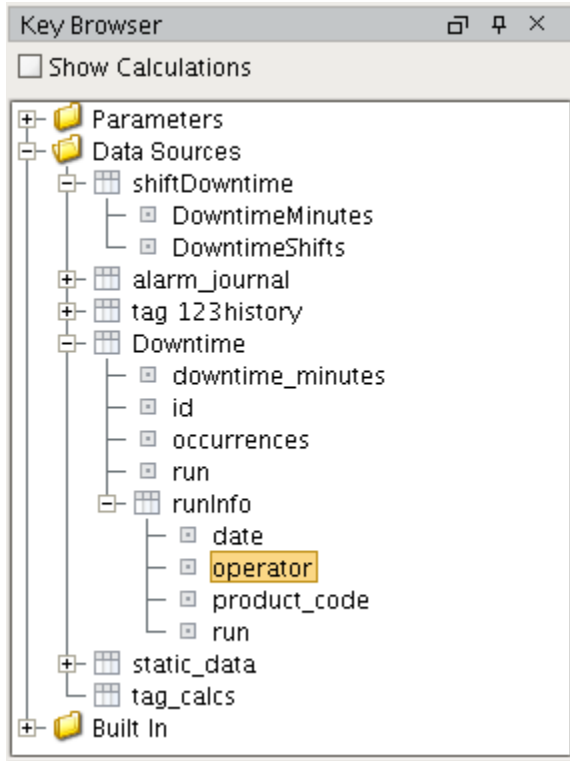


# Data Keys

## Data Keys

In the Reporting Module, we use **Data Keys** to pull values from data sources and show them on the report. In simple terms, Data Keys are placeholders for your data. The simplest reference to data is a simple Data Key. At report generation time, these keys resolve to the values (or sets of values) provided by the data source. Additionally, Data Keys may be used as expressions, which are referred to as Keychain Expressions.

As you add Parameters and Data Sources to the [Data section](#) of your report, they will appear in the Key Browser's **Parameters** and **Datasources** folders.



## On this page

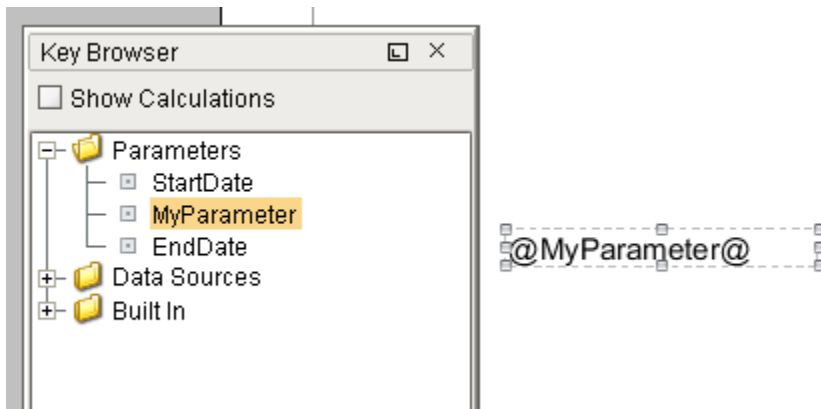
...

- Data Keys
  - Data Keys on the Report
- Built-In Keys
  - Built-In Data Key Description
- Show Calculations Property
- Dynamic Data Keys
  - Configuring a Dynamic Data Key
- Data Key Usage
  - Data Keys as Paths
  - Array Index of Data
  - Colors in Expressions

