Endeca® Latitude

Latitude Studio User's Guide

Version 2.2.2 • December 2011

Contents

Copyright and disclaimer

Preface

Endeca® Latitude applications guide people to better decisions by combining the ease of search with the analytic power of business intelligence. Users get self-service access to the data they need without needing to specify in advance the queries or views they need. At the same time, the user experience is data driven, continuously revealing the salient relationships in the underlying data for them to explore.

The heart of Endeca's technology is the MDEX Engine.™ The MDEX Engine is a hybrid between an analytical database and a search engine that makes possible a new kind of Agile BI. It provides guided exploration, search, and analysis on any kind of information: structured or unstructured, inside the firm or from external sources.

Endeca Latitude includes data integration and content enrichment tools to load both structured and unstructured data. It also includes Latitude Studio, a set of tools to configure user experience features including search, analytics, and visualizations. This enables IT to partner with the business to gather requirements and rapidly iterate a solution.

About this guide

This guide contains information on configuring the Latitude Studio application, and on using Latitude Studio to build component-based applications for exploring and analyzing business data from MDEX Engines.

Who should use this guide

The *Latitude Studio User's Guide* provides details on using Latitude Studio.

The target audience for this guide includes:

- Application administrators who configure and monitor Latitude Studio.
- Data architects who configure the data sources used to provide the data for Latitude Studio components.
- Power users who use Latitude Studio to build applications to navigate, search, and analyze data from MDEX Engines.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: ¬

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Endeca Customer Support

The Endeca Support Center provides registered users with important information regarding Endeca software, implementation questions, product and solution help, training and professional services consultation as well as overall news and updates from Endeca.

You can contact Endeca Standard Customer Support through the Support section of the Endeca Developer Network (EDeN) at *http://eden.endeca.com*.

Part 1

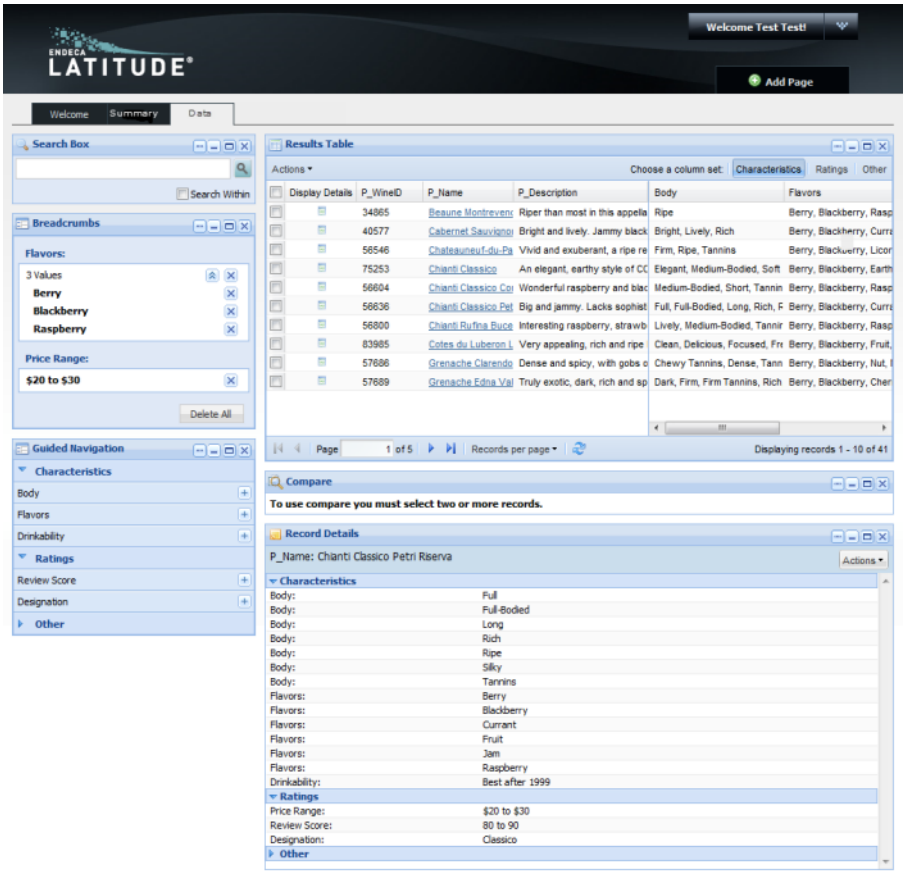
Latitude Studio Introduction and Component Summary

Chapter 1

Introduction to Latitude Studio

About Latitude Studio

Latitude Studio is a web-based tool for building and using applications to search, analyze, and monitor data from MDEX Engines.



Power users use Latitude Studio's set of data display and visualization tools to quickly build these applications.

The applications consist of collections of graphical components. The standard components provided with Latitude Studio can include functions to:

- Navigate to or search for specific data
- Display detailed information about data
- Manipulate and analyze the data
- Highlight specific data values

Business end users can then use these applications to explore the data and perform analysis to discover trends, uncover relationships, monitor changes, and investigate issues.

About the Liferay Portal

Latitude Studio is built upon the Liferay Portal Enterprise Edition.

Liferay Portal is an open-source JSR-286 portal technology.

Latitude Studio extends the basic Liferay functionality to provide enhanced:

- User management
- Security
- Cross-component interaction
- Performance-optimized communication with the Endeca MDEX Engine

This version of Latitude Studio is built upon Liferay Portal 5.2 Enterprise Edition Service Pack 5.

Roles and responsibilities for Latitude Studio application development

Configuring Latitude Studio, and building Latitude Studio applications, involves the following roles and responsibilities.

Role	Responsibilities
Power user	• Understanding the use cases and business case, so they can assemble Latitude Studio components to match these cases. • Building application pages. • Creating charts. • Configuring the controls for the various components.
System administrator	• Deploying and maintaining the server that will host the Latitude Studio application. • Connecting Latitude Studio to enterprise systems. • Setting up Latitude Studio logging. • Establishing and maintaining the Latitude Studio connection to its underlying database. • Managing ongoing clustering and load balancing. • Deploying extension points provided by the developer.

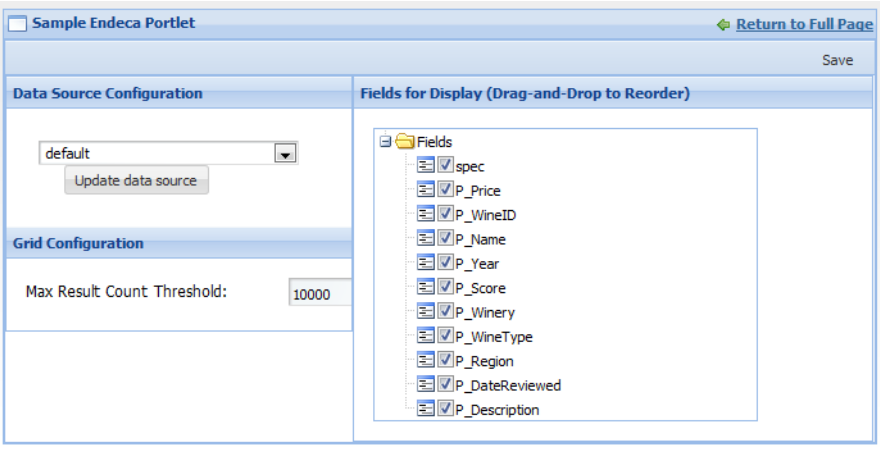
Role	Responsibilities
Developer	• Modifying existing components, or creating new ones, to better meet an organization's unique business requirements. • Developing Latitude Studio extension points for items such as security.
Designer	• Creating new themes.

About plugins and developing custom components

Plugins are `.war` files that allow you to add functionality to Latitude Studio. Custom components, themes, and layout templates are all considered plugins.

Because plugins are separate from the core application, they can be developed and deployed without disrupting application performance.

Latitude Studio includes a Component SDK that component developers can use to modify or create portlet components. The Sample Endeca Portlet component provides a starting point for creating custom components.



For more information about the Component SDK, see the *Latitude Developer's Guide*.

Obtaining more information about the Endeca MDEX Engine

This guide assumes that you already have experience working with the Endeca MDEX Engine.

This includes having a solid understanding of the Endeca Latitude Query Language (LQL) and other forms of data manipulation.

For more information about the MDEX Engine, consult the documentation for the version you are using. The documentation is available in the Knowledge Base of the Endeca Developer's Network (EDeN).

Where to find Liferay Portal documentation

For more information about how to perform various administration and customization tasks, you can see Liferay's documentation.

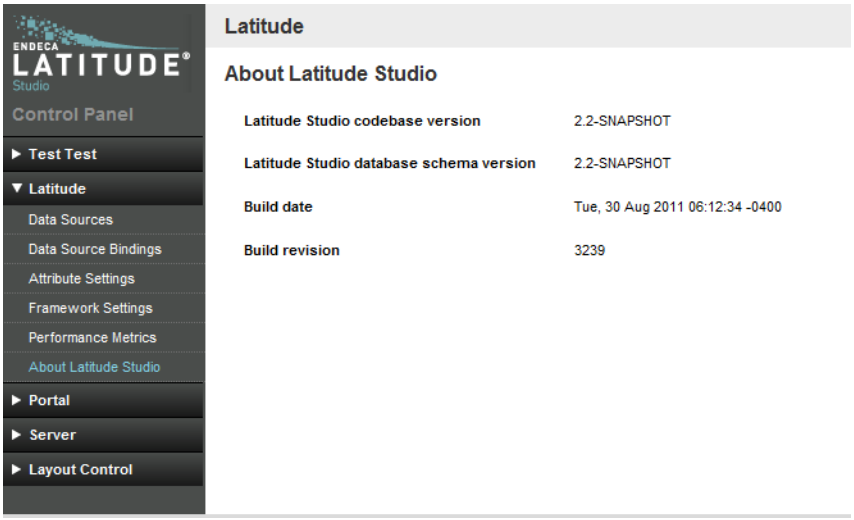
Specifically, the *Liferay Portal Administrator's Guide* provides extensive information about installing, configuring, and maintaining a portal.

You can download the guide from the **Documentation** section of the Liferay web site (`www.liferay.com`). It is also packaged with the Latitude 2.2 documentation on EDeN.

Liferay also offers developer assistance in the form of blogs, wikis, and forums. These resources are available from the **Community** section of the Liferay web site.

Getting the Latitude Studio build and version

The **About Latitude Studio** component on the **Control Panel** provides build and version information for the application.



Chapter 2

Overview of the Latitude Studio User Interface

About the Latitude Studio user interface

The look and feel of a Latitude Studio application varies according to the product version as well as to any customization performed by the application's developers. There are, however, some common elements.

A typical Latitude Studio application includes the following main controls and work areas:

Number	Item	Description
1	Dock	The Dock gives users access to function areas such as: • **Control Panel**, for administrative functions • **Add Component**, for adding Latitude and Liferay components to the page • **Layout Template**, for selecting the page layout • **Manage Pages**, for administering pages
2	Add Page	**Add Page** allows power users to add pages to the application's interface.
3	Pages	The tabbed pages are added by power users or application developers, and contain the various application components. Placing the components on several pages instead of on a single page provides a more useful navigation experience for end users.
4	Portal workspace	The portal workspace is where users access the components. In the example, the **Guided Navigation** component provides end users with contextual navigation across the application's data set. The **Results Table** component shows the records returned by the end user's navigation.

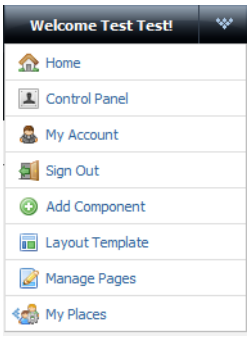
Dock menu functions

The Liferay Portal Dock, located in the upper right corner of the application window, contains the main menu for the portal application. It is typically labeled "Welcome, <*user name*>!"

To display the application menu, hover the mouse over the Dock.

If you are not logged in, then only the **Home** and **Sign In** links are available.

After you log in, the remaining options become available. The available options are based on the permissions for your user role.



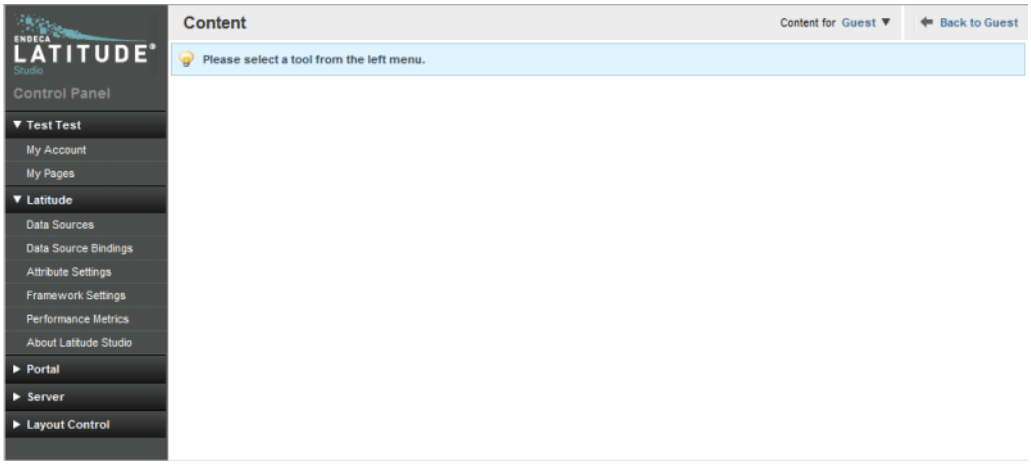
The Dock menu can include the following options:

Dock Link	Purpose
Home	Takes users to the home page.
Control Panel	Provides access to administrative functions.
My Account	Lets all users edit their user information, including changing their login password.
Sign Out	Logs users out of the application.
Add Component	Allows power users to add components to an application.
Layout Template	Allows power users to apply a different layout to a page.
Manage Pages	Allows power users to manage the application pages, including: • Adding and deleting pages • Changing the page display order • Performing other page administration functions
My Places	Displays the community and organization pages.

Control Panel functions

The Liferay Portal **Control Panel**, available from the Dock, contains several tools for system administrators and power users.

The left menu is grouped into sections, each containing a number of user and administrator tool components. Each section can be expanded or collapsed.



User name section

This is the logged-in user's personal space, which contains the following functions:

My Account	Allows you to edit your user information, including changing your password.
My Pages	Allows you to edit your public and private pages.

Latitude section

Power users most commonly use the **Control Panel** to get access to these administrative components:

Data Sources	Used to view and reload data sources.
Data Source Bindings	Used to associate data sources with components.
Attribute Settings	Used to create and modify attribute groups, and configure attributes.
Framework Settings	Provides access to state, security, and other settings.
Performance Metrics	Displays information about component and MDEX Engine query performance.
About Latitude Studio	Provides build and version information for Latitude Studio.

Portal section

The **Portal** section is intended for portal administrators to manage the user community. The administrative components in this section are:

Users	Used to add and modify user accounts.
Organizations	Used to create and edit Organizations, which are hierarchical collections of users.
Communities	Used to create and edit Communities, which are collections of users who have a common interest.
User Groups	Used to create and edit User Groups, which are simple, arbitrary collections of users.

Roles	Used to manage user roles, which are used to define permissions across their scope. The types of roles are: • Portal role - grants access across the portal. • Community role - grants access only within a single Community. • Organization role - grants access only within a single Organization. The portal administrator can use these roles to control permissions in the application, such as who is able to see specific pages.
Password Policies	Sets the password rules, such as the password strength and the password expiration.
Settings	Contains most of the global portal settings, including authentication, e-mail configuration, and the default landing page.
Monitoring	If monitoring is enabled, monitors all the live sessions in the portal.
Plugins Configuration	Used to configure which portal roles have access to the plugins.

Server section

Server Administration	Allows you to perform administrative tasks for the overall portal server, including: • Resource management • Setting log levels • Migrating data between databases • Configuring the e-mail server • Shutting down the Liferay Portal server
Portal Instances	Allows you to add and configure multiple portal instances on a single server.
Plugins Installation	Allows you to view the installed plugins and add new ones.

Layout Control section

Web Content	Allows you to add Web content, structures, templates, and RSS feeds.
Links	Allows you to add folder links.

Chapter 3

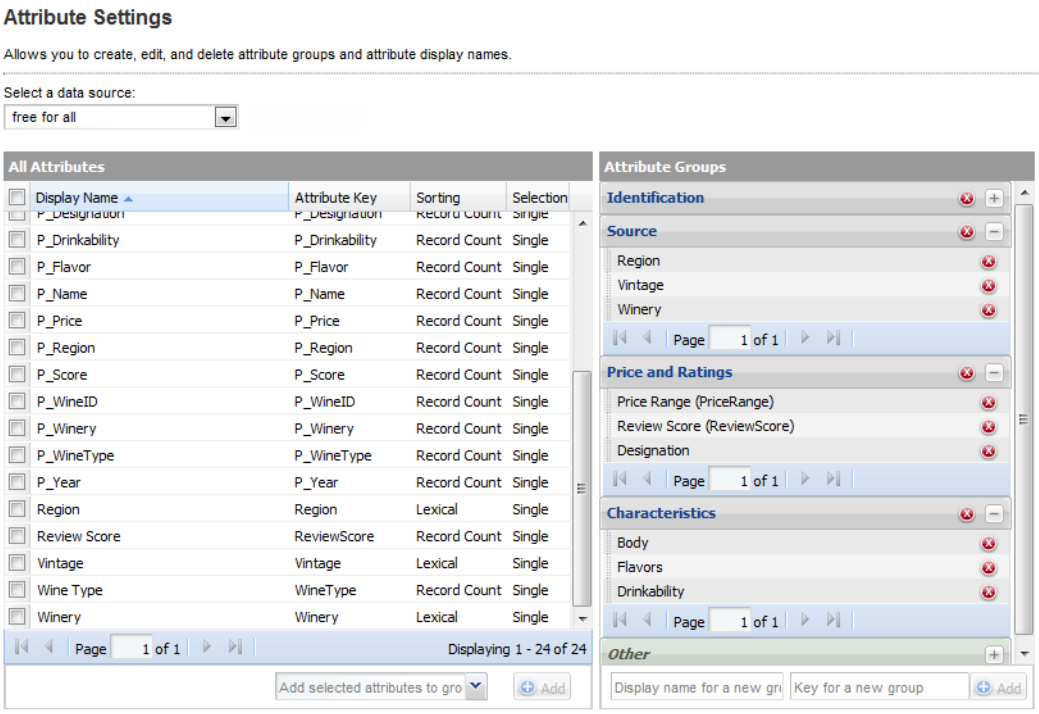
Summary of Latitude Studio Components

Administrative components

The administrative components, available from the **Control Panel**, are primarily used by administrators for higher-level configuration and monitoring of Latitude Studio.

Attribute Settings

Used to configure the attributes and attribute groups within each data source.



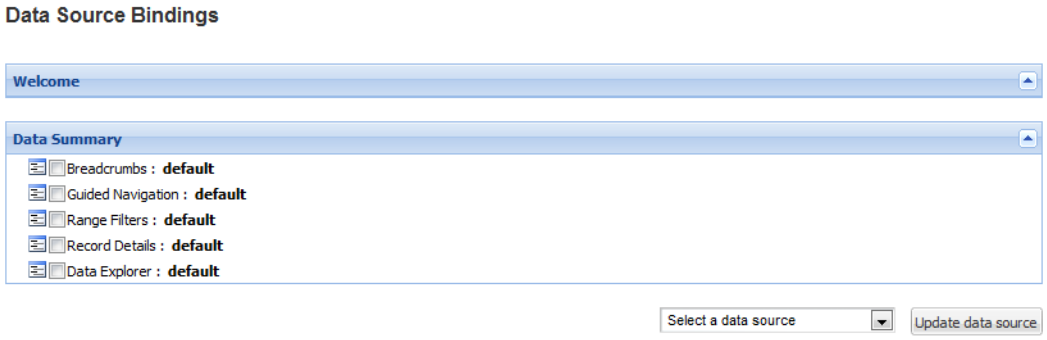
Using the **Attribute Settings** component, power users can:

- Create and remove attribute groups
- Add attributes to attribute groups
- Remove attributes from attribute groups
- Configure attribute display names

- Configure the attribute value sorting and selection options for the **Guided Navigation** component

Data Source Bindings

Used to assign a data source to one or more of the components on the Latitude Studio application pages.

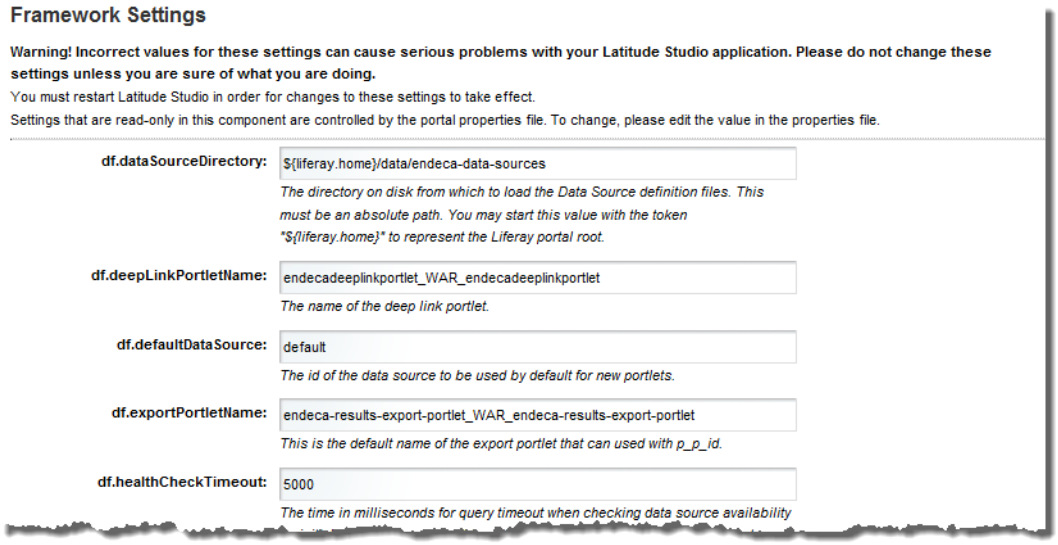


Data Sources

Contains the current list of data sources, including the update information and whether the data source is available. Used to test and reload the configured data sources for Latitude Studio.

Framework Settings

Provides access to several configuration settings for the Latitude Studio application, including state and security options.

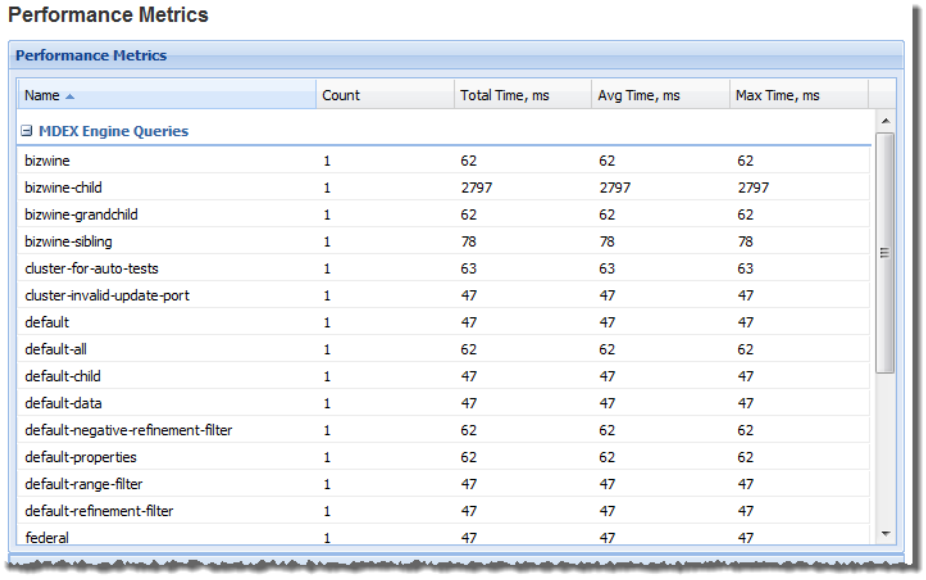


Performance Metrics

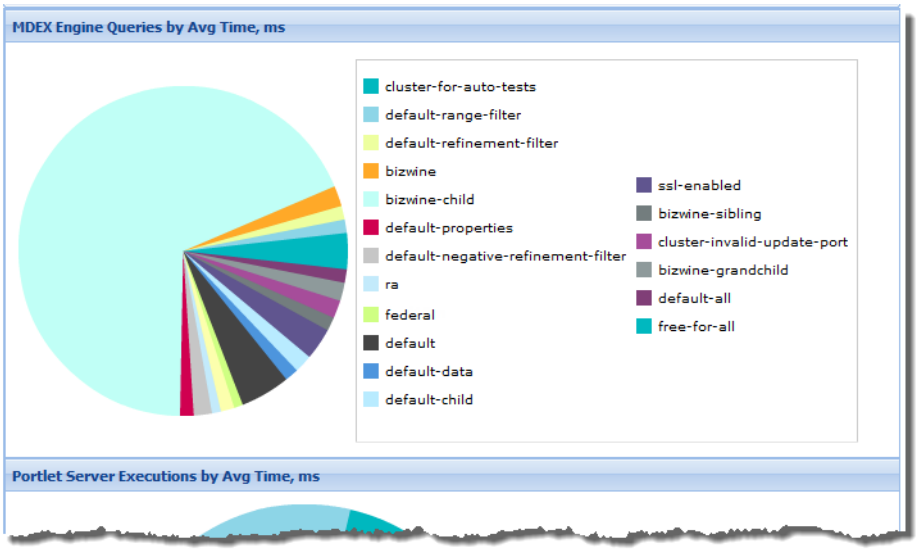
Provides insight into the number of queries performed by Latitude Studio, and the time taken to complete those queries. Contains data from the `LatitudeStudio-metrics.log` file. The component can be configured to include information for:

- MDEX Engine queries
- Portlet server executions
- Portlet client executions

At the top of the component is a table summarizing the query information.

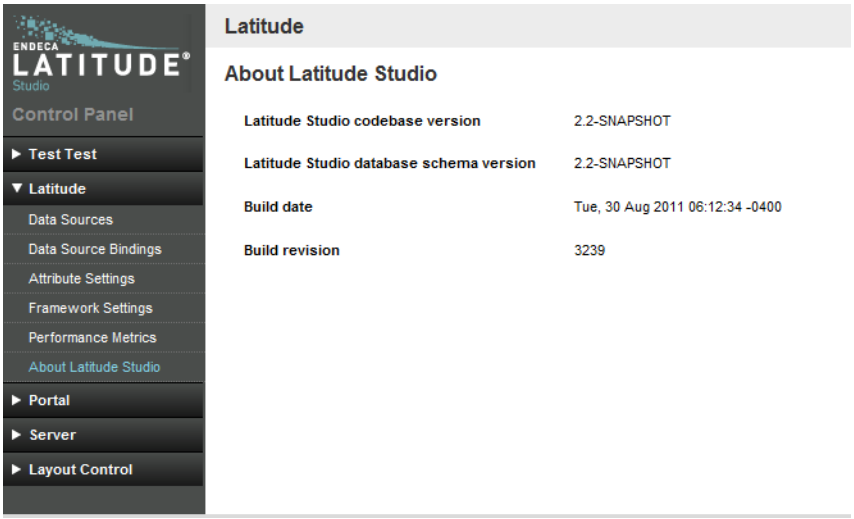


The table is followed by a set of pie charts to provide a visual summary of the performance information.



