

# G5 Series

G5.API

---

**.Net API Access**

V 1.20.0.x

---

REGATRON AG  
Feldmuehlestrasse 50  
9400 Rorschach  
SWITZERLAND

Tel +41 71 846 67 67  
[www.regatron.com](http://www.regatron.com)  
[support@regatron.com](mailto:support@regatron.com)

## General Information on this Document

### Identification of the Document

Identifier	G5.API
Version	V01.60

### Usage of the Document

This System Manual describes the system components and serves as a guide and also as a reference work. Familiarize yourself with the contents of the document to operate the device efficiently. The document must be available at all times to the personnel who are operating the device.

### Copyright Information

© 2024 Regatron AG

This document is protected by copyright.

All rights, including translation, re-printing and duplication of this manual or parts of it, reserved. No part of this work is allowed to be reproduced, processed, copied or distributed in any form, without the written approval of Regatron.

The information in this documentation corresponds to the development situation at the time of going to print and is therefore not of a binding nature. Regatron AG reserves the right to make changes at any time for the purpose of technical progress or product improvement, without stating the reasons. In general we refer to the applicable issue of our "Terms of delivery".

# Table of Contents

<b>General Information on this Document .....</b>	<b>2</b>
Identification of the Document .....	2
Usage of the Document.....	2
Copyright Information .....	2
<b>Table of Contents .....</b>	<b>3</b>
<b>1 Overview .....</b>	<b>4</b>
1.1 Content .....	4
1.2 Precondition / Dependencies .....	4
1.3 Licensing.....	4
<b>2 General Architecture .....</b>	<b>5</b>
2.1 Integration of the G5.API.....	5
2.2 Basic architecture of the G5.API.....	6
2.3 Error handling.....	7
2.4 Logging .....	8
<b>3 Integrating the G5.API in Python.....</b>	<b>9</b>
3.1 Installing the pythonnet package .....	9
3.2 Working with the G5.API in Python .....	9
3.2.1 Import needed modules.....	9
3.2.2 Access the G5.API functionality.....	9
3.2.3 Error handling of the G5.API .....	10
<b>4 Integrating the G5.API in PowerShell .....</b>	<b>11</b>
4.1 Working with the G5.API in PowerShell.....	11
4.1.1 Import the G5.API .....	11
4.1.2 Access the G5.API functionality.....	11
4.1.3 Error handling of the G5.API .....	12
<b>5 Integrating the G5.API in C# .Net.....</b>	<b>13</b>
5.1 Working with the G5.API in C# .Net.....	13
5.1.1 Import the G5.API .....	13
5.1.2 Access the G5.API functionality.....	13
5.1.3 Error handling of the G5.API .....	13
<b>6 Change log.....</b>	<b>14</b>

# 1 Overview

The G5.API can be used to access the G5 device programmatically and perform a certain set of operations.

It cannot substitute the G5.Control, G5.SASControl or G5.ESSControl PC software, but offers some methods to automate operations.

The main configuration and error analysis can only be done with the G5.Control or G5.ESSControl software.

For more details please see the doxygen documentation and the example applications and scripts.

Access the doxygen documentation with the G5.Api\_Documentation.html file.

## 1.1 Content

This document provides a guideline how to access the G5.API implemented with .Net. It does not explain all the method that the API exposes, it only provides a basic overview and some examples how to use.

To get a complete method and class description of the G5.API, please refer to the Doxygen documentation.

## 1.2 Precondition / Dependencies

The G5.API runs on Windows 10 (32bit and 64bit) and Windows 11. Other Operating Systems that support .Net 6 may work as well but have not been tested.

To be able to use the G5.API the following frameworks and libraries have to be installed on the target system:

- Microsoft Visual C++ Redistributable for Visual Studio 2015-2022 x86
- Microsoft Visual C++ Redistributable for Visual Studio 2015-2022 x64
- Microsoft .NET 6

**Note:**

All dependencies are installed on your device by installing the G5.Control software.

## 1.3 Licensing

By using the G5.API you accept the license agreement under "License Agreement G5.Control.pdf".