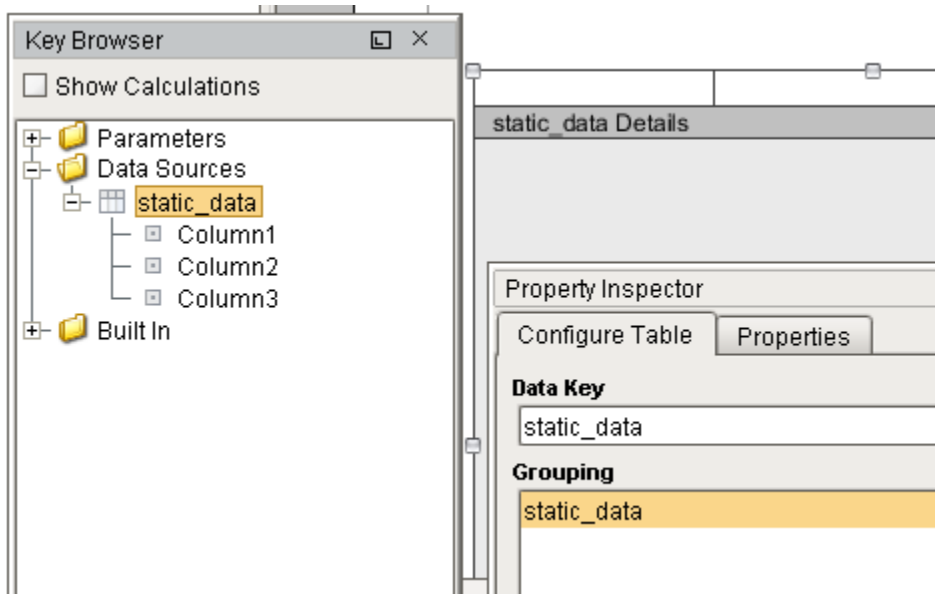
## Data Keys on the Report

Data Keys are enclosed in the "@" character when utilized by components in the report. They may be typed manually, or dragged directly from the Key Browser.

Keys that contain a single value will create a [Text Shape](#) when dragged onto the report.



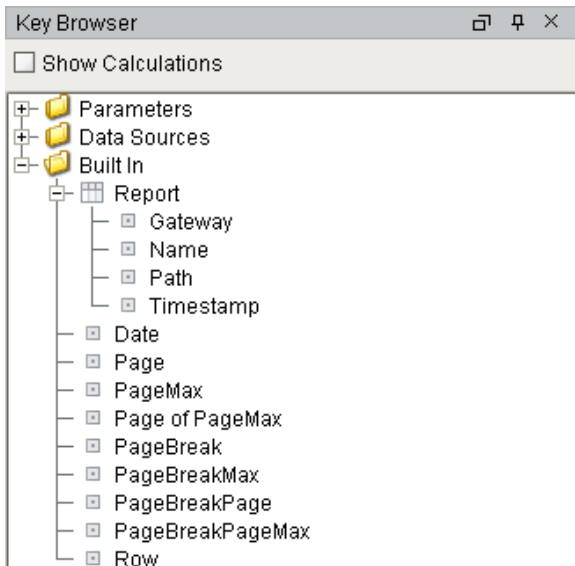
Keys that represent datasets will create a Table component, and configure the Data Key property to use the key.



## Built-In Keys

Built-In Keys provide a lot of useful information on your report at a glance. The Built-In keys are found in the Key Browser. Expand the Built-In folder and you'll see all the default keys, including a Report folder. The keys in the Report folder are specifically related to the report: Gateway name that the report is located in, report name, folder path from the Project Browser to the report, and the Timestamp of the Gateway. The other Data Keys are related to information you may want to add to a report like the date you are viewing or printing the report, page number, number of total pages, and more.

Here's a screenshot of the Key Browser showing all the default Built-In Keys that can be used on a report.



The tables below show the Built-In Report Data Keys and Built-in Data Keys along with a brief description of each key.



**INDUCTIVE  
UNIVERSITY**

---

**Built-in Keys**

[Watch the Video](#)

