1 File Specification

Graphs are stored in a json serialized node link format. This format benefits from native support from NetworkX, as well as easy compatibility with D3.js and other packages for working with network data. Furthermore, the format is simple enough to be directly de-serialized into a dictionary or map and used right away.

1.1 Node Link File structure

Each graph or multi-graph is represented as a single json file. This file contains a nested map formatted as follows:

Node Link JSON file structure:

```
{"directed": true / false,
"multigraph": true / false},
"graph": {attribute_1:value,...,attribute_g:value},
"nodes": [{"id":node_id,attribute_1:value,...,attribute_n:value},
          ...,
          {"id":node_id,attribute_1:value,...,attribute_n:value}],
"links": [{"source":node,"target":node,"key":id,attribute_1:value,...,attribute_e: value},
          ...,
          {"source":node,"target":node,"key":id,attribute_1:value,...,attribute_e: value}]}
```

Keywords the are reserved are in quotes. Every json file will have the **directed** and **multigraph** flags. The **graph** field contains a dictionary of all graph attributes.

Nodes
The **nodes** field contains a list of dictionaries. Each of these inner dictionaries has a required **id** field - which stores that node's unique string or integer id - and all associated attributes and their values.

Edges (Links)
The **links** field also contains a list of dictionaries. Each of these inner dictionaries has a required **source** field and a **target** field - which stores the unique string or integer id of the nodes this edge connect - and fields for each edge attribute and their values. They also have a **key** field required only if this is a multi-graph, which defines the edge set this edge belongs to. If this edge is weighted, a **weight** attribute must be set. Note that the **weight** keyword is reserved for the edgeweight and may be treated differently than a generic attribute.