

Visual Action Platform

User Guide

Version 7.4



Visual Action Software, Inc.

Contents

1	Overview	7
1.1	About the Visual Action Platform.....	7
2	Installation.....	8
2.1	Requirements	8
2.1.1	Hardware Requirements.....	8
2.1.2	Software Requirements	9
2.1.3	Optional Dependencies.....	11
2.2	Getting Started.....	12
2.3	Installing the Web Server	13
2.3	Installing Required Software and Dependencies.....	15
2.4	Installing the Platform	18
2.5	Configuring Permissions and Security.....	21
2.5.1	File Permissions	21
2.6	Testing.....	22
3	Platform Environment	24
3.1	Interface.....	24
3.2	Desktop.....	25
3.2.1	Interactions	25
3.3	Navbar.....	28
3.3.1	Logo.....	28
3.3.2	Application Title	28
3.3.3	Menu Button.....	29
3.4	Menu.....	30
3.4.1	Menu Items	31
3.5	Workflow Drawer	33
3.5.1	Opening and Closing.....	33
3.5.2	Navigating the Drawer.....	34
3.5.3	Deploying Extensions	35

3.5.4	Getting More Extensions.....	35
3.6	Editor	36
3.6.1	Workflow Extension Settings.....	36
3.6.2	Application Settings and Preferences	38
3.7	Page Manager	39
3.7.1	Opening and Closing.....	39
3.7.2	Adding an Application Page.....	39
3.7.3	Editing a Page Name.....	40
3.7.4	Deleting a Page	40
3.8	General.....	42
3.8.1	API Keys.....	42
3.8.2	Hover Info.....	43
3.8.3	Keyboard Shortcuts (Keybinds)	43
4	Applications	44
4.1	Getting Started.....	45
4.2	Creating a Brand-New Application.....	47
4.3	Beginning from a Template Application.....	48
4.4	Editing an Existing Application.....	49
4.4.1	From a File.....	49
4.4.2	From Last App.....	49
4.5	Naming the Application	51
4.6	Connecting to Data	52
4.6.1	Data Manager	53
4.6.2	Creating a Data Object	54
4.6.3	Editing a Data Object	55
4.6.4	Naming a Data Object	56
4.6.5	Connecting to a Data Source	56
4.6.6	Using Custom Data Logic	58
4.6.7	Connection Status	60

4.6.8	Data Caching	63
4.6.8	Live Updating	65
4.6.9	Column Data Types.....	66
4.6.10	Choosing a Primary Data Object.....	70
4.6.11	Deleting a Data Object	71
4.7	Adjusting Settings and Preferences	72
4.7.1	Data Services	73
4.7.2	Look and Feel	74
4.7.3	Preferences.....	75
4.8	Adding, Placing, Collapsing, and Removing Extensions	76
4.8.1	Adding Extensions.....	76
4.8.2	Placing Extensions	77
4.8.3	Collapsing Extensions.....	78
4.8.4	Removing Extensions	78
4.9	Configuring Flaremaps and Extensions.....	79
4.10	Reporting	81
4.10.1	Report to Generate	81
4.10.2	Report Source	82
4.10.3	Report Options	84
4.11	Saving an Application.....	85
4.12	Publishing an Application	87
4.13	Advanced References	89
4.13.1	Custom Themes.....	89
4.13.2	Custom Styling Technical Reference	90
4.14	Getting Help.....	94
4-A	Advanced Platform Configurations.....	95
4-A.1	Custom Logic.....	96
4-A.2	Data Logic.....	97
4-A.3	App Logic.....	99

4-A.4	Report Logic.....	100
5	Flaremap	102
5.1	About Flaremaps	103
5.2	Deploying a Flaremap	104
5.3	Flaremap Components.....	105
5.3.1	Flaremap Field.....	105
5.3.2	Presets.....	106
5.3.3	Controls.....	106
5.3.4	Filters	109
5.3.5	Mouseover Boxes.....	113
5.3.6	Header Controls	115
5.4	Flaremap Configuration.....	117
5.4.1	Naming a Flaremap.....	117
5.4.2	Creator Groups	118
5.4.3	Creator Cards	120
5.4.4	Creating Group Options.....	121
5.4.5	Creating Size Options.....	123
5.4.6	Creating Color Options.....	124
5.4.7	Creating (Standard) Filters.....	128
5.4.8	Creating Indicators.....	130
5.4.9	Creating Sliders	133
5.4.10	Setting the Track Cell Icon.....	136
5.4.10	Creating Cell Labels	137
5.4.11	Creating Cell Mouseovers.....	138
5.4.11.1	Quick Settings	141
5.4.11.2	Advanced	144
5.4.11.3	Applying Cell Mouseovers.....	147
5.4.12	Creating Group Mouseovers	147
5.4.12.1	Applying Group Mouseovers.....	152

- 5.4.13 Creating Presets 153
 - 5.4.14 Saving Settings 155
- 5.5 Advanced Flaremap Configuration..... 157
 - 5.5.1 Cell Display..... 157
 - 5.5.2 Manual Configurations 160
- 5.6 Mini/Mobile Flaremaps..... 161
- 6 Index..... 162
- 7 Contact..... 163

1 Overview

1.1 About the Visual Action Platform

Built on Visual Action's long-standing commitment to operational and process relevance, the Visual Action Platform 7.4 allows any type of user to work within a modular environment and tune the interface and logic to meet their own unique needs.

The Platform's modular user environment includes in-app access to suites of Workflow Extensions through a Workflow Extension Drawer.

The Visual Action Platform allows for tailored experiences by and for each user. The Platform enables users to control design, manage layout, and choose specific extensions from an array of Workflow Suites. Users may also tune their selected Workflow Extensions in entirely unique ways.

2 Installation

2.1 Requirements

An installation of an instance of the Visual Action Platform is lightweight and simple, but does include various requirements. The Platform is designed with the intention of running with minimal dependencies; however, several optional features introduce new requirements.

Please note that the included requirements are listed as the minimum requirements for the Platform to run. It is possible, and in some cases likely, that the Platform will perform on systems with less than the minimum hardware requirements or without the listed dependencies. Visual Action, however, cannot guarantee proper functionality or performance of the Platform or any Platform applications in deployments not meeting the requirements listed here.

2.1.1 Hardware Requirements

The Visual Action Platform is web-based application, which runs from a live web server. A suitable physical server is required to run the hosting web server. The physical server may be on-premise, external, or cloud-provided.

The Visual Action Platform only requires a small amount of dedicated storage space (hardware storage space requirements include space for the operating system), and does not require a dedicated server.

Minimum Hardware Specification

The minimum performance specifications for the physical server are listed below. Minimum requirements are based on minimal user quantities at a given time and for acceptable performance. A Network Engineer or Administrator from your organization will be able to determine the optimal hardware for a deployment based on what's available.

Hardware	Minimum Specification
Processor (CPU)	Dual-core processor, 1.5 GHz
Memory (CPU)	4 GB DDR4 2000 MHz
Storage	20 GB
LAN Interface	100 Mbps
Power Supply (PSU)	450 W
Motherboard	Supportive of memory interface rates of 2000 MHz and multi-threaded CPU of 1.5 GHz

2.1.2 Software Requirements

The Visual Action Platform requires a web server environment in which to run. There are various software requirements for this environment.

Server Software and Support Requirements

The below requirements specify the server software and language requirements for the hosting web server.

Software/Language	Requirement
-------------------	-------------

Operating System (OS)	A supported operating system capable of hosting a web server that meets the other software requirements. Some examples include, but are not limited to: • Windows Server 2016 or later • Windows 10 or later • Linux Debian 10 or later Visual Action recommends Linux Debian 11.
Web server	Web server software supporting each of the below software versions. Visual Action recommends Apache HTTP Server 2.4.51 or higher for Linux.
PHP	PHP 7.4.26 or higher (with support for 7.4 features)

Client Software Requirements

The below requirements specify the client software requirements for Platform users.

Software	Requirement
Web Browser	A supported and updated browser supporting HTML 5, CSS 3, and JavaScript ES6. Any browser meeting these requirements will function, but for best performance, Visual Action recommends the latest version of one of the following: • Mozilla Firefox, • Google Chrome, or • Microsoft Edge

2.1.3 Optional Dependencies

The following dependencies are required only for the use of specific features within the Platform. They may be left uninstalled, but certain features will not work (indicated in table).

Dependency	Required For
PHP MySQL Extension (mysqli)	Connection to MySQL databases
PHP SQLite3 Extension (sqlite3)	Connection to SQLite databases
PHP PostgreSQL Extension (pgsql)	Connection to PostgreSQL databases
PHP MS SQL Server Extension (php_sqlsrv_74)	Connection to Microsoft SQL Server databases
PHP Oracle Extension (pdo_oci)	Connection to Oracle databases
PHP Firebird Extension (pdo_firebird)	Connection to Firebird databases
PHP ODBC Extensions (pdo_odbc)	Connection to databases using ODBC

Note: For customized deployments (custom database queries, Extensions, or other deployment features), additional dependencies may be required.

2.2 Getting Started

This Getting Started section will walk through the basic steps of initializing a deployment environment as well as the installation of the Platform itself. It is not a complete guide, as such documentation should be made available by the provider of the software you are using for each step. The following sections intend to provide the necessary details relative to setting up a Platform deployment, and aim to use an example to point engineers in the right direction. The following instructions are based on a sample environment using Apache 2 as the web server and Microsoft Windows 10 or 11 as the client machine operating system.

2.3 Installing the Web Server

To run a web application, such as the Visual Action Platform, a web server is required. A web server provides the hosting mechanisms for serving up web pages to users and the backend mechanisms for managing other web-related services and user sessions. Below are the basic instructions for installing Apache 2. Choose the instruction set corresponding to the operating system running on the server machine.

Linux

Begin by navigating to the Apache Software Foundation website. As of the release date of this documentation, this website is located [here](https://httpd.apache.org/) (<https://httpd.apache.org/>).

In the sidebar to the left of the page, navigate to the “Download!” page.

Download the latest stable release of the Apache HTTP Server (or the desired version of choice).

Alternatively, you may use a program such as `curl` to download the appropriate file from the Apache Software Foundation website or their trusted alternative download sources.

Verify the integrity of the downloaded file(s) to ensure that you have obtained an official, unaltered copy of the Apache HTTP Server software. Integrity verification is an integral component of security, and Visual Action recommends not skipping this step in a deployment. The Apache Software Foundation provides a link to its PGP Keys file on the downloads page as well as from other pages on the website. More information on verifying the Apache HTTP Server download can be found [here](https://httpd.apache.org/dev/verification.html) (<https://httpd.apache.org/dev/verification.html>).

Once you have verified the integrity of the files, follow the instructions on the Apache Foundation website for installing the HTTP Server (or follow the associated instructions on the Apache Foundation website for instructions on installing from other forms).

Refer to the Apache HTTP Server documentation for information on how to complete the installation and setup of the web server. Be sure to complete configuration of the hosted directory and to configure SSL and load Digital Certificates as applicable.

Windows

Begin by navigating to the Apache Software Foundation website. As of the release date of this documentation, this website is located [here](https://httpd.apache.org/) (<https://httpd.apache.org/>).

In the sidebar to the left of the page, navigate to the "Download!" page.

Locate the latest stable release of the Apache HTTP Server (or the desired version of choice).

Click the provided link for "Files for Microsoft Windows". This link may only be provided under the latest stable release.

This link will bring you to a page with information on using Apache HTTP Server on Microsoft Windows. You may choose to download and compile the software from source yourself, if desired, as the Apache HTTP Server Project does not provide Windows binaries. Alternatively (and recommended for new users), you may use one of the multiple popular options containing a binary package as well as other features. This page will list some of these options. Further Windows examples in this documentation will feature [WampServer](#) as the package of choice. Select your package of choice and follow the associated instructions for downloading it.

If provided from the download source, verify the integrity of the downloaded file(s) to ensure that you have obtained an official, unaltered copy of the Apache HTTP Server software. Integrity verification is an integral component of security, and Visual Action recommends not skipping this step in a deployment. The Apache Software Foundation provides a link to its PGP Keys file on the downloads page as well as from other pages on the website. More information on verifying the Apache HTTP Server download can be found [here](https://httpd.apache.org/dev/verification.html) (<https://httpd.apache.org/dev/verification.html>).

Once you have verified the integrity of the files, follow the instructions on the package provider's website for installing the Apache HTTP Server. See the next section before installing to ensure that you install the necessary and optional software during the setup process.

2.3 Installing Required Software and Dependencies

By default, your web server should support HTML 5, CSS 3, and JavaScript ES6 or newer, as these are core languages relative to the web. If you find that your installation encounters errors using any of these languages, please consult the appropriate documentation or support channel of the software you are using for further details and assistance in resolving the issue(s).

Beyond these three languages, the Visual Action Platform requires the use of PHP 7.4.26 or newer. Follow the instructions below for installing PHP in your web server environment, using the instructions applicable to the server's operating system.

Linux

Begin by navigating to the PHP Group website. As of the release date of this documentation, this website is located [here](https://www.php.net/) (<https://www.php.net/>).

In the navigation bar across the top of the webpage, navigate to the "Downloads" page.

Locate and download the current stable release (or the desired version of choice, as long as it is version 7.4.26 or later).

Alternatively, you may use a program such as [curl](#) to download the appropriate installation file.

Verify the integrity of the downloaded file(s) to ensure that you have obtained an official, unaltered copy of the PHP software. Integrity verification is an integral component of security, and Visual Action recommends not skipping this step in a deployment. The PGP Group provides checksums directly following each download file link on the Downloads page.

Follow the instructions provided in the PHP Documentation for installing PHP within your webserver environment. The PHP Documentation is available in a variety of languages [here](https://www.php.net/docs.php) (<https://www.php.net/docs.php>).

During the installation, while configuring and building PHP, you will be permitted to optionally include extensions to enable when running PHP in Apache. This is the step where you will want to enable the extensions listed above (in Section 2.1.3) in support of any database connections you will need. Extensions can be enabled later in Apache, provided they are installed.

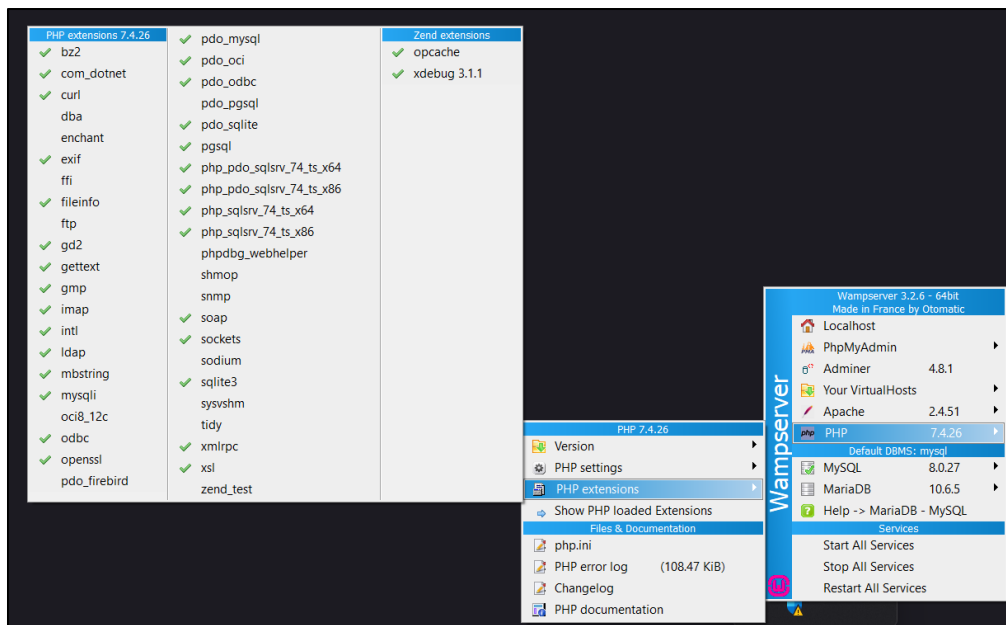
The relevant extensions used by the Platform, and their installation and usage instructions, may all be found on the PHP Group website [here](https://www.php.net/manual/en/refs.database.php) (<https://www.php.net/manual/en/refs.database.php>).

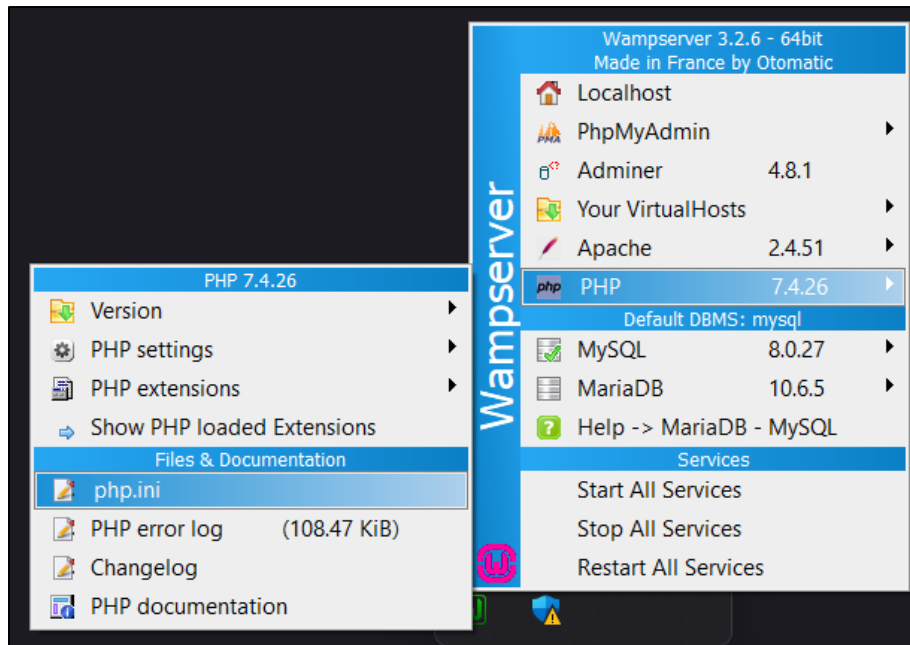
Once you've installed PHP in your web server, configured the web server, and enabled your PHP extensions, restart the web server. It is now prepared for hosting.

Windows

When installing Apache through a precompiled package such as WampServer, you will be provided the option to install additional utilities such as PHP and MySQL. You should choose to install PHP 7.4.26 or later during this process. See the WampServer documentation for further details on this process.

Once the web server environment is installed, along with PHP, most necessary extensions will be available by default. In WampServer, you can use the system tray icon to open up the PHP > PHP extensions menu and enable the required extensions listed above for database support (this will require a restart of the web server). You may also need to modify the installation's php.ini file to enable these extensions.





If you find that some extensions required are not already available in your installation, navigate [here](https://www.php.net/manual/en/refs.database.php) (<https://www.php.net/manual/en/refs.database.php>) to find instructions on installing the necessary database extensions.

Once you've installed PHP in your web server, configured the web server, and enabled your PHP extensions, restart the web server. It is now prepared for hosting.

NOTE: for standard deployments, using an HTTP server such as Apache, the following steps are essentially the same whether running Apache via WampServer for Windows or via Linux. This, further instructions will not be separated for each operating system.

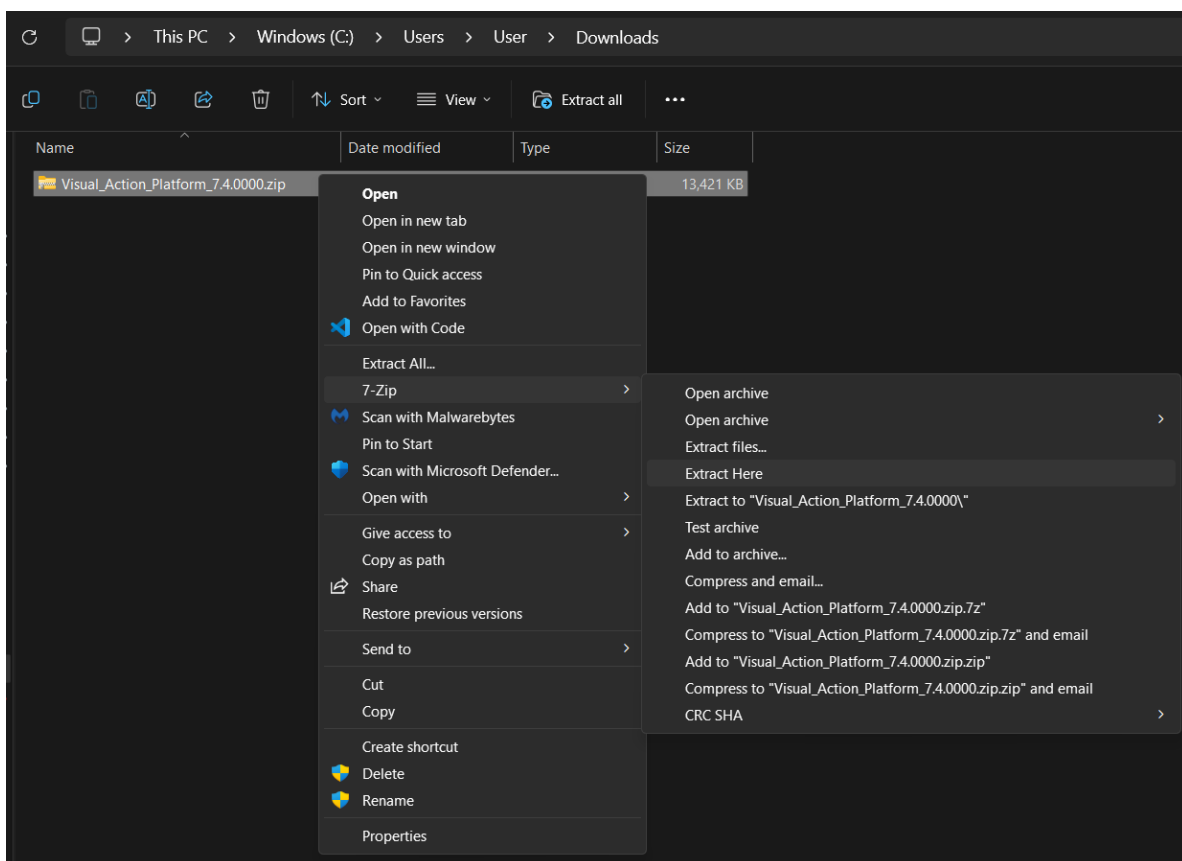
2.4 Installing the Platform

The Visual Action Platform is as easy to install as any traditional webpage. The process simply involves obtaining the Platform files and deploying the application files into the desired location on your web server.

Begin by logging into the [Visual Action Software Releases](https://www.visualaction.com/software/) site (<https://www.visualaction.com/software/>) and downloading the latest release of the Visual Action Platform along with its corresponding “readme” file.

Verify the integrity of the downloaded file. The checksum(s) is available on the downloads page directly following the link to download the Platform. Integrity verification is an integral component of security, and Visual Action recommends not skipping this step in a deployment.

Once you’ve downloaded and verified the Platform files, use your extraction software of choice to extract the compressed folder containing the Platform files.



Once you've extracted the contents of the compressed folder, open it in an explorer window (or list its contents via the command line).

You will see four items within the Platform folder:

1. Flaremap Development Kit Technical Reference [PDF File]
2. Visual Action Platform 7.4 User's Guide [PDF File]
3. Visual_Action_Platform [Directory]
4. Flaremap_Development_Kit [Directory]

If you have just extracted the Platform files and see a difference in any of the contents, please permanently delete all Platform files, redownload the package from the Visual Action website, and verify the integrity of your new download.

The `Visual_Action_Platform` folder contains the deployment-ready Visual Action Platform application files. To install the Platform on your web server, simply cut or copy this directory from the current folder (extraction location) and place it in the appropriate directory in your hosted environment on the web server, where you want it to be accessible.

The default hosting directory for Apache HTTP Server on Linux is `/var/www/html/`. Your network administrators may choose a different location or a specific subdirectory for installation.

It is recommended that the Platform files remain in their own subdirectory to avoid conflicts with files related to other applications or services, so it is not recommended that you remove the Platform application files from the copied directory and place them in another directory which is shared with other applications. However, you may change the directory name from `"Visual_Action_Platform"` to another desired name in order to achieve the desired URL for accessing the Platform.

Once you've copied the Platform files to the intended hosted directory (and renamed the parent folder to the desired resource name), you've installed the Visual Action Platform!

Note that you may deploy multiple instances of the Platform, but they should be kept separate from one another in the server's directory structure to avoid conflicts between deployments.

It is recommended that you copy the two PDF files from the downloaded Platform files into a directory that may be made available to users. This documentation is also available online, but it is important that users are given copies of/access to the Platform documentation. You may want to make a backup copy for your records as well.

The final item, the "Flaremap_Development_Kit", contains the files needed for using the Flaremap Development Kit. Since this is not a hosted application, and works standalone in a browser with or without PHP, it is recommended to distribute the entire compressed file to developers who need to work with the Development Kit. They may extract and use the Development Kit files as needed, and may copy any files necessary from the Kit to other locations where necessary for embedding and integrating with other software. You may want to move the compressed file containing the FDK to a location from which it can be copied and redistributed to developers.

