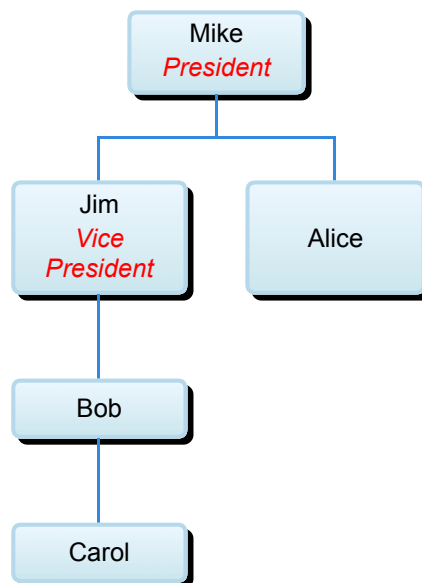# Organization Chart

Overview

Org charts are diagrams of a hierarchy of nodes, commonly used to portray superior/subordinate relationships in an organization. A family tree is a type of org chart.

Example

CODE IT YOURSELF ON JSFIDDLE

```html
<html>
  <head>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader
    <script type="text/javascript">
      google.charts.load('current', {packages:["orgchart"]});
      google.charts.setOnLoadCallback(drawChart);

      function drawChart() {
        var data = new google.visualization.DataTable();
        data.addColumn('string', 'Name');
        data.addColumn('string', 'Manager');
        data.addColumn('string', 'ToolTip');

        // For each orgchart box, provide the name, manager, and tooltip to sl
        data.addRows([
          [{v:'Mike', f:'Mike<div style="color:red; font-style:italic">Presid
```

```
              '', 'The President'],
          [{v:'Jim', f:'Jim<div style="color:red; font-style:italic">Vice Pres
           'Mike', 'VP'],
          ['Alice', 'Mike', ''],
          ['Bob', 'Jim', 'Bob Sponge'],
          ['Carol', 'Bob', '']
        ]);

        // Create the chart.
        var chart = new google.visualization.OrgChart(document.getElementById
        // Draw the chart, setting the allowHtml option to true for the toolt:
        chart.draw(data, {allowHtml:true});
      }
    </script>
     </head>
  <body>
    <div id="chart_div"></div>
  </body>
</html>
```

Loading

The package name is `'orgchart'`.

```
google.charts.load('current', {packages: ['orgchart']});
```

The visualization's class name is `google.visualization.OrgChart`.

```
var visualization = new google.visualization.OrgChart(container);
```

Data Format

A table with three string string columns, where each row represents a node in the orgchart.
Here are the three columns:

- Column 0 - The node ID. It should be unique among all nodes, and can include any
  characters, including spaces. This is shown on the node. You can specify a formatted
  value to show on the chart instead, but the unformatted value is still used as the ID.

- Column 1 - [*optional*] The ID of the parent node. This should be the unformatted value
  from column 0 of another row. Leave unspecified for a root node.

- Column 2 - [*optional*] Tool-tip text to show, when a user hovers over this node.

Each node can have zero or one parent node, and zero or more child nodes.

Custom Properties

You can assign the following custom properties to data table elements, using the `setProperty()` (https://developers.google.com/chart/interactive/docs/reference#DataTable_setProperty) method of `DataTable` (https://developers.google.com/chart/interactive/docs/reference):

Property Name	
selectedStyle	**Applies To:** DataTable row An inline style string to assign to a specific node when selected. You must set the option `allowHtml=true` for this to work, and it must be set before calling `draw()` on the visualization. This overrides the `selectionColor` option for the specified node. Example: `myDataTable.setRowProperty(2, 'selectedStyle', 'background-color:#00FF00');`
style	**Applies To:** DataTable row An inline style string to assign to a specific node. This is overridden by the `selectedStyle` property. You must set the option `allowHtml=true` for this to work, and it must be set before calling `draw()` on the visualization. This overrides the `color` option for the specified node. Example: `myDataTable.setRowProperty(3, 'style', 'border: 1px solid green');`

Configuration Options

Name	
allowCollapse	Determines if double click will collapse a node. **Type:** `boolean` **Default:** `false`
allowHtml	If set to true, names that includes HTML tags will be rendered as HTML.

	Type: `boolean` Default: `false`
color	**Deprecated. Use nodeClass instead.** The background color of the orgchart elements. Type: `string` Default: `'#edf7ff'`
nodeClass	A class name to assign to node elements. Apply CSS to this class name to specify colors or styles for the chart elements. Type: `string` Default: `default class name`
selectedNodeClass	A class name to assign to selected node elements. Apply CSS to this class name to specify colors or styles for selected chart elements. Type: `string` Default: `default class name`
selectionColor	**Deprecated. Use selectedNodeClass instead.** The background color of selected orgchart elements. Type: `string` Default: `'#d6e9f8'`
size	'small', 'medium' or 'large' Type: `string` Default: `'medium'`

Methods

Method	
`collapse(row, collapsed)`	Collapses or expands the node. • `row` - Index of the row to expand or collapse. • `collapsed` Whether to collapse or expand the row, where true mea Return Type: none
`draw(data, options)`	Draws the chart. Return Type: none
`getChildrenIndexes(row)`	Returns an array with the indexes of the children of the given node.

	Return Type `Array.<number>`
getCollapsedNodes	Returns an array with the list of the collapsed node's indexes. **Return Type:** `Array.<number>`
getSelection()	Standard `getSelection()` implementation. Selection elements are a elements. Can return more than one selected row. **Return Type:** Array of selection elements
setSelection(selection)	Standard [setSelection()](https://developers.google.com/chart/interactive/docs/reference#vis) (https://developers.google.com/chart/interactive/docs/reference#vis implementation. Treats every selection entry as a row selection. Suppo of mutiple rows. **Return Type:** none

Events

Name
collapse
onmouseover
onmouseout
select

collapse	Event triggered when `allowCollapse` is set to `true` and the user double clicks on a node with children. **Properties:** `collapsed` - A boolean indicating whether this is a 'collapse' or 'expand' event. `row` - The zero-based index of the row in the data table, corresponding to the node being clicked.
onmouseover	Triggered when the user hovers over a specific row. **Properties:** `row` - The zero-based index of the row in the data table, corresponding to the node being moused over.
onmouseout	Triggered when the user hovers out of a row. **Properties:** `row` - The zero-based index of the row in the data table, corresponding to the node being moused out from.
select	Standard select event **Properties:** None

ready	The chart is ready for external method calls. If you want to interact with the chart, and call methods after you draw it, you should set up a listener for this event *before* you call the `draw` method, and call the methods only after the event is fired. **Properties:** None

Data Policy

All code and data are processed and rendered in the browser. No data is sent to any server.