## 2 General Architecture

### 2.1 Integration of the G5.API

To use the G5.API you only have to add a reference to the Regatron.G5.Api.dll to your application.

The Regatron.G5.Api.dll has dependencies to the Regatron.G5.FunctionGenerator.Native.dll and the Regatron.G5.MathModelEN50530Native.dll. Both files have to be placed in the same folder as the Regatron.G5.Api.dll.

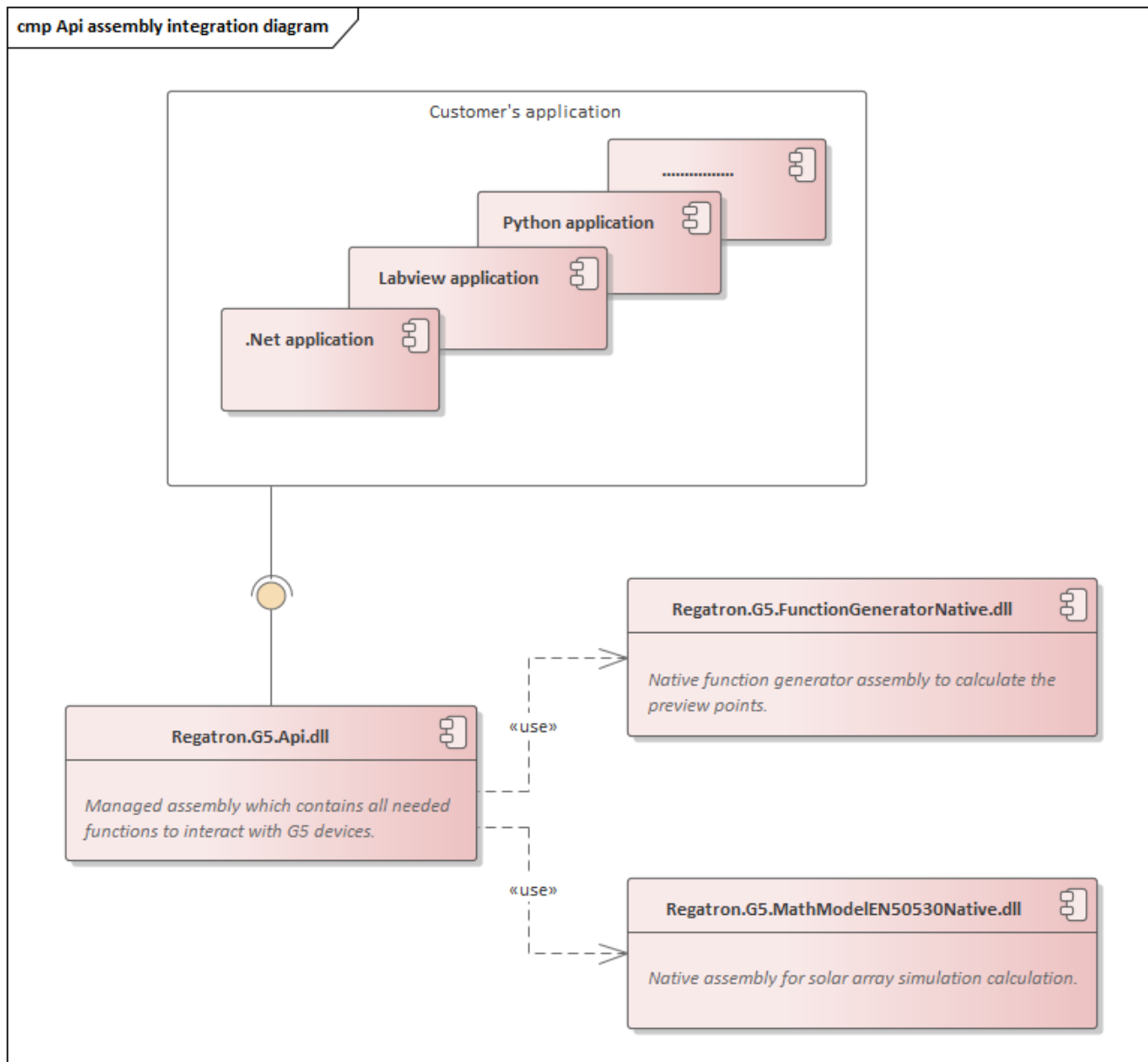


Fig. 1: Integration of the Regatron.G5.Api.dll

## 2.2 Basic architecture of the G5.API

In Figure 2 you can see the basic architecture of the G5.API.

