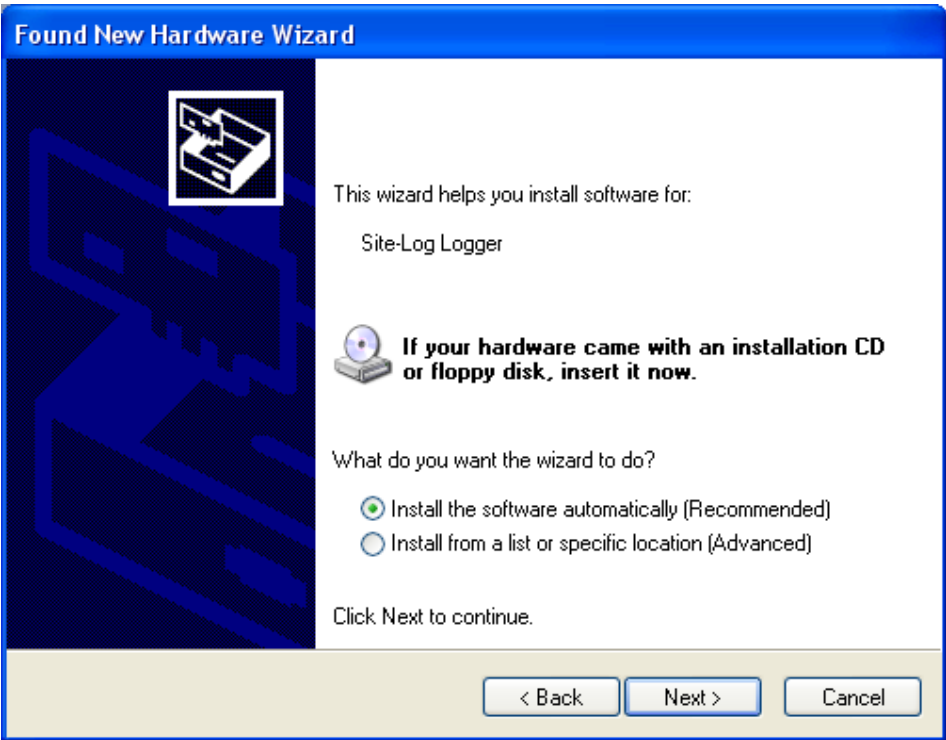
---

Connect the logger to the computer's USB port. Windows Vista and Windows 7 will automatically recognize the data logger.

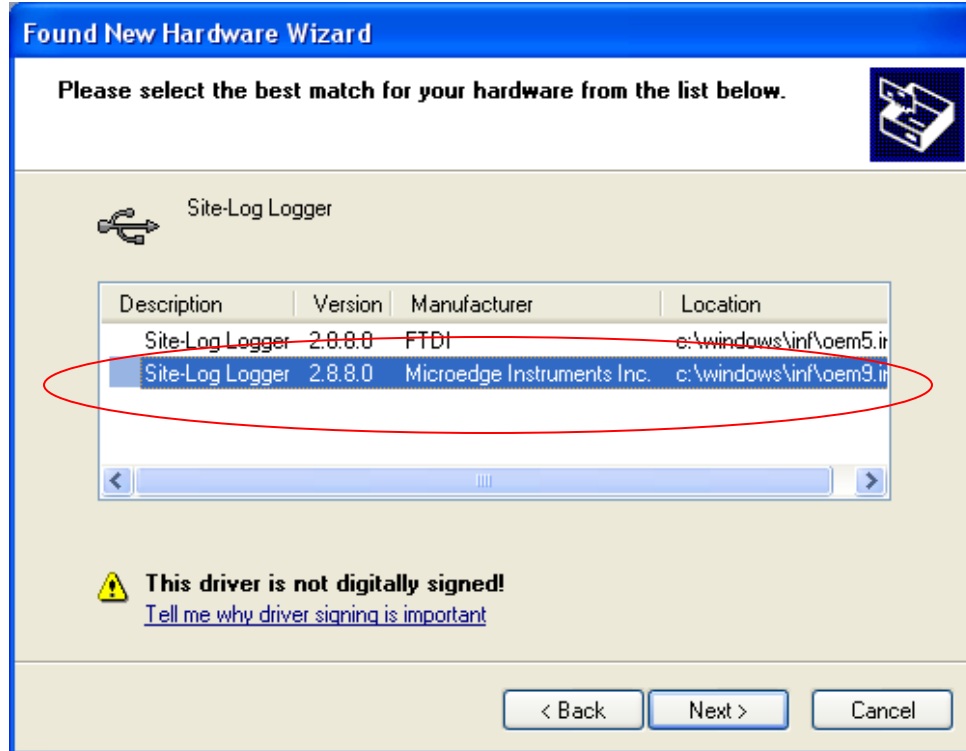
For Windows XP user, the following dialog window will appear:



Select "No, not this time" from options available and then Click "Next >" to proceed with the installation.



Select "Install the software automatically (Recommended)" as shown in the above figure and then click "Next >".

