

# **Oracle® Big Data Discovery Cloud Service**

Studio User's Guide

**E65365-05**

November 2016

Copyright © 2016, 2016, Oracle and/or its affiliates. All rights reserved.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

Preface .....	xiii
About this guide .....	xiii
Audience .....	xiii
Conventions.....	xiii
Contacting Oracle Customer Support .....	xiv
<b>Part I Overview of Studio</b>	
<b>1 Signing In and Managing Your Account</b>	
Signing in to Studio .....	1-1
Resetting your password .....	1-2
Changing your password .....	1-2
About locales .....	1-3
Configuring the locale for your account .....	1-4
Selecting a locale from the user menu .....	1-5
<b>2 Using the Studio Catalog</b>	
About the Catalog.....	2-1
Browsing data sets and projects .....	2-2
Filtering and searching data sets and projects.....	2-3
Editing data set or project names, descriptions, and tags.....	2-4
<b>Part II Searching and Filtering in Studio</b>	
<b>3 Searching in Studio</b>	
About Studio search .....	3-1
Displaying and using the results of a search .....	3-2
About keyword searches .....	3-4
About Boolean searches .....	3-6
Enabling or disabling full text search and typeahead search on an attribute.....	3-8

<b>4</b>	<b>Filtering Based on Refinements</b>	
	About using refinements for filtering .....	4-1
	Selecting refinement values for filtering .....	4-2
	Selecting a range of refinement values .....	4-4
	Selecting a recurring date/time .....	4-5
	About implicit refinements .....	4-7
	Managing Selected Refinements.....	4-7
	About the Selected Refinements panel .....	4-7
	Removing refinements from the Selected Refinements panel .....	4-9

### Part III Managing Data Sets

<b>5</b>	<b>Managing a Data Set</b>	
	Options for loading data into Studio .....	5-1
	About creating a data set in the Catalog .....	5-2
	Creating a data set from a file.....	5-3
	Creating a data set from a JDBC data source .....	5-5
	About updating a data set in the Catalog .....	5-6
	Updating a data set from a file .....	5-7
	Updating a data set from a JDBC data source.....	5-8
	Setting attribute metadata in data sets .....	5-9
	About notifications for data set changes .....	5-11
	Managing access to a data set .....	5-13
	Deleting a data set.....	5-15
<b>6</b>	<b>Exploring a Data Set</b>	
	About exploring a data set .....	6-1
	Selecting the data to explore .....	6-2
	Switching Explore between tile view and table view.....	6-2
	Filtering and sorting displayed attributes in Explore or Transform .....	6-3
	Working with an individual attribute in Explore .....	6-4
	Using the scratchpad .....	6-5
	Keyboard shortcuts for Scratchpad .....	6-9

### Part IV Creating and Configuring Projects

<b>7</b>	<b>Creating, Localizing, and Deleting Projects</b>	
	About projects .....	7-1
	Creating a project.....	7-2
	Localizing the project name, description, page names, and component titles .....	7-2
	Deleting a project using the Catalog .....	7-3

<b>8</b>	<b>Managing Access to Projects</b>	
	Project and User Roles .....	8-1
	Project Types .....	8-2
	Assigning project roles.....	8-3
<b>9</b>	<b>Managing Project Pages</b>	
	Adding a page to a project .....	9-1
	Renaming a page.....	9-2
	Changing the page layout .....	9-2
	Changing the data set binding for a page .....	9-3
	Deleting a page.....	9-3
	Adding and deleting components on a page.....	9-3
<b>Part V</b>	<b>Viewing and Using Projects</b>	
<b>10</b>	<b>Navigating Within a Project</b>	
	Navigating to a project.....	10-1
	Navigating within a project.....	10-1
	Displaying information about the project data .....	10-2
<b>11</b>	<b>Creating and Using Bookmarks</b>	
	About bookmarks .....	11-1
	Displaying and navigating to bookmarks.....	11-2
	Creating, editing, and deleting bookmarks .....	11-2
	Generating bookmark links.....	11-4
	Emailing bookmark links.....	11-4
	Bookmark data saved for each component and panel .....	11-5
<b>12</b>	<b>Creating and Managing Snapshots</b>	
	About snapshots and snapshot galleries .....	12-1
	Displaying the list of snapshots and snapshot galleries .....	12-1
	Creating and managing individual snapshots .....	12-2
	Creating a snapshot.....	12-2
	Managing saved snapshots.....	12-3
	Exporting the entire snapshot list.....	12-5
	Emailing the entire snapshot list .....	12-6
	Creating and managing snapshot galleries.....	12-6
	Creating a snapshot gallery .....	12-6
	Editing and deleting snapshot galleries.....	12-8
	Emailing a snapshot gallery.....	12-9
	Exporting a snapshot gallery to a PDF file .....	12-9

## Part VI Managing Project Data

### 13 Managing Project Data Sets

Adding a data set to an existing project .....	13-1
Loading the full data set in a project.....	13-2
Configuring a project data set for incremental updates .....	13-3
Converting a project to a BDD Application .....	13-5
Removing a data set from a project.....	13-6

### 14 Linking Project Data Sets

About data set links.....	14-1
Creating a data set link .....	14-2
Editing a link key .....	14-5
Managing aggregation methods for a linked view .....	14-6
Controlling automatic filtering between linked data sets .....	14-10

### 15 Defining Project Data Set Views

About views.....	15-2
About base views.....	15-2
About linked views .....	15-2
Displaying the list of views for a project.....	15-4
Displaying details for a view .....	15-5
Creating views.....	15-6
Creating a new view .....	15-6
Copying an existing data view .....	15-7
Configuring a view .....	15-10
Editing and localizing the view display name and description .....	15-10
Editing the view definition .....	15-12
Guidelines for using EQL to define a view .....	15-12
Configuring the view attributes .....	15-16
Defining predefined metrics for a view .....	15-29
Previewing the content of a view .....	15-32
Deciding whether to publish a custom view .....	15-33
Deleting a view.....	15-33

### 16 Managing Attribute Groups for a Project

About attribute groups and the Attribute Groups page.....	16-1
About groups for linked views.....	16-2
Displaying the groups for a selected view .....	16-4
Displaying attribute information for a group.....	16-5
Displaying the list of attributes for a group .....	16-5
Displaying the full list of metadata values for an attribute .....	16-6

Selecting the metadata columns to display for each attribute .....	16-6
Creating an attribute group.....	16-7
Deleting an attribute group .....	16-7
Configuring an attribute group .....	16-8
Changing the group name and use.....	16-8
Configuring the group membership .....	16-9
Configuring the sort order for a group .....	16-10
Saving the attribute group configuration and reloading the attribute group cache.....	16-11
<b>17 Exporting Project Data Sets</b>	
About exporting project data .....	17-1
Exporting project data.....	17-2
<b>Part VII Transforming Project Data</b>	
<b>18 Overview of Transforming Project Data</b>	
About transformations and transformation scripts .....	18-2
About the Transform page user interface .....	18-3
Transformation script workflow .....	18-4
Transformations workflow diagram.....	18-4
About Groovy.....	18-5
About transform functions .....	18-6
Transform locking.....	18-6
Preview mode.....	18-7
<b>19 Transforming Project Data Sets</b>	
Overview of transforming an attribute .....	19-2
Transform page controls.....	19-2
Toggling the Transform data grid view .....	19-2
Filtering and sorting the displayed attributes and values .....	19-3
Displaying whitespace characters in attribute values.....	19-4
Basic transform operations.....	19-5
Splitting String attribute values .....	19-6
Trimming spaces from String values.....	19-6
Selecting subsets of date/time values .....	19-7
Changing the capitalization of String values .....	19-7
Performing mathematical operations on numeric values .....	19-8
Conversion transform operations.....	19-8
Changing the data type of an attribute .....	19-8
Advanced transform operations.....	19-9
Obtaining geocode information from attribute values .....	19-10
Extracting parts of a date from attribute values .....	19-13
Extracting names of people, places, or organizations from attribute values.....	19-15

Extracting key phrases from attribute values .....	19-15
Providing a whitelist to extract terms from attribute values .....	19-16
Binning numeric attribute values.....	19-17
Grouping attribute values.....	19-18
Managing null values .....	19-19
Shaping transform operations .....	19-20
Filtering records from the project data set.....	19-21
Deleting an attribute .....	19-22
About aggregating attribute values.....	19-23
Aggregating attribute values.....	19-24
About joining data sets.....	19-26
Joining data sets.....	19-27
Creating custom transformations.....	19-30
Creating a custom transformation .....	19-30
Writing custom transform code .....	19-31
Exception handling and debugging .....	19-37
Managing and executing the transformation script .....	19-41
Reordering, editing, and deleting transformations.....	19-41
Running the transformation script against a project data set.....	19-42
Creating a new data set from the transformed data.....	19-43
Sharing a transform script .....	19-44
Publishing a transformation script .....	19-44
Loading a transformation script.....	19-45
Un-publishing a transformation script.....	19-46
Tips for better performance.....	19-46

## 20 Transform Function Reference

Data types .....	20-1
Data type conversions.....	20-2
Groovy reserved keywords and unsupported functions.....	20-3
Conditional statements and multi-assign values.....	20-5
List of transform functions .....	20-8
Conversion functions.....	20-8
Date functions .....	20-9
Enrichment functions.....	20-13
Geocode functions.....	20-25
Math functions.....	20-26
Set functions .....	20-28
String functions.....	20-29
URL functions .....	20-33



## Part VIII Working with Components

### 21 Component Overview

Component list .....	21-1
Common component controls.....	21-7
Paging through data .....	21-7
Using a component to refine data.....	21-8
Displaying details for a component item.....	21-10
Comparing items within a component .....	21-10
Printing component data.....	21-12
Exporting data from a component.....	21-13
Common component configuration workflows.....	21-14
About configuring components .....	21-15
Renaming a component.....	21-15
Configuring attribute value formats.....	21-16
Selecting the aggregation method to use for a metric.....	21-20
Configuring cascading for dimension refinement.....	21-21
About the component Actions menu.....	21-22
Selecting the actions to include in the Actions menu.....	21-23
Configuring actions for attribute values.....	21-24
About component hyperlinks.....	21-27
About using non-HTTP protocol links.....	21-27
Selecting the target page for a refinement or hyperlink .....	21-27
Recommendations for attribute values that are complete URLs.....	21-29
Indicating whether to encode inserted attribute values .....	21-30
Configuring hyperlinks to external URLs .....	21-31
Configuring a Pass Parameters Actions menu action.....	21-31

### 22 Containers

Adding a Column Container .....	22-1
Adding a Tabbed Container.....	22-2

### 23 Data Record Displays

Results List.....	23-1
Sorting records in a Results List.....	23-2
Configuring a Results List.....	23-3
Results Table.....	23-7
About the Results Table layout .....	23-8
Configuring a Results Table.....	23-12
Using Results Table values to refine the data .....	23-30
Navigating to other URLs from a Results Table.....	23-30
Exporting data from a Results Table .....	23-30

## 24 Charts

Chart .....	24-1
Bar chart.....	24-4
Line / Area chart.....	24-6
Pie / Sunburst chart.....	24-7
Scatter chart.....	24-8
Bar-Line chart.....	24-13
Thematic Map .....	24-14
Configuring a Chart.....	24-18
Saving a snapshot of a chart .....	24-23
Histogram Plot .....	24-24
Selecting Histogram Plot data and display options.....	24-25
Box Plot.....	24-25
Selecting Box Plot data.....	24-27
Parallel Coordinates Plot .....	24-27
Refining Parallel Coordinates Plot data.....	24-29
Selecting Parallel Coordinates Plot data .....	24-30

## 25 Other Data Visualizations

Map .....	25-1
Configuring the Map connection to Oracle MapViewer.....	25-2
Using the Map.....	25-3
Configuring the Map .....	25-9
Pivot Table .....	25-18
Using the Pivot Table.....	25-18
Configuring the Pivot Table.....	25-21
Summarization Bar .....	25-27
Configuring Summarization Bar display options .....	25-29
Adding, renaming, and deleting summary items .....	25-29
Configuring a metric value summary item .....	25-31
Configuring a dimension value spotlight.....	25-32
Configuring metric and dimension value display and actions .....	25-34
Configuring conditional formatting .....	25-35
Configuring flag summary items.....	25-36
Tag Cloud.....	25-39
Configuring Tag Cloud display options.....	25-40
Selecting and configuring dimensions .....	25-41
Configuring term size and order.....	25-42
Configuring available metrics .....	25-43
Timeline.....	25-44
Adding and editing metric timelines .....	25-45
Setting the displayed date unit and date range for the metric timelines.....	25-47

Saving the current timeline configuration as the default .....	25-48
--	-------

## **26 Web-Based Content**

IFrame.....	26-1
Configuring an IFrame .....	26-2
Web Content Display .....	26-3
Configuring a Web Content Display .....	26-3

## **Index**



---

# Preface

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Apache Spark to turn raw data into business insight in minutes, without the need to learn specialist big data tools or rely only on highly skilled resources. The visual user interface empowers business analysts to find, explore, transform, blend and analyze big data, and then easily share results.

## About this guide

This guide contains information about using Oracle Big Data Discovery to explore, analyze, and transform data.

## Audience

This guide is written for business analysts and data scientists who want to explore, analyze, and transform data within the Hadoop ecosystem.

Oracle Big Data Discovery is designed for those who look for insights in data:

- Data Analyst
- Data Scientist
- Analytics Manager
- Problem Solver
- Casual Analyst

## Conventions

The following conventions are used in this document.

### Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
<b>User Interface Elements</b>	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
Code Sample	This formatting is used for sample code segments within a paragraph.

Typeface	Meaning
<i>Variable</i>	This formatting is used for variable values. For variables within a code sample, the formatting is <i>Variable</i> .
File Path	This formatting is used for file names and paths.

### Symbol conventions

The following table describes symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

### Path variable conventions

This table describes the path variable conventions used in this document.

Path variable	Meaning
\$ORACLE_HOME	Indicates the absolute path to your Oracle Middleware home directory, where BDD and WebLogic Server are installed.
\$BDD_HOME	Indicates the absolute path to your Oracle Big Data Discovery home directory, \$ORACLE_HOME/BDD- <i>&lt;version&gt;</i> .
\$DOMAIN_HOME	Indicates the absolute path to your WebLogic domain home directory. For example, if your domain is named <i>bdd- &lt;version&gt;_domain</i> , then \$DOMAIN_HOME is \$ORACLE_HOME/user_projects/domains/ <i>bdd- &lt;version&gt;_domain</i> .
\$DGRAPH_HOME	Indicates the absolute path to your Dgraph home directory, \$BDD_HOME/dgraph.

## Contacting Oracle Customer Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. This includes important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.

# Part I

---

## Overview of Studio





---

# Signing In and Managing Your Account

Studio users may have access through Single Sign-On integration, or they may have accounts specific to Studio.

## Signing in to Studio

Navigating to Studio results in automatic login when using Single Sign-On. If your Studio deployment does not use SSO authentication, you must sign in with either your Studio or LDAP credentials.

## Resetting your password

If you forget your Studio password, then you can have Studio reset it and email it to you. Studio cannot reset LDAP passwords.

## Changing your password

If you log in using a Studio account, you can use the **My Account** page to change your password.

## About locales

Studio is available in multiple locales. The locale determines the language that you see on the screen, as well as the default formatting for data values such as numbers and dates.

## Signing in to Studio

Navigating to Studio results in automatic login when using Single Sign-On. If your Studio deployment does not use SSO authentication, you must sign in with either your Studio or LDAP credentials.

---

**Note:** A Studio Administrator can create and manage users and groups directly, or they can configure access through an external user management system. For details about the two approaches to user accounts, including how to integrate with LDAP and single sign-on (SSO), see the *Administrator's Guide*.

---

To sign in to Studio:

1. Navigate to Studio.

If Single Sign-On authentication is enabled, you are automatically logged in to the application. Otherwise, the **Welcome** page appears.

2. Type your email address or username in the top field, as prompted.

A BDD deployment is configured to take one value or the other. Check the field label to verify that you have entered the required information.

3. In the **Password** field, type your Studio password.
4. Click **Sign In**.

## Resetting your password

If you forget your Studio password, then you can have Studio reset it and email it to you. Studio cannot reset LDAP passwords.

To reset your password:

1. Click the **Forgot your password?** link.

The **Forgot Your Password?** dialog displays.

### Forgot Your Password?

[Sign In](#)

Email Address:



Text Verification:

**Send New Password**

2. In the **Email Address** field, type your email address.  
This must be the address associated with your Studio account.
3. In the **Text Verification** field, type the Captcha displayed above the field.
4. Click **Send New Password**.

Studio resets your password and sends it to the provided email address.

## Changing your password

If you log in using a Studio account, you can use the **My Account** page to change your password.

---

---

**Note:** If you use Single Sign-On or LDAP authentication, you must reset your password within the respective system. You cannot reset an SSO or LDAP password from within Studio.

---

---

To change your Studio password:

1. In the right corner of Studio, click **User Options > My Account**.

The **My Account** page displays.

## My Account

\* Required

### User Details

---

Screen Name:* <input type="text" value="rwiggum"/>	Password: <input type="password"/>
Email Address:* <input type="text" value="ralph.wiggum@ssotest.com"/>	Retype Password: <input type="password"/>
First Name:* <input type="text" value="Ralph"/>	<b>Display Settings</b>
Middle Name: <input type="text"/>	Locale: <input style="border: 1px solid #ccc;" type="text" value="United States - English"/>
Last Name:* <input type="text" value="Wiggum"/>	Time Zone: <input style="border: 1px solid #ccc;" type="text" value="(UTC ) Coordinated Universal Time"/>

Role: Restricted User

---

▶ [INHERITED ROLES](#)

---

Cancel
Save

2. In the **Password** field, type the new password.
3. In the **Retype Password** field, type the new password again.
4. To save the new password, click **Save**.

Studio verifies that the values you typed matched and that the new password is valid.

## About locales

Studio is available in multiple locales. The locale determines the language that you see on the screen, as well as the default formatting for data values such as numbers and dates.

Studio is configured with a default locale, but you may choose to override it for your account.

Big Data Discovery supports the following locales:

- Chinese - Simplified

- English - UK
- English - US
- Japanese
- Korean
- Portuguese - Brazilian
- Spanish

Note that this is a subset of the languages supported by the Dgraph.

For more details on how Studio determines the locale and how to configure the available and default locales, see the *Administrator's Guide*.

#### [Configuring the locale for your account](#)

When an administrator first creates an account, the preferred locale is set to **Use Browser Locale**, which indicates to use the browser locale for Studio. If Studio doesn't support the browser locale, then Studio uses its default locale.

#### [Selecting a locale from the user menu](#)

The user menu includes an option to select a locale.

## Configuring the locale for your account

When an administrator first creates an account, the preferred locale is set to **Use Browser Locale**, which indicates to use the browser locale for Studio. If Studio doesn't support the browser locale, then Studio uses its default locale.

Restricted Users cannot access the **My Account** menu. If you are a Restricted User, see [Selecting a locale from the user menu](#).

To select a different preferred locale for your account:

1. From the user menu, select **My Account**.

The **My Account** page displays.

## My Account

\* Required

### User Details

Screen Name:\*  Password:

Email Address:\*  Retype Password:

First Name:\*  Display Settings

Middle Name:  Locale:

Last Name:\*  Time Zone:

Role: Restricted User

▶ INHERITED ROLES

2. Under **Display Settings**, from the **Locale** list, select your preferred locale.
3. To save the change, click **Save**.

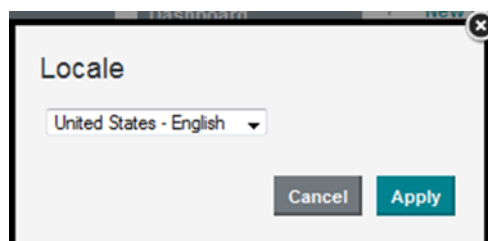
## Selecting a locale from the user menu

The user menu includes an option to select a locale.

If you change the locale from the user menu while signed in to Studio, your account updates accordingly.

To select the locale to use:

1. From the user menu, select **Change locale**.
2. On the change locale dialog, from the **Locale** list, select the locale.



Locale

**3. Click **Apply**.**

Studio displays in the selected locale. This includes:

- User interface labels
- Names of attributes displayed on components
- Values of attributes
- Formatting of data values

Values listed on the **Selected Refinements** panel are not affected.

If you are signed in, then your account also updates to use the locale as your preferred locale.

---

## Using the Studio Catalog

The Catalog is the main entry point for Studio. It provides access to all projects and data sets that you have permission to access. Clicking the logo in the header brings you back to the Catalog from any location in Studio.

### [About the Catalog](#)

The Catalog lets you browse, search, and filter across data sets and projects in Studio.

### [Browsing data sets and projects](#)

From the Catalog, you can sort displayed items and get additional information about each project and data set.

### [Filtering and searching data sets and projects](#)

You can view the full lists of data sets and projects on the Catalog. You can also use the **Refine By** panel to filter the lists, or use the search field to search for a project or data set.

### [Editing data set or project names, descriptions, and tags](#)

From the Catalog, you can rename a data set or project, add or edit the description, and add or remove searchable tags.

## About the Catalog

The Catalog lets you browse, search, and filter across data sets and projects in Studio.

Each project and data set is represented by a tile. Clicking directly on a project or data set name takes you to that project or data set, while clicking elsewhere on the tile opens a collapsible detail pane.

From the Catalog, you can:

- Search and filter data sets and projects. Once you locate a data set or project that you are interested in, you can browse its metadata before exploring the data set or project itself.
- Edit data set and project names, descriptions, and searchable tags.
- Create or delete data sets and projects for which you have administrator privileges.

### **The Projects tab**

Projects are sets of pages containing visualizations such as charts and tables. They are used to visualize data from one or more data sets.

The Catalog opens to the Projects tab by default. If you haven't searched or filtered the list, the **Projects** tab displays groups of:

- **Recently Viewed Projects** - Projects you have viewed recently

- **Most Popular Projects** - Projects that are viewed the most by all users

To display a list of projects you have access to, click **Projects**.

### The Data Sets tab

Data sets are collections of data that you can explore and add to projects.

If you haven't searched or filtered the list, the **Data Sets** tab displays groups of:

- **Recently Viewed Data Sets** - Data sets you've used recently
- **Most Popular Data Sets** - Data sets that are used the most by all users
- **Newly Added Data Sets** - Data sets that were added most recently

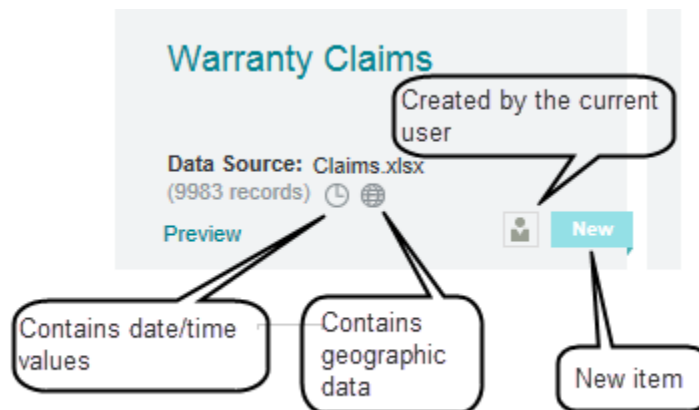
To display the list of data sets you have access to, click **Data Sets**. Restricted users do not have permission to view **Data Sets**.

## Browsing data sets and projects





From the Catalog, you can sort displayed items and get additional information about each project and data set.

### Icons used on the Catalog tiles

Each tile uses icons to provide additional information about the project or data set.



These icons include:

Icon	Definition
	Indicates an item that was created within the last week.
	Indicates an item that you created (the currently logged in user).
	Indicates a data set that includes date/time values.
	Indicates a data set that includes geocode values.



### Displaying a preview of a data set

The **Preview** link for each data set displays a small sample (the first 15 records and 50 attributes) of the data set.

### Displaying details for a data set or project

Clicking the tile displays a panel with additional details about an item, including:

- Description, if available
- Tags
- Types of attributes in a data set, or in data sets used in the selected project
- Who created it and when
- Available actions
- Number of views
- When it was last updated
- For projects:
  - Data sets used in the project
  - Pages in the project
  - Number of snapshots and bookmarks
- For data sets:
  - Data source
  - Origin system name and type
  - Data set permissions
  - Projects where the data set is used
  - Other data sets used in the same projects as this data set

You can use the paging icons at the left and right to page through the details panels for the previous or next data set or project.

In addition to accessing the details from the tile, you can also display a subset of the details by clicking the **Show Data Set Properties** icon in the **Explore** or **Transform** screens.

## Filtering and searching data sets and projects

You can view the full lists of data sets and projects on the Catalog. You can also use the **Refine By** panel to filter the lists, or use the search field to search for a project or data set.

### Displaying the full list of projects or data sets

From the Catalog, to display the full list of projects or data sets, click the **View all** link on the **Projects** or **Data Sets** tab, or the **View more** link for the project or data set groups.

## Filtering the Catalog lists

The **Refine By** panel provides options for filtering the Catalog lists.

For projects, many of the refinement options apply to included data sets. For example, when you refine the Catalog to only show items with geocode attributes, projects are included in the results if a project data set includes a geocode attribute.

The available refinements for the Catalog are grouped into the following categories:

Refine By Category	Description
<b>Projects</b>	Filter based on project details including the name, searchable tags, the creator, and the creation or modification dates.
<b>Data Sets</b>	Filter based on data set details or contents. In addition to the name, tags, creator, and creation/modification dates you can also filter by the number of records or attributes or by specific datetime or geocode information.
<b>Attributes</b>	Filter based on attributes within a data set. These include raw attribute names, standardized names, attribute tags, and semantic types.
<b>Data Sources</b>	Filter based on the types of source systems used to provide data. For example, you can limit results to data from Hadoop, or data uploaded from delimited files.

For details on using refinements for filtering, see [Filtering Based on Refinements](#).

## Searching the Catalog

You can use the search field to search for Catalog items. See [Searching in Studio](#).

## Sorting data sets or projects

If you have searched or filtered on the Catalog or clicked the **View All / View More** links, a **Sort** dropdown is present to the far right of the **Projects** and **Data Sets** tabs.

To sort the results, select the value to use for the sort:

- **Name** - The project or data set name
- **Created Date / Recently Added** - The project or data set creation date
- **Most Popular** - The number of unique visitors to a project or data set

To switch between ascending and descending order, click the sort order icon.

## Editing data set or project names, descriptions, and tags

From the Catalog, you can rename a data set or project, add or edit the description, and add or remove searchable tags.

Tags are words or phrases that provide meaningful information about a data set or project. You can add one or more tags to a data set or a project. Once you add a tag, you can then find data sets and projects that contain those tags using the Catalog search feature or by using the **Refine By** feature against either **Project Tag** or **Data Set Tag**.

To edit data set or project names, descriptions, and tags:

1. In the Catalog, locate the data set or project you want to modify.
2. Click the tile to open the details panel.
3. To rename a data set or project:
  - a. Click the title.
  - b. Modify the title text.
  - c. Press Enter.

The name updates to the new value.
4. To add a description:
  - a. Click the description field below the title.

The title opens to an editable text field.
  - b. Enter or modify the description.
  - c. Press Enter.

The description updates to the new value.
5. To add a tag:
  - a. Click the text field below the **Tags** heading.
  - b. Type the tag text.

Autocomplete may display suggested results in the tag field.
  - c. Press Enter to add the tag.

The tag is added to the tag list.
  - d. Optionally, continue to add tags.
6. To remove a tag, click its delete icon.



# Part II

---

## Searching and Filtering in Studio



---

## Searching in Studio

The Studio search feature provides quick access to data sets, projects, pages, and data from throughout Studio.

### [About Studio search](#)

You can access the search box from anywhere in Studio.

### [Displaying and using the results of a search](#)

To start a search, begin typing your search text into the search box. As you type, Studio displays matching search results.

### [About keyword searches](#)

When you complete a keyword search in a project, the data is refined to only include records with the matching search terms. In the Catalog, the list of projects and data sets are filtered to only include items with the matching search terms.

### [About Boolean searches](#)

For Boolean searches, Studio displays a **Boolean Search** panel when you include a logical operator in the query. Selecting the Boolean Search link from the panel runs the search based on the operators listed below.

### [Enabling or disabling full text search and typeahead search on an attribute](#)

Each String attribute can be individually enabled or disabled for full text and typeahead search. By default, String attributes that are longer than approximately 200 characters are enabled for full text search and disabled for typeahead search, while attributes shorter than 200 characters are enabled for typeahead search and disabled for full text search.

## About Studio search

You can access the search box from anywhere in Studio.

You can run a keyword search from:

- **The Catalog** - Search against project or data set details such as name, description, or searchable tags. Search matches that correspond to an attribute filtering value are highlighted in the **Refine By** sidebar.
- **Data Sets and Project Data Sets** - Search against String attribute values.

---

**Note:** The String attribute must be enabled for either full text search or typeahead search. If an attribute is not searchable in a data set, you must add the data set to a project in order to modify attribute metadata and enable search and/or typeahead on the attribute. For more information, see [Enabling or disabling full text search and typeahead search on an attribute](#).

---

- **Discover pages within a project** - Filter visualizations by searching against attribute names and values. The results panel also displays suggested filter refinements that match the keyword search.
- **Control Panel pages** - Search against pages in the Control Panel. For example, searching for "user" shows results for the **Users** and **User Groups** pages.

Results may also include "Did You Mean" suggestions, automatic spelling correction, and typeahead suggestions.

## Displaying and using the results of a search

To start a search, begin typing your search text into the search box. As you type, Studio displays matching search results.

The search results are divided into panels to reflect the different types of results.

For searches from the Catalog, the results are also divided into results from data sets, projects, project refinements (tags), and keyword search.

The screenshot shows the 'Big Data Discovery' search interface. At the top right, there is a search bar containing the text 'wine'. Below the search bar, the results are organized into three main columns:

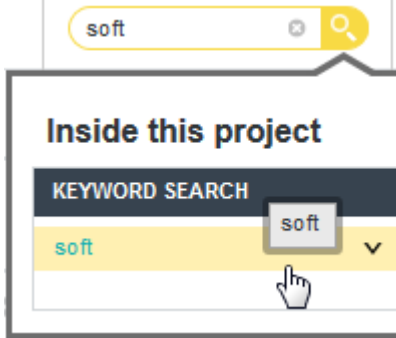
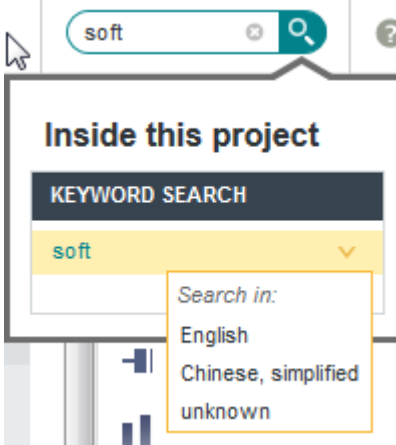
- DATA SETS REFINEMENTS:** This column is divided into two sections. The top section, 'Data Set', lists 'winedataset' and 'wine'. The bottom section, 'Data Source', lists 'default.wine'. The 'Original Source' section shows a file path: 'C:\Users\danizhan\Desktop\wine.csv' and 'wine.csv'.
- KEYWORD SEARCH:** This column displays the search term 'wine'.
- PROJECTS:** This column lists three projects: 'Bill Wine' (Created: 10/28/14, Author: bill okeefe), 'Wine2139' (Created: 10/28/14, Author: Admin Admin), and 'Henry' (Created: 10/28/14, Author: Admin Admin, Data sets: wine, winedataset).

Below the 'PROJECTS' column, there is a section titled 'DATA SETS' which lists 'wine' (Last modified: 2014-10-17T13:14:46.419Z) and 'winedataset' (Last modified: 2014-10-23T11:28:46.111Z).

Each section contains a different type of search result. Each type of result has an associated action you can perform from that result. The possible search results include:



Result Type	Description	Available Action
<b>Attributes</b>	<p>Included in searches from a data set or within a project.</p> <p>A list of attributes whose names match the entered search terms.</p> <p>Next to each attribute is the name of the attribute group that it belongs to.</p> <p>Attributes are grouped by data set.</p>	<p>When searching from a project page, to display the available values for refining by the attribute, click the attribute name.</p> <p>When searching from the Project Settings page, click the attribute name to display the <b>Views</b> page.</p> <p>The base view for the data set containing the attribute is selected.</p>
<b>Refinements</b>	<p>Only included in searches from a data set or project page.</p> <p>A list of attribute values, grouped by data set and attribute, that match the entered search terms.</p>	<p>To refine the data using a displayed value, click the value.</p> <p>You can also use Ctrl-click to add a value to the multi-select queue.</p>
<b>Projects</b>	<p>Included in all searches.</p> <p>A list of projects where the search term matches any of the following:</p> <ul style="list-style-type: none"> <li>• Project name or description</li> <li>• Page names</li> <li>• Data sets</li> <li>• Bookmarks</li> <li>• Snapshots</li> <li>• Tags</li> </ul> <p>The results only include projects that you have access to.</p>	<p>To navigate to the project, click the project name.</p> <p>You can also click individual matching items such as pages, bookmarks, or snapshots.</p>
<b>Pages</b> <b>Bookmarks</b> <b>Snapshots</b> <b>Galleries</b>	<p>Included in all searches.</p> <p>Results include matching pages, bookmarks, snapshots, or galleries for any project that you have access to. If searching from within a project, results are divided between those in the current project and those from other projects.</p>	<p>Select a result to navigate to it.</p> <p>Snapshot results link to snapshot details, while bookmark results link to the bookmarked page.</p>
<b>Project Settings</b>	<p>Only included in searches from a project page or from the Project Settings page, for project or Studio administrators.</p> <p>A list of pages in <b>Project Settings</b> whose names match the entered search terms.</p>	<p>To navigate to the page, click the page name.</p>
<b>Control Panel</b>	<p>Included in all searches run by Studio Administrators.</p> <p>A list of pages in the Control Panel whose names match the entered search terms.</p>	<p>To navigate to the page, click the page name.</p>
<b>Data Sets</b>	<p>Included in all searches.</p> <p>A list of available data sets with matches in their name or description.</p>	<p>To display the data set in <b>Explore</b>, click the data set name.</p>

Result Type	Description	Available Action
Keyword Search	<p>Included for searches from the Catalog, a data set, or a project page.</p> <p>Contains a link with the search text.</p> <p>For data sets and projects, the search is within the attribute values from the project data.</p> <p>On the Catalog, the search is within the information about the available data sets and projects.</p> <p>If you include Boolean syntax in the search text, then Studio also displays a <b>Boolean Search</b> section. Clicking the suggested search query enters it as a Boolean search.</p>	<p>To complete the text search, click the search term or press <b>Enter</b>. You can also click the search icon next to the search field.</p> <p>For project data, if the data is available in more than one locale, hovering the mouse over the keyword in the results displays a drop-down:</p>  <p>To select a locale for the search, click the drop-down icon and select the locale.</p>  <p>Once you select a locale, Studio uses it until you change it. If the current search locale is different from your locale, then Studio displays the current search locale next to the keyword.</p>

## About keyword searches

When you complete a keyword search in a project, the data is refined to only include records with the matching search terms. In the Catalog, the list of projects and data sets are filtered to only include items with the matching search terms.

---

**Note:** Keyword Search results may not display in all data sets. For results to be available, at least one attribute in the data must have String values that are larger than 200 characters on average, or else one or more attributes must be manually enabled for full text search through the Transform page of a project.

---

### Keyword searches

For non-Boolean searches, Studio looks for items that have all of the terms. If it finds any, it stops looking and returns those records.

If none of the items have all of the search terms, then records are returned based on the partial search rules:

- The item must contain at least one of the search terms.
- The item cannot be missing more than two of the search terms.

For example, if you type `California red berry sweet`, then:

- An item containing only one term would not be a match.
- An item containing two or more terms would be a match.

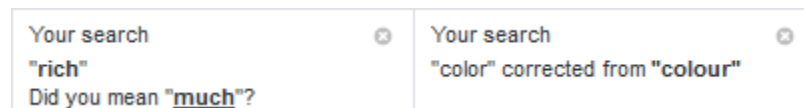
To use other logical operators such as AND and OR, see [About Boolean searches](#).

### Wildcard searches

As you type a keyword in the search box, Studio implicitly adds a trailing wildcard (\*) after each character and performs character-by-character matching as you type. Adding a wildcard character to the search syntax has no additional effect.

### Search refinements

When you complete a text search, the refinement appears in the **Selected Refinements** panel. "Did You Mean" and spelling-corrected results also appear in the **Selected Refinements** panel:



For more information on how refinements display, see [About the Selected Refinements panel](#).

### Displaying snippets for search terms

On the **Results List** and **Results Table** components, the search terms may be highlighted. For attributes that support snippeting, the search snippet displays. The snippet displays the portion of the attribute value that contains the search terms.

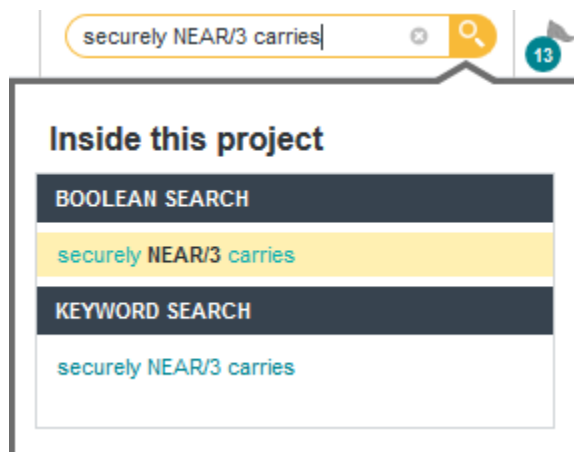
### Effect of relevance ranking

For non-Boolean searches, relevance ranking can affect the display order of results on the following components:

Component	Effect of Relevance Ranking
Results List	If there is no default sort order for the <b>Results List</b> , and you haven't applied a sort to the list, then when you submit a search, the results sort based on search relevance. If a specific sort order has been applied to the table, then the relevance ranking settings are not applied.
Results Table	If there is no default sort order for the <b>Results Table</b> , and you haven't applied a sort to the table, then when you submit a search, the results sort based on search relevance. If a specific sort order has been applied to the table, then the relevance ranking settings are not applied.
Map	When you submit a the search, the results automatically sort based on search relevance. A <b>Search Relevance</b> option is added to the <b>Sorted by</b> list. You can then switch between the <b>Search Relevance</b> option and the other available sorting options.

## About Boolean searches

For Boolean searches, Studio displays a **Boolean Search** panel when you include a logical operator in the query. Selecting the Boolean Search link from the panel runs the search based on the operators listed below.




---

**Note:** Boolean Search results may not display in all data sets. For results to be available, at least one attribute in the data must have String values that are larger than 200 characters on average, or else one or more attributes must be manually enabled for full text search through the Transform page of a project.

---

Boolean Search Operator	Purpose	Example Usage and Results
AND	Returns results with all specified terms.	<p>fruit AND berry</p> <p>Returns results with both "fruit" and "berry" such as:</p> <p>This wine has notes of fruit, cinnamon, berry, and oak.</p>
OR	Returns results with any specified terms.	<p>"best buy" OR cheap</p> <p>Returns results with either "best buy" or "cheap" such as:</p> <p>The product quality is cheap.</p>
NOT	Negates the following term. Carries an implicit "AND" when paired with other terms, which is overridden if you specify "OR NOT"	<p>"best buy" NOT cheap</p> <p>Returns results with "best buy" but not "cheap" such as:</p> <p>At this price, this is a best buy.</p> <p>But not:</p> <p>The price is cheap for what you get, a definite best buy.</p>
NEAR/ <NUM>	Returns results where the right side term is within <NUM> words of the left side term. Terms can be a single word or multi-word phrases, but cannot include wildcards or Boolean operators.	<p>blue NEAR/4 red</p> <p>Returns results where "blue" is within 4 words (inclusive) of "red", such as:</p> <p>red orange yellow green blue indigo violet</p>
ONEAR/ <NUM>	Functions identically to NEAR/<NUM>, except that terms must be in the specified order.	<p>red ONEAR/2 blue</p> <p>Returns results where "blue" occurs within 2 words AFTER "red", such as:</p> <p>red white blue</p>

### Operator interaction and precedence

By default, a keyword search runs as an AND search that uses all entered terms. You can use Boolean operators to set more precise search logic. For example, the query `camcorder AND NOT digital` searches for records with the word "camcorder" and then removes any results with the word "digital" before returning the set of matches.

Operator precedence is determined in the following order:

1. Any sub-expressions in parentheses are evaluated first
2. NOT is evaluated before other operators
3. AND is evaluated after NOT
4. OR is evaluated after AND

For example, the expression "A OR B AND C NOT D" is interpreted as "A OR (B AND C AND (NOT D))".

## Enabling or disabling full text search and typeahead search on an attribute

Each String attribute can be individually enabled or disabled for full text and typeahead search. By default, String attributes that are longer than approximately 200 characters are enabled for full text search and disabled for typeahead search, while attributes shorter than 200 characters are enabled for typeahead search and disabled for full text search.

---

---

**Note:** You can also use the "--disableSearch" command of the Data Processing CLI to globally disable search for all string attributes. This command disables both full text search and value search. This global command can be useful in cases where you have a large number of attributes in a project data set, and it is easier to disable all of them using the Data Processing CLI and then enable just a couple of them using Studio. For additional information, see "Record and value search settings for string attributes" in the *Data Processing Guide*.

---

---

To enable or disable full text search and typeahead search for an attribute:

1. In your project, select **Transform**.
2. Locate the String attribute that has text you want to be available for full text search.
3. Mouse over the attribute column header.

A menu button appears.

4. Click the menu and select **Edit Details**.

The Quick Look dialog opens.

5. To enable or disable typeahead search, select the corresponding **Typeahead search** option.

This adds a transformation step to the transform script. If you wish to undo your selection before committing, remove the step from the script.

6. To enable or disable full text search, select the corresponding **Full text search** option.

This adds a transformation step to the transform script. If you wish to undo your selection before committing, remove the step from the script.

7. Click **Done**.
8. To apply changes immediately, expand the **Transform Script** pane and click **Commit to Project**.

After the script completes, the attribute is enabled or disabled for full text search and typeahead search.





---

## Filtering Based on Refinements

Throughout Studio, you can use the refinements panel to refine projects, data sets, or data.

### About using refinements for filtering

The **Refine By** panel lets you filter the currently displayed items and data.

### Selecting refinement values for filtering

You can select one or more values on the **Refine By** for filtering. Selected refinements appear in the **Selected Refinements** panel.

### Selecting a range of refinement values

For some attributes, you can refine by a range of values instead of choosing specific values.

### Selecting a recurring date/time

Date/time attributes allow you to refine based on a recurring value. You can filter results to show only records that recur at a certain time.

### About implicit refinements

A refinement to one attribute can affect the available refinements for other attributes. This is referred to as an implicit refinement.

### Managing Selected Refinements

The **Selected Refinements** panel displays all refinements currently applied to the Catalog or to data sets within a project.

### About the Selected Refinements panel

The **Selected Refinements** panel displays all of your currently selected refinements and enables you to modify and remove them. It appears at the top of the page whenever you apply a refinement to the Catalog or to a data set within a project.

### Removing refinements from the Selected Refinements panel

You can use the **Selected Refinements** panel to remove refinements and individual refinement values. You can also use it to remove all selected refinements at once.

## About using refinements for filtering

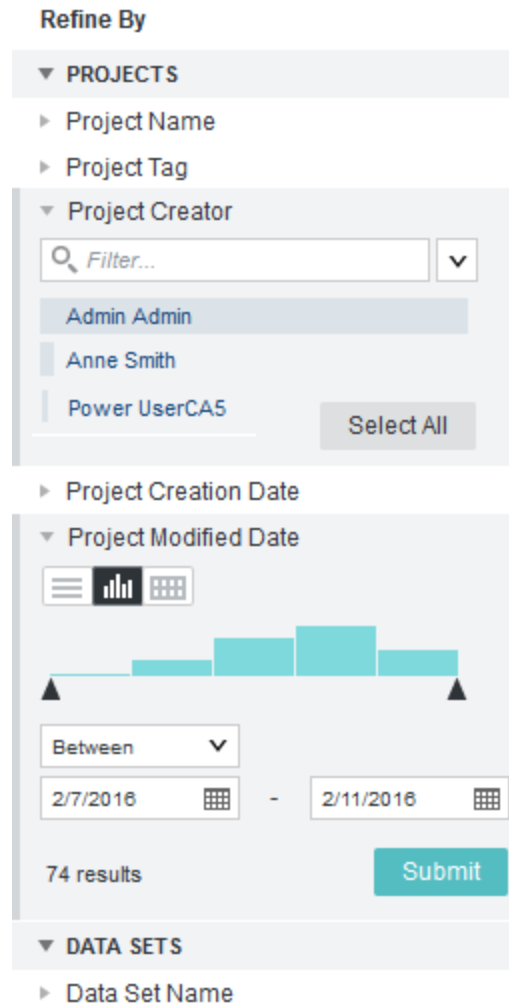
The **Refine By** panel lets you filter the currently displayed items and data.

The **Refine By** panel is collapsed by default.

On the Catalog, the **Refine By** panel allows you to filter the lists of available data sets and projects based on characteristics such as data types, creator, and searchable tags.

On the **Explore** and **Transform** pages, the **Refine By** panel allows you to filter data based on attribute values. You display the panel by clicking the Filter icon on the left side of the screen.

Refinements display in collapsible groups. For each refinement, you can also show and hide all available values.



Depending on the refinement, you can either select a specific value or values, or specify a range of values.

## Selecting refinement values for filtering

You can select one or more values on the **Refine By** for filtering. Selected refinements appear in the **Selected Refinements** panel.

To select refinement values for filtering:

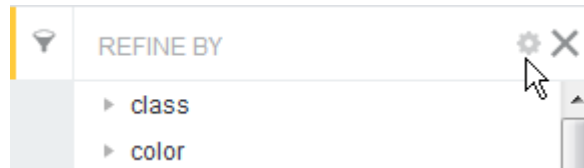
1. In the **Refine By** panel, expand the attribute you want to use for filtering.
2. To display additional attribute values, click **More**

Studio initially shows the first 20 values. Clicking **More** displays the first 100 values, and each additional click displays another 100 values up to a maximum of 2000. To display only the first 20 values again, click **Fewer**.

By default, values display in descending order based on the number of matches for each value. The shaded bar indicates the relative number of matches.

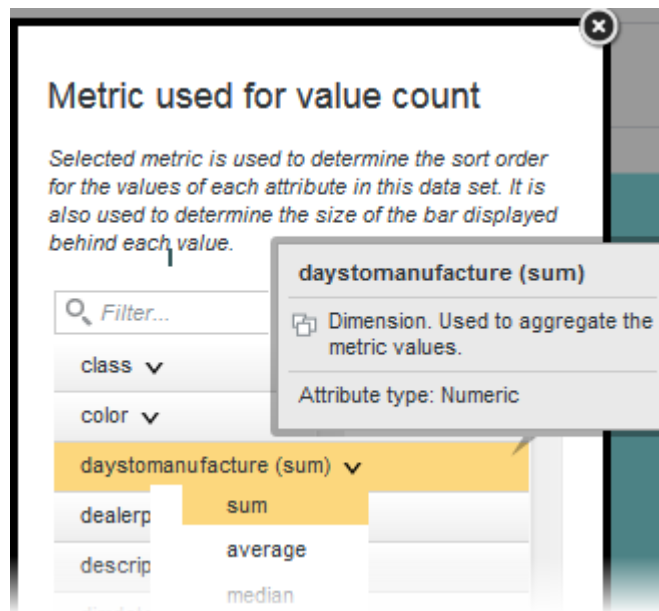
To search across attribute values, use the Search field. To change the sort order, click the drop-down arrow next to the Search field.

3. When filtering from the **Explore** or **Transform** pages, you can optionally change the metric used to sort attribute values:
  - a. Mouse over the **Refine By** header and click the Configuration icon:



The **Metric used for value count** dialog appears.

- b. Click the attribute you want to modify and select the aggregation menu from the drop-down next to the attribute:



- c. Click **Apply**.

The sort list updates to reflect the currently selected metric value.

4. If you are selecting a value for a date/time attribute, use the drop-down to select the desired **Year**, **Year-Month**, or **Year-Month-Day** combination.
5. To select a single refinement value, click the value in the list.

The data is filtered to only display records with that value. If the attribute does not allow you to select multiple values or if there are no matching records for any other values, the attribute is disabled on the **Refine By** panel.

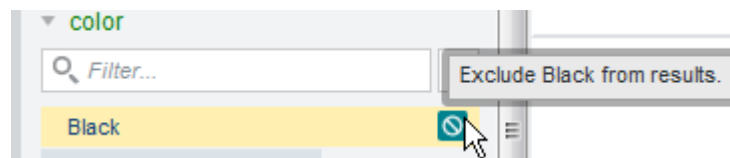
- To select multiple refinement values, Ctrl-click each value and click **Apply refinements** in the multi-select panel.

Optionally, click the **Select All** button to select all values.

Depending on the refinement, the data is filtered to include records that either:

- Have any one or more of the selected values
- Have all of the selected values

- To exclude a single refinement value, mouse over the value and click the Exclude icon:



The data is filtered to only display records without that value.

- To exclude multiple refinement values, Ctrl-click the exclude icon for each value and click **Apply refinements** in the multi-select panel.

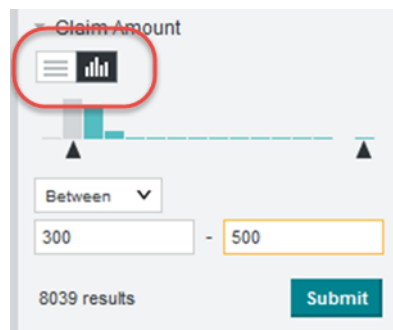
The data is filtered to include records that have none of the selected values.

## Selecting a range of refinement values

For some attributes, you can refine by a range of values instead of choosing specific values.

By default, range selections are enabled for numeric, date, and time values and display the current minimum and maximum values for the selected attribute. Range filters display a histogram which shows how the values are distributed within the range. Hover over a histogram bar to see the range of values for that bar and the number of matching records in that range.

If you want to choose a specific value instead of a range, use the value list toggle icon:



For information on selecting refinement values from a list, see [Selecting refinement values for filtering](#).

Date/time values also offer a recurring range selection option. See [Selecting a recurring date/time](#).

To select a range of refinement values:

1. In the **Refine By** panel, expand the attribute you want to use for filtering.
2. In the range drop-down, select the range type:
  - **Between** two values or dates/times (the default)
  - **Less than** a value or **Before** a date/time
  - **Greater than** a value or **After** a date/time
3. Specify the necessary values by entering them in the fields (or by using the date picker for dates).

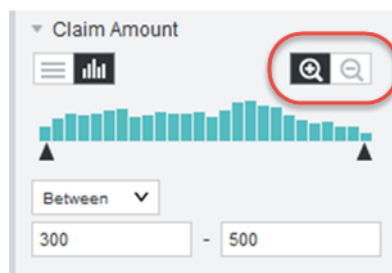
Alternately, drag the black range selection arrows to select value ranges from the histogram.

4. Click **Submit**.

The selected range is added to the **Selected Refinements** panel.

If you submit a different range value for the same attribute, the existing refinement is replaced by the new refinement.

5. After you submit a range filter refinement, you can use the range type toggle to change the displayed range on the histogram. This does not affect the selected refinement.



You can either display:

- The full range of available values (limited by any other selected refinements).
- The current range of available values based on refinements to the attribute.

## Selecting a recurring date/time

Date/time attributes allow you to refine based on a recurring value. You can filter results to show only records that recur at a certain time.

You can select a value for:

- Month
- Day of the month

- Hour of the day
- Minute of the hour
- Second of the minute

---

---

**Note:** Available date/time attribute values are determined by attribute configuration in the data set.

---

---

For example, you could include:

- Dates in May of any year
- Dates on the third of any month
- Date values with time values during the noon hour

You can also select values from different units. For example, you could refine the data to only include dates on May 6 of any year.

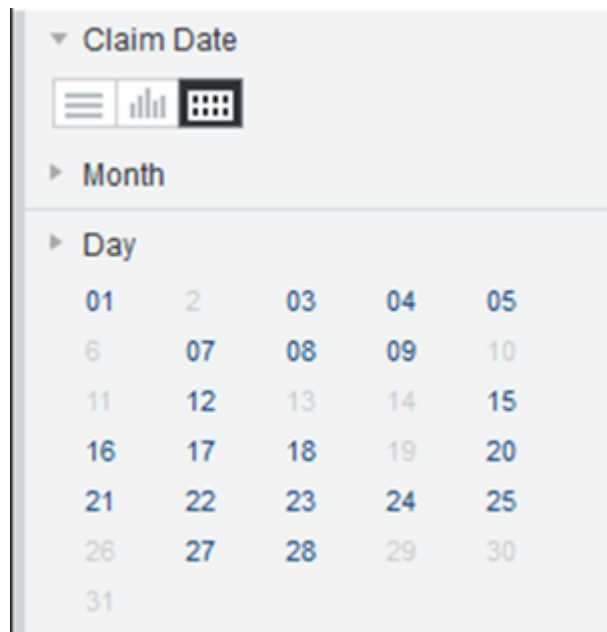
To select a recurring date/time:

1. In the **Refine By** panel, expand the attribute you want to use for filtering.
2. Click the recurring date button:



If no date/time values recur for the selected attribute, a "No recurring date/time refinements available" message displays.

3. If recurring values are available, click the drop-down for the unit you wish to use:



4. Select a value.
5. Click **Submit**.

The selected recurring dates/times are added to the **Selected Refinements** panel.

If you submit a new set of recurring dates/times for the same attribute, the existing refinement is replaced by the new refinement.

## About implicit refinements

A refinement to one attribute can affect the available refinements for other attributes. This is referred to as an implicit refinement.

For example, if vehicle sales data includes both the make and the model, then when users refine by the vehicle model Mustang, which is produced by Ford, then the make attribute is implicitly refined by Ford.

If the data also includes the sales month, but no Mustangs were sold in January, then January is removed as an available value for the sales month.

Enabling filtering between linked data sets also leads to implicit refinements.

Implicit refinements are not displayed on the **Selected Refinements** panel. However, if there are no available refinements for an attribute because of implicit refinements, the attribute is disabled, and Studio displays a message explaining that the attribute is not available because of implicit refinements.

## Managing Selected Refinements




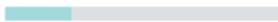
The **Selected Refinements** panel displays all refinements currently applied to the Catalog or to data sets within a project.

## About the Selected Refinements panel

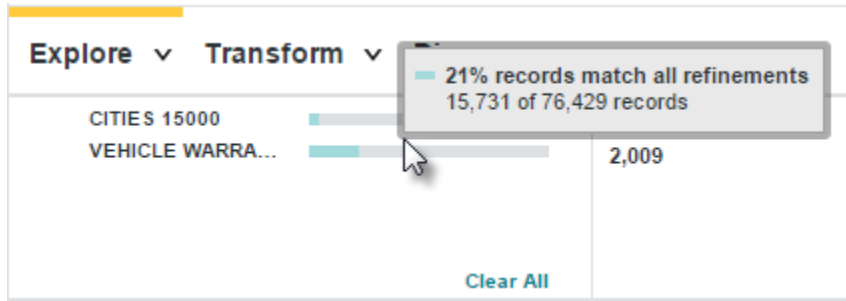
The **Selected Refinements** panel displays all of your currently selected refinements and enables you to modify and remove them. It appears at the top of the page whenever you apply a refinement to the Catalog or to a data set within a project.

### The data set list

Within a project, the left side of the **Selected Refinements** panel lists the data sets that are currently being filtered.

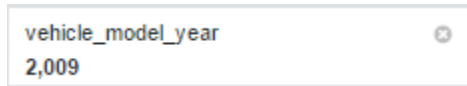
CITIES 15000 	vehicle_model_year 	vehicle_make 	a
VEHICLE WARRA... 	2,009	CHEVROLET FORD DODGE JEEP	V S F C
<a href="#">Clear All</a>			

Next to the name of each data set is a bar that indicates the percentage of its records that match the refinements. You can hover your mouse over this bar to view the data set's details, which include the exact percentage and number records that match.

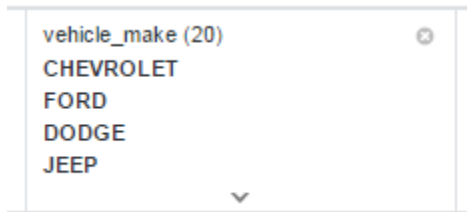


### Refinement tiles

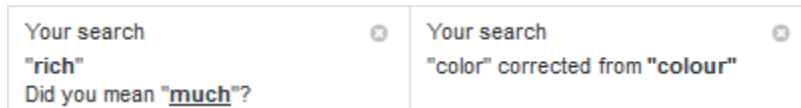
The **Selected Refinements** panel displays each refinement as a tile that contains the name of the refinement and its value.



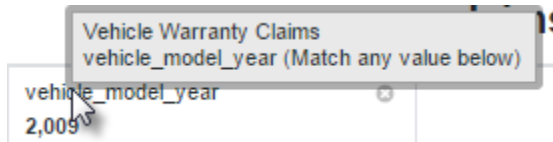
If the refinement has multiple values, the tile displays the full list in the order you selected them. If there are more selected values than the tile can display at once, you can click the arrow at the bottom of the tile to view the entire list.



Refinements that include "Did You Mean" suggestions or automatic spelling corrected are displayed differently to indicate the suggestion or correction:



You can hover over a refinement's name to view what it applies to. In the Catalog, this shows whether it applies to a project or a data set; within a project, it shows the name of the data set it applies to.

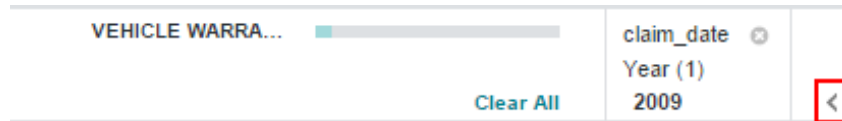


If the **Selected Refinements** panel contains too many tiles to display at once, you can use the arrows to the right and left of the panel to scroll through them.

### Hiding the Selected Refinements panel

Within a project, you can collapse the **Selected Refinements** panel so that only the list of data sets displays.

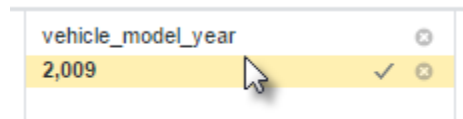




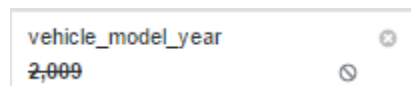
### Negative refinements

By default, all refinement values are positive, meaning the data is filtered to include records that match them. You can reverse this and filter for records that *don't* match a refinement by making it negative.

To make a positive refinement value negative, hover your mouse over its name and select the check mark.

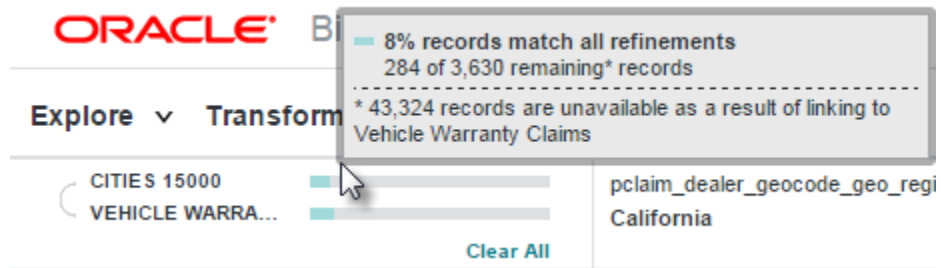


The checkmark turns into a slashed circle and the value is crossed out. If there are multiple values selected, the negative value will be moved to the bottom of the list.



### Refinements in linked data sets

When data sets in a project are linked, the data set list on the **Discover** page displays a line connecting them and each data set's details specify the number of its records that are unavailable because of the linking.



Refinements applied to one of the data sets are automatically applied to the other.

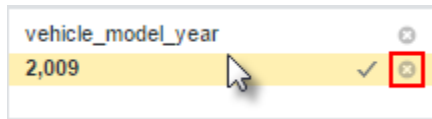
## Removing refinements from the Selected Refinements panel

You can use the **Selected Refinements** panel to remove refinements and individual refinement values. You can also use it to remove all selected refinements at once.

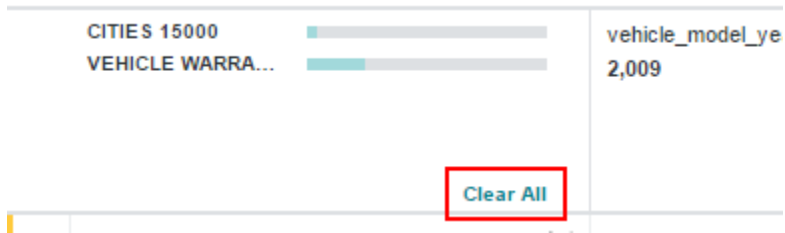
To remove a single refinement, click the delete icon in its tile.



To remove a single refinement value, hover your mouse over the value's name and click the delete icon. To remove all values, click the delete icon for the refinement.



To remove all currently selected refinements, click **Clear All** in the leftmost tile of the **Refinements Panel**.



# Part III

---

## Managing Data Sets



---

## Managing a Data Set

You can create new data sets, update them, and remove them from the Catalog. Depending on how a data set is created, you can then assign access to it in order to expose it to Studio users.

### Options for loading data into Studio

You can load data into Studio by either creating a new data set or updating a data set.

### About creating a data set in the Catalog

You can create a new data set in the Catalog by uploading personal data files into Studio and also by importing and filtering a JDBC data source into Studio.

### About updating a data set in the Catalog

In Studio, you can update a data set that has a source type of Excel, delimited file, or JDBC by using **Reload Data set**. A reload operation updates a data set to include all records and attributes that have been added, modified, or deleted since the last reload or since the time the data set was created.

### Setting attribute metadata in data sets

You can set metadata properties on attributes to make a data set easier to search and navigate. Metadata properties include display names, attribute types, semantic types, tags, and notes. On added to a data set, these metadata values become searchable in the search box of Studio, and the metadata values display as new navigation options in the Catalog's **Refine By** menu.

### About notifications for data set changes

Studio provides a **Notifications** panel for you to view notifications about the progress of any data set changes you initiated. Notifications can include information about creating new data sets, updates to data sets including refreshes and full data loads, transformations, exports, deletions, and so on.

### Managing access to a data set

Users must have access to a data set in order to see it in the Catalog or search results, explore the data, or add the data set to a project.

### Deleting a data set

If you have the role of User, Power User, or Administrator, you can delete any data set from the Catalog. If you have the role of Viewer, you cannot delete a data set.

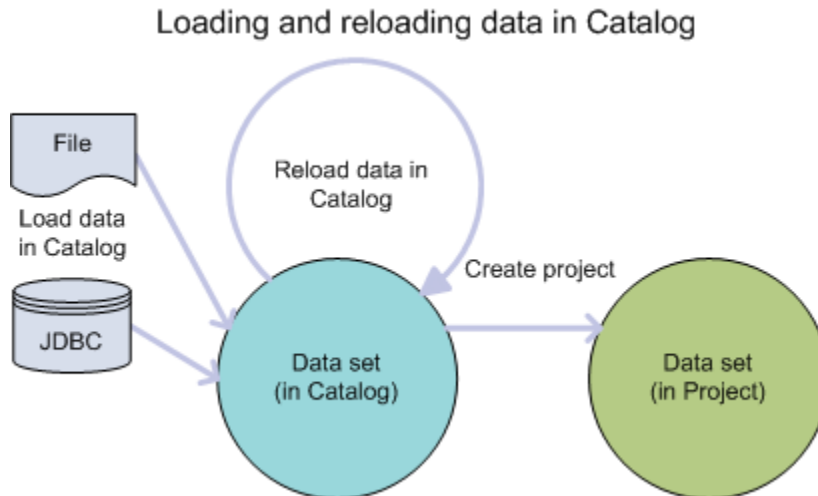
## Options for loading data into Studio

You can load data into Studio by either creating a new data set or updating a data set.

There are four data loading operations:

- Loading a file to create a new data set.
- Loading a JDBC data source to create a new data set.
- Reloading a file to update the data set.
- Reloading a JDBC data source to update the data set.

Here is the typical workflow of loading a data set and reloading it in the Catalog before you add it to a project:



In this workflow, the following actions take place:

1. You load a data set from a file or JDBC data source. This is the initial load of the data set into the Catalog.
2. Reload the data set as necessary.
3. Explore the data and add it to a project if you want to use Transform and Discover. If you transform or fully load a data set in a project, the data set is now private to the project.

## About creating a data set in the Catalog

You can create a new data set in the Catalog by uploading personal data files into Studio and also by importing and filtering a JDBC data source into Studio.

### Permissions

A Studio user must have the role of User, Power User, or Administrator to load data from a file. Creating a data set from a JDBC data source additionally requires you to provide the user name and password of the person who has database credentials.

### Maximum sample size for file upload

By default, any file you upload is processed to result in a maximum sample size of up to 1,000,000 records. Files that contain more than 1,000,000 records are down sampled to approximately 1,000,000 records. Files with less than 1,000,000 records contain the full record set in the resulting data set.

If necessary, you can increase the default setting from 1,000,000 records to another value. For details, see the `bdd.sampleSize` documentation in the *Administrator's Guide*.

### Potential upload timeouts

If a file upload fails due to a connection timeout, the file may be too large to upload from Studio. You can work around this issue by asking your Hive database administrator to import the source file into a Hive table, and then you run the Data Processing CLI utility to process the table. After data processing, a new data set, based on the file, is available in the Catalog.

### Duplicate columns and multi-value attributes

Note that if your personal data file has columns with identical headings, then in the new data set, the columns are converted into a single multi-value attribute. For example, if your data includes:

Item	Color	Color	Color
T-Shirt	Red	Blue	Green
Sweatshirt	Red	White	

Then in the final data set, the result is:

Item	Color
T-Shirt	Red, Blue, Green
Sweatshirt	Red, White

### Anti-Virus and Malware

Studio can load Excel spreadsheets and delimited files. Oracle strongly encourages you to use anti-virus products prior to uploading files into Studio. Studio converts these files into the Hadoop Avro format, uploads the data to HDFS, and then registers a Hive table for the data. The original file is then discarded.

#### [Creating a data set from a file](#)

You can create a new data set in Studio by uploading personal data from files. After upload, the data is available as a data set in the Catalog.

#### [Creating a data set from a JDBC data source](#)

If a Studio administrator has already created a data connection and added a JDBC data source, then you can import and filter a JDBC data source into Studio. After import, the data source is available as a data set in the Catalog. For information about creating data sources, see the *Administrator's Guide*.

## Creating a data set from a file

You can create a new data set in Studio by uploading personal data from files. After upload, the data is available as a data set in the Catalog.

Studio supports Microsoft Excel files, delimited files such as CSV, TSV, and TXT, and also compressed file such as ZIP, GZ and GZIP. A compressed file may include only one delimited file.

Microsoft Excel files must have a suffix of XLS or XLSX and may have been created using any of the following versions:

- Excel 2016
- Excel 2013
- Excel 2010
- Excel 2007
- Excel 97 - 2003

To create a data set from a file:

1. Click the **Add Data Set** option on the Catalog.

This option adds the new data set to the Catalog. You can also add a new data set from within a project.

2. Click **Create a data set from a file**.
3. Click **Browse**, locate the file, and click **Open**.
4. Click **Next**.
5. In the **Preview** page, you can both edit attributes and limit the data before you upload it.
  - a. To exclude an attribute from the data set, deselect its check box.
  - b. To modify the name of an attribute as it appears in the data set, select the column header and edit the name of the attribute.
6. Expand **Basic Settings** and set the following options:

Setting	Description
<b>My data includes header row</b>	Specify if the file includes a header row. If you deselect this option, Studio creates an alphabetized list in place of attribute names.
<b>Skip the first 0 rows</b>	If necessary, you can indicate how many rows to skip from the top of the file.
<b>Sheet</b>	<i>Microsoft Excel files only.</i> Select which sheet to load from the list. A data set corresponds to one sheet. Run the wizard again if you need to process multiple sheets.
<b>Fields are delimited by</b>	<i>Delimited file formats only.</i> Select a field delimiter from the list. This selection often corresponds to the file format, comma separated (CSV), tab separated (TSV/TAB), etc.
<b>Fields are quoted</b>	<i>Delimited file formats only.</i> Specify whether the fields contain quoted values.
<b>Quote Character</b>	<i>Delimited file formats only.</i> If you enabled <b>Fields are quoted</b> , select the quote character.



Setting	Description
Quote escape character	<i>Delimited file formats only.</i> If you enabled <b>Fields are quoted</b> , select the quote escape character.

7. Expand **Advanced Settings** and set the following options:

Setting	Description
Character Encoding	<i>Delimited file formats only.</i> Select the file encoding. If you are unsure, you may have to open the file in a full featured text editor and use the editor to detect the encoding.
Language	Specify the language of text data in your file. This setting is used during data processing and then used for value and keyword searches.

8. Click **Next**.

9. On the **Create your data set** page:

- a. Specify a name for the data set as it appears in the Catalog.
- b. Optionally, specify a description for the data set.
- c. Optionally, specify a Hive table name. By default, the Hive table name is the same as the data set name. If you create a data set by the same name as an existing data set, you must specify a different Hive table name. Studio maps the data set name to a unique table name.

10. Click **Create** and then click **Add Another Data Set** if you have more files to upload or click **Return to Catalog** while the file is being processed.

A new data set based on the file is available in the Catalog.

If the file upload fails due to a connection timeout, the file may be too large to upload from Studio. You can work around this issue by asking your Hive database administrator to import the source file into a Hive table, and then you run the Data Processing CLI utility to process the table. After data processing, a new data set, based on the file, is available in the Catalog .

## Creating a data set from a JDBC data source

If a Studio administrator has already created a data connection and added a JDBC data source, then you can import and filter a JDBC data source into Studio. After import, the data source is available as a data set in the Catalog. For information about creating data sources, see the *Administrator's Guide*.

The Studio user must have the role of User, Power User, or Administrator to import JDBC data.

To load data from a JDBC data source:

1. Click the **Add Data Set** option on the Catalog.

This option adds the new data set to the Catalog. You can also add a new data set from within a project.

2. Click **Create a data set from a database**.
3. On the **Select data source** page, select a row corresponding to the data source you want to import and click **Next**.
4. Provide the user name and password of the person who has database credentials to access the data and click **Continue**.
5. In the **Preview & filter data** page, you can both edit attributes and limit the data before you upload it:
  - a. To exclude an attribute from the data set, deselect its check box.
  - b. To modify the name of an attribute as it appears in the data set, select the column header and edit the name of the attribute.
  - c. To filter an attribute by an attribute value, select the funnel icon in an attribute header. (This adds a filter to the **Filter By** pane.) And then select a sample value that you want to filter by. For example, if you have an attribute named `Country_Name`, you can select the filter icon and then select `United States`. That filters the records down to the set of records where `Country_Name` matches `United States`.
  - d. If you know the language of text data in your attributes, select the source language from **Default search language**. (This setting is used during data process and then used for value and keyword searches.)
6. Click **Next**.
7. On the **Select data source** page:
  - a. Specify a name for the data set as it appears in the Catalog.
  - b. Optionally, specify a description for the data set.
  - c. Optionally, specify a Hive table name. By default, the Hive table name is the same as the data set name. If you create a data set by the same name as an existing data set, you must specify a different Hive table name. Studio maps the data set name to a unique table name.
8. Click **Create**.

A new data set, based on the data source, is available in the Catalog.

## About updating a data set in the Catalog

In Studio, you can update a data set that has a source type of Excel, delimited file, or JDBC by using **Reload Data set**. A reload operation updates a data set to include all records and attributes that have been added, modified, or deleted since the last reload or since the time the data set was created.

Studio does not automatically reload a data set after you add it to a project. Once a data set has been transformed or loaded within a project it becomes a private copy.

After you reload a data set in the Catalog, Studio automatically reloads the data set into a project if the project is using the public copy.

However, once a data set has been transformed in a project it becomes a copy that is specific to that project.

If the public version of this data set in Catalog is reloaded (that is, a newer version of this public data set appears in Catalog), then the next time you enter the project, Studio notices that you are using a private copy of a recently updated public version of this data set, and prompts you to "Accept data updates" by re-committing the transform script to the new version of the data set. This commit creates a new private copy of the reloaded public data set. The transform script is applied to that private copy.

---

---

**Note:** To update data sets that were created directly in Hive, use the Data Processing CLI. For details, see the *Data Processing Guide*.

---

---

#### Updating a data set from a file

The **Reload Data set** operation updates a data set from a file and makes the updated data set available in the Catalog.

#### Updating a data set from a JDBC data source

The **Reload Data set** operation reloads a data set from a JDBC data source and makes the updated data set available in the Catalog.

### Updating a data set from a file

The **Reload Data set** operation updates a data set from a file and makes the updated data set available in the Catalog.

To update a data set from a file:

1. In the Catalog, locate the data set you want to update.
2. Open the quick look for the data set, and from the **Actions** menu, select **Reload Data set**.

This starts the personal data upload wizard.

3. Click **Browse**, locate the file, and click **Open**.
4. Click **Next**.
5. In the **Preview** page, you can both edit attributes and limit the data before you upload it:
  - a. To exclude an attribute from the data set, deselect its check box.
  - b. To modify the name of an attribute as it appears in the data set, select the column header and edit the name of the attribute.
  - c. To specify that the file includes a header row, select **My data includes header row**. If you deselect this option, Studio creates an alphabetized list in place of attribute names.
  - d. If necessary, you can indicate how many rows to skip from the top of the file.
  - e. If the file is an MS Excel file, you can select which sheet to load.
  - f. If you know the language of text data in your attributes, expand **Advanced Settings** and select the source language from **Language**. (This setting is used during data processing and then used for value and keyword searches.)

6. Click **Next**.
7. Click **Reload**.

After processing, an updated data set is available in the Catalog.

## Updating a data set from a JDBC data source

The **Reload Data set** operation reloads a data set from a JDBC data source and makes the updated data set available in the Catalog.

To update a data set from a JDBC data source:

1. In the Catalog, locate the data set you want to update.
2. Open the quick look for the data set, and from the **Actions** menu, select **Reload Data set**.  
  
This starts the personal data upload wizard.
3. Provide the user name and password of the person who has database credentials to access the data and click **Continue**.
4. In the **Preview & filter data** page, you can both edit attributes and limit the data before you upload it:
  - a. To exclude an attribute from the data set, deselect its check box.
  - b. To modify the name of an attribute as it appears in the data set, select the column header and edit the name of the attribute.
  - c. To filter an attribute by an attribute value, select the funnel icon in an attribute header. (This adds a filter to the **Filter By** pane.) And then select a sample value that you want to filter by. For example, if you have an attribute named `Country_Name`, you can select the filter icon and then select `United States`. That filters the records down to the set of records where `Country_Name` matches `United States`.
  - d. If you know the language of text data in your attributes, select the source language from **Default search language**. (This setting is used during data processing and then used for value and keyword searches.)
5. Click **Next**.
6. On the **Select data source** page:
  - a. Specify a name for the data set as it appears in the Catalog.
  - b. Optionally, specify a description for the data set.
  - c. Optionally, specify a Hive table name. By default, the Hive table name is the same as the data set name. If you create a data set by the same name as an existing data set, you must specify a different Hive table name. Studio maps the data set name to a unique table name.
7. Click **Create**.
8. Click **Reload**.

## Setting attribute metadata in data sets

You can set metadata properties on attributes to make a data set easier to search and navigate. Metadata properties include display names, attribute types, semantic types, tags, and notes. On added to a data set, these metadata values become searchable in the search box of Studio, and the metadata values display as new navigation options in the Catalog's **Refine By** menu.

For example, if you add a semantic type named `Person` to an attribute, you see a new navigation option in **Catalog > Refine by > Attributes > Semantic type for Person**.

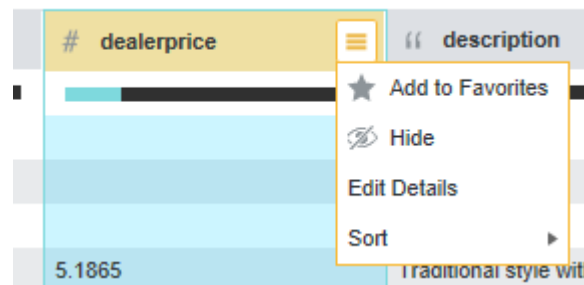
You can set the metadata properties on a public data set in the Catalog and on a project data set. Metadata changes to public data sets display to all Studio users. Changes to projects with restricted access display only to users with project access.

Changes to attribute metadata that you make in one project do not affect other projects using the same data set. However, changes to attribute metadata that you make in a public data set is inherited by projects using that data set until you run a transform to modify the data set.

To set attribute metadata:

1. From the **Explore** page:
  - a. Select the Data Set from the bottom tabs.
  - b. Locate the attribute you wish to modify.
  - c. Click the arrow at the bottom of the frame to open the Attribute Details.
  - d. Select the **Details** tab.
2. From the **Transform** page:
  - a. Select the Data Set from the bottom tabs.
  - b. Locate the column for attribute you wish to modify and select the column.
  - c. In the attribute menu, click **Edit Details**.

For example:



3. To change the display name:
  - a. Click the attribute name to enable the edit frame.
  - b. Click to edit.

- c. Enter a new display name.

To change a long name, press **CTRL+A** to select the entire text field.

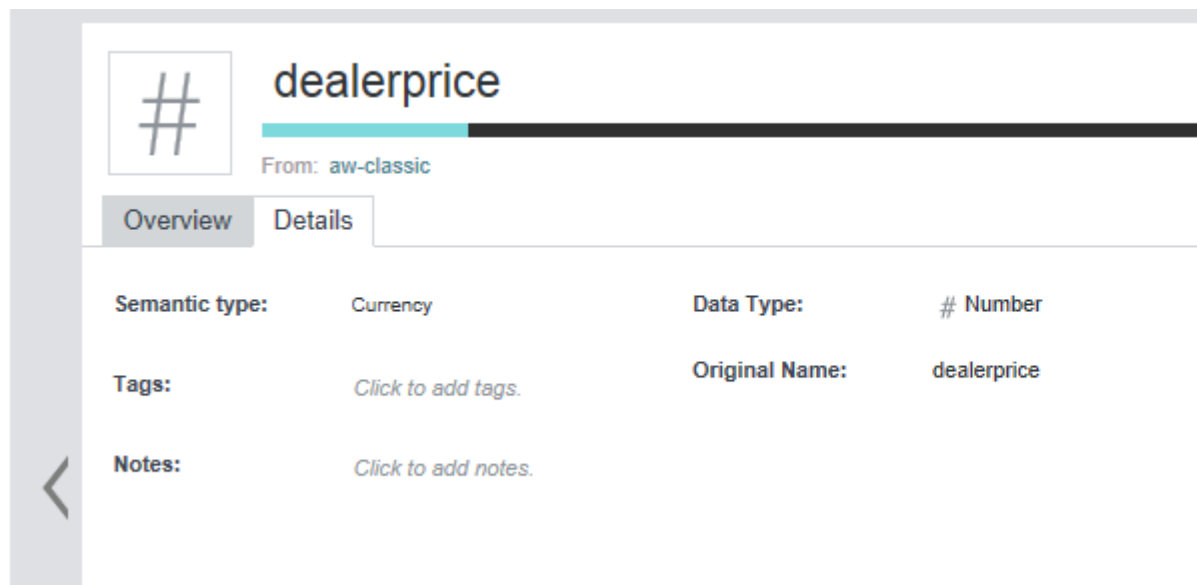
- d. Click on another area of the Details panel to save the name.

The new display name is used in place of the original attribute name.

- 4. To add or change the semantic type:

- a. From the **Semantic type** list, choose an option that more finely describes the attribute.

For example, this `dealerprice` attribute has a data type of number and also has a semantic type of currency:



- b. Click on another area of the Details panel to save the semantic type.

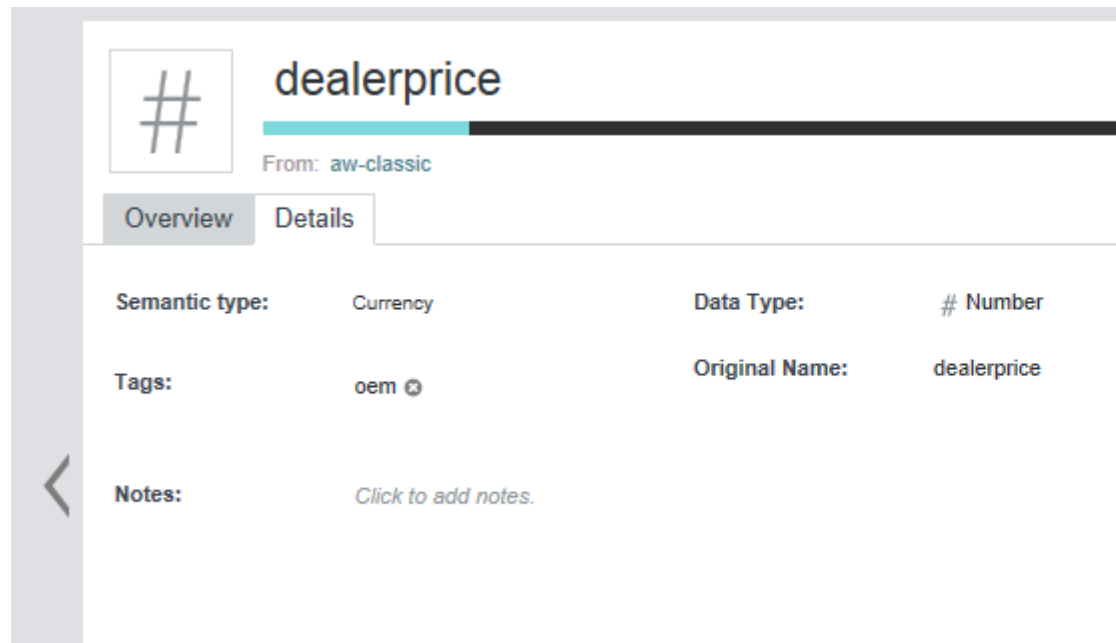
- 5. To add or remove tags:

- a. Mouse over the **Tags** area until the edit frame appears.

- b. Click to edit.

- c. To add a tag, type a tag name.

For example, this `dealerprice` attribute has a tag of `oem`:

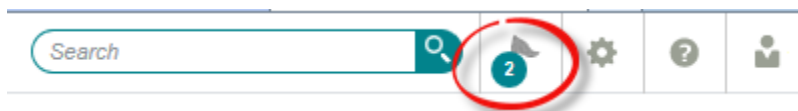


- d. Click on another area of the Details panel to save the new tag.
  - e. Repeat Steps a-d to add additional tags.
  - f. To remove a tag, click the corresponding x icon.
6. To add or remove a note:
- a. Mouseover the **Notes** area until the edit frame appears.
  - b. Click to edit.
  - c. Specify note text to describe the attribute.
  - d. Click on another area of the Details panel to save the note.
- Notes include a "Last Modified" entry above the text field that indicates the username and time associated with the last edit.
- e. To remove a note, edit the field and remove any content.
- Press **CTRL+A** to select all text in a large note for deletion.
7. Click x in the top of the Details pane to return to the Explore or Transform page.

## About notifications for data set changes

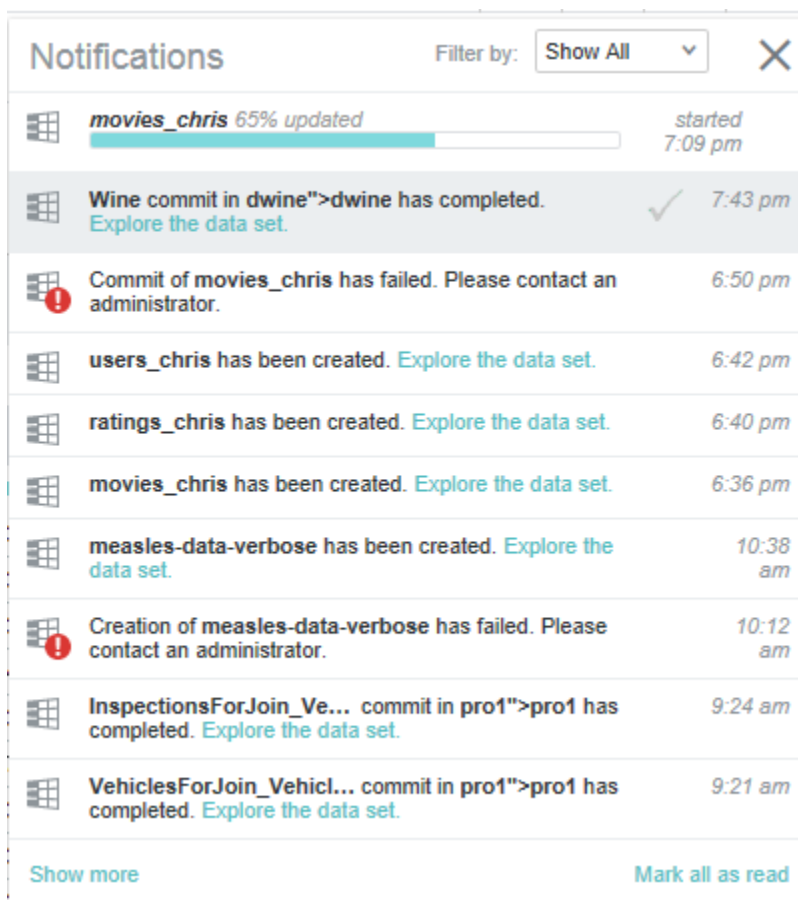
Studio provides a **Notifications** panel for you to view notifications about the progress of any data set changes you initiated. Notifications can include information about creating new data sets, updates to data sets including refreshes and full data loads, transformations, exports, deletions, and so on.

To display the **Notifications** panel, click the **Notifications** icon (a flag) in the header of Studio:



The icon also might have a number indicating the amount of unread notification messages since the last time you read the notifications or clicked **Mark all as read**.

After you open the panel, you see all notifications for data sets you have access to. For example, here is a snapshot of a **Notifications** panel:



Changes that are in progress display with a corresponding progress bar inside the **Notifications** panel, and at the moment the change completes, there is a brief banner notification across the Studio header to alert you that the change is complete.

Once you receive a notification that changes are complete, you can explore the changes by clicking **Explore the data set**.

### Default permissions for notifications

When you open the **Notifications** panel, Studio displays only notifications for projects that you have privileges to view. So for example, if a **project restricted user** is a member of only Project A, then the user gets notifications for only Project A. On the other hand, an administrator gets all notifications for all data sets and project changes. In short, if you can see a data set or project in the Catalog, then you receive notifications for changes to that data set or project.



### Configuring prune settings for notifications

An administrator can configure how long to store notifications based on a specified number of days or based on a specified number of notifications. For details, see the "Studio settings list" topic in the *Administrator's Guide*.

## Managing access to a data set

Users must have access to a data set in order to see it in the Catalog or search results, explore the data, or add the data set to a project.

Any user with access to a data set can manage access rights to the data.

Default data set permissions are configured by Studio Administrators in the Control Panel. By default, Studio uses the following settings:

- Data sets created by personally uploading files are private. Aside from Administrators, only the file uploader has access.
- Data sets created by ingesting data from Hive are public.
- Data sets created from duplicating or exporting data are public.

To modify this behavior, see the "Studio settings list" topic in the *Administrator's Guide*.

Users with Read access to a data set can

- See the data set in search results or by browsing the Catalog
- Explore the data set
- Add the data set to a project

Users with Write access to a data set can

- Modify data set metadata
- Manage access to the data set

---

**Note:** A user without any access to a data set can still explore the data they are a Project Restricted User or Project Author on a project that uses the data set. Project Authors can use the Transform operations to create a duplicate data set and gain access to the new data set. Similarly, a user with Read-only access to a data set can create a project using that data set and then execute transformations against the data if the default data set permissions include Write access. If you are working with sensitive information, consider this when assigning project roles and data set permissions. See [Project and User Roles](#) for more information.

---

To manage access to a data set:

1. From the **Data Sets** tab in the Catalog, select the data set to modify and click the **edit** link beside the data set **Access** level:

Access: Private [edit](#)



The **Data Set Security Setup** pane displays.

2. To add or remove individual users:

- a. Click **+User**.

The **Add Users** dialog appears.

- b. To add users, drag user names from the **Available Users** list to the **Selected Users** list.

Optionally, use the Search field to filter the **Available Users** list.

- c. To remove users, click the **x** next to a user name in the **Selected Users** list or click **Clear All** to remove all selected users.

- d. Click **Save**.

3. To add or remove groups:

- a. Click **+Group**.

The **Add Groups** dialog appears.

- b. To add groups, drag group names from the **Available User Groups** list to the **Selected User Groups** list.

Optionally, use the Search field to filter the **Available User Groups** list.

- c. To remove groups, click the **x** next to a group name in the **Selected User Groups** list or click **Clear All** to remove all selected groups.

You can also remove groups from the **Group access** table on the **Sharing** page by clicking the **x** in the right side column.

---

---

**Note:** You cannot remove or modify permissions for the **All Big Data Discovery admins** group.

---

---

- d. Click **Save**.

4. To set access for users or groups:

- a. Select an access level in the **Access** column:

- **No Access** — The user group cannot access the data set. The data set does not show up for this user or group in the Catalog.
- **Default Access** — The user group has default access to the data set. The "default" access level is set on the Studio Settings page in the Control Panel. See the *Administrator's Guide* for more information.
- **Read-only** — The user or user group can access the data set and modify the data set as a part of a project, but they cannot modify the data set metadata or set permissions.
- **Read/Write** — The user or user group has full access to the data set, including setting permissions.

---

---

**Note:** For individual users you cannot set permissions to "Default Access" or "No Access." Assigning individual permissions is outside of the scope of a default permissions model, and restrictive access to data should be based on a whitelist, where a user has minimal permissions and is given access to specific data sets, rather than a blacklist where a user has global access and only certain data sets are restricted.

---

---

b. Click **Save**.

5. Click **Save**.

## Deleting a data set

If you have the role of User, Power User, or Administrator, you can delete any data set from the Catalog. If you have the role of Viewer, you cannot delete a data set.

You cannot delete a data set that is currently being used in a project.

To delete a data set, click **Delete** on the data set details panel.

---

---

**Note:** Deleting a data set from the Catalog does not delete its corresponding Hive table in the Hive database. For details about the relationship between a data set and its Hive table, see the *Data Processing Guide*.

---

---



---

## Exploring a Data Set

**Explore** allows you to explore a data set, either from the Catalog or from within a project.

### About exploring a data set

**Explore** allows you to explore the structure and data values in a data set.

### Selecting the data to explore

You can explore a data set from the Catalog or from within a project.

### Switching Explore between tile view and table view

By default, **Explore** displays each attribute as a tile containing a visualization showing the distribution of values for that attribute.

### Filtering and sorting displayed attributes in Explore or Transform

You can filter and sort displayed attributes in the **Explore** and **Transform** views to more easily navigate a data set. If you are within a project data set, these preferences apply for all project users.

### Working with an individual attribute in Explore

From **Explore**, you can filter the values, select a different visualization, and display a details panel with additional options.

### Using the scratchpad

You can use the scratchpad component on the **Explore** page to try out different visualizations with one or more attributes before adding them to a **Discover** page.

### Keyboard shortcuts for Scratchpad

Studio provides a number of keyboard shortcuts to simplify interacting with Scratchpad. See the following table.

## About exploring a data set

**Explore** allows you to explore the structure and data values in a data set.

---

**Explore** ▾ **Transform** ▾ **Discover** ▾

Using **Explore**, you can find out:

- What attributes are available
- The types of values for those attributes
- The distribution of those values across the data set

- The quality of the values – in particular, whether values are missing
- How attributes correlate to each other

After exploring a data set, you should have a better idea of the visualizations you may want to use to show the data values and relationships, and the transformations that you may want to apply.

## Selecting the data to explore

You can explore a data set from the Catalog or from within a project.