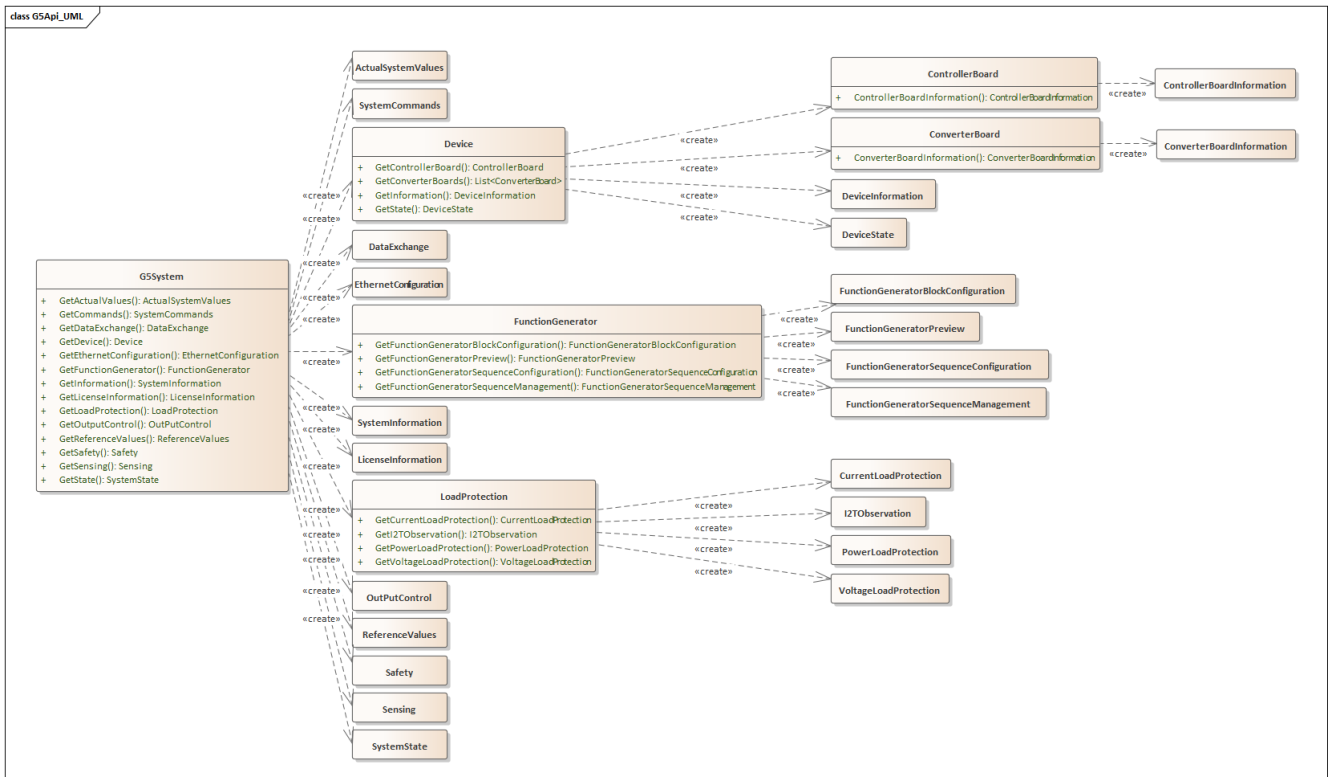


Fig. 2: Structure diagram of the G5.API

To use the G5.API functionality you only have to get an instance of G5System. To create a G5System you can use the static factory method `G5System.CreateSystem()`.

The G5System instance serves as a kind of factory for some sub-objects in the G5.API. You can not instantiate the objects directly. To instantiate an object call the corresponding Get-method in the G5System instance.