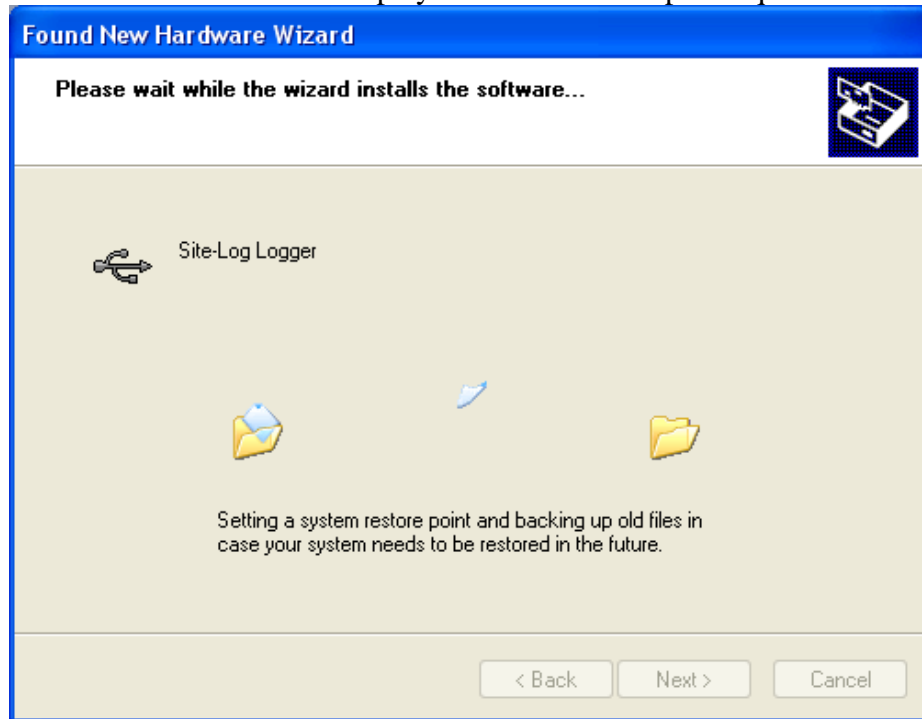


Select the item with Manufacturer of Microedge Instruments Inc and click "Next>" to proceed.

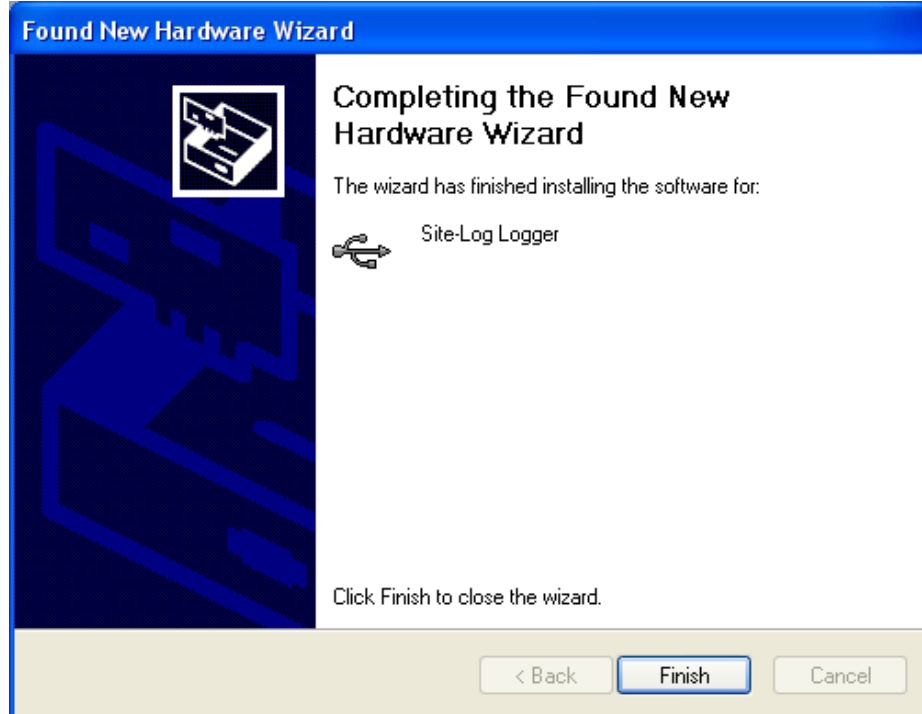
In the following message dialog, click "Continue Anyway" to continue with the installation:



The screen below will be displayed as Windows copies required driver files:



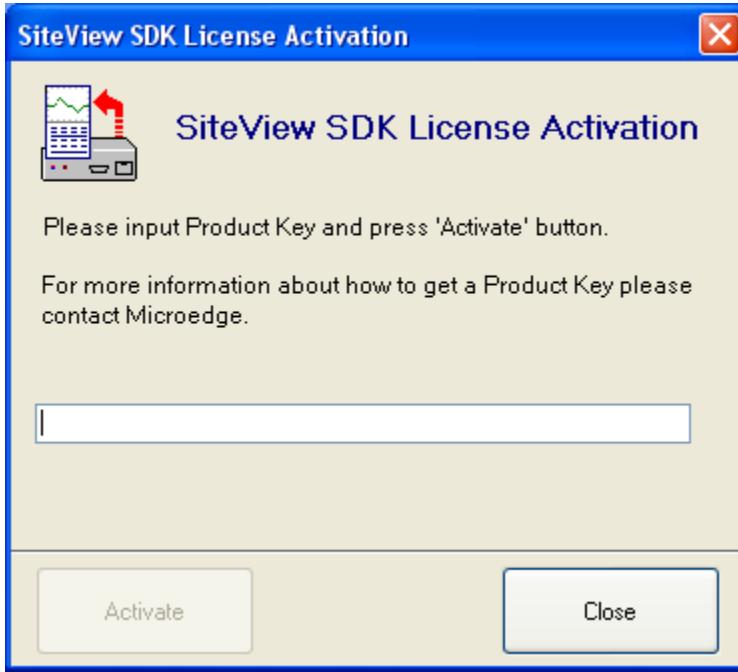
Windows should then display a message indicating the installation was successful:



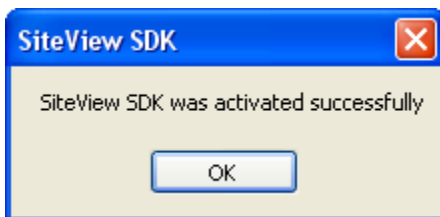
## Activate SiteView SDK

---

At the first time when you instantiate any class of the SDK, you will see below dialog appears for you to activate the SDK:



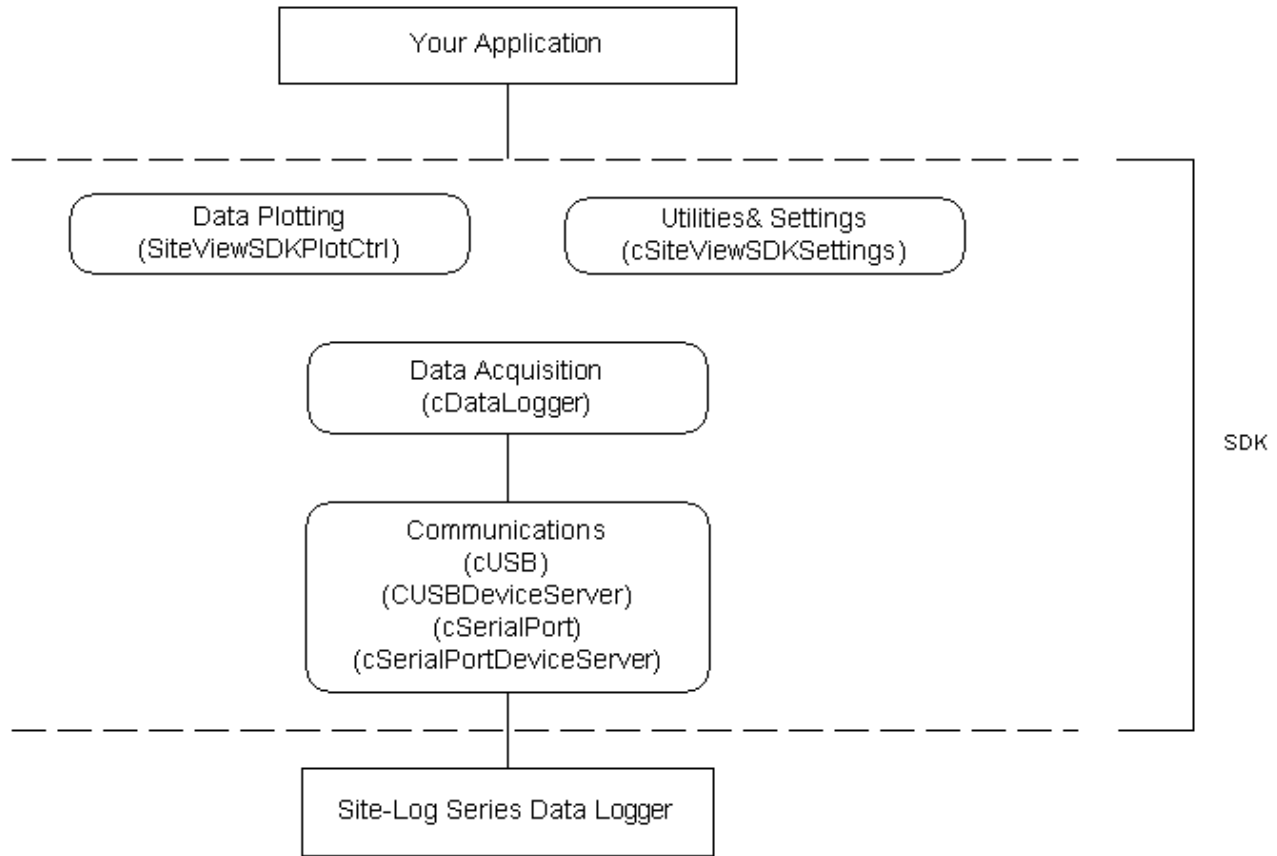
Enter the Product Key, then click **Activate** button. If the Product Key is accepted the following confirmation dialog will appear:



Click **OK** button to finish the activation. From now you can start using SiteView SDK.

# CHAPTER 3 - ENTITIES AND FUNCTIONS

## SiteView SDK System Architecture



### **cUSB**

cUSB class handles the communications with data loggers via USB ports of a computer.

### **Methods**

#### *FindDataLoggers*

```
public cDataLogger[] FindDataLoggers()
```

#### **Usage:**

Find a list of data loggers from the USB ports of the computer.

**Return Values:**

cDataLogger[]

**Arguments:**

Void

**Example(C#):**

```
using SiteViewSDK;

cUSB Conn = new cUSB();
cDataLogger[] Loggers = Conn.FindDataLoggers();

if (Loggers.Length > 0)
    MessageBox.Show(string.Format("Total data loggers found in USB ports: {0}",
    Loggers.Length));
```

## **cUSBDeviceServer**

---

cUSBDeviceServer class handles the communications with data loggers via USB Device Server connections.

### **Methods**

***FindDataLoggers***

public cDataLogger [] FindDataLoggers(cDeviceServerEntryProperty Entry)

**Usage:**

Find a list of data loggers from a given IP address

**Return Values:**

cDataLogger[]

**Arguments:**

cDeviceServerEntryProperty Entry

**Example(C#):**

```
using SiteViewSDK;

cUSBDeviceServer Conn = new cUSBDeviceServer();
cDeviceServerEntryProperty entry = new cDeviceServerEntryProperty();
entry.Description = "example";
entry.IP = "192.168.0.100";
entry.Port = 5678;
```



```
entry.Packet = 5000;
entry.Password = "12345";
entry.Retries = 5;
entry.Timeout = 5000;
```

```
cDataLogger[] Loggers = Conn.FindDataLoggers(entry);
```

```
if (Loggers.Length > 0)
    MessageBox.Show(string.Format("Total data loggers found: {0}", Loggers.Length));
```

### *FindDataLoggers*

```
public cDataLogger[] FindDataLoggers(string IP, int Port, string Description, string Password)
```

#### **Usage:**

Find a list of data loggers from a given IP address.

#### **Return Values:**

cDataLogger[]

#### **Arguments:**

string IP: The IP address of the USB Device Server.

int Port: The port number of the USB Device Server.

string Description: The description of the connection.

string Password: The password to access the USB Device Server.