About Latitude Studio

Provides version and build information for the Latitude Studio application.

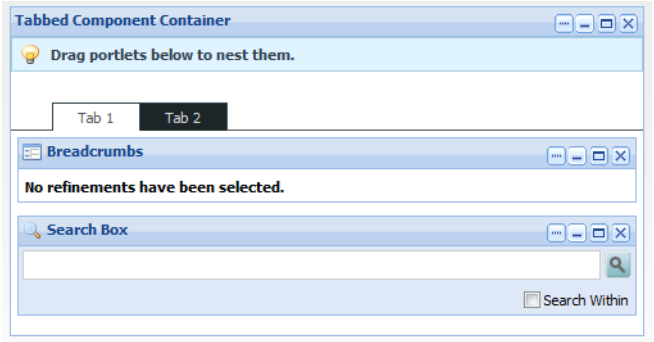


Layout components

Layout components, available from the **Layout** section of the **Add Component** dialog, allow power users to group other components on the page.

Tabbed Component Container

Used to create a tabbed interface. Each tab can contain a different set of components.



For the **Tabbed Component Container**, power users determine:

- The number of tabs to display
- The tab display order
- The label on each tab
- The layout of each tab
- Whether to display the component border around the container

Results components

Results components, available from the **Latitude** section of the **Add Component** dialog, allow end users to view lists of records from a data source.

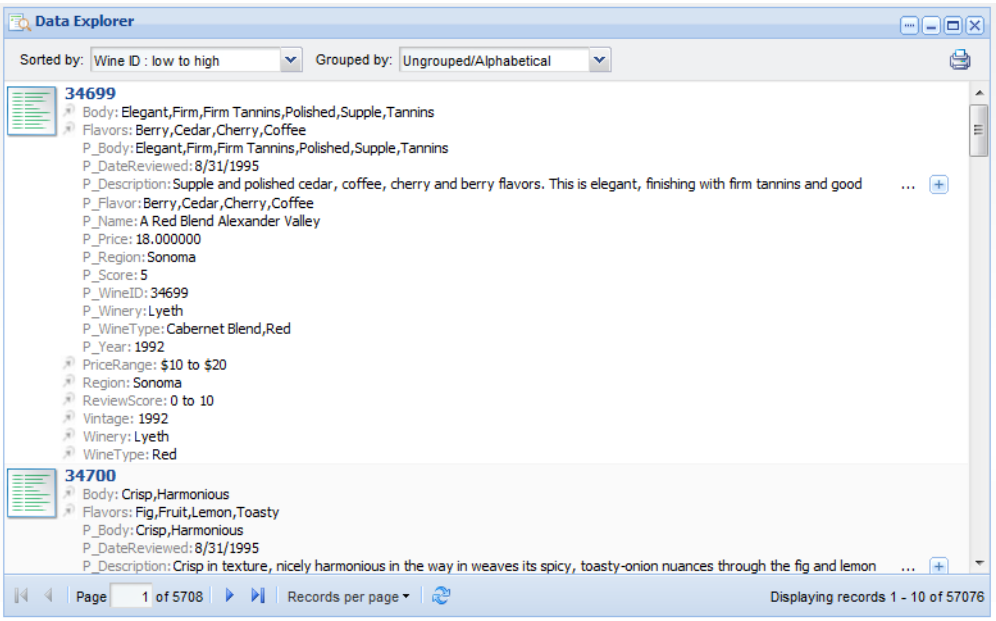
Data Explorer

Displays each record in the current refinement as a complete set of attribute-value pairs. The attributes can be displayed:

- In alphabetical order
- By data type
- Within their attribute groups

The **Data Explorer** component is designed to allow power users or data administrators to verify newly loaded data.

Users can print the list.

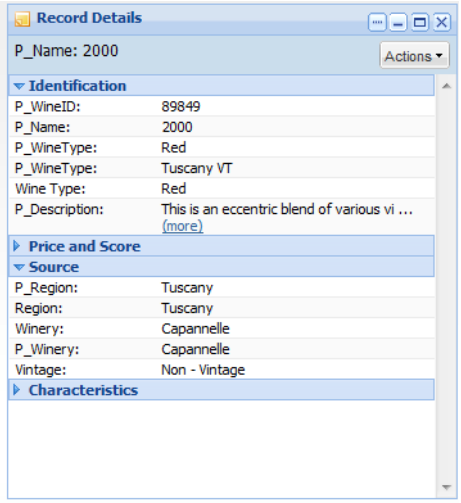


For the **Data Explorer** component, power users can configure:

- The data source
- The default grouping for the attributes
- The pagination and sorting options for the list

Record Details

Displays a set of attribute values for a selected record from a **Data Explorer**, **Results List**, **Results Table**, or **Compare** component.



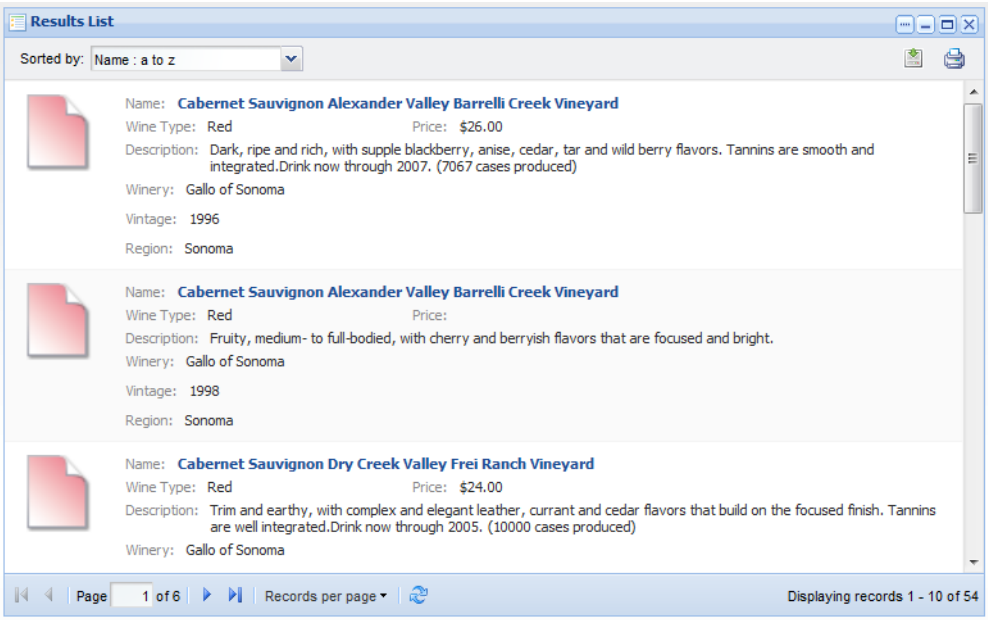
For the **Record Details** component, power users can configure:

- The data source
- The available actions
- The attribute groups to include

Results List

Displays the list of records for the current refinement. The list format is similar to regular Web search results. Each record contains a selected set of attributes.

End users can export and print the list. They also may be able to use the attributes to refine the data set or display related content.



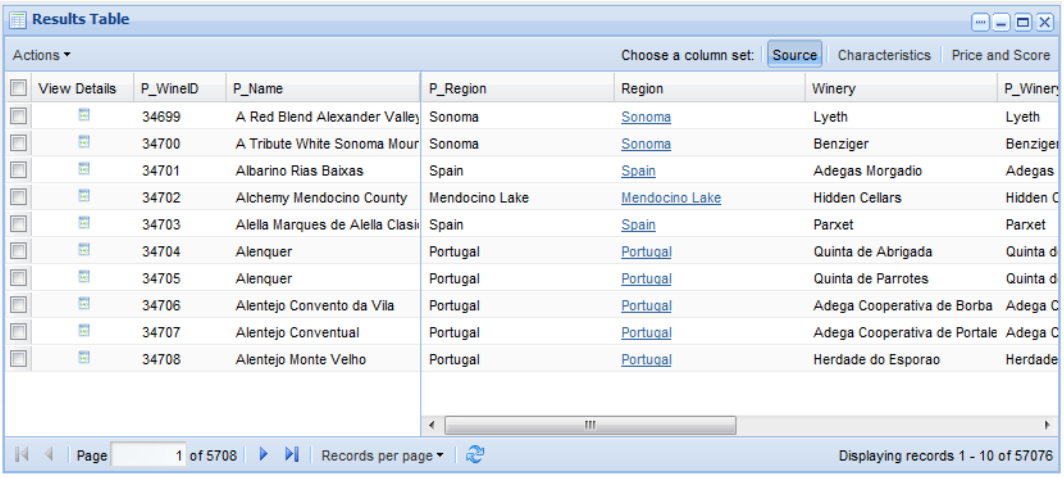
For the **Results List** component, power users can configure:

- The data source
- The pagination and sorting options for the list
- The attributes to include for each record, including whether to use the attributes for refinement or to display related content
- The images to display next to each record

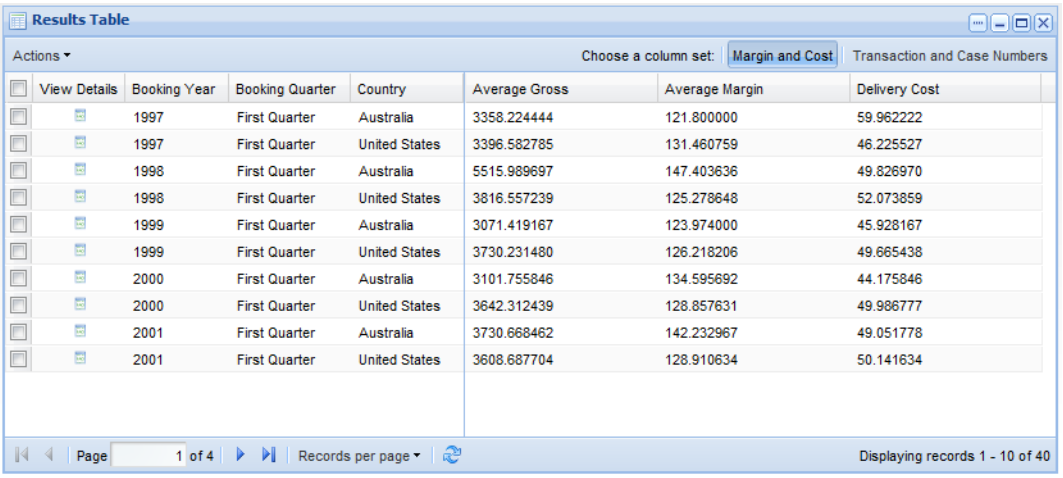
Results Table

Displays a table containing either:

- A list of records for the current refinement



- A set of metrics generated from an LQL query



End users can export and print the list. For a records-based table, they also may be able to use a **Compare** component to compare selected records.

The table can also include links to display a **Record Details** component, refine the data set, or display related content.

For the **Results Table** component, power users can configure:

- The data source
- An LQL query to return a set of metrics
- The groups of columns to display in the table
- The pagination and sorting options for the table
- The available actions for the table

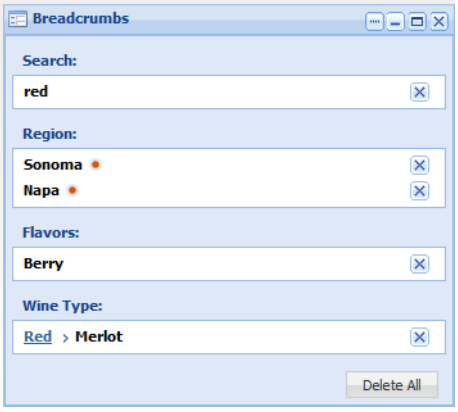
Filtering components

Filtering components, available from the **Latitude** section of the **Add Component** dialog, allow end users to search, navigate, and filter the data displayed in other components.

Breadcrumbs

Tracks the attributes and search terms that an end user has used for filtering.

End users can then remove items from the list of current filters.

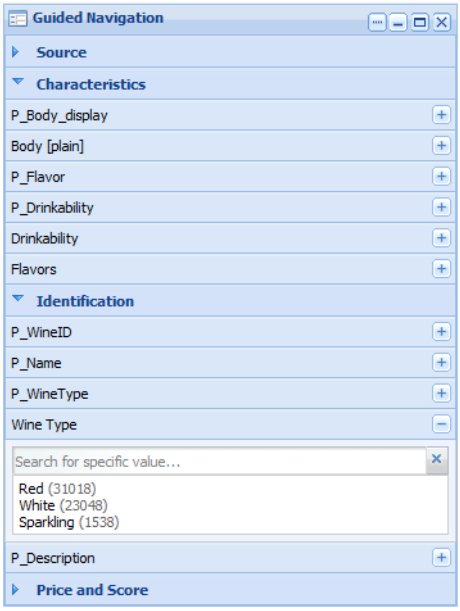


For the **Breadcrumbs** component, power users can configure:

- The data source to use
- Whether to allow end users to collapse lists of multiple values selected from the same attribute

Guided Navigation

Allows end users to filter data to only include specific attribute values. The selected values are added to the **Breadcrumbs** component.



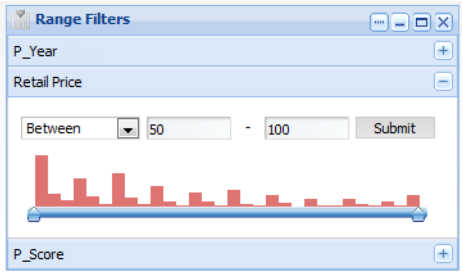
For the **Guided Navigation** component, power users can configure:

- The data source
- The attribute groups to include
- Whether to allow negative refinements, where end users search for records that do NOT have a selected value
- The number of values to display

In addition to the information in this guide, you also can find on EDeN short videos about using and configuring the **Guided Navigation** component.

Range Filters

Allows end users to filter data by specifying a range of numeric or date values. When end users apply a range filter, it is added to the **Breadcrumbs** component.



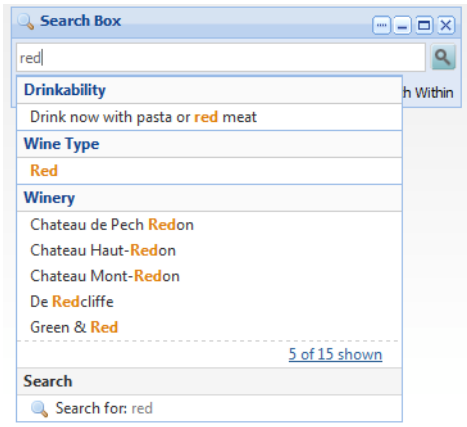
For the **Range Filters** component, power users can configure:

- The data source
- The attribute to use
- The minimum and maximum values for the range
- For numeric values, whether to show the distribution of values within the range

Search Box

Allows end users to filter data using a search. When end users perform a search, the search term is added to the **Breadcrumbs** component.

As end users type, they may be prompted to select a matching attribute value, or just search for the entered text.



For the **Search Box** component, power users can configure:

- The available search configurations. Each configuration can have a different data source.
- Whether to provide type-ahead suggestions
- The rule for determining a match

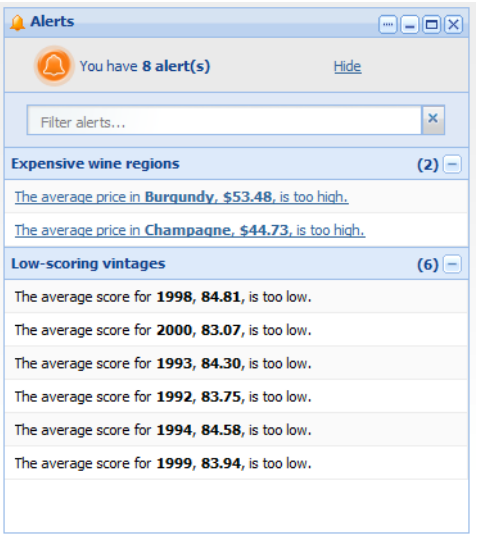
In addition to the information in this guide, you also can find on EDeN short videos about using and configuring the **Search Box** component.

Data Visualization components

Data Visualization components, available from the **Latitude** section of the **Add Component** dialog, provide more detailed or alternate views of the data.

Alerts

Displays messages to users to highlight values for those records. For example, alerts can be used to flag attribute values that fall outside of a specific range.



For the **Alerts** component, power users can configure:

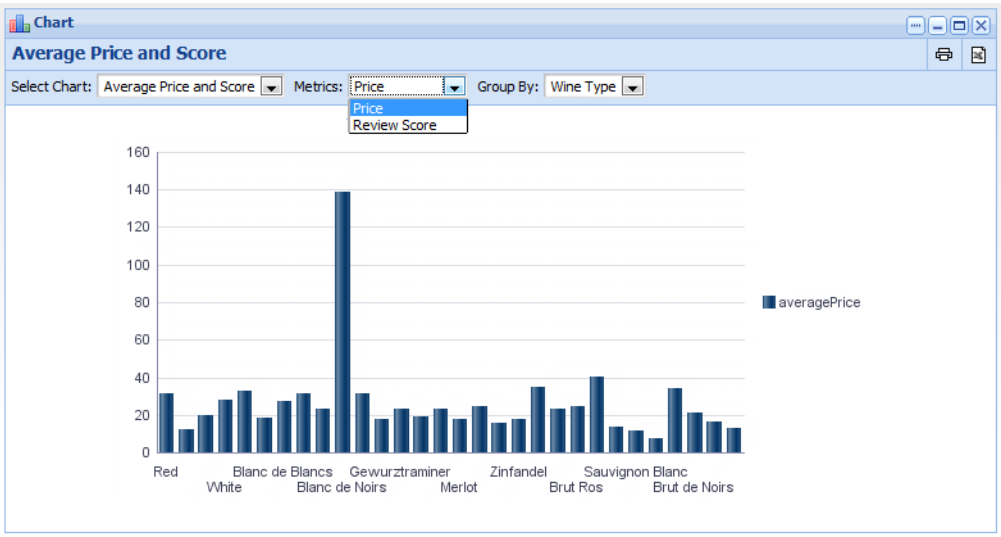
- Whether end users can use an alert to refine the data
- The data source for each alert
- The LQL query for each alert
- The text of the alert messages

Chart

Displays data using standard chart formats, including:

- Bar charts
- Line charts
- Pie charts

A single **Chart** component can contain multiple charts. End users also may be able to select different metrics values or grouping attributes in order to change the chart display.

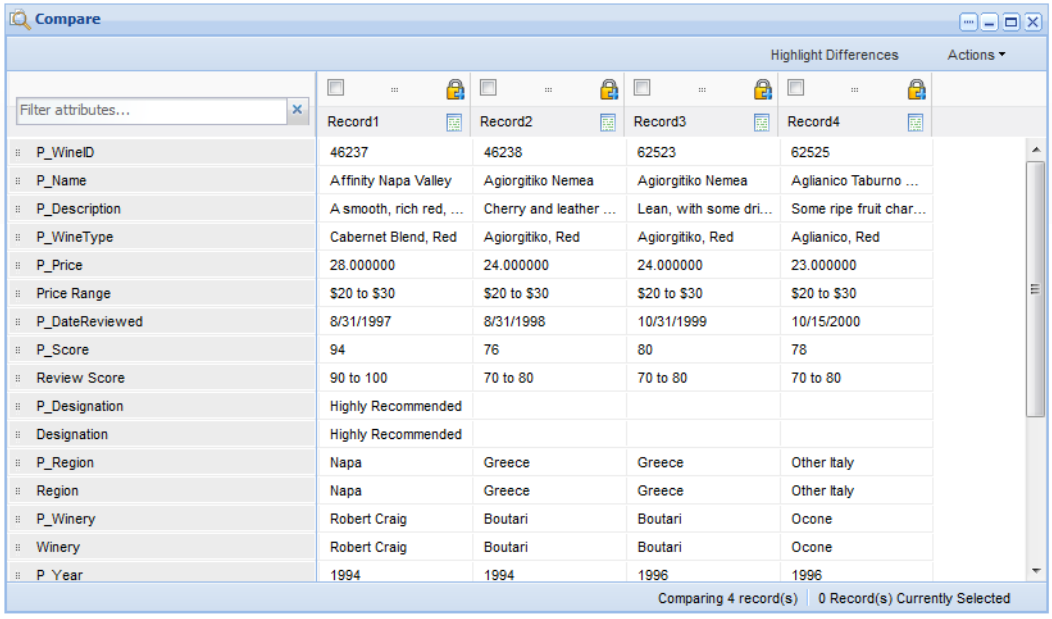


For the **Chart** component, power users can configure:

- The data source
- The chart style and labels
- The LQL query for the chart data
- Whether end users can change the chart metrics and grouping

Compare

Allows end users to do a side-by-side comparison of attribute values for records selected from a **Results Table** component.



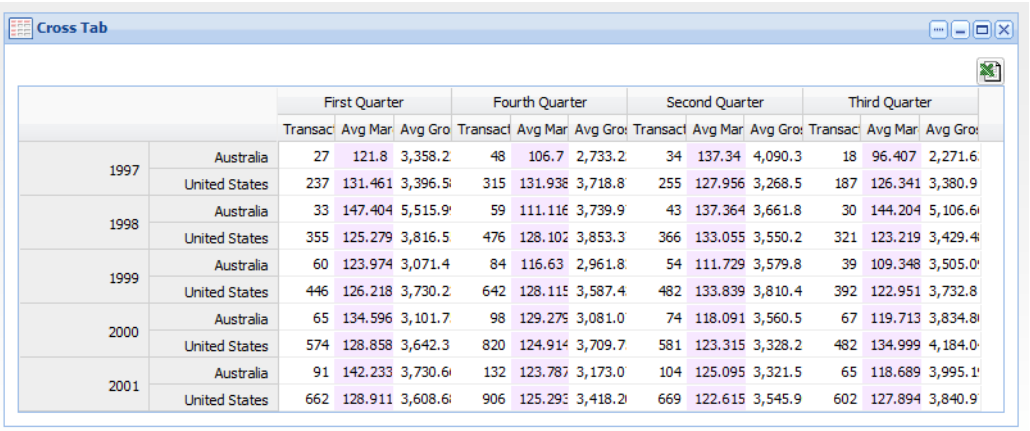
For the **Compare** component, power users can configure:

- The data source
- The attributes to include in the comparison
- Whether to allow highlighting to emphasize different values

Cross Tab

Generates a table that allows end users to perform comparisons and identify trends across several cross sections of data.

End users can also export the data to a spreadsheet.

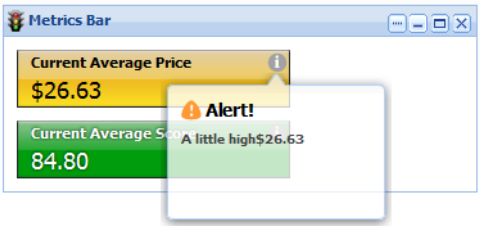


For the **Cross Tab** component, power users can configure:

- The data source
- The LQL query to generate the metrics and groupings
- The display of the metrics and groupings
- The dimensions of the table and window

Metrics Bar

Provides a quick view of metrics that summarize various aspects of the data. End users can display additional messages related to the metric value.



For the **Metrics Bar** component, power users can configure:

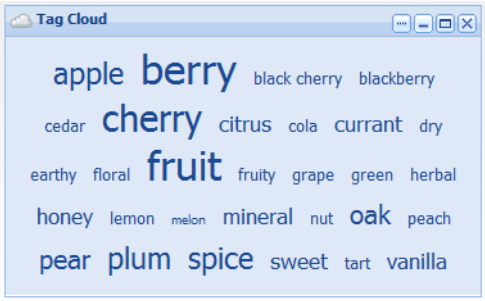
- The data source
- The LQL queries to generate the metrics
- The display format for each metric
- The message to display for the metric based on the current value

In addition to the information in this guide, you also can find on EDeN short videos about using and configuring the **Metrics Bar** component.

Tag Cloud

Displays the distribution of text values for an attribute in the current data. Values that occur more frequently are in larger and bolder type.

End users may be able to use the values to refine the data.



For the **Tag Cloud** component, power users can configure:

- The data source
- The attribute to use
- The number of values to display
- Whether end users can use the values to refine the data

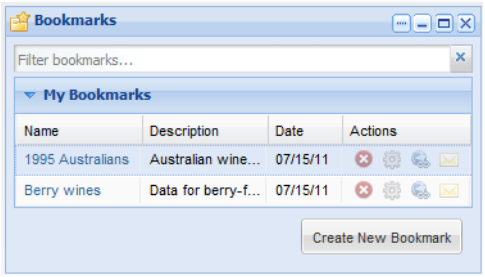
Personalization components

Personalization components, available from the **Latitude** section of the **Add Component** dialog, allow end users to customize the display of a Latitude Studio application.

Bookmarks

Allows end users to save the current state of the Latitude Studio application in order to return to it at a later time.

They may also be able to create links to and email bookmarks, to allow other users to see the same view of the data.



For the **Bookmarks** component, power users control whether end users can:

- Create bookmarks
- View and manage the current list of bookmarks

Part 2

Configuring and Monitoring Latitude Studio

Chapter 4

Configuring Framework Settings

About the Framework Settings component

The **Framework Settings** component on the **Control Panel** provides access to state, security, and other settings.

Note: If you do not see the **Framework Settings** in the **Control Panel**, it probably means you did not install the `endeca-framework-settings-portlet-<version>.war` file. Please review your installation settings.

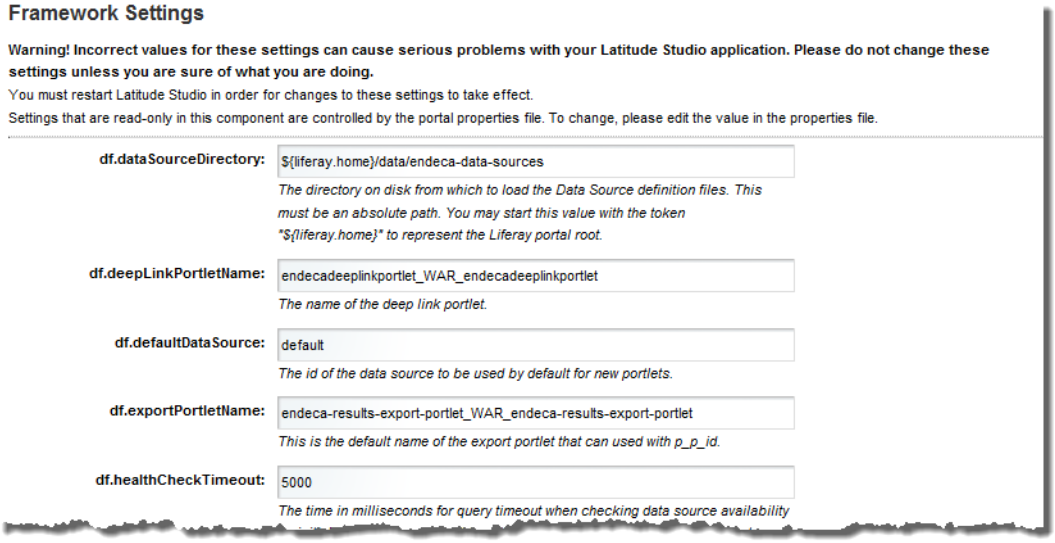
The default values of these settings are created automatically upon first use. You can only edit the settings. You cannot add or delete settings.