Some of the sub-objects themselves create sub-objects and provide Get-methods for this purpose. An extract of the object graph you can see in Figure 2.

The example below shows you one specific task in pseudo code. As you can see, you never call a constructor of an object directly.

To get all other methods provided by each object please refer to the doxygen documentation through the `G5.Api_Documentation.html` file.

**Example: You want to get the version of the controller board firmware.**

```

G5SystemInstance = G5System.CreateSystem();
DeviceInstance = G5SystemInstance.GetDevice(DeviceID);
ControllerBoardInstance = DeviceInstance.GetControllerBoard();
ControllerBoardInformationInstance = ControllerBoardInstance.GetInformation();
Version = ControllerBoardInformationInstance.GetDspFirmwareVersion();
  
```

## 2.3 Error handling

The G5.API wraps all internal errors in a `G5ApiException` which is thrown outside the G5.API to inform the using application about the error.

Therefore your application needs to handle the `G5ApiException` for every G5.API method call. The example below shows an exemplary pseudo code.

To get detailed information about the structure of the `G5ApiException` please refer to the doxygen documentation through the `G5.Api_Documentation.html` file.

**Example : You want to connect to a device via comport 2 and baudrate 230400.**

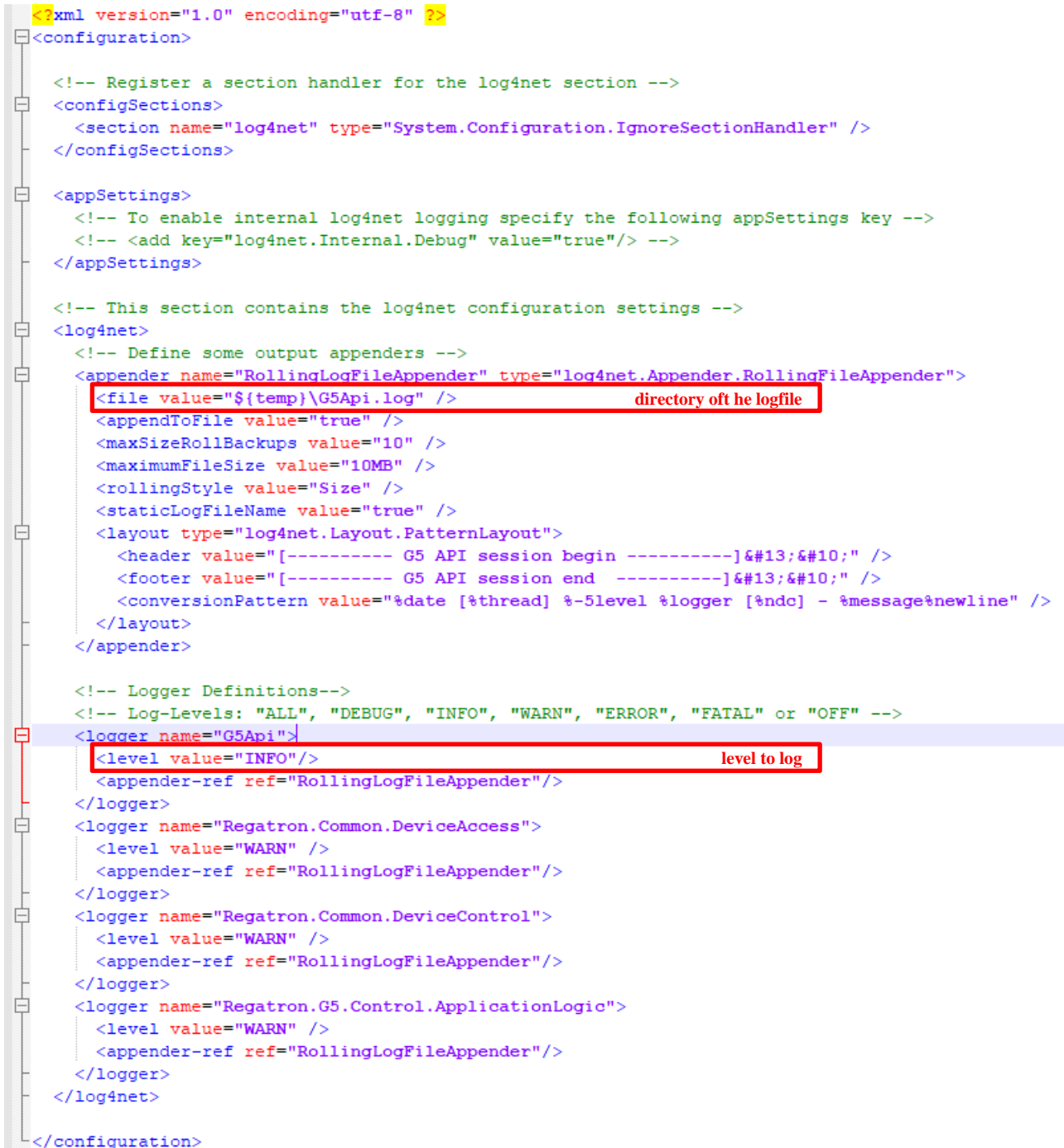
```
try
{
    G5SystemInstance = G5System.CreateSystem();
    operationResult = G5SystemInstance.Connect(2, 230400);
}
catch (G5ApiException g5ApiException)
{
    someErrorHandlingByTheCustomerApplicationIfTheConnectHadSomeErrors;
}
```