#### **Example(C#):**

```
using SiteViewSDK;
```

```
cUSBDeviceServer Conn = new cUSBDeviceServer();
cDataLogger[] Loggers = Conn.FindDataLoggers("192.168.0.100", 5678, "Example", "12345");
if (Loggers.Length > 0)
    MessageBox.Show(string.Format("Total data loggers found: {0}", Loggers.Length));
```

### *GetDeviceServerEntries*

```
public cDeviceServerEntryProperty GetDeviceServerEntry(string IP, ushort Port, string Description)
```

#### **Usage:**

Get a list of Device Server entries you have saved in the computer.

#### **Return Values:**

cDeviceServerEntryProperty[]

**Arguments:**

Void

***GetDeviceServerEntry***

`public cDeviceServerEntryProperty GetDeviceServerEntry(string IP, ushort Port, string Description)`

**Usage:**

Get a specific Device Server entry you have saved in the computer.

**Return Values:**

`cDeviceServerEntryProperty`

**Arguments:**

`string` IP: the IP address

`ushort` Port: the port number

`string` Description: the description of the entry

***DeleteDeviceServerEntry***

`public bool DeleteDeviceServerEntry(cDeviceServerEntryProperty entry)`

**Usage:**

Delete a specific Device Server entry you have saved in the computer.

**Return Values:**

`bool`: true if success

**Arguments:**

`cDeviceServerEntryProperty` entry

***SetDeviceServerEntry***

`public bool SetDeviceServerEntry(cDeviceServerEntryProperty entry)`

**Usage:**

Update a specific Device Server entry you have saved in the computer.

**Return Values:**

`bool`: true if success

**Arguments:**

`cDeviceServerEntryProperty` entry

## **cSerialPort**

---

`cSerialPort` class handles the communications with data loggers via serial port connections.

## Methods

### *FindDataLoggers*

```
public cDataLogger[] FindDataLoggers()
```

**Usage:**

Find a list of data loggers from all COM ports of a computer.

**Return Values:**

cDataLogger []

**Arguments:**

Void

**Example(C#):**

```
using SiteViewSDK;  
  
cSerialPort Conn = new cSerialPort();  
cDataLogger[] Loggers = Conn.FindDataLoggers();  
  
if (Loggers.Length > 0)  
    MessageBox.Show(string.Format("Total data loggers found: {0}", Loggers.Length));
```

### *FindDataLogger*

```
public cDataLogger FindDataLogger(string Port)
```

**Usage:**

Find a data loggers from a given COM port of a computer.

**Return Values:**

cDataLogger

**Arguments:**

string Port

**Example(C#):**

```
using SiteViewSDK;  
  
cSerialPort Conn = new cSerialPort();  
cDataLogger Logger = Conn.FindDataLogger("COM1");  
  
if (Logger != null)  
    MessageBox.Show("A data logger was found");
```

### *FindDataLogger*

`public cDataLogger FindDataLogger(string Port, int Packet, int BaudRate)`

**Usage:**

Find a data loggers from a given COM port of a computer.

**Return Values:**

`cDataLogger`

**Arguments:**

`string` Port: the serial port.

`int` Packet: the packet size.

`int` BaudRate: the baud rate (valid values: 115200, 76800, 57600, 38400, 28800, 19200, 14400, 9600,4800, 2400)

**Example(C#):**

```
using SiteViewSDK;
```

```
cSerialPort Conn = new cSerialPort();
```

```
cDataLogger Logger = Conn.FindDataLogger("COM1", 1000, 57600);
```

```
if (Logger != null)
```

```
    MessageBox.Show("A data logger was found");
```

## **cSerialPortDeviceServer**

---

`cSerialPortDeviceServer` class handles the communications with data loggers via Serial Device Server connections.

### **Methods**

#### *FindDataLogger*

`public cDataLogger FindDataLogger(cDeviceServerEntryProperty Entry)`

**Usage:**

Find a data loggers from a given ip address

**Return Values:**

`cDataLogger`

**Arguments:**

`cDeviceServerEntryProperty` Entry

### Example(C#):

```
using SiteViewSDK;

cSerialPortDeviceServer Conn = new cSerialPortDeviceServer();
cDeviceServerEntryProperty entry = new cDeviceServerEntryProperty();
entry.Description = "example";
entry.IP = "192.168.0.100";
entry.Port = 5678;
entry.Packet = 5000;
entry.Retries = 5;
entry.Timeout = 5000;

cDataLogger Logger = Conn.FindDataLogger(entry);

if (Logger != null)
    MessageBox.Show("A data logger was found");
```

### *FindDataLogger*

```
public cDataLogger FindDataLogger(string IP, int Port, string Description)
```

#### Usage:

Find a data loggers from a given IP address.

#### Return Values:

cDataLogger

#### Arguments:

**string** IP: The IP address of the USB Device Server.  
**int** Port: The port number of the USB Device Server.  
**string** Description: The description of the connection.

### Example(C#):

```
using SiteViewSDK;

cSerialPortDeviceServer Conn = new cSerialPortDeviceServer();
cDataLogger Logger = Conn.FindDataLogger("192.168.0.100", 5678, "Example");

if (Logger != null)
    MessageBox.Show("A data logger was found");
```

### *GetDeviceServerEntries*

```
public cDeviceServerEntryProperty [] GetDeviceServerEntries()
```

#### Usage:

Get a list of Device Server entries you have saved in the computer.

**Return Values:**

cDeviceServerEntryProperty[]

**Arguments:**

Void

***GetDeviceServerEntry***

public cDeviceServerEntryProperty GetDeviceServerEntry(string IP, ushort Port, string Description)

**Usage:**

Get a specific Device Server entry you have saved in the computer.

