Business Value Dashboard

Software Version: 10.63

User Guide

Document Release Date: December 2017
Software Release Date: November 2017

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Seattle SpinCo, Inc and its subsidiaries (“Seattle”) products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Seattle shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated, valid license from Seattle required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

AMD, the AMD Arrow symbol and ATI are trademarks of Advanced Micro Devices, Inc.

Citrix® and XenDesktop® are registered trademarks of Citrix Systems, Inc. and/or one more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

Google™ and Google Maps™ are trademarks of Google Inc.

Intel®, Itanium®, Pentium®, and Intel® Xeon® are trademarks of Intel Corporation in the U.S. and other countries.

iPad® and iPhone® are trademarks of Apple Inc.

Java is a registered trademark of Oracle and/or its affiliates.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft®, Windows®, Lync®, Windows NT®, Windows® XP, Windows Vista® and Windows Server® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

NVIDIA® is a trademark and/or registered trademark of NVIDIA Corporation in the U.S. and other countries.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries.

SAP® is the trademark or registered trademark of SAP SE in Germany and in several other countries.

UNIX® is a registered trademark of The Open Group.

Contents

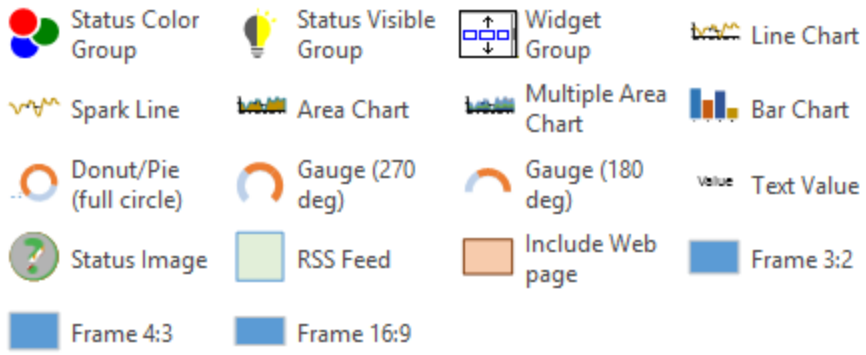
User Guide

This guide provides more advanced details about learning how to use BVD. It includes reference information, procedures, and tips that you may find useful.

Read this section when you are comfortable with the basic configuration steps, and refer back to it when necessary. The information in this section is organized as a reference.

Widgets

The topics in this section describe all available dashboard widgets. You can create the widgets from the following BVD shapes in Visio:



You can edit the properties of a widget in Visio (by editing the Shape Data) or in BVD (by editing a dashboard in Dashboards). We recommend that you make your changes in BVD because the dashboard editor simplifies this task by offering values in drop-down lists for you to choose.

Some advanced operations can only be done in Visio itself. See "Tips and Tricks" on page 44 for details.

You can upload an SVG file as often as you want. If the file already exists, BVD replaces the previous version of the file with the newer version and opens the associated dashboard for you to edit. The newer version does not overwrite existing dashboard properties or widgets, only new widgets are added.

Rule Operators

Rules use operators to compare current and given values. The available operators are:

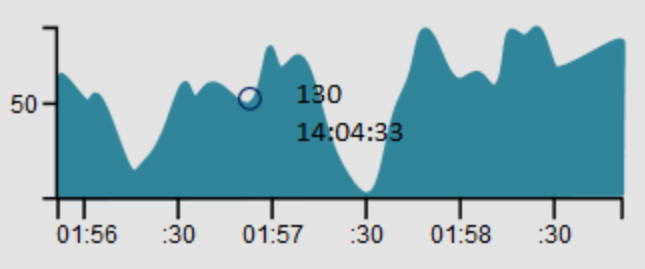
==	equal
<	less than
>	greater than
<=	less than or equal
>=	greater than or equal
!=	not equal

Available widgets:

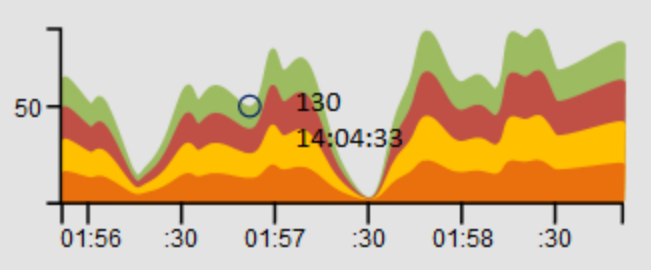
Area and Multiple Area Chart Widgets

Area charts are used to display quantitative data. They are based on line charts.

If you have only one set of data to display, use a simple **Area Chart** widget:



To display multiple sets of data, use a **Multiple Area Chart** widget:



You can customize the x- and y-axes by changing the font and color in Visio.

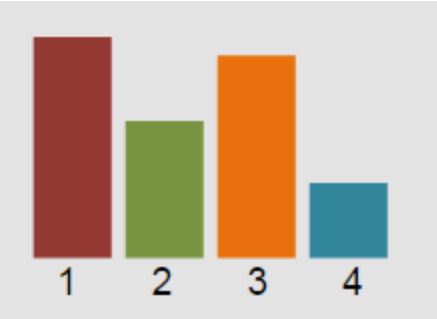
See also:

- "Data Channel" on page 31
- "Data Field" on page 32
- "Min Value" on page 38
- "Max Value" on page 37
- "Chart Autoscale" on page 27
- "Mouse Over" on page 39
- "Show Chart Numbers" on page 40
- "Chart Colors" on page 28
- "Chart Period" on page 29
- "Number Format" on page 39
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Bar Chart Widgets

Use the **Bar Chart** widget to display a bar chart. Bar charts are useful when you want to compare multiple values from the same data channel. Each bar in a bar chart corresponds to a field in the data received over the associated data channel.

The **Data Field** and the **Chart Colors** properties refer to the bars using a numbering scheme. Bar number 1 is the leftmost bar with numbering continuing to the right:

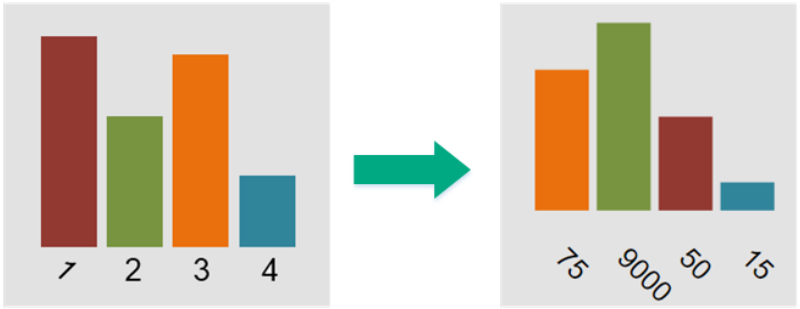


For details on how to set the colors of the individual bars, see "Chart Colors" on page 28.

To make room for larger labels, you can rotate the bar chart labels in Visio.

The **Angle** property in Visio determines the orientation of the label. A default bar chart label has a start angle of 0 (zero). To rotate the labels, set the angle of the first label to the angle at which you want all labels to appear. You can do this either by clicking on the label and rotating it or by changing the value of the label's **Angle** property:
Access the tab **View**. Click **Task Panes > Size & Position**. Click on the first label of the bar chart and adjust the **Angle** property.