Settings only appear after the feature(s) that use them have been executed at least once.

Settings that have been configured in `portal-ext.properties` are displayed, but cannot be edited.

Configuring Framework Settings from the Control Panel

You use the fields on the **Framework Settings** page to modify the settings. You cannot modify settings that already have been configured in `portal-ext.properties`. If a setting has been configured directly in the `portal-ext.properties` file, then the field on the **Framework Settings** page is locked.



The **Framework Settings** component contains the following settings:

Framework Setting	Meaning
df.dataSourceDirectory	The directory on disk from which to load data source definition files.
df.deepLinkPortletName	The name of the deep link component.
df.defaultDataSource	The name of the data source to use as the default.
df.defaultExporter	The default exporter class.
df.exportPortletName	The default name of the export portlet.
df.healthCheckTimeout	The time (in milliseconds) for query timeout when checking data source availability on initialization.
df.maxExportAnalyticsRecords	The maximum allowable number of LQL records that can be exported.
df.maxExportBaseErrors	The maximum allowable number of non-LQL records that can be exported.
df.mdexCacheManager	The fully-qualified class name to use for the MDEX Cache Manager. Changing this setting is currently experimental and unsupported, and should be used only for research purposes. This interface will change in upcoming releases.
df.mdexSecurityManager	The fully-qualified class name to use for the MDEX Security Manager.
df.mdexStateManager	The fully-qualified class name to use for the MDEX State Manager.

Framework Setting	Meaning
df.metadataCacheEnabled	If set to `false`, retrieves this metadata directly from the data source whenever required (this will be slow). This setting will take effect for new sessions. Note that the use of this setting in a production environment is not supported.
df.viewTransitionDefaultContext	When power users specify a transition from a component to a different page, this is the default context path used to get to that page. The default value is `/web/guest/`, which is the default guest area of Latitude Studio. If you create your application within a Liferay Portal community, then you would specify the relative path to that community. For example, if you created your application within a community called My Community, you would set this setting to `/web/my-community/`. By setting the path here, you do not have to include it in the target page field in the individual component settings.
df.wsConnectionTimeout	The time in milliseconds before the connection to the MDEX Engine web services times out. The default value is 300000. If you experience frequent timeouts, then you may need to adjust this setting, or check your hardware configuration.

On the **Framework Settings** page, to change a setting:

1. Provide a new value in the setting configuration field.

   Note: Take care when modifying these settings, as incorrect values can cause problems with your Latitude Studio application.

   If the setting is configured in `portal-ext.properties`, then you cannot change the setting from this page. You must set it in the file.

2. Click **Update Settings**.
3. To apply the changes, restart Latitude Studio.

Configuring Framework Settings in portal-ext.properties

By default, you configure settings from the **Framework Settings** page. You also can add one or more of the settings to the `endeca-portal\portal-ext.properties` file.

Configuring settings in `portal-ext.properties` makes it easier to migrate settings across different environments. For example, after testing the settings in a development system, you can simply copy

the properties file to the production system, instead of having to reset the production settings manually from the **Control Panel**.

In the file, the format for adding a setting is:

```
<settingname>=<value>
```

Where:

- *<settingname>* is the name of the setting from the **Framework Settings** page.
- *<value>* is the value of the setting.

For example, to set the default data source in the file, the entry would be:

```
df.defaultDataSource=mydefault
```

If a property is configured in `portal-ext.properties`, you cannot edit it from the **Control Panel**. The field on the **Framework Settings** page is read only.

To move the configuration for a setting to the properties file after Latitude Studio has been started:

1. Stop the server.
2. Add the setting to `portal-ext.properties`.
3. Restart Latitude Studio.

   On the **Framework Settings** page of the **Control Panel**, the setting is now read only. You can no longer edit the value from the **Control Panel**.

Chapter 5

Configuring Logging for Latitude Studio

About logging in Latitude Studio

Latitude Studio uses the Apache log4j logging utility.

The Latitude Studio log files include:

- A main log file with most of the logging messages
- A second log file for performance metrics logging.

You can also use the **Performance Metrics** component to view performance metrics information.

About the log4j configuration XML files

The primary log configuration is managed in `portal-log4j.xml`, which is packed inside the portal application file `WEB-INF/lib/portal-impl.jar`.

To override settings in `portal-log4j.xml`, you use the file `portal-log4j-ext.xml`, which is located in the portal application's `/WEB-INF/classes/META-INF/` directory.

Both files are in the standard log4j XML configuration format, and allow you to:

- Create and modify appenders
- Bind appenders to loggers
- Adjust the log verbosity of different classes/packages

By default, `portal-log4j-ext.xml` specifies a log verbosity of INFO for the following packages:

- `com.endeca`
- `com.endeca.portal.metadata`
- `com.endeca.portal.instrumentation`

It does not override any of the default log verbosity settings for non-Latitude components.

Note: If you adjust the logging verbosity, it is updated for both log4j and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

About the Latitude Studio log files

For Latitude Studio, one log file contains all of the log messages, and a second file is used only for metrics logging.

About the main Latitude Studio log file

In the Latitude Studio log file configuration, the main root logger prints all messages to:

- The console, which typically is redirected to the application server's output log (For Tomcat, `catalina.out` and for WAS, `SystemOut.log`)
- A file called `LatitudeStudio.log`

The main logger does not print messages from the `com.endeca.portal.instrumentation` classes. Those messages are printed to the metrics log file.

Location of LatitudeStudio.log

By default, the logger tries to create `LatitudeStudio.log` in the following directory:

Tomcat:	The default location of `LatitudeStudio.log` is: *<value of catalina.home>*`/logs` For example, if `catalina.home` is set to `C:\endeca-portal\tomcat-6.0.29`, then `LatitudeStudio.log` is located in: `C:\endeca-portal\tomcat-6.0.29\logs`
WAS 7:	The default location of `LatitudeStudio.log` is either the value of `SERVER_LOG_ROOT` or the value of `LOG_ROOT`. 1. If `SERVER_LOG_ROOT` is defined, then the file is located in: *<value of SERVER_LOG_ROOT>* `SERVER_LOG_ROOT` might be set to something like `/usr/local/WAS/AppServer/profiles/AppSrv01/logs/server1` 2. If `SERVER_LOG_ROOT` is not defined, then the file is located in: *<value of LOG_ROOT>* `LOG_ROOT` might be set to something like `/usr/local/WAS/AppServer/profiles/AppSrv01/logs`

If the logger can't place the file in the default directory, then you typically can find `LatitudeStudio.log` in one of the following locations:

Tomcat - startup script:	If you started Tomcat by running a startup script, the log file is located where the script was run. For example, if you ran the startup script from `tomcat-<version>/bin`, the log file also is in `tomcat-<version>/bin`.

Tomcat - Windows service:	If you registered and started Tomcat as a Windows service, the log file may be in `C:\Windows\System32` or `C:\Windows\SysWOW64`.
Tomcat - Eclipse server:	If Tomcat is a server inside of Eclipse, the log files may be located in the root of the Eclipse directory.
WAS 7	For WAS 7, the log file is located relative to the profile's working directory. For example, `/usr/local/WAS/AppServer/profiles/AppSrv01`.

Specifying an absolute path to the log file

The custom appender used for Latitude Studio logging contains the logic for determining the log file directory.

```
<appender name="FILE" class="com.endeca.portal.util.ContainerAwareRolling¬
FileAppender">
 <param name="File" value="LatitudeStudio.log"/>
 <param name="MaxFileSize" value="10MB"/>
 <param name="MaxBackupIndex" value="10"/>
 <layout class="org.apache.log4j.PatternLayout">
  <param name="ConversionPattern" value="%d{ISO8601} %-5p [%c{1}] %m%n" />
</layout>
</appender>
```

If you want to specify an absolute file path for the log files, then you will need to update the appender configuration to use the regular log4j appender class.

In `portal-log4j-ext.xml`, make the following changes to the appender configuration:

1. Change the class to be `org.apache.log4j.RollingFileAppender`.
2. Change the value of the `File` parameter to specify the full path to the file.

For example:

```
<appender name="FILE" class="org.apache.log4j.RollingFileAppender">
 <param name="File" value="C:\endeca-portal\logs\LatitudeStudio.log"/>
 <param name="MaxFileSize" value="10MB"/>
 <param name="MaxBackupIndex" value="10"/>
 <layout class="org.apache.log4j.PatternLayout">
  <param name="ConversionPattern" value="%d{ISO8601} %-5p [%c{1}] %m%n" />
</layout>
</appender>
```

About metrics logging

An additional file appender captures metrics logging, including all log entries from the `com.endeca.portal.instrumentation` classes.

The metrics log file, `LatitudeStudio-metrics.log`, is in the same directory as `LatitudeStudio.log`.

You also can view metrics data on the **Performance Metrics** component.

For details on metrics logging, see *Monitoring the Performance of Queries* on page 49.

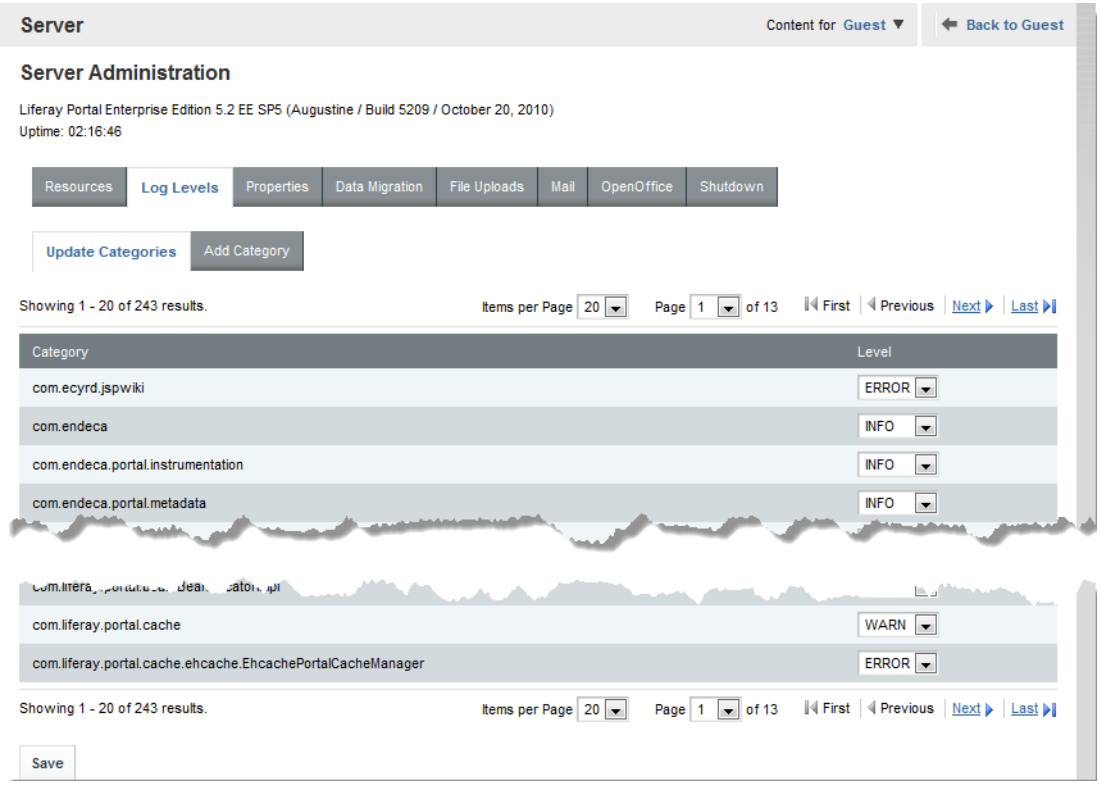
Using the Control Panel to adjust logging verbosity

For debugging purposes in a development environment, you can use the **Control Panel** to dynamically adjust logging verbosity levels for any class hierarchy.

Note: When you adjust the logging verbosity, it is updated for both `log4j` and the Java Utility Logging Implementation (JULI). Code using either of these loggers should respect this configuration.

To adjust logging verbosity from the **Control Panel**:

1. In Latitude Studio, point the cursor at the Dock.
2. From the drop-down menu, choose **Control Panel**.
3. From the **Control Panel** menu, choose **Server Administration**.
4. In the **Server Administration** page, click the **Log Levels** tab.



5. On the **Update Categories** tab, locate the class hierarchy you want to modify.
6. From the logging level drop-down list, select the logging level.

Note: When you modify a class hierarchy, all classes that fall under that class hierarchy also are changed.

7. When you have finished adjusting log levels, click **Save**.

You also can use the **Add Category** tab to set the verbosity for a specific class or package.

How log4j.properties is used in Latitude Studio

In Latitude Studio, a `log4j.properties` configuration file is used to start up the Tomcat bundle. Otherwise, all log4j configuration comes from the XML configuration files.

Using log4j.properties when starting up the Tomcat bundle

The version of `log4j.properties` in `common/endorsed/log4j.properties.jar` is used to configure logging for the Tomcat bundle. Because `log4j` is initialized before Latitude Studio in the Tomcat bundle, this file is needed to ensure that there is some preliminary `log4j` configuration.

This version of `log4j.properties` provides minimal configuration, so that the initial messages are logged to the console in the same format as the default configuration in `portal-log4j-ext.xml`. Its settings only affect a small number of messages printed as the server is starting.

When Latitude Studio starts, it loads its XML configuration file, which overrides the settings in `log4j.properties`. Therefore administrators should not need to modify `log4j.properties`.

Other log4j.properties files

In addition to the `log4j.properties` file used for the Tomcat startup, all deployed portlets, as well as the Latitude Studio application itself, have their own `log4j.properties` files, located in `WEB-INF/classes`.

Because Latitude Studio uses XML configuration files, the settings in these files have no effect.

Additional sources for information on log4j

For more information about `log4j`, see the Apache web site and the Liferay documentation. .

The *Apache log4j site* provides general information about and documentation for `log4j`.

The *Liferay documentation*, including the *Liferay Portal Administrator's Guide*, provides more information about `log4j` logging in Liferay.

Chapter 6

Monitoring the Performance of Queries

You can get access to performance metrics data both from the metrics log file and from the **Performance Metrics** component. A setting in `portal-ext.properties` controls the amount of metrics data to record.

Configuring the amount of metrics data to record

To configure the metrics you want to include, you use a setting in `portal-ext.properties`. This setting applies to both the `LatitudeStudio-metrics.log` file and the **Performance Metrics** component.

The metrics logging can include:

- MDEX Engine queries by data source
- Portlet server executions by component. The server side code is written in Java.

  It handles configuration updates, configuration persistence, and MDEX queries. The server side code generates results to send back to the client side code.

  Server executions include portlet render, resource, and action requests.
- Portlet client executions by component. The client side code lives in the browser and is written in JavaScript.

  It issues requests to the server code, then renders the results as HTML. The client code also handles any dynamic events within the browser.

By default, only the MDEX Engine queries and portlet server executions are included.

You use the `df.performanceLogging` setting in `portal-ext.properties` to configure the metrics to include. The setting is:

```
df.performanceLogging=<metrics to include>
```

Where <*metrics to include*> is a comma-separated list of the metrics to include. The available values to include in the list are:

`QUERY`	If this value is included, then the component includes information for MDEX queries.
`PORTLET`	If this value is included, then the component includes information on portlet server executions.

`CLIENT`	If this value is included, then the component includes information on portlet client executions.

In the default configuration, where only the MDEX queries and portlet server executions are included, the value is:

```
df.performanceLogging=QUERY,PORTLET
```

To include all of the available metrics, you would add the `CLIENT` option:

```
df.performanceLogging=QUERY,PORTLET,CLIENT
```

Note that for performance reasons, this configuration is not recommended.

If you make the value empty, then the log file and **Performance Metrics** component also are empty.

```
df.performanceLogging=
```

About the metrics log file

The `LatitudeStudio-metrics.log` file contains the metrics logging information. It is located in the same directory as the `LatitudeStudio.log` file.

The metrics log file contains the following columns:

Column Name	Description
Total duration (msec)	The total time for this entry (End time minus Start time).
Start time (msec since epoch)	The time when this entry started. For MDEX queries and server executions, uses the server's clock. For client executions, uses the client's clock.
End time (msec since epoch)	The time when this entry was finished. For MDEX queries and server executions, uses the server's clock. For client executions, uses the client's clock.
Session ID	The session ID for the client.
Page ID	If client instrumentation is enabled, the number of full page refreshes or actions the user has performed. Used to help determine how long it takes to load a complete page. Some actions that do not affect the overall state of a page, such as displaying attributes on a **Guided Navigation** component, do not increment this counter.
Portlet ID	This is the ID associated with an individual instance of a component. It generally includes: • The type of component • A unique identifier For example, if a page includes two **Chart** components, the ID will can be used to differentiate them.

Column Name	Description
Entry Type	The type of entry. For example: • `PORTLET_RENDER` - Server execution in response to a full refresh of a component • `DISCOVERY_SERVICE_QUERY` - MDEX query • `CLIENT` - Client side JavaScript execution • `PORTLET_RESOURCE` - Server side request for resources • `PORTLET_ACTION` - Server side request for an action
Miscellaneous	A URL encoded JSON object containing miscellaneous information about the entry.

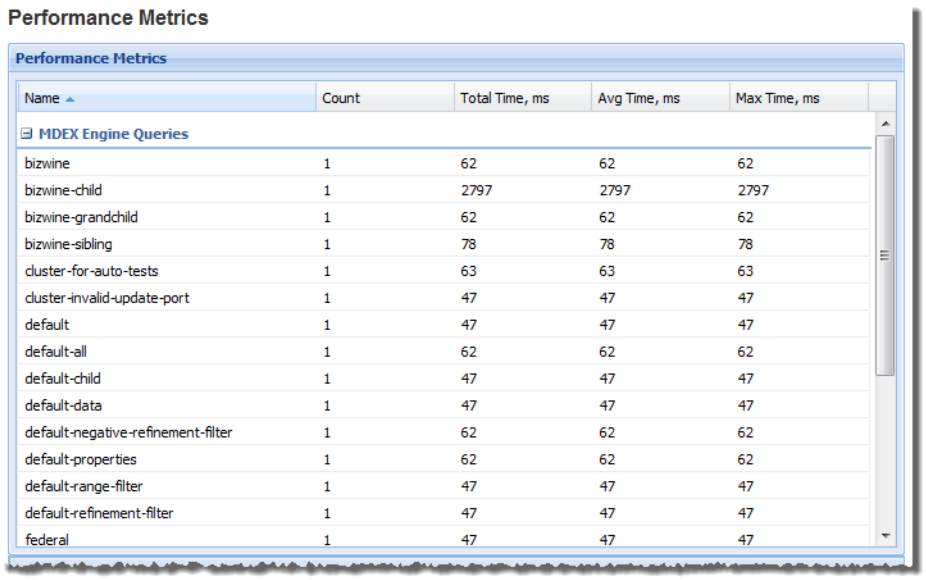
About the Performance Metrics component

The **Performance Metrics** component on the **Control Panel** displays information about component and MDEX Engine query performance.

It uses the same logging data recorded in `LatitudeStudio-metrics.log`.

However, unlike the log file, the **Performance Metrics** component uses data stored in memory. Restarting Latitude Studio clears the **Performance Metrics** data.

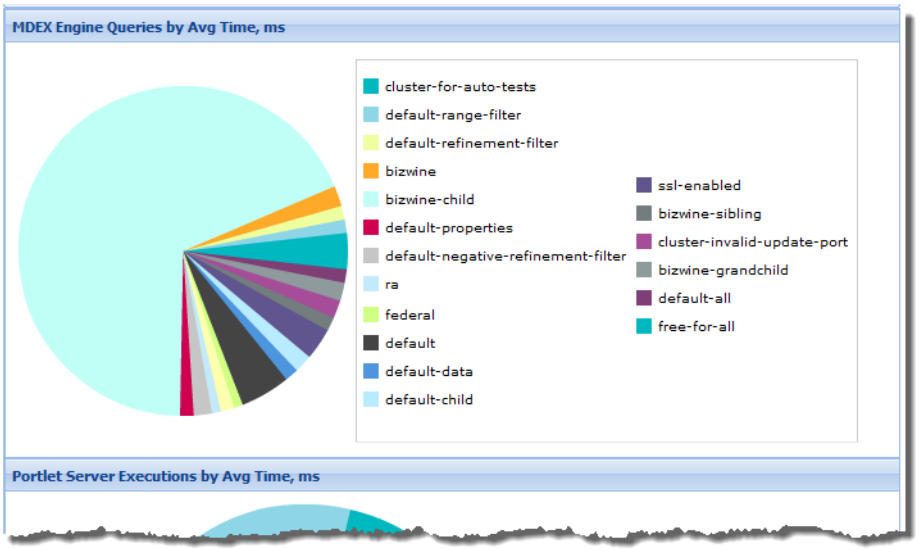
For each type of included metric, the table at the top of the component contains a collapsible section.



For each data source or component, the table tracks:

- Total number of queries or executions
- Total execution time
- Average execution time
- Maximum execution time

For each type of included metric, there is also a pie chart summarizing the average query or execution time per data source or component.



Note: MDEX Engine query performance does not correlate directly to a page, as a single page often uses multiple MDEX Engine queries.

Chapter 7

Controlling Access to Latitude Studio

In order to view and configure Latitude Studio applications, users must log in.

About managing users in Latitude Studio

Notes on the scope of this information

Latitude Studio uses Liferay's user and role management functions.

This section provides some basic information about configuring and managing users in Latitude Studio, including:

- User roles
- Creating and editing users
- Integrating with an LDAP system

This content assumes you are using the default organization and a single community. For additional details on Liferay's user management functions, see the *Liferay Administrator's Guide*.

How users get access to Latitude Studio

The **Welcome** page of a Latitude Studio application includes a **Sign In** component to allow users to log in to Latitude Studio.

Users can be created directly from within Latitude Studio. You also can integrate with an LDAP system to import users directly.

About user roles

In Latitude Studio, each user is assigned a user role. The user's role controls the functions that the user has access to.

While Latitude Studio comes with several roles, the roles you are most likely to use are:

Administrator	Administrators have full privileges. They can: • View Latitude Studio pages • Configure public pages and components

	• Use all of the **Control Panel** components
Power User	Power Users have fewer privileges. They can: • View Latitude Studio pages • Configure pages in communities for which they are an owner or administrator • Use the **Control Panel** to edit their account information They also have access to **Latitude** components on the **Control Panel**.
User	The User role is the role you would assign to your end users. Users can view Latitude Studio pages, and can use the **Control Panel** to edit their account information. They cannot configure Latitude Studio pages or components.
Guest	This is the role associated with users who have not logged in to Latitude Studio. Guest users can view pages, but do not have any access to the **Control Panel**.

If you are using multiple communities, then users can also be assigned roles for a specific community. For example, an administrator may be an administrator for a specific community, and not for the entire application.

About the default user

When you first install Latitude Studio, a default user is created.

The default user is an administrator and has full privileges to:

- View Latitude Studio pages
- Create and configure pages
- View and configure **Control Panel** settings

To log in as the default user, use the following user name and password:

Email address:	test@endeca.com
Password:	test

For better security, after logging in for the first time, you should probably either:

- Change the password for this user
- Create new administrative users for your instance, and then remove this user

Creating and editing users in Latitude Studio

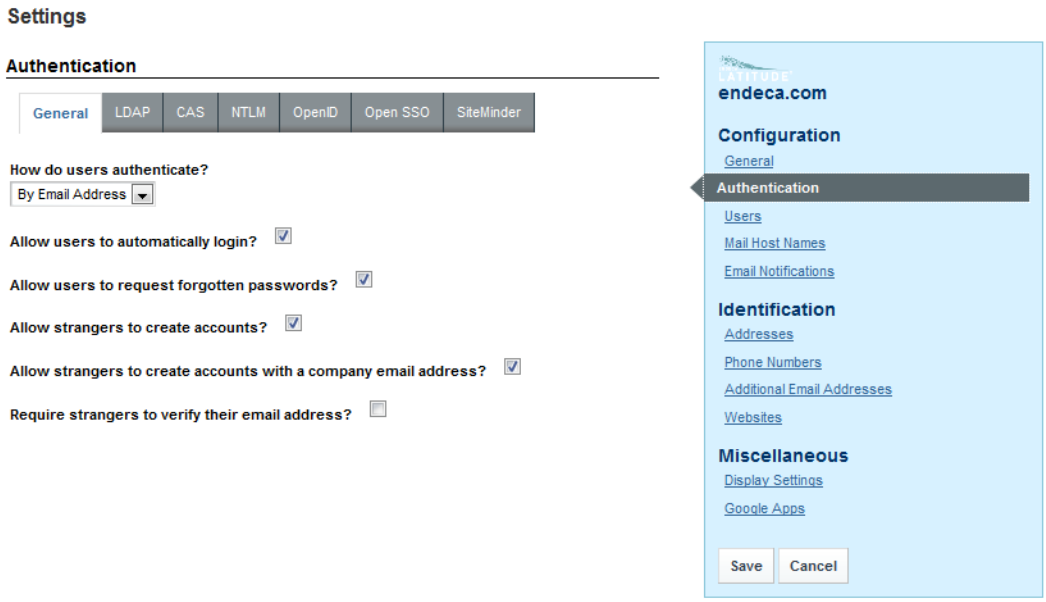
The **Users** component provides options for creating and editing Latitude Studio users.

Configuring the type of user name for Latitude Studio

Each Latitude Studio user has both an email address and a screen name. By default, users log in to Latitude Studio using their email address.

To change the configuration so that users log in with their screen name:

1. From the Dock menu, click **Control Panel**.
2. On the **Control Panel** menu, click **Settings**.
3. In the **Settings** page menu to the right, click **Authentication**.
4. On the **General** tab, select the name used to log in.



5. Click **Save**.

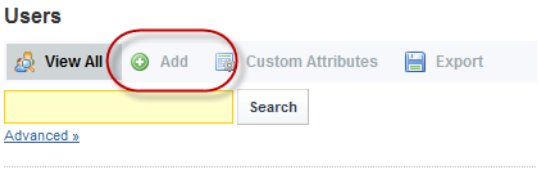
Creating a new user

Even if you are importing users from LDAP, you may still want to create a few users directly in Latitude Studio.

For example, for a small development instance, you may just need a few users to develop and test pages. Or if your LDAP users for a production site are all end users, you may need a separate user account for administering and editing the site.

To add a new user:

1. Display the **Users** component on the **Control Panel**.
   a) From the Dock menu, select **Control Panel**.
   b) On the **Control Panel**, click **Users.**
2. On the **Users** page, click **Add**.



3. On the **Details** page, to provide the minimum required information:
   a) In the **Screen Name** field, type the screen name for the user.
   b) In the **Email Address** field, type the user's email address.
   c) In the **First Name** field, type the user's first name.
   d) In the **Last Name** field, type the user's last name.
4. Click **Save**.

