# class Grid

**Public:**

Functions:

* *Grid*()
* *Grid*(string catalogFilepath, string componentsListFilepath)
* ~*Grid*()
* gridBus\* *newBus*()
* loadNode\* *newLoad*()
* sourceNode\* *newSource*()
* filterNode\* *newFilter*()
* esmNode\* *newESM*()
* void *addNewCatalog*()
* void *addNewComponent*()
* int *numBuses()*;
* int *numLoads()*;
* int *numSources()*;
* int *numFilters()*;
* int *numESMs()*;
* int *numEdges()*;

Variables:

Structs:

* struct *busListElement*
  + gridBus\* *bus*
  + double *busVoltage*

**Private:**

Functions:

* int *findNumCatalogEntries* (string catalogFilepath)
* int *findNumComponents* (string componentsListFilepath)
* Qstring *newName(Qstring type)*

Variables:

* vector<busListElement\*>\* *busList*
* vector<loadNode\*>\* *loads*
* vector<sourceNode\*>\* *sources*
* vector<filterNode\*> *filters*
* vector<esmNode\*>\* *ESMs*
* vector<gridEdge\*>\* *edges*;
* vector<int>\* *componentsList*
* vector<component\*>\* *activeComponents*
* vector<int>\* *Catalog*
* commonModeGrid\* *cmEqModel*

# class gridNode

**Public:**

Functions:

* *gridNode*()
* ~*gridNode*()
* bool *validityCheck*()
* string *getType()*
* void *setType(*std::string newType*)*
* string *getName ()*
* void *setName(*std::string newName*)*

**Private:**

Functions:

* void *loadFromDataFile*(string filepath)

Variables:

* int *serialNumber*
* string *name*
* string *thumbnailImagePath*
* string *type*

## class gridEdge::gridNode

**Public:**

Functions:

* *gridEdge*()
* ~*gridEdge*()
* bool *validityCheck*()

**Private:**

Functions:

* void *loadFromDataFile*(string filepath)

Variables:

* double *voltageA*
* double *breakerRating*
* int *numPhasesA*
* int *numPhasesB*
* double *conduit\_Diameter*
* double *conduit\_Capacitance*
* double *conduit\_Resistance*
* double *conduit\_Inductance*
* bool *pwrTypeA*
* bool *pwrTypeB*
* double *converter\_Capacitance*
* double *converter\_Resistance*
* double *converter\_Inductance*
* bool *transformer\_3pTypeA*
* bool *transformer\_3pTypeB*
* double *transformer\_Capacitance*
* double *transformer\_Resistance*
* double *transformer\_Inductance*

## class gridElement::gridNode

**Public:**

Functions:

* *gridElement*()
* ~*gridElement*()
* bool *validityCheck*()

**Private:**

Functions:

Variables:

### class sourceNode::gridElement

**Public:**

Functions:

* sourceNode()
* ~sourceNode()
* bool validityCheck()

**Private:**

Functions:

Variables:

### class loadNode::gridElement

**Public:**

Functions:

* loadNode()
* ~loadNode()
* bool validityCheck()

**Private:**

Functions:

* void loadFromDataFile(string filepath)

Variables:

* int serialNumber
* string name
* string thumbnailImagePath

### class esmNode::gridElement

**Public:**

Functions:

* esmNode()
* ~esmNode()
* bool validityCheck()

**Private:**

Functions:

* void loadFromDataFile(string filepath)

Variables:

### class filterNode::gridElement

**Public:**

Functions:

* filterNode()
* ~filterNode()
* bool validityCheck()

**Private:**

Functions:

Variables:

## class gridBus::gridNode

**Public:**

Functions:

* *gridBus()*
* *~gridBus()*
* bool *validityCheck()*
* double *getVoltage()*

**Private:**

Functions:

* void *loadFromDataFile*(string filepath)

Variables:

* vector<double>\* *busTrace*
* vector<double>\* *exportTrace*
* int *numBreakers*
* double *bus\_Capacitance*
* double *bus\_Resistance*
* double *bus\_Inductance*
* double *­voltage*
* vector<loadNode\*> *loads*
* vector<sourceNode\*> *gensets*
* vector<gridBus\*> *childSWBDs*
* vector<gridBus\*> *equalSWBDs*
* vector<gridBus\*> *parentSWBDs*
* vector<filterNode\*> *filters*
* vector<esmNode\*> *ESMs*

# class component::Qlabel

This class is used for vsualizing the grid in the drag and drop viewport. Each instance contains a reference to a node within the grid

**Public:**

Functions:

* *Component()*
* *Component(gridNode\* inputNodeRef)*
* *~Component()*
* Std::string *checkNodeType*()
* gridNode *getNodeData*()
* void *updateThumbnail()*
* void *updateName()*

**Private:**

Functions:

Variables:

* string *name*
* gridNode\* *nodeRef*
* Qpixmap *thumbnail*

# class commonModeGrid

**Public:**

Functions:

* *commonModeGrid*()

**Private:**

Functions:

Variables: