Schleifenbauer Dot Net Application Programming Interface

Launch IT
B.C.E. Vaessen

Version 1.2
20-12-2010

1 Contents

1 Contents....2
2 Introduction....3
3 Prerequisites....3
4 Setting up a project....4
4.1 Getting started using C#.Net in Visual Studio....4
4.2 Getting started using Visual Basic .Net in Visual Studio....7
5 Exported classes....8
5.1 Class Diagram....9
5.2 he SBRAPI class....10
5.3 The Gateway class....11
5.4 The PDU class....12
5.5 The SBRCommunicator class....13
5.6 The Message class....14
5.7 The pduIdentification class....15
5.8 The pduConfiguration class....16
5.9 The pduStatus class....17
5.10 The pduSettings class....18
5.11 The pduOutlet class....19
5.12 The pduOutputMeasure class....20
5.13 The pduInputMeasure class....21
5.14 The pduGeneralMeasures class....22
5.15 The SBRError class....23
6 Example application....24
6.1 Getting Started....24
6.2 The User Interface....24

2 Introduction

The Schleifenbauer Application Programming Interface (API) is a Dynamic Link Library (DLL), allowing control of the PDUs using programming languages that can interact with DLLs, such as Visual Basic, C++ and C# (dot net). The API exports classes, of which objects can be built that represent hardware units.

This manual describes the methods, properties and events for the exported interfaces. The purpose of the manual is to support a programmer in rapidly creating applications controlling or monitoring the Schleifenbauer PDUs. Diagrams have been modelled in the Unified Modelling Language (UML)

3 Prerequisites

To be able to develop applications for the Schleifenbauer PDUs and use the examples, it is advised to use:

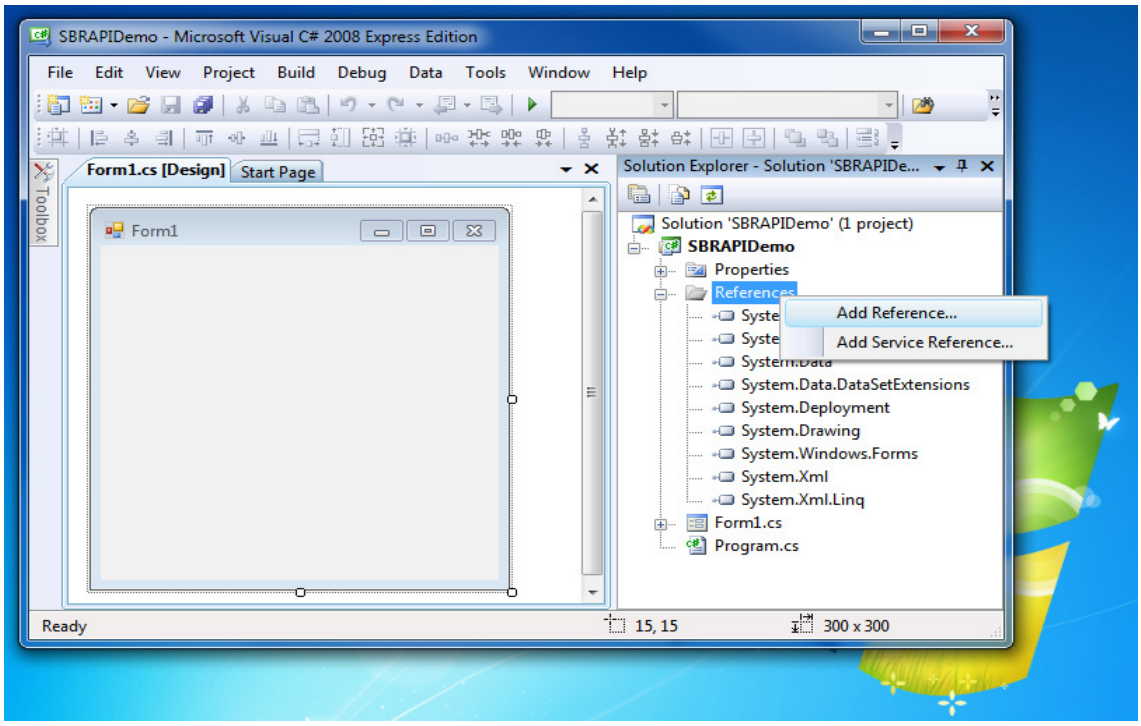
- Visual Studio 2008
- Dot Net 3.5 or later.
- Windows XP, Vista, 7 or later
- Basic programming experience in VB, C++ or C#