For example, to rotate the labels of the following bar chart, set the angle of the first label to -45 degrees. As a result, all other labels will be rotated to the same degree in BVD.



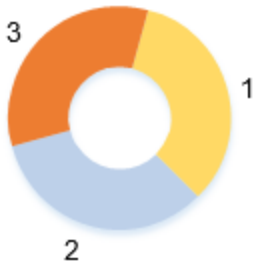
See also:

- "Data Channel" on page 31
- "Data Field" on page 32
- "Max Value" on page 37
- "Chart Autoscale" on page 27
- "Show Chart Numbers" on page 40
- "Chart Colors" on page 28
- "Number Format" on page 39
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Donut/Pie Widgets

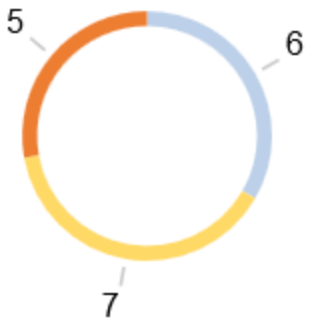
Use the **Donut/Pie** widget to display a donut or pie chart. Each slice in a donut/pie widget corresponds to a field in the data received over the associated data channel.

The **Data Field** and **Chart Colors** properties refer to the donut/pie slices using a numbering scheme. Slice number 1 is the top right slice with numbering continuing clockwise:



In comparison to the Donut Chart widget, you can modify the hole size of the Donut/Pie widget directly in Visio, by using the yellow marker that appears when you click on the chart.

Right-click the widget and select **Hide Labels** if you do not want the widget to be displayed with labels for the received data.



> **Note:** To also modify the start angle and donut size in Visio, use a Gauge widget.

See also:

- "Data Channel" on page 31
- "Data Field" on page 32
- "Max Value" on page 37
- "Chart Colors" on page 28
- "Number Format" on page 39
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Feed Widgets

The **Feed** widget enables you to display information feeds, similar to well-known RSS feeds. Feeds must be sent as JSON-encoded data. The data channel must include the field's `title` and `link` in order to display the feed. New feed items are always added to the top of the list.

The Feed widget displays the following items:

- The creation time stamp of the item.
- The title field as text (retrieved from the `title` data field).
- The link field as hyperlink (retrieved from the `link` data field).

> **Example feed data:**
>
> ```
> {
>    "time":1437633749317,
>    "type":"test",
>    "title":"Tests show UK Quran manuscript is among world's oldest",
>    "link":"http://rss.cnn.com/~r/rss/cnn_latest/~3/HgufPus_pOs/index.html"
> }
> ```

> **Note:** Do not rotate Feed widgets. This breaks the widget functionality.

You can style the displayed feed items by providing a custom CSS definition in the BVD Settings. For more information, see System Settings.

> **Example**
>
> ```
> .feedItem .ts {display: none;}
> .feedItem .even {background-color: #262627;}
> .feedItem a:hover {text-decoration: none;}
> .feedItem section {hight:30px; padding: 3px; margin: 0}
> ```

See also:

- "Data Channel" on page 31
- "Feed Max Items" on page 34
- "Transparent Background" on page 42

Frame Widgets

BVD offers the following Frame shapes:

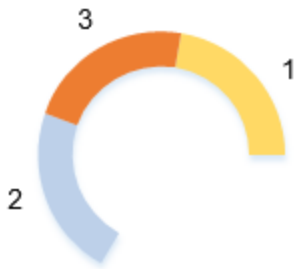
- Frame 3:2
- Frame 4:3
- Frame 16:9

The shapes do not have any shape data defined. Their purpose is to help you lay out your dashboards based on predefined ratios.

Gauge Widgets

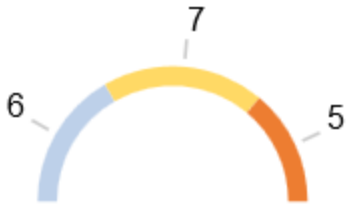
Use the **Gauge** widget to display a gauge chart. By default, gauge widgets are available with a 180 or 270 degrees start angle. Each slice in a gauge chart corresponds to a field in the data received over the associated data channel.

The **Data Field** and the **Chart Colors** properties refer to the gauge slices using a numbering scheme. Slice number 1 is the top right slice with numbering continuing clockwise:



You can modify the start and end angle, donut size, and hole size of the gauge widget directly in Visio, by using the yellow markers that appear when you click on the chart.

Right-click the widget and deselect **Hide Labels** if you want the widget to be displayed with labels for the received data.

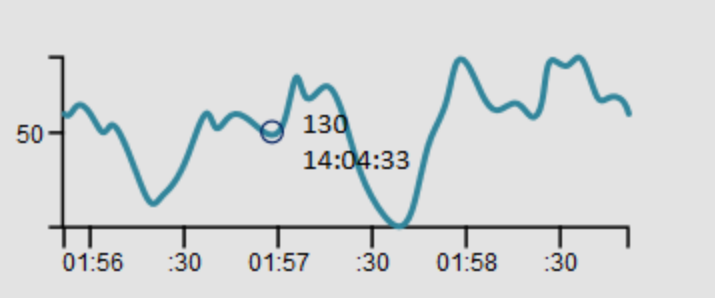


See also:

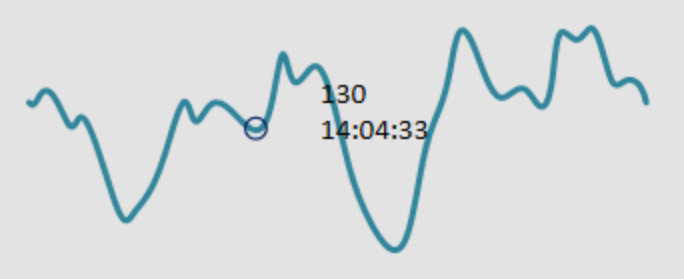
- "Data Channel" on page 31
- "Data Field" on page 32
- "Max Value" on page 37
- "Chart Colors" on page 28
- "Number Format" on page 39
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Line Chart and Sparkline Widgets

Use the **Line Chart** widget to show a line chart with axes and coordinates:



The **Sparkline** widget creates a sparkline chart. Typically, a sparkline is a very small chart, shown without axes or coordinates:



> **Tip:** Position sparklines next to an absolute number to provide you with a quick reference to the data trend.

You can change the style of the line in Visio.

See also:

- "Data Channel" on page 31
- "Data Field" on page 32
- "Min Value" on page 38
- "Max Value" on page 37
- "Chart Autoscale" on page 27
- "Mouse Over" on page 39
- "Chart Period" on page 29
- "Number Format" on page 39
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Status Color Group

Group the **Status Color Group** shape with other shapes in Visio to make them change color depending on the values received. You can define the values that cause a color change using the Coloring Rule property.

Use the group widget when you want to change the color of non-BVD shapes or to color a large number of shapes. Then it is easier to group the shapes with the Status Color Group and set the coloring rule in the group widget.

The Status Color Group widget can color the following SVG elements: `<path>`, `<rect>`, `<ellipse>`, `<circle>`, and `<polygon>`.

See also:

- "Data Channel" on page 31
- "Coloring Rule" on page 30
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Status Image Widgets

Use the Status Image widget to display an image depending on the value received for the Status Field or depending on a rule.