You may want to clean up the location of the extracted files once you have placed copies of each item in the appropriate locations.

2.5 Configuring Permissions and Security

2.5.1 File Permissions

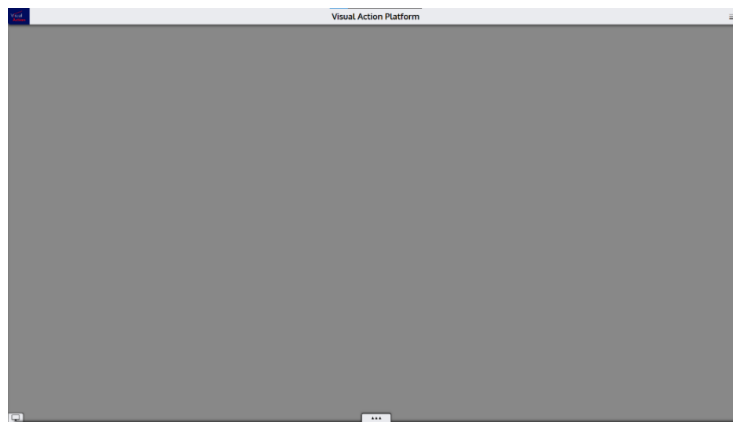
In order to achieve optimal security, permissions for standard users should be set to read-only or read and execute for your Platform deployment. Users should be able to access and retrieve the Platform files via a browser, but should not have permissions to modify Platform files and resources.

Only admins and authorized personnel should be given access to modify Platform files. Some users may be given authorization to modify the `keys.js` file (which stores API keys for the Platform) and other config files, but not the rest of the Platform resources.

2.6 Testing

Once your deployment of the Visual Action Platform is installed and configured, you'll want to ensure that everything is working properly before notifying users that they may access the Platform environment.

First, navigate to the URL at which your Platform instance is hosted. If the Platform environment loads to a brand new app, similar to that shown below, you've achieved access to the Platform.



Reaching this page indicates that your web server is indeed serving the Platform application. Next, you should ensure that the PHP portions of the Platform are working correctly. The easiest way to check that PHP is working is to open up the Workflow Extension Drawer (see Section 3.5.1) and look to see that the Extension libraries and the Extensions within a couple of them are loaded. While checking this does not indicate that all PHP functions are working properly, such as database extensions, it is a quick way to verify that PHP is running and that the Platform's PHP instructions are being processed.

You may also add a database in the Data Manager at any point to test the proper installation and configuration of your PHP database extensions (see Section 4.6.1).

Finally, test access to the Platform from a suitable browser on a system other than the server itself. Ideally, test access from an actual user's machine.

If you are able to load and navigate the Platform as desired, your deployment is ready to use! Open up a new application and start building!

Should you encounter any issues during testing, review the other sections of this guide under Section 2, or contact Visual Action for technical support (see Section 7).

3 Platform Environment

3.1 Interface

The interface of the Visual Action Platform 7.4 is where most user actions are taken and where most control of an application is performed. The Visual Action Platform 7.4 is designed to allow most users to design and edit applications to meet their specific needs directly through the interface – without the need to dive deeper into technical configuration files or external programs.

3.2 Desktop

The Desktop is a large, open area on the screen in which a user may design and use an application. Some users may begin with a Desktop that has been preconfigured, while others beginning with a brand-new application will see a blank slate.



The graphic above shows an empty Desktop environment. All modifiable elements of an application that may be interacted with by a user will be laid out within this Desktop area. These elements are automatically contained within the usable Desktop environment; there is no need for the user to worry about designing an application with unreachable elements.

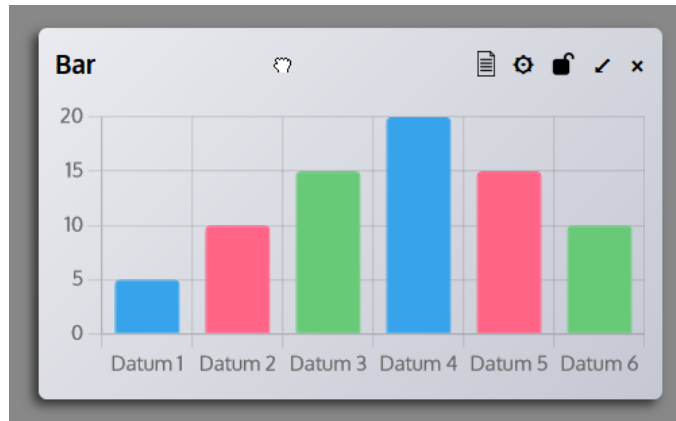
A single Platform application may contain up to 5 different Desktops (application pages). See section 3.7 (Page Manager) for more details.

3.2.1 Interactions

Elements within the Desktop are interactive, allowing users to design and configure an application to their own specifications.

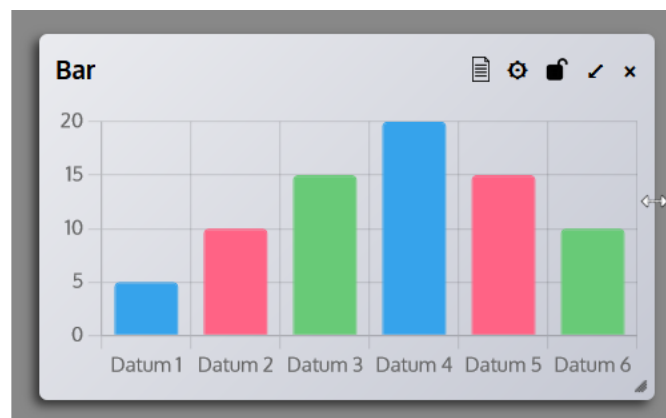
Dragging

Elements added to the Desktop can be moved freely by clicking and holding on the extension header (upper ribbon). Upon clicking and holding, a grabbing cursor will appear, meaning the element can be dragged to the desired location. Releasing the mouse will leave the element placed in its new location.



Resizing

Elements on the Desktop can also be freely resized. Resizable elements are indicated by the appearance of the  symbol in the lower-right corner of an element when it is hovered over. While hovering over the outer edges of an element, directional arrows will appear, indicating in which direction(s) the element may be resized from that edge. When these arrows appear, clicking and dragging (without releasing the mouse) allows the user to transform the element to the desired shape and size in the applicable direction. Releasing the mouse will leave the element with its new size and shape.



The edges of an element allow it to be resized horizontally (left or right edge) or vertically (top or bottom) only. Corners allow the element to be resized in both directions simultaneously.

See section 4.9 Configuring Flaremaps and Extensions for more details on all of the dynamic interactions and controls of elements and Workflow Extensions on the Desktop.

3.3 Navbar

The topmost element of the Visual Action Platform 7.4 interface is the Navigation Toolbar, or “Navbar.” The Navbar is always visible to the user, even in a brand-new application environment.



A Visual Action Platform 7.4 application’s Navbar initially includes three components:

3.3.1 Logo

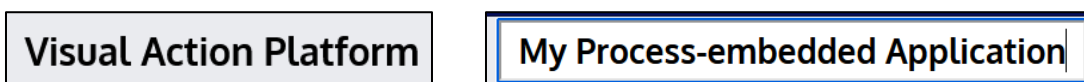
At the left of the Navbar, organization or project logos are displayed (see 4.7.2 Look and Feel).



3.3.2 Application Title

The center of the Navbar contains the title of the currently open application. For new applications, this will default to “Visual Action Platform”.

The title of an application can be edited right from within the Navbar. Simply click on the title text to allow editing.



3.3.3 Menu Button

At the right of the Navbar is the Menu button, which opens the Platform Menu containing the general utilities for managing settings, applications, and more. The Menu is described in further detail in the next section (3.4 Menu).



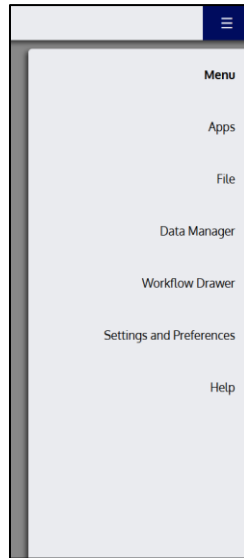
Supplemental Information and Utilities

Sometimes, the Navbar contains “widgets,” or supplemental components that provide additional information or controls, such as live update statuses, timers, clocks, data controls, and more. Navbar widgets are typically custom application components, and none are displayed by default.

● Last updated: 4/2/2024, 4:06:11 PM

3.4 Menu

The Menu provides access to all of the Platform's controls. For many overarching functions and settings within the Platform, the Menu is the place to begin.



To open the Menu, click the Menu button () in the upper right of the Navbar.

3.4.1 Menu Items

Once opened, the Menu contains many options. The table below describes each of those options.

Menu Item	Description
Dropdown Item	
Apps ↓	Reveals options related to your recently used apps and the currently open app.
Open Last App	Recovers and opens the most recently used Platform application.
Clear	Clears (resets) the current working app (will result in loss of current app without a saved copy).
File ↓	Reveals options related to saving, opening, and creating applications.
Save	Saves the current application under the last-used file name. If the application hasn't yet been "saved as," this will initially prompt for a file name.
Save As	Saves the current application with the ability to set the file name.
Open	Opens an existing application.
New	Begins a new application and clears the current working Platform environment.
Data Manager	Opens the Data Manager utility, where Data Objects and data connections can be created, edited, and managed.
Workflow Drawer	Toggles the Workflow Extension Drawer. This is an alternative to the pull-tab of the Drawer at the bottom of the screen.
Settings and Preferences	Opens the application's Settings and Preferences in the Editor.
Help ↓	Links to resources for finding information and assistance with using the Platform.
User Guide	Opens an online PDF of the most current software documentation.

Contact Us	Takes you to www.visualaction.com/contact , where you can contact us for help.
------------	--

3.5 Workflow Drawer

The Workflow Drawer houses all of the Workflow Extensions available for use within an application. The Workflow Drawer is the hub from which a user begins creating the interactive components of an application.



When designing an application, a user may select whichever Extensions are preferred from the Workflow Drawer to use within the application. There is no limit on the number of Extensions a user may add to an application, but it's important to keep in mind the usability of an application when deploying a large number of Extensions. Too many Extensions packed into a single interface may make it difficult to navigate an application. Try making multiple apps!

3.5.1 Opening and Closing

There are multiple ways to access the Workflow Drawer from within the Platform environment:

Pull Tab

While in Designer Mode, a small pull tab is always present at the bottom of the Desktop, in the center:



Clicking on this tab opens the Workflow Drawer. When the Drawer is open, clicking this tab will close the Drawer.

Menu

The Workflow Drawer can be toggled from within the Menu as well. Open the Menu and select "Workflow Drawer" (see section 3.4 Menu).

Shortcut

The Workflow Drawer can also be toggled using the keyboard shortcut `Ctrl + Alt + E`. See section 3.8.3 Keyboard Shortcuts (Keybinds) for more keyboard shortcuts.

The Workflow Drawer can always be closed by clicking the `x` button in the upper right corner of the Drawer when it is open.



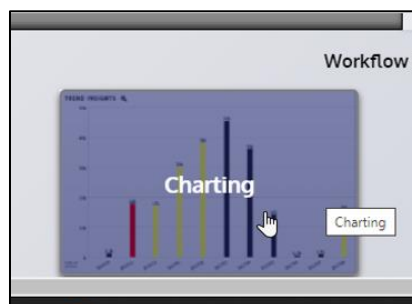
The Workflow Drawer will automatically close after deploying a Workflow Extension or when another panel, such as the Editor, is opened.

The Workflow Drawer is disabled in published applications, meaning the above options for accessing it are also disabled. The pull tab will be hidden.

3.5.2 Navigating the Drawer

Once opened, the Workflow Drawer will present all of the Workflow Suites which are available to the user. Workflow Suites are available if there is at least one Extension within that Suite.

Begin by clicking on the Suite relevant to the type of Workflow Extension you need. For example, to add a Flaremap, click on the "Flaremap" Suite; to add a bar chart, click on the "Charting" Suite.

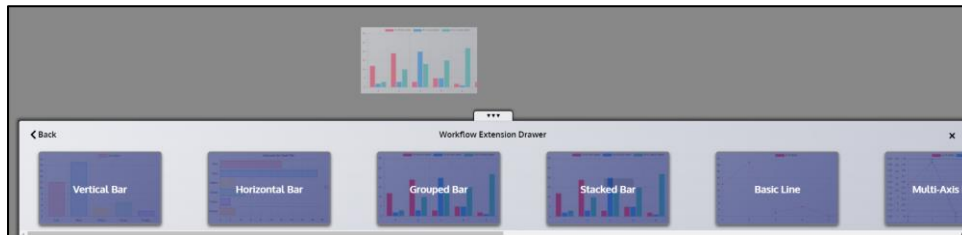


Once a Suite has been selected, the Workflow Drawer will populate with all of the available Workflow Extensions within that Suite.



3.5.3 Deploying Extensions

Once the Drawer has been populated with the contents of a Workflow Suite, locate the Extension you would like to add to the application. Once located, click and drag the Extension's card out of the Drawer and onto the Desktop environment. A "ghost" instance of the card will follow your mouse as you drag, indicating that the card has been grabbed. Move the cursor to a place relatively near where you would like the Extension to be, then release the mouse button. Upon releasing, a new instance of the selected Extension will be deployed to the interface.



Once deployed, the new Extension can be arranged and sized to best fit the interface as desired.

3.5.4 Getting More Extensions

In the Workflow Drawer, while looking at the available Suites of Workflow Extensions, a card titled "+ Get More Extensions" will always be available at the end of the row. Click this card, or manually navigate to www.visualaction.com/contact/ to contact Visual Action about obtaining more Extensions. The "Contact Us" option under "Help" in the Menu will also take you to this page.

3.6 Editor

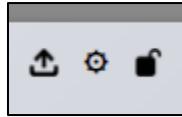
The Editor panel is the most dynamic and interactive panel in the Visual Action Platform. The Editor changes to include the modifiable settings for almost any editable component of an application, including Workflow Extensions, Application Settings, and User Preferences.



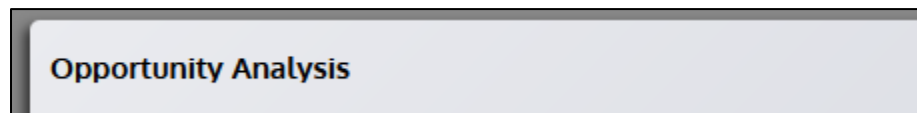
The Editor appears on the left edge of the screen. When clicking on a button that opens the settings for some application component, the Editor will automatically open to reveal the appropriate settings. The Editor can be closed by clicking the × button in the upper right corner. The Editor will automatically close if a conflicting panel, such as the Workflow Drawer, is opened.

3.6.1 Workflow Extension Settings

Workflow Extension settings are likely what will be edited most often within the Editor. Each Workflow Extension developed within the Visual Action Platform, excluding a few special Extensions, has user-controlled settings that allow the user to make the Extension look, feel, and function as desired. An individual Extension's settings may be accessed by clicking the gear icon (⚙️) in the upper right corner of the Extension.

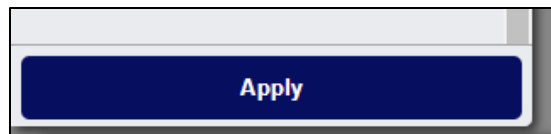


After clicking an Extension's settings (gear) icon, the Editor panel will open and/or update to reflect all of the available settings for that Extension which the user may modify. Except for a select few Extensions, the first setting will always be the Extension Name (the title shown within the header of the Extension).

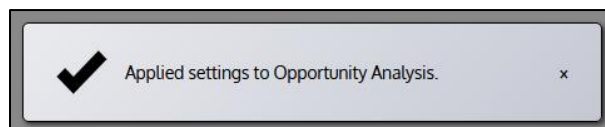


If the Extension has many settings, the panel will allow the user to scroll down to see all of the available options.

An "Apply" button is fixed at the bottom of the Editor panel. After making desired changes to an Extension's configuration, click this "Apply" button to execute the changes. The Workflow Extension whose settings are being modified will automatically update itself with its new settings.



NOTE: Certain settings changes may cause no visible changes to an Extension, and it therefore may be difficult to recognize that an Extension was successfully updated. A notification should appear in the bottom right corner of the interface indicating that an Extension's settings have been successfully updated.



NOTE: There is no "Apply" button when adjusting application settings and preferences. See the next section for more information.

3.6.2 Application Settings and Preferences

There are many settings and preferences for each application that may be controlled by the user. These can be accessed via the Menu, by clicking "Settings and Preferences."

Once clicked, the Editor panel will open with all of the available options which a user may control. The user may make any desired changes; changes are automatically saved. There is no need to press an "Apply" button, as with Extension settings.

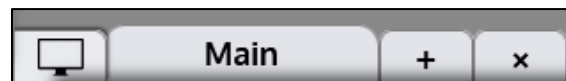
See section 4.7 Adjusting Settings and Preferences for further information on the available application settings and user preferences.

3.7 Page Manager

Oftentimes, multiple, separated pages are needed within the same application to space out components or divide them according to their place in a process. The Page Manager is where application pages can be added, edited, and removed.

3.7.1 Opening and Closing

To open (or to close) the Page Manager, navigate to the lower-left corner of the screen. A small tab with a desktop icon () is present. Clicking this tab will toggle the Page Manager, which slides up from the bottom of the screen and contains a tab for each existing application page as well as some additional control buttons. By default, only one page, titled “Main”, exists in an application.



3.7.2 Adding an Application Page

Adding a new page to an application is simple. With the Page Manager open, click the + button. A new application page will be created, along with a tab in the Page Manager to represent it. The application will automatically turn to the newly created page, which is blank (so don't worry when any existing Extensions seemingly disappear).



You can now add Flaremaps and Extensions to this new application page.

Note: Each Platform application can contain up to five (5) different pages.

3.7.3 Editing a Page Name

Application pages (including the main page) can be renamed. To rename an application page, locate its tab in the Page Manager, then double-click on its name. This will reveal an input field.



Enter the desired name for the page in this input box. Page names can have a maximum of 10 characters.



Once a name has been entered, click anywhere outside of the input field. The new name will be applied to that page and will be reflected in its tab in the Page Manager.

Note: Renaming a page **does not** affect the overall application title displayed in the Navbar.

3.7.4 Deleting a Page

Before deleting a page, be sure that you no longer want any of the Flaremaps or Extensions which exist on that page. Once the page is deleted, all elements on that page (and their corresponding configurations) will also be deleted, and this action cannot be undone.

To delete a page, first click that page's tab in the Page Manager. The page you want to delete must be selected as the active page (this will help verify that there is nothing on the page you want to save).



Then, click the × button at the right end of the Page Manager to delete the page. The page will be deleted and its tab removed, and you will be automatically returned to the main application page. Again, be sure you want to delete the page you are on; the Platform will not double-check before deleting the page when this button is clicked.



Note: The main application page (always the first tab in the Page Manager) may not be deleted.

3.8 General

3.8.1 API Keys

Various Workflow Extensions require API keys to function. The specific documentation of a Workflow Extension will describe the key(s) you or your organization must possess for the Extension to function properly.

API keys are stored in a common file to make it easy to configure them once for all apps. This file only needs to be updated to add or change keys when keys are updated or new Extensions are added.

The key file is located in the Platform installation directory, under `/config/keys.js`.

NOTE: This is a slightly technical process. Do not edit this file if you are not sure what you are doing or if your organization has already set it up for you. **Do not** edit any portion of the file other than what is described below or inside a specific Workflow Extension's documentation. **Do not** edit the filename.

This file may be edited with any text editor. It contains a simple JSON object named "APIKeys". To insert your organization's key, locate the identifier for the Workflow Extension(s) you are adding a key for. It will be to the left of a colon within all the data inside curly braces ({ }).

To the right of the colon next to that identifier, there will be double quotes. If they are empty, paste the key in-between the quotes. If there is an existing key, replace the entirety of it, within the quotes, with the new key. Quotes must remain present. Do not delete the comma after the end-quote. See the image below for an example of how one line should look when entering a key. The highlighted text shows where you would place your API key.

```
1 var APIKeys = {  
2   GoogleMaps: "[your_API_key]",  
3 }  
4
```

Your key file may contain more rows between the curly braces, depending on the Extensions you have. Again, each Workflow Extension's documentation will specify what should be included in this file to achieve functionality.

3.8.2 Hover Info

There are various buttons and items within the Platform that do not have labels. They are described within this Guide, but while using the Platform, if you question what an item is, hover the cursor over the item for a second. A small box will appear with a label, indicating what the item is.

3.8.3 Keyboard Shortcuts (Keybinds)

To make interaction with the Platform most convenient for the user, various components may be accessed quickly using the keyboard shortcuts provided in the table below.

Component	Shortcut
Settings and Preferences	Ctrl + Alt + S
Menu	Ctrl + Alt + C
Workflow Drawer	Ctrl + Alt + E
Data Manager	Ctrl + Alt + D
Reporting Manager	Ctrl + Alt + R

4 Applications

This section details the creation of applications as well as many of the individual controls available to users while designing and using applications. The previous section intended to describe where components are found within the Platform, while this section will provide much more detail on the specific features and usage of those components, and is written from a training perspective.

4.1 Getting Started

Different users might begin their work on applications from different starting points. Use the sections below to guide you to the relevant starting point within this guide.

Creating a Brand-New Application

To begin a brand-new application from scratch, continue on to the next section, 4.2 Creating a Brand-New Application.

Be sure to have at least one data source available and accessible before continuing.

Beginning From a Template Application

If you are beginning from a Visual Action Template Application, continue to section 4.3 Beginning from a Template Application.

Be sure to have at least one data source available and accessible before continuing.

Editing an Existing Application

If you are editing an existing application, be sure you have an editable copy of the application (.vas file extension) and not a published version of the application (.vap file extension).

If you are editing an existing application that:

- might need to be reverted to its previous state,
- only needs temporary edits,
- has high importance to you and/or your organization,
- was not originally created by you, or
- needs to be preserved for any other reason,

please be absolutely sure you have your own **copy** of the editable version of the application, and not the only copy or the original copy. In other words, please make sure the original application is **backed up** before editing. Visual Action Software, Inc. is not responsible for the loss of applications due to overwriting, editing, or failure to back up.

Once you have your editable copy of the application, continue on to section 4.4

Editing an Existing Application. Sections prior should be referenced if changes are needed to components of an application that are typically set at creation and often unchanged, such as the update interval for the data. This section assumes that most edits to existing applications involve Workflow Extensions.

4.2 Creating a Brand-New Application

If it's the first time you have used the Visual Action Platform on the device and browser you are using, it is likely you will by default be met with a blank, brand-new environment. Otherwise, the Platform attempts to load the most previously used application.



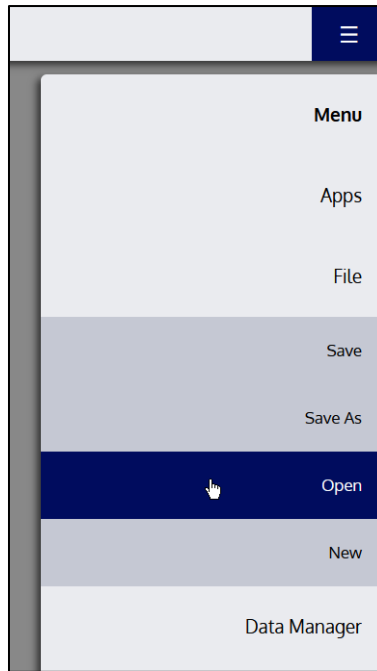
This can also happen if the browser's local storage was cleared or if some hardware or software event causes the Platform to be unable to read the data related to the previously used application.

In all other instances, an application likely exists in the environment when you navigate to the Platform. Use the Menu to select "File," then "New/Clear" to clear the environment and create a new application. The Platform will prompt you for confirmation before resetting the Platform environment.

Once confirmed, the Platform will reset everything to a new application state, and it is ready for you to begin creating!

4.3 Beginning from a Template Application

If you're starting from a template application from Visual Action, begin by opening the template application file from within the Platform. Open the Menu, click "File" > "Open," and locate the template file you were given.



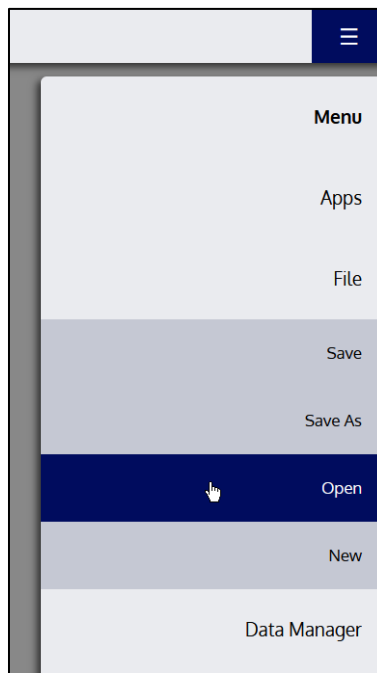
The template application will load. Continue on to the next section to learn how to modify the existing components of the application.