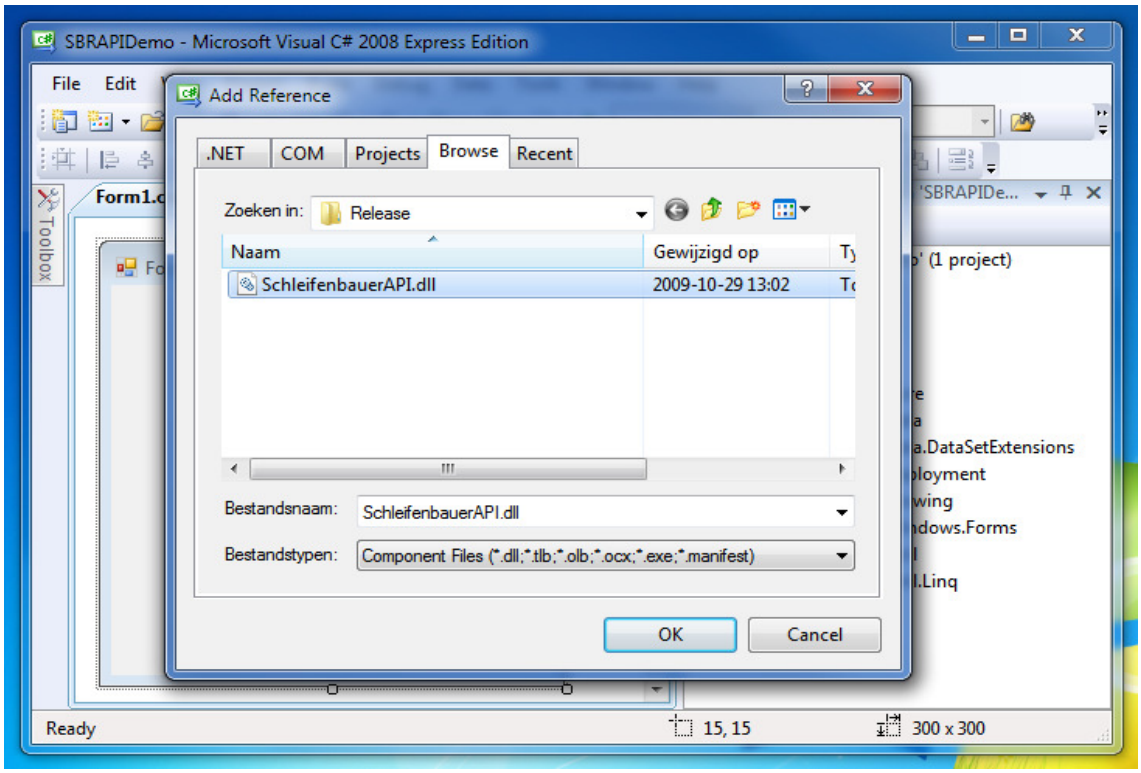
Visual Studio Express can be downloaded free of charge, at the following location:
http://www.microsoft.com/express/download/

4 Setting up a project

4.1 Getting started using C#.Net in Visual Studio

4.1.1 Adding the Schleifenbauer API as a reference





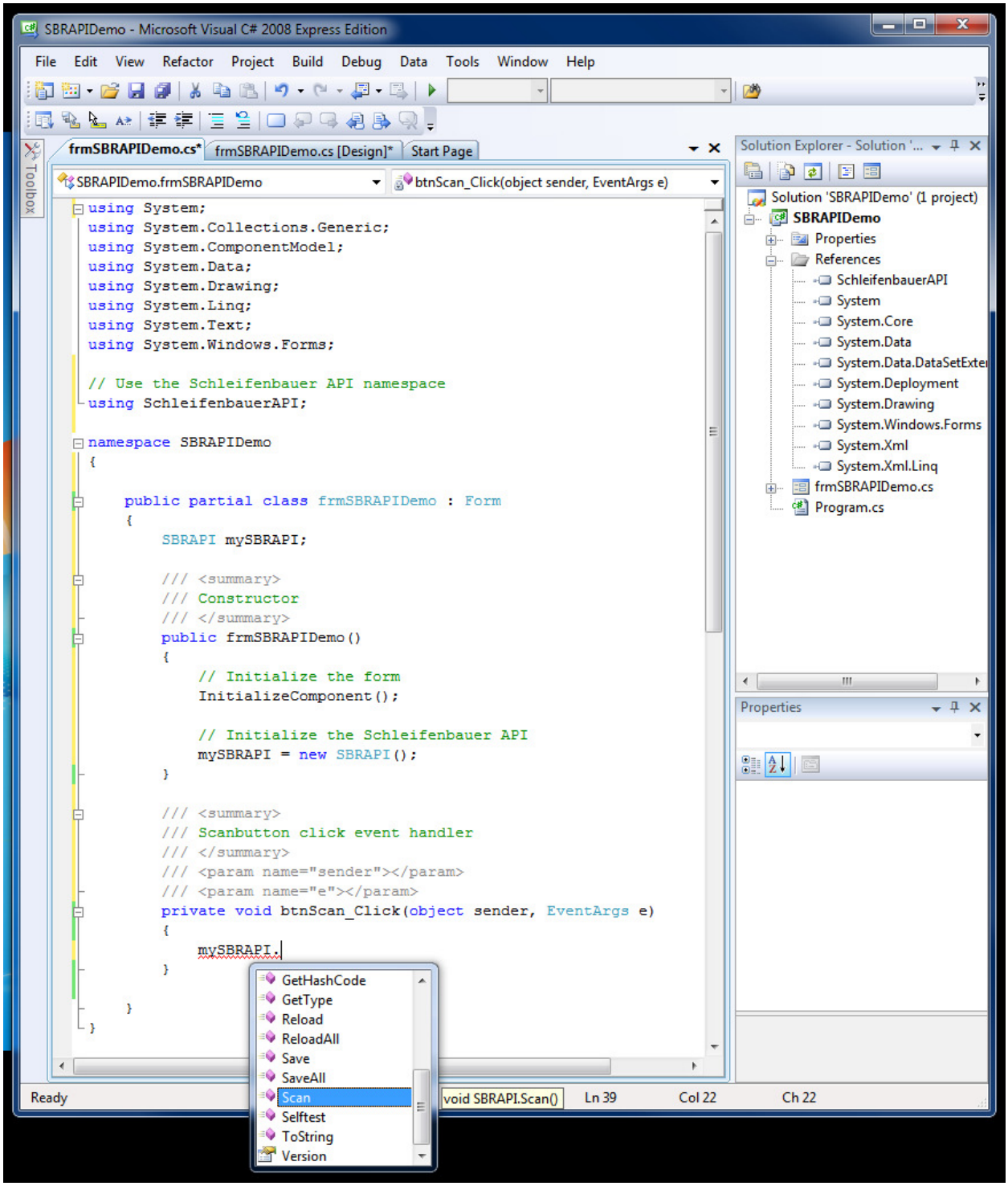
4.1.2 Constructing objects of the exported classes