**Return Values:**

cDeviceServerEntryProperty

**Arguments:**

string IP: the IP address

ushort Port: the port number

string Description: the description of the entry

***DeleteDeviceServerEntry***

public bool DeleteDeviceServerEntry(cDeviceServerEntryProperty entry)

**Usage:**

Delete a specific Device Server entry you have saved in the computer.

**Return Values:**

bool: true if success

**Arguments:**

cDeviceServerEntryProperty entry

***SetDeviceServerEntry***

public bool SetDeviceServerEntry(cDeviceServerEntryProperty entry)

**Usage:**

Update a specific Device Server entry you have saved in the computer.

**Return Values:**

bool: true if success

**Arguments:**

cDeviceServerEntryProperty entry

## **cDataLogger**

---

cDataLogger class represents a data logger entity. You manage the data logger hardware via the methods and properties of this class.

### **Methods**

#### *UpdateStatus*

`public bool UpdateStatus()`

**Usage:**

Retrieve the new status from the data logger hardware.

**Return Values:**

`bool`: true if success

**Arguments:**

Void

**Example(C#):**

```
using SiteViewSDK;

cUSB Conn = new cUSB();
cDataLogger[] Loggers = Conn.FindDataLoggers();
if (Loggers.Length > 0)
{
    if(Loggers[0].UpdateStatus())
        MessageBox.Show("The logger status is updated.");
}
}
```

#### *GetCurrentMeasurements*

`public double[] GetCurrentMeasurements()`

**Usage:**

Retrieve the current measurements from the data logger hardware

**Return Values:**

`double[]`: each measurement for each enabled channel

**Arguments:**

Void

**Example(C#):**

```

using SiteViewSDK;

cUSB Conn = new cUSB();
cDataLogger[] Loggers = Conn.FindDataLoggers();
if (Loggers.Length > 0)
{
    double[] values = Loggers[0].GetCurrentMeasurements();
    if (values != null && values.Length > 0)
        MessageBox.Show(string.Format("Total measurements: {0}", values.Length));
}

```

### *StartNewSession*

```
public bool StartNewSession(DateTime TimeToStart)
```

#### **Usage:**

Configure the data logger hardware to start a new session from the given start time.

#### **Return Values:**

**bool:** true if success

#### **Arguments:**

**DateTime** TimeToStart: the specific time to start recording.

#### **Example(C#):**

```

using SiteViewSDK;

cUSB Conn = new cUSB();
cDataLogger[] Loggers = Conn.FindDataLoggers();
if (Loggers.Length > 0)
{
    if(Loggers[0].StartNewSession(DateTime.Now))
        MessageBox.Show("The logger started a new logging session");
}

```

### *StartNewSession*

```
public bool StartNewSession(DateTime TimeToStart, DateTime TimeToEnd)
```

#### **Usage:**

Configure the data logger hardware to start a new session from the given start time.



**Return Values:**

**bool:** true if success

**Arguments:**

**DateTime** TimeToStart: the specific time to start recording.

**DateTime** TimeToEnd: the specific time to end recording if the logging mode is not FIFO. If the logging mode is FIFO, when the memory is full both start and end time will move forward.

***StartDownloadData***

```
public bool StartDownloadData(string Filename, DateTime Start, DateTime End,
enumSaveFileType SaveFileType)
```

or

```
public bool StartDownloadData(string Filename, enumSaveFileType SaveFileType)
```

**Usage:**

Start to download the data blocks from the data logger. This function starts a new thread for downloading. When the download is finished, the data is saved in the give file. Please use the following two events to track the process of the download:

OnLoggerDownloading: Fired during the download with percentage of download

OnLoggerDownloadFinished: Fired after the download with success or failure status

**Return Values:**

**bool:** true if success

**Arguments:**

**string** Filename:the filename to save the data.

**DateTime** Start: the time when the data begins.

**DateTime** End: the time when the data ends.

**enumSaveFileType** SaveFileType: specify how the file will be saved: svf or text...

**Example(C#):**

```
using SiteViewSDK;
```

```
cUSB Conn = new cUSB();
```

```
cDataLogger[] m_Loggers;
```

....

```
m_Loggers = Conn.FindDataLoggers();
```

```
if (m_Loggers.Length > 0)
```

```
{
```

```
    string file = "c:\\temp.txt";
```

```
    m_Loggers[0].OnLoggerDownloading += new
```

```
    LoggerDownloadingHandler(frmUSB_OnLoggerDownloading);
```

```

    m_Loggers[0].OnLoggerDownloadFinished += new
    LoggerDownloadingHandler(frmUSB_OnLoggerDownloadFinished);

    DateTime start = new DateTime(2011, 10, 10, 1, 0, 0);
    DateTime end = new DateTime(2011, 10, 10, 2, 0, 0);
    m_Loggers[0].StartDownloadData(file,
    SiteView.enumSaveFileType.CSV_WITH_STATUS);

}

....

delegate void frmUSBOnLoggerDownloading(int Percent);
void frmUSB_OnLoggerDownloading(int Percent)
{
    if (InvokeRequired)
        this.BeginInvoke(new
    frmUSBOnLoggerDownloading(frmUSB_OnLoggerDownloading), new object[]
    { Percent });
    else
    {
        if (Percent != 100)
            this.Text = string.Format("Downloading data: {0}%", Percent);
        else
        {
            m_Loggers[0].OnLoggerDownloading -= new
            LoggerDownloadingHandler(frmUSB_OnLoggerDownloading);

            this.Text = "USB Demo";
            MessageBox.Show("Download finished");

        }
    }
}

}

delegate void frmUSBOnLoggerDownloadFinished(bool Success);
void frmUSB_OnLoggerDownloadFinished(bool Success)
{
    if (InvokeRequired)
        this.BeginInvoke(new

```

```

frmUSBOnLoggerDownloadFinished(frmUSB_OnLoggerDownloadFinished), new
object[] { Success });
else
{
    if (Success)
    {
        this.Text = "Download finished";
        MessageBox.Show("Download finished");
    }
    else
    {
        this.Text = "Download failed";
        MessageBox.Show("Download failed");
    }
    m_Loggers[0].OnLoggerDownloadFinished -= new
    LoggerDownloadingHandler(frmUSB_OnLoggerDownloadFinished);
}
}
}
}

```

### ***DownloadData***

**public bool** DownloadData(**string** Filename, **DateTime** Start, **DateTime** End, **enumSaveFileType** SaveFileType)

#### **Usage:**

Download the data blocks from the data logger and wait until it finished.