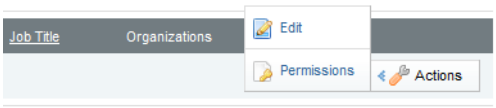
   The new user is added, and the configuration menu is updated to add the rest of the options.
5. To create the password for the user:
   a) On the user page, on the configuration menu to the right, click **Password**.
   b) On the **Password** page, enter the password to assign to the new user.
   c) Click **Save**.
6. To add the user to a community:
   a) On the user page, from the list to the right, click **Communities**.
   b) Click the **Select** link.
   c) On the communities list, click the community to add the user to.
7. To assign a role to the user:
   a) On the user page, from the list to the right, click **Roles.**
   b) Select the roles for the user.
   c) Click **Save**.
8. For the Power User role, in order for the user to be able to manage pages, they must be a Community Administrator or Community Owner. On the **Roles** page for the user:
   a) Under **Community Roles**, click the **Select** link.
   b) On the community roles list, click the role you want to assign to the user.
   c) Click **Save**.

Editing a Latitude Studio user

The `Users` component also allows you to edit a user's account.

From the **Users** component, to edit a user:

1. Click the **Actions** button next to the user.
2. In the **Actions** menu, click **Edit**.



3. After making your changes, click **Save**.

Integrating with an LDAP System to manage users

If you have an LDAP system, you can allow users to use those credentials to log in to Latitude Studio.

About using LDAP

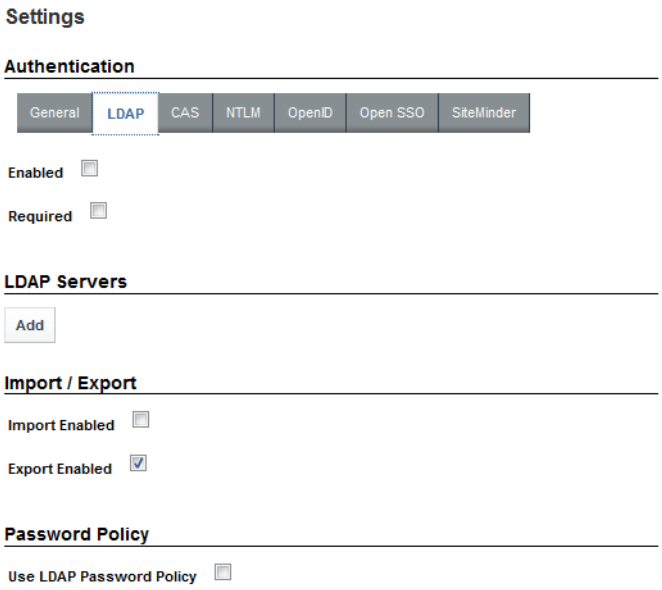
LDAP (Lightweight Directory Access Protocol) allows you to have users connect to your Latitude Studio application using their existing LDAP user accounts, rather than creating separate user accounts from within Latitude Studio.

Configuring the basic LDAP settings

You configure the LDAP connection from the Control Panel. The basic settings include whether LDAP is enabled, and whether to import or export LDAP settings.

To display the LDAP configuration page and configure the basic settings:

1. From the Dock menu, click **Control Panel**.
2. On the **Control Panel** menu, click **Settings**.
3. In the **Settings** page menu to the right, click **Authentication**.
4. Click the **LDAP** tab.



5. On the **LDAP** tab:
   a) To enable LDAP authentication, check the **Enabled** checkbox.
   b) To only allow users to log in using an LDAP account, check the **Required** checkbox.

      If this box is checked, then any users that you create manually in Latitude Studio cannot log in.

      To make sure that users you create manually can log in, make sure that this box is not checked.

6. The LDAP Servers section lists the configured servers.

   For details on adding an LDAP server, see *Adding an LDAP server* on page 58.

7. The **Import/Export** section is used to configure importing and exporting of LDAP user data:
   a) If the **Import Enabled** checkbox is checked, then when you start Latitude Studio, it can import all of your LDAP groups and users.

      If the box is not checked, then Latitude Studio synchronizes each user as they log in.

      It is recommended that you leave this box unchecked.

   b) If the **Export Enabled** checkbox is checked, then any changes to the user in Latitude Studio are exported to the LDAP system.

      It is recommended that you uncheck this checkbox.

8. To save the LDAP settings, click **Save**.

Adding an LDAP server

On the **LDAP** tab for authentication settings, the **LDAP Servers** section lists the LDAP servers configured for this instance.

To add and configure your LDAP server:

1. Under **LDAP Servers**, to add a new server to the list, click the **Add** button.
2. In the **Server Name** field, type the name of the LDAP server.
3. To populate the rest of the fields with default values based on a specific type of server:
   a) Under **Default Values**, click the radio button for the type of server you are using.
   b) Click **Reset Values**.
4. The **Connection** settings cover the basic connection to LDAP:

Base Provider URL:	The location of your LDAP server. Make sure that the machine on which Liferay is installed can communicate with the LDAP server. If there is a firewall between the two systems, make sure that the appropriate ports are opened.
Base DN:	The Base Distinguished Name for your LDAP directory. For a commercial organization, it may look something like: `dc=companynamehere,dc=com`
Principal:	The user name of the administrator account for your LDAP system. This ID is used to synchronize user accounts to and from LDAP.
Credentials:	The password for the administrative user.

   After providing the connection information, to test the connection to the LDAP server, click the **Test LDAP Connection** button.

5. The **Users** section contains settings for finding users in your LDAP directory.

Authentication Search Filter:	Determines the search criteria for user logins. By default, users log in using their email address. If you have changed this setting, you must to modify the search filter here. For example, if you changed the authentication method to use the screen name, you would modify the search filter so that it can match the entered login name: `(cn=@screen_name@)`
Import Search Filter:	Depending on the LDAP server, there are different ways to identify the user. The default setting (`objectClass=inetOrgPerson`) usually is fine, but to search for only a subset of users or for users that have different object classes, you can change this.

6. Under **User Mapping**, map your LDAP attributes to the Latitude Studio user fields:

   You can map the following attributes:

   - Screen Name
   - Password
   - Email Address
   - Full Name
   - First Name
   - Middle Name
   - Last Name
   - Job Title
   - Group

   After setting up the attribute mappings, to test the mappings, click **Test LDAP Users**.

7. Under **Groups**, map your LDAP groups.

   In the **Import Search Filter** field, type the filter for finding LDAP groups, then map the following fields:

   - Group Name
   - Description
   - User

   To test the group mappings, click **Test LDAP Groups**. The system displays a list of the groups returned by your search filter.

8. To save the LDAP server, click **Save**.

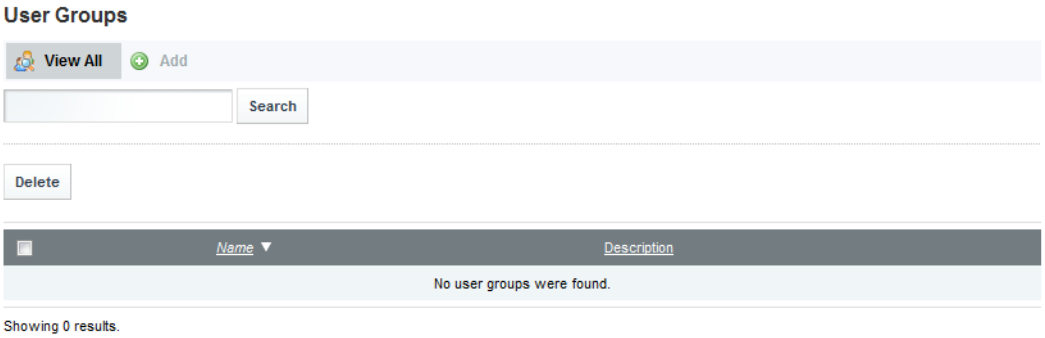
Assigning roles based on LDAP user groups

For LDAP integration, it is recommended that you assign roles based on your LDAP groups.

To ensure that users have the correct roles as soon as they log in, you create groups in Latitude Studio that have the same name as your LDAP groups, then assign the correct roles to each group.

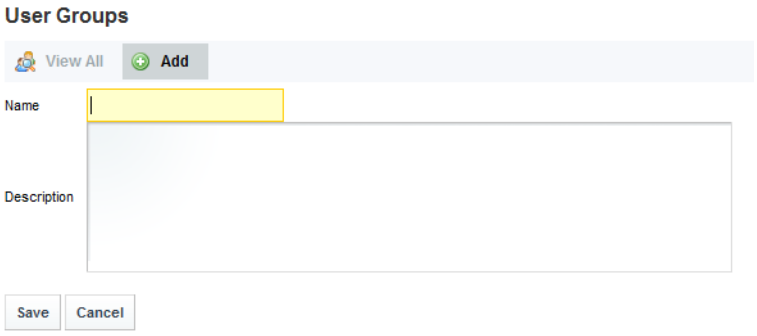
To create a group and then assign a role to that group:

1. Display the **Groups** component on the **Control Panel**.
   a) From the Dock menu, select **Control Panel**.
   b) On the **Control Panel**, click **User Groups**.



2. On the **User Groups** page, to add a new group:
   a) Click the **Add** button.

      The new group page is displayed.



   b) On the new group dialog, in the **Name** field, type the name of the group.

      Make sure the name matches the name of a group from your LDAP system.

   c) In the **Description** field, type a description of the group.
   d) Click **Save.**

      The group is added to the **User Groups** list.

3. To assign the group to a role:
   a) In the **Control Panel** menu, click the **Roles** option.
   b) On the **Roles** page, for the role you want to assign the group to, click the **Actions** button.
   c) In the menu, click **Assign Members**.
   d) Click the **User Groups** tab.
   e) To display the list of available groups to assign to the role, click the **Available** tab.
   f) Check the checkbox next to the group, then click the **Update Associations** button.

      The group is added to the **Current** tab as a group assigned that user role.

Part 3

Working with Data in Latitude Studio

Chapter 8

Working with Data Sources

About data sources

Every instance of a component that needs to query the MDEX Engine is backed by a selected data source. A data source represents a pointer to a specific MDEX Engine or load balancer, and is used to maintain the application state for each user's session.

Each data source is a .json file that contains:

- Connection information for the MDEX Engine or load balancer
- Optional filters to restrict the returned data

Each component in your Latitude Studio application can connect to a different data source.

Specifying a default data source

One of the data sources in your Latitude Studio application must be the default data source. When a data-source-backed component is added to a page, it automatically is bound to this data source.

When you initially start Latitude Studio, the `id` of the default data source is set to **default**. If you do not change this setting, you must include a data source that has an `id` of **default**.

When you install Latitude Studio, the `endeca-portal\data\endeca-data-sources` directory includes a `default.json` data source file, which has an implicit `id` of **default**. This file only includes host and port information.

To configure a different `id` for the default data source:

1. From the **Control Panel**, go to the **Framework Settings** page.
2. On the **Framework Settings** page, in the `df.defaultDataSource` field, type the `id` of the data source to use as the default.

   Make sure that there is an existing data source with that `id` value.
3. Restart **Latitude Studio**.

Changing the data source for a Latitude component

If more than one data source has been configured for the application, then you can select a different data source for an individual component.

From the edit view of a component, to select a different data source for that component:

1. From the data source drop-down list, select the new data source.
2. Click **Update data source**.

   A message displays indicating that the component has been bound to the new data source.

   The data-specific component configuration also is cleared.

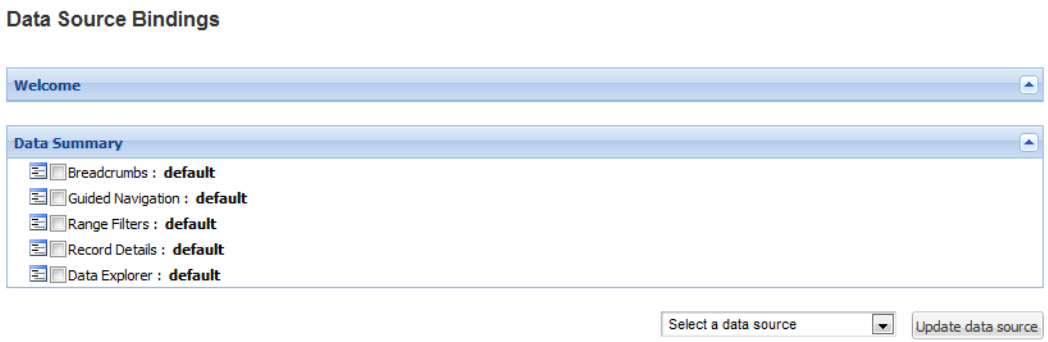
3. To exit the edit view, click **Return to Full Page**.

Changing the data source for multiple components at the same time

The **Data Source Bindings** component of the **Control Panel** allows power users to assign the same data source to multiple components, rather than having to update each component individually.

For example, you could add all your components to the page, then before you configure the components, use the **Data Source Bindings** component to set the data sources.

The component lists the data source selected for each component on each page.



On the **Data Source Bindings** component, to change the data source for multiple components:

1. Check the checkbox next to each component you want to update.
2. From the drop-down list, select the data source you want to use.
3. Click **Update data source**.

Configuring data sources

Latitude Studio includes sample data sources you can use as a basis for creating your own data sources. Each data source contains at minimum the connection information for the MDEX Engine.

Sample data sources provided with Latitude Studio

Latitude Studio ships with sample data sources. You can use these samples as models for your own data sources.

The sample data sources are located in the `endeca-portal\data\endeca-data-sources` directory:

- `base-filters-data-source.json.sample` - Provides an example of a data source with filters to restrict the returned data.
- `child-data-source.json.sample` - Provides an example of a child data source.
- `secured-data-source.json.sample` - Provides an example of a data source with security filters.
- `simple-data-source.json.sample` - Provides an example of a basic data source.

This directory also contains:

- `default.json` - Contains host and port information only. This is the default data source when you first install Latitude Studio.
- `default-schema.json` - Identical to `default.json`, except that it is configured to display configuration records only.

Adding data sources to Latitude Studio

You can add new data sources to Latitude Studio. Your Latitude components can then access those data sources.

To add a data source to Latitude Studio:

1. Create a new JSON file in `endeca-portal\data\endeca-data-sources`.

   For examples, see the sample data sources located in the same directory.

2. After creating the new file, either:

   - Restart Latitude Studio.
   - On the **Data Sources** component of the **Control Panel**, click **Update data sources**.

Note: If your data source does not appear, it probably means that your data source definition file contains invalid JSON syntax. To confirm this, look for a message about invalid syntax in the Latitude Studio log `df.log`. Check the log, edit your syntax, and then try the steps above again.

Basic data source syntax

Data source files are written as a JSON string. Each data source must contain the MDEX Engine connection settings, and can optionally contain identification and description information.

Connection settings (required)

Every data source must at a minimum contain the following MDEX Engine connection settings:

`server`	The name of the server on which the MDEX Engine is running.

`port`	The port on which the MDEX Engine is listening.

Here is an example of the most basic data source configuration, with just the connection information.

```
{
 "server":"server01.lab.acme.com",
 "port":"15000"
}
```

Identification and description settings

The data source can optionally contain the following identification and description settings:

`id`	The identifier of the data source. Used internally. The `id` value cannot contain spaces. If you do not provide a value for `id`, then it is set to the name of the data source file. For example, if the data source file is `parts.json`, and you do not provide a value for `id`, then `id` is set to `parts`. If you do not provide a value for `id`, make sure that the data source file name does not contain any spaces.
`name`	The name of the data source. This is the value displayed on the Latitude Studio UI, including on the **Data Sources** component and in data source drop-down lists. If you do not provide a value for `name`, then it is set to the name of the data source file.
`description`	A longer description of the data source. Used for logging and debugging.

For example:

```
{
 "server":"server01.lab.acme.com",
 "port":"15000",
 "id":"wine1",
 "name":"Wine Transactions",
 "description":"Transaction data for the Midwest Region"
}
```

Adding filters to a data source

The `baseFunctions` setting in the data source file allows you to include filters in order to restrict the returned data.

The basic format for the `baseFunctions` setting is:

```
"baseFunctions": [
 {
  "class":"com.endeca.portal.data.functions.<class name>",
  "<property name>":"<property value>"
 }
       ]
```

Where:

- *<class name>* is the name of the filter class.
- *<property name>* is the name of a configuration property for the filter.
- *<property value>* is the value of the configuration property.

Latitude Studio comes with the following filters:

Class Name and Description	Configuration Properties
`NegativeRefinementFilter` Used to filter data to exclude records that have the provided attribute value.	`attributeValue`: String `attributeKey`: String `ancestors`: String[]
`RangeFilter` Used to filter data to include records with attribute values within the specified range.	`attributeKey`: String `rangeOperator`: `(LT | LTEQ |GT |GTEQ| BTWN |GCLT |GCGT | GCBTWN)` `value1`: numeric `value2`: numeric (optional) `value3`: numeric (optional)
`RecordFilter` This is the most commonly used type of filter. Can be configured to include multiple filters with Boolean logic.	`recordFilter`: String
`RefinementFilter` Used to filter data to include records with the specified attribute values.	`attributeValue`: long `attributeKey`: long `multiSelect`: `(AND |OR | NONE)` (optional) `navigable`: `(true|false)` (optional)
`SearchFilter` Used to filter the data to include records that have the provided search terms.	`searchInterface`: String (Either a search interface or an attribute enabled for text search) `terms`: String `matchMode`: `(ALL | PARTIAL | ANY | ALLANY | ALLPARTIAL | PARTIALMAX | BOOLEAN)` `enableSnippeting`: boolean (optional; default is `false`) `snippetLength`: int (required if `enableSnippeting` is `true`) To enable snippeting, set `enableSnippeting` to `true`, and provide a value for `snippetLength`.

Example data source with filters

The following example data source file contains a `RecordFilter` and a `RefinementFilter`. The data is filtered only include transaction records from the Midwest region for the year 1999.

```
{
 "server":"server01.lab.acme.com",
 "port":"15000",
 "id":"wine1",
 "name":"Wine Transactions",
 "description":"Transaction data for the Midwest Region",
 "baseFunctions": [
  {
   "class":"com.endeca.portal.data.functions.RecordFilter",
   "recordFilter":"Regions:Midwest"
  },
  {
   "class":"com.endeca.portal.data.functions.RefinementFilter",
   "attributeValue":"1999",
   "attributeKey":"Year"
  }
        ]
}
```

Configuring parent and child data sources

Latitude Studio data sources can have parents and children. A child data source is essentially a subset of its parent data source.

For example, a parent data source contains all of the transaction records for a company. A child data source may contain only transactions from the United States.

Child data sources may themselves have children. For example, the child data source containing United States transactions may have a child data source that contains only transactions for New York, or transactions processed by a specific distributor.

Effect of query state changes on parent and child data sources

Important: The interactions between data sources can vary based on the data source State Manager your portal is using. The information here only applies to the default implementation of the State Manager provided with Latitude Studio.

The current query state for a child data source is the current query state of the parent data source plus any filters from the child data source. For example, if the parent data source has been refined to only include transactions for 1999, then the child data source only includes transactions for the United States (the child data source filter) from 1999 (the parent data source refinement).

Changes to the query state for any data source are applied to the entire family hierarchy that the data source belongs to:

- Changes to the query state for a child data source are applied to its parent data source.

  For example, if an end user selects a refinement from a **Guided Navigation** component that is bound to a child data source, components bound to the parent data source also are updated with that refinement.

  If that parent data source is itself a child data source, the change is then applied to its parent as well.

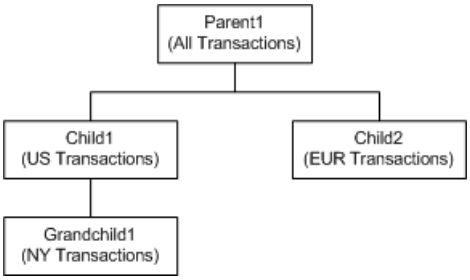
The operation is applied to each parent in turn until it reaches a data source that does not have a parent.

- Changes to the query state for a parent data source are applied to all of its children.

  If that child data source itself has children, the change is then applied to those children as well.

  The operation is applied to each child in turn until it reaches a data source that does not have any children.

For example, for the following data source hierarchy:



- If an end user refines Parent1 to only show transactions for 1999:
  1. Child1 and Child2 are refined to only show transactions for 1999, plus their original base filters.
  2. The refinement to Child1 causes Grandchild1 to be refined to only show transactions for 1999, plus its original base filters and the filters from Child1.
- If an end user refines Child1 to only show credit card transactions:
  1. Parent1 and Grandchild1 are refined to only show credit card transactions, plus their original filters.
  2. The refinement to Parent1 causes Child2 to be refined to only show credit card transactions, plus its original base filters.
- If an end user refines Grandchild1 to only show cash transactions:
  1. Child1 is refined to only show cash transactions, plus its original base filters.
  2. The refinement to Child1 causes Parent1 to be refined to only show cash transactions, plus any original filters.
  3. The refinement to Parent1 causes Child2 to be refined to only show cash transactions, plus its original base filters.

Creating a child data source

A child data source has the same server and port settings as its parent data source.

To identify the data source as a child data source, and also identify its parent, you add the following setting:

`parentDataSource`	The `id` of the parent data source.

You then configure `baseFunctions` setting with the filters for the child data source.

Example of a child data source

In the following example of a child data source, the ID of the parent data source is `all-transactions`.

This `importer-transactions` child data source includes data from `all-transactions`, filtered to only include records for the Importer supplier type.

```
{
   "server":"server01.lab.acme.com",
   "port":"5555",
   "id":"importer-transactions"
   "name":"Importer Transactions"
   "parentDataSource":"all-transactions",
   "baseFunctions": [
        {"class":"com.endeca.portal.data.functions.RecordFilter",
         "recordFilter":"Supplier_Types:Importer"
        }
   ]
}
```

The sample data source `child-data-source.json.sample` provides a template for defining a child data source.

Configuring role-based security for data sources

You can also configure a data source to restrict access to data based on user roles.

The data source settings related to role-based security are:

`securityEnabled`	Whether to enable the security filters for queries to the data source. If set to `"true"`, then the data source uses the filters configured under `securityFilters`.
`securityFilters`	Defines all of the security filters to be used by the data source.  **Note:** Record filters are the only supported type of `securityFilter`.
`rolePermissions`	Maps the user roles to the security filters. For each mapping, the format is: `"<role name>" : "<filterName>"` where: • *<role name>* is the name of the user role . • *<filterName>* is the name of the security filter.
`inheritSecurity`	For a child data source, whether the data source should inherit the security filters configured in its ancestor data sources. If set to `"true"`, then the list of security filters to apply includes both: • All of the security filters for every ancestor data source • All of the security filters defined in the child data source

Example of a data source with security filtering

In the following example, users with the role "French Wine" can only see data from the Bordeaux and Burgundy regions.

```
{
 "server":"server01.lab.acme.com",
 "port":"15000",
 "id":"ds-id",
 "name":"Descriptive DataSource name",
 "description":"Detailed information about this DataSource",
"securityEnabled": "true",
"inheritSecurity": "true",
"securityFilters": {
     "frenchFilter": {
           "class":"com.endeca.portal.data.functions.RecordFilter",
           "recordFilter":"OR(Region:Bordeaux,Region:Burgundy)"
     }
},
"rolePermissions": {
     "French Wine": ["frenchFilter"]
}
}
```

The sample data source `secured-data-source.json.sample` provides a template for configuring a secured data source.

Connecting to a secured MDEX Engine

You can set up Latitude Studio to connect to a secured (HTTPS) MDEX Engine.

Several of the steps below refer to the *Latitude Administrator's Guide*. Before attempting these steps, make sure you have a copy of that guide at hand.

Note: The steps below assume you are using the Latitude Studio Tomcat bundle.

1. Stop Latitude Studio.
2. Use the `enecerts` utility to generate the SSL certificate files for the Dgraph.

   For instructions, see the *Latitude Administrator's Guide*.

3. Generate the Java KeyStore (JKS) files.

   For instructions, see the *Latitude Administrator's Guide*.

4. Place the JKS keys into the directory containing the JSON data source files.

   This is typically, but not always, the `endeca-portal\data\endeca-data-sources` directory.

5. In the appropriate JSON data source file, add the `sslConfig` setting, which contains the following settings:

`caFile`	The name of the truststore file.
`caPassword`	The password for the truststore file.
`certFile`	The name of the keystore file.
`certPassword`	The password for the keystore file.

For example:

```
"sslConfig": {
  "caFile":"truststore.ks",
  "caPassword":"pwd1",
  "certFile":"keystore.ks",
  "certPassword":"pwd2"
 }
```

Note: SSL configuration file paths are relative to the directory containing the JSON data source files. This is typically, but not always, the `endeca-portal\data\endeca-data-sources` directory.

6. Restart Latitude Studio.

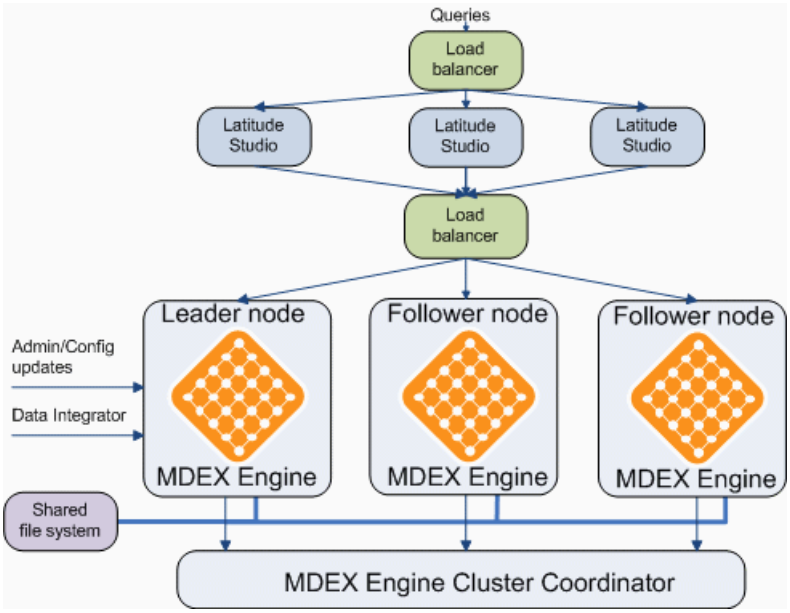
Example of a data source connected to secured MDEX Engine

The following data source connects to a secured MDEX Engine.

```
{
 "server":"server01.lab.acme.com",
 "port":"15000",
 "sslConfig": {
  "caFile":"truststore.ks",
  "caPassword":"pwd1",
  "certFile":"keystore.ks",
  "certPassword":"pwd2"
 }
 "id":"ds-id",
 "name":"Descriptive DataSource name",
 "description":"Detailed information about this DataSource",
 "baseFunctions": [
  {
   "class":"com.endeca.portal.data.functions.RecordFilter",
   "recordFilter":"Regions:Midwest"
  },
  {
   "class":"com.endeca.portal.data.functions.RangeFilter",
   "property":"P_Price",
   "rangeOperator":"GTEQ",
   "value1": "25"
  },
  {
   "class":"com.endeca.portal.data.functions.RefinementFilter",
   "attributeValue":"123",
   "attributeKey":"121"
  }
        ]
}
```

Configuring a data source for a clustered environment

The MDEX Engine can use a clustered configuration with multiple nodes, where one node is the leader node, and the other nodes are follower nodes.



Latitude Studio communicates with the MDEX Engine cluster through a load balancer. The data source is configured to connect to the load balancer.

You also configure the data source to identify the leader node to apply any updates to. If you do not provide the update settings, then you cannot use the **Attribute Settings** component to manage the data source's attribute and group settings.

