

# Python API



# Introduction

This document will walk you through the basics of how to use the Python API with REPORTER. This new feature is available starting with REPORTER v21.0.

For more information on the Python API, see the [Python API reference manual](#).



# Starting a gRPC server

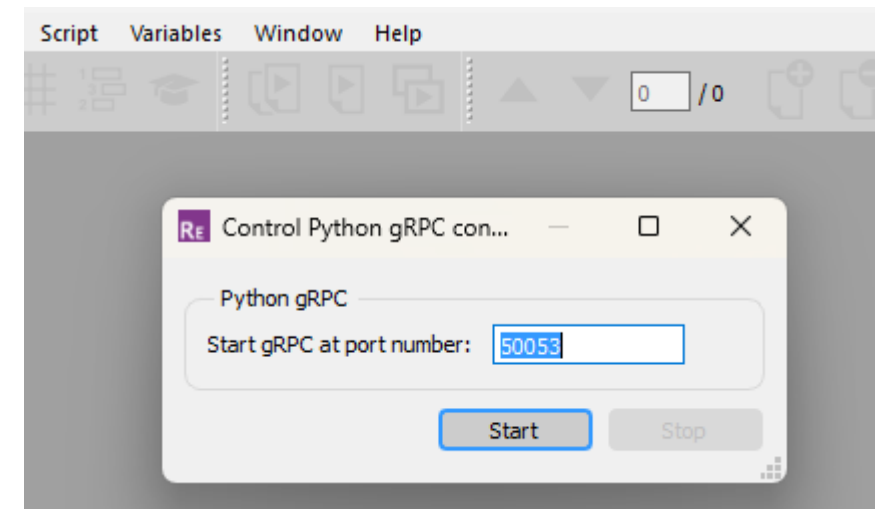
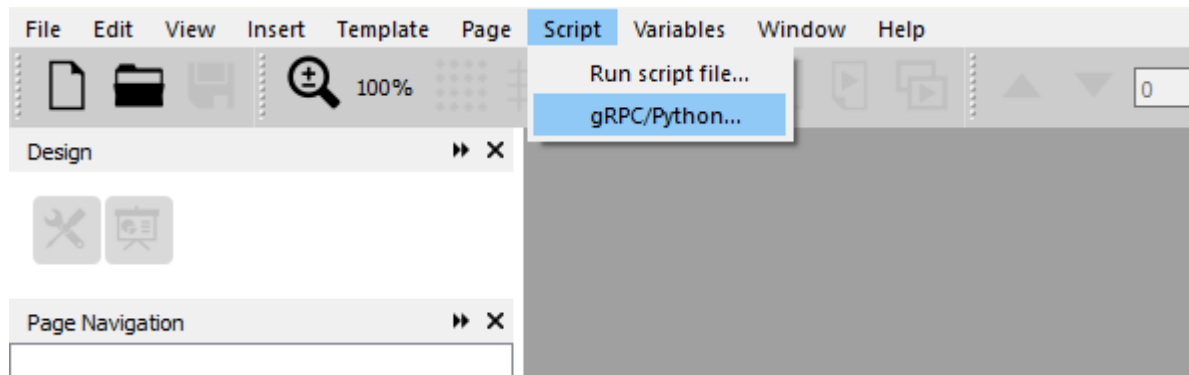
- gRPC is the framework that enables the communication between Python and REPORTER. There are multiple ways in which you can create a gRPC server:
  1. Via the graphical menu interface in REPORTER
  2. Via the command line, by supplying an additional argument when starting REPORTER
  3. Directly from a Python script
- The next slides will demonstrate each of these scenarios to aid you in starting work with the Python API.



# Starting a gRPC server

## 1. Via the graphical menu interface in REPORTER:

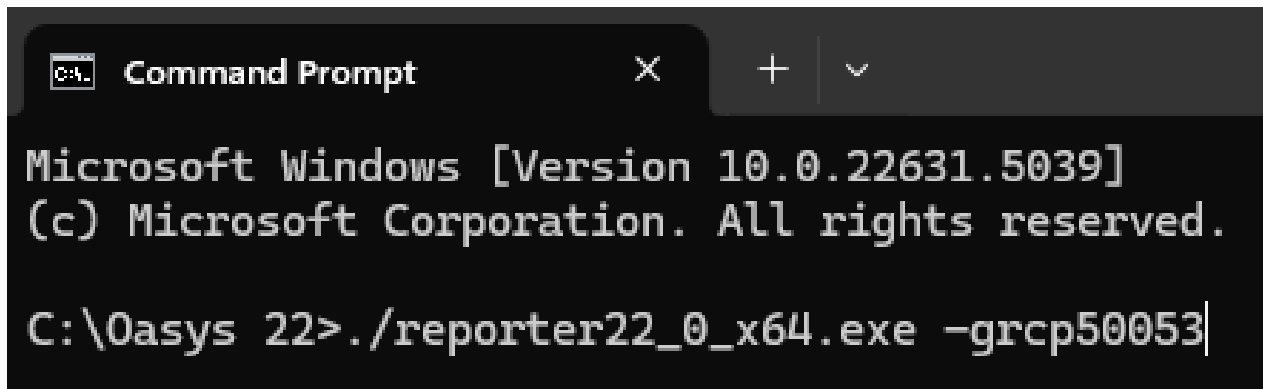
- Open REPORTER as you normally would, by double clicking on the executable or running it via the command line;
- In the main menu, go to **Script -> gRPC/Python**;
- In the pop-up dialog box, you can start the server with the default port number supplied or with a custom one.