4.4 Editing an Existing Application

If you want to make changes to an existing application, there are multiple options.

4.4.1 From a File

To edit your application from a saved copy, open the editable version of the application by opening the Menu and clicking “File” > “Open” and locating your `.vas` file for the application.



Once the Platform has loaded your application, continue on to section 4.5 Naming the Application to learn how to edit all parts of your application, or on to section 4.8 Adding, Placing, Collapsing, and Removing Extensions to skip directly to Flaremaps and Workflow Extensions.

4.4.2 From Last App

The Platform also provides an “Open Last App” option that will recover and load the most recently edited application in the Platform. The Open Last App feature intends to provide a convenience to users by making it easier to access a recently used app in the event of a tab closure, browser crash, failure to save, or other event, without requiring the user to locate the application’s save file to reload it.

The Platform-managed Last App has the benefit of always keeping the most up-to-date copy of a recently edited application, as the Platform automatically saves applications in the browser whenever a change is made. However, there is a downside. The Platform cannot automatically write the save data out to a save file without user-permission. Thus, it is still important to remember to save a local copy of your app (in a save file) periodically, as if the browser’s Local Storage is ever cleared, or if the app you are trying to edit is not most recently used apps, it will not be accessible to you using the Last App feature. Remember: the Last App is a convenience, not a failsafe.

To locate the Last App, open the Menu, select “Apps,” then select “Open Last App” to open it.

Note: Selecting “Open Last App” will automatically open the most recently saved app, and it is consistently updated with each edit made to *any* application. Therefore, if you are working on multiple apps across multiple Platform sessions (tabs or windows), using this feature might overwrite the app you are working on if the last edit was made to a different app. Be sure to save all your apps to a file for the most reliable preservation.

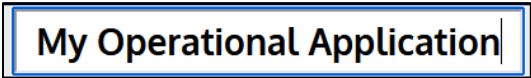
4.5 Naming the Application

Before diving into the bulk of creating an application, it's easy to initialize it by giving it its own title. New applications will begin with a default title of "Visual Action Platform"



Visual Action Platform

To edit the title, locate the existing title in the center of the Navbar at the top of the screen. Hover over the title to see the cursor change, indicating that it may be edited as text, then click to reveal a text input box. Enter an appropriate name for your application, then click anywhere else on the interface to stop editing.



My Operational Application

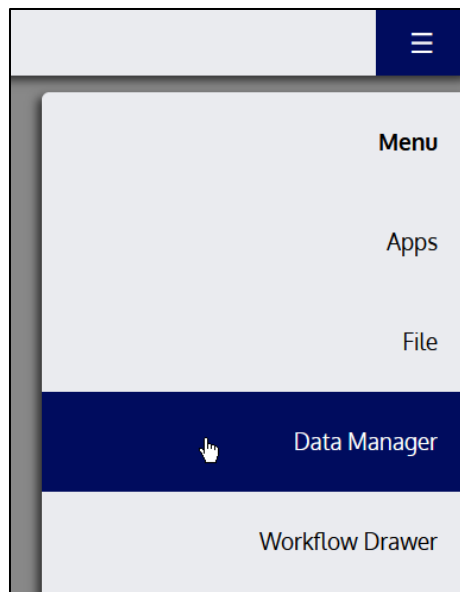
NOTE: Titles must have at least one character. If the title is left blank, the last remaining character in the text input box will be replaced.

If you need to edit the application's title later, or want to skip this step for now, this can always be done while in Designer Mode using the same procedure.

4.6 Connecting to Data

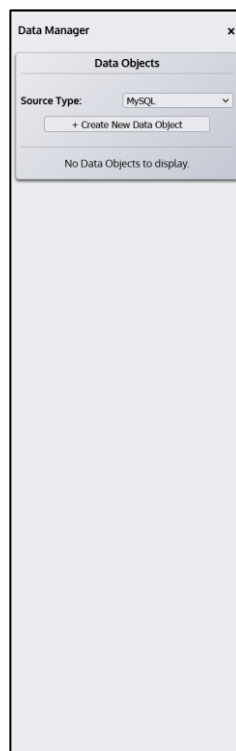
When beginning to build a new application, the most critical early step is to connect the application to at least one data source so you have some data to build your application around.

Start by opening the Menu and navigating to “Data Manager,” or by pressing `Ctrl + Alt D`.



4.6.1 Data Manager

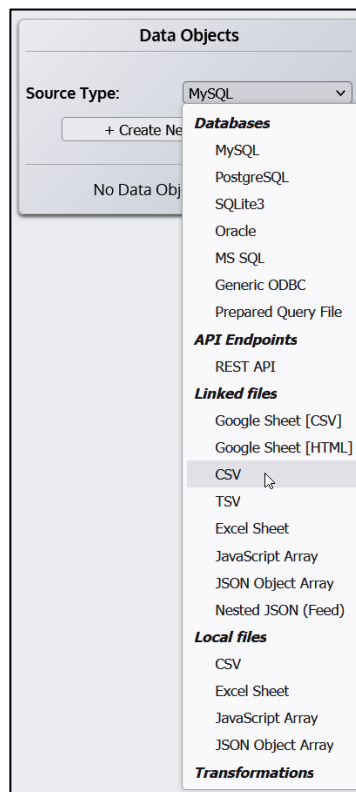
The Data Manager within the Platform is where all Data Objects and their related information may be managed. A Data Object represents a collection of information relevant to a source of data, including its source location, live update settings, column parameters, and more. Once you've followed the above instructions, you'll see the Platform Data Manager open to the left, in the Editor Panel.



By default, in a new application, there will be no Data Objects present in the Data Manager. Continue on to section 4.6.2 Creating a Data Object to learn how to create a new Data Object and connect it to your data.

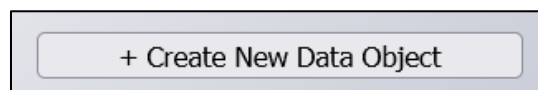
4.6.2 Creating a Data Object

To create a new Data Object and connect it to your data source, begin by selecting the type of data source you are connecting to from the “Source Type” dropdown at the top of the Data Manager.

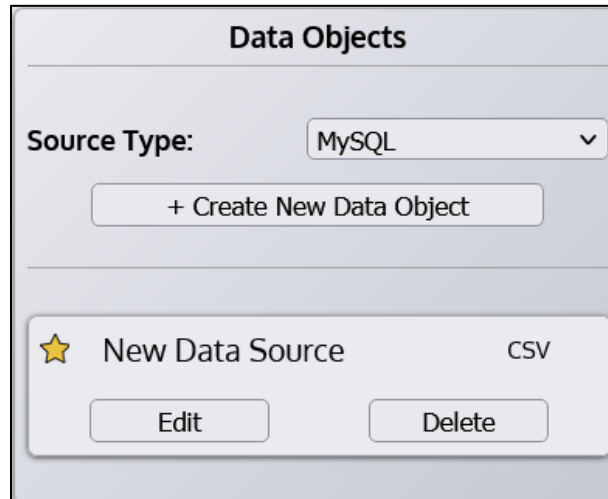


The Platform supports several types of data sources, including Databases, linked files (URLs), and local files. Note that local files, which are uploaded from a local source (e.g.: on the disk of your computer), are intended primarily for use as “test files.” Local files will need to be loaded manually each time the page is refreshed, as security restrictions prevent the browser from automatically accessing the storage on your local system.

Once you’ve selected the source type for your data, create the new Data Object for it by clicking the “+ Create New Data Object” button below the “Source Type” dropdown.



A new Data Object tile, representing the Data Object you just created, will appear in the list of Data Objects.



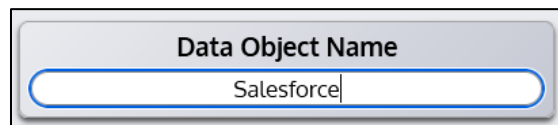
You can create multiple Data Objects, and each will have its own tile listed here. The list of Data Objects makes it easy to see and manage all of the Data Objects you have created within an application. From this list, you can see the names and data source types of each Data Object, select a Data Object to be the default, and choose to edit or delete a specific Data Object. More details on each of these options follow in the next sections.

4.6.3 Editing a Data Object

To begin connecting your new Data Object to data and setting it up for use, click “Edit” on the tile of the Data Object you just created (named “New Data Source” by default) to open it for editing. Once you’ve opened a Data Object for editing, you’ll see a list of several different configuration groups relating to that Object. Most of these are the same for any Data Object, but some parameters are different depending on the data source type to which you are connecting.

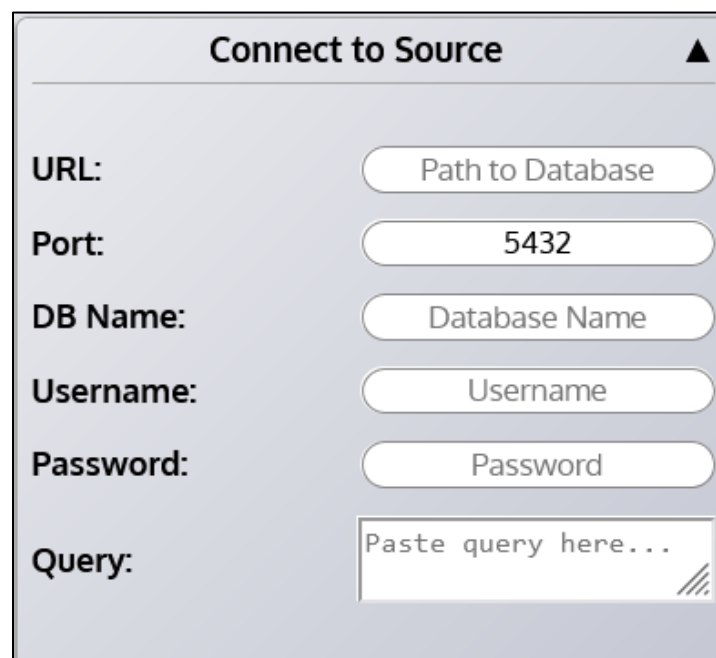
4.6.4 Naming a Data Object

The first option presented when editing any Data Object is the “Data Object Name”. Use the input field (containing “New Data Source” by default) to enter a recognizable name for your Data Object.

A screenshot of a user interface element titled "Data Object Name". It features a light gray rectangular box with a thin black border. Inside the box, at the top, is the text "Data Object Name" in a bold, dark font. Below this text is a horizontal input field with a blue border and rounded ends. The input field contains the text "Salesforce" in a standard black font, with a vertical cursor line positioned at the end of the text.

4.6.5 Connecting to a Data Source

There are various methods for connecting your Data Object to its data source, depending on the type of source you are connecting to. After you’ve instantiated a new Data Object of a specific type, once you click “Edit” to modify it, the appropriate parameters for that type of Data Object will appear in the “Connect to Source” section.

A screenshot of a dialog box titled "Connect to Source" with a small upward-pointing triangle icon in the top right corner. The dialog box has a light gray background and a thin black border. It contains several labeled input fields arranged vertically. The labels are in a bold, dark font: "URL:", "Port:", "DB Name:", "Username:", "Password:", and "Query:". The corresponding input fields are rounded rectangles with a light gray background and a thin black border. The "URL:" field contains the text "Path to Database". The "Port:" field contains the text "5432". The "DB Name:" field contains the text "Database Name". The "Username:" field contains the text "Username". The "Password:" field contains the text "Password". The "Query:" field is a larger text area with a light gray background and a thin black border, containing the text "Paste query here..." and a small icon of three diagonal lines in the bottom right corner.

Following are the descriptions of the general parameters required for each category of data source type. Note that some specific sources require additional information (for which input fields will be presented to the user) that others don't, such as Excel Sheets, which require the name of the sheet being referenced.

Databases

Generally, there are up to six parameters required for connecting to a database.

1. **URL:** The URL path at which the database can be reached.
2. **Port:** The port on which the database is serving.
3. **DB Name:** The name of the database to access.
4. **Username:** A valid username with permission to access the database.
5. **Password:** The password for the previously entered username.
6. **Query:** The query to submit to the database in order to retrieve data.

There may be other parameters required. For example, when connecting to a database using ODBC, a "Driver" field will appear. You must enter the driver name *exactly* as it appears on your server or local system in order for the ODBC connection to be able to find and connect to your database using it.

Linked files

Generally, linked files only require one parameter. That parameter is the URL at which the data source may be accessed.

The entered URL must be the full path to the data source, unless it is locally available on the machine and can be accessed with a relative path.

Some linked file types require an additional parameter:

- **Excel Sheets:** Requires a "Sheet name" parameter, which is the name of the sheet within the Excel workbook from which to gather data. Default's to "Sheet1," the default name of the first sheet in an Excel workbook.

Local files

Generally, local files also only require one parameter: the path to the file. Unlike with linked files, the path will be determined using a file explorer, allowing the user to select data from his or her local files.

Please note: due to security constraints, local files cannot be automatically accessed when the page is refreshed or an application is reloaded. For privacy-related reasons, the Visual Action Platform *does not* save any data within the application files. Therefore, it is recommended that local files only be used for designing/testing applications, and not for final version of applications which will be released for use. Flat files will need to be manually reloaded on page refresh.

Some local file types require an additional parameter:

- **Excel Sheets:** Requires a "Sheet name" parameter, which is the name of the sheet within the Excel workbook from which to gather data. Default's to "Sheet1," the default name of the first sheet in an Excel workbook.

Once you've entered the necessary parameters to connect to your data source, click the "Connect" button at the bottom of the Data Manager Editor panel. Read on to the next section to find out how to use the "Connection Status" section of the Data Manager to determine if the Platform has been able to retrieve your data.

NOTE: Data Object connection parameters may be edited at any time. If the Platform is unable to retrieve your data, or if the parameters for reaching that data change, simply edit the Data Object, check the connection parameters, and adjust accordingly. Whenever a connection parameter changes, the Platform will automatically disconnect from the data source. You must click "Connect" again to initiate a new connection given the new parameters.

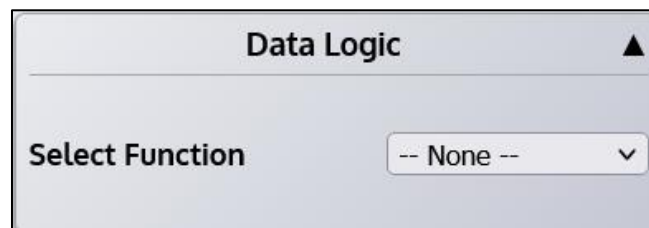
4.6.6 Using Custom Data Logic

The Logic Layer of the Visual Action Platform enables custom logic to be used to process and manipulate data before it is provided to the rest of the application.

Logic can be implemented in a variety of ways, but ultimately rolls up to a singular function that is called by the Platform when loading a data object. To read more about the technical details of writing and configuring custom data logic, see section 4-A.2 Data Logic.

Collected data is passed to custom data logic functions **before** it is processed and used by the Platform, which is why the “Data Logic” configuration appears prior to the “Connection Status” box.

To have a Data Object’s data passed through a custom function, simply select the function you want to call on the data from the “Select Function” menu under “Data Logic.” Then, whenever data for that Data Object is collected, the Platform will first pass it to the chosen custom function.

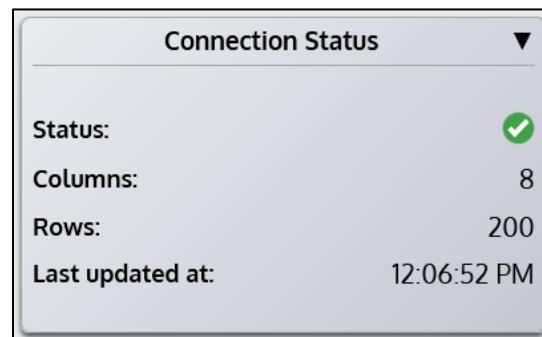


By choosing nothing, or selecting the “–None –” option, the Platform will bypass the custom logic step and continue to prepare the data normally for the application.

Tip: If you need to call multiple functions on a single Data Object, group those calls into a single master function in your Logic file, then call that master function from the Data Object editor.

4.6.7 Connection Status

The "Connection Status" of a Data Object is important for monitoring the Platform's connection to your data source. This section indicates whether or not the Platform has actually collected data from your data source (rather than simply if it is able to reach the source), as well as how much data was collected and when.



Once you have pointed the Platform to your data source and clicked "Connect," look to the "Connection Status" section to verify whether the connection was successful. Please note, depending on several factors, there may be a delay between the time you click "Connect" and the time the Platform has processed all the data and updated this section.

Below are the descriptions of each piece of information in the "Connection Status" section and how each can be used to verify proper connection to your data.

Status

The "Status" indicator informs you of whether or not the Platform has data, from your source, loaded into this Data Object. There are two icons which the "Status" indicator may present:



DISCONNECTED: Indicates that data was not successfully loaded from your data source, and therefore no data is present within the current Data Object.



CONNECTED: Indicates that data was successfully loaded from your data source, and there is usable data present and ready within the current Data Object.

Connection Status ▼	
Status:	✖
Columns:	--
Rows:	--
Last updated at:	Never

Connection Status ▼	
Status:	✔
Columns:	8
Rows:	200
Last updated at:	12:06:52 PM

If your Data Object displays the red "X" icon, indicating no data, the cause may be one of the following issues:

- The Data Object is newly created, and the "Connect" button has not yet been pressed to initiate a connection to the data source.
- One or more of the connection parameters in the "Connect to Source" section is incorrect, and the connection to the data source could not be properly made.
- The connection parameters are correct, but the Platform is unable to retrieve data from the source due to a failed connection or denial of permission from the source.
- If the Data Object already existed, and it successfully loaded data previously, the location of the data source may have changed since the last time it loaded, and the connection parameters should be updated.

If none of the above circumstances are applicable to your situation, and you are still having trouble connecting to your data source, please contact support.

Columns

The "Columns" indicator informs you of the number of data columns the Platform read when loading data. When the "Status" indicator shows a "Connected" status, this will update with a number reflecting the number of collected columns in the data. When data is "Disconnected," this will show "--". Use this value to confirm that the correct number of columns were loaded from your data source.

If the value does not match what you expected to come from your data, there may be an issue with your connection parameters or with the permissions allowing some data to be shared from your source.

Row

The "Rows" indicator informs you of the number of data rows the Platform read when loading data. When the "Status" indicator shows a "Connected" status, this will update with a number reflecting the number of collected rows in the data. Please note: this count does not include the header row from the data. When data is "Disconnected," this will show "--". Use this value to confirm that the correct number of rows were loaded from your data source. If the value does not match what you expected to come from your data, there may be an issue with your connection parameters or with the permissions allowing some data to be shared from your source.

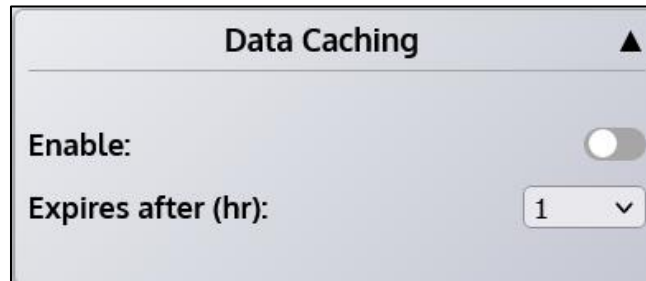
Last updated at

The "Last updated at" indicator presents the last time (in local time) data was retrieved from the source. If connecting to a source for the first time (i.e.: in a new Data Object), this will show "Never," otherwise it will show the last time data was updated. This time will be updated each time data is loaded. Use this time to see that data was updated when you loaded the app, refreshed the page, or when live updating is on.

If the "Connection Status" section of your Data Object looks good, continue on to finish setting up your Data Object.

4.6.8 Data Caching

Every Data Object within the Platform may enable data caching. Data caching allows the Platform to cache the last fetched data in browser storage (never in the application file) and skip retrieving new data until a specified expiry period is met (or the option to cache data is turned off).



Before enabling data caching, it is important to understand what data caching does and how it works. Data caching helps reduce load times when refreshing an application and can also be used to prevent excessive queries against a data source when it's likely that there is no new data available yet. For example, if a source of data in an application only gets new data once a day, but your application live updates every hour (perhaps to fetch updated data from an additional source), you can have data for the daily source cached so that the Platform only retrieves data from that source once per day. There are several other uses for data caching, but these are some of the primary use cases.

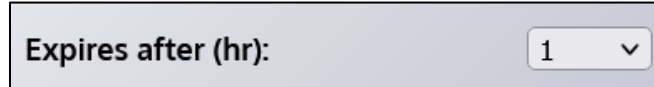
Data caching is always off for a Data Object by default. A user must enable data caching before any data will persist. When data caching is enabled, the Platform will save the currently loaded dataset into an object in the browser's storage. Normally, no data is saved in an application file or in persistent browser storage as a security feature. Even with data caching enabled, no data is stored in the application file, however, it will be persisted in the browser's storage until the cache expiry time. Under certain circumstances, this storage may not be automatically cleared. No data that the Platform stores in browser storage is ever sent or used outside of the Platform environment by the Platform itself. Nonetheless, it is important to ensure that, prior to using data caching features in your application, it is not a violation of any relevant security policies of your organization to store sensitive data in persistent browser storage.

Once data caching is enabled, the Platform will store the current data for the specified interval. During this interval, unless data caching is turned off, when the page is refreshed, the Platform will retrieve this stored data rather than retrieving the latest data from the source. If the page is refreshed and the expiry time has been reached, the Platform will fetch data from the source. At this point, if data caching is still enabled, it will store the new data and reset the cache expiry time.

To enable data caching, toggle the “Enable” switch under the Data Object’s “Data Caching” section.



Beneath this toggle, there is a dropdown menu labeled “Expires after (hr).” Use this dropdown to select how long cached data should persist before the Platform will retrieve data from the source again. You may select as few as one hour and as many as 24 hours (1 day).

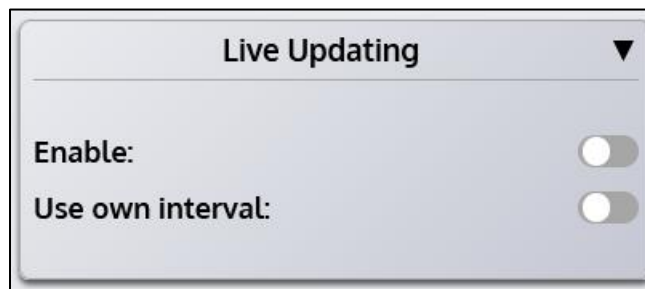


At any time, you can disable data caching for a Data Object by toggling off the previously referenced “Enable” switch in the Data Object’s “Data Caching” section. Once disabled, the Platform will continue to fetch data from the source each time the page is refreshed or the application is loaded.

As with most Data Object settings, data caching applies to each Data Object individually. Data caching must be enabled or disabled within the settings for each Data Object to which you want it applied. Different Data Objects can have different settings for whether their data is cached or not.

4.6.8 Live Updating

Every Data Object within the Platform may be registered for live updates. Live updates allow the Platform to automatically recollect the most recent data from a data source on a specified interval.



Before setting up live updates for a Data Object it is important to note that there is also a global toggle for Live Updating. Global settings apply to the entire application rather than to a specific component. Global Live Updating *must* be enabled for a Data Object to live update, even if live updates are enabled within the Data Object itself. See section 4.7.1 Data Services for more details on global Live Updating.

To enable a Data Object to live update, toggle the “Enable” switch under the Data Object’s “Live Updating” section. Again, remember that global Live Updating *must* be enabled in the Settings and Preferences menu before this Data Object will be able to automatically update (if global updating is not enabled, the toggle switch will still indicate that live updating is enabled for the Data Source, but updating will not occur).

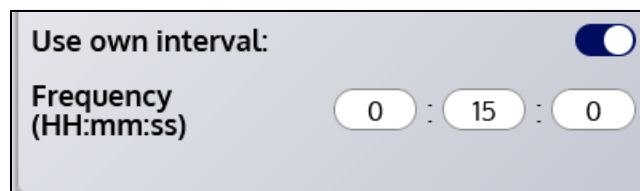


Enabling live updating on a Data Object will, by default, allow it to update on the global interval (set in the Settings and Preferences Menu). All Data Objects set to update on the global interval will update at the same time, when the global interval passes. Each individual Data Object, however, may be assigned its own interval for updates. This is useful if certain data needs to be updated more or less frequently than other data within an application.

To assign a custom interval to a Data Object, toggle the “Use own interval” switch under the Data Object’s “Live Updating” section.



Toggling this switch will reveal a “Frequency” input, where you can enter the hours, minutes, and seconds desired between data updates. The minimum frequency is 5 seconds, and the input will automatically update to this minimum if a lower time is entered.



Please note: When modifying a Data Object’s live update settings (toggling a custom interval on or off or changing that interval), live updating for the Data Object will automatically be disabled (you will notice the “Enable” switch turn off). This is to prevent issues with update timing that can often be confusing while modifying these controls (e.g.: accidentally setting too short of an interval and causing the data to be updated too rapidly while trying to adjust Data Object settings). Remember to re-enable live updating for your Data Object once you’ve made the desired changes to its update parameters.