When you edit the data source's attribute and group settings from the **Attribute Settings** component:

1. The changes are first sent to the leader node.
2. The leader node then sends the changes to the follower nodes.

   Note that it may take a few minutes to propagate the changes to the follower nodes.

When configuring the data source for an MDEX Engine cluster, the main settings are:

`server`	The server for the load balancer.
`port`	The port number for the load balancer.
`updateServer`	The server for the leader node.
`updatePort`	The port number for the leader node.
`updateSslConfig`	Optional. If applicable, the SSL settings for connecting to the leader node. Includes the following settings: • `caFile` - The truststore file. • `caPassword` - The truststore password. • `certFile` - The name of the keystore file. • `certPassword` - The keystore password.

For example:

```
{
    "server":"loadbalancer1.acme.com",
    "port":"15000",
    "name":"Read-only data source",
```

```
    "updateServer":"leader1.acme.com",
    "updatePort":"18000",
    "updateSslConfig": {
        "caFile":"truststore.ks",
        "caPassword":"pwd1",
        "certFile":"keystore.ks",
        "certPassword":"pwd1"
        }
}
```

Viewing configuration records for a data source

You can use Latitude Studio components to view configuration records from a data source.

About configuration records

The data represented by a data source consists of both data records and configuration records. Configuration records also may be referred to as schema records.

Data records contain the actual data content. You use these records in the Latitude Studio applications you create for end users.

Configuration records define the attributes in the data records. For each attribute in the data records, there is an attribute configuration record with settings such as:

- The data type for the attribute value
- Whether the attribute value must be unique for each data record
- Whether a data record can have multiple values for the attribute
- Behavior for search and guided navigation
- The groups the attribute belongs to

There is also a global configuration record with general settings for features such as search and spelling correction.

For more details on the structure of these configuration records, see the *Latitude Developer's Guide*.

There are also configuration records for each setting in the attribute and global configuration records. For example, the attribute configuration records include an `mdex-property_Type` setting to determine the data type for the attribute value. There is also an attribute configuration record for `mdex-proper¬ty_Type`.

You can use Latitude Studio components to view all of the attribute configuration records for a data source.

Configuring data sources to display configuration records

By default, a data source only displays the data records.

To configure a data source to only display configuration records, you add the setting `"mdexRecords":"propertyRecords"` to the data source file. For example:

```
{
  "server":"server01.lab.acme.com",
  "port":"15000",
  "name":"mydata schema records"
```

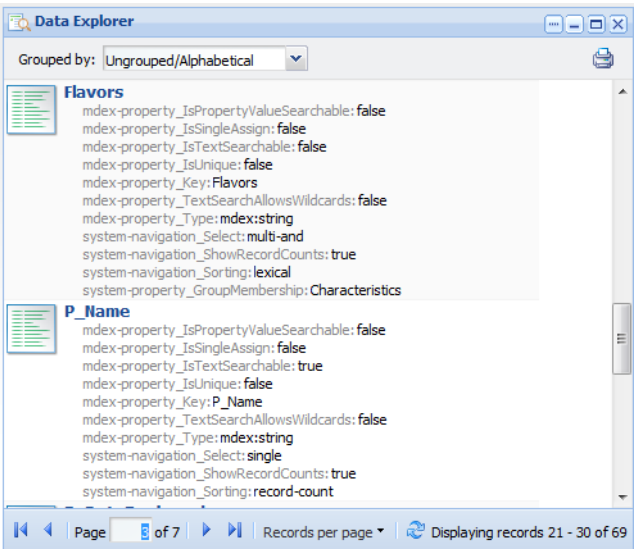
```
  "mdexRecords":"propertyRecords"
}
```

Note: You cannot configure different attribute groups for a configuration-only version of a data source. If you create two data sources with the same host and port, with one displaying data records and the other displaying configuration records, they are simply different views of the same source data, and any change to the attribute groups in one data source automatically updates the groups in the other data source.

Using Latitude Studio components to display configuration records

When a data source is configured to only display configuration records, then when you use that data source for a Latitude Studio component, only the configuration records are shown.

For example, the following **Data Explorer** component is bound to a data source configured to only display configuration records:



With configuration-only data sources, you can use Latitude Studio functions such as search and Guided Navigation to analyze the attribute configuration.

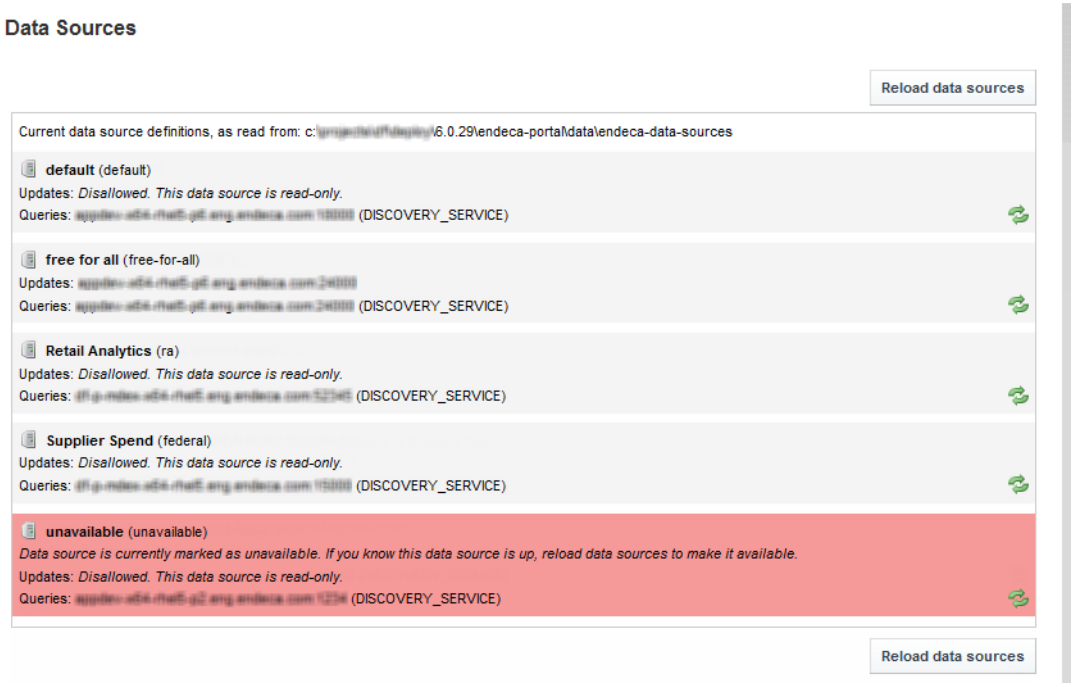
Testing and reloading data sources

The **Data Sources** component on the **Control Panel** allows you to view your list of data sources and test the connection to them. You also can reload updated configuration based on edits you have made on disk.

For each data source, the **Data Sources** component lists:

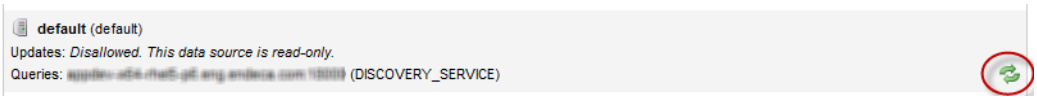
- The data source name and ID
- The data source host and port
- The update host and port, or an indication that the data source is read-only

Any unavailable data sources are highlighted.



To test the data source's connection, click the icon next to the data source. The icon changes, and mousing over it provides a confirmation message.

To clear the Latitude Studio's session cache of attributes and attribute groups for a data source, click the reset cache icon for that data source.



To reload the updated configuration based on edits made on disk, click the **Reload data sources** button.

Chapter 9

Managing Attributes and Attribute Groups

From the **Attribute Settings** component, power users can create, edit, and delete attribute groups. For attributes, power users can edit the display name, and configure how attribute value selection and sorting work on the **Guided Navigation** component.

About attributes and attribute groups

Within a data source, each record is made up of a set of attributes. Attributes are displayed within attribute groups.

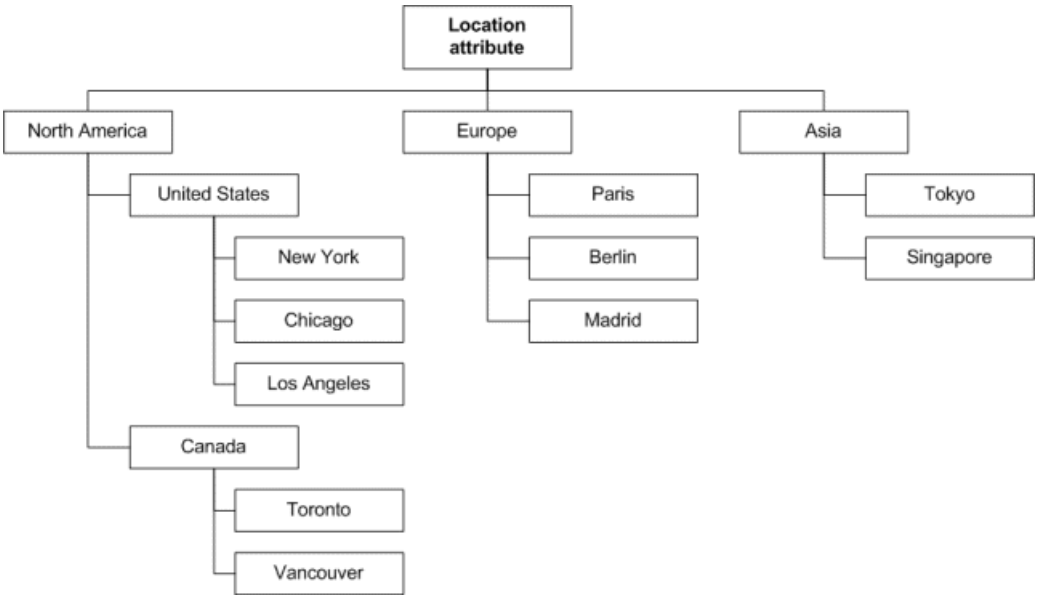
About attributes

An attribute provides a specific piece of information about a record. The attributes for a data source are defined by the MDEX Engine.

In a Latitude Studio application, the components may display the attribute values for a record. End users can also use attribute values to search and filter the data.

An attribute may have a defined set of available values. For example, a Size attribute may be limited to the values Small, Medium, and Large. These are referred to as managed attributes. For other attributes, such as a Description or Price attribute, there may not be specific available values. These are called standard attributes.

For managed attributes, the values may be organized into a hierarchy, to help end users drill down to a value. For example, for a Location attribute, the actual assigned value may be a city. For navigation purposes, however, the values may be grouped by country or continent.



An attribute also may be multi-assign. For multi-assign attributes, the same record can have more than one value for that attribute.

Most of the attribute configuration occurs when the MDEX Engine is created. From the **Attribute Settings** component in Latitude Studio, power users can:

- Change the display name for each attribute
- For each attribute, for the **Guided Navigation** component:
  - Determine the order in which to display the selected attribute values
  - Configure whether end users can select multiple values, and whether matching records must contain all of the selected values

About attribute groups

In Latitude Studio, lists of attributes are displayed within attribute groups, including:

- On the edit view of components
- On the **Guided Navigation** component
- On the **Record Details** component
- On the **Compare** component

When a data source is first created, all of the attributes are in a single group named `Other`. This group is always in place, and always contains any attribute that you have not added to one of your own groups.

You can then define and attributes to your own attribute groups. You can organize the attributes in any way that makes sense for your data. For example, one group might contain descriptive attributes (Size, Color), and another identifying attributes (SKU, Name). Within each group, you control the attribute display order.

When you add attributes to your own attribute groups, they are removed from the `Other` group. However, you can add the same attribute to more than one of your own groups.

Note: You cannot configure different attribute groups for a configuration-only version of a data source. If you create two data sources with the same host and port, with one displaying data records and the other displaying configuration records, they are simply different views of the

same source data, and any change to the attribute groups in one data source automatically updates the groups in the other data source.

Displaying the attributes and groups for a selected data source

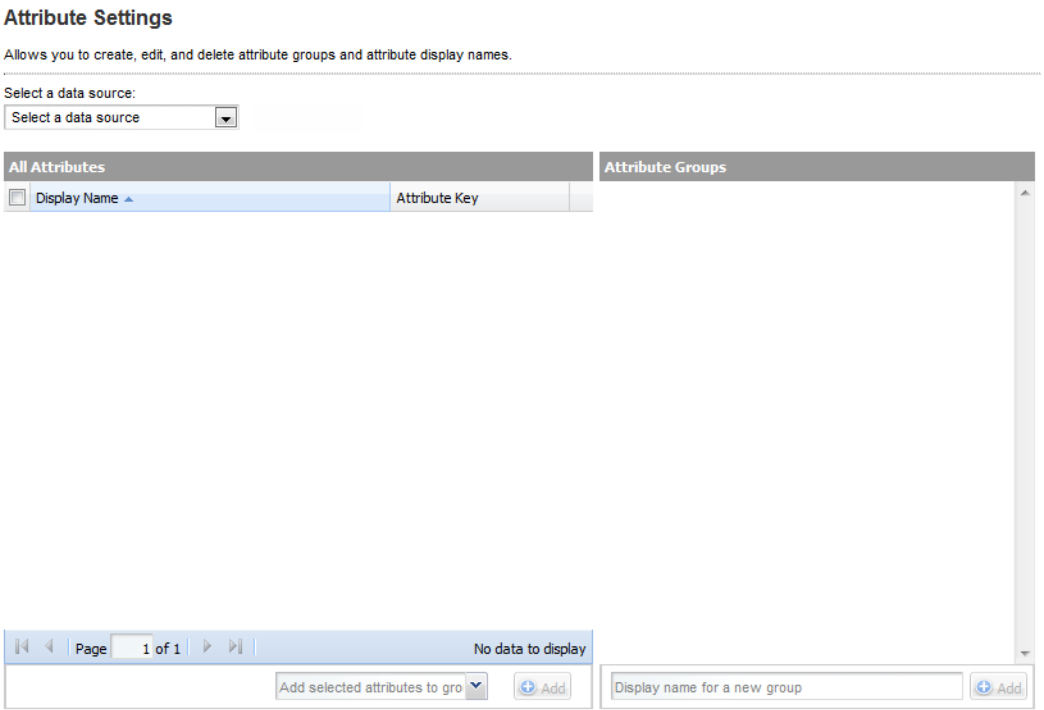
The **Attribute Settings** component displays the lists of attributes and groups for a selected data source. From the **Attribute Settings** component, power users can create, edit, and delete attribute groups, and make some changes to the attribute configuration.

To display the attributes and groups for a data source:

1. In Latitude Studio, click the Dock in the upper-right corner of the page.
2. From the drop-down menu, select **Control Panel**.
3. From the **Latitude** section of the **Control Panel** menu, select **Attribute Settings**.

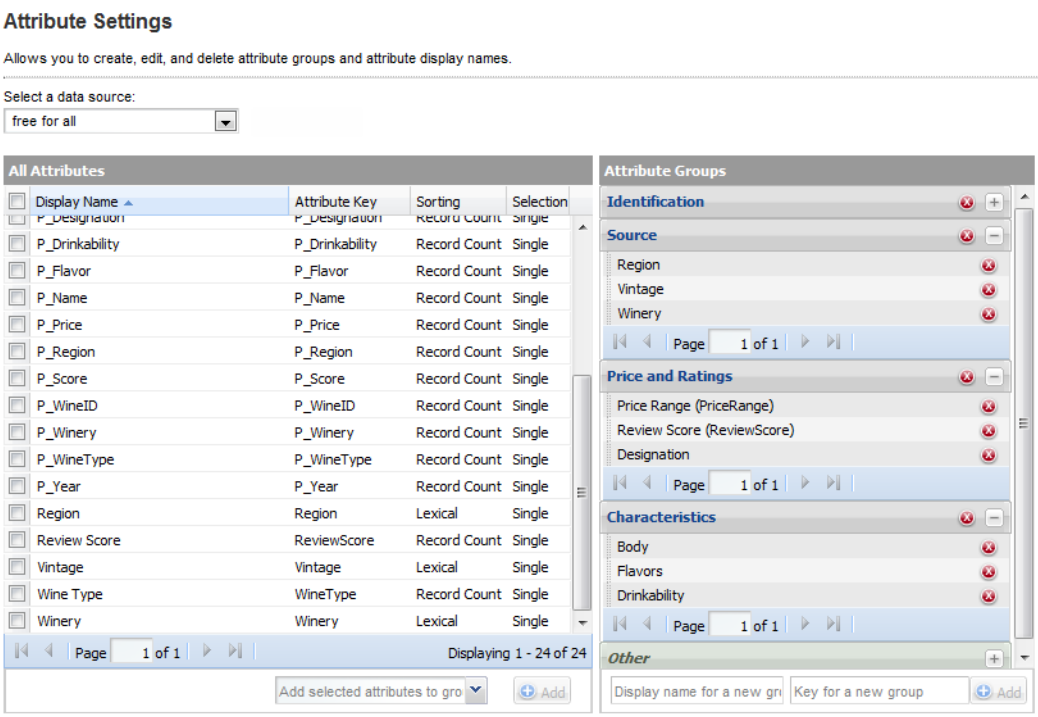
   The **Attribute Settings** component is displayed.

   When the component is first displayed, no data source is selected. The attribute and attribute group lists are empty.



4. From the **Select a data source** drop-down list, select the data source you want to work with.

   The **All Attributes** and **Attribute Groups** lists are populated with the current attributes and attribute groups for the selected data source.



Configuring attributes

The **Attribute Settings** component includes options to configure the attribute display name and, for **Guided Navigation**, how attribute value selection and sorting work.

Changing the display name of an attribute

Display names provide an end-user-friendly way to refer to an attribute in a Latitude Studio application. From the **Attribute Settings** component, power users can edit the attribute display names.

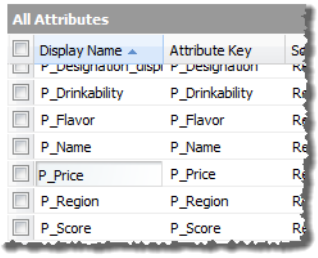
Changing an attribute's display name does not change the attribute's name as tagged on the data record.

Tagged attribute names on records are in an NCName format, which is restricted to letters, numbers, underscores, dashes, and dots. Display names, however, can include non-NCName characters such as spaces and colons. For example, the tagged attribute name on the record might be "WineType", while the display name could be "Wine Type".

From the **Attribute Settings** component, to change the display name of an attribute:

1. In the **All Attributes** list, in the **Display Name** column for the attribute, double-click the cell containing the current display name.

   The cell changes to an editable field.



2. In the field, type in a new display name.

   Because the display name is displayed to end users, you should choose a reasonably short name that end users will understand.

3. Press **Enter**.

   The **All Attributes** list is updated with the new display name. Any component that lists the display name also is updated immediately.

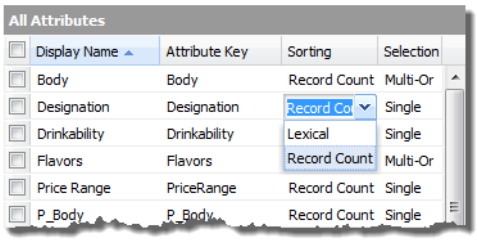
Configuring attribute value selection and sorting for Guided Navigation

For each attribute, the **Attribute Settings** component includes configuration options related to the selection and sorting of attribute values on the **Guided Navigation** component.

In the **All Attributes** list, to configure the value selection and sorting for each attribute:

1. The **Sorting** column determines the order in which to display multiple selected attribute values on the **Guided Navigation** component. To configure how an attribute's values are sorted:
   a) Double-click the **Sorting** column for the attribute.

      The cell becomes a drop-down list.



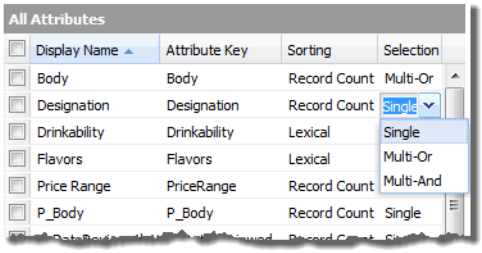
   b) From the drop-down list, select the type of sorting to use. The options are:

Lexical	Indicates to display the selected values in alphabetical or numeric order. For example, if the end user chooses the values Red (15 records), Green (25 records), and Blue (5 records), then if the sorting is lexical, the values are displayed as: • Blue (5 records) • Green (25 records) • Red (15 records)

Record Count	Indicates to display the selected values in descending order by the number of matching records. For example, if the end user chooses the values Red (15 records), Green (25 records), and Blue (5 records), then if the sorting is by record count, the values are displayed as: • Green (25 records) • Red (15 records) • Blue (5 records)

2. The **Selection** column determines whether users can select multiple values, and whether the records must contain all of the selected values. To select the selection option:
   a) Double-click the **Selection** column for the attribute.

   The cell becomes a drop-down list.



   b) From the drop-down list, select the type of selection. The available values are:

Single	Indicates that end users can only select one value for the attribute.
Multi-Or	Indicates that end users can select more than one attribute value. 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).
Multi-And	Indicates that end users can select more than one attribute value. For multi-and, a record matches only if it has all of the selected attribute values. Multi-and should only be used with multi-assign 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).

Configuring attribute groups

The **Attribute Settings** component contains options for creating and populating attribute groups.

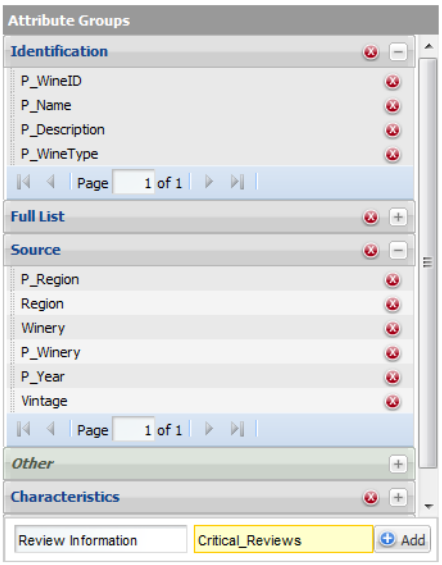
Creating a new attribute group

From the **Attribute Settings** dialog, you can create new attribute groups.

To add an attribute group to the selected data source:

1. Under **Attribute Groups**, in the **Key for a new group** text box, type the attribute group key.

   The key must be NCName-compliant. It can only contain letters, numbers, underscores, and dots. It cannot contain any spaces or other characters.

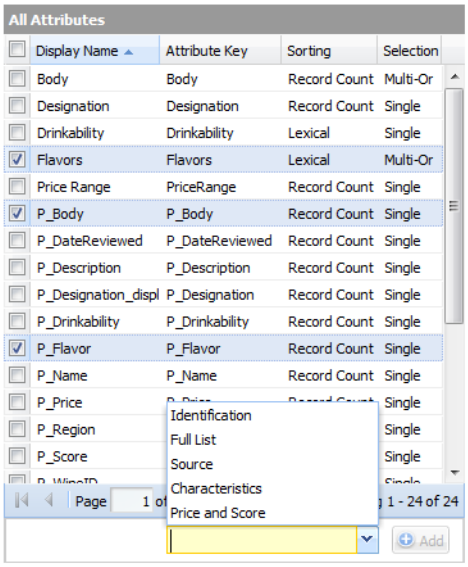


2. In the **Display name for a new group** field, type the display name for the new group.
3. Click **Add**.

   Note that the **Add** button is not enabled until you have provided valid values for both the group key and the display name.

   When you click **Add**, the empty attribute group is added to the **Attribute Groups** panel.

4. To add attributes to the new group:
   a) In the **All Attributes** list, check the attributes that you want to add to the new group.
   b) From the **Add selected attributes to group** drop-down list, select the new group.



c) Click **Add**.

The selected attributes are added to the new group.

If an attribute was not already added to one of your own groups, then it is removed from the default `Other` group.

If an attribute was already added to one of your groups, it is not removed from that group.

After the attribute group is added, it is displayed in the end user and edit views of the relevant components.

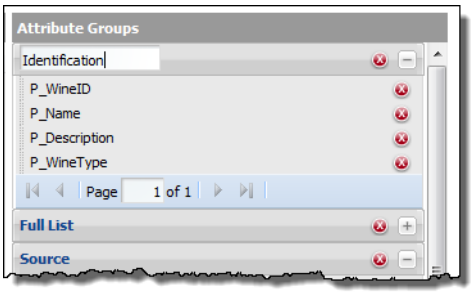
Changing the display name of an attribute group

From the **Attribute Settings** component, you can change the display name of any of your own groups. You cannot change the attribute group key, and you cannot change the name or content of the `Other` group.

To change the display name of an attribute group:

1. In the **Attribute Groups** list, double-click the group name.

   The display name becomes an editable field.



2. Type the new display name for the group.

3. Press **Enter**.

The new display name is updated automatically on the Latitude Studio components.

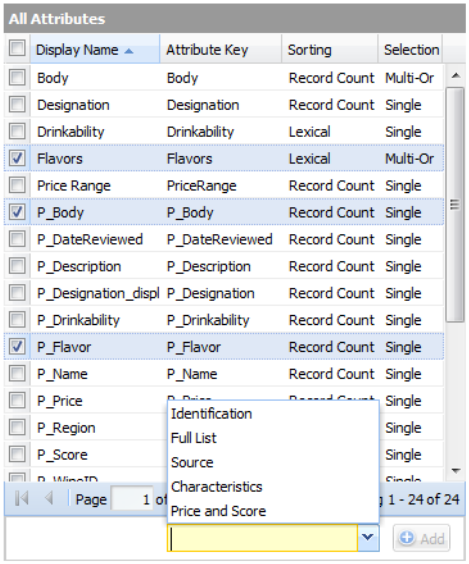
Adding attributes to an attribute group

From the **Attribute Settings** component, you can add any attribute to any of your attribute groups. An attribute can belong to more than one attribute group.

You cannot add attributes to the `Other` default attribute group. The `Other` group is read-only, and automatically contains any attributes that have not been added to any other group.

From the **Attribute Settings** dialog, to add attributes to an attribute group:

1. In the **All Attributes** list, check the attributes that you want to add to the group.
2. From the **Add selected attributes to group** drop-down list, select the group to add the attributes to.



3. Click **Add**.

   The selected attributes are added to the new group.

   If an attribute was not already added to one of your own groups, then it is removed from the default `Other` group.

   If an attribute was already added to one of your groups, it is not removed from that group.

Setting the display order of attributes within a group

From the **Attribute Settings** component, within your attribute groups, you can set the order in which to display the attributes. You cannot change the display order for the default `Other` group.

To change the display order of an attribute within an attribute group:

1. In the **Attribute Groups** list, click the attribute that you want to move.

2. Drag the attribute to the new location in the list for that group.

   Note that you can only drag the attribute within its current group. You cannot drag it into a different group.

3. Release the mouse.

   The attribute is moved to the new location in the list.

Removing attributes from an attribute group

From the **Attribute Settings** component, you can remove attributes from your attribute groups.

Removing an attribute from an attribute group does not remove it from other attribute groups or from the MDEX Engine.

You cannot remove attributes from the default `Other` default attribute group.

To remove an attribute from an attribute group:

1. In the **Attribute Groups** list, open the attribute group you want to manage.
2. Click the delete icon next to the attribute name.

   The attribute is removed from the group.

   If the attribute no longer belongs to any of your own groups, then it is restored to the default `Other` group.

