GridDyn API Users Guide
Version 0.5
Friday 12th August, 2016

Philip Top Ph.D.

This work was performed under the auspices of the U.S. Department of Energy by
Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.
LLNL-TR-XXXXXXX

Contents

1 Introduction

GridDyn is a power system simulator developed at Lawrence Livermore National Laboratory. The name is a concatenation of Grid Dynamics, and as such usually pronounced as "Grid Dine". This document details some of the common interfaces to the GridDyn simulation object itself, as well as highlighting some of the input and output functions associated with the simulation object. For more details on the actual functions and calls and objects used throughout GridDyn the user is encouraged to explore the Doxygen produced documentation for a more interactive description.

2 Overview

2.1 gridDynSimulation class

Principle control of a simulation is done through the gridDynSimulation object. The simulation object is accessed in C++ code by including the "griddyn.h" header file. This object contains the necessary functionality to run a simulation, contain all the power grid components. Additional IO capabilities are provided by the "gridDynFileOperations.h" header file and the "gridDynFileInput.h" header files. These additional files give access to a the IO libraries for importing models and creating files from the simulation results.

2.2 gridDynRunner

The gridDynRunner object is a simple wrapper around a gridDynSimulation object to help with inclusion in other simulation environments.

2.3 Other Objects

Other objects and models can be created and some modeling could be done with them but they are not intended to be used generally. Documentation for them can be found in the Doxygen documentation and in the Future Developers Guide(In Development).

2.4 future API's

Work is ongoing to wrap GridDyn in an FMU for co-simulation object. This will include a shared object library containing Griddyn and an interface to access it, as well as some additional functionality required to interface as an FMU. Some future ideas include adding a Python interface but the concept is still in planning stages.

3 GridDynSimulation Glass