4.6.9 Column Data Types

Once you have successfully loaded your data into the application, it’s important to verify that each column of data is properly configured. The Platform will perform auto-detection when loading data, during which it will determine the column headers (names or keys) and make its best predictions as to an appropriate alias (friendly name) and data type for each column. It’s important to review these before continuing to ensure that the Platform properly recognizes the type of each data and that the friendly names are something that the users of your application will recognize and understand.

When loading data from a source, after clicking “Connect” to fetch the data, you will see the bottom section, “Column Data Types,” populate with a table displaying each column from the data and each column’s associated configurations. If the data object already existed and was connected to data, this table may already be present, and may update if the Platform recognizes changes in the data since the last update.

Column Data Types ▼			
Sales_Rep	★	Sales Rep	TEXT ▼
Sales_Manager	☆	Sales Manager	TEXT ENUM ▼
Sales_Region	☆	Sales Region	TEXT ENUM ▼
Total_Sales_Q4	☆	Total Sales Q4	NUMBER ▼
Quota_Q4	☆	Quota Q4	NUMBER ▼
Quota_Percent	☆	Quota Percent	NUMBER ▼

Each row of this table contains four parts: the original column name (also referred to as a column key), the “Primary Key” button, the column alias (friendly name), and the column data type. Below are detailed descriptions of each of these parameters.

Column Name

The column name is the identifier (header, key, etc.) for a column of the data. This comes directly from the data and is used to reference that data, and thus cannot be edited. It is shown at the top-left of each row in the columns table.

Primary Key

The “Primary Key” button is a ★ (star) icon that is used to indicate which column of the data represents the Primary Key. The Primary Key of a dataset is the column key (name/header) which is associated with a column that contains *only* unique entries. In other words, it represents a column which can be used to uniquely identify a row within the dataset because there are no duplicates in the

column (each value can be uniquely matched to a single data row). The Platform will attempt to automatically determine the column which seems to be the Primary Key of the data, but it is important to ensure that the appropriate column is selected. Only one column may be used as the Primary Key.

Alias (friendly name)

An alias, or a friendly name, is a name you may enter that is easier or more comfortable to read while using the application. For example, "Sales Rep" might be much more comfortable to read within an application than the actual column name in the data, "sales_rep". When you load your data, the Platform will do its best to automatically create an alias from each column name.

For each column in the data, you may enter any custom name you desire. Keep in mind that it may be best to use organizational terms that other users will understand if they are to use the application.

Data Type

The data type for each column tells the Platform what kind of information is contained in that column. There are four general types used in the Visual Action Platform:

Data Type	Description
TEXT	Regular text information in any format, typically without (many) repeats of the same value across multiple rows or which does not need to be categorized based on the value.
TEXT ENUM	Text data that is enumerable, meaning that there are repeats of values across multiple rows. This data type indicates that the text data in that column is not typically unique to each row, and is used for options that you would like to Group by in a Flaremap. Select this type if you have repeating text values in that data column.
NUMBER	Any purely numerical data.
DATE	Dates, which are text data formatted into any standardized date format.

It is very important to select the appropriate data type for each column. These can always be changed later, but it is very helpful in application design to ensure that data types are properly associated from the start.

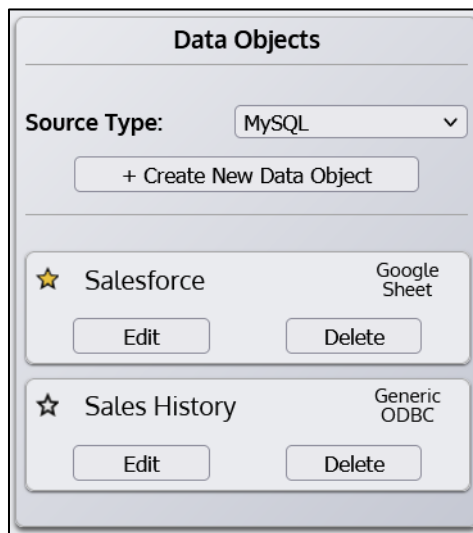
The Platform will auto-detect data types for each column to the best of its ability, but you should check each column to verify that the data types are correct. If they are not, or you would like to make a custom change, select the appropriate data type from the dropdown list.

Once you have changed and/or verified the information in the Column Data Types section, your Data Object should be ready to use! You can return to the main Data Objects panel of the Data Manager to continue creating or modifying Data Objects, or you can close the Data Manager and move on to creating your application.

4.6.10 Choosing a Primary Data Object

If you've created multiple Data Objects for use in your application, you can choose which to deem the "Primary" data object. The Primary Data Object is the data object by which Flaremaps and Workflow Extensions are driven by default. You can always change which Data Object an Extension or Flaremap is using in its own settings, but it's helpful to choose the most frequently used Data Object as the Primary.

In the Data Manager, under "Data Objects" where each of the Data Objects you've created shows up, there will be a tile for each Data Object. Each tile will contain a ☆ (star) icon used to indicate which Data Object is the Primary. If you only have one Data Object, this will automatically be selected and may not be changed. If you have multiple Data Objects, the first one you created will be marked as Primary by default.

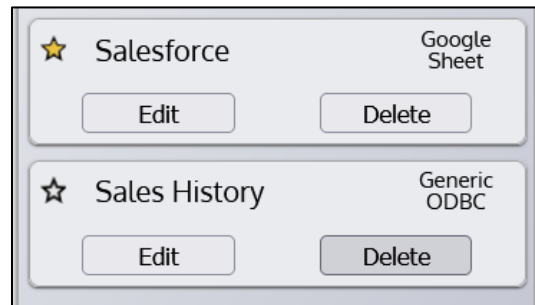


To change which Data Object is the Primary, simply select the ☆ (star) icon in the tile of the Data Object you want to make Primary. If you delete a Data Object that is marked as the Primary, a new Primary will automatically be selected.

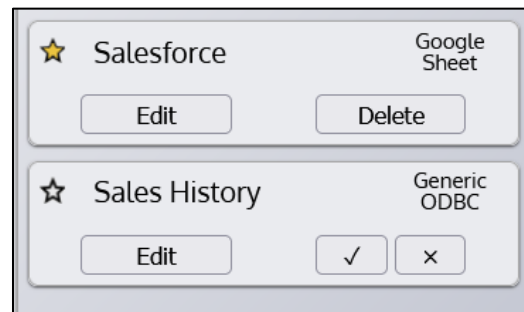
4.6.11 Deleting a Data Object

If you need to remove a Data Object from your application for any reason, simply open the Data Manager and locate the tile for the Data Object you'd like to remove.

Click the "Delete" button on that Data Object's tile.



You will be prompted to confirm deletion with another pair of buttons.



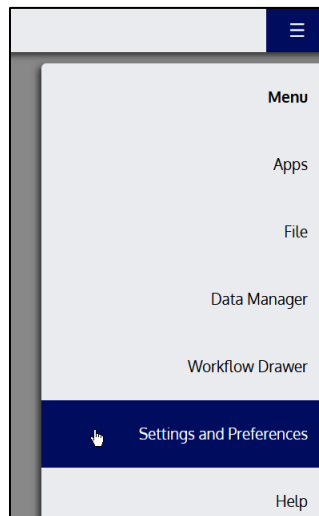
To confirm deletion, click the ✓ (checkmark) button. To cancel deletion, click the × (cancel) button.

NOTE: Deletion of a Data Object cannot be undone. The only way to recover your Data Object is to retrieve it from a saved copy of the application, or to recreate it. While the Data Manager does **not** modify data at the source, it still may be inconvenient to remove a Data Object from your application while you still need it. Be sure you want to delete a Data Object before confirming.

4.7 Adjusting Settings and Preferences

It's almost time to begin building your application, but there are some important and helpful options in the Settings and Preferences to be familiar with first. Some options do not need to be adjusted yet, but many may be useful to use during the design of your application.

Navigate to the Menu and click "Settings and Preferences."



The Editor panel will open to show you all the available controls you have over the application and its performance.



Adjust any desired settings and preferences to meet your needs. Use the information below as a reference for information about the function of the available options.

When complete, close the Editor panel by clicking the **x** in the upper right corner of the panel. Changes to Settings and Preferences are automatically saved.

4.7.1 Data Services

Data Services contains controls that allow you to manage how the application handles data in the background.

Service	Option	Description
Live Updating	Enabled	Toggle switch indicating whether live updating should take place or not. Entirely enables or disables live updating for the application; if disabled here, no Data Objects will be updated, even if updating is enabled within that specific Data Object.
	Frequency (HH:mm:ss)	Indicates the global interval on which the application should automatically check for data updates, in hours, minutes, and/or seconds. Minimum of 5 seconds.

4.7.2 Look and Feel

Look and Feel contains controls allowing you to set up the application however is most comfortable for you and its users to interact with it. This includes controls that make it more comfortable during editing as well as during usage.

Service	Option	Description
Logo	URL	Allows input of a URL to a custom image to be used as a logo for the application (appears in the Navbar, in the upper-left corner).
Theme	Theme	Indicates which theme (style set) to use for the application. There are three options: • Light (the default, light color scheme) • Dark (an included dark color scheme) • Custom (enables custom CSS to be applied as a theme) See Section 4.13.1 for further details on custom themes.
Grid Snapping	Snap Extensions to each other	Indicates whether to snap the edges of Extensions to each other when moving them around the Desktop.
	Snap Extensions to grid	Indicates whether to use an invisible grid to align Extensions when moving them around the Desktop. When off, Extensions move smoothly and freely.
	Grid size	Sets the (invisible) grid size to use when "Snap Extensions to grid" is enabled. There are three sizes: • Small (10px × 10px) • Medium (20px × 20px) • Large (30px × 30px) Clicking the dropdown for this option automatically enables "Snap Extensions to grid."
Tabbed Extensions	Enable	Enables Tabbed Extensions, which allows multiple Workflow Extensions to be merged together into a single, tab-separated Extension element. When enabled, a Tabbed Extension can be formed by simply dragging and dropping one Extension onto another (a blue border will indicate that it can be dropped). If a Tabbed Extension has already been formed prior to this option being disabled, the Extensions within it will still support being Tabbed after this option is disabled.

Animations	Animate resizing	Indicates whether to animate the resizing of Extensions. When disabled, Extensions move immediately with the mouse when resized, without smooth animations. Disabling animations can help with performance on slower machines.
	Animation speed	Sets the speed of the animation while resizing. There are two speeds: • Slow • Fast

4.7.3 Preferences

Preferences contains further preferences which do not have their own category.

Service	Option	Description
Autosave	Enabled	Toggle switch indicating whether the application should automatically save. NOTE: Applications auto-save within the browser whenever changes are made to avoid losing your work. This option is for auto-saving to a file for permanent storage.
	Frequency (minutes)	Indicates how often the application should automatically save itself to the last save location. Minimum of 1 minute.
Reporting	Default document format	Indicates the default file format to be used when generating reports. File format can always be changed when generating reports, but can save time during generation if a specific file type is typically used. Multiple standard formats are available: • Microsoft Word document (.docx) • Webpage (.html) • Portable Document Format (.pdf)

4.8 Adding, Placing, Collapsing, and Removing Extensions

Once you have at least one data source loaded into your application, you can start adding Extensions to the interface.

4.8.1 Adding Extensions

Begin by opening the Workflow Drawer using the pull tab at the bottom-center of the screen or one of the other methods mentioned in section 3.5.1 [Opening and Closing](#). Once the Drawer is open, locate the Suite from which you would like to deploy

A Note on Extensions in this Guide

Often, the first element added to the interface is a Flaremap. Any Extension may be added at any time, but for the purposes of this Guide, we will begin with a Flaremap.

an Extension.

Locate the Extension you would like to add to the interface, then click and drag its card out onto the Desktop environment, releasing the mouse once you are near where you want the Extension. Upon mouse release, the Extension will immediately be deployed.

Repeat this process to add each Extension you would like to the interface, or continue on to the next section if you'd like to configure an Extension before adding more.

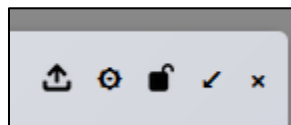
4.8.2 Placing Extensions

You can now drag the recently deployed Extension(s) around the interface by grabbing anywhere in its header. Using the edges of the Extension, you can also resize it. Move and resize your Extension to get the desired location and size. Extensions may always be moved and resized later, except in published apps. Oftentimes, after configuring your Extensions and adding more, you will need to readjust the placement and size of some of them.

If you've got your Extension placed and sized comfortably, you can lock it in place by clicking the  (lock) icon in the Extension's controls. This will prevent the Extension from being moved or resized. The icon will toggle to a closed lock to indicate that the Extension is locked.



To unlock an Extension and allow it to be moved and resized again, simply click the  (lock) icon again (this will return the icon to an open lock).



If you've added multiple Extensions, repeat this process with each one to design an application layout that meets your needs.

4.8.3 Collapsing Extensions

Every Extension may also be collapsed (minimized) for convenience. This may be helpful when designing an application with many Extensions, or when actively using an application in Designer Mode. It can help users to have many Extensions configured and available, but kept nicely out of the way when not immediately needed.



To collapse (minimize) an Extension, click the ✓ (arrow) in the Extension controls. The Extension will collapse to the minimum width with only its header visible. To expand (maximize) an Extension, click the ↗ (reversed arrow) in the header.

All collapsed Extensions are the same size, making it convenient to stack and organize them.

4.8.4 Removing Extensions

If you no longer need an Extension, or accidentally added the wrong Extension to your interface, simply click the ✕ icon in the Extension controls to remove it.

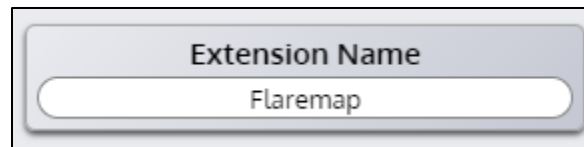
4.9 Configuring Flaremaps and Extensions

Almost every Workflow Extension has its own set of custom settings and controls which can be configured to tailor its performance to the user's needs.

To access a Workflow Extension's settings, click the ⚙️ (gear) icon in the Extension's controls. This will open the Editor panel with the Extension's controls.



One common setting for most Extensions is the Extension name, which appears at the very top of its settings. Enter any name for the Extension, then click anywhere off of the input box. This is the only Extension setting that automatically updates without pushing the "Apply" button at the bottom of the Editor panel.

A screenshot of a settings panel. At the top, there is a label 'Extension Name' in bold. Below it is a text input field containing the word 'Flaremap'. The input field has a light gray border and a subtle shadow.

Navigate through the settings within the Editor panel to make the changes you need to the selected Extension.

Please note that this User Guide is for the Platform, and does not describe the individual controls available for each Extension. Read the box below for more information on where to find documentation on individual Extensions and their respective settings.

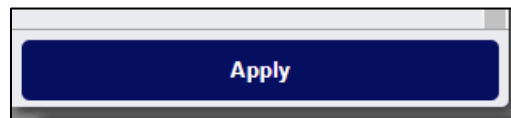
Workflow Extension Documentation

Because each Suite contains different Workflow Extensions, and because different users may have different Extensions, documentation regarding the specific controls and usage of each Extension are separate from this User Guide. Documentation for each Extension is available on our website, and is quickly accessible through the Extension itself. Click on the (document) icon in the Extension header to open a tab directly to the documentation for the Extension you are using.



Please refer to the associated documentation of an Extension to get details on the specific controls and usage.

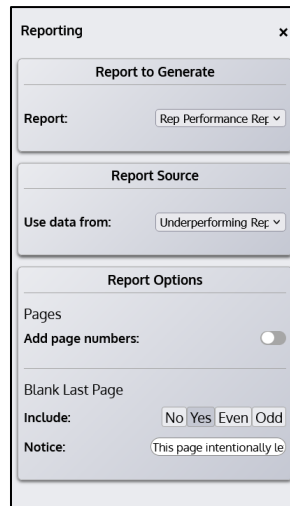
Click the “Apply” button, which is fixed to the bottom of the Editor panel while editing Extension settings, to apply the changes you’ve made. The Editor panel will remain open after clicking “Apply,” so you may iterate through multiple stages of edits to test out the Extension without having to reopen its settings each time.



Once you have modified your Extension as desired, close the Editor panel by clicking the ✕ in the upper right corner of the panel. It will also close automatically if you open another conflicting panel, such as the Workflow Drawer.

4.10 Reporting

The Visual Action Platform features a built-in Reporting Engine to assist users in generating distribution-ready reports dynamically from the analyses they performed during a session.



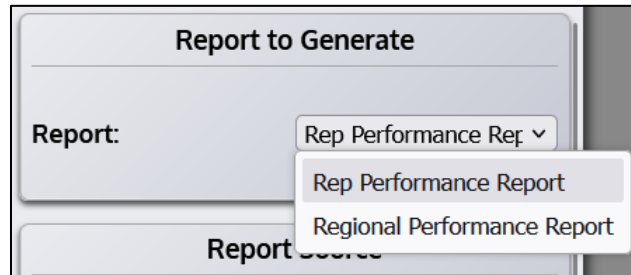
The screenshot shows a 'Reporting' dialog box with three main sections: 'Report to Generate', 'Report Source', and 'Report Options'. The 'Report to Generate' section has a 'Report:' dropdown menu with 'Rep Performance Rep' selected. The 'Report Source' section has a 'Use data from:' dropdown menu with 'Underperforming Rep' selected. The 'Report Options' section includes a 'Pages' section with an 'Add page numbers:' toggle switch, a 'Blank Last Page' section with a dropdown menu showing 'No | Yes | Even | Odd', and a 'Notice:' section with a text field containing 'This page intentionally le'.

To open the Reporting Manager, open the Menu and select "Reporting" or use the keyboard shortcut (see section 3.8.3 Keyboard Shortcuts (Keybinds)).

4.10.1 Report to Generate

The first option in the Reporting Manager falls under "Report to Generate," and is where the specific report to generate is chosen. Reports which are available to generate are driven by report logic that is included in the Logic Layer. To learn more about how to assemble and link report logic so that reports can be automatically generated, see section 4-A.4 Report Logic.

In the dropdown to the right of "Report," any reports are available to be generated will be listed. Select the appropriate report title from this list to choose which report to generate.



4.10.2 Report Source

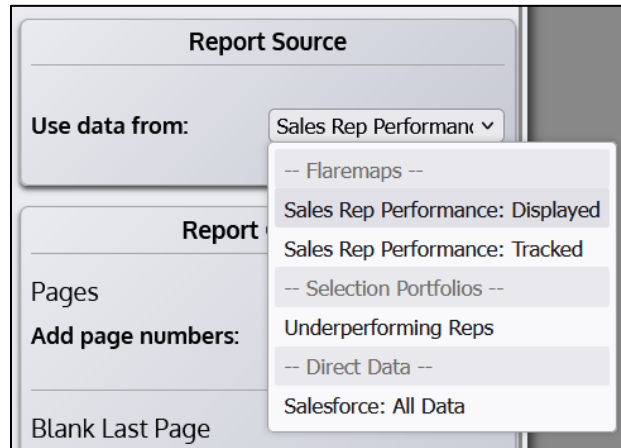
The Report Source section in the Reporting Manager identifies which collection of data within the interface should be included in the report. This is how a user can select which analyzed data points should be reported on.

There are three primary sources of report data:

- **Flaremaps:** build report from currently displayed cells or from tracked cells
- **Selection Portfolios:** build report from items (cells) added to the portfolio
- **Data Objects:** build report from all rows in a dataset

Users can select which information to report on, depending on how their analysis was tracked (filtering out cells, tracking cells, adding cells to a portfolio, etc.).

Each available source of report data will appear under its corresponding group (see above source types) in the dropdown labeled “Use data from”. Select the source you want to use from this dropdown.



Note: The category names will not change and are there to help users identify the type of each available source; however, the source names themselves will reflect names assigned by users. For example, if a Flaremap was named “Inventory Items”, selections pertaining to that Flaremap in the list will begin with “Inventory Items” (not “Flaremap”).

4.10.3 Report Options

The Reporting Manager provides controls for various features made available by the Reporting Engines for use in any report. See below for a description of each of the available options and what they do.

Pages

Add page numbers: Enables the Platform Reporting Engine to automatically add page numbers to the lower right corner of each report page. The first page (typically a title page) will be skipped.

Blank Last Page

Include: Indicates whether the Platform Reporting Engine should include a blank page as the last page of the report (also known as a “vacat page”). There are four options for including the blank last page:

- **No:** Do not include a blank last page.
- **Yes:** Always include a blank last page.
- **Even:** Only include a blank last page if the last page is an even-numbered page (including the blank last page itself).
- **Odd:** Only include a blank last page if the last page is an odd-numbered page (including the blank last page itself).

Notice: Specifies what text to display on a blank last page, if included (also known as a “vacat notice”). This text will appear in small print in the center of the blank page. The default notice reads: “This page intentionally left blank.”

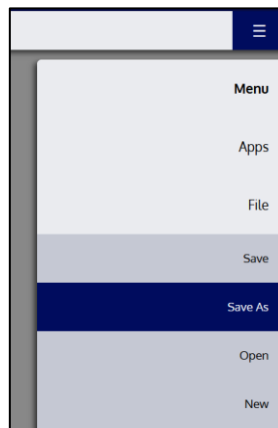
4.11 Saving an Application

Both during and after working on an application, you will want to be sure to save your work. As mentioned previously, regularly saved application files are editable applications. The next section (4.12) describes how to publish non-editable applications.

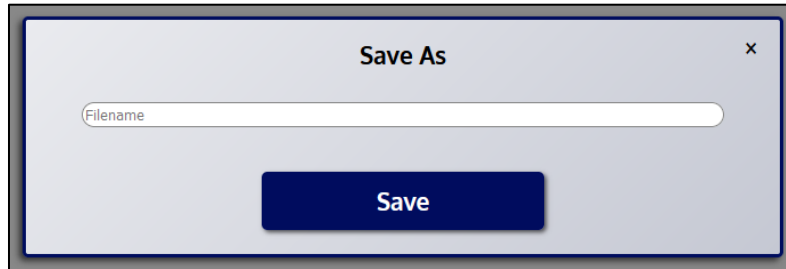
An Important Note on Saving Applications

This section is for saving your editable application to file as a backup, for editing later, for sharing, or for other reasons. Applications are automatically saved in the browser on a frequent basis while editing to ensure you do not lose your work; however, it is still very important that you save your own copy of your application somewhere safe. Should the browser storage get cleared for any reason (including resetting the Platform environment), the Platform will not be able to automatically recover your application, and you will need a saved copy of it.

There are two options in the Menu (under "File") for saving an application: "Save" and "Save As."



If it is your first time saving your application to file, click "Save As." A dialog box will appear and you will be prompted for a file name. Once you've entered the name you want to use, click the "Save" button. Depending on your browser settings, the file may immediately download to the automatic download location, or you may be prompted to choose where to save it.



If you would like to automatically save to the last file name you used, click "Save." If you have not yet used the "Save As" option to name your file, you will first be taken to the "Save As" dialog box to enter a name. Following this, the "Save" option will automatically use the last name you used.

Depending on your browser settings, you may still be prompted to choose a location to save the file, even when using the quick "Save" option. This is a browser security option and cannot be overridden by the Platform.

NOTE: The Desktop environment will be disabled while the "Save As" dialog box is open.

To cancel saving (using "Save As"), click the **x** in the upper right corner of the dialog box.

4.12 Publishing an Application

After you've completed designing an application, you may want to make it available for use without all of the controls of the Platform (non-editable). This can be accomplished by publishing an application.

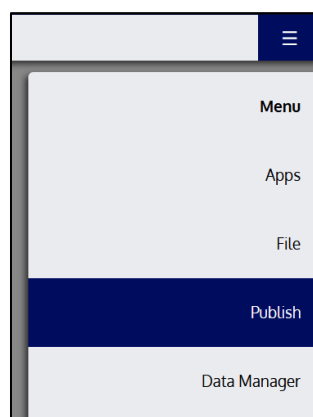
A published application contains all the Extensions and configurations of an application, while disabling the ability of the application to be modified in any way. This is called "Publish Mode," as opposed to "Designer Mode," which is the normal mode of the Platform in which all the controls and configurations for editing and application are available.

A Note About Published Applications

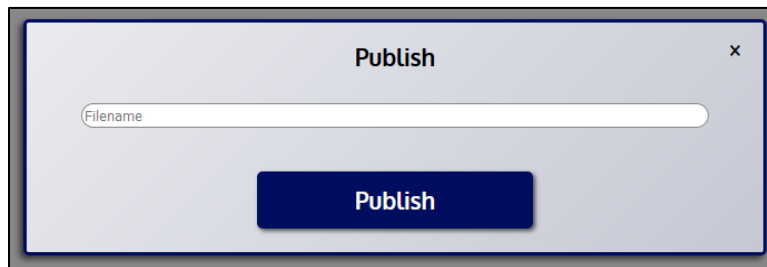
Published applications are a restricted form of application which allows users to interact with and use the application without being able to edit it. Applications are usable in both edit and publish modes, but published applications may be helpful for presentations or to prevent users from editing an application when not permitted or appropriate.