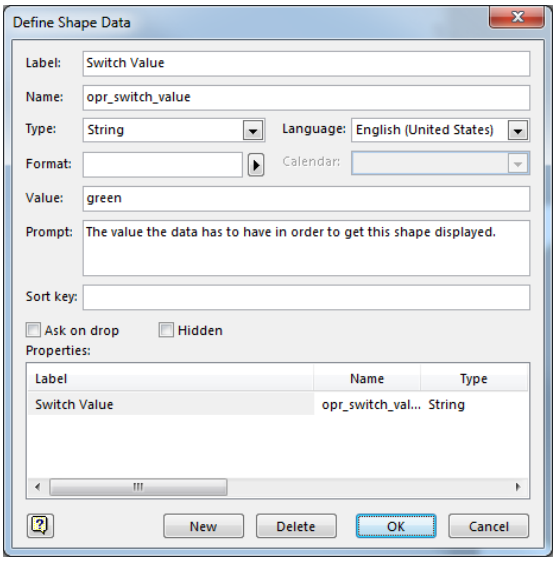
The Status Image shape is a group of shapes. Each shape in the group is an image, and each image has the shape data Switch Value defined. For example, the "error" image in the BVD Status Image shape has the Switch Value "red". BVD then selects the image to display depending on the values received in the data channel. You can choose the data field to use as input by selecting it in the Status Field (default: `status`), or you can define an image selection rule. For example, if the result of the image selection rule is "red", the "error" image is displayed.

You can take a look at the Status Image shape in Visio by right-clicking the shape and selecting **Group > Open Status Image**.

> **Caution:** Never ungroup the Status Image shape.

The shape consists of the following four images:

Image Name	Switch Value	Image
ok	green	
warning	yellow	
error	red	
unknown	grey	



Do it yourself - Create your own Status Image

1. Make sure Visio is running in developer mode:

   File > Options > Advanced > Run in developer mode

2. Drag the BVD Status Image shape to your drawing.
3. Right-click the Status Image shape and select **Group > Open Status Image**.

   You are now inside the shape. You can start editing, deleting, or adding shapes as required.
4. If you add a new shape, you must add the shape data **Switch Value** to the shape:
   a. Right-click the added shape and select **Data > Define Shape Data**.
   b. Add a property with the name `opr_switch_value` of the type `String`.
   c. Set the value of the property to a value of the Status Field or one that can be selected by an Image Selection Rule.

After uploading the exported SVG file to BVD, select the data channel for your Status Image widget. Then either select a data field for the Status Field property or configure an image selection rule. You

also need to set a default value for situations when no value is available or the value is not one of the defined switch values.

See also:

- "Data Channel" on page 31
- "Status Field" on page 41
- "Default Value" on page 33
- "Image Selection Rule" on page 36
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Status Visible Group

Group the invisible **Status Visible Group** Visio shape with other shapes to show or hide the widgets depending on the result of the rule defined in the **Visibility Rule** property.

> **Tip:** Use the group widget when you want to show or hide non-BVD shapes, or to show or hide a large number of shapes. Then it is easier to group the shapes with the Status Visible Group and set the visibility rule in the group widget. You can also set the visibility of a widget based on its **Visibility Rule** property. Use this property to show or hide individual widgets.

See also:

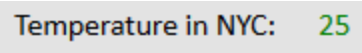
- "Data Channel" on page 31
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Text Value Widgets

Use the Text Value widget to display values or to color text. Text attributes like font, alignment, and rotation are not updated.

For example, you could use the text value widget to display the current temperature in your store in New York City. As temperature measurements arrive, the number changes reflecting the current

temperature measured in the store. Additionally, you could configure the text value widget to change the color of the value displayed depending on the current temperature:



See also:

- "Data Channel" on page 31
- "Data Field" on page 32
- "Number Format" on page 39
- "Coloring Rule" on page 30
- "Visibility Rule" on page 42
- "Hyperlink" on page 34

Web Page Widgets

Web Page widgets enable you to show web pages in a dashboard. For example, you can include web pages that stream television, video, or audio.

The website must be accessible via HTTPS because the BVD UI is also only accessible via HTTPS. However, if the website sends an X-Frame-Options HTTP header restricting the embedding of this page to same origin, the browser will not display this web page within the dashboard.

> **Note:** Do not rotate Web Page widgets. This breaks the widget functionality.

You can add variables to the URL, either based on the data fields in the data channel, or based on the variables from a dashboard template. Variables are replaced with their assigned value when a dashboard or an instance is viewed.

See also:

- "URL" on page 42

Widget Groups

Use the **Widget Group** to display a dynamic list of individual widgets within one widget container. Every widget that is grouped within a widget group receives data from the same data channel. The

JSON data that is sent via this data channel must be an array of objects that contain the list items. The widget group works like a list that is dynamically populated with list items based on the data that is being sent.

> **Note:** Adding Feed widgets, Web Page widgets and Frames to widget groups is not supported. Other than that, all widget types as well as non-widget shapes can be added to widget groups.
>
> If you display historical data from a Data Collector in a widget group, charts with time dimensions cannot be added.

The display of the widgets inside the widget group is limited by the size of the widget group box. If the content of the widgets is larger than the space in the box, the content is cut off. For example, if you have a Text Field Widget with a long value inside the widget group, the text is cut off if the text is longer than the space available within the widget group box.

You can modify the color of the box and the scroll bar individually. The vertical scroll bar appears if the length of the content that is displayed in the widget group exceeds the size of the box.

To add widgets to the widget group, put another widget on top of the widget group, for example the Text Value Widget. Then select both widgets, right-click and select **Group** > **Group**.

Learn more

Send JSON data

The data that is being fed into the widget group via the data channel property must be defined in JSON data and contain an array of values similar to the following:

```
{
        "items" : [
```

```
        {
                "id": "1",
                "value": "content1"
         },
        {
                "id": "2",
                "value": "content2"
         },
         {
                "id": "3",
                "value": "content3"
         }
        ]
}
```

