FACTSET 〉 SEE THE ADVANTAGE

# STACH FORMAT DOCUMENTATION - VERSION 2.0 DEVELOPER'S MANUAL AND REFERENCE MARCH 2021

Document Version 2

FACTSET 〉 SEE THE ADVANTAGE

# STACH Format Documentation – Version 2.0

## Contents



## 1. Introduction

Given the lack of industry standard, FactSet's engineering team developed the STACH format as the JSON standard for representing tabular data. As this is a new standard, there will be ongoing changes to the format. FactSet has come up with the second version of STACH that has enhanced consumption capabilities through seamless integration of JSON data by allowing our clients greater ability to pick up JSON formats suitable for specific client use cases. Along with, column organized format already supported with earlier versions, STACH supports the row organized and the simplified row formats. To simplify integration around this standard, FactSet will provide language classes to represent the format for the commonly used languages.

For more on the comparative advantages of column vs row organized formats refer the documentation here. Please refer to Appendix of the API documentation for an example of STACH V2 formats.

## 2. Column Organized: Format Features

The column organized format is best for manipulation. For instance, the column organized format does not flatten the columns that make up the collapsed column into one column like the row organized format does. This means that the data is still available for manipulation, such as processing the data, creating a rendered chart, or creating a rendered table. The goal of the column organized format is to fully describe the complexity of render-able tables and to serialize to the smallest size possible.

### 2.1 Column Organized Package

The Column Organized response package houses all associated tables in each response. The version is used to validate which version of the STACH schema is expected. This property is most useful when data is stored persistently. Upon reading the data, the version can be validated against the SDK used to ensure compatibility. For this reason, when persisting STACH data, it is recommended to store a Package.

```
{
  "primaryTableIds": [
    "exampleTableId"
  ],
  "tables": {
    "exampleTableId": {}
  }
}
```

### 3.1 Tables

This property details all tables in the response. It is a dictionary mapping the table identifiers to its contents.

```
…
"primaryTableIds": [
    "mainTableId"
],
"tables": {
      "mainTableId": {
            "definition": {…},
            "data": {…}
      },
      "headerTableId": {
            "definition": {…},
            "data": {…}
```

```
        }
}
…
```

In the above example, primary table can be found in the "`primaryTableIds`" property. The contents of the primary table can be found in the "`tables`" object. Tables "`mainTableId`" and "`headerTableId`" are shown for illustration and should not be hardcoded, an actual STACH response would have GUI Ids to represent the tables respectively.

## 2.2 Table Schema

This property defines the table schema including header table location and its columns.

```
…
"definition": {
        "headerTableId": "headers",
        "columns": […]
}
…
```

## 2.3 Column Schema

This property defines the column schema. A column can either be a dimension column or a simple data column. A dimension column contains additional information about another peer column.

```
…
"definition": {
        "headerTableId": "headers",
        "columns": [{
                        "id": "c_continent1",
                        "name": "continent1",
                        "description": "Continent 1",
                        "type": "STRING",
                        "isDimension": true,
                        "parentId": "c_region"
                },{
                        "id": "c_continent2",
                        "name": "continent2",
                        "description": "Continent 2",
                        "type": "DOUBLE",
                        "format": {
                                "format": "{0:0.00;-0.00;--}",
                                "nullFormat": "--",
```

```
                        "halign": "RIGHT"
                    }
            },
            {…}
        ]
},
"data": {…}
…
```

### Hidden Columns

A column of data can represent metadata which can be used for interactive rendering (e.g., for a tooltip or hyperlink). If the column is hidden in the report, the "`isHidden`" property is set to true. Hidden columns serve as inputs to a reference column. For example, Column3 is defined as hiddenColumn1+hiddenColumn2.

### Data Types

STACH V2 is not opinionated about what data types can be represented in a table and leaves it open ended. The intent is to standardize data types independently. It means that any value, including strings, numbers, Booleans, objects, arrays, and nulls can be represented in a table's cell. The type property was changed from a DataType to a string as a result.

### Column Format

The "nullFormat" property denotes the null value for a data item. The null value placeholder for each data type is "null". The "format" property is used to render a raw value into a human readable string. Refer Microsoft's Formatting Types for what strings can be used. The "halign" and "valign" properties are used to specify how the data should be aligned within the table cell.