## Built-In Data Key Description

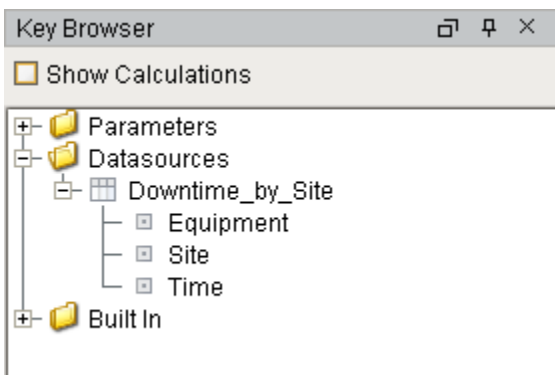
Key	Description										
Report	This key has multiple sub-keys that provide meta-data about the report. <table border="1" data-bbox="375 296 912 533"> <thead> <tr> <th>Key</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Gateway</td> <td>Name of the Report's Ignition Gateway</td> </tr> <tr> <td>Name</td> <td>Name of the Report</td> </tr> <tr> <td>Path</td> <td>Path to the Report in the Project Browser</td> </tr> <tr> <td>Timestamp</td> <td>The Gateway's current timestamp</td> </tr> </tbody> </table>	Key	Description	Gateway	Name of the Report's Ignition Gateway	Name	Name of the Report	Path	Path to the Report in the Project Browser	Timestamp	The Gateway's current timestamp
Key	Description										
Gateway	Name of the Report's Ignition Gateway										
Name	Name of the Report										
Path	Path to the Report in the Project Browser										
Timestamp	The Gateway's current timestamp										
Date	The current date/time										
Page	The current page										
PageMax	The total number of pages in the generated report										
Page of PageMax	Shows current page number and the total number of pages in the report										
PageBreak	The number of explicit page breaks encountered										
PageBreakMax	The total number of explicit page breaks in generated report										
PageBreakPage	The number of pages since last explicit page break										
PageBreakPageMax	The total number of pages in current explicit page break										
Row	Shows the current row number. Must be used in a table										

## Show Calculations Property

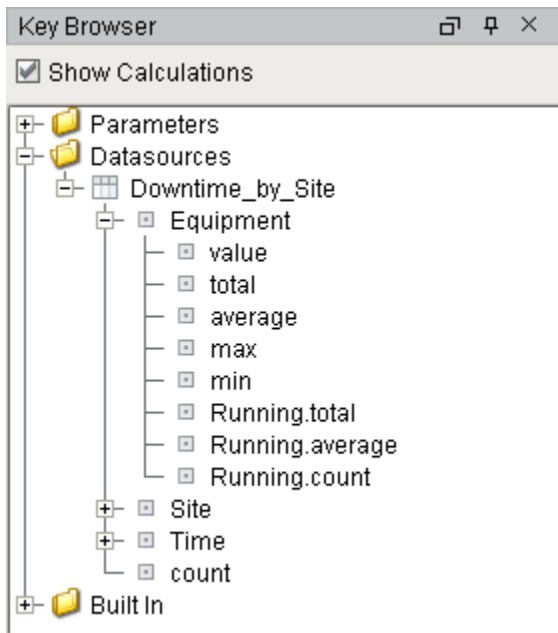
In the Key Browser, the **Show Calculations** property will add several aggregates to each key. These allow your reports to easily display things like the total of a key. These calculations are typically used in the [summary row](#) on the Table component.

Once Show Calculations is enabled, the Key Browser will refresh, and each key will be expandable. Expanding a key will show the available calculations.

### Show Calculations Disabled



### Show Calculations Enabled



Calculation Keys work like any other key: they may be dragged onto the report, and utilized in [Keychain Expressions](#).

## Dynamic Data Keys

Normally, Data Keys may only be used to display the value of a key, such as the Text property on a [Text Shape](#) component. However, they can not be used in the same manner to modify other properties on a report component. Instead, you can utilize Dynamic Data Keys.

Dynamic Data Keys allow you to use the value of a Data Key on a non-string property. With Dynamic Data Keys, you can modify properties on report components, such as the background color or width, based on the value of a key. This is very similar to the binding system used by components in the [Vision Module](#).



**Use Dynamic Data Key**

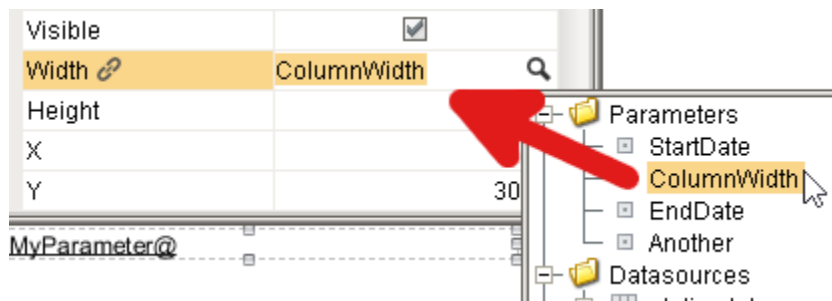
[Watch the Video](#)

## Configuring a Dynamic Data Key

There are two ways to configure a dynamic data key. Note that the syntax of keys differs in Dynamic Data Keys: the "@" are omitted, as demonstrated below.