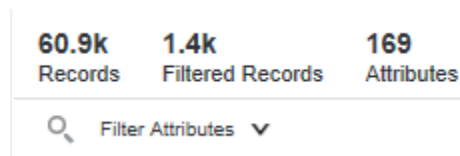
### Selecting a data set to explore

To explore a data set, click a data set name in the **Data Sets** tab of the Catalog.

To explore a project data set, click the data set tab from the data sets view of a project.

### Explore header information and links

At the top of **Explore** is a summary of the data. For example:



This summary includes:

- The number of records in the data set sample. Or if this number is less than 1,000,000 this is the number of records in the full data set.
- An option to switch between Precise mode and Approximate mode. This option displays only in data sets that are larger than the sample size. By default, this is 1,000,000 records. Precise mode provides a precise calculation for sum operations performed against the data sample. Approximate mode provides an approximation for sum operations across the full data set.
- The number of records in the entire data set
- The number of records in the sample that match the current filters (the **Filtered Records** value).
- The number of attributes in the data set
- A **Filter Attributes** drop down to locate attributes by name, data type, and by whether they are marked as hidden or favorite.
- A **Sort** control to sort attributes.

The page header can also contains links to:

- Show/Hide the Scratchpad
- Show/Hide data set properties

## Switching Explore between tile view and table view

By default, **Explore** displays each attribute as a tile containing a visualization showing the distribution of values for that attribute.

The default visualization is based on an attribute's data type.

A data quality bar shows the percentage of valid values (blue) and null values (black). You can click a section of the data quality bar to refine the data to only include records with valid, invalid, or null values.

You use the toggle switch at the top right of the list to switch between the tile view and a table view. The table view contains the data set records.

## Filtering and sorting displayed attributes in Explore or Transform

You can filter and sort displayed attributes in the **Explore** and **Transform** views to more easily navigate a data set. If you are within a project data set, these preferences apply for all project users.

---

**Note:** Filtering attributes is a navigation convenience and does not affect the current search or refinement state for attribute values.

---

To filter and sort displayed attributes in **Explore** or **Transform**:

1. To mark an attribute as a favorite, click the star icon on the attribute tile in **Explore** or select **Add to favorites** from the attribute details in **Transform**.
2. To mark an attribute as hidden, click the eye icon on the attribute tile in **Explore** or select **Hide** from the attribute details in **Transform**.
3. To filter, click the **Filter Attributes** drop-down on the left side above the displayed attribute tiles (**Explore**) or table (**Transform**).

The filtering submenu appears.

4. Apply filters by selecting the corresponding checkbox(es) from each submenu.

You can also click the Search icon next to the drop-down to see matching available filters sorted by type. For example, entering "st" as a query may suggest **Data Type > string** and **Notes > this is a test**.

- For **Attribute name**, **Tags**, and **Notes** filters, you can enter a search query to filter the list of options to a manageable size.
- Selecting the **Hidden** filter temporarily disables all other filters. Removing it re-enables the other selections.
- Selecting the **Favorites** filter removes the **Hidden** filter.

Applied filters display alongside the drop-down, and the attribute tiles or table are reduced to the subset of attributes that match all filters.

5. To remove filters, either click the **X** icon next to a filter or click **Clear all** to remove all filters.
6. To sort attributes, click the **Sort** drop-down and select a sort type:
  - **Name** - sort by name
  - **Null percentage** - sort by the percentage of null values for each attribute

- **Information potential** - sort by an information potential algorithm that considers factors such as distribution of attribute values across records. This is the default.
- **Relationship to an attribute** - sort by correlation with another attribute.  
For example, a list of wine attributes sorted by relation to "body" may show that "region" is strongly related.

In **Explore**, you can also sort by relationship to an attribute by clicking the arrow icon for that attribute tile.

## Working with an individual attribute in Explore

From **Explore**, you can filter the values, select a different visualization, and display a details panel with additional options.

### Filtering the attribute values

The **Available Refinements** panel allows you to filter attribute values by selecting specific values or ranges of values. The refinements are added to the **Selected Refinements** panel at the top of the page.

You can also use the individual visualizations to refine the data based on the attribute values.

### Hiding attributes or marking attributes as favorite

You can flag attributes that you are particularly interested in, or hide those that have little value to you. If you hide an attribute or mark it as a favorite, then when exploring a data set from the Catalog, that attribute shows as favorite or hidden for all users who explore the data set outside of a project.

### Displaying a different visualization for an attribute

In tile view, each attribute may have multiple visualization options. For example, for a numeric attribute, you might be able to switch between a histogram or a box plot.

If there are multiple visualization options, then to switch to a different visualization, use the paging arrows to the left and right of the attribute visualization.

### Displaying the details panel for an attribute

When you click the attribute name on the attribute tile, the details panel for the attribute displays.

You can use the paging icons to the left and right of the details panel to page through the details panels for each attribute.

The panel includes options to:

- Indicate the attribute is a favorite or hidden
- Sort the attribute values
- Add the attribute to the scratchpad

The details panel also contains the following tabs:



Tab	Description
Overview tab	<p>Displays the attribute visualization, with a <b>Summary Statistics</b> section listing information about the attribute and its values.</p> <p>Also displays thumbnails representing the available visualizations for the attribute.</p> <p>To change to a different visualization, either use the paging icons to the left and right of the visualization, or click the visualization thumbnail.</p>
Details tab	<p>Displays some details about the attribute, including:</p> <ul style="list-style-type: none"> <li>• Display name</li> <li>• Enrichments that were applied to the original values</li> <li>• Data type</li> </ul> <p>It also displays the list of values for the attribute, sorted by relative frequency within the data sample.</p>

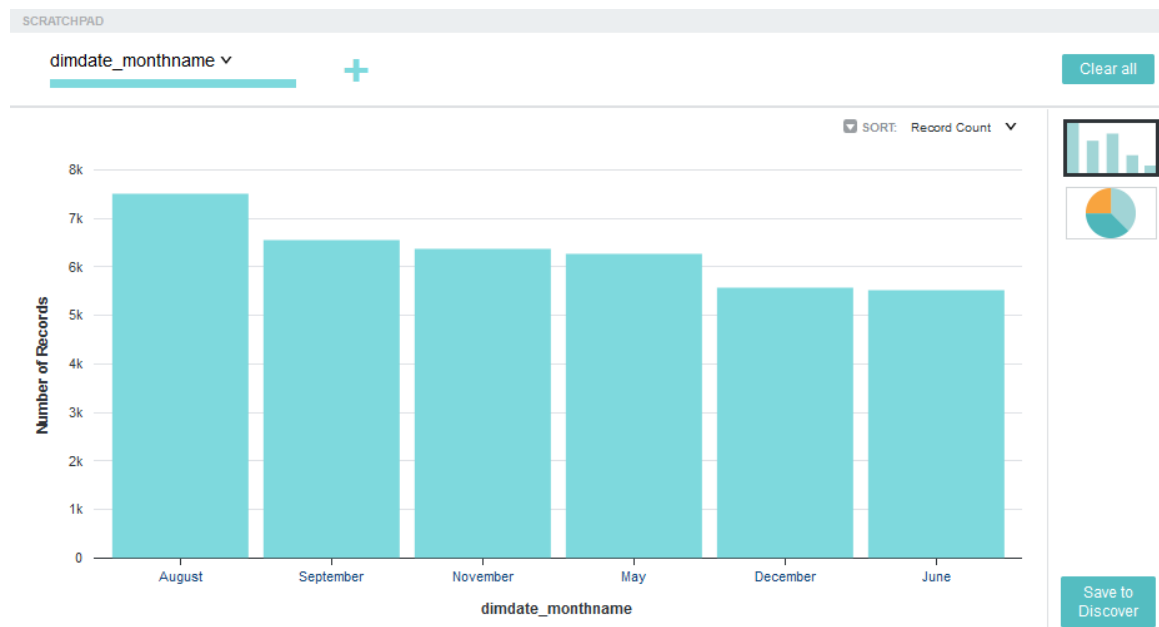
## Using the scratchpad

You can use the scratchpad component on the **Explore** page to try out different visualizations with one or more attributes before adding them to a **Discover** page.

### About the scratchpad

The scratchpad consists of:

- The scratchpad drop zone, which lists the attributes currently displayed on the scratchpad
- The scratchpad visualizations



The visualizations available in the scratchpad can include charts, box plots, histograms, and maps. These function similarly to the chart components; for information, see [Charts](#).

The type and number of visualizations available for an attribute or combination of attributes varies. If the scratchpad contains only one attribute, it displays the same visualization as the attribute's tile and allows you to switch to the other visualizations available in the attribute's details panel. If the scratchpad contains multiple attributes, it displays the most appropriate visualization for the current set.

---

---

**Note:** The scratchpad cannot display more than 1000 marks in a visualization. Refine your filtering criteria if you exceed this limit. Some combinations of attributes are incompatible and cannot be displayed by the scratchpad at all.

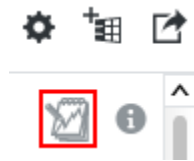
---

---

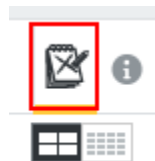
You can page through the available visualizations by clicking the arrows on either side of the scratchpad. You can also display a specific visualization by clicking its thumbnail next to the scratchpad.

### Displaying and hiding the scratchpad

The scratchpad displays when you add an attribute to it, or when you click the **Show Scratchpad** icon at the top right of the **Explore** page.



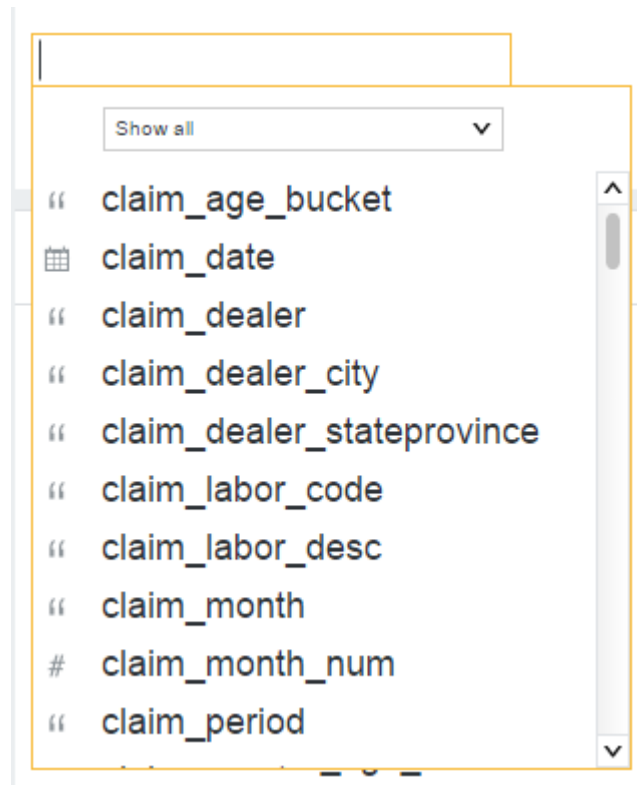
To hide the scratchpad, click either the **Hide Scratchpad** icon or the white arrow at the bottom of the scratchpad.



### Adding attributes

You can add attributes to the scratchpad in multiple ways:

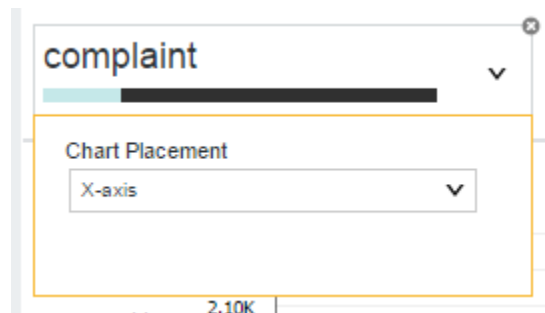
- Hover your mouse over an attribute tile and click **Add to scratchpad**.
- Click the down arrow on an attribute tile to open its details panel and click **Add to scratchpad**.
- Click anywhere within the scratchpad drop zone and either enter the name of the attribute you want to add or select it from the list of available attributes.



As you add attributes to the scratchpad, the other attributes may be sorted to put the most relevant first.

### Configuring visualizations

Each type of visualization is configurable; for example, you can change the chart placement and sort order of a histogram plot. To view options for configuring the visualization, click the down arrow next to the attribute's name in the scratchpad drop zone.



You can swap attributes between zones in a visualization by dragging and dropping them on top of each other in the drop zone. Attempting to drag and drop attributes shows a preview icon that indicates whether values are swapped or dropped upon completion. For example, you can re-order inner and outer rings in a Sunburst chart. Dragging and dropping attributes preserves any configured sort or aggregation behavior unless swapping a dimension (non-countable values, such as Strings) with a metric.

You can also modify the sort order using the **Sort** arrow, when available.

## Refining visualizations

Any refinements applied to your project are automatically applied to visualizations within the scratchpad. You can also click on specific values within the visualization to set them as refinements.

You can modify and remove your selected refinements using the **Selected Refinements** panel. For more information, see [Managing Selected Refinements](#).

## Removing attributes

To remove an individual attribute from the scratchpad, hover your mouse over its name in the drop zone and click the delete icon.

To remove all currently selected attributes, click **Clear all** in the top right of the scratchpad.

## Saving a visualization to a Discover page

If you are working within a project, you can save your visualization to a **Discover** page when you are satisfied with it.

---

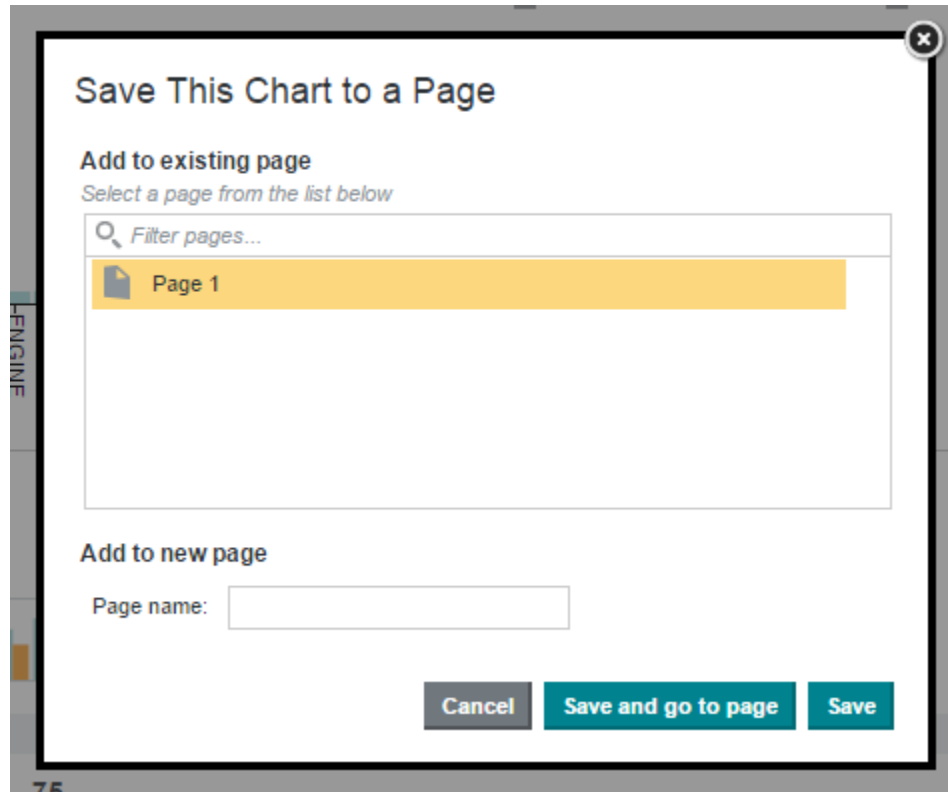
---

**Note:** Saving a visualization to a project page causes the project to open to the **Discover** page instead of the **Explore** page. If you are not working within a project, you must create one before saving your visualization.

---

---

To save a visualization to a **Discover** page, click **Save to Discover** page, then either select an existing page to save it to, or provide a name for a new page.



## Keyboard shortcuts for Scratchpad

Studio provides a number of keyboard shortcuts to simplify interacting with Scratchpad. See the following table.

**Table 6-1** *Keyboard shortcuts for Scratchpad*

Shortcut	Description
s	Moves the cursor focus to Scratchpad and displays the attribute list.
spacebar	Displays the attribute list.
Up/Down arrow	Moves up or down the attribute list.
Enter	If an attribute is highlighted in the list, Enter adds the attribute to the Scratchpad.
Left/Right arrow	If you have multiple attributes in the drop zone, arrows move the focus forwards or backwards between attributes.
Tab/Shift + Tab	Moves the focus to the next / previous object in the user interface.
Backspace	If the cursor is at the end of an attribute, this removes the attribute.
Esc	Cancels an action.



# Part IV

---

## Creating and Configuring Projects





---

## Creating, Localizing, and Deleting Projects

Studio users can create projects to transform data and explore data sets in greater detail. Project Authors and Administrators can localize or delete existing projects.

### About projects

Any edit operation, such as transforming data, must take place within a project rather than in a data set. Creating a new project also provides several other features that are not available when exploring a data set. In a project you can:

### Creating a project

You can create a project from the Catalog, or while exploring a data set that you wish to include.

### Localizing the project name, description, page names, and component titles

You can use the **Project Settings > Localization** page to localize a project's name, description, page names, and component titles.

### Deleting a project using the Catalog

Both Project Authors and Studio Administrators can delete individual projects from the Catalog or they can use the **Projects** page on the Control Panel to delete multiple projects at a time.

## About projects

Any edit operation, such as transforming data, must take place within a project rather than in a data set. Creating a new project also provides several other features that are not available when exploring a data set. In a project you can:

- Create and save project pages in **Discover**.
- Add multiple data sets, and link them in the **Data Set Manager** editor. In other words, a project provides the structure to join, aggregate, and transform multiple data sets. Note however, that a project does not make a copy of a data set.
- Modify attribute metadata in the **Attribute Groups** editor. For example, you can preserve changes for display name settings or dimension settings as part of project.
- Create and modify data views in the **Project Settings > Data Views**.

There are several ways to create a project:

- From the Catalog, you can immediately create a new project before you explore or transform the data. This is a good option if you know you are going to make changes to the data and prefer to create a new project right away. In other words, you typically do this *early in the discovery process*.

- From **Explore**, you can create a new project after examining the source data. You typically do this *later in the discovery process*.
- Or, you can create a new project only after being prompted by Studio during an edit operation that you want to perform. You typically do this *only if required*.

If you have a role of Administrator, Power User, or User, you can create new projects. If you only have the Restricted User role, you cannot create projects.

## Creating a project

You can create a project from the Catalog, or while exploring a data set that you wish to include.

---

---

**Note:** Studio does not require unique project names. Try to think of a name that is unlikely to cause confusion with existing projects, and avoid naming your project after the data set(s) it uses.

---

---

To create a new project:

1. From the Catalog:
  - a. Select a data set to include.
  - b. Expand the data set preview menu and click **Add to Project**.
  - c. Enter a name and description for the new project.
  - d. Click **Add**.
2. From the **Explore** page:
  - a. Click **Add to Project**.
  - b. Enter a name and description for the new project.
  - c. Click **Add**.

The new project opens to the **Explore** page.

## Localizing the project name, description, page names, and component titles

You can use the **Project Settings > Localization** page to localize a project's name, description, page names, and component titles.

If you localize these items, then when end users select a different locale, these items can be displayed in the correct languages.

To localize the project name, description, page names, and component titles:

1. On the project menu bar, click the settings icon.
2. In the project settings menu, click **Localization**.

The **Localization** page contains a table listing:

- Project name
- Project description
- Pages
- Under each page, the list of components

Localization

Select a locale to customize text for that language and location.  
To change default values for all locales, edit "United States | English" locale values.

United States | English ▼

Element	Use Current Default	Override Default
Project Name	<input type="radio"/>	<input checked="" type="radio"/> Sales Discovery Dashboard
Project Description	<input type="radio"/>	<input checked="" type="radio"/> Dashboard for exploring recent sales data.
Page Title	<input type="radio"/>	<input checked="" type="radio"/> Wine
Component: Chart	<input checked="" type="radio"/> Chart	<input type="radio"/>
Component: Results Table	<input checked="" type="radio"/> Results Table	<input type="radio"/>

For the project name and description, you can only localize these values in other languages. You use the **Project Configuration** page to change the default values. For page names and component titles, you can change the value for both the default language and for other languages.

- To provide localized values for a specific locale:
  - From the locale list, select the locale.

By default, each item in the list has **Use Current Default** selected, indicating to use the value for the default locale.

Localization

Select a locale to customize text for that language and location.  
To change default values for all locales, edit "United States | English" locale values.

Spain | Spanish ▼

Element	Use Current Default	Replace Default
Project Name	<input checked="" type="radio"/> Sales Discovery Dashboard	<input type="radio"/>
Project Description	<input checked="" type="radio"/> Dashboard for exploring recent sales data.	<input type="radio"/>
Page Title	<input checked="" type="radio"/> Wine	<input type="radio"/>
Component: Chart	<input checked="" type="radio"/> Gráfico	<input type="radio"/>
Component: Results Table	<input checked="" type="radio"/> Tabla de resultados	<input type="radio"/>

- For each item you want to localize, click **Replace Default**.
  - In the field, type the localized value, then press **Enter**.
- To save the changes to the localization, click **Save**.

## Deleting a project using the Catalog

Both Project Authors and Studio Administrators can delete individual projects from the Catalog or they can use the **Projects** page on the Control Panel to delete multiple projects at a time.

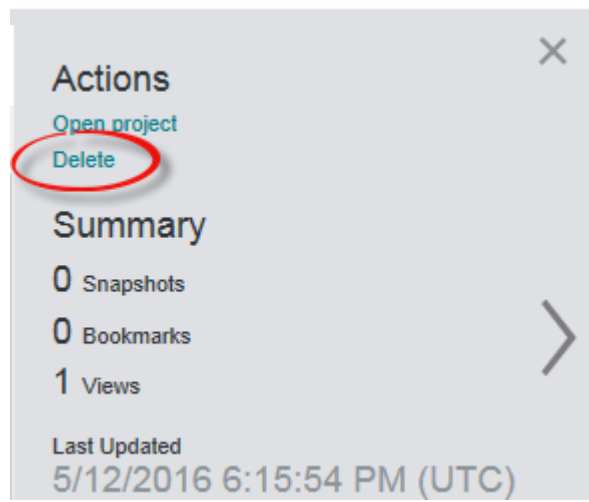
For details about using the Control Panel to delete projects, see the *Administrator's Guide*.

To delete a project using the Catalog:

- On the Catalog page, click the project tile.

2. On the project details panel, click the delete link.

For example:



When you delete a project, Studio automatically deletes any bookmarks and snapshots for that project.

---

# Managing Access to Projects

Projects are listed as either Private (the default), Public, or Shared (any project with manually configured permissions for users or user groups). You can configure access to projects by granting project roles to individual users or to user groups.

## Project and User Roles

A user who creates a project is assigned the Project Author role. Other users or groups may be either Project Authors or Restricted Project Users. Project Authors can add, remove, or set permissions for users and user groups.

## Project Types

The project type listed in the Catalog project detail panel reflects the assigned roles.

## Assigning project roles

You can grant access to a project by assigning project roles to users and user groups. Sharing settings are accessible from the **Project Settings** page or from the project details panel in the Catalog.

## Project and User Roles

A user who creates a project is assigned the Project Author role. Other users or groups may be either Project Authors or Restricted Project Users. Project Authors can add, remove, or set permissions for users and user groups.

- [Project and User Roles](#)
- [Project and User Roles](#)

## Project Roles

Project roles grant access privileges to project content and configuration. The roles are:

Role	Description
<b>Project Author</b>	<p>Project authors can:</p> <ul style="list-style-type: none"> <li>• Configure and manage a project</li> <li>• Add or remove users and user groups</li> <li>• Assign user and user group roles</li> <li>• Transform project data</li> <li>• Export project data</li> </ul> <p>Project authors cannot:</p> <ul style="list-style-type: none"> <li>• Create new data sets</li> <li>• Access the Studio Control Panel</li> </ul>
<b>Project Restricted User</b>	<p>Project Restricted Users can:</p> <ul style="list-style-type: none"> <li>• View a project and navigate through the configured pages</li> <li>• Add and configure project pages and components</li> </ul> <p>Project restricted users cannot:</p> <ul style="list-style-type: none"> <li>• Access <b>Project Settings</b></li> <li>• Create new data sets</li> <li>• Transform data</li> <li>• Export project data</li> </ul>

## User Roles

The table below summarizes the overlap between project roles and Studio user roles.

Role	Description
<b>Administrator</b>	Administrators always have the <b>Project Author</b> role for all projects.
<b>Power User</b>	Power users do not gain any access to a project beyond what is configured for the <b>All Big Data Discovery users</b> group.
<b>User</b>	Users do not gain any access to a project beyond what is configured for the <b>All Big Data Discovery users</b> group.
<b>Restricted User</b>	Restricted users do not gain any access to a project beyond what is configured for the <b>All Big Data Discovery users</b> group.

If you wish to assign project permissions by Studio roles, you can use the Control Panel **User Settings > User Groups** page to create a User Group that maps to a role, and then use the **Project Settings > Sharing** page within a project to assign permissions for that User Group.

For additional details about user roles in Studio, see "About Role Privileges" in the *Administrator's Guide*.

## Project Types

The project type listed in the Catalog project detail panel reflects the assigned roles.

Project types reflect user access settings. The project types are as follows:

Project Type	Description
Private	<ul style="list-style-type: none"> <li>The project creator and Studio Administrators are the only users with access</li> <li>The <b>All Big Data Discovery users</b> group is set to <b>No Access</b></li> </ul> <p>Projects are Private by default. Access must be granted by the creator or by a Studio Administrator.</p>
Public	<ul style="list-style-type: none"> <li>The <b>All Big Data Discovery users</b> group is set to <b>Project Authors</b></li> </ul> <p>Public projects grant view access to Studio users.</p>
Shared	<p>The project has been modified in any of the following ways:</p> <ul style="list-style-type: none"> <li>Users other than the creator are added to the project</li> <li>User Groups other than <b>All Big Data Discovery admins</b> and <b>All Big Data Discovery users</b> are added to the project</li> <li>The <b>All Big Data Discovery users</b> group is set to <b>Project Restricted Users</b></li> </ul> <p>Projects are set to Shared to indicate changes from the default Public or Private permissions.</p>

## Assigning project roles

You can grant access to a project by assigning project roles to users and user groups. Sharing settings are accessible from the **Project Settings** page or from the project details panel in the Catalog.

You must have Project Author permissions for a project in order to assign roles.

You can assign **Project Restricted User** or **Project Author** access to individual users or to user groups. Users gain the greater permissions of either their group permissions or their individually assigned permissions. For details on permission levels, see [Project and User Roles](#).

To share a project:

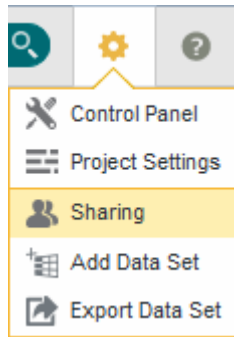
1. Navigate to the sharing settings:
  - a. From the Catalog, select the project to modify and click the **edit** link beside the project **Access** level:

Access: Private [edit](#)



The **Sharing Setup** pane displays.

- b. From within a project, click **Configuration Options** in the top right corner of the page and select **Share**:



The **Project Settings** screen opens to the **Sharing** page.

2. To add or remove individual users:

- a. Click **+User**.

The **Add Users** dialog appears.

- b. To add users, drag user names from the **Available Users** list to the **Selected Users** list.

Optionally, use the Search field to filter the **Available Users** list.

- c. To remove users, click the **x** next to a user name in the **Selected Users** list or click **Clear All** to remove all selected users.

You can also remove users from the **User access** table on the **Sharing** page by clicking the **x** in the right side column.

- d. Click **Save**.

3. Click the drop down in the **Access** column to set a user's permissions for the project.

4. To add or remove groups:

- a. Click **+Group**.

The **Add Groups** dialog appears.

- b. To add groups, drag group names from the **Available User Groups** list to the **Selected User Groups** list.

Optionally, use the Search field to filter the **Available User Groups** list.

- c. To remove groups, click the **x** next to a group name in the **Selected User Groups** list or click **Clear All** to remove all selected groups.

You can also remove groups from the **Group access** table on the **Sharing** page by clicking the **x** in the right side column.

---

---

**Note:** You cannot remove or modify permissions for the **All Big Data Discovery admins** group, and you cannot remove the **All Big Data Discovery users** group.

---

---

- d. Click **Save**.



5. Click the drop down in the **Access** column to set a group's permissions for the project.
6. Click **Save**.



---

## Managing Project Pages

A project can contain any number of pages where you create components to visualize data.

### Adding a page to a project

When you first create a project, it contains a single empty page. From the project pages view, you can then add additional pages.

### Renaming a page

After you create a page, you can change the display name that appears on its tab.

### Changing the page layout

You can select the layout to use for each page in your project.

### Changing the data set binding for a page

Each page within a project is data bound to a specific data set. This approach provides a more granular page-level data binding rather than project-level data binding. If necessary, you can change the data set associated with a particular page.

### Deleting a page

You can delete pages from a project. When you delete a page, you also delete any components on the page and bookmarks and snapshots created for that page.

### Adding and deleting components on a page

If you have Project Author or Project Restricted User permissions for a project, you can add and delete components. The components provide functions to display and create different visualizations of the project data.

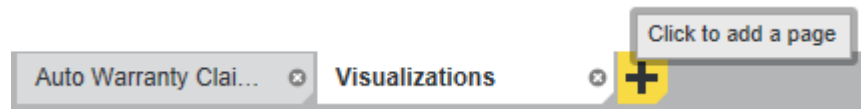
## Adding a page to a project

When you first create a project, it contains a single empty page. From the project pages view, you can then add additional pages.

Creating a project page causes your project to open to the first **Discover** page instead of the **Explore** page. Projects open to the **Explore** page if they include only an empty Page 1 (created by default for new projects), or if all pages have been removed.

To add a page to a project:

1. Select **Discover**.
2. In the page footer, click the add page icon next to the page tabs.



3. On the **Add Page** dialog, type a name for your new page, and if your project contains multiple data sets, also select a primary data set to be the main focus for this page.

This is page-level data binding. Any components you add to this page are bound to the primary data set by default.

4. Click **Add**.

The new page is added to the project and then automatically displayed, so that you can add components to it.

See [Adding and deleting components on a page](#).

Access to pages within a project depends on whether the user or the user group have access to this project. If a given user has the Project Restricted User or Project Author roles for a project, they can add and configure page components within that project. For details, see [Project and User Roles](#).

## Renaming a page

After you create a page, you can change the display name that appears on its tab.

When you rename a page, you are only changing the display name, so you do not have to change any existing page transitions.

To rename a page:

1. Click the page tab.
2. Click the page tab again.

The page name displays in an editable field.



3. In the field, type the new name.
4. To cancel the name change, and restore the previous name, press **Esc**.
5. To save the new name, click outside of the tab, or press **Enter**.

You can also use the **Localization** page to edit and localize the page name. See [Localizing the project name, description, page names, and component titles](#).

## Changing the page layout

You can select the layout to use for each page in your project.

The page layout determines how components are organized on each page. A page may have one, two, or three columns. One column width may be fixed, or all of the column widths may be proportional.

To change the layout of a page:

1. In a project, select **Discover**.
2. In the right bar, click the pencil icon to edit page settings.
3. Expand **Page Templates** and select the layout you want to apply to the page.

The new layout is applied to the page, with any existing components organized accordingly.

## Changing the data set binding for a page

Each page within a project is data bound to a specific data set. This approach provides a more granular page-level data binding rather than project-level data binding. If necessary, you can change the data set associated with a particular page.

To change the data set binding for a page:

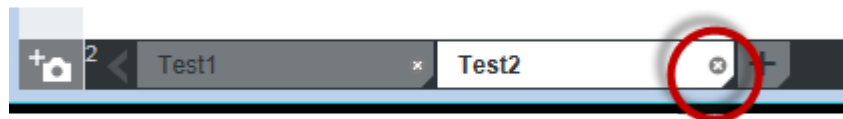
1. Select **Discover**.
2. In the footer, select the page you want to modify.
3. In the right bar, click the pencil icon to edit page settings.
4. Expand **Page Data Settings** and select a different primary data set from the list.

## Deleting a page

You can delete pages from a project. When you delete a page, you also delete any components on the page and bookmarks and snapshots created for that page.

To delete a page:

1. Click the delete icon in the top corner of the page that you want to delete.



2. To delete the page, click OK.

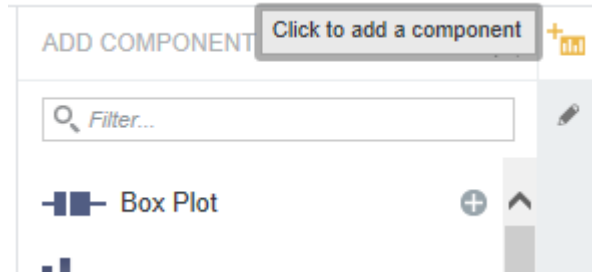
## Adding and deleting components on a page

If you have Project Author or Project Restricted User permissions for a project, you can add and delete components. The components provide functions to display and create different visualizations of the project data.

To add and delete components on a page:

1. To add a component:
  - a. Click the add component icon in the upper right of the panel.

For example:



- b. To display a description of a component, hover the mouse over the component icon.
        - c. To find a specific component, use the search field.
        - d. To add a component to the page, either:
          - Click the add icon for the component. If the page contains more than one column, then the component is added to the leftmost column.
          - Drag the component from the add page/component panel to the appropriate location on the page.
2. To delete a component from a page, click its delete icon.



For details regarding each component's function and configuration workflow, see [Working with Components](#).

# Part V

---

## Viewing and Using Projects





---

## Navigating Within a Project

From the Catalog, you can select a project to view. You can also display information about the project and its associated data sets.

### [Navigating to a project](#)

To navigate to a project, click the project name in the Catalog. The project details panel also contains a link to display the project.

### [Navigating within a project](#)

New projects initially open on the **Explore** page to allow you to explore one or more data sets in a project. Projects that already have one or more **Discover** pages configured instead open to the first **Discover** page.

### [Displaying information about the project data](#)

In the header of both the **Explore** and **Transform** pages, there is an information icon that provides information about the data sets used in the project:

## Navigating to a project

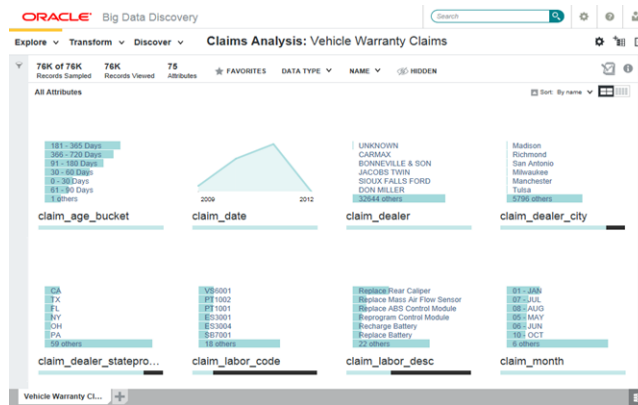
To navigate to a project, click the project name in the Catalog. The project details panel also contains a link to display the project.

## Navigating within a project

New projects initially open on the **Explore** page to allow you to explore one or more data sets in a project. Projects that already have one or more **Discover** pages configured instead open to the first **Discover** page.

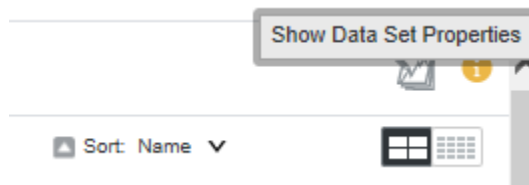
In **Explore**, each data set is represented on a tabbed page. For details on exploring a data set, see [Exploring a Data Set](#).

You can also switch to the **Transform** or **Discover** pages, and, if you have role privileges, you can also display project settings. In **Discover**, each tabbed page provides an area to add components to analyze the data and discover insights hidden in data. For details about using specific components, see [Working with Components](#).



## Displaying information about the project data

In the header of both the **Explore** and **Transform** pages, there is an information icon that provides information about the data sets used in the project:



To display information about the project, click the **Show Data Set Properties** icon.

For each data set, the information includes:

- The data source name
- When the data was last updated
- Information about the status of the data set, to indicate when a data set is loading or has a problem
- Information about other related data sets in the project

The **Data Set Relationships** tab shows how the data sets are linked.

---

## Creating and Using Bookmarks

You can create bookmarks for your own use or to share with other users.

### About bookmarks

Bookmarks allow you to save a given navigation and component state so you can return to it at a later time or email it to other users.

### Displaying and navigating to bookmarks

The bookmarks panel displays bookmarks created by the current user for the current project. It also includes any shared bookmarks created by other users.

### Creating, editing, and deleting bookmarks

You can create new bookmarks. You can also edit or delete bookmarks that you have created.

### Generating bookmark links

Users can get access to bookmarks using a hyperlink. From the **Actions** menu on the bookmark details panel, you can generate the link to the bookmark.

### Emailing bookmark links

Users can get access to bookmarks using a hyperlink. From the **Actions** menu on the bookmark details panel, you can email the link.

### Bookmark data saved for each component and panel

Bookmarks include different levels of detail for the types of components and panels on a page.

## About bookmarks

Bookmarks allow you to save a given navigation and component state so you can return to it at a later time or email it to other users.

---

**Note:** A bookmark is only guaranteed to work if the page that it points to has not been changed. If components have been added, removed, or modified, then the bookmark may no longer be valid.

---

You delete bookmarks manually, or in the following situations:

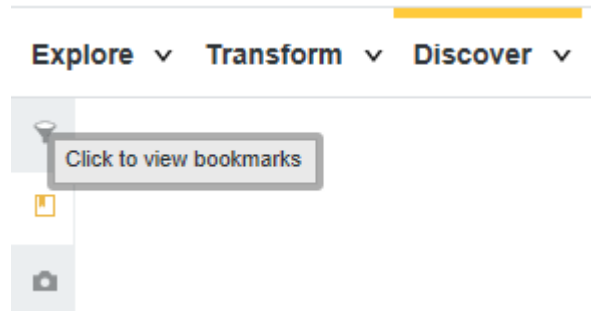
- When a page or project is removed, any bookmarks for that page or project are also removed.
- When a user is removed, any bookmarks created by that user are also removed.

## Displaying and navigating to bookmarks

The bookmarks panel displays bookmarks created by the current user for the current project. It also includes any shared bookmarks created by other users.

To display and navigate to bookmarks:

1. Within a project, click **Discover**.
2. Click the bookmarks icon in the left panel:



The **Bookmarks** pane displays.

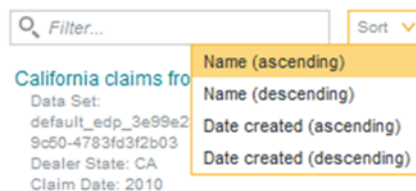
Bookmarks you create are listed under **My Bookmarks**. If you configure a bookmark to be available to other users, it is marked with a shared icon. The **Shared Bookmarks** list contains shared bookmarks created by other users.

3. To search for a bookmark, use the **Filter** field.

Studio filters the list to display bookmarks where entered text is included in the bookmark name, bookmark description, or the refinements associated with the bookmark. Search results include both your bookmarks and shared bookmarks.

4. To sort bookmarks, click **Sort** at the top of the **Bookmarks** pane and select a sort type.

You can sort the list alphabetically by bookmark name, or based on the order that bookmarks were created.



5. To navigate to a specific bookmark, click the bookmark name.

## Creating, editing, and deleting bookmarks

You can create new bookmarks. You can also edit or delete bookmarks that you have created.

To create, edit, or delete a bookmark:

1. To create a new bookmark:

- a. Within a project, click **Discover**.
- b. Click the bookmarks icon in the left panel.
- c. At the bottom of the **Bookmarks** pane, click **Add New Bookmark**.
- d. In the **Bookmark Name** field, enter a name.

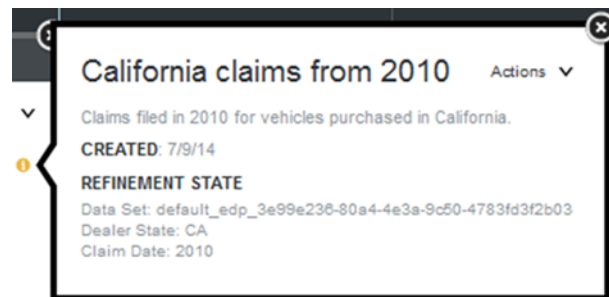
All bookmarks you create must have a unique name.

- e. In the **Description** field, provide an optional description for the bookmark.
- f. To make the bookmark a shared bookmark available to all users with access to the current project, select the **Share this bookmark with all project users** check box.
- g. Click **Save**.

2. To edit a bookmark you created:

- a. Click the information icon next to the bookmark name.

The bookmark details panel displays:



- b. Select **Actions > Edit**.
- c. Use the **Edit Bookmark** dialog to modify the bookmark name, description, or sharing status:

 A screenshot of the "Edit Bookmark" dialog box. It has a title bar with a close icon. The dialog contains the following fields:
 

- Bookmark Name:** A text input field containing "California claims from 2010".
- Description:** A text area containing "Claims filed in 2010 for vehicles purchased in California."
- Share this bookmark with all project users:** A checked checkbox.

 At the bottom right, there are two buttons: "Cancel" and "Save".

- d. Click **Save**.
3. To delete a bookmark you created:
  - a. Click the information icon next to the bookmark name.
  - b. Select **Actions > Delete**.

A confirmation dialog displays.
  - c. Click **Click to delete**.

## Generating bookmark links

Users can get access to bookmarks using a hyperlink. From the **Actions** menu on the bookmark details panel, you can generate the link to the bookmark.

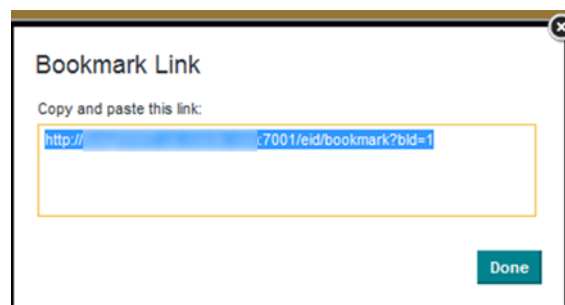
Bookmarks that are shared via links have the following properties:

- Anyone using a bookmark link must log in to Studio. Users cannot navigate to a bookmark for a project or page if they do not have permission to access it.
- Personal bookmarks (bookmarks that do not have the **Share this bookmark with all project users** check box selected) can be shared by providing a link. Users accessing a linked bookmark can create their own copy by using the **Add New Bookmark** button from the linked page or project.

To generate a link to a bookmark:

1. Click the information icon next to the bookmark name.
2. Select **Actions > Create Link**.

The **Bookmark Link** dialog displays. The link is selected automatically so that you can immediately copy it to the clipboard.



## Emailing bookmark links

Users can get access to bookmarks using a hyperlink. From the **Actions** menu on the bookmark details panel, you can email the link.

To send snapshots by email, the outbound email server for Big Data Discovery must be configured. For information on how to configure the outbound email server, see the *Administrator's Guide*.

---

**Note:** Emailing bookmark links is not supported by Big Data Discovery Cloud Service. The feature is supported by Big Data Discovery.

---

To email a link to a bookmark:

1. From the bookmark **Actions** menu, select **Email**.

The **Email Bookmark** dialog displays, with the **Subject** and **Message** fields pre-populated.

2. In the **To** field, type the email address for the recipient.

If you are sending the snapshot to multiple recipients, then for each additional recipient, click the + icon to add a field for the email address.

3. The **From** field contains your email address, which is used as the reply-to address on the email message.

Note that this is not the **From** address. The **From** address on the email is the address associated with the outbound email server.

For information on configuring the outbound email server, see the *Administrator's Guide*.

4. In the **Subject** field, you can customize the subject line for the email.
5. In the **Message** field, you can add any additional text to the email message.

Make sure you do not change the bookmark link.

6. To send the email, click **Send**.

## Bookmark data saved for each component and panel

Bookmarks include different levels of detail for the types of components and panels on a page.

Bookmarks save the following information:

<b>Component or Panel</b>	<b>Persisted States</b>
<b>Available Refinements</b>	<ul style="list-style-type: none"><li>• Whether the panel is expanded or collapsed</li><li>• Which data set is expanded</li><li>• Expanded and collapsed groups and attributes</li><li>• Toggle selection for the refinement type (value list vs. range filter vs. recurring date)</li></ul>
<b>Box Plot</b>	<ul style="list-style-type: none"><li>• Selected Y-axis metric</li><li>• Selected X-axis attribute</li><li>• Boxes per page</li></ul>
<b>Chart</b>	<ul style="list-style-type: none"><li>• All chart information. This includes the selected metric, selected dimensions, sort order and sort options, alternative chart suggestions, and all state information in the <b>Chart Settings</b> panel.</li></ul>
<b>Histogram Plot</b>	<ul style="list-style-type: none"><li>• Selected metric</li><li>• Number of bins</li></ul>
<b>Map</b>	<ul style="list-style-type: none"><li>• Records per page</li><li>• Sort order</li><li>• Map searches</li></ul>
<b>Parallel Coordinates Plot</b>	<ul style="list-style-type: none"><li>• Selected view</li><li>• Displayed metrics</li><li>• Aggregation method</li><li>• Selected dimension</li></ul>
<b>Pivot Table</b>	None
<b>Results List</b>	<ul style="list-style-type: none"><li>• Sort order</li><li>• Records per page</li></ul>
<b>Results Table</b>	<ul style="list-style-type: none"><li>• Sort order</li><li>• Number of records per page</li></ul>
<b>Selected Refinements</b>	<ul style="list-style-type: none"><li>• Whether the panel is expanded or collapsed</li><li>• Whether refinements for each data set are hidden</li><li>• Display order of the refinement tiles</li><li>• Whether each refinement tile is expanded</li><li>• Scroll position for navigating through the refinement tiles</li></ul>
<b>Summarization Bar</b>	None
<b>Tabbed Component Container</b>	Selected tab
<b>Tag Cloud</b>	<ul style="list-style-type: none"><li>• Display format (cloud or list)</li><li>• Selected dimension</li><li>• Selected metric</li></ul>
<b>Timeline</b>	<ul style="list-style-type: none"><li>• Selected time frame</li><li>• Selected date/time attributes</li><li>• Selected metrics</li></ul>

---



---

# Creating and Managing Snapshots

Studio allows you to create and manage snapshots of project pages.

## About snapshots and snapshot galleries

A snapshot captures an image of a component or page at a specific moment in time. You can save these snapshots in Studio.

## Displaying the list of snapshots and snapshot galleries

If you click the snapshot (camera) icon, available on the **Discover** page, Studio displays the snapshots panel.

## Creating and managing individual snapshots

You can create, view details, and perform actions on individual snapshots.

## Exporting the entire snapshot list

The **Actions** menu on the snapshots panel includes an option to export the entire list of snapshots. This feature saves the images to a single PDF file.

## Emailing the entire snapshot list

The **Actions** menu on the snapshots panel includes an option to email the entire list of snapshots. This feature saves the images to a single PDF file and emails the file.

## Creating and managing snapshot galleries

You can group snapshots into snapshot galleries, which you can then email to other users.

## About snapshots and snapshot galleries

A snapshot captures an image of a component or page at a specific moment in time. You can save these snapshots in Studio.

You can email individual snapshots, save them as image files, and return to the project state reflected by the snapshot.

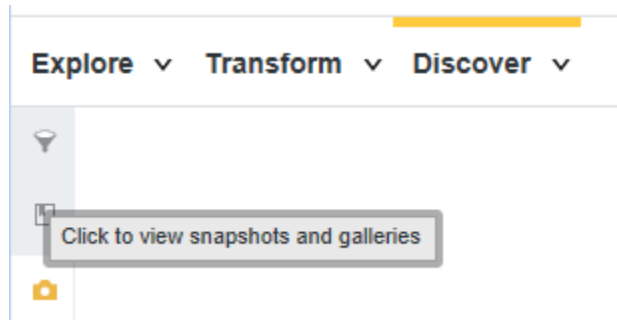
You can also create galleries of related snapshots. For example, a snapshot gallery could show the progression of chart data as different refinements are applied.

You can email a snapshot gallery and save a gallery to a PDF file.

## Displaying the list of snapshots and snapshot galleries

If you click the snapshot (camera) icon, available on the **Discover** page, Studio displays the snapshots panel.

For example:



The snapshots panel includes:

- The **Snapshots** list, showing the snapshots you have created for this project.
- The **Galleries** list, showing the snapshot galleries you have created for this project.

To find a specific snapshot, use the search field at the top of the **Snapshots** list. As you type, Studio filters the list to display snapshots with the entered text in the snapshot name, snapshot description, or the refinements associated with the snapshot.

To find a specific gallery, use the search field at the top of the **Galleries** list. As you type, Studio filters the list to display galleries with the entered text in the gallery name or gallery description.

## Creating and managing individual snapshots

You can create, view details, and perform actions on individual snapshots.

### Creating a snapshot

From either the project footer or the snapshots panel, you can create snapshots of a project page. The snapshots are added to the **Snapshots** list.

### Managing saved snapshots

From the **Snapshots** list for a project, you can edit, email, save, and delete snapshots. You can also use the snapshot as a bookmark to display the project page in the state reflected by the snapshot.

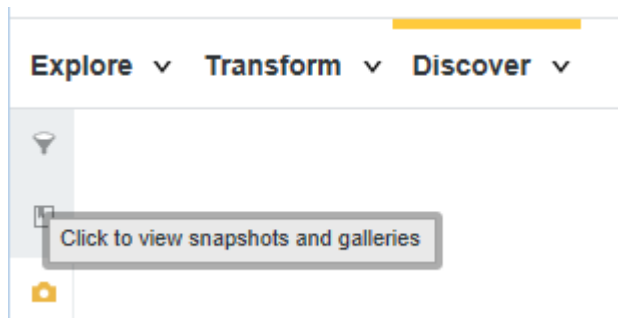
## Creating a snapshot

From either the project footer or the snapshots panel, you can create snapshots of a project page. The snapshots are added to the **Snapshots** list.

To create a new snapshot:

1. Select **Discover** if you haven't already.
2. Click the snapshot (camera) icon to display the Snapshots panel

For example:



3. In the project footer click **New snapshot** or on the snapshots panel, click **New snapshot**.

Both bring up the same dialog. The **Add Snapshot** dialog displays. The dialog includes a generated default name for the snapshot.

4. In the **Snapshot Name** field, update the name for the snapshot.
5. In the **Description** field, type a description of the snapshot.
6. Click **Save**.

## Managing saved snapshots

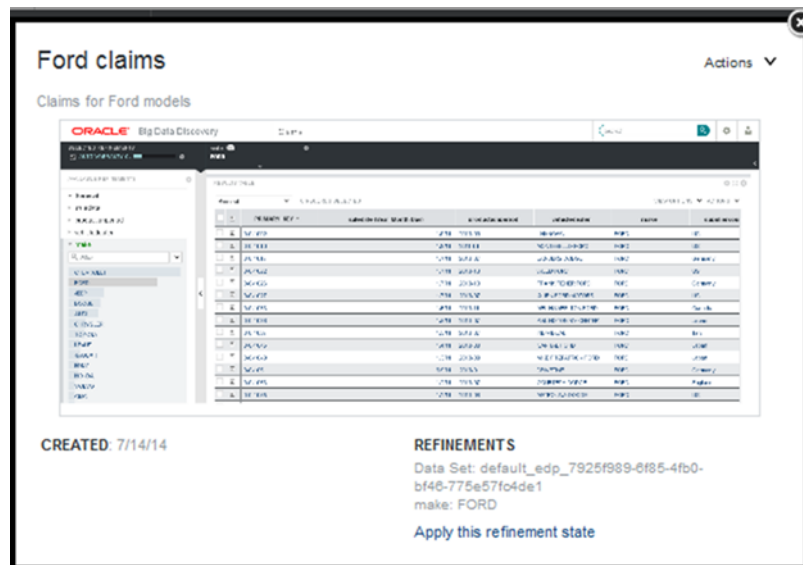
From the **Snapshots** list for a project, you can edit, email, save, and delete snapshots. You can also use the snapshot as a bookmark to display the project page in the state reflected by the snapshot.

Note that in order to be able to send snapshots by email, the outbound email server for Studio must be configured. For information about how to configure the outbound email server, see the *Administrator's Guide*.

To manage the snapshots in the **Snapshots** list:

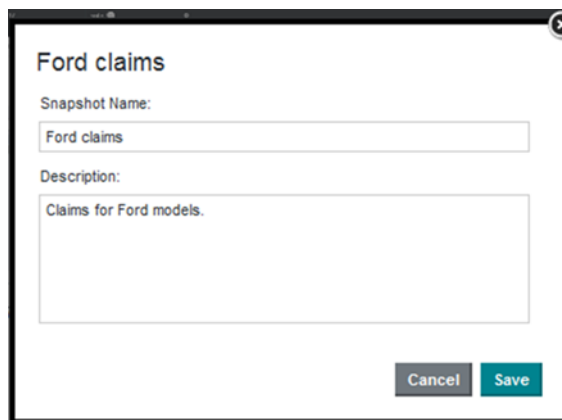
1. To display details for a snapshot, click the snapshot.

The snapshot details include the name, description, creation date, source of the data, and the refinements that were in place when you created the snapshot.



The **Actions** menu contains options to edit, email, save, and delete the snapshot. You can also navigate to the refinement state represented by the snapshot.

2. To edit a snapshot:
  - a. From the snapshot **Actions** menu, select **Edit**.
  - b. On the edit dialog, you can edit the name and description of the snapshot.



- c. To save the changes, click **Save**.
3. To email a snapshot:
  - a. From the snapshot **Actions** menu, select **Email**.

The **Email Snapshot** dialog displays.

- b. In the **To** field, type the email address of the recipient.

If you are sending the snapshot to multiple recipients, then for each additional recipient, click the + icon to add a field for the email address.

- c. The **From** field shows your email address as the "Reply to:" address on the email message.

Note that this is not the **From** address. The **From** address on the email is the address associated with the outbound email server.

For information on configuring the outbound email server, see the *Administrator's Guide*.

- d. In the **Subject** field, edit the subject line of the email.

- e. In the **Message** field, edit the text for the email message.

- f. To send the snapshot, click **Send**.

4. To save a snapshot to a file, from the **Actions** menu, click **Save to Disk**.

Depending on your browser, you are prompted to open or save the image.

5. To display the project page using the refinement state associated with the snapshot, on the snapshot details, click **Apply this refinement state**.

6. To delete a snapshot, from the snapshot **Actions** menu, select **Delete**.

You are prompted to confirm that you want to delete the snapshot from the list. If they snapshot is part of a gallery, it is still stored there.

## Exporting the entire snapshot list

The **Actions** menu on the snapshots panel includes an option to export the entire list of snapshots. This feature saves the images to a single PDF file.

To export the entire set of snapshots:

1. In the **Actions** menu, click **Save All to Disk**.
2. Use the file save dialog to name and save the file.

## Emailing the entire snapshot list

The **Actions** menu on the snapshots panel includes an option to email the entire list of snapshots. This feature saves the images to a single PDF file and emails the file.

Note that in order to be able to send snapshots by email, you must configure the outbound email server for Studio. For information on how to configure the outbound email server, see the *Administrator's Guide*.

---

---

**Note:** Emailing snapshots is not supported by Big Data Discovery Cloud Service. The feature is supported by Big Data Discovery.

---

---

To email entire set of snapshots:

1. In the **Actions** menu, click **Email All**.
2. On the email dialog, in the **To** field, type the email address of the recipient.

To add a field for an additional recipient, click +.

3. In the **Subject** field, type the subject line for the email.
4. In the **Message** field, type the text of the email.
5. Click **Send**.

The snapshots are sent in a PDF file attached to the email.

## Creating and managing snapshot galleries

You can group snapshots into snapshot galleries, which you can then email to other users.

### [Creating a snapshot gallery](#)

You can create snapshot galleries to group a related set of snapshots. For example, you can put together a "slide show" of screen shots showing a series of refinement states.

### [Editing and deleting snapshot galleries](#)

Within a snapshot gallery, you can add snapshots to the gallery, delete snapshots from the gallery, and change the order of the snapshots in the gallery. For each snapshot, you can provide a name and description that is specific to the gallery.

### [Emailing a snapshot gallery](#)

You can email an entire snapshot gallery to one or more email addresses. The gallery is attached to the message as a PDF file.

### [Exporting a snapshot gallery to a PDF file](#)

You can export an entire snapshot gallery to a PDF file.

## Creating a snapshot gallery

You can create snapshot galleries to group a related set of snapshots. For example, you can put together a "slide show" of screen shots showing a series of refinement states.

From the snapshots panel, to create a new snapshot gallery:

1. On the snapshots panel, click **New Gallery**.

The new gallery dialog displays.

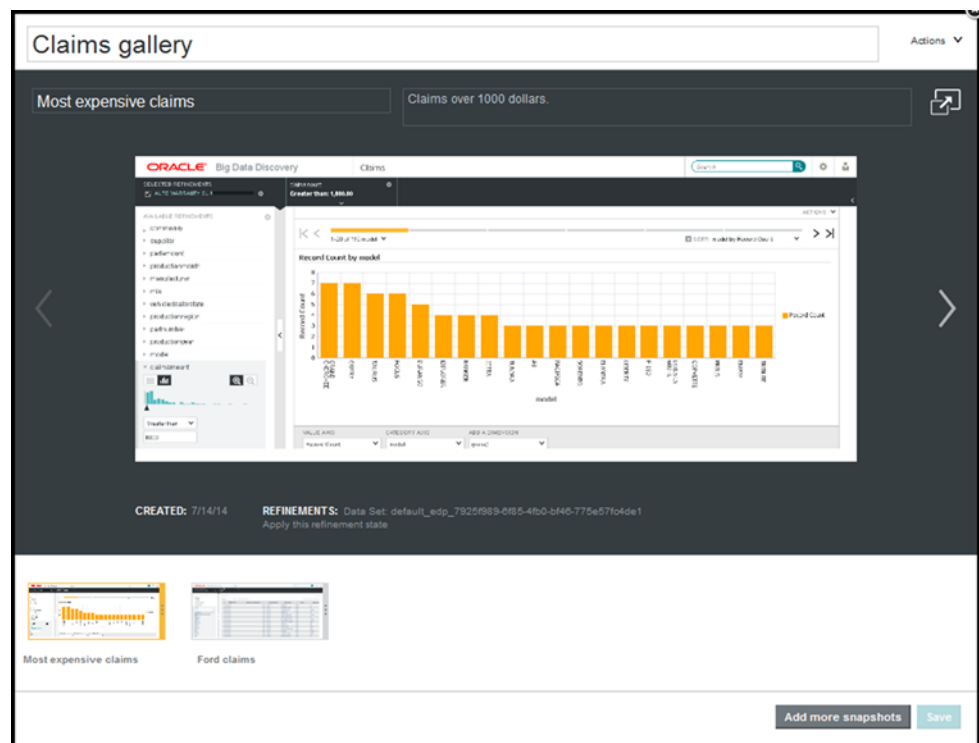


2. In the name field, type the name of the new gallery.
3. In the list of snapshots, click each snapshot you want to add to the new gallery.  
To deselect a snapshot, click it again.
4. To save the new gallery, click **Save**.

The gallery panel for the new gallery displays.

The first snapshot is selected and displays at the top of the panel.

At the bottom of the panel is the full list of snapshots in the gallery.



## Editing and deleting snapshot galleries

Within a snapshot gallery, you can add snapshots to the gallery, delete snapshots from the gallery, and change the order of the snapshots in the gallery. For each snapshot, you can provide a name and description that is specific to the gallery.

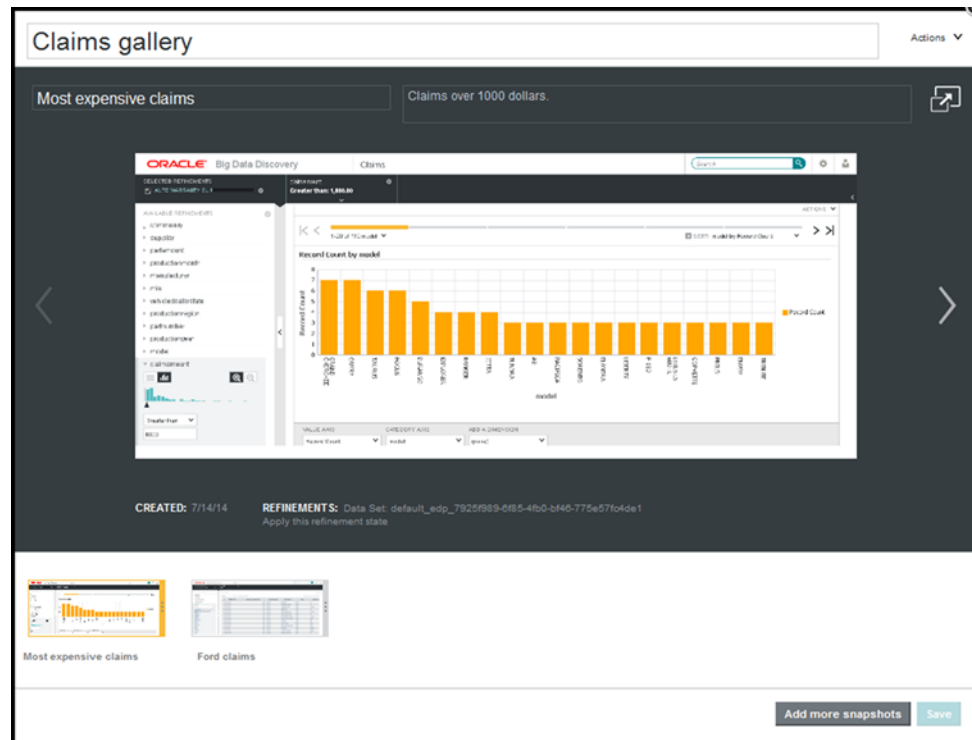
To edit and delete a snapshot gallery:

1. To display the details for a gallery, in the **Galleries** list, click the information icon for the gallery.

The gallery panel for the selected gallery displays.

The first snapshot is selected by default, with the full list of snapshots in the gallery displayed at the bottom of the panel.

You can use the next and previous icons to page through the snapshot list.



2. To determine the order of the snapshots in the gallery, drag and drop the snapshots in the gallery snapshots list.
3. To display and edit the details for a snapshot in the gallery, in the snapshots list, click the snapshot.

You can use the name and description fields at the top of the panel to customize the name and description of the snapshot in the context of the gallery.

This name and description is specific to this instance of the snapshot within the gallery.

4. To page through the snapshots in full screen mode:
  - a. Click the full screen icon at the top right.



- b. Use the next and previous icons to page through the snapshots.
        - c. To exit the full screen mode, either click the return icon, or press the **Esc** key.
5. To add snapshots to the gallery:
  - a. Click **Add more snapshots**.
  - b. On the add snapshots dialog, click each snapshot that you want to add.  
To deselect the snapshot, click it again.
  - c. When you have selected and deselected the snapshots as needed, click **Save**.
6. To delete a snapshot from the gallery:
  - a. Hover the mouse over the snapshot.
  - b. Click the delete icon.
7. To save the changes to the snapshot gallery, click **Save**.
8. To delete a snapshot gallery, in the **Galleries** list, click the delete icon for the snapshot gallery.

## Emailing a snapshot gallery

You can email an entire snapshot gallery to one or more email addresses. The gallery is attached to the message as a PDF file.

Note that in order to be able to send a snapshot gallery by email, you must configure the outbound email server for Big Data Discovery. For information on how to configure the outbound email server, see the *Administrator's Guide*.

---

---

**Note:** Emailing a snapshot gallery is not supported by Big Data Discovery Cloud Service. The feature is supported by Big Data Discovery.

---

---

To email a snapshot gallery:

1. In the **Galleries** list, click the information icon for the snapshot gallery.
2. In the gallery **Actions** menu, click **Email**.
3. On the **Email Gallery** dialog, in the **To** field, type the email address of the recipient.  
To add another recipient, click **+**.
4. In the **Subject** field, edit the subject line of the email.
5. In the **Message** field, edit the text for the email message.
6. Click **Send**.

## Exporting a snapshot gallery to a PDF file

You can export an entire snapshot gallery to a PDF file.

To export a snapshot gallery:

1. In the **Galleries** list, click the information icon for the snapshot gallery.
2. In the gallery **Actions** menu, click **Save to Disk**.
3. Use the file save dialog to name and save the PDF file.

# Part VI

---

## Managing Project Data



---

## Managing Project Data Sets

The **Data Set Manager** provides options add, remove, or load a full data set within a project and also to enable or disable links among multiple data sets in a project.

### [Adding a data set to an existing project](#)

Studio provides several options for adding a data set to a project.

### [Loading the full data set in a project](#)

After you add a data set to a project, you can choose to load the full data set into the project. This is useful for comprehensive data analysis and building a BDD application. Remember that without this full data load, Studio displays a sampled data set of approximately 1 million records if the full data set is larger than 1 million records.

### [Configuring a project data set for incremental updates](#)

To prepare your data set for an incremental update, you must first load a full data set and provide a record identifier so that Studio can determine the incremental changes to apply during the update.

### [Converting a project to a BDD Application](#)

A *BDD Application* is a logical designation of a project that contains one or more data sets that you transform, update, and maintain for long-lasting data analysis, reporting, and sharing. In this release, you can think of an application as a production-quality project. This topic describes the high-level workflow of converting a project into a BDD application and provides pointers to more detailed instructions as necessary.

### [Removing a data set from a project](#)

You can have any number of data sets associated with a project. If a data set is no longer useful in a particular project, you can remove the data set from that project. This removal affects only the active project. It does not affect any other projects that a data set may be added to.

## Adding a data set to an existing project

Studio provides several options for adding a data set to a project.

### **From the Catalog**

On the Catalog, the data set details panel includes an option to add the data set to a project.

Click **Add to Project**, then select the project.

### From Explore

When exploring a data set from the Catalog, the page header includes an option to add the data set to a project.

Click **Add to Project**, then select the project.

### From within a project

From the **Explore** or **Transform** pages of a project, to add a data set to the project, click the + icon to the right of the data set tabs.

You can select an existing data set, or create a new data set.

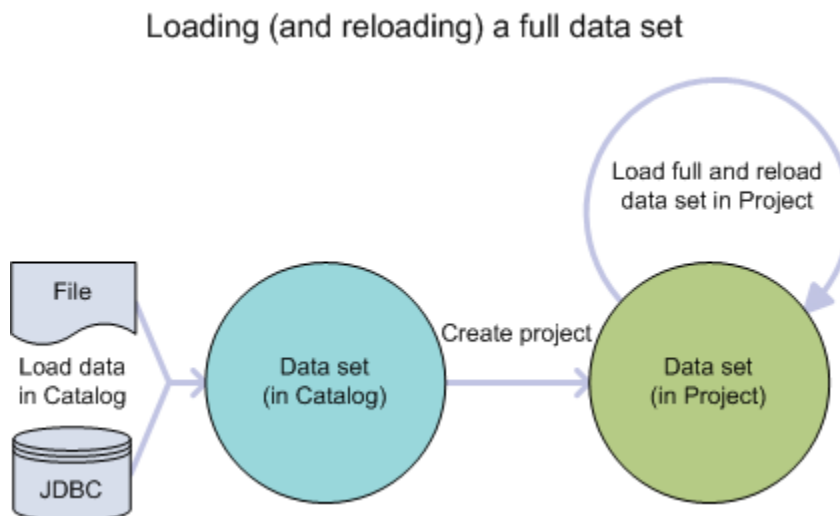
## Loading the full data set in a project

After you add a data set to a project, you can choose to load the full data set into the project. This is useful for comprehensive data analysis and building a BDD application. Remember that without this full data load, Studio displays a sampled data set of approximately 1 million records if the full data set is larger than 1 million records.

The **Load Full Data Set** option behaves as follows:

- It loads all records stored in the Hive table for a data set. This includes any table updates performed by a system administrator. The full data load happens during the initial full data load only. After the first full data load, the action changes to **Reload Data Set** and you can reload the data set any number of times.
- It increases a sampled data set up to the full size of the data set.
- If the project contains a transformation script that you have committed, then Studio runs that script against the full data set. This way, all transformations apply to the full data set in the project.

The following diagram shows the workflow of loading a full data set into a project:



In this workflow, the following actions take place:

1. You load a data set from a file or JDBC data source. This is the initial load of the data set into the Catalog.

2. You can then explore the data and add it to a project to use **Transform** and **Discover**.
3. You load the full data set and reload the data set as necessary.

Notice that loading the full data set affects only the data set in a specific project: it does not affect the data set as it displays in the Catalog.

To check if a data set has already been fully loaded into a project, go to the **Data Set Manager** page and see if the **Record Data Volume** property indicates `Full data set is loaded`.

To load the full data set in a project:

1. From the Configuration Options menu, select **Project Settings**.
2. Select **Data Set Manager** and expand the options next to the data set name.
3. Select **Load Full Data Set**.
4. In the confirmation dialog, select **Load Full Data Set** again.
5. Return to **Explore** or **Transform** to monitor the progress of the load operation.

Depending on the size of the data set, the load may take some time to complete. After the operation finishes, you can check the **Data Volume** information to see that the `Full data set is loaded` and the **Explore** header indicates the full record count.

## Configuring a project data set for incremental updates

To prepare your data set for an incremental update, you must first load a full data set and provide a record identifier so that Studio can determine the incremental changes to apply during the update.

An incremental update is performed with DP CLI. However, in Studio, you prepare your data set for this type of update so that you can then run an incremental update against a project data set with DP CLI.

An **incremental update** lets you add newer data to an existing data set in project, without removing already loaded data. It is most useful when you keep already loaded data, but would like to continue adding new data.

---

---

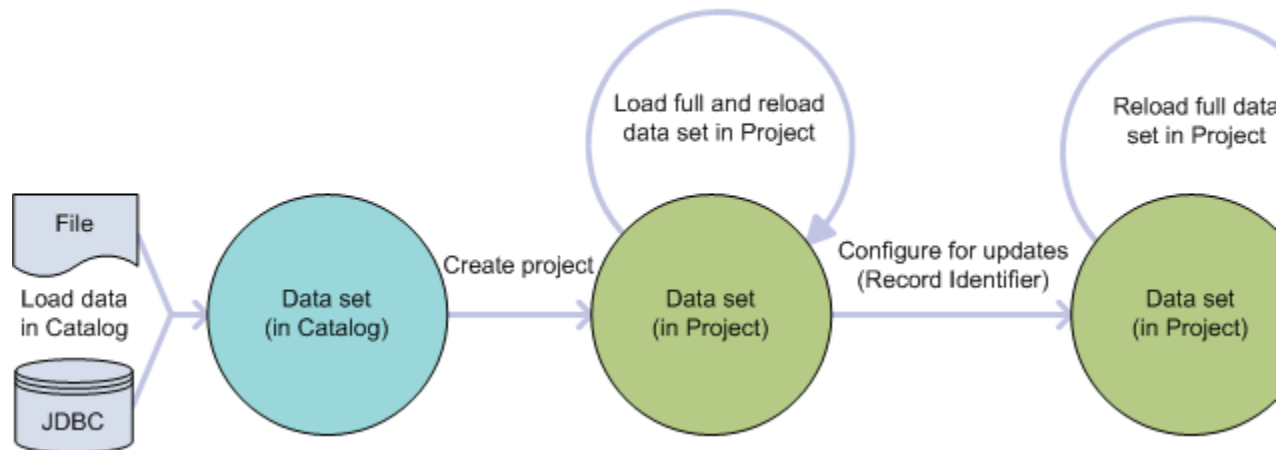
**Note:** You cannot run an incremental update from Studio. For details and a diagram of the incremental update workflow, see the *Getting Started Guide*. For full information on how to run an incremental update with DP CLI, see the *Data Processing Guide*.

---

---

The following diagram shows the **Configure for Updates** action. This action prepares your data set for running incremental updates with DP CLI:

## Configuring a data set for updates



In this diagram, from left to right, the following actions take place: you load a data set into Studio from a file or a JDBC source. Next, you add a data set to a project and load a data set in full. Now you can run **Configure for Updates**. Notice that you can only run **Configure for Updates** for a data set that is already in project and has already been loaded in full.

The action **Configure for Updates** performs two tasks: it loads a full data set (by re-running the operation for loading a full data set), and lets you configure a record identifier. The record identifier is then used by Data Processing CLI when it runs an incremental update for the data set.

The **record identifier** must be unique enough to determine the delta between records in the Hive table and records in the project data set. Practically speaking, this means that in some project data sets you might have to provide a record identifier that is the combination of several attributes. In other projects, you might have a single attribute that works as a unique record identifier without any additional combination.

Studio helps you identify which attribute in your data set is a good candidate for a record identifier. When you select an attribute from the list as a record identifier, Studio calculates the percentage of records in your data set that have unique values for the combination of attributes. The uniqueness score must be 100% or the Data Processing workflow will fail and return an exception to Studio.

To check if a data set already has a record identifier defined in Studio, go to the **Data Set Manager** page and see if there is a **Record Identifiers** property specified.

To configure a project data set for incremental updates:

1. From the Configuration Options menu, select **Project Settings**.
2. Select **Data Set Manager** and expand options next to the data set name.
3. Select **Configure for Updates**.
4. From the list, select an attribute for the data set. If the **Key Uniqueness** is not 100% for the attribute, click + **Attribute** to add another attribute and repeat to improve uniqueness.

Studio combines these attributes into a new unique record identifier for the data set.



## 5. Click **Configure for Updates**.

After you click this option, a **Load Full Data Set** operation starts automatically as a background process.

At this point you can schedule and run incremental updates using the `IncrementalUpdate` command of the Data Processing CLI. For details, see the *Data Processing Guide*.

## Converting a project to a BDD Application

A *BDD Application* is a logical designation of a project that contains one or more data sets that you transform, update, and maintain for long-lasting data analysis, reporting, and sharing. In this release, you can think of an application as a production-quality project. This topic describes the high-level workflow of converting a project into a BDD application and provides pointers to more detailed instructions as necessary.

Before you convert a project to an application there are a number of tasks that you must have already performed. These tasks are common to any project and are not specific to building a BDD Application:

1. Create a data set. You can do this by uploading your source data in Studio or by creating a Hive table and running a data processing workflow.
2. Add one or more data sets to a project. For details, see [Adding a data set to an existing project](#).
3. Set access permissions on data sets and projects. For details, see [Managing Access to Projects](#).
4. If necessary, transform the project data set to clean up the attributes and attribute values. This might include reformatting attributes, changing data types, splitting attributes into more specific attributes, creating new attributes, and so on. For details, see [About the Transform page user interface](#).
5. Build components to visualize the data in Discover that you want to preserve in a more productized way.

To convert a project to a BDD Application:

1. Optionally, if you created one or more data sets by uploading source data into Studio, you should change the source of each data set to point to a new Hive table. This step is not required if data set is based on a table created directly in Hive. (This step is necessary for data encoding reasons.)
  - a. In Studio, find the data set's current Hive table name by clicking **Show Data Set Properties**. The name is of the form `default.my_uploaded_data`.
  - b. Browse to the Hive query editor for your Hadoop distribution. For example, in a Cloudera environment, this is the Hive Query Editor in Hue.
  - c. Run the following Hive command and specify the data set's Hive table name:
 

```
SHOW CREATE TABLE `default.my_uploaded_data`
```
  - d. Copy the resulting create table command and modify it in a text editor. Modify the table definition to point to the new location in HDFS of the data set's source files. Specify the column data types using Hive types; by default, the table description shows all attributes as strings. Also, change the table encoding

properties like Row Format, Storage Format, and SerDe as necessary to match the source file encoding. Optionally change other table properties as necessary like `dataSetDisplayName` or `comment`.

- e. Run drop table to remove the existing Hive table.

```
DROP TABLE `default.my_uploaded_data`
```

- f. Run the new create table command to re-create the table with the same name and columns but new source file location and encoding.
- g. Test that the new table definition is correct by browsing in Hue to Metastore Tables. Find the re-created Hive table and click on Sample. Optionally run a Hive query to count the number of rows in the table to confirm the expected data set size.
- h. Repeat these steps for each data set in the project that has a **Data source type** value of Excel, Delimited, or JDBC.

**Important:** Now that you have a new Hive table that has the same name as the old Hive table, do not use the **Actions > Reload Data set** feature on the quick look for this specific data set in the Catalog. The **Reload Data set** feature invokes the file upload wizard and allows you to overwrite the production Hive table that you just created.

2. Optionally, specify a record identifier for the project if you plan to run incremental updates to the data.

For details, see [Configuring a project data set for incremental updates](#) .

3. Load the full data set into the project.

This step provides all the source data so that visualization and analysis includes the full record set. For details, see [Loading the full data set in a project](#).

4. Schedule updates to data sets using the Data Processing CLI and cron jobs.

For details, see "DP CLI configuration" and "DP CLI cron job" in the *Data Processing Guide*.

## Removing a data set from a project

You can have any number of data sets associated with a project. If a data set is no longer useful in a particular project, you can remove the data set from that project. This removal affects only the active project. It does not affect any other projects that a data set may be added to.

To remove a data set from a project:

1. Select either **Explore** or **Transform**.
2. In the footer, locate the data set you want to remove from the project
3. Click the remove icon for the data set.

For example:



4. Alternatively, you can also remove a data set from a project on the **Data Set Manager** page:
  - a. Select the **Project Settings > Data Set Manager** page.
  - b. Expand the data set to display its information and select **Remove from Project**.



---

## Linking Project Data Sets

You can link together project data sets that have corresponding attributes.

### [About data set links](#)

You create a link between two data sets by selecting one or more attributes from one data set and associating them with corresponding attributes from another data set. These corresponding pairs of attributes are called a **link key**.

### [Creating a data set link](#)

You create a link between two data sets by setting one or more corresponding attributes from each as the link key. You can do this from either the **Data Set Manager** or the **Data Set Relationships** tab of the data set details footer panel.

### [Editing a link key](#)

You can edit a link key by adding or removing attribute pairs.

### [Managing aggregation methods for a linked view](#)

When you first create a data set link, the attributes in the linked view are aggregated by the default aggregation methods set for their data types. However, you can change the aggregation methods used for individual attributes from either the **Data Set Manager** or the **Data Set Relationships** tab of the data set details footer panel.

### [Controlling automatic filtering between linked data sets](#)

Automatic filtering of linked data sets is enabled by default, but you can disable it for individual project pages from the **Page Data Settings** panel.

## About data set links

You create a link between two data sets by selecting one or more attributes from one data set and associating them with corresponding attributes from another data set. These corresponding pairs of attributes are called a **link key**.

For each data set that is linked to others, Studio creates a **linked view**, which contains attributes from that data set, as well as those it is linked to. For more information, see [About linked views](#).

While you cannot edit the linked views themselves, you can modify the data that appears in them by adding or removing link key attributes, changing the methods used to aggregate attributes within the view, and enabling or disabling automatic filtering.

If you run updates for each of the linked data sets, such as a full data set load or an incremental update, then data set links are preserved and continue to work.

## Creating a data set link

You create a link between two data sets by setting one or more corresponding attributes from each as the link key. You can do this from either the **Data Set Manager** or the **Data Set Relationships** tab of the data set details footer panel.

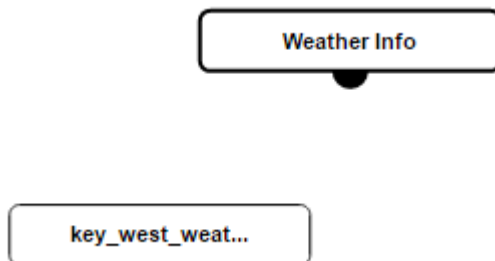
To create a data set link:

1. Open either the **Data Set Manager** or the **Data Set Relationships** panel:
  - To open the **Data Set Manager**, click **Project Settings**, then select **Data Set Manager**.
  - To open the **Data Set Relationships** panel, click the data set details icon in the footer, then select the **Data Set Relationships** tab.

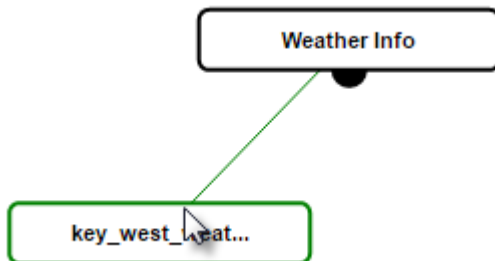
A diagram of the data sets in your project displays.

2. In the diagram, click one of the data sets to select it.

A handle appears below the selected data set.



3. Click and drag the handle to the data set you want to link to and release the mouse when the second data set becomes highlighted in green.



The **Edit Link** dialog opens.

**Edit link**

A data set link is defined by a link key, corresponding attributes that share common values between the two data link key may be compound, consisting of two or more attribute pairs, such as First Name and Last Name.

Enable link

To create a link key, select corresponding attributes from each data set.

**Weather Info** link to **key\_west\_weather**

Select attribute pair

Cancel

4. Verify that the **Enable link** check box is selected.

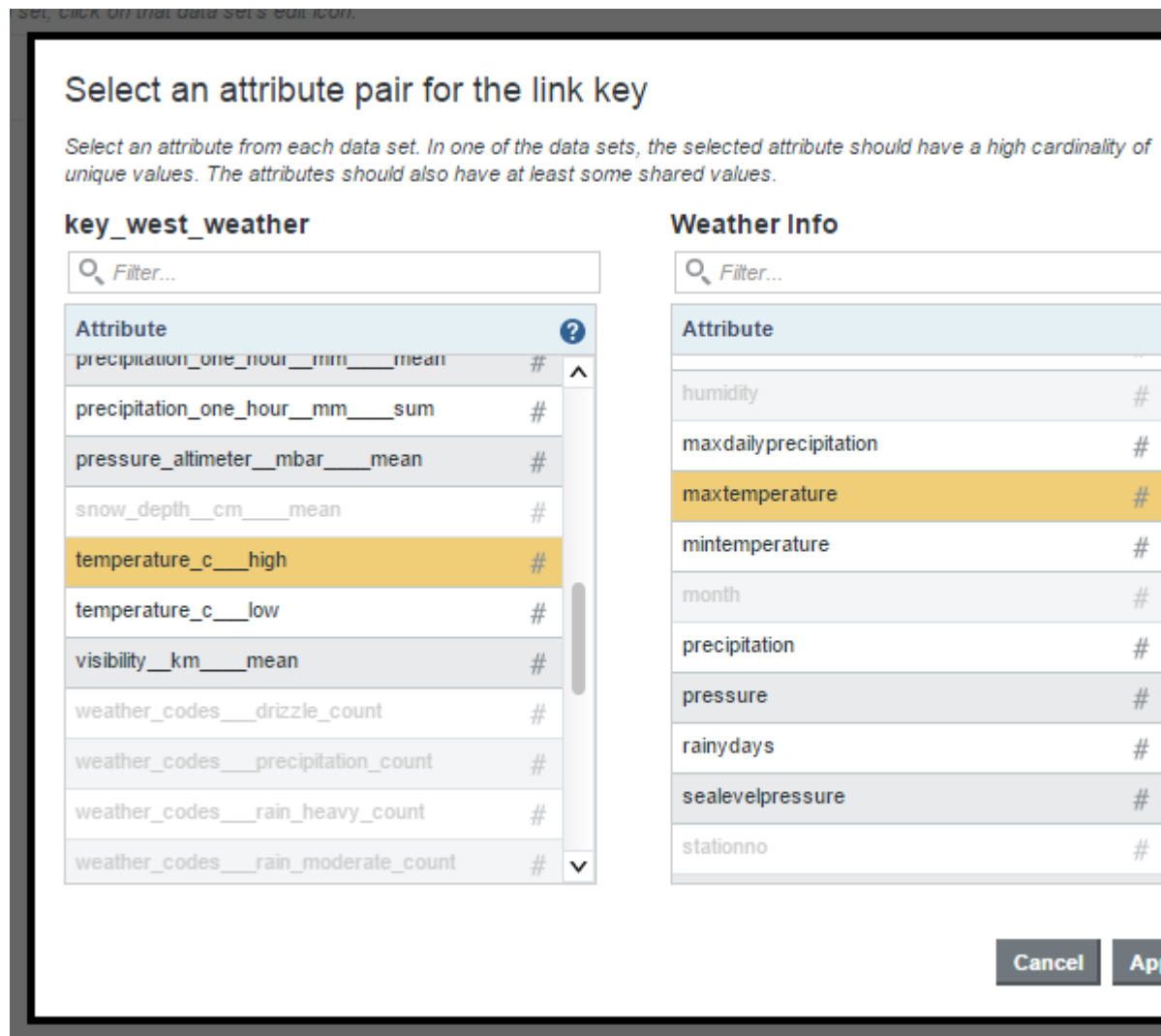
Deselecting this disables the link.

5. Add at least one attribute pair:

- a. Click **Select attribute pair**.

- b. In the attribute pair selection dialog, select an attribute from each data set.

When you select an attribute from one list, the other is automatically filtered for attributes of the same data type.



c. Click **Apply**.

The **Edit link** dialog is updated with the selected attribute pair.



### Edit link

*A data set link is defined by a link key, corresponding attributes that share common values between the two data link key may be compound, consisting of two or more attribute pairs, such as First Name and Last Name.*

Enable link

*To create a link key, select corresponding attributes from each data set.*

**Weather Info**

link to

**key\_west\_weather**

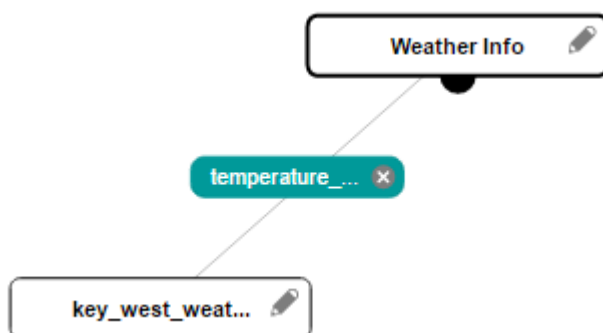
=

**Add attribute pair**

**Cancel**

6. Click **Save**.

The diagram displays the two linked data sets with the link key between them.



## Editing a link key

You can edit a link key by adding or removing attribute pairs.

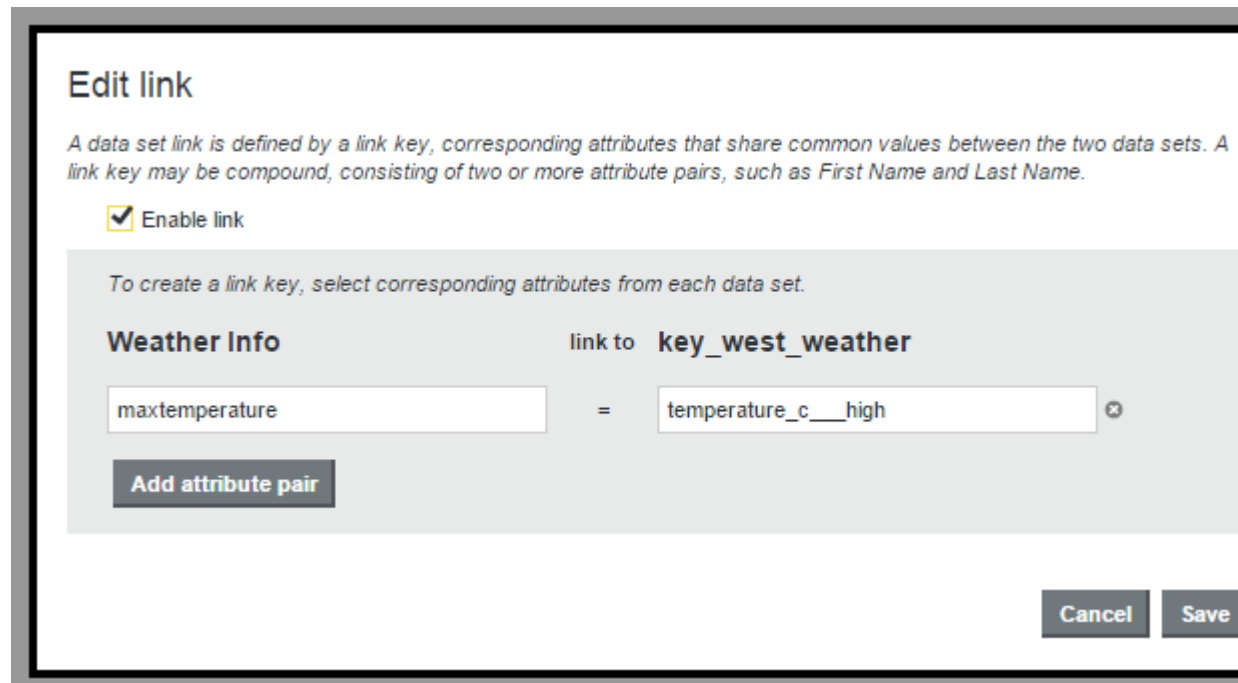
To edit the link key for a data set link:

1. Open either the **Data Set Manager** or the **Data Set Relationships** panel:
  - To open the **Data Set Manager**, click **Configuration Options > Project Settings**, then select **Data Set Manager**.
  - To open the **Data Set Relationships** panel, click the data set details icon in the footer, then select the **Data Set Relationships** tab.

A diagram of the data sets and linked views in your project displays.

2. Click the link you want to edit.

The **Edit link** dialog displays.



3. Make the desired changes:
  - Click **Add attribute pair** to add a new attribute pair.
  - Click the delete icon next to an attribute pair to remove it.
4. When you are satisfied with your changes, click **Save**.

The combined views in your project are updated to reflect your changes.

## Managing aggregation methods for a linked view

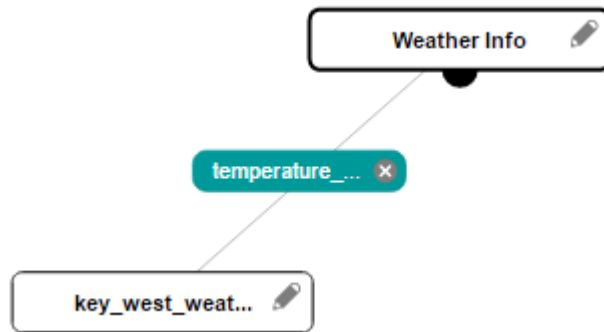
When you first create a data set link, the attributes in the linked view are aggregated by the default aggregation methods set for their data types. However, you can change the aggregation methods used for individual attributes from either the **Data Set Manager** or the **Data Set Relationships** tab of the data set details footer panel.

For more information on aggregation methods, see [Selecting the available and default aggregation methods for an attribute](#).

To manage the aggregation methods for a linked view:

1. Open either the **Data Set Manager** or the **Data Set Relationships** panel:
  - To open the **Data Set Manager**, click **Configuration Options > Project Settings**, then select **Data Set Manager**.
  - To open the **Data Set Relationships** panel, click the data set details icon in the footer, then select the **Data Set Relationships** tab.

A diagram of the data sets and linked views in your project displays.



2. Click the pencil icon for the view you want to modify.

The manage aggregation dialog displays.

If the selected view contains more than one data set, the dialog contains a drop down menu for selecting the data set whose aggregation methods you want to modify.

The dialog also contains a table that displays sample values from the selected data set. The two leftmost columns contain values from the Record ID and Link Key attributes of the selected view.

The remaining columns contain attributes and values from the selected data set. The aggregation method used for each attribute is listed below the attribute's name.