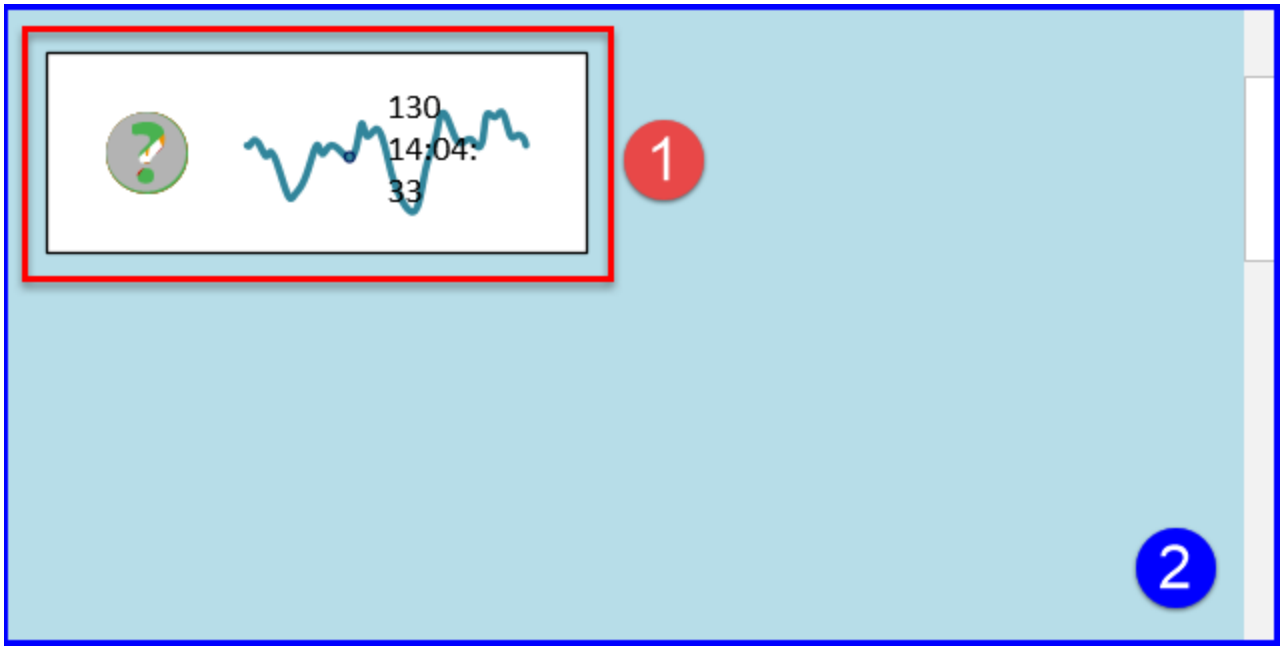
The JSON data must contain an array of values called `items`. The array elements must be JSON objects with a property called `id` that uniquely identifies the single instances of the widgets inside the group.

Data updates must always contain data for all instances to be displayed. If an ID is missing in the updated data, the corresponding widget instance is removed. If a new ID is added in a data update, a new widget instance is added.

Widget Groups with multiple widgets or shapes

A Widget Group can contain more than one widget or shape. Note the following when working with complex widget groups:

- You must group the widgets/shapes inside the group before grouping them with the widget group. For example, if you have two widgets and a basic rectangle shape inside a group widget, group them as follows:



First, you select all widgets and shapes in (1), right-click and select **Group** > **Group**. Then you select the grouped widgets/shapes and the widget group box (2), right-click and again select **Group** > **Group**.

- Widgets inside the group are sorted in the same way as the data in the JSON array. You can control the display order by changing the order of the values in the array.
- In BVD, you can select the widgets inside the widget group and change their properties. However, you cannot change the data channel for those widgets as the channel is set by the group. If you hover over the data channel of the widget group, the latest received values are displayed in a tooltip.

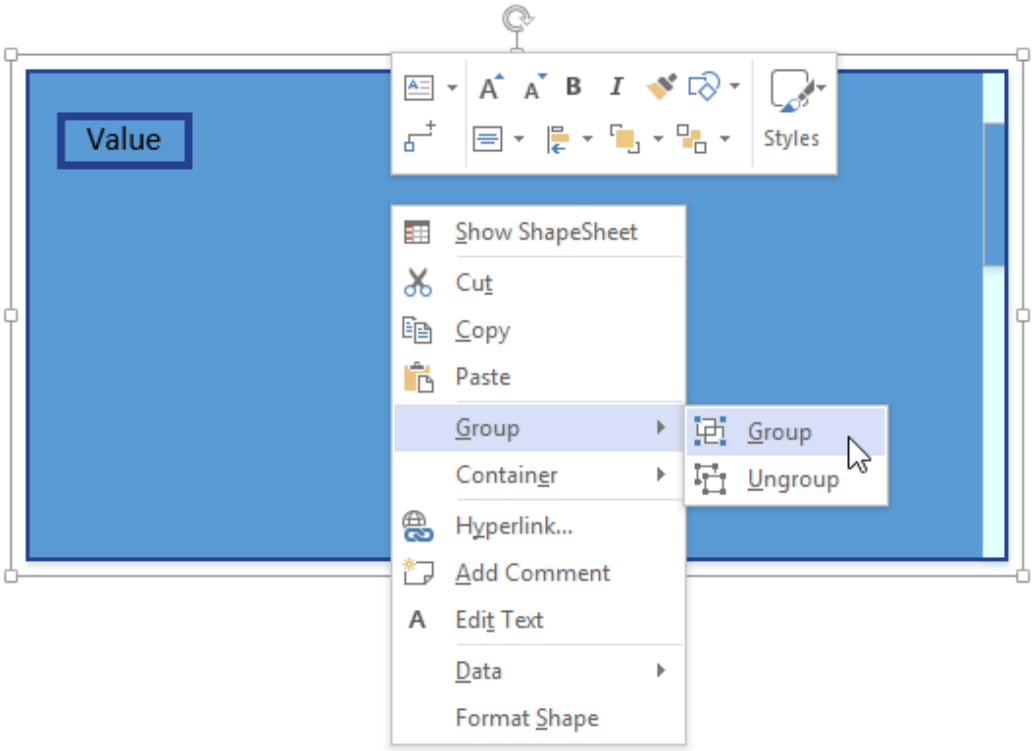
Examples

Example (simple): adding a Text Value Widget to a widget group

As an example, create a dashboard that displays the values of multiple dimensions in a widget group.

1. In Visio, create a new drawing and drag the Widget Group from the BVD stencil into the free space.
2. *Optional*. Change the color of the widget group box and scroll bar.
3. Drag the Text Value Widget on top of the Widget Group. Select both widgets, right-click and select **Group** > **Group**.

Note that you only have to add one Text Value Widget into the Widget Group. The widget will be dynamically replicated inside the widget group as often as required to display all of the data being sent via the data channel.



4. Export the dashboard (or manually save it as an SVG file), then upload the new dashboard to BVD in the Dashboards page.
5. Send nested JSON data to BVD. For this example, we will use the following JSON data:

```
{
        "items" : [
       {
               "id": "1",
               "value": "content1"
        },
       {
               "id": "2",
               "value": "content2"
        },
        {
               "id": "3",
```

```
                "value": "content3"
        },
        {
                "id": "4",
                "value": "content4"
        },
       {
                "id": "5",
                "value": "content5"
        },
        {
                "id": "5",
                "value": "content5"
        },
        {
                "id": "6",
                "value": "content6"
        },
       {
                "id": "7",
                "value": "content7"
        },
        {
                "id": "8",
                "value": "content8"
        },
        {
                "id": "9",
                "value": "content9"
        }
       ]
}
```

Send the data to the receiver URL as described in Create custom integrations.

6. In BVD, edit the dashboard you uploaded earlier. Select the widget group, and choose the data channel that contains the items property.
7. *Optional*. Select the Text Widget inside the widget group to change its properties. Note that you cannot change the data channel for widgets inside a widget group as the channel is set by the group.
8. Click **Save**.