# Starting a gRPC server

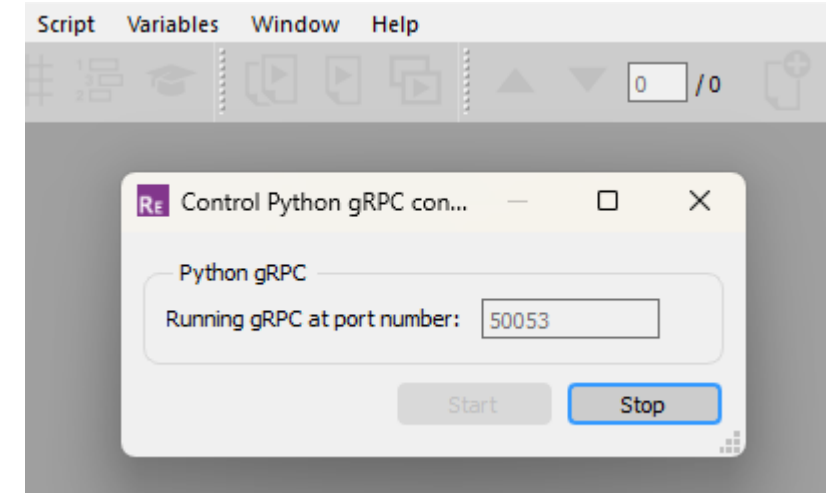
2. Via the command line, by supplying an additional argument when starting REPORTER:

- You can use the argument “**-grpc=50053**” (or any other valid, available port);
- This will start REPORTER in batch mode, as well as create a gRPC server connection;
- You can verify the server was started successfully by inspecting the gRPC/Python menu shown in the previous slide. This should show the **Start** button greyed out and display the port number the server is running on.



```
Microsoft Windows [Version 10.0.22631.5039]
(c) Microsoft Corporation. All rights reserved.

C:\Oasys 22>./reporter22_0_x64.exe -grpc50053
```



# Starting a gRPC server

## 3. Directly from a Python script

- Use the gRPC package “start” method to initiate a REPORTER session which automatically creates and connects to a gRPC server on the default port;
- This will also start REPORTER in batch mode;
- This method provides you with the same freedom as starting REPORTER via the command line, e.g. you can supply any arguments you wish.

```
example.py
File Edit View

import Oasys.REPORTER as re

connection = re.start(abspath="C:\\Oasys 22\\reporter22_0_x64.exe")

t = re.Template()
p = re.GetPage(0)

i = re.Item(p, re.Item.TEXT, "example")

i.text = "Example"
i.fontSize = 48
i.textColour = re.Colour.Red()

re.LogPrint("Hello")

t.Close()

re.terminate(connection)
```

```
connection = re.start(abspath="C:\\Oasys 22 Debug\\reporter22_0_x64.exe",
args=["-log=C:\\Oasys 22 Debug\\logExample.txt"])
```

# Connecting to a gRPC server

- The third method of starting a gRPC server for REPORTER also automatically handles the connection to it from the Python script.
- The first 2 methods of starting a gRPC server are external to any Python scripts, and therefore require a different approach to connect the already running REPORTER instance to the Python script:
  - This is easily done by using the gRPC package “connect” method and by supplying the port number on which the server was started.

```
connection = re.connect(port=50053)
```

```
example.py
File Edit View

import Oasys.REPORTER as re
connection = re.connect(port=50053)

t = re.Template()
p = re.GetPage(0)

i = re.Item(p, re.Item.TEXT, "example")

i.text = "Example"
i.fontSize = 48
i.textColour = re.Colour.Red()

re.LogPrint("Hello")

t.Close()

re.terminate(connection)
```

# Contact us

## Global / UK

T: +44 121 213 3399

E: [dyna.support@arup.com](mailto:dyna.support@arup.com)

## India

T: +91 40 69019723 / 98

E: [india.support@arup.com](mailto:india.support@arup.com)

## China

T: +86 21 3118 8875

E: [china.support@arup.com](mailto:china.support@arup.com)

## USA

T: +1 415 940 0959

E: [us.support@arup.com](mailto:us.support@arup.com)

Subscribe to  
our newsletter:



Follow us on:



@Oasys LS-DYNA  
Environment



@Oasys LS-DYNA  
Environment



@Oasys



@Oasys

<https://www.oasys-software.com/dyna/>