### Manage aggregation for the key\_west\_weather view

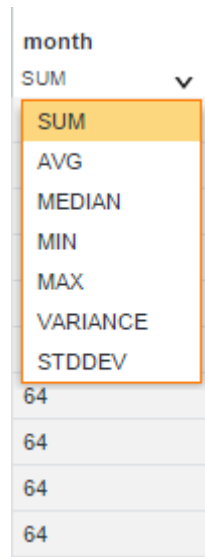
When a data set is linked to other data sets, a custom view is automatically created. For the data sets that are linked to this view, you can configure the aggregation for the data set attributes.

key_west_weather data (730 records)		Weather Info data			
PRIMARY_KEY Record ID	temperature_... Link Key	month SUM ▼	rainydays SUM ▼	vaporpressure SUM ▼	precipitation SUM ▼
0-0-210	33.9	43	32.2		373.1
0-1-177	33.9	43	32.2		373.1
0-0-209	33.9	43	32.2		373.1
0-0-203	33.9	43	32.2		373.1
0-1-176	33.3	64	51.8	11.4	638.4
0-0-233	33.3	64	51.8	11.4	638.4
0-0-208	33.3	64	51.8	11.4	638.4
0-0-232	33.3	64	51.8	11.4	638.4
0-1-163	33.3	64	51.8	11.4	638.4
0-0-226	33.3	64	51.8	11.4	638.4

3. If the selected view is linked to multiple data sets, select the name of the data set you want to modify from the drop down menu.
4. Click the down arrow next to the aggregation method you want to modify.

Note that this will not be available for String attributes.

A drop down menu displays, listing the available aggregation methods for that attribute.



5. Select one of the available aggregation methods, then click **Save**.

The dialog is updated to display the new values for the modified attribute.

**Manage aggregation for the key\_west\_weather view**

✓ Your aggregations were successfully modified

When a data set is linked to other data sets, a custom view is automatically created. For the data sets that are linked to this view, you can configure the aggregation for the data set attributes.

key_west_weather data (730 records)		Weather Info data			
PRIMARY_KEY Record ID	temperature_... Link Key	month MEDIAN	rainydays SUM	vaporpressure SUM	precipitation SUM
0-0-350	19.4	4	147.4	133.6	1,763
0-0-345	19.4	4	147.4	133.6	1,763
0-1-17	19.4	4	147.4	133.6	1,763
0-1-364	22.8	5	133.9	136.8	1,789.1
0-1-359	22.8	5	133.9	136.8	1,789.1
0-1-6	22.8	5	133.9	136.8	1,789.1
0-1-355	22.8	5	133.9	136.8	1,789.1
0-1-35	18.3	5	98.7	103.5	1,370.7
0-0-76	22.8	5	133.9	136.8	1,789.1
0-1-1	22.8	5	133.9	136.8	1,789.1

- When you are finished, click **Close**.

## Controlling automatic filtering between linked data sets

Automatic filtering of linked data sets is enabled by default, but you can disable it for individual project pages from the **Page Data Settings** panel.

When automatic filtering is enabled for a pair of linked data sets, you can filter both data sets simultaneously using attributes from their link key.

For example, two data sets called Sales and Products are linked based on product SKU and have automatic filtering enabled. If you filter Sales for a specific range of SKU values, Products is also filtered for the same range of values automatically, and vice versa.

To control automatic filtering for a pair of linked data sets:

- On the project page for the linked view, click the pencil icon in the right bar.

The **Page Settings** panel opens.

- Click **Page Data Settings**.

The panel displays the data settings for the current page. By default, the **All <data set> records** option is selected

3. Select the **Disable data set filtering on this page**.

**PAGE SETTINGS** ✕

▼ **PAGE DATA SETTINGS**

**Primary Data Set**

key\_west\_weather ▼

**BEHAVIOR FOR LINKED DATA SETS**  
Select a filter and refinement behavior for linked data sets on this page

All key\_west\_weather records  
Primary data set is not pre-filtered by secondary data sets

Only key\_west\_weather records with linked Weather Info  
Primary data set is pre-filtered by linked data sets

**Disable data set filtering on this page**  
Data sets do not filter or refine each other on this page

key\_west\_w... — — Weather Info

▶ **PAGE TEMPLATES**

The page is updated to reflect the change. Refinements applied to both data sets are now only applied to their original data sets.





---

## Defining Project Data Set Views

You can use the **Views** page to configure a set of views for project data. Views are used by components to determine the data to display.

### About views

A view is a logical collection of information that is derived from the records in project data sets.

### About base views

Each data set is automatically configured with a base view, which contains all of the physical attributes from the data set. The base view uses the name and description from the data set.

### About linked views

Linked views are created automatically when you link two data sets.

### Displaying the list of views for a project

On the project settings page, the **Data Views** page allows you to view and edit the views for the project data.

### Displaying details for a view

To display the details for a view, in the views list, click the view name.

### Creating views

From the **Views** page, you can either create a new empty view, or make a copy of an existing view.

### Configuring a view

From the **Views** page, you can edit each view's display name, description, definition, attributes, and predefined metrics.

### Previewing the content of a view

From the **Views** page, you can display a preview list of the records in a view.

### Deciding whether to publish a custom view

For custom views, the **Publish** check box controls whether the view is published. Only published views are available to use in a project.

### Deleting a view

From the **Views** page, you can delete custom views. You cannot delete base views or linked views. If you delete a view, any component that uses that view can no longer display correctly. Also, any other view whose definition includes references to this view becomes invalid.

## About views

A view is a logical collection of information that is derived from the records in project data sets.

For example, the original records in a data set may represent individual sales transactions. However, from those transaction records, you could derive:

- A list of products
- A list of customers
- A summary of sales data grouped by year

Each of those could be a view.

All views are made up of attributes. Some attributes may be the physical attributes from a data set, while others may be calculated from the original data.

For example, for a list of products derived from a list of sales transactions, the product name comes directly from the data, while the total sales for that product is calculated from the individual sales records.

Views can optionally have predefined metrics, which are values calculated from view attributes using EQL expressions.

For example, for sales transaction data, a predefined metric might be the profit margin.

## About base views

Each data set is automatically configured with a base view, which contains all of the physical attributes from the data set. The base view uses the name and description from the data set.

While you cannot change the definition of a base view, or add predefined metrics to a base view, you can configure base view attributes.

## About linked views

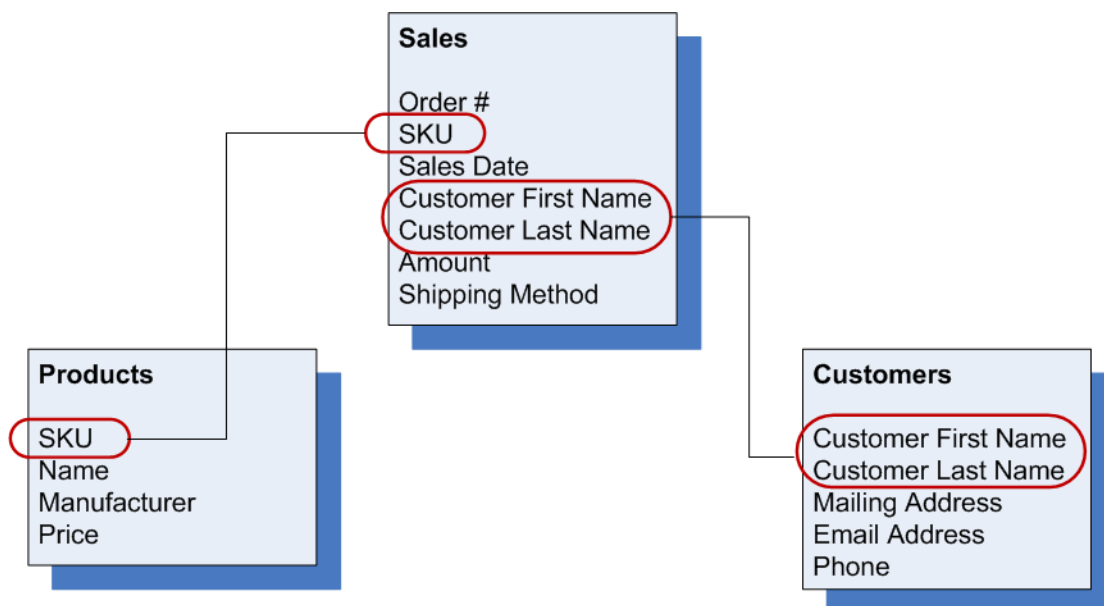
Linked views are created automatically when you link two data sets.

For each data set that has links to other data sets, Studio creates a combined linked view that contains:

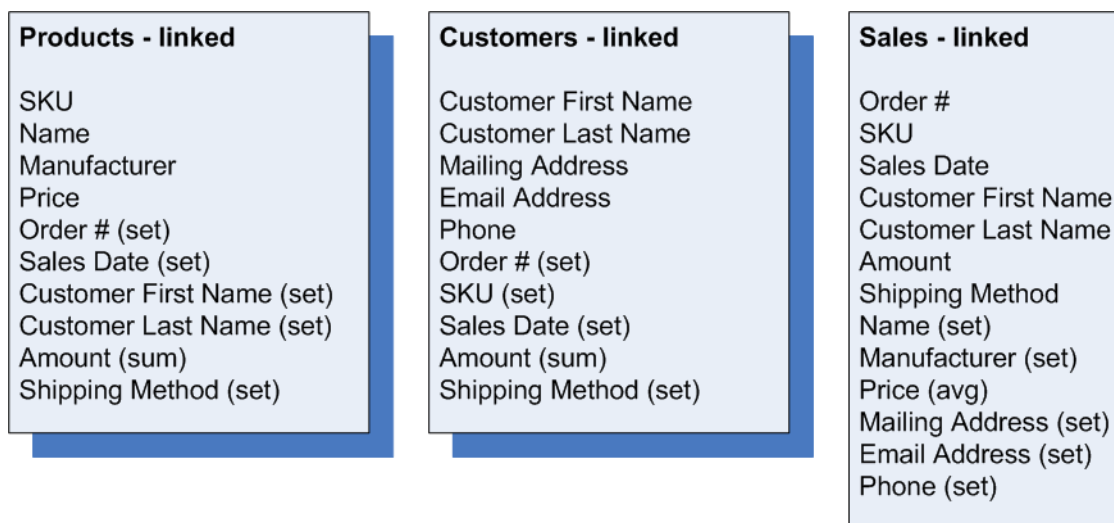
- All of the attribute for that data set
- The attributes from the other data sets, aggregated based on the link key

Linked views are read-only, and do not allow for any configuration or customization. You also cannot copy a linked view.

So for the following links among the Products, Customers, and Sales data sets:



Studio creates the following linked views:



Linked View	Description
Products - linked	<p>Contains the Products records with Sales attributes added in. The Sales attributes are aggregated by product SKU.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>The sales amount (from Sales) for each product is the total for all of the sales for the product.</li> <li>The customer name (also from Sales) for each product is a list of customers that purchased the product.</li> </ul>

Linked View	Description
Customers - linked	<p>Contains the Customers records with Sales attributes added in. The Sales attributes are aggregated by the Customer First Name and Last Name.</p> <p>For example:</p> <ul style="list-style-type: none"><li>• The sales amount (from Sales) for each customer is total for all sales to that customer.</li><li>• The product SKU (from Sales) for each customer is a list of SKUs for products purchased by that customer.</li></ul>
Sales - linked	<p>Contains the Sales records with the Products and Customers attributes added in. The Products information is aggregated by product SKU, and the Customers information is aggregated by the Customer First Name and Customer Last Name.</p> <p>In this case, because each sales record has a single product and customer, there is still only a single value for each product and customer field.</p>

---

## Displaying the list of views for a project

On the project settings page, the **Data Views** page allows you to view and edit the views for the project data.

To display the list of views for a project:

1. On the project menu bar, click the settings icon.
2. In the project settings menu, click **Data Views**.

On the **Data Views** page, the **Data Views** list is populated with the views for the project. For each view, the list shows the display name, key, type, source data sets, and identifying attributes for the view records. The first base view is selected by default.

## Data Views

Configure the data views for your project. Select a view from the list at the top to see and configure its details.

### Views

Publish	View Name	View Key	Type	D
✓	Product	edp_cli_edp_39361033-...	Base	P
✓	Sales	edp_cli_edp_7a6bae5d-...	Base	S
✓	Product - linked	edp_cli_edp_39361033-...	Linked	S
✓	Sales - linked	edp_cli_edp_7a6bae5d-...	Linked	P

Selected View: Product (edp\_cli\_edp\_39361033-51e8-4e06-810e-914054a1046c\_busgl0307\_wls\_ENTITIES\_10850)

Display name: Product ?

Default view description: ?

Record Identifier(s): PRIMARY\_KEY

▼ Attributes

Display Name	Attribute Key	Refinable	Multi-Value	Data Type
product_catego... ↻	product_categoryn...	✓	—	“ String
product_class ↻	product_class	✓	—	“ String
product_color ↻	product_color	✓	—	“ String
product_daysto... ↻	product_daystoma...	✓	—	# Number (long)

## Displaying details for a view

To display the details for a view, in the views list, click the view name.

The view details section is populated with

- Whether the view is published.  
If the view is not published, then it is not available to use for components. You cannot publish views that do not have a valid view definition.
- View name
- View description
- Identifying attributes for the view
- Attributes list.

The attributes list does not display for views that do not have a valid view definition.

- Predefined metrics list.

The predefined metrics list does not display for views that do not have a valid view definition.

✓	Sales	edp_cli_edp_7a6bae5d-...	Base	Sales
✓	Product - linked	edp_cli_edp_39361033-...	Linked	Sales, Product
✓	Sales - linked	edp_cli_edp_7a6bae5d-...	Linked	Product, Sales

Selected View: Sales (edp\_cli\_edp\_7a6bae5d-4109-41bd-962d-4fcf2a1d8061\_busgl0307\_wls\_ENTITIES\_10850) View Definition

Display name: Sales ?  
 Default view description: ?  
 Record Identifier(s): PRIMARY\_KEY

▼ **Attributes**

Display Name	Attribute Key	Refinable	Multi-Value	Data Type	Standard F
sales_activityd... ↻	sales_activitydates	✓	✓	📅 Date/Time ✎	DateTime
sales_calendar... ↻	sales_calendarqu...	✓	—	# Number (long)	Number
sales_calendar... ↻	sales_calendarse...	✓	—	# Number (long)	Number
sales_calendar... ↻	sales_calendaryear	✓	—	# Number (long)	Number
sales_carriertra... ↻	sales_carriertracki...	✓	—	“ String	(string)
sales_custome... ↻	sales_customerpo...	✓	—	“ String	(string)
sales_daynam... ↻	sales_daynameof...	✓	—	“ String	(string)

## Creating views

From the **Views** page, you can either create a new empty view, or make a copy of an existing view.

### Creating a new view

One option for creating a new view is to create a completely new view.

### Copying an existing data view

You can create a data view by copying another data view. You cannot copy a linked data view. The new view initially has the same definition, attributes, and metrics as the original view. Studio also copies the attribute groups.

## Creating a new view

One option for creating a new view is to create a completely new view.

To create a new view:

1. Click **Configuration Options > Project Settings**, then navigate to the **Data Views** page.
2. On the views list, click **+ View**.  
The **Create View Definition** dialog displays.
3. In the **Display Name** field, type a name for the view.
4. In the **Description** field, type a description of the view.
5. In the EQL text area, enter the view definition.

For details on defining a view definition, see [Guidelines for using EQL to define a view](#).

To the right of the text area is the **Available Attributes** list, which contains the list of base attributes. You can show or hide this list.

You can use the attributes list as a reference as you compose the view definition. To add an attribute to the definition, you can also double-click or drag and drop the attribute from the list. Studio automatically adds quotes around the inserted attribute, in case the attribute name contains reserved characters in EQL.

Note that if you include a base attribute that has locale-specific versions, then for those locale-specific versions to be available in the new view, you must manually include them in the view definition.

6. By default, when you save a new view, it is automatically published. To not publish the new view automatically, deselect the **Publish when saving** check box.
7. To validate the definition, click **Validate**.

If the definition is not valid, then Studio displays error messages to help you troubleshoot the definition.

8. To save the new view, click **Save**.

If the view definition is valid, then the view is saved and can be published.

If the view definition is not valid, then you are prompted to confirm that you want to save an invalid view. If you save the view, it is flagged as invalid and cannot be published.

## Copying an existing data view

You can create a data view by copying another data view. You cannot copy a linked data view. The new view initially has the same definition, attributes, and metrics as the original view. Studio also copies the attribute groups.

To copy an existing:

1. Select **Project Settings**.
2. Select **Data Views**.
3. Identify the view you want to copy.
4. In the **Actions** column, click the copy icon for the data view you want to copy.

The **Create View Definition** dialog displays. By default:

- The view display name is "Copy of *<copiedViewName>*".
- The view key in the view definition is set to "*<copiedViewKeyName>*Copy".
- The view description and the rest of the view definition are identical to the copied view.



## Create View Definition

To define a view, please enter the name, description, and an EQL definition.

Display Name:

Description:

Enter the EQL to define the view [?](#)

```

1 DEFINE "edp_cli_edp_7a6bae5d-4109-41bd-962d-4fcf2a1d6061_busgl0307_wls_ENTI
2 ARB("sales_resellerlatlongb_geo_subregion") AS "sales_resellerlatlongb_geo_s
3 ARB("sales_reseller_latlongd_geo_subregion") AS "sales_reseller_latlongd_geo
4 ARB("sales_resellerlatlong_geo_city") AS "sales_resellerlatlong_geo_city",
5 ARB("sales_resellerstateprovincename") AS "sales_resellerstateprovincename",
6 ARB("sales_daynameofweek") AS "sales_daynameofweek",
7 ARB("sales_resellerlatlong_geo_postcode") AS "sales_resellerlatlong_geo_post
8 ARB("sales_totalproductcost") AS "sales_totalproductcost",
9 ARB("sales_reseller_latlongc_geo_country") AS "sales_reseller_latlongc_geo_c
10 ARB("sales_resellerkey") AS "sales_resellerkey",
11 ARB("sales_reseller_latlongd") AS "sales_reseller_latlongd",
12 ARB("sales_reseller_latlongc") AS "sales_reseller_latlongc",
13 SET_UNIONS("sales_surveyresponse_key_phrases") AS "sales_surveyresponse_key
14 ARB("sales_fiscalyear") AS "sales_fiscalyear",
15 ARB("sales_employeefullname") AS "sales_employeefullname",
16 ARB("sales_reseller_latlongd_geo_country") AS "sales_reseller_latlongd_geo_c
17 ARB("sales_resellerlatlongb") AS "sales_resellerlatlongb",
18 ARB("sales_reseller_latlongc_geo_region") AS "sales_reseller_latlongc_geo_re
19 ARB("sales_weeknumberofyear") AS "sales_weeknumberofyear",
20 ARB("sales_customerponumer") AS "sales_customerponumer",
21 ARB("sales_calendarsemester") AS "sales_calendarsemester",
22 ARB("sales_resellerlatlongb_geo_city") AS "sales_resellerlatlongb_geo_city",
23 ARB("sales_reseller_latlongd_geo_city") AS "sales_reseller_latlongd_geo_city",
24 ARB("sales_daynumberofmonth") AS "sales_daynumberofmonth",
25 ARB("sales_territoryregion") AS "sales_territoryregion",
26 ARB("sales_territorycountry") AS "sales_territorycountry",
27 ARB("sales_productname") AS "sales_productname",
28 ARB("sales_territorygroup") AS "sales_territorygroup",
29 ARB("sales_fiscalquarter") AS "sales_fiscalquarter",
30 ARB("sales_resellerlatlongb_geo_country") AS "sales_resellerlatlongb_geo_co
31 ARB("sales_employeelastname") AS "sales_employeelastname",
32 ARB("sales_resellerstateprovincecode") AS "sales_resellerstateprovincecode",
33 ARB("sales_reseller_latlongc_geo_postcode") AS "sales_reseller_latlongc_geo
34 ARB("sales_monthname") AS "sales_monthname",
35 ARB("sales_productstandardcost") AS "sales_productstandardcost",
36 ARB("sales_resellercountryregioncode") AS "sales_resellercountryregioncode",
37 ARB("sales_surveyresponse") AS "sales_surveyresponse",
38 ARB("sales_resellerlatlongb_geo_postcode") AS "sales_resellerlatlongb_geo_po
39 SET_UNIONS("sales_activitydates") AS "sales_activitydates",
40 ARB("sales_resellername") AS "sales_resellername",
41 ARB("sales_resellercountryregionname") AS "sales_resellercountryregionname",
42 < >

```

Publish when saving

- Update the view name, description, view key, and definition as needed.

Note that you cannot change the view key after you save the new view, so make sure you update the value before saving.

Remember also that the view key must be NC-Name compliant.

- By default, when you save a new view, it is automatically published. To not publish the new view automatically, deselect the **Publish when saving** check box.
- To validate the new view, click **Validate**.
- To save the new view, click **Save**.

## Configuring a view

From the **Views** page, you can edit each view's display name, description, definition, attributes, and predefined metrics.

### Editing and localizing the view display name and description

On the **Data Views** page, when you select a view from the view list, the view details include the **Display name** and **Default view description** fields.

### Editing the view definition

From the **Views** page, after you create a custom view, you can update the view EQL definition. Note that if you change the definition, you could affect components that are using the view, as well as other views that may use attributes from this view.

### Guidelines for using EQL to define a view

The view definition EQL defines the content of the view. When defining a view, you need to keep the following guidelines in mind. For complete details on EQL syntax, see the *EQL Reference*.

### Configuring the view attributes

For each attribute in the view, you can configure the display name and aggregation methods.

### Defining predefined metrics for a view

Custom views can also have predefined metrics, which are values calculated from the view attributes.

## Editing and localizing the view display name and description

On the **Data Views** page, when you select a view from the view list, the view details include the **Display name** and **Default view description** fields.

Selected View: Sales (edp\_cli\_edp\_7a6bae5d-4109-41bd-962d-4fcf2a1d6061\_busgl0307\_wls\_ENTITIES\_10850) View Definition

Display name:	Sales <span style="color: #0070c0; font-size: 1.2em;">?</span>
Default view description:	<span style="color: #0070c0; font-size: 1.2em;">?</span>
Record Identifier(s):	PRIMARY_KEY

The display name and description are used when showing the list of available views to assign to a component.

For custom views, you can create localized versions of the name and description.

Base views use the name and description of the data set. To edit or localize the name and description of a base view, you edit the data set.

Linked views use automatically generated names and descriptions.

To edit and localize the view display name and description for a custom data view:

1. Copy a base view if you haven't already. This is your custom data view.
2. To edit the default display name and description:
  - a. In the **Display name** field, type the display name for the view.
  - b. In the **Default view description** field, type a description of the view.
3. To localize the view display name and description, click the localize icon next to either field.
4. To localize the view display name:
  - a. On the **Configure Locale Options** dialog, click the **Display Name** tab.

The **Default display name** field contains the default name. If you edit the name here, it will be reflected on the main **Views** page.
  - b. From the locale list, select a locale that you want to provide a version of the display name for.
  - c. To use the default display name for the selected locale, leave the **Use default value** check box selected.
  - d. To provide a localized display name, deselect the check box. In the field, type the localized name.
5. To localize the view description:
  - a. On the **Configure Locale Options** dialog, click the **Description** tab.

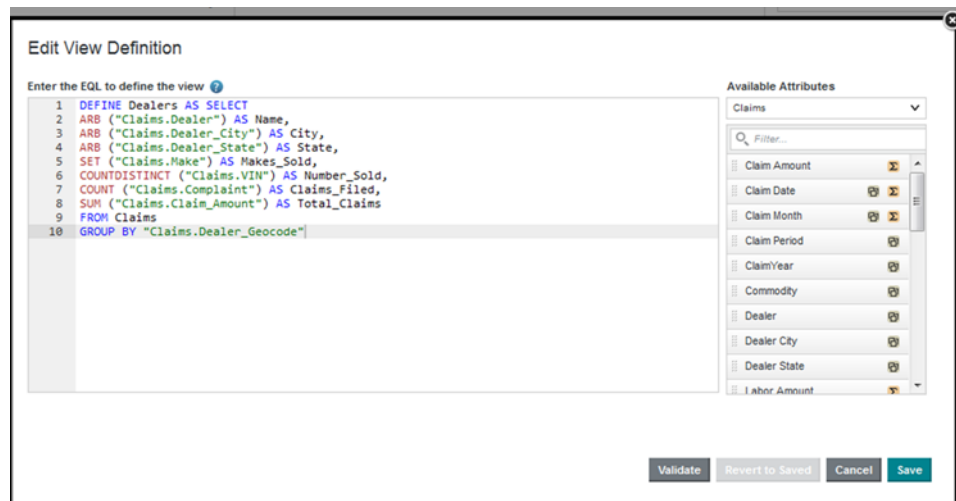
The **Default description** text area contains the default description. If you edit the description here, it is reflected on the main **Views** page.
  - b. From the locale list, select a locale that you want to provide a version of the description for.
  - c. To use the default description for the selected locale, leave the **Use default value** check box selected.
  - d. To provide a localized description, deselect the check box. In the text area, type the localized description.
6. After providing all of the localized versions of the name and description, click **Apply**.
7. To save the changes to the view, click **Save View**.

## Editing the view definition

From the **Views** page, after you create a custom view, you can update the view EQL definition. Note that if you change the definition, you could affect components that are using the view, as well as other views that may use attributes from this view.

To edit a view definition:

1. Click **Configuration Options > Project Settings**, then navigate to the **Data Views** page.
2. In the view list, click the view.
3. Click **Edit Definition**.
4. On the **Edit View Definition** dialog, update the view definition.



For details on the rules for setting a view definition, see [Guidelines for using EQL to define a view](#).

5. To validate the new definition, click **Validate**.

If the definition is invalid, then Studio displays error messages to help you troubleshoot the definition.

6. To save the new definition, click **Save**.

If the view definition is valid, then the attribute and metrics lists are updated to reflect the new definition.

If the view definition is not valid, then you are prompted to confirm that you want to save an invalid view. If you save the view, it is flagged as invalid and cannot be published.

## Guidelines for using EQL to define a view

The view definition EQL defines the content of the view. When defining a view, you need to keep the following guidelines in mind. For complete details on EQL syntax, see the *EQL Reference*.

## View DEFINE and SELECT statements

A view definition may contain multiple `DEFINE` and `SELECT` statements.

The last statement must be a `DEFINE` statement that contains all of the view attributes. The name of that `DEFINE` statement is used as the view key, and cannot be changed. If you try to change the view key for an existing view, the corresponding components will no longer work.

## View RETURN statements

Although the *EQL Reference* uses `RETURN` statements in its code examples, you should not add `RETURN` statements to any views you create in Studio. Studio automatically wraps your EQL with a `RETURN` statement.

You can copy EQL examples from the *EQL Reference* into a view and then delete the `RETURN` statement. This way, Studio wraps the EQL with an implicit `RETURN` and the example works correctly. If you explicitly add the `RETURN` the view fails.

## Providing the source view for the view attributes

For all custom view definitions, you must use the `FROM` parameter to provide the original source of the view attributes.

For example, if you create a copy of a base view, you will see that the view definition includes the view key for that base view.

```
DEFINE Customers as SELECT
ARB(Customer_Name) as Customer_Name,
ARB (Customer_Address) as Customer_Address,
ARB(Cities) as Cities,
ARB(States) as States,
ARB(CustomerZip) as CustomerZip,
ARB(LatLong) as LatLong,
ARB(Business_Types) as Business_Types,
ARB(CustomerAgreedDaysCredit) as CustomerAgreedDaysCredit,
ARB(Credit_Rating) as Credit_Rating,
ARB(CustomerDiscount) as CustomerDiscount,
SUM(Number_of_Cases_Sold) as TotalCustomerCases,
CountDistinct(Transaction_Id) as TotalCustomerTransactions,
SUM(GrossDollars) as TotalCustomerGross,
SUM(MarginDollars) as TotalCustomerMargin,
SET(Shipping_Companies) as ShippingCompaniesUsed
FROM "wine-sales"
GROUP BY Customer_Id
```

When identifying the source view, make sure to use the view key, and not the view display name.

Also, when referring to another view, you can only refer to the final `DEFINE` statement that identifies the attributes for other view. You cannot refer to any intermediate `DEFINE` or `SELECT` statements in that view definition.

## Grouping a view

All custom views should have at least one grouping (`GROUP BY`) attribute.

Studio uses the `GROUP BY` attributes as the identifying attributes for the view records.

If there are no `GROUP BY` attributes, then there are no identifying attributes, and the view cannot be used for **Record Details** or **Compare**. On component edit views, the **Record Details** and **Compare** options are then automatically disabled.

## Using multi-value attributes for grouping

When using a multi-value attribute for grouping, you must indicate whether to group by the individual values, or by the sets of values assigned to the records.

- To group by the sets of values, use `GROUP BY`.
- To group by the individual values, use `GROUP BY MEMBERS`. The syntax for `GROUP BY MEMBERS` is slightly different from `GROUP BY`:

```
GROUP BY MEMBERS(attributeName) as attributeName
```

For example, the following data shows available product colors, and the number of outlets that carry each product:

ItemColors	AvailableOutlets
Red, Blue	3
Red	5
Red, Blue, White	6
Red, Blue, White	4
Red, Blue	1

If you are calculating the total number of available outlets grouped by the item color, then:

- If you use `DEFINE NewView as SELECT TotalOutlets as sum(AvailableOutlets) FROM Sales GROUP BY ItemColors`, then the result is:

ItemColors	TotalOutlets
Red, Blue	4
Red	5
Red, Blue, White	10

- If you use `DEFINE NewView as SELECT TotalOutlets as SUM(AvailableOutlets) FROM Sales GROUP BY MEMBERS(ItemColors) as ItemColors`, then the result is:

ItemColors	TotalOutlets
Red	19
Blue	14
White	10

## Providing aggregation methods when grouping view attributes

When defining a view, if you are using grouping, then you must provide an aggregation method for all of the attributes in the view, except for the grouping attributes.

If you are using aggregation to derive attributes from the physical attributes, then you can use the applicable aggregation methods for the attribute data type. See [Aggregation methods and the data types that can use them](#).

In some cases, you may want to display the actual values of physical attributes in a grouped view. For example, when you are grouping by product identifier, then you may want to include the product name, product description, and available sizes. When using grouping for a view, to include the actual values of physical attributes:

- For a single-value attribute, use the ARB aggregation method. For example:

```
SELECT ARB(Region) as Region
```

Note that if there are different values for an attribute within the grouping, Studio chooses a single, arbitrary value.

- For a multi-value attribute:
  - To get a single, arbitrary value for the attribute from each record, use the ARB aggregation method. For example:

```
SELECT ARB(ItemColors) as ItemColors
```

- To get the complete set of values for the records, use the SET\_UNIONS aggregation method. For example:

```
SELECT SET_UNIONS(ItemColors) as ItemColors
```

If you are not using grouping, then you do not need an aggregation method. For example:

```
SELECT ProductName as ProductName
```

In this case, you do not need to specify the name, so you could also use:

```
SELECT ProductName
```

### Naming view attributes

When naming attributes in a view:

- Do not change the name of (also referred to as aliasing) attributes from the physical data.

For example, for a "Region" attribute, when you add the attribute to a view, define it as "Region".

```
SELECT ARB(Region) AS Region
```

Do not define the attribute under a different name, such as:

```
SELECT ARB(Region) AS RegionList
```

End users can only refine by attributes that are present in the physical data. If you alias an attribute, then end users cannot use it for refinement.

- Do not give a derived attribute the same name as an attribute from the physical data.

For example, for an attribute that averages the values of the "Sales" attribute, do not define the new attribute as:

```
SELECT AVG(Sales) as Sales
```

Instead, use something like:

```
SELECT AVG(Sales) as AvgSales
```

This is also to prevent end users from trying to refine by an attribute that is not in the physical data.

### Sample view definition

Here is an example of an EQL query for generating a list of products, including price, available colors, and sales numbers, from a single data set consisting of individual transactions.

```
DEFINE Products AS SELECT
ARB(ProductSubcategoryName) AS ProductSubcategoryName,
ARB(ProductCategoryName) AS ProductCategoryName,
ARB(Description) AS Description,
SET_UNIONS(AvailableColors) AS AvailableColors,
AVG("FactSales_SalesAmount") AS AvgSales,
SUM("FactSales_SalesAmount") AS SalesSum,
AVG(StandardCost) AS AvgStandardCost,
AVG(ListPrice) AS AvgListPrice,
(AvgListPrice - AvgStandardCost) AS Profit
FROM Sales
GROUP BY ProductName
```

## Configuring the view attributes

For each attribute in the view, you can configure the display name and aggregation methods.

### About the view attributes list

On the **Views** page, the attributes list for a view consists of the attributes from the view definition. This includes both the defined attributes and the grouping attributes.

### Editing and localizing an attribute display name and description

The attribute display name is used in Studio as the default label for the attribute. By default, the attribute display name is the same as the attribute key. The description provides additional information. You can edit both values, and also create localized versions for each of the supported locales.

### Determining the available subsets of date/time attributes

For a date/time attribute, you can select the available combinations of date "parts" that can be used to display and filter by the attribute.

### Configuring the default display format for an attribute

By default, the display format for attribute values is based on the attribute's data type plus the user's current locale. You can change the default display format used for each attribute. Within specific components, you can then make some updates to the display format.

### Selecting the available and default aggregation methods for an attribute

Aggregation methods are used to aggregate attribute values into metrics displayed on components such as **Chart**, **Pivot Table**, and **Results Table**.



### [Aggregation methods and the data types that can use them](#)

Aggregation methods are types of calculations used to group attribute values into a metric for each dimension value. For example, for each country (each value of the Country dimension), you might want to retrieve the total value of transactions (the sum of the Sales Amount attribute).

### [Configuring the Available Refinements behavior for base view attributes](#)

For each attribute in a base view, the **Views** page attributes list allows you to configure settings for how to handle selection and sorting of attribute values on the **Available Refinements** panel.

## About the view attributes list

On the **Views** page, the attributes list for a view consists of the attributes from the view definition. This includes both the defined attributes and the grouping attributes.

The list includes the following information about each attribute:

Attribute List Column	Description
<b>Display Name</b>	The display name for the attribute. You can edit and localize the display name at any time. See <a href="#">Editing and localizing an attribute display name and description</a> .
<b>Attribute Key</b>	The attribute identifier. Read-only. You cannot change the attribute key unless you update the view definition.
<b>Refinable</b>	Whether the attribute can be used for refinement. Only attributes from the physical data can be used for refinement. Read-only.
<b>Searchable</b>	Whether the attribute is enabled for either full text or typeahead search. Read-only. To enable or disable an attribute for search, see <a href="#">Enabling or disabling full text search and typeahead search on an attribute</a> .
<b>Multi-Value</b>	Whether the attribute can be assigned multiple values. For a base view, this indicates whether the physical attributes are configured to be multi-value. For views you create, an attribute can be multi-value if you used the set aggregation for that attribute in the view definition. Read-only.
<b>Data Type</b>	The data type for the attribute. You cannot change the data type. For base views, for date/time attributes, you can configure the available subsets of date/time units that can be used for filtering and display. See <a href="#">Determining the available subsets of date/time attributes</a> .

Attribute List Column	Description
<b>Standard Format</b>	<p>The default display format for the attribute values. For most data types, you can customize the default display format.</p> <p>String attributes can only be customized if they are multi-value. See <a href="#">Configuring the default display format for an attribute</a>.</p>
<b>Dimension</b>	<p>Whether the attribute can be used as a dimension for aggregating metric values.</p> <p>To allow the attribute to be used as a dimension, select the check box.</p> <p>Note that multi-value date/time attributes cannot be used as dimensions.</p>
<b>Available Aggregations</b>	<p>The available aggregation methods for aggregating the values for this attribute into a metric.</p> <p>You can enable or disable available aggregation methods. See <a href="#">Selecting the available and default aggregation methods for an attribute</a>.</p>
<b>Default Aggregation</b>	<p>The default aggregation method.</p> <p>You can change the default aggregation method. See <a href="#">Selecting the available and default aggregation methods for an attribute</a>.</p>
<b>Refinement Behavior</b>	<p>Displayed and editable for base views only.</p> <p>The type of refinement allowed for the attribute.</p> <p>Indicates whether users can select more than one value to refine by.</p> <p>See <a href="#">Configuring the Available Refinements behavior for base view attributes</a>.</p>
<b>Include in Available Refinements</b>	<p>Displayed and editable for base views only.</p> <p>Whether to include the attribute on the <b>Available Refinements</b> panel.</p> <p>If this box is not selected, then even if the attribute belongs to a group that is included on the <b>Available Refinements</b> panel, the attribute is not displayed.</p> <p>See <a href="#">Configuring the Available Refinements behavior for base view attributes</a>.</p>
<b>Description</b>	<p>A short description of the attribute.</p> <p>You can edit and localize the attribute description at any time. See <a href="#">Editing and localizing an attribute display name and description</a>.</p>

### Editing and localizing an attribute display name and description

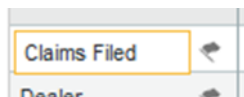
The attribute display name is used in Studio as the default label for the attribute. By default, the attribute display name is the same as the attribute key. The description provides additional information. You can edit both values, and also create localized versions for each of the supported locales.

In the attributes list on the **Views** page, to edit and localize an attribute's display name and description:

1. To edit the display name:

a. Click the display name.

The name displays in an editable field.



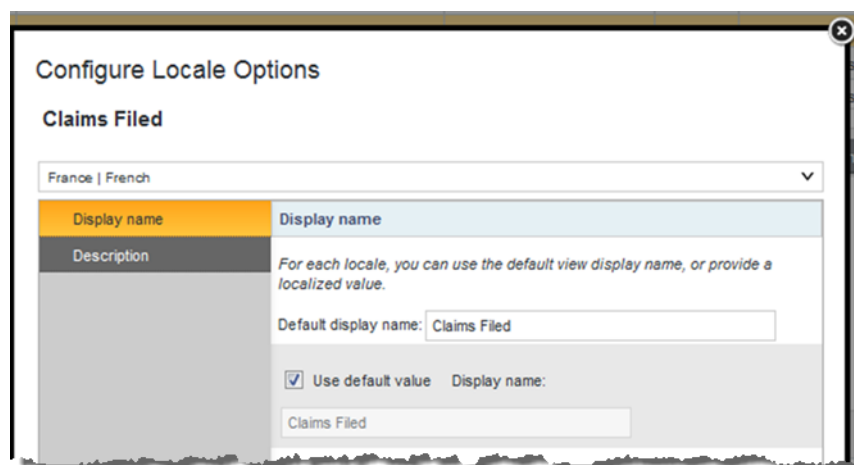
b. In the field, type the new name.

2. To localize the display name, in the **Display Name** column, click the localize icon.

On the **Display Name** tab of the **Configure Locale Options** dialog, the **Default display name** field contains the value you entered in the **Display Name** column. If you edit the value here, it also is updated in the attributes list.

For each locale:

a. From the locale list, select the locale for which to configure the display name value.



b. To use the default value, leave the **Use default value** check box selected.

c. To provide a localized value for the selected locale, deselect the check box. In the **Display name** field, enter the localized value.

d. After entering the localized values, click **Apply**.

3. From the **Description** column, to edit the description:

a. Click the description.

The description displays in an editable field.

b. In the field, type the new description.

4. To localize the description, in the **Description** column, click the localize icon.

On the **Description** tab of the **Configure Locale Options** dialog, the **Default description** field contains the value you entered in the column. If you edit the value here, it also is updated in the attributes list.

For each locale:

- a. From the locale list, select the locale that you want to configure the description of.
  - b. To use the default description, leave the **Use default value** check box selected.
  - c. To provide a localized value for the selected locale, deselect the check box. In the **Description** text area, enter the localized value.
  - d. After entering the localized values, click **Apply**.
5. To save the changes to the attribute display name and description, click **Save View**.

### Determining the available subsets of date/time attributes

For a date/time attribute, you can select the available combinations of date "parts" that can be used to display and filter by the attribute.

For example, you may want to allow users to filter by the specific month or combination of month and year.

On a component, users may choose to display an individual date/time element (Year, Month, Day, Hour, Minute, Second), or may be able to select one of the following date/time combinations:

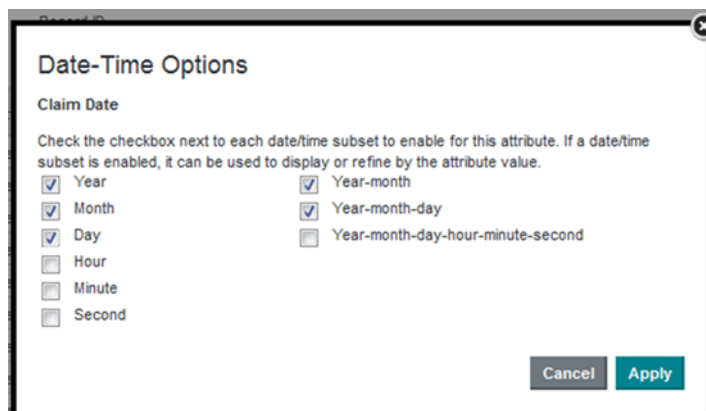
Date/Time Display Option	Sample Value
Year-Month	January 2012
Year-Month-Day	January 1, 2012
Year-Month-Day-Hour-Minute-Second	January 1, 2012 3:15:05 am

To configure the available subsets of date/time units:

1. In the **Data Type** column for a date/time attribute, click the edit icon.

Data Type	Standard
String	(string)
String	(string)
Date/Time	DateTim
String	(string)
Date/Time	DateTim
String	(string)
String	(string)

2. On the **Date-Time Options** dialog, select the check box next to each option to enable.



3. Click **Apply**.

### Configuring the default display format for an attribute

By default, the display format for attribute values is based on the attribute's data type plus the user's current locale. You can change the default display format used for each attribute. Within specific components, you can then make some updates to the display format.

Note that you only configure the display format for the default locale version of the attribute. You do not configure formatting for the locale-specific attributes. Because the formatting configuration already takes the current locale into account, you do not need to configure the formatting for the locale-specific attributes separately.

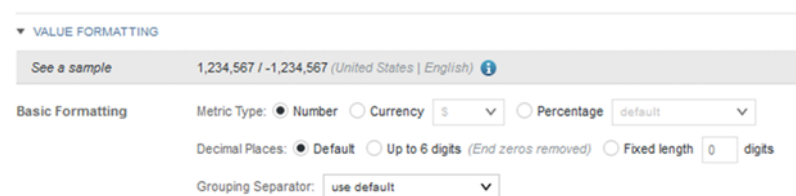
From the attributes list on the **Views** page, to change the default display format for an attribute:

1. Click the edit icon in the **Standard Format** column.

The **Format Option** dialog displays.

At the top of the dialog is a sample value that displays the current formatting selection based on your current locale.

2. For numeric attributes:



- a. Under **Metric Type**, click the radio button to indicate the type of number.

A numeric attribute may be displayed as a regular number, a currency value, or a percentage.

Note that if you set a numeric attribute to be a currency or percentage, that selection cannot be overridden at the component level.

- b. If the number is a currency value, then from the **Currency** list, select the type of currency. This selection determines the symbol to display next to each value.

- c. If the number is a percentage, then from the **Percentage** list, select whether to display the percentage (%) sign with the value.

Note that if you use percentage as the format, Studio automatically multiplies the value by 100. So if the actual value is 0.05, Studio displays the value as 5%.

- d. Under **Decimal Places**, to display all numbers using the same default number of decimal places, leave **Default** selected.

To display all of the decimal places from the original value, up to 6 places, click **Up to 6 digits**. Studio truncates any decimal places after 6, and removes any trailing zeroes.

To specify the number of decimal places to use for all numbers, click **Fixed length**, then in the field, type the number of decimal places to display.

- e. From the **Grouping Separator** list, select whether to display the grouping separator (used to separate thousands).
- f. The **Advanced Formatting** section provides additional format options.

**Advanced Formatting** *Changes below will override localized formatting based on each user's locale.*

Number Format: varies based on locale

Decimal Separator: varies based on locale

Grouping Separator: varies based on locale

For each of the advanced items, you can choose to have the display determined automatically based on the user's locale, or select a specific option.

- g. From the **Number Format** list, select how to display positive and negative numbers.
  - h. From the **Decimal Separator** list, select the character to use for the decimal point.
  - i. From the **Grouping Separator** list, select the character to use for the grouping (thousands) separator.
3. For a Boolean attribute, you can determine the values to display for each Boolean value.

By default, 1 displays as the localized version of "True", and 0 displays as the localized version of "False". To set specific (but non-localized) values to display:

[See a sample](#) true / false (United States | English) ⓘ

**Basic Formatting** Boolean Display:  Default  Custom Values

True:

False:

- a. Click **Custom Values**.
- b. In the **True** field, type the value to display if the attribute value is 1 (True).
- c. In the **False** field, type the value to display if the attribute value is 0 (False).

4. For a geocode, to set a specific number of decimal places to display for the latitude and longitude values:

▼ VALUE FORMATTING

See a sample 44.756530 -91.473097 (United States | English) ⓘ

Basic Formatting    Decimal Places:  Default  Fixed length  digits

- a. Click **Fixed length**.
  - b. In the field, type the number of decimal places.
5. For a date/time attribute:

See a sample 1/18/99 (United States | English) ⓘ

Basic Formatting    Date Display: use default ▼

Time Display: use default ▼

- a. From the **Date Display** list, select the format to use for the date.  
The selected format is adjusted for locale. For example, for some locales the month displays first, and for others the day displays first.
  - b. From the **Time Display** list, select the format to use for the time.
6. For a time attribute, from the **Time Display** list, select the format to use for the time.

See a sample 3:30 PM (United States | English) ⓘ

Basic Formatting    Time Display: use default ▼

7. For all value types, the **Multi-Value Formatting** section indicates how to display multiple values.

▼ MULTI-VALUE FORMATTING

**Multi-Value Formatting**    *Only apply when a record has a multiple values or as the result of a set aggregation.*  
Changes to any of the settings below will override localized formatting based on each user locale.

Multi-value Separator: varies based on locale ▼

The setting is applied whenever multiple values are displayed for an attribute, including when an attribute is by definition multi-value, and when the set aggregation is applied to an attribute.

From the **Multi-value Separator** list, select the character to use to separate the values.

8. To confirm the format configuration, click **Apply**.
9. To save the changes to the view settings, click **Save View**.

## Selecting the available and default aggregation methods for an attribute

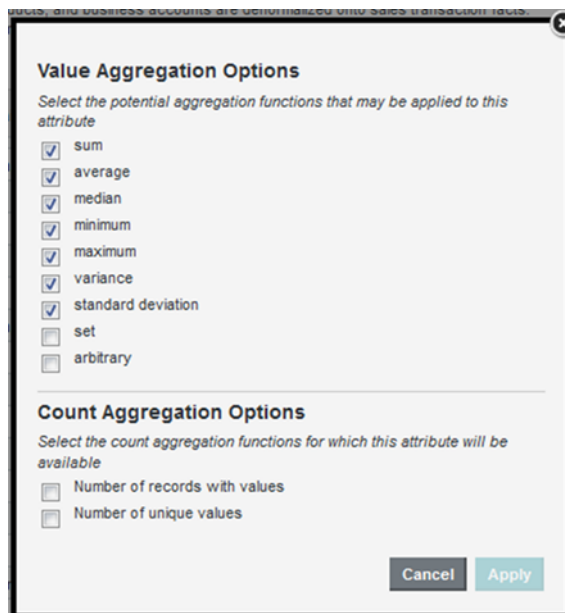
Aggregation methods are used to aggregate attribute values into metrics displayed on components such as **Chart**, **Pivot Table**, and **Results Table**.

Based on its data type, each attribute can support specific aggregation methods. You can then configure attributes to only support a subset of the applicable aggregation methods. For details on the available aggregation methods, see [Aggregation methods and the data types that can use them](#).

Each attribute also is assigned a default aggregation method. You can change the default used for each attribute.













From the attributes list on the **Views** page, to configure the available and default aggregation methods for an attribute:

1. To configure the available aggregation methods for an attribute:
  - a. In the **Available Aggregations** column, click the edit icon.
  - b. On the aggregation methods dialog, to enable an available aggregation method, select the check box.



- c. To disable an aggregation method, deselect the check box.
  - d. To confirm the settings, click **Apply**.
2. From the **Default Aggregation** list, select the default aggregation to use.



	Default Aggregation		
	set	⌵	M
	set	⌵	M
	set	⌵	M
	set	⌵	M
	sum	⌵	S
	sum	⌵	M
	average	⌵	M
	median	⌵	M
	minimum	⌵	M
	maximum	⌵	M
	variance	⌵	S
	standard deviation	⌵	S

- To save the changes to the view configuration, click **Save View**.

### Aggregation methods and the data types that can use them

Aggregation methods are types of calculations used to group attribute values into a metric for each dimension value. For example, for each country (each value of the Country dimension), you might want to retrieve the total value of transactions (the sum of the Sales Amount attribute).

On the component edit view, you select the aggregation method from a list. When using these aggregation methods in definitions of views and predefined metrics, you use the EQL syntax.

The available aggregation methods are:

Aggregation Method	Description	EQL Syntax
sum	Calculates the total value for the metric. You can use this aggregation method for numbers. You cannot use this aggregation method for multi-value attributes.	<code>sum(attributeName)</code>
average	Calculates the average value for the metric. You can use this aggregation method for numbers, dates, and times. You cannot use this aggregation method for multi-value attributes.	<code>avg(attributeName)</code>

Aggregation Method	Description	EQL Syntax
median	<p>Calculates the median value for the metric.</p> <p>You can use this aggregation method for numbers, dates, and times.</p> <p>You cannot use this aggregation method for multi-value attributes.</p>	<code>median(<i>attributeName</i>)</code>
minimum	<p>Selects the minimum value for the metric.</p> <p>You can use this aggregation method for numbers, dates, and times.</p> <p>You cannot use this aggregation method for multi-value attributes.</p>	<code>min(<i>attributeName</i>)</code>
maximum	<p>Selects the maximum value for the metric.</p> <p>You can use this aggregation method for numbers, dates, and times.</p> <p>You cannot use this aggregation method for multi-value attributes.</p>	<code>max(<i>attributeName</i>)</code>
variance	<p>Calculates the variance (square of the standard deviation) for the metric values.</p> <p>You can use this aggregation method for numbers, dates and times.</p> <p>You cannot use this aggregation method for multi-value attributes.</p>	<code>variance(<i>attributeName</i>)</code>
standard deviation	<p>Calculates the standard deviation for the metric values.</p> <p>You can use this aggregation method for numbers, dates and times.</p> <p>You cannot use this aggregation method for multi-value attributes.</p>	<code>stddev(<i>attributeName</i>)</code>
set	<p>Instead of performing a calculation, creates a list of all of the unique values.</p> <p>You can use this aggregation method for any attribute.</p>	<code>set(<i>attributeName</i>)</code> For multi-value attributes, use <code>set_unions(<i>attributeName</i>)</code> .

Aggregation Method	Description	EQL Syntax
arb	<p>Instead of performing a calculation, arbitrarily selects one of the values.</p> <p>You can use this aggregation method for any attribute.</p> <p>The arb aggregation method is typically not provided as an option for aggregating metric values on components.</p>	<code>arb(attributeName)</code>

When configuring components, the following count aggregation options are applied using separate system metrics.

For the other aggregation methods, such as sum and average, you select an attribute, and then apply the aggregation method. For the count aggregation methods, you first select the system metric, then select the attribute to use.

When using these aggregation methods in definitions of views and predefined metrics, you use the EQL syntax.

Aggregation Method	Description	EQL Syntax
Number of records with values	<p>Calculates the number of records with a value for the selected attribute.</p> <p>You can use this aggregation method for all data types.</p> <p>You cannot use this aggregation method for multi-value attributes.</p> <p>You apply this aggregation by adding the <b>Number of records with values</b> system metric to the component.</p>	<code>count(attributeName)</code>
Number of unique values	<p>Calculates the number of unique values for the metric.</p> <p>You can use this aggregation method for all data types.</p> <p>You cannot use this aggregation method for multi-value attributes.</p> <p>You apply this aggregation by adding the <b>Number of unique values</b> system metric to the component.</p>	<code>countdistinct(attributeName)</code>

To show how each of the aggregation methods work, we'll use the following data:

Country	Amount of Sale	Shipping Company
US	125.00	UPS
US	50.00	UPS

Country	Amount of Sale	Shipping Company
US	150.00	FedEx

Based on these values, the aggregation for Amount of Sale and Shipping Company for the US would be:

Aggregation Method	Amount of Sale (US)	Shipping Company (US)
sum	325.00	Cannot be aggregated
avg	108.33	Cannot be aggregated
median	125.00	Cannot be aggregated
min	50.00	Cannot be aggregated
max	150.00	Cannot be aggregated
variance	2708.33	Cannot be aggregated
standard deviation	52.04	Cannot be aggregated
set	125.00, 50.00, 150.00	UPS, FedEx
arb	Could be either 125.00, 50.00, or 150.00	Could be either UPS or FedEx
Number of records with values	3	3
Number of unique values	3	2

### Configuring the Available Refinements behavior for base view attributes

For each attribute in a base view, the **Views** page attributes list allows you to configure settings for how to handle selection and sorting of attribute values on the **Available Refinements** panel.

In the attributes list for a base view, to configure the value selection and sorting for each attribute:

1. From the **Refinement Behavior** list, select whether users can select multiple values, and if so, whether the records must contain all of the selected values.

Refinement Behavior	Include in Available Refinements
Single	<input checked="" type="checkbox"/>
Single	<input checked="" type="checkbox"/>
Multi-AND	<input checked="" type="checkbox"/>
Multi-OR	<input checked="" type="checkbox"/>
Single	<input checked="" type="checkbox"/>

The available options are:

Option	Description
Single	Indicates that end users can only refine by one value at a time.
Multi-OR	Indicates that end users can refine by more than one value at a time. For multi-or, a record matches if it has at least one of the selected values. So if an end user selects the values Red, Green, and Blue, then matching records only need to have one of those values (Red or Green or Blue). Note that date/time attributes are always multi-or. You cannot change the refinement behavior for a date/time attribute.
Multi-AND	Indicates that end users can refine by more than one value at a time. For multi-and, a record matches only if it has all of the selected attribute values. Multi-and should only be used with multi-value attributes. So if an end user selects the values Red, Green, and Blue, then matching records must have all of those values (Red and Green and Blue).

2. In the **Include in Available Refinements** column, to include the attribute on the **Available Refinements** panel if it belongs to a group that displays, select the check box.

## Defining predefined metrics for a view

Custom views can also have predefined metrics, which are values calculated from the view attributes.

### About predefined metrics

Predefined metrics are values calculated from view attributes. For example, if the attributes for a view include product price and cost information, you could calculate predefined metrics such as the ratio of retail price to cost.

### Adding a predefined metric to a view

Adding a predefined metric to a view consists of specifying the metric key and EQL definition.

### Configuring a predefined metric for a view

From the predefined metrics list on the **Views** page, you can configure the metric name, description, definition, and display. The display name and definition can be localized for each of the available locales.

### Deleting a predefined metric

You can delete predefined metrics from a view. Before deleting a predefined metric, try to make sure that it is not being used by a component.

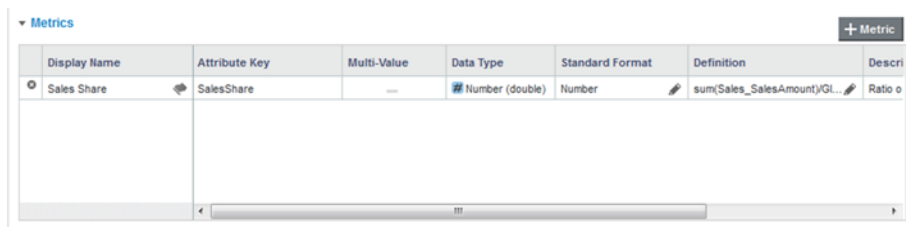
## About predefined metrics

Predefined metrics are values calculated from view attributes. For example, if the attributes for a view include product price and cost information, you could calculate predefined metrics such as the ratio of retail price to cost.

Predefined metrics are optional. By default, views do not have predefined metrics. They are always added manually.

Predefined metrics should only be used for calculated values that are only meaningful within the context of the view, and that stand on their own as values. When configuring a component, you can assign simple aggregations such as sum or average to any attribute, so you don't need to create those as predefined metrics. Predefined metrics are more appropriate for more complex values such as ratios.

On the **Views** page, the **Metrics** section, which lists the predefined metrics, displays below the view attribute list. By default, the section is collapsed. To expand the section, click the section heading.



Display Name	Attribute Key	Multi-Value	Data Type	Standard Format	Definition	Description
Sales Share	SalesShare	...	Number (double)	Number	sum(Sales_SalesAmount)/GL...	Ratio o

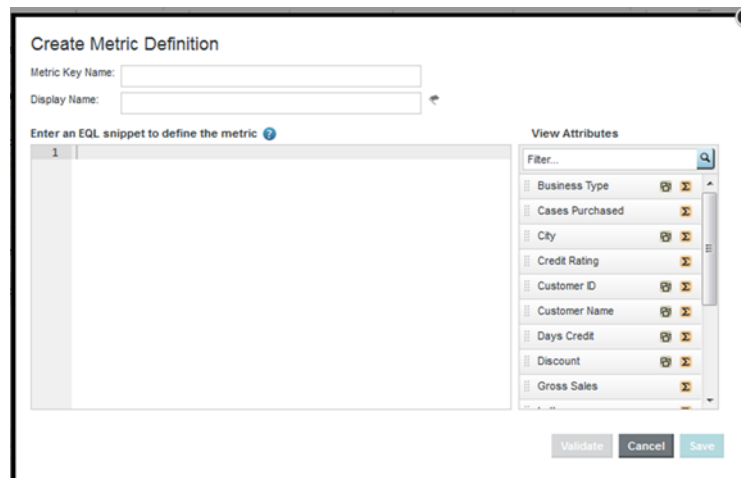
## Adding a predefined metric to a view

Adding a predefined metric to a view consists of specifying the metric key and EQL definition.

On the **Views** page, to add a predefined metric to a view:

1. From the **Metrics** list, click **+ Metric**.

The **Create Metric Definition** dialog displays.



2. On the **Create Metric Definition** dialog, in the **Metric Key Name** field, type the key name for the metric.

The key name must be NCName-compliant.

Once you save the new metric, you cannot change the key name.

3. In the **Display name** field, type the display name for the metric.

This is the name used to select the metric to add to a component. You can edit and localize the display name at any time.

4. In the text area, type the EQL expression for calculating the predefined metric value.

The expression is an EQL snippet that is inserted into a `SELECT` statement when a query using the predefined metric is issued. The snippet can only refer to attributes from the current view. For example:

```
count(1) WHERE (SalesAmountSum > 3000000)
```

To the right of the EQL text area is the list of attributes in the current view for you to use as a reference. You can show or hide the attribute list.

For information on aggregation methods, including the EQL syntax for including them in predefined metrics, see [Aggregation methods and the data types that can use them](#). For detailed information on EQL syntax, see the *EQL Reference*.

5. To validate the expression, click **Validate**.

If the expression is invalid, Studio displays error messages to help you troubleshoot.

6. To confirm adding the metric, click **Save**.

### Configuring a predefined metric for a view

From the predefined metrics list on the **Views** page, you can configure the metric name, description, definition, and display. The display name and definition can be localized for each of the available locales.

Display Name	Attribute Key	Multi-Value	Data Type	Standard Format	Definition	Descri
Sales Share	SalesShare	---	Number (double)	Number	sum(Sales_SalesAmount)/GL...	Ratio o

To configure a predefined metric:

1. In the **Display name** field, type a display name for the predefined metric.

Display names do not need to be NCName-compliant.

2. To localize the display name, in the **Display name** column, click the localize icon.

On the **Configure Locale Options** dialog, the **Default display name** field contains the value you entered in the column. If you edit the value here, it also is updated in the metrics list. For each locale:

- a. From the locale list, select the locale that you want to configure the display name value of.

- b. To use the default value, select the **Use default value** check box.
  - c. To provide a localized value for the selected locale, deselect the check box. In the **Display name** field, enter the localized value.
  - d. After entering the localized values, click **Apply**.
3. To configure the display format for a predefined metric, click the edit icon in the **Standard Format** column.

The display format configuration is the same as that for attributes. See [Configuring the default display format for an attribute](#).

4. To change the definition of a predefined metric, click the edit icon in the **Definition** column.
5. In the **Description** field, type a brief description of the predefined metric.
6. To localize the description, in the **Description** column, click the localize icon.

On the **Configure Locale Options** dialog, the **Default description** field contains the value you entered in the column. If you edit the value here, it also is updated in the metrics list. For each locale:

- a. From the locale list, select the locale that you want to configure the description of.
  - b. To use the default description, select the **Use default value** check box.
  - c. To provide a localized value for the selected locale, deselect the check box. In the **Description** field, enter the localized value.
  - d. After entering the localized values, click **Apply**.
7. To save any changes to the predefined metrics, click **Save View**.

### Deleting a predefined metric

You can delete predefined metrics from a view. Before deleting a predefined metric, try to make sure that it is not being used by a component.

From the **Views** page, to delete a predefined metric from the current view, click the delete icon for that metric.

To save the change to the view configuration, click **Save View**.

## Previewing the content of a view

From the **Views** page, you can display a preview list of the records in a view.

To preview the contents of a view:

1. Click **Configuration Options > Project Settings**, then navigate to the **Data Views** page.
2. In the views list, click the view.
3. Click **Preview**.

The **Preview** dialog displays, containing a list of records for the view.



The list only displays the view attributes. Because the preview list is not grouped by a specific dimension, it cannot display the predefined metrics.

4. To close the **Preview** dialog, click **Close**.

## Deciding whether to publish a custom view

For custom views, the **Publish** check box controls whether the view is published. Only published views are available to use in a project.

Base views are always published.

Custom views can only be published if the view definition is valid.

## Deleting a view

From the **Views** page, you can delete custom views. You cannot delete base views or linked views. If you delete a view, any component that uses that view can no longer display correctly. Also, any other view whose definition includes references to this view becomes invalid.

Note that Big Data Discovery cannot automatically check whether a view is being used in the definition of another view.

To delete a view, in the **Actions** column of the view list, click the delete icon for that view.



---

## Managing Attribute Groups for a Project

You can group attributes within each project data view in order to organize the information.

### [About attribute groups and the Attribute Groups page](#)

Attribute groups are used to group attributes within a view. For example, for a view containing sales information, one group might contain attribute related to the payment (credit card type, cost, billing address) and another group might contain attributes related to the shipping (shipping address, shipping company, shipping cost).

### [About groups for linked views](#)

Studio automatically creates a linked view for each linked data set. For each linked view, Studio also creates a default set of attribute groups.

### [Displaying the groups for a selected view](#)

When the **Attribute Groups** page is first displayed, the groups for the first base view are displayed by default. You can then change the selected view.

### [Displaying attribute information for a group](#)

For each group on the **Attribute Groups** page, you can display a list of attributes and attribute metadata.

### [Creating an attribute group](#)

When you create a group, you give the group a name, select default options for how the group is used, and then select the attributes that belong to the group.

### [Deleting an attribute group](#)

From the **Attribute Groups** page, you can delete attribute groups.

### [Configuring an attribute group](#)

From the **Attribute Groups** page, for each group, you can configure the display name, the group membership, and the attribute display order.

### [Saving the attribute group configuration and reloading the attribute group cache](#)

Changes to the group configuration do not become permanent until you save them. To immediately reflect the saved group configuration changes on the project components, reload the attribute group cache.

## About attribute groups and the Attribute Groups page

Attribute groups are used to group attributes within a view. For example, for a view containing sales information, one group might contain attribute related to the payment (credit card type, cost, billing address) and another group might contain attributes related to the shipping (shipping address, shipping company, shipping cost).

For each group, you select the specific attributes in the group.

**ATTRIBUTE GROUPS**

View: Wine Reload Attribute Cache Revert to Last Save Save

Attribute Groups + Group *Price and Score Attributes* + Attributes

**Permanent Groups**

- All
- Other

**Created Groups**

- Identification
- Source
- Characteristics
- Price and Score**

Filter...

Attributes ordered by: Custom order

Display Name	Attribute Key	Info	Data Type	Dimension
Sales Price	P_Price		#	---
Price Range	PriceRange		##	✓
Date Reviewed	P_DateReviewed		##	✓
Base Score	P_Score		#	---
Score Range	ReviewScore		##	✓
Designation	Designation		##	✓

From the **Attribute Groups** page, you can create and edit the attribute groups for the current project data. To display the **Attribute Groups** page:

1. Click **Configuration Options > Project Settings**.
2. Click **Attribute Groups**.

## About groups for linked views

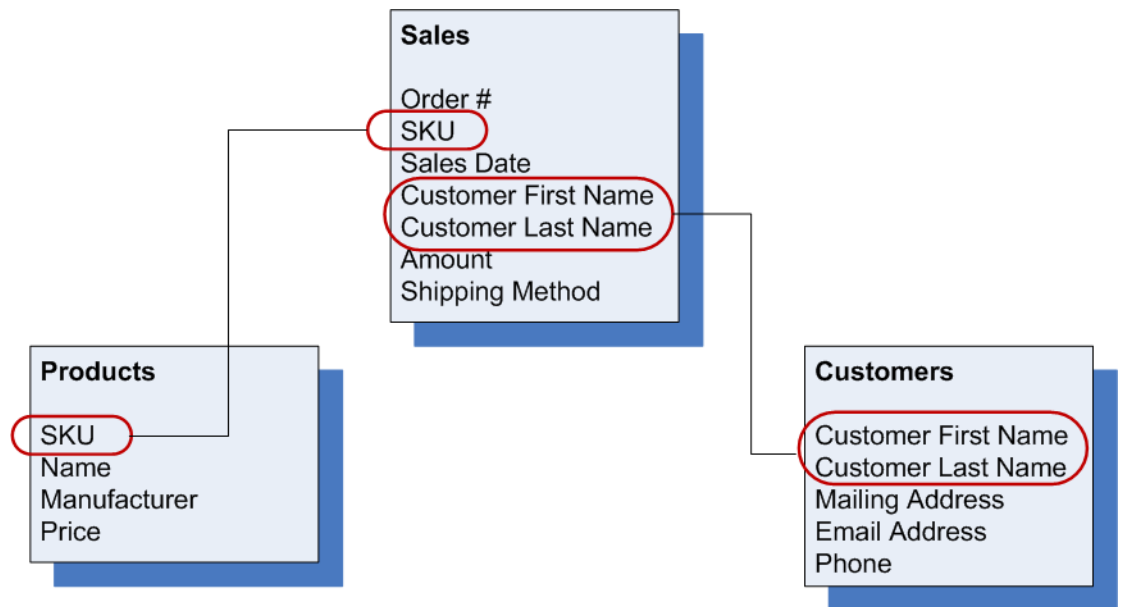
Studio automatically creates a linked view for each linked data set. For each linked view, Studio also creates a default set of attribute groups.

The default attribute groups for a linked view consist of:

- The attribute groups from the source data set
- For each data set that the source data set is linked to, an attribute group containing all of the attributes in that data set, except for the link key attributes. The group is named for the data set.

For example, as shown in the following diagram:

- The Sales and Products data sets are linked by the SKU attribute
- The Sales and Customers data sets are linked using the Customer First Name and Customer Last Name attributes.



For these links, Studio creates the following views:

- Sales - linked
- Products - linked
- Customers - linked

If the base views for the data sets contain the following groups:

Data Set	Base View Groups
Sales	Order Creation Customer Information Shipping Details
Products	Product Summary Product Pricing
Customers	Customer Name Customer Contact Information

Then for the linked views, Studio creates the following default attribute groups:

Linked View	Default Attribute Groups
Sales - linked	Order Creation Customer Information Shipping Details Products (all Product attributes, except for SKU) Customers (all Customer attributes, except for Customer First Name and Customer Last Name)
Products - linked	Product Summary Product Pricing Sales (all Sales attributes, except for SKU)

Linked View	Default Attribute Groups
Customers - linked	Customer Name Customer Contact Information Sales (all Sales attributes, except for Customer First Name and Customer Last Name)

These are default groups only. You can create, edit, and remove groups for linked views the same way you would for base or custom views.

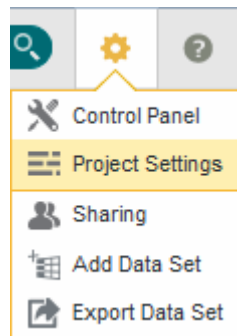
## Displaying the groups for a selected view

When the **Attribute Groups** page is first displayed, the groups for the first base view are displayed by default. You can then change the selected view.

To display the groups for a selected view:

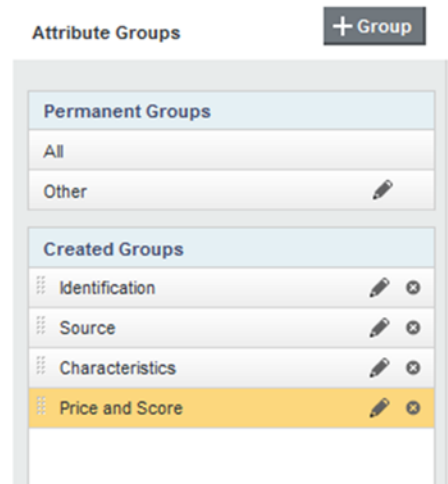
to select the view for which to display the attribute groups:

1. Click **Configuration Options > Project Settings**:



2. Select the **Attribute Groups** page.
3. From the **View** list, select the view.
4. If there are unsaved changes in the current view, you are prompted to either save the changes, discard the changes, or cancel and return to the current view.

When you select a view, the **Attribute Groups** section is updated to display the groups for the current view.



The **Permanent Groups** list contains the **All** group and the **Other** group. These groups are automatically part of every view.

- The **All** group always contains all of the attributes in the view.
- The **Other** group always contains the attributes that are not members of any of the created groups.

The **Created Groups** list contains the groups you create for this view, as well as groups created automatically for linked data set views.

For the **Created Groups** list, you can set the default display order.

## Displaying attribute information for a group

For each group on the **Attribute Groups** page, you can display a list of attributes and attribute metadata.

### Displaying the list of attributes for a group

From the **Attribute Groups** list on the **Attribute Groups** page, to display the current contents of a group, click the group name. The attribute list is updated to display the current membership of the group.

### Displaying the full list of metadata values for an attribute

From the **Attribute Groups** page, to display the full list of metadata values for an attribute, hover the mouse over the information icon in the **Info** column.

### Selecting the metadata columns to display for each attribute

By default, the attribute list on the **Attribute Groups** page displays basic information for each attribute. You can add columns to and remove columns from the attribute list.

## Displaying the list of attributes for a group

From the **Attribute Groups** list on the **Attribute Groups** page, to display the current contents of a group, click the group name. The attribute list is updated to display the current membership of the group.

The attribute list consists of attribute metadata for each attribute. By default, the list includes the attribute name, attribute key, data type, and whether the attribute is a dimension.

Identification Attributes + Attributes

Filter... Attributes ordered by : Custom order

Display Name	Attribute Key	Info	Data Type	Dimension
Name	P_Name		["]	
Description	P_Description		["]	
Wine Type	WineType		["]	

## Displaying the full list of metadata values for an attribute

From the **Attribute Groups** page, to display the full list of metadata values for an attribute, hover the mouse over the information icon in the **Info** column.

The metadata includes the list of groups that the attribute belongs to.

## Selecting the metadata columns to display for each attribute

By default, the attribute list on the **Attribute Groups** page displays basic information for each attribute. You can add columns to and remove columns from the attribute list.

To change the displayed columns:

1. Hover the mouse over the right edge of any column.  
An arrow displays.
2. Click the arrow.
3. In the menu, click **Columns**.

The full list of available columns displays.

Columns currently in the list are selected.

Display Name	Attribute Key	Info	Data Type	Dimen
Name	Sort Ascending		["]	
Description	Sort Descending		["]	
Wine Type	wine type		["]	

Columns

- Attribute Key
- Info
- Data Type
- Dimension
- Refinement Behavior
- Include In Available Refinements
- Refinable
- Group Membership

4. To add a column, select the check box.  
To remove a displayed column, deselect the check box.



## Creating an attribute group

When you create a group, you give the group a name, select default options for how the group is used, and then select the attributes that belong to the group.

From the **Attribute Groups** page, to create a new attribute group:

1. Click **+Group**.
2. On the **New Group** dialog, in the **Group name** field, enter the name of the new group.

3. For a group in a base view, to have this group enabled on the **Available Refinements** panel, select the **Include this group in Available Refinements** check box.

Note that because the **Available Refinements** panel only uses base views, this check box is not displayed for groups created in other views.

Also, even if you enable the group, the attributes in the group are only displayed if they are configured on the **Views** page to display on the **Available Refinements** panel. See [Configuring the Available Refinements behavior for base view attributes](#).

4. To have this group enabled when displaying record details information for this view, including on the **Record Details** and **Compare** dialogs, and in the default configuration for a **Results Table** component, select the **Include this group in record details** check box.
5. Configure the group membership.  
See [Configuring the group membership](#).
6. After completing the configuration, click **Apply**.

The new group is added to the group list. You can then drag the group to set its default display order.

## Deleting an attribute group

From the **Attribute Groups** page, you can delete attribute groups.

You cannot delete the default **Other** or **All** attribute groups.

Deleting an attribute group does not delete its attributes from the view or from other attribute groups.

To delete an attribute group, in the **Created Groups** list, click the delete icon for the attribute group. The attribute group is removed from the **Created Groups** list.

When you save the attribute group changes for the current view, then the attribute group is no longer displayed on components.

If an attribute does not belong to any of your other groups, it is restored to the default **Other** group.

## Configuring an attribute group

From the **Attribute Groups** page, for each group, you can configure the display name, the group membership, and the attribute display order.

### Changing the group name and use

The group name is used as the display name for the group in Studio. You can edit and localize the name. You can also change whether a group is used in **Available Refinements** and for record detail display.

### Configuring the group membership

For each group, you select the specific attributes that belong to the group.

### Configuring the sort order for a group

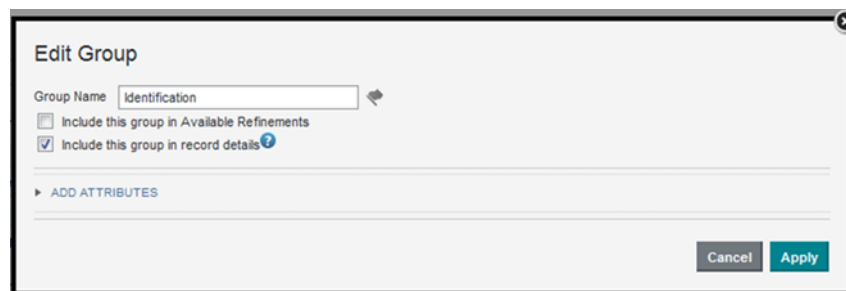
For each group, you can either sort the list using a metadata value, or set the sort order manually. The sort order on the **Attribute Groups** page determines the display order of attributes on components.

## Changing the group name and use

The group name is used as the display name for the group in Studio. You can edit and localize the name. You can also change whether a group is used in **Available Refinements** and for record detail display.

To change the name and default use of an existing group:

1. In the groups list, click the edit icon for the group.
2. On the **Edit Group** dialog, in the **Group Name** field, type the new display name for the group.

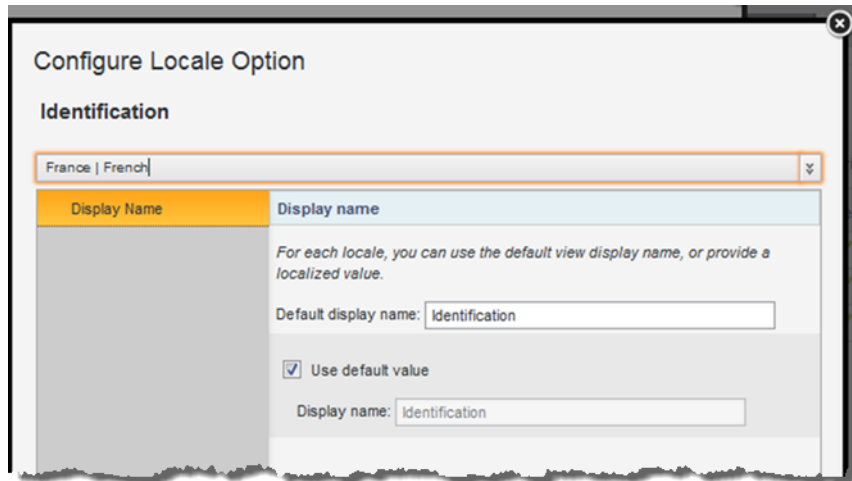


3. To localize the display name, click the localize icon next to the **Group Name** field.

On the localization dialog, the **Default display name** field contains the value you entered in the **Group Name** field. If you edit the value here, it also is updated on the **Edit Group** dialog and in the group list.

For each locale:

- a. From the locale list, select the locale that you want to configure the display name value of.



- b. To use the default value, leave the **Use default value** check box selected.
  - c. To provide a localized value for the selected locale, deselect the check box. In the **Display name** field, enter the localized value.
  - d. After entering the localized values, click **Apply**.
4. For a group in a base view, to have this group enabled on the **Available Refinements** panel, select the **Include this group in Available Refinements** check box.

Note that because the **Available Refinements** component only uses base views, this check box is not displayed for groups in other views.

Also, even if you enable the group, the attributes in the group are only displayed if they are configured on the **Views** page to display on the **Available Refinements** panel. See [Configuring the Available Refinements behavior for base view attributes](#).

5. To have this group enabled when displaying record details information for this view, including on the **Record Details** and **Compare** dialogs, and in the default configuration for a **Results Table** component, check the **Include this group in record details** check box.
6. To save the changes, click **Apply**.

## Configuring the group membership

For each group, you select the specific attributes that belong to the group.

If an attribute has locale-specific versions, you must add them manually to the group membership.

From the **Attribute Groups** page, to add and remove attributes for a group:

1. To add attributes:

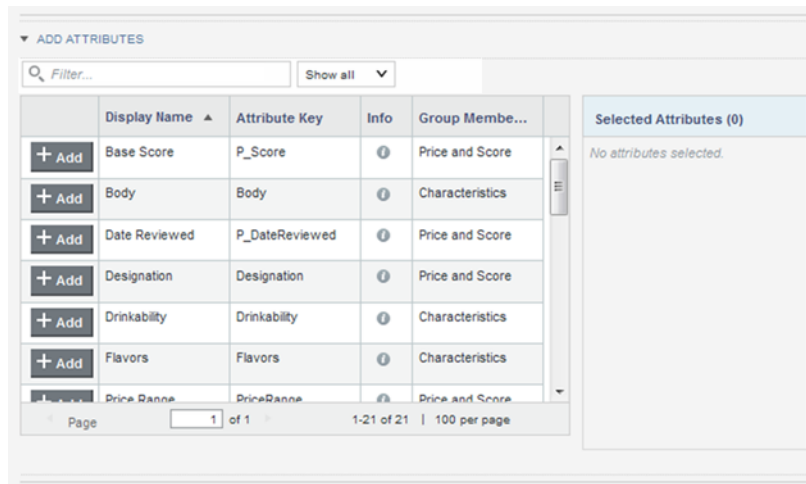
- a. Display the **New Group** or **Edit Group** dialog.

The **New Group** dialog displays when you create a new group.

To display the **Edit Group** dialog for an existing group, either:

- From the groups list, click the edit icon.
- From the attributes list for the group, click **+Attributes**.

The **Add attributes** section contains the list of attributes in the view that do not belong to the group.



- b. To filter the list, either:

- Use the search field above the attribute list to search based on the attribute display name.
- Use the data type list to only display attributes for a specific data type.

- c. In the attribute list, click **+Add** next to each attribute you want to add to the group.

The attributes are added to the **Selected Attributes** list. **+Add** for the selected attributes is disabled.

To remove a selected attribute from the list of attributes to add, click the delete icon for that attribute. **+Add** for that attribute is then re-enabled.

- d. To add the selected attributes, click **Apply**.

2. To remove an attribute from a group, in the attribute list, click the delete icon for that attribute.

## Configuring the sort order for a group

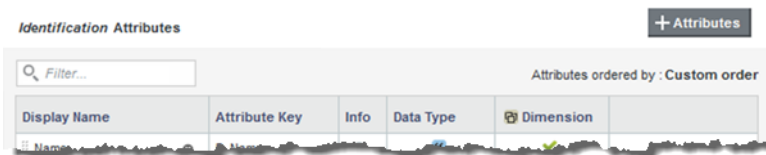
For each group, you can either sort the list using a metadata value, or set the sort order manually. The sort order on the **Attribute Groups** page determines the display order of attributes on components.

By default, the group is sorted by attribute display name in alphabetical order. For attributes that do not have a display name, the attribute key is used.

The **Attributes ordered by** field at the top of the attribute list shows the current sort order, including the metadata used and whether the sort is in ascending or descending order.

When you click a column heading, the sort order is updated to use that value for the sort. Clicking the same column heading twice switches the sort between ascending and descending order.

You also can move an attribute to a specific location in the list. When you do that, the sort order is updated to be **Custom order**.



If you then click a column heading, the sort order changes to use that metadata value.

## Saving the attribute group configuration and reloading the attribute group cache

Changes to the group configuration do not become permanent until you save them. To immediately reflect the saved group configuration changes on the project components, reload the attribute group cache.

When you have unsaved changes, you can use **Revert to Last Save** to undo those changes.

To save the current group configuration, click **Save**.

When Studio first loads the project data, it stores the attributes and groups in the cache. To reload the cache, so that the project components immediately reflect any updates to the group configuration, click **Reload Attribute Cache**.



---

## Exporting Project Data Sets

You can export data from project views and **Results Tables** to either HDFS or your computer.

### About exporting project data

You can export data from a project view and from a **Results Table** component. An export creates a file based on your data that you can either store in HDFS or download to your computer.

### Exporting project data

You can export data from a project view or a **Results Table** to either HDFS or your computer.

## About exporting project data

You can export data from a project view and from a **Results Table** component. An export creates a file based on your data that you can either store in HDFS or download to your computer.

The exported data is based on the current refinement state of your project, so it will not include attributes that have been filtered out or hidden. You will be able to preview a small sample of the exported data, as well as a list of attributes that will not be included, before you export.

When you export project data, you must specify whether you want the file exported to HDFS or your computer. If you export to HDFS, you must specify a directory to save the file to. If you export to your computer, the file is automatically downloaded once it has been generated.

You must also select the type of file your data is exported to. This can be one of the following:

- Delimited
- Avro
- A Hive table. This option is only available when you export to HDFS.

---

**Note:** The exported data uses attribute display names by default. However, if you are exporting to an Avro file, any attribute display names that contain invalid Avro characters will be replaced with their attribute keys.

---

### Transposing attributes

If the data set you're exporting contains multi-value attributes, you can have some or all of them transposed. Transposed attributes are replaced with sets of Boolean attributes representing their possible values.

---

---

**Note:** You cannot transpose attributes when exporting to an Avro file.

---

---

For example, consider a multi-value attribute named Color that has the unique values "Red", "Blue", and "Green". If the Color attribute were transposed, it would be replaced in the exported data by the Boolean attributes Red, Blue, and Green. A record that originally had its Color attribute set to "Red, Blue" would now contain the following:

- A Red attribute set to "True"
- A Green attribute set to "False"
- A Blue attribute set to "True"

Because multi-value attributes can have numerous possible values, you can specify a limit for the number of new columns added for each transposed attribute. Columns will only be added for the top values until the limit is reached.

### Problems viewing non-ASCII attribute values

If the data set you export contains non-ASCII attributes, the non-ASCII attribute values may be corrupt when you open the CSV file. For example, this data corruption problem occurs with Chinese, Japanese, and Korean attribute values and other non-ASCII language encodings.

To work around this issue:

1. Export the data set from Studio.
2. Instead of opening the CSV file, save the file locally.
3. Start Microsoft Excel and import the CSV file with UTF-8 encoding. You find this option in Excel under **Data > Get External Data From Text > Import** and then select UTF-8 in the import wizard.

After import, the attribute values render correctly.

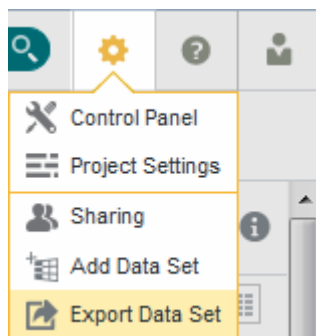
## Exporting project data

You can export data from a project view or a **Results Table** to either HDFS or your computer.

To export project data:

1. If you are exporting from a project view, click the Control Panel icon and select **Export Data Set**:





If you are exporting from a **Results Table**, select **Actions > Export**.

The **Export Data** dialog opens.

2. If you are exporting from a project view, select the data set to export from the **Data Selection** drop down list.

The **Preview** displays the first three records of the selected data set.

3. In the **Filename** field, enter a name for the file the data will be exported to.

By default, this is the name of the selected data set followed by the current date; for example, `Vehicle_Warranty_Claims_03022015`.

4. From the **Export to** list, select the location you want to export the file to:

- **HDFS**
- **My Computer**

5. If you are exporting to HDFS, do the following:

- a. In the **File Path** field, enter the path to the HDFS directory you want to export to.

By default, this path begins with `/user/bdd/exports`.

- b. If you want to create a new Hive table based on the exported data, select the **Create Hive table** check box and enter a name for the table in the **Table name** field.

By default, this is the name of the data set followed by the current date; for example, `Vehicle_Warranty_Claims_030215`.

Optionally, enter a description for the table in the **Add description** field.

6. From the **Type** list, select the type of file to export to:

- **Delimited**
- **Avro**

Note that if you select **Avro**, you will not be able to transpose any of your data's multi-value attributes.

7. If you are exporting to a delimited file, select delimiters for fields and multi-value attributes.

If you select **Other...**, you must also provide a new delimiter. This must be a single unicode character with a value of 127 or less.

8. If you want to transpose your data's multi-value attributes:

---

---

**Note:** This option is *not* available for Avro files.

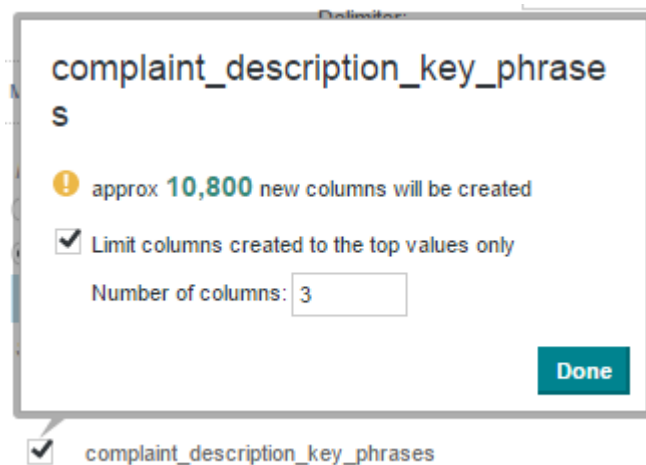
---

---

- a. Click **Multi-value attribute handling** and select **Transpose unique values**.
- b. Select the attribute(s) you want to transpose.

You can select all attributes by selecting the **Apply to all multi-value attributes** check box. However, if you want to limit the number of new columns created for each transposed attribute, you must select them one at a time.

- c. In each selected attribute's pop up, optionally check the **Limit columns created to the top values only** check box and enter a column limit.



This prevents your data from becoming cluttered with columns from attributes that have large numbers of possible values.

9. Click **Export**.

# Part VII

---

## Transforming Project Data



---

## Overview of Transforming Project Data

This section provides conceptual information about transforming project data sets using both default Studio transformations and the custom transform functions (Groovy code).

Use this section together with the *Transform API Reference* (generated Groovydoc) that is packaged with Big Data Discovery and also available on the Oracle Technology Network.

### [About transformations and transformation scripts](#)

**Transformations** are changes you can make to your project data set, after the source data has been processed and loaded into Studio. Transformations are effectively an ETL process to clean your data. Transformations can overwrite an existing attribute, modify attributes, or create new attributes. Any number of transformations can be combined into a transformation script. You run the script against the project data set. And if desired, you can create a new data set in the Catalog based on the transformed version.

### [About the Transform page user interface](#)

Provides basic information about the header and menu features.

### [Transformation script workflow](#)

At a high level, creating a transformation script and running it against your project data set involves the following steps:

### [Transformations workflow diagram](#)

When you run a transformation script against a project data set and commit it, Studio modifies the project data set (this is the private copy of the data set), but does not modify the underlying data set as it appears in the Catalog to other Studio users (the public copy of the data set).

### [About Groovy](#)

Groovy is a dynamically-typed scripting language. Code written in the Java language is valid in Groovy, so users not familiar with Groovy may resort to the Java syntax. All transform functions available for you in Studio are written in Groovy.

### [About transform functions](#)

**Transform functions** are customized Groovy functions available in Studio that you can include in your transformation scripts. Each transform function performs a specific operation on your data, from simple ones, such as converting an attribute to a different data type, to more complex ones, such as determining the overall sentiment of a document or a String of text.

### Transform locking

The **Transform** page in Studio has a locking mechanism to ensure that multiple users can't transform the same project data set at the same time.

### Preview mode

You can preview a transformation at any time by clicking **Preview**, to see the effect it will have on your project data set. Preview mode is also a useful debugging tool, as it detects any runtime errors or corner case exceptions your transformation contains.

## About transformations and transformation scripts

**Transformations** are changes you can make to your project data set, after the source data has been processed and loaded into Studio. Transformations are effectively an ETL process to clean your data. Transformations can overwrite an existing attribute, modify attributes, or create new attributes. Any number of transformations can be combined into a transformation script. You run the script against the project data set. And if desired, you can create a new data set in the Catalog based on the transformed version.

The **Transform** page allows you to make corrections and enhancements to a project data set. The transformations are within the scope of the project and do not affect the data set in the Catalog for other Studio users.

For example, you can use **Transform** page to modify:

- Change an attribute's data type
- Change capitalization of values
- Remove attributes or records
- Populate missing values
- Split columns into new ones (by creating new attributes)
- Add or remove attributes, or overwrite existing attributes
- Correct invalid or inconsistent values. For example, an attribute may have multiple versions of the same value (Wal-Mart, Walmart, Wal\*Mart).
- Group or bin values
- Extract information from values

You can also provide additional information about the data or provide new ways to use the values. For example, you can:

- Identify commonly used terms
- Analyze sentiment

### Custom transforms in a transformation script

You can also use the Groovy scripting language and a list of predefined Groovy-based **transform functions** available in Studio, to create a **transformation script**. Transformation scripts are collections of various transformations; they can contain any of the **transform functions**.

You can also write your own transformations from scratch using Groovy, within the same **Transform** page of Studio, using the **Transformation Editor**.

## Running a transformation script

You run a transformation script by clicking **Commit to project** in the **Transform** editor. When you commit a transformation script to a project, the script runs against the project data set and modifies the project data set with each transform step in the script. However, the operations do not affect the public data set in the Catalog on which that project is based.

You can preview the result of changes before you commit the script. While the script is running, you can check its progress in the **Notifications** panel.

After you transform a project data set, you can optionally create a new data set based on your modifications. This allows other Studio users to access the new data set in Catalog.

## Size of the data set to transform

If the source data is larger than one million records, the source data is automatically sampled during the ingest process to give you a data set limited to a maximum of one million records.

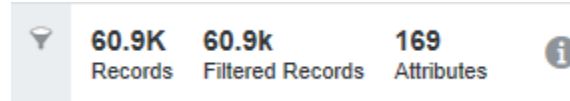
If the source data is smaller than one million records, the project contains the full data set rather than a sampled data set.

## About the Transform page user interface

Provides basic information about the header and menu features.

### Transform page header information

At the top of **Transform** is a summary of the project data set. For example:



This summary includes:

- The number of records in the sample being used
- The number of records in the entire data set
- The number of records in the sample that match the current refinements
- The number of attributes in the data set
- A link to show/hide data set properties

### Transform menu

Underneath the header is the attribute menu. It contains all the transformations that you can apply to an attribute. For example:



This menu includes transforms grouped into the following categories:

- **Shaping** - transforms to filter attributes, delete attributes, aggregate attributes, and join data sets.
- **Basic** - transforms to perform basic text manipulation (split, trim, and so on).
- **Convert** - transforms to convert an attribute data type to another data type.
- **Advanced** - transforms for more sophisticated operations such as extracting information from attributes and creating new attributes based on geography, entities, key phrases, and so on.
- **Editor** - displays the **Transformation Editor** where you can write custom transformation steps and add them to a transformation script.

The page header can also contain:

- A value indicating the total number of transforms in the transformation script.
- A value indicating the number of uncommitted transforms in the transformation script.