#### **Return Values:**

**bool:** true if success

#### **Arguments:**

**string** Filename: the filename to save the data.

**DateTime** Start: the time when the data begins.

**DateTime** End: the time when the data ends.

**enumSaveFileType** SaveFileType: specify how the file will be saved: svf or text...

### Example(C#):

```
using SiteViewSDK;

cUSB Conn = new cUSB();
cDataLogger [] m_Loggers;

....

m_Loggers = Conn.FindDataLoggers();
if (m_Loggers.Length > 0)
{
    string file = "c:\\temp.txt";
    DateTime start = new DateTime(2011, 10, 10, 1, 0, 0);
    DateTime end = new DateTime(2011, 10, 10, 2, 0, 0);
    if(m_Loggers[0].DownloadData(file,
        SiteView.enumSaveFileType.CSV_WITH_STATUS))
        MessageBox.Show("Download finished");
}
}
```

### *DownloadData*

```
public bool DownloadData(string Filename, enumSaveFileType SaveFileType)
```

#### **Usage:**

Download the whole data from the data logger and wait until it finished

#### **Return Values:**

**bool:** true if success

#### **Arguments:**

**string** Filename: the filename to save the data.

**enumSaveFileType** SaveFileType: specify how the file will be saved: svf or text...

### *ConfigureWifiLogger\_WifiSettings*

```
public bool ConfigureWifiLogger_WifiSettings ()
```

#### **Usage:**

For PRECISE-LOG data logger, it is used to save WIFI related settings. Those settings were changed with related properties like: WifiLogger\_WifiMode, WifiLogger\_WifiSta\_SSID etc.

#### **Return Values:**

**bool:** true if success

## Properties

Property	Type	Description
DataStartTime	DateTime	Get the start time of the downloaded data.
DataEndTime	DateTime	Get the end time of the downloaded data.
TotalCapacityReadings	Integer	Get the total data in readings that the data logger can record.
TotalReadings	Integer	Get the total data in readings for the current logging session.
DataFIFO	Boolean	Get if the data is full and the new data is overwriting the oldest data.
FIFOMode	Boolean	Get/Set if the data logger is set to overwrite the oldest data by new data when the user configurable data is full.
LEDOn	Boolean	Get/Set if the on-board LED will be on when the data logger is sampling.
LoggerModel	String	Get the model of the data logger.
LoggerID	String	Get the logger identification.
LoggerSerialNumber	String	Get the logger serial number.
Firmware	String	Get the logger firmware version.
Description	String	Get/Set the logger description. Max 30 characters.
SamplingInterval	Unsigned Integer In millisecond	Get/Set the sampling interval for the new logging session.
Channels	cDataLoggerChannel []	Get the list of the data logger channels. See cDataLoggerChannel class for details
DownloadPercent	Integer	Get the current download percentage (0 to 100%)
DeviceType	enumTypeOfDevice	Get the device type
<b>Wifi related settings for PRECISE-LOG data logger</b>		
WifiLogger_WifiMode	enumWifiMode	Disabled: the wifi function is

		disabled. Server: the logger will act as a server
WifiLogger_IsConnectedToWifi	Boolean	True: If the logger is connected to the access point
WifiLogger_WifiSta_SSID	String	The Access Point's name (ID)
WifiLogger_WifiSta_Password	String	The password of the AP
WifiLogger_WifiSta_Authentication	enumWifiAuth	Authentication method of AP
WifiLogger_WifiSta_Encryption	enumWifiEncry	Encryption method of AP
WifiLogger_Password	String	The password to access the logger
WifiLogger_Wifi_IPPort	Integer	The IP port of its network (1 – 65535)

## **cDataLoggerChannel**

---

cDataLoggerChannel class represents a data logger channel entity. You manage the data logger hardware via the methods and properties of this class.

### **Methods**

#### *GetAvailableChannelTypes*

```
public string [] GetAvailableChannelTypes()
```

#### **Usage:**

Get the available channel types the current channel contains. A list of Channel Types for voltage channel can be: 20V, 10V, 5V, 2V... If the channel is user configurable, you can change the channel type for different input range.

#### **Return Values:**

`string []`: a list of channel type strings

#### **Arguments:**

Void

#### *SetChannelType*

```
public bool SetChannelType(string chtype)
```

#### **Usage:**

Change the channel type of current channel . A list of Channel Types for voltage channel can be: 20V, 10V, 5V, 2V... If the channel is user configurable, you can change the channel type for different input range.

**Return Values:**

**bool:** true indicate the change type changed successfully.

**Arguments:**

**string** chtype: the new channel type string.

***GetAvailableEquations***

**public string[]** GetAvailableEquations()

**Usage:**

Change the channel type of current channel . A list of Channel Types for voltage channel can be: 20V, 10V, 5V, 2V... If the channel is user configurable, you can change the channel type for different input range.

**Return Values:**

**string[]:** A list of available equations for this channel.

**Arguments:**

void

***SetEquation***

**public bool** SetEquation(**string** EquationName)

**Usage:**

Change the channel equation.

**Return Values:**

**bool:** true indicate the channel has changed the equation successfully.

**Arguments:**

**string** EquationName: the new equation name.

**Properties**

<b>Property</b>	<b>Type</b>	<b>Description</b>
ChannelType	String	Get the channel type.
IsCustomChannle	Boolean	Get if the current channel is a custom channel.
Description	String	Get/Set the description of the logger. Max: 30 characters.
CaliZero	Short	Get/Set the calibration Zero value. Range: -32768 to +32767.
CaliSpan	Short	Get/Set the calibration Span value. Range: -32768 to +32767.
UnitSymbol	String	Get the symbol for the unit.