Deleting attribute groups

From the **Attribute Settings** component, you can delete your own attribute groups.

You cannot delete the default `Other` attribute group.

Deleting an attribute group does not delete its attributes from the MDEX Engine or from your other attribute groups.

From the **Attribute Settings** component, to delete an attribute group:

1. In the **Attribute Groups** panel, click the delete icon for the attribute group.

   You are prompted to confirm the delete operation.

2. On the confirmation message, to delete the attribute group, click **Yes**.

The attribute group is removed from the MDEX Engine and from the **Attribute Groups** panel. It no longer displays on the Latitude Studio components.

If an attribute does not belong to any of your other groups, it is restored to the default `Other` group.

Part 4

Building a Latitude Studio Application

Chapter 10

About Building Latitude Studio Applications

What is a Latitude Studio application?

Latitude Studio applications are business intelligence tools that provide end users with ways to view, navigate, compare, and analyze data.

Each page consists of a set of components. Each component is designed to perform a specific type of function, such as:

- Displaying lists of records or record attributes
- Filtering the data displayed on other components
- Displaying visual representations of data
- Highlighting specific values

Sample page layouts for different types of end user tasks

About these sample layouts

You can set up each page in your application to support different types of data exploration and discovery.

For example, one page may enable end users to view and compare different subsets of the data. Other pages may foster more in-depth analysis and visualization of the data. Or you may want to provide a quick visual snapshot of the current data, highlighting important values and showing trends over time.

These sample page layouts show how different components and arrangements of components can support these different types of user tasks. They can be used as a starting point for designing your own applications.

Sample layout - status and alerts dashboard

This type of layout is designed to present end users with a quick summary of key data points, to allow them to monitor changes to the data over time.

The **Metrics Bar** and **Alerts** components are used to highlight important details in the data such as key performance indicators.

The **Chart** component provides a visual display of key data.

The **Cross Tab** component provides a broader set of data.

Sample layout - visualization dashboard

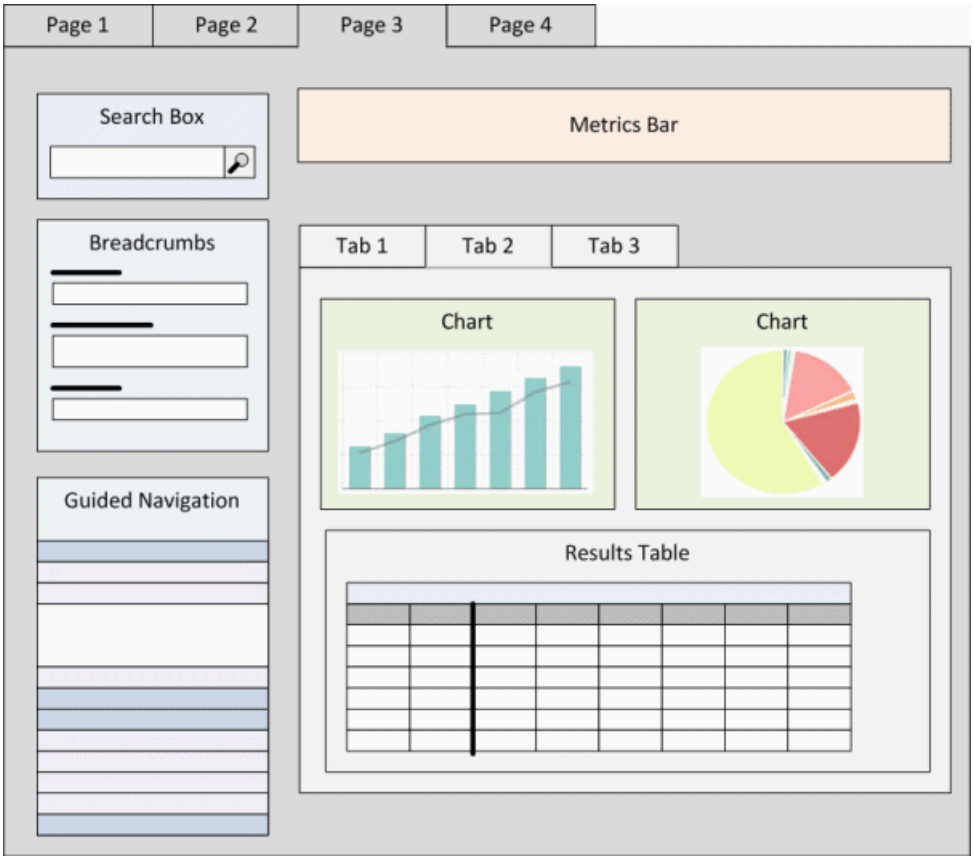
This page layout also provides a quick snapshot of data, but in a more visual format.

Each **Chart** component highlights a specific set of data, to compare values or show trends.

The **Cross Tab** component provides a broader set of data.

Sample layout - visual discovery and analysis

This type of layout allows end users to explore and filter the data in order to do more analysis and comparison.



The **Search Box** and **Guided Navigation** components allow users to refine the displayed data using search terms or selected attribute values. As users refine the data, the other components may be updated to only include the data for the current refinement.

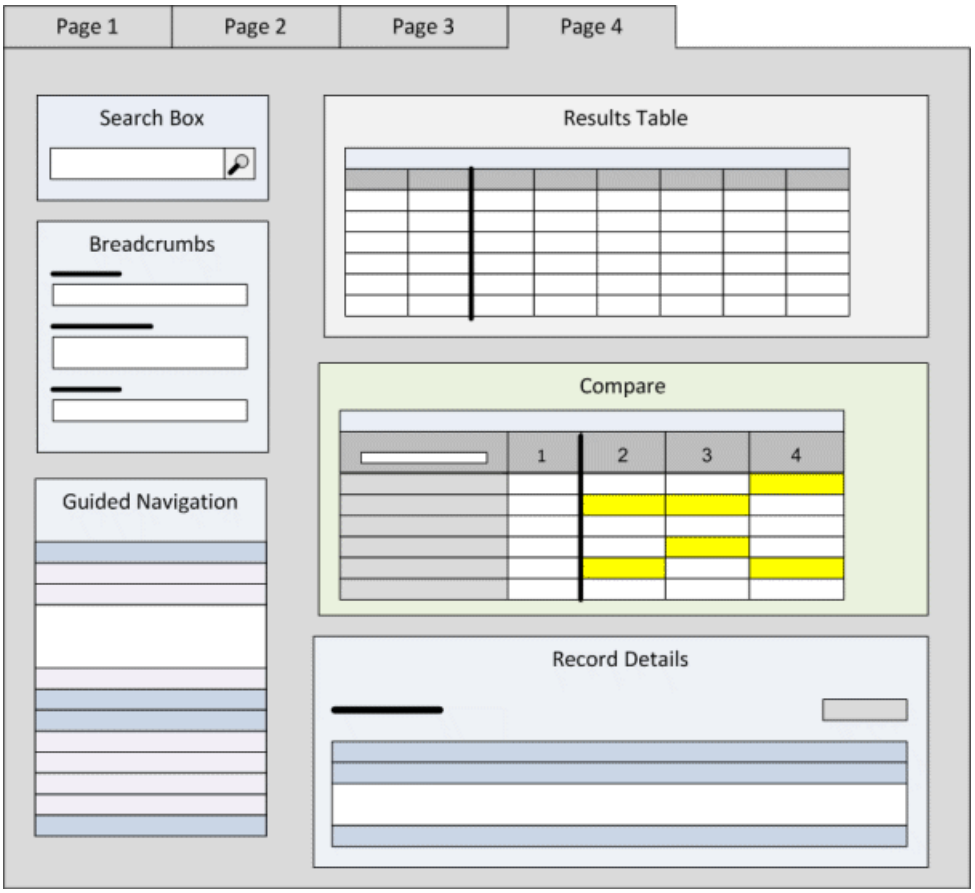
The **Breadcrumbs** component displays the current refinements, and is used to remove refinements.

The **Results Table** component provides a list of records or generated values, with the **Chart** components providing a more visual analysis of the current data.

This layout also includes a tabbed container to provide different views.

Sample layout - unstructured visual discovery

This type of layout provides basic tools to explore and understand the data.



The **Search Box** and **Guided Navigation** components allow users to refine the displayed data using search terms or selected attribute values. As users refine the data, the other components may be updated to only include the data for the current refinement.

The **Breadcrumbs** component displays the current refinements, and is used to remove refinements.

The **Results Table** component contains a list of records or generated metrics. From the **Results Table**, users can:

- Use the **Record Details** component to see an expanded list of attributes for a selected record
- Use the **Compare** component to analyze differences among selected records

Importing the sample Latitude Studio pages

Latitude Studio provides a set of sample pages you can use as a starting point for working with components and viewing data and configuration settings.

One page, called **Data Explorer**, contains components bound to the `default` data source. This page allows you to explore the data in the default data source.

The other page, **Schema Explorer**, contains similar components. The components on the **Schema Explorer** page are bound to the `default-schema` data source, which only displays configuration records. This page allows you to explore the configuration settings for the default data source.

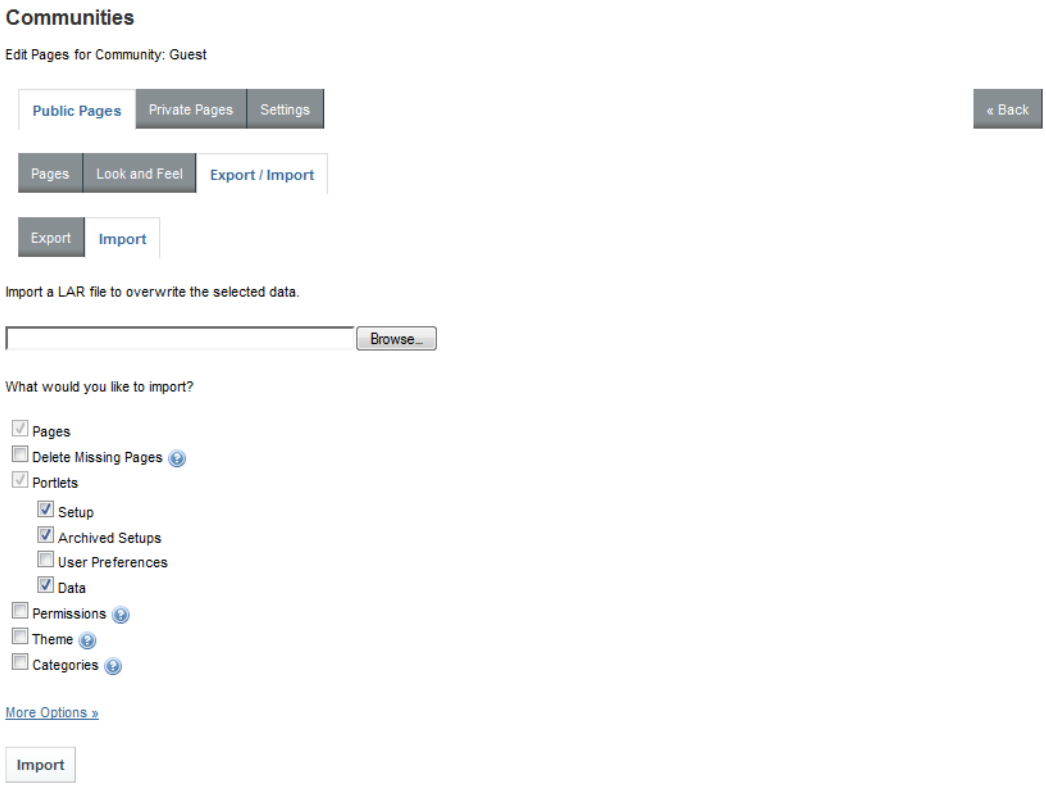
The sample pages are in a LAR file called `SchemaAndDataExplorers.lar`. The file is in the `endeca-portal` directory.

To import the sample pages into Latitude Studio:

1. To display the **Manage Pages** options from the **Communities** component:
   a) From the Dock menu, select **Control Panel**.
   b) On the **Control Panel**, click **Communities**.
   c) On the **Communities** page, for the community you want to import the pages into, click the **Actions** button, then click **Manage Pages**.

   If you have not set up any other communities, then the only community is the **Guest** community.

2. On the **Communities** page for the selected community, click the **Export/Import** tab.
3. Click the **Import** tab. The **Import** tab contains the options for importing pages from a LAR file.



4. To search for and select the `SchemaAndDataExplorers.lar` file, click the **Browse** button.
5. Click the **Import** button.

   The **Data Explorer** and **Schema Explorer** pages are added to the application.

Chapter 11

Managing Pages

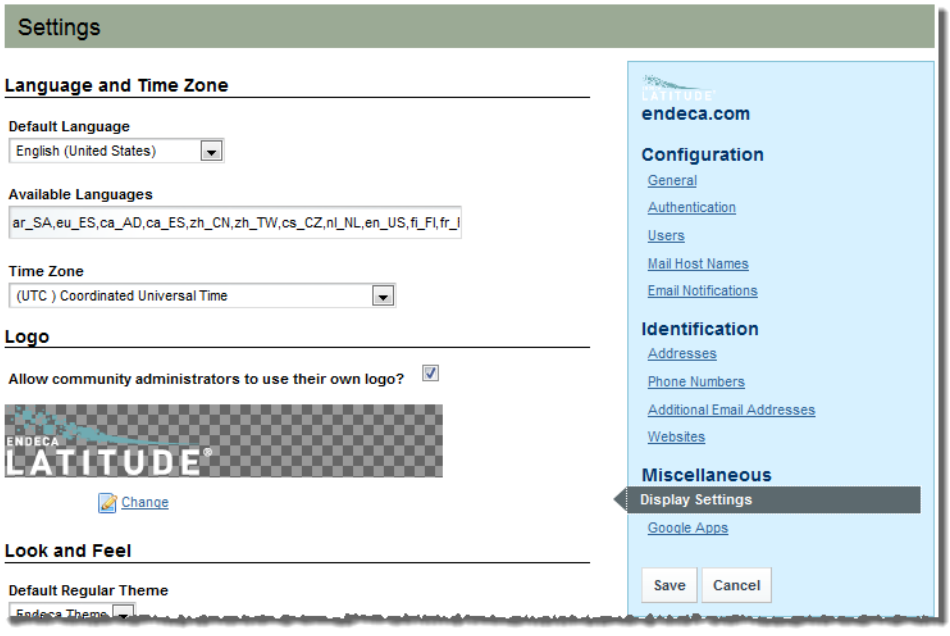
Each Latitude Studio application is made up of one or more pages.

Changing the company logo at the top of the application

By default, the logo at the top of the Latitude Studio application is the Endeca Latitude logo. You can replace this logo with your own company logo.

To change the company logo:

1. In the **Control Panel**, in the **Portal** section, click **Settings**.
2. On the **Settings** page, in the right-hand menu, click **Display Settings**.



3. In the **Logo** section of the **Settings** page, click **Change**.
4. Browse to the logo .png file you want to use, open it, then click **Save**.
5. On the **Settings** page, click **Save**.

   The new logo is applied to the application's pages.

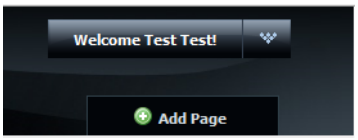
Adding a page

When you first launch Latitude Studio, the interface consists of a single page. However, you can add additional pages.

Adding pages to your application maximizes application logic while minimizing visual clutter. Spreading components among several pages also helps improve the performance of the application.

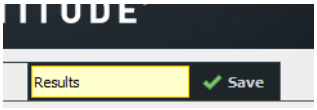
To add a page to your Latitude Studio application:

1. Click **Add Page**.



   An empty page label is created.

2. In the label field, type a name for your new page.

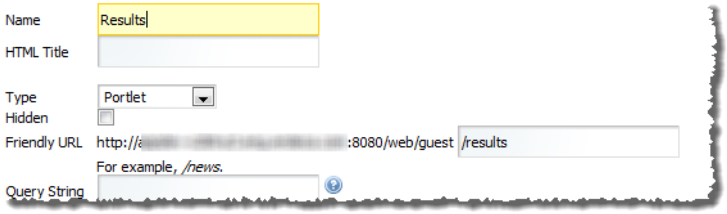


3. Click **Save**.

   A new named tab is added to the application.

The name that you give a new page is actually its display name. This is the name shown in the page tab.

When the page is created, it also is given a friendly URL. You can see the friendly URL on the **Manage Page** dialog for that page. For example:



Because the display name is different from the friendly URL, you can change the display name without affecting any page transitions.

Once you have created additional pages, you can create page transitions to target the output of individual components to the page you specify. You can also apply a new theme to the page.

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.
3. In the field, type the new name.

4. Click **Save**.

You can also rename a page from the **Manage Pages** dialog for that page.

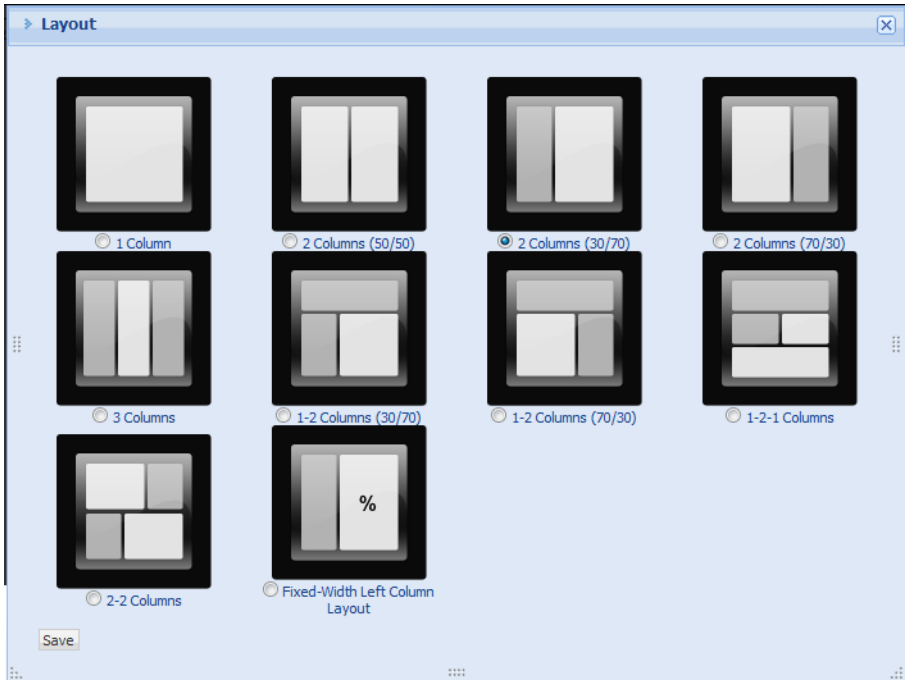
Applying a layout template to a page

Liferay provides a number of default layout templates that allow you to establish the layout of the pages in your Latitude Studio application.

Layout templates determine how components are organized on each page. For example, a page may contain two horizontal columns of different widths, or a banner over two equally-wide columns.

To change the layout of a page:

1. Click the page tab.
2. In the Dock menu, click **Layout Template**.
3. In the **Layout** window, click the radio button for the layout you want to apply to the page.



4. Click **Save**

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

Note: If the default layout templates supplied by Liferay do not meet your needs, your developer can create custom layout template plugins for you, and your system administrator can make them available to your application.

Applying a theme to a page

Themes are hot deployable plugins that you can use to customize the appearance of your application. By using your own theme, your Latitude Studio application can adhere to the look-and-feel standards used across all of your company's Web sites and Web application.

By default, Latitude Studio ships with the Endeca theme and the Liferay Classic theme.

You also can download themes from the Liferay Community Plugin Library on the Liferay Web site:

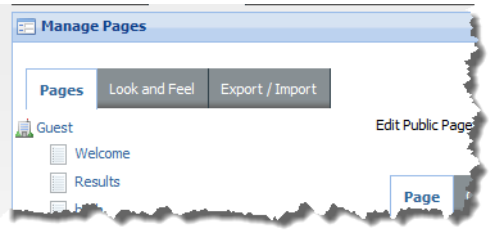
```
http://www.liferay.com/downloads/liferay-portal/community-plugins
```

These themes are contributed by the Liferay community of users.

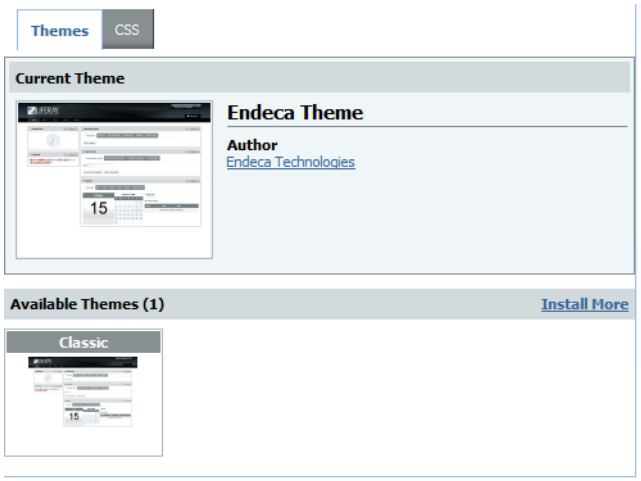
Your application developer can also create customized themes for your application. For information on theme development, see the *Liferay Developer's Guide*, which is available from the Liferay Web site.

Each page can have a different theme. To apply an existing theme to a page:

1. From the Dock menu, choose **Manage Pages**.
2. On the **Pages** tab of the **Manage Pages** dialog, click a page.



3. Click the **Look and Feel** tab.
4. The **Themes** tab shows the theme that is currently being used by the page and lists any available themes. To apply another theme, click that theme.



If the new theme was applied successfully, a message displays to confirm that the request was processed.

Deleting a page

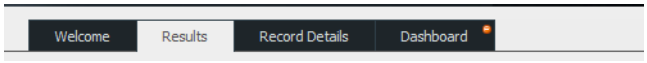
You can delete pages from the application. When you a delete a page, any components on the page also are deleted.

You cannot delete the currently selected page. In other words, before deleting a page, you must navigate to a different page.

To delete a page:

1. Navigate to a different page from the one you want to delete.
2. Place the cursor on the page tab of the page to be deleted.

   A delete icon is displayed in the right upper corner of the tab.



3. Click the delete icon.

   A delete confirm prompt is displayed.

4. To delete the page, click **OK**.

You can also delete a page from the **Manage Pages** dialog. As with the procedure above, you cannot delete the currently selected page.

Chapter 12

Adding and Configuring Components

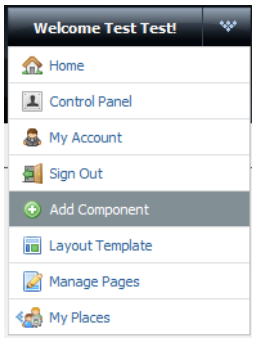
On each page, you add and configure components in order to allow end users to view and analyze the data in a meaningful way.

Adding a Latitude component to a page

Latitude Studio contains several standard Latitude components. These components allow you to add Latitude functionality to your application.

To add a Latitude component to your Latitude Studio application:

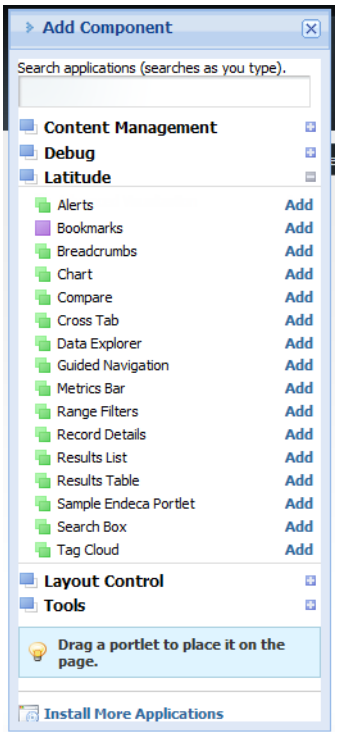
1. Point the cursor at the Dock.
2. From the Dock menu, select **Add Component**.



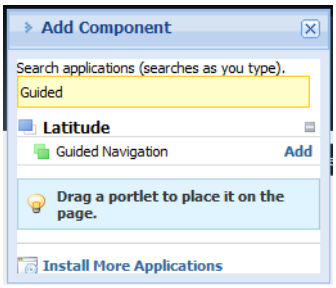
   The **Add Component** dialog is displayed.

3. In the **Add Component** dialog, expand the **Latitude** category.

   A list of the available Latitude components appears.



You can also use the search field to search for a specific component.



4. To add a component to the main page layout, either:
   - Click the **Add** link for the component.
   - Drag the component from the **Add Component** dialog to the page.

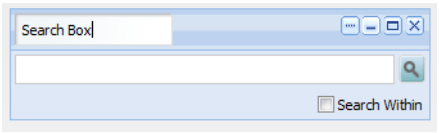
Renaming components

After you add a component to your application, you can rename it to reflect your own terminology or the content being displayed.

To rename a component:

1. Click the title bar of the component.

   The default title becomes editable.



2. Overwrite the existing title with your new one.
3. Click outside of the title bar.

   The component now uses the new title.

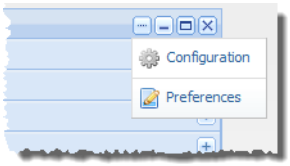
Editing components

You edit Latitude Studio components in order to further customize your end users' experience. For example, you can specify the data source to use, the attribute groups to display, and the format of displayed data.

The general process for editing a component is the same for each component. For detailed information about configuring specific components, see the component documentation.

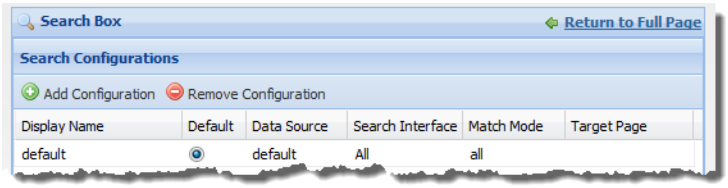
To edit a Latitude component:

1. In the component's title bar, click **....** In the drop-down menu, click **Preferences**.



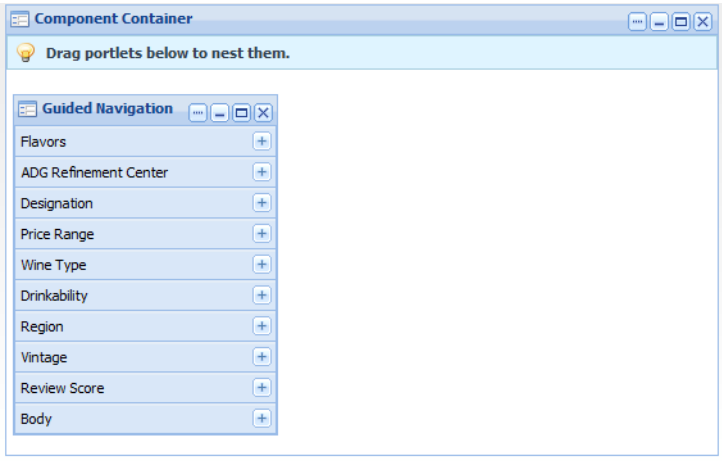
   The edit view for the component is displayed.

2. Edit the settings for the component.
3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.



Using a Component Container to group components

The **Component Container**, available from the **Layout Control** section of the **Add Component** menu, allows power users to organize components by grouping them together in a single container.