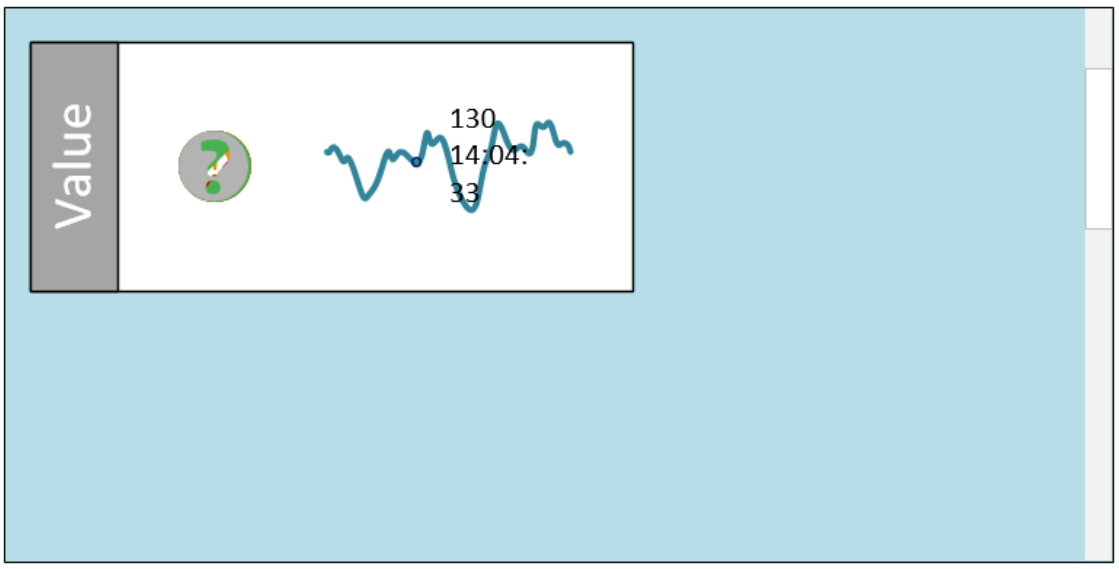
9. Select and view the dashboard in the **Dashboards** menu. The widget group displays all of the list items one after the other. You can use the scroll bar to see all of the values.
10. *Optional*. To see how the widget group displays the content dynamically, add additional list items to the JSON data, and delete or modify existing list items.

Example (complex): adding multiple widgets/shapes to a widget group

As a bit more complex example, create a dashboard that displays the values in a group containing a Text Widget, a Status Image Widget, and a Spark Line Widget.

1. In Visio, create a new drawing and drag the Widget Group from the BVD stencil into the free space.
2. *Optional*. Change the color of the widget group box and scroll bar.
3. Drag and drop a rectangle basic shape into the Widget Group and adjust its size. Then add a Text Value Widget, a Status Image Widget, and a Spark Line Widget on top of the frame. Select the three widgets and the rectangle, right-click and select **Group** > **Group**.

   > **Note:** It is important where in the widget group you place your widgets and shapes. The space between the grouped widgets/shapes and the right side of the widget group box stays the same in the final BVD dashboard. Additionally, the space between the grouped widgets/shapes and the top of the widget group box defines the space between the repeated elements in the final dashboard.



4. Select the grouped widgets and the widget group box, and again select **Group** > **Group**.

5. Export the dashboard (or manually save it as an SVG file), then upload the new dashboard to BVD in the Dashboards page.
6. Send nested JSON data to BVD. For this example, we will use the following JSON data:

```
{
        "items" : [
       {
                "id": "1",
                "value": "content1"
        },
       {
                "id": "2",
                "value": "content2"
        },
        {
                "id": "3",
                "value": "content3"
        },
        {
                "id": "4",
                "value": "content4"
        },
       {
                "id": "5",
                "value": "content5"
        },
        {
                "id": "5",
                "value": "content5"
        },
        {
                "id": "6",
                "value": "content6"
        },
       {
                "id": "7",
                "value": "content7"
        },
        {
                "id": "8",
                "value": "content8"
        },
```

```
        {
                "id": "9",
                "value": "content9"
        }
       ]
}
```

Send the data to the receiver URL as described in Create custom integrations.

7. In BVD, edit the dashboard you uploaded earlier. Select the widget group, and choose the data channel that contains the items property.
8. Select the widgets inside the widget group to change their properties. For the Status or Data Field, select the data you want the widget to display. Note that you cannot change the data channel for widgets inside a widget group as the channel is set by the group.
9. Click **Save**.
10. Select and view the dashboard in the **Dashboards** menu.
11. *Optional*. To see how the widget group displays the content dynamically, add additional list items to the JSON data, and delete or modify existing list items.

See also:

- "Data Channel" on page 31

Widget Properties

The topics in this section describe all available widget properties.

Chart Autoscale

Chart Autoscale automatically scales the maximum values and, if available, minimum values.

Bar charts only. If you set Max Value to 0, the bar with the highest value will be shown in full height; the height of all other bars is shown relative to the highest bar.

Default: not selected

Examples

The following three sparklines have the same size and show the same data. The scaling, however, differs because of different min and max values or Autoscale:

Chart Colors

The **Chart Colors** property enables you to set colors for your charts. Provide a semicolon-separated list of RGB color codes in hexadecimal notation.

You can choose the coloring of the first four colors by changing the colors in Visio. However, the colors you specify in the Chart Colors property override the colors defined in Visio.

If you specify only one color, BVD uses this color for the first data field, and uses the Visio-defined colors for data fields two to four. If more than four data fields but no colors are defined, BVD randomly chooses a color for data fields five and higher.

Default: not defined

Example Donut Chart

The donut chart in the OMi sample dashboard uses the following color codes:

```
FF0000;FF9933;C8C800;33CC33;B2B2B2
```

They color the donut chart like this:

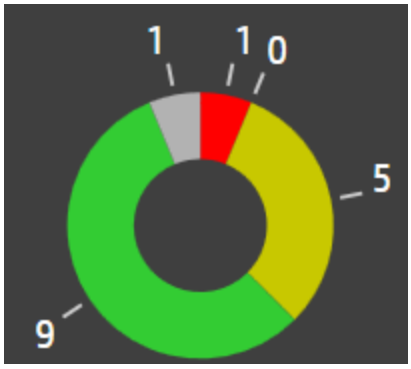


Chart Period

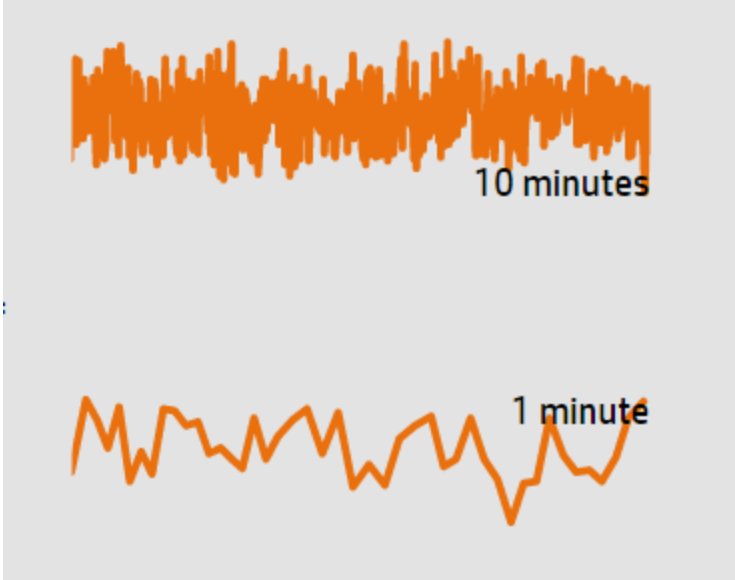
Use the **Chart Period** property to set the period of time (in minutes) for which the widget will display data.