Published applications are still run within the Platform environment, and thus must still be accessed through a valid, licensed copy of the Platform.

To publish an application, open the Menu and click "Publish."



This will prompt you to save the file in the same way the Platform would when clicking “Save As.” When publishing, however, you will notice that the file will save as a `.vap` file, rather than a `.vas` file. This indicates that this file is a published application that cannot be edited.



Enter a file name for the published app file, then click “Publish.” Depending on your browser settings, the file may immediately download to the automatic download location, or you may be prompted to choose where to save it. This is a browser security option and cannot be overridden by the Platform.

NOTE: The Desktop environment will be disabled while the “Publish” dialog box is open.

To cancel publishing, click the **x** in the upper right corner of the dialog box.

Open a published application the same way you would open any other application within the Platform. Open the Menu, click “File,” then “Open,” and choose your published application file.

Once you choose to open a published application, the Platform will automatically detect that it is published and will put itself into Publish Mode. This will remove many options from the Menu that allow the application to be edited, as well as remove the Workflow Drawer and Editor panels. It will also restrict the ability to modify Data Source connections and the layout of the interface.

Once published, the application cannot be edited. To make changes to an application, use an editable (`.vas`) copy of the application that you saved while working on it. To republish an application, simply edit the editable copy of the application, then repeat the above process to republish the new version.

4.13 Advanced References

There are several advanced features of the Visual Action Platform that are not critical components of basic application-building, but which can significantly enhance the look and performance of your applications.

4.13.1 Custom Themes

The Visual Action Platform enables you to modify the styling of the page to create a custom theme for your application. While there are two themes included (a light and a dark theme), custom changes are often desired to personalize and improve applications.

There are two ways to apply a custom theme to your application. First, open the “Settings and Preferences” panel from the menu or by using the keyboard shortcut `Ctrl + Alt + S`. Under the “Look and Feel” section, find “Theme,” and select “Custom.” You will see two input options appear.



The screenshot shows a settings panel titled "Theme". It contains the following elements:

- A "Theme:" label followed by three buttons: "Light", "Dark", and "Custom". The "Custom" button is highlighted.
- An "Upload CSS file:" label followed by a "Browse..." button and a "No fil." button.
- An "Or paste CSS here:" label followed by a large text area. Inside the text area, it says "Paste custom styles here." and "If this box is not empty, its contents will be applied rather than those of an uploaded CSS file."
- An "Apply Theme" button at the bottom.

Upload CSS file

The first option for applying your custom theme is to upload a CSS file. Many developers or designers find it easier to write and edit their CSS in an external editor. This option allows those users to write and edit their CSS styles in a different editor, saving it as a CSS file. Then, the CSS file can simply be uploaded into the application, and the styles from the file will be applied.

Simply click "Browse" to locate your CSS file. Select the file you have your custom theme in, then click "Open". Click the "Apply Theme" button to apply the styles from your file.

Note that if any custom CSS is pasted in the input area (described in the next section) below the file upload, those styles will be applied rather than those from the uploaded file.

Paste CSS

As another option allowing easy pasting and live editing of styles within the Platform, you may also paste in the custom theme styles you want to apply in the box beneath "Or paste CSS here:". Simply paste your CSS styles in and click the "Apply Theme" button to apply the styles. This input box takes precedence over an uploaded file. You can edit the contents of this box and apply changes to test how a styling change will look in your application.

The next section includes the technical reference, guiding you as to which CSS variables, classes, and IDs should be modified to achieve specific results with your custom styling.

4.13.2 Custom Styling Technical Reference

To enable custom themes, the Visual Action Platform makes use of CSS variables whose names are exposed to the user. You can modify the values of these variables to apply custom styling to your application.

Below is a table referencing each accessible variable and the application element it affects. For reading clarity, the preceding dashes in CSS variable names are excluded from this table, but keep in mind that the naming format for CSS variables requires that all variables begin with two dashes, for example: `--my-color-variable`.

Referencing other CSS variables (of the appropriate type) within a variable is permitted and should work properly within the Platform.

Almost all variables applying to Platform styles should be included within the `:root` selector. Place all of your styling inside the `:root` selector, like this:

```
:root {

    /* All custom styling goes here */

}
```

The appropriate selector will be indicated for any styles that **do not** belong within the `:root` selector.

Custom Styling Reference Table

Variable	Applies To	Value Type
Theme Styles These style attributes apply to the overall theme, and are applied by default to multiple elements. Used these variables to apply a theme to the entire application.		
background	The default background color used in the theme.	Color
foreground	The default foreground color used in the theme.	Color
foreground2	The alternative foreground color to use for the theme when a different foreground from <code>--foreground</code> is needed.	Color
foreground-gradient	A gradient to use for foreground items.	Color (linear-gradient)

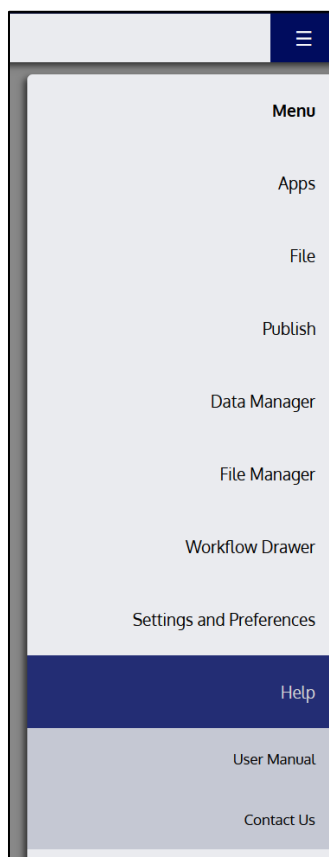
foreground-gradient-invert	An inverted version of <code>--foreground-gradient</code> .	Color (linear-gradient)
shadow	The default color for the box shadow of elements.	Color
color-primary	The primary theme color. Used automatically to theme elements without custom-styling each component.	Color
color-secondary	The secondary theme color. Used automatically to theme elements without custom-styling each component.	Color
color-tertiary	The tertiary theme color. Used automatically to theme elements without custom-styling each component.	Color
color-primary-pressed	The “pressed” version of <code>--color-primary</code> , used when a button of the primary theme color is clicked/pressed.	Color
font-color-primary	The primary font color of the theme. Used for most text by default.	Color
font-color-secondary	The secondary font color of the theme. Used where an alternative to <code>--font-color-primary</code> is needed.	Color
filter-bw	A filter to apply to black-and-white icons used throughout the application. Can be switched to appropriately invert icons when a light/dark theme is used, respectively.	CSS3 invert filter
Individual Element Styles These styles apply to specific, individual elements of the application. Use these variables to customize specific pieces of the application, rather than affect all elements using the above theme styles.		
desktop-background-color	The background color of the Desktop (interface where Flaremap and Extensions are deployed).	Color
workflow-drawer-color	The color of text in the Workflow Drawer.	Color
workflow-drawer-background-color	The background color of the Workflow Drawer.	Color
workflow-drawer-tab-color	The color of the arrows in the Workflow Drawer pull tab.	Color

workflow-drawer-tab-background-color	The background color of the Workflow Drawer pull tab.	Color
editor-panel-font-color	The color of the text in the Editor Panel.	Color
editor-panel-background-color	The background color of the Editor Panel.	Color
extension-font-color	The color of text in a Workflow Extension. This typically only affects the Extension header, and rarely effects content in the body of Extensions, which usually has its own style rules.	Color
extension-background	The background style of Workflow Extensions.	Color

4.14 Getting Help

If you need any help while using the Visual Action Platform, there are many options available to you.

Open the Menu and click “Help” to view the in-Platform help options. The first option, “User Guide,” will open the most up-to-date version of this User Guide from the Visual Action website. If you are viewing this document digitally, you can also check for updated documentation by clicking [here](#).



If you need further assistance beyond what is offered by this User Guide or the documentation accompanying a Workflow Suite or Extensions, click “Contact Us,” or navigate directly to www.visualaction.com/contact, and use one of the available methods to get in touch with us for help. You can also use the information contained in section 7 Contact to contact us.

4-A Advanced Platform Configurations

Beyond the user-friendly controls provided by the Platform interface, there are several advanced capabilities which the Platform supports. Such capabilities often require more technical knowledge and configuration. Use this section as a reference for these features. Please see the Technical Reference or contact Visual Action for more details about advanced configurations.

4-A.1 Custom Logic

An extremely powerful and useful component of many applications is custom logic written to support extended functionality. Logic is often used to provide additional manipulation and analysis of data or to add custom components to an interface.

While custom logic is very powerful, it needs to be written (programmed). Logic can be coded manually, or developed via services by Visual Action. Below is some general information about custom logic in the Platform:

- Logic functions and files are written in JavaScript (JS). With the appropriate technical expertise and programmatic connections, other languages may be used, but ultimately, the functions called within Platform applications must be written in proper JavaScript.
- There is a single, globally accessible variable where all Logic Layer functions are stored. The variable is simply named `Logic`.
 - `Logic` is a JavaScript object, with several properties that are also JavaScript objects. Each of these objects is a category for a specific type of logic functions.
 - While `Logic` is globally accessible (and thus, can be overwritten), it is best not to overwrite it. The Platform expects to access `Logic` and its default properties, and modifying them may result in performance errors. Similarly, it is advised that the default properties of `Logic` not be overwritten, but only added to.
 - New properties may be added to the `Logic` variable. The Platform, however, provides no user interface for accessing custom properties of `Logic`. Only functions defined within the default properties of `Logic` are accessible via the user interface.
- Logic functions have different parameter and return value requirements, depending on which type of logic function they are.

The following sections detail the technical parameters and processes for developing and including common types of custom logic in the Logic Layer of the Visual Action Platform.

4-A.2 Data Logic

One of the most frequently used types of custom logic is that for processing and manipulating data. The Platform allows users to easily select from an available set of custom logic functions to use when connecting to data.

Data Logic functions, when selected, are called after the Platform fetches the raw data from a source and converts it to a standard, object-based (JSON) format. The converted data is passed to the selected logic function, and the Platform expects the modified dataset, still in JSON format, to be returned from the logic function.

Tip: Make a copy of the data passed into the logic function by the Platform, modify that object in your logic, and return it at the end of the function. Alternatively, you can modify the original object in-place to ensure that you return the right object.

Below are the requirements for all custom Data Logic functions.

Data Logic Function Requirements

Definition Location: All Data Logic functions must be defined in the `Logic.Data` object.

Function Identifier: Each Data Logic function identifier must be unique among all property identifiers in the `Logic.Data` object.

Parameters:

- (1) **Required** The first parameter **must** receive the converted dataset that is fetched and passed to the function by the Platform.
 - (2...n) **Optional** Any parameters beyond the first parameter are optional and custom to the defined function.
-

Return: The function **must** return the processed data in JSON format. This must be the only return value (e.g.: an array containing the data in one of its indices is not valid). Even if the function does not modify data, it must return the dataset, as the Platform expects to receive data back from all Data Logic functions.

Example Data Logic Function Definition

```
Logic.Data['exampleFunction'] = (data,optionalParam='default') => {  
  for(let row in data) {  
    data[row]["sum_of_samples"] = data[row]["sample_a"] + data[row]["sample_b"];  
  }  
  
  return data;  
};
```

4-A.3 App Logic

App Logic functions are useful for applying custom functionality to an application as a whole, as opposed to Data Objects, Reports, or other components specifically. App logic is often used to add some additional component to the global interface of an application, such as a Navbar module/widget. App logic can be used to define any custom logic that should be applied to the entire application.

App Logic functions are called once an application is successfully loaded.

Below are the requirements for all custom App Logic functions.

App Logic Function Requirements

Definition Location: All App Logic functions must be defined in the `Logic.App` object.

Function Identifier: Each App Logic function identifier must be unique among all property identifiers in the `Logic.App` object.

Parameters: There are **no required parameters** for App Logic functions, and the Platform will not pass any arguments any App Logic function. Therefore, you can choose which parameters are defined – if any – and in which order, for your App Logic functions.

Return: There is **no required return value** for App Logic functions. You may return a value from your App Logic functions to be handled by other functions, but the Platform will not do anything with values returned from an App Logic function and will simply discard the returned data.

Example App Logic Function Definition

```
Logic.App['onLoad'] = () => {  
  // Call other custom functions for assessing if user is a first-time user and for  
  // creating a modal window  
  let firstTime = isFirstTimeUser();  
  if(firstTime) generateWelcomeModal();  
};
```

4-A.4 Report Logic

Report Logic functions are used for defining custom reports to generate from analyses performed in Platform Applications. Report Logic must include all the necessary information to generate and present the desired report when a user selects it from the Reporting Engine.

Report Logic functions are only called when the referenced report they supply is chosen for generation from the Report Engine control panel.

Report Logic functions are responsible for collecting, manipulating, and inserting the report data provided by the Platform and for generating and outputting the report content using that data.

Report Logic functions **are not** responsible for additional report features that are provided by the Platform (which have controls in the Reporting Engine control panel), such as adding page numbers and adding blank pages to the end of a report. If you wish, you may include these features manually in your report functions, but the Platform can handle these tasks automatically.

Report Logic Function Requirements

Definition Location: All Report Logic functions must be defined in the `Logic.Report` object.

Function Identifier: Each Report Logic function identifier must be unique among all property identifiers in the `Logic.Report` object.

Parameters:

- (1) **Required** The first parameter **must** receive the report settings provided by the Platform. Report settings are stored as a JavaScript object.
 - (2) **Required** The second parameters **must** receive the report data provided by the Platform. Report data is provided as an array of JSON objects, as it is stored for use by the application itself.
 - (3...n) **Optional** Any parameters beyond the first parameter are optional and custom to the defined function.
-

Return: The function must return the entire HTML content for the page displaying the report.

Report Logic HTML Content and Printing Requirements

The following requirements should be used to guide you in developing your custom Report Logic functions. The Platform Reporting Engine provides standard style rules and utilities for presenting and printing reports. These requirements ensure that your custom report will display correctly when generated via the Platform's Reporting Engine and that users can properly print reports on standard A4 Letter paper.

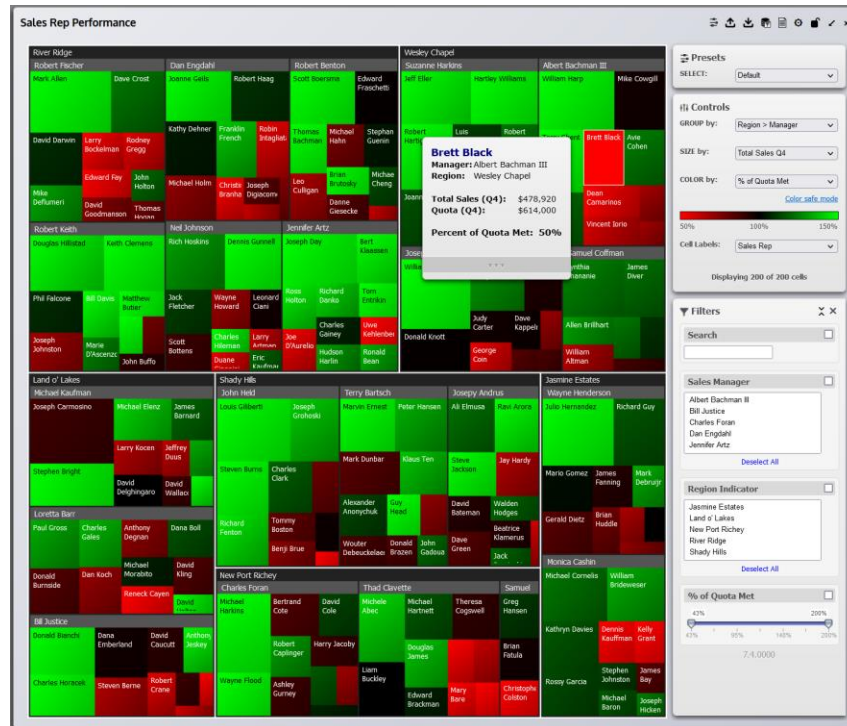
- Report pages must be assembled in HTML `div` elements with the class "page".
- For viewing reports on-screen, an HTML `div` element with the class "page-break" should be included before each generated page (`div` element of class "page"). The `div` elements of class "page-break" should contain no internal content between their opening and closing `div` tags.
- When printing, users should deselect their browser's option to print page headers and footers.
- When printing, users should select to print pages with no margins.

5 Flaremap

Flaremaps, powered by Flaremap JS 7.4, are the core of the operational analytics capabilities of the Visual Action Platform.

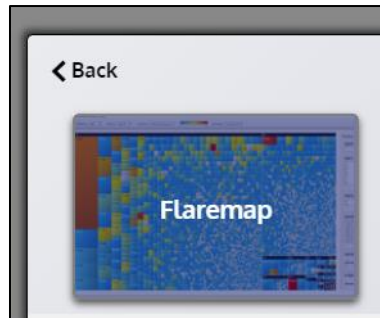
5.1 About Flaremaps

Flaremaps are holistic, data-dense visualizations that are designed to drive anomalies to the surface. Flaremaps use “cells” to represent data, with the cells grouped, sized, and colored based on selected data attributes. Flaremaps allow a dataset to be pivoted on many dimensions, leading to a multitude of perspectives and insights.



5.2 Deploying a Flaremap

Flaremaps, while a core component of the Platform, are nicely contained within the Workflow Drawer alongside Workflow Extensions. This makes them easy to access and deploy in the same way you would a Workflow Extension.



Before adding a Flaremap, be sure you have added a Data Object. If you have not yet added a Data Object, see section 4.6.2 [Creating a Data Object](#) to learn how to do so.

To deploy a Flaremap, open the Workflow Drawer, choose the "Flaremap" suite, and click and drag a "Flaremap" out onto the Desktop.

A Flaremap element within the Platform shares many of the same dynamic features and controls as Workflow Extensions, including the ability to be moved and resized, as well as the ability to be locked and collapsed.

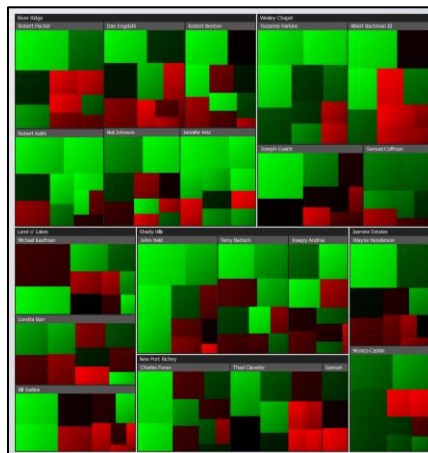
Move and size the Flaremap as you would with any Workflow Extension. Once ready, continue on to learn about the components of a Flaremap and how to configure them within the Platform.

5.3 Flaremap Components

There are three main components to the Flaremap interface: the Flaremap field, controls, and filters, as well as some additional utilities.

5.3.1 Flaremap Field

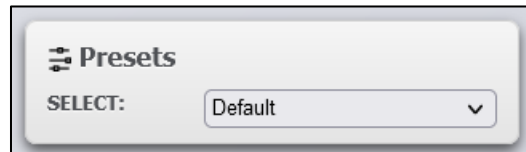
The Flaremap field covers most of a Flaremap's interface, visualizing data using the methodology of a treemap.



Flaremaps use cells to represent data from the connected data source, with each cell representing a record (row) from the data. The Flaremap field is where all of these cells are displayed.

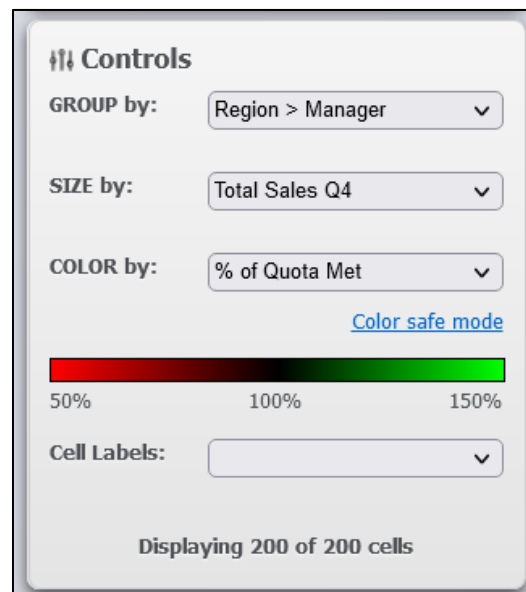
5.3.2 Presets

The Presets panel appears to the upper-right of the Flaremap field, just above the Controls panel. This panel presents Flaremap Presets – preconfigured combinations of group, size, color, label, and filter selections – that have been created, and allows allow you to select and apply them.



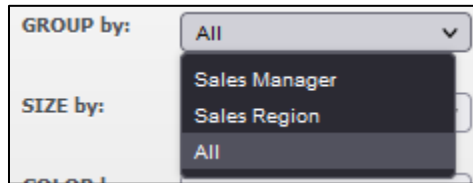
5.3.3 Controls

The Controls panel appears to the upper-right of the Flaremap field. This panel is where you can switch between options for grouping, sizing, and coloring cells to quickly drive the information you need to the surface. This panel also contains controls for toggling cell labels.

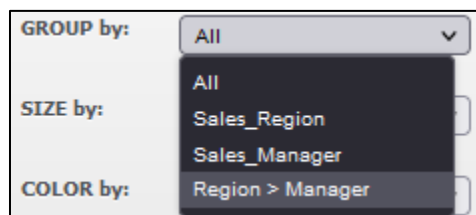


GROUP by

The "GROUP by" dropdown allows the user to select which option for grouping the cells is used. Groups are driven by (text) values that are shared across multiple rows of data.

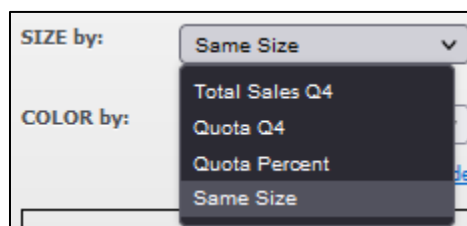


Grouping in Flaremaps also includes hierarchies, or "groups within groups." Groups may be created based on multiple data columns, and cells will be sub-grouped based on each selection.



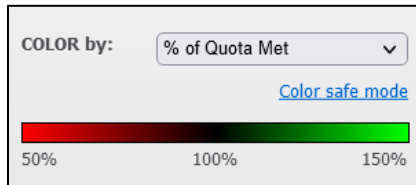
SIZE by

The "SIZE by" dropdown allows the user to select which option is used to size the cells. Sizes are driven by numerical values in the data.



COLOR by

The “COLOR by” dropdown allows the user to select which option is used to color the cells. Colors may be driven by both numerical data (spectrum) or text data (legend).

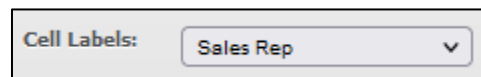


Each color by option is accompanied by either a spectrum or legend, respective to the data type of the column being used to color by, beneath the “COLOR by” dropdown. This is where the user can see what each color represents.

Beneath the “COLOR by” dropdown is a toggle labeled “Color safe mode.” Color safe mode is a feature that automatically alters colors to assist colorblind users in seeing the Flaremap properly. Click this button to toggle color safe mode. To exit this mode, click the button again, which will now say “Standard mode.”

Cell Labels

The “Cell Labels” dropdown allows the user to select what data, if any, is displayed directly on the cells themselves.



Cell Display Count

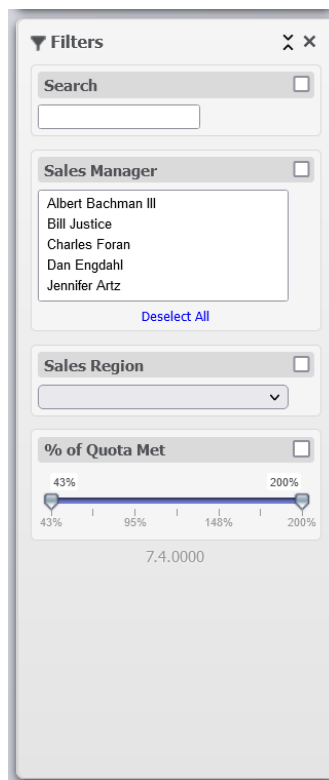
At the bottom of the Controls panel, there is a count of the cells displayed on the Flaremap. This automatically updates when filtering to indicate how many cells are visible out of the total number of cells in the data at any given time. It will initially show that all cells are visible.

Displaying 200 of 200 cells

Displaying 10 of 200 cells

5.3.4 Filters

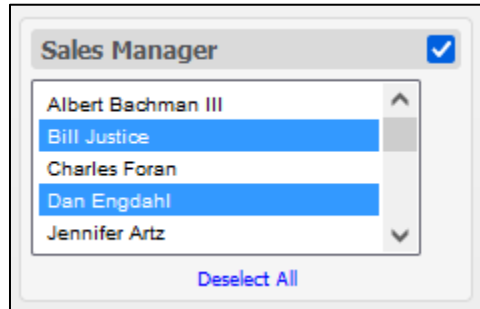
The Filters panel appears to the lower-right of the Flaremap field. Filters do just that – filter (hide or display) cells based on various rules or combinations of rules. Filtering is a powerful and fast way to narrow down data into an applicable subset.