## 2.4 Logging

In case an operation is not successful a `G5ApiException` is thrown which already provides some debugging information. If this information is not yet sufficient the G5.API provides a logging framework. This can be used by the customer application developer to get more detailed information about the error.

To make the G5.API log information you have to copy the log configuration file `log.config` into the execution directory of the customer application.

In the log configuration file you can configure the logging level by changing the level and define the output directory of the generated logfile by changing the path. The default path is the temporary folder accessible by typing `%temp%` in the explorer.



```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>

  <!-- Register a section handler for the log4net section -->
  <configSections>
    <section name="log4net" type="System.Configuration.IgnoreSectionHandler" />
  </configSections>

  <appSettings>
    <!-- To enable internal log4net logging specify the following appSettings key -->
    <!-- <add key="log4net.Internal.Debug" value="true"/> -->
  </appSettings>

  <!-- This section contains the log4net configuration settings -->
  <log4net>
    <!-- Define some output appenders -->
    <appender name="RollingLogFileAppender" type="log4net.Appender.RollingFileAppender">
      <file value="$(temp)\G5Api.log" />
      <appendToFile value="true" />
      <maxSizeRollBackups value="10" />
      <maximumFileSize value="10MB" />
      <rollingStyle value="Size" />
      <staticLogFileName value="true" />
      <layout type="log4net.Layout.PatternLayout">
        <header value="----- G5 API session begin -----]&#13;&#10;" />
        <footer value="----- G5 API session end -----]&#13;&#10;" />
        <conversionPattern value="%date [%thread] %-5level %logger [%ndc] - %message&#13;&#10;" />
      </layout>
    </appender>

    <!-- Logger Definitions-->
    <!-- Log-Levels: "ALL", "DEBUG", "INFO", "WARN", "ERROR", "FATAL" or "OFF" -->
    <logger name="G5Api">
      <level value="INFO"/>
      <appender-ref ref="RollingLogFileAppender"/>
    </logger>
    <logger name="Regatron.Common.DeviceAccess">
      <level value="WARN" />
      <appender-ref ref="RollingLogFileAppender"/>
    </logger>
    <logger name="Regatron.Common.DeviceControl">
      <level value="WARN" />
      <appender-ref ref="RollingLogFileAppender"/>
    </logger>
    <logger name="Regatron.G5.Control.ApplicationLogic">
      <level value="WARN" />
      <appender-ref ref="RollingLogFileAppender"/>
    </logger>
  </log4net>
</configuration>
```

Fig. 3: G5.API log configuration file `log.config`



## 3 Integrating the G5.API in Python

This chapter briefly describes some important points regarding accessing the API from a python script. This setup has been tested with Python 3.10.8 (64-bit) and Pythonnet 3.0.1.

The example script itself can be found in the folder Examples → Python.

This example implements a command prompt application to do some basic interactions with the device.