## 2.4 Table Data

This property defines the table contents.

```
…
"definition": {…},
"data": {
        "rows": […],
        "columns": {…},
        "metadata": {…}
}
…
```

## 2.5 Row Data

This property defines the row contents.

```
…
"data": {
       "rows": [{
                     "id": "r_0"
              },
              {…},
              …
       ],
       "columns": {…},
       "metadata": {…}
}
…
```

## 2.6 Column Data

This property defines the column data.

```
…
"data": {
       "rows": […],
       "columns": {
              "c_region": {…},
              "c_continent1": {
                           "values": [
                                  "null",
                                  "null",
                                  "null",
                                  "Americas",
                                  "Asia Pacific",
                                  "Europe",
                                  "Middle East and Africa",
                                  "null",
                                  "null",
```

```
                                    "null",
                                    "null",
                                    "null",
                                    "null",
                                    "null",
                                    "null"
                             ]
                },
                "c_continent2": {…},
                "c_f0": {…},
                "c_b0": {…},
                …
        },
        "metadata": {…}
}
…
```

## 2.7 Metadata

Metadata helps in providing the information beyond the raw data, e.g. accounts, benchmarks, dates etc. values used while creating the calculation. It can be attributed to the entire table, or to a particular column, row, or a cell.

The locations object specifies to what level the metadata is applicable. The items object defines the actual metadata item whose reference is present in location object. Each item key can either be a value or a reference to another table in the JSON.

```
…
"data": {
        "rows": […],
        "columns": {…},
        "metadata": {
                "items": {
                        "m_h1": {
                                "name": "Currency",
                                "stringValue": "USD"
                        },
                        "m_h2": {
                                "name": "Benchmark",
                                "stringValue": "BENCH:SP50"
                        },
                        "m_fund0Url": {
                                "name": "fund0Url",
```

```
                        "refValue": {
                              "tableId": "main",
                              "columnId": "c_fund0Url"
                        }
                  },
                  "m_regionUrl": {…},
                  "m_rm": {…}
            },
            "locations": {
                  "table": [
                        "m_h1",
                        "m_h2"
                  ],
                  "columns": {
                        "c_f0": {
                              "ids": [
                                    "m_format",
                                    "m_fund0Url"
                              ]
                        },
                        "c_b0": {
                              "ids": [
                                    "m_format"
                              ]
                        },
                        "c_region": {
                              "ids": [
                                    "m_regionUrl",
                                    "m_rm"
                              ]
                        }
                  }
            }
      }
}
…
```

## 2.8 Column Headers

Column Header tables contain additional information about columns in the primary tables. Every row in a column header table links to a column in the primary table.

```
…
"tables": {
       "main": {
              "definition": {
                     "headerTableId": "headers",
                     "columns": […]
              },
              "data": {…}
       },
       "headers": {
              "definition": {
                     "columns": […]
              },
              "data": {
                     "rows": […],
                     "columns": {…},
                     "metadata": {…}
              }
       }
}
…
```

### Example

| | 2017 | | | | | |
|---|---|---|---|---|---|---|
| | Exposure | | Total Risk | | Contribution to Risk | |
| | Fund | Bench | Fund | Bench | Fund | Bench |
| Max | 88.3 | 89.62 | 17.17 | 15.67 | 86.07 | 89.18 |
| Total | 100 | 100 | 8.6 | 8.38 | 100 | 100 |
| Developed Markets | 88.3 | 89.62 | 8.51 | 8.39 | 86.07 | 89.18 |
| Americas | 46.03 | 54.98 | 10.21 | 9.63 | 50.74 | 61.23 |
| Asia Pacific | 9.09 | 12.48 | 11.94 | 10.06 | 7.37 | 9.77 |
| Europe | 33.15 | 21.95 | 8.71 | 8.28 | 27.96 | 17.93 |
| Middle East and Africa | | 0.21 | | 13.58 | | 0.24 |
| Emerging Markets | 8.71 | 7.76 | 14.12 | 10.65 | 10.48 | 8.17 |
| Americas | 0.3 | 1.12 | 22.7 | 15.44 | 0.36 | 1.41 |
| Asia Pacific | 8.41 | 5.41 | 14.22 | 11.02 | 10.12 | 5.47 |
| Europe | | 0.33 | | 12.78 | | 0.25 |
| Middle East and Africa | | 0.89 | | 14.92 | | 1.03 |
| Frontier Markets | 0.36 | | 17.17 | | 0.19 | |
| Other | | 0.53 | | 15.67 | | 0.59 |
| [Cash] | 0.15 | | 8.45 | | 0.06 | |

