1. Clone git repository Gridlab-d <https://github.com/1scotthp/gridlab-d>
2. Clone git repository Omnet++ <https://github.com/1scotthp/GLD-Omnet->
3. Download MSYS2 (if not already installed) and follow directions here <https://www.msys2.org/>
4. Follow directions for “Building gridlab-d from source”

<http://gridlab-d.shoutwiki.com/wiki/Building_GridLAB-D_on_Windows_with_MSYS2>

1. Update Environment path variables as described in link above (very important to do this correctly)
2. If you want to use eclipse, follow directions from

<http://gridlab-d.shoutwiki.com/wiki/MinGW/Eclipse_Installation>

but only from “Running gridlab-d on Eclipse” on. The instructions for compiling and building no longer work

Testing co-simulation

1. Open cmd window and run gridlab-d in server mode (and verbose for troubleshooting). Make sure the .glm file being used is set to pause at the beginning of the simulation so that Omnet++ can control the simulation progress.
2. Build and run Omnet++ simulation

**Networked Distribution Grid Simulator**

This project aims to develop a networked grid simulator by integrating the GridLAB-D power distribution system simulator with OMNeT++ cyber communication network simulator. The idea is to achieve a low-level integration of the two simulators while supporting time synchronization and timed data exchange between them. It is expected that significant computational performance can be achieved by eliminating the overhead of an integration middleware such as HLA-RTI or other co-simulation frameworks.

**Implementation Architecture**

OMNeT++ is a discrete-event simulator, which can provide up to nanosecond of time-stepping and accuracy. On the other hand, GridLAB-D is a steady-state and time-stepped simulator. In GridLAB-D the step-size is normally supported to the minimum of 1 second. However, using its *delta-mode* the step-size could be further reduced.

For time-synchronization between GridLAB-D and OMNeT++, the OMNeT++ acts as the main controller of time-progression and the GridLAB-D simulator is progressed periodically to catch-up with OMNeT++. This essentially means that the simulators are progressed forward in time in a lock-stepped manner.

For exchanging messages between the two simulators, we implemented a special module in OMNeT++ that serves as a broker for relaying messages from and to GridLAB-D. We have come up with three distinct ways of achieving this:

* The data exchange with GridLAB-D is accomplished by executing it in *server mode* where it can receive object parameter updates via HTTP requests and can be queried for their values as well using HTTP requests. This is the easiest method to implement the integration using the server mode of GridLAB-D.
* A more efficient implementation (performance wise) might be to actually implement a brand new "ioInterface" module (requires C++) responsible for reading and updating objects.
* As both GridLAB-D and OMNeT++ are in C++, probably the most efficient implementation might involve directly linking them into a single executable.