After adding the SchleifenbauerAPI DLL to the references, it is available for usage. All classes exported by the API are located in the *SchleifenbauerAPI* namespace. To use this namespace, simply include the following line on the top of your code:

```
using SchleifenbauerAPI;
```

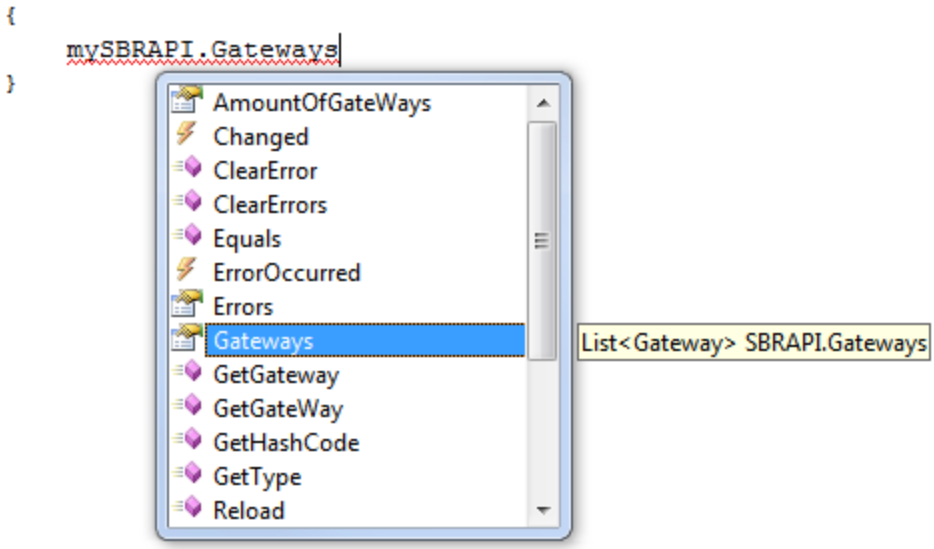
It is now possible to declare and construct the objects provided by the API:

```
SBRAPI mySBRAPI
mySBRAPI = new SBRAPI();
```



4.1.3 Properties and methods

Each of the exported classes contains properties, events and methods for comfortable use in custom built applications. Browse them by typing an object and pressing "." and the browser window shows the object's fields as shown in the image below.



When setting the property, only the local software representation of the class is updated. Call a Save method to update the devices to apply the changes. In order to cancel the changes, call the Reload method of the object.