BVD stores only the last 500 data records received from the data senders. To be able to display data, you must align the chart period with the rate data is received. For example, if BVD receives one value per day, the maximum chart period would be 720,000 minutes (500 days). If data is received every 15 minutes, the maximum chart period would be 7,500 minutes (5.2 days).

Default: 10 minutes

Examples

The following two sparklines have the same size and show the same data. The upper line shows the data for the last 10 minutes, the lower line only the last minute.



Coloring Rule

The **Coloring Rule** property enables you to determine the color to display depending on the outcome of a rule.

You can add multiple coloring rules separated by semicolons:

Rule format: `<rule>;<rule>;...`

Coloring rules can contain a value only; for example, the value `#008000` in a coloring rule colors the widget green. Rules can also contain conditions that must be matched; for example, `#008000:temperature<60` colors a widget green when the value of the data field `temperature` is less than 60.

Rules are evaluated from left to right. When one condition is matched, no additional rules will be evaluated. If no rule matches, the default is applied; therefore, when you define a set of rules, always insert the default as the last rule.

Coloring rules have the following format:

`<color>[:<property><operator><value>]`

`<color>`

RGB color code; for example, `#000000`.

`<property>`
The name of the data property to use to calculate the color.

`<operator>`
The operation that is used to compare the current value of the property with the given value. For a list of operators, see "Rule Operators" on page 6.

`<value>`
The value the operator works on.

> **Example**
>
> `#AABBCC:temperature<60;#7FFF00:temperature<30;#00ff00`
>
> If the value of the data field `temperature` is less than 60, the color #AABBCC is used. If the value is less than 30, the color #7FFF00 is used. In all other cases, #00ff00 is used as the color.

Data Channel

The **Data Channel** property enables you to select the data stream over which you want to feed data into the widget. Before you can select a data stream, you must send data to BVD, addressed to the data channel you want to select for your widget.

> **Example**
>
> To show the rise and fall of the temperature in your store in New York City, select the following data channel for your widget:
>
> 
>
>
> The data received includes the following:
>
> ```
> type: Temperature
> element: Monitor
> value: 20.9
> status: #008000
> ```

Data Field

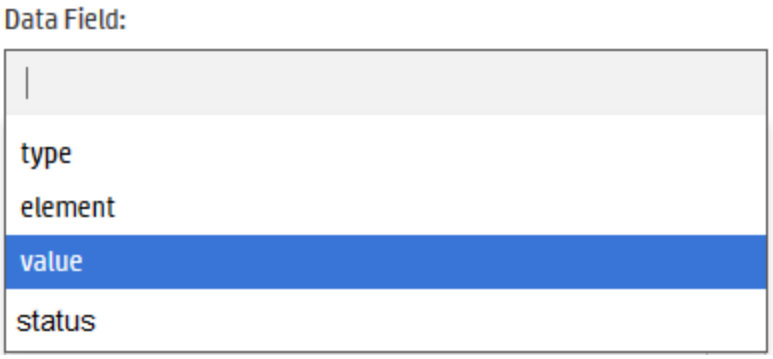
Select one or more data fields in the data that BVD received through the data channel to display in your widget.

Single Data Field

Sparkline, line, and area charts as well as status image and text value widgets support only one data field.

For single data fields, BVD assumes that the data includes the data field `value` and uses the values received for `value` in the widget.

Default: `value`

> **Example Line Chart**
>
> The temperature data BVD is receiving from your New York City store already happens to include the `value` data field, so you do not need to select it. Otherwise select the data field with your temperature values.
>
> 
>

Multiple Data Fields

Multiple area, bar, and donut charts support multiple data fields. Select a data field for each area, bar, or slice in the chart. If you select more than four data fields, BVD automatically chooses the colors of the additional fields. For details, see "Chart Colors" on page 28.

Donut charts only. If you select only one data field for a donut chart, a second one will be automatically generated based on the max value. For details, see "Max Value" on page 37.

Example Donut Chart

The donut chart in the OMi sample dashboard displays the five data fields representing the number of events per severity, resulting in a donut with five slices:

Default Value

The value set in **Default Value** is used when the data in the Status data field is empty or is not one of the defined switch values. For example, if `status: blue` or if `status: <empty>`, then the selected default value will be used by the Status Image widget.

You can only select one of the switch values defined in the Status Image widget.

The switch values of the BVD default Status Image widget are grey, red, green, and yellow.

Default: grey

Donut Size

The **Donut Size** property configures the size of a donut chart within the 360 degrees of the circle of the donut chart. By default, a donut completes a full circle (360 degrees). To create a gauge-like semicircle, set the donut size to 180 degrees.

Default: 360

Donut Hole Size

The **Donut Hole Size** property configures the size (in percent) of the hole in the middle of a donut chart. To create a pie chart, set the donut hole size to 0 percent.

Default: 35

Feed Max Items

The **Feed Max Items** property sets the maximum number of items to display in the Feed widget.

Hyperlink

The **Hyperlink** property enables you to link a widget to another dashboard or to a URL. When a user then clicks the widget, the linked dashboard or URL opens. Linked dashboards open in the current browser window, while external URLs open in a new window.

You can add variables to the URL in order to link widgets to specific content based on the data channel. Use the values assigned to the data fields, or use the variables from a dashboard template. You can also combine both types of variables in one URL.

Data fields as variables

Use the data fields of your received data as variables in the URL. Specify the data field names as variables. The data field names are replaced with their associated values when the dashboard is viewed. You can add data field variables to the URL in the format `#{dataField_name}`.

```
https://example.com/#{dataField_name}
```

When the value associated with the variable changes, the URL changes accordingly.

> **Tip:** You can specify only the data field name, for example `#{url}`, if the data field has a complete URL as value, for example `https://example.com`.

> **Example**
>
> A company wants to link a widget to their website, searching for the host name specified in the data channel:
>
> 
>
>
> They specify the following URL, which later replaces the query value with the value of the dimension `host`:
>
> `https://example.com/q=#{host}`
>
> When viewing the dashboard, the dimension `host` is replaced with its assigned value in the URL:
> `https://example.com/q=Host+A`

If the variable is not defined, the variable expression in the URL is replaced with an empty string.

Template variables

You can add variables from dashboard templates to the URL. Variables are replaced with their assigned value when an instance is viewed. You can add variables to the URL in the format `${variable}`.

> **Example**
>
> A company uses a URL in a dashboard template to show the floor plan of their company location. They use different instances for each location and want to show the floor plan specific to each location.
>
> The following URL replaces the query value with the value of the variable `location` that is assigned to the instance:
>
> `https://example.com/office-locations/floor-plans?site=${location}`
>
> When viewing the instance with the assigned value `Atlanta`, the floor plan for the company location in Atlanta is shown. The `location` variable is replaced with `Atlanta` in the URL:
> `https://example.com/office-locations/floor-plans?site=Atlanta`

If the variable is not defined or the template is converted back into a dashboard, the variable expression in the URL is replaced with an empty string.

For more information on templates and variables, see Template Manager.

Image Selection Rule

The **Image Selection Rule** property enables you to determine the image to display depending on the outcome of a rule.

> **Note:** Image selection rules override the values received for the Status Field.