## Transformation script workflow

At a high level, creating a transformation script and running it against your project data set involves the following steps:

1. Create a transformation script using transforms and optionally custom transform functions.
2. Use preview to debug your transformation and view its effects on your data.
3. Save the transformation to your transformation script.
4. Edit your transformation script by rearranging, modifying, and deleting individual transformations.
5. Run your script against the project data set. This updates your copy of the project data set (it is a sample of the source Hive table), and makes it available in **Discover** area of Studio.

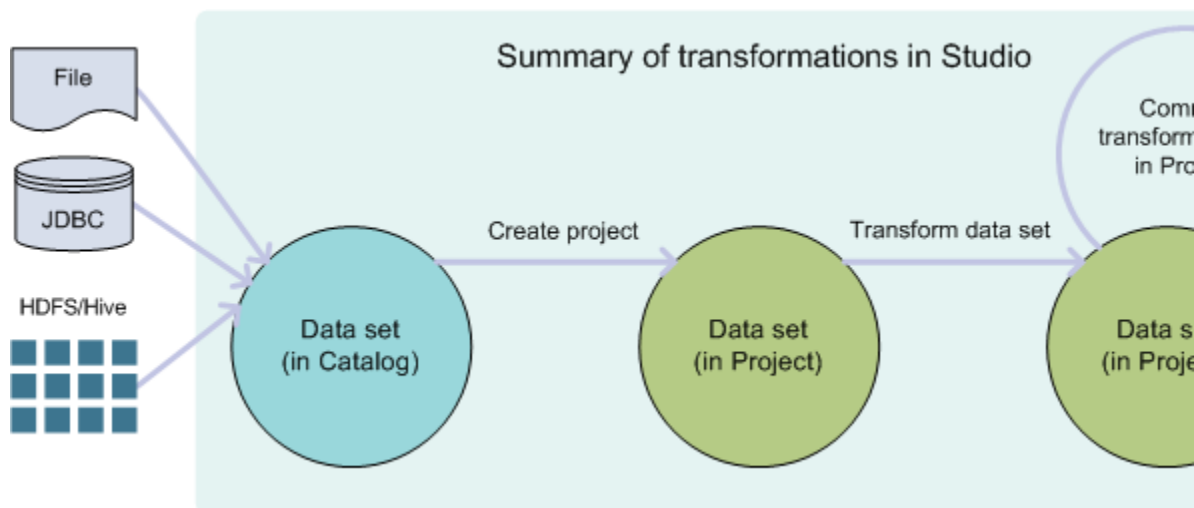
## Transformations workflow diagram

When you run a transformation script against a project data set and commit it, Studio modifies the project data set (this is the private copy of the data set), but does not modify the underlying data set as it appears in the Catalog to other Studio users (the public copy of the data set).

As shown in this workflow diagram, you create a project based on a data set and transform the private copy of the data set in your project by committing the transformation script.

As you change the transformation script, you can commit it again as necessary to continue modifying the project data set. The transformations apply to your version of the data set in project (this is the private copy of the data set); they do not affect the public version of this data set in Catalog.





### Interaction of transformations with updates

Here is a summary of how changes you make in **Transform** interact with updates:

- **Data sets loaded in Studio and from the JDBC source.** If you have a newer version of the Studio-uploaded file, you can reload it in Studio. The reloaded public version of the data set appears in the Catalog. See [About updating a data set in the Catalog](#).

You can apply a transformation to the private copy of this data set in your project.

When a reloaded version of this data set appears in the Catalog, the private copy in your project is not automatically updated. Studio prompts you to `Accept data updates`. If you do, Studio creates a new private copy of the reloaded public data set and your transformations are applied to that private copy.

- **Data sets backed by Hive and loaded with DP CLI.**

If you have not transformed this data set in your project, you can update it with DP CLI.

Once you transform the DP CLI-loaded data set in your project, you can update it with newer or additional data using DP CLI if the data set has been already loaded in full or is configured for updates using **Configure for Updates**.

If the data set has not been loaded in full or configured for updates, and you apply transformations, they are applied to the private version of this data set in your project. This means that to run transformations on data sets which you want to continuously update with DP CLI, the data sets must be loaded in full or configured for updates.

## About Groovy

Groovy is a dynamically-typed scripting language. Code written in the Java language is valid in Groovy, so users not familiar with Groovy may resort to the Java syntax. All transform functions available for you in Studio are written in Groovy.

Groovy was chosen as the basis for the Transform API because it is flexible and easy to use. Additionally, although it is a dynamic language, it can use static compilation and static type checking to make it less error-prone at runtime.

Studio lets you use many features of the Groovy language when writing your own custom transformations; it does, however, impose a few restrictions for security reasons. For more information, see [Unsupported Groovy language features](#).

You can find more information on Groovy, including tutorials, in the [Groovy documentation](#).

## About transform functions

**Transform functions** are customized Groovy functions available in Studio that you can include in your transformation scripts. Each transform function performs a specific operation on your data, from simple ones, such as converting an attribute to a different data type, to more complex ones, such as determining the overall sentiment of a document or a String of text.

Studio provides these types of custom transform functions:

- **Conversion functions** convert values to different data types.
- **Date functions** perform actions on Date objects, such as adding a specific amount of time to a Date.
- **Enrichment functions** are based on Data Enrichment modules in Studio. You can use them to extract complex information from your data.
- **Geocode functions** perform actions on Geocode objects, such as calculating the distance between two Geocode objects.
- **Math functions** perform mathematical operations on numerical values.
- **Set functions** perform different actions on sets of values on multi-value attributes in Studio, such as obtaining the size of the value set, checking whether a set is empty, or converting a multi-value attribute to a single-value attribute. Set functions only work on multi-value (also known as multi-assign) attributes.
- **String functions** perform different actions on String values, such as concatenating two String values, or splitting a single String into multiple values.

## Transform locking

The **Transform** page in Studio has a locking mechanism to ensure that multiple users can't transform the same project data set at the same time.

**Transform** locks a data set when a user previews or saves a transformation (that is, when the user clicks **Preview** or **Apply to Script** from the **Transformation Editor**). The lock remains set for 30 minutes, which is extended each time the user previews or saves a transformation, or while they are actively working in the **Transformation Editor**.

**Transform** locks the data set when a user clicks **Commit to Project**, which runs the transformation script against the project's data set. This lock applies to all users, including the one who ran the script.

A lock on a data set remains set until any of these conditions are met:

- You close the browser window that is running Studio.
- You navigate to a different URL in the browser window that is running Studio.

- The script finishes running.
- Studio times out from inactivity after a period of 30 minutes.
- The session is ended by the user signing out of Studio.

When a data set is locked, users can only perform the following actions:

- View the data set
- Copy transformation scripts
- Toggle rows and values
- Sort columns

## Preview mode

You can preview a transformation at any time by clicking **Preview**, to see the effect it will have on your project data set. Preview mode is also a useful debugging tool, as it detects any runtime errors or corner case exceptions your transformation contains.

When you click **Preview**, the **Transformation Editor** updates the transformation script.

When you preview a transformation, **Transform** finds and displays runtime errors that weren't detected by the static parser. It is therefore recommended that you preview your transformations and fix any errors they contain before saving them to your transformation script. You can revert the changes made in the preview by clicking **Cancel**.

Preview only updates your project data set in Studio's internal files backing the **Transform**; it does not affect any data sets in a Dgraph database, and the results are not visible to other users of your project in Studio. Additionally, your project data set is still associated with its source Hive data, so any changes made to the source are still reflected within your project.



---

# Transforming Project Data Sets

This section describes how to correct and enrich data sets within a project.

## Overview of transforming an attribute

All transforms follow the same basic work flow.

## Transform page controls

There are a number of general controls on the **Transform** page to toggle the data grid, filter and sort, and display whitespace in attribute values.

## Basic transform operations

Simple transforms such as splitting attributes, trimming attributes, or changing the case of attributes are listed on the **Basic** tab of the **Transform** page.

## Conversion transform operations

Conversion transforms convert the data type of an attribute that you select to another data type. They are listed on the **Convert** tab of the **Transform** page.

## Advanced transform operations

More advanced transforms such as creating geo heirarchy attributes, extracting entities or phrases, managing null values and so on are listed on the **Advanced** tab of the **Transform** page.

## Shaping transform operations

Shaping transforms are listed on the **Shaping** tab of the **Transform** page.

## Creating custom transformations

If the default transforms do not meet your needs, you can also create and add custom transformations to a transformation script on the **Editor** tab of the **Transform** page.

## Managing and executing the transformation script

The transformation script tracks and executes the transformations.

## Creating a new data set from the transformed data

After running a transformation script, you can create a new data set in the Catalog based on the transformed project data.

## Sharing a transform script

The ability to publish and share scripts provides a useful starting point for other projects to build a transform script.

## Tips for better performance

There are some simple things you can do while designing a transform script to improve its overall performance.

## Overview of transforming an attribute

All transforms follow the same basic work flow.

The high-level process:

1. You select the transformation that you want to apply to an attribute. If the transformation does not require any configuration, then it is added immediately to the transformation script. If the transformation does require configuration, then a **Transform Editor** displays so you can provide the transformation-specific configuration.
2. Determine the target attribute for the transformed values. Some transformations require that you use the transformation to create a new attribute. In this case, in the **New Attribute Name** field, type the name for the new attribute.

Other transformations allow you to choose whether to create a new attribute or use the transformed values to overwrite the current attribute.

- The default option is to create a new attribute.
  - To overwrite the current attribute, click **Apply to existing attribute**.
3. To display the effect of the transformation on the **Transform** data grid, click **Preview**.
  4. To add the transformation to the transformation script, click **Add to Script**.

## Transform page controls

There are a number of general controls on the **Transform** page to toggle the data grid, filter and sort, and display whitespace in attribute values.

### [Toggling the Transform data grid view](#)

Most of the **Transform** page consists of the data grid. In the grid, each column represents an attribute from the project data set.

### [Filtering and sorting the displayed attributes and values](#)

**Transform** provides sorting and filtering options to help you focus on the attributes you want to work with.

### [Displaying whitespace characters in attribute values](#)

To help gauge the amount of cleanup that might be needed in an attribute value, you can display spaces, tabs, and paragraph breaks as whitespace characters.

## Toggling the Transform data grid view

Most of the **Transform** page consists of the data grid. In the grid, each column represents an attribute from the project data set.

The data grid can display either:

- An short list of records from the project data sample. This is a row view. In this view, each row represents a record.
- Lists of attribute values. For each attribute, this view lists the unique values found in the data sample, listed in descending order by frequency.

To switch to the record list, click the row view icon.

To switch to the value list view, click the value list icon. This view shows attribute value distribution in a column display. (On the Transform page, this a column display of the same information you see in an attribute tile on the Explore page.)

The header for each attribute column contains:

- An icon to indicate the data type for the attribute.
- The attribute menu, used to access transformation options for that attribute.
- A data quality bar to indicate the number of records with valid or missing values for the attribute. You can click either group of values in the quality bar to refine the data.

## Filtering and sorting the displayed attributes and values

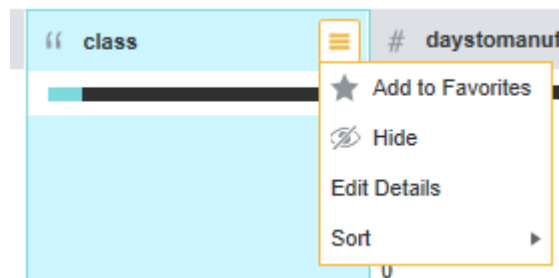
**Transform** provides sorting and filtering options to help you focus on the attributes you want to work with.

### Setting attributes as hidden or favorite

The **Favorite** and **Hide** options allow you to mark attributes that you are more interested in or that you do not want to see. These options only apply within a project.

To set an attribute as a favorite:

1. Select a column to highlight it.
2. From the attribute menu, select **Add to Favorites**. For example:



This move the column to the left most position of the data grid.

And you can then use the **Filter Attributes > Favorites** option to only display the attributes you mark as a favorite.

To hide an attribute:

1. Select a column to highlight it.
2. From the attribute menu, select **Hide**.

This removes the attribute from the grid.

And you can then use the **Filter Attributes > Hidden** filter option to only display the hidden attributes.

To remove the favorite or hidden flag for the attribute, click the **Star** or **Hide** option again.

When you set an attribute as hidden or favorite in the **Transform** page, the attribute is also treated the same way in **Explore** and for all users of the project.

### Filtering the displayed attributes

In the **Transform** header are options for filtering the displayed attributes based on:

- The attribute name
- Any tags the attribute may have
- The attribute data type
- Any notes the attribute may have
- Any semantic types assigned to the attribute
- Whether an attribute is marked as favorite or hidden

### Filtering the attribute values

You can use the **Available Refinements** panel to filter the displayed data based on attribute values. See [Filtering Based on Refinements](#).

You can also use the data quality bar to refine the data to only include records with valid, invalid, or missing values.

The refinement state can be included in custom transformations, and also used to select records to delete from the project data set.

### Sorting the displayed attributes

In **Transform**, you can sort the attributes:

- Alphabetically by name
- By the percentage of null values for the attribute
- Based on an algorithm that places attributes with more information potential at the top of the list. This is the default.
- Based on the relationship to a selected attribute.

This sort suggests how closely related the attribute is to others in the data set and sorts the attributes accordingly. This sort suggests what one attribute can tell you about the distribution of other attributes. For example, you might sort a `complaint` attribute in a manufacturing data set and find that a `supplier name` attribute is closely related.

This sort does not work with attributes that have hundreds of distinct values, so those attributes are marked as ineligible for sort by relationship. Also, this sort does not evaluate null values; attributes that have a large number of null values are moved toward the end of the sorted list.

When you sort based on an attribute, the selected attribute is moved to the front of the list, and the other attributes are displayed in order based on that attribute.

To sort alphabetically by name, select **By name** from the **Sort** list.

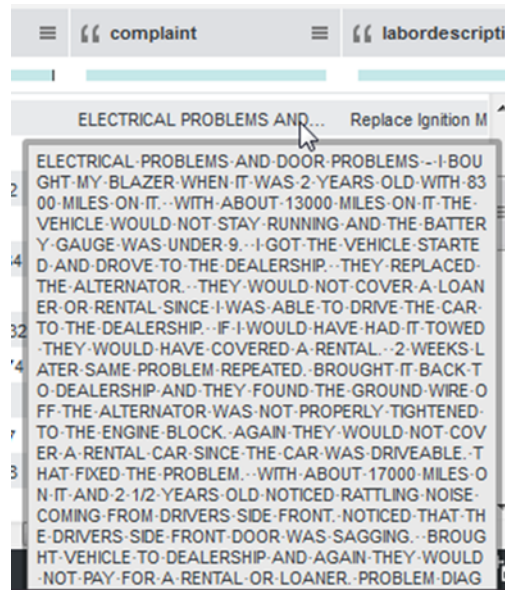
To sort based on the relationship to a selected attribute, select **Relationship to an attribute** from the **Sort** list and choose an attribute to sort by.

## Displaying whitespace characters in attribute values

To help gauge the amount of cleanup that might be needed in an attribute value, you can display spaces, tabs, and paragraph breaks as whitespace characters.



When the whitespace characters are shown, tooltip displays the full value with the whitespace characters when you hover the mouse over the value.



To display whitespace characters in attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute and select the column.
4. Click **Show whitespace**.

## Basic transform operations

Simple transforms such as splitting attributes, trimming attributes, or changing the case of attributes are listed on the **Basic** tab of the **Transform** page.

### Splitting String attribute values

The **Split** transformation creates one or more new attributes from the original value. You can only split String attributes.

### Trimming spaces from String values

The **Trim** transformation removes white spaces from the beginning or end of String attribute values.

### Selecting subsets of date/time values

For date/time attributes, you can truncate the date/time values to only include a specific subset of the date/time units.

### Changing the capitalization of String values

When you change the capitalization, the transformation is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

### Performing mathematical operations on numeric values

For numeric attributes, you can perform mathematical operations. The available operations change based on whether the attribute you select is an integer, long, or double value.

## Splitting String attribute values

The **Split** transformation creates one or more new attributes from the original value. You can only split String attributes.

For example, the values of a Vehicle Type attribute for vehicles consists of the vehicle make and model separated by a colon (Honda:Civic, Toyota:Camry). You could then use the split transformation to create a Make attribute and a Model attribute.

You can split a value:

- Based on a delimiter.

For example, you might split the attribute after a comma or colon. If the delimiter occurs multiple times, the value is split after the first occurrence.

- At a specific position in the text.

For example, you might split the attribute after the fourth character.

So if a Vehicle Type included the year (1995 Civic, 2003 Camry), then by splitting the value after the fourth character, you could create a Year attribute and a Model attribute.

- After specific text.

For example, split the attribute after the word "type".

To split String attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute that contains values you want to split into multiple values and select the column.
4. From the transform menu, select **Basic > Split**.
5. Choose a split method from the list and its corresponding delimited or position (as described above).
6. In **New Attribute Name**, specify the name of the attribute you want to create based on the split.
7. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Trimming spaces from String values

The **Trim** transformation removes white spaces from the beginning or end of String attribute values.

To trim spaces from String values:

From the attribute menu, under **Quick Transformation**, select **Basic > Trim**.

When you trim white spaces, the transformation is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

1. In the Catalog, select a project.
2. Select **Transform**.
3. Select the column of the attribute you want to change.
4. From the transform menu, select **Basic > Trim**.

When you select a **Trim** transformation, it is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Selecting subsets of date/time values

For date/time attributes, you can truncate the date/time values to only include a specific subset of the date/time units.

For example, if the attribute values include both the date and a time, you can truncate the values to only include the date.

To select subsets of date/time values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Select the column of the attribute you want to truncate.
4. From the transform menu, select **Basic > Truncate date**.

When you select a **Truncate date** transformation, it is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Changing the capitalization of String values

When you change the capitalization, the transformation is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

To change the capitalization of String values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute that contains values you want to modify and select the column.

4. Choose any of the following:

- To change the value to be all uppercase, select **Basic > Uppercase** .
- To change the value to be all lowercase, select **Basic > Lowercase**.
- To change the value to use title case, select **Basic > Titlecase**.

## Performing mathematical operations on numeric values

For numeric attributes, you can perform mathematical operations. The available operations change based on whether the attribute you select is an integer, long, or double value.

The options can include:

- Getting the absolute value
- Rounding decimal values
- Change a real number to the largest previous integer (floor) or the smallest following integer (ceiling).

To perform mathematical operations on numeric values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Select the column of the attribute you want to change.
4. From the transform menu, select **Basic** and the mathematical operation you want to perform on the attribute.

When you select a numeric transformation, it is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Conversion transform operations

Conversion transforms convert the data type of an attribute that you select to another data type. They are listed on the **Convert** tab of the **Transform** page.

### [Changing the data type of an attribute](#)

You can change the data type assigned to an attribute.

## Changing the data type of an attribute

You can change the data type assigned to an attribute.

For example, an attribute might contain mostly date values, but be classified as a String because of missing values or an unrecognized format. To take advantage of the features associated with date/time attributes, you can change the data type to date/time, then use other transformations to clean up the values.

Any data type can be transformed to a String. For other data types:

- Strings can be converted to date/time, numeric, or geocode data types
- Date/time values can be converted to numbers, or to times
- Long numbers can be converted to dates or geocodes

When converting values to or from date/time or time values, you must specify the date/time format to use. The acceptable formats use the Java SimpleDateFormat syntax (<http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>).

When converting to geocode values, you must specify the latitude and longitude.

To change the data type of an attribute:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Select the column of the attribute you want to change.
4. From the transform menu, select **Convert to <type>** and the data type transform you want.

When you select a **Convert to <type>** transformation, it is immediately added to the transformation script. There is no configuration, and the change always overwrites the selected attribute.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Advanced transform operations

More advanced transforms such as creating geo heirarchy attributes, extracting entities or phrases, managing null values and so on are listed on the **Advanced** tab of the **Transform** page.

### [Obtaining geocode information from attribute values](#)

You can create a full set of geocode attributes (country, region, subregion, etc.) based on any String attribute that has partial address information. The full set of geocode attributes allows you to display the geographic information in a Thematic Map.

### [Extracting parts of a date from attribute values](#)

You can extract date patterns from a Date Time attribute and use that information to create a new attribute or overwrite an existing attribute.

### [Extracting names of people, places, or organizations from attribute values](#)

The **Entity Extraction** transformation searches attributes for names of people, places, or organizations. The resulting attribute contains a delimited list of the found entities. If the transformation does not find any entities, the resulting attribute is empty.

### [Extracting key phrases from attribute values](#)

The **Extract key phrases** transform extracts key phrases from a String attribute and creates a list of phrases in a new multi-assign attribute. The transform calculates key phrases using the TF/IDF algorithm, which takes the total number of times each term appears within the String and

offsets that value by the number of times it appears within a larger body of work.

#### Providing a whitelist to extract terms from attribute values

If you know the exact words you want to extract from an attribute, then you can use the **Tag from Whitelist** transformation to extract them and create a new attribute. You can also extract them, replace them with another term, and create a new attribute.

#### Binning numeric attribute values

The binning transformation takes values for a numeric attribute and groups them into bins. The bins are then used as the values of a new attribute.

#### Grouping attribute values

Grouping attribute values takes the attribute values and puts them into groups, with each group given a unique name.

#### Managing null values

The **Manage null values** transformation replaces null values for an attribute.

## Obtaining geocode information from attribute values

You can create a full set of geocode attributes (country, region, subregion, etc.) based on any String attribute that has partial address information. The full set of geocode attributes allows you to display the geographic information in a Thematic Map.

For details about using the geocode attributes in a chart, see [Thematic Map](#). For details about using geocode (geotag) functions and their data type requirements, see [Enrichment functions](#).

To obtain geocode information from attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute, of type String, that has geographic information.  
For example, this might be a city, state, or postal code attribute.
4. From the transform menu, select **Advanced > Create geo heirarchy**.
5. In **Select a <location> attribute...**, do one of the following depending on your data set:

(where <location> represents each control such as country, region, subregion, city and postcode)

- If your data set already has a <location> attribute, select **Attribute** from the list and specify the attribute name.
- If your data set does not have a <location> attribute, create a country attribute manually by selecting **Static Value** from the first list and selecting a country name from the second list.

- If the attribute is not applicable, select **N/A**. In other words, this attribute may be unknown; it may not exist, or may not apply. For example, some countries do not have providences, states, or counties.
  - Repeat this step to specify the values of **Select a country attribute...**, **Select a region attribute...**, **Select a subregion attribute...**, **Select a city attribute...** and **Select a postcode attribute...**
6. In **New Attribute Name (prefix)**, specify just the prefix value for the new geocode attributes.
  7. Decide how to handle ambiguous location matches in the new geocode attribute values by selecting one of the following:
    - **Return the most populated relevant location** - sets the ambiguous value to the most populated location match. For example, an ambiguous match to Boston is set to Boston Massachusetts rather than Boston, Georgia.
    - **Return a null value** - sets the ambiguous value to null.
  8. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

#### **Example 19-1 Example**

Suppose your source data has attributes named `city`, `county`, and `zip_code` with values that represent city names, county names, and postal codes in the state of California:

All Attributes		
city	county	zip_code
Truckee	Nevada	96162
Truckee	Nevada	96161
Truckee	Nevada	96160
South Lake Tahoe	El Dorado	96158
Vinton	Plumas	96135
South Lake Tahoe	El Dorado	96157
Tulelake	Siskiyou	96134
South Lake Tahoe	El Dorado	96156
Topaz	Mono	96133
Davis	Yolo	95618
South Lake Tahoe	El Dorado	96155
Janesville	Lassen	96114
Termo	Lassen	96132
Bakersfield	Kern	93390

However, the source data does not have country or state information. You can create the geocode attributes by setting the **Create Geographic Heirarchy** as follows:

### Geographic Hierarchy

Select a country attribute...

Static Value ▼

United States ▼

Select a region attribute...

Static Value ▼

California ▼

Select a subregion attribute...

Attribute ▼

county ▼

In the event of ambiguous location matches:

Return the most populated relevant location

Return null value

After you click Preview, you see that Studio created the following new geocode attributes highlighted in yellow:

All Attributes			
city	county	test_geo_city	test_geo_country
Truckee	Nevada	Foresthill	US
Truckee	Nevada	Truckee	US
Truckee	Nevada	Truckee	US
South Lake Tahoe	El Dorado	South Lake Tahoe	US
Vinton	Plumas	East Quincy	US
South Lake Tahoe	El Dorado	South Lake Tahoe	US
Tulelake	Siskiyou	Tulelake	US
South Lake Tahoe	El Dorado	South Lake Tahoe	US
Topaz	Mono	Smith Valley	US
Davis	Yolo	Davis	US
South Lake Tahoe	El Dorado	South Lake Tahoe	US
Janesville	Lassen	Janesville	US
Termo	Lassen	Susanville	US
Bakersfield	Kern	Lamont	US

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).



## Extracting parts of a date from attribute values

You can extract date patterns from a Date Time attribute and use that information to create a new attribute or overwrite an existing attribute.

For example, this may be useful if an attribute has a timestamp and you want to extract the day of the week value and create a new attribute called Day.

To extract parts of a date from attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute of type Date Time that has date information that you want to extract.
4. From the transform menu, select **Basic > Extract date part**.
5. Specify a date pattern for the attribute values in one of the following ways:
  - In **Select a data part...**, select a date format from the list of pre-defined format patterns.
  - Alternatively, you can enable **Create a custom data part** and specify a custom date pattern. Specify only the date pattern not a label for the pattern. For example, if you want to create a custom pattern for Year-Month-Day, you specify only `YYYY-MM-dd`.
6. Select an output time zone for the date pattern. This setting normalizes the attribute values to the time zone you specify.
7. Select an output locale from the available list. This setting converts character (non-numeric) date information to the locale you specify.

For example, if you extract a day pattern, such as `dd`, and set the output locale to Japanese, the new attribute values contain the day names in Japanese.
8. Indicate whether you want the **Extract Date Part** transformation to apply to the attribute you selected previously, or indicate whether you want to create a new attribute using the extracted date information.
9. If you selected **Create New Attribute**, then specify an attribute name.
10. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

### **Example 19-2 Example**

Suppose your source data has attributes named `dropoff_datetime` and you want to extract the day value and use it to create `dropoff_day`. You can create the attributes by setting the **Extract Date Part** as follows:

Shaping	Basic	Convert	Advanced	Editor
Truncate date	Extract date part			
Choose a part of your date time to extract				
Select Output Timezone: (UTC-05:00) Eastern Standard Ti... ▼			Output Locale: Eng	
Day (Full Name) - EEEE ▼		<input type="checkbox"/> Create a custom date part		

After you click **Preview**, you see that Studio creates the following new `dropoff_day` attribute highlighted in yellow:

All Attributes			
dropoff_datetime	dropoff_day	dropoff_geocode	dropoff_geocode
2013-12-03 08:02:00 UTC	Tuesday	40.731731 -73.983635	New York City
2013-12-02 11:48:00 UTC	Monday	40.766193 -73.956978	Manhattan
2013-03-28 11:43:44 UTC	Thursday	40.721569 -73.983788	New York City
2013-12-01 03:58:00 UTC	Saturday	40.754436 -73.973656	Long Island City
2013-03-29 11:06:00 UTC	Friday	40.762314 -73.969208	Manhattan
2013-03-27 00:04:03 UTC	Tuesday	40.717293 -74.012428	New York City
2013-10-06 23:23:56 UTC	Sunday	40.750393 -73.980492	Long Island City
2013-12-01 01:50:00 UTC	Saturday	40.694756 -73.994331	New York City
2013-11-16 09:35:00 UTC	Saturday	40.748928 -73.991417	Long Island City
2013-03-28 18:50:00 UTC	Thursday	40.762459 -73.968140	Manhattan

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Extracting names of people, places, or organizations from attribute values

The **Entity Extraction** transformation searches attributes for names of people, places, or organizations. The resulting attribute contains a delimited list of the found entities. If the transformation does not find any entities, the resulting attribute is empty.

For example, for the text: "In New York, the Metropolitan Museum of Art is the largest museum. Other popular museums include the Guggenheim (designed by Frank Lloyd Wright) and the Museum of Modern Art."

- If you run **Entity Extraction** to extract places, the resulting value would be something like "New York, Metropolitan Museum of Art, Guggenheim, Museum of Modern Art".
- If you run **Entity Extraction** to extract people, the resulting value would be "Frank Lloyd Wright".

To extract entity names from an attribute's values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute of type String that contains entity information that you want to extract and select the column.
4. From the transform menu, select **Advanced > Extract entities**.
5. Select the type of entities you want to extract (people, places, or organizations). You select one or all of the available entities.
6. Specify a prefix for the new attribute.

Studio creates a new attribute with the prefix value combined with a suffix of `<prefix>_person` (for People), `<prefix>_location` (for Places), and `<prefix>_organization` (for Organizations).

7. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Extracting key phrases from attribute values

The **Extract key phrases** transform extracts key phrases from a String attribute and creates a list of phrases in a new multi-assign attribute. The transform calculates key phrases using the TF/IDF algorithm, which takes the total number of times each term appears within the String and offsets that value by the number of times it appears within a larger body of work.

To extract key phrases from an attribute's value:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute that you want to extract phrases from and select the column.

4. From the transform menu, select **Advanced > Extract key phrases**.
5. In **Input language**, specify the language of the attribute value.  
This improves the accuracy of key phrase identification by applying a language specific identification model.
6. Select **Use smart casing for input text** to better handle documents that are predominantly in either title case or upper case.  
Smart casing helps the transform better identify key phrases by first converting upper case text to lower case and then running the phrase extraction on the lower-case text.
7. In **New Attribute Name**, specify the name of the attribute you want to create. By default, Studio creates a multi-assign attribute to store key phrases.
8. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Providing a whitelist to extract terms from attribute values

If you know the exact words you want to extract from an attribute, then you can use the **Tag from Whitelist** transformation to extract them and create a new attribute. You can also extract them, replace them with another term, and create a new attribute.

Each line in the editor represents a single tagging action. Types of available tagging actions include:

Tagging Action	Syntax	Examples
Extract a single term.	Specify the term on its own line.	red  If the term <code>red</code> is found, add it to the output attribute.
Extract a single term and replace it with another term.	Specify the term on its own line, add a tab, and specify the replacement term.	red<tab>color  If <code>red</code> is found it is replaced by <code>color</code> in the output attribute.
Extract several terms and replace it with one term. Replace a selected term or terms with another term.	This is the same as single term replacement but you use a new line for each term and its replacement.	red<tab>color yellow<tab>color blue<tab>color  If <code>red</code> , <code>yellow</code> , or <code>blue</code> are found, the term is extracted and replaced by <code>color</code> in the output attribute.

To provide a whitelist to extract terms from attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute, of type String, that has text you want to extract.
4. From the transform menu, select **Advanced > Tag from whitelist**.
5. Click **Enter Terms**.
6. Specify the terms you want to extract according to the syntax in the table above.

For example:

Enter Terms		<input type="checkbox"/> Match case	<input type="checkbox"/> Match whole words only
1	red color		
2	yellow color		
3	blue color		

7. If desired, specify whether you want a case-sensitive match or a whole word match (no sub-String matches) for the terms.
8. Click **Done**.
9. In **New Attribute Name**, specify the name of the attribute you want to create.
10. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Binning numeric attribute values

The binning transformation takes values for a numeric attribute and groups them into bins. The bins are then used as the values of a new attribute.

For example, from a Price attribute, you might create a new Price Range attribute such that:

- If Price is less than 20 dollars, then Price Range is "Inexpensive".
- If Price is greater than 20 dollars but less than 40 dollars, then Price Range is "Moderately Priced".
- If Price is greater than 40 dollars but less than 60 dollars, then Price Range is "Expensive".
- If Price is greater than 60 dollars, then Price Range is "Very Expensive".

To bin the values of a numeric attribute:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate one of the attributes that you want to bin and select the column.
4. From the transform menu, select **Advanced > Bin numeric values**.
5. For each bin that you add, specify the following:
  - a. A bin name. The bin names are used as the display name for the new attribute value.
  - b. A low value for the bin range. For the low value, you indicate whether values in the bin must be greater than or greater than or equal to the specified value.
  - c. A high value for the bin range. For the high value, you indicate whether values in the bin must be less than or less than or equal to the specified value.
6. For values that don't fit in any of the bins, select whether to:
  - Leave them as is without any binning.
  - Place them in another bin named Other.
7. Provide the name of the new attribute to contain the bins. When you bin values, you must always create a new attribute.
8. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Grouping attribute values

Grouping attribute values takes the attribute values and puts them into groups, with each group given a unique name.

You can use the groups to replace the current attribute values, or create a new attribute.

You might replace the current attribute values if you want to standardize values that are actually the same value, but with slightly different formatting or spelling across the data set.

For example, Walmart might show up as Wal-Mart, walmart, and Wal\*Mart. To fix these inconsistencies, you could create a Walmart group that includes all of these variants.

You might create a new attribute if you want to provide an attribute with a higher level of abstraction.

For example, you might want to group the values of a State attribute into a new Region attribute. You could then create visualizations based on either State or Region.

To group the values for an attribute:

1. In the Catalog, select a project.

2. Select **Transform**.
3. Locate one of the attributes that you want to group and select the column.
4. From the transform menu, select **Advanced > Group values**.
5. Click **+ Value Group**.  
A new empty group is added to the list, with a default name of "New Value Group 1". Each group name becomes an attribute value.
6. To rename the group, double-click the group name. Type the new name, then press **Enter**.
7. To populate a group, drag values from the value list and drop them on the group name.
8. Either specify a new attribute name for the group, or apply the new attribute group to the attribute you selected in Step 3.
9. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Managing null values

The **Manage null values** transformation replaces null values for an attribute.

You can fill in the null values using either:

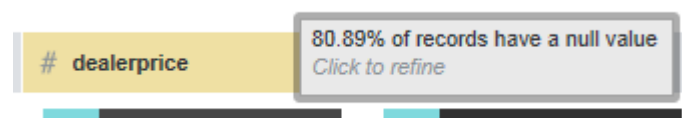
- The mostly commonly used value
- For numeric values, the mean of the attribute values
- A specific value that you specify.

For example, you could fill in all of the null values with "Not Provided" or "Not Applicable".

To fill in null values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Locate an attribute that contains null values you want to modify and select the column.

Remember the data quality bar shows the percentage of null values in black. For example:



4. From the transform menu, select **Advanced > Manage null values**.
5. Specify a replacement for the null values in one of the following ways:
  - Choose **Replace with Static Value > Most Frequent Value**. This option provides a pre-calculated value in grey.
  - Choose **Replace with Static Value > Attribute Mean**. This option provides a pre-calculated mean value in grey.
  - Choose **Replace with Static Value > Null filling value** and specify a value of your choice.
  - Choose **Delete Records** to delete all records that have a null value for the selected attribute.
6. Optionally, select **Create null indicator attribute** to create a new attribute, based on the column name you selected, that gets populated with either 1 or 0 for the value.

A value of 1 means a null value was replaced. A value of 0 means the original value was not modified because it was not null.
7. If you selected **Replace with Static Value**, then indicate whether to apply the change to the current attribute or specify a new attribute name.
8. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Shaping transform operations

Shaping transforms are listed on the **Shaping** tab of the **Transform** page.

Shaping includes all of the following data set modifications:

- Filtering rows out of a data set
- Deleting attributes from a data set
- Aggregating the values of one or more attributes into one or more new derived attribute
- Joining one or more data sets together

### [Filtering records from the project data set](#)

You can use the refinement state as a basis for removing records from the project data set. You can either remove records that match the refinement state, or remove records that do NOT match the refinement state.

### [Deleting an attribute](#)

You can delete an attribute from a project data set by using the quick transform menu option **Delete Attribute**.

### [About aggregating attribute values](#)

You aggregate attribute values by applying an operation (sum, average, min/max, etc.) to one or more attributes and then grouping the derived result by an attribute.



### Aggregating attribute values

The **Aggregate** transformation aggregates attributes values to derive one or more new attributes and group the result by an attribute.

### About joining data sets

Studio can perform a left outer join, inner join, or full outer join against one or more data sets based on a key, or compound key, that you create.

### Joining data sets

You can join one or more data sets by specifying primary and secondary sources, a join key, a join type, and then running the transform script.

## Filtering records from the project data set

You can use the refinement state as a basis for removing records from the project data set. You can either remove records that match the refinement state, or remove records that do NOT match the refinement state.

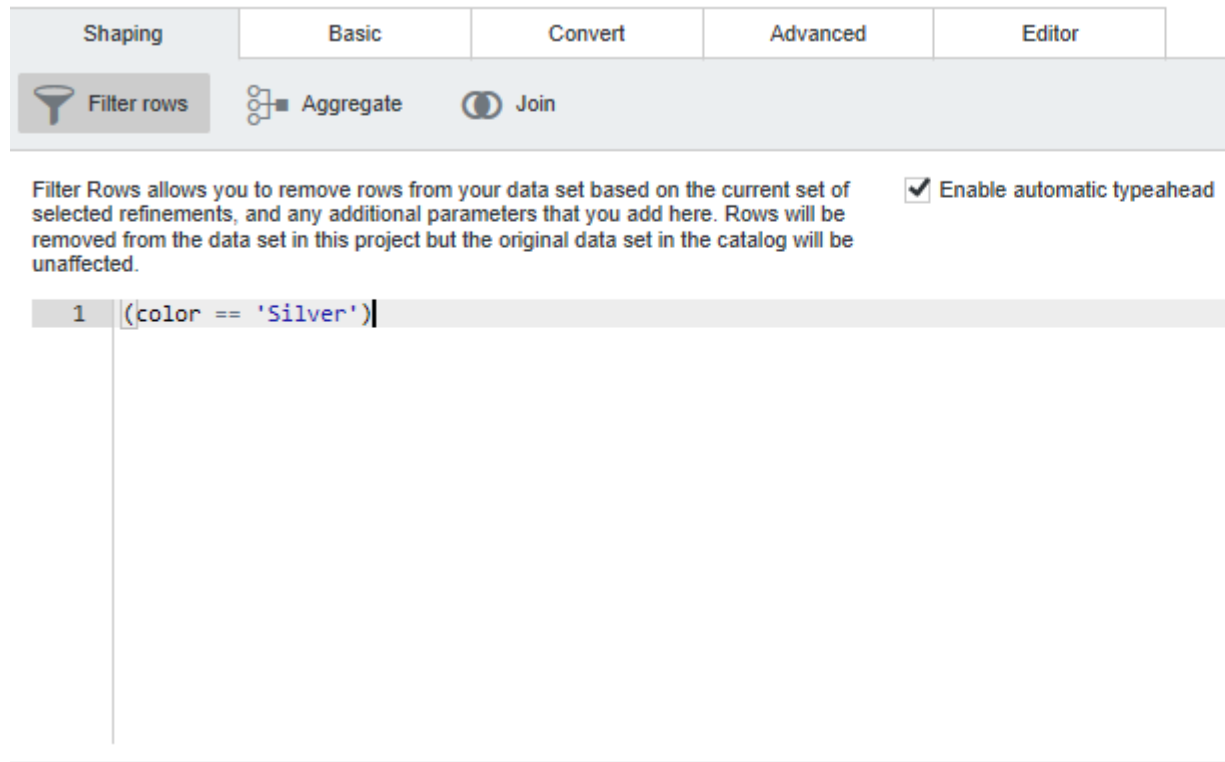
For example, you might want to filter out all records with null values for certain attributes. Or you may want to filter out all records except those with Canada as the country value.

If you filter the records, then they are permanently removed from the project data set. However, they are still in the original data set.

To delete records from the project data set:

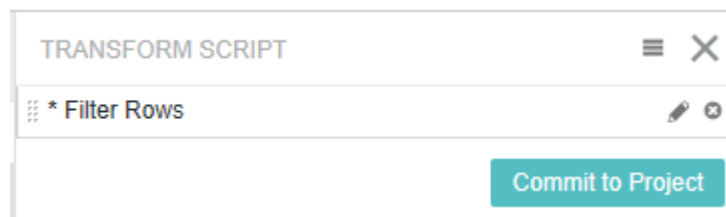
1. In the Catalog, select a project.
2. Select **Explore** and refine the data set to either represent the records you want to keep or exclude.
3. Select **Transform**.
4. From the transform menu, select **Shaping > Filter rows**.
5. In the **Transformation Editor**, verify that the transform reflects the refinement state you selected.

For example, this transform filters out records that have a color value of Silver:



6. Choose a radio button to indicate whether to delete records that do NOT match the refine by selection or delete records that match the refine by selection.
7. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.
8. Click the numbered bubble to expand the **Transform Script** window.
9. Click **Commit to Project** to run the script. If necessary, click the pencil icon to edit the transform before committing it.

For example:



## Deleting an attribute

You can delete an attribute from a project data set by using the quick transform menu option **Delete Attribute**.

To delete an attribute:

1. In the Catalog, select a project.

2. Select **Transform**.
3. Locate an attribute that you want to remove from the project data set.
4. From the transform menu, select **Shaping > Delete attribute**.

There is no editor that displays. The delete operation removes the attribute column from the table and also adds a **Delete** transformation step to the Transform Script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## About aggregating attribute values

You aggregate attribute values by applying an operation (sum, average, min/max, etc.) to one or more attributes and then grouping the derived result by an attribute.

For example, you could aggregate the average of a `Standard Cost` attribute and group the result by a `Product Category`. In that case, you get the average cost of each item in a particular product category.

Or, you could aggregate the minimum and maximum values of a `Standard Cost` attribute and group the results by a `Product Category`. In that case, you get the lowest cost of each item in a category and the highest cost of each item in a category.

Aggregating attribute values provides a way to consolidate data into a new derived attribute for further examination. And in some cases, the new derived attribute values are often used before you perform additional operations such joining one data set with another data set.

Some attributes have a data type that make them unavailable for aggregation. Specifically, multi-assign and geocode attributes display in the **Aggregate** editor as unavailable.

Some operators can only transform attributes of a particular data type. The `average`, `sum`, `standard dev` and `variance` operators can only be used with numeric attributes. The `min` and `max` operators can be used with numeric attributes and date time attributes.

### Aggregation operators

The supported aggregation operators are `sum`, `average`, `min`, `max`, `records with values`, and `unique values`. In addition to these operators, you might also have `set`, `variance`, and `standard dev` if the `df.advancedSparkAggregationsEnabled` property is set to `true` on the **Control Panel > Studio Settings** page.

**Table 19-1** Aggregation operators

Operator	Description
<code>sum</code>	Finds the sum of all values in the attribute.
<code>average</code>	Finds the average of all values in the attribute.
<code>min</code>	Finds the minimum value of all values in the attribute.
<code>max</code>	Finds the maximum value of all values in the attribute.
<code>records with values</code>	Finds the number of records with values in the attribute.

**Table 19-1 (Cont.) Aggregation operators**

Operator	Description
unique values	Finds the number of unique values in the attribute.
set	Optionally enabled. Finds the set of matching values in the attribute.
variance	Optionally enabled. Finds the variance between a set of values.
standard dev	Optionally enabled. Finds the standard deviation for the attribute values.

**Interaction with data set updates**

Aggregated attributes are calculated based on values from one point in time. If you run a transformation script with an aggregation and then reload or update a data set, you have to re-run the transform script with the aggregation using the updated data. In other words, the values in an aggregated attribute have been collapsed during the aggregation, and you need the un-aggregated values to run subsequent updates.

**Aggregating attribute values**

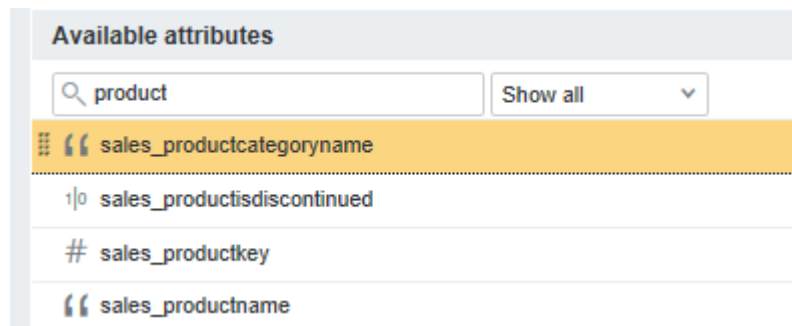
The **Aggregate** transformation aggregates attributes values to derive one or more new attributes and group the result by an attribute.

To aggregate attribute values:

1. In the Catalog, select a project.
2. Select **Transform**.
3. From the transform menu, select **Shaping > Aggregate**.
4. Specify the attribute to group by, as follows:
  - a. From the **Available attributes** list, locate and select an attribute that you want to group by.

You can filter attributes by name or by data type to locate them.

This example shows a selected the `sales_productcategoryname` attribute:



- b. Optionally, examine the **Available attributes preview** pane to check whether the attribute you selected to group by has the attribute value distribution that you expect. (This pane shows the value list view. )

This example shows the attribute value distribution of the `sales_productcategoryname` attribute:

**Available attributes preview**

sales\_productcategoryname

Bikes	24,800
Components	18,687
Clothing	12,267
Accessories	5,101

- c. If the attribute you examined is what you want to group by, drag and drop the attribute onto the **Grouping attributes** pane.
5. Specify the new aggregated attributes, as follows:

- a. From the **Available attributes** list, locate and select an attribute that you want to aggregate.

You can filter attributes by name or by data type to locate them.

- b. Drag and drop the attribute onto the **Aggregated attributes** pane and select an operator for the aggregation. (The default operator is average, but you can select any operator from the drop-down list.)

Studio creates a new attribute name with a new suffix of the form `_operatorName`.

- c. Optionally, click the name to make it editable and edit the new attribute name.

This example shows two new aggregated attributes. One creates a minimum value for a product's standard cost and the second creates a maximum value for a product's standard cost.

**Aggregated attributes**

# sales_productstandardcost_max	(max)	▼
# sales_productstandardcost_min	(min)	▼

6. Click **Preview** and examine the **Aggregated preview** pane to see the result of aggregating the values and grouping by the attribute you chose previously.

To continue the above example, Studio creates the new aggregated attributes `sales_productstandardcost_min` and

`sales_productstandardcost_max` and then grouped them by `sales_productcategoryname`. As a result of the aggregation, you can see the high and low cost grouped by the category:

Aggregation preview		
sales_productcategoryname	# sales_productstandardcost_max	# sales_productstandardcost_min
Components	868.6342	8.9866
Bikes	2,171.2942	294.5797
Accessories	44.88	0.8565
Clothing	41.5723	3.3623

7. Click **Add to Script** to save the transformation step to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## About joining data sets

Studio can perform a left outer join, inner join, or full outer join against one or more data sets based on a key, or compound key, that you create.

The primary data set displays the join changes and the secondary data set is not modified.

The updated records have an additional attribute, named *Data Source Name* that specifies which data sources, by name, contributed to the new records. For example, in a left join, every record is tagged with *dataset1*, but there might be some records that have no matching data from *dataset2*. Those records would only contain the value *dataset1*. All other records would contain both *dataset1* and *dataset2* as attribute values.

### Differences between joining and linking data sets

Joining data sets performs a full SQL join combining records from the primary and secondary sides of the join. Joining materializes new records that replace the data in the primary data set of your project.

Linking data sets links the data at query time for temporary use in Discover components. Both data sets continue to be stored as separate data sets with only a key (link) that connects them during queries. Linked data sets are not permanently joined in any way.

A link is logically similar to a database view. Links provide a way to temporarily look at relationships in multiple data sets without the level of persistence and data processing required by a join. For more information about linking, see [Linking Project Data Sets](#).

### Updates to joined data

Joins have the same update model as all other transforms. If you load a full data set or run an incremental update, Studio notifies you that the data set has changed and you can accept or reject the update in your project. If you accept an update, Studio reads the changes from the Hive table and re-runs the join operation.

### Limits to the sample size after joins

A join operation could result in a larger data set size. However, the `bdd.sampleSize` setting, on the **Data Processing Settings** page, limits the *sample size* that results from transforms such as Join, Aggregate, and FilterRows. In other words, a join operation never results in a sample size that exceeds the value of `bdd.sampleSize`.

### Joining data sets

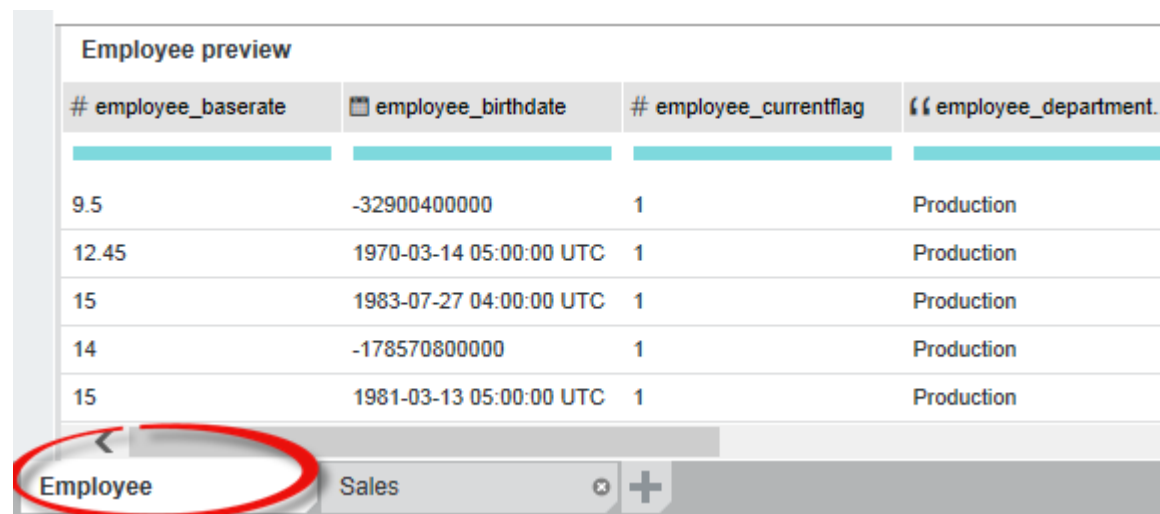
You can join one or more data sets by specifying primary and secondary sources, a join key, a join type, and then running the transform script.

Before performing a join, both data sets must be added to the project. If you are performing a self join, only one data set is required.

To join data sets:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Select a data set tab that you want to be the primary data set.

For example, this project has an employee data set and a sales data set. Selecting Employee makes it the primary data set:

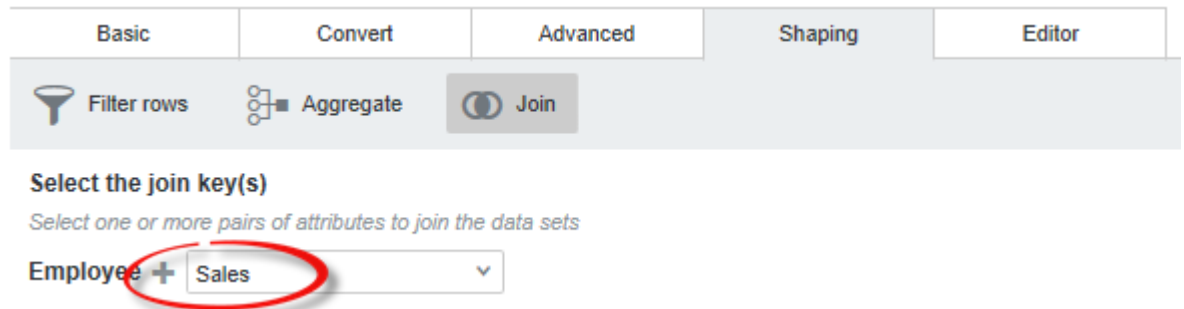


# employee_baserate	employee_birthdate	# employee_currentflag	employee_department
9.5	-32900400000	1	Production
12.45	1970-03-14 05:00:00 UTC	1	Production
15	1983-07-27 04:00:00 UTC	1	Production
14	-178570800000	1	Production
15	1981-03-13 05:00:00 UTC	1	Production

Employee Sales

4. From the transform menu, select **Shaping > Join**.
5. Confirm that the secondary data set that you want to join is selected.

For example, this project joins an employee data set to a sales data set:



6. Create a unique join key from attributes that are the same data type.
  - a. Click an attribute name from the *primary* side of the join to highlight it.

The attribute is added to the *left* side of the **Select Keys** list. Also note that Studio restricts available attributes on the right side to those of the same data type.
  - b. Select a key operator (equals, less than, greater than, etc) to determine how the key is built.

For example, `employee_fullname` on the primary side might *equal* `sales_employeefullname` on the secondary side.
  - c. Click an attribute name from the *secondary* side of the join to highlight it  

The attribute is added to the *right* side of the **Select Keys** list. For example, this key is `employee_fullname equals sales_employeefullname`:



The screenshot shows the 'Join' configuration interface. At the top, there are tabs for 'Shaping', 'Basic', 'Convert', 'Advanced', and 'Editor'. Below these are icons for 'Filter rows', 'Aggregate', and 'Join'. The 'Join' tab is selected. The main area is titled 'Select the join key(s)' with a subtitle 'Select one or more pairs of attributes to join the data sets'. Below this, there is a dropdown menu showing 'Employee + Sales'. The interface is divided into two columns: 'Employee attributes' and 'Sales attributes'. Each column has a search filter and a 'Show all' dropdown. The 'Employee attributes' list includes: # employee\_baserate, # employee\_birthdate, # employee\_currentflag, employee\_departmentname, employee\_emailaddress, employee\_firstname, and employee\_fullname. The 'Sales attributes' list includes: sales\_carriertrackingnumber, sales\_customerponumer, sales\_daynameofweek, sales\_employeefullname, sales\_employeelastname, sales\_fiscalquarter, and sales\_monthname. The 'employee\_fullname' and 'sales\_employeefullname' attributes are highlighted in yellow.

- d. If a single attribute is not sufficient to guarantee uniqueness across the joined record set, click the + icon and repeat the above steps a - c to build a compound key using a second set of attributes.
7. Select the join type of Left Outer, Inner, or Full Outer.
8. Optionally, click **Preview** to see the results of the join.
9. Click **Next**.
10. Specify which attributes you want in the joined data set by checking the corresponding attribute names. You can filter attributes, sort by data type, and select or deselect all.

The available attribute list contains all attributes from both the primary and secondary data sets. At the top of the list, there is also a **Source data set** attribute. If this attribute is enabled, Studio creates a new multi-assign attribute with values that specify which data sources, by name, contributed to the new records.

11. Optionally, click an attribute name to edit the name if necessary.
12. Click **Add to Script** to save the join to the script.

If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

You can monitor the progress of the join from the **Notifications** panel.

## Creating custom transformations

If the default transforms do not meet your needs, you can also create and add custom transformations to a transformation script on the **Editor** tab of the **Transform** page.

### [Creating a custom transformation](#)

You can create a custom transformation by writing custom Groovy code in the **Transformation Editor**. After you save the custom transformation, it is added as one individual execution step to the transformation script.

### [Writing custom transform code](#)

Transformations can contain attributes and records from your project data set as variables, and can create new attributes to hold the transformed values.

### [Exception handling and debugging](#)

These topics describe exception handling in **Transform** and show how to debug individual transformations.

## Creating a custom transformation

You can create a custom transformation by writing custom Groovy code in the **Transformation Editor**. After you save the custom transformation, it is added as one individual execution step to the transformation script.

The **Transformation Editor** provides access to lists of functions and attributes to insert into a transformation script. The individual Groovy functions are documented in [Transform Function Reference](#).

Also note that like other scripting languages, Groovy has a set of reserved keywords that have special meanings and therefore can't be used as variable or function names in transformation scripts. For details, see [Unsupported Groovy language features](#).

To create a custom transformation:

1. In the Catalog, select a project.
2. Select **Transform**.
3. From the transform menu, select **Editor**.
4. To insert the current refinement state into the script, click the **Use refinement state as a conditional statement** link.
5. To insert a function into the custom transform code:
  - a. Click **Functions**.
  - b. To find a specific function, use the filter field to filter by function name, or choose a category of functions from the drop down to filter by grouping such as data type.
  - c. To add a function, either double-click it, or drag and drop it into the script.
6. To insert an attribute name into the custom transform code:
  - a. Click **Attributes**.

- b. To find a specific attribute, use the filter field. You can filter the list by name or data type.
  - c. To add an attribute, either double-click it, or drag and drop it into the script.
7. Add any additional Groovy code as necessary for your data cleaning.
  8. If you are applying the transform to an existing attribute, select **Apply transformation to [attribute name]**

Applying a transformation to the selected attribute overwrites the attribute with the transformed data.

9. If you are creating a new attribute, provide the attribute name in **Create New Attribute**.

Setting the transformation to output to a new column adds a new attribute to your project data set. The new name can only contain alphanumeric characters and underscores (\_). If the name you enter contains unsupported characters, the outline of the text box turns red and you receive an error message if you try to preview or save the transformation.

10. From the **Data Type** list, select the data type to assign to the resulting attribute.

Studio automatically selects an appropriate data type, but you can override this.

11. If the new attribute should be multi-assign, deselect the **Single Assign** check box.

12. Either click **Preview** to see the previewed results of running the transformation, or click **Add to Script** to save the transformation step to the script.

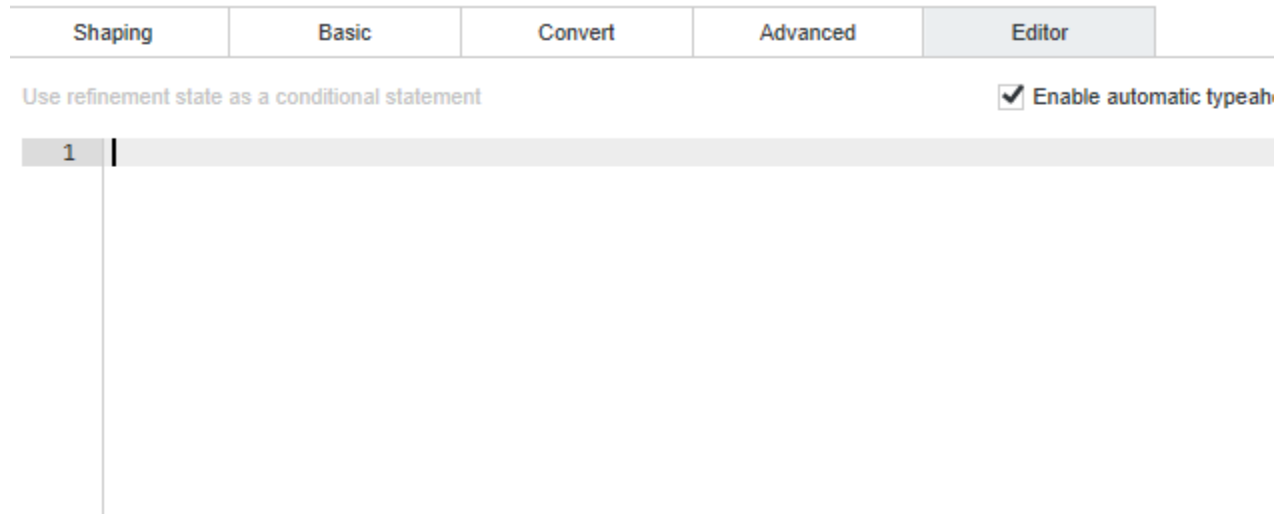
If you are done making changes to the project data set, you can commit the changes. See [Running the transformation script against a project data set](#).

## Writing custom transform code

Transformations can contain attributes and records from your project data set as variables, and can create new attributes to hold the transformed values.

### The Transformation Editor

The **Transformation Editor** becomes available when you select the **Editor** tab of the Transform page.



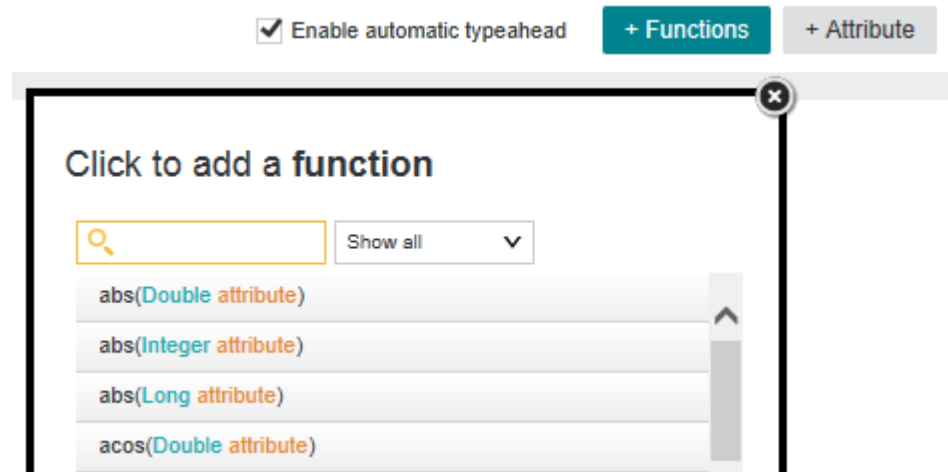
In the editor:

- **Syntax highlighting** enables color-coding of different elements in your transformation to indicate their type.
- **Auto complete** lets you view a list of autocomplete suggestions for the word you're typing, by pressing **Ctrl+space**. Use the arrow keys to navigate this list and press **Enter** to select the highlighted item.
- **Error checking** includes a built-in static parser that performs error checking when you preview or save your transformation. For more information, see [Exception handling](#).

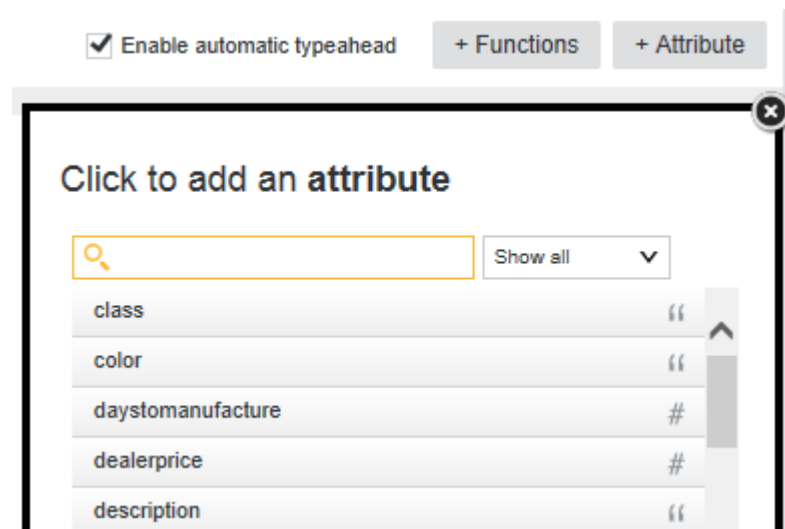
You can write code in the editor in two different ways, depending on your programming experience level:

- If you are comfortable with Groovy, you can type directly into the **Transformation Editor**. Your code can contain any of the supported Groovy language features, and custom transform functions available in Studio.
- If you have limited experience with Groovy, you can create transformations using predefined lists of custom transform functions and available attributes:
  - To view the list of transform functions, click **Functions** above the **Transformation Editor**. In the **Functions** list, you can search, filter, and learn about each function by hovering the mouse over its name.

For example:



- To view the list of your data set's attributes, click **Attributes**. The **Attributes** list displays an icon next to each attribute's name indicating its data type. For example:



To add items from either list to your transformation, click and drag its name into the **Transformation Editor**.

To add an item from the **Attributes** list as a parameter to a function, drag the attribute's name directly on top of the function's placeholder text.

### Formats for variables

You can use attributes from your project data set as variables in your transformation scripts. This allows you to pass attributes to transform functions as parameters and perform other operations on them.

To include an attribute in a transformation, you can reference it using the formats described below. The specific formats you can use for a given attribute depend on whether its name meets the following requirements:

- Names should consist of a letter or underscore ( `_` ) followed by zero or more alphanumeric characters ( `a-zA-Z, 0-9` ) and underscores.
- Names can't contain any of the reserved keywords from either the **Transform** area in Studio, or from Groovy. For more information, see [Groovy reserved keywords and unsupported functions](#).

---



---

**Note:** Unlike other variables, attributes don't need to be declared.

---



---

This table describes the formats you can use to include attributes as variables in transformations.

Format syntax	Description
<code>&lt;attribute&gt;</code>	Attribute names that consist of a letter or underscore ( <code>_</code> ) followed by zero or more alphanumeric characters ( <code>a-zA-Z, 0-9</code> ) and underscores. Names can't contain any of the reserved keywords from either the <b>Transform</b> area in Studio, or from Groovy.
<code>row["_&lt;attribute&gt;"]</code>	The map format. This can be used for all attributes, including those whose names don't meet the naming requirements described above. You can use single or double quotes.
<code>row."&lt;attribute&gt;"</code>	The long dot format. This can be used for all attributes, including those whose names don't meet the naming requirements described above. You can use single or double quotes.
<code>row.&lt;attribute&gt;</code>	The short dot format. This can only be used for attributes whose names consist of alphanumeric characters, dollar signs ( <code>\$</code> ), and underscores.

---



---

**Note:** The format you use for an attribute affects how it is handled by the static parser. For more information, see [Exception handling and troubleshooting your scripts](#).

---



---

### Functional and dot notation and function chaining

You must use proper syntax when adding transform functions to your script, or your script won't run properly. You can reference all transform functions using functional notation, as described in this topic.

```
<function>(<argument1>[,<argumentN>])
```

For example, the following code applies the `geotagAddress` function to an attribute called `address`:

```
geotagAddress(address)
```

You can use dot notation to include original Groovy functions that aren't specific to Studio:

```
<attribute>.<function>()
```

For example, the following code uses the `toString` function to convert an attribute called `quantity` to a `String`:

```
quantity.toString()
```

---



---

**Note:** You can only use dot notation for original Groovy functions. For the BDD-specific transform functions, you must use functional notation.

---



---

## Function chaining

Function chaining allows you to apply multiple functions to an attribute in a single statement. You chain functions by passing an attribute to one function, then passing that function to another function. The innermost function (the one receiving the attribute as a parameter) is evaluated first, and the outermost function is evaluated last.

For example, the following code takes an IP address, determines the city it originated from, then converts the name of the city to uppercase:

```
// Performs two transformations on a single attribute using one line of code:

toUpperCase(geotagIPAddressGetCity(IP_address))
```

The following code produces the same result as the code above, but is more verbose:

```
// The same two transformations as above, without chaining.
// 'city_name' is a temporary variable that stores the output of
geotagIPAddressGetCity()

def city_name = geotagIPAddressGetCity(IP_address)
toUpperCase(city_name)
```

As you can see in the examples, function chaining makes your code cleaner and easier to read. Additionally, not having to include placeholder variables, such as `city_name` in the second example, helps make your code less error prone.

## Extending the transform function library

You can create an external Groovy script that extends the transform function library, and then call the function in a custom transformation. This provides an extension point to the set of default Groovy-based transformation functions provided with BDD.

The custom Groovy script you create must be packaged in a JAR that you make available as a plug-in to both Studio and the Data Processing component. Next, you can run your custom transformation script in Studio with the `runExternalPlugin` function.

---



---

**Note:** Extending the transform function library is not supported by Big Data Discovery Cloud Service. The feature is supported by Big Data Discovery.

---



---

This sample Groovy script (named `MyPlugin.groovy`) is used as an example:

```
def pluginExec(Object[] args) {
    String input = args[0] //args[0] is the input field from the BDD Transformation
    Editor

    //Return a lowercase version of the input.
```

```
    input.toLowerCase()
}
```

The script defines the `pluginExec()` method, which takes an Object array (named `args`) as an argument. `args[0]` corresponds to the string attribute to be transformed and is assigned to the `input` variable. Next, the Transform function `toLowerCase()` is called to change the input data to lower case and return it to be stored in a single-assign string attribute.

You can create a more complex transformation script in Groovy, for example, one that imports external Java libraries and uses them in the script. Documentation for the Groovy language is available at: <http://www.groovy-lang.org/documentation.html>

---

**Note:** For security reasons, you should not use Groovy `System.xxx` methods, such as the `System.getProperty()` method.

---

To extend the transform function library and run the custom transformation script:

1. Create a directory for your custom transformation plug-ins:

```
mkdir /opt/custom_lib
```

This directory will also store any Java libraries that are imported by the Groovy script.

2. In the custom plug-ins directory, create a Groovy script, such as the `MyPlugin.groovy` example shown above.
3. Package the Groovy script into a JAR with the `jar cf` command:

```
jar cf MyPlugin.jar MyPlugin.groovy
```

The Groovy script must be located at the root of the JAR. Any additional files used by the script can also be included in the JAR.

4. Go to the `$BDD_HOME/dataprocessing/edp_cli/config` directory, and edit the `sparkContext.properties` file, to add the `spark.executor.extraClassPath` property pointing to the directory you created in Step 1.

Make sure you add an asterisk character at the end of the path to indicate that all JARs in the directory should be included.

The edited file should look like this example:

```
#####
# Spark additional runtime properties, see
# https://spark.apache.org/docs/1.0.0/configuration.html
# for examples
#####
spark.executor.extraClassPath=/opt/custom_lib/*
```

Note that if there already exists a `spark.executor.extraClassPath` property in the file, any libraries referenced by the property should be moved to the `custom_lib` directory so they are still included in the Spark class path.

5. If you have a clustered (multi-node) Hadoop environment, repeat Steps 1-4 on each node that runs YARN. This ensures that your custom transformation script can be used regardless of which YARN node runs the transformation workflow.



6. Run the new plug-in in the Studio's **Transform** page:
  - a. Make sure the data set is present in the **Transform** page.
  - b. Open the Transformation Editor.
  - c. From the **Functions** menu, select the **runExternalPlugin** function.
  - d. In the function's signature, specify the name of the Groovy script (in quotes) and the attribute to be processed. For example:
 

```
runExternalPlugin("MyPlugin.groovy", surveys)
```
  - e. In **Configure Output Settings**, configure either **Apply to existing attribute** or **Create New Attribute**.
  - f. Click **Add to Script**. (Do not select **Preview** as previews do not work with custom transformation scripts.)
  - g. Click **Commit to Project**.

After the transformation script is applied to the data set, the new or existing attribute should show in the new data set.

## Exception handling and debugging

These topics describe exception handling in **Transform** and show how to debug individual transformations.

### Script evaluation

Transformation scripts are evaluated top-down on each input row. This means that each transformation in the script is applied in order to the first input row, then again to the second row, and so on. This is illustrated by the following pseudo code:

```
for each input row R
  for each transform T
    R <- apply T to R
```

Additionally, each transformation can see the results of the transformations that ran before it. This is important to understand, as transformations within a script can be dependent on others. You should be aware of these dependencies when editing transformations or rearranging their order within your script.

### Dynamic typing vs. static typing

This topic is provided for reverence only as it explains the differences between dynamic and static typing. Understanding the differences between dynamic and static typing is key to understanding the way in which transformation script errors are handled, and how it is different from the way Groovy handles errors. This will also help you interpret errors created by your transformation script.

---

**Note:** It is important to know that the Groovy implementation within Big Data Discovery enforces static typing. For information on exception handling in **Transform**, which uses a static parser overriding Groovy's dynamic typing behavior, see [Exception handling and troubleshooting your scripts](#).

---

There are two main differences between dynamic typing and static typing that you should be aware of when writing transformation scripts.

First, dynamically-typed languages perform type checking at runtime, while statically typed languages perform type checking at compile time. This means that scripts written in dynamically-typed languages (like Groovy) can compile even if they contain errors that will prevent the script from running properly (if at all). If a script written in a statically-typed language (such as Java) contains errors, it will fail to compile until the errors have been fixed.

Second, statically-typed languages require you to declare the data types of your variables before you use them, while dynamically-typed languages do not. Consider the two following code examples:

```
// Java example
int num;
num = 5;
```

```
// Groovy example
num = 5
```

Both examples do the same thing: create a variable called `num` and assign it the value 5. The difference lies in the first line of the Java example, `int num;`, which defines `num`'s data type as `int`. Java is statically-typed, so it expects its variables to be declared before they can be assigned values. Groovy is dynamically-typed and determines its variables' data types based on their values, so this line is not required.

Dynamically-typed languages are more flexible and can save you time and space when writing scripts. However, this can lead to issues at runtime. For example:

```
// Groovy example
number = 5
numbr = (number + 15) / 2 // note the typo
```

The code above should create the variable `number` with a value of 5, then change its value to 10 by adding 15 to it and dividing it by 2. However, `number` is misspelled at the beginning of the second line. Because Groovy does not require you to declare your variables, it creates a new variable called `numbr` and assigns it the value `number` should have. This code will compile just fine, but may produce an error later on when the script tries to do something with `number` assuming its value is 10.

### Exception handling and troubleshooting your scripts

**Transform** uses a static parser to override some of Groovy's dynamic typing behavior and detect parsing errors, such as undefined variables, when you preview or save your transformations.

---

---

**Important:** Because the static parser forces Groovy to behave like a statically-typed language, you cannot use Groovy's dynamic typing features in your transformations. For example, while undeclared variables are normally allowed in Groovy, they produce parsing errors in **Transform**.

---

---

The static parser also verifies that the attributes referenced directly in your script match those defined in your data set's schema. Any attributes that don't match (for example, ones that are misspelled) produce an error.

---

---

**Important:** The static parser does not verify that parameters included in your transformation match their syntax as referenced in the row map of some custom functions, such as enrichment functions. If you incorrectly reference a parameter from a function, your transformation script will not validate, but the parser will not specify an error. Therefore, check the Transform API Reference (either in this document or in the Groovydoc), to verify that you correctly reference function parameters in the row map.

---

---

If you include attributes as variables in your transformation scripts, the format you use for an attribute affects how it is handled by the static parser. For information about attribute formats, see [Formats for variables](#).

If your transformation contains any parsing errors, **Transform** displays the resulting messages in the **Transformation Error** dialog box when you preview or save the transformation. Additionally, the **Transformation Editor** displays a red **X** icon next to each line that contains an error. You can hover over these icons to view more information about the error.

You should close the dialog box, fix the errors, then preview your transformation again to verify that all errors have been fixed. You cannot save your transformation to your script until it is free of errors.

### Troubleshooting exceptions for set functions

You can run the following set functions from the Transform API only on multi-assign attributes (these attributes are known as multi-value attributes in Studio):

- cardinality()
- isSet()
- isEmpty()
- isMemberOf()
- toSet()
- toSingle()

These functions belong to the in-line transformations you can do in **Transform**. These set functions are applicable to sets of values on attributes that are multi-assign.

If you run any of these functions from **Transform** in Studio, and the attribute on which you attempt to run them is a single-assign (single-value) attribute, the Transform API may throw NULL or an exception, depending on the Dgraph type of the attribute.

---

---

**Note:** You can check if an attribute is multi-value by looking at a data set in **Explore**, and selecting a table view. A column that will have more than a single value in a cell indicates that this column represents a multi-value attribute. You can also check the value of the Multi-Value column for your data set in **Project Settings > Data Views**.

---

---

To summarize, if you receive an exception when attempting to run a transformation, check if the attribute you run the transformation on is a single-value. In this case, set transformation functions do not apply.

### Security exceptions

If your transformation script contains any of the Groovy language features that are not supported, the parser throws a security exception, which displays in the **Transformation Error** dialog box. Remove the code that caused the error.

For more information on the Groovy language features that can cause security exceptions, see [Groovy reserved keywords and unsupported functions](#).

### Troubleshooting runtime exceptions

The static parser can't detect all errors, particularly runtime exceptions caused by anomalies in your data. **Transform** typically handles these errors by returning null values for data it can't process.

If you want to know more about why your transformation script is producing null values, you can wrap your code in a `try` block and set its output to a new temporary attribute of type `String` (it will show up in your project's data set table as a new column for an attribute of type `String`):

```
try {
    <transformation script> // replace this with your transformation script code
    'OK'
} catch (Exception ex) {
    ex.getMessage()
}
```

When you preview the transformation script, any error messages it produces will be output to the temporary column of type `String`. Once you have debugged it, you can delete the `try` block and remove the temporary attribute.

### Troubleshooting dateTime conversions

If you implicitly convert a `dateTime` object (that is, an `mdex:dateTime` attribute) to a `String` object, you might get a different result depending on where the cluster Spark job ran. In particular, the resulting time zone may not be what you expected.

For example, assume the following statements:

```
// mydatetime is an mdex:dateTime object
mydatetime = 1/1/1970 03:00:00 PM UTC
// convert to mdex:string-set object
new_set= toSet(mydatetime)
```

If you have a wide-spread cluster, you may not know where the job will be run. If the job is run on a machine configured for the New York, NY time zone, the result will be "Jan 1, 1970 10:00:00 AM" because EST is 5 hours ahead of UTC. However, if the job is run on a machine configured for the San Jose, CA time zone, the result will be "Jan 1, 1970 07:00:00 AM".

If your intention is to use `mydatetime` as a string, it is best to explicitly convert it. For example:

```
new_set = toSet(toString(mydatetime,"MMM d, yyyy HH:mm:ss a","UTC"))
```

In this case, the time zone is specified by the user, instead of by the Spark node.

### Transform logging

If your transformation script fails to commit, you can learn more about the cause of the failure by looking through the Data Processing logs.

Data Processing writes its logs to a user-specified directory on each Data Processing node in the cluster. The default location is the `$BDD_HOME/logs/edp` directory.

Each transformation is identified within the logs by the name of the data set it was applied to and the name of the project it originated from. You can use this information to locate the messages related to your script and determine which functions caused the failure.

## Managing and executing the transformation script

The transformation script tracks and executes the transformations.

### Reordering, editing, and deleting transformations

The transformation script is contained in a collapsible panel at the right of the **Transform** page.

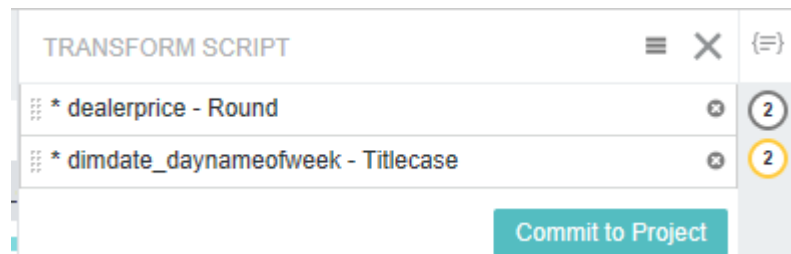
### Running the transformation script against a project data set

A project's full data set is not affected by the transformations until you run the transformation script against the project's data set. Until that time, Studio commits the transformations to the data sample that is currently associated with the project. Also, if you later change the sample size, the transformations that you have already applied are automatically applied to the sample.

## Reordering, editing, and deleting transformations

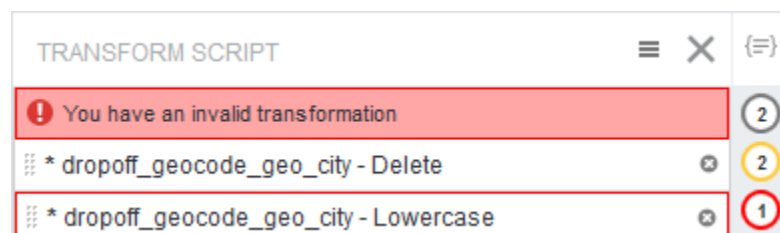
The transformation script is contained in a collapsible panel at the right of the **Transform** page.

As you create each additional transformation, it is added to the end of the transformation script. The total number of transforms in a script displays in a grey bubble. The number of uncommitted transforms in a script display in the yellow bubble. For example, this transformation script has two transforms that are uncommitted:



To display the list of transforms in the transformation script, click the lined icon above the bubbles.

Transformations that haven't been committed to a project data set are marked by \*. Invalid transformations are marked with a red outline. The **Commit to Project** button is unavailable if any such transformations exist, and a header message indicates that there are errors:



From the **Transform Script** editor, you can:

- Change the order for committing transformations to the project.  
The transformations are committed in the order they appear. To change the order, drag and drop individual transformations to a new location in the list.
- Edit individual transformations.  
To edit a transformation, click its pencil icon, make the required changes to the Groovy code, and then click **Save**. Note that some Transforms do not have an edit option, as with the first two examples in the screenshot above.
- Delete individual transformations.  
To delete a transformation, click its delete icon. **Transform** alerts you if you delete a transformation that other transformations are dependent on.

## Running the transformation script against a project data set

A project's full data set is not affected by the transformations until you run the transformation script against the project's data set. Until that time, Studio commits the transformations to the data sample that is currently associated with the project. Also, if you later change the sample size, the transformations that you have already applied are automatically applied to the sample.

The transformed data set is only available within your project. The **Commit** operation does not add a new data set to the Catalog, nor does it modify the source data in Hive.

---

---

**Note:** Due to the way BDD converts Hive source table data types to its own data types, applying your script to the project's data set may result in some omitted data types. For example, some complex Hive data types that do not match the Dgraph data types are omitted. For more information, see [Data type conversion](#).

---

---

To run the transformation script against the project data set:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Expand the **Transformation Script** panel by clicking the grey bubble that indicates the project contains transforms that have not yet been committed.
4. Click **Commit to Project**.

**Commit to Project** runs the transformations against the project data set but does not modify the underlying data set as it appears in the Catalog to other Studio users. This means that other Studio users can explore the untransformed data set in the Catalog, and project users can explore the transformed data set.

If you want to run transformations and make them available to other Studio users, commit the transformations to the project, and then create a new data set. The resulting new data set contains the transform changes and is available in the Catalog. See [Creating a new data set from the transformed data](#).

## Creating a new data set from the transformed data

After running a transformation script, you can create a new data set in the Catalog based on the transformed project data.

If you create a new data set, Studio runs the transformation script against the entire data set, not just the data sample in the project, and Studio applies the transformations, and enrichments to the new data set. A new sample is generated and the data set is available in the Catalog.

---

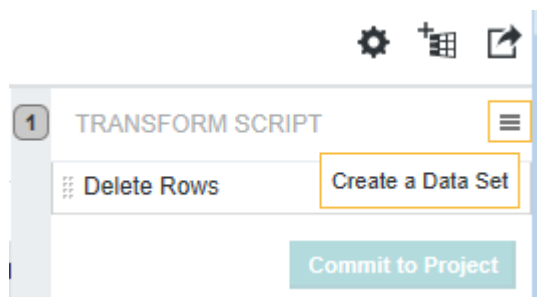
**Note:** Due to the way BDD converts Hive source table data types to its own data types, applying your script to the source data set may result in some omitted or modified data types. For example, some complex Hive data types that do not match the Dgraph data types are omitted. For more information, see [Data type conversion](#).

---

To create a new data set from the transformed data:

1. Select **Transform**.
2. From the transformation script menu, select **Create a Data Set**.

For example:



3. In the **Data Set Name** field, type the name of the new data set.
4. If desired, provide any notes about the new data set in the **Description** field.
5. Optionally, expand **Advanced Hadoop Options** and specify a Hive table name. By default, the Hive table name is the same as the data set name. If you create a data set by the same name as an existing data set, you must specify a different Hive table name. Studio maps the data set name to a unique table name.
6. Click **Create Data Set**.

(You can check the progress of this process by looking at the Oozie Web UI tool.)

After BDD finishes creating a new table in the Hive database and performing data processing, the data set becomes available in the Catalog. If you do not see the new data set in Catalog, then the script failed. You can learn more about why it failed by checking the Data Processing logs. For more information, see [Transform logging](#).

## Sharing a transform script

The ability to publish and share scripts provides a useful starting point for other projects to build a transform script.

### Publishing a transformation script

You can publish a transformation script to make it available for other Studio users to load into projects and run against a project data set.

### Loading a transformation script

If there are published transformation scripts available, you can load them into a project and run them against a similar project data set, or if the data sets are different, you can modify the script as necessary.

### Un-publishing a transformation script

If you no longer want a published transformation script to be available for use in other project data sets, you can un-publish the script. Un-publishing does not remove the script from projects in which it was previously loaded.

## Publishing a transformation script

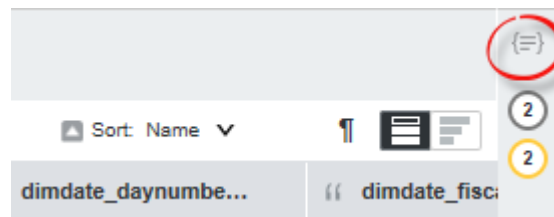
You can publish a transformation script to make it available for other Studio users to load into projects and run against a project data set.

You do not have to apply a transformation script to a project data set before publishing it. You can create a transformation script and publish it at any time.

To publish a transformation script:

1. In the Catalog, select a project.
2. Select **Transform**.
3. Click the icon to view the transformation script.

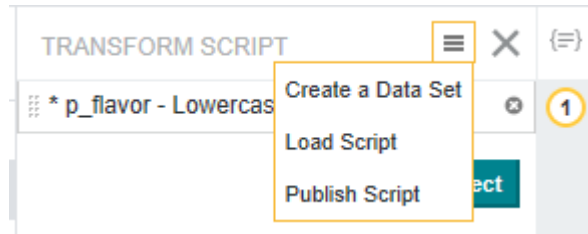
For example:



4. From the transformation script menu, select **Publish Script**.

For example:





5. On the **Publish Script** dialog, provide a unique name for the script and an optional description.
6. Click **Publish Script**.

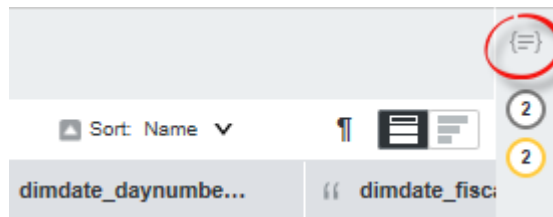
## Loading a transformation script

If there are published transformation scripts available, you can load them into a project and run them against a similar project data set, or if the data sets are different, you can modify the script as necessary.

To load a transformation script:

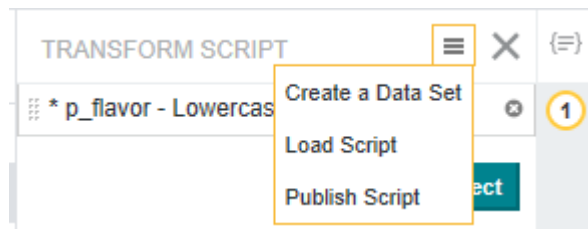
1. In the Catalog, select a project.
2. Select **Transform**.
3. Click the icon to view the transformation script.

For example:



4. From the transformation script menu, select **Load Script**.

For example:



5. On the **Load Script** dialog, select the name of the script you want to load. By default, you see all scripts that were created for the current data set.

You may need to deselect **Filter to current data set** in order to show all scripts that are available to Studio.

6. Click **Append** to append the script to the end of the existing transformation steps, or click **Replace** to overwrite the existing script.

The **Replace** option replaces the old script with a new script and also reverts any changes that were committed using the old script. The revert operation effectively resets the data set to a point before any transforms were committed. You then commit the new (replaced) script against the project data set.

## Un-publishing a transformation script

If you no longer want a published transformation script to be available for use in other project data sets, you can un-publish the script. Un-publishing does not remove the script from projects in which it was previously loaded.

To un-publish a transformation script:

1. Go to the Catalog and locate the data set associated with the published script.
2. Open the details pane for the data set.
3. Select the **Script** tab.
4. Click the x icon to remove the script from the data set.

## Tips for better performance

There are some simple things you can do while designing a transform script to improve its overall performance.

- The **Filter rows** transform should be placed as early in the transform script as possible.
- In a **Join** transform, the primary data set should be the largest data set.

---

## Transform Function Reference

This section lists data types, discusses data type conversions that take place when transformation scripts are applied, provides a list of reserved words and unsupported features of Groovy, and contains examples of custom transform function usage. It also includes the reference documentation for the custom transform functions in Studio.

Use this section together with the *Transform API Reference* (generated Groovydoc documentation for the custom functions).

### Data types

In transformation scripts, the attribute's data type is represented as a Groovy data type. This topic discusses how the Dgraph data types match the Groovy data types.

### Data type conversions

When you apply your transformation script to the project data set or click **Create a new data set** from within **Transform**, the Data Processing component converts most of the Hive data types to its corresponding Dgraph data types. However, this can result in some of the original data types being changed or omitted. This topic discusses these data type conversions in detail.

### Groovy reserved keywords and unsupported functions

This topic lists reserved keywords and those Groovy language features that are not supported in Studio.

### Conditional statements and multi-assign values

This section contains examples of transformations using conditional statements and multi-assign values.

### List of transform functions

This section lists and describes custom transform functions in Big Data Discovery.

## Data types

In transformation scripts, the attribute's data type is represented as a Groovy data type. This topic discusses how the Dgraph data types match the Groovy data types.

When you create a project based on a data set found in Catalog, this data set is indexed by Data Processing, and each attribute in the project's data set is assigned a Dgraph data type (also known as the `mdex:<type>`). All Dgraph data types begin with `mdex:`, and those used for multi-assign attributes end with `-set`. For more information on Dgraph data types, see the *Data Processing Guide*.

In transformation scripts, Groovy data types are used, as described in the following table.

When you commit your script, it outputs data with Groovy data types. These data types are then converted to the appropriate Dgraph data types when the data is written to a data set.

In addition to the settings shown in this table, the following two considerations apply:

- Multi-assign Dgraph attributes correspond to Set Groovy types. For example, a Dgraph type `mdex:int-set` (which is a type used in the Dgraph for multi-assign attributes of type Integer), corresponds to a Java type `set <integer>`.
- Long data types are converted to Integer data types if they are small enough.

Groovy data type	Corresponding mdex data types
Boolean	<code>mdex:boolean</code> , <code>mdex:boolean-set</code>
Integer	<code>mdex:int</code> , <code>mdex:int-set</code>
Long	<code>mdex:long</code> , <code>mdex:long-set</code>
Double	<code>mdex:double</code> , <code>mdex:double-set</code>
Date	<code>mdex:dateTime</code> , <code>mdex:dateTime-set</code> , <code>mdex:time</code> , <code>mdex:time-set</code>
Geocode	<code>mdex:geocode</code> , <code>mdex:geocode-set</code>
String	<code>mdex:String</code> , <code>mdex:String-set</code>

## Data type conversions

When you apply your transformation script to the project data set or click **Create a new data set** from within **Transform**), the Data Processing component converts most of the Hive data types to its corresponding Dgraph data types. However, this can result in some of the original data types being changed or omitted. This topic discusses these data type conversions in detail.

For information on complex types in Hive tables, see <https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Types>. The types that are present in your source Hive tables depend on the Hadoop environment you use.

For information on which data types are supported by Big Data Discovery, see the *Data Processing Guide*.

The following table describes how different Hive data types are affected by transformation scripts. The table lists the data types the source Hive table can contain and shows the data types in the Dgraph (`mdex:<type>`) to which they are converted.

Source Hive table data type (before the transformation script is applied)	Dgraph data type	Target Hive table data type (after the transformation script is applied)
BOOLEAN	<code>mdex:boolean</code>	BOOLEAN
TINYINT	<code>mdex:long</code>	BIGINT; this type is converted to Long during ingest.
SMALLINT	<code>mdex:long</code>	BIGINT ; this type is converted to Long during ingest.

Source Hive table data type (before the transformation script is applied)	Dgraph data type	Target Hive table data type (after the transformation script is applied)
INT	mdex:long	BIGINT; this type is converted to Long during ingest.
BIGINT	mdex:long	BIGINT
FLOAT	mdex:double	DOUBLE
DOUBLE	mdex:double	DOUBLE
DECIMAL	mdex:double	DOUBLE ; this may result in loss of precision.
DATE	mdex:dateTime	TIMESTAMP
TIMESTAMP	mdex:dateTime	TIMESTAMP
STRING	Discovered mdex:<type>	STRING (or other primitive types)
CHAR	Discovered mdex:<type>	STRING (or other primitive types)
VARCHAR	Discovered mdex:<type>	STRING (or other primitive types)
ARRAY (complex)	Multi-assign of the ARRAY type. For example, for an ARRAY of decimals, it becomes a multi-assign attribute of mdex:double.	ARRAY (complex) of the types obtained from the Dgraph type.
STRUCT (complex)	None	Multiple fields of this format: struct_(structName)_(fieldName)
BINARY	None	Unsupported; the entire field or column is omitted.
MAP (complex)	None	Unsupported; the entire field or column is omitted.
UNION (complex)	None	Unsupported; the entire field or column is omitted.