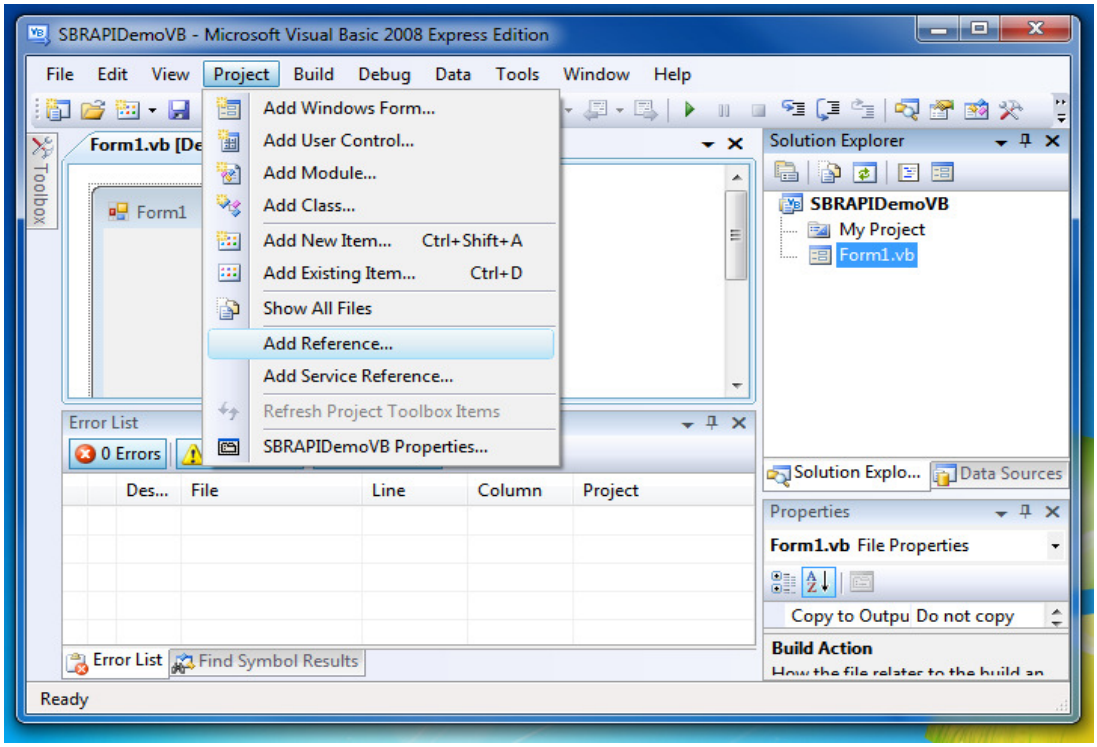
4.1.4 Handling events

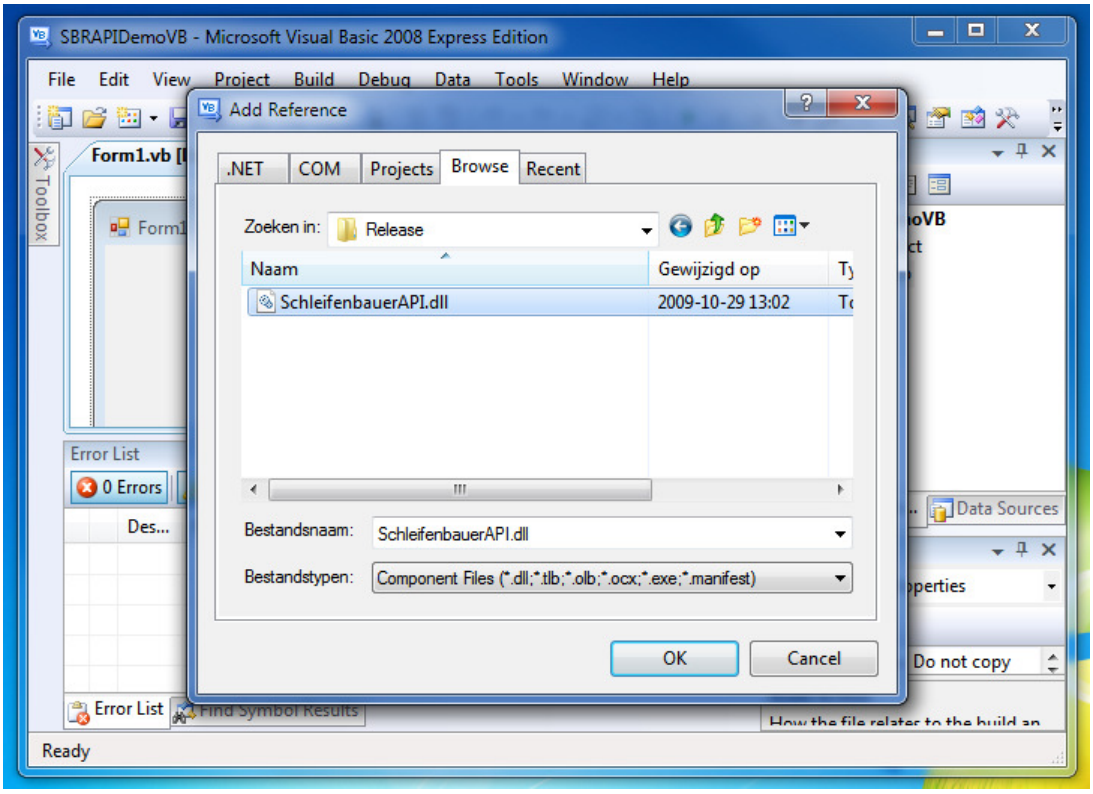
Creating event handlers and connecting them to the Schleifenbauer API:

```
/// <summary>
/// Constructor
/// </summary>
public frmSBRAPIDemo()
{
    …
    …
    …

    // Add an event handler for the error event
    mySBRAPI.ErrorOccurred += new ErrorOccurred(mySBRAPI_ErrorOccurred);

}

/// <summary>
/// My event handler for error handling
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
void mySBRAPI_ErrorOccurred(string sender, SBRError e)
{
    MessageBox.Show("Error", "An error occurred in the SBR API",
          MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

4.2 Getting started using Visual Basic .Net in Visual Studio

4.2.1 Adding the Schleifenbauer API as a reference

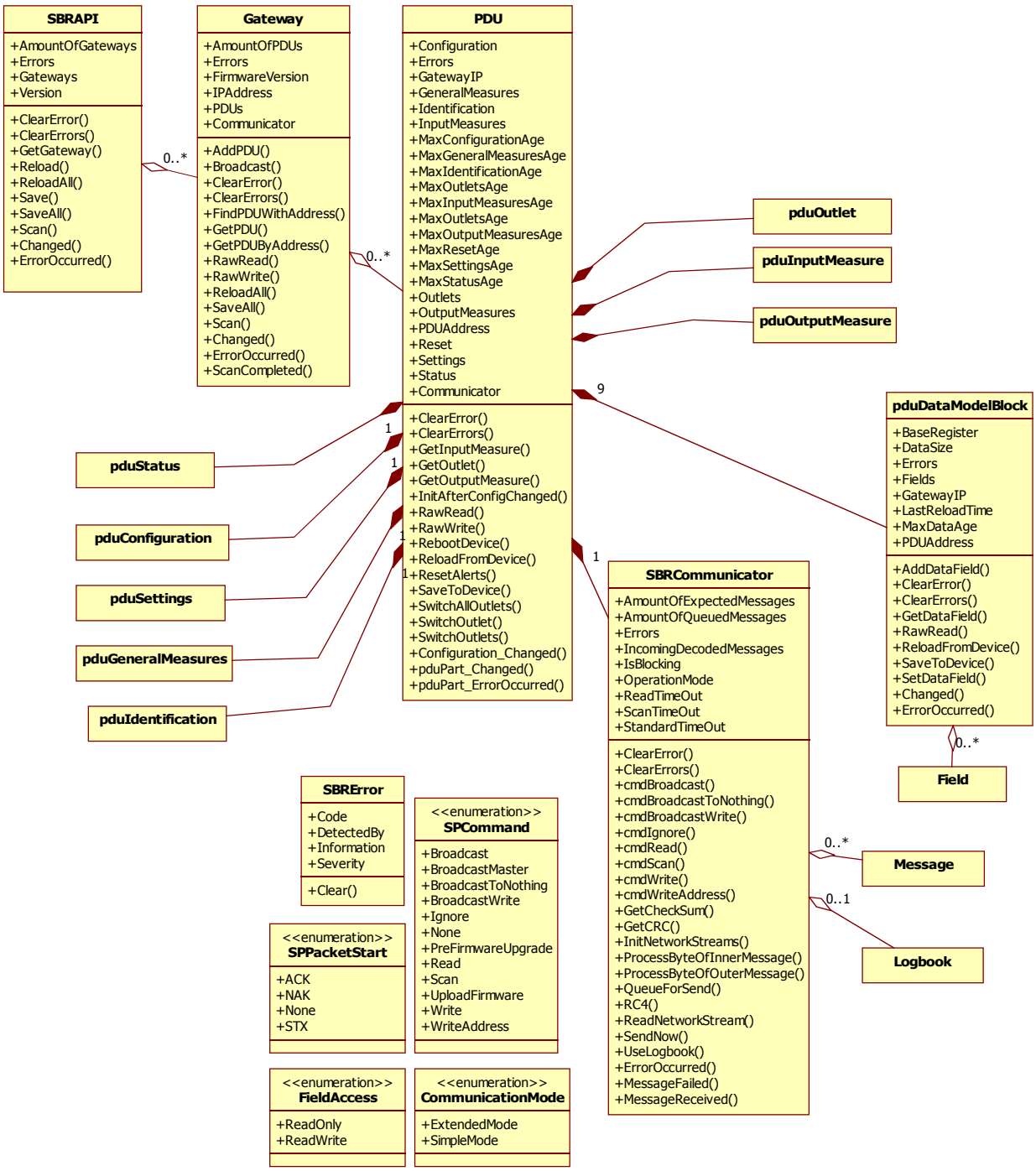




5 Exported classes

SBRAPI	The main entry point to the gateways
Gateway	The software representation of a gateway, which contains a list of all PDUs
PDU	The software representation of a PDU
pduIdentification	Information on the identity of a PDU, including version information
pduStatus	Status information
pduConfiguration	All configuration properties of a PDU, such as amount of inputs and outlets
pduSettings	Settings such as thresholds for alerts
pduOutlet	A software representation of an outlet, for switching
pduOutputMeasure	Contains the measurements of an output, one object for each output
pduInputMeasure	Contains the measurements of an input, one object for each phase
pduGeneralMeasures	General measurements, such as temperatures and temperature peaks
pduDataModelBlock	Represents a block of the SP Datamodel, e.g. Settings or Configuration
SBRError	An object containing error information
SBRCommunicator	Responsible for low-level communication to the Gateway
Message	An SPBus message

5.1 Class Diagram



5.2 he SBRAPI class

The *SBRAPI* class is the main entry point to the API. It is able to scan the network for gateways and keeps a list of these gateways.

5.2.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
Version	A string containg version information
Gateways[]	A List of all gateways found in the network
AmountOfGateways	Contains the amount of gateways in the list

5.2.2 Methods

Scan()	Scans the network and populates the list of Gateways
Selftest()	Performs a self test of the API and the devices
GetGateway(idx)	Returns the *Gateway* object specified by an index
GetGateway(ip)	Returns the *Gateway* object specified by an IP address
ReloadAll()	Calls a reload of all gateways and their properties
SaveAll()	Saves all properties of all gateways
Reload(idx)	Calls a reload of all properties of gateway idx
Save(idx)	Saves all properties of gateway idx
Reload(ip)	Calls a reload of all properties of the gateway specified by the ip address
Save(ip)	Saves all properties of the gateway specified by the ip address
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.2.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.3 The Gateway class

The Gateway provides access to all connected PDUs.

5.3.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
FirmwareVersion	Contains a version string
IPAddress	The network address of the gateway
PDUs[]	A List of *PDU* objects
AmountOfPDUs	Contains the amount of PDUs in the PDU list

5.3.2 Methods