You can add multiple rules separated by semicolons:

Rule format: `<rule>;<rule>;...`

Rules can contain a value only; for example, the value `green` in an image selection rule selects the switch value `green`. Rules can also contain conditions that must be matched; for example, `green:statusColor==verde` selects the switch value `green` when the value of the data field `statusColor` is `verde`.

Rules are evaluated from left to right. When one condition is matched, no additional rules will be evaluated. If no rule matches, the default is applied; therefore, when you define a set of rules, always insert the default as the last rule.

Image selection rules have the following format:

`<switch value>[:<property><operator><value>]`

`<switch value>`
: Switch value assigned to an image in the Status Image shape. The switch values of the default Status Image shape are `green`, `yellow`, `red`, and `grey`.

`<property>`
: The name of the data property to use to calculate the color.

`<operator>`
: The operation that is used to compare the current value of the property with the given value. For a list of operators, see "Rule Operators" on page 6.

`<value>`
: The value the operator works on.

> **Example**
>
> `green:statusColor==verde;yellow:statusColor==amarillo;red:statusColor==rojo;grey`
>
> If the value of the data field statusColor is `verde`, the image with the switch value `green` is selected. The value `amarillo` selects the image `yellow`, `rojo` selects `red`, and the image `grey` is displayed in all other cases.

Max Value

Max Value in Bar Charts

Set **Max Value** to set the maximum value the chart should display.

Default: 100

Max Value in Donut Charts

Set **Max Value** to set a maximum value for the chart. Setting the Max Value property is only relevant for donut charts with only one data field. If two or more fields are selected, Max Value is disabled.

Default: 100

> **Example Donut Chart**
>
> If the current value of the data field is 30 and Max Value is set to 100, the donut chart will display two slices: one slice with the value 30 and the other slice with the value 70.

Max Value in Line, Sparkline, Area, and Multiple Area Charts

Line, sparkline, area, and multiple area charts have a min and a max value property. Set Min Value and Max Value to adjust the range of data displayed in the chart. Data outside this range is cut from the chart.

If Chart Autoscale is used, the min and max values are ignored.

Default: 100

Examples

The following three sparklines have the same size and show the same data. The scaling, however, differs because of different min and max values or Autoscale:

Min Value

Set **Min Value** and **Max Value** to adjust the range of data displayed in the chart. Data outside this range is cut from the chart. If Chart Autoscale is used, the min and max values are ignored.

Default: 0

Examples

The following three sparklines have the same size and show the same data. The scaling, however, differs because of different min and max values or Autoscale:

Mouse Over

The **Mouse Over** property enables you to enable or disable the mouse over tooltip. The tooltip offers detailed information on the current data point. The indicator and the tooltip font can be customized in Visio.

Default: selected

Number Format

Use the **Number Format** property to format or manipulate the values displayed below the charts.

Number format works as documented here: http://numeraljs.com/

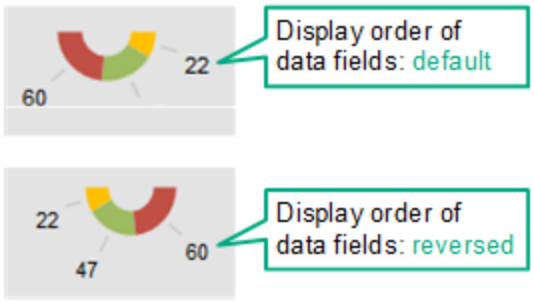
Number format respects the current locale for formatting.

> **Example**
>
> `'$0,0.00'`
>
> This example changes the number 1000.234 to the string $1,000.23.

Reverse Order of Data Fields

Use the **Reverse the Display Order of the Data Fields** property to change the display order of the donut slices (or data fields) from clockwise to counterclockwise. This property is useful when your donut chart starts at an angle contrary to the natural reading direction.

For example, to create the following gauge-like semicircle donut, set the start angle to 90 and the donut size to 180. The first slice starts at 90 degrees with the data sources organized clockwise. To change the order of the slices to counterclockwise, click the **Reverse the Display Order of the Data Fields** check box in the donut widget properties.



Default: not selected

Show Chart Numbers

The **Show Chart Numbers** property has the following effect:

- **Bar and donut charts.** Shows or hides the numbers.

  You can customize the formatting of the numbers (for example, change color or font) by reformatting the number "1" in Visio.

- **Line, area, and multiple area charts.** Shows or hides x- and y-axes. See also "Showing or hiding

x- and y-axes" on page 45.

You can customize the x- and y-axes by changing the font and color in Visio.

Default: selected

> **Tip:** Use the **Number Format** property to format or manipulate the values.

Start Angle

The **Start Angle** property determines the location of the first slice in a donut chart. By default, the first slice (or data field) starts at 0 (zero) degrees within the 360 degrees of the circle of the donut chart.

Default: 0 (zero)

Status Field

The **Status Field** property enables you to select a data field in the received data that contains a switch value for the Status Image widget.

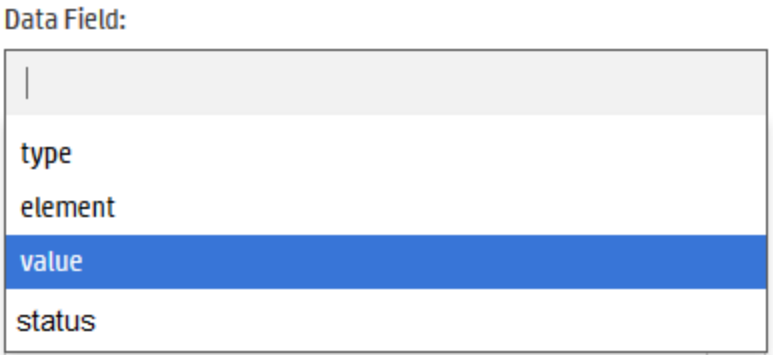
By default, BVD assumes that the data includes the data field `status` and uses the values received for `status` to update the color.

> **Note:** Image selection rules override the values received for the Status Field.

Default value: `status` (providing switch values)

Example

The temperature data BVD is receiving from your New York City store already happens to include the `status` data field, so you do not need to select it. Otherwise select the data field that contains your switch values.



Transparent Background

Select **Transparent Background** to hide the placeholder shape and show the feed as a transparent overlay.

URL

Specifies the **URL** of the web page you want to show in this widget.

Example

`https://marketplace.microfocus.com/`

For more information, see "Hyperlink" on page 34.

Visibility Rule

The **Visibility Rule** property enables you show or hide the widget based on the outcome of a rule.

Tip: You can also use the Status Visible Group widget to show or hide a widget. Use the group widget when you want to show or hide non-BVD shapes, or to show or hide a large number of

shapes. Then it is easier to group the shapes with the Status Visible Group and set the visibility rule in the group widget. See also "Status Visible Group" on page 17.

Visibility rules have the following format:

```
<property><operator><value>
```

`<property>`
The name of the data property to use to calculate the visibility.

`<operator>`
The operation that is used to compare the current value of the property with the given value. For a list of operators, see "Rule Operators" on page 6.

`<value>`
The value the operator works on.

Example

```
errors>=10
```

If the value of the data field `errors` is greater than or equal to ten, the widget is shown in the dashboard. In all other cases, the widget is hidden.

Tips and Tricks

This section includes:

Font usage

For BVD to be able to render the text as designed in Visio, you must make the fonts used in Visio available to the web browser where you view the dashboards. If the web browser does not have access to the fonts, the system default fonts are used.