UnitName	String	Get the name of the unit.
UnitDecimal	Integer	Get the decimal point of the unit.
Enabled	Boolean	Get/Set if the channel will be enabled or disabled.
Equation	String	Get the equation name of current channel.
Index	Integer	Get the index of current channel.
CoefficientA	Single(Float)	Get/Set the temperature coefficient value a. Coefficient values are only used for External Thermistor channel
CoefficientB	Single(Float)	Get/Set the temperature coefficient value b. Coefficient values are only used for External Thermistor channel
CoefficientC	Single(Float)	Get/Set the temperature coefficient value c. Coefficient values are only used for External Thermistor channel
Measurement	Double	Get the last retrieved reading
StrainGaugeFactor	Float	If this channel is Strain Gauge channel, set/get the strain gauge factor
LinearEquation_Low	Single(Float)	Get/Set the low value of “Linear” equation
LinearEquation_High	Single(Float)	Get/Set the high value of “Linear” equation

## **cSiteViewSDKSettings**

---

cSiteViewSDKSettings class contains a list of functions for application level settings.

### **Functions**

#### *GetAvailableChannelTypes*

```
public void DisplayProductKeyDialog()
```

#### *Usage:*

Display Product Key dialog to input a new Product Key.

#### *Return Values:*

Void

#### *Arguments:*

Void

#### *DisplayUnitEditor*

```
public void DisplayUnitEditor()
```

#### *Usage:*

Display Unit Editor dialog.



**Return Values:**

Void

**Arguments:**

Void

***DisplayUnitEditor***

**public void** DisplayUnitEditor(**Form** ParentWindow)

**Usage:**

Display Unit Editor dialog.

**Return Values:**

Void

**Arguments:**

Form ParentWindow: the parent window of this dialog

***DisplayPlotPreferences***

**public void** DisplayPlotPreferences()

**Usage:**

Display Plot Preferences Editor dialog.

**Return Values:**

Void

**Arguments:**

Void

***DisplayPlotPreferences***

**public void** DisplayPlotPreferences(**Form** ParentWindow)

**Usage:**

Display Plot Preferences Editor dialog.

**Return Values:**

Void

**Arguments:**

Form ParentWindow: the parent window of this dialog

***DisplayEquationEditor***

**public void** DisplayEquationEditor()

**Usage:**

Display Equation Editor dialog.

**Return Values:**

Void

**Arguments:**

Void

***DisplayEquationEditor***

**public void** DisplayEquationEditor(**Form** ParentWindow)

**Usage:**

Display Equaiton Editor dialog.

**Return Values:**

Void

**Arguments:**

Form ParentWindow: the parent window of this dialog

***About***

**public void** About()

**Usage:**

Display About window to view software version and reactivate the software.

**Return Values:**

Void

**Arguments:**

Void

## **cSiteViewSDKPlotCtrl**

---

cSiteViewSDKPlotCtrl is a standard windows control for data plotting. You can add the control to a window and load a SiteView file.

### **Methods**

***OpenFiles***

**public bool** OpenFiles(**string**[] filenames)

**Usage:**

Load a list of SiteView files.

**Return Values:**

bool: true indicates the files are loaded to the plotting control successfully.

**Arguments:**

**string**[] filenames: a list of SiteView file.

### ***OpenFile***

```
public bool OpenFile(string filename)
```

#### **Usage:**

Load a list of SiteView files.

#### **Return Values:**

bool: true indicates the files are loaded to the plotting control successfully.

#### **Arguments:**

string filenames: a SiteView file.

## **cDeviceServerEntryProperty**

---

cDeviceServerEntryProperty class contains a list of properties of a device server entry.

### **Properties**

<b>Property</b>	<b>Type</b>	<b>Description</b>
Description	String	Get/Set the description about this entry
IP	String	Get/Set the IP address
Port	Integer	Get/Set the Port of the server
Packet	Unsigned Integer	Get/Set the packet size for downloading
Timeout	Unsigned Integer	Get/Set the total timeout for communications
Retries	Integer	Get/Set the total retries for communications
Password	String	Get/Set the password for accessing the server (only used for cUSBDeviceServer)

## **Enumerations**

---

```
public enum enumSaveFileType  
{  
  
    SVF,  
    CSV_WITHOUT_STATUS,  
    CSV_WITH_STATUS  
  
};
```

Specify how the file will be saved.

SVF:	Save the downloaded file as SiteView file. SVF file can be opened by SiteView software.
CSV_WITHOUT_STATUS:	Save the downloaded file as CSV file with dataset only. (TAB Seperated)
CSV_WITH_STATUS	Save the downloaded file as CSV file with status and dataset. (TAB Seperated)

```
public enum enumWifiMode
{
    WifiMode_Disabled,
    WifiMode_Client,
    WifiMode_Client_PowerSave,
    WifiMode_Server
};
```

Specify the Wifi mode of the on-board Wifi moduel.