There are three main types of filters (and one special type):

Filters

Basic filters, known as just “filters” or “select filters,” display only the cells that match a certain value within the associated data column.

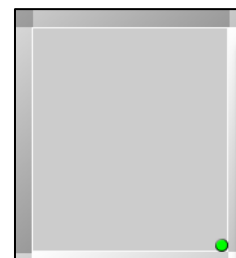
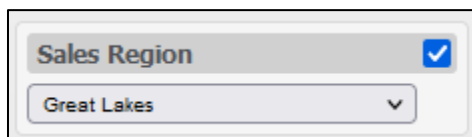


Filters can be individual, meaning only one value to filter by may be selected from a dropdown, or multiple, meaning that multiple values can be selected to filter to all cells which match *any* of the selections. These types can be distinguished on the interface by looking for a dropdown versus a list containing at least two visible items.

To select multiple items within a multiple-select filter, hold Ctrl while clicking each item. Use the “Deselect All” button at the bottom of the filter to deselect every option in that filter.

Indicators

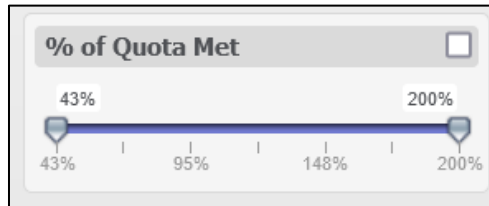
Indicators provide a way to “filter” items with a visual cue. Similar to basic filters, there are individual and multiple modes of indicators. The difference is that the cells matching the criteria of the selected items in an indicator will have a visible beacon placed on top of them. This is opposed to the previous type of filter, which actually *hides* the cells that don’t match any of the selected criteria.



Indicators come in various styles, which can be set in the Flaremap configuration. See section 5.4.8 Creating Indicators to learn how to adjust these styles.

Sliders

Sliders provide a way of accurately filtering numerical data. Unlike the other two filter types, sliders provide – you guessed it – a range slider that can be adjusted to a specific range.



When a slider's range is adjusted, the cells will be filtered to display only the cells whose data (for that slider's selected column) falls within the selected range.

Search

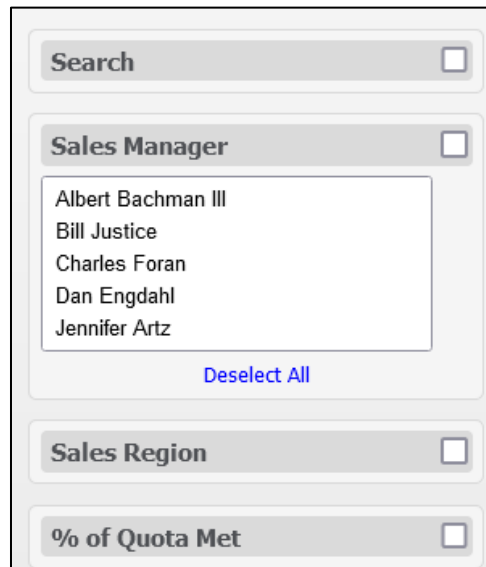
There is one special type of filter that is always present in the Filters panel, whether you create more filters or not. This filter is a "Search" filter.

When you configure your Flaremap, the Search filter will automatically be configured to search any text columns in the data. This filter is always present and may not be removed. You cannot create additional Search filters, as the default one will cover all searchable columns.

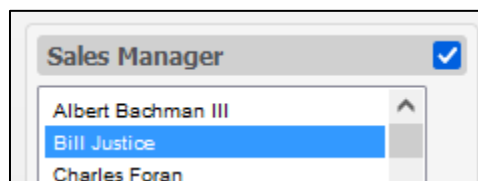
Use the Search filter to quickly find a specific cell or set of cells with an identifiable piece of information by typing into the input box within the filter. The Flaremap will automatically filter cells as you type. Once the software can no longer find any cells with matching data, the Flaremap will indicate that "No results match your criteria."

Filters are each represented within their own box inside the Filters panel. When you add a filter in the Flaremap configuration and apply changes, you will see them automatically populate themselves in this panel (the default Flaremap will begin with an almost empty Filters panel).

Each individual filter may be collapsed by clicking on its header (title bar). Expand the filter again when needed by clicking again on the filter's header.



When you click inside a filter to use it, the filter will automatically be enabled. This can be seen by the checkbox in the upper-right corner of the filter automatically becoming checked.



You can disable a filter without deselecting your options or adjusting your parameters by unchecking this box. This saves users from having to reselect items or reset a slider range when toggling between using a filter and not using it. It enables users to quickly switch between which filters are currently applied.

In the upper right of the Filters panel, you will notice two buttons: a "Collapse All" button (✕) and a "Deselect All" button (x).



Collapse/Expand All

The "Collapse All" button (✕) will collapse all filters within the Filters panel. This is helpful in Flaremaps with many filters, as it keeps filters in a smaller form, allowing the user to expand only the relevant filters at any given time.

Likewise, after collapsing all filters, this button will change to an "Expand All" (⬆) button. Clicking this button will do the opposite, opening up all filters within the Filters panel.

Deselect All

The "Deselect All" button (✕) will uncheck (disable) all filters which are currently active. This allows users to quickly turn off all filters and restore the full dataset to the Flaremap field.

NOTE: This is different from the "Deselect All" seen at the bottom of multiple-select filters and indicators, which only deselects all selected options within that specific filter.

5.3.5 Mouseover Boxes

Flaremap mouseover boxes (or simply "mouseovers") drill-down into the data composing a cell or group to provide detailed information and aggregate statistics. Mouseovers appear when hovering over a cell or a group on the Flaremap field. There are two types of mouseover boxes: cell and group mouseovers.

Cell Mouseovers

Cell mouseovers provide specific attributes about the data record a cell represents.

Terry Ghent
Manager: Albert Bachman III
Region: Wesley Chapel

Total Sales (Q4): \$588,240
Quota (Q4): \$342,000

Percent of Quota Met: 172%

▲ ▲ ▲

Zoom in
 Track this cell

Group Mouseovers

Group mouseovers provide aggregate details and statistics about a group of cells and all of the cells that comprise that group.

River Ridge
 Summary Statistics

60 sales reps

	Average	Total
Total Sales	\$371,422	\$22,285,310
Quota	\$351,250	\$21,075,000

▲ ▲ ▲

Zoom in
 Track visible cells in this group
 Untrack visible cells in this group

5.3.6 Header Controls

As mentioned previously, the Flaremap shares many controls and capabilities with Workflow Extensions. This includes the controls in its element's header.

For Flaremaps, however, three new controls are introduced: "Import Flaremap Configuration," "Export Flaremap Configuration," and "Export Filtered Data."

Import Flaremap Configuration

The  (upload) icon in the Flaremap element's header provides an option for users to upload a Flaremap configuration file. This allows users coming from older versions of the software (Flaremap JS 6.9 and below) to upload and import their existing Flaremap configurations, such that they may continue with the same Flaremap they were already using without having to recreate it. Users will be able to immediately use the Flaremap as-is, or use the new and improved controls of the Platform to edit their existing configuration further.

NOTE: When importing a Flaremap configuration file from another application or from the Flaremap Development Kit, it is important to ensure that the file is formatted correctly. The Platform will attempt to cleanse the file of some contents that are not allowed, but you should check the file and perform the following before importing:

- Remove JavaScript comments from the file, both single-line `"/ /"` and multi-line `("/* * /")`
- Remove the jQuery onload call (`"$(function() {"`) and all of its contents
- Remove the variable definition outside the JSON (i.e.: `"var viewConfig = "`)

Export Flaremap Configuration

The  (download) icon in the Flaremap element's header allows the user to export and save the Flaremap's current configuration file in developer format. This allows users to use the Platform to build and design Flaremaps and subsequently use that Flaremap in existing applications built in older versions of Flaremap JS or in embedded Flaremaps being developed using the Flaremap Development Kit.

NOTE: The Flaremap configuration file output from the Platform is suitable for use in embedded Flaremaps and other applications, but will not contain custom items such as developer comments or custom functions that run on page load.

Export Filtered Data

Most data management within the Platform is done via the Data Sources interface. Some users, however, may want to export the subset of filtered data they have arrived at using the Flaremap for other purposes.

Clicking the  (export filtered data) button will export only the filtered (visible) cells' data to a CSV file.

5.4 Flaremap Configuration

In the Visual Action Platform, Flaremaps are now creatable right through the interface!

Begin the configuration of your Flaremap by clicking the ⚙️ (gear) icon in the upper right corner of the header.



Please note that some sections within this section on configuration include quick-reference tables at the end for users to quickly address what specific controls do. Read through the text if you need to learn how to perform an action within the Platform, or use the tables if you just need to quickly recall the function of a specific control.

5.4.1 Naming a Flaremap

A Flaremap can be named in the same way an Extension would be. Once you have opened the Flaremap configuration in the Editor panel, the first option will allow you to enter a name for the Flaremap.

A screenshot of a configuration window. At the top, there is a label 'Extension Name'. Below it is a text input field with the word 'Flaremap' entered inside it.

Enter any desired name, then click out of the input box. The name will automatically update.

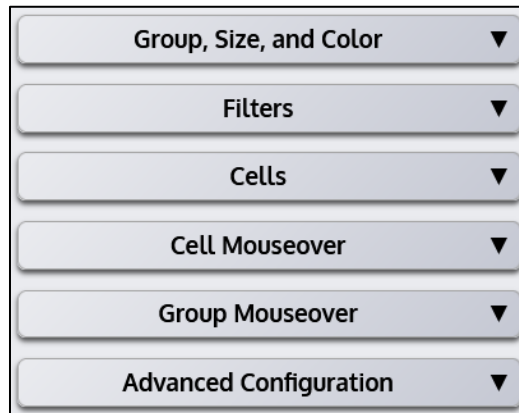
5.4.2 Creator Groups

There are many options to be created and configured when it comes to Flaremaps. Because of this, these options are categorized into smaller groups within the Editor panel.

The screenshot shows the 'Flaremap Editor' panel with the following sections and options:

- Extension Name:** A text input field containing 'Flaremap'.
- Dataset:** A dropdown menu currently showing 'Salesforce'.
- Quick Configuration:** Two buttons: 'Auto-configure' and 'Clear all options'.
- Group, Size, and Color:** A section header with an expand/collapse arrow. Below it are three options, each with a plus icon:
 - Group Options
 - Size Options
 - Color Options
- Filters:** A section header with an expand/collapse arrow. Below it are three options, each with a plus icon:
 - Filters
 - Indicators
 - Sliders
- Cells:** A section header with an expand/collapse arrow. Below it are two options:
 - Track icon: A dropdown menu showing a checkmark icon.
 - Cell Labels: An option with a plus icon.
- Cell Mouseover:** A section header with an expand/collapse arrow. Below it is one option:
 - Launch Mouseover Creator
- Group Mouseover:** A section header with an expand/collapse arrow.
- Apply:** A large blue button at the bottom of the panel.

Each settings group may be collapsed and expanded. As the Editor panel fills up, it may be helpful to collapse a group to limit the need to scroll. Click the upper banner of a settings group to collapse it. The arrow at the right of the banner will reverse. Click the banner again to expand the group.



Group, Size, and Color

Within this settings group, Flaremap group (and hierarchy), size, and color options may be created.

Filters

Within this settings group, each type of filter may be created.

Cells

This settings group is where cell-specific configurations are made. Cell Labels may be set up here.

Cell Mouseover

From this settings group, the Mouseover Creator for the respective Flaremap may be launched, using which the user can design a cell mouseover for the Flaremap.

Group Mouseover

From this settings group, the Group Mouseover Creator for the respective Flaremap may be launched, using which the user can design a group mouseover for the Flaremap.

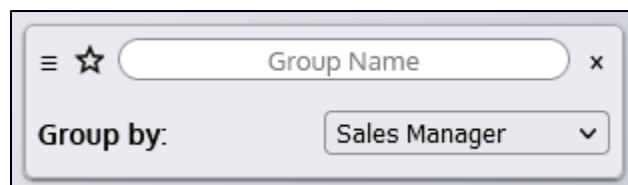
Advanced Configuration

This group begins collapsed by default. This group contains advanced settings for configuring the Flaremap. This section should only be used by those experienced with

creating Flaremaps in previous versions of Flaremap JS, or by advanced users who have thoroughly familiarized themselves with the **Flaremap Development Kit Technical Reference**.

5.4.3 Creator Cards

Each component you create within the Flaremap will appear in its own “card” within its settings group. These cards share some common characteristics.



Sort Handle

The ≡ (triple-bar or “hamburger”) icon in the upper left of a card indicates that the card can be dragged to sort. You may drag cards around to sort them and place them in the order you wish them to appear in the Flaremap options.

Default Toggle

A ☆ (star) icon in the upper left of a card represents a toggle for making that option the default. There may only be one default for each type of option. For example, there is one default for groups, one for sizes, one for colors, etc. Selecting a default is *not* required. When you do select a default option, the star will fill (★).

Option Name

In the top center of every card is an input box for a corresponding option name. This is the visible name the option will be given when showing up in lists within the Flaremap interface.

All options require names. If an option name is left blank, it will automatically be given a name relative to the column of data selected for that option.

Delete Button

In the upper right of a card is an × button. Clicking this button will delete the option.

Quick-Reference Table

Option	Description	Display
Sort Handle	Allows an option to be sorted for a particular position in the option list.	≡
Default Toggle	Toggles the default option for an option type. Only one per option type may be selected.	☆ Not set as Default ★ Set as Default
Option Name	Specifies the name of an option, which appears within the Flaremap interface.	Group Name
Delete Button	Removes a created option.	×

5.4.4 Creating Group Options

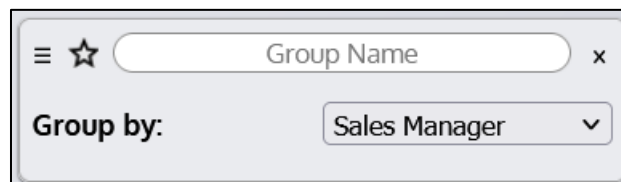
The first option you will see to create for your Flaremap is a group option.

Across from “Group Options,” to the right, you will see two + buttons. By default, these buttons look like this: , but this can change depending on the theme/style selected for the application.

The left button creates a standard, single-level group option. The right button creates a hierarchy, or multi-level, group (within group) option.

Create group

Begin by clicking the left + button on the “Group Options” header. You will see a new group option card appear.



The image shows a group option card with a header bar containing a sort handle (≡), a default toggle (☆), a text input field labeled "Group Name", and a delete button (×). Below the header, there is a "Group by:" label followed by a dropdown menu currently showing "Sales Manager" with a downward arrow.

You can name the group option now, or wait until you select the column of data to group by to enter an appropriate name.

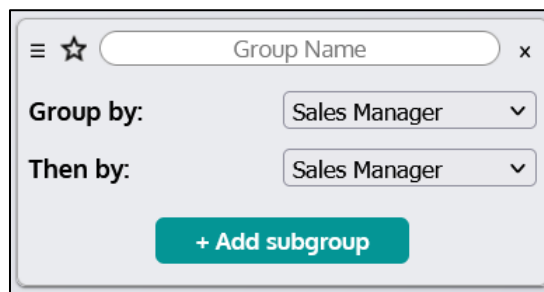
In the “Group by” dropdown, select the column of data you would like this option to group by. Recall that group options come from columns of data type **TEXT|ENUM**.

If you haven’t already, give the option an appropriate name.

Repeat this process for as many standard group options as you would like.

Create hierarchy

Begin by clicking the right + button on the “Group Options” header. You will see a new hierarchy option card appear, but this option will contain both “Group by” and “Then by.”

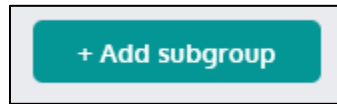


You can name the hierarchy option now, or wait until you select the columns of data to group by to enter an appropriate name.

In the “Group by” dropdown, select the column of data you would like this option to initially group by. This option will be the outermost group of the hierarchy. Recall that group options come from columns of data type **TEXT|ENUM**.

In the “Then by” dropdown, select the next column of data you would like this option to group by. This will be the first subgroup under the initial group you previously chose. Hierarchies must have at least one subgroup. To group by just one column of data, delete this option using the × in the upper right corner, then go to the previous section and create a regular group option.

To add more subgroups, click the “+ Add subgroup” button. This will add another “Then by” dropdown. Follow the procedure in the previous paragraph to create a further subgroup.



NOTE: A hierarchy requires there to be a standard group option for each column used within the hierarchy. If you haven’t created a group option for one of the columns used in your hierarchy option, the Platform will automatically create one for you (and therefore, you will see it as a standard group option in your Flaremap and in the Flaremap’s settings next time you open the Editor panel).

NOTE: Too many subgroups can quickly get messy within the Flaremap. You may want to test out how your hierarchy looks as you add layers by applying your changes one step at a time.

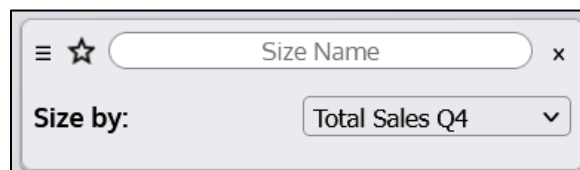
If you haven’t already, give the option an appropriate name.

Repeat this process for as many hierarchy group options as you would like.

NOTE: An “All” group option, which contains all cells, is automatically created and is always available. The “All” group option cannot be modified or removed.

5.4.5 Creating Size Options

The next option you will see to create for your Flaremap is a size option.



Begin by clicking the + button to the right of the “Size Options” header. You will see a new size option card appear.

You can name the size option now, or wait until you select the column of data to size by to enter an appropriate name.

In the “Size by” dropdown, select the column of data you would like this option to size by. Recall that size options come from columns of data type **NUMBER**.

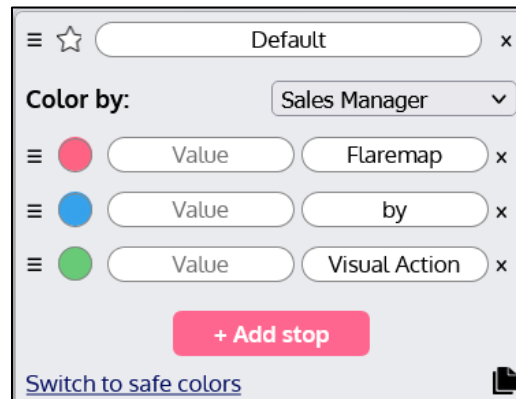
If you haven’t already, give the option an appropriate name.

Repeat this process for as many size options as you would like.

NOTE: A “Same Size” size option, which makes all cells the same size, is automatically created and is always available. The “Same Size” size option cannot be modified or removed.

5.4.6 Creating Color Options

The third option (the final option in the “Group, Size, and Color” creator group) you will see to create for your Flaremap is a color option.



Begin by clicking the + button to the right of the “Color Options” header. You will see a new color option card appear.

You can name the color option now, or wait until you select the column of data to color by to enter an appropriate name.

In the “Color by” dropdown, select the column of data you would like this option to color by. Recall that color options can be used on both **TEXT** (including **TEXT|ENUM**) as well as **NUMBER** data types; however, the legend type (and therefore configuration) will be slightly different depending on the data type.

Color Spectrums for NUMBER Data Type

If you have chosen a numerical data column for your color option, you will enter numerical values as stops for colors, and this will create a **spectrum**.

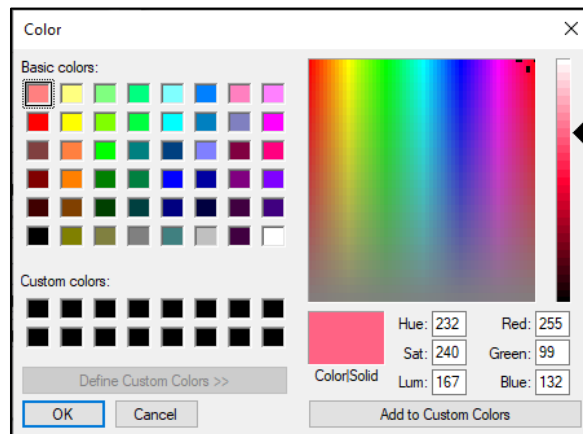
Click “+ Add stop” to add a color stop. You will see a color stop creator line appear.



Click the color bubble to the left (default color is black), then select the desired color for this stop.



Google Chrome



Mozilla Firefox

NOTE: Color-pickers are a feature of the browser, and thus the interface where you choose your colors may not look like the above examples.

Once you have chosen a color, click inside the input box that says “Value” (immediately to the right of the color bubble). Enter the **numerical** value that should correspond to this color on the spectrum.

NOTE: Flaremaps will automatically create a gradient spectrum from the colors you enter. Enter the main colors you want used; there is no need to add dozens of colors to achieve a spectrum with a gradient.

Finally, click in the second input box (to the right), labeled "Legend Label." Enter the text that you want displayed on the color spectrum while this color option is being used.

NOTE: Try to keep labels short, as once you create many color stops, long names may start to overlap each other. Labels may also be left blank to leave space in the spectrum.

Repeat this process for each color stop you want to add to the spectrum.

Color Legends for TEXT or TEXT|ENUM Data Type

If you have chosen a textual data column for your color by, you will enter text values as stops for colors, and this will create a **legend**.

Click "+ Add stop" to add a color stop. You will see a color stop creator line appear.



Click the color bubble to the left (default color is black), then select the desired color for this stop.

Once you have chosen a color, click inside the input box that says "Value" (immediately to the right of the color bubble). Enter the **text** value that should correspond to this color on the spectrum.

Finally, click in the second input box (to the right), labeled "Legend Label." Enter the text that you want displayed alongside this color in the legend while this color option is being used.

Repeat this process for each color stop you want to add to the legend.

Once you have added all the colors you need to your option, give the option an appropriate name, if you haven't already.

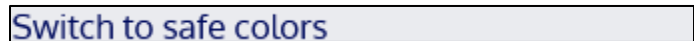
Repeat this process for as many color options as you would like.

NOTE: A "Default" color option is created when a Flaremap is first deployed. You may modify this option to make it one of your own color options, or simply delete it and create a new option of your own.

Creating Color-safe (Colorblind) Options

The Visual Action Platform and Flaremap Development Kit support the creation of "color-safe" color options, which define alternative color spectrums and legends that assist those affected by different forms of colorblindness in interpreting Flaremaps appropriately.

Color-safe options can be defined for each color option that has been created. To switch to editing the color-safe version of a color option you've created, simply locate that option in the list, and click "Switch to safe colors."



Switch to safe colors

The color bubbles in the creator will change to show the colors selected for the color-safe option. By default, they are all black. You can now select colors to use for the color-safe version of this color option.

Note: You do not need to change or modify the data column, values, or labels used when changing the color-safe colors.

Note: Flaremaps will automatically apply color-safe options from a set of predefined legends if a custom color-safe option is not defined.

You can return to editing the standard color option by clicking "Switch to standard colors" while editing safe colors.

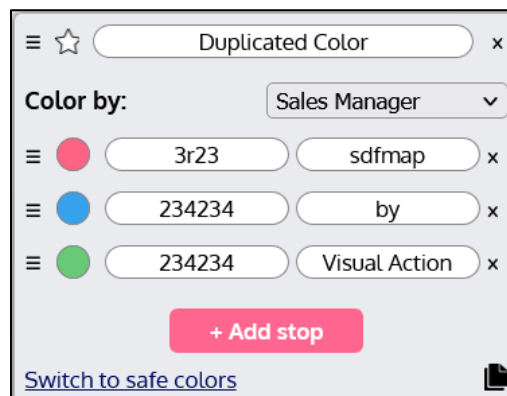
Duplicating a Color Option

If you want to reuse a color spectrum or legend that you've already created, without having to reselect each of the individual colors, color options can be easily duplicated.

To duplicate a color option, click the  icon in the lower-right corner of the color option you'd like to duplicate.



A duplicate of the color option will be created and added to the end of the color options list. All of its parameters will be identical to the color you duplicated, except the name, which will be identified as "Duplicated Color."



5.4.7 Creating (Standard) Filters

In the next creator group, "Filters," the first option you will see to create for your Flaremap is "Filters." These are the standard select filters.

Begin by clicking the + button to the right of the "Filters" header. You will see a new filter card appear.

You can name the filter now, or wait until you select the column of data to filter by to enter an appropriate name.

In the “Filter by” dropdown, select the column of data you would like this option to filter by. Recall that filter data options come from columns of data type **TEXT|ENUM**.

Why Only TEXT|ENUM for Filters?

Data columns that are TEXT|ENUM have repeated values, whereas TEXT columns tend to have mostly unique values. This is why TEXT|ENUM is used for group options.