## Groovy reserved keywords and unsupported functions

This topic lists reserved keywords and those Groovy language features that are not supported in Studio.

### Reserved Keywords

Reserved keywords are words that have special meanings in Groovy language and therefore cannot be used as variable or function names in Groovy scripts. The following table lists Groovy's reserved keywords:

---

abstract	as	assert
----------	----	--------

boolean	break	byte
case	catch	char
class	const	continue
def	default	do
double	else	enum
extends	false	final
finally	float	for
goto	if	implements
import	in	instanceof
int	interface	long
native	new	null
package	private	protected
public	return	short
static	strictfp	super
switch	synchronized	this
threadsafe	throw	throws
transient	true	try
void	volatile	while

---

Additionally, the following keywords are reserved by the transform functions used in Studio:

---

DEFAULTLANG	MILLISECONDS	SECONDS
MINUTES	HOURS	DAYS
WEEKS	MONTHS	YEARS
DATEFORMAT_DEFAULT		

---

Attributes with reserved keywords for names can only be referenced through the row map format; referencing them directly will produce an error. For more information on the row map format, see [Formats for variables](#).

### Unsupported functions

For security reasons, **Transform** does not support all of Groovy's original classes. For example, `System.xxx` methods are not supported. Transformation scripts that contain methods of unsupported classes produce errors (such as Security Exception error messages) and cannot be saved to the script in Studio.

You can use any of the functions listed in the **Transformation Editor's Functions** list, as well as functions from the following classes (other original Groovy functions are not supported).

---



---

**Note:** If a function you need to use is unsupported, contact Oracle Customer Support.

---



---

This table lists the supported Groovy classes:

Math	Integer	Float
Double	Long	BigDecimal
Date	Geocode	Object
Closure	String	Set
Array	InvokerHelper	Exception
Rowbinding		

## Conditional statements and multi-assign values

This section contains examples of transformations using conditional statements and multi-assign values.

### **Example 20-1** Simple conditional statement

The following code uses an if...else statement to assign a flag to a record based on the value of the `tip_amount` attribute.

```
// if the value of tip_amount is greater than 0, assign the record the 'Tip' flag:
if(tip_amount>0){
    'Tip'
}
// if the above statement is false, assign the record the 'No Tip' flag:
else{
    'No Tip'
}
```

### **Example 20-2** Advanced conditional statement

The following code assigns each record a flag based on the `tip_amount` attribute. However, this one uses a series of if...else statements to assign different flags to each percentage range:

```
// if the tip was more than 25% of the total fare, assign the record the Large Tip
flag:
if(tip_amount/fare_amount>.25){
    'Large Tip'
}

// if the tip was less than or equal to 25% and higher than 18%, assign the record
the Standard Tip flag:
else if(tip_amount/fare_amount<=.25 || tip_amount/fare_amount>.18){
    'Standard Tip'
}

// if the tip was less than or equal to 18%, assign the record the Small Tip flag:
else if(tip_amount/fare_amount<=.18 || tip_amount/fare_amount>0){
    'Small Tip'
}
```

```
// if all of the above statements failed assign the record the No Tip flag:
else{
    'No Tip'
}
```

### **Example 20-3 Advanced conditional statement using a variable**

This example performs the same operation as the previous example, but uses a variable to store the tip percentage rather than calculating it in each statement.

```
// calculate the tip percentage and assign it to PercentageTip
def PercentageTip = tip_amount/fare_amount

// if the value of PercentTip is greater than .25 (25%),
// assign the record the Large Tip flag
if(PercentTip>.25){
    'Large Tip'
}

// if the value of PercentTip is less than or equal to .25 and higher than .18
// assign the record the Standard Tip flag
else if(PercentTip<=.25 && PercentTip>.18){
    'Standard Tip'
}

// if the value of PercentTip is less than or equal to .18 and higher than 0
// assign the record the Small Tip flag
else if(PercentTip<=.18 && PercentTip>0){
    'Small Tip'
}

// if all of the above statements failed assign the record the No Tip flag
else{
    'No Tip'
}
```

### **Example 20-4 Advanced conditional statement using date logic**

The following code uses a series of else...if statements to create a multi-assign value. It uses the `diffDatesWithPrecision` function to calculate the amount of time between the current date and the attribute `dropoff_datetime` in days. It then assigns the record a list of String values that specify the different ranges of time it falls into.

```
if(diffDates(dropoff_datetime,today(),'days',true)<=30){
    toSet('Last 30 Days','Last 90 Days','Last 180 Days')
}
else if(diffDates(dropoff_datetime,today(),'days',true)<=90){
    toSet('Last 90 Days','Last 180 Days')
}
else if(diffDates(dropoff_datetime,today(),'days',true)<=180){
    toSet('Last 180 Days')
}
else{
    toSet('Greater than 180 Days')
}
```

### **Example 20-5 Examples with multi-assign values**

The following examples demonstrate how to work with multi-assign values.



This example takes a multi-assign attribute named `ItemColor` and uses `toUpperCase` function to convert its values to upper case:

```
ItemColor.collect
{toUpperCase(it)}
```

The following code iterates through the values in a multi-assign attribute called `MedalsAwarded` and adds a specific number of points for each type of medal it finds to the variable `MedalValue`. It then returns `MedalValue`, which contains the total number of points awarded for each medal in the attribute.

```
def MedalValue=0

for (int i in 0..cardinality(MedalsAwarded)-1) {
    if(indexOf(MedalsAwarded[i], 'Gold')>=0){
        MedalValue=MedalValue+3;
    }
    else if(indexOf(MedalsAwarded[i], 'Silver')>=0){
        MedalValue=MedalValue+2;
    }
    else if(indexOf(MedalsAwarded[i], 'Bronze')>=0){
        MedalValue=MedalValue+1;
    }
}
MedalValue
```

For example, if `MedalsAwarded[ 'Gold' , 'Silver' , 'Gold' ]`, the final value of `MedalValue` would be 8.

Here is another option for iterating through the values in a multi-assign attribute:

```
def MedalValue=0

for (x in MedalsAwarded) {
    if(indexOf(x, 'Gold')>=0){
        MedalValue=MedalValue+3;
    }
    else if(indexOf(x, 'Silver')>=0){
        MedalValue=MedalValue+2;
    }
    else if(indexOf(x, 'Bronze')>=0){
        MedalValue=MedalValue+1;
    }
}
MedalValue
```

### **Example 20-6 Multi-assign value operations using method chaining**

The following code iterates through a multi-assign attribute called `PartIdentifier` and replaces its identification numbers with X to mask them.

```
PartIdentifier.{collect(trim(replace(substring(it,1,10), '[0-9]', 'X')))}
```

The above example uses method chaining to perform a number of operations on the values of `PartIdentifier` in a single statement. The first part of the statement, `PartIdentifier.collect` calls the `collect` method on the `PartIdentifier` attribute. `collect` runs the code in the outermost set of parentheses on each of the values in the `PartIdentifier` multi-assign attribute.

`collect` first calls the `substring` method, which returns the substring of the String `it`, defined by the character in position 1 through position 10, where `it` is the implicit Groovy variable ranging over the numbers of `PartIdentifier`.

This substring is passed to the `replace` method, which replaces all numeric characters ('[0-9]') with `x`. The `trim` method then takes the masked String and removes all leading and trailing whitespace from it.

## List of transform functions

This section lists and describes custom transform functions in Big Data Discovery.

### Conversion functions

Conversion functions change a value from one data type to another.

### Date functions

Date functions perform actions on Date objects, such as obtaining the month information from a specific date or adding time to a date.

### Enrichment functions

Enrichment functions are based on Data Enrichment modules used as part of data processing in Big Data Discovery. You can use these functions to extract meaningful information from your data and modify attributes to make them more useful for analysis.

### Geocode functions

Geocode functions perform different actions on Geocode objects, such as calculating the distance between two Geocode values or obtaining a Geocode's latitude coordinate.

### Math functions

Math functions perform mathematical operations on your data.

### Set functions

Set functions perform various functions on values for multi-assign attributes, such as obtaining the size of the set, checking whether a set is empty, or converting an attribute from a multi-value to a single-value attribute.

### String functions

String functions perform different actions on Strings, such as converting an entire String to uppercase or removing whitespace from a String.

### URL functions

The URL functions return parts of a specific URL, such as the host name used in the URL.

## Conversion functions

Conversion functions change a value from one data type to another.

This table describes the conversion functions that **Transform** supports. The same functions are described in the *Transform API Reference* (Groovydoc).

These functions rely on the Java `DateFormat`: <http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>.

User Function	Return Data Type	Description
<code>toBoolean(Number n)</code>	Boolean	Converts a number to a Boolean value. For example, <code>toBoolean(1)</code> evaluates to <code>true</code> . Note that only <code>0</code> evaluates to <code>false</code> . Any number other than <code>0</code> (including negative numbers) evaluates to <code>true</code> .
<code>toBoolean(String s)</code>	Boolean	Converts a String to a Boolean value. For example, <code>toBoolean("yes")</code> returns <code>false</code> . Note that this method only evaluates to <code>true</code> if the String is <code>"true"</code> . Capitalization is ignored, so <code>"true"</code> , <code>"TRUE"</code> , and <code>"tRuE"</code> are all <code>true</code> .
<code>toBoolean(Boolean b)</code>	Boolean	Converts a Boolean to a Boolean value. For example, <code>toBoolean("FALSE")</code> returns <code>false</code> .
<code>toDouble(Boolean b)</code> <code>toDouble(Number n)</code> <code>toDouble(String s)</code>	Double	Converts a Boolean, Number, or String to a Double.
<code>toLong(Boolean b)</code> <code>toLong(Date d)</code> <code>toLong(Double d)</code>	Long	Converts a Boolean, Date, or Double to a Long.
<code>toLong(String s)</code>	Long	Converts a String representation of a long to a Long. For example, <code>toBoolean("123")</code> returns <code>123</code> . Invalid Strings (for example, one that contains letters) return <code>null</code> .
<code>toString(Object arg)</code>	String	Converts an Object of any data type to a String.

## Date functions

Date functions perform actions on Date objects, such as obtaining the month information from a specific date or adding time to a date.

This table describes the Date functions that **Transform** supports. The same functions are described in the *Transform API Reference* (Groovydoc).

Date formats adhere to the **SimpleDateFormat** class in Java. For information on **SimpleDateFormat** class, see: <http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>.

User Function	Return Data Type	Description
<code>addTime(Date attribute, Integer timeToAdd, String timeUnit)</code>	Date	Adds time to a Date object. The time unit used must be one of the following constants: <ul style="list-style-type: none"> <li>• <code>MILLISECONDS</code></li> <li>• <code>SECONDS</code></li> <li>• <code>MINUTES</code></li> <li>• <code>HOURS</code></li> <li>• <code>DAYS</code></li> <li>• <code>WEEKS</code></li> <li>• <code>MONTHS</code></li> <li>• <code>YEARS</code></li> </ul>
<code>diffDates(Date firstDate, Date secondDate, String timeUnit, Boolean precisionFlag)</code>	Double	Calculates the difference between two dates as a long in a specific time unit. The time unit must be one of the following constants: <ul style="list-style-type: none"> <li>• <code>MILLISECONDS</code></li> <li>• <code>SECONDS</code></li> <li>• <code>MINUTES</code></li> <li>• <code>HOURS</code></li> <li>• <code>DAYS</code></li> <li>• <code>WEEKS</code></li> <li>• <code>MONTHS</code></li> <li>• <code>YEARS</code></li> </ul> <p>The <code>precisionFlag</code> optional parameter specifies whether to get the difference with a millisecond precision. If not used, it defaults to <code>false</code>.</p>
<code>getDayOfMonth(Date attribute, String timeZone, String locale)</code>	Integer	Returns a day-of-month value for the Date. You can specify a time zone and locale as optional parameters; the defaults are <code>null</code> and <code>"en"</code> , respectively.
<code>getDayOfWeek(Date attribute, String timeZone, String locale)</code>	Integer	Returns a day-of-week value for the Date. You can specify a time zone and locale as optional parameters; the defaults are <code>null</code> and <code>"en"</code> , respectively.
<code>getDayOfYear(Date attribute, String timeZone, String locale)</code>	Integer	Returns a day-of-year value for the Date. You can specify a time zone and locale as optional parameters; the defaults are <code>null</code> and <code>"en"</code> , respectively.
<code>getHour(Date attribute, String timeZone, String locale)</code>	Integer	Returns an hour value for the Date. You can specify a time zone and locale as optional parameters; the defaults are <code>null</code> and <code>"en"</code> , respectively.
<code>getMillisecond(Date attribute, String timeZone, String locale)</code>	Long	Returns a millisecond value for the Date, based on a time zone and locale parameters that you specify. The defaults are <code>null</code> and <code>"en"</code> , respectively.
<code>getMinute(Date attribute, String timeZone, String locale)</code>	Integer	Returns a minute value for the Date. You can specify a time zone and locale as optional parameters; the defaults are <code>null</code> and <code>"en"</code> , respectively.

User Function	Return Data Type	Description
<code>getMonth(Date attribute, String timeZone, String locale)</code>	Integer	Returns a month value for the Date. You can specify a time zone and locale as optional parameters; the defaults are null and "en", respectively.
<code>getSeconds(Date attribute, String timeZone, String locale)</code>	Long	Returns a seconds value for the Date. You can specify a time zone and locale as optional parameters; the defaults are null and "en", respectively.
<code>getYear(Date attribute, String timeZone, String locale)</code>	Integer	Returns a year value for the Date. You can specify a time zone and locale as optional parameters; the defaults are null and "en", respectively.
<code>isDate(String attribute, String dateFormat)</code>	Boolean	Determines whether a String is a valid Date value with a specific format.
<code>toDate(Long epochDate)</code>	Date	Converts a Long to a Date object.
<code>toDate(String attribute, String dateFormat, String timeZone, String locale)</code>	Date	Converts a String to a Date object using a specific date format. You can specify a time zone and locale as optional parameters; the defaults are null and "en", respectively. Invalid date Strings return null.
<code>today(String timeZone, String locale)</code>	Date	Returns the current date. You can specify a time zone and locale as optional parameters; the defaults are null and "en", respectively.
<code>toString(Date date, String dateFormat, String locale)</code>	String	Converts a Date to a String. You must specify the Date's format and a locale, which default to <code>DATEFORMAT_DEFAULT</code> and "en", respectively.
<code>truncateDate(Date date, String timeUnit)</code>	Date	Truncates a Date based on a given time unit. The time unit used must be one of the following constants: <ul style="list-style-type: none"> <li>• <code>MILLISECONDS</code></li> <li>• <code>SECONDS</code></li> <li>• <code>MINUTES</code></li> <li>• <code>HOURS</code></li> <li>• <code>DAYS</code></li> <li>• <code>WEEKS</code></li> <li>• <code>MONTHS</code></li> <li>• <code>YEARS</code></li> </ul> For example, <code>truncateDate((toDate("2015/03/31 21:34:56")), MONTHS)</code> returns <code>2015-03-01 00:00:00 UTC</code> .

### Date constants

Date constants define the default Date format and the time units that can be passed to Date functions.

This table describes the Date constants that **Transform** supports.

Constant Name	Data Type	Description
DATEFORMAT_DEFAULT	Object	Defines the default Date format: "yyyy/MM/dd HH:mm:ss"
DAYS	Object	Defines the constant for days: "days"
HOURS	Object	Defines the constant for hours: "hours"
MILLISECONDS	Object	Defines the constant for milliseconds: "milliseconds"
MINUTES	Object	Defines the constant for minutes: "minutes"
MONTHS	Object	Defines the constant for months: "months"
SECONDS	Object	Defines the constant for seconds: "seconds"
WEEKS	Object	Defines the constant for weeks: "weeks"
YEARS	Object	Defines the constant for years: "years"

#### **Example 20-7** Date extracting example

This example pulls out the year-month.

```
toString(pickup_datetime, 'yyyy-MM')
```

#### **Example 20-8** Date calculation example

The following code uses the `diffDates` function to calculate the number of days to `pickup_datetime`:

```
diffDates(today(),pickup_datetime,DAYS,true)
```

`today()` obtains the current date, `pickup_datetime` is the pickup date, `DAYS` specifies the time unit in which to return the result, and `true` specifies that the difference be calculated with precision.

#### **Example 20-9** toDate examples

This example first uses the `toLong` function to convert a `String` to a `Long` and then uses the result as the input to the `toDate` function:

```
toDate(toLong("1240596879"))
```

The following are some sample date formats for the `toDate` function:

```
toDate(date_String,'EEE MM/DD/YYYY hh:mm:ss a')
toDate(date_String,'EEE MM/DD/YYYY hh:mm:ss a','EST')
toDate(date_String,'EEE MM/DD/YYYY hh:mm:ss a','Etc/GMT-10')
toDate(date_String,'EEE MM/DD/YYYY hh:mm:ss a','Etc/Universal','en')
toDate(date_String,'EEE MM/DD/YYYY hh:mm:ss a','GMT0','de')
toDate(date_String,DATEFORMAT_DEFAULT)
toDate(date_String,DATEFORMAT_DEFAULT,'EST')
toDate(date_String,DATEFORMAT_DEFAULT,'Etc/GMT-10')
```

## Enrichment functions

Enrichment functions are based on Data Enrichment modules used as part of data processing in Big Data Discovery. You can use these functions to extract meaningful information from your data and modify attributes to make them more useful for analysis.

The same functions are described in the *Transform API Reference* (Groovydoc).

More information on the Data Enrichment modules is available in the *Data Processing Guide*.

**Transform** supports the following enrichment functions:

- [detectLanguage](#)
- [extractKeyPhrases](#)
- [extractNounGroups](#)
- [extractWhiteListTags](#)
- [geotagIPAddress](#)
- [geotagIPAddressGetGeocode](#)
- [geotagStructuredAddress](#)
- [geotagUnstructuredAddress](#)
- [geotagUnstructuredAddressGetGeocode](#)
- [getEntities](#)
- [getSentiment](#)
- [getTermSentiment](#)
- [reverseGeotag](#)
- [runExternalPlugin](#)
- [stripTagsFromHTML](#)
- [toPhoneticHash](#)

### **detectLanguage**

Finds the language of a given String attribute and returns an Oracle language code (for example, *es* for Spanish). For accurate results, the text should contain at least ten words.

The syntax is:

```
detectLanguage(String attribute)
```

where:

- `attribute` is the String attribute on which to perform language detection.

The results are returned in a single-assign attribute.

Example:

```
detectLanguage(labor_description)
```

might return "en" for the labor\_description String attribute.

### **extractKeyPhrases**

Extracts key phrases from a String attribute and returns a list of phrases in a multi-assign attribute. The function calculates key phrases using the TF/IDF algorithm, which takes the total number of times each term appears within the String and offsets that value by the number of times it appears within a larger body of work. Offsetting the value helps filter out frequently-used terms like "the" and "it". The body of work used as the control is selected internally based on the String's language; for example, the model used for English is based on a New York Times corpus. The `extractKeyPhrases` function is a wrapper function for the TF/IDF Term extractor enrichment module.

The number of key phrases returned by `extractKeyPhrases` is a function of the TF/IDF curve. By default, it stops returning terms when the score of a given term falls below ~68%.

The syntax is:

```
extractKeyPhrases(String attribute, String languageCode, Boolean smartCasing)
```

where:

- `attribute` is a String attribute that is to be processed. It is recommended that you convert the text to lowercase first, especially if it is in all caps.
- `languageCode` is an optional String parameter that specifies the language name or code (for example "en", "English", "German") to improve accuracy. Supported languages are English (UK/US), Portuguese (Brazilian), Spanish, French, German, and Italian. When specified, it forces the function to use a model specific to that language. When not specified, or when passed as `null` (this is the default), the function will automatically detect the language model. Throws an error if a non-supported language is specified.
- `smartCasing` is an optional parameter that, when set to `true`, specifies that the function automatically handle documents that are predominantly in either title case or upper case. If this parameter is not used, it defaults to `true`.

Examples:

```
extractKeyPhrases(toLowerCase(comments))
```

```
extractKeyPhrases(surveys, 'en', true)
```

When you create a new attribute as a result of using this function, make sure the attribute is of type multi-assign.

### **extractNounGroups**

Returns a String containing noun groups. A noun group is any noun, such as "movie" or "building". This is a wrapper function for the Noun Group Extractor enrichment module. This module finds and returns noun groups from a String attribute in each of the supported languages. It is used in tag cloud visualization, for finding commonly occurring themes in the data.



The syntax is:

```
extractNounGroups(String attribute, String languageCode)
```

where:

- `attribute` is the `String` attribute to be processed.
- `languageCode` is an optional parameter that specifies the language name or code (for example "en", "English", "German") to improve accuracy. Supported languages are English (UK/US), Spanish, French, German, and Italian. When specified, it forces the function to use a model specific to that language. When not specified, or when passed as `null` (this is the default), the function will automatically detect the language model. Throws an error if a non-supported language is specified.

Example:

```
extractNounGroups(labor_description, 'en')
```

would return nouns (such as "Battery" or "Mass Air Flow Sensor") for the `labor_description` `String` attribute.

When you create a new attribute as a result of using this function, make sure the attribute is of type multi-assign.

### **extractWhiteListTags**

Uses a dictionary-matching algorithm that locates elements of a finite set of `Strings` (the whitelist) within input text. The function finds all occurrences of any whitelist terms and returns a list of matching expansions. The input text is matched against a whitelist. A whitelist is newline-delimited. This is a wrapper function for the `Whitelist Tagger` enrichment module.

Each line may be either a comment (indicated with a `#` as the first character), or a matching directive comprised of either one or two values (separated by the `delimiter` character). The second value is used to rewrite the match output.

Here is a simple example whitelist:

- helium
- neon
- argon
- krypton
- xenon
- radon

It could be rewritten as follows, with a forward slash (`/`) as the delimiter:

- helium/He
- neon/Ne
- argon/Ar
- krypton/Kr
- xenon/Xe

- radon/Rn

When this whitelist is run on the text "The only noble gas is radon", it would produce an output list of ['Rn']

The syntax is:

```
extractWhiteListTags(String attribute, String whitelist, String languageCode,  
                    boolean caseSensitive, boolean matchWholeWords, String delimiter)
```

where:

- `attribute` is the `String` attribute to process.
- `whitelist` is a document containing whitelisted entries. This should be a plain text file containing a newline-delimited list of literals and configuration terms.
- `languageCode` is an optional `String` parameter that specifies the `String`'s language to improve accuracy. Set to `English` by default. Supported languages are whitespace-delimited languages only.
- `caseSensitive` is an optional `Boolean` parameter that indicates whether input is case-sensitive (the default is `false`).
- `matchWholeWords` is an optional `Boolean` parameter that indicates whether to match whole words only (when set to `false`, which is the default), or parts of words (when set to `true`). Ensures that "red" does not match "reduce".
- `delimiter` is an optional `String` parameter that specifies the delimiter character used to parse a whitelist entry into the "match" and "output" values. The TAB character (`\t`) is the default. Note that each whitelist entry can use only one delimiter (all delimiters after the first one are ignored).

Note that what the `delimiter` character does is to separate a whitelist entry into its match and output values, as in these examples:

```
delimiter=','  
Rn,86  
Ne,10  
He,2
```

```
delimiter='/'  
Rn/86  
Ne/10  
He/2
```

```
no delimiter specified (uses the default <tab> character)  
Rn<tab>86  
Ne<tab>10  
He<tab>2
```

In this `extractWhiteListTags` example, the first line defines a whitelist named `tagList`, the second line defines a document, and the third line uses the `extractWhiteListTags` Transform enrichment function, which first matches the input text against the specified whitelist. Next, finds and extracts all occurrences of any terms listed in the whitelist (in English), as `WhitelistTags`, and then returns a list of matching expansions:

```
def whitelist = '''  
helium/He  
neon/Ne
```

```

argon/Ar
krypton/Kr
xenon/Xe
radon/Rn
'''
def document = 'The noble gases make a group of chemical elements with similar
properties:
under standard conditions, they are all odorless, colorless, monatomic gases with
very low
chemical reactivity. The six noble gases that occur naturally are helium (He), neon
(Ne),
argon (Ar), krypton (Kr), xenon (Xe), and the radioactive RADON (Rn).'
```

```

extractWhitelistTags(document, whitelist, 'en', false, true, '/')
```

Note that the language specified is English (`en`), the matches are not case-sensitive (`false`), and unbounded, thus match the whole words only (`false`). The `/` delimiter is used for parsing the whitelist.

### geotagIPAddress

Converts an IP address to a Geocode String address according to the admin level. Administrative divisions vary depending on the country, so the returned values may be different than expected. This is a wrapper function for the IP Address GeoTagger data enrichment module.

The syntax is:

```
geotagIPAddress(String IPAddress, String adminLevel)
```

where:

- `IPAddress` is a valid IP address to process, in type `String`.
- `adminLevel` is an optional `String` parameter that specifies an administrative division to return. This can be set to only one of the following constant or literal values (case-sensitive):
  - `ADMIN_LEVEL_CITY` or `'City'` for a city match.
  - `ADMIN_LEVEL_COUNTRY` or `'Country'` for a country match.
  - `ADMIN_LEVEL_REGION` or `'Region'` for a region match, such as a state in the United States.
  - `ADMIN_LEVEL_REGIONID` or `'RegionID'` for the ID of the region in the GeoNames database, such as "6254926" for Massachusetts.
  - `ADMIN_LEVEL_SUBREGION` or `'SubRegion'` for a sub-region match, such as a county in the United States.
  - `ADMIN_LEVEL_SUBREGIONID` or `'SubRegionID'` for the ID of the sub-region in the GeoNames database, such as "4943909" for Middlesex County in Massachusetts.
  - `ADMIN_LEVEL_POSTCODE` or `'Postcode'` for a postal code, such as a zip code in the US.

The returned data types for the `adminLevel` are:

- String for adminLevel = City, Country, Postcode, Region, SubRegion, RegionID, SubRegionID
- Geocode for adminLevel = Geocode

Examples:

```
geotagIPAddress('148.86.25.54', 'City')
```

```
geotagIPAddress('148.86.25.54', ADMIN_LEVEL_CITY)
```

Both examples return "New York City" as a single-assign string attribute.

### **geotagIPAddressGetGeocode**

Converts an IP address to a Geocode and returns its geocode field as an Object. This is a wrapper function for the IP Address GeoTagger data enrichment module that returns a single attribute as a Geocode type.

The syntax is:

```
geotagIPAddressGetGeocode(String IPAddress)
```

where:

- `IPAddress` is a valid IP address to process, in type String.

Example:

```
geotagIPAddressGetGeocode('148.86.25.54')
```

Returns a geocode of "40.714270 -74.005970" as a single-assign Geocode attribute.

### **geotagStructuredAddress**

Geotags an address, based on structured fields.

The syntax is:

```
geotagStructuredAddress(String country, String region, String subregion, String city,  
                        String postcode, Boolean returnByPopulation, String adminLevel)
```

where:

- `country` is the field for the country of the address (use null when unknown).
- `region` is the field for the region of the address (use null when unknown). A region would be a state in the US.
- `subregion` is the field for the sub-region of the address (use null when unknown). A sub-region would be a country in the US.
- `city` is the field for the city of the address (use null when unknown).
- `postcode` is the field for the postal code of the address (use null when unknown). A postal code would be a zip code in the US.
- `returnByPopulation` is an optional Boolean parameter that, if set to true, returns the location with largest population. The default is false.

- `adminLevel` is an optional String parameter that specifies a specific field to return. This can be set to only one of the following constant or literal values (case-sensitive):
  - `ADMIN_LEVEL_CITY` or `'City'` returns the city of the address.
  - `ADMIN_LEVEL_COUNTRY` or `'Country'` returns the country of the address.
  - `ADMIN_LEVEL_REGION` or `'Region'` returns the region of the address, such as a state in the United States.
  - `ADMIN_LEVEL_REGIONID` or `'RegionID'` returns the ID of the region in the GeoNames database, such as "6254926" for Massachusetts.
  - `ADMIN_LEVEL_SUBREGION` or `'SubRegion'` returns the sub-region of the address, such as a county in the United States.
  - `ADMIN_LEVEL_SUBREGIONID` or `'SubRegionID'` returns the ID of the sub-region in the GeoNames database, such as "4952349" for Suffolk County in Massachusetts.
  - `ADMIN_LEVEL_POSTCODE` or `'Postcode'` returns the postal code of the address.
  - `ADMIN_LEVEL_GEOCODE` or `'Geocode'` returns the geocode of the least hierarchical administrative level. This is the default.

Note that the `adminLevel` parameter is the only optional parameter and therefore is the only parameter that can be omitted. All other parameters must be specified, either with a value or as null.

The function returns the value requested by the `adminLevel` parameter. The returned data types for the `adminLevel` are:

- String for `adminLevel = City, Country, Postcode, Region, SubRegion, RegionID, SubRegionID`
- Geocode for `adminLevel = Geocode`

If the address resolves to a number of locations and `returnByPopulation` is true, the function will pick the location with the largest population and then return its geocode.

Example 1:

```
// Get the geocode for San Francisco in the US and return the location with the
largest population.
geotagStructuredAddress( 'us', null, null, 'san francisco', null, true, 'Geocode')
```

Returns "39.76 -98.5" (the geocode of San Francisco, California).

Example 2:

```
// Get the region (state in the US) in which the Boston with the highest population
is located.
geotagStructuredAddress('us', '', '', 'boston', '', true, 'Region')
```

Returns "Massachusetts" (because Boston, Massachusetts has the largest population of all the Boston locations in the US).

### **geotagUnstructuredAddress**

Converts a valid address in a String attribute to a Geocode String address according to the admin level. Administrative divisions vary depending on the country, so the returned values may be different than expected.

This is a wrapper function for the Address GeoTagger data enrichment module. It adds a multi-assign attribute (column) to your data set that contains the Geocode address.

The syntax is:

```
geotagUnstructuredAddress(String addressText, String adminLevel, String addressGrain, Boolean validateAddress)
```

where:

- `addressText` is the address String to process. This must be less than or equal to 350 characters.
- `adminLevel` is an optional String parameter that specifies an administrative division to return. This can be set to only one of the following constant or literal values (case-sensitive):
  - `ADMIN_LEVEL_CITY` or 'City' for a city match.
  - `ADMIN_LEVEL_COUNTRY` or 'Country' for a country match.
  - `ADMIN_LEVEL_REGION` or 'Region' for a region match, such as a state in the United States.
  - `ADMIN_LEVEL_REGIONID` or 'RegionID' for the ID of the region in the GeoNames database, such as "6254926" for Massachusetts.
  - `ADMIN_LEVEL_SUBREGION` or 'SubRegion' for a sub-region match, such as a county in the United States.
  - `ADMIN_LEVEL_SUBREGIONID` or 'SubRegionID' for the ID of the sub-region in the GeoNames database, such as "4943909" for Middlesex County in Massachusetts.
  - `ADMIN_LEVEL_POSTCODE` or 'Postcode' for a postal code, such as a zip code in the US.
- `addressGrain` is an optional String parameter that specifies an administrative division to help the GeoTagger find the most likely match for a given level. This can be set to only one of the following constant or literal values (case-sensitive):
  - `ADMIN_LEVEL_CITY` or 'City' for a city match.
  - `ADMIN_LEVEL_COUNTRY` or 'Country' for a country match.
  - `ADMIN_LEVEL_REGION` or 'Region' for a region match.
  - `ADMIN_LEVEL_SUBREGION` or 'SubRegion' for a sub-region match.
  - `ADMIN_LEVEL_NONE` or 'None' returns the most populous location that most closely matches the address String. This is the default value.
- `validateAddress` is an optional Boolean parameter that specifies whether the GeoTagger should validate the address.

The returned data types for the `adminLevel` are:

- String for `adminLevel` = City, Country, Postcode, Region, SubRegion, RegionID, SubRegionID
- Geocode for `adminLevel` = Geocode

The following example shows how to retrieve Geocode addresses for country names in the "countries" String attribute:

```
geotagUnstructuredAddress(countries, 'Country', , true, 'en')
geotagUnstructuredAddress('New York, NY 10029', 'Region', 'SubRegion', false)
```

### **geotagUnstructuredAddressGetGeocode**

Converts an IP address to a Geocode and returns its `geocode` field as an Object. This is a wrapper function for the Address GeoTagger module.

The syntax is:

```
geotagUnstructuredAddressGetGeocode(String addressText, String addressGrain, Boolean validateAddress)
```

where:

- `addressText` is the address String to process. This must be less than or equal to 350 characters.
- `addressGrain` is an optional String parameter that helps the GeoTagger to find the most likely match for a given level. This can be set to only one of the following values (case-insensitive):
  - `ADMIN_LEVEL_CITY` or 'City' for a city match.
  - `ADMIN_LEVEL_COUNTRY` or 'Country' for a country match.
  - `ADMIN_LEVEL_REGION` or 'Region' for a region match, such as a state in the United States.
  - `ADMIN_LEVEL_SUBREGION` or 'SubRegion' for a sub-region match, such as a county in the United States.
  - `ADMIN_LEVEL_NONE` or 'None' returns the most populous location that most closely matches the address String. This is the default value.
- `validateAddress` is an optional Boolean parameter that specifies whether the GeoTagger should validate the address.

Examples:

```
geotagUnstructuredAddressGetGeocode(cities, ADMIN_LEVEL_CITY, false)
```

```
geotagUnstructuredAddressGetGeocode(cities, 'City', false)
```

### **getEntities**

Returns all entities, of a specified type, from an input String attribute. The entities are returned as a list of Strings. This function creates a new multi-assign column in your data set for the entity results. This is a wrapper function for the Named Entity Recognition extractor module. Supports only English input text.

The syntax is:

```
getEntities(String attribute, String entityType)
```

where:

- `attribute` specifies the String attribute that is to be processed.
- `entityType` is a String parameter that specifies the type of entity to extract. You can specify only one of the following constant or literal values (case-sensitive):
  - `ENTITY_TYPE_PERSON` or `'Person'` returns all the person entities found in the attribute.
  - `ENTITY_TYPE_ORGANIZATION` or `'Organization'` returns all the organization entities found in the attribute.
  - `ENTITY_TYPE_LOCATION` or `'Location'` returns all the location entities found in the attribute. Location entities are names of places, such as "Boston" or "Canada".

Examples:

```
getEntities(claims, ENTITY_TYPE_LOCATION)
```

```
getEntities(reviews, 'Person')
```

### **getSentiment**

Returns a String containing the overall sentiment of a String attribute. The attribute's sentiment can be one of the following:

- `POSITIVE`
- `NEGATIVE`

This is a wrapper function for the Sentiment Analysis (document level) data enrichment module.

The syntax is:

```
getSentiment(String textAttribute, String languageCode)
```

where:

- `attribute` specifies the String attribute that is to be processed.
- `languageCode` is an optional parameter that specifies the language name or code (for example "en", "English", "German") to improve accuracy. Supported languages are English (UK/US), Spanish, French, German, and Italian. When specified, it forces the function to use a model specific to that language. When not specified, or when passed as `null` (this is the default), the function will automatically detect the language model. Throws an error if a non-supported language is specified.

Example:

```
getSentiment(comments, 'English')
```

In the example, "comments" is a String attribute.



## getTermSentiment

Extracts phrases in sentences that have a positive or negative sentiment. The function call will specify the type of phrases to be extracted and which type of sentiment. A list of the desired terms (as Strings) is returned.

The syntax is:

```
getTermSentiment(String textAttribute, String termAttribute, String
sentimentCategory, String languageCode)
```

where:

- `textAttribute` is the String attribute to process.
- `termAttribute` is a String parameter that specifies the type of terms to extract, based on their sentiment (as set by the `sentimentCategory` argument). You can specify only one of the following values (case-sensitive) for the term type:
  - `ENTITY_TYPE_PERSON` or `'Person'` locates passages that contain person entities and returns the sentiment of those passages.
  - `ENTITY_TYPE_ORGANIZATION` or `'Organization'` locates passages that contain organization entities and returns the sentiment of those passages.
  - `ENTITY_TYPE_LOCATION` or `'Location'` locates passages that contain location entities and returns the sentiment of those passages.
  - `NOUN_GROUPS` or `'NounGroups'` extracts noun groups in sentences, based on their specified sentiment.
  - `KEY_PHRASES` or `'KeyPhrases'` extracts key phrases in sentences, based on their specified sentiment.
- `sentimentCategory` is specifies the type of sentiment to be considered for the terms. You can specify `SENTIMENT_POSITIVE` (or `'Positive'`) for negative for positive sentiment or `SENTIMENT_NEGATIVE` (or `'Negative'`) for negative sentiment. All values are case-sensitive.
- `languageCode`. An optional parameter that specifies the language name or code (for example `"en"`, `"English"`, `"German"`) to improve accuracy. Supported languages are English (UK/US), Spanish, French, German, and Italian. When specified it forces the function to use a model specific to that language. When not specified, or when passed as `null` (this is the default), the function will automatically detect the language model. Throws an error if a non-supported language is specified.

Examples:

```
getTermSentiment(comments, 'KeyPhrases', 'Positive')
getTermSentiment(comments, KEY_PHRASES, SENTIMENT_POSITIVE)
```

```
getTermSentiment(companies, 'Organization', 'Negative')
getTermSentiment(companies, ENTITY_TYPE_ORGANIZATION, SENTIMENT_NEGATIVE)
```

```
getTermSentiment(reviews, 'NounGroups', 'Positive')
getTermSentiment(reviews, NOUN_GROUPS, SENTIMENT_POSITIVE)
```

## reverseGeotag

Returns the geocode address for a specified administrative division. Searches for the administrative division within the specified radius from the entered Geocode. This is a

wrapper function for the Reverse GeoTagger data enrichment module that returns a single value.

The syntax is:

```
reverseGeotag(Geocode geoAttribute, String adminLevel, Double proximityThreshold)
```

where:

- `geoAttribute` is the Geocode to process.
- `adminLevel` is String parameter that specifies an administrative division to return. This can be set to only one of these constant or literal values (case-sensitive):
  - `ADMIN_LEVEL_CITY` or 'City' for a city match.
  - `ADMIN_LEVEL_COUNTRY` or 'Country' for a country match.
  - `ADMIN_LEVEL_REGION` or 'Region' for a region match, such as a state in the United States.
  - `ADMIN_LEVEL_REGIONID` or 'RegionID' for the ID of the region in the GeoNames database, such as "6254926" for Massachusetts.
  - `ADMIN_LEVEL_SUBREGION` or 'SubRegion' for a sub-region match, such as a county in the United States.
  - `ADMIN_LEVEL_SUBREGIONID` or 'SubRegionID' for the ID of the sub-region in the GeoNames database, such as "4943909" for Middlesex County in Massachusetts.
  - `ADMIN_LEVEL_POSTCODE` or 'Postcode' for a postal code, such as a zip code in the US.
- `proximityThreshold` is an optional Double parameter that specifies the maximum distance in miles allowed for input geocode and output geographic location. If this parameter is not specified, the default of 100 miles is used. If the distance exceeds the threshold, null is returned.

The function returns the geotagged address as requested by the `adminLevel` parameter. The returned data types for the `adminLevel` are:

- String for `adminLevel = City, Country, Postcode, Region, SubRegion, RegionID, SubRegionID`
- Geocode for `adminLevel = Geocode`

The following example uses two values to create a Geocode object, then returns the Geocode's `city` field:

```
reverseGeotag(toGeocode(42.35843, -71.05977), 'CITY', 'en', 50)
```

### **runExternalPlugin**

Runs a custom, external Groovy script as defined in an external file of `pluginName` and returns the result of the script.

The syntax is:

```
runExternalPlugin(String pluginName, String attribute, Map options)
```

where:

- `pluginName` is the name (base name and extension) of the Groovy script file (for example, `MyPlugin.groovy`).
- `attribute` is the input String passed to the script.
- `options` is an options Map, which contains any options to be used by the Groovy script. The default is to be empty.

For information on creating the external plug-in, see [Extending the transform function library](#).

### **stripTagsFromHTML**

Removes any HTML, XML and XHTML markup tags from the input String and returns the result as a String. This is a wrapper function for the Tag Stripper data enrichment module.

The syntax is:

```
stripTagsFromHTML(String attribute)
```

where:

- `attribute` is the HTML String to process.

The function returns plain text.

### **toPhoneticHash**

Produces a String hash of the input text (English only) that represents the phonetics of the text.

A word's phonetic hash is based on its pronunciation, rather than its spelling. One application for phonetic hashes is search engines. If a search term does not return any results, the search engine can compare the term's phonetic hash to the hashes of other terms and return results for the term that is the best fit. For example, "purple" and "pruple" have the same phonetic hash (PRPL), so a search for the misspelled term "pruple" would still yield results for "purple".

The syntax is:

```
toPhoneticHash(String attribute)
```

where:

- `attribute` is the String attribute to process.

Example:

```
toPhoneticHash(terms)
```

In the example, "terms" is a String attribute.

## **Geocode functions**

Geocode functions perform different actions on Geocode objects, such as calculating the distance between two Geocode values or obtaining a Geocode's latitude coordinate.

This table describes the Geocode functions that **Transform** supports. The same functions are described in the *Transform API Reference* (Groovydoc).

---

**Important:** For those geocode functions where type Double is required for inputs, make sure you enter values that are within valid ranges. The range for valid latitude values is from -90.0 to 90.0; the range for valid longitude values is from -180.0 to 180.0. Also, note that geocode functions do not accept type Long.

---

User Function	Return Data Type	Description
distance(Geocode geo1, Geocode geo2)	Double	Calculates the distance between two Geocode values, in kilometers. Note that geocode functions do not accept type Long.
getLatitude(Geocode geo)	Double	Returns the latitude coordinate of a Geocode value. Note that geocode functions do not accept type Long.
getLongitude(Geocode geo)	Double	Returns the longitude coordinate of a Geocode value. Note that geocode functions do not accept type Long.
isGeocode(String s)	Boolean	Determines whether a String is a valid Geocode value.
toGeocode(String s)	Geocode	Converts a String to a Geocode value.
toGeocode(Double lat, Double lon)	Geocode	Converts a pair of latitude and longitude coordinates to a Geocode value. For the inputs to this function, ensure that you enter valid latitude and longitude values. The range for valid latitude values is from -90.0 to 90.0. The range for valid longitude values is from -180.0 to 180.0.

## Math functions

Math functions perform mathematical operations on your data.

This table describes the math functions that **Transform** supports.

User Function	Return Data Type	Description
abs(Double d)	Double	Calculates the argument's absolute value.
abs(Integer i)	Integer	
abs(Long l)	Long	
acos(Double d)	Double	Calculates the arccosine of a double. The returned angle is between 0.0 and pi.
asin(Double d)	Double	Calculates the arcsine of a double. The returned angle is between $-\pi/2$ and $\pi/2$ .
atan(Double d)	Double	Calculates the arctangent of a double. The returned angle between $-\pi/2$ and $\pi/2$ .
atan2(Double y, Double x)	Double	
atan2(Double y, Double x)	Double	Calculates the angle theta from the conversion of rectangular coordinates (x, y) to polar coordinates (r, theta).
cbrt(Double d)	Double	Calculates the cube root of a double.

User Function	Return Data Type	Description
<code>ceil(Double d)</code>	Double	Returns the smallest (i.e., closest to negative infinity) double value that is greater than or equal to the argument, and is equal to a mathematical integer.
<code>copySign(Double a, Double b)</code>	Double	Returns the first floating-point argument with the sign of the second floating-point argument.
<code>cos(Double a)</code>	Double	Calculates the trigonometric cosine of an angle.
<code>cosh(Double d)</code>	Double	Calculates the hyperbolic cosine of a double.
<code>exp(Double d)</code>	Double	Returns Euler's number $e$ raised to the power of a double value.
<code>expm1(double x)</code>	Double	Returns $e^x - 1$ .
<code>floor(Double d)</code>	Double	Returns the largest (i.e., closest to positive infinity) double value that is less than or equal to the argument and is equal to a mathematical integer.
<code>getExponent(Double d)</code>	Integer	Returns the unbiased exponent used in the representation of a double.
<code>hypot(Double x, Double y)</code>	Double	Returns $\sqrt{x^2 + y^2}$ without intermediate overflow or underflow.
<code>log(Double d)</code>	Double	Returns the natural logarithm (base $e$ ) of a double.
<code>log10(Double d)</code>	Double	Returns the base 10 logarithm of a double.
<code>log1p(Double d)</code>	Double	Returns the natural logarithm of the sum of a double and 1.
<code>max(Double a, Double b)</code>	Double	Returns the greater of the two arguments.
<code>max(Integer a, Integer b)</code>	Integer	
<code>max(Long a, Long b)</code>	Long	
<code>min(Double a, Double b)</code>	Double	Returns the lesser of the two arguments.
<code>min(Integer a, Integer b)</code>	Integer	
<code>min(Long a, Long b)</code>	Long	
<code>nextAfter(Double a, Double b)</code>	Double	Returns the floating-point number adjacent to the first argument in the direction of the second.
<code>nextUp(Double a)</code>	Double	Returns the floating-point value adjacent to the argument in the direction of positive infinity.
<code>pow(Double a, Double b)</code>	Double	Returns the value of the first argument raised to the power of the second.
<code>rint(Double a)</code>	Double	Returns the double value that is closest in value to the argument and is equal to a mathematical integer.
<code>random()</code>	Double	Returns a positive double value that is greater than or equal to 0.0 and is less than 1.0.

User Function	Return Data Type	Description
<code>round(Double a, Integer precision)</code>	Double	Returns the closest value to the argument, with ties rounding up. The <code>precision</code> optional parameter specifies whether to round the value with the specified precision. If not used, it defaults to <code>null</code> .
<code>scalb(Double a, Integer b)</code>	Double	Returns $a \times 2^b$ rounded as if performed by a single, correctly-rounded floating-point multiply to a member of the float value set.
<code>signum(Double a)</code>	Double	Returns the signum of the argument: 0 if the argument is 0, 1.0 if the argument is greater than 0, -1.0 if the argument is less than 0.
<code>sin(Double a)</code>	Double	Calculates the trigonometric sine of an angle.
<code>sinh(Double a)</code>	Double	Calculates the hyperbolic sine of the argument.
<code>sqrt(Double a)</code>	Double	Calculates the correctly-rounded positive square root of the argument.
<code>tan(Double a)</code>	Double	Calculates the trigonometric tangent of an angle.
<code>tanh(Double a)</code>	Double	Calculates the hyperbolic tangent of a.
<code>toDegrees(Double angle)</code>	Double	Converts an angle measured in radians to an approximately equivalent angle measured in degrees.
<code>toRadians(Double angle)</code>	Double	Converts an angle measured in degrees to an approximately equivalent angle measured in radians.
<code>truncateNumber(Double number, Integer precision)</code>	Double	Truncates a number using the specified precision.
<code>ulp(Double a)</code>	Double	Returns the size of a ULP of the argument.

#### **Example 20-10** Time conversion example with floor

This example uses the `floor` function to convert `trip_time_in_secs` to minutes:

```
floor(trip_time_in_secs/60)
```

`trip_time_in_seconds` is first divided by 60 to determine the number of minutes in the trip. The `floor` function then rounds this number down and returns it as a double.

## Set functions

Set functions perform various functions on values for multi-assign attributes, such as obtaining the size of the set, checking whether a set is empty, or converting an attribute from a multi-value to a single-value attribute.

This table describes the Set functions that **Transform** supports. The same functions are described in the *Transform API Reference* (Groovydoc).

User Function	Return Data Type	Description
<code>cardinality(Object attribute)</code>	Long	Inputs the set of values in a multi-assign attribute and obtains the size of that set. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.
<code>isEmpty(Object attribute)</code>	Boolean	Inputs the set of values in a multi-assign attribute and checks whether this set is empty (has no assignments). Returns <code>true</code> if the set is empty. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.
<code>isMemberOf(Object attribute, Object value)</code>	Boolean	Checks whether the value belongs to the set of values in a multi-assign attribute. Returns <code>true</code> if the value belongs to the set. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.
<code>isNull(Object attribute)</code>	Boolean	Determines whether an attribute has any assigned values. Returns <code>true</code> if the attribute is null (has no assigned values). Works on both multi-assign and single-assign attributes.
<code>isSet(Object attribute)</code>	Boolean	Checks if an attribute is a multi-assign attribute. Returns <code>true</code> if yes. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.
<code>toSet(Object... attributes)</code>	Object[]	Converts an Object argument list to an Object array. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.
<code>toSingle(Object attribute)</code>	Object	Converts a multi-value attribute into a single-value attribute. Returns a single value chosen randomly from a set of values for the attribute. Works only on multi-value (multi-assign) attributes. Throws an exception if run on a single-assign attribute.

## String functions

String functions perform different actions on Strings, such as converting an entire String to uppercase or removing whitespace from a String.

This table describes the String functions that **Transform** supports. The same functions are described in the *Transform API Reference* (Groovydoc).

User Function	Return Data Type	Description
<code>concat(String... arguments)</code>	String	Combines a list of String arguments into a single String.
<code>concatWithToken(String joinToken, String... arguments)</code>	String	Combines a list of String arguments into a single String using a join token. For example:  <code>concatWithToken(" ", "merlot", "cabernet", "malbec")</code>  would return:  "merlot cabernet malbec"
<code>contains(String originalString, String substring)</code>	Boolean	Determines whether a String contains a substring. For example, <code>contains("Boston", "Bos")</code> would return true.
<code>find(String originalString, String pattern)</code>	String, null	Returns the first instance of a substring or regular expression within a String. Returns null if no match is found.
<code>findAll(String originalString, String pattern)</code>	String, null	Returns a (possibly empty) list of all occurrences of a regular expression (in String format) found within a String.
<code>indexOf(String originalString, String substring)</code>	Integer	Returns the index of a substring within a String.
<code>isDouble(String s)</code>	Boolean	Determines whether a String is a Double.
<code>isLong(String s)</code>	Boolean	Determines whether a String is a Long.
<code>length(String s)</code>	Integer	Returns the length of a String.
<code>replace(String attribute, String searchString, String replaceString, Boolean useRegex)</code>	String	Replaces every instance of a substring or regular expression within a String with a new text String. The <code>useRegex</code> argument specifies whether <code>searchString</code> is a literal ( <code>false</code> , the default) or a regular expression ( <code>true</code> ).
<code>splitToSet(String originalString, String delimiter)</code>	String	Splits an original String based on a specified regular expression delimiter character.
<code>stripIndent(String s)</code>	String	Removes leading spaces from a String.
<code>substring(String s, Integer start, Integer end)</code>	String	Returns a substring from the original String, based on its start point and end point. For example, <code>substring("cabernet", 0, 3)</code> returns cab.
<code>substring(String s, Integer start)</code>	String	Returns a substring from the original String, based on its start point. The returned substring will be from the start point to the end of the original String. For example, <code>substring("cabernet", 5)</code> returns net.



User Function	Return Data Type	Description
<code>toLowerCase(String s, String locale)</code>	String	Converts a String to lowercase. You can optionally specify the String's locale; this defaults to "en".
<code>toUpperCase(String s, String locale)</code>	String	Converts a String to title case. For example, <code>toUpperCase("sOME STRING")</code> would return "Some String". You can optionally specify the String's locale; this defaults to "en".
<hr/> <b>Note:</b> For lists of comma-separated values that do not include spaces, only the first item in the list will be converted to title case. For example, <code>toUpperCase("apple, cherry, plum")</code> returns "Apple, cherry, plum". <hr/>		
<code>trim(String s)</code>	String	Removes leading and trailing whitespace from a String.
<code>urlDecode(String s, String characterEncoding)</code>	String	Decodes a String into application/x-www-form-urlencoded format using a specific encoding scheme. You can optionally specify the encoding scheme; this defaults to "UTF-8".
<code>urlEncode(String s, String characterEncoding)</code>	String	Translates a String into application/x-www-form-urlencoded format using a specific encoding scheme. You can optionally specify the encoding scheme; this defaults to "UTF-8".

### Example 20-11 Escaping special characters in regular expressions

Some String functions (such as `find`, `findAll`, `replace`, and `splitToSet`) use a regular expression (regex) as a String argument. Some characters have special meaning in the regular expression. For example, the dot (.) matches any character except the new line, the dollar sign (\$) is an end-of-line anchor, and the vertical pipe (|) separates a series of alternatives.

If you use a special character in a regular expression in a Groovy function, you will have to double-escape it in the argument. For example, if you do not escape the pipe character:

```
def attrs='aa|bb|cc|dd'
splitToSet(attrs, "|")
```

returns:

```
d,b,c,a|
```

If you double-escape the pipe character:

```
def attrs='aa|bb|cc|dd'  
splitToSet(attrs, "\\|")
```

returns:

```
dd,aa,bb,cc
```

Alternatively, you can use the open and close square brackets to match a single special character, which gives you the same effect as double-escaping the character. Therefore, the last example can be written as::

```
def attrs='aa|bb|cc|dd'  
splitToSet(attrs, "[|]")
```

returns:

```
dd,aa,bb,cc
```

### **Example 20-12 Locating String patterns using *find* and regular expressions**

This example shows how to use the `find` function. Assume you have an attribute `amz_desc`, and it has the following value:

```
The future is here people and it has arrived in the form of an LED digital bracelet  
watch. That's right. You'll never  
have to live the disappointing life of not owning a digital bracelet watch. This  
revolutionary piece of technology is not  
only stylish but it will completely change the way you read time.
```

You can use the following transformation code with `find` function and regular expressions in it. This script locates a String pattern that begins with "LED" and ends with "h.", where in between, there can be zero or more of any characters (excluding a new line):

```
find(amz_desc, 'LED.*h\\.')
```

This script produces the following result:

```
LED digital bracelet watch. That's right. You'll never have to live the  
disappointing life  
of not owning a digital bracelet watch.
```

In the output, you can see that the script a part of the first sentence, and includes it because it starts with "LED". Next, the script looks for the last occurrence of "h.", which is a letter "h" followed by a period at the end of the sentence.

Note also that the script must escape the second ".", because the script wants the second "." to be treated as a regular period in the end of the sentence, and not as a regular expression for any character excluding a new line. Typically, to escape a character, "\" is used, however, in this case, "\"" must be used twice "\"\". This is because the transformation script must pass the "\"" literally ( as text) to the Groovy language, which then treats it as an escape character for the "." period.

### **Example 20-13 String replacement using *replace***

This example replaces Strings:

```
replace(cost, '\\$', '', false)
```

**Example 20-14 SubString replacement using trim and replace**

This example removes County suffix from pickup\_county attribute:

```
trim(replace(pickup_county, 'County', ''))
```

The above code uses method chaining to perform multiple actions with a single statement. `replace` first locates the substring `County` in the attribute `pickup_county` and replaces it with a blank String ( `' '`), which essentially removes it. `trim` then removes all leading and trailing whitespace from the result.

**Example 20-15 SubString replacement using regular expressions**

The following code masks the number in the medallion attribute by replacing it with 'X':

```
replace(medallion, '[0-9]', 'X', true)
```

The `replace` function locates all numeric characters in the `medallion` attribute using the regular expression `[0-9]`, which defines a range of characters. It then replaces any characters that match this pattern with the String `X`. The `true` argument specifies that a regular expression will be used instead of a literal String.

## URL functions

The URL functions return parts of a specific URL, such as the host name used in the URL.

This table describes the URL (Uniform Resource Locator) functions that **Transform** supports.

User Function	Return Data Type	Description
<code>getURLAuthority(String url)</code>	String	Returns the authority part of the URL as a single-assign string attribute. The authority part comprises an optional user-information part, a host name (e.g., domain name or IP address), and an optional port number.
<code>getURLHost(String url)</code>	String	Returns the host name in the URL as a single-assign string attribute.
<code>getURLPath(String url)</code>	String	Returns the path in the URL as a single-assign string attribute. The path begins with a single forward slash, not a double forward slash. Returns <code>null</code> if the URL does not contain a single forward slash.
<code>getURLPort(String url)</code>	Long	Returns the port number of the URL as a single-assign Long attribute. Returns <code>-1</code> if the URL does not contain a port number.
<code>getURLProtocol(String url)</code>	String	Returns the protocol name of the URL as a single-assign string attribute. Protocol examples are <code>http</code> , <code>https</code> , and <code>ftp</code> . Returns <code>null</code> if the URL does not contain a valid protocol.

User Function	Return Data Type	Description
<code>getURLQuery(String url)</code>	String	Returns the query part of the URL as a single-assign string attribute. The query part is an optional part, separated by a question mark ("?"), that contains additional identification information that is not hierarchical in nature. Returns <code>null</code> if the URL does not contain a query part.
<code>getURLRef(String url)</code>	String	Returns the reference (ref) part of the URL as a single-assign string attribute. The reference part (also called a fragment) is indicated by a sharp-sign (#) followed by more characters. Returns <code>null</code> if the URL does not contain a reference part.
<code>getURLUserInfo(String url)</code>	String	Returns the user-information part of the URL as a single-assign string attribute. The user-information part is terminated with an at-sign (@). Returns <code>null</code> if the URL does not contain a user-information part.

### **Example 20-16 URL function examples**

#### **getURLAuthority example:**

```
getURLAuthority('http://jjones:hello@www.example.com:1080/docs/resource1.html')
```

returns `jjones:hello@www.example.com:1080` as a single-assign string attribute.

#### **getURLHost example:**

```
getURLHost('http://www.example.com:1080/docs/resource1.html')
```

returns `www.example.com` as a single-assign string attribute.

#### **getURLPath example:**

```
getURLPath('http://www.example.com:1080/docs/resource1.html')
```

returns `/docs/resource1.html` as a single-assign string attribute.

#### **getURLPort example:**

```
getURLPort('http://www.example.com:1080/docs/resource1.html')
```

returns `1080` as a single-assign long attribute.

#### **getURLProtocol example:**

```
getURLProtocol('http://www.example.com:1080/docs/resource1.html')
```

returns `http` as a single-assign string attribute.

#### **getURLQuery example:**

```
getURLQuery('http://www.example.com:1080/ShowOrder.aspx?ID=10248')
```

returns `ID=10248` as a single-assign string attribute.

#### **getURLRef example:**

```
getURLRef('http://www.example.com:1080/docs/chapter1.html#introduction')
```

returns `introduction` as a single-assign string attribute.

**getURLUserInfo example:**

```
getURLUserInfo('http://jjones:hello@www.example.com:1080/docs/resource1.html')
```

returns `jjones:hello` as a single-assign string attribute.



# Part VIII

---

## Working with Components





## Component Overview

This section includes an overview of the components in Big Data Discovery, as well as the interface elements that are common across components.

### Component list

The tables below provide an overview of available components, sorted by component type.

### Common component controls

Many components share a subset of user-facing options. These are summarized here.

### Common component configuration workflows

Many components share a subset of configuration workflows. These are summarized here.

### About the component Actions menu

A component can include an **Actions** menu to allow end users to perform component-level actions such as printing the component and exporting data.

## Component list

The tables below provide an overview of available components, sorted by component type.

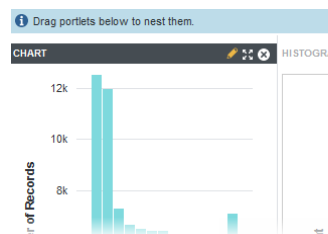
---

**Note:** The Summary column lists the full functionality for a component. Many components can be configured to expose only a subset of functionality to users.

---

### Containers

Containers allow users to group components on the page.

Component Name	Summary	Screenshot
<a href="#">Column Container</a>	Used to group components on the page. Users can configure the number of columns and overall layout.	

**Component Name**   **Summary**

[Tabbed Container](#)

Used to create a tabbed interface, with each tab containing a different set of components.

**Screenshot**



**Data Record Displays**

Data record displays allow users to view lists of records.

**Component Name**   **Summary**

[Results List](#)

Displays the list of records for the current refinement. Each record displays a subset of attributes as configured for the component. Users can export and print the list. They can also compare records, display details for a record, and refine the data.

[Results Table](#)

Displays a table containing either a list of records for the current refinement, or a set of metrics aggregated by one or more dimensions.

The table can include links to display the **Record Details** dialog, refine the data set, or display related content. Selected values can also be highlighted.

Users can export and print the list. They can also use the **Compare** dialog to compare selected rows.

**Screenshot**

**Charts**

Studio provides several types of charts to use to visualize data and the distribution of data values.

**Component Name**    **Summary**

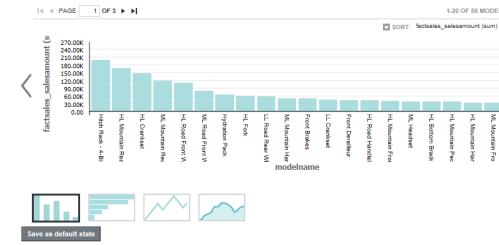
**Screenshot**

[Chart](#)

A **Chart** displays data using various chart formats, including:

- Bar charts
- Line / Area charts
- Pie / Sunburst charts
- Scatter / Bubble charts (including binned scatter charts)
- Bar-Line charts
- Thematic maps

Depending on the configured chart type, users may be able to select the data view and metrics to plot. They can also save the currently displayed chart as an image.

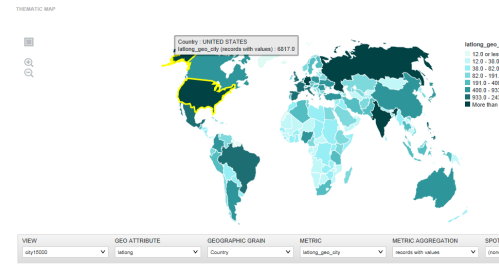


[Chart > Thematic Map](#)

Plots regions on a map and distinguishes the regions by color and record count per metric.

For a **Thematic Map**, you can select:

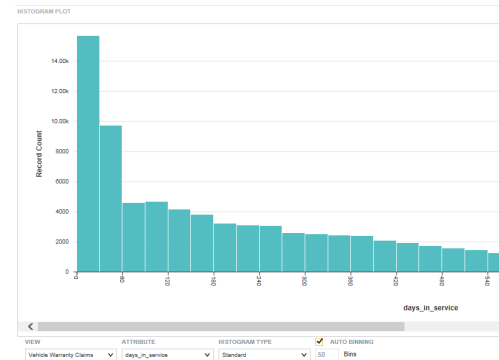
- The data view to use for the chart data
- The geocode attribute to display on the map.
- The geographic grain used to display the map. For example, you can display the map by country, region, county, and so on.



[Histogram Plot](#)

A **Histogram Plot** is a basic visualization for showing the distribution of values for a single metric. The values are grouped into bins, each representing a range of values and displaying the number of values in that range.

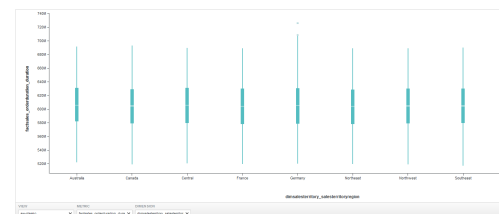
Users can select the data view, the X-axis metric, the number of X-axis bins, and the type of histogram to display



[Box Plot](#)

A **Box Plot** provides a capsule summary of values for a metric for each value of a String attribute. For example, you can show a summary of the values for Number Ordered against each value of Product Category.

Users can select the data view, as well as the values to plot on each axis.



**Component Name**      **Summary**

**Parallel Coordinates Plot**

A **Parallel Coordinates Plot** plots the values of multiple metrics for each value of a single dimension. Users can select the data view, the dimension, and the metrics and aggregation methods to use.

**Screenshot**



**Other Data Visualizations**

Other visualizations allow users to explore and visualize data on the page.

**Component Name**      **Summary**

**Custom Visualization Component**

A Custom Visualization Component is an extension to Studio that lets you create customized visualizations in cases where the default components in Studio do not meet your specific data visualization needs.

**Screenshot**

You implement a unique rendering for a Custom Visualization Component to meet your data visualization needs. For details about creating a Custom Visualization Component, see the *Extensions Guide*.

**Map**

Displays one or more map layers. Each layer represents a set of geographic locations. Users can search for locations and display details for a specific location.



**Component Name**    **Summary**

**Screenshot**

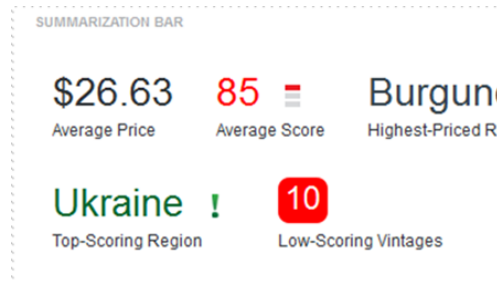
**Pivot Table**

Generates a table that allows users to perform comparisons and identify trends across several cross sections of data.  
Users can change the table layout and export the data to a spreadsheet.

Region	Designation	1992		1993	
		Price(avera...	Review Sco...	Price(avera...	Review Sco...
Alsace	Auslese			\$30.00	90.00
	Best Buy	\$9.00	86.00	\$9.00	86.00
	Botrytis	\$19.00	87.00		\$185.00
	Brut	\$20.00	85.00	\$15.00	85.00
	Cellar Select	\$43.50	85.00	\$40.00	85.00
	Cellar Select...	\$43.50	85.00	\$40.00	85.00
	Cremant	\$20.00	85.00	\$15.00	85.00
	Cuvee	\$22.45	84.85	\$27.60	86.07
	Grand Cru	\$27.60	85.88	\$42.88	86.47
	Highly Reco...			\$34.00	89.97
Alsace	Late Harvest	\$30.00	87.00		
	Reserve	\$13.61	81.80	\$16.00	84.45
	Rose	\$20.50	88.50	\$56.29	87.14
	Sec				\$40.77

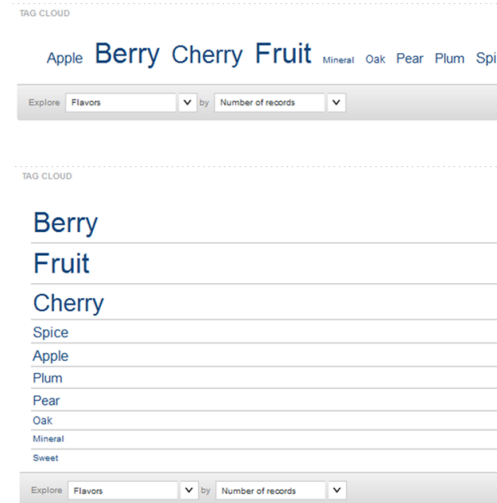
**Summarization Bar**

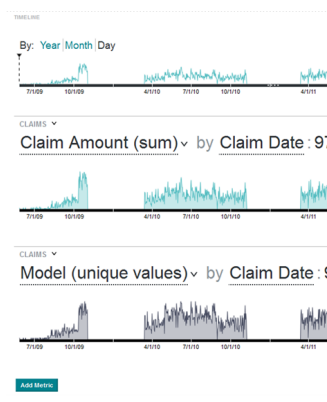
Displays a set of summary items that contain summary values from the data. A summary value can be a metric (such as average sales), a dimension value associated with the lowest or highest value of a metric (such as the region with the highest sales), or a number of flags. Flags indicate the dimension values that have associated metric values which have passed a specified threshold.



**Tag Cloud**


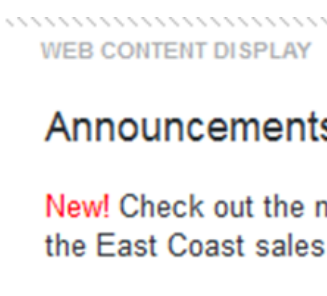
Displays a distribution of terms based on the value of an associated metric. Larger fonts indicate a higher metric value.  
The component can display in cloud view, with the terms in alphabetical order, or in list view, with the terms ordered by the metric value.  
Users can refine by the displayed terms.



Component Name	Summary	Screenshot
<p><a href="#">Timeline</a></p>	<p>Displays one or more metric values over the period of time. It consists of a Master Timeline to set the date/time range, and individual metric timelines showing the value of a metric for a selected date/time attribute.</p> <p>Users can:</p> <ul style="list-style-type: none"> <li>• Add and remove metric timelines</li> <li>• Select the metric and date/time attribute to use for each timeline</li> <li>• Select a specific subset of the available date/time range to display on the metric timelines</li> </ul> <p>There is no edit view for the component, but administrators can save a configuration to be the default displayed when users first display the page.</p>	

**Web-based Content**

Data record displays allow users to view lists of records.

Component Name	Summary	Screenshot
<p><a href="#">IFrame</a></p>	<p>Displays the content of an external URL on the page.</p>	
<p><a href="#">Web Content Display</a></p>	<p>Displays HTML content. Users can create and edit the content within the component.</p>	

## Common component controls

Many components share a subset of user-facing options. These are summarized here.

### Paging through data

Components that display data may include a pagination toolbar. From the pagination toolbar, you can navigate through the data and also determine the amount of data to display per page.

### Using a component to refine data

The **Available Refinements** panel and **Search Box** functions are specifically designed for refinement, but other components can allow you to click attribute values in order to refine the data.

### Displaying details for a component item

Some component hyperlinks display the **Record Details** dialog, which displays information from the current component item. For example, you could display details for a row in a **Results Table**, a record in a **Results List**, or a location on a **Map** component.

### Comparing items within a component

Some components include an **Actions** > **Compare** option to compare selected items.

### Printing component data

Some components include an **Actions** > **Print** option to print data.

### Exporting data from a component

Some components include an **Actions** > **Export** option to export data to a CSV file with UTF-16LE encoding.

## Paging through data

Components that display data may include a pagination toolbar. From the pagination toolbar, you can navigate through the data and also determine the amount of data to display per page.

The following components include a pagination toolbar:

- Data Record Displays
  - [Results List](#)
  - [Results Table](#)

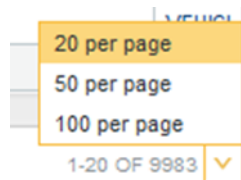
To page through component data:

1. At the left of the pagination toolbar are the navigation controls. To navigate using the paging toolbar:



- To navigate to the next or previous page, click the next or previous page button.
- To navigate to the first or last page, click the first or last page icon.

- To jump to a specific page, type the page number in the field, then press **Enter**.
2. The right of the toolbar shows the current subset of records being displayed (for example, "1-50 of 500"). To change the amount of data displayed on each page:
    - a. Click the drop-down icon next to the record subset information.
    - b. From the pop-up list, select the number of items to display per page.



## Using a component to refine data

The **Available Refinements** panel and **Search Box** functions are specifically designed for refinement, but other components can allow you to click attribute values in order to refine the data.

When you click a value that is enabled for refinement, other components that are tied to data from the same data set are updated to reflect the change.

Refinements are always added to the **Selected Refinements** panel, to allow you to:

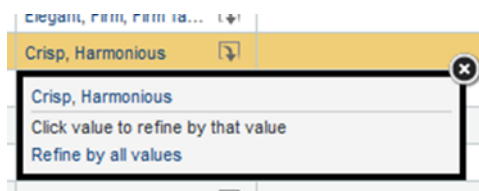
- See what you have refined by
- Remove refinements

See [About the Selected Refinements panel](#).

### Refining from a list of values

For a list of values:

- To refine by an individual value, click that value
- To refine by all of the values, click the icon that displays after the list, then click **Refine by all values**.



If you can only refine by one value at a time, then the **Refine by all values** option is not displayed.

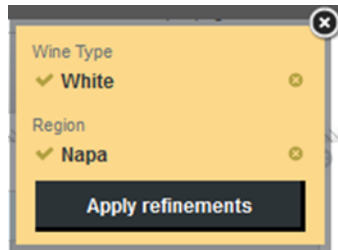
### Selecting multiple values to refine by

You can also create a list of values to refine by. You can select the values from different attributes and components, including components on different pages. For example, you could refine by a Region value displayed on a **Results Table**, and a Wine Type value displayed on a **Summarization Bar**.

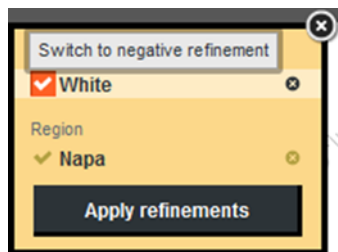
To do this, Ctrl-click each value. The values must all be enabled for refinement. When you Ctrl-click a value, it is added to the list of values to refine by. You can also Ctrl-



click a **Refine by all values** link to add all of the values to the list. As you navigate between pages, the list remains displayed until you either confirm or cancel the refinement.



To toggle a selection between positive and negative refinement, click the icon in front of the value.



If the list of refinements does not match any records, Studio displays a warning message.

To remove a value from the list, click its delete icon. If you remove all of the values, then the list is closed.

By default, the list is either at the right of the page, if you selected the first value from a component, or immediately to the right of the **Available Refinements** panel, if you selected the first value from there. You can, however, use the top bar to drag the list to another location on the page.

The list only applies to the current page. If you navigate to a different page, the list closes automatically.

To refine the data using the selected values, click **Apply refinements**.

To close the list and cancel the refinement, click the close icon.

### Date hierarchies and dimension cascading

Date/time values have an implicit hierarchy that uses the available subsets of date/time units. For example, if a component is currently displaying the year for a date value, then when you refine by a specific year, Studio could display data for the months of that year. If you then refine by a specific month, Studio could display data for the days of that month.

Components may also be configured to allow cascading of values. Cascading means that when you refine by a specific attribute value, the component is updated to display values from a different attribute. For example:

1. When you refine by the country United States, the component displays data for all of the states in the United States.
2. When you refine by California, the component displays data for cities in California.

## Displaying details for a component item

Some component hyperlinks display the **Record Details** dialog, which displays information from the current component item. For example, you could display details for a row in a **Results Table**, a record in a **Results List**, or a location on a **Map** component.

The **Record Details** dialog displays attributes within the context of their attribute groups.

If the original item reflects a record from a data set, the **Record Details** dialog displays all of the attribute groups that an administrator configured to include in record details. For example, a **Results List** item may only display a few attributes, but the **Record Details** dialog contains all of the applicable attribute groups for the associated record.

For aggregated data, the **Record Details** dialog only includes the displayed data:

- The dimensions are displayed in their attribute groups
- The metrics are displayed in the Other group

From the **Record Details** dialog, you can print the displayed data, or export the data to a spreadsheet. See [Printing component data](#) and [Exporting data from a component](#).

## Comparing items within a component

Some components include an **Actions > Compare** option to compare selected items.

The following components include a **Compare** option:

- Data Record Displays
  - [Results List](#)
  - [Results Table](#)

To compare items:

1. Select the records to compare:
  - a. Select the check box for each item you want to compare.
  - b. Select **Actions > Compare**.

The **Compare** dialog displays with the selected items:

The screenshot shows a 'Compare' dialog window with the title 'Compare'. At the top, it says 'COMPARING 4 RECORD(S)' and '0 RECORD(S) CURRENTLY SELECTED'. There is a search box labeled 'Filter attributes...'. Below this, there are four columns labeled 'Record 1', 'Record 2', 'Record 3', and 'Record 4'. The rows are grouped into several categories: Identification (6 attributes), Characteristics (4 attributes), Price and Score (2 attributes), and Other (2 attributes). Each row contains data for the four records.

	Record 1	Record 2	Record 3	Record 4
<b>Identification (6 attributes)</b>				
Name	Albarino Rias Baixas	Alentejo Convento da Vila	Alentejo Tinto da Talha	Aloxe-Corton
Description	Dry, tart and crisp, offering s...	Fresh, jammy and peppery in...	Soft, generous and fruity, wi...	Simple and appealing, with a...
Wine Type	White	Red	Alentejo, Red	Burgundy Cote de Beaune, ...
Vintage	1992	1992	1992	1992
Winery	Adegas Morgadio	Adega Cooperativa de Borba	Roquevale	Bouchard Pere & Fils
Region	Spain	Portugal	Portugal	Burgundy
<b>Characteristics (4 attributes)</b>				
Drinkability	Drink now			
Flavors	Apple, Dry, Fruit, Grape, Gra...	Jam, Pepper	Berry, Blackberry, Blueberry,...	Black Cherry, Cherry
Body	Crisp	Fresh	Soft	Bright, Simple
Wine Type	White	Red	Alentejo, Red	Burgundy Cote de Beaune, ...
<b>Price and Score (2 attributes)</b>				
Date Reviewed	6/15/1993	4/15/1994	4/15/1994	12/15/1994
Review Score	25	78	79	82
<b>Other (2 attributes)</b>				
Price	\$18.00	\$4.00	\$7.00	\$30.00
Wine ID				

When comparing records, the **Compare** dialog displays all attributes that display in the **Record Details** view. For example, a **Results Table** row may only display a subset of attributes in the component itself, but the **Compare** dialog shows the full set of attributes for the selected records.

If the original items have been aggregated, the **Compare** dialog only includes the displayed data. The dimensions used for aggregation display in their attribute groups, and the metrics display in the **Other** group.

2. Optionally, select a baseline item:

- a. To select an item as the baseline, click the lock icon in the header row.

The selected item becomes the first column in the list and cannot be moved. The lock icon is changed to indicate that the item is the baseline. If another item was previously selected as the baseline, it becomes a non-baseline item:

The screenshot shows the 'Compare' dialog window with the title 'Compare'. At the top, it says 'COMPARING 3 RECORD(S) TO BASELINE RECORD 3' and '0 RECORD(S) CURRENTLY SELECTED'. There is a search box labeled 'Filter attributes...'. Below this, there are four columns labeled 'Record 3', 'Record 1', 'Record 2', and 'Record 4'. The rows are grouped into several categories: Identification (6 attributes), Characteristics (4 attributes), Price and Score (2 attributes), and Other (2 attributes). Each row contains data for the four records.

	Record 3	Record 1	Record 2	Record 4
<b>Identification (6 attributes)</b>				
Name	Alentejo Tinto da Talha	Albarino Rias Baixas	Alentejo Convento da Vila	Aloxe-Corton
Description	Soft, generous and fruity, wi...	Dry, tart and crisp, offering s...	Fresh, jammy and peppery in...	Simple and appealing, with a...
Wine Type	Alentejo, Red	White	Red	Burgundy Cote de Beaune, ...
<b>Characteristics (4 attributes)</b>				
Drinkability	Drink now			
Flavors	Apple, Dry, Fruit, Grape, Gra...	Jam, Pepper	Berry, Blackberry, Blueberry,...	Black Cherry, Cherry
Body	Crisp	Fresh	Soft	Bright, Simple
Wine Type	White	Red	Alentejo, Red	Burgundy Cote de Beaune, ...
<b>Price and Score (2 attributes)</b>				
Date Reviewed	6/15/1993	4/15/1994	4/15/1994	12/15/1994
Review Score	25	78	79	82
<b>Other (2 attributes)</b>				
Price	\$18.00	\$4.00	\$7.00	\$30.00
Wine ID				

You can drag any non-baseline records left or right in the display, to allow you to do a side-by-side comparison of selected records.

- b. To remove the designation as a baseline record, click the lock icon again.

The record column remains at the left of the table, but can now be moved.

3. Reorder or search for relevant attributes to compare:

- a. Drag and drop attribute groups to move them up or down in the display.

You can reorder attributes within a group, but cannot drag attributes to other groups.

- b. To expand or collapse an attribute group, click the group name.
  - c. To expand all attribute groups, select **Actions > Expand all groups**.
  - d. To collapse all attribute groups, select **Actions > Collapse all groups**.
  - e. To search for a specific attribute, enter text in the **Filter** field.
4. Highlight the differences between selected items:
    - a. select **Actions > Highlight Differences**.

If you haven't selected a baseline, then the **Compare** dialog highlights attribute values that are different across any of the selected items.

* Identification (6 attributes)				
Name	Albarino Rias Baixas	Alentejo Convento da Vila	Alentejo Tinto da Talha	Aloxe-Corton
Description	Dry, tart and crisp, offering s...	Fresh, jammy and peppery in...	Soft, generous and fruity, wi...	Simple and appealing, with a...
Wine Type	White	Red	Alentejo, Red	Burgundy Cote de Beaune, ...
Vintage	1992	1992	1992	1992
Winery	Adegas Morgadio	Adega Cooperativa de Borba	Roquevale	Bouchard Pere & Fils
Region	Spain	Portugal	Portugal	Burgundy

If there is a baseline, the **Compare** dialog highlights attribute values in non-baseline items that differ from the baseline values:

* Identification (6 attributes)				
Name	Alentejo Tinto da Talha	Albarino Rias Baixas	Alentejo Convento da Vila	Aloxe-Corton
Description	Soft, generous and fruity, wi...	Dry, tart and crisp, offering s...	Fresh, jammy and peppery in...	Simple and appealing, with a...
Wine Type	Alentejo, Red	White	Red	Burgundy Cote de Beaune, ...
Vintage	1992	1992	1992	1992
Winery	Roquevale	Adegas Morgadio	Adega Cooperativa de Borba	Bouchard Pere & Fils
Region	Portugal	Spain	Portugal	Burgundy

- b. To remove the highlighting, select **Actions > Hide Highlights**.
5. To remove records:
    - a. Select the check box for each record to remove.
    - b. Select **Actions > Remove Selected**.
  6. To reset the **Compare** dialog to its initial display, select **Actions > Reset view**.

## Printing component data

Some components include an **Actions > Print** option to print data.

The following components include a **Print** option:

- Data Record Displays
  - [Results List](#) - only prints the current page
  - [Results Table](#)
  - Record Details - While not a component, the Record Details screen is accessible from the Results List, Results Table, and Map.
- Other Data Visualizations

- [Pivot Table](#) - only prints the currently displayed data

To print component data:

1. In the component **Actions** menu, select **Print**.

A print preview displays. Any refinements that have been applied to the data are included at the top of the printout.

2. Configure any printer settings as desired and click **Print**.

## Exporting data from a component

Some components include an **Actions** > **Export** option to export data to a CSV file with UTF-16LE encoding.

The export option is not available if you are viewing the project on an iPad.

The following components include an **Export** option:

- Data Record Displays
  - [Results List](#) - Exported data includes the displayed attributes for all exported records. You can select specific records to export.
  - [Results Table](#) - Exported data includes all columns from the available column sets. You can select specific rows to export.  
For aggregated tables, only the displayed dimensions are included in the export.
  - Record Details - While not a component, the Record Details screen is accessible from the Results List, Results Table, and Map. Exported data from this screen includes a single row for the displayed data.
- Other Data Visualizations
  - [Pivot Table](#) - Exported data includes the data used to create the Pivot Table display.

To export data from a component:

1. In the component **Actions** menu, select **Export**.

The **Export Data** dialog displays.

2. In the **File name:** field, set output file format:

By default, the output file name is the data view name, with spaces replaced by underscores.

3. In the **File type:** field, set the output type:

- **Avro** saves the file as an Avro file in HDFS.
- **Delimited** saves the file as a CSV.

- a. For **Delimited** files, set the following using drop-downs:

- **Field Delimiter** - The character used to separate fields in a record. You can select **Other** to enter a custom character.

- **Quote Character** - The character used for quotation marks, either double or single quote.
- **Multi-Value Delimiter** - The character used to separate multiple values in the same multi-assign field. You can select **Other** to enter a custom character.

For example, if you choose **Pipe (|)**, **Single quote (')**, and **Semicolon (;)**, a row from the exported file might look something like:

```
|'10234'|'Bourdeaux'|'Valmaison'|'Best buy';'Recommended'|'$10'|
```

4. In the **Export to:** drop-down, select an export destination:

- **My Computer** is the default option, and prompts to open or download the file, based on your browser.
- **HDFS** exports the file to HDFS so that it is available to other Studio users. This option is available to Administrators and Power Users.

a. For **HDFS** exports, set the following:

- **HDFS File Path** - The file path on the configured HDFS export directory. Ensure that any changes return a checkmark when validating path uniqueness.
- **Create Hive table** - For Avro files, select this checkbox to create a new Hive table for exported data.

b. If you are creating a Hive table, set the name and description:

- **Hive table name:** - The name of the Hive table.

By default, this is the data view name with spaces replaced by underscores and an incremented number appended for uniqueness. Ensure that any changes return a checkmark when validating that the new path is unique.

- **Description:** - A description of the table.

5. If the component has multi-value attributes selected, then under **Multi-Value Attribute Handling**, click a radio button to indicate whether to leave multi-value attributes without any transformations, or to transpose columns for multi-value attributes.

Transposing multi-value attributes adds a column for each unique value of the attribute. For each record, each column includes a Boolean value that indicates whether the attribute value is present for that record.

6. Click **Export**.

An Export Data screen displays while data is formatted and prepared for export.

7. Save or open the exported file.

The exact prompt varies based on your browser.

## Common component configuration workflows

Many components share a subset of configuration workflows. These are summarized here.

### About configuring components

You can customize components to determine details for the displayed data and formatting. Many components include a default configuration, while others (such as charts) require configuration before they display any data.

### Renaming a component

After you add a component to your project, you can rename it to reflect your own terminology or the content being displayed.

### Configuring attribute value formats

For values displayed on a component, you can configure the display format.

### Selecting the aggregation method to use for a metric

For metrics other than system metrics or predefined metrics, when you first select a metric to add to a component, Studio uses the default aggregation method for that attribute. You can then select a different aggregation method.

### Configuring cascading for dimension refinement

When users refine by dimension values on a component, you can configure the dimension to use cascading.

## About configuring components

You can customize components to determine details for the displayed data and formatting. Many components include a default configuration, while others (such as charts) require configuration before they display any data.

The steps below summarize the workflow for configuring data display in a component.

Configuring data display in a component consists of the following high-level steps:

1. Select the data view from the available data set views for the project.  
Some components use multiple data views. The **Summarization Bar**, for example, can use a separate data view for each summary item.
2. Select the attribute(s) or aggregated value(s) to display.
3. For attributes that are not System Metrics or Predefined Metrics, select the aggregation methods to use.
4. Configure the display format for each value displayed in the component.
5. Configure any attribute actions for displayed values.
6. Configure the component **Action** menu.

For detailed instructions for specific components, see the corresponding component documentation.

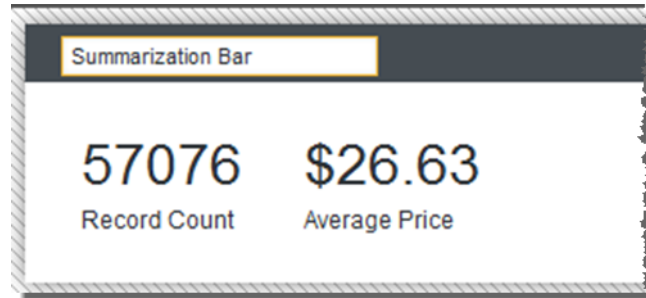
## Renaming a component

After you add a component to your project, you can rename it to reflect your own terminology or the content being displayed.

To rename a component:

1. Click the title of the component.

The title field becomes editable.



2. In the field, type the new title for the component.

3. Press **Enter**.

The new title replaces the previous title.

You can also use the **Localization** page to edit and localize the component title. See [Localizing the project name, description, page names, and component titles](#).

## Configuring attribute value formats

For values displayed on a component, you can configure the display format.

Components can display:

- Attribute values from a data view
- Predefined metrics from a data view
- Component-specific metrics that are derived from the attribute values in a data view

For attribute values and predefined metrics, the default display format is configured in the **Data Views** page. See [Configuring the default display format for an attribute](#). You can override these defaults on a per-component basis.

For component-specific metrics, the default display depends on the type of aggregation:

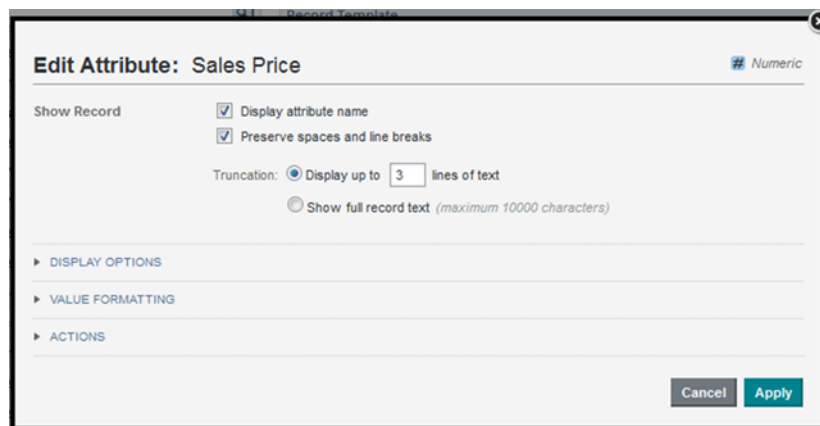
Aggregation Method	Default Display Format
Sum	Uses the default display format for the original attribute.
Average	
Median	
Min	
Max	
Set	
Arb	



Aggregation Method	Default Display Format
Variance Standard Deviation	Uses the default format (adjusted for locale) for a decimal number.
Count (Number of records with values) Count Distinct (Number of unique values)	Uses the default format (adjusted for locale) for an integer number.

To configure the attribute value format for an attribute in a component:

1. Open the **Edit Attribute** dialog for the attribute you wish to configure:



2. To hide the attribute name in the Results List, deselect the **Display attribute name** check box.

3. To configure String values:

- a. To ignore line breaks and white space in the attribute value, deselect the **Preserve spaces and line breaks** check box.
- b. Configure the display limit for the attribute:
  - To limit attribute value display to a specific number of lines of text, select **Truncation > Display up to** and enter the number of lines to display.
  - To display the full text of the attribute value up to the maximum allowable length, select **Truncation > Show full record text**.
- c. Expand the **Display Options** section and configure the **Text Format** settings.

These include:

- Text size
- Text weight (Bold or Normal)
- Text alignment within the template block

At the top of the display settings is a sample value showing the current configuration adjusted for your locale.

4. To configure numeric values:

- a. Expand the **Value Formatting** section and configure the **Basic Formatting** settings.

These include:

- **Metric Type** - Used to display numeric values as currency or percentage values. When selecting Percentage, the value is multiplied by 100. Note that any attribute that is already formatted as a currency or percentage by default cannot be reconfigured.
- **Decimal Places** - Used to limit the number of decimal places that display for the value.
- **Grouping Separator** - Used to display or hide the grouping separator for thousands values.

- b. Configure **Advanced Formatting** settings.

These include:

- **Number Format** - Use a negative sign or parentheses to indicate negative values.
- **Decimal Separator** - Use a period or comma as a decimal point character.
- **Grouping Separator** - Use a period or comma as a grouping separator character.

5. To configure Boolean values:

- a. Expand the **Value Formatting** section and configure the **Basic Formatting** settings.

These include:

- **Default** - Display the default configuration
- **Custom Values** - Specify values to display in place of default "true" (1) or "false" (0) values.

6. To configure geocode values:

- a. Expand the **Value Formatting** section and configure the **Basic Formatting** settings.

These include:

- **Default** - Display the default configuration.
- **Fixed length** - Specify the number of decimal places to display for the latitude and longitude values.

7. To configure date/time values:

- a. Expand the **Value Formatting** section and configure the **Basic Formatting** settings.

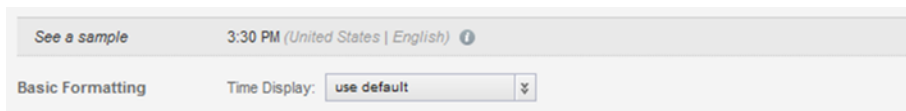
These include:

- **Date Display** - Select the format to use for the date. Available formats are determined by the locale. For some locales, the month displays first, and for others the day displays first.
- **Time Display** - Select the format to use for the time.

8. To configure time values:

- a. Expand the **Value Formatting** section and configure the **Basic Formatting** settings.

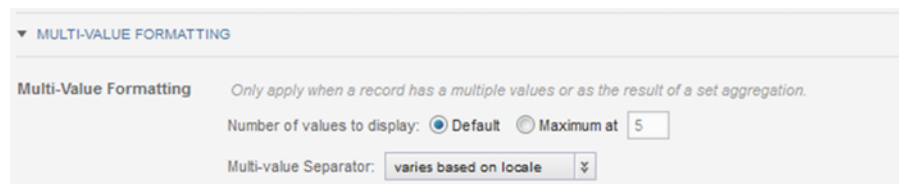
These include:



- **Time Display** - Select the format to use for the time.

- To configure attributes that are multi-value or use the Set aggregation:
  - Expand the **Multi-Value Formatting** section and configure the **Multi-Value Formatting** settings.