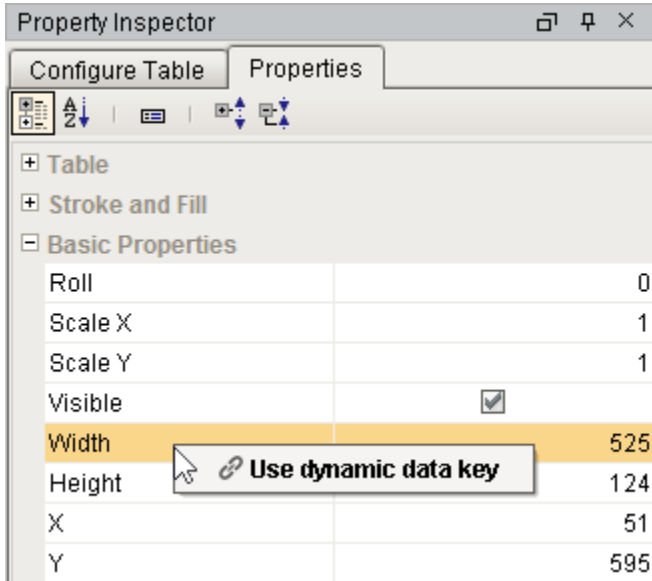
### Drag-and-Drop

The easiest approach is by simply dragging a data key from the **Key Browser** directly to a property on a report component.

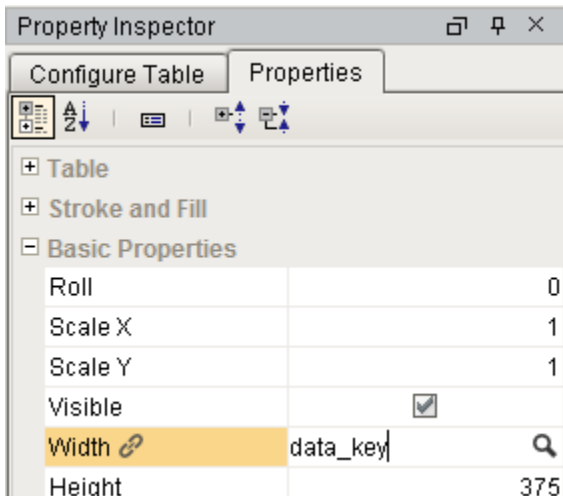


## Right-Click

1. With a report component selected, look in the **Property Inspector**, and left click on the name of a property you wish to place the dynamic data key on.
2. Once selected, you can right-click on the property name and a menu will appear.
3. Click on the **Use dynamic data key** menu item.



4. This will place a dynamic data key on the property. An icon of a link (🔗) will appear next to the property name, and a default key will be applied to the property.
5. Next we will want to override the default value with one of the keys from the Key Browser. Simply left click on the value field and a magnifying glass (🔍) icon will appear.



6. Click on the (🔍) icon. From here a popup of available keys will appear. Select the key you wish to use, and then click OK.

## Data Key Usage

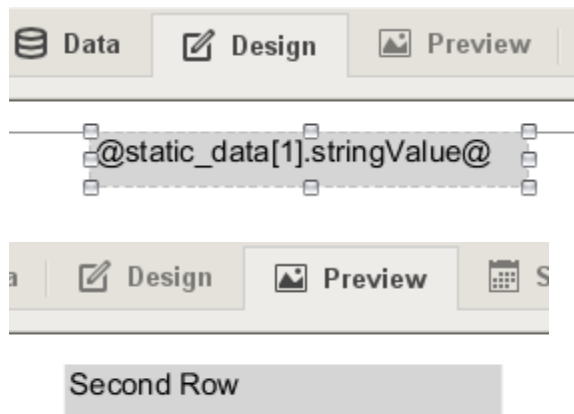
### Data Keys as Paths

Data Keys are *relative*, and use 'dot notation' to reference children. Meaning, if we have a nested data structure, we can use Data Key paths (also known as *Keychains*) to reference the nested data. In the key browser image below, we have a nested data source called Downtime. Downtime contains a number of columns, and then contains a reference to additional data called `runInfo`. If we wanted to access the highlighted `operator` data, we could use the keychain dot notation in the Designer - **@Downtime.runInfo.operator@**. Nested data sources are outside the scope of this page, but you can learn about Data source nesting in the [Nested Queries](#) section.