WifiMode_Disabled:	Disable the Wifi function (Turn off the Wifi module)
WifiMode_Client:	Reserved
WifiMode_Client_PowerSave	Reserved
WifiMode_Server	The logger will act as a server and SiteView can access it by its IP address

```
public enum enumWifiAuthentication
{
    Wifi_Open,
    Wifi_Shared,
    Wifi_WPAPSK,
    Wifi_WPA2PSK
};
```

Specify the Wifi authentication of the access point the logger will join

Wifi_Open:	No authentication
Wifi_Shared:	Shared authentication
Wifi_WPAPSK	WPAPSK authentication
Wifi_WPA2PSK	WPA2PSK authentication

For detailed authentication mode please refer to [Wi-Fi Protected Access](#)

```
public enum enumWifiEncryption
{
    Wifi_Encry_None,
    Wifi_Encry_WEP_H,
    Wifi_Encry_WEP_A,
};
```

Wifi\_Encry\_TKIP,  
Wifi\_Encry\_AES

};

Specify the Wifi encryption of the access point the logger will join

Wifi_Encry_None:	No encryption
Wifi_Encry_WEP_H	Wired Equivalent Privacy H encryption
Wifi_Encry_WEP_A	Wired Equivalent Privacy A encryption
Wifi_Encry_TKIP	Temporal Key Integrity Protocol encryption
Wifi_Encry_AES	Advanced Encryption Standard

For detailed authentication mode please refer to [Wi-Fi Protected Access](#)

# CHAPTER 4 - MORE EXAMPLES

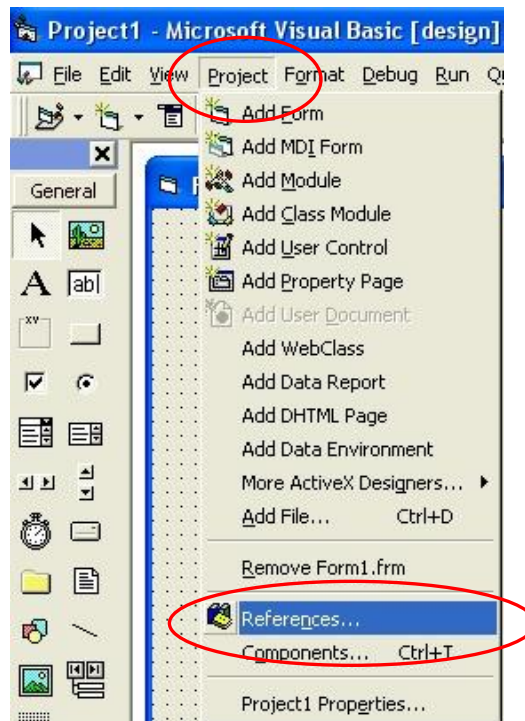
## Visual Basic 6 User

SiteView SDK is developed in .NET development environment and called Managed Code where Visual Studio 6 (including Visual C++6, Visual Basic6) is called Unmanaged Code (Native Code).

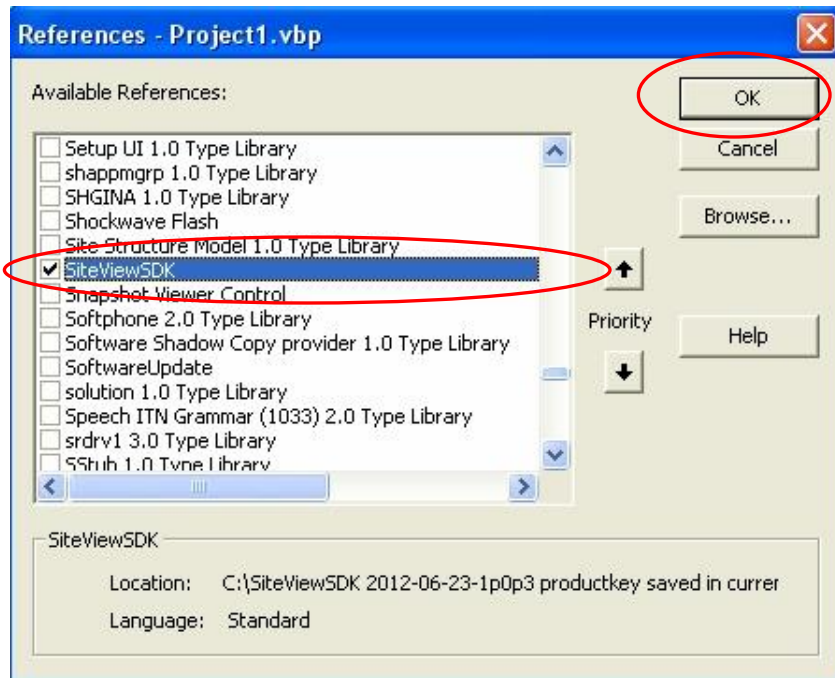
The deployment of SiteView SDK is callable by Unmanaged Code and the following code snippets demonstrate how to invoke SiteView SDK by Visual Basic 6.

## Reference SiteView SDK

1. Open Visual Basic 6 and create a blank project.
2. Click “Project” -> “References...” menu item:



3. Find “SiteView SDK” and check it. Click “OK” button:



## Find Data Loggers in USB Ports

```

Dim usb As New cUSB

Dim loggers() As New cDataLogger

loggers = usb.FindDataLoggers()

Dim i As Integer

i = UBound(loggers)
If i >= 0 Then
    'get realtime values
    Dim realtimevalues() As Double
    realtimevalues = loggers(i).GetCurrentMeasurements()
    'get the logger description
    Me.Caption = loggers(i).Description
    'get the start and end time of data
    Dim starttime As Date
    Dim endtime As Date
    starttime = loggers(i).DataStartTime
    endtime = loggers(i).DataEndTimet
    'get total data points
    Dim total As Long
    total = loggers(i).TotalReadings
End If

```

## Open SiteView File

In Global area type in the following line:

```
Dim graph As VBControlExtender
```

In Form\_Load() function:

```
Private Sub Form_Load()  
  
Set graph = Me.Controls.Add("SiteViewSDK.SiteViewSDKPlotCtrl", "SiteViewSDKPlotCtrl1")  
graph.Visible = True  
graph.Object.OpenFile "c:\graph.svf"  
  
End Sub
```

In Form\_Resize() function:

```
Private Sub Form_Resize()  
graph.Width = Me.Width  
graph.Height = Me.Height  
End Sub
```

## Display About Dialog

```
Dim settings As New SiteViewSDK.cSiteViewSDKSettings  
settings.About
```