# gRPC LabVIEW installation and usage on Linux RT HowTo (V.0.2.3):

Tested on environment:

Server tested on:

NI-PXIe-8840Quad-Core-030EEB7A

NI Linux Real-Time x64 4.14.146-rt67-cg-8.6.0f0-x64-48 (20.6)

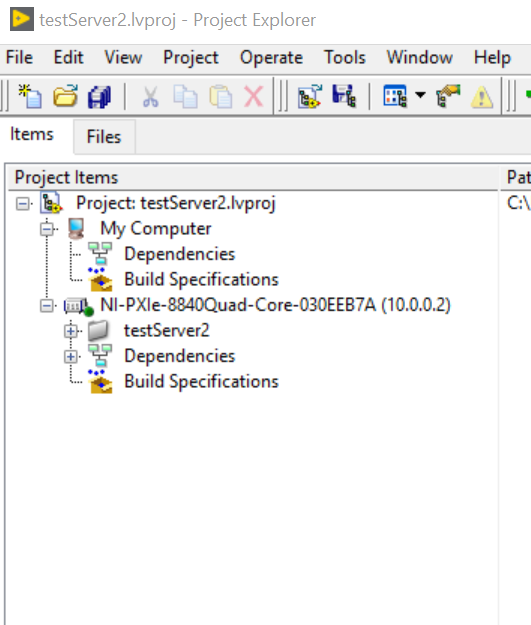
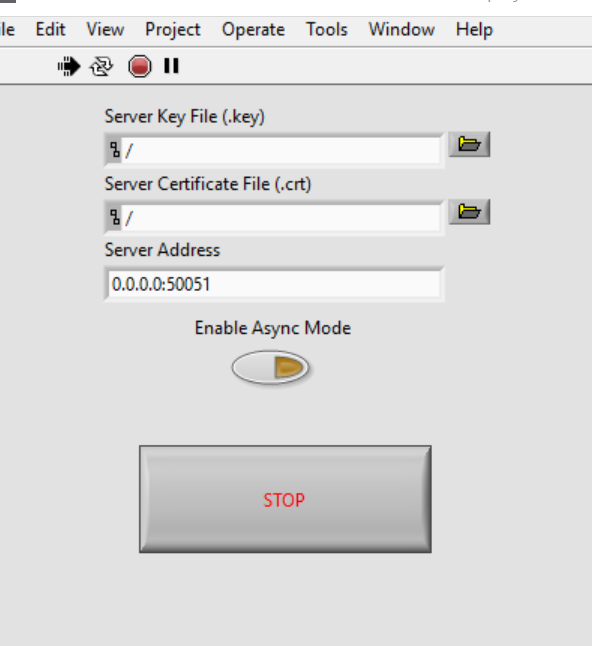
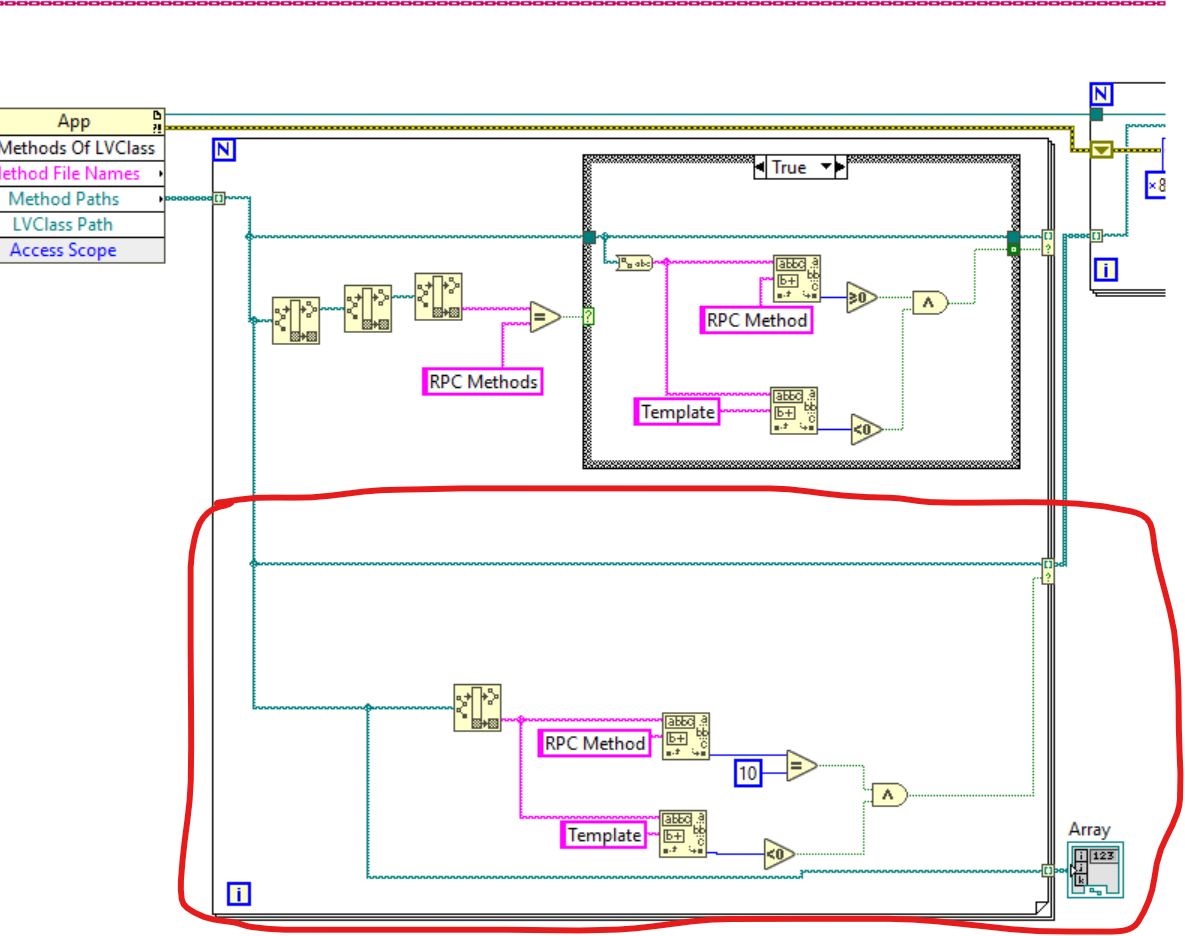
Client code using Python 3.7