## Array Index of Data

You can reference an individual object in a list using standard array indexing syntax (brackets) like this: `@dataSource[0].columnName@`, where "dataSource" is a data source that contains a child data key named `columnName`. Assuming a data source with the values listed below, we can retrieve the value of "Second Row" by specifying index 1 and the column `stringValue`: **@static\_data[1].stringValue@**

```
static_data Example
indexColumn, stringValue
0, "First Row"
1, "Second Row"
2, "Third Row"
4, "Fourth Row"
```



## Colors in Expressions

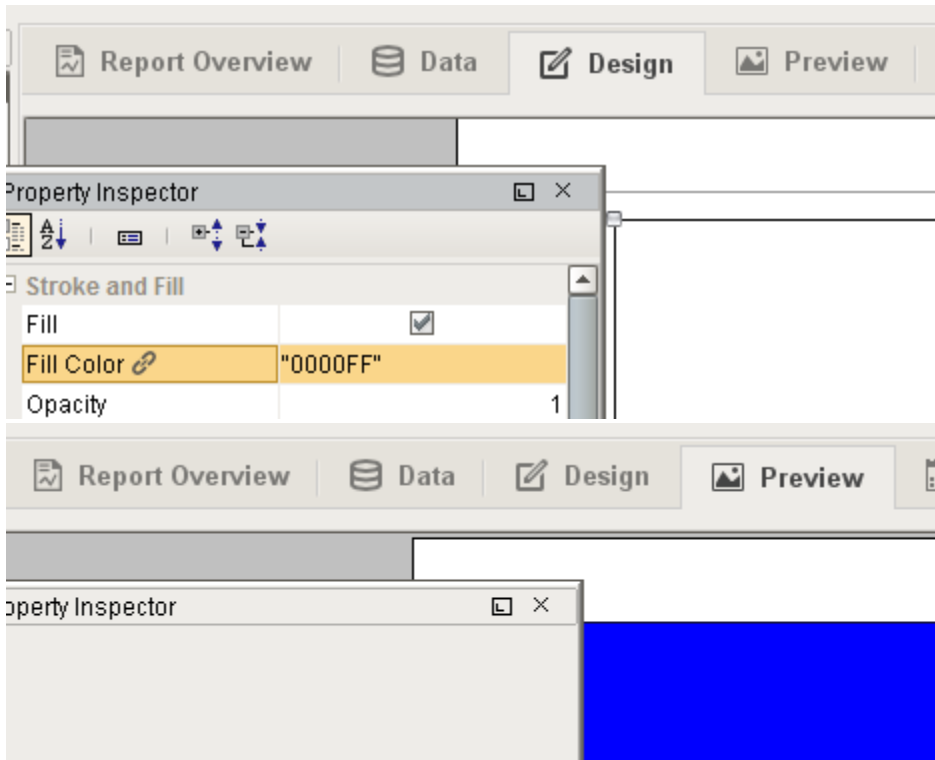
Colors may be references in Keychain Expressions in several ways.

### Colors in Hexadecimal

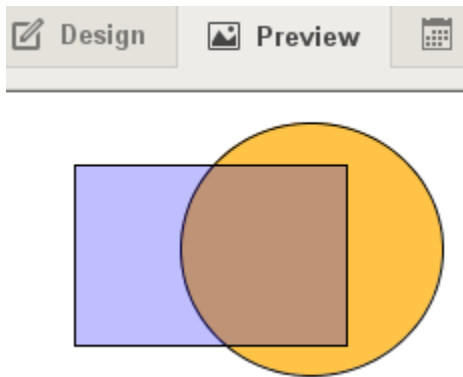
First, hexadecimal case-insensitive color codes may be used. The code must be wrapped in quotation marks to be evaluated correctly. Note that the color change will only appear when the report is executed. The easiest way to test the expression is to switch to the Preview Panel.

Below we see a Blue hexadecimal code of **"0000FF"** is used on the Fill Color of a Rectangle. The Fill Color on the Rectangle was originally set to White. Because the expression will not evaluate in the Design Panel, the Rectangle will appear as a White color.

However, switching over to the Design Panel will generate the report, and evaluate the expression. This in turn returns a Blue Fill Color.

