**1/29/2015 Proposed JSON Structure: Assumptions**

**1.)** **Footnotes are placed at end of a line of text (subtitle, row/column names, etc)**

*Example:*

Currently supported: "A 2013 column name1"

Currently not supported: "A 20131 column name"

*Reason for limitation:*

Footnotes are bound to elements (headers, title, etc), not to specific text/ characters.

*Possible solution:*

If we decide we need mid-title footnote functionality (as an example), I currently propose a “catch-all” footnote type that binds to specific CSS classes or id’s, so we could support something like:

"title":{

"label":"A <span class = “mid-title-footnote”>2013</span> title”

}

and then, in the footnotes object:

{

"type": "class",

"class": "mid-title-footnote",

"text": "Some footnote text"

}

**2.) Row/column headers must be unique**

*Example:*

Currently supported:

|  |  |  |  |
| --- | --- | --- | --- |
| 2013 | | 2014 | |
| Case1 | Case2 | Case1 | Case2 |

Currently not supported:

|  |  |  |  |
| --- | --- | --- | --- |
| 2013 | | 2013 | |
| Case1 | Case2 | Case1 | Case3 |

*Reason for limitation:*

Currently row/column combinations are referenced in the metadata as:

“2013 >>> Case1”

In order to bind footnotes to headers or single cells (using row/column header combinations)

*Possible solution:*

If this uniqueness assumption cannot be met, we’d have to use a reference system by index, i.e.

|  |  |  |  |
| --- | --- | --- | --- |
| [0] | | [1] | |
| [0][0] | [0][1] | [1][0] | [1][1] |

Which has the advantage of uniqueness, but the disadvantage of being less human-readable.

**3.)** **Hierarchical rows must only contain data for the deepest headers**

*Example:*

Currently supported:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parent |  | | | |
| Child |  | | | |
| Grandchild | 1 | 2 | 3 | 4 |
| Grandchild | 5 | 6 | 7 | 8 |

Currently not supported:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parent | 1 | 2 | 3 | 4 |
| Child |  |  |  |  |
| Grandchild | 1 | 2 | 3 | 4 |
| Grandchild | 5 | 6 | 7 | 8 |

*Reason for limitation:*

The “data” object does not contain any formatting data, e.g. blank rows. It assumes that, if the examples above were full tables, then row 0 would be the first grandchild and row 1 would be the second grandchild. That is, data rows are assumed to be the deepest elements in the row trees, and data columns are assumed to be the deepest elements in the column trees.

*Possible solution:*

Inserting “null” rows or columns in the “data” object could solve this problem, but it would mean that data and metadata were linked in relatively unpleasant, circular fashions. As is, in the “data” object, data[i][j] refers to the i’th row of data and the j’th column of numbers, with blank rows determined programmatically.

**4.)** **Title objects need some fixed format**

*Example:*

We can hard code in deeply nested title structures, for example, but currently the title/ subtitle structure is not as flexible or robust as the row/ column header structure

*Reason for limitation:*

Current subtitle footnote looks like:

{

"type": "subtitle",

"text": "Includes AMT liability on Form..."

},

Which of course binds to the subtitle, and assumes all tables with subtitles have a single subtitle.

*Possible solution:*

We could come up with a flexible, hierarchical subtitle structure, but it might

make the most sense to determine title/ subtitle use cases and hard code

them in?

**Note:**

*The “conditional styles” object is just a skeleton right now, but I tried to give some ideas about how we could bind styles to:*

1. *Any value below 0.05 (which would also need a footnote)*
2. *Columns 1 and 2 (0 indexed) in our example table.*

*Again, I like the idea of keeping this object separate, as opposed to including styles in the column objects or data object, if that makes sense.*