Scan()	Scans the bus and populates the list of PDUs
ReloadAll()	Reloads all properties
SaveAll()	Saves all properties
GetPDU(idx)	Returns the *PDU* specified by an index
Broadcast(addr, data)	Broadcasts a list of data bytes to all connected PDUs
RawWrite(pduidx, addr, data)	Performs a raw write to the specified PDU
RawRead(pduidx, addr, len)	Performs a raw read from the specified PDU
SelfTest()	Performs a self test of the gateway and the connected PDUs
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.3.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.4 The PDU class

5.4.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
AmountOfInputs	The amount of phases for the current PDU
AmountOfOutputs	The amount of outlets for the current PDU
UpdateInterval	The amount of ms for automatic reloading. 0 means no automatic reload.
Identification	An object containing all information needed for identifying a PDU
Configuration	This object contains all configuration properties
Status	Object providing the status information of the PDU
Reset	Object which provides various reset options
Settings	Contains the settings
GeneralMeasures	An object containg information on temperatures
Outlets[]	List of *pduOutlets*, allowing switching of the outlets
OutputMeasures[]	List of *pduOutputMeasures*, containing various output measurements
InputMeasures[]	List of *pduInputMeasures*, containing various input measurements

5.4.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
GetInputMeasure(idx)	Returns a *pduInputMeasure* object, specified by the phase
GetOutputMeasure(idx)	Returns a *pduOutMeasure* object, specified by the outlet index
GetOutlet(idx)	Returns a *pduOutlet* object, specified by the index
SwitchOutlet(idx, state)	Switches outlet idx to the state as boolean.
SwitchOutlets(lst, state)	Switches all outlets in the list of indices to the state specified
SwitchAllOutlets(state)	Switches all outlets at once
RebootDevice()	Reboots the PDU, keeping the outlet states
ResetAlerts()	Resets the alerts
RawWrite(addr, data)	Performs a raw write to the datamodel of the PDU
RawRead(addr, length)	Performs a raw read of the datamodel of the PDU
SelfTest()	Performs a self test of this PDU
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.4.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.5 The SBRCommunicator class

The SBRCommunicator class is used for all low level communication through the Gateway. This class is public and can be used independetly from the other classes, however, only experienced programmers should do this.

5.5.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
AmountOfExpectedMessages	The amount of messages still awaiting an answer
AmountOfQueuedMessages	The amount of messages queued for sending
ReadTimeOut	TimeOut in msec for read replies
ScanTimeOut	TimeOut for scan replies
StandardTimeOut	TimeOut for other messages
OperationMode	
Simple(SPGW)	Answers from the Gateway is data only
Extended (SAPI)	Answers from the Gateway according to SPBus protocol (default)

5.5.2 Methods

cmdBroadcast()	Broadcast command, from master to slaves; no reply expected
cmdBroadcastToNothing()	Broadcast command, from master to nothing, do not reply
cmdBroadcastWrite()	Broadcast write command, from master to slaves, no reply expected
cmdIgnore()	Ignore Command, from master to no specific slave; never reply