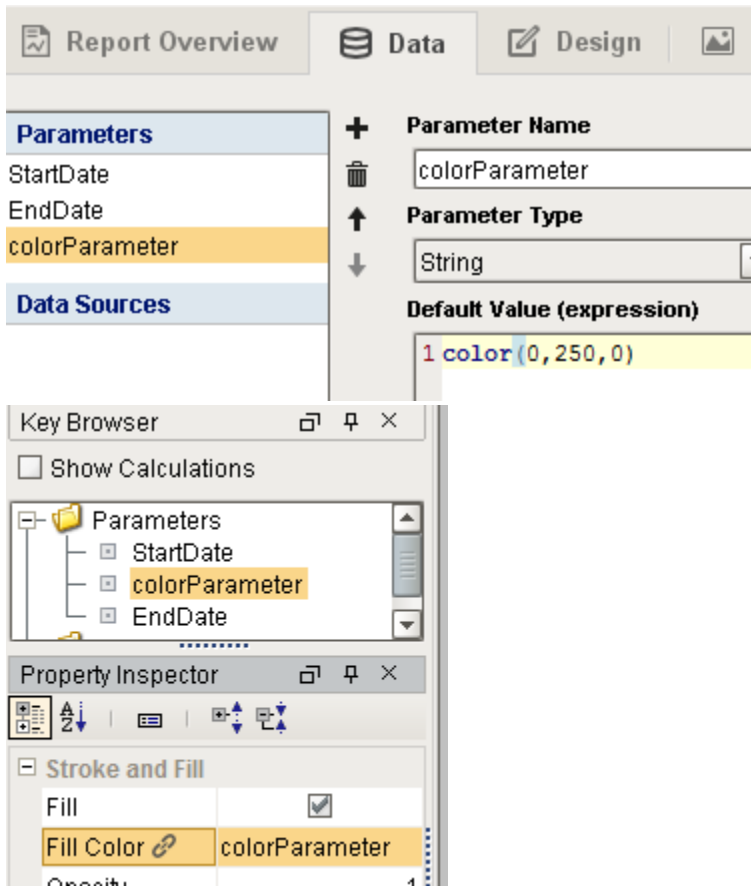


The 7th and 8th digits may be added to specify an alpha channel, or the opacity of the color: **00** is fully transparent, while **FF** is fully opaque. Below, we see a similar rectangle overlapping an ellipse, but with a code of **"0000FF40"**. This represents ~25% opacity, so objects behind the rectangle will be visible, and the fill color will only be slightly opaque.



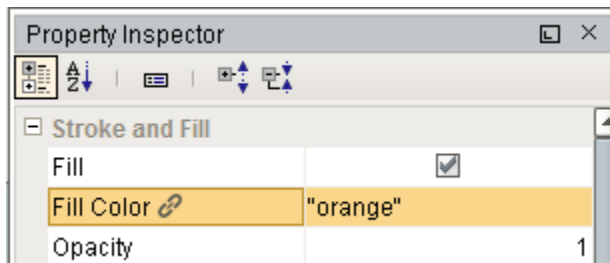
## Parameters as Colors

You may also leverage Report Parameters to specify colors. This typically involves creating a parameter with a string datatype, and using the `color` expression function. Once created, you can simply create a dynamic data key reference on the property by dragging the parameter from the **Key Browser** onto the property in the **Property Inspector**. This way, you can have several components use the same color, and modify the color in a single location.



## Strings as Colors

Additionally, case-insensitive string color names may be used to return a color. Again, the value must be wrapped in quotation marks.



The following string values may be used:



String Value Color Reference

Value	Example	Value	Example	Value	Example	Value	Example	Value	Example
"beige"		"gold"		"lavender"		"pink"		"tan"	
"black"		"goldenRod"		"light Gray"		"plum"		"teal"	
"blue"		"gray"		"lime"		"powderBlue"		"violet"	
"brown"		"green"		"magenta"		"purple"		"white"	
"crimson"		"hotPink"		"maroon"		"salmon"		"yellow"	
"cyan"		"indigo"		"navy"		"silver"		"clear"	Zero opacity. Similar to disabling the <b>Fill</b> property.
"darkGray"		"ivory"		"olive"		"skyBlue"			
"fuchsia"		"khaki"		"orange"		"red"			

In This Section ...