Topology Manager

This project is made using C#, I chose C# because I had experience with it, because it has several powerful libraries to deal with JSON objects and finally because visual studio contains automated testing and code analysis tool that are optimized for this language and an easy way to build and deploy my project.

The program can also be run, and the APIs can be used on a console program that contains a “help” command to show how the APIs can be called on the console.

Automated Testing is made using a separate project which several runs tests on the main one, Output of the code analysis is included in the repo.

Classes:

-Topology_API

A class containing all the other APIs as its methods.

-Topology

A nested class designed to contain the converted JSON Topology objects from the files, it expects a string with name “id” and JsonArray with the name “components”.

-Component

A class used in the “Topology” class to contain the converted JSON objects from it JsonArray mentioned in “Topology” class, it expects 2 strings with names “id” and “type”, a Dictionary object “netlist” and an

arbitrary named Dictionary object that contains values describing the component's physical properties.

APIs:

All APIs take - as input – the reference of the list "topologies" which keeps track of all the topologies in the memory during runtime, it is initialized as an empty list of class "Topology".

-ReadJson(filename,topologies)

This API takes as input a string describing the path to the json file containing the topologies, it then converts the file contents into a list nested objects belonging to class "Topology", returns a boolean value indicating success or failure.

-WriteJson(TopologyID,topologies)

This API takes the ID of the topology we want to save on a file and creates a new file to write on or overwrites an existing file with the serialized version of the selected topology, returns a boolean value indicating success or failure.

-QueryTopologies(topologies)

Writes all saved topologies on the consoles and returns the topologies list.

-DeleteTopology(TopologyID,topologies)

This API takes the ID of the topology we want to delete and removes it form the topologies list, returns a boolean value indicating success or failure.

-QueryDevices(TopologyID,topologies)

This API takes the ID of the topology we want to query the devices from, returns a list of “Component” objects.

-QueryDevicesWithNetlistnode(TopologyID,Netlistnode,topologies)

This API takes the ID of the topology we want to query the devices from and the name of the node on which all devices are connected to, returns a list of “Component” objects of which their netlist Dictionary contains the value of Netlistnode.