To change the layout of the container, from the component menu, select the **Configuration** option.

Note: Do not add a **Component Container** component to a **Tabbed Container** component.

Using a Tabbed Component Container to group components

The **Tabbed Component Container** groups a page into multiple tabs.

About the Tabbed Component Container

The **Tabbed Component Container**, available from the **Layout Control** section of the **Add Component** menu, creates a tabbed interface.

Each tab can contain a different set of components. For example, you might choose to put a **Results Table** component containing customer-based information on one tab, and another **Results Table** component containing product-based information on another tab.

When working with a **Tabbed Component Container**:

- Do not add a **Component Container** component to a **Tabbed Component Container**.

  You also cannot add a **Tabbed Component Container** to another **Tabbed Component Container**.
- After placing the **Tabbed Component Container** on the page, make sure to refresh the page before you add other components to the tabs.

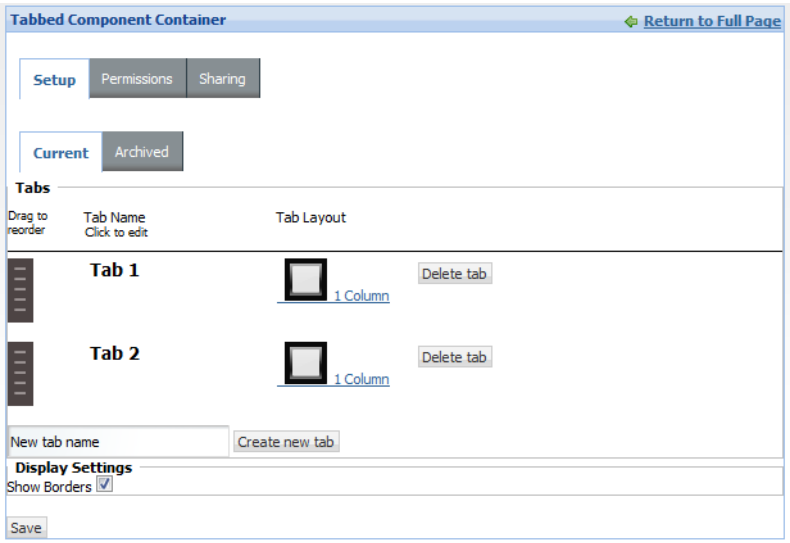
Configuring the Tabbed Component Container

For a **Tabbed Component Container** component, power users can add and remove tabs, and determine whether to display a border around the components on the tab.

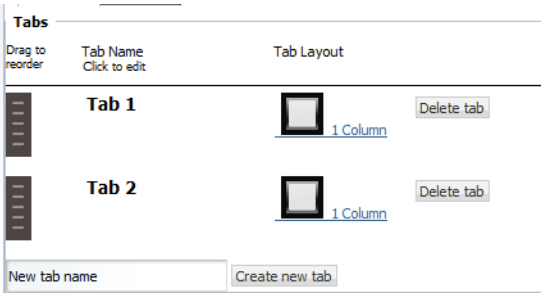
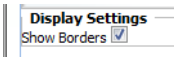
To display the edit view for a **Tabbed Component Container** component, click the **...** button for the component. From the drop-down menu, select **Configuration**.

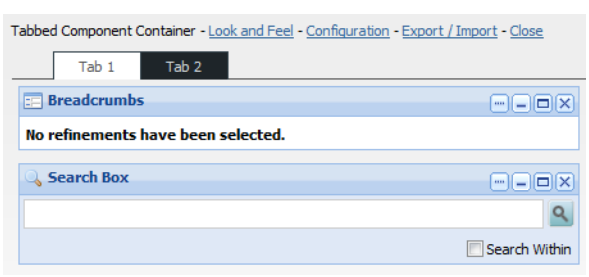
Note: Only the configuration tasks on the **Setup** tab are relevant to our purposes. You may disregard the other tabs.

From the **Setup** tab, you can either edit the current tabbed component container, or select an archived version (if available).



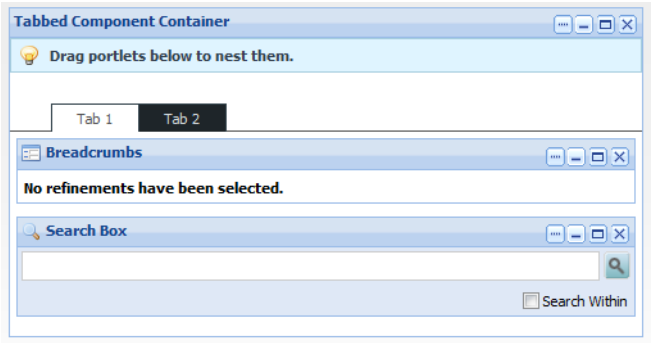
The tab configuration includes the following sections:

Tabs:	From this section, you can: • Add or delete tabs • Select the layout for each tab • Change the display name for each tab • Set the display order of the tabs 
Display Settings:	In this section, to draw the regular component border around the Tabbed Component container, check **Show Borders**.  If **Show Borders** is unchecked, then the **Tabbed Component Container** title bar does not display. The buttons are replaced by links.



To save the changes to the **Tabbed Component Container**, click **Save**.

After configuring the **Tabbed Component Container**, you can then drag other components onto each tab and configure them as usual.



Chapter 13

Using the Latitude Query Language (LQL)

About LQL

The Latitude Query Language (LQL) is designed specifically to query and manipulate data from an MDEX Engine. LQL is used to support all of the search and navigation functions in Latitude Studio.

Power users also need to be able to write LQL statements in order to generate the data to display for the following components:

- **Alerts**
- **Chart**
- **Cross Tab**
- **Map**
- **Metrics Bar**
- **Results Table**

LQL is very similar to SQL, and includes support for:

- Aggregation functions.
- Numeric functions.
- Composite expressions to construct complex derived functions.
- Grouped aggregations such as cross-tabulated totals over one or more managed attributes.
- Top-k according to an arbitrary function.
- Cross-grouping comparisons such as time period comparisons.
- Intra-aggregate comparisons such as computation of the percentage contribution of one region of the data to a broader subtotal.

LQL syntax

Note: The syntax descriptions use the BNF notation.

Basic syntax of LQL queries and statements

A query is an ordered list of statements. Each statement generates a single named set of records.

```
<Query>        ::=  <Statements>
<Statements>   ::=  <Statement> [; <Statements>]
```

The basic syntax for a statement is:

```
<Statement>    ::=  (DEFINE | RETURN) <Key> AS <SELECT>
                       [<FROM>]
                       [<WHERE>]
                       [<GROUP BY>]
                       [<HAVING>]
                       [<ORDER BY>]
                       [<PAGE>]
```

A statement specifies:

	Item	Description and Syntax
1	A name for the resulting record set (`DEFINE | RETURN <recordSetName> AS`)	In most cases, you will use `RETURN`. Use `DEFINE` when you are generating a set of records to use in a later statement.
2	The attributes (or derived attributes) that the set of records should contain (`SELECT` expressions)	`<Select>  ::=  SELECT <Assigns>` `<Assigns> ::=  <Assign> [, <Assigns>]` `<Assign>  ::=  <Expr> AS <Key>` Expressions can perform a variety of functions. You can use commas to separate multiple expressions within a statement.
3	The input record set (`FROM`)	`<From>  ::=  FROM <Key>` `::=  FROM NavStateRecords` `::=  FROM AllBaseRecords` The input records provide the values used in expressions and `WHERE` filters. The input record set is either: • The result of a previous statement • The records for the current navigation state (`NavStateRecords`). This is the default value. If you do not provide a `FROM` clause, then this is the value used. • The complete set of records (`AllBaseRecords`)
4	An optional filter to apply to the input (`WHERE`)	`<Where> ::=  WHERE <Filter>` As in SQL, the `WHERE` clause filters the inbound records before any operations are performed on them.
5	A way to group the output records (`GROUP BY`)	Used to map the input records to the output records. `<GroupBy>   ::=  GROUP` `::=  GROUP BY <Groupings>` `<Groupings> ::=  <Grouping> [, <Groupings>]` `<Grouping>  ::=  <Key>` `::=  <Key>:<int>`

	Item	Description and Syntax
		• If you provide a `GROUP BY` list of attributes, then all records containing identical values for those attributes are grouped (or bucketed) together. • If you only specify `GROUP`, all input records are grouped into a single output record. • If you do not specify either `GROUP` or `GROUP BY`, then no grouping occurs. The number of output records is the same as the number of input records. The colon operator allows you to group by a hierarchical attribute at a specified level of the hierarchy.
6	An optional filter to apply to output records (`HAVING`)	`<Having> ::= HAVING <Filter>` As in SQL, the `HAVING` clause filters the results of the operations on the input records.
7	An optional way to determine the sort order of the output records (`ORDER BY`)	`<OrderBy> ::= ORDER BY <OrderList>` `<OrderList>::= <Order> [, <OrderList>]` `<Order> ::= <Key> [ASC | DESC]` The `ORDER BY` clause includes the comma-separated list of attribute(s) to use for the sorting, and whether to sort from smallest to largest (`ASC`) or largest to smallest (`DESC`). The default order is to sort in ascending order.
8	An optional way to page through the output records, in other words, to return only a subset of the output (`PAGE(i,n)`)	`<Page> ::= PAGE(<int>,<int>)` In the syntax, the first value is the number of the record to start at, and the second value is the number of records to return.

Elements of LQL statements

Using SELECT AS expressions to calculate derived attributes

LQL statements typically use expressions to compute one or more derived attributes.

Each aggregation operation can declare an arbitrary set of named expressions, sometimes referred to as derived attributes, using SELECT AS syntax. These expressions represent aggregate analytic functions that are computed for each aggregated record in the statement result.

Important: Derived attribute names must be NCName-compliant. They cannot contain spaces or special characters. For example, the following statement would not be valid:

```
RETURN foo AS SELECT AVG(Price) AS "Average Price"
```

The space would have to be removed:

```
RETURN foo AS SELECT AVG(Price) AS "AveragePrice"
```

Supported aggregation functions

LQL supports the following aggregation functions:

Function	Description
AVG	Computes the arithmetic mean value for a field.
COUNT	Counts the number of records with valid non-null values in a field for each GROUP BY result.
COUNTDISTINCT	Counts the number of unique, valid non-null values in a field for each GROUP BY result.
MAX	Finds the maximum value for a field.
MIN	Finds the minimum value for a field.
MEDIAN	Finds the median value for a field.
STDDEV	Computes the standard deviation for a field.
ARB	Selects an arbitrary but consistent value from the set of values in a field.
SUM	Computes the sum of field values.
VARIANCE	Computes the variance (that is, the square of the standard deviation) for a field.

Supported numeric functions

LQL supports the following numeric functions:

Function	Description
addition	The addition operator `(+)`.
subtraction	The subtraction operator `(-)`.
multiplication	The multiplication operator `(*)`.
division	The division operator `(/)`.
ABS	Returns the absolute value of `n`. If `n` is `0` or a positive integer, returns `n`. Otherwise, `n` is multiplied by -1.
CEIL	Returns the smallest integer value not less than `n`.
EXP	Exponentiation, where the base is `e`. Returns the value of `e` (the base of natural logarithms) raised to the power of `n`.
FLOOR	Returns the largest integer value not greater than `n`.
LN	Natural logarithm. Computes the logarithm of its single argument, the base of which is `e`.
LOG	Logarithm. `log(n, m)` takes two arguments, where `n` is the base, and `m` is the value you are taking the logarithm of.

Function	Description
MOD	Modulo. Returns the remainder of n divided by m. LQL uses the fmod floating point remainder, as defined in the C/POSIX standard.
ROUND	Returns a number rounded to the specified decimal place. The unary version drops the decimal (non-integral) portion of the input. The binary version allows you to set the number of spaces at which the number is rounded: • Positive second arguments specified to this function correspond to the number of places that must be returned after the decimal point. For example, ROUND(123.4567, 3) = 123.457 • Negative second arguments correspond to the number of places that must be returned before the decimal point. For example, ROUND(123.4567, -3) = 100.0
SIGN	Returns the sign of the argument as -1, 0, or 1, depending on whether n is negative, zero, or positive.
SQRT	Returns the nonnegative square root of n.
TRUNC	Returns the number n, truncated to m decimals. If m is 0, the result has no decimal point or fractional part. The unary version drops the decimal (non-integral) portion of the input, while the binary version allows you to set the number of spaces at which the number is truncated.
SIN	The sine of n, where the angle of n is in radians.
COS	The cosine of n, where the angle of n is in radians.
TAN	The tangent of n, where the angle of n is in radians.
POWER	Returns the value of n raised to the power of m.
TO_DURATION	Casts an integer into a number of milliseconds so that it can be used as a duration. When TO_DURATION is given a value of type double, it removes the decimal portion and converts the integer portion to a duration.

COALESCE expression

The COALESCE expression allows for user-specified null-handling.

You can use the COALESCE expression to evaluate records for multiple values and return the first non-null value encountered, in the order specified. The following requirements apply:

- You can specify two or more arguments to COALESCE
- Arguments that you specify to COALESCE should all be of the same type
- You cannot specify managed attributes as arguments to COALESCE. However, if you a managed attribute is mapped to a standard attribute, you can specify this standard attribute to COALESCE to create a valid query.

- The COALESCE expression can only be used in a SELECT clause, and not in other clauses (such as WHERE)

In the following example, all records without a specified price are treated as zero in the computation:

```
AVG(COALESCE(price, 0))
```

COALESCE can also be used without aggregation, for example:

```
SELECT COALESCE(price, 0) AS price_or_zero WHERE
```

Using the COUNT and COUNTDISTINCT functions

The COUNT function returns the number of records that have a value for an attribute. COUNTDISTINCT counts the number of distinct values for an attribute.

Using COUNT to count the number of records with valid values

The COUNT function counts the number of records that have valid non-null values in a field for each GROUP BY result.

For example, the following records include Size and Color attributes:

```
Record 1: Size=small, Color=red, Color=white
Record 2: Size=small, Color=blue, Color=green
Record 3: Size=small, Color=black
Record 4: Size=small
```

The following statement returns the number of records for each size that have a value for the Color attribute:

```
RETURN result AS SELECT COUNT(Color) as Total GROUP BY Size
```

The statement result is:

```
Record 1: Size=small, Total=3
```

Because all of the records have the same value for Size, there is only one group, and thus only one record. For this group, the value of Total is 3, because only three of the records have valid Color assignments.

Using COUNTDISTINCT to get the number of distinct values for an attribute

The COUNTDISTINCT function returns the number of unique, valid non-null values in a field for each GROUP BY result.

COUNTDISTINCT can only be used for single-assign attributes, and not for multi-assigned attributes. Using a multi-assign attribute generates misleading results.

For example, for the following records:

```
Record 1: Size=small, Color=red
Record 2: Size=small, Color=blue
Record 3: Size=small, Color=red
Record 4: Size=small
```

The following statement returns for each size the number of different values for the Color attribute:

```
RETURN result AS SELECT COUNTDISTINCT (Color) as Total GROUP
  BY Size
```

The statement result is:

```
Record 1: Size=small, Total=2
```

Because all of the records have the same value for Size, there is only one group, and thus only one record. For this group, the value of Total is 2 because there are two unique, valid, non-null values for the Color attribute: red and blue.

Using GROUP BY or GROUP to group the statement output

The GROUP BY option buckets a set of records into a resulting set of aggregated records. Most LQL statements in Latitude Studio are aggregation operations.

To define the set of resulting buckets, a statement must specify a set of GROUP BY attributes. The cross product of all values in these grouping attributes defines the set of candidate buckets.

The results are automatically pruned to include only non-empty buckets.

Basic usage of GROUP BY

For example, suppose we have sales transaction data with records consisting of the following attributes:

```
{ TransId, ProductType, Amount, Year, Quarter, Region,
  SalesRep, Customer }
```

For example:

```
{ TransId = 1, ProductType = "Widget", Amount = 100.00,
  Year = 2009, Quarter = "09Q1", Region  = "East",
  SalesRep = "J. Smith", Customer = "Customer1" }
```

If an LQL statement uses Region and Year as GROUP BY attributes, the statement results contain an aggregated record for each valid, non-empty combination of Region and Year. In LQL, this example is expressed as:

```
DEFINE RegionsByYear AS
GROUP BY Region, Year
```

resulting in the aggregates of the form { Region, Year }, for example:

```
{ "East", "2008" }
{ "West", "2009" }
{ "East", "2009" }
```

Specifying the hierarchy level for a managed attribute

You can also group by a specified depth of each managed attribute.

For example, the Region attribute contains the hierarchy Country, State, and City. To group the results at the State level (one level below the root of the managed attribute hierarchy), you would use the following syntax:

```
GROUP BY "Region":1
```

Using a GROUP BY that is an output of a SELECT expression

A GROUP BY key can be the output of a SELECT expression), as long as that expression itself does not contain an aggregation function.

For example, the following syntax is a correct usage of GROUP BY:

```
SELECT COALESCE(Person, 'Unknown Person')
as Person2, ... GROUP BY Person2
```

The following syntax is incorrect and results in an error, because Sales2 contains an aggregation function (SUM):

```
SELECT SUM(Sales) as Sales2, ... GROUP
BY Sales2
```

Specifying only Group

You can also use a GROUP statement to aggregate results into a single bucket.

For example, the following statement uses the SUM statement to return a single sum across a set of records:

```
RETURN "ReviewCount" AS SELECT
SUM(number_of_reviews) AS "NumReviews"
GROUP
```

This statement returns one record for NumReviews. The value is the sum of the values for the attribute number_of_reviews.

Using WHERE or HAVING to filter input or output

LQL supports the WHERE and HAVING filtering options. You use a WHERE clause to filter the input records for an expression, and a HAVING clause to filter the statement output records.

Within a WHERE or HAVING clause, you can use filter operations such as:

- Numeric and string value comparison functions (equal, not equal, greater than, less than, between, and so on)
- Boolean operators (AND, OR, or NOT)
- The IN filter, which checks whether a value belongs to a record set generated from a previous statement. The previous statement must have a GROUP BY value.

The syntax for the filtering operations is:

```
<Filter>
               ::=  <Key> <Compare> <Literal>
               ::=  <Key> IS [NOT] NULL
               ::=  <Filter> AND <Filter>
               ::=  <Filter> OR <Filter> | NOT <Filter>
               ::=  [<KeyList>] IN <Key>
<Compare>      ::=  = | <> | < | > | <= | >=
<KeyList>      ::=  <Key> [, <KeyList>]
```

For the IN filter, the number and type of keys in the <KeyList> must match the number and type of keys used in the statement referenced by the IN clause.

In the following example, the results only include sales reps who generated at least $10,000:

```
RETURN Reps AS
SELECT SUM(Amount) AS SalesTotal
GROUP BY SalesRep
HAVING SalesTotal > 10000
```

In this example, the amounts are only calculated for sales in the Western region. Then within those results, only sales reps who generated at least $10,000 are included:

```
RETURN Reps AS
SELECT SUM(Amount) AS SalesTotal
WHERE Region = 'West'
GROUP BY SalesRep
HAVING SalesTotal > 10000
```

In the following example, a single statement contains two expressions. The first expression computes the total for the all of the records, and the second expression computes the total for a particular sales representative:

```
RETURN QuarterTotals AS SELECT
  SUM(Amount) AS SalesTotal,
  SUM(Amount) WHERE SalesRep = 'John Smith' AS JohnTotal
GROUP BY Quarter
```

This would return for each quarter both the total overall sales and the total sales for John Smith.

Using ORDER BY to sort the statement results

`ORDER BY` operators control the order of the records produced by the statement operations.

You can sort the records by any of their attribute or derived attribute values, and can specify whether to sort in ascending or descending order. You can use any combination of values and sort orders.

In this example, the amount is calculated for each sales rep. The resulting records are in sorted by total amount in descending order:

```
DEFINE Reps AS
SELECT SUM(Amount) AS Total
GROUP BY SalesRep
ORDER BY Total DESC
```

Using FROM to use the output of one statement as the input of another statement

By default, the source of records for an LQL statement is the result of the containing search and navigation query. You can use the FROM syntax to use the output of a previous statement as the input of a subsequent statement.

For example, a statement might compute the total number of sales transactions for each quarter and sales rep. To then compute the average number of transactions per sales rep, a subsequent statement groups those results by quarter.

```
DEFINE RepQuarters AS
SELECT COUNT(TransId) AS NumTrans
GROUP BY SalesRep, Quarter ;

RETURN Quarters AS
SELECT AVG(NumTrans) AS AvgTransPerRep
FROM RepQuarters
GROUP BY Quarter
```

The `RepQuarters` statement generates a list of records. Each record contains the values `{ SalesRep, Quarter, NumTrans }`. For example:

```
{ J. Smith, 09Q1, 10 }
{ J. Smith, 09Q2, 3 }
{ F. Jackson, 08Q4, 10 }
...
```

The `Quarters` statement then uses the results of the `RepQuarters` statement to generate a list with the values `{ Quarter, AvgTransPerRep }`. For example:

```
{ 08Q4, 10 }
{ 09Q1, 4.5 }
{ 09Q2, 6 }
...
```

Using PAGE to return a subset of the results

By default, a statement returns all of the results. In some cases, it is useful to request results in smaller increments for presentation to the user (such as presenting the sales reps ten at a time, with links to page forward and backward).

The `PAGE(i,n)` operator limits the returned records to `n` records starting with the record at index `i`.

For example, the following statement groups the records by SalesRep, and then returns the 11th through 20th results:

```
DEFINE Reps AS
GROUP BY SalesRep
PAGE(10,10)
```

You can also use paging in combination with `ORDER BY` to generate "top-k" type queries. This example returns the top 10 sales reps by total sales:

```
DEFINE Reps AS
SELECT SUM(Amount) AS Total
GROUP BY SalesRep
ORDER BY Total DESC
PAGE(0,10)
```

For Latitude Studio components such as the Results Table that provide their own paging capabilities, you should not use the LQL paging statement. Use caution with `PAGE` statements within Latitude Studio components, and always test to see if there is an incompatibility.

Using inter-statement references

Multiple LQL sub-queries can be specified within the context of a single navigation query, each corresponding to a different analytical view, or to a sub-total at a different granularity level.

You can nest statements within one another to create layered LQL.

Expressions also can use values from other computed statements. This is often useful when coarser subtotals are required for computing analytics within a finer-grained bucket.

For example, when computing the percent contribution for each sales representative in a given year, you must also calculate the overall total for the year. You can use inter-statement references to create these types of queries.

Syntax for inter-statement references

The syntax for an inter-statement reference is:

```
<LookupExpr>   ::=  <Key>[<LookupList>].<Key>
<LookupList>   ::=  <empty>
               ::=  <SimpleExpr> [,<LookupList>]
```

The square brackets are used to identify the record set and grouping attribute, and the dot is used to identify the field.

Example of referencing a value from another statement

For example, suppose we want to compute the percentage of sales per ProductType per Region. One aggregation computes totals grouped by Region, and a subsequent aggregation computes totals grouped by Region and ProductType.

This second aggregation would use expressions that referred to the results from the Region aggregation. That is, it would allow each Region and ProductType pair to compute the percentage of the full Region subtotal represented by the ProductType in this Region.

```
DEFINE RegionTotals AS
SELECT SUM(Amount) AS Total
GROUP BY Region

RETURN ProductPcts AS
SELECT
  100 * SUM(Amount) / RegionTotals[Region].Total AS PctTotal
GROUP BY Region, ProductType
```

The first statement computes the total product sales for each region. The next statement then uses the RegionTotals results to determine the percentage for each region, making use of the inter-statement reference syntax.

- The bracket operator indicates to reference the RegionTotals result that has a group-by value equal to the ProductPcts value for the Region attribute.
- The dot operator indicates to reference the Total field in the specified RegionTotals record.

Example of referencing a value within the current statement

The above example references values in a separate statement, but you can also use the reference operator to reference values within the current statement.

This is useful for computing trends that change over time, such as year-on-year sales change, which could be expressed as:

```
RETURN YearOnYearChange AS
SELECT SUM(Amount) AS TotalSales,
  SUM(Amount) - YearOnYearChange[Year-1].TotalSales AS Change
GROUP BY Year
```

Working with date and time values in LQL

LQL also provides functions for working with date, dateTime, and duration attribute types.

Using TRUNC to round down dateTime values

The TRUNC function can be used to round a dateTime value down to a coarser granularity. For example, this may be useful when you want to group your statement results data for each quarter using a dateTime attribute .

The syntax of the TRUNC function is:

```
<TruncExpr>     ::=  TRUNC(<expr>,<DateTimeUnit>)
<dateTimeUnit>  ::=  SECOND | MINUTE | HOUR |
                     DATE | WEEK | MONTH | QUARTER | YEAR
```

For example, the dateTime attribute TimeStamp has a value representing 10/13/2009 11:35:12.000. The list below shows the results of using the TRUNC operator to round the TimeStamp value at each level of granularity. The values are displayed here in a format that is easier to read - the actual values would use the standard Endeca dateTime format.

```
TRUNC("TimeStamp", SECOND)  = 10/13/2009 11:35:12.000
TRUNC("TimeStamp", MINUTE)  = 10/13/2009 11:35:00.000
TRUNC("TimeStamp", HOUR)    = 10/13/2009 11:00:00.000
TRUNC("TimeStamp", DATE)    = 10/13/2009 00:00:00.000
TRUNC("TimeStamp", WEEK)    = 10/08/2009 00:00:00.000
```

```
TRUNC("TimeStamp", MONTH)   = 10/01/2009 00:00:00.000
TRUNC("TimeStamp", QUARTER) = 10/01/2009 00:00:00.000
TRUNC("TimeStamp", YEAR)    = 01/01/2009 00:00:00.000
```

Here is a simple example of using this functionality. In the following statement, the total value for the Amount attribute is grouped by quarter. The quarter is obtained by using the TRUNC operation on the TimeStamp attribute:

```
RETURN Quarters AS
SELECT SUM(Amount) AS Total,
       TRUNC(TimeStamp, QUARTER) AS Qtr
GROUP BY Qtr
```

Using EXTRACT to extract a portion of a dateTime value

The EXTRACT function extracts a portion of a dateTime value, such as the day of the week or month of the year. This can be useful in situations where the data must be filtered or grouped by a slice of its timestamps, for example to compute the total sales that occurred on any Monday.

The basic syntax of the EXTRACT function is:

```
<ExtractExpr>   ::=  EXTRACT(<expr>,<DateTimeUnit>)
<DateTimeUnit>  ::=  SECOND | MINUTE | HOUR | DAY_OF_WEEK |
                     DAY_OF_MONTH | DAY_OF_YEAR | DATE | WEEK |
                     MONTH | QUARTER | YEAR
```

For example, the dateTime attribute "TimeStamp" has a value representing 10/13/2009 11:35:12.000. The following list shows the results of using the EXTRACT operator to extract each component of that value:

```
EXTRACT("TimeStamp", SECOND)       = 12
EXTRACT("TimeStamp", MINUTE)       = 35
EXTRACT("TimeStamp", HOUR)         = 11
EXTRACT("TimeStamp", DATE)         = 13
EXTRACT("TimeStamp", WEEK)         = 40   /* Zero-indexed */
EXTRACT("TimeStamp", MONTH)        = 10
EXTRACT("TimeStamp", QUARTER)      = 3    /* Zero-indexed */
EXTRACT("TimeStamp", YEAR)         = 2009
EXTRACT("TimeStamp", DAY_OF_WEEK)  = 4    /* Zero-indexed */
EXTRACT("TimeStamp", DAY_OF_MONTH) = 13
EXTRACT("TimeStamp", DAY_OF_YEAR)  = 286  /* Zero-indexed */
```

Here is a simple example of using this functionality. The following statement groups the total value of the Amount attribute by quarter, and for each quarter computes the total sales that occurred on a Monday (DAY_OF_WEEK=1):

```
RETURN Quarters AS
SELECT SUM(Amount) AS Total
       TRUNC(TimeStamp, QUARTER) AS Qtr
WHERE EXTRACT(TimeStamp,DAY_OF_WEEK) = 1
GROUP BY Qtr
```

Using arithmetic operations on date and time values

In addition to using the TRUNC and EXTRACT functions, you also can use normal arithmetic operations with date and time values

For example, you could:

- Add a duration to a time or a dateTime to obtain a new time or dateTime.
- Subtract two times or dateTimes to obtain a duration.