Using TEXT columns for filters would likely result in the list of options for the filter being close to, if not exactly, as long as the list of rows in the data. In most cases, it’s illogical to use a filter in this way to locate a single, or perhaps a couple, cell(s).

Navigating to a single cell is much better done using the Search filter, as if the user already knows exactly which filter option(s) he or she would click on to find that single cell, that user could begin typing in the box. This is usually faster than scrolling through a large list of options as well.

In the “Select type” buttons, choose “Single” or “Multiple.” “Single” will create a dropdown that only lets one enumerated value (think of these as groups) from the chosen column be filtered by at a time. “Multiple” will enable multiple enumerated values to be filtered by at once.

If you chose “Multiple,” enter a value of at least 2 into the “List length” input box. This indicates how many options from the list you can see at any given time within the filter, without scrolling.

If you haven’t already, give the filter an appropriate name.

Repeat this process for as many filters as you would like.

Quick-Reference Table

Option	Description	Accepted Values	Default Value
Filter by	The column of data associated with this filter.	Any column name presented in the dropdown list.	First option in the list
Select type	Determines whether one or multiple items from the list may be used to filter by at a time.	Single or multiple (radio buttons).	Multiple
List length	When a type of "Multiple" is used, indicates how many options are visible in the filter's list without scrolling. When type is "Single," option is hidden.	Any positive integer.	5

5.4.8 Creating Indicators

The next option you will see to create for your Flaremap is "Indicators."

Begin by clicking the + button to the right of the "Indicators" header. You will see a new indicator card appear.

You can name the indicator now, or wait until you select the column of data to "indicate" (filter) by to enter an appropriate name.

In the "Indicate" dropdown, select the column of data you would like this option to trigger on (indicate). Recall that indicator data options come from columns of data type **TEXT|ENUM**, for the same reasons as described above for filters.

NOTE: If you are looking for a way to track an individual cell or a few cells as they move around the Flaremap, you can use the "Track this cell" option in the mouseover menu after clicking on the cell you want to track.

In the "Select type" buttons, choose "Single" or "Multiple." "Single" will create a dropdown that only lets one enumerated value (think of these as groups) from the chosen column be indicated at a time. "Multiple" will enable multiple enumerated values to be indicated at once.

If you chose "Multiple," enter a value of at least 2 into the "List length" input box. This indicates how many options from the list you can see at any given time within the indicator, without scrolling.

In the "Indicator color" dropdown, choose the color you would like this indicator to display.

Choose a style from the "Indicator style" dropdown. "Static" will remain a solid color, "blinking" will alternate between a white flash and the chosen color, and "pulsing" will smoothly fade to white, then back to the chosen color.

Choose a position for the indicator to appear on the cell in the "Cell position" dropdown.

NOTE: Indicators may not be positioned on the center of a cell. The center of a cell is used for markers when tracking cells.

If you haven't already, give the indicator an appropriate name.

Repeat this process for as many indicators as you would like.

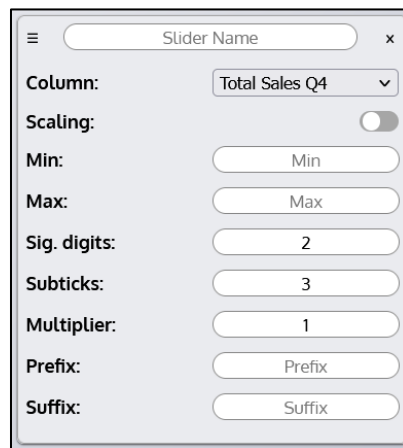
Quick-Reference Table

Option	Description	Accepted Values	Default Value
Indicate	The column of data associated with this indicator.	Any column name presented in the dropdown list.	First option in the list
Select type	Determines whether one or multiple items from the list may be used to filter by at a time.	Single or multiple (radio buttons).	Multiple
List length	When a type of "Multiple" is used, indicates how many options are visible in the indicator's list without scrolling. When type is "Single," option is hidden.	Any positive integer.	5
Indicator color	The color of the indicator on the cell.	Any color presented in the dropdown list.	Green
Indicator style	The visual style of the indicator.	Any style presented in the dropdown list: • Solid: Solid color • Blinking: Blinking between chosen color and white • Pulsing: Smooth fade between chosen color and white	Pulsing
Cell position	The position on a cell where the indicator appears.	Any position presented in the dropdown list: • Top Left • Top Right • Bottom Right • Bottom Left • Top Center	Bottom Right

5.4.9 Creating Sliders

The third option (the final option in the “Filters” creator group) you will see to create for your Flaremap is “Slider.”

Begin by clicking the + button to the right of the “Sliders” header. You will see a new slider card appear.

A screenshot of a configuration dialog box for a slider. The dialog has a title bar with a hamburger menu icon, a text input field labeled "Slider Name", and a close button (x). Below the title bar, there are several settings: "Column:" with a dropdown menu showing "Total Sales Q4"; "Scaling:" with a toggle switch that is currently turned off; "Min:" with a text input field containing "Min"; "Max:" with a text input field containing "Max"; "Sig. digits:" with a text input field containing "2"; "Subticks:" with a text input field containing "3"; "Multiplier:" with a text input field containing "1"; "Prefix:" with a text input field containing "Prefix"; and "Suffix:" with a text input field containing "Suffix".

You can name the slider now, or wait until you select the column of data to filter by to enter an appropriate name.

In the “Column” dropdown, select the column of data you would like this option to filter by. Recall that slider data options come from columns of data type **NUMBER**.

Toggle “Scaling” if you want to scale the slider range. By default, slider ranges are linear. Enabling scaling will draw the first 25% of values over approximately the first 50% of the slider range.

If you want a custom minimum value for the slider range, enter that value in the “Min” input box. Leave this box blank if you want the minimum value from the data to be used.

If you want a custom maximum value for the slider range, enter that value in the “Max” input box. Leave this box blank if you want the maximum value from the data to be used.

In the "Sig. digits" input box, enter the number of decimal places (significant digits) you want to display in the slider labels. A value of 0 will round numbers to whole numbers. Significant digits only display if there are non-whole numbers in the data column selected.

"Subticks" indicates the number of minor ticks to be drawn on the slider range. Enter a custom number if you would like more ticks (and labels) to be drawn on the range.

If desired, enter a multiplier in the "Multiplier" input box. A multiplier will multiply all values by the entered number. This may be useful, for example, when creating a slider for very large numbers, especially in conjunction with prefixes and suffixes.

In the "Prefix" and "Suffix" inputs, enter any text you would like to appear before and after, respectively, each label on the slider. For example, when dealing with monetary values in the thousands, you may want to enter a prefix of "\$" and a suffix of "K" (with a multiplier of 0.001). This (with the multiplier) would turn a value of 100,000 from the data into "\$100K" on the slider range.

If you haven't already, give the slider an appropriate name.

Repeat this process for as many sliders as you would like.

Quick-Reference Table

Option	Description	Accepted Values	Default Value
Column	The column of data associated with this slider.	Any column name presented in the dropdown list.	First option in the list
Scaling	Enables scaling of the slider range. When disabled, a linear scale is drawn. When enabled, the first 25% of values will be drawn across approximately the first 50% of the slider range.	Enabled or disabled (toggle).	Disabled
Min	The minimum value for the slider range. When blank, the minimum value from the selected data column is used.	Any number greater than the minimum within the data.	N/A
Max	The maximum value for the slider range. When blank, the maximum value from the selected data column is used.	Any number less than the maximum within the data.	N/A
Sig. digits	The number of significant digits (decimal places) to use after non-whole numbers.	Any non-negative integer.	2
Subticks	The number of subticks (and therefore labels) to include on the slider range.	Any positive integer.	3
Multiplier	A value to multiply slider values by.	Any number.	1
Prefix	Text to place before the value in a slider label.	Any text.	N/A
Suffix	Text to place after the value in a slider label.	Any text.	N/A

5.4.10 Setting the Track Cell Icon

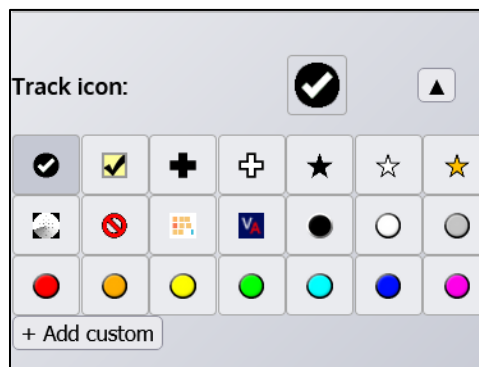
In the next creator group, “Cells,” the first option you will see is a control for setting the “track icon.”



When using Flaremaps, cells can be “tracked.” Tracked cells have a marker placed in their centers, which persists even when cells rearrange due to changed groups, sizes, colors, labels, and filters.

By default, the icon representing a tracked cell looks like this: . This icon can be changed.

Click the dropdown arrow next to the current track icon (the one displayed). This will reveal a list of included track icons.



Select the icon you want to use for tracked cells from the available options. You will see the current track icon update to the icon you selected.

If you want to use a custom icon, click the “+ Add custom” button below the icon list to upload a custom icon. Once uploaded, select it from the list to use it.

5.4.10 Creating Cell Labels

In the “Cells” creator group, the second option you will see to create for your Flaremap is “Cell Labels.” These are the labels that go directly on top of cells.

Begin by clicking the + button to the right of the “Cell Labels” header. You will see a new cell label card appear.

The screenshot shows a configuration window for a cell label. At the top, there is a header bar with a hamburger menu icon, a star icon, a text input field labeled "Cell Label Name", and a close button (x). Below the header, the main area is titled "Contents:". Under this title is a large text input area containing the placeholder text "Cell contents". At the bottom of the window, there is a row of controls: an "Insert" button, the text "value of:", and a dropdown menu currently showing "Sales Rep" with a downward arrow.

Enter a name for the cell label option.

In the “Cell contents” input area, enter the desired configuration for your cell labels.

Enter any text you want to appear in the cell label in the “Cell contents”. To insert a value corresponding to the cell’s data, use the dropdown to the right of “Insert value of” to select the column whose data you want in the cell label. Once you’ve selected a column, click “Insert,” and the Platform will insert the appropriate template into the “Cell contents” box.

You may also manually enter column data templates into the “Cell contents” box if you are familiar with the technical configuration parameters of Flaremaps (column within curly braces { }, e.g.: {sales_rep}). Remember to use the actual column name (key) in templates, rather than the alias created in the Data Manager. See the **Flaremap Development Kit Technical Reference** for more detailed information on cell label configuration.

You may enter text of your own in cell labels as well. For example, if you want to include an ID number (with a column name of “client_id” in the cell label, and also indicate that this is an ID, you could write:

Example: Client ID #: {client_id}

To make a line break and continue your label onto a new line, simply hit `Return` (or `Enter`) and continue writing on the next line.

See the **Flaremap Development Kit Technical Reference** for more detailed information on cell label configuration.

Repeat this process for as many cell label options as you would like.

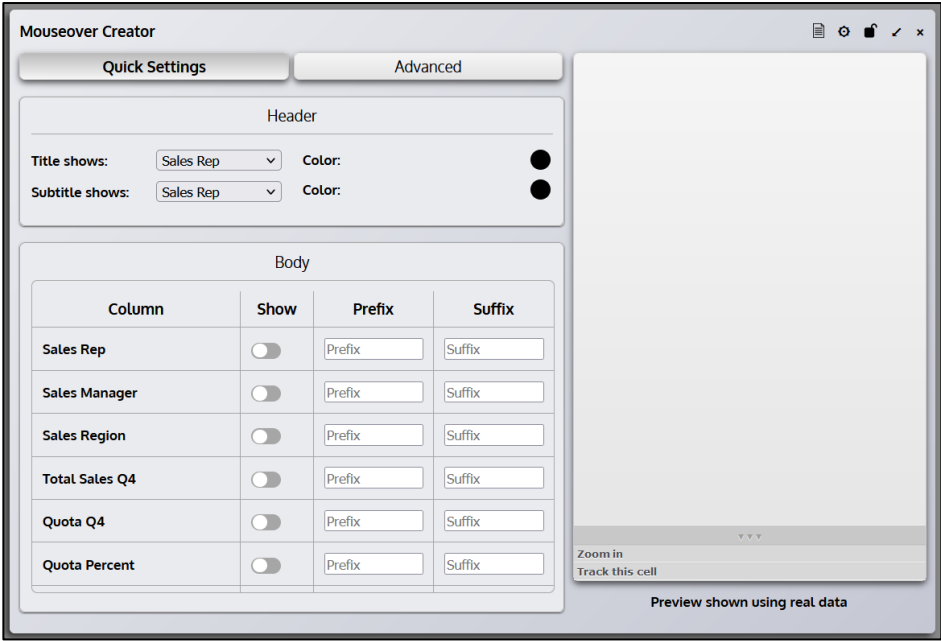
5.4.11 Creating Cell Mouseovers

In the next creator group, “Cell Mouseover,” you are able to launch the Mouseover Creator to create custom cell mouseover boxes.

To launch the Mouseover Creator, click “Launch Mouseover Creator”.



This will open the Mouseover Creator, a special Extension that can only be opened while editing a Flaremap. Only one Mouseover Creator per Flaremap may be open at a time.



There are three main components to the Mouseover Creator: the designer tabs, the composer, and the previewer.

Designer Tabs

The Mouseover Creator’s designer tabs appear at the top left of the Extension. These tabs allow you to toggle between using the “Quick Settings,” for quickly generating a useful mouseover box, and the “Advanced” creator, which allows much more control and customization over the contents of the cell mouseover.



Composer

The Composer section of the Mouseover Creator is comprised of the entire space beneath the Designer Tabs on the left of the Extension. This area populates with the controls used for creating a mouseover, depending on the Designer Tab which is currently selected.

Header

Title shows:Sales Rep

Color:

Subtitle shows:Sales Rep

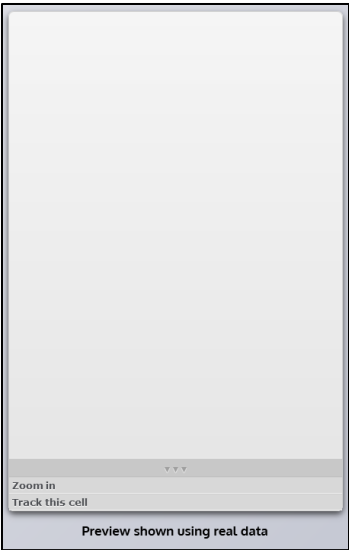
Color:

Body

Column	Show	Prefix	Suffix
Sales Rep	☐	Prefix	Suffix
Sales Manager	☐	Prefix	Suffix
Sales Region	☐	Prefix	Suffix
Total Sales Q4	☐	Prefix	Suffix
Quota Q4	☐	Prefix	Suffix
Quota Percent	☐	Prefix	Suffix

Previewer

The Previewer of the Mouseover Creator is shown to the right of the Extension and resembles a mouseover box which would appear over the actual Flaremap. This previewer is always present and updates as you edit to display a sample of what your current mouseover design will look like.



The previewer includes mouseover menu items at the bottom to represent the menu that will appear in the actual mouseover box. The menu and its items within the preview do not perform any actions, and menu items cannot be edited.

The following sections will describe how to use both the Quick Settings and Advanced options to create a mouseover box using the Mouseover Creator.

5.4.11.1 Quick Settings

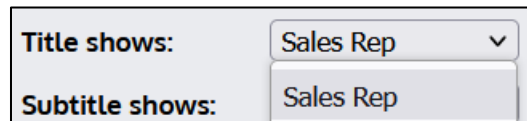
The Quick Settings version of the composer provides a quick and easy way to create a useful cell mouseover without having control over several details.

Header

The “Header” section of the Quick Settings composer is where the title and subtitle shown in the mouseover can be selected.

The title of the mouseover box is the largest text which appears topmost in the mouseover box. It is often filled with a unique identifier for a cell (such as is value in the Primary Key column of the data).

To select a title, choose the column from which you want data shown in the “Title shows” dropdown.

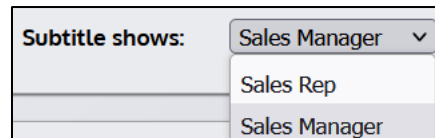


To change the color of the title, use the “Color” bubble to the right of the “Title shows” option to select a color.



The subtitle of a mouseover box appears directly beneath the title and shows a piece of data that is also relevant to the cell. It is not necessarily unique.

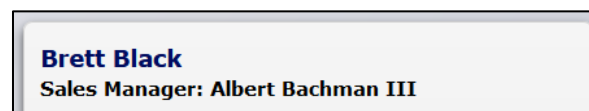
To select a subtitle, choose the column from which you want data shown in the “Subtitle shows” dropdown.



To change the color of the subtitle, use the “Color” bubble to the right of the “Subtitle shows” option to select a color.



The Previewer will automatically update as you make changes to display your mouseover.



Body

The “Body” section of the Quick Settings composer is where you can choose which data from a cell is displayed inside the mouseover box.

The “Body” section will automatically be populated with a row for each column in the dataset to which the Flaremap is linked. There are 4 columns shown in this table:

- Column: identifies the column from the data being represented
- Show: indicates whether to show data from that column
- Prefix: text which is displayed before that column’s value(s)
- Suffix: text which is displayed after that column’s value(s)

For each column you want displayed in the mouseover box, toggle the “Show” switch on. If you want text displayed before or after that column’s values in the mouseover box, enter that text accordingly into the “Prefix” and “Suffix” input fields.

Column	Show	Prefix	Suffix
Sales Rep	☐	Prefix	Suffix
Sales Manager	☐	Prefix	Suffix
Sales Region	☒	Prefix	Suffix
Total Sales Q4	☒	\$	Suffix
Quota Q4	☒	\$	Suffix
Quota Percent	☒	Prefix	%

Once you've finished, your mouseover box will be shown in the Previewer.

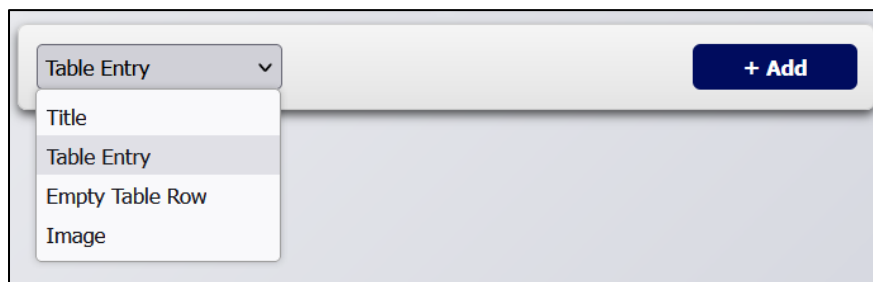
Brett Black	
Sales Manager: Albert Bachman III	
Sales Region:	Wesley Chapel
Total Sales Q4:	\$478920
Quota Q4:	\$614000
Quota Percent:	50%

5.4.11.2 Advanced

The Advanced version of the Mouseover Creator composer provides several additional controls for creating and customizing components of your cell mouseover.

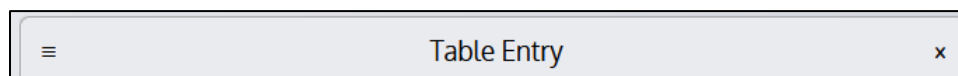
The topmost component of the Advanced composer is where you can select the type of element to insert into the mouseover box. There are four options:

- Title: a title element (single value with no label)
- Table entry: a single value with a label
- Empty table row: a blank space
- Image: an image



Select the item you want to create from the dropdown list, then click "+ Add" to add it to the mouseover box.

You will see a creator for the item type you selected appear in the composer. Each type will be described in detail below, but each creator shares some common controls.



Sort Handle

The ≡ (triple-bar or "hamburger") icon in the upper left of a creator can be used to drag and sort the order of items in the mouseover.

Item Type

In the top center of a creator, the type of item which it represents is shown.

Delete Button

In the upper right of a creator is an **x** button. Clicking this button will delete the item and remove it from the mouseover.

Each of the item types you can add to the cell mouseover in the Advanced composer are described below:

Title

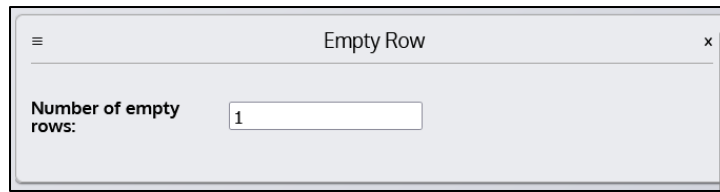
A title element shows a single value with no label. You can adjust the data shown in a title, as well as enter a prefix and suffix and change its color, font size, and alignment.

Table Entry

A table entry is similar to a title in that it shows a single value and its prefix, suffix, color, and font size can be changed. A table entry, however, contains a label which can be customized, and is always left-aligned.

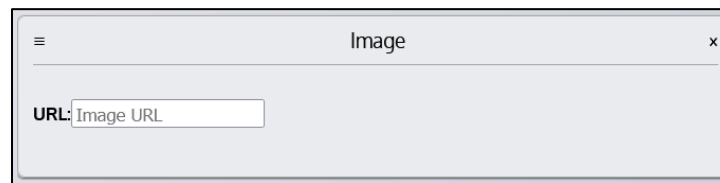
Empty Table Row

An empty table row creator allows you to specify a number of empty rows which may be used to separate sections of information in your mouseover box.

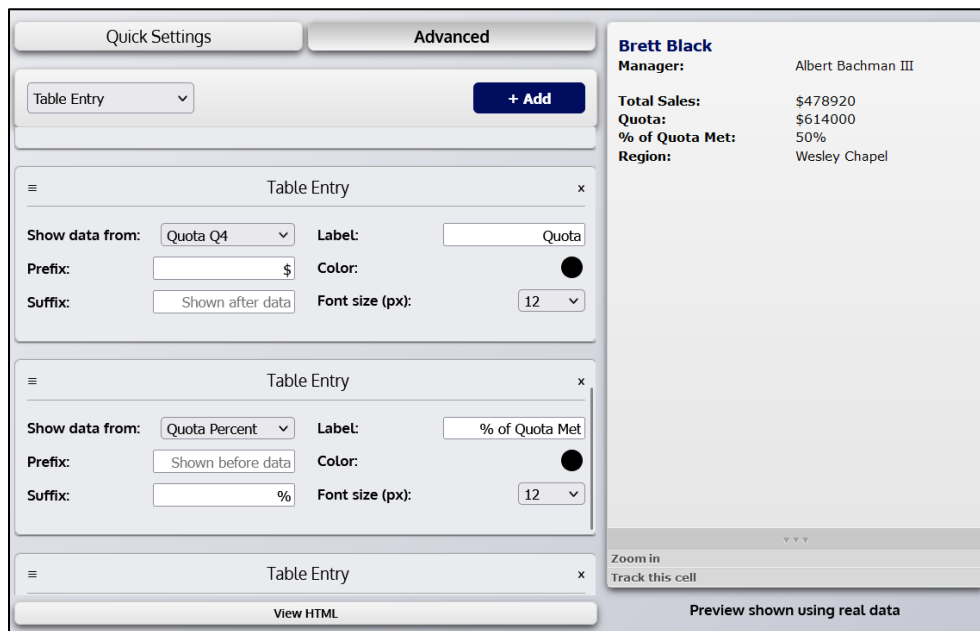


Image

An image creator allows you to enter a URL to an image, which will be displayed in your mouseover box.



Use the different creator types to design and customize your mouseover to meet your specific needs. The Previewer will continually update to reflect the changes you make in your design.



Some users may want to view the HTML generated when designing a mouseover, either for use in other (Flaremap Development Kit) applications or to modify manually to customize even further.

To view and edit the HTML of your mouseover, while in the Advanced composer, click the “View HTML” button at the bottom of the composer. This will reveal the HTML editor. You may edit the HTML manually here. To return to the composer, click the “Hide HTML” button beneath the code editor, or click the tab for the composer you want to use.

NOTE: Any edits made using the composer after editing the HTML manually will override any manual edits.

5.4.11.3 Applying Cell Mouseovers

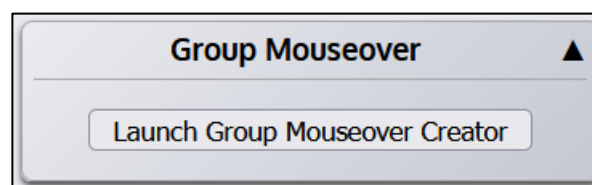
Once you’ve completed designing your cell mouseover using either of the available composers, click the “Apply” button in the Flaremap’s settings to apply the changes (and your cell mouseover) to the Flaremap. Hover over cells in the Flaremap to see your mouseover.

You can relaunch the Mouseover Creator to continue designing or to change your cell mouseover at any time.