### 3.1 Installing the pythonnet package

To be able to use .Net libraries under Python you need to install the pythonnet package. Please make sure that the IronPython package is not installed or you will get an error when you try to import clr.

To install the pythonnet package go to your python command prompt and write the following command.

```
pip install pythonnet
```

Alternatively you can use the requirements.txt file to install all needed packages.

```
pip install -r requirements.txt
```

### 3.2 Working with the G5.API in Python

Copy the file Regatron.G5.Api.dll, Regatron.G5.FunctionGeneratorNative.dll and Regatron.G5.MathModelEN50530Native.dll into the same folder as your Python script is.

#### 3.2.1 Import needed modules

If you want to copy the Regatron.G5.Api.dll in a random folder, you need to add the path to this folder into your script.

```
import sys  
sys.path.append("Your_Folder_Path")
```

Within the G5.API uint16 are used and Python does not support this type so we need the numpy module.

```
import numpy
```

To get access to the pythonnet package functionality you have to import the clr.

```
import clr
```

To get access to the G5.API you have to add a reference to it.

```
clr.AddReference('Regatron.G5.Api')
```

#### 3.2.2 Access the G5.API functionality

As Figure 2 states the G5System class is the base class to get access to all sub classes. So the first thing you have to do is to get an instance of the G5System.

```
from Regatron.G5.Api.System import G5System  
g5Instance = G5System.CreateSystem()
```

To get an overview of the provided methods and properties please refer to the doxygen documentation through the G5.Api\_Documentation.html file.

### 3.2.3 Error handling of the G5.API

All internal errors of the G5.API are wrapped inside a `G5ApiException`.

For example a desired reference value is outside the allowed device range, then it is wrapped in a `G5ApiException` and thrown outside the G5.API.

So you always have to put a try-except-clause around every G5.API method call.

The following code snippet shows an example for error handling when setting reference values.

```
from Regatron.G5.Api import G5ApiException

try:
    referenceValuesInstance = g5Instance.GetReferenceValues()
    referenceValuesInstance.SetVoltage(float(voltage))
except G5ApiException as exception:
    doSomeErrorHandling()
```

## 4 Integrating the G5.API in PowerShell

The Windows PowerShell example G5.Api.Example.ps1 shows how to access the G5.API with a Powershell Script.

If the G5.API is downloaded from the internet, make sure that under Security the Unblock-Button is set or not visible. This can be done with a "right click" on the Regatron.G5.Api.dll → Properties → General → Security → Unblock. Otherwise the Powershell Script cannot be started.

### 4.1 Working with the G5.API in PowerShell

The example script G5.Api.Example.ps1 runs on Windows PowerShell Version 7.2 and newer. Other older versions which support CLR4.0 may work as well, but have not been tested. If you need to upgrade your PowerShell follow the instructions on the link <https://learn.microsoft.com/de-de/powershell/scripting/install/installing-powershell-on-windows?view=powershell-7.2>. On the same page there are informations about how to use it as well.

Before running the script you have to make sure that scripts are enabled in your shell. If not, please follow this instructions: [https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about\\_execution\\_policies?view=powershell-7.2](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_execution_policies?view=powershell-7.2) for more information.

For example if you want to enable scripts to the actual process enter this instruction:

```
Set-ExecutionPolicy -ExecutionPolicy Unrestricted -Scope Process.
```

#### 4.1.1 Import the G5.API

Copy the actual Regatron.G5.Api.dll, Regatron.G5.FunctionGeneratorNative.dll and Regatron.G5.MathModelEN50530Native.dll right next to the script G5.Api.Example.ps1.

The Regatron.G5.Api.dll is loaded to the process by the following command:

```
Add-Type -AssemblyName $FullPathToG5Api
```

#### 4.1.2 Access the G5.API functionality

As Figure 2 states the G5System class is the base class to get access to all sub classes. So the first thing you have to do is to get an instance of the G5System.

```
$g5SystemInstance = [Regatron.G5.Api.System.G5System]::CreateSystem()
```

To get an overview of the provided methods and properties please refer to the doxygen documentation through the G5.Api\_Documentation.html file.