These include:



- **Number of values to display** - Set the number of values to display.
- **Multi-value Separator** - Select the character to use to separate the values.

- Click **Apply**.

## Selecting the aggregation method to use for a metric

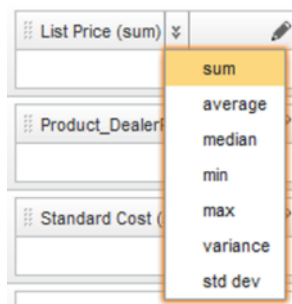
For metrics other than system metrics or predefined metrics, when you first select a metric to add to a component, Studio uses the default aggregation method for that attribute. You can then select a different aggregation method.

For information on aggregation methods and how they work, see [Aggregation methods and the data types that can use them](#).

You configure the default and available aggregation methods for each attribute on the **Data Views** page. For information on configuring the available and default aggregation methods, see [Selecting the available and default aggregation methods for an attribute](#).

To change the aggregation method used for a metric value:

- On the edit view of the component, the selected metric includes a drop-down icon to allow you to select a different aggregation method.



- The configuration dialog for the metric also contains an aggregation method list you can use to change the aggregation method.



## Configuring cascading for dimension refinement

When users refine by dimension values on a component, you can configure the dimension to use cascading.

Cascading means that when the data is refined to a single value for the dimension value, the component is updated to use a different dimension.

When there are no more dimensions to cascade to, the component remains on the last dimension in the cascade.

For example, a component includes a Country dimension. The Country dimension is configured to cascade to State and then to Supplier. With this configuration:

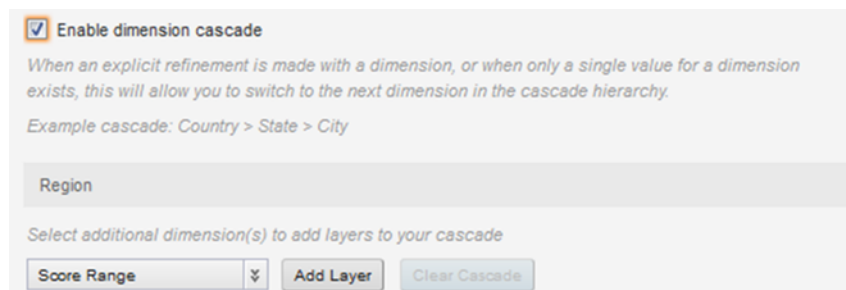
1. When users refine the data to only show records for the United States, the component uses the State dimension (for states within the United States).
2. If users then refine the data to only show records for California, the component uses the Supplier dimension (for suppliers within California).
3. If users then refine the data by a specific supplier, the component displays the data for the selected supplier, and the cascade stops.

Note that for multi-or or multi-and dimensions, where users can refine by more than one value, then if the data is refined by any one of the dimension values, the cascade continues to the next dimension, even if there are still available values to refine by.

When configuring a dimension, to configure cascading for refinement:

1. In the **Dimension Cascade** section, select the **Enable dimension cascade** check box.

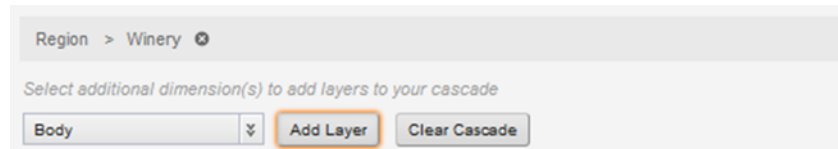
The cascade configuration displays. The current dimension is automatically at the top of the cascade.



You can select the next dimension in the cascade from the list.

2. To add a new layer to the cascade, select the dimension for that layer from the list, then click **Add Layer**.

The new layer is added to the end of the cascade.



3. To remove a layer from the cascade, click the delete icon next to that layer.
4. To clear the entire cascade, click **Clear Cascade**.

## About the component Actions menu

A component can include an **Actions** menu to allow end users to perform component-level actions such as printing the component and exporting data.

Some components also support options that require users to first select one or more items, such as comparing records, refining by attribute values in selected records, or using POST to pass parameters to a URL.

The available actions for a component **Actions** menu are:

Action	Description
<b>Print</b>	Allows users to print the component.
<b>Export</b>	Allows users to export data from a component.
<b>Compare</b>	<p>Displays the <b>Compare</b> dialog to allow users to do a more detailed comparison of selected items.</p> <p>The <b>Actions</b> menu can only contain one <b>Compare</b> option, which is enabled by default.</p> <p>If the view used for the component does not have any identifying attributes, then the <b>Compare</b> check box is disabled and locked.</p>
<b>Pass Parameters</b>	<p>Allows users to create a hyperlink that generates an HTTP POST request containing attribute values from the selected records as POST parameters.</p> <p>The <b>Pass Parameters</b> option is only available for the <b>Results List</b> and <b>Results Table</b> components.</p> <p>The <b>Actions</b> menu can contain multiple <b>Pass Parameters</b> options, each with a different label, URL, and parameters.</p> <p>By default, there are no <b>Pass Parameters</b> options in the menu.</p>
<b>Refinement</b>	<p>Allows users to refine the data using selected attribute values from the selected records. The attributes must allow users to refine using multiple values (have multi-or or multi-and refinement behavior).</p> <p>For example, users could select three records and then refine the data to only include records that have values for available colors and available sizes that are used in those selected records.</p> <p>The <b>Refinement</b> option is only available for the <b>Results List</b> and <b>Results Table</b> components.</p> <p>The actions menu can contain multiple <b>Refinement</b> options, each with a different label and attributes.</p> <p>By default, there are no <b>Refinement</b> options in the menu.</p>

### Selecting the actions to include in the Actions menu

On the **Actions** tab of the component edit view, you use the **Actions menu** list to select the available options in the **Actions** menu. For some actions, you can only enable or disable them. For other actions, you can add one or more instances to the menu.

### Configuring actions for attribute values

For many components, you can allow users to click displayed values in order to refine data, display record details, or navigate to a specified page or external URL.

### About component hyperlinks

Some components can include hyperlinks from the displayed records to external URLs.

### About using non-HTTP protocol links

While Studio does not prevent you from inserting links that use a non-HTTP protocol, such as `file://` or `ftp://`, using these types of links is not recommended.

### Selecting the target page for a refinement or hyperlink

You can configure refinement actions to display a different page. You can also create hyperlink actions that navigate to a different page.

### Recommendations for attribute values that are complete URLs

It is possible that your source data contains complete URLs (including the protocol, host, port, and path) that are ingested when the data is loaded. For example, company data could include links to pages on the company web site.

### Indicating whether to encode inserted attribute values

When you insert an attribute value as a URL parameter, you can specify whether Studio encodes the value before inserting it into the URL.

### Configuring hyperlinks to external URLs

Components can include hyperlinks to external URLs, which can present security risks.

### Configuring a Pass Parameters Actions menu action

For a **Pass Parameters** option in a component **Actions** menu, you can configure the URL and whether to display it in a new browser window. You can also include attribute values as POST parameters.

## Selecting the actions to include in the Actions menu

On the **Actions** tab of the component edit view, you use the **Actions menu** list to select the available options in the **Actions** menu. For some actions, you can only enable or disable them. For other actions, you can add one or more instances to the menu.

To select the options to include in the **Actions** menu:

1. To exclude the **Print** option, deselect the **Print** check box. To restore the **Print** option in the **Actions** menu, select the check box.

There are no configuration options for the **Print** option.

2. To exclude the **Export** option, deselect the **Export** check box. To restore the **Export** option in the **Actions** menu, select the check box.

There are no configuration options for the **Export** option.

3. To exclude the **Compare** option, deselect the check box. To restore the **Compare** option in the **Actions** menu, select the **Compare** check box.

For the **Compare** option, you can configure the label to display in the column heading for each record. To configure the column heading:

- a. Click the edit icon for the **Compare** option.
- b. On the **Compare Configuration** dialog, in the **Header** field, type the header to display.

The default is `Record {counter}`, which displays the word "Record" plus a column number based on the order in which you selected the records. For example, Record 1, Record 2, etc.

In addition to the `{counter}` token, you can also use the `{attributeKey}` token, where *attributeKey* is the key name (not the display name) of an attribute for which to display the associated value. For example, entering `{Name}` would indicate to use the value of the Name attribute for the record as the column heading.

The attribute must be in a group that is configured to display for record details. If the attribute is not included in the attributes on the **Compare** dialog, then the value is not displayed in the column heading.

4. The **Actions** menu list initially does not contain any **Pass Parameters** options. You can add one or more of these options, each with unique settings:
  - a. To add a **Pass Parameters** option, click **+Action**. From the menu, select **Pass parameters**.
  - b. By default, the option is enabled when you add it. To exclude the option, deselect its check box. To reenable the option, select the check box.
  - c. To remove a **Pass Parameters** option, click its delete icon.
5. The **Actions** menu list initially does not contain any **Refinement** options. You can add one or more of these options, each with unique settings:
  - a. To add a **Refinement** option, click **+Action**. From the list, select **Refinement**.
  - b. By default, the option is enabled when you add it. To exclude the option, deselect its check box. To reenable the option, select the check box.
  - c. To remove a **Refinement** option, click its delete icon.
6. To change the display order of the **Actions** menu options, drag each option to the appropriate location in the list.

## Configuring actions for attribute values

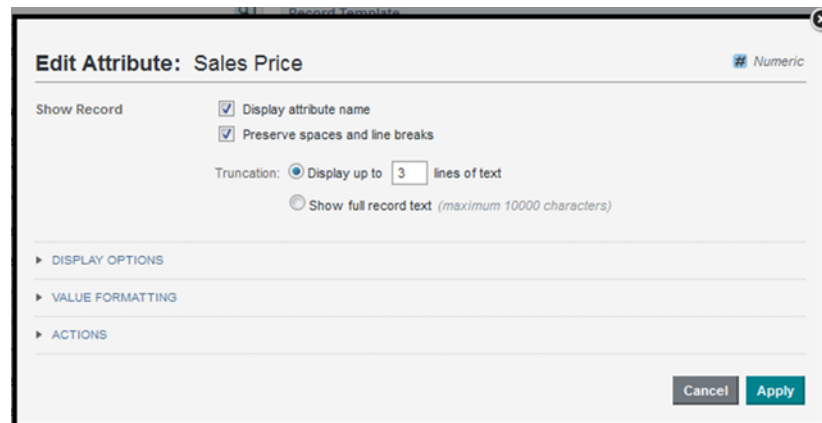
For many components, you can allow users to click displayed values in order to refine data, display record details, or navigate to a specified page or external URL.

When configuring a displayed value, there is an **Actions** section to configure the action.

To configure actions for an attribute in a component:



1. Open the **Edit Attribute** dialog for the attribute you wish to configure:



2. Expand the **Actions** section.
3. If more than one action is available, select the action from the list.

The available actions are:

Action Name	Description
<b>No Action</b>	Indicates that the value is not hyperlinked. For attributes that do not support refinement, this is the default option.
<b>Show Details</b>	When users click the value, the <b>Record Details</b> dialog displays already populated with the details for that record or row.
<b>Refinement</b>	When users click the value, the data is refined to only include records with that value. If the displayed attribute supports refinement, then this action is selected by default.
<b>Hyperlink</b>	When users click the value, they navigate to the specified URL. A hyperlink can be to another page in the same project, or to an external URL. Hyperlinks to external URLs can include attribute values as parameters.

4. To configure a **Show Details** action:
  - a. Set title of the **Record Details** dialog in the **Window Name** field.
5. To configure a **Refinement** action:
  - a. Set the **Target Page** to the desired destination page for the attribute value link. Values include:
    - **Current page** - Stays on the current page.
    - **Other page** - Navigates to the specified page.

For details on providing a target page for refinement, see [Selecting the target page for a refinement or hyperlink](#).

6. To configure a **Hyperlink** action:
  - a. Optionally, display a description tooltip for the action by selecting the **Display action description in tooltip** check box.
  - b. If you have enabled an action tooltip, enter text in the **Action Description** field.
  - c. To create a new browser window for the destination page, select the **Open link in a new window** check box.
  - d. Enter the hyperlink destination in the **URL** field.

You can link to:

- A different page in the same project. For details on providing a target page for a hyperlink, see [Selecting the target page for a refinement or hyperlink](#).
- An external URL. Ensure the URL is correctly formed and that you have properly encoded any special characters.

You must provide the full URL, starting with the protocol (HTTP, HTTPS, FTP, etc.)

A hyperlink to an external URL can include attribute values.

The values may be query parameters:

```
http://www.acme.com/index.htm?p1="Red"&p2="1995"
```

Values may also be part of the URL path:

```
http://www.acme.com/wines/1995/
```

To add attribute values to an external URL:

- a. Click **+ Add URL Parameters**.
- b. On the add parameters dialog, in the attribute list, select the check box next to each attribute to add.
- c. When you are finished selecting attributes, click **Apply**.

The selected attributes are displayed in a table, with each attribute assigned an ID to use when inserting the attribute into the URL.

The attributes are also inserted as query parameters, where the parameter name is the attribute key, and the parameter value is  $\{IDNumber\}$ , where *IDNumber* is the ID for that attribute. For example: `http://www.acme.com/index.htm?Region={0}&WineType={1}`

By default, the value is encoded. To not encode the value, change the format to  $\{\{IDNumber\}\}$ . For example:  $\{\{0\}\}$

You can also use the ID numbers to insert the attribute values manually.

For details on component hyperlinks and encoding inserted attribute values, see [Configuring hyperlinks to external URLs](#).

- d. To remove a URL parameter from the table, click its delete icon.

If you did not edit the inserted query parameter, then Big Data Discovery also removes it from the URL.

If you did edit the inserted query parameter, then you must remove the parameter from the URL manually.

If you inserted the attribute value manually, then you also must remove it manually.

7. Click **Apply**.

## About component hyperlinks

Some components can include hyperlinks from the displayed records to external URLs.

In some cases, the URLs are static, with the exact same target displayed for every record.

More commonly, a URL contains one or more variable placeholders, each representing an attribute from the underlying data. At runtime, Big Data Discovery replaces the placeholders with the actual attribute values for the current record.

For example, for a list of products, a **Results List** component might include a hyperlink to the product web page. To find the correct page, the hyperlink includes the product identifier as a dynamic URL parameter.

When configuring these hyperlinks and inserting dynamic URL parameters, you should be aware of the security risks and recommendations associated with them, including:

- Encoding versus not encoding the inserted values
- Using non-HTTP (file or FTP) links
- Storing full URLs as attribute values

## About using non-HTTP protocol links

While Studio does not prevent you from inserting links that use a non-HTTP protocol, such as file:// or ftp://, using these types of links is not recommended.

Most modern browsers are configured by default to disallow file URLs on web pages retrieved over HTTP. This is a security precaution to ensure web pages cannot execute files on a user's system.

When users click these types of links, the browser ignores them and does not display the link target.

Similarly, not all browsers are guaranteed to support FTP links, and users may not have FTP clients available to get access to content hosted over this protocol.

If users need to reach content on a file system or FTP server, it is recommended that you set up a configuration such as a web servlet to stream files through an HTTP interface. This allows all of your links to be secure HTTP links, and ensures consistent access for users with all browser configurations.

## Selecting the target page for a refinement or hyperlink

You can configure refinement actions to display a different page. You can also create hyperlink actions that navigate to a different page.

For example, when users refine by a value on a **Results Table** component on one page, users could be redirected to a different page. Note, however, that the refinement affects all of the components where it applies, no matter what page the components are displayed on.

On the component edit view, the target page setting for a refinement action uses radio buttons to indicate whether to stay on the current page or navigate to a different page.



### Using the internal page name

When specifying the target page for a refinement or hyperlink, you must use the internal page name used in the URL, not the display name used on the page tab.

Studio creates the internal name automatically when you create the page. The internal name removes spaces and special characters.

For example, if the page name displayed on the page tab might be **Data Results**, then the page name in the URL might be **data-results**.

While you can change the display name for a page, the internal page name does not change.

### Selecting a tab on a Tabbed Component Container

If the target page includes a **Tabbed Component Container** component, then to specify the tab that is selected, you append to the page name:

```
#tabComponentName[tabNumber]
```

Where:

- *tabComponentName* is the name of the **Tabbed Component Container**.
- *tabNumber* is the number (1, 2, 3, etc.) of the tab to select.

So for example, for the following target value:

```
analyze#Sales Numbers[1]
```

- The end user is redirected to the *analyze* page.
- On the page, the first tab of the **Sales Numbers** tabbed component is selected.

To select the tab to display for multiple tabbed components, use a double colon (::) to delimit the components.

For example, for the following target value:

```
analyze#Sales Numbers[1]::Quarterly Forecast[2]
```

- The user is redirected to the *analyze* page.
- On the **Sales Numbers** tabbed component, tab 1 is selected.
- On the **Quarterly Forecast** tabbed component, tab 2 is selected.

## Using component IDs to specify a Tabbed Component Container

Because the double colon (::) is part of the target page syntax, you should avoid using it in your tab titles. You also should avoid multiple tabbed component containers with duplicate titles.

If you cannot avoid these naming features, then when defining a target page, you must use a component's ID rather than its name.

To find a component's ID:

1. Hover your mouse over the tab until the URL appears in the browser status bar.
2. Extract the `p_p_id` parameter from the URL.

```
/dashboard?p_p_id=nested_tabs_INSTANCE_7r3l&p_p_lifecycle=1&p_p_state=
```

For example, for the following target value:

```
analyze#nested_tabs_INSTANCE_0CbE[2]::nested_tabs_INSTANCE_Ja6E[1]
```

- The end user is redirected to the `analyze` page.
- On the tabbed component with ID `nested_tabs_INSTANCE_0CbE`, tab 2 is selected.
- On the tabbed component with ID `nested_tabs_INSTANCE_Ja6E`, tab 1 is selected.

## Recommendations for attribute values that are complete URLs

It is possible that your source data contains complete URLs (including the protocol, host, port, and path) that are ingested when the data is loaded. For example, company data could include links to pages on the company web site.

If you store the full value, and then insert the value as the hyperlink URL, you would not be able to encode it. Because Big Data Discovery does not parse the attribute value as part of an absolute URL, encoding it would corrupt the URL, and the link would not work.

For example, `http://www.mycompany.com/page1` would become `http%3A%2F%2Fwww.mycompany.com%2Fpage1`.

Because of this, storing the full URL in your data is not recommended.

For these types of attributes, during the data ingest process, it is recommended that you use one of the following approaches:

1. **Use one or more attributes to store only the parameter values for each record's URL.**

When you configure the URL in a component, you would then manually type the standard part of the URL into the component configuration, and use encoded attributes for the query String parameters. For example:

```
http://server.mycompany.com/path/to?file={0}
```

0 is a number from the list of selected parameters.

You can only use this approach if all of the URLs have the same structure. If this is not the case, then you may need to use one of the other approaches.

**2. Store the structural portions of the URL in a separate attribute from the parameter values.**

In this approach, the structural portions of the URL such as the protocol, hostname, port, and context path/delimiters are stored in one or more attributes. The parameter portions of the URL that represent identifiers are stored in separate attributes as in approach 1 above.

When you enter the URL in the component, you would not encode the structural attributes, but would encode the parameters. For example:

```
{{0}}/path/to/{1}?file={2}
```

0, 1, and 2 are numbers from the list of selected parameters.

**3. Use a single attribute for the full URL, but have the data ingest process encode any non-structural portions of the URL, such as query String parameters.**

This prevents script injection and addresses any disallowed characters. When you enter the URL, you can then use the attribute value without further encoding. For example:

```
{{0}}
```

0 is a number from the list of selected parameters.

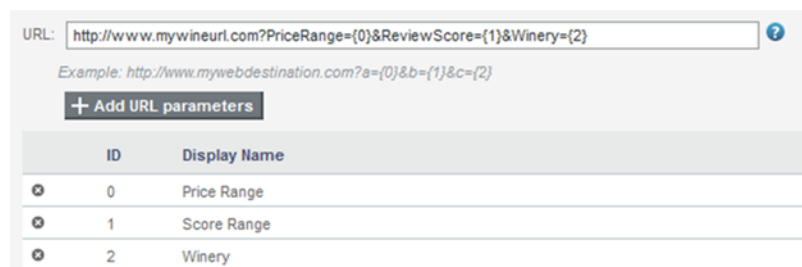
## Indicating whether to encode inserted attribute values

When you insert an attribute value as a URL parameter, you can specify whether Studio encodes the value before inserting it into the URL.

URLs must use ASCII characters, and must avoid using certain reserved characters used as part of the URL syntax. URL encoding replaces any non-conforming characters with a percent sign followed by a hexadecimal numeric representation of the character.

For example, the forward slash (/) is a reserved character. If a URL includes the phrase "CD/DVD", then URL encoding would change that to "CD%2FDVD".

To insert parameters into a URL, you first select a list of attributes, each of which is assigned a number. You then use the number to insert the parameter.



The format uses the single and double braces to indicate encoding versus not encoding. By default, the values are encoded, and the parameters are enclosed by a single set of braces:

```
{parameterNumber}
```

For example:

```
http://www.acme.com/index.htm?p1={0}&p2={1}
```

To have Studio not encode a hyperlink parameter, use a double set of braces:

```
{{parameterNumber}}
```

For example:

```
http://www.acme.com/index.htm?p1={{0}}&p2={{1}}
```

It is important to understand that avoiding encoding can pose a security risk, because data is rendered in the user's browser without intervention, potentially allowing scripts or HTML to be injected directly into a user's browser.

Similarly, if the attribute contains reserved characters that are disallowed in a URL, the resulting hyperlink may be invalid or incorrect and therefore not work for the end user.

As a result, using the non-encoding syntax is not recommended. Unless you know that the underlying data is already encoded, you should always use the encoding syntax.

## Configuring hyperlinks to external URLs

Components can include hyperlinks to external URLs, which can present security risks.

## Configuring a Pass Parameters Actions menu action

For a **Pass Parameters** option in a component **Actions** menu, you can configure the URL and whether to display it in a new browser window. You can also include attribute values as POST parameters.

To configure a **Pass Parameters** option:

1. Click its edit icon.
2. On the configuration dialog, in the **Action name** field, type the label to use for this option.
3. To display the URL in a new browser window, select the **Open link in a new window** check box.
4. In the **URL** field, type the URL to post to.

Make sure that the URL is correctly formed, and that special characters are properly encoded. You must provide the full URL, starting with the protocol (HTTP, HTTPS, and so on).

5. To add attribute values as POST parameters:

- a. Click **+ Add POST parameters**.

The list of available attributes displays.

- b. For each attribute you want to add, select its check box.

- c. When you are finished selecting the attributes to add, click **Apply**.

The selected attributes display in a list.

**New Pass Parameters**

Hyperlink

Action name:

Open link in a new window

URL:

POST parameters: [?](#) [+ Add POST parameters](#)

Display Name	Parameter Name
<input checked="" type="radio"/> Sales Price	P_Price
<input checked="" type="radio"/> Base Score	P_Score

▶ ACTION CONDITION

- d. To remove a parameter, click its delete icon.
- 6. To save the action configuration, click **Apply**.



## Containers

Containers allow you to group components on a page. Containers can also be nested within each other for further organization. You can move a container along with any components inside it as a single unit.

### Adding a Column Container

The Column Container groups components into one or more columns in order to organize and position them on a page.

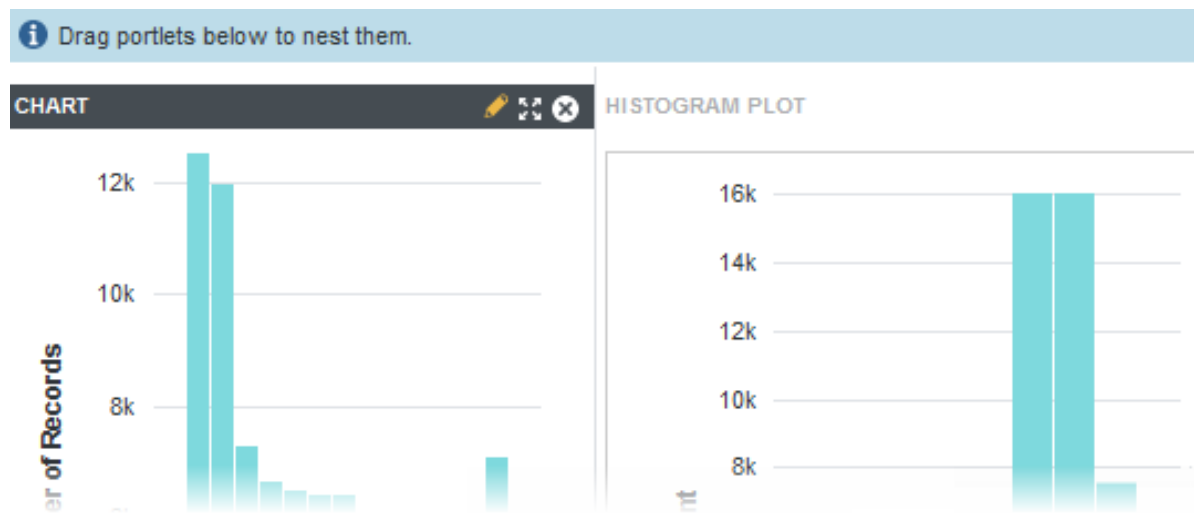
### Adding a Tabbed Container

The Tabbed Container groups components into multiple tabs. Each tab may contain one or more components.

## Adding a Column Container

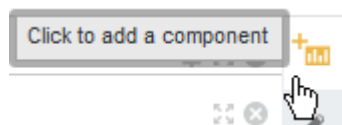
The Column Container groups components into one or more columns in order to organize and position them on a page.

By default, a new Column Container is configured as two equal-width columns, as in the example below:



To add a Column Container:

1. In the **Discover** View of your project, click the Add Component icon in the right sidebar:



The **Add Component** panel appears.

2. Select the **Column Container** and click the + icon or drag it to the desired location on the page.
3. To configure the column layout:
  - a. Click the **Options** gear on the right side of the Column Container menu bar to access the Component Edit View.
  - b. From the **Layout Template** menu, select the layout to use.

Any components already inside the container remain in their specified columns, or are repositioned to the first column if the column they are in is not present in the new layout.

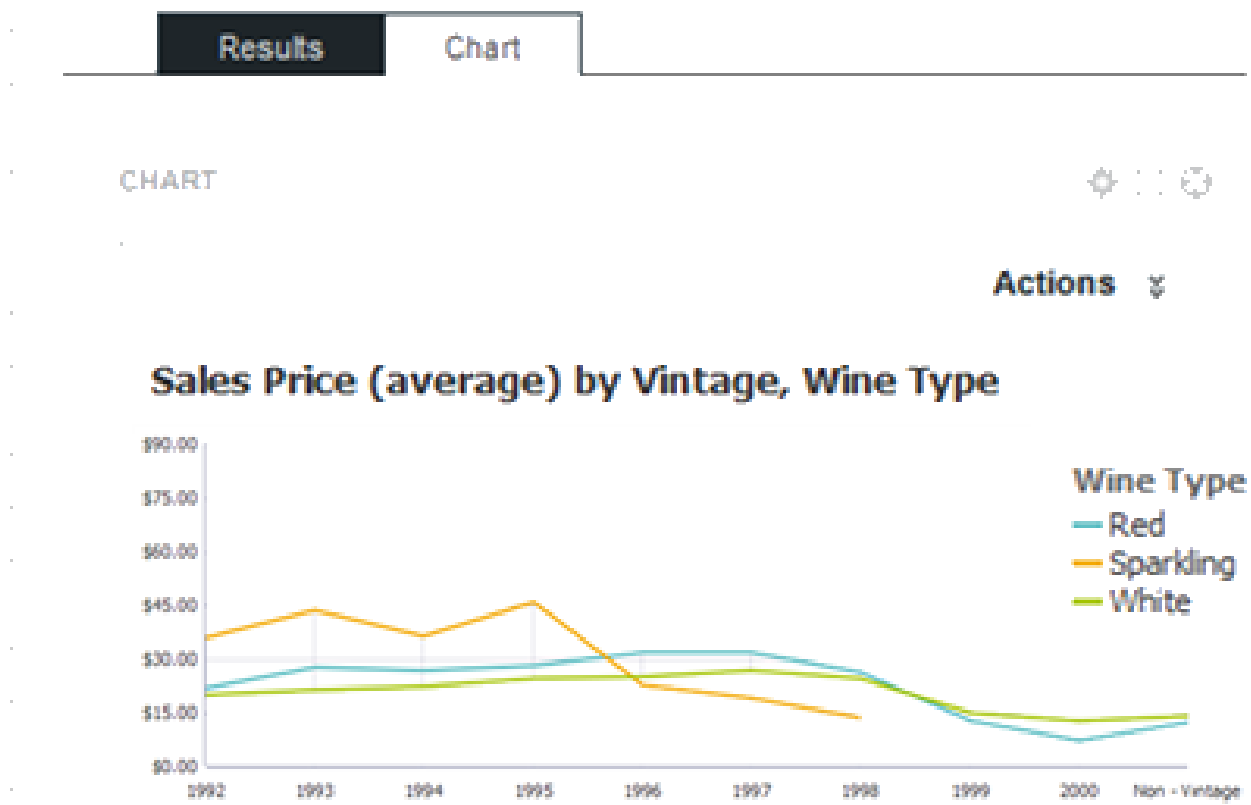
- c. Click **Save**.
- d. Click **Exit** in the upper right corner.

After configuring the container, you can drag other components into it and configure them as normal.

## Adding a Tabbed Container

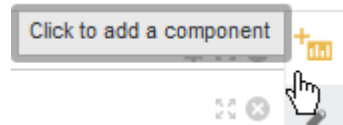
The Tabbed Container groups components into multiple tabs. Each tab may contain one or more components.

By default, the container includes two tabs. The example below shows a Line Chart and a Results List included as separate tabs:



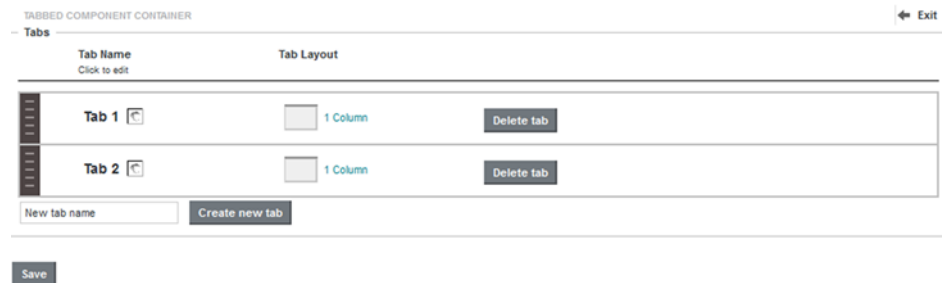
To add a Tabbed Container:

1. In the **Discover** View of your project, click the Add Component icon in the right sidebar:



The **Add Component** panel appears.

2. Select the **Tabbed Component Container** and click the + icon or drag it to the desired location on the page.
3. Click the **Options** gear on the right side of the Tabbed Container menu bar to access the Component Edit View:



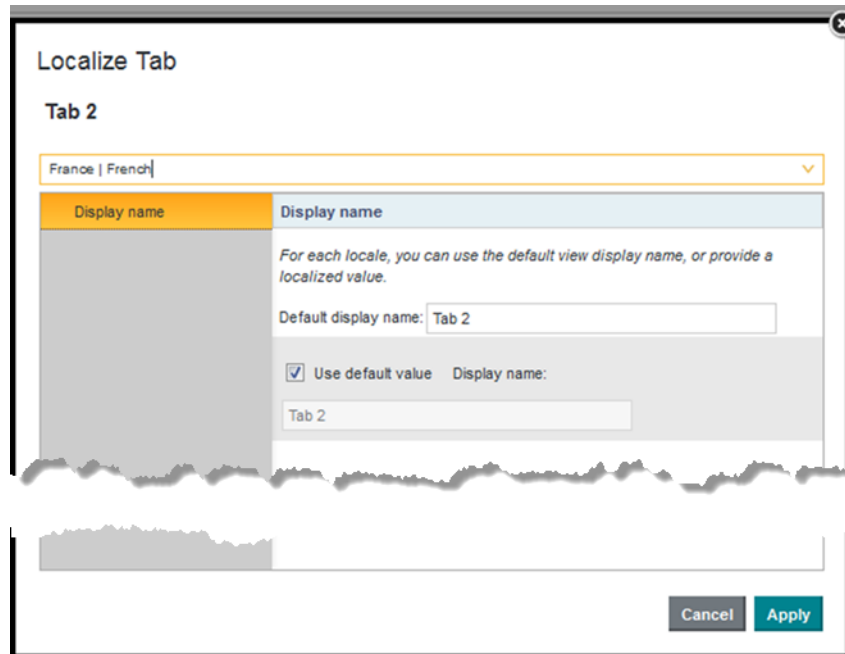
4. To add a new tab:
  - a. In the **New tab name** field, enter the name of the new tab.
  - b. Click **Create new tab**.
5. To change the layout of a tab:
  - a. Click the layout in the **Tab Layout** column.
 

The **Choose a Layout** dialog appears.
  - b. Select the layout to use.
  - c. Click **OK**.
6. To change the tab order, click and drag the handle on the left side to move each tab to the desired location.
7. To delete a tab, click **Delete tab** for the tab.
8. To rename a tab:
  - a. Click the tab name.
 

The tab name becomes an editable field.
  - b. In the field, enter the new label for the tab.
  - c. Click **OK**.

9. To localize a tab name:
  - a. Click the localization icon next to the tab name.

The **Localize Tab** dialog displays:



- b. From the locale list, select the locale for which to provide a value.
- c. Deselect the **Use default value** check box.

By default, each locale uses the display name configured in the Component Edit View.

- d. In the **Display name** field, enter the localized tab name.
- e. Click **Apply**.

10. Click **Save**.

11. Click **Exit** in the upper right corner.

After configuring the container, you can drag other components into it and configure them as normal.

## Data Record Displays

These components provide a detailed view of record results for a set of refinements.

### Results List

The **Results List** component displays a list of records in a list format similar to regular web search results. It is designed to provide a meaningful summary of results, and is especially useful for records that include unstructured data such as long text fields.

### Results Table

The **Results Table** component displays a set of data in a table format.


## Results List

The **Results List** component displays a list of records in a list format similar to regular web search results. It is designed to provide a meaningful summary of results, and is especially useful for records that include unstructured data such as long text fields.

In the list, each record displays a configurable set of attributes:

### RESULTS LIST

2 RECORDS SELECTED 

p\_body: [Soft, Tannins](#)   
 p\_datereviewed: 10/31/97  
 p\_description\_newName: [Straightforward Barolo, quite light but with an attractive toasted oak, cherry, mineral and](#)  
 p\_drinkability: [Best after 0](#)

p\_body: [Complex, Elegant](#)   
 p\_datereviewed: 11/30/93  
 p\_description\_newName: [An excellent spätlese in which the sweet and dry elements mix in an elegant, complex](#)  
 p\_drinkability: [Best after 1995](#)

p\_body: [Deep, Firm, Rich, Ripe](#)   
 p\_datereviewed: 3/15/94

Displayed attributes can include links that refine the data, display record details, or navigate to a URL. Each record may also include a link that displays the **Record Details** dialog for the record.

The Results List includes the following common functions, summarized in [Common component controls](#):

- [Paging](#)

- [Actions > Compare](#)
- [Actions > Print](#)
- [Actions > Export](#)

#### [Sorting records in a Results List](#)

The Results List **View Options** menu includes a **Sort** option to sort the records based on the displayed attributes.

#### [Configuring a Results List](#)

A newly-added **Results List** component displays five attributes for each record. You determine how the component displays and how end users can interact with each displayed attribute.

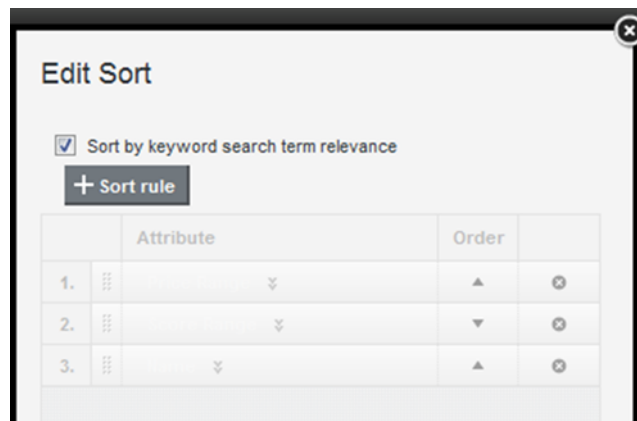
## Sorting records in a Results List

The Results List **View Options** menu includes a **Sort** option to sort the records based on the displayed attributes.

To sort records:

1. Select **View Options > Sort**.

The **Edit Sort** dialog displays:



If there is currently a search refinement in place, the **Sort by keyword search term relevance** check box is enabled. Deselect the check box if you want to sort by one of the displayed attributes.

2. To add a sort rule:
  - a. Click **+Sort rule**.
  - b. In the **Attribute** column, click **Select a column** and browse to or search for an attribute to sort by.
  - c. Click **Apply**.
  - d. Click the triangle icon in the **Order** column to switch between Sort Ascending or Sort Descending.

The mouseover tooltip displays the current sort order.

- e. Click **Apply**.
3. To remove a sort rule:
  - a. Click the **X** icon in the rightmost column.
  - b. Click **Apply**.
4. To reset the sort order, select **View Options > Reset to default**.

## Configuring a Results List

A newly-added **Results List** component displays five attributes for each record. You determine how the component displays and how end users can interact with each displayed attribute.

To configure a Results List:

1. Choose a data set view:
  - a. In the component menu bar, click the Options icon to navigate to the Edit view:



The Edit view opens to the **Data Selection** tab.

The tab contains the list of views for the project. Views that are not valid for the selected component or sub-component are disabled in the UI.

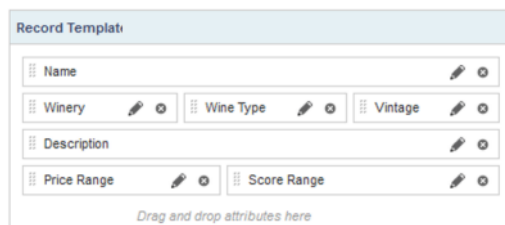
The View list includes the view name, description, whether the view is a base or custom view, the data sets the view is created from, and the identifying attributes for the view.

- b. (Optional) To display the list of attributes in a view, click its information icon.
- c. Select a view by enabling the radio button.
- d. Click **Save**.

A message bar displays the result of the save operation.

2. Select the attributes to display for each record in the list:
  - a. In the Edit view, navigate to the **List Template** tab.

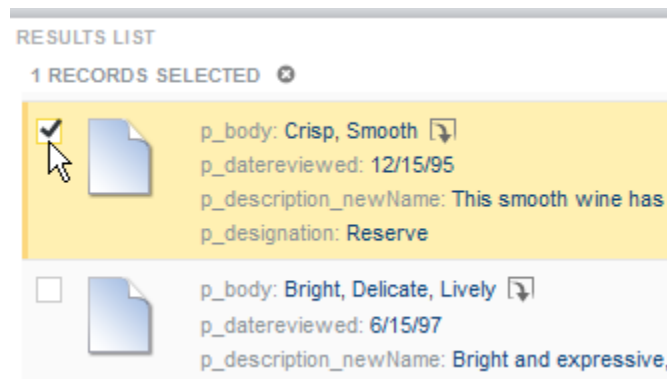
The Results List record display includes multiple rows, each of which can contain up to 3 attributes. You can add or remove rows and attributes from the **Record Template** pane in this tab:



- b. To add an attribute to the template, drag the attribute from the attributes list to the desired location in the Record Template.  
You can create a new row for the attribute or add it to an existing row.
  - c. To change the location of a selected attribute, drag it to another location in the template.
  - d. To remove an attribute from the template, click its delete icon.
3. Configure the format of each displayed attribute:
    - a. In **Record Template** pane, click the edit icon for an attribute you wish to configure.
    - b. Configure the attribute as described in [Configuring attribute value formats](#).
  4. Configure available actions for each displayed attribute:
    - a. In **Record Template** pane, click the edit icon for an attribute you wish to configure.
    - b. Configure the actions as described in [Configuring actions for attribute values](#).
  5. Click **Save**.

### Configuring images to display next to each record

You can configure the images that display next to each record in a Results List component.



Studio supports the following image sources:

Image Type	Description
Icons selected from the image gallery	This option is typically used when the icons represent a category or other high-level classification of the records. Studio provides a basic set of icon images. You can also upload your own using .gif, .jpeg/.jpg, .bmp, and .png file types. When you upload images, they become available to use for any <b>Results List</b> component in the same project.



Image Type	Description
Images from a URL that you provide	This is typically used when most of the records in the system have associated images published to a Web or application server (for example, a unique preview image for products represented by each record). The URL can include tokens substitute in attribute values as parameters.

To configure images to display next to each record:

1. In the component menu bar, click the Options icon to navigate to the Edit view:



2. In the Edit view, navigate to the **Images** tab.
3. To disable image display, select **Type of image > None**.

This is the default configuration.

4. To display an icon from the image gallery:

- a. Select **Type of image > Icon from gallery**.
- b. Click **Select Attribute**.

The attribute selection dialog displays the attributes that can be used for the image selection. The attribute must have fewer than 15 values, and must be displayed on the component.

- c. Click the attribute to use and click **Apply**.
- d. To select an image for an attribute value, click **Select**.

The **Select Image** dialog opens.

- e. Click the image you want to use or click **No Image** to disable images for a value.

To upload your own image, click **Browse**, locate the image, and click **Upload**. The uploaded image is added to a new **My Images** category on the **Select Image** dialog. If the image is too large, Studio crops it.

- f. Click **OK**.

The list updates with the selected image.

- g. To select a different image for an attribute value, click the edit icon.
- h. To clear a selected image and display the default image, click the delete icon.
- i. Hide or change the default icon by clicking the **None** radio button below the attribute value list, or by clicking **Change Icon**.

5. To display images from a URL:

- a. Under **Type of image**, select **Thumbnail from URL**.

- b. In the field, type the URL for the image file.
- c. To add attribute values as parameters in the URL, click **Add URL Parameters**.
- d. On the add parameters dialog, in the attribute list, check the check box next to each attribute to add.
- e. When you are finished selecting attributes, click **Apply**.

The selected attributes are displayed in a table, with each attribute assigned an ID to use when inserting the attribute into the URL.

The attributes are also inserted as query parameters, where the parameter name is the attribute key, and the parameter value is `{IDNumber}`, where `IDNumber` is the ID for that attribute. For example: `http://www.mywines.com/images?Designation={0}`

By default, the value is encoded. To not encode the value, change the format to `{{IDNumber}}`. For example: `{{0}}`

You can also use the ID numbers to insert the attribute values manually.

For details on component hyperlinks and encoding inserted attribute values, see [Configuring hyperlinks to external URLs](#).

- f. To remove a URL parameter from the table, click its delete icon.

If the parameter is unedited, Studio also removes it from the URL. If you have edited the inserted query parameter or inserted the attribute value manually, you must manually remove it from the URL.

6. Hide or change the default icon by clicking the **None** radio button or by clicking **Change Icon**.

The default image setting determines the image to display if Studio can't find a configured image, or if the **No Image** option is selected for a value. You can also use this setting to display the same image for all records.

### Configuring display and pagination options for a Results List

For a **Results List** component, you can configure the component height, whether to enable pagination, and, if pagination is enabled, whether to allow end users to select the number of results per page.

To configure pagination and navigation options for a Results List:

1. In the component menu bar, click the Options icon to navigate to the Edit view:



2. In the Edit view, click the **Display Options** tab.
3. In the **Component height** field, type the height in pixels for the component.
4. To allow end users to select the number of results to display on each page, select the **Enable end user results per page controls** check box.

- a. In the **Available results per page options** field, type a comma-separated list of selectable records per page values.
  - b. In the **Default results per page** drop-down, select the default number of results to display per page.
5. If you do not allow end users to select the number of results per page, enter the number of results to display in the **Default results per page** field.

## Results Table

The **Results Table** component displays a set of data in a table format.

The data displayed in the **Results Table** component is either:

- A flat list of records from a selected view. Each row represents a single record. The columns contain attribute values for that record:

RESULTS TABLE

0 RECORDS SELECTED

Wine ID	Name	Description	Wine Type	Vintage	Winery
34707	Alentejo Conventual	Some plum and cherry fla...	Red	1992	Adega Cooperativa de Po...
34708	Alentejo Monte Velho	This shows suppl, flesh...	Red	1992	Herdade do Esporao
34709	Alentejo Tinto da Anforma	Smells like a melange of s...	Alentejo, Red	1992	J.P. Vinhos
34710	Alentejo Tinto da Talha	Soft, generous and fruity...	Alentejo, Red	1992	Roquevale
34711	Alentejo Vinha do Monte	Here's a bruiser. Deeply ...	Red	1992	Sogrape
34712	Alentejo	Fruity, concentrated and ...	Red	1992	Adega Cooperativa de B...
34713	Alliance California	Rustic, rough-and-tumble...	Red, Rhone Blend	1992	R.H. Phillips
34714	Aloxe-Corton	Simple and appealing, w/E...	Burgundy Cote de Be...	1992	Bouchard Pere & Fils
34715	Aloxe-Corton	Interesting floral and rhu...	Burgundy Cote de Be...	1992	Jaffelin
34716	Aloxe-Corton	Light and sweet-tasting, ...	Burgundy Cote de Be...	1992	Louis Latour

PAGE 1 OF 2854

1:20 OF 57076

- An aggregated list of metric values calculated from attribute values. The metric values are aggregated using dimension values displayed at the left of the table. Each row contains the values for a unique combination of dimension values:

RESULTS TABLE

0 RECORDS SELECTED

Region	Vintage	Record Count	Price (average)	Review Score (aver...
Alsace	1992	116	\$18.95	84.04
Alsace	1993	105	\$35.15	86.13
Alsace	1994	145	\$37.09	85.88
Alsace	1995	188	\$32.99	86.23
Alsace	1996	260	\$28.80	86.29
Alsace	1997	270	\$30.36	85.46
Alsace	1998	196	\$23.83	85.30
Alsace	1999	11	\$12.73	83.55
Alsace Summary		1,291	\$29.46	85.67
Argentina	1992	10	\$10.42	81.68
Grand Summary (474 results)		55,099	\$26.74	84.84

PAGE 1 OF 28

1:20 OF 555

The Results Table includes the following common functions, summarized in [Common component controls](#):

- [Paging](#)

- [Actions > Compare](#)
- [Actions > Print](#)
- [Actions > Export](#)

#### [About the Results Table layout](#)

A **Results Table** always contains a set of persistent columns, and can also contain one or more column sets.

#### [Configuring a Results Table](#)

A newly-added **Results Table** component uses default configuration. You can configure the table type, layout, and available end user options.

#### [Using Results Table values to refine the data](#)

You can use the values in a **Results Table** to refine the data.

#### [Navigating to other URLs from a Results Table](#)

A **Results Table** can include hyperlinks to other URLs. The hyperlink can pass parameters in the form of attribute values from a row or rows.

#### [Exporting data from a Results Table](#)

**Results Table** supports exporting data to HDFS or the desktop.

## About the Results Table layout

A **Results Table** always contains a set of persistent columns, and can also contain one or more column sets.

Persistent columns are always displayed to the left of the table.

- For a record list, any column can be a persistent column. One use of persistent columns in a record list might be to display identifying information for each record, such as an ID or name.
- For aggregated tables, the persistent columns always contain the dimension values used to aggregate the metric values.

For example, the table could have persistent columns for the Country and Product Line dimensions. The values in the other columns are then aggregated by the country and product line values.

To define persistent columns, see [Creating persistent columns](#).

For an aggregated table, the persistent columns are by default locked to the left of the table, and do not scroll horizontally with the other columns. For a record list, the persistent columns are unlocked by default.

The columns for the currently selected column set display after the persistent columns. For aggregated tables, the columns contain the aggregated metric values. To define column sets, see [Creating column sets](#).

Aggregated tables can also contain summary rows. There can be summary rows to aggregate values across a dimension value, as well as a summary row that aggregates the values across the entire table. The same aggregation method is used for a metric throughout the table.

RESULTS TABLE

0 RECORDS SELECTED

VIEW OPTIONS ACTIONS

Region	Vintage	Record Count	Price (average) ↕	Review Score (aver... ↕
Alsace	1992	116	\$18.95	84.04
Alsace	1993	105	\$35.15	86.13
Alsace	1994	145	\$37.09	85.88
Alsace	1995	188	\$32.99	86.23
Alsace	1996	260	\$28.80	86.29
Alsace	1997	270	\$30.36	85.46
Alsace	1998	196	\$23.83	85.30
Alsace	1999	11	\$12.73	83.55
Alsace Summary		1,291	\$29.46	85.67
Argentina	1997	10	\$10.47	81.68
Grand Summary (474 results)		55,099	\$26.74	84.84

PAGE 1 OF 28

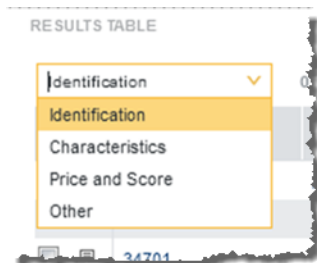
1-20 OF 555

You can page through and sort results. You may be able to select the columns to display, and use displayed values to refine the data or display related content.

### Selecting the column set to display on a Results Table

If a **Results Table** has multiple available column sets, then a list of the column sets displays above the table.

From the column set list, select the column set you want to display.



The column set columns are displayed to the right of the persistent columns.

### Changing the layout of a Results Table

You may be able to change the **Results Table** layout, including the columns that are displayed, the display order, and whether to display summary rows for an aggregated table.

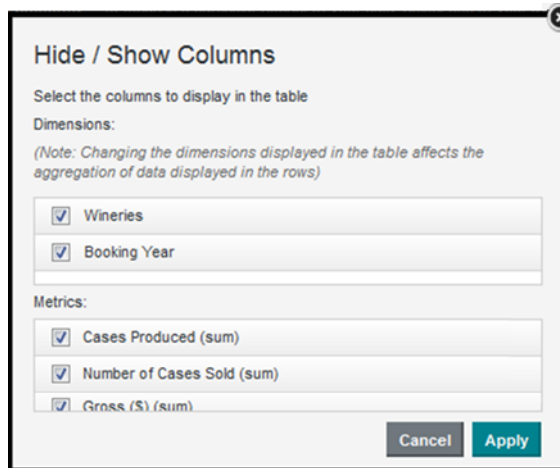
#### Showing, hiding, and moving columns

You may be able to show and hide columns in the **Results Table**. For aggregated tables, when you show a hidden dimension, it is always added to the dimension columns area.

To show, hide, or move columns:

1. To hide or show columns:
  - a. From the **View Options** menu, select **Hide / Show Columns**.

The complete list of available columns displays. For aggregated tables, the dimension and metric columns are grouped as separate lists:



- b. To show or hide a column, select or deselect its check box.
  - c. Click **Apply** to save changes.
2. To move a column, click the column heading and drag it to a new location.

You can only move persistent columns or dimensions within the persistent columns area of the table. You cannot move columns from a column set into the persistent columns.

You can restore default column configuration by selecting **View Options > Reset table to default**.

#### Changing the aggregation method used for a metric

In an aggregated **Results Table**, each metric has an associated aggregation method (for example, sum or average). If the aggregation is not built into the metric, then you can assign a different aggregation method.

For example, you could change a column to display the average of the values across the selected dimension values instead of the sum.

To change the aggregation method for a metric:

1. Click the arrow immediately to the right of the column heading.
2. From the list, select the aggregation method.

#### Showing and hiding summary rows on an aggregated table

If an aggregated **Results Table** is configured to allow summary rows, then you can show or hide those rows.

To show or hide the dimension summary rows:

1. Click **View Options**.
2. In the **View Options** menu, to show or hide the summary rows for each dimension, select or deselect the **Summary rows** check box.
3. To show or hide the summary row for the entire table, select or deselect the **Grand summary row** check box.

### Showing and hiding conditional formatting

A **Results Table** can include conditional formatting, where individual cells that contain numbers are highlighted based on the value in the cell. For example, the table may be configured to highlight in red all Sales values that are lower than 1000.

To show or hide available formatting, select or deselect the **View Options > Conditional formatting** check box.

### Sorting a Results Table

For a non aggregated **Results Table** or an aggregated table without summary rows, you can sort the table using any of the displayed columns. You can also create a compound sort, where the table is sorted by multiple columns in order.

For example, you can configure a compound sort that sorts by Country, then by Year. In this case, all records for "United States" are grouped together, then sorted by Year within the group.

To set the sorting for a table:

1. To sort by a single column, click the column heading.

The rows are sorted by the column values in ascending (A-Z or low-high) order.

To reverse the sort order, click the column heading again.

2. To set up a compound sort:

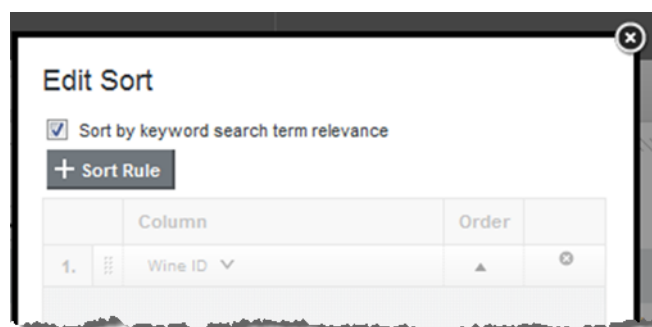
- a. Select **View Options > Sort**.

The **Edit Sort** dialog displays. It shows the current sort order for the table. The rule at the top of the list is used first.

---

**Note:** If there is currently a search refinement in place, then the **Sort by keyword search term relevance** check box is selected.

---



You can deselect the check box in order to sort based on the table columns.

- b. To add a sort rule to the list, click **+Sort Rule**.

An empty sort rule is added to the bottom of the list.

	Column	Order	
1.	Wine ID ▾	▲	⊗
2.	Select a column ▾	▲	⊗

- c. To select a column to use for a sort rule, click the arrow for the column name, then from the list, select the column.  
You cannot select a column that is already being used in a sort rule.
- d. To switch the sort direction for a sort rule, click the sort direction arrow in the **Order** column.
- e. To change the order for applying sort rules, drag and drop each rule to the appropriate location in the list.
- f. To remove a sort rule, click its delete icon.
- g. To save and implement the compound sort, click **Apply**.

#### Sorting an aggregated table with summary rows

For an aggregated **Results Table** that includes summary rows, when you sort by a specific column other than the first dimension, the sort is applied within the parent dimension.

For example, an aggregated table is grouped by Country, then by Year, then by Product Line. The table includes Total Sales and Average Sales for each combination of Country and Year, with summary rows to show the total and average across the entire year and the entire country.

So if you sort by Year, the year values are only sorted in the context of each Country. If you sort by Total Sales, the sales values are sorted within each year.

## Configuring a Results Table

A newly-added **Results Table** component uses default configuration. You can configure the table type, layout, and available end user options.

To configure a Results Table:

1. Choose a data set view:
  - a. In the component menu bar, click the Options icon to navigate to the Edit view:



The Edit view opens to the **Data Selection** tab.

The tab contains the list of views for the project. Views that are not valid for the selected component or sub-component are disabled in the UI.

The View list includes the view name, description, whether the view is a base or custom view, the data sets the view is created from, and the identifying attributes for the view.



- b. (Optional) To display the list of attributes in a view, click its information icon.
- c. Select a view by enabling the radio button.
- d. Click **Save**.

A message bar displays the result of the save operation.

2. Select the type of Results Table to display:

- a. In the Edit view, navigate to the **Table Type** tab.
- b. Select a table type:
  - To create a flat list of records, select **Details table**.
  - To create an aggregated list of metric values, select **Analytics table**.

---

**Important:** If you change the table type after configuring the table layout, you lose the entire table configuration.

---

3. If you are displaying a flat list of records, configure the columns:

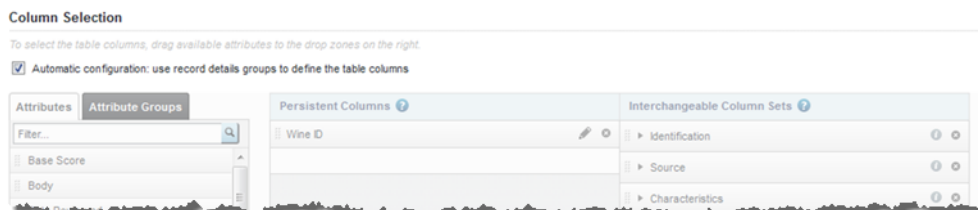
For a record list **Results Table**, the default column configuration contains the identifying attributes (for a base view) and the view group-by attributes (for other views), and the default record detail groups. The identifying or group-by attributes are used for the persistent columns, and the record detail groups are used for the column sets.

Each view is defined using group-by attributes. For example, from a set of sales data, you could create a view listing the available products. The view might be grouped by product ID, and so the grouping attribute would be the product ID.

Base views do not have group-by attributes, but have attributes that are flagged as identifying attributes for the record. For a data set created by loading personal data, the display name of the identifying attribute is `PRIMARY_KEY`.

Each view can be configured with a set of attribute groups. Each group is configured to indicate whether to use it by default when viewing record details.

- a. In the Edit view, navigate to the **Column Selection** tab.
- b. To use the default configuration, leave the **Automatic configuration: use record details groups to define the table columns** check box selected:



- c. To use custom configuration, deselect the **Automatic configuration: use record details groups to define the table columns** check box.

The default configuration remains, but the lists are enabled to allow you to change the configuration.

If you check the check box again, the default configuration is restored, and any customization is lost.

- d. Add or remove persistent columns as described in [Creating persistent columns](#).
  - e. Optionally, add or remove column sets as described in [Creating column sets](#).
4. If you are displaying an aggregated list of metric values, select the dimensions to use for the aggregation and the metrics to display:
    - a. Add or remove dimension columns as described in [Adding dimension columns to an aggregated table](#).
    - b. Optionally, add or remove column sets as described in [Adding column sets to an aggregated table](#).
  5. Configure selected columns as described in [Configuring column formatting](#).
  6. Add action columns:
    - a. Add action columns as described in [Adding action columns for row-level Results Table actions](#).
    - b. Configure conditions for actions as described in [Configuring conditions for enabling Results Table actions](#).
  7. Configure sorting and a summary row as described in [Configuring the default Results Table sort order and summary row display](#).
  8. Configure the table display as described in [Configuring the Results Table display and navigation](#).
  9. Click **Save**.

### Configuring the Results Table display and navigation

The **Display Options** tab of the **Results Table** edit view contains options for displaying and navigating through the table.

To configure these settings:

1. Under **Table height**, in the text field, type the number of rows to display at one time.
2. Use the pagination fields to configure whether users can select the number of results per page, and the available options.
3. In the **Default column width** field, type the default width in pixels of a table column.
4. To allow end users to show and hide columns, check the **Allow end user to show and hide columns** check box.

If the check box is selected, then the **View Options** menu includes the **Hide / Show Columns** option.

## Configuring the default Results Table sort order and summary row display

The **Sorting & Summaries** tab on the **Results Table** edit view allows you to configure the default sort order to use for the table. For aggregated tables, the tab also determines whether to enable the dimension and table summary rows.

For record list tables and aggregated tables without dimension summary rows, you create a list of sort rules, which are applied in order. For each sort rule, you can select a different column and determine whether to sort in ascending or descending order.

Default Sort Order: + Sort Rule

	Column	Order	
1.	Winery ▼	▲	⊗
2.	Vintage ▼	▲	⊗
3.	Name ▼	▲	⊗

For aggregated tables with dimension summary rows, the sort rules use the dimension columns from left to right, with each dimension sorted in ascending order. You cannot change the order of the sort rules, and you cannot add or remove sort rules. You can only:

- Change the sort direction of each dimension (ascending or descending)
- Change the attribute used in the last sort rule. By default, it is the last dimension in the dimension list, but you can instead select a metric column. The values in the metric column then determine the order of the values for the last dimension column.

Note that on the end user view, only the currently visible columns can affect the sort order. Dimensions that are hidden by default are not available to include in the default sort order.

On the **Summaries & Sorting** tab, to configure the sorting:

1. In the **Maximum number of sort rules** field, type the maximum number of rules that can be used in a compound sort.
2. For record list tables, or aggregated tables that do not display dimension summary rows, to configure the list of sort rules:
  - a. To add a new sort rule, click **+Sort Rule**.  
If the maximum number of sort rules already has been added, then the button is disabled.
  - b. To change the column assigned to a sort rule, click the arrow next to the column name, then select the new column.
  - c. To switch the sort direction for a sort rule, click sort arrow in the **Order** column.
  - d. To change the order for applying sort rules, drag each sort rule to the appropriate location in the list.
  - e. To remove a sort rule, click its delete icon.

3. For an aggregated table that does display dimension summary rows, to configure the sort rules:
  - a. To switch the sort direction for a sort rule, click the sort direction icon in the **Order** column.
  - b. To change the column used in the last sort rule, click the arrow next to the column name, then select the new column.
4. For an aggregated table, the **Summaries** section of the **Sorting & Summaries** tab also allows you to configure whether to display summary rows by default:
  - a. By default, the dimension summary rows are not displayed. To display by default the dimension summary rows, check the **Show summary rows by default** check box.
  - b. By default, the grand summary row displays. To not show the grand summary row, deselect the **Show a grand summary row by default** check box.

### Configuring column formatting

From the **Column Selection** tab of the **Results Table** edit view, you can configure the width, formatting, and other options for all of the table columns.

Note that for a record list table, if you are using the default column configuration, then you cannot configure the columns. Only default settings are used.

To configure the display of a column:

1. From the **Column Selection** tab, to display the **Edit column** dialog, click the edit icon for the column.
2. For aggregated tables, select the aggregation method for each metric.  
See [Selecting the aggregation method to use for a metric](#).
3. For a date/time attribute, from the **Date/time Subset** list, select the date/time subset to display.

The selected date/time subset is included in the column heading on the end user view.



Date of Sale (Year-Month-Day)
1/17/97
10/23/97

4. For persistent columns and dimensions, you can lock the column, so that it is always visible at the left of the table.

To lock the column, check the **Lock column** check box.

To unlock the column, deselect the check box. If the column is not locked, it is still at the left of the table, but can scroll horizontally.

5. By default, all of the columns are visible. However, you can configure a column to be hidden when the table is first displayed.

To hide the column by default, check the **Hide column** check box.

On the **Column Selection** tab, hidden columns are flagged with an icon.

6. For record list tables, you can indicate whether to highlight search terms that appear in the attribute value. To highlight search terms, check the **Highlight search term in context** check box.
7. Optionally, configure the **Attribute Cascade** section to configure a dimension cascade when users refine by the dimension value as described in [Configuring cascading for dimension refinement](#).
8. In the **Display Options** section:
  - a. Under **Alignment**, to select the horizontal alignment to use for the column value, click its radio button.
  - b. Under **Width**, to override the default width, click **Custom**, then in the field, type the width in pixels of the column.
9. The **Value Formatting** section is used to set the format for displaying the column values.

For details on formatting displayed values, see [Configuring attribute value formats](#).

10. The **Actions** section allows you to configure the action that occurs when users click the value.

When the dialog is first displayed, the section is collapsed. To expand or collapse the section, click the section heading.

For general information on configuring column actions, see [Configuring actions for attribute values](#).

In addition to the basic action configuration, for **Results Table** columns, you can use the **Action Condition** setting to also set a condition to determine whether to enable the action based on the value of an attribute in the table. For example, you can configure the action to only be enabled if the **Date Reviewed** column has a non-null value.

Action Condition    If  is  then

For details on configuring action conditions, see [Configuring conditions for enabling Results Table actions](#).

### Creating persistent columns

On the **Column Selection** tab of the **Results Table** edit view for a record list table, the **Persistent Columns** list contains the columns that are always displayed. In the default column configuration, the list contains the identifying or group-by attributes for the view.

You can create a record list table containing only persistent columns.

By default, the persistent columns are not locked, meaning that on the end user view they can scroll horizontally. If you use the default column configuration, you cannot lock the columns.

If you use a custom column configuration, you can lock the persistent columns. If you lock a column, it is moved to the top of the **Persistent Columns** list, and is locked to the left of the table in the end user view. Locked columns do not scroll horizontally.

Persistent Columns ?		
Wine ID	🔒	✏️ 🗑️
Name	🔒	✏️ 🗑️
Wine Type		✏️ 🗑️

On the **Column Selection** tab, to manually select the columns for the **Persistent Columns** list:

1. To add a column to the **Persistent Columns** list, drag an attribute from the **Attributes** tab to the empty slot in the **Persistent Columns** list.
2. To set the display order of the columns, drag each attribute to the appropriate location in the list. The column at the top of the list displays at the far left of the table.
3. To remove a column from the **Persistent Columns** list, click its delete icon.

### Creating column sets

A column set is a group of columns. While persistent columns always display, only one column set displays at a time. If there are multiple column sets, then end users select the displayed column set from a list.

Column sets are optional. You can configure a table with only persistent columns.

For a record list table, a column set may be either:

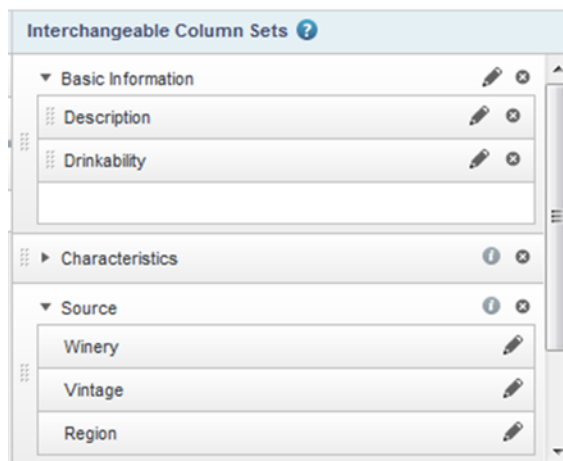
- An attribute group from the original view. For this type of column set, you cannot change the column set name, and you cannot add or remove columns from the set.

- A manually created column set. For this type of column set, you name the set and select the attributes that are in the set.

For the default column configuration, the columns sets consist of the attribute groups configured to be used by default for record details.

On the **Column Selection** tab of the edit view, the column sets are displayed in the **Interchangeable Column Sets** list. If you are not using the default column configuration, then the list always contains an empty set, used to add a new set to the list.

You can expand and collapse each column set to show or hide the list of columns.



The order of the column sets in the list determines the display order in the end user list. The column set at the top of the list is selected by default on the end user view.

To create and configure column sets:

1. To create a new column set from an attribute group, drag the group from the **Attribute Groups** tab to the **Interchangeable Column Sets** list.

The column set uses the attribute group name. You cannot change the name of a column set created from an attribute group.

The column set uses the attributes and attribute display order configured for the attribute group. For column sets created from an attribute group, you cannot add or remove columns, and you cannot change the display order of the columns.

2. To create and configure manual column sets:
  - a. To create a new manual column set, drag an attribute from the **Attributes** tab into the empty column set slot.
 

A new empty set slot is added to the bottom of the list. The default name for a new manual column set is "New set" plus a number if needed to differentiate the name from other column sets.
  - b. To edit the name of a manual column set, click its edit icon.
  - c. To add an attribute to a manual column set, drag the attribute from the **Attributes** tab to the empty slot in the column set.

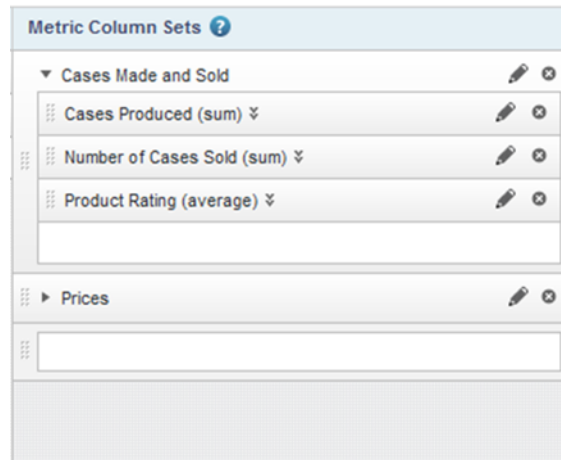
When you add an attribute to a set, a new empty attribute slot is added.

- d. To control the display order of the columns in a manual column set, drag the attribute to the appropriate location in the list. You can also drag attributes to other manually created sets.
  - e. To remove a column from a manual column set, click its delete icon.
3. To determine the display order of the sets in the end user list, drag each set to the appropriate location in the list.
  4. To remove a column set, click its delete icon.

#### Adding column sets to an aggregated table

For an aggregated **Results Table**, a column set is a group of metric values. All metric values display in column sets. While the dimension columns are always displayed, only one column set displays at a time. If there are multiple column sets, then end users use a list to select the column set to display.

On the **Column Selection** tab of the edit view, the **Metric Column Sets** list contains the column sets. Each column set can be collapsed to allow you to see more of the list.



The list always contains an empty column set slot, to allow you to add new sets.

The display order of the **Metric Column Sets** list determines the order they display in within the end user list. The column set at the top of the list is selected by default when the component is first displayed.

To manage the column sets:

1. To add a new column set, drag a metric from the attributes list into the empty column set slot.

A new column set is created containing the selected metric, and a new empty column set slot is added.

The default name for a new column set is "New set", plus a number if needed to differentiate the name from other column sets.

2. To add a column to an existing column set, drag a metric into the empty metric column slot for the column set.
3. To determine the order of the column sets in the end user list, drag each set to the appropriate location in the list.



4. Within each column set, to determine the display order of the columns, drag each column to the appropriate location in the column set.
5. To configure the column set, click its edit icon.

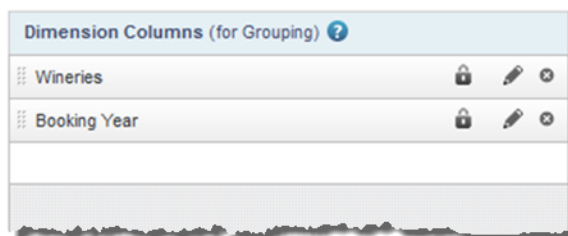
From the **Edit column set** dialog, you can configure the column set name.

6. To remove a column set, click its delete icon.

### Adding dimension columns to an aggregated table

On the **Column Selection** tab of an aggregated **Results Table** edit view, the **Dimension Columns** list contains the dimensions used to aggregate the metrics in the column sets. Dimension columns are always displayed to the left of the table.

By default, the dimension columns are locked, meaning that they do not scroll horizontally in the end user view.



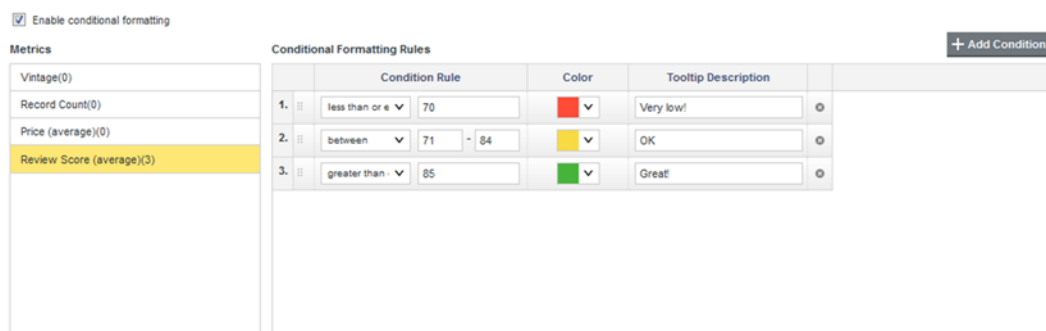
If you do not select any dimension columns, then the table consists of a single row, with the metric values aggregated across the entire data set.

To manage the **Dimension Columns** list:

1. To add a dimension to the list, drag the dimension from the attributes list to the **Dimension Columns** list.
2. To determine the display order of the dimension columns, drag each dimension to the appropriate location in the list.
3. To remove a dimension column, click its delete icon.

### Highlighting Results Table values that fall within a specified range

The **Conditional Formatting** tab of the **Results Table** edit view allows you to highlight specific numeric values in a **Results Table**.



To configure whether to use conditional formatting, and to set the values to highlight:

1. To allow conditional formatting, make sure that the **Enable conditional formatting** check box is selected.

If the check box is not selected, then there is no conditional formatting.

2. In the list, click the attribute or metric for which to configure conditional formatting.

Only numeric attributes and metrics are included in the list.

Each item in the list includes the number of conditional formatting rules configured for it.

3. To add a new range of values to highlight for that attribute, click **Add condition**.

4. To configure a condition:

- a. From the **Condition Rule** list, select the type of comparison.

- b. In the field (or fields, for the **between** or **a range from** options), enter the value(s) against which to do the comparison.

Note that for the **between** option, the values are inclusive. So if you specify a range between 20 and 30, values of 20 and 30 also are highlighted.

- c. From the **Color** list, select the color to use for the highlighting.

- d. In the **Tooltip Description** field, type a brief description of the highlighting. The description is included in the tooltip for each highlighted cell.

5. The conditions are applied based in the order they are listed. So a condition at the top of the list has a higher priority than a condition lower in the list.

To change the priority of the conditions, drag and drop them to the appropriate location in the list.

6. To remove a condition, click its delete icon.

### **Adding action columns for row-level Results Table actions**

You can add actions columns to the **Results Table** to allow end users to perform an action on the specific row.

#### **Available row-level actions**

For row-level action columns in a **Results Table**, the link to the action can be either a text String or an icon.

The available row-level actions are:

Action	Description
<b>Detail</b>	<p>Displays the <b>Record Details</b> dialog, which contains data associated with the selected row.</p> <p>For record list tables, the component displays the attribute groups configured on the <b>Views</b> page to be used to display record details.</p> <p>For aggregated tables, the component displays the columns from the current row.</p> <p>There can be only one <b>Detail</b> action column, which is disabled by default.</p> <p>If the view used for the table does not have any identifying attributes, then the <b>Detail</b> check box is disabled and locked.</p> <p>For a <b>Detail</b> action column, you can configure the title of the <b>Record Details</b> dialog.</p>
<b>Hyperlink</b>	<p>Allows users to navigate to a specified URL. The link can include attribute values from the current row.</p> <p>Users can create multiple <b>Hyperlink</b> action columns, each with unique settings.</p> <p>By default, there are no <b>Hyperlink</b> action columns.</p>
<b>Refinement</b>	<p>Refines the data based on one or more attribute values from the current row.</p> <p>Users can create multiple <b>Refinement</b> action columns, each with unique settings.</p> <p>By default, there are no <b>Refinement</b> action columns.</p>

### Selecting the action columns to include

On the **Actions** tab of the **Results Table** edit view, you use the **Record actions** list to manage the action columns. For the **Detail** action, you can only enable or disable the action. For other actions, you can add one or more instances to the table.

To select the row-level action columns:

1. To include the **Detail** action, check its check box. To exclude the **Detail** action, deselect the check box.
2. The table initially does not contain any **Hyperlink** options. You can add one or more of these columns, each with unique settings:
  - a. To add a **Hyperlink** action column, click **+ Action**. From the menu, select **Hyperlink**.
  - b. By default, the column is enabled when you add it. To exclude the action column, but not remove it, deselect its check box.
  - c. To remove a **Hyperlink** action column from the list, click its delete icon.
3. The table initially does not contain any **Refinement** action columns. You can add one or more of these columns, each with unique settings:
  - a. To add a **Refinement** action column, click **Action**. From the menu, select **Refinement**.

- b. By default, the column is enabled when you add it. To exclude the action column, but not remove it, deselect its check box.
- c. To remove a **Refinement** action column, click its delete icon.

### Configuring a Hyperlink action column

For a **Hyperlink** row-level action column in a **Results Table**, you can configure the name, URL, and whether to display the URL in a new browser window. You can also include attribute values as URL parameters.

To configure a **Hyperlink** row-level action column:

1. On the **Actions** tab of the **Results Table** edit view, click the edit icon for the **Hyperlink** action.
2. In the **Action name** field, type a name for the action.

3. To display the hyperlink in a separate browser window, check the **Open link in a new window** check box.
4. In the **URL** field, type the URL to link to.

Make sure that the URL is correctly formed, and that any special characters are properly encoded.

5. The URL can include attribute values. The values could be query parameter names or values:

```
http://www.acme.com/index.htm?p1="Red"&p2="1995"
```

Or could be part of the URL path:

```
http://www.acme.com/wines/1995/
```

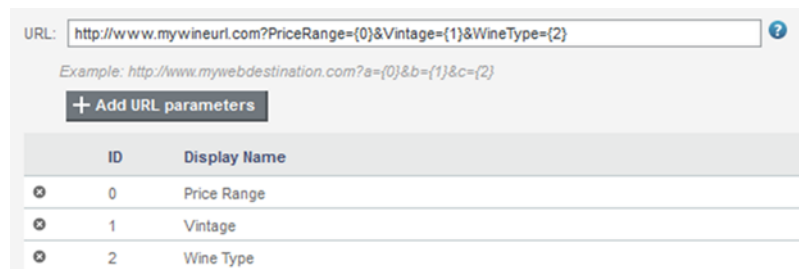
To add attribute values to the URL:

- a. Click **Add URL Parameters**.
- b. On the add parameters dialog, in the attribute list, check the check box next to each attribute to add.
- c. When you are finished selecting attributes, click **Apply**.

The selected attributes are added to a table, with each attribute assigned an ID number.

The attributes are also inserted as query parameter values, where the parameter name is the attribute key, and the parameter value is  $\{IDNumber\}$ , where  $IDNumber$  is the ID for that attribute.

For example: `http://www.acme.com/index.htm?Region={0}&WineType={1}`



By default, the value is encoded. To not encode the value, change the format to  $\{\{IDNumber\}\}$ .

For example:  $\{\{0\}\}$

You can also use the ID numbers to insert the attribute values manually.

For details on component hyperlinks and encoding inserted values, see [Configuring hyperlinks to external URLs](#).

- d. To remove a URL parameter from the table, click its delete icon.

If you did not edit the inserted query parameter, then Big Data Discovery also removes it from the URL.

If you did edit the inserted query parameter, then you must remove the parameter from the URL manually.

You must also remove manually any attribute values that you added manually.

6. To save the configuration, click **Apply**.

### Configuring a Refinement action column

For a **Refinement** action column on a **Results Table**, you can select the attributes to use for the refinement, and the page on which to apply the refinement.

To configure a **Refinement** action column:

1. On the **Actions** tab of the **Results Table** edit view, click the edit icon for the action.
2. On the configuration dialog, in the attributes list, check the check box next to each attribute to use for the refinement.

Note that for aggregated tables, you can only use the dimension columns for refinement.

3. Use the **Target Page** setting to specify the page to display when users do the refinement.

For information on specifying a different page in the project, see [Selecting the target page for a refinement or hyperlink](#).

### Configuring how each Hyperlink or Refinement action column displays

The configuration dialog for each **Hyperlink** or **Refinement** action column in a **Results Table** contains a **Display Options** section. This section is collapsed by default, and is the same for each action. It determines how the column displays in the table.

In the **Display Options** section, to configure the display for an action column:

1. To expand or collapse the section, click the section heading.

2. Under **Alignment**, click a radio button to indicate how to align the action icon or hyperlink within the column.
3. By default, action columns display at the far left of the table, in front of the persistent columns. Under **Position**, to change the column location:
  - a. Click the radio button for the custom location.
  - b. From the location list, select whether to display the column before or after the selected column set.
  - c. From the column sets list, select the column set to display the action column in.  
The list includes all of the column sets in the table, plus an entry to display the column in the persistent columns.

4. Under **Width**, to have the width set automatically, click **Default**.

To set a specific width, click **Custom**. In the field, type the default width in pixels.

5. Under **Behavior**, configure how to display the action column to end users.

To display the action name as a hyperlink:

- a. Click **Hyperlinks**.
- b. To also display the action name in the column heading, check the **Display action name in column header** check box.

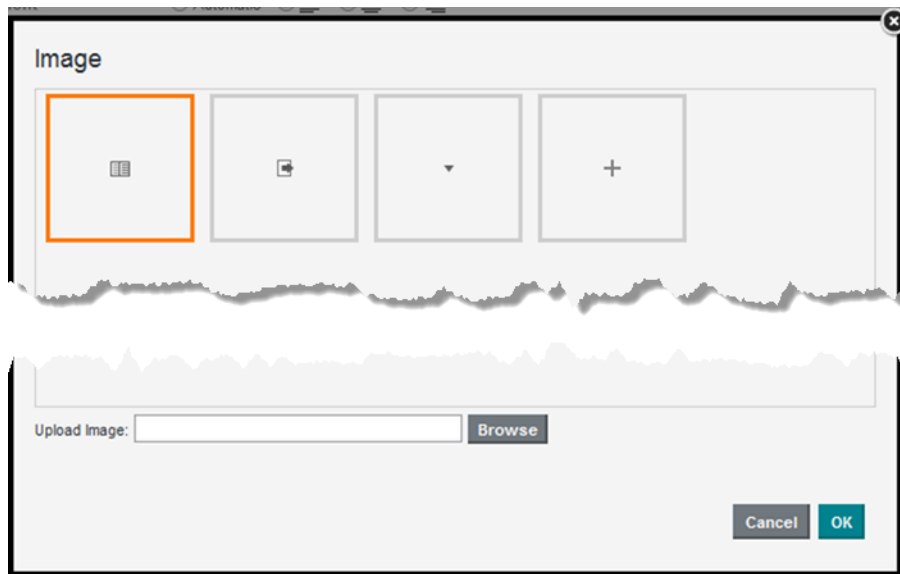
To display a clickable icon:

- a. Click **Icons**. This is the default.

When you display an icon in the action column, the column heading is empty.

- b. To display an icon other than the default, click **Custom icon**.
- c. To select the custom icon, click **Select Icon**.

The image selection dialog displays. The dialog lists any images that have been uploaded for any **Results Table** component in the current project.

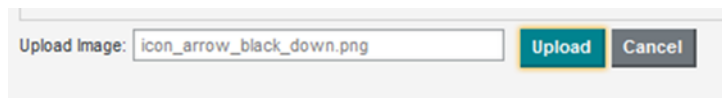


The list does not include images from other component types or projects.

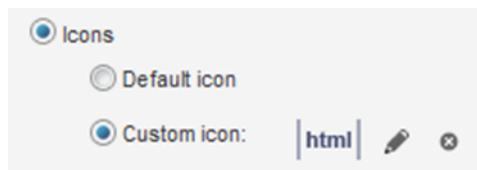
- d. To search for and select a new image file to upload, click **Browse**.

Studio supports the .gif, .jpeg/.jpg, .bmp, and .png file types.

- e. After you select the file, to clear the current selection, click **Cancel**.



- f. To add the image to the list, click **Upload**.
- g. To select an image to use for the action, click the image in the list, then click **OK**.  
If the image is too large, Studio crops it.
- h. To change to a different image, click the edit icon next to the selected image.



- i. To remove the selected image, click the delete icon.

### Configuring conditions for enabling Results Table actions

For most **Results Table** actions, including column actions, action columns, and the **Actions** menu, you can set up a condition to determine whether the action is enabled.

Note that for the **Actions** menu, you can only configure conditions for the **Refinement** and **Pass Parameters** actions. For action columns, you can only configure conditions for the **Refinement** and **Hyperlink** actions. **Print**, **Export**, **Compare**, and **Details** actions never have conditions.



For column values, the **Action Condition** fields display in the **Column Actions** section of the column configuration dialog.

---

Action Condition    If **Select Attribute** is  then

---

For **Actions** menu and action column actions, the configuration dialog contains a separate **Action Condition** section.

---

▼ ACTION CONDITION

---

Action Condition    If **Select Attribute** is  then

---

To configure the condition to control whether the action is available:

1. To select the attribute for the condition, click **Select Attribute**. On the attribute selection dialog, click the attribute to use, then click **Apply**.

You can select any of the columns in the table.

2. From the **is** list, select the comparison operator to use for the condition.

The options are:

Option	Description
equal to	Indicates that the condition applies if the column value is equal to the comparison value.
not equal to	Indicates that the condition applies if the column value is not equal to the comparison value.
null	Indicates that the condition applies if the column value is null.
not null	Indicates that the condition applies if the column value is a value other than null.

3. For the equal to and not equal to comparisons, in the value field, type the value to use for the comparison.
4. From the **then** list, select whether to enable or disable the action if the condition is met.

When an action has an associated condition, then on the end user view:

Action Type	Effect of Action Condition
Column values	For rows that should not have the action, the column value is not hyperlinked.
Action columns	For rows that should not have the action, the action column is empty. It does not display the icon or hyperlink.

Action Type	Effect of Action Condition
Actions menu	<p>For <b>Actions</b> menu options, users must first select at least one row to perform the action on.</p> <ul style="list-style-type: none"><li>• If all of the selected rows should not have the action, the action is disabled (grayed out) in the <b>Actions</b> menu.</li><li>• If at least one of the selected rows can have the action, the action is enabled.</li></ul>

## Using Results Table values to refine the data

You can use the values in a **Results Table** to refine the data.

The table can include options to:

Option	Description
Refine by a specific attribute value	For a record list table, the displayed values may be hyperlinked to allow you to refine by that value. When you click the hyperlink, the data is refined to only include records with that attribute value.
Refine by multiple attributes in the same row	A record list table can also include a column with a link to refine by selected attribute values in the row.
Refine by attribute values from multiple rows	<p>The <b>Actions</b> menu can also include options to refine by selected attribute values from the selected rows.</p> <p>To use this option, you must first check at least one row. You can select multiple rows across different pages.</p> <p>To deselect all of the selected rows, click delete icon next to the number of selections.</p>

## Navigating to other URLs from a Results Table

A **Results Table** can include hyperlinks to other URLs. The hyperlink can pass parameters in the form of attribute values from a row or rows.

For individual rows, a column can contain a hyperlink to another URL. The URL can include attribute values for that record as part of the URL or as query parameters for the URL.

The **Actions** menu can also contain options to send attribute values from selected rows as POST parameters.

## Exporting data from a Results Table

**Results Table** supports exporting data to HDFS or the desktop.

The **Actions** > **Export** option allows you to export all of the table columns as an Avro or CSV file to either HDFS or the desktop. See [Exporting data from a component](#) and [Exporting project data](#) for details.

Studio provides chart components that you can use to visualize data and the distribution of data values. The chart components include the Chart, Box Plot, Histogram Plot, and the Parallel Coordinates Plot.

#### Chart

The **Chart** component displays a graphical chart based on the project data. It supports several sub-types and includes options for selecting the specific data to display.

#### Histogram Plot

A **Histogram Plot** is a basic visualization for showing the distribution of values for a single metric.

#### Box Plot

A **Box Plot** provides a capsule summary of metric values for each value of a selected String attribute.

#### Parallel Coordinates Plot

A **Parallel Coordinates Plot** displays the values of several metrics for each value of a single grouping dimension.

## Chart

The **Chart** component displays a graphical chart based on the project data. It supports several sub-types and includes options for selecting the specific data to display.

The chart sub-types are:

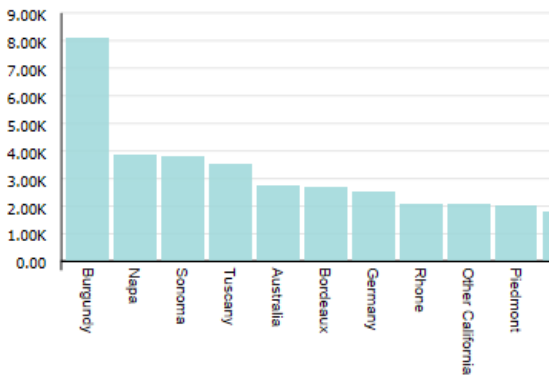
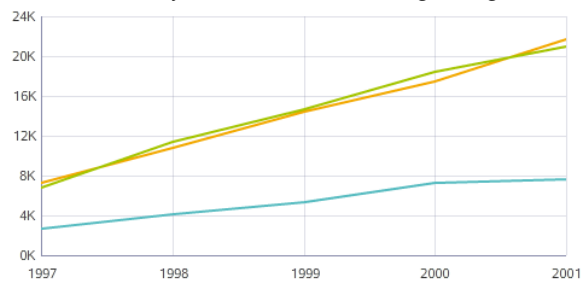
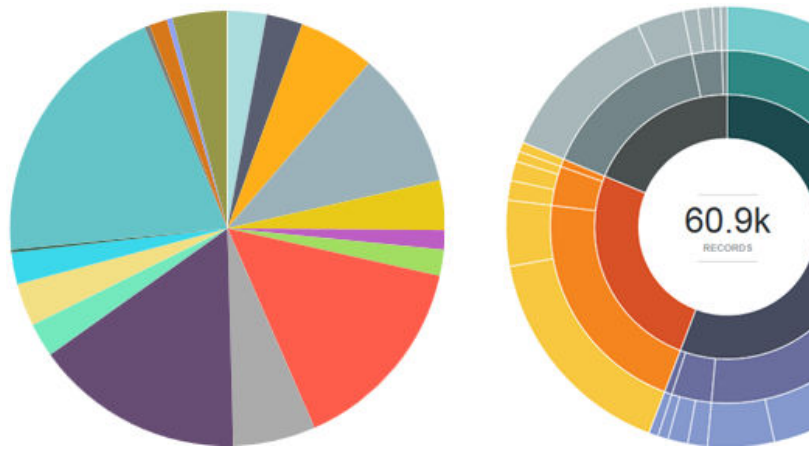
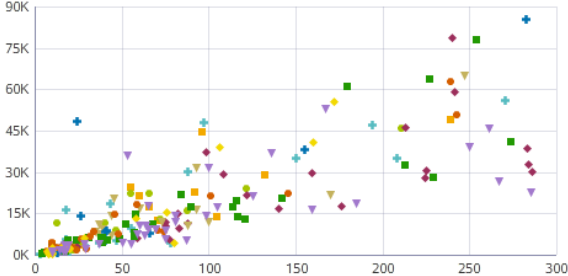
Chart Type	Description
<p><a href="#">Bar chart</a></p>	<p>Bar charts show one or more metric values aggregated across a group dimension. They are useful for precise comparisons of one or more values.</p> 
<p><a href="#">Line / Area chart</a></p>	<p>Line charts show one or more metric values aggregated across a group dimension. They are useful for showing changes or trends.</p> 
<p><a href="#">Pie / Sunburst chart</a></p>	<p>Pie charts show a single metric aggregated across a group dimension. They are useful for showing how each value contributes towards a total. Sunburst charts show concentric rings of metric values, in order to display contribution towards a total while also showing relationships between hierarchical data.</p> 

Chart Type	Description
Scatter chart	<p>Scatter charts display data points, with each point representing a dimension value. Points can also be aggregated into bins in a binned scatterplot, or measured against a third metric and displayed as scaled bubbles in a bubble chart. These charts are useful for showing correlations between metrics.</p> 
Bar-Line chart	<p>Bar-Line charts show two or more metric values aggregated across a group dimension. They are useful for showing quantity alongside changes in trends over time.</p> 
Thematic Map	<p>The Thematic Map shows geographic data.</p> 

**Bar chart**

Bar charts show one or more metric values aggregated across a group dimension. They are useful for precise comparisons of one or more values.

**Line / Area chart**

Line charts show one or more metric values aggregated across a group dimension. They are useful for showing changes or trends.

### [Pie / Sunburst chart](#)

Pie charts show a single metric aggregated across a group dimension. They are useful for showing how each value contributes towards a total. Sunburst charts show concentric rings of metric values, in order to display contribution towards a total while also showing relationships between increasingly finer-grained metrics.

### [Scatter chart](#)

Scatter charts display data points, with each point representing a dimension value. Points can also be aggregated into bins in a binned scatterplot, or measured against a third metric and displayed as scaled bubbles in a bubble chart. These charts are useful for showing correlations between metrics.

### [Bar-Line chart](#)

Bar-Line charts show two metric values aggregated across a group dimension. They are useful for showing quantity alongside changes in trends over time.

### [Thematic Map](#)

The **Thematic Map** component shows the data graphically and allows you to refine the data by regions including country, region, county, etc. You can refine data by drilling down into more granular regions in a thematic map and see the results associated with a sub-region.