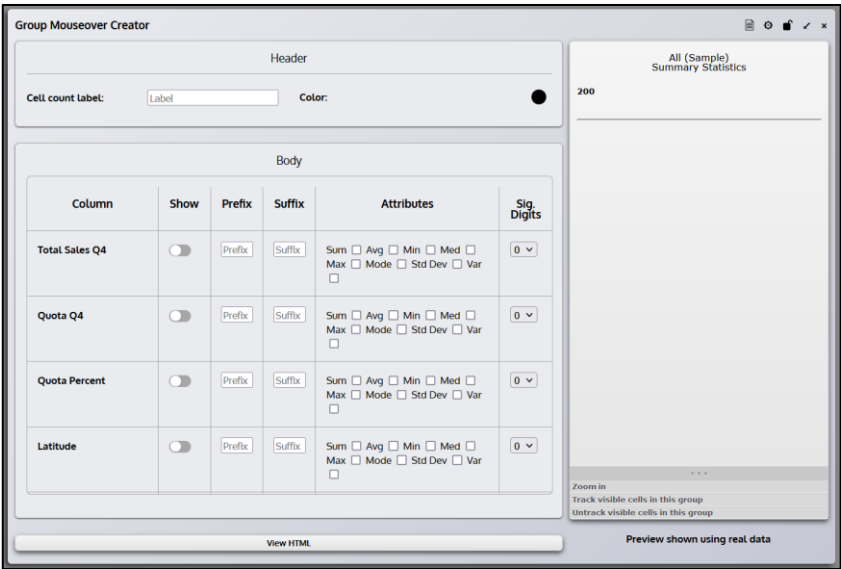
5.4.12 Creating Group Mouseovers

In the next creator group, “Group Mouseover,” you are able to launch the Group Mouseover Creator to create custom group mouseover boxes.

To launch the Group Mouseover Creator, click “Launch Group Mouseover Creator”.



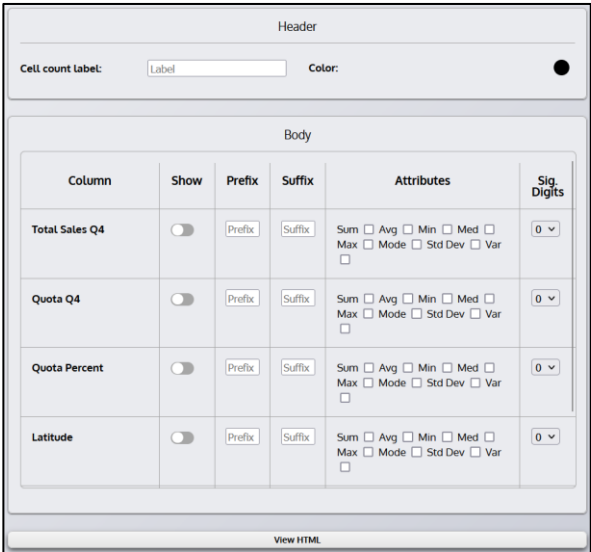
This will open the Group Mouseover Creator, another special Extension that can only be opened while editing a Flaremap. Only one Group Mouseover Creator per Flaremap may be open at a time.



There are two main components to the Group Mouseover Creator: the composer and the previewer.

Composer

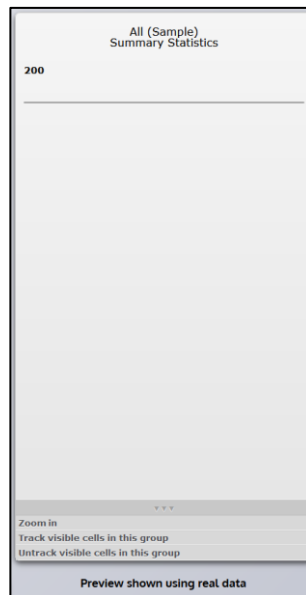
The Composer section of the Group Mouseover Creator fills the left portion of the Extension. This area populates with the controls used for creating a group mouseover.



Previewer

The Previewer of the Group Mouseover Creator is shown to the right of the Extension and resembles a group mouseover box which would appear over the actual Flaremap.

This previewer is always present and updates as you edit to display a sample of what your current group mouseover design will look like.



The previewer includes group mouseover menu items at the bottom to represent the menu that will appear in the actual group mouseover box. The menu and its items within the preview do not perform any actions, and menu items cannot be edited.

The following sections describe how to use the Group Mouseover Creator to produce a group mouseover.

Header

The “Header” section of the composer is where the cell count label and its color can be selected. The title of a group mouseover is automatically generated based on the group’s name.

The cell count label refers to the label used to identify the cells within a group. Each group mouseover displays the total number of cells within that group, and cells represent something from the data. For example, if your cells represent sales representatives, you might use the words “sales reps” as your cell count label. Then, when a group mouseover is displayed, it will display something like “200 sales reps”.

Enter your desired cell count label in the “Cell count label” input field. To change its color, use the “Color” bubble to the right of the “Cell count label” input field.

Cell count label:	Label	Color:	
-------------------	------------------------------------	--------	---

Body

The “Body” section of the composer is where you can choose which aggregate data from a group is displayed.

The “Body” section will automatically be populated with a row for each column of numerical data in the dataset which the Flaremap is linked to. There are 6 columns shown in this table:

- **Column:** identifies the column from the data being represented
- **Show:** indicates whether to show data from that column
- **Prefix:** text which is displayed before that column’s value(s)
- **Suffix:** text which is displayed after that column’s value(s)
- **Attributes:** indicates which aggregate attributes to display in the table
- **Sig. Digits:** indicates how many decimal places should be shown after values

For each numerical column whose aggregate attributes you want displayed in the group mouseover box, toggle the “Show” switch on. If you want text displayed before or after that column’s values in the mouseover box, enter that text accordingly into the “Prefix” and “Suffix” input fields.

Column	Show	Prefix	Suffix	Attributes	Sig. Digits
Total Sales Q4	☒	\$	Suffix	Sum ☐ Avg ☐ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾
Quota Q4	☒	\$	Suffix	Sum ☐ Avg ☐ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾
Quota Percent	☒	Prefix	%	Sum ☐ Avg ☐ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾
Latitude	☐	Prefix	Suffix	Sum ☐ Avg ☐ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾

For each column you've selected, you can choose which aggregate attributes should be displayed in the group mouseover box's table. Aggregate attributes are calculated from the set of values from a column across each cell in a group. There are 8 available attributes:

- Sum: sum
- Avg: average/mean
- Min: minimum
- Med: median
- Max: maximum
- Mode: mode
- Std Dev: standard deviation
- Var: variance

Check the box for each attribute you want displayed for each column. Use the "Sig. Digits" dropdown to limit the number of digits following the decimal point in attribute values.

Column	Show	Prefix	Suffix	Attributes	Sig. Digits
Total Sales Q4	☒	\$	Suffix	Sum ☒ Avg ☒ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	1 ▾
Quota Q4	☒	\$	Suffix	Sum ☒ Avg ☒ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾
Quota Percent	☒	Prefix	%	Sum ☐ Avg ☒ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	3 ▾
Latitude	☐	Prefix	Suffix	Sum ☐ Avg ☐ Min ☐ Med ☐ Max ☐ Mode ☐ Std Dev ☐ Var ☐	0 ▾

Once you've finished, your group mouseover box will be shown in the Previewer.

All (Sample) Summary Statistics		
200		
	Average	Total
Total Sales Q4	\$393726.8	\$78745360
Quota Q4	\$369730	\$73946000
Quota Percent	106.305%	
▼ ▼ ▼		
Zoom in		
Track visible cells in this group		
Untrack visible cells in this group		
Preview shown using real data		

5.4.12.1 Applying Group Mouseovers

Once you've completed designing your group mouseover using the composer, click the "Apply" button in the Flaremap's settings to apply the changes (and your group mouseover) to the Flaremap. Hover over groups in the Flaremap to see your group mouseover.

You can relaunch the Mouseover Creator to continue designing or to change your group mouseover at any time.

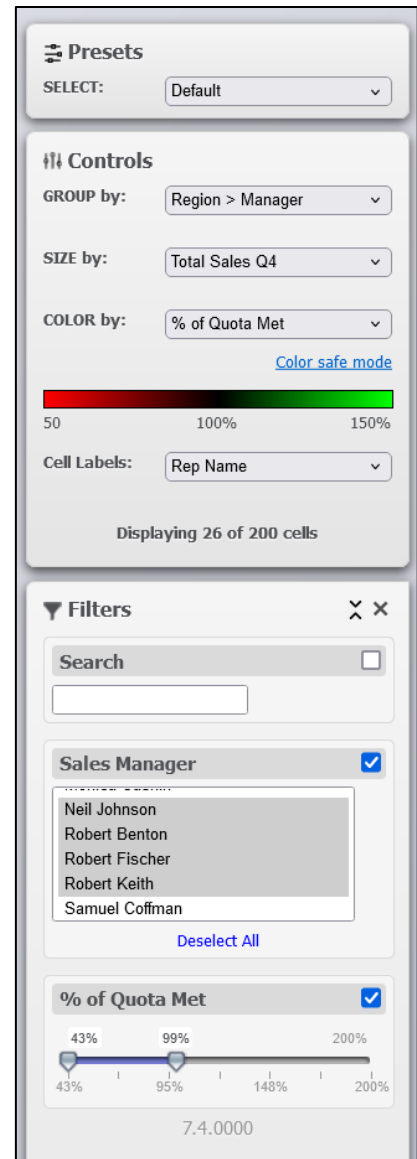
5.4.13 Creating Presets

Creating a Flaremap preset is a fast and efficient way to reorient the Flaremap to particular set of options that achieve a specific goal. Flaremap presets apply a pre-configured set of selections (group, size, color, cell labels, and filters) to the Flaremap with a single click. Presets are often built around use cases, allowing users to quickly toggle the Flaremap's options to drive out qualifying anomalies, but can be used for many other tasks.

Configuring a Preset

To create a Flaremap preset, first select the Flaremap options which should be applied (see example to the right).

- Select the group option to apply.
- Select the size option to apply.
- Select the color option to apply.
- Select a cell label option to apply, if any.
- Enable and adjust any and all filters to apply, if any.



Saving a Preset

Once you've selected all of the Flaremap options to be applied by a preset, locate the Flaremap Presets toggle in the Flaremap (Extension) header. The presets toggle looks like this: .

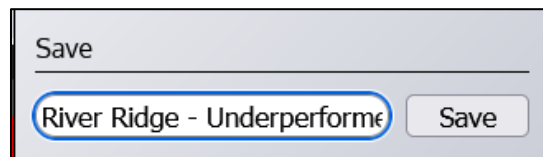


Click the presets toggle. The Flaremap Presets menu will open.



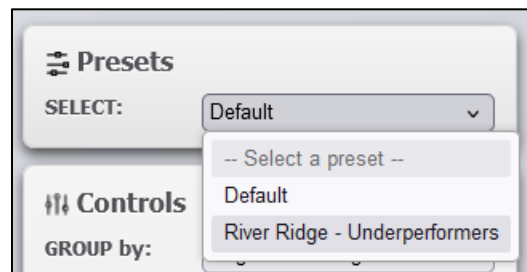
The dialog box is titled "Flaremap Presets". It has two sections: "Save" and "Delete". The "Save" section contains a text input field labeled "Preset Name" and a "Save" button. The "Delete" section contains a dropdown menu and a "Delete" button.

Under "Save", in the input field that says "Preset Name", enter a name to assign to the preset. Each preset name must be unique. A preset may not be named "Default".



The dialog box is titled "Flaremap Presets". The "Save" section shows the text "River Ridge - Underperformers" entered into the "Preset Name" field. The "Delete" section is empty.

Once a qualifying name has been entered, click the "Save" button. The Flaremap will update, and the preset you just created will now be available in the "Presets" control panel within the Flaremap.



The control panel shows two sections: "Presets" and "Controls". The "Presets" section has a "SELECT:" label and a dropdown menu currently showing "Default". The "Controls" section has a "GROUP by:" label and a dropdown menu. The dropdown menu for "SELECT:" is open, showing a list of options: "-- Select a preset --", "Default", and "River Ridge - Underperformers".

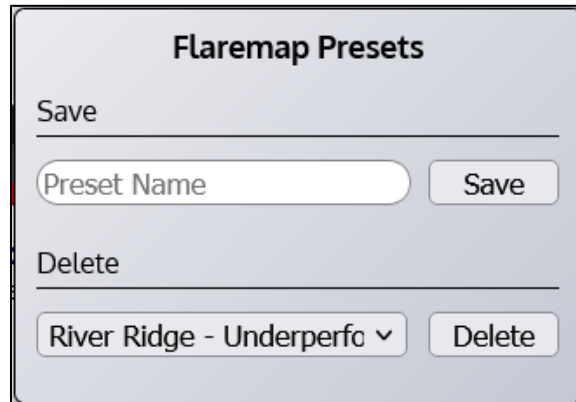
Simply select your preset from the list to apply it. Select "Default" to revert the Flaremap to its default options. Presets can be toggled between at any time.

Repeat this process for as many Flaremap Presets as necessary. There is no limit on the number of presets that can be configured for a Flaremap.

Note: Flaremap Presets are associated with only a single Flaremap. To create an identical preset for another Flaremap, it must be created again within that Flaremap.

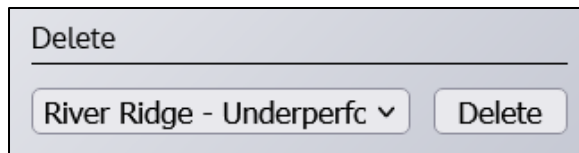
Deleting a Preset

To delete a Flaremap Preset, first open the Flaremap Presets menu using the preset toggle (described above in “Saving a Preset”).



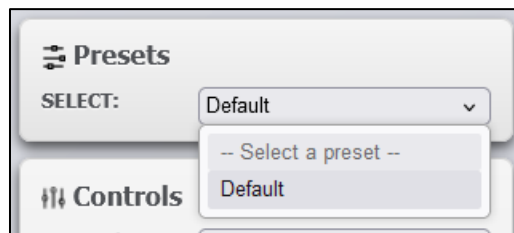
The screenshot shows a dialog box titled "Flaremap Presets". It has two sections: "Save" and "Delete". The "Save" section contains a text input field labeled "Preset Name" and a "Save" button. The "Delete" section contains a dropdown menu showing "River Ridge - Underperfo" and a "Delete" button.

In the dropdown under “Delete”, locate and select the preset you want to delete. Once selected, click the “Delete” button to remove it.



This is a close-up of the "Delete" section of the dialog box. It shows a dropdown menu with the text "River Ridge - Underperfc" and a "Delete" button.

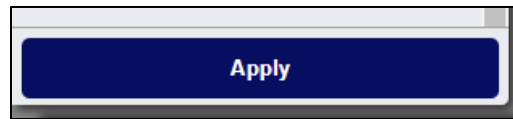
The Flaremap will update, and the deleted preset will no longer be an available option in the Flaremap’s “Presets” control panel.



The screenshot shows a control panel with a "Presets" section. It has a "SELECT:" label and a dropdown menu showing "Default". Below the dropdown is a list of options: "-- Select a preset --" and "Default".

5.4.14 Saving Settings

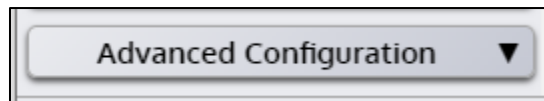
Once you have configured your Flaremap, click the “Apply” button at the bottom of the Editor to immediately apply your changes to the Flaremap. You will receive a notification that the settings were successfully applied, and you will see the Flaremap update immediately.



You can apply changes frequently throughout your creation of a Flaremap to test options out. There is no need to fully configure every portion of the Flaremap before applying changes.

5.5 Advanced Flaremap Configuration

The final creator group available in the Flaremap configuration options is “Advanced Configuration.” This contains advanced controls for experienced Flaremap users and technical personnel.



Only modify controls in this section if you are well-experienced with creating Flaremaps in older versions of the software (Flaremap JS 6.9 and below) and/or are thoroughly familiar with the contents of the **Flaremap Development Kit Technical Reference**.

5.5.1 Cell Display

This control group contains options for the drawing and display of cells within the Flaremap. They may be modified to improve accessibility of the Flaremap as well as to achieve custom Flaremap designs.

A configuration panel with a light gray background and a thin black border. It contains five settings: "Gutter size (px)" with a text input field containing "0"; "Invert cell sizes:" with a toggle switch that is currently turned off; "Min. cell size (px)" with a text input field containing "10"; "Cell size ratio" with a text input field containing "0"; and "Cell paint type" with a dropdown menu showing "Gradient light → dar" and a downward arrow.

Gutter size (px)

The gutter size defines the width of the lines between cells in the Flaremap field. By default, there is no space between cells. Increase this number to increase the line width between cells. Width is measured in pixels (px).

Invert cell sizes

Inverting the size of cells uses different logic than the typical Flaremap algorithm's logic. Cells that would normally be the smallest (lowest value)

become the largest cells, and the largest become the smallest, respectively. This feature is useful when the most important values are actually the smallest values rather than the largest.

Min. cell size (px)

This value enforces a minimum size for cells, in pixels (px). No cell may be drawn smaller than the minimum cell size. This is useful when there may be many small cells that become very tiny relative to the largest cells, and often therefore become hard to pinpoint with the mouse cursor.

Cell size ratio

This value is an approximate ratio between the smallest and largest cells in the Flaremap field. A value of 0 enforces no ratio. A value of 1 makes all cells the same size. Values between 2 and 50 will scale cells between relatively similar in size (2) and significantly dissimilar in size (50).

Cell paint type

The paint type of a cell indicates the graphic style used to paint a cell's color. There are three options. "Solid" paints the cell with a solid color. "Gradient light → dark" (default) gradients the cell color with a lighter color in the upper-left corner and a darker color in the lower-right corner. "Gradient dark → light" inverts this logic, with the darkest color in the upper-left corner and the lightest color in the lower-right corner.

Quick-Reference Table

Option	Description	Accepted Values	Default Value
Gutter size (px)	Size, in pixels, of the space between cells.	Any non-negative integer.	0
Invert cell sizes	Inverts cells sizes, such that the lowest value is represented by the largest cell, rather than the largest value being represented by the largest cell.	Enabled or disabled (toggle).	Disabled
Min. cell size (px)	The minimum size, in pixels, that a cell may be.	Any non-negative integer.	10
Cell size ratio	Defines a ratio for approximate similarity in size between the largest and smallest cells.	• 0: No ratio • 1: All cells are the same size • 2-50: Scales cells between relatively similar in size (2) and significantly dissimilar in size (50)	0
Cell paint type	Indicates the graphic paint type of cells.	• Solid: Solid color for entire cell. • Gradient light → dark: Gradient from light to dark from upper-left to lower-right. • Gradient dark → light: Gradient from dark to light from upper-left to lower-right.	Gradient light → dark

5.5.2 Manual Configurations

The Manual Configurations section allows technical and experienced users to manually enter Flaremap configuration parameters. This may be especially useful for those building new Flaremaps, but who are very familiar with older versions of the software (Flaremap JS 6.9 and below).

Manual Configurations

You can directly paste your own Flaremap configuration key-value pairs into the box below, one per line, separated with commas. Include quotations around necessary keys. No colons allowed in key names. Object braces ({ }) not necessary.

This may be useful for copying old configurations, controlling specific settings, or achieving advanced functionality, but will likely cause problems with the Flaremap extension if you do not know what you are doing.

Manual configurations made here will override corresponding configurations set above.

viewConfig

Clear

Enter viewConfig key-value pairs here...

Advanced users may enter parameters for the viewConfig (Flaremap configuration JSON object). Establishing these parameters is described in the **Flaremap Development Kit Technical Reference**.

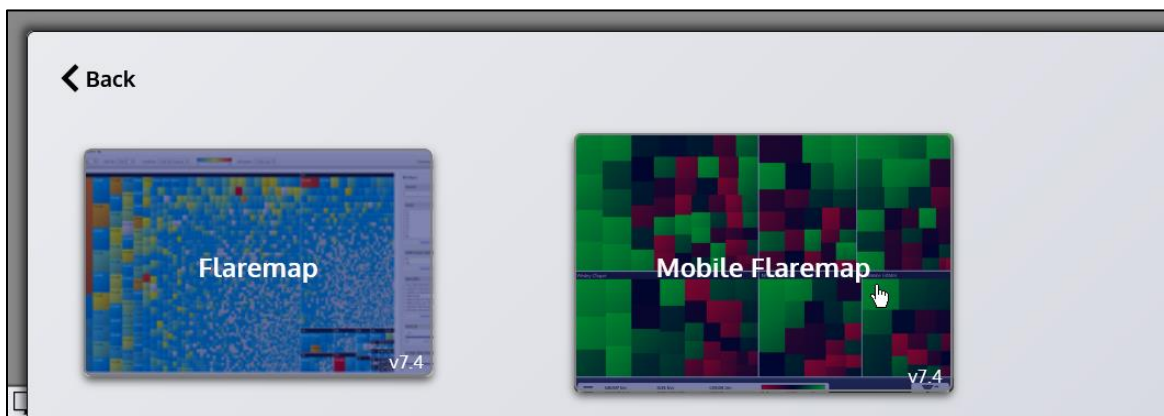
Manual configurations will be applied when the “Apply” button is used, along with other Flaremap settings modified in the Editor panel. Manual configurations will override settings made in the above controls in the Editor. If not being used, leave these input areas blank, or click the “Clear” button near each one to ensure they are empty, such that improper parameters don’t override other settings and cause the Flaremap to malfunction.

5.6 Mini/Mobile Flaremaps

Flaremaps also feature a compact (mobile-friendly) display. When using embedded Flaremaps, this compact display is automatically utilized when viewing the Flaremap on a mobile device. This compact interface, however, can be useful in several other circumstances. Sometimes, when using a second or third Flaremap on a single interface, it's helpful to have a more compact view of the secondary (or tertiary) Flaremaps. Perhaps you simply want the Flaremap controls to be collapsed because users won't be changing them frequently.

Whatever the case may be, the Platform makes the compact Flaremap available to easily deploy into applications. This version, when used in the Platform, is forced to always show in the compact, mobile-friendly format.

To deploy a Mini Flaremap, navigate to the Workflow Extension Drawer, and choose the "Flaremap" Suite (the same location where you can find standard Flaremaps). Click and drag a "Mini Flaremap" out onto the Desktop.



All of the same settings and controls available for a standard Flaremap are also available for a Mini Flaremap. The only difference is the actual display/interface of the Flaremap field and controls within the Extension body.

6 Index

Advanced Flaremap Configuration, 157
 Animations, 75
 Application, 28, 36, 38, 39, 40, 45, 47, 48, 49, 51, 85, 87
 Autosave, 75
 Cell Labels, 108, 119, 136, 137
 Cells, 103, 105, 106, 107, 108, 109, 110, 111, 116, 123, 124, 131, 137, 157, 158, 159
 Collapsing, 49, 76, 78
 Color, 108, 119, 124, 125, 126
 Controls, 106, 109, 115
 Data, 22, 31, 43, 52, 53, 54, 55, 56, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 82, 88, 97, 98, 99, 104, 115, 116, 125, 126, 137
 Data sources, 31, 52, 53
 Data Object, 53, 54, 55, 56, 58, 59, 60, 61, 62, 63, 64, 65, 66, 69, 70, 71, 73, 104
 Desktop, 25, 26, 27, 33, 35, 74, 76
 Dragging, 25
 Editor, 31, 34, 36, 37, 38, 72, 73, 79, 80
 Extensions, 7, 33, 35, 36, 37, 46, 74, 75, 76, 77, 78, 79, 81, 94
 Filter, 109, 110, 111, 112, 113, 119, 128, 129, 130, 132, 133
 Filters, 109, 110, 111, 112, 113, 119, 128, 133
 Flaremap, 19, 20, 34, 68, 70, 83, 92, 102, 104, 105, 106, 107, 108, 109, 111, 112, 113, 115, 116, 117, 119, 120, 121, 123, 124, 126, 127, 128, 130, 131, 133, 137, 138, 140, 142, 146, 147, 148, 150, 152, 153, 154, 155, 156, 157, 158, 160, 161, 163
 Grid Snapping, 74
 Group, 68, 107, 119, 121, 122, 124
 Help, 31, 35, 94
 Indicators, 110, 111, 130
 Installation, 8, 12, 13, 15, 16, 17, 19, 22, 42
 Interactions, 25
 Interface, 7, 9, 24, 28, 33, 35, 37, 51, 76, 77, 78, 82, 88, 92, 95, 96, 99, 105, 110, 116, 117, 120, 121, 125, 161
 Interface, 24
 Lock, 77
 Menu, 29, 30, 31, 33, 35, 38, 43, 47, 48, 49, 52, 72, 94
 Mouseover Boxes, 138
 Naming, 49, 51
 Navbar, 28, 29, 30, 51
 Preferences, 31, 36, 38, 43, 72, 73, 75
 Publishing, 87
 Pull Tab, 33
 Reporting, 75
 Resizing, 26
 Search, 111
 Settings, 31, 36, 38, 43, 72, 73
 Size, 107, 119, 123, 124, 159
 Sliders, 111, 133
 Title, 28
 Workflow Extension, 7, 27, 31, 34, 36, 37, 79

7 Contact

For questions, comments, or any other assistance, get in touch with us using any of the following:



Visual Action Software, Inc.

+1 972 808 0400

info@visualaction.com

www.visualaction.com/contact