The above table can have its own column header table rotated 90° as shown below:

| Year | Category | Portfolio Type |
|---|---|---|
| 2017 | Exposure | Fund |
| 2017 | Exposure | Bench |
| 2017 | Total Risk | Fund |
| 2017 | Total Risk | Bench |
| 2017 | Contribution to Risk | Fund |
| 2017 | Contribution to Risk | Bench |

## 2.9 Version

This property defines the version of STACH format.

```
…
"version": "2.0"
…
```

# 3 Row Organized: Format Features

The purpose of the row organized format is to make it as simple as possible to consume tabular data. It is easier to deserialize or utilize the schema.

## 3.1 Package

STACH aims to represent complex Portfolio Analysis (PA) workflows using tabular representation with the following characteristics:

- Tables store derived data and its associated metadata.
- Package houses all associated tables in each response
- Version is used to validate which version of the STACH schema is expected

```
{
  "version": "2.0",
  "tables": {
    "exampleTable": {}
  }
}
```

## 3.2 Tables

This details all tables available within the response with unique ids. A row organized table is made up of three pieces; definition & headersDefinition which includes primary and header TableDefinition respectively along with the data section which represents data for the entire table including both header and body rows. It also houses other properties like tableMetadata and columnMetadata.

```
…
"tables": {
  "35f25fb9-0c7c-4964-a828-54f77c5f7894": {
       "definition": {
             "columns": […]
       },
       "headersDefinition": {
             "columns": […]
       },
      "data": {
        "rows": […],
       "tableMetadata": {…},
```

```
        "columnMetadata": {…}
      }
   }
}
…
```

### 3.3 Definition

This table defines the primary table definition. It contains an array of ColumnDefinition that represents the columns.

```
…
      "definition": {
        "columns": [
          {
            "id": "c_function",
            "name": "function",
            "description": "Function",
            "type": "string",
            "isDimension": true,
            "nextSiblingId": "c_region"
          },
          {
            "id": "c_region",
            "name": "region",
            "description": "Region",
            "type": "string",
            "isDimension": true
          },
          {
            "id": "c_continent1",
            "name": "continent1",
            "description": "Continent 1",
            "type": "string",
            "isDimension": true,
            "parentId": "c_region"
          },
          {
            "id": "c_continent2",
            "name": "continent2",
            "description": "Continent 2",
            "type": "string",
```

```
            "isDimension": true,
            "parentId": "c_region"
          },
          {…}
        ]
      },
…
```

### 3.4 Header Definition

This table details the headers table definition. It contains an array of header column definitions.

```
…
      "headersDefinition": {
        "columns": [
          {
            "id": "c_year",
            "name": "year",
            "description": "Year",
            "type": "int32",
            "isDimension": true
          },
          {
            "id": "c_category",
            "name": "category",
            "description": "Category",
            "type": "string",
            "isDimension": true,
            "parentId": "c_year"
          },
          {
            "id": "c_portType",
            "name": "portType",
            "description": "Portfolio Type",
            "type": "string",
            "isDimension": true,
            "parentId": "c_category"
          }
        ]
      },
…
```

### 3.5 Data

The data table houses an array of row objects representing each row in the actual table. It represents data for the entire table including both header and the actual data values. The header rows are identified with “rowType” as “Header” and the data rows are represented as “Values” with key-value pair where the key corresponds to the column’s id.