cmdRead()	Read command, from master to specific slave; ACK or NAK reply required
cmdScan()	Scan command, from master to slaves; ACK, some error or no reply
cmdWrite()	Write command, from master to specific slave; ACK or NAK reply required
cmdWriteAddress()	Writes an address to a certain device, indexed by a serial number
StopScan()	Call this method to stop scanning. In essence, this command clears all messages in ExpectedMessages that are of the type *scan*
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.5.3 Events

ErrorOccurred(sender, e)	An error occurred. e is a SBRError object
MessageFailed(msg)	A Message was received, but contains failures or NAK
MessageReceived(msg)	A Message was received, and validated

5.6 The Message class

A Message object represents an SPBus message.

5.6.1 Properties

Address	The address of the PDU this message is from or going to
Command	One of the SPBus commands, such as Read, Scan, Broadcast etc.
ID	A message ID
PacketStart	STX, NAK or ACK
RegStart	The start register in the DataModel of the PDU
RegLength	The amount of bytes in the data array
CRC	Cyclic Redundancy Check
Data	The data as an array of bytes
Error	An error code returned by a PDU

DateCreated	Timestamp placed at the moment of creation of this object
DateTransmission	Timestamp placed at the moment of transmission of this message

Complete	True in case the message is complete, False if still being constructed
Failed	True in case of a NAK or an error on processing, otherwise False

AsBytes	Returns the message as an array of bytes

5.6.2 Methods

ToString()	Prints the message as a string, for example for logging purposes

5.6.3 Events

The Message class does not generate events.

5.7 The pduIdentification class

The pduIdentification class contains all information needed for identifying a PDU.

5.7.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
SPDMVersion	An integer representing the datamodel version
FirmwareVersion	The units firmware version as an integer
SalesOrderNumber	Sales order information as a string
ProductID	Product identifier as a string
SerialNumber	Serial number information as a string
HardwareAddress	The hardware serial number as a string (xx-xx-xx)
UnitAddress	The address of the unit, as an integer

5.7.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.7.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.8 The pduConfiguration class

The pduConfiguration class allows for configuration of the PDU. All of the properties can be read from and written to.

5.8.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
AmounfOfPhases	0, 1 or 3, resp. no input metering, single or three phase system
AmountOfOutlets	Total number of outlets, including hardwires without switch/measure modules
AmountOfSwOutlets	Number of switched outlets
AmountOfMeasOutlets	Number of measured outlets
MaximumLoad	Maximum rated load of device per phase, usually either 16 or 32A
AmountOfTmpSensors	Amount of temperature sensors (0=none, 1=internal, 2=int and ext)

5.8.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.8.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.9 The pduStatus class

This class contains information on the status of the PDU. All properties are read only.

5.9.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