### [Configuring a Chart](#)

Studio creates a new **Chart** component based on a set of default values. You can re-configure the defaults with different chart data and chart display options. This is often necessary to change the X and Y-axis settings.

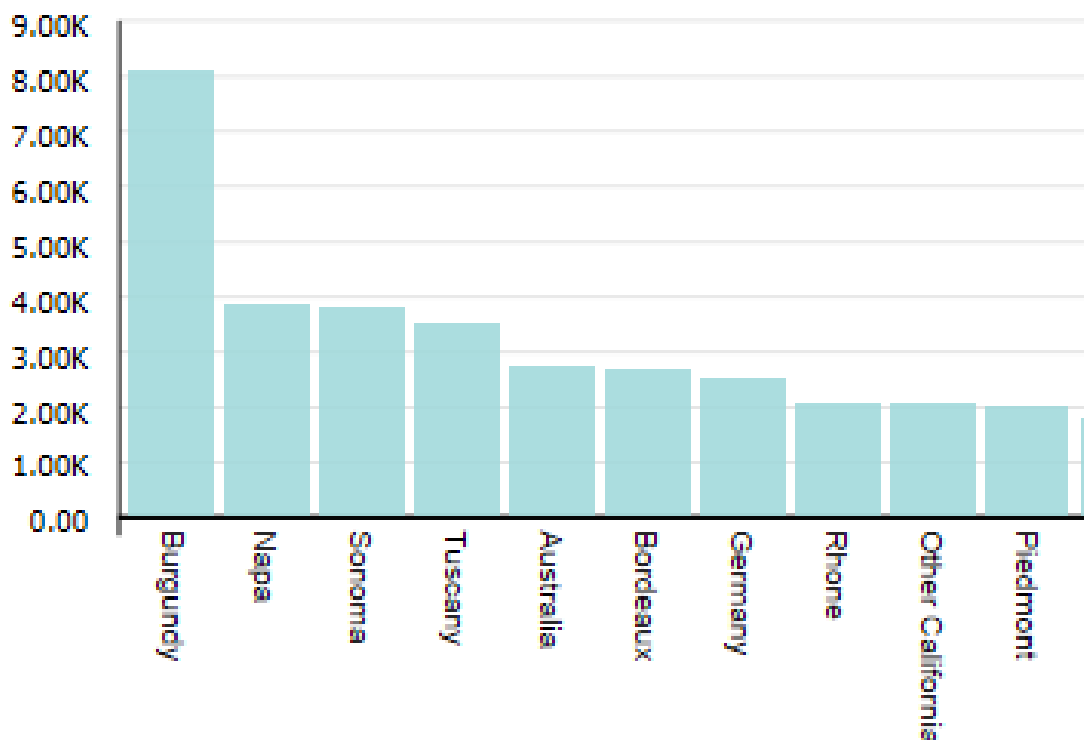
### [Saving a snapshot of a chart](#)

You can create a snapshot of a displayed Chart component. The snapshot includes any current refinements to the chart data.

## Bar chart

Bar charts show one or more metric values aggregated across a group dimension. They are useful for precise comparisons of one or more values.

Bar charts can display multiple metrics at the same time. Bar charts support both group dimensions (category axis) and series (color) dimensions. They can display either vertically (the default) or horizontally.



Bar charts feature the following variants:

- **Bar** — A basic bar chart.
- **Clustered bars** — A bar chart that displays different metrics alongside each other for comparison. For example, you might want to plot review scores and production cost across different car models.
- **Stacked bars** — A bar chart that displays multiple values atop each other to show how they contribute towards a varying total. For example, you might want to plot total sales for three different regions across each fiscal quarter.
- **Percentage stacked bars** — A bar chart that displays multiple values atop each other to show percentages of a total by metric. All bars are scaled to the same height (100%) to emphasize the difference in percentages.

#### About using a Bar chart

Mousing over a bar displays the X and Y axis values for that data point. Highlighting a color in the legend highlights the corresponding bar.

You can refine displayed data by clicking a single bar, or by clicking and dragging to select a set of bars, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple bars, then click **Apply refinements** in the selection box when finished. You can also select or multi-select values directly from the X-axis.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

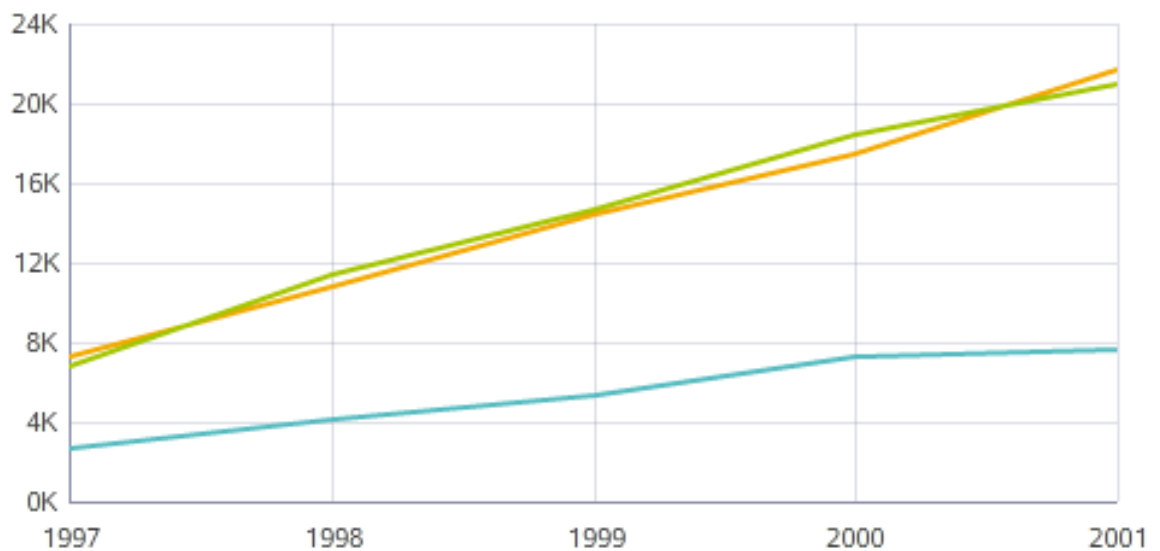
### About configuring a Bar chart

For variants that show multiple bars, you can either plot multiple Y-axis attributes (such as total sales and sales by region A, B, and C) or you can plot a single Color attribute that is automatically rendered as a separate bar for each attribute value. You cannot do both.

For detailed configuration steps, see [Configuring a Chart](#).

## Line / Area chart

Line charts show one or more metric values aggregated across a group dimension. They are useful for showing changes or trends.



Line charts can display multiple metrics as separate lines. They feature the following variants:

- **Line** — A basic line chart.
- **Area** — A line chart that fills in the area under the line, making it suitable for visually representing volume.
- **Colored lines** — A line chart that displays multiple metrics as different colored lines. This type of chart is generally preferable to an area chart if comparing metrics that do not contribute towards some total, or when showing a large number of metrics.
- **Stacked area** — An area chart that displays multiple metrics as filled areas. This type of chart makes it easier to see how multiple values contribute towards a cumulative total.
- **Percentage stacked area** — An area chart that displays multiple values atop each other to show percentages of a total by metric. The top of the chart represents 100%, so the visualization includes no whitespace unless a given X-axis value has no corresponding data.



### About using a Line / Area chart

Mousing over a point on the line displays the X and Y axis values for that data point. Highlighting a color in the legend highlights the corresponding line or area.

You can refine displayed data by clicking a single point, or by clicking and dragging to select a set of points, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple points, then click **Apply refinements** in the selection box when finished. You can also select or multi-select values directly from the X-axis.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

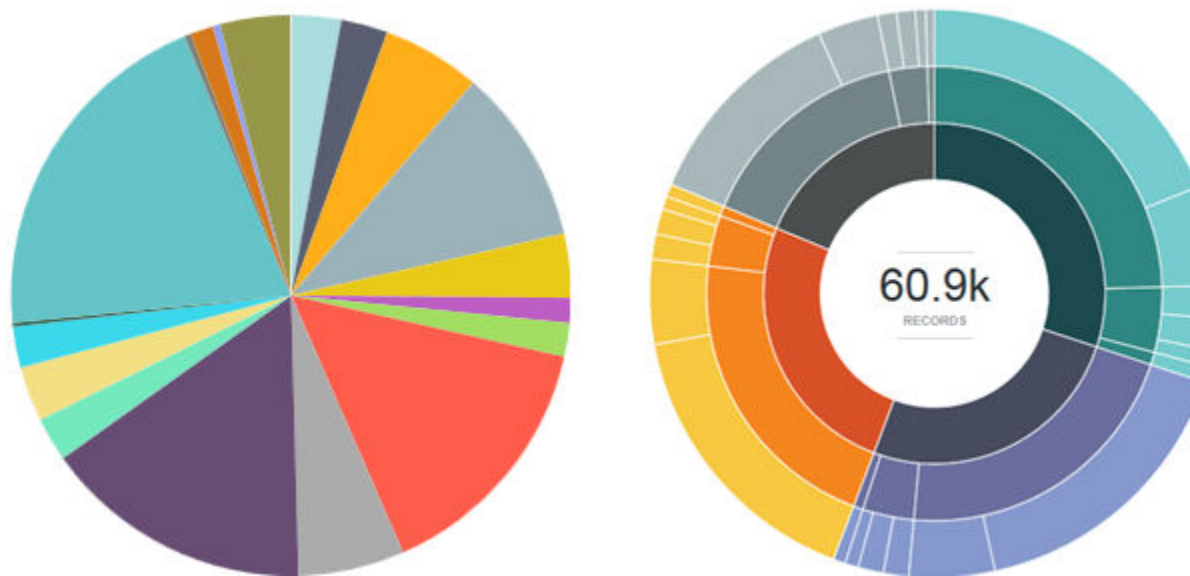
### About configuring a Line / Area chart

Selecting a Color metric for a Line or Area chart automatically changes the sub-type to Stacked area. Switching to a Line or Area chart with a Color metric selected instead displays the metric as a Trellis.

For detailed configuration steps, see [Configuring a Chart](#).

## Pie / Sunburst chart

Pie charts show a single metric aggregated across a group dimension. They are useful for showing how each value contributes towards a total. Sunburst charts show concentric rings of metric values, in order to display contribution towards a total while also showing relationships between increasingly finer-grained metrics.



Metric values display as pie wedges, with each wedge sized proportionally to the relative size of the value. There are two variations of pie chart:

- **Pie** — A basic pie chart.
- **Sunburst** — A multilevel pie chart hierarchical data is displayed from an inside ring outwards, with another selected metric used as the unit of measurement for rendering segments. For example, you might configure a sunburst to use Total Sales as the scale metric, and organize the rings with Fiscal Quarter as the inner

ring, territories as the next ring, and companies as the third ring. A sunburst can contain up to five rings.

### About using a Pie / Sunburst chart

Mousing over a wedge or ring segment displays the corresponding metric value as well as its percent of the total. Mousing over any outer Ring values in a Sunburst chart also highlights the selected value in other segments of the chart for easier visibility. Highlighting a color in the legend highlights the corresponding wedge or segment.

You can also search for values in a Sunburst chart by using the search box above the color legend. This allows you to refine the visualization by a metric value that might otherwise be displayed only as part of the "Others" group.

You can refine displayed data by clicking a segment. To select multiple segments, hold the CTRL key and click each one, then click **Apply refinements** in the selection box when finished. You can also select refinements in the same manner from the color legend.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

### About configuring a Pie / Sunburst chart

A Sunburst chart can be configured to show colors one of two ways:

- **One hue per ring** — Each ring uses a different base color, making it easier to distinguish which level a data point belongs to (inner ring, second ring, third ring, etc).
- **Hierarchical (fan gradient)** — Each inner ring value uses a different base color, making it easier to distinguish which data is associated with a given root value in the hierarchy. This may also display data multiple times in the legend if an outer ring value is present for multiple inner ring values. For this reason, when displaying hierarchical data try to choose an inner data metric where the values each have non-overlapping data for outer ring values.

Converting a Sunburst chart back to a pie chart renders the chart based on the innermost ring. If a second Rings metric is present, it is used to display a Trellised set of charts. Any additional metrics are removed from the configuration.

---

---

**Note:** You cannot specify multi-assign attributes for a Sunburst chart.

---

---

When configuring the chart, note that Pie charts use an independent metric (Slice size) and render dependent values from one configured Color metric. Sunburst charts use one dependent metric (Segment size) and render one or more independent metrics (the Rings). The Segment size metric must be both numeric and countable — scaling by total sales works, but attempting to scale by median sale value does not create an accurate visualization. Keep this in mind if you create your own metrics and want to use them in a Sunburst chart.

For detailed configuration steps, see [Configuring a Chart](#).

## Scatter chart

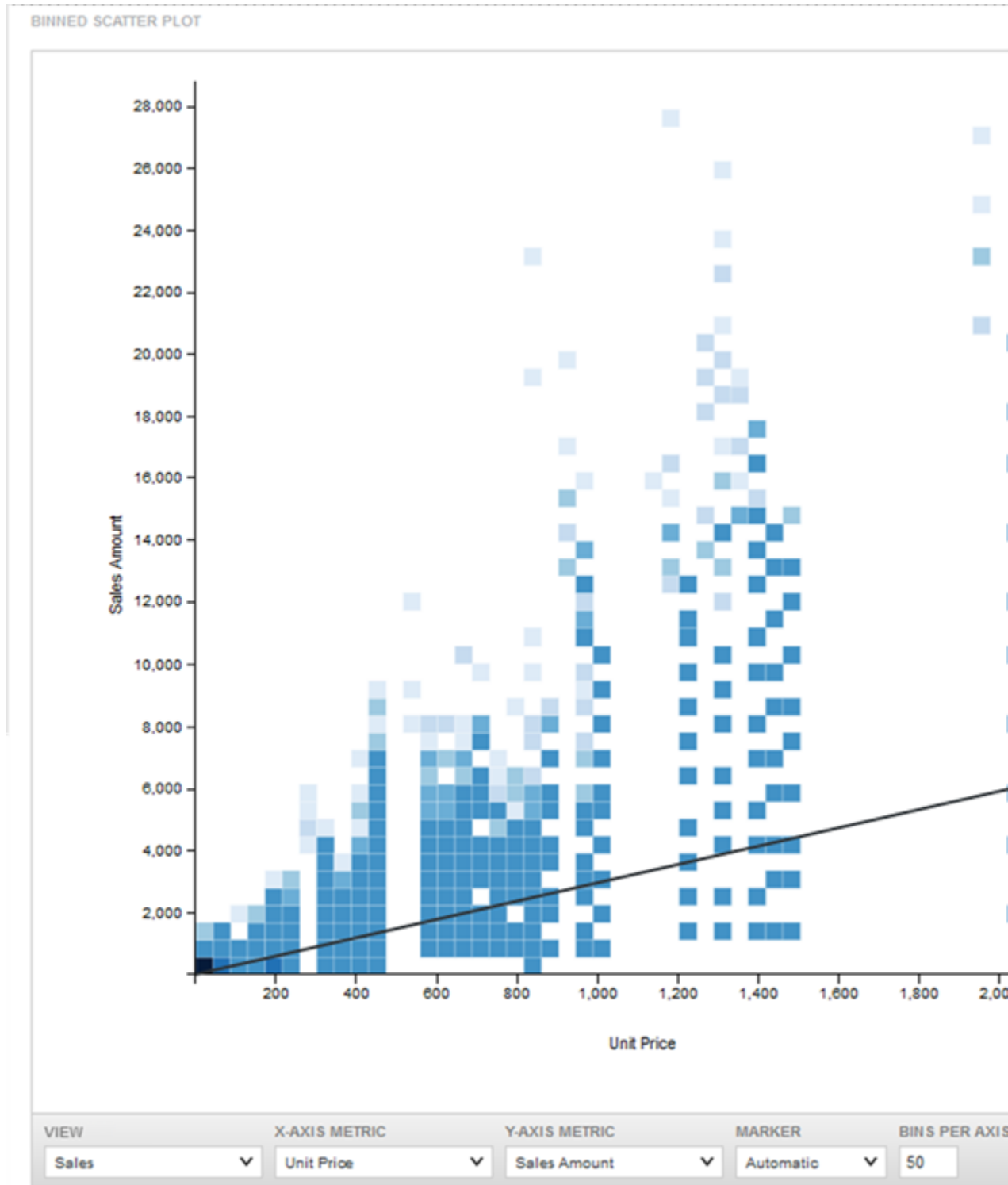
Scatter charts display data points, with each point representing a dimension value. Points can also be aggregated into bins in a binned scatterplot, or measured against a

third metric and displayed as scaled bubbles in a bubble chart. These charts are useful for showing correlations between metrics.

A Scatter chart can show a line of best fit if there is any clear trend in the distribution of data points.

As an example of binning, one bin may contain all points where both the X and Y values are between 0 and 50. The bin to the right contains all points where X is between 50 and 100, and Y is still between 0 and 50. By default, scatter charts automatically switch from individual points to bins when there are more than 1000 data points in the chart.

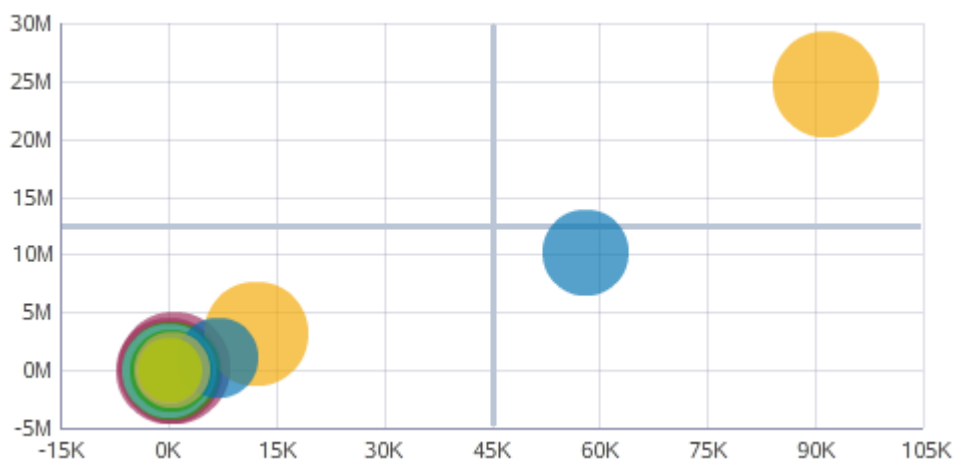
Bins are shaded to indicate the relative number of points in that bin: more points in a bin means darker shading. For example:



Scatter charts feature the following variants:

- **Scatter** — A basic Scatter chart displays points in either binned or unbinned form.
- **Grouped Scatter** — Specifying a Color metric renders a Scatter chart as a Grouped Scatter chart. Records are grouped based on their values in the Color metric, with a single point shown per Color metric value.

- **Bubble** — Specifying both Color and Size metrics renders a Scatter chart as a Bubble chart. Like a Grouped Scatter plot, each Color metric value displays a single point on the chart. The size of the point is determined by the Size metric value:



- **Scatterplot Matrix** — A collection of binned scatterplots and correlation values displaying the relationships between three or more metrics.

#### About using a Scatter chart

Mousing over a bin or bubble displays all of the corresponding metric values. Highlighting a color in the legend highlights the corresponding point.

You can refine displayed data by clicking a single point or bubble, or by clicking and dragging to select a set of points, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple points, then click **Apply refinements** in the selection box when finished. You can also select refinements in the same manner from the color legend or from an axis.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

#### About configuring a Scatter chart

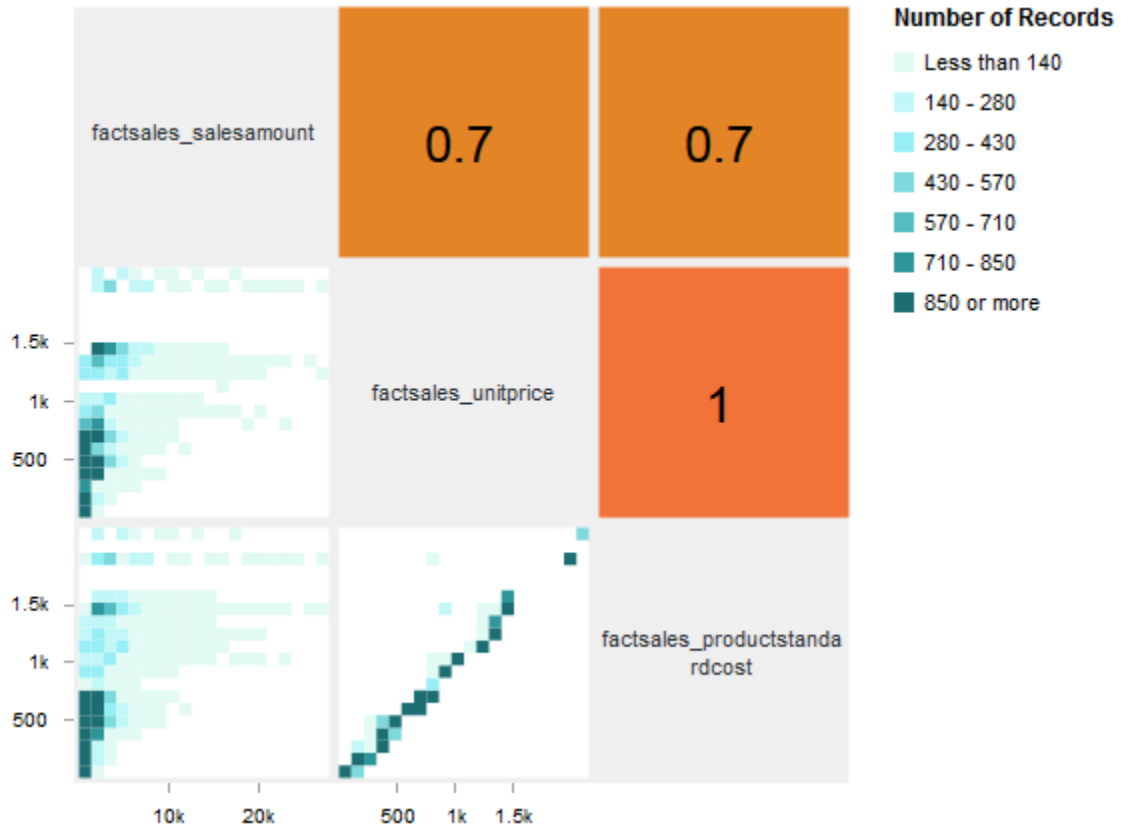
The Scatter chart automatically sets a subtype based on the configured metrics. Converting a Bubble chart to a Grouped Scatter chart clears the Size metric selection. Similarly, converting a Grouped Scatter chart to a Scatter chart clears the Color metric.

You can set whether binning behavior is automatic or manual in the Display Options. You can also set whether the chart renders a line of best fit.

For detailed configuration steps, see [Configuring a Chart](#).

#### Scatterplot Matrix

The **Scatterplot Matrix** is a Scatter chart sub-type that plots multiple matrices for comparing attributes against each other. You can also add an optional Color attribute for grouping.



**About using a Scatterplot Matrix**

To find the plot or correlation for any two attributes, look at where their rows and columns intersect. For example, for three attributes the matrix is organized as follows:

Attribute 1 ▼	Correlation of Attribute 1 to Attribute 2	Correlation of Attribute 1 to Attribute 3
Attribute 2  Attribute 1	Attribute 2 ▼	Correlation of Attribute 2 to Attribute 3
Attribute 3  Attribute 1	Attribute 3  Attribute 2	Attribute 3 ▼

Mousing over a chart or a correlation value highlights the attributes being compared. For example, in the diagram above, mousing over the lower-left corner highlights the

"Attribute 1" and "Attribute 3" squares to make it clear what is being compared and rendered in the chart, and it also highlights the "Correlation of Attribute 1 to Attribute 3" value in the upper-right corner so you can see if the two attributes have closely correlated values, or are independent.

Mousing over a bin within one of the charts also shows the value ranges for the bin in a tooltip, while mousing over a correlation value displays the exact correlation. Clicking and dragging a range of values in any of the scatter plot visualizations adds two range filter refinements, one for each of the plotted attributes, to the **Selected Refinements** panel.

The **Number of Records** legend indicates the number of records in each bin based on the bin shade, so that you can interpret each of the individual scatterplots.

### About configuring a Scatterplot Matrix

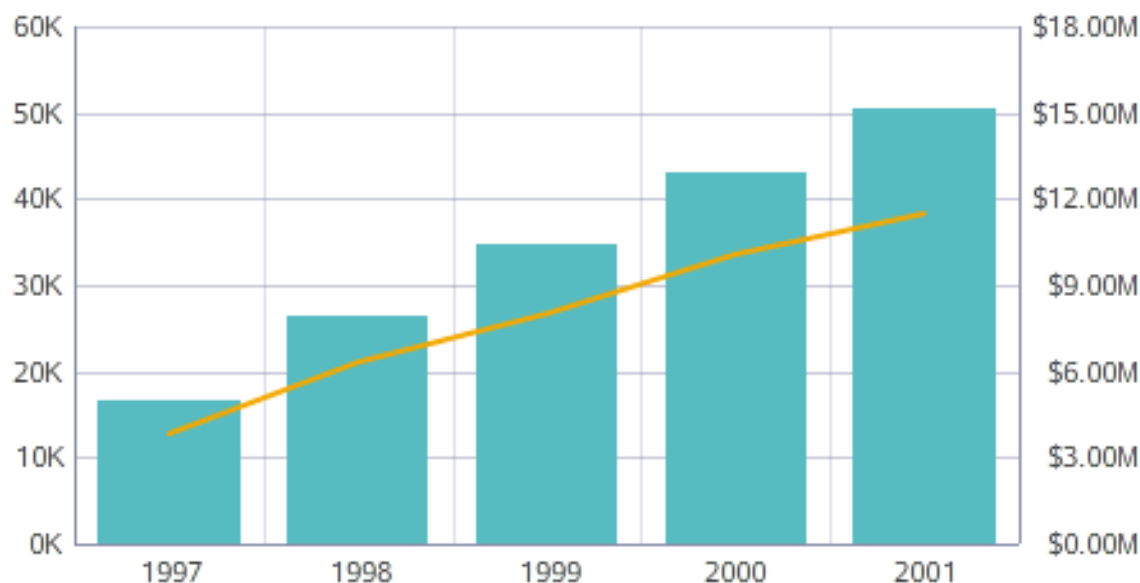
A Scatterplot Matrix requires at least three metric attributes to render, though you can add more. Each additional attribute adds a row and a column to the layout.

Selecting a Color attribute color codes bins based on the most common value of that attribute.

## Bar-Line chart

Bar-Line charts show two metric values aggregated across a group dimension. They are useful for showing quantity alongside changes in trends over time.

The first Y-axis metric displays as a bar, and the second displays as a line. The chart uses a dual-axis plot, where bars are plotted against the left vertical axis and the line is plotted against the right vertical axis.



### About using a Bar-Line chart

Mousing over a bar or a point on the line displays the X and Y axis values for that data point.

You can refine displayed data by clicking a single bar, or by clicking and dragging to select a section of the chart, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple bars, then click **Apply**

**refinements** in the selection box when finished. You can also select or multi-select values directly from the X-axis.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

### About configuring a Bar-Line chart

You cannot render only a line or only bars; the chart requires a metric for each axis and for the line in order to render. For detailed configuration steps, see [Configuring a Chart](#).

## Thematic Map

The **Thematic Map** component shows the data graphically and allows you to refine the data by regions including country, region, county, etc. You can refine data by drilling down into more granular regions in a thematic map and see the results associated with a sub-region.

The component can only be used if the data set contains at least one geocode attribute along with five related attributes that qualify the geocode attribute with additional region information.

The naming syntax for the geocode attribute varies slightly depending on whether it is a latitude/longitude geocode or an IP address/address geocode. This naming difference only affects the main geocode attribute but not the five related attributes.

Here is the minimum set of attributes required to render a **Thematic Map** component:

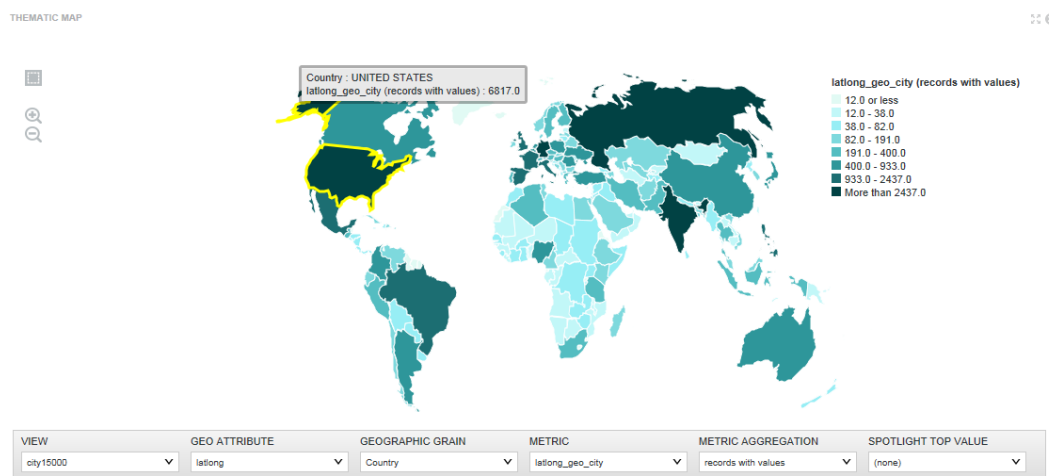
- `<attributeName>` for geocode attributes or `<attributeName>_geo_geocode` for IP address or address attributes
- `<attributeName>_geo_country`
- `<attributeName>_geo_region`
- `<attributeName>_geo_subregion`
- `<attributeName>_geo_regionid` — This attribute must be present in the data set but it is not exposed in the tabular view of **Explore**.
- `<attributeName>_geo_subregionid` — This attribute must be present in the data set but it is not exposed in the tabular view of **Explore**.

The five related attributes are produced in any of the following ways:

- The data enrichment process creates the geocode attributes automatically during data processing. See the *Administrator's Guide* for information about enabling auto-enrichment using the `bdd.enableEnrichments` setting.
- You can create the geocode attributes manually by writing a custom transformation script on the **Transform** page of Studio.
- You can create the geocode attributes using the Geohierarchy Tagger transform.

For example, if you create the region attributes manually and they are based on an IP attribute, you call the `geotagIPAddressGetGeocode` transform function to create a geocode attribute based on an IP attribute. (See the *Extensions Guide* for details about functions.)





## Using the Thematic Map

You can create a new **Thematic Map** component in either the scratchpad or the **Discover** page.

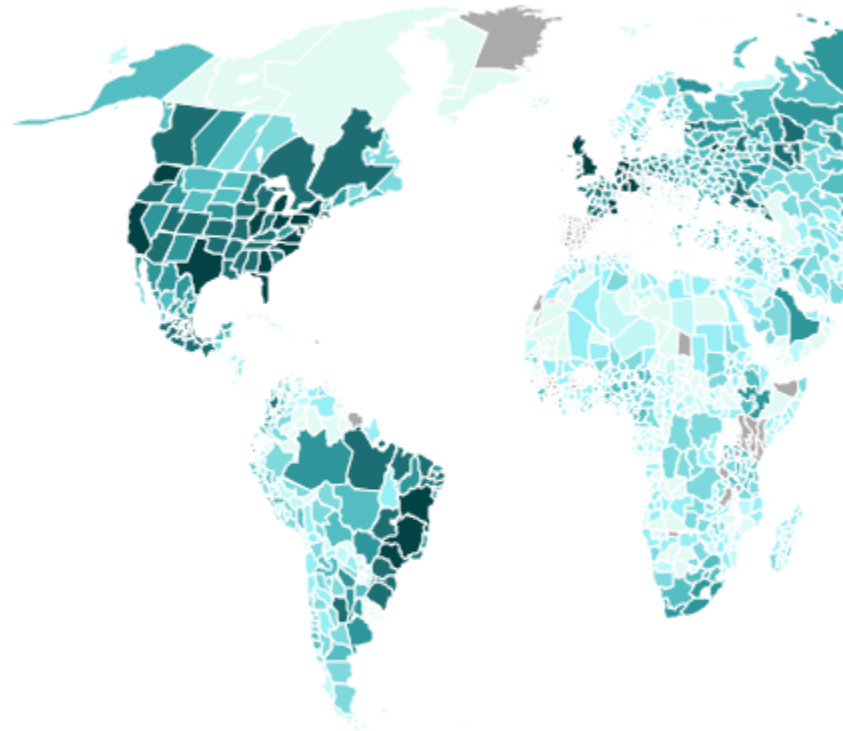
### Choosing the geocode attribute for the thematic map

From the **Geo Attribute** list, you can select an attribute name on which to aggregate records in the thematic map.

### Choosing a geographic grain region for the thematic map

From the **Geographic Grain** list, you can select a region name on which to aggregate records in the thematic map. For example, this thematic map aggregates records based on State rather than Country:

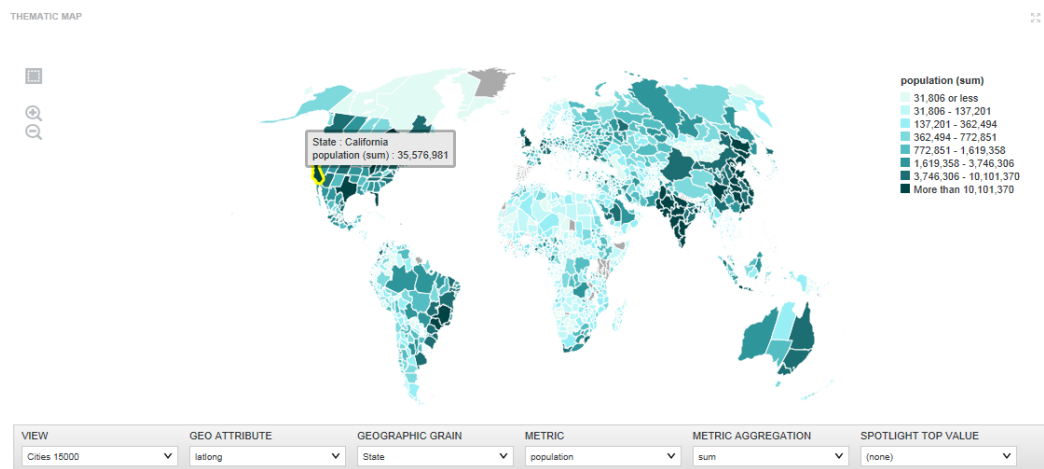
## THEMATIC MAP



VIEW	GEO ATTRIBUTE	GEOGRAPHIC GRAIN	METRIC
Cities 15000 ▼	latlong ▼	State ▼	Record Count

### Choosing a metric and metric aggregation

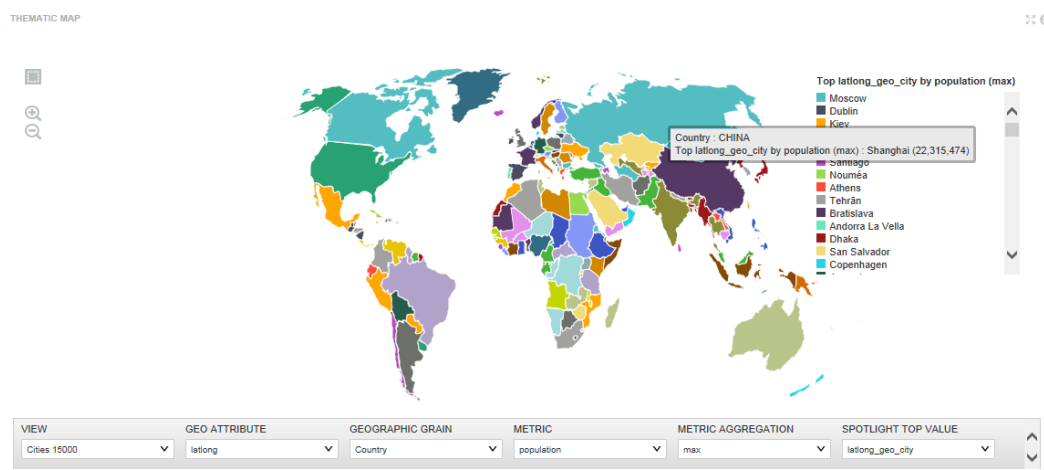
From the **Metric** list, you can select a metric name on which to aggregate records in the thematic map. Optionally, from the **Metric Aggregation** list, you can select a method to aggregate the value of the metric you selected in the **Metric** option. For example, this thematic map first aggregates records based on State and then sums the records for the state based on population:



### Spotlighting values on the map

The **Spotlight Top Value** list colors regions according to the largest value of the metric selected in the **Metric** option.

For example, this map shows the city in each country that has the maximum population. The grain is set to country, and the metric and metric aggregation instruct the component to calculate records based on maximum population, and lastly, the **Spotlight Top Value** groups the records by city. So you can see that in China, the city of Shanghai has the largest population:



### Zooming in/out and redrawing the map

You can use Zoom In and Zoom Out features on the left edge of the component to manipulate the map.

You can multi-select regions using the Marque drawing tool or by using ctrl+click to select regions one at a time. You can also drag the thematic map to different locations on the page and then use the tools to zoom to finer views of the data.

### Using custom views with a Thematic Map

The Thematic Map chart does not render correctly if it is based on a custom view. This is because two region attributes that are required to render the Thematic Map are not

exposed in the base view and therefore the attributes are not copied into the custom view. To work around this issue, manually copy the following two attributes into the custom view EQL:

```
ARB("dimgeography_latlong_geo_regionid") AS "dimgeography_latlong_geo_regionid",
ARB("dimgeography_latlong_geo_subregionid") AS
"dimgeography_latlong_geo_subregionid",
```

## Configuring the Thematic Map connection to Oracle MapViewer

The **Thematic Map** component uses Oracle's MapViewer (11.1.1.7.2 or greater) to display the map. Studio includes framework settings to configure the connection to MapViewer.

By default, Studio is configured to use the public instance of MapViewer. If you are using that instance, and your browser has access to the Internet, then you should not need to make any configuration changes.

Studio includes the following framework settings related to the MapViewer connection. If you are using your own instance of MapViewer, or if your Web browser does not have Internet access, then you will need to change these settings. For details on configuring framework settings in Studio, see the *Administrator's Guide*.

Framework Setting	Description
df.mapViewer	<p>The URL of the MapViewer instance. By default, this is the URL of the public instance of MapViewer.</p> <p>If you are using your own internal instance of MapViewer, then you must update this setting to connect to your MapViewer instance.</p>

## Configuring a Chart

Studio creates a new **Chart** component based on a set of default values. You can re-configure the defaults with different chart data and chart display options. This is often necessary to change the X and Y-axis settings.

The default **Chart** component configuration starts with:

- A vertical Bar chart
- The base view of the default project data set
- For the chart metrics, the first 20 available metrics in the view.

Each metric uses its default aggregation method. The available metrics also include the Record Count system metric, which is selected by default.

- For the chart group (category axis) and series (color) dimensions, allows end users to select from the first 20 available dimensions in the view.

By default, the category axis displays the first dimension. The chart is initially displayed with no color dimension selected or trellis dimension selected. You can then select a color dimension and trellis dimension if desired.

To configure a Chart component:

1. In the **Discover** page of your project, locate the chart you want to configure, and click the pencil icon to display the **Chart Settings** panel.

2. From the **Data View** list, select a data set view to associate with the chart.

3. From the **Chart Type** list, select a chart type.

- **Bar** - Bar charts can display multiple metrics at the same time. Bar charts support both group dimensions (category axis) and series (color) dimensions.
- **Line** - Line charts can display multiple metrics at the same time. Line charts support both group dimensions (category axis) and series (color) dimensions.
- **Pie** - Pie charts display one metric at a time and only support group dimensions (wedge colors). Sunburst charts display multiple metrics and only support group dimensions (ring segments).
- **Scatter** - Scatter charts both support X-axis and Y-axis metrics. Only one X-axis and Y-axis metric can be used at a time. Scatter charts support color and detail dimensions.
- **Bar-Line** - Bar-Line charts can display multiple bar and line metrics at the same time. Bar-Line charts only support group dimensions (category axis). They do not support series dimensions.

a. If applicable, select a chart sub-type.

Charts have the following sub-types:

- **Bar** - Bars, Clustered Bars, Stacked Bars, Percentage Stacked Bars.
- **Line** - Line, Area, Colored Lines, Stacked Area, Percentage Stacked Area.
- **Pie** - Pie, Sunburst.
- **Scatter** - Scatter, Grouped Scatter, Bubble.
- **Bar-Line** - No subtypes.

Many sub-types require a Color attribute for rendering, and Bubble charts require a Size attribute.

The chart type you select determines the subsequent configuration options. Required zones display in the attribute list with a solid outline if they are not configured.

You can drag configured attributes between zones as long as they are valid for the new location, and any aggregation, sorting, or formatting is preserved as part of the transition. Dragging an attribute over another attribute in a zone will swap them.

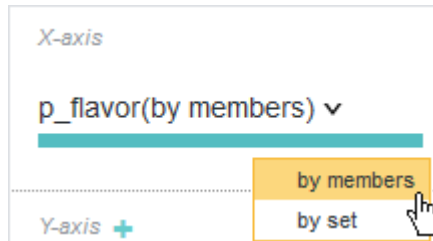
4. Configure and format the **X-axis** or **Slice / Segment size** value:

Pie / Sunburst charts scale all wedges or ring segments based on the configured value.

a. Type an attribute name or select one from the dropdown to set the attribute to plot on the horizontal axis.

To remove an attribute, mouseover it and click the **X** in the corner of the zone. You can change attributes by removing the current selection and adding the new attribute.

- b. For attributes with multiple grouping options, select an option from the dropdown:



- **by members** - Default. Each individual value displays as a mark on the X-axis. Any records that include a value count towards the Y-axis value.
- **by set** - Each individual value and each unique combination of values displays as a mark on the X-axis. Any records with the value or with the specified combination of values count towards the Y-axis value.

For example, a wine with the flavors "Berry" and "Cherry" is included in both the "Berry" and "Cherry" Y-axis totals when displaying flavor by members. If the chart displays flavor by set, then the wine is included in the "Berry", "Cherry", and "Berry, Cherry" Y-axis totals.

- c. Mouseover the X-axis value and select **Options > Format** in the upper right corner:



- d. Specify X-axis **Title** text, font size, and **Labels** font size.
5. If available, configure X-axis sorting:
- a. Mouseover the X-axis zone and select **Options > Sort** in the upper right corner.

Some attribute types do not support sorting configuration.

- b. Configure **Ascending** or **Descending** sorting either by **Alphabetic/Sequential** values for the X-axis attribute, or you can sort based on a selected attribute.
- c. Configure the number of marks to display from the **Show up to: ... marks** dropdown.

This value is approximate. Studio adjusts the setting internally to ensure even breakpoints and a reasonable visualization.

- d. Optionally, display any results that don't fit on the first page of the visualization as a consolidated **Others** mark on the X-axis.

You can use the pagination controls to see additional pages if you do not display an **Others** value. This can help you decide whether consolidating the values is worthwhile, or if doing so hides too much useful information.

## 6. Configure and format the **Y-axis** or **Rings** value(s):

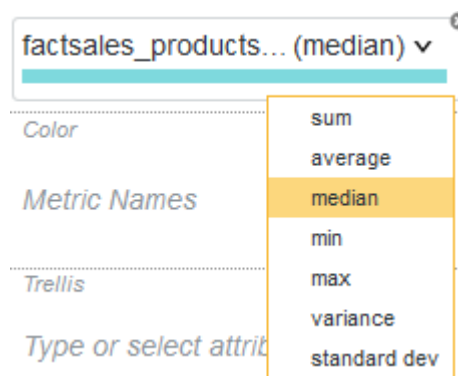
Pie charts instead use the **Color** metric for displaying dependent values. See Step 7.

For Bar-Line charts, repeat these steps for both the **Left Y-Axis (Bar)** and **Right Y-Axis (Line)**.

- a. Type an attribute name or select one from the dropdown to set the attribute to plot on the vertical axis.

To remove an attribute, mouseover it and click the **X** in the corner of the zone. You can change attributes by removing the current selection and adding a new attribute. Some chart types, such as Stacked Bar charts, support multiple Y-axis values.

- b. If required, select an aggregation method from the dropdown:



By default, aggregation is set to **sum**.

- c. Mouseover the **Y-axis** zone and click **Format** in the upper right corner.
- d. Specify Y-axis **Title** text, font size, and **Labels** tick mark number and size:
- e. Specify additional Y-axis **Scale**, **Range**, **Symbol**, and **Decimal** configuration.

By default, the Y-axis scales linearly. **Log** (logarithmic) scaling is useful for values with an extreme range of values, where relatively small values may be hard to differentiate from each other when compared to a few much larger values.

## 7. Configure and format the **Color** value.

- a. Select an attribute to plot as colors.

If you are plotting multiple Y-axis values (for example, plotting manufacturing price and shipping price as a single bar in a stacked Bar chart), the Y-axis values are automatically mapped to the color of the visualization. You cannot assign a separate **Color** attribute.

- b. Mouseover the **Color** zone and select **Options > Format** in the upper right corner.
- c. Select the color scheme from the **Scheme** dropdown.
- d. Enable or disable the Legend display using the **Show legends** check box.

- e. Specify Legend **Title** text, font size, and **Labels** font size.
- 8. If available, configure **Color** sorting:
  - a. Mouseover the **Color** zone and select **Options** > **Sort** in the upper right corner.
  - b. Configure **Ascending** or **Descending** sorting either by **Alphabetic/Sequential** values for the Color attribute, or you can sort based on a selected attribute.
  - c. Optionally, display the final color in the palette as a consolidated **Others** color.

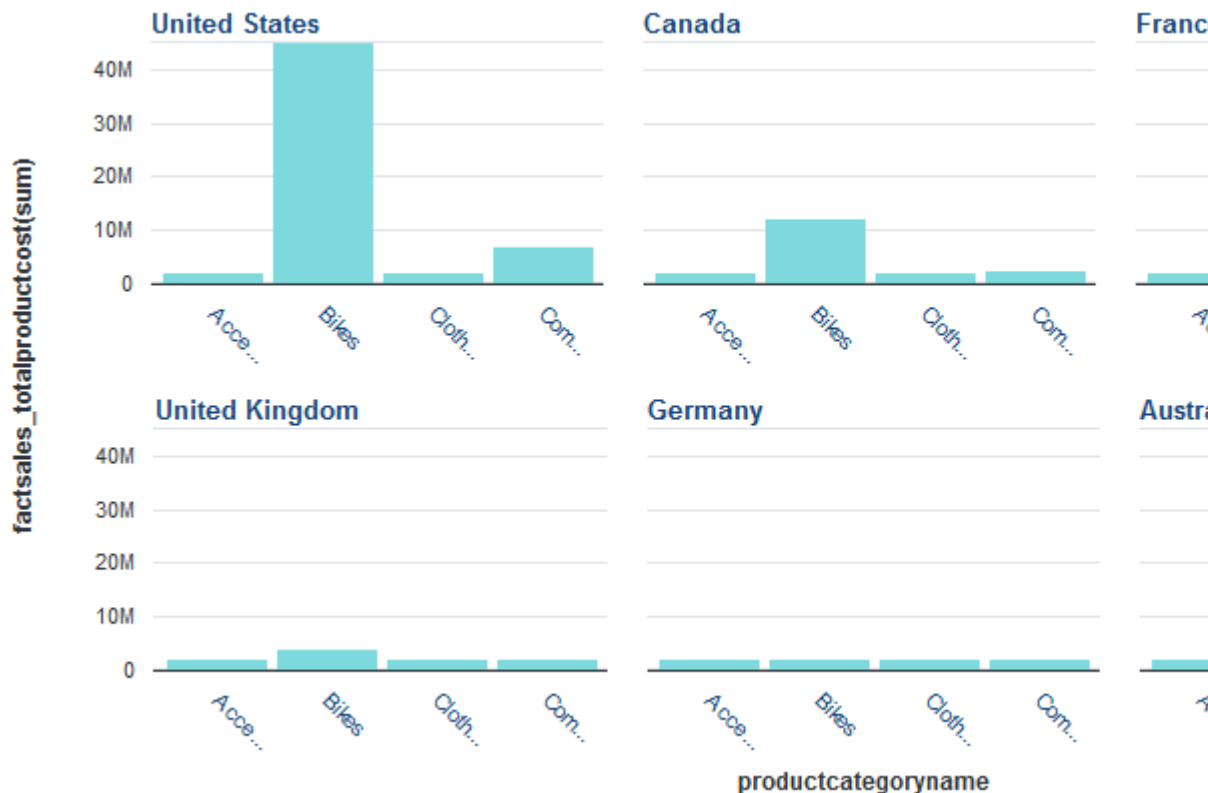
Pie charts always include an **Others** color.

- 9. For a Bubble chart, configure and format the **Size** value.

This value determines which attribute scales Bubble size. Specifying a **Size** value for any other type of Scatter chart converts it to a Bubble chart.

- 10. For Bar, Line, and Pie charts, optionally specify a **Trellis** attribute.

Specifying a trellis attribute splits the visualization into a set of smaller charts. For example, a Bar chart that plots product cost based on category can be displayed as a Trellis based on region, so that it shows separate charts for data from the United States, Canada, France, etc:



- a. Mouseover the **Trellis** zone and select **Sort** in the upper right corner.
- b. Configure **Ascending** or **Descending** sorting for the charts either by **Alphabetic/Sequential** values for the Trellis attribute, or sort based on another attribute.



## 11. Expand and configure the **Display Settings**:

- a. Set the **Chart width**, in pixels.
- b. Set the **Chart height**, in pixels.
- c. Enable or disable **Show chart title** to show or hide title text. If enabled, select **Automatic** or **Custom** text to display.
- d. For Bar charts, select between **Vertical Bars** and **Horizontal Bars**.
- e. For Sunburst charts, select a **Color pattern**.

Note that depending on configured data, the **Hierarchical (fan gradient)** option may display metric values multiple times in outer rings. For example, using Month as an inner ring and Day of the Week as an outer ring results in all 7 days of the week rendering 12 times (once per month).

- f. For Scatter charts, configure binning behavior:

With the **Automatic** option, the chart displays points if there are fewer than 1000 data points, and bins if there are more. You can adjust this threshold in the **Maximum number of points to show** field.

You can also configure the number of bins per axis to control how granular the display is.

- g. For Scatter charts, set the **Point opacity**.
- h. For Scatter charts, enable or disable **Show line of best fit** and **Show coefficient**.

**Show line of best fit** displays a line that approximates the linear relationship of the plotted points. **Show coefficient** displays the correlation of data with the approximated line when a user mouses over the line in the chart.

12. When you are finished, click **Save as default state**.

## Saving a snapshot of a chart

You can create a snapshot of a displayed Chart component. The snapshot includes any current refinements to the chart data.

To save a snapshot of a chart:

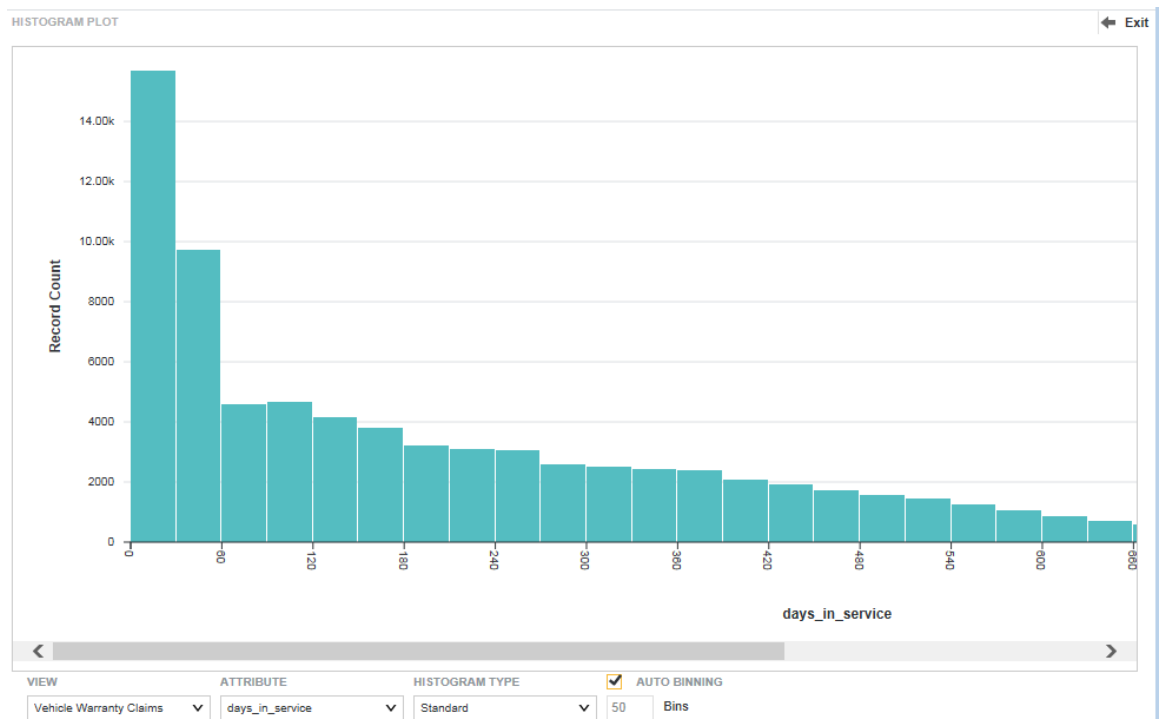
1. Click the camera icon in the left bar to expose the **Snapshots** pane.
2. Select **New Snapshot**.
3. On the **Add Snapshot** dialog:
  - a. Provide the name of the snapshot.
  - b. Provide an optional description of the chart snapshot.
  - c. Click **Save**.

The snapshot displays in the **Snapshots** pane. You can click to view the snapshot and also perform a number of addition actions from the **Actions** menu (edit, email, save to disk, or delete).

## Histogram Plot

A **Histogram Plot** is a basic visualization for showing the distribution of values for a single metric.

On the **Histogram Plot**, the X-axis contains bars with binned data reflecting ranges of attribute values. Values are left inclusive and right exclusive. For example, the first bin might reflect values between 0 and 29, the second bin values between 30 and 59, and so on. The Y-axis shows the number of records with values that fall within each range of values.



### About using a Histogram Plot

Mousing over the bar for a bin displays the range of values for the bin, as well as the record count for the bin.

You can refine displayed data by clicking a the bar for a single bin, or by clicking and dragging to select a set of bins, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple bins, then click **Apply refinements** in the selection box when finished. You can also select or multi-select values directly from the X-axis.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

#### Selecting Histogram Plot data and display options

You can select the data view, metric, histogram type, and number of bins when viewing a **Histogram Plot**.

## Selecting Histogram Plot data and display options

You can select the data view, metric, histogram type, and number of bins when viewing a **Histogram Plot**.

To select the **Histogram Plot** data and display options:

1. Select the view to use for the chart data from the **View** list at the bottom of the chart.
2. Select the metric for which to plot the value ranges on the X-axis from the **Attribute** list.
3. Select the type of histogram to display from the **Histogram Type** list:

Histogram Type	Description
<b>Cumulative</b>	The cumulative histogram adds up the number of records through each range of values. So each point in the histogram represents the total number of records with values up to, but not including, that range. For example: <ul style="list-style-type: none"> <li>• The bar for range 1-10 shows the number of records with a value between 1 and 9.</li> <li>• The bar for range 10-20 shows the number of records with a value between 1 and 19.</li> <li>• The bar for range 20-30 shows the number of records with a value between 1 and 29.</li> </ul>
<b>Relative</b>	The relative histogram shows the relative frequency for each range of attribute values. Relative histograms also include a kernel density line.
<b>Standard</b>	The standard histogram shows the actual number of records for each range of attribute values. Standard histograms also include a kernel density line.

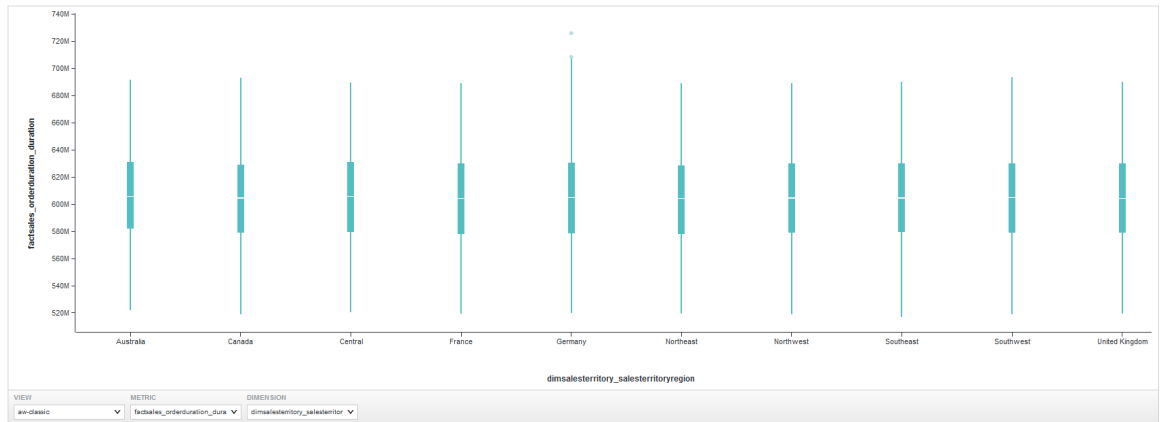
4. By default, the **Auto Binning** check box is selected, indicating that Studio automatically determines the number of bins to display. To set the number manually:
  - a. Deselect the **Auto Binning** check box.
  - b. In the **Number of Bins** field, type the number of bins to display on the X-axis. Click outside the field to update the visualization.

The larger the number of bins, the smaller the range of values in each bin.

For example, if you have 50 bins for 1000 data points, each bin represents a range of 20 (0-19, 20-39, etc.). If you have 100 bins, each bin represents a range of 10 (0-9, 10-19, etc.).

## Box Plot

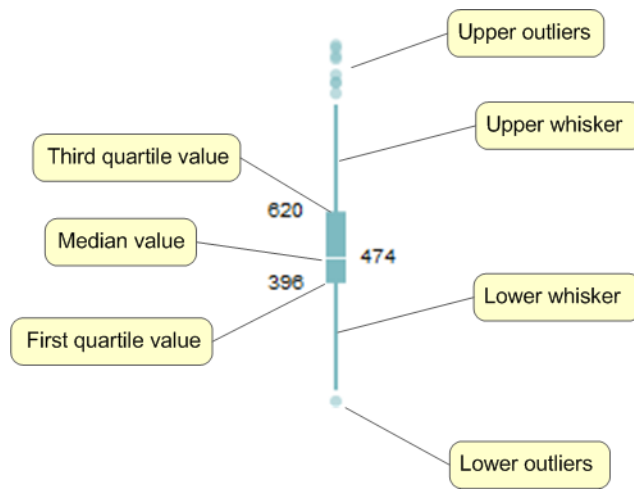
A **Box Plot** provides a capsule summary of metric values for each value of a selected String attribute.



For example, you can show a summary of the values for sales cycle duration plotted by region, as above.

**About using a Box Plot**

For each dimension value, the **Box Plot** displays a box showing:



- First and third quartile metric values for that dimension value
- Median metric value for that dimension value
- Upper and lower whisker lines.

To calculate the end points of the whisker lines, Big Data Discovery first gets the difference of the third and first quartile values, then multiplies it by 1.5.

For the lower whisker line, it then subtracts the resulting value from the first quartile value. The lower whisker line also cannot extend past the absolute minimum value.

For the upper whisker line, it adds the value to the third quartile value. The upper whisker line also cannot extend past the absolute maximum value.

- Any outlier values that fall above or below the whisker line end points. The chart always includes the single lowest outlier and single highest outlier in order to show the full range of values.

Mousing over a quartile displays the X-axis value and the quartile value range.

You can refine displayed data by clicking a single quartile on a plotted box, or by clicking and dragging to select a set of quartiles, then clicking a second time to apply the selection as a refinement. Alternately, hold the CTRL key and select multiple quartiles, then click **Apply refinements** in the selection box when finished.

This adds two refinements to the **Selected Refinements** panel:

- The X-axis attribute value(s).
- The cumulative range of values represented by the selected quartiles.

Selected refinements display in the Selected Refinements Panel and can be toggled to set them as negative refinements. See [About the Selected Refinements panel](#) for details.

### Selecting Box Plot data

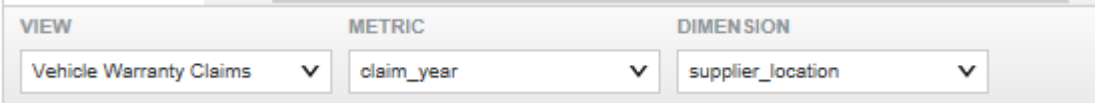
You can select the data to display when viewing a **Box Plot** chart.

## Selecting Box Plot data

You can select the data to display when viewing a **Box Plot** chart.

To select **Box Plot** data:

1. Select the view to use for the chart data from the **View** list at the bottom of the chart:



VIEW	METRIC	DIMENSION
Vehicle Warranty Claims ▼	claim_year ▼	supplier_location ▼

2. Select the metric to display on the Y-axis from the **Metric** list.

The axis automatically scales to show the full range of values, including outliers.

3. Select the String attribute to display on the X-axis from the **Dimension** list.

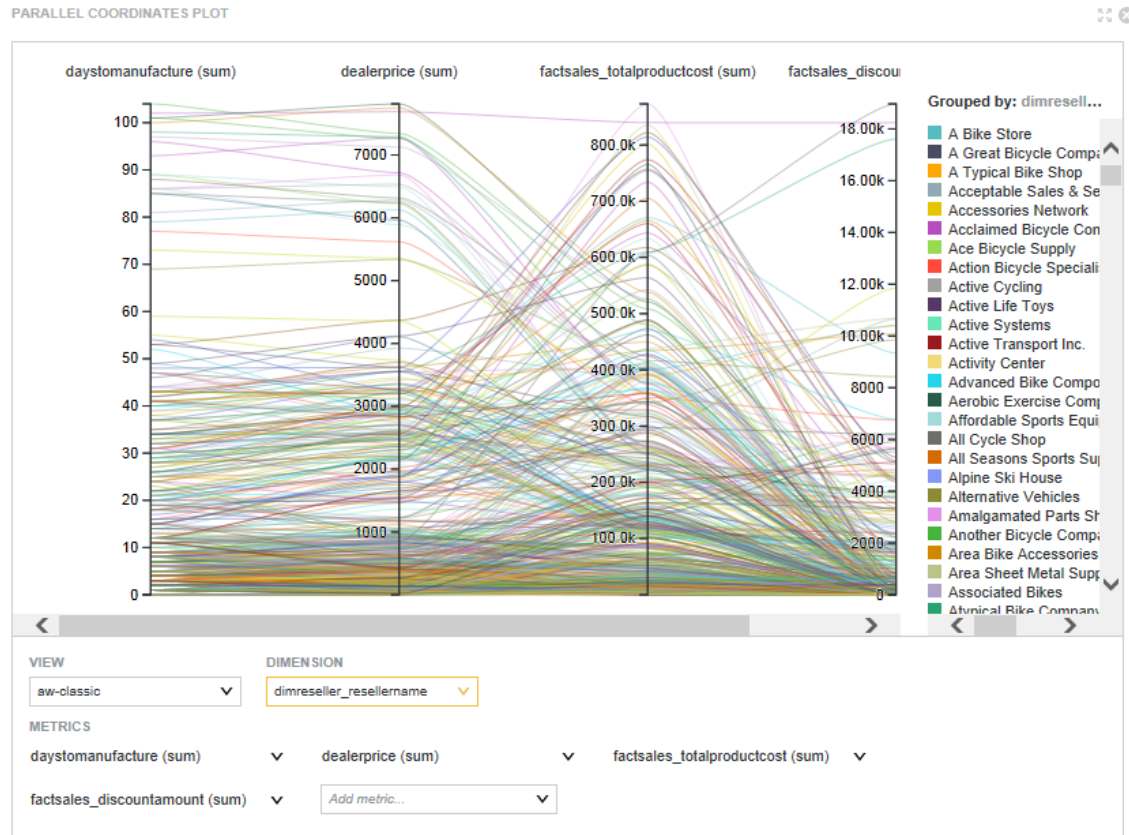
## Parallel Coordinates Plot

A **Parallel Coordinates Plot** displays the values of several metrics for each value of a single grouping dimension.

---

**Note:** The **Parallel Coordinates Plot** chart does not support multi-assign attributes.

---

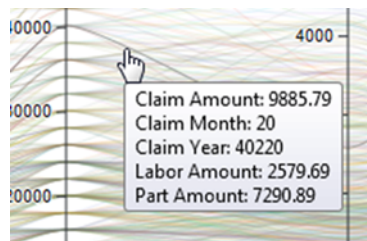


On the chart:

- Each metric is represented by a vertical axis.
- Each dimension value is represented by a colored line, which crosses each metric axis based on the metric value for the dimension value.

To focus on the metric values for a specific dimension value, hover the mouse over either a specific line or a specific entry in the chart legend.

The line for that dimension value is highlighted, and the rest of the lines grayed out. A tooltip displays listing the metric values.



For example, a **Parallel Coordinates Plot** can display the average price, average rating, number of cylinders, and average mileage for types of car models.

### Refining Parallel Coordinates Plot data

You can select metric values in a **Parallel Coordinates Plot** to refine the displayed data.

### Selecting Parallel Coordinates Plot data

You can select the data view, dimension, and metrics to display when viewing a **Parallel Coordinates Plot**.

## Refining Parallel Coordinates Plot data

You can select metric values in a **Parallel Coordinates Plot** to refine the displayed data.

You can either:

- Use the legend or plotted lines to select individual dimension values.
- Refine the dimension to only include values associated with a specific range of metric values.

For example, you can refine the Model attribute to only include models for which the average price is between \$15,000 and \$20,000.

To refine the data displayed in a **Parallel Coordinates Plot**:

1. To refine by a specific dimension value, do one of the following:

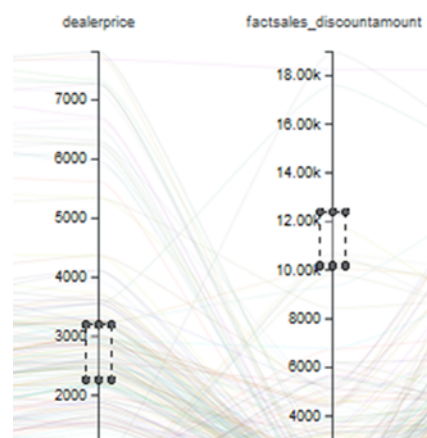
- In the chart legend, click a dimension value.
- On the chart, click the line plotted for a dimension value

You can also use Ctrl-click to select multiple individual values.

2. To refine by dimension values that are within a range of metric values:

- a. Use the mouse to select a range of values on a metric axis.

You can create multiple selected areas on different axes.



To clear the selected areas, click the **Clear selections** link above the chart legend.

- b. Click any of the selected areas.

The dimension values for all lines that run through the selected ranges are added to the **Selected Refinements** panel.

## Selecting Parallel Coordinates Plot data

You can select the data view, dimension, and metrics to display when viewing a **Parallel Coordinates Plot**.

To select **Parallel Coordinates Plot** data:

1. From the **View** list, select the view to use as the source of the plot data.
2. From the **Dimension** list, select the grouping dimension to display on the chart.
3. To add a new metric to the chart, select the metric to add from the **Add Metric...** list.

You can select any numeric attribute.

4. To select the aggregation method for a metric, click its drop-down icon and select the operation.

Note that when the data is refined to a single record, then on the **Parallel Coordinates Plot**, the aggregation is removed, and the chart shows the actual metric values for the record.

5. To remove a metric from the chart, click its delete icon.
6. To change the order of the metric axes:
  - a. Hover the mouse over the axis label at the top of the axis you want to move.
  - b. Click and drag the axis to the new location.



---

## Other Data Visualizations

These components provide you with other options for visualizing values in your data.

### Map

The **Map** component allows you to analyze data based on geographic location. It can only be used if the data contains at least one geocode attribute.

### Pivot Table

The **Pivot Table** component allows users to perform comparisons and identify trends across several cross sections of data.

### Summarization Bar

The Summarization Bar allows users to quickly view metric or dimension values that summarize aspects of the underlying data.

### Tag Cloud

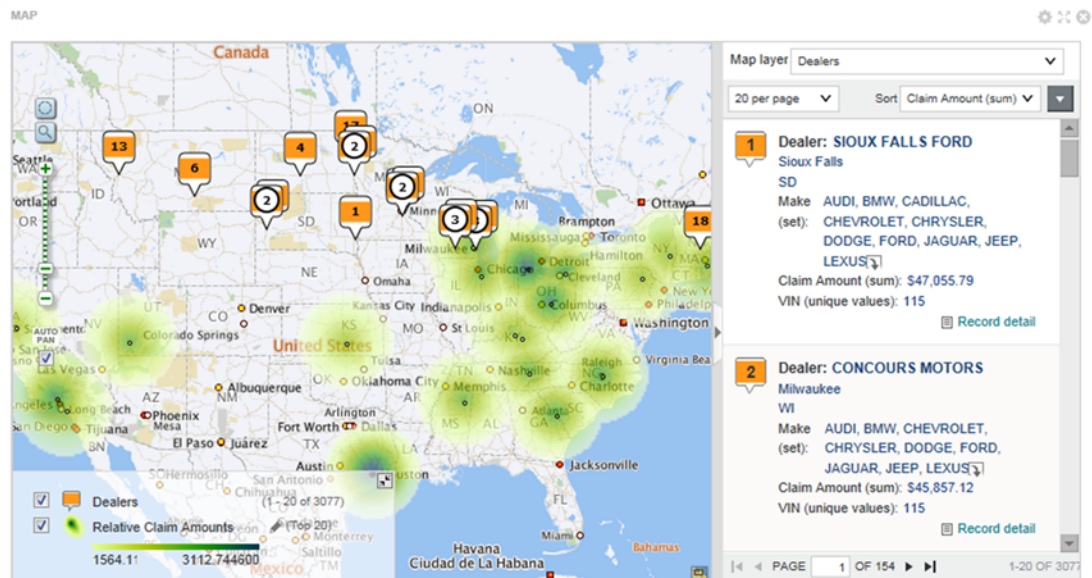
The **Tag Cloud** component allows users to quickly compare a set of displayed terms based on the value of an associated metric, such as the number of times the term occurs.

### Timeline

The **Timeline** component allows you to see changes in the value of one or more metrics over the same time period. For example, one timeline could track total sales against the sale date, and another could track total claims against the claim date.

## Map

The **Map** component allows you to analyze data based on geographic location. It can only be used if the data contains at least one geocode attribute.



### Configuring the Map connection to Oracle MapViewer

The **Map** component uses Oracle MapViewer (11.1.1.7.2 or greater) to display the map. Studio includes framework settings to configure the connection to MapViewer.

### Using the Map

You use the **Map** component to view geographic locations.

### Configuring the Map

The **Map** component configuration allows you to configure the map layers.

## Configuring the Map connection to Oracle MapViewer

The **Map** component uses Oracle MapViewer (11.1.1.7.2 or greater) to display the map. Studio includes framework settings to configure the connection to MapViewer.

By default, Studio is configured to use the public instance of MapViewer. If you are using that instance, and your browser has access to the Internet, then you should not need to make any configuration changes.

Studio includes the following framework settings related to the MapViewer connection. If you are using your own instance of MapViewer, or if your browser does not have Internet access, then you must change these settings. For details on configuring framework settings in Studio, see the *Administrator's Guide*.

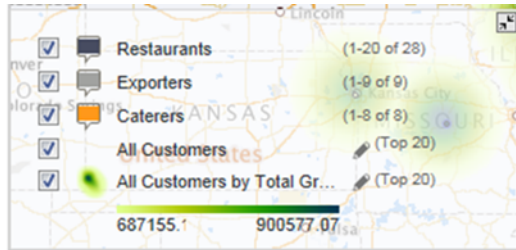
Framework Setting	Description
<code>df.mapLocation</code>	<p>The URL for the MapViewer eLocation service. The eLocation service is used for the text location search, to convert the location name entered by the user to latitude and longitude.</p> <p>By default, this is the URL of the global eLocation service.</p> <p>If you are using your own internal instance, and do not have Internet access, then set this setting to "None", to indicate that the eLocation service is not available. If the setting is "None", Studio disables the text location search.</p> <p>If this setting is not "None", and Studio is unable to connect to the specified URL, then Studio disables the text location search.</p> <p>Studio then continues to check the connection each time the page is refreshed. When the service becomes available, Studio enables the text location search.</p>
<code>df.mapViewer</code>	<p>The URL of the MapViewer instance. By default, this is the URL of the public instance of MapViewer.</p> <p>If you are using your own internal instance of MapViewer, then you must update this setting to connect to your MapViewer instance.</p>
<code>df.mapTileLayer</code>	<p>The name of the MapViewer Tile Layer. By default, this is the name of the public instance.</p> <p>If you are using your own internal instance, then you must update this setting to use the name you assigned to the Tile Layer.</p>

## Using the Map

You use the **Map** component to view geographic locations.

### Showing and hiding map layers

At the bottom left of the **Map** component is the layer list, which shows the available layers configured for the component. You can use the icon in the top-right corner of the layer list to expand or collapse the list.



The layer list contains a check box for each numbered point and point layer. To hide a numbered point or point layer, deselect its check box.

If any heat layers are available for the **Map** component, then there is a single heat layer check box, with a list containing all of the available heat layers. From the heat layer list, select the heat map to display. To hide the heat layer entirely, deselect the heat layer check box.



### Types of map layers

Each **Map** component is configured with one or more map layers. Each map layer uses a specific layer type.

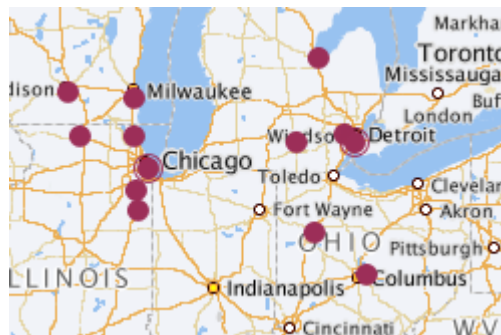
The available map layer types are:

Layer Type	Description
Numbered point layer	A numbered point layer displays a numbered point for each location on the map.



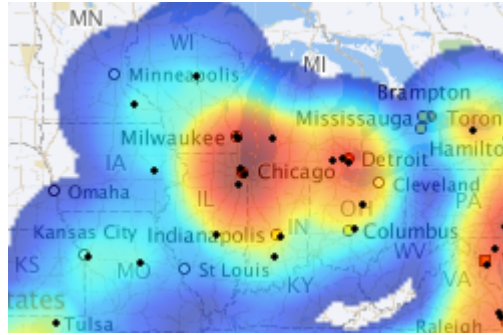
The layer includes a separate list containing the details for each point. The **Map** component can display multiple numbered point layers at the same time, with color used to differentiate the layers.

Point layer A point layer displays a point for each location on the map.



The **Map** component can display multiple point layers at the same time, with color used to differentiate the layers.

Layer Type	Description
Heat map layer	<p>A heat map layer can display:</p> <ul style="list-style-type: none"> <li>• A point for each location on the map.</li> <li>• A shaded cloud with color gradients to show either the relative density of the points on the map, or the change in an associated metric value between locations. The cloud colors are also reflected on the map points.</li> </ul>



You can choose whether to display just the points, just the cloud, or both.

The **Map** component can only display one heat layer at a time.

### Using the numbered points list

When a numbered point layer displays on the **Map** component, the component by default displays the numbered point list to the right of the map.

You can use the expand/collapse icon to the left of the list to show or hide the list.

The list also uses the standard pagination to allow you to page through the list and control the number of points to display per page. See [Paging through data](#).

From the numbered point list:

1. If there is more than one numbered point layer, then from the **Map Layer** list, select the layer for which to display the numbered point list.
2. To change the sort order for the numbered point list:
  - a. From the **Sort** list, select the attribute to use for the sort.

Note that if the data is currently refined using a keyword search, then the list also includes a **Search Relevance** option.

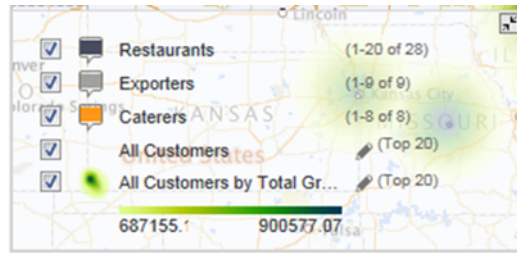
- b. To switch the sort direction, click the sort direction toggle. When you change the sort for a numbered point layer, it affects the points displayed on the map.

### Changing the display of points and heat map layers

From the layer list, for point and heat map layers, you can change the amount of information displayed.

To adjust the display of point and heat map layers:

1. In the layer list, the point layer name is followed by the number of points displayed. To change the point layer display:



- a. Click the edit icon for the point layer.
  - b. On the layer dialog, from the **Display** list, select whether to display the top or bottom set of points.
  - c. From the number list, select the number of points to display.
  - d. From the **Based on** list, select the value to use to determine the top or bottom points.
2. In the layer list, the heat layer information includes the number of points. To change the heat layer display:



- a. Click the edit icon for the heat map.
- b. On the layer dialog, from the **Display** list, select whether to display the top or bottom set of points.
- c. From the number list, select the number of points to display.
- d. Under **Display heat as**, click the radio button to indicate whether to display only the points, only the cloud, or both the points and the cloud.
- e. Use the **Range** slider to determine the range for calculating the cloud.

The range determines how much a point affects the cloud calculation. The larger the range, the more impact each point has on the cloud calculation.

### Searching the map

The **Map** component can include both text and range filter search tools to allow you to find specific locations.

The text search allows you to display locations based on their proximity to a specified place name. For example, you can display locations within 10 miles of Boston, Massachusetts.

---

**Note:** The version of MapViewer used by the component may be limited to selected geographic areas. If you try to do a text search based on a location outside those geographic areas, an error displays.

---

The range search allows you to only display locations in a selected area of the map.

To do text and range searches on the **Map** component:

1. To do a text search:

- a. Click the Geospatial Text Search icon above the zoom bar.

The **Location Search** pop-up displays.

- b. In the field, type the search text.
- c. In the **Show locations within** field, specify the number of miles or kilometers, then select the unit of measurement.
- d. Click the search icon.

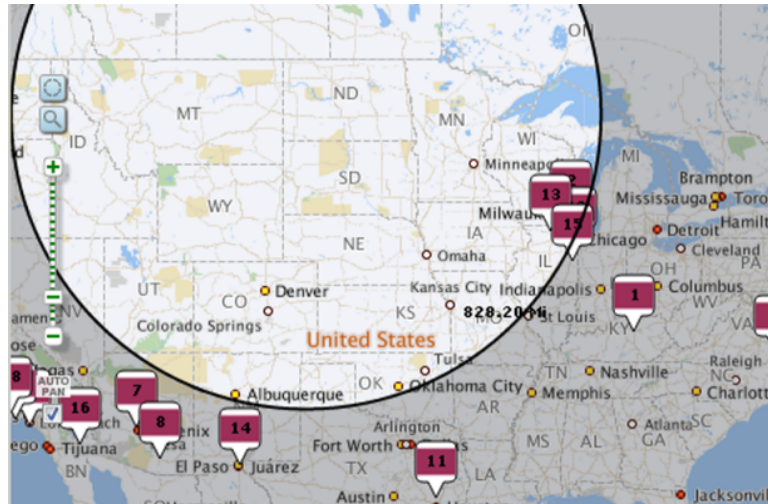
The map is updated to only show pins for locations within the specified area.

If the **Auto Pan** check box is selected, the map shifts automatically to display all of the pins for the first page of results. This also adds a **Range Filter** entry to the **Selected Refinements** panel.

2. To do a range search:

- a. Click the range search icon.
- b. On the map, click and drag the mouse from the middle to the edge of the area to select.

As you drag the mouse, the distance from the point you started at displays.



- c. When you have selected the area you want, release the mouse.

The map is updated to only display pins for locations within that selected area.

A **Range Filter** entry also is added to the **Selected Refinements** panel.

If the **Auto Pan** check box is selected, the map shifts automatically to display all of the pins for the first page of results.

### Displaying details for a map point

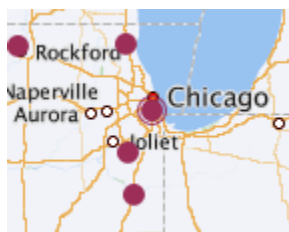
When you click a map point, Studio displays additional information about that location.



If several numbered points are clustered together, then the **Map** component displays the number of points in the cluster.

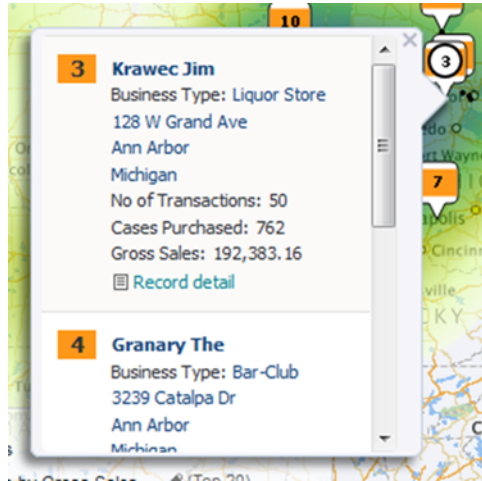


For point layers, when there is a cluster of points, Studio displays a circle around the cluster.



When you click a cluster of points, the **Map** component displays the details for all of the points in the cluster.





The values in the point details may be configured to allow you to refine the data. See [Using a component to refine data](#).

The point details may also include a link to display record details. See [Displaying details for a component item](#).

## Configuring the Map

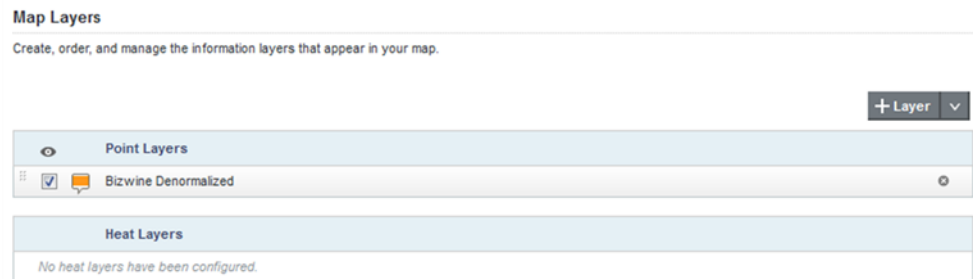
The **Map** component configuration allows you to configure the map layers.

### About the default Map configuration

When the **Map** component is first added, it uses the default configuration. For each base view that has a geocode attribute, the component includes a numbered point layer. The layer name is set to the view name.

The details for the default layers contain the identifying attributes for the view.

The automatically created layers are visible by default.

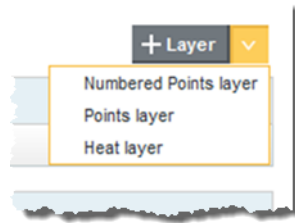


### Adding, editing, and deleting map layers

From the **Map Layers** tab of the **Map** component edit view, you manage the list of layers.

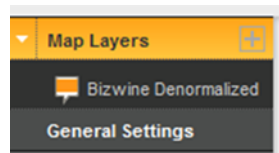
To add, edit, and delete map layers:

1. To add a new layer to the map, either:
  - Hover the mouse over **+Layer**, then in the list, click the type of layer to add.



For details on the map layer types, see [Selecting the map layer type](#).

- In the edit view menu, click + next to the **Map Layers** entry in the edit view menu.



A new layer is added to the list.

If you selected the layer type when you added the layer, then the **Data Selection** tab displays. See [Selecting the view to use for the map layer](#).

If you did not select a layer type, then by default the layer is a numbered point layer, and the **Layer Type** tab displays. See [Selecting the map layer type](#).

- To edit a layer, click the layer name.
- To change the name of a layer:
  - Double-click the layer name.  
The name displays in an editable field.
  - Type the new name.
  - Press **Enter**.
- To delete a layer, click its delete icon.

### Configuring a map layer

For each map layer, you select the data to use and configure the display.

#### Selecting the view to use for the map layer

The **Data Selection** tab for the map layer contains the list of views from the project data. Map layers can only use views that contain a geocode attribute. The geocode attribute cannot be multi-value.

Views that do not contain a geocode attribute are disabled.

#### Selecting the map layer type

By default, a new map layer is a numbered point layer. After you create the layer, you can select the type of layer.


The available layer types are:

Layer Type	Description
<b>Numbered Points</b>	<p>Displays each location on the map as a numbered point. The numbered points are also displayed in a corresponding list.</p> <p>End users can page through the list of numbered points. This is the default layer type for a new map layer.</p>
<b>Points</b>	<p>Displays each location on the map as a point.</p>
<b>Heat Layer</b>	<p>Can display:</p> <ul style="list-style-type: none"> <li>• A point for each location on the map. The points are color coded based on the value of a selected metric.</li> <li>• Shaded clouds to show the change in metric values across a geographic area.</li> </ul> <p>End users can display one or the other or both.</p>


On the **Layer Type** tab for a layer, to select the layer type, click the icon for the type.

**Customers: Layer Type**


Select the style to use to render the map layer



Numbered Points



Points



Heat Layer

**Numbered Points**

- Records appear on the map as numbered pins
- Each point can show more details when clicked
- A corresponding list appears with details for each numbered pin on the page
- Numbered points layers support pagination

**Defining the map points for the map layer**

On the **Points Definition** tab for the map layer, you select the geocode attribute to use for the map points, and whether the map layer uses aggregated points.

For an aggregated layer, you also select the group dimensions to use for the aggregation.

For a heat map layer, you select the metric to use to determine the point colors.

To define the map points:

1. From the **Geospatial attribute** list, select the geocode attribute to use for the map locations.

### Customers: Points Definition

Select attributes to define the layer's points

Geospatial attribute:

LatLong

Detail points
  Aggregated points

- Detail point map layers render the data points based on the grain of the records in the data view

2. You also select whether the map locations need to be aggregated.
  - To not aggregate the map locations, click **Detail points**.
  - To aggregate the map locations, click **Aggregated points**.

For example, if the view contains a simple list of stores, with the geocode being the store location, you would not need to aggregate the locations, and could use the detail points option.

But if the view contains a list of sales transactions, with the geocode being the store location for the sale, then to generate a single list of store locations, you might aggregate by a dimension such as the store name.

3. If you are creating an aggregated map layer, then you configure the list of grouping dimensions to use for the aggregation.
  - a. To add a dimension to the list, drag a dimension from the **Available attributes** list to the **Grouping dimensions** list.

### Customers: Points Definition

Select attributes to define the layer's points

Available attributes

Filter...

- Booking Month
- Booking Quarter
- Booking Year
- Business Types
- Cities
- Contact Names
- Contact Positions
- Countries of Origin
- Customer Address
- Customer Id
- Customer Name
- CustomerAgreedDaysCredit
- CustomerDiscount
- CustomerZin

Geospatial attribute:

LatLong

Detail points
  Aggregated points

- Aggregated point map layers group the data points by one or more selected dimensions

Grouping dimensions (optional):

Customer Id

- b. To determine the order for applying aggregation, drag each dimension to the appropriate location in the list. The dimension at the top of the list is applied first.
  - c. To remove a dimension from the list, click its delete icon.
4. For a heat map layer, you can select a metric to use for setting the layer point and heat cloud colors.

For example, if the locations are stores, then the heat map layer colors could reflect the total sales at each store.

If you do not select a metric, then the heat map colors are based on the relative density of the points on the map. So if the locations are stores, then the heat map colors would change based on the number of stores in a given area.

To select the metric, click **Select Metric**. On the metric dialog, click the metric, then click **Apply**.

For predefined metrics or system metrics, the aggregation method is built in.

For other attributes, the default aggregation method is assigned. You can then use the list to select a different aggregation method.

To select a different metric, click the delete icon, then click **Select Metric**.

### Configuring the layer name and display options

On the **Layer Properties** tab for the map layer, you configure the layer name and options for displaying the map locations, and, for heat map layers, the cloud overlay.

On the **Layer Properties** tab:

1. In the **Layer name** field, type the name of the layer.


The layer name is used to identify the layer on the map legend.

For numbered point layers, the name is also used in the layer list on the numbered point list.

2. If the view contains multiple geocode attributes, then from the **Geo filtering attribute** list, select the attribute to use when searching or filtering the locations.

By default, the map layer uses the same geocode attribute for both the map points and the filtering.

3. For point layers and heat map layers, from the **Size of points** list, select the size of the map location points.
4. For numbered point and point map layers, from the **Layer color** list, select the color to use to display the points.
5. For a heat map layer, to configure the color and metric value range:

<p><b>Color Options</b></p> <p>Layer color:  </p> <p><input checked="" type="radio"/> Dynamically determine min / max values on color ramp</p> <p><input type="radio"/> Fixed values on color ramp</p> <p>Minimum: <input type="text"/></p> <p>Maximum: <input type="text"/></p>	<p><b>Heat Options</b></p> <p><input type="radio"/> Show heat points by default</p> <p><input type="radio"/> Show heat cloud by default</p> <p><input checked="" type="radio"/> Show both by default</p> <p><input checked="" type="checkbox"/> Enable end user to adjust heat range</p>
---	--

- a. From the **Layer color** list, select the color range to use for the map points and cloud.
  - b. To automatically calculate the minimum and maximum metric values for the color range, click **Dynamically determine min/max values on color ramp**.
  - c. To specify the minimum and maximum values, click **Fixed values on color ramp**, then in the **Minimum** and **Maximum** fields, type the minimum and maximum values.
6. For a heat map layer, under **Heat Options**:
- a. Click the radio button to indicate whether to display by default the location points, heat cloud, or both.
  - b. To allow end users to adjust the heat range, check the **Enable end user to adjust heat range** check box.

### Selecting and configuring the details to display for each map point

On the **Details Template** tab for the map layer, you configure the information to display when users click a point on the map. For detail points, you can also determine whether to include a separate link to display record details.

<p>Available attributes</p> <p><input type="text" value="Filter..."/></p> <ul style="list-style-type: none"> <li>Business Type</li> <li>Cases Purchased</li> <li>City</li> <li>Credit Rating</li> <li>Customer ID</li> <li>Customer Name</li> <li>Days Credit</li> <li>Discount</li> <li>Gross Sales</li> </ul>	<p><input checked="" type="checkbox"/> Use same configuration for pin and list detail templates</p> <p><b>Pin Details Template</b></p> <ul style="list-style-type: none"> <li>Customer Name</li> <li>Business Type</li> <li>Street Address</li> <li>City</li> <li>State</li> <li>No of Transactions</li> <li>Cases Purchased</li> <li>Gross Sales</li> </ul> <p><input checked="" type="checkbox"/> Show link to record details</p>
---	---

To configure the map location details:

1. For a numbered point layer, you can configure a different set of details to display in the numbered points list and when users click the point on the map. To enable

the different sets of details, deselect the **Use same configuration for pin and list detail templates** check box.

Use same configuration for pin and list detail templates

Pin Details Template	List Details Template
Customer Name	Customer Name
Business Type	Business Type
Street Address	Credit Rating
City	Days Credit
State	Discount
No of Transactions	
Cases Purchased	
Gross Sales	

Show link to record details

2. To add an item to a details template, drag it from the **Available attributes** list to the template.
3. To determine the display order of the template items, drag each item to the appropriate location in the list.
4. To remove an item from a template, click its delete icon.
5. To configure the item display, click its edit icon. On the configuration dialog:
  - a. For a metric other than a system or predefined metric, from the **Aggregation Function** list, select the aggregation method to use.
  - b. For a date/time attribute, from the **Date/time Subset** list, select the subset of date/time units to display (for example, Year, Year-Month, Year-Month-Day).
  - c. To include the attribute display name as well as the value, check the **Show attribute name in the template** check box.