# Setup a server

## Overview worksteps:

1. Download gRPC LabVIEW template from <https://github.com/ni/grpc-labview>
2. Download or create Proto file.
3. Use gRPC template to generate server and VI:s from Proto-file
4. Generate Python client using the compiled python output from Proto-file

## Detailed worksteps:

1. Download gRPC LabVIEW
   1. [https://github.com/ni/grpc-labview/releases/tag/v0.2.3 grpc-labview.zip](https://github.com/ni/grpc-labview/releases/tag/v0.2.3%20grpc-labview.zip)
   2. Unzip grpc-labview.zip to an arbitrary directory
2. Download or create Proto fil
   1. Option1. In the downloaded repo, there is an example server containing a proto file and a client file
   2. Option2. A protofile and a prebuilt client can be found <https://grpc.io/docs/languages/python/quickstart/>
   3. Build you own proto file https://developers.google.com/protocol-buffers/docs/overview
3. Generate server and VI:s from Proto-file
   1. Create a LabVIEW project that you can generate the server to
   2. Open LabVIEW gRPC.lvproj from LabVIEW gRPC directory
   3. Follow the guide <https://github.com/ni/grpc-labview/blob/master/docs/ServerCreation.md> and choose the desired Proto file  
        
      ---FOR LINUX RT---
   4. Move the generated server to your real time target  
      
   5. Put your \*.SO file on the RT target and check that Get Server DLL Path.vi points to it
   6. Write the server address and port  
        
      ---For Sync / Sequential execution ---
   7. Press “play” to use non Async (still async but sequential)  
        
        
      ---For Async execution ---
   8. To use Async with interactive mode
      1. Make a change to Start Async. The change to is marked with red in the image below.  
         
      2. Select your server (testServer2 from the above image) and deploy all (this enables dynamic references to work)
      3. Press “play”
4. Generate or use client t python
   1. Generate needed code
      1. python -m grpc\_tools.protoc -I PathToProtosDir --python\_out=. --grpc\_python\_out=. Path/myProtoFile.proto
      2. For more information: python -m grpc.tools.protoc -h
   2. Write client, see example <https://grpc.io/docs/languages/python/quickstart/> or <https://github.com/ni/grpc-labview/blob/master/examples/query_server/Clients/python/queryserver.py>