DeviceStatusCode	An integer representing the internal status or error code. 0 is undefined, 1 is ok
TemperatureAlert	0 in case of no alert, 1 for internal temperature, 2 for external temperature
InputCurrentAlert	An alert due to current exceeding threshold. 0=no alert, 1..3 for input phase
OutputCurrentAlert	An alert due to current exceeding threshold. 0=no alert, 1..27 for outlet
InputVoltageAlert	An alert has been raised due to a voltage dup. 0=no alert, 1..3 for input phase
CurrentDropAlert	An alert has been raised due current dropping to zero, e.g. indicating fuse blown

5.9.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.9.3 Events

Changed(sender)	One or more of the properties have been changed.
HighTemp(sender, idx)	A temperature alert has occurred
HighInputCurrent(sender, idx)	The input current of phase idx exceeded the threshold
HighOutputCurrent(sender, idx)	The output current of outlet idx exceeded the threshold
InputVoltageDip(sender, idx)	A voltage dip occurred on input idx
CurrentDrop(sender, idx)	Current drop, or fuse blown detected
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.10 The pduSettings class

The pduSettings class contains all settings of a PDU.

5.10.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
DeviceName	User configurable device name or identifier
DeviceLocation	User configurable device location identifier
VanityTag	String to be displayed as vanity tekst in display
PeakDuration	A current peak should last at least [PeakDuration] milliseconds to be detected
DipDuration	A voltage dip should last at least [DipDuration] milliseconds to be detected
FixedOutletDelay	Minimal delay between two successive relay switches in milliseconds
PowerSaverMode	Backlight on time in seconds; 0 keeps display always on.
OutletPowerUpMode	behaviour of outlet on power-up: 0=off 1=same state as at power down 2=same state as at power down, but delayed by individual delay timer.
MaximumTemperature	An alert is generated in case the temperature is above this threshold
DisplayOrientation	0=no display; display off 1=vertical, display on top 2=vertical, upside down 3=horizontal, display at left size 4=horizontal, display at right side

5.10.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.10.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.11 The pduOutlet class

5.11.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
SwitchState	The actual state of the outlet as a boolean
Scheduled	Pending activity as boolean
UnlockState	A boolean representing the unlock state. True means unlocked.
MaxAmps	A double representing maximum current per outlet in A
Groups	An integer for user configurable grouping of individual outlets
Name	A string representing the name of the outlet
IndivDelay	Delay before the individual outlet switches on at power-up in seconds

5.11.2 Methods

SwitchOn()	Switches on the outlet (unlocks outlet, carried out immediately)
SwitchOff()	Switches the outlet off (unlocks outlet, carried out immediately)
Switch(state)	Switches the outlet to the specified state as boolean (unlocks outlet, carried out immediately)
Toggle()	Toggles the outlet, inverting the state (unlocks outlet, carried out immediately)
ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.11.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.12 The pduOutputMeasure class

The pduOutputMeasure class contains all output measurements for one output.