#### 4.1.3 Error handling of the G5.API

All internal errors of the G5.API are wrapped inside a `G5ApiException`.

For example a desired reference value is outside the allowed device range, then it is wrapped in a `G5ApiException` and thrown outside the G5.API.

So you always have to put a try-except-clause around every G5.API method call.

The following code snippet shows an example for error handling when setting reference values.

```
from Regatron.G5.API import G5ApiException

try
{
    $referenceValuesInstance = $g5SystemInstance.GetReferenceValues()
    $referenceValuesInstance.SetVoltage(float(voltage))
}
Catch [Regatron.G5.Api.G5ApiException]
{
    doSomeErrorHandling
}
```

## 5 Integrating the G5.API in C# .Net

The C#.Net example solution G5.Api.Example.sln shows how to access the G5.API with C# .Net with Visual Studio. The solution is implemented with WPF.

### 5.1 Working with the G5.API in C# .Net

The example solution G5.Api.Example.sln runs on VisualStudio 2022.

#### 5.1.1 Import the G5.API

Copy the actual Regatron.G5.Api.dll, Regatron.G5.FunctionGeneratorNative.dll and Regatron.G5.MathModelEN50530Native.dll right next to the solution file G5.Api.Example.sln.

Now you have to set the correct platform (x64 or x86) corresponding to the copied Regatron.G5.Api.dll. When you set the wrong platform you will get a BadImageFormatException.

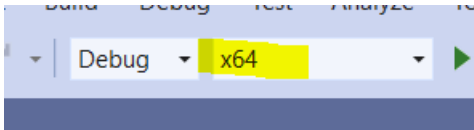


Fig. 4: Platform selection

#### 5.1.2 Access the G5.API functionality

As Figure 2 states the G5System class is the base class to get access to all sub classes. So the first thing you have to do is to get an instance of the G5System.

```
G5System G5SystemInstance = G5System.CreateSystem();
```

To get an overview of the provided methods and properties please refer to the doxygen documentation through the G5.Api\_Documentation.html file.

#### 5.1.3 Error handling of the G5.API

All internal errors of the G5.API are wrapped inside a G5ApiException.

For example a desired reference value is outside the allowed device range, then it is wrapped in a G5ApiException and thrown outside the G5.API.

So you always have to put a try-catch-clause around every G5.API method call.

The following code snippet shows an example for error handling when setting reference values.

```
try
{
    ReferenceValuesInstance = G5SystemInstance.GetReferenceValues();
    ReferenceValuesInstance.SetVoltage(float(voltage));
}
catch (G5ApiException g5ApiException)
{
    doSomeErrorHandling();
}
```

## 6 Change log

This change log contains a list of changes in the previous versions of the documentation

Version	Date	Changes in documentation
1.0	2021-December-2	Initial documentation
1.1	2022-April-28	Chapter 1.2 Precondition / Dependencies adapted
1.2	2022-November-22	Chapter 2.4 Logging: screenshot adapted
1.3	2023-January-5	Chapter 2.1 Dependency to the Regatron.G5.MathModelEN50530Native.dll added.
1.4	2023-April-04	Chapter 1.2 Windows 11 added to the list of supported operating systems.
1.5	2023-November-14	Description to get an instance of G5System modified, to match the multi connection changes.
1.6	2024-August-19	G5.API version number updated .NET requirement changed from .NET Framework 4.7.2 to .NET6 runtime Matlab and LabView examples removed

For detailed technical information, contact your local sales partner or REGATRON.

Regatron AG  
Feldmuehlestrasse 50  
9400 Rorschach  
SWITZERLAND

[sales@regatron.com](mailto:sales@regatron.com)  
[www.regatron.com](http://www.regatron.com)

Regatron Inc.  
100 Overlook Center, 2<sup>nd</sup> Floor  
Princeton, NJ 08540  
USA

[inquiries@us.regatron.com](mailto:inquiries@us.regatron.com)  
[www.us.regatron.com](http://www.us.regatron.com)

All product specifications and information contained herein are subject to change without notice.

Filename: G5\_API\_GettingStarted

Class: Project specific use only