



# The Galileo Interchange Format

The Galileo Interchange Format is the wire protocol between the galileo module and a dashboard and the disk format of a Galileo Dashboard file. The extension of the file is .gd.json

## Overall Structure

The Galileo Interchange Format is simply the JSONified form of the data structure underlying a Galileo Dashboard. Its overall structure is:

```
{
  "tables": <dictionary of tables>,
  "filters": <dictionary of filters>,
  "views": <dictionary of views>,
  "charts": <dictionary of charts>,
  "morphs": <list of morphs>
}
```

where each dictionary is of the form {<objectName>: <structure>}.

## Morphic Properties

All physical objects (filters, charts, text, shapes, and images) have a field "morphIndex" and a substructure "morphicProperties". The morphIndex gives the z-order of the morph; 0 is the frontmost morph, n the rearmost. No two object should share the same morphIndex; the morphIndex should be a total order on physical objects.

The morphicProperties structure describes its physical properties. The fields of the "morphicProperties" structure are:

Field	Structure	Purpose	Example	Note
fill	Color string	Color Object		
position	Coordinate	Top-left corner position	pt(128, 73)	x = 128, y = 73
extent	Coordinate	Bounding Box width/height	pt(10, 5)	width: 10, height: 5
rotation	real	In radians, clockwise	1.57	rotated 90° clockwise
border	Border object	Substructure with border properties. See below.		
opacity	real	1 = solid, 0 = transparent	0.5	semi-transparent
Clipmode	Clip enum	Handle bounding box overflow	scroll	scroll bars appear

A Color Object is a four-tuple:

Field	Type	Semantics
r	0-1 real	red intensity (0 = none, 1 = full)
g	0-1 real	green intensity (0 = none, 1 = full)
b	0-1 real	blue intensity (0 = none, 1 = full)
a	0-1 real	opacity (0 = transparent, 1 = solid)

A Coordinate is a structure of the form 'pt(x, y)', where x and y are reals.

The Clip enum is one of visible, scroll, or auto. Visible is overflow the bounds, hidden is do not show the overflow, scroll is show scrollbars

A border structure is a dictionary with a number of fields. Each field is a structure of the form {"top": <top>, "bottom": <bottom>, "right": <right>, "left":<left>}, where <top>, <bottom>, <right>, and <left> specify the values for each side of the structure. For example:

{ "width": { "top": 2, "bottom": 2, "right": 4, "left": 4 } } specifies that the width, in pixels, of the top and bottom borders are 2 and the width of the side borders are 4.

The fields are here. The structure refers to the structure of each of the top, bottom, right, and left components.

Field	Structure	Purpose	Example	Note
Width	Real	Border width in pixels	2	Must be > 0
Radius	Real	Side arc radius, as a percentage of side length	10	0 = Straight side 100 = full arc
Type	BorderType	style of the border line	dotted	border composed of dots
Color	Color object	color of the border line		

A BorderType is an enum (as a string). It is one of: none, hidden, solid, dotted, dashed, ridged, double, groove, or inset

## Tables

A table is a dictionary with two entries, columns and rows. A column is a list of entries of the form {"name": <columnName>, "type": <columnType>} where column type must be one of 'string', 'number', 'boolean', 'date', 'datetime', 'timeofday'.

The rows field is a list of lists, where each list represents a row of the table. Each component list *must*:

- Be of the same length as the list of columns
- Be type-compliant with the list of columns. The ith entry in the list must be of the type specified in the ith entry of the column list

## Filters

A Filter is a structure of the form: {"savedFilter": <filterSpecification>, "morphIndex": <index>, "morphicProperties": <properties>}. See the discussion above for morphicIndex and morphicProperties. Each filter specification has a field "part", which gives the URL for the lively.next component used to build the slider. The filter specification is specific to the filter type, and these are given here:

- Slider

Field	Type	Role
part	part URL	Url of the prototype for the filter
columnName	String	Name of the column this filters
minVal	Number	Minimum possible value for this filter
maxVal	Number	Maximum possible value for this filter
value	Number	Current value for this filter
increment	number	distance between consecutive values
type	enum	type of filter

For as Slider, the filter type is always "NumericSelect".

- List, Dropdown

Field	Type	Role
-------	------	------

part	part URL	Url of the prototype for the filter
columnName	String	Name of the column this filters
choices	List	List of choices for this filter
selection		Current selection for this filter
type	enum	type of filter

Selection is the type of the choices of the list; the filter type is always "Select".

- Range, Double Slider

Field	Type	Role
part	part URL	Url of the prototype for the filter
columnName	String	Name of the column this filters
minVal	Number	Minimum possible value for this filter
maxVal	Number	Maximum possible value for this filter
min	Number	Current minimum of the range selected
max	Number	Current maximum of the range selected
increment	number	distance between consecutive values
type	enum	type of filter

The filter type is always "Range".

## Views

A View is an extremely simple structure; it has three components:

Field	Type	Role
table	string	name of the underlying table
filters	list of strings	Unordered list of the names of the filters used to find the rows
columns	list of strings	<i>ordered</i> list of the names of the columns in this view

## Charts

A Chart is also a simple structure. It has four fields:

Field	Type	Role/Notes
chartType	enum	type of the chart (chosen from a supported chart library)
options	object	chart options (library specific)
viewOrTable	string	name of the View or Table that is the data source for the chart
morphIndex	number	order of the chart in the scene (0 = front)
morphicProperties	object	see above

## Morphs

A morph is a simple structure. Since Morphs are not stored in dictionaries, but rather in lists, the name of the morph is in the morph structure. Every morph has four fields:

Field	Type	Role
name	string	name of the morph
type	enum morphType	morph type: list below
morphIndex	number	z-order of the morph
morphicProperties	object	morphic properties

The morph types are Rectangle, Ellipse, Image, and Text. The Image morph has one additional field:

Field	Type	Role
imageUrl	URL	URL of the image (can be a data URL)

The Text morph has one additional field:

Field	Type	Role
textProperties	object	Text-specific properties

The text properties are given here:

Field	Type	Role
fontFamily	string	Name of the font family
fontSize	number	Size of the font, in pts
fontWeight	enum fontWeight	Weight, fine to bold
fontStyle	list of styles	Weight, fine to bold
fontColor	Color Object	text color
padding	number	padding between text and bounding box
textAlign	enum alignment	text alignment
textDecoration	enum decoration	underlined or not
lineWrapping	enum wrapping	whether to wrap text
fixedHeight	boolean	if true, bounding box height independent of tex
fixedWidth	boolean	if true, bounding box width independent of text
textString	string	the text string itself

- A fontWeight is one of "Fine", "Medium", "Bold", "Extra Bold"
- textAlign is one of "center", "left", "right", "justified"
- fontStyle is one of "normal", "italic", "oblique"
- textDecoration is one of "underline" or "none"
- linewrapping is one of "by words", "anywhere", "only by words", "none"