All metadata items associated with the entire table or row or column as well as cells reside in the Data section. The headerCellDetails lists cell properties like rowspan, colspan, source, columnIndex. The cellDetails lists properties like group_level, cell_definition, cell_metadata. The tableMetadata is a map of metadata at table location.


```
…
      "data": {
        "rows": [
          {
            "rowType": "Header",
            "cells": [
              "Function",
              "Region",
              "Continent 1",
              "Continent 2",
              2017,
              "Abbr",
              2017
            ],
            "headerCellDetails": {
              "0": {
                "rowspan": 3,
                "source": "PRIMARY"
              },
              "1": {
                "rowspan": 3,
                "source": "PRIMARY",
                "columnIndex": 1
              },
              "2": {
                "rowspan": 3,
                "source": "PRIMARY",
                "columnIndex": 2
              },
              "3": {
                "rowspan": 3,
                "source": "PRIMARY",
                "columnIndex": 3
```

```
            },
            "4": {
              "colspan": 2,
              "source": "HEADERS"
            },
            "5": {
              "rowspan": 3,
              "source": "PRIMARY",
              "columnIndex": 6
            },
            "6": {
              "colspan": 4,
              "source": "HEADERS"
            }
          }
        },
      {...},
        {
          "values": {
            "c_function": "Max",
            "c_region": null,
            "c_continent1": null,
            "c_continent2": null,
            "c_f0": 88.3,
            "c_b0": 89.62,
            "c_contAbbr": null,
            "c_f1": 17.17,
            "c_b1": 15.67,
            "c_f2": 86.07,
            "c_b2": 89.18
          }
        },
        {
          "values": {
            "c_function": "Total",
            "c_region": null,
            "c_continent1": null,
            "c_continent2": null,
            "c_f0": 100,
            "c_b0": 100,
            "c_contAbbr": null,
            "c_f1": 8.6,
```

```
            "c_b1": 8.38,
            "c_f2": 100,
            "c_b2": 100
          }
        },
        {
          "values": {
            "c_function": null,
            "c_region": "Developed Markets",
            "c_continent1": null,
            "c_continent2": null,
            "c_f0": 88.3,
            "c_b0": 89.62,
            "c_contAbbr": null,
            "c_f1": 8.51,
            "c_b1": 8.39,
            "c_f2": 86.07,
            "c_b2": 89.18
          },
          "cellDetails": {
            "c_region": {
              "cellMetadata": {
                "m_regionUrl": {
                  "value": "https://www.google.com/search?q=developed+markets"
                }
              }
            },
            "c_f0": {
              "cellMetadata": {
                "m_fund0Url": {
                  "value": "https://www.google.com/search?q=x"
                }
              }
            }
          }
        }
```

# 4 Simplified Row: Format Features

Simplified row format aims at minimizing references and other extraneous formatting information. It keeps the response simpler to use by scaling down on metadata and organizing the crux of definition and data.

## 4.1 Package

```
{
  "version": "2.0",
  "tables": {
    "exampleTable": {}
  }
}
```

## 4.2 Tables

Tables in simplified row format are organized into two pieces, definition, and data. It houses both the primary and header table schema in the definition section rather than organizing them across separate properties as is the case with row organized format.

```
…
"tables": {
  "35f25fb9-0c7c-4964-a828-54f77c5f7894": {
       "definition": {
             "columns": […]
       },
      "data": {
        "rows": […],
       "tableMetadata": {…}
      }
   }
}
…
```

### 4.3 Definition

This property houses both the primary and header table schema rather than organizing them across as separate properties as is the case with row organized format.


```
…
"definition": {
      "columns": [{
            "id": "col_0",
            "name": "total0",
            "description": "total0",
            "type": "string",
            "isDimension": true
      },
      {
            "id": "col_1",
            "name": "group1",
            "description": "group1",
            "type": "string",
            "isDimension": true
      },
      {
            "id": "col_2",
            "name": "group2",
            "description": "group2",
            "type": "string",
            "isDimension": true
      },
      {
            "id": "col_3",
            "name": "EndingWeight:FACTSET149:7",
            "description": "Difference",
            "type": "real"
      }…
```

## 4.4 Data

The data table consists of an array of row objects that represents each row of the entire table. Both the header and data rows are represented as “Values” with key-value pair where the key is the column’s id. It excludes cell details as compared to the row organized. Metadata related to the table are encased in tableMetadata.


```
…
"data": {
      "rows": [
       {
            "values": {
                  "col_0": "Total",
                  "col_1": null,
                  "col_2": null,
                  "col_3": null
            }
      },
      {
            "values": {
                  "col_0": "Commercial Services",
                  "col_1": null,
                  "col_2": null,
                  "col_3": 0.882507652252207
            }
      },
      {
            "values": {
                  "col_0": "Commercial Services",
                  "col_1": "Advertising/Marketing Services",
                  "col_2": null,
                  "col_3": 0.0981594725806263
            }
      },
      {
            "values": {
                  "col_0": "Commercial Services",
                  "col_1": "Advertising/Marketing Services",
                  "col_2": "Interpublic Group of Companies, Inc.",
                  "col_3": 0.030606947459382
```

```
                }
        },
        {
                "values": {
                        "col_0": "Commercial Services",
                        "col_1": "Advertising/Marketing Services",
                        "col_2": "Nielsen Holdings Plc",
                        "col_3": 0.0248581269700406
                }
        }
…
```