**Edit Attribute: Gross Sales** Numeric Attribute

---

Display Options

Show attribute name in the template

Allow sorting by this attribute (when sorting is enabled)

---

▶ VALUE FORMATTING

---

▶ ACTIONS

---

Cancel Apply

- d. To allow users to sort by the attribute, check the **Allow sorting by this attribute** check box.
- e. Under **Value Formatting**, configure the display format for the value. See [Configuring attribute value formats](#).
- f. Under **Actions**, configure the action for when the user clicks the value.

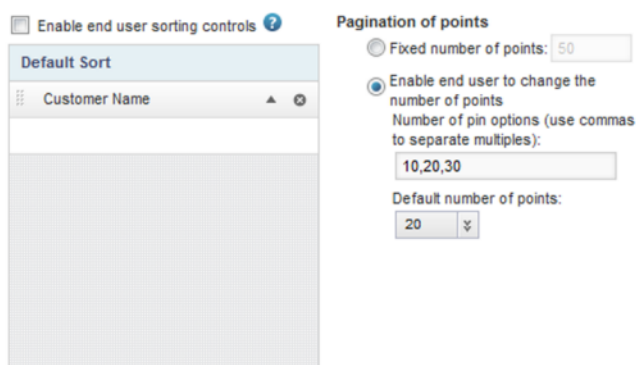
For information on configuring actions for displayed values, see [Configuring actions for attribute values](#).

- g. To save the configuration, click **Apply**.
6. If the points are not aggregated, then to display a link to record details, check the **Show link to record details** check box.

No check box displays for aggregated points and you cannot show a record details link.

### Configuring pagination and sorting for the map layer

On the **Sorting and Pagination** tab for the map layer, you configure the available sort options and pagination for the layer.



To configure the sorting and pagination:

1. To allow end users to change the sort order, check the **Enable end user sorting controls** check box.
 

Selecting this check box lets end users sort the map using any of the values in the details template.
2. To configure the default sort order:
  - a. To select the attributes to include in the default sort, drag the attributes from the **Available attributes** list to the **Default Sort** drop zone.
  - b. To determine the order for applying the sort, drag each attribute to the appropriate location in the list.
  - c. For each sorting attribute, use the sort order icon to indicate whether to sort in ascending or descending order.
  - d. To remove an item from the default sort, click its delete icon.
3. To always display the same fixed number of locations on the map:
  - a. Click **Fixed number of points**.
  - b. In the field, enter the number. For a numbered points layer, this is the number of points to display per page. Users can then page through the list.



4. To allow end users to change the number of locations to display:
  - a. Click **Enable end user to change number of points**.
  - b. In the **Number of pin options** field, type a comma-separated list of options for the number of locations to display.
  - c. From the **Default number of points** list, select the default number of locations to display.

### Determining which map layers are visible

If you are using the default map layer configuration, then all of the automatically generated map layers are visible on the end user view of the **Map** component. If you are not using the default configuration, then you can control whether the layer is visible.

On the **Map Layers** tab, to not display the map layer on the end user view of the **Map** component, deselect the layer's check box.

### Configuring general display settings for the Map

On the **General Settings** tab of the **Map** component edit view, you configure general settings for the component as a whole. These settings are not specific to a map layer.

On the **General Settings** tab:

1. Under **Filtering**:

#### Filtering

- Enable geospatial filtering via marquee
- Enable geospatial range filtering via text search

- a. To allow users to search the map using the range search, check the **Enable geospatial range filtering via marquee** check box.
- b. To allow users to search the map using the location text search, check the **Enable geospatial filtering via text search** check box.

2. Under **Display Settings**:

#### Display Settings

- Show the numbered list overlay panel by default when numbered point layers are visible on the map

Map height:  pixels

Units of measure:

- a. If the map includes a numbered pin layer, then to display the numbered pin list by default when the component is first displayed, check the **Show numbered list overlay panel by default** check box.
- b. In the **Map height** field, type the height in pixels for the **Map** component.

- c. From the **Units of measure** list, select the unit of measurement to use for the map.

To use the preferred unit of measurement based on the current locale, select **use locale default**.

Otherwise, you can have the map always use either miles or kilometers.

## Pivot Table

The **Pivot Table** component allows users to perform comparisons and identify trends across several cross sections of data.

The values in the header rows and columns represent every possible grouping of the selected data fields.

Each body cell contains a metric value that corresponds to the intersection of the values in the heading rows and columns. In the following example, each cell contains either the average price or average score for a specific combination of region, designation, and vintage.

Vintage ▲		1992		1993		1994	
Region ▲	Designation ▲	Price(avera...	Review Sco...	Price(avera...	Review Sco...	Price(avera...	Review Sco...
Alsace	Auslese			\$38.00	90.00		
	Best Buy	\$9.00	88.00	\$9.00	88.00	\$10.00	85.00
	Botrytis	\$19.00	87.00			\$185.00	90.00
	Brut	\$20.00	85.00	\$15.00	85.00		
	Cellar Select	\$43.50	92.50	\$40.00	92.00		
	Cellar Selecti...	\$43.50	92.50	\$40.00	92.00		
	Cremant	\$20.00	85.00	\$15.00	85.00		
	Cuvee	\$22.45	84.65	\$27.60	86.07	\$31.29	86.00
	Grand Cru	\$27.60	85.88	\$42.88	86.47	\$45.76	86.51
	Highly Reco...			\$34.00	90.67		
	Late Harvest	\$30.00	87.00				
	Reserve	\$13.61	81.89	\$16.00	84.40	\$29.00	84.25
	Rose	\$20.50	88.50	\$56.29	87.14	\$40.77	88.23
Sec							

Cells may also be highlighted based on the displayed value. In the example, values are highlighted in red if they are below a certain number, and in green if they are above a certain number.

### Using the Pivot Table

In addition to viewing the **Pivot Table** data, you may be able to change the table layout.

### Configuring the Pivot Table

For a **Pivot Table** component, you select the view to use and configure how the table displays.

## Using the Pivot Table

In addition to viewing the **Pivot Table** data, you may be able to change the table layout.

The Pivot Table includes the following common functions, summarized in [Common component controls](#):

- [Actions > Print](#)
- [Actions > Export](#)

### Navigating through Pivot Table data

You use the scroll bars to scroll through the **Pivot Table** rows and columns. You can also use the scroll bars to jump to a location.

When you hover the mouse over a location on a scroll bar, Studio calculates and displays the dimension values for that relative location in the **Pivot Table**.



For the vertical scroll bar, the row dimension values are shown. For the horizontal scroll bar, the column dimension values are shown.

If you then click the scroll bar, the **Pivot Table** jumps to that location.

### Displaying summaries and highlighting in the Pivot Table

The **Pivot Table** component can include summaries of individual row or column dimensions as well as summary rows and columns for the entire table. Individual values may also be highlighted. The **Pivot Table** may be configured to allow you to show or hide the summaries, and to show or hide the highlighting

If you can control these features, then a **View Options** button displays at the top of the table.

To enable and disable these options:

1. Click **View Options**.

A list of check boxes displays. There are options for enabling and disabling the highlighting, the dimension summaries, and the table summaries.

2. To show or hide the value highlighting, select or deselect the **Conditional Formatting** check box.
3. To show or hide the summaries for each dimension, select or deselect the **Summaries** check box.
4. To show or hide the table summaries, which aggregate all of the values in a row or column, select or deselect the **Grand Summary** check box.

### Sorting the Pivot Table dimension values

For each **Pivot Table** row and column dimension, you can determine the order for displaying dimension values.

At the top left of the **Pivot Table** are the headings with the dimension names.



To switch the display order of the values for a dimension, click the dimension heading.

### Changing the layout of the Pivot Table

You may be able to change the layout of the **Pivot Table**, including the dimensions and metrics that are displayed and the row and column display order.

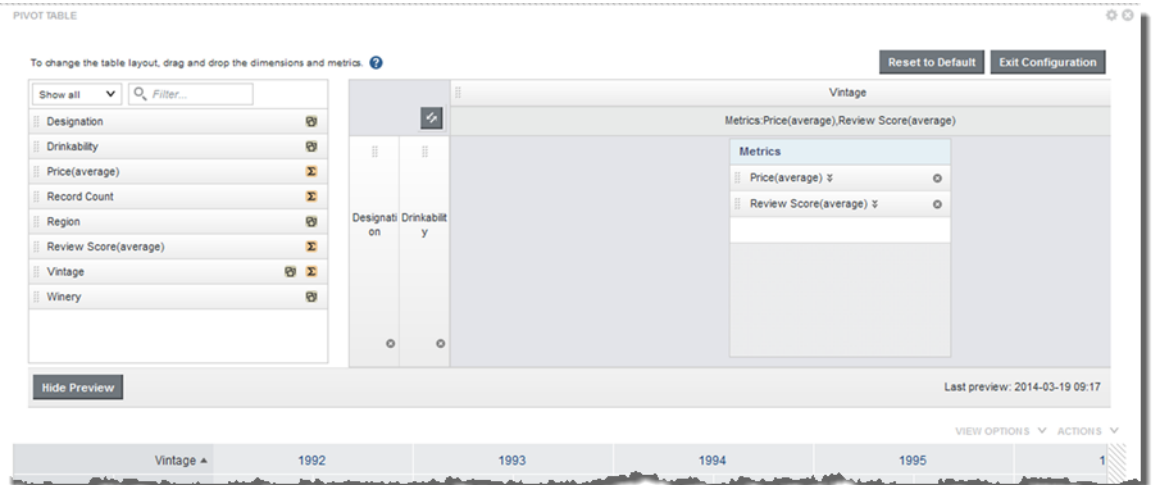
If the **Pivot Table** is configured to allow you to change the table layout, then a **Configure** button displays at the top of the component.

To change the layout of the **Pivot Table** table:

1. Click **Configure**.

On the configuration view, the list at the left shows the available metrics and dimensions that are not currently displayed on the table.

The rest of the configuration view shows a mockup of the displayed dimension columns and rows, with the list of displayed metrics in the middle. A metrics placeholder indicates whether the metrics are displayed as rows or columns.



2. To change the order of the displayed dimension rows or columns, drag the dimension to the new location.
3. To remove a dimension or metric, click its delete icon.

The dimension or metric is removed from the table and added to the available list.

4. To add a new dimension, drag the dimension from the available list to the appropriate location in the row or column groups.

5. To add a new metric, drag the metric from the available list to the appropriate location in the metrics list.

If the metrics are displayed as columns, then the metric at the top of the list displays in the leftmost column.

If the metrics are displayed as rows, then the metric at the top of the list displays in the top row.

6. To swap all of the rows and columns, including the metrics, click the swap icon in the top left corner of the layout.
7. To only switch whether the metrics are displayed as rows or columns, click the swap icon in the metrics placeholder.
8. To change the display order of the metric rows or columns, drag each metric to the appropriate location in the list.
9. As you make changes, a preview of the resulting table displays below the configuration. It is updated automatically as you make changes.
  - a. To hide the preview, click **Hide Preview**.
  - b. To show the preview again, click **Show Preview**.
10. To revert the layout back to the original default, click **Reset to Default**.
11. To exit the configuration view, click **Exit Configuration**.

The currently displayed layout is used.

### Using the Pivot Table to refine the project data

You can use the **Pivot Table** dimension values to refine the project data.

If a dimension allows refinement, then when you click the dimension values in the row and column headings, the data is refined by that value.

When you click a metric value, then the data is refined by all of the dimension values that apply to that cell and that allow refinement.

So for example, if a cell displays an average price of \$25.55 for Red wines in the Bordeaux region for the year 1992, then when you click the value \$25.55, the data is refined by the values:

- Red for the Wine Type attribute
- Bordeaux for the Region attribute
- 1992 for the Vintage attribute

For more details on refining by data on a component, see [Using a component to refine data](#).

## Configuring the Pivot Table

For a **Pivot Table** component, you select the view to use and configure how the table displays.

## Configuring the layout of the Pivot Table

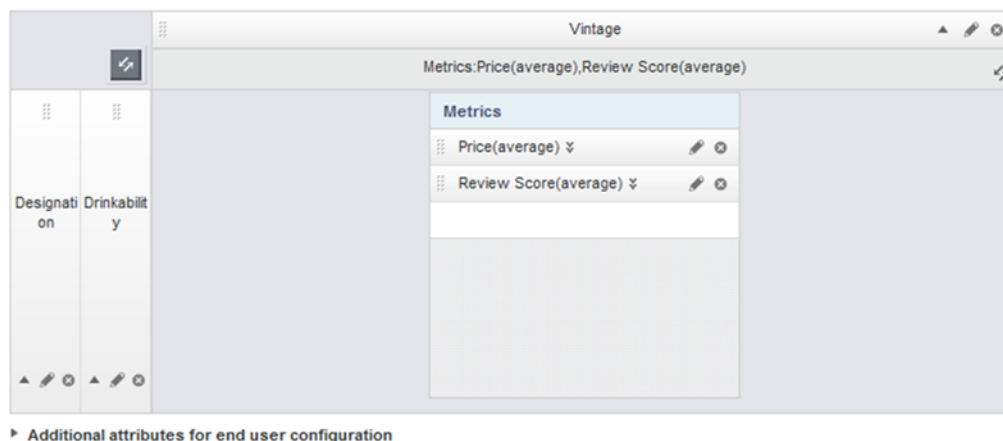
Once you select the view to use, you can configure the layout of the dimensions and metrics on the **Pivot Table** table.

### About configuring the Pivot Table layout

You use the **Table layout** tab of the **Pivot Table** edit view to configure the layout of the **Pivot Table**.

The layout includes:

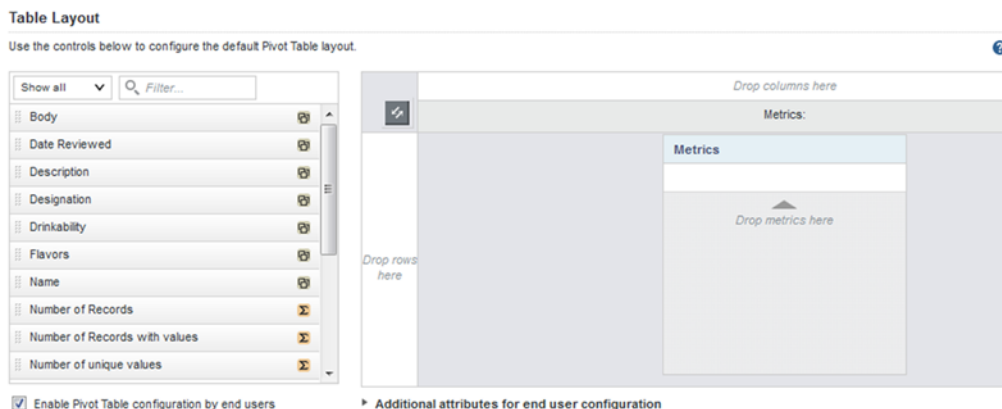
- The dimensions to use to aggregate the **Pivot Table**
- How to group the dimension values into nested rows and columns. The table must have at least one row-based dimension.
- The metrics to display on the **Pivot Table**
- Whether to display those metrics as rows or columns. By default, the metrics display as columns, with the headings in the last row of column headings.



### Adding dimensions and metrics to the Pivot Table

Each **Pivot Table** consists of dimension rows and columns to determine the aggregation, and metrics to determine the displayed values.

At the right of the **Table Layout** tab is a mockup of the **Pivot Table** format, with drop zones to add dimensions to the row and column groups, and to add metrics. A placeholder block indicates whether the metrics are displayed as columns or rows.



To populate the default view of the **Pivot Table**:

1. To add a dimension to a row or column group, drag the dimension from the list to either the row or column group drop zone.

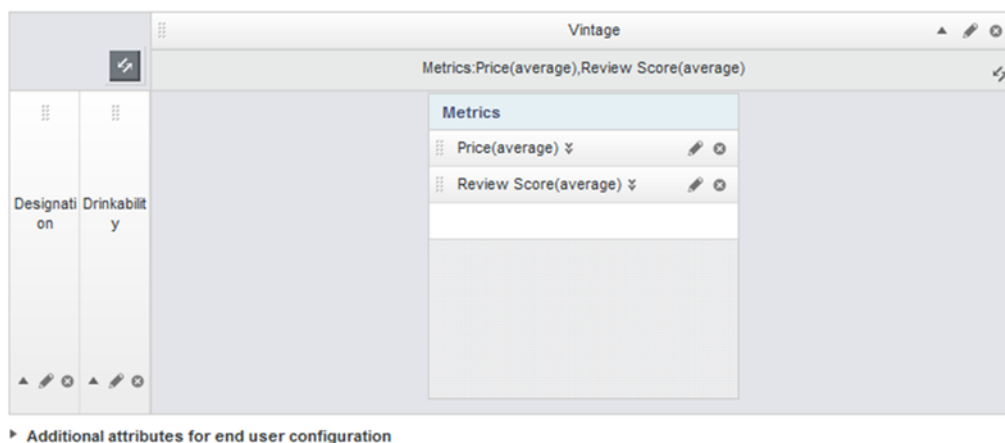
You must add at least one dimension to the row group zone.

If there are existing row or column dimensions, you can drop the new dimension before or after those dimensions.

You can also drag and drop the dimensions within the mockup to change the order of the row and column groups.

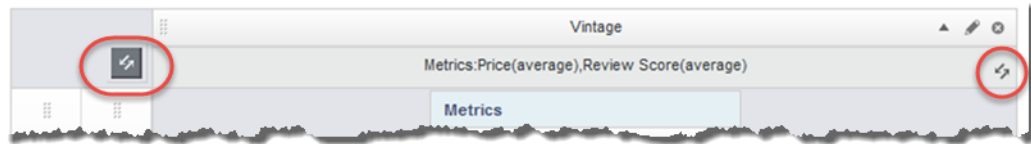
2. To add a metric to the **Pivot Table**, drag the metric from the list to the metric drop zone.

The order of the metrics in the list determines the order (from left to right for columns or top to bottom for rows) of the metrics in the **Pivot Table**. The placeholder also lists the selected metrics.



### Changing the orientation of the Pivot Table

The **Table Layout** tab on the **Pivot Table** edit view includes options to switch whether dimensions and metrics are displayed as rows or columns.



To switch the **Pivot Table** row and column groups, click the swap icon at the top left of the mockup. The swap includes the metrics, so for example if your metrics were displayed as columns, when you click **Swap**, they are changed to display as rows.

To only change the location of the metrics, click the swap icon on the metrics placeholder.

### Adding dimensions and metrics

If you are allowing end users to change the configuration of the **Pivot Table**, you can provide additional available dimensions and metrics that are not displayed on the default view of the table.

On the **Table Layout** tab of the **Pivot Table** edit view, the **Available attributes for end user configuration** section below the **Pivot Table** mockup contains the lists of these available dimensions and metrics.

▼ Additional attributes for end user configuration

Dimensions	Metrics
Region	Number of Records
Winery	

To add the additional dimensions and metrics:

1. To expand or collapse the section, click the section heading.
2. To add additional available dimensions, drag the dimensions from the attributes list to the **Dimensions** list.
3. To add additional available metrics, drag the metrics from the attributes list to the **Metrics** list.

### Configuring a dimension

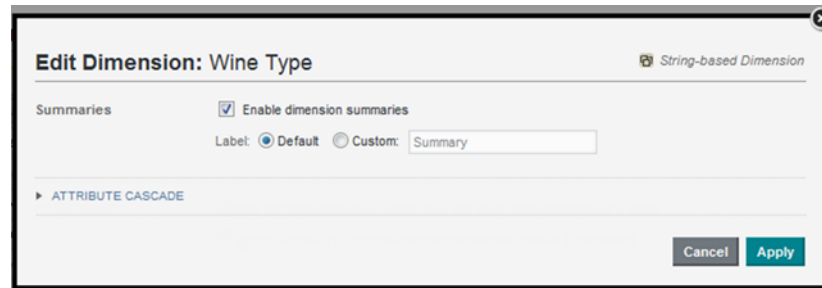
For each **Pivot Table** dimension, you can configure the label for the dimension, whether to display a summary row or column for the dimension, and rules for refining by the dimension value.

From the **Table Layout** tab of the **Pivot Table** edit view, to configure a dimension:

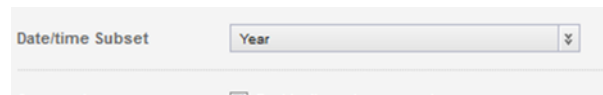
1. Click the edit icon for the dimension you want to configure.

The **Edit Dimension** dialog displays.





2. If the dimension is a date/time attribute, then from the **Date/time subset** list, select the default date/time subset to display.



By default, Studio selects the largest date/time unit that is available for that attribute.

3. On the edit dimension dialog, to display a summary row or column for the dimension, check the **Enable dimension summaries** check box.
4. Under **Label**, by default, the summary label is "Summary", and **Default** is selected.

To provide a custom label, click **Custom**, then type the new label into the field.

5. Use the **Attribute Cascade** section to configure cascading for the dimension.

For details on dimension cascading and how to configure it, see [Configuring cascading for dimension refinement](#).

6. To save the dimension configuration, click **Apply**.

### Configuring a metric

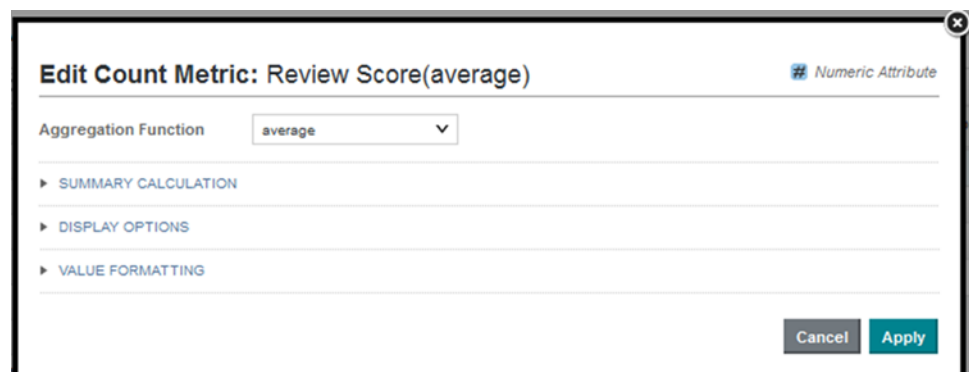
For a **Pivot Table** metric, you can configure the aggregation method for the metric, the tooltip for summary values, and the format of the metric value.

For information on selecting the aggregation method for a metric, see [Selecting the aggregation method to use for a metric](#).

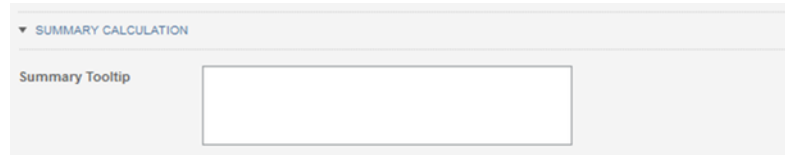
To configure all of the options for a metric:

1. On the **Table Layout** tab, click the edit icon for the metric you want to edit.

The **Edit Count Metric** dialog displays.

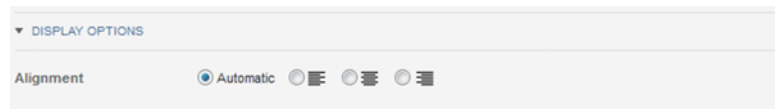


- The **Summary Calculation** section contains the **Summary Tooltip** field, used to add a description to the tooltip displayed when users hover the mouse over a metric value in a summary row or column.



- The **Display Options** section contains a setting to determine the horizontal alignment of the value in the metric column. When the dialog is first displayed, the section is collapsed.

To expand or collapse the section, click the section heading.



To select the alignment, click the radio button.

- The **Value Formatting** section contains settings to customize the format of the metric values.

For details on formatting displayed values, see [Configuring attribute value formats](#).

- To save the metric configuration, click **Apply**.

### Allowing end users to configure the table layout

You can configure the **Pivot Table** to allow end users to change the layout.

On the **Table Layout** tab of the **Pivot Table** edit view, to allow end users to change the layout, check the **Enable Pivot Table configuration by end users** check box.



### Highlighting Pivot Table metric values that fall within a specified range

The **Conditional Formatting** tab of the **Pivot Table** edit view allows you to highlight specific metric values.



To configure whether to use conditional formatting, and select the values to highlight:

1. To allow conditional formatting, make sure that the **Enable conditional formatting** check box is selected. It is selected by default.

If the check box is not selected, then there is no conditional formatting.

2. In the **Metrics** list, click the metric for which to configure conditional formatting.

For each metric, the list displays the number of conditional formatting rules created for it.

3. To add a new range of values to highlight for that metric, click **+Add Condition**.

4. To configure a condition:

- a. From the **Condition Rule** type list, select the type of comparison.

- b. In the field (or fields, for the **between** and **a range from** options), enter the value(s) against which to do the comparison.

Note that for the **between** option, the values are inclusive. So if you specify a range between 20 and 30, values of 20 and 30 also are highlighted.

- c. From the **Color** list, select the color to use for the highlighting.

- d. In the **Tooltip Description** field, type the text to display in the tooltip for a highlighted cell.

5. The conditions are applied based in the order they are listed. So a condition at the top of the list has a higher priority than a condition lower in the list.

To change the priority of the conditions, drag and drop them to the appropriate location in the list.

6. To remove a condition, click its delete icon.

### Configuring display options for the Pivot Table

The **Display Options** tab of the **Pivot Table** edit view contains options for setting size and other display options for the **Pivot Table** table.

To configure the **Pivot Table** display options:

1. Under **Table height**, in the field, type the number of rows that are visible at a time.
2. In the **Column width** field, type the width in pixels to use for each column.
3. Under **Grid display**, to hide any empty rows or columns, check the **Hide empty rows/columns** check box.
4. Under **Summaries**, use the check boxes to indicate whether to display the summary and grand summary rows by default.

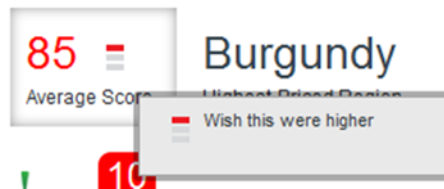
## Summarization Bar

The Summarization Bar allows users to quickly view metric or dimension values that summarize aspects of the underlying data.

Each summary item contains one of the following types of values:

Summary Item	Value Description
Metric value	The actual value of a specific metric, such as total sales or average profit.
Dimension value	The dimension value associated with either the top or bottom value of an associated metric value, such as the product category with the highest total sales.
Number of flags	The number of flags. Flags are based on the value of a metric for a selected dimension or dimensions. For example, a flag summary item can reflect the number of product categories with more than 500 total sales.

Values can use icons and color highlighting to provide additional context for that value. You can also display a tooltip containing a description and value details:



For flags, you can click the number to display the list of values.



Some summary values can redirect to a configured URL. Dimension values may be refinable -- see [Using a component to refine data](#) for details.

#### Configuring Summarization Bar display options

The **Display Options** tab of the **Summarization Bar** edit view contains settings to control the label position and text size for all of the summary items.

#### Adding, renaming, and deleting summary items

When you first add a **Summarization Bar** component to a project page, it automatically displays a single metric value summary item that shows the number of records in the base view of the first data set. You can add, rename, reorder, or remove displayed items.

#### Configuring a metric value summary item

For a metric summary value, on the **Define Item** tab of the **Summarization Bar** edit view, you select the metric value to display.

### Configuring a dimension value spotlight

For a dimension value spotlight summary item, on the **Define Item** tab of the **Summarization Bar** edit view, you select the dimension and metric to use.

### Configuring metric and dimension value display and actions

For metric and dimension value spotlight summary items, the summary item includes a label above or below the displayed metric or dimension value. You can also configure the display options and action for the summary item.

### Configuring conditional formatting

For a metric or dimension spotlight value summary item, you can configure different formatting based on the metric value.

### Configuring flag summary items

For a flag summary item, on the **Define Flags** tab of the **Summarization Bar** edit view, you select the dimensions and set up the metric conditions.

## Configuring Summarization Bar display options

The **Display Options** tab of the **Summarization Bar** edit view contains settings to control the label position and text size for all of the summary items.

To configure the display options:

1. On the **Summarization Bar** edit view, click the **Display Options** tab.

### Display Options

*Edit the display options for all of your summary items*

Display name position:  Below value  Above value

Value text size: smaller  larger

Label text size: smaller  larger

2. Under **Display name position**, click a radio button to indicate whether to display the summary item display name above or below the summary item value.
3. Under **Value text size**, use the slider to adjust the size in pixels of the summary item value.

You can set the size anywhere between 11 and 48 pixels. The default size for the summary item value is 28 pixels.

4. Under **Label text size**, use the slider to adjust the size in pixels of the summary item display name.

You can set the size anywhere between 11 and 48 pixels. The default size for the summary item display name is 12 pixels.

## Adding, renaming, and deleting summary items

When you first add a **Summarization Bar** component to a project page, it automatically displays a single metric value summary item that shows the number of

records in the base view of the first data set. You can add, rename, reorder, or remove displayed items.

Summary items can be one of three types:

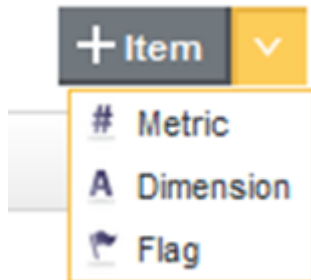
Summary Item Type	Description
<b>Metric Value</b>	Displays a single aggregated metric value, such as total sales or average score.
<b>Dimension Value Spotlight</b>	Displays a single dimension value. The displayed dimension value is associated with the top or bottom value for a single metric. For example, you could display the name of the product category that has the highest or lowest total sales.
<b>Flag</b>	Displays the number of values for a selected dimension or combination of dimensions that meet a set of metric conditions. For example, you could display the number of combinations of product category and sales region that had more than 500 sales and for which the percentage of total sales was at least 90% of expected sales.

To add, rename, or delete summary items:

1. In the component menu bar, click the Options icon:



2. To add a new summary item to the **Summarization Bar**, either:
  - Click the drop-down arrow beside **+ Item** and select the type of summary item to add from the list:



The **Select Data** tab displays.

- In the Edit View menu, click **+** next to the **Summary Items** heading.  
This adds a metric item. Click the **Select Item Type** tab to select a different type.
3. To configure a summary item, click the item name.

- a. To rename a summary item, select the **Summary Items** tab and double-click the item name.  
The name displays as an editable field.
  - b. Type the new name, then press **Enter**.
  - c. To return to the **Summary Items** list after editing a summary item, click the **Summary Items** option in the left menu.
4. To determine the order for displaying summary items, drag each item in the list to the desired location.
  5. To delete a summary item, click its x icon.

## Configuring a metric value summary item

For a metric summary value, on the **Define Item** tab of the **Summarization Bar** edit view, you select the metric value to display.

**Average Price**

*Define the metric value to display on this summary item.*

Metric: Sales Price (avg) ▼

Display Name:  Use metric name:  
Sales Price (avg)

Custom name:

▶ DISPLAY OPTIONS

▶ ACTIONS

To select and configure the summary item value for a metric summary item:

1. To select the metric:
    - a. Click **Select Metric**.
- 
- Average Price**
- Define the metric value to display on this summary item.*
- Metric: Select Metric...
- b. On the **Select a Metric** dialog, click the metric you want to use. You can use the filter field to search for a specific metric.
  - c. Click **Apply**.
2. To select a different aggregation method for a metric, you can use the list next to the metric name.

You cannot select a different aggregation method for a predefined metric.

3. To select a different metric:
  - a. Click the delete icon for the current metric.
  - b. Click **Select Metric**.
  - c. On the **Select a Metric** dialog, click the metric you want to use.
  - d. Click **Apply**.
4. To configure the metric:
  - a. Click its edit icon.
  - b. For metrics that are not predefined, the configuration dialog includes a list to select the aggregation method.
  - c. The **Value Formatting** section allows you to customize the format of the displayed value.



For details on customizing a displayed value on a component, see [Configuring attribute value formats](#).

## Configuring a dimension value spotlight



For a dimension value spotlight summary item, on the **Define Item** tab of the **Summarization Bar** edit view, you select the dimension and metric to use.

**Highest-Priced Region**

*Select the dimension to display, and the metrics to use to determine the dimension value.*

Dimension:   

Top value  Bottom value

Metric:   

Display Name:  Use dimension & metric names:  
Top Region by Sales Price (avg)

Custom name:

▶ DISPLAY OPTIONS

▶ ACTIONS

To select the summary item values:

1. To select the dimension for which to display a value:
  - a. Click **Select Dimension**.



**Highest-Priced Region**

*Select the dimension to display, and the metrics to use to determine the dimension value.*

Dimension:

Metric:

- b. On the **Select a Dimension** dialog, click the dimension you want to use.
  - c. Click **Apply**.
2. To display the dimension value with the highest metric value, click **Top Value**.  
To display the dimension value with the lowest metric value, click **Bottom Value**.
3. To select a different dimension:
  - a. Click the delete icon for the current dimension.
  - b. Click **Select Dimension**.
  - c. On the **Select a Dimension** dialog, click the dimension you want to use.
  - d. Click **Apply**.
4. To select the metric value:
  - a. Click **Select Metric**.
  - b. On the **Select a Metric** dialog, click the metric you want to use. You can use the filter field to find a specific metric.
  - c. Click **Apply**.
5. To select a different aggregation method for the metric, you can use the list next to the metric name.  
You cannot select a different aggregation method for a predefined metric.
6. To select a different metric:
  - a. Click the delete icon for the current metric.
  - b. Click **Select Metric**.
  - c. On the **Select a Metric** dialog, click the metric you want to use.
  - d. Click **Apply**.
7. To configure the metric or dimension:
  - a. Click its edit icon.
  - b. For date/time dimensions, from the **Date/time Subset** list, select the date/time subset to display on the component.  
By default, the largest available subset is used.

- c. For metrics that are not predefined, the configuration dialog includes a list to select the aggregation method.
- d. For both the dimension and the metrics, the **Value Formatting** section allows you to customize the format of the displayed value.

For details on customizing a displayed value on a component, see [Configuring attribute value formats](#).

## Configuring metric and dimension value display and actions

For metric and dimension value spotlight summary items, the summary item includes a label above or below the displayed metric or dimension value. You can also configure the display options and action for the summary item.

On the **Summarization Bar** edit view, on the **Define Item** tab for the summary item:

1. The **Display Name** setting determines the label displayed below the metric or dimension value.

Display Name:

Use dimension & metric names:  
Top Region by Sales Price (avg)

Custom name:

You can either display the dimension and metric names, or use a custom name. The custom name is also used as the name of the summary item.

2. Under **Display Options**, use the **Tooltip** setting to provide a description of the summary item.

▼ DISPLAY OPTIONS

Tooltip: Provide an additional description or details for the summary item metric. The description can include tokens to represent values associated with the selected metric. ?

Example: "Rating: 93"

Width: Summary items are truncated if they don't fit within the defined maximum width.

Default max width (350 pixels)  Custom max width  pixels

In the tooltip text, use the `{MetricValue}` token to represent the metric value.

3. The **Width** setting determines the maximum width of the summary item block.

You can use the default maximum width, or set a custom maximum width.

If the value doesn't fit within the maximum width of the summary item, then Studio truncates it.

4. For a metric summary item, you can allow users to click the metric value in order to navigate to a different page in the project or to an external URL.

To enable the hyperlink, you must at least provide the page name or URL. See [Configuring actions for attribute values](#).

5. For a dimension value spotlight item, you can allow users to click the dimension value in order to either refine by the value, or to navigate to a different page in the project or to an external URL.

Under **Actions**, from the **Type** list, select the action. For details on configuring an action from a component value, see [Configuring actions for attribute values](#).

## Configuring conditional formatting

For a metric or dimension spotlight value summary item, you can configure different formatting based on the metric value.

Conditional formatting based on: [?](#)

Metric value  Percentage of another metric

Style: [Style icons] + Condition

Display	When value is...	Description
<span style="border: 1px solid gray; padding: 2px;">[Style icons]</span>	less than <span style="border: 1px solid gray; padding: 2px;">85</span>	<span style="border: 1px solid gray; padding: 2px;">Wish this were higher</span>
<span style="border: 1px solid gray; padding: 2px;">[Style icons]</span>	greater than <span style="border: 1px solid gray; padding: 2px;">90</span>	<span style="border: 1px solid gray; padding: 2px;">Great score!</span>

▼ DISPLAY OPTIONS

Set summary item text to black for all conditions

Only display this summary item if it meets a condition [?](#)

On the **Conditional Formatting** tab for a summary item, to configure the conditional formatting:

1. Select the type of value to use to determine whether the condition is met:
  - a. Click the radio button to indicate the value to use.

The options are:

Comparison Type	Description
<b>Metric value</b>	Indicates to apply the conditional formatting based on the actual summary item value.
<b>Percentage of another metric</b>	Indicates to apply the conditional formatting based on the ratio of the summary item metric value to another metric. For example, if a summary item displays total sales, then you could configure formatting based on the ratio of total sales to expected sales. So you might configure the total sales to display in green if they are greater than or equal to 100% of the expected sales.

- b. For a ratio to another metric, to select the other metric, click **Select Metric**.
  - c. On the **Select a Metric** dialog, click the metric you want to use, then click **Apply**.
2. From the **Style** list, select the style to use for the conditional formatting.  
The style determines the available set of icons to display on the summary item.
3. To add a condition to the list, click **+Condition**.

4. To configure a condition:
  - a. From the **Display** list, select the icon to display when the condition is met.  
 Note that you can have multiple rows that use the same display. For example, you may want to use the same formatting to highlight both very low and very high outlying values.
  - b. Under **When value is**, select the comparison to use, then enter the appropriate values.
  - c. In the **Description** column, type a description to display in the summary item tooltip when the condition is met.
5. To remove a condition, click its delete icon.
6. In the **Display Options** section:
  - a. By default, the text color on the summary item matches the icon color, if color is used. To always display the summary item value in black, check the **Set summary item text to black for all conditions** check box.
  - b. To only display the summary item if the value meets one of the conditions, check the **Only display this summary item if it meets a condition** check box.  
 For example, if you create a condition for total sales for when the value is less than 1,000,000, then if this check box is selected, the summary item does not display at all when total sales are greater than or equal to 1,000,000.

## Configuring flag summary items

For a flag summary item, on the **Define Flags** tab of the **Summarization Bar** edit view, you select the dimensions and set up the metric conditions.

If you select multiple dimensions, then the conditions are applied to combinations of those dimension values.

For example, if you select the product category and region dimensions, and set the flag condition to total sales greater than 500, then the flag summary item displays if any combination of product category and region has total sales greater than 500.

To configure a flag summary item:

1. In the **Display Name** field, type the text to display on the summary item.

**Low-Scoring Vintages**

Select a combination of dimensions and conditions to determine the type of information to flag and when to flag it.

Display Name:

Flag Color: ▾

Select Dimension(s)

Select one or more dimensions to identify the values that you're interested in. To change the order of the dimensions, drag them up or down in the list.

Vintage ✎ ⊗

+ Dimension

The display name is also used as the name of the summary item.

2. From the **Flag Color** list, select the background color to use for the number of values that match the flag criteria.

3. To add a dimension to the flag summary item:
  - a. Click **+ Dimension**.
  - b. On the **Select a Dimension** dialog, click the dimension you want to use.
  - c. Click **Apply**.

If you select a date/time attribute as a flag dimension, then by default, Studio uses the largest available date/time subset. For example, if the attribute has Year, Year-Month, and Year-Month-Day enabled, then the Year subset is used.

4. The dimension values are applied in the order they are listed. To change the order, drag each dimension to the appropriate location in the list.
5. To remove a dimension from the list, click its delete icon.
6. Use the **Define Conditions** list to select the list of conditions that must be met in order for the flag summary item to display.

A flag summary item is only displayed on the end user view if at least one value matches all of the conditions in the list.

**Define Conditions**  
 Define the conditions (metrics and thresholds) that determine when a dimension value is flagged. A dimension value is only flagged if ALL of the conditions are met. To change the order of the conditions, drag them up or down in the list. + Condition

Condition based on	Metric	Threshold
Metric value	Base Score (avg)	less than 90

- a. To add a condition to the list, click **+Condition**.
  - b. To remove a condition from the list, click its delete icon.
  - c. To change the order for applying conditions, drag and drop the conditions to the appropriate location in the list.
7. For each condition in the **Define Conditions** list:
    - a. From the **Condition Based On** list, select whether the condition is based on:
      - A single metric value. For example, display the flag summary item if any of the dimension values have more than 500 sales.
      - A ratio of two metrics. For example, display the flag summary item if the total sales for any of the dimension values is greater than 90% of the projected sales.
    - b. To select a metric, click **Select Metric**.
    - c. To select a different aggregation method for a metric, you can use the list next to the metric name. You cannot select a different aggregation method for a predefined metric.
    - d. In the **Threshold** column, configure the condition comparison. From the list, select the comparison operator, then in the field or fields, type the comparison value or values.
  8. To configure a dimension or metric:

- a. Click its edit icon.
- b. For date/time dimensions, from the **Date/time Subset** list, select the date/time subset to display on the component.  
By default, the largest available subset is used.
- c. For metrics that are not predefined, the configuration dialog includes a list to select the aggregation method.
- d. For both metrics and dimensions, the **Value Formatting** section allows you to customize the format of the displayed value.

For details on customizing a displayed value on a component, see [Configuring attribute value formats](#).

**9. Under List of flags:**

**List of Flags**  
 Max number of flags to display:   
 Overview content:

- a. In the **Max number of flags to display** field, type the maximum number of flags to display when users click the flag summary item.  
For example, even if 20 dimension values match the flag conditions, you can restrict the list of flags to only show 10.
- b. In the **Overview content** field, type a brief description to display above the list.

**10. Under Display Options:**

**DISPLAY OPTIONS**

Width: *Summary items are truncated if they don't fit within the defined maximum width.*  
 Default max width (350 pixels)  Custom max width  pixels

Display name for column of dimensions:   
 Use default name  Custom name

Select metrics to display as columns:    
 Use default name  Custom name

- a. Under **Width**, to customize the maximum width of the item, click **Custom max width**, then in the field, type the width in pixels.
- b. Under **Display name for column of dimensions**, specify the column heading for the dimension column on the list of flags.  
By default, the dimension names are used. To specify a custom name, click **Custom name**, then in the field, type the custom heading.
- c. Under **Select metrics to display as columns**, to include the metric values in the list of flags, check the check box.

- d. If you are displaying the metrics, then to customize the column heading for the metric values, click **Custom name**, then in the field, type the custom heading.
11. In the **Actions** section, configure the page to display when end users refine by a dimension value in the list of flags.
- For information on configuring a target page for refinement, see [Selecting the target page for a refinement or hyperlink](#).

## Tag Cloud

The **Tag Cloud** component allows users to quickly compare a set of displayed terms based on the value of an associated metric, such as the number of times the term occurs.

For example, a **Tag Cloud** component might display a list of wine flavors, organized based on the number of wines that include each flavor. The component can also display the metric value associated with each term. Alternately, **Tag Cloud** terms can display in a simple list.

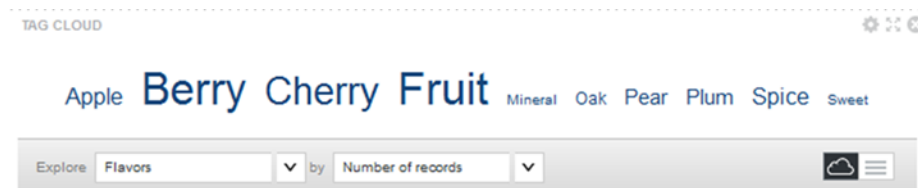
The Tag Cloud includes the following common functions, summarized in [Common component controls](#):

- [Refining Data](#)

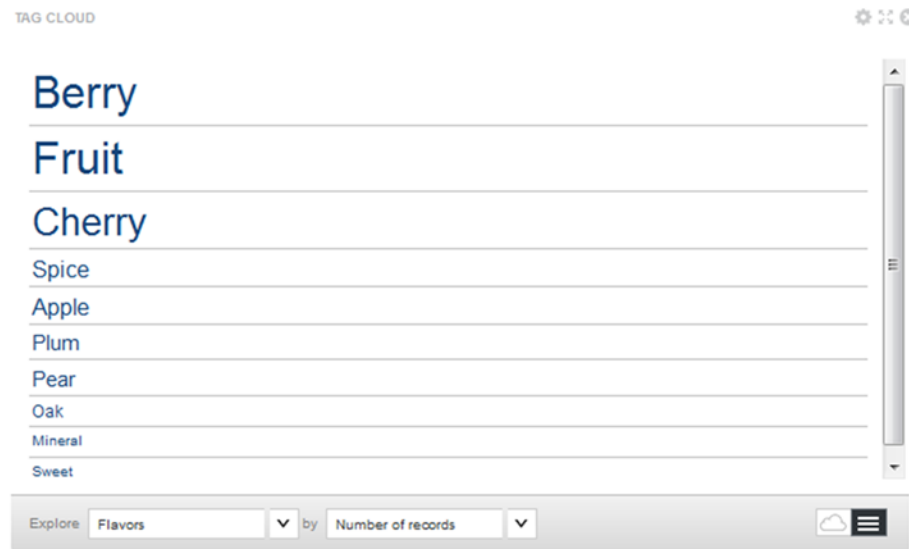
### Toggling cloud and list view

Use the toggle icon to switch between the two layout types.

In cloud layout, the terms display in alphabetical order, and are not separated by a line break:



In list layout, each term displays on a separate line. The terms are displayed in descending order based on the value of the associated metric:



### Selecting data

If there are multiple sets of terms available, then the component includes an **Explore** list. From the **Explore** list, select the set of terms to display.

If there are multiple metrics available, a **by** list displays. Select a metric from the list to use when determining relative size or display order for terms:



### Configuring Tag Cloud display options

The **Display Options** tab of the **Tag Cloud** edit view provides additional display configuration options for the **Tag Cloud**.

### Selecting and configuring dimensions

The terms displayed on the Tag Cloud are values from a selected dimension. You can configure a list of available dimensions for end users to select from, then configure those dimensions to control capitalization and sorting behavior.

### Configuring term size and order

On the **Tag Cloud**, the value of the selected metric determines the relative size of the **Tag Cloud** terms. For list view, it also determines the display order.

### Configuring available metrics

For each **Tag Cloud** metric, you can configure the aggregation function to use. You also determine whether to display the metric values on the **Tag Cloud**, and the format to use for those values.

## Configuring Tag Cloud display options

The **Display Options** tab of the **Tag Cloud** edit view provides additional display configuration options for the **Tag Cloud**.



The configuration includes the number of terms to display, the default format, and the text size for the terms.

### Display Options

Configure the size, format, and optional features for the tag cloud.

Maximum number of keywords:   
Enter a value between 1 and 100

Maximum component height for list view:  pixels

Tag Cloud format:  Cloud  List  
 Allow end user to toggle format

Text size:  10px  32px

To configure the display:

1. In the **Maximum number of keywords** field, type the maximum number of terms to display on the **Tag Cloud**.
2. In the **Maximum component height for list view** field, type the maximum height in pixels of the **Tag Cloud** component when displayed in list view.

If the list does not fit within the specified height, then end users can scroll through the list.

3. Under **Tag Cloud** format:
  - a. Click a radio button to select the default display format (**Cloud** or **List**) for the tag cloud.
  - b. By default, end users can toggle between cloud and list view. To prevent them from changing the format, deselect the **Allow end user to toggle format** check box.
4. Under **Text size**, use the sliders to set the minimum and maximum size in pixels of the **Tag Cloud** terms.

By default, the text size ranges from 10 pixels to 32 pixels.

You can set the text size down to a minimum of 8 pixels, and up to a maximum of 48 pixels.

You can also put the sliders on top of each other to set a single text size for all of the terms. For example, for a list view that includes the number of values next to each term, you may just want to display all of the terms at a size of 11 pixels.

## Selecting and configuring dimensions

The terms displayed on the Tag Cloud are values from a selected dimension. You can configure a list of available dimensions for end users to select from, then configure those dimensions to control capitalization and sorting behavior.

To select and configure Tag Cloud dimensions:

1. In the component menu bar, click the Options icon:



2. Navigate to the **Configuration** tab.

The **Dimensions** list displays with any selected dimensions:

Dimensions (terms in cloud) ?	Metrics (size of terms) ?
<ul style="list-style-type: none"> <li>⋮ Flavors <span style="float: right;">✎ ✕</span></li> <li>⋮ Designation <span style="float: right;">✎ ✕</span></li> </ul>	<ul style="list-style-type: none"> <li>⋮ Number of records <span style="float: right;">✎ ✕</span></li> <li>⋮ Sales Price (average) ▾ <span style="float: right;">✎ ✕</span></li> <li>⋮ Base Score (average) ▾ <span style="float: right;">✎ ✕</span></li> </ul>

By default, the Tag Cloud component contains the first available multi-value dimension that allows refinement.

3. To add a dimension to the **Dimensions** list, drag it from the attributes list.  
Note that you cannot use date/time attributes in a Tag Cloud.
4. To rearrange the order of dimensions in the **Explore** list, click and drag dimensions.
5. To remove a dimension from the **Dimensions** list, click its delete icon.
6. To configure a dimension:
  - a. Click the edit icon for the dimension you want to configure  
The **Edit Dimension** dialog displays.
  - b. Expand **Attribute Cascade** to enable dimension cascading when users refine by a Tag Cloud term.  
See [Configuring cascading for dimension refinement](#).
  - c. Expand **Value Formatting** to control capitalization.
    - To display terms exactly as they are stored in the data, select **Default (no change)**.
    - To display all terms in lowercase, select **Lower case**.
  - d. Click **Apply**.

## Configuring term size and order

On the **Tag Cloud**, the value of the selected metric determines the relative size of the **Tag Cloud** terms. For list view, it also determines the display order.

On the **Configuration** tab of the **Tag Cloud** edit view, the **Metrics** list contains the available metrics for end users to select from.

Dimensions (terms in cloud) ?	Metrics (size of terms) ?
Flavors	Number of records
Designation	Sales Price (average) ▾
	Base Score (average) ▾

In addition to regular attributes and system metrics, the **Tag Cloud** also provides a Relevancy metric. The Relevancy metric applies when the data has been refined, and determines how relevant each term is to the current refinement. To determine the relevancy of each term, Studio compares the frequency of the term in the current refinement to the frequency in the data as a whole.

- If a term occurs more frequently in the current refinement than in the data as a whole, then that term is more relevant, and is larger.
- If a term does not occur any more frequently in the current refinement, then the term is less relevant, and is smaller.

For example, a **Tag Cloud** contains wine types. The Sparkling wine type occurs in 30% of the records. When users refine by a specific region, if the Sparkling wine type now occurs in more than 30% of the records, then the Sparkling wine type is more relevant. If the Sparkling wine type only occurs in 30% or fewer of the records, then it is less relevant.

To select the available **Tag Cloud** metrics:

1. To add a metric to the **Metrics** list, drag the metric from the attributes list to the **Metrics** list.

Note that you cannot use date/time attributes in a **Tag Cloud**.

2. To determine the order for displaying metrics in the **by** list, drag each metric to the appropriate location in the list.

The metric at the top of the list is selected by default when the **Tag Cloud** component is first displayed.

3. To remove a metric from the list, click its delete icon.

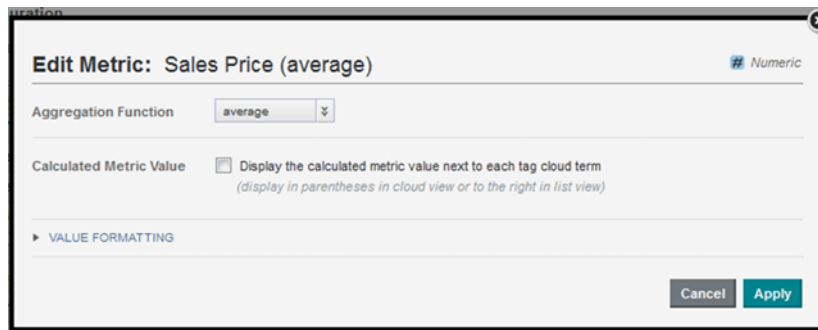
## Configuring available metrics

For each **Tag Cloud** metric, you can configure the aggregation function to use. You also determine whether to display the metric values on the **Tag Cloud**, and the format to use for those values.

On the **Configuration** tab of the **Tag Cloud** edit view, to configure a metric:

1. In the **Metrics** list, click the edit icon for the metric.
2. On the **Edit Metric** dialog, from the **Aggregation Function** method list, select the aggregation method to use to calculate the metric value.

For information on selecting a metric aggregation method, see [Selecting the aggregation method to use for a metric](#).



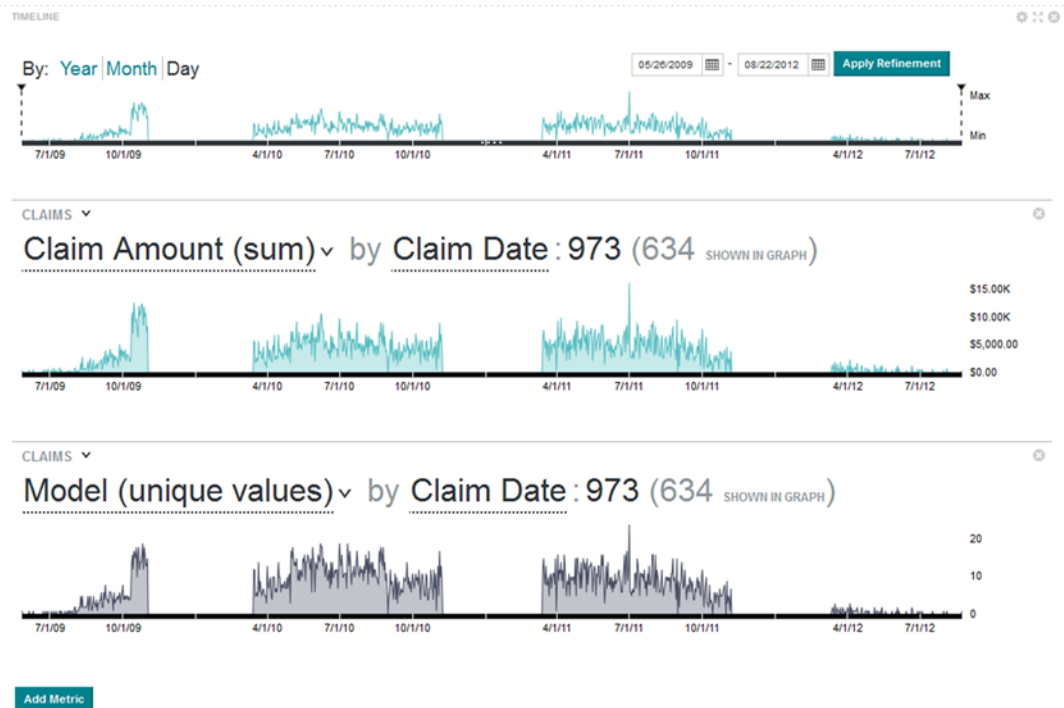
3. To display the calculated metric value for each **Tag Cloud** term, check the **Display calculated metric value next to each tag cloud term** check box.
4. If you are displaying the metric values, then use the **Value Formatting** section to customize the display format.

For details on customizing the format of values displayed on a component, see [Configuring the format of values displayed on a component](#).

5. To save the configuration, click **Apply**.

## Timeline

The **Timeline** component allows you to see changes in the value of one or more metrics over the same time period. For example, one timeline could track total sales against the sale date, and another could track total claims against the claim date.



The component consists of:

Timeline Type	Description
Master timeline	<p>Used to set the range of dates to use on all of the metric timelines. The full range of available dates is based on the earliest and latest dates that are available for the date attributes displayed on the metric timelines.</p> <p>From the Master Timeline, you can also select a date subset to use on the charts. You can only select date spans that are available for all of the selected date attributes.</p> <p>Note also that on the Master Timeline, the metric values are plotted based on the relative minimum and maximum values for each metric. The metrics do not use a single scale.</p>
Metric timelines	<p>Each metric timeline is a bar or line chart that is associated with a specific data set. To plot the chart, the metric timeline uses:</p> <ul style="list-style-type: none"> <li>• A date/time attribute, for the horizontal axis of the chart. If a data set does not include at least one date/time attribute, then you cannot use it on the <b>Timeline</b> component.</li> <li>• A metric value to plot against the date/time values</li> </ul> <p>Each metric timeline includes the number of matching records for the current refinement.</p>

You can add and remove metric timelines, and use a date range to refine the data.

Administrators can also save a configuration as the default to display for all users.

#### [Adding and editing metric timelines](#)

On the **Timeline** component, you can add multiple metric timelines. For each metric timeline, you select the view, date/time attribute, and metric to use for the display. You can also change the display order of the metric timelines.

#### [Setting the displayed date unit and date range for the metric timelines](#)

On the **Timeline** component, you can use the Master Timeline to refine the project data based on a specific date/time range, or refine by a date value on a metric timeline. You can also change the date unit to use on the horizontal axis for the charts.

#### [Saving the current timeline configuration as the default](#)

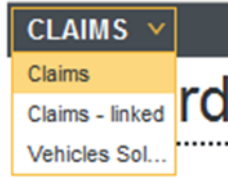
If you are able to edit the project, then you can also save the current configuration of the **Timeline** component as the default for all users.

## Adding and editing metric timelines

On the **Timeline** component, you can add multiple metric timelines. For each metric timeline, you select the view, date/time attribute, and metric to use for the display. You can also change the display order of the metric timelines.

To add and edit metric timelines:

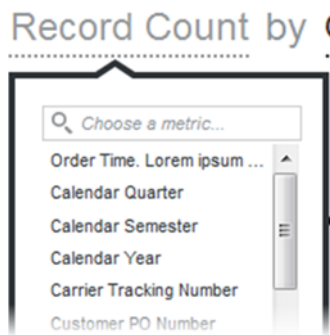
1. To add a new metric timeline, click **Add Metric**.
2. Click the View name of the new timeline to select a data view:



If a view does not include at least one date/time attribute that can be used as a dimension, it cannot be used for a timeline.

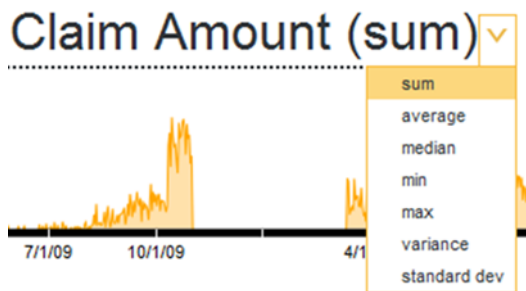
3. To select the metric to display on the metric timeline:
  - a. Click the metric name.

Studio displays a list of available attributes to use for the metric.



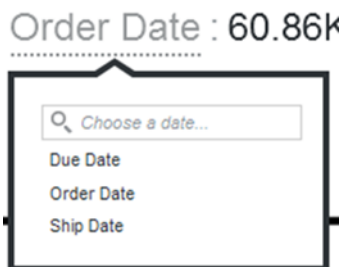
You can also use the search field to find a specific attribute.

- b. Click the attribute you want to use for the metric.
  - c. If the metric is not a predefined metric, then use the aggregation method list to select the aggregation method to use to calculate the metric value.



4. To select the date/time attribute to use on the metric timeline:
  - a. Click the attribute name.

Studio displays the list of available date/time attributes.



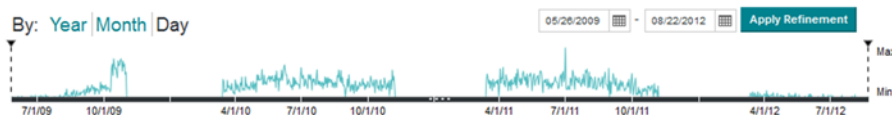
You can also use the search field to find a specific attribute.

- b. Click the date/time attribute you want to use.
5. To change the display order of the metric timelines, drag the timeline to the new location.
6. To delete a metric timeline, click its delete icon.

If there is only one metric timeline, then you cannot delete it. You can only change its configuration.

## Setting the displayed date unit and date range for the metric timelines

On the **Timeline** component, you can use the Master Timeline to refine the project data based on a specific date/time range, or refine by a date value on a metric timeline. You can also change the date unit to use on the horizontal axis for the charts.



Note that if date/time refinements have already been applied, then those values are displayed above the Master Timeline.

To change the displayed date subset and date unit for the metric timelines:

1. Available date units (Year, Month, Day, etc) are listed above the Master Timeline. To display a different date unit, click it.

You can only use date units that are available for all of the selected date/time attributes. If one date/time attribute supports Year, Month, and Day, but another only supports Year and Day, you can only select from Year and Day.

2. To use the Master Timeline to refine the data:
  - a. Select the date/time range.

To do this, use the sliders or the date picker widget.

For each metric timeline, the **Timeline** component displays the preview of the number of records that match the new unit selection.

EMPLOYEES ▾

Record Count by Hire Date : 296 (45 SHOWN IN GRAPH)**b. Click Apply Refinement.**

The date range refinements are added to the **Selected Refinements** panel.

The metric timelines are updated to show the selected date range, and the number of records for each metric timeline is also updated.

3. To refine by a date value on a metric timeline, double-click the point on the metric timeline.

The value for that point is added as a refinement for the date attribute displayed on that timeline.

## Saving the current timeline configuration as the default

If you are able to edit the project, then you can also save the current configuration of the **Timeline** component as the default for all users.

When users first display the page, the **Timeline** component initially displays using the last saved default.

To save the current timeline configuration as the default:

1. In the component menu bar, click the Options icon:



The **Timeline Preferences** dialog opens.

2. Click **Save as Default**.



---

## Web-Based Content

You can use components to display Web-based content on a page. Content can come from external URLs, or it can be created and stored in Big Data Discovery.

### Iframe

The **Iframe** component can display the content of an external URL.

### Web Content Display

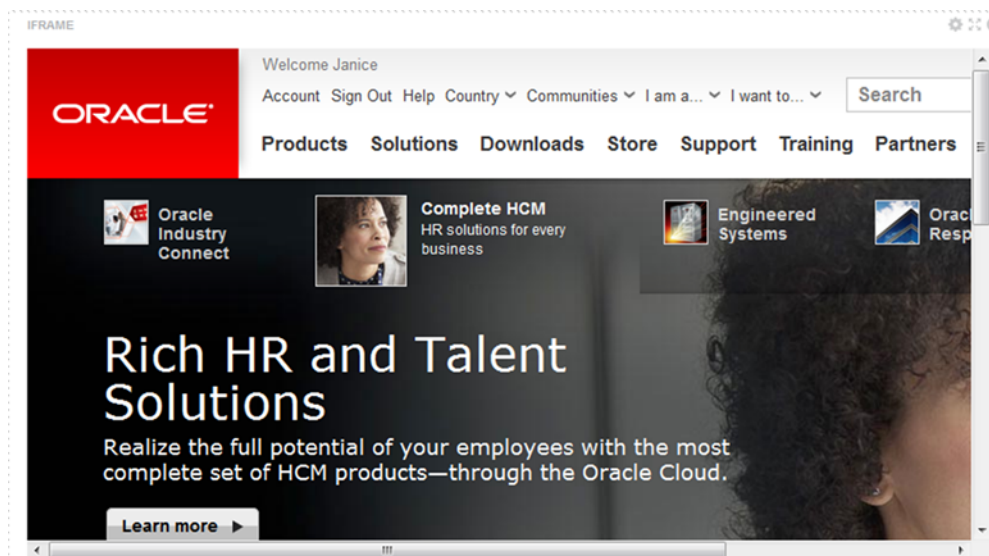
The **Web Content Display** component allows you to create and display HTML content in your project.

## Iframe

The **Iframe** component can display the content of an external URL.

Link targets from the URL also display within the **Iframe** component, unless they are specifically configured to display in another browser window.

For example, you might want to display your company's home page, or provide access to tools to allow users to share their findings.



---

**Note:** Some sites include JavaScript that prevents those sites from loading in an iFrame. The common behavior of such sites is that they briefly appear to load before redirecting to the top-level location for the browser. You cannot display these sites in an iFrame component unless you have access to the page source and can remove the JavaScript in question.

---

## Configuring an IFrame

For an **IFrame** component, you configure the URL to display, and determine whether to require additional authentication in order to view the content.

## Configuring an IFrame

For an **IFrame** component, you configure the URL to display, and determine whether to require additional authentication in order to view the content.

To configure an **IFrame** component:

1. In the component menu bar, click the Options icon to navigate to the Edit view:



2. In the **Source URL** field, type the target URL.
3. If the URL is relative to the context path for Studio, check the **Relative to Context Path** check box.
4. If the page you are embedding requires authentication, configure the authentication information:

- a. Check the **Authenticate** check box.

The authentication fields display.

- b. From the **Authentication Type** list, select the type of authentication.

**Basic** authentication simply provides the user name and password required by the embedded page.

**Form** authentication uses POST or GET to validate the user.

- c. For **Basic** authentication, to specify the user name and password, enter the values in the fields.
- d. For **Form** authentication, from the **Form Method** list, indicate whether to use GET or POST to validate the user.

Authentication	
Authenticate	<input checked="" type="checkbox"/>
Authentication Type	Form
Form Method	Post
User Name	Field Name: <input type="text"/> Value: <input type="text"/>
Password	Field Name: <input type="text"/> Value: <input type="text"/>
Hidden Variables	var1=hello;var2=world

For the user name and password, provide the field names and values to send.

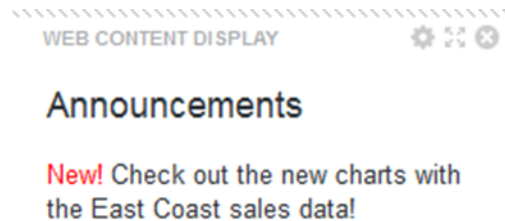
In the **Hidden Variables** field, provide any hidden variables to include in the authentication request.

5. Use the **HTML Attributes** field to provide any additional display parameters for the content.

6. To save the configuration, click **Save**.
7. To return to the end user view, click **Exit**.

## Web Content Display

The **Web Content Display** component allows you to create and display HTML content in your project.



### Configuring a Web Content Display

For the **Web Content Display** component, you provide the content to display. You can also customize the component height.

## Configuring a Web Content Display

For the **Web Content Display** component, you provide the content to display. You can also customize the component height.

To configure the **Web Content Display** component:

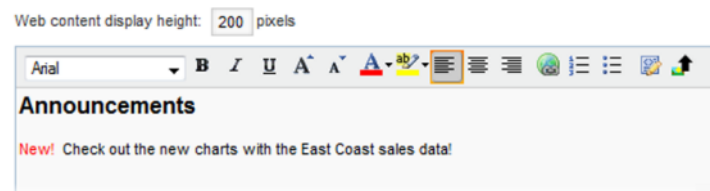
1. In the component menu bar, click the Options icon to navigate to the Edit view:



2. In the **Web content display height** field, enter the desired height.

By default, the component is 200 pixels high.

3. Use the text editor to create or edit the component content.



The text editor provides options to:

- Select a font
- Highlight text
- Change text alignment
- Create numbered and bulleted lists

Note that you cannot include script tags or a separate style tag. If you include these items, they are removed when you save the component.

4. To add a hyperlink:
  - a. Select the link text.
  - b. In the text editor toolbar, click the hyperlink icon.

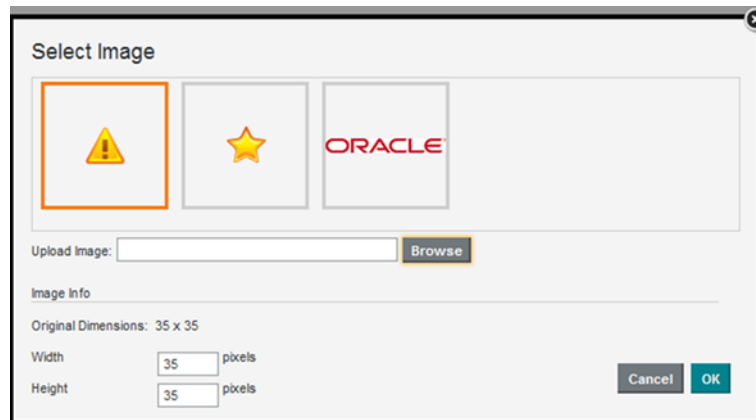
The hyperlink dialog opens.
  - c. In the **URL:** field, type the URL.
  - d. In the **Text to display:** field, enter the display text.
  - e. Click **OK**.

When end users click links in a **Web Content Display** component, the target URL automatically displays in a new browser window.

To edit an existing link, click anywhere in the link text, then click the hyperlink icon.

5. To insert an image:
  - a. In the text editor toolbar, click the image upload icon.

The image upload dialog displays. The dialog displays any images that have been uploaded for this or any other **Web Content Display** component in the current project.



The list does not include images uploaded in other components or other projects.

The images are stored in the database.

- b. To use an existing image in the list, click the image, then click **OK**.
  - c. If you need to upload a new image, click **Browse** to search for and select the image.

Big Data Discovery supports the .gif, .jpeg/.jpg, .bmp, and .png file types.

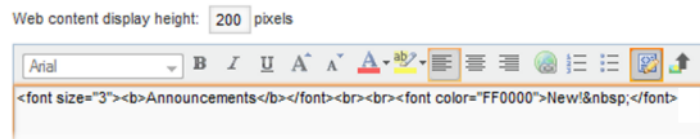
To clear the image file selection, click **Cancel**.

- d. To upload the selected file, click **Upload**.

The image is added to the list, and the dialog is updated to display the image width and height.

You can use the **Width** and **Height** fields to customize the displayed size of the image. The image always maintains its original aspect ratio, so editing one field automatically adjusts the other.

- e. To insert the currently selected image, click **OK**.
6. To display the source HTML, click the source view icon.





---

---

# Index

## A

---

### actions

- configuring for displayed values, [21-24](#)
- Results Table, [23-22](#)
- Results Table, conditions for actions, [23-28](#)
- target page, specifying, [21-28](#)

### Advanced transforms, [19-9](#)

### aggregating, attribute values, [19-23](#)

### aggregation methods

- EQL syntax, [15-25](#)
- list of, [15-25](#)
- selecting for a metric, [21-20](#)

### anti-virus warning, [5-3](#)

### attribute groups

- attribute metadata, displaying full, [16-6](#)
- attribute metadata, selecting columns to display, [16-6](#)
- attributes, displaying the list of, [16-5](#)
- creating, [16-7](#)
- deleting, [16-7](#)
- displaying on the Attribute Groups page, [16-4](#)
- linked views, [16-2](#)
- renaming, [16-8](#)
- selecting attributes, [16-9](#)
- sort order, configuring, [16-10](#)

### Attribute Groups page

- attribute group cache, reloading, [16-11](#)
- attribute groups, creating, [16-7](#)
- attribute groups, deleting, [16-7](#)
- attribute groups, renaming, [16-8](#)
- attribute groups, saving, [16-11](#)
- attribute groups, selecting attributes, [16-9](#)
- attribute metadata, displaying full, [16-6](#)
- attribute metadata, selecting columns to display, [16-6](#)
- attributes, displaying for a group, [16-5](#)
- displaying groups for a view, [16-4](#)
- sort order, configuring for an attribute group, [16-10](#)

### attribute values, aggregating, [19-24](#)

### attributes

- aggregation methods, [15-25](#)

### attributes (*continued*)

- available aggregation methods, selecting, [15-24](#)
- Available Refinements behavior, configuring, [15-28](#)
- default aggregation method, selecting, [15-24](#)
- descriptions, editing and localizing, [15-18](#)
- display names, editing and localizing, [15-18](#)
- storing URLs as values, [21-29](#)

### Available Refinements

- about, [4-1](#)
- bookmarks, saved data for, [11-6](#)
- dates, refining by, [4-5](#)
- implicit refinements, about, [4-7](#)
- linked data sets, controlling automatic filtering, [14-10](#)
- range filters, [4-4](#)
- value list, using for refinement, [4-2](#)

## B

---

### Basic transforms, [19-5](#)

### BDD application, [13-5](#)

### Bin Values transformation, [19-17](#)

### Binned Scatter Plot chart, [24-9](#)

### bookmarks

- about, [11-1](#)
- bookmarks list, displaying, [11-2](#)
- creating, [11-2](#)
- deleting, [11-4](#)
- editing, [11-3](#)
- emailing, [11-4](#)
- links, generating, [11-4](#)
- navigating to, [11-2](#)
- saved data for components and panels, [11-5](#)

### Box Plot

- about, [24-25](#)
- bookmarks, saved data for, [11-6](#)
- data, [24-27](#)
- display options, [24-27](#)

## C

---

### Catalog

## Catalog *(continued)*

- about, [2-1](#)
- data sets, displaying details, [2-3](#)
- data sets, previewing, [2-3](#)
- data sets, sorting, [2-4](#)
- filtering, [2-4](#)
- icons on the tiles, [2-2](#)
- projects, displaying details, [2-3](#)
- projects, navigating to, [10-1](#)
- projects, searching, [2-4](#)
- projects, sorting, [2-4](#)

## Chart

- about, [24-1](#)
- bookmarks, saved data for, [11-6](#)
- default configuration, [24-18](#)
- selecting chart type, [24-18](#)
- snapshot, creating, [24-23](#)
- trellis, [24-18](#)

## Column Container

- about, [22-1](#)
- adding, [22-1](#)
- configuring, [22-1](#)

## components

- Actions menu, about, [21-22](#)
- Actions menu, selecting actions, [21-23](#)
- actions, configuring for component values, [21-24](#)
- adding, [9-3](#)
- Chart, [24-1](#)
- Column Container, [22-1](#)
- comparing items, [21-10](#)
- deleting, [9-3](#)
- dimension cascades, configuring, [21-21](#)
- displaying Record Details, [21-10](#)
- exporting data, [21-13](#)
- formatting displayed values, [21-16](#)
- hyperlinks, configuring external, [21-31](#)
- localizing titles, [7-2](#)
- paging, [21-7](#)
- Pass Parameters action, configuring, [21-31](#)
- Pivot Table, [25-18](#)
- printing, [21-12](#)
- renaming, [21-15](#)
- Results List, [23-1](#)
- Results Table, [23-7](#)
- Tabbed Container, [22-2](#)
- using for refinement, [21-8](#)

## Convert transforms, [19-8](#)

## D

---

- data connection, creating data sets from, [5-5](#)
- data loading options, [5-2](#)
- data sets
  - about reloading, [5-7](#)
  - adding tags, [2-4](#)
  - creating from file, [5-4](#)

## data sets *(continued)*

- data connection, creating from, [5-5](#)
- data set access, [5-13](#)
- deleting, [5-15](#)
- deleting from projects, [13-6](#)
- groups, adding, [5-13](#)
- joining, about, [19-26](#)
- joins, creating, [19-27](#)
- linking, about, [14-1](#)
- links, aggregation methods for linked view, [14-6](#)
- links, creating, [14-2](#)
- links, editing the link key, [14-5](#)
- loading full, [13-2](#)
- managing access, [5-13](#)
- notifications, [5-11](#)
- projects, adding to, [13-1](#)
- reloading from file, [5-7](#)
- reloading from JDBC, [5-8](#)
- removing tags, [2-4](#)
- renaming, [2-4](#)
- supported file types, [5-3](#)
- tags, [2-4](#)
- updating, [13-4](#)
- upload permissions, [5-2](#)
- upload timeouts, [5-3](#)
- users, adding, [5-13](#)

## Data Sets tab, [2-2](#)

## data sources, creating data sets from, [5-5](#)

## data, exporting from projects, [17-1](#)

## E

---

## Entity Extraction Transformation, [19-15](#)

## EQl

- aggregation method syntax, [15-25](#)
- predefined metric definitions, [15-31](#)
- view definition guidelines, [15-12](#)

## exception handling, [19-38](#)

## Explore

- attribute values, filtering, [6-4](#)
- attribute values, hiding, [6-4](#)
- attribute visualization, changing, [6-4](#)
- data set, selecting, [6-2](#)
- details panel, displaying, [6-4](#)
- display type, toggling, [6-3](#)
- scratchpad, saving, [6-8](#)
- scratchpad, shortcuts, [6-9](#)
- scratchpad, using, [6-5](#)

## G

---

## Groovy

- about, [18-5](#)
- data types, [20-1](#)
- reserved keywords, [20-3](#)
- unsupported functions, [20-4](#)



Group Values transformation, [19-18](#)

---

## H

### Histogram Plot

about, [24-24](#)

bookmarks, saved data for, [11-6](#)

data, [24-25](#)

display options, [24-25](#)

### hyperlinks from components

encoding, [21-30](#)

non-HTTP protocol links, [21-27](#)

URLs as attribute values, [21-29](#)

---

## I

### IFrame

about, [26-1](#)

configuring, [26-2](#)

incremental updates with DP CLI, [13-4](#)

---

## J

joining data sets, [19-27](#)

---

## L

layout, changing for a page, [9-2](#)

loading a full data set, [13-2](#)

### locales

configuring user preferred, [1-4](#)

selecting from the user menu, [1-5](#)

supported, [1-3](#)

logging in, [1-1](#)

---

## M

### Map

about, [25-1](#)

bookmarks, saved data for, [11-6](#)

default configuration, [25-9](#)

details template, configuring for a layer, [25-14](#)

display settings, configuring, [25-17](#)

filtering, configuring, [25-17](#)

heat map layer, changing the display, [25-5](#)

map layer data view, [25-10](#)

map layer display options, configuring, [25-13](#)

map layer name, configuring, [25-13](#)

map layer points, defining, [25-11](#)

map layer type, selecting, [25-10](#)

map layers, [25-3](#), [25-4](#)

map layers, adding, [25-9](#)

map layers, configuring visibility, [25-17](#)

map layers, deleting, [25-9](#)

map layers, editing, [25-9](#)

map layers, renaming, [25-9](#)

### Map (*continued*)

map layers, selecting the view, [25-10](#)

map points, displaying details for, [25-8](#)

numbered points list, using, [25-5](#)

Oracle MapViewer, [25-2](#)

pagination, configuring, [25-16](#)

point layer, changing the display, [25-5](#)

range search, [25-6](#)

sorting, configuring, [25-16](#)

text search, [25-6](#)

---

## N

notifications, [5-11](#)

---

## P

### pages

adding, [9-1](#)

deleting, [9-3](#)

layout, changing, [9-2](#)

localizing names, [7-2](#)

renaming, [9-2](#)

target page, specifying for actions, [21-28](#)

### Parallel Coordinates Plot

about, [24-27](#)

bookmarks, saved data for, [11-6](#)

data, selecting, [24-30](#)

refining data, [24-29](#)

Pass Parameters action, configuring, [21-31](#)

### password

changing, [1-2](#)

resetting forgotten, [1-2](#)

### Pivot Table

about, [25-18](#)

bookmarks, saved data for, [11-6](#)

column width, configuring, [25-27](#)

conditional formatting, configuring, [25-26](#)

dimensions, adding, [25-22](#)

dimensions, adding additional available, [25-24](#)

dimensions, configuring, [25-24](#)

end user configuration, enabling, [25-26](#)

end user layout configuration, [25-20](#)

hiding empty rows and columns, [25-27](#)

highlighting, showing and hiding, [25-19](#)

metrics, adding, [25-22](#)

metrics, adding additional available, [25-24](#)

navigating in, [25-19](#)

sorting dimension values, [25-19](#)

summaries, default display, [25-27](#)

summaries, showing and hiding, [25-19](#)

swapping rows and columns, [25-23](#)

table height, configuring, [25-27](#)

using for refinement, [25-21](#)

### predefined metrics

adding to a view, [15-30](#)

- predefined metrics (*continued*)
  - configuring, [15-31](#)
  - deleting from a view, [15-32](#)
- project roles
  - about, [8-1](#)
  - adding, [8-3](#)
  - assigning, [8-3](#)
  - removing, [8-3](#)
  - Studio roles, [8-2](#)
  - types of, [8-1](#)
- Project Settings page
  - Attribute Groups page, [16-2](#)
  - Localization page, [7-2](#)
  - Views page, [15-4](#)
- project types, [8-2](#)
- projects
  - about, [7-1](#)
  - adding tags, [2-4](#)
  - attribute groups, configuring, [16-1](#)
  - creating, [7-2](#)
  - creating from Catalog, [7-2](#)
  - creating from data set, [7-2](#)
  - data sets, adding, [13-1](#)
  - data sets, deleting, [13-6](#)
  - data sets, displaying information about, [10-2](#)
  - deleting, [7-3](#)
  - exporting data, [17-2](#)
  - exporting data, about, [17-1](#)
  - groups, adding, [8-3](#)
  - groups, removing, [8-3](#)
  - localizing the name and description, [7-2](#)
  - navigating within, [10-1](#)
  - pages, adding, [9-1](#)
  - pages, changing the layout, [9-2](#)
  - pages, data binding, [9-3](#)
  - pages, deleting, [9-3](#)
  - pages, renaming, [9-2](#)
  - project roles, assigning, [8-3](#)
  - project types, [8-2](#)
  - removing tags, [2-4](#)
  - renaming, [2-4](#)
  - roles, [8-1](#)
  - tags, [2-4](#)
  - users, adding, [8-3](#)
  - users, removing, [8-3](#)
  - views, configuring for project data, [15-1](#)
- Projects tab, [2-1](#)

## R

---

- range filters
  - switching to values, [4-4](#)
  - using, [4-4](#)
- record identifier, [13-4](#)
- refinements
  - about, [4-1](#)

- refinements (*continued*)
  - dates, refining by, [4-5](#)
  - implicit refinements, about, [4-7](#)
  - range filters, [4-4](#)
  - value list, using for refinement, [4-2](#)
- reloading data in the Catalog, [5-7](#)
- Results List
  - about, [23-1](#)
  - attributes, selecting, [23-3](#)
  - bookmarks, saved data for, [11-6](#)
  - configuring, [23-3](#)
  - data view, [23-3](#)
  - default configuration, [23-3](#)
  - images, [23-4](#)
  - images, configuring, [23-4](#)
  - pagination, [23-6](#)
  - sorting, [23-2](#)
- Results Table
  - about, [23-7](#)
  - action columns, available, [23-22](#)
  - action columns, configuring the display, [23-26](#)
  - action columns, selecting, [23-23](#)
  - actions, configuring conditions for, [23-28](#)
  - aggregated table, adding dimension columns, [23-21](#)
  - attributes, selecting, [23-12](#)
  - bookmarks, saved data for, [11-6](#)
  - changing aggregation, [23-10](#)
  - column sets, adding to an aggregated table, [23-20](#)
  - column sets, configuring, [23-18](#)
  - column sets, selecting on the end user view, [23-9](#)
  - columns, formatting, [23-16](#)
  - conditional formatting, configuring, [23-21](#)
  - conditional formatting, showing and hiding, [23-11](#)
  - configuring, [23-12](#)
  - data view, [23-12](#)
  - data, exporting, [23-30](#)
  - default configuration, [23-12](#)
  - display options, configuring, [23-14](#)
  - hide columns, [23-9](#)
  - Hyperlink action column, configuring, [23-24](#)
  - move columns, [23-9](#)
  - navigating to other URLs, [23-30](#)
  - persistent columns, selecting, [23-18](#)
  - Refinement action column, configuring, [23-25](#)
  - show columns, [23-9](#)
  - sorting, [23-11](#)
  - sorting with summary rows, [23-12](#)
  - sorting, configuring, [23-15](#)
  - summary rows, configuring, [23-15](#)
  - summary rows, showing and hiding, [23-10](#)
  - table type, selecting, [23-12](#)
  - types of rows and columns, [23-8](#)
  - using to refine data, [23-30](#)
- roles

roles (*continued*)  
  project roles, [8-1](#)  
  user roles, [8-2](#)  
runtime exceptions, troubleshooting, [19-40](#)

## S

---

Scatter Plot Matrix chart, [24-11](#)

search

- about, [3-1](#)
- Boolean, [3-6](#)
- keyword search, match mode, [3-5](#)
- relevance ranking, effect of, [3-5](#)
- results, displaying, [3-2](#)
- results, using, [3-2](#)
- snippets, displaying, [3-5](#)
- wildcards, [3-5](#)

security exceptions, troubleshooting, [19-39](#)

Selected Refinements

- bookmarks, saved data for, [11-6](#)
- data set list, [4-7](#)
- linked data sets, [4-9](#)
- negative refinements, [4-9](#)
- refinement tiles, [4-8](#)
- refinements, removing, [4-9](#)
- Selected Refinements panel, hiding, [4-8](#)

Shaping transforms, [19-20](#)

snapshots

- creating, [12-2](#)
- deleting, [12-5](#)
- details, displaying, [12-3](#)
- editing, [12-4](#)
- emailing all, [12-6](#)
- emailing individual, [12-4](#)
- galleries, creating, [12-6](#)
- galleries, deleting, [12-8](#)
- galleries, editing, [12-8](#)
- galleries, emailing, [12-9](#)
- galleries, exporting to PDF, [12-9](#)
- galleries, full screen mode, [12-8](#)
- refinement state, navigating to, [12-5](#)
- saving all to a PDF, [12-5](#)
- saving to a file, [12-5](#)
- snapshots panel, displaying, [12-1](#)

Split transformation, [19-6](#)

static and dynamic typing, about, [19-38](#)

Summarization Bar

- about, [25-27](#)
- action, configuring for summary items, [25-34](#)
- adding items, [25-30](#)
- bookmarks, saved data for, [11-6](#)
- conditional formatting, configuring, [25-35](#)
- deleting items, [25-30](#)
- dimension value, selecting for a dimension spotlight item, [25-32](#)
- display name location, configuring, [25-29](#)

Summarization Bar (*continued*)

- display name, configuring for summary items, [25-34](#)
- flag items, configuring, [25-36](#)
- item types, [25-30](#)
- metric value, selecting for a dimension spotlight item, [25-32](#)
- metric value, selecting for a metric summary item, [25-31](#)
- renaming items, [25-30](#)
- text size, configuring, [25-29](#)
- tooltip, configuring for summary items, [25-34](#)
- using, [25-27](#)
- width, configuring for summary items, [25-34](#)

## T

---

Tabbed Component Container

- bookmarks, saved data for, [11-6](#)

Tabbed Container

- about, [22-2](#)
- adding, [22-2](#)
- configuring, [22-2](#)

Tag Cloud

- about, [25-39](#)
- bookmarks, saved data for, [11-6](#)
- cloud, [25-39](#), [25-40](#)
- configuring, [25-41](#)
- display type, [25-39](#), [25-40](#)
- display, configuring, [25-41](#)
- list, [25-39](#), [25-40](#)
- metrics, configuring, [25-43](#)
- metrics, selecting available, [25-42](#)
- selecting dimensions, [25-41](#)
- values, [25-39](#), [25-40](#)

text search

- disabling, [3-8](#)
- enabling, [3-8](#)

Thematic Map

- about, [24-14](#)
- attribute, selecting, [24-15](#)
- Oracle MapViewer, [24-18](#)

Timeline

- about, [25-44](#)
- bookmarks, saved data for, [11-6](#)
- date range, setting, [25-47](#)
- date unit, setting, [25-47](#)
- default configuration, saving, [25-48](#)
- metric timelines, adding, [25-45](#)
- metric timelines, deleting, [25-45](#)
- metric timelines, editing, [25-45](#)
- metric timelines, reordering, [25-45](#)
- refinement, using for, [25-47](#)

Transform

- about, [18-2](#)
- attributes, adding custom, [19-30](#)

Transform (*continued*)

- attributes, favorite, [19-3](#)
- attributes, filtering, [19-4](#)
- attributes, hiding, [19-3](#)
- binning values, [19-17](#)
- capitalization, changing for Strings, [19-7](#)
- custom transformation, creating, [19-30](#)
- data grid, toggling the display, [19-2](#)
- data set size determining, [18-3](#)
- data set, creating new, [19-43](#)
- data type, changing for an attribute, [19-8](#)
- date/time values, truncating, [19-7](#)
- deleting records, [19-21](#)
- Entity Extraction, [19-15](#)
- Extract Date Part, [19-13](#)
- Extracting key phrases, [19-15](#)
- Geographic Hierarchy, [19-10](#)
- grouping values, [19-18](#)
- locking, [18-6](#)
- logging, [19-40](#)
- menu, about, [18-3](#)
- null values, filling in, [19-19](#)
- numeric transformations, performing, [19-8](#)
- page header, about, [18-3](#)
- preview mode, [18-7](#)
- splitting values, [19-6](#)
- static parser, [19-38](#)
- Tag from Whitelist, [19-16](#)
- transform script, reordering, [19-41](#)
- Transformation Editor, [19-31](#)
- Transformation Editor, using, [19-30](#)
- transformation script, loading, [19-45](#)
- transformation script, publishing, [19-44](#)
- transformation script, un-publishing, [19-46](#)
- transformation, configuring, [19-2](#)
- transformation, deleting attribute, [19-22](#)
- transformation, selecting, [19-2](#)
- transformations, deleting, [19-41](#)
- trimming spaces, [19-6](#)
- whitespace characters, displaying, [19-5](#)
- workflow, [18-4](#)
- workflow diagram, [18-4](#)

transform functions

- about, [18-6](#)
- conversion functions, [20-8](#)
- date functions, [20-9](#)
- dynamic and static typing, [19-38](#)
- enrichment functions, [20-13](#)
- geocode functions, [20-25](#)
- math functions, [20-26](#)
- set functions, [20-28](#)
- String functions, [20-29](#)
- URL functions, [20-33](#)

Transform functions

- date constants, [20-11](#)

Transform, performance, [19-46](#)

Transformation Editor, [19-31](#)

transformation scripts

- about, [18-2](#)
- committing, [19-42](#)
- conditional statements, [20-5](#)
- creating custom transformation plug-ins with Groovy, [19-35](#)
- data type conversion, [20-2](#)
- data types, [20-1](#)
- evaluation, [19-37](#)
- exception handling, [19-38](#)
- function chaining, [19-35](#)
- function syntax, [19-34](#)
- multi-assign values, [20-5](#)
- variable formats, [19-33](#)
- workflow, [18-4](#)

transformations

- about, [18-2](#)
- extending, [19-35](#)

Trim transformation, [19-6](#)

typeahead search

- disabling, [3-8](#)
- enabling, [3-8](#)

## U

---

upload timeouts, [5-3](#)

user roles, [8-2](#)

## V

---

variable formats, [19-33](#)

views

- attribute aggregation methods, selecting available, [15-24](#)
- attribute aggregation methods, selecting default, [15-24](#)
- attribute descriptions, editing and localizing, [15-18](#)
- attribute display names, editing and localizing, [15-18](#)
- attributes, default display format, [15-21](#)
- Available Refinements behavior for attributes, configuring, [15-28](#)
- base views, about, [15-2](#)
- copying, [15-7](#)
- creating a new view, [15-6](#)
- date/time subsets, configuring available, [15-20](#)
- deleting, [15-33](#)
- description, editing and localizing, [15-10](#)
- display name, editing and localizing, [15-10](#)
- displaying for a project, [15-4](#)
- editing the definition, [15-12](#)
- EQL guidelines, [15-12](#)
- linked views, about, [15-2](#)
- predefined metrics, adding, [15-30](#)

views (*continued*)

- predefined metrics, configuring, [15-31](#)
- predefined metrics, deleting, [15-32](#)
- previewing, [15-32](#)
- publishing custom, [15-33](#)

Views page

- attribute aggregation methods, selecting available, [15-24](#)
- attribute aggregation methods, selecting default, [15-24](#)
- attribute descriptions, editing and localizing, [15-18](#)
- attribute display names, editing and localizing, [15-18](#)
- attributes list columns, [15-17](#)
- attributes, default display format, [15-21](#)
- Available Refinements behavior, configuring, [15-28](#)
- date/time subsets, configuring available, [15-20](#)
- displaying, [15-4](#)

Views page (*continued*)

- predefined metrics, adding to a view, [15-30](#)
- predefined metrics, configuring, [15-31](#)
- predefined metrics, deleting, [15-32](#)
- view definition, editing, [15-12](#)
- view description, editing and localizing, [15-10](#)
- view display name, editing and localizing, [15-10](#)
- views, copying, [15-7](#)
- views, creating new, [15-6](#)
- views, deleting, [15-33](#)
- views, displaying details, [15-5](#)
- views, previewing, [15-32](#)
- views, publishing custom, [15-33](#)

## W

---

Web Content Display

- about, [26-3](#)
- configuring, [26-3](#)