5.12.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
TotalkWh	Total kWh of this output
SubTotalkWh	Subtotal kWh of this output
PowerFactor	The effective power factor in percent
ActualCurrent	Actual apparent, RMS current
PeakCurrent	Peak apparent RMS current. Last value since last reset of the alerts.
ActualVoltage	Actual voltage of the output

5.12.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.12.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.13 The pduInputMeasure class

Contains the input measurements for a phase (input)

5.13.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors
TotalkWh	Total kWh for this input
SubTotalkWh	Subtotal kWh for this input
PowerFactor	The effective power factor in percent
ActualCurrent	Actual apparent, RMS current
PeakCurrent	Peak apparent RMS current. Last value since last reset of the alerts.

5.13.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.13.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.14 The pduGeneralMeasures class

Contains general measurements, such as temperatures.

5.14.1 Properties

Errors[]	A List of *SBRError* objects containing information on the errors

InternalTemperature	Actual internal temperature in degrees C
ExternalTemperature	Actual external temperature in degrees C
InternalPeakTemp	Internal temperature peak since last reset
ExternalPeakTemp	External temperature peak since last reset

5.14.2 Methods

ReloadFromDevice()	Loads the properties from the device, updating the object
SaveToDevice()	Saves the properties in the device
ClearErrors()	Clears the list of errors
ClearError(idx)	Removes the error specified by idx

5.14.3 Events

Changed(sender)	One or more of the properties have been changed.
ErrorOccurred(sender, e)	An error occurred. e is a SBRError object

5.15 The SBRError class

All objects part of the Schleifenbauer API contain the error object, which can be used for looking up information on the last error.

5.15.1 Properties

Information	String containing error information
Code	An error code
DetectedBy	String containg information on the class and method which generated this error
Severity	An integer indicating how severe the error is

5.15.2 Methods

Clear()	Clears the error

5.15.3 Events

Changed(sender)	One or more of the properties have changed.

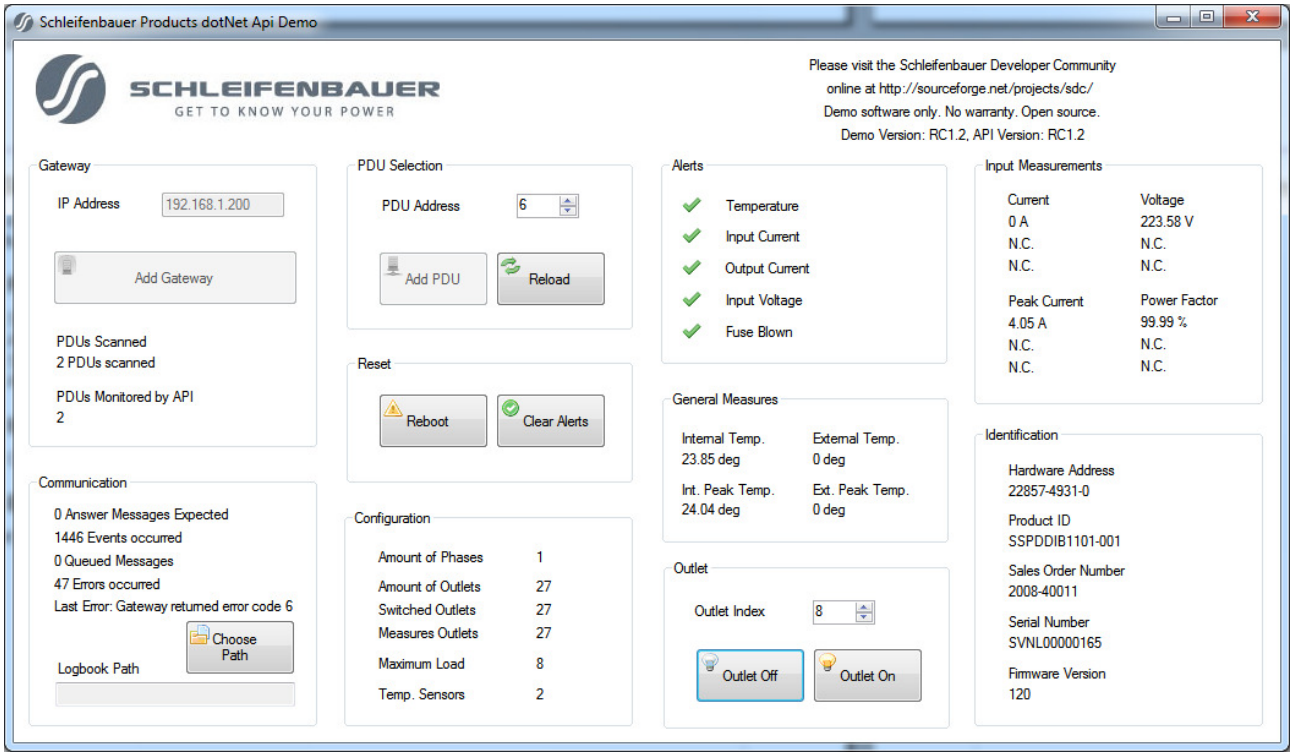
6 Example application

6.1 Getting Started

1. make sure Dot Net 3.5 is installed on your system, or download it online for free.
2. Copy the files *SchleifenbauerAPI.dll* and *SBRAPIDemo.exe* to a folder of your choice.
3. Make sure your Firewall allows communication to the Gateways.
4. Start *SBRAPIDemo.exe*

6.2 The User Interface

The user interface of the Demo Application demonstrates several of the functions provided by the API.



After entering a valid IP address, press *Add Gateway*. Scanning will start. Within a couple of seconds, the label showing the amount of Scanned PDUs should update. Within the PDU Selection group, browse through the PDUs using the Numeric Input Field. A red sign on the *Add PDU* Button indicates that the PDU has not been found by the scan. If the PDU is available, add it for monitoring by pressing the *Add PDU* Button now. The other fields on the form will become enabled and should update approximately each second now.