For example, if you use the Windows font Calibri in Visio, and then view your dashboard in a browser on a Linux system, the browser will substitute Calibri with a Linux system font because Calibri is not installed.

To enable platform-independent text rendering, use Google Fonts when designing your dashboard drawings in Visio. BVD then directs the browser to load the fonts from http://www.google.com/fonts when displaying a BVD dashboard.

You can also use custom fonts but you must set up a publicly accessible web server that serves the fonts and specify a CSS definition for your custom font in the Settings page. See Use Custom Fonts in Your Dashboards for details.

Exporting Visio drawings to SVG

When you save a Visio drawing as an SVG file make sure that the following settings are selected:

- Save as type: **Scalable Vector Graphics (*.svg)**
- Select: **Include Visio data in the files**
- Tip: Press **Ctrl+A** to select everything in the drawing. This ensures that your entire drawing is exported and not the currently selected element only.

Alternatively, click the **Export Dashboard** button in the **Dashboard** ribbon, if you have installed the BVD **Visio Add-in**.

Showing or hiding x- and y-axes

Line, area, and multiple area charts by default show x- and y-axes. You can hide the axes by clearing the **Show Chart Numbers** check box in the widget properties.

The availability of the check box is controlled by the Visio shape data **Show Chart Numbers**, which is by default set to TRUE. If you change this to FALSE and then re-import the exported SVG file, the check box is removed from the widget properties. To re-enable the check box, change the setting to TRUE in Visio and re-import the drawing to BVD.

Displaying small fonts in Firefox

Firefox displays small fonts in SVGs larger than their intended size. To work around this problem, make the original SVG file bigger and let the browser scale it down.

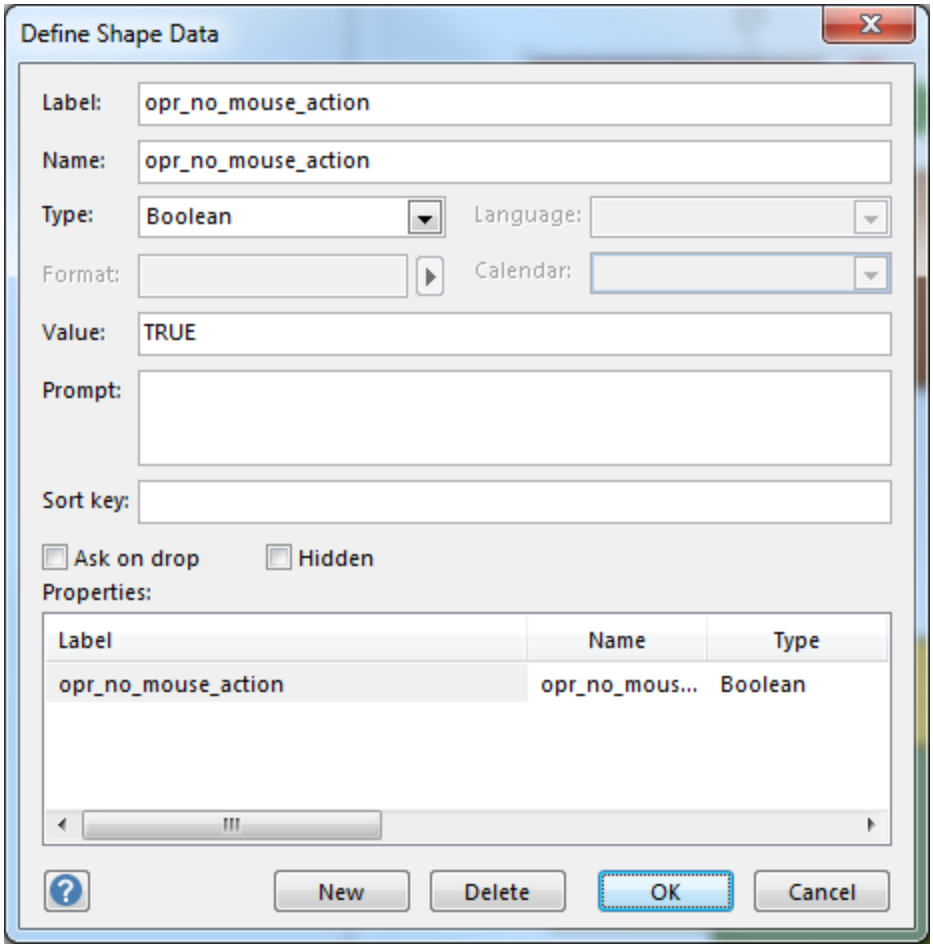
Displaying horizontal or vertical lines with gradient line color

SVG files do not display horizontal or vertical lines with gradient line color. The lines need to deviate from being horizontal or vertical.

Showing widget tooltips even if the widget is overlaid by another shape

> **Note:** To add data to a shape, Visio must be running in developer mode: **File > Options > Advanced > Run in developer mode**.

If you have placed a Visio shape on top of a BVD shape, you can configure the obscuring shape to display the BVD tooltips by adding the shape data **opr_no_mouse_action**:



Set the value of **opr_no_mouse_action** to TRUE. This makes the obscuring shape transparent to the mouse and enables the BVD widget to display tooltips on mouse over.

Inserting Twitter feeds in a dashboard

Although the name suggests it, Twitter feeds cannot be inserted directly in a dashboard using the Feed widget. You would first need to convert the tweets to JSON format and then send the converted tweets to BVD.

The steps below describe an alternative method to include tweets using the Web Page widget:

1. *Prerequisite.* You need a web server that is configured to serve HTML files and allows the inclusion of its pages into the BVD page (X-Frame-Option HTTP Header).
2. Place an HTML file with the following content on the web server:

```
<html>
   <head>
      <style>
         iframe {
                height: 100%;
                }
      </style>
</head>
<body style="margin:0">
</body>
```

3. Place the HTML snippet provided by Twitter between the body tags of the HTML file.
4. Add a Web Page widget to your dashboard and set its URL property to the URL of the page located on your web server. For details, see "Web Page Widgets" on page 18.

Linking dashboards

You can link dashboards by inserting any widget and selecting the target dashboard in the **Hyperlink** property. When a user clicks the widget, the linked dashboard opens and replaces the current dashboard in the browser.

If the link should be a simple button, without status updates, use the Text Value widget as follows:

1. Insert a Text Value widget in your Visio drawing, change the default text "Value" to what will be your link, and style the widget as desired. Export the drawing to SVG and then upload the SVG file to BVD.
2. In BVD, edit the Text Value widget:
   a. Do not select a **Data Channel**. This will cause an error, which you can ignore.
   b. Use the **Hyperlink** drop-down list to select the dashboard you want to link to.
3. Save your changes to the dashboard. Then view the dashboard and test the link.

For details, see "Text Value Widgets" on page 17.

Improving loading time of dashboards with raster graphics

Raster graphics images in dashboards increase the size and therefore the loading time of the dashboards. To reduce the size of the images, compress them in Visio before saving your drawing to SVG. In Visio, select the image, then click **Format > Compress Picture**. Increasing the compression reduces the file size but also the quality of the image.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on User Guide (Business Value Dashboard 10.63)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-asm@hpe.com.

We appreciate your feedback!