- Add or subtract two durations to obtain a new duration.

Handling of characters in LQL

LQL accepts all Unicode characters.

```
<Literal>    ::=  <StringLiteral>  |  <NumericLiteral>
```

String literals	String literals must be surrounded by single quotes. Embedded single quotes and backslashes must be escaped by backslashes. Examples: `'jim'` `'àlêx\'s house'`
Numeric literals	Numeric literals can be integers or floating point numbers. However, they do not support exponential notation, and they cannot have trailing f|F|d|D to indicate float or double. `34` `.34`
Identifiers	An identifier must be enclosed in double quotes if: • The identifier contains characters other than letters, digits, and underscores. • The identifier starts with a digit. • The identifier uses the same name as an LQL keyword. For example, if an attribute is named WHERE or GROUP. If an identifier is in quotes, then you must use a backslash to escape double quotes and backslashes. Examples: `àlêx4` `"4th street"` `"some ,*#\" funny \\;% characters"`

How LQL handles special data values

Handling null attribute values

If an attribute value is missing for a record, then the attribute is referred to as being null. For example, if a record does not contain an assignment for a Price attribute, LQL defines the Price value as null.

The following table outlines how LQL handles null values for each type of operation:

Type of operation	How LQL handles null values
Arithmetic operations and non-aggregating functions	The value of any operation on a null value is also defined as null. For example, if a record has a value of 4 for Quantity and a null value for Price, then the value of `Quantity + Price` is considered to be null.
Aggregating functions	LQL ignores records with null values. For example, if there are 10 records, and 2 of them have a null value for a Price attribute, all aggregating operations ignore the 2 records, and instead compute their value using only the other 8 records. If all 10 records have a null Price, then most aggregations such as `SUM(Price)` also result in null values. The exceptions are `COUNT` and `COUNTDISTINCT`, which return zero if all the records have a null value. (That is, the output of `COUNT` or `COUNTDISTINCT` is never null.)
Grouping expressions	LQL ignores any record that has a null value in any of the group keys, and does not consider the record to be present in any group.
Filters	When doing a comparison against a specific value, the null value will not match the specified filter. For example, if record A has price 5, and record B has no price value, then: • `WHERE price = 5` matches A • `WHERE NOT(price = 5)` matches B • `WHERE price <> 5` matches neither A nor B • `WHERE NOT(price <> 5)` matches both A and B • `WHERE price = 99` matches neither A nor B • `WHERE NOT(price = 99)` matches both A and B • `WHERE price <> 99` matches A • `WHERE NOT(price <> 99)` matches B
Sorting	For any sort order specified, LQL returns: 1. Normal results 2. Records for a NaN value 3. Records with a null value

Handling of NaN, inf and -inf results

Operations in LQL adhere to the conventions for NaN, `inf` and `-inf` defined by the IEEE 7540 2008 standard for handling floating point numbers.

In cases when it has to perform operations involving floating point numbers, or operations involving division by zero or null values, LQL expressions can return NaN, `inf`, and `-inf` results.

For example, NaN, `inf` and `-inf` values could arise in your LQL calculations when:

- A zero divided by zero results in NaN
- A positive number divided by zero results in `inf`
- A negative number divided by zero results in `-inf`

For most operations, LQL treats NaN, `inf` or `-inf` values the same way as any other value.

For the following special cases, however, you may find it useful to know how LQL defines these special values.

Type of operation	How LQL handles `NaN`, `inf` and `-inf`
Arithmetic operations	Arithmetic operations with `NaN` values result in `NaN` values.
Filters	`NaN` values do not pass filters. Any comparison involving a `NaN` value is false.
Sorting	`NaN` is treated as "less than" `-inf` (`NaN < -inf`). For any sort order specified, LQL returns: 1. Normal records 2. Records with a NaN value 3. Records with a null value

Handling of records with multiple values for an attribute

An attribute may allow a record to have multiple values.

To show how LQL handles these types of records, for a record tagged with both `Blue` and `Green`:

- `WHERE Color = Blue` matches the record
- `WHERE Color <> Blue` matches the record
- `WHERE NOT(Color = Blue)` does not match the record
- `WHERE NOT(Color <> Blue)` does not match the record

Samples of typical types of LQL queries

Here are some sample queries for a number of representative use cases.

These examples are based on sales transaction data with records consisting of the following attributes:

```
{ TransId, ProductType, Amount, Year, Quarter, Region,
  SalesRep, Customer }
```

For example:

```
{ TransId = 1, ProductType = "Widget", Amount = 100.00,
  Year = 2009, Quarter = "09Q1", Region  = "East",
  SalesRep = "J. Smith", Customer = "Customer1" }
```

Top-k LQL example

This LQL statement calculates the best 10 SalesReps based on total sales from the first quarter of 1991:

```
RETURN BestReps AS
SELECT SUM(Amount) AS Total
WHERE Quarter = '09Q1'
GROUP BY SalesRep
ORDER BY Total DESC
PAGE(0,10)
```

Subset comparison LQL example

For each Quarter and Region, this example computes the percent of sales for which the ProductType is "Widgets". It combines totals on all elements of a grouping (total sales) with totals on a filtered set of elements (sales for "Widgets" only).

```
RETURN Results AS
SELECT
  (SUM(Amount) WHERE ProductType='Widgets') /
   SUM(Amount) AS PctWidgets
GROUP BY Quarter, Region
```

Nested aggregation LQL example

This example computes the average number of transactions per sales representative grouped by Quarter and Region.

This query represents a multi-level aggregation. First, transactions must be grouped into sales reps to get per-rep transaction counts. Then these rep counts must be aggregated into averages by quarter and region.

```
DEFINE DealCount AS
SELECT COUNT(TransId) AS NumDeals
GROUP BY SalesRep, Quarter, Region ;

RETURN AvgDeals AS
SELECT AVG(NumDeals) AS AvgDealsPerRep
FROM DealCount
GROUP BY Quarter, Region
```

Example of using an IN filter for segmentation (pie chart)

This query shows how the IN filter can be used to populate a pie chart showing sales divided into six segments: one segment for each of the five largest customers, and one segment showing the aggregate sales for all other customers.

The first statement gathers the sales for the top five customers, and the second statement aggregates the sales for all customers not in the top five.

```
RETURN Top5 AS SELECT
SUM(Sale) AS Sales
GROUP BY Customer
ORDER BY Sales DESC
PAGE(0,5);

RETURN Others AS SELECT
SUM(Sale) AS Sales
```

```
WHERE NOT [Customer] IN Top5
GROUP
```

Inter-statement references LQL example

This example computes for each quarter the percentage of sales for each product type.

This query requires calculating information in one statement in order to use it in another statement.

To compute the sales of a given product as a percentage of total sales for a given quarter, the quarterly totals must be computed and stored. The calculations for quarter/product pairs can then retrieve the corresponding quarterly total.

```
DEFINE QuarterTotals AS
SELECT SUM(Amount) AS Total
GROUP BY Quarter ;

RETURN ProductPcts AS
SELECT
  100 * SUM(Amount) / QuarterTotals[Quarter].Total AS PctTotal
GROUP BY Quarter, ProductType
```

Chapter 14

Creating Links Between Pages in Latitude Studio

For components that allow refinement or linking, power users can configure the links to target a different page of the Latitude Studio application.

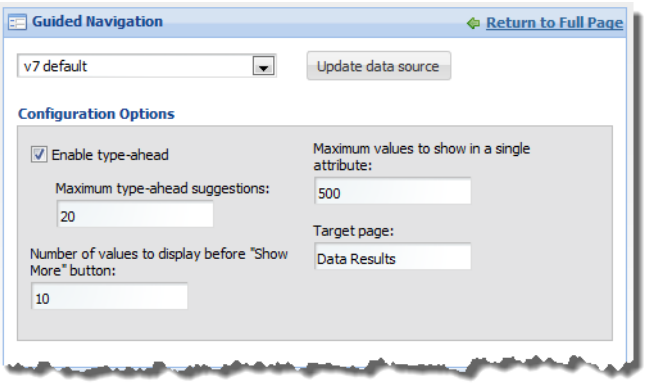
About page transitions

Page transitions allow a component on one page in your Latitude Studio application to pass data to a component on another page.

For example, a **Results Table** component on one page could be configured to target the **Record Details** component on a different page.

On the edit view of the source component, power users specify the target for a page transition.

For example, when users select an attribute on this **Guided Navigation** component, the refinement is applied to the **Data Results** page.



Page transition syntax

Power users can target a page using a full context path or a relative context path.

Relative context paths

For a relative context path, the power user only specifies the name of the page. The default context path is then added in front of the target page name.

The default context is set using the Framework Setting **df.viewTransitionDefaultContext**. The default value for the setting is `/web/guest/`. If your application is created within a Liferay Portal community, you can change the setting to be the path to that community.

For example, if the user enters `Analyze` in the target page field for a component, and the default context path is `/web/my-community/`, the end user is redirected to `/web/my-community/Analyze`.

Full context paths

For a full context path, the entire path is provided, and the default context path setting is ignored.

For example, for the following target value:

```
/web/spend/Analyze
```

no matter what the default context path is, end users are redirected to the `/web/spend/Analyze` page.

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.
- *tabNumber* is the number (1, 2, 3, etc.) of the tab to select.

So for example, for the following target value:

```
/web/spend/Analyze#Sales Numbers[1]
```

- The end user is redirected to the `/web/spend/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:

```
/web/spend/Analyze#Sales Numbers[1]::Quarterly Forecast[2]
```

- The user is redirected to the `/web/spend/Analyze` page.
- On the **Sales Numbers** tabbed component, tab 1 is selected.
- On the **Quarterly Forecast** tabbed component, tab 2 is selected

Creating page transitions using component IDs

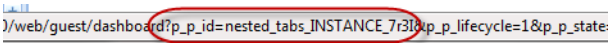
In order to implement certain kinds of page transitions, you need to use component IDs, rather than names.

Because the double colon (::) is part of the page transition 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 page transition target, 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 status bar at the bottom of the screen.
2. Extract the `p_p_id` parameter from the URL.



For example, for the following target value:

```
/web/spend/Analyze#nested_tabs_INSTANCE_0CbE[2]::nested_tabs_INSTANCE_Ja6E[1]
```

- The end user is redirected to the `/web/spend/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.

Chapter 15

Using Deep Linking to Create Links from External Sites

About deep linking

Deep linking allows users to construct URLs to navigate from an external site to a target page in Latitude Studio. These URLs are essentially external bookmarks to Latitude Studio.

Pages targeted by deep linking inbound portal URLs must already exist within Latitude Studio, and contain zero or more Latitude components bound to one or more data sources.

The URLs encode the data source filter state for any or all of the data sources defined in the application.

Any query filter supported by the data source JSON files is supported in the URL. Navigation by attribute values is also supported.

Users construct these URLs based on a query URL template, which define filters for a number of data sources for that user request.

When implementing deep linking, keep the following restrictions in mind:

- Only attribute keys (not display names) will work in deep links. Any deep link created using display names will fail.
- If non-supported filters are specified, the filters are ignored, but no error message is issued. The non-supported filters are `EQLFilter` and `RecordAggregator`.

Format of the deep linking URL

Deep linking uses the `deeplink` URL query parameter to create external bookmarks to the Latitude Studio. Deep links can include record filters and can set the current state of a tabbed container.

The general form of the URL is:

```
http://portalhost:port/path/to/page?deeplink=JSON-ARRAY-OF-DATASOURCE-STATES
```

The following example includes a record filter to only include records with a Designation of Best Buy:

```
http://localhost:8080/web/guest?deeplink=[{"default":
{"queryFunctions":[{"class":"RecordFilter","recordFilter":
"Designation:BestBuy"}]}}]
```

If the page you are navigating to includes a **Tabbed Component Container**, then you can use the `pageTransitionTabState` parameter to direct the user to a specific tab. To do this:

```
http://portalhost:port/path/to/page?com.endeca.discovery.
pageTransitionTabState=tabcomponentname[tabnumber]&deeplink=JSON-ARRAY-OF-
DATASOURCE-STATES
```

Where:

- *tabcomponentname* is the name of the component.
- *tabnumber* is the number of the tab.

In this example, in addition to providing a record filter, the link directs the user to the second tab of the Stores tabbed component:

```
http://localhost:8080/web/guest??com.endeca.discovery.
pageTransitionTabState=Stores[2]&deeplink=[{"default":
{"queryFunctions":[{"class":"RecordFilter","recordFilter":
"Designation:BestBuy"}]}}]
```

Syntax for the deeplink parameter

The `deeplink` parameter value is a JSON array of data source states keyed by data source ID.

Data source state consists of query functions and `NavByValue` JSON objects. The syntax is as follows:

```
deepLink=[
  { "dataSourceId":
    {
      "queryFunctions": [data source json as defined elsewhere],
      "navByValue": {
        "attribute1":"attributevalue1",
        "attribute2":"attributevalue2",
        "attribute3":["attributevalue","attributevalue","attributevalue"]
      }
    }
  },
  { "dataSourceId2": ...},
  { "dataSourceId3": ...}
]
```

Below is an example of the JSON array:

```
http://localhost:8080/web/guest?deeplink=[
{
 "default": {
  "queryFunctions":[
  {
   "class":"RecordFilter",
   "recordFilter":"Designation:Best Buy"
  },
  {
   "class":"RangeFilter",
   "property":"P_Price","rangeOperator":"BTWN","value1":"50","value2":"100"

  }
  ],
  "navByValue":{
   "Wine Type":"Red",
   "Region":"Other France",
   "Body":["Fresh","Full","Rich","Ripe"]
```

```
  }
 }
},
{ "v7-wine":{
  "queryFunctions":[
  {
   "class":"RecordFilter",
   "recordFilter":"Designation:Best Buy"
  }
  ],
     "navByValue":{
   "Wine Type":"Red",
   "Region":"Other France",
   "Body":["Fresh","Full","Rich","Ripe"]
  }
}
]
```

For more information about data source syntax in general, see the data sources chapter.

Note: Microsoft Internet Explorer has a URL character limit of 2083 characters. Be aware of this limit when creating a deep linking URL with a large number of filters.

About using NavByValue filters to refine deep linking queries

A `NavByValue` filter is a convenient way to refine a query using attribute values.

To do the same refinement with a `RefinementFilter`, you would need to look up the attribute value ID for each attribute value in the refinement and then pass the IDs into the `RefinementFilter`. Further, there are different ways to look up attribute value IDs depending on the version of the MDEX Engine being queried.

The `NavByValue` filter handles these steps for you.

A `NavByValue` filter allows the deep link to both:

- Refine the query using attribute values instead of attribute value IDs.
- Reflect query refinements in other components on the page. This is similar to making refinements through the **Guided Navigation** component.

Examples of deep linking URLs

Here are some examples of deep linking URLs.

Record filter: The following URL applies a record filter for wines from the 2000 vintage:

```
http://localhost:8080/web/guest/my-page?deeplink=[{"default":
{queryFunctions:[{"class":"RecordFilter","recordFilter":"Vintage:2000"}]}}]
```

Page and tab transition with a record filter: This URL combines a tab transition that points to a specific nested tab on the target page with a record filter:

```
http://localhost:8080/web/guest/my-page?com.endeca.discovery.
pageTransitionTabState=Charts[2]&deeplink=[{"default":{"queryFunctions":
[{"class":"RecordFilter","recordFilter":"Vintage:1999"}]}}]
```

Clearing refinements and searches as part of a deep link

To make sure that the data source is in its default state when the link is used, with no refinements or searches applied, you pass it an empty filter.

Here is an example of a `deeplink` parameter passing an empty array.

```
http://localhost:8080/web/guest/my-page?deeplink=[{"default":{}}]
```

How security is handled with deep linking

User authentication and page authorization for the deep linking feature are handled by the Liferay Portal.

- Authentication, if required, is handled before deep link processing.
- Page authorization is handled after deep link processing. Deep link processing is performed before redirecting to a URL-specified page or tab.

Chapter 16

Recommendations for Better Performance

When building a Latitude Studio application, to prevent the application from slowing down, keep the following recommendations in mind.

Reduce the number of components per page

One way to ensure good performance is to keep the number of components per page down to the minimum needed.

Because each component makes a separate query to the MDEX, adding a component adds additional work to load the page.

Rendering each component also adds to the work.

To help avoid this issue, you can group components across multiple pages. For example, you can have one page devoted to basic data navigation, and another with charts for data analysis.

Avoid overly complex LQL queries

Another way to improve performance is to keep your LQL queries as simple as possible.

Evaluating an LQL query can take up a lot of resources, so the more complex the LQL queries on a page, the slower the page will be.

Also, the LQL queries are evaluated in parallel as much as possible, so the more LQL queries that are issued, the fewer resources each query has to execute.

Keep the LQL threshold small

Some components that use LQL, such as **Results Table** and **Chart**, include an LQL threshold setting that controls the maximum number of records to process. For better performance, you should make this value relatively small.

The components do not evaluate the LQL results until the number of records is lower than the defined threshold. A smaller threshold allows the page to perform better when working with large sets of data.

When the data has been refined to a smaller number of records, the LQL is processed and the component data is displayed.

Display the minimum number of columns needed

For components such as **Results Table**, **Cross Tab**, and **Chart**, only display the columns you need.

On a similar note, for the **Guided Navigation** component, only display the attribute groups you need, and avoid expanding attribute values by default.

Retrieving all of the data needed to generate a large number of columns can cause the components to render much more slowly.

For the **Results Table**, you can use attribute groups to limit the number of columns shown at any given time.

Chapter 17

Using Liferay Components in Your Application

Latitude Studio includes a number of Liferay components in its installation by default. You can integrate these components with Latitude components in order to build a richer application.

Liferay component support

The level of documentation and support Endeca provides for the Liferay components included in Latitude Studio differs from that provided for Latitude components.

Because these components were not developed by Endeca, Endeca cannot control their interface or guarantee that they will be available in subsequent versions. Endeca provides only high-level documentation for Liferay components.

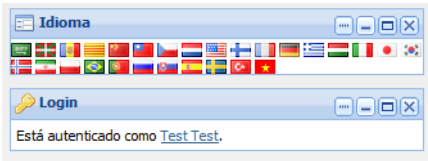
For more detailed information, consult the Liferay documentation and forums, available at *http://www.liferay.com.*

Changing the locale of the server

You can use the **Languages** component to change the locale of the server.

The **Languages** component is available from the **Tools** section of the **Add Component** menu.

From the **Language** component, to select an alternate language, click the flag icon associated with your target language.



When you select a different language, Latitude Studio displays the component messages from your resource bundle in your target language. Because the portal itself is also localized, menus and other portal controls also display in your target language.

Using Liferay Web Content Management components

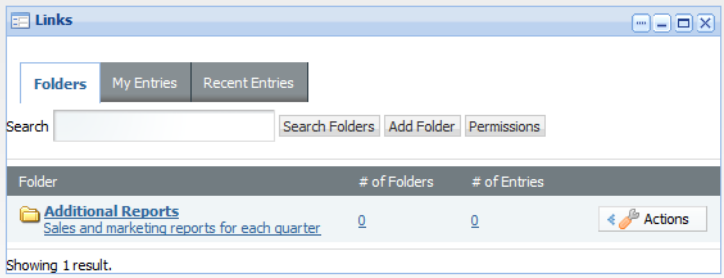
You can use Liferay Web Content Management (WCM) components, available from the **Content Management** section of the **Add Component** menu, to integrate document management and publishing capabilities into your application.

These components are often used together to configure, display, and save links to Web content.

Links component

The **Links** component allows users to save and manage their own Web content links in folders.

These links can be tagged for later search and shared with other users.

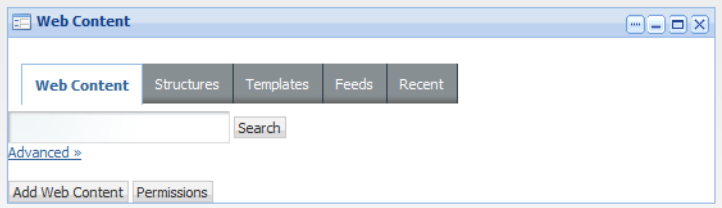


Web Content component

The **Web Content** component allows power users to configure and manage the administrative aspects of Liferay Web Content Management.

This includes:

- Enabling users to write and publish articles to the site.
- Creating article templates.
- Controlling article-creation workflow and versioning.
- Managing article search and metadata.



The content is displayed in the **Web Content Display** component.

Web Content Display component

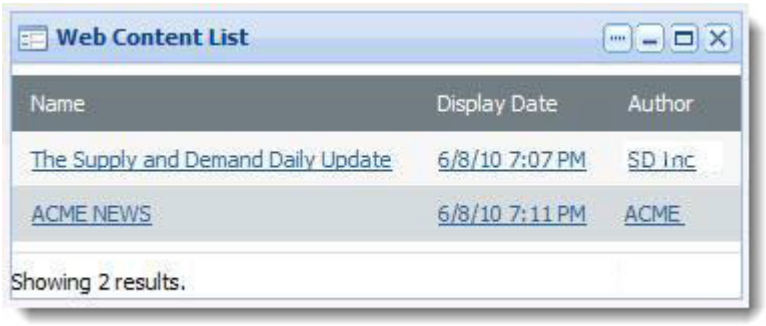
The **Web Content Display** component allows you to request and display articles in your application.

The appearance of the articles can be controlled by WCM templates, as configured in **Web Content** component.

Web Content List component

The **Web Content List** component displays a list of all Web content articles that are available in the application.

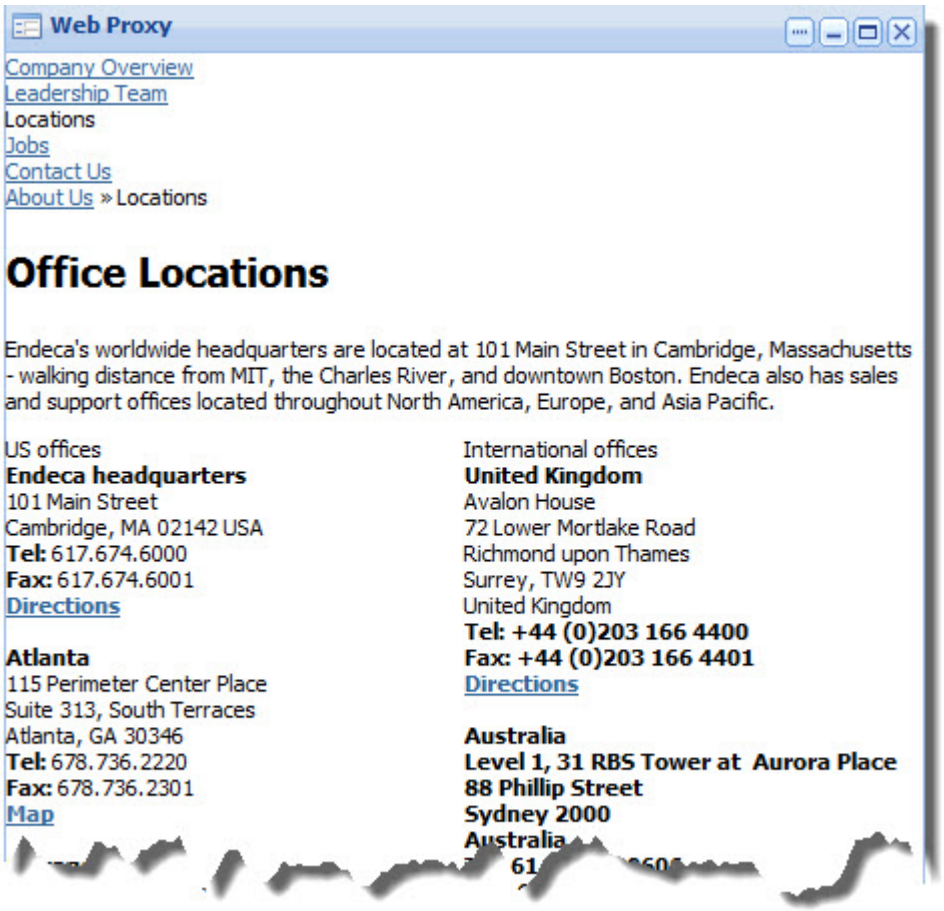
The article list is automatically updated when new articles are made available, and can be sorted by various criteria.



Web Proxy component

The **Web Proxy** component makes it possible to display any Web site as if it were a component.

Power users can configure the appearance and authentication of the content.


Web Proxy
Company Overview
Leadership Team
Locations
Jobs
Contact Us
About Us » Locations
Office Locations
Endeca's worldwide headquarters are located at 101 Main Street in Cambridge, Massachusetts - walking distance from MIT, the Charles River, and downtown Boston. Endeca also has sales and support offices located throughout North America, Europe, and Asia Pacific.
US offices
Endeca headquarters
101 Main Street
Cambridge, MA 02142 USA
Tel: 617.674.6000
Fax: 617.674.6001
Directions
Atlanta
115 Perimeter Center Place
Suite 313, South Terraces
Atlanta, GA 30346
Tel: 678.736.2220
Fax: 678.736.2301
Map
International offices
United Kingdom
Avalon House
72 Lower Mortlake Road
Richmond upon Thames
Surrey, TW9 2JY
United Kingdom
Tel: +44 (0)203 166 4400
Fax: +44 (0)203 166 4401
Directions
Australia
Level 1, 31 RBS Tower at Aurora Place
88 Phillip Street
Sydney 2000
Australia

Chapter 18

Exporting and Importing Latitude Studio Pages

To back up your application, or to migrate pages between environments, you can export and import Latitude Studio pages.

About exporting and importing pages

The export and import process is mostly used to migrate pages between environments.

Most Latitude Studio applications use multiple environments for their development process. The environments can include:

- Development, for creating new content
- Testing, to test content that is ready to go to production
- Staging, to stage new content. In some cases, the testing and staging systems are combined into a single environment.
- Production, the site available to end users

When developing new content, instead of having to recreate the content on each environment, you can export the pages from one environment, and then import them into another environment.

You also can use the export function to back up a set of pages.

The pages are exported to a LAR file. LAR stands for Liferay Archive File. See the *Liferay Administrator's Guide* for additional information on LAR files.

Note that the export and import functions only work between applications that are based on the same version of Liferay.

Exporting pages from Latitude Studio

What is included in the export?

When you export pages, the exported material only includes the pages and the components. The export does not contain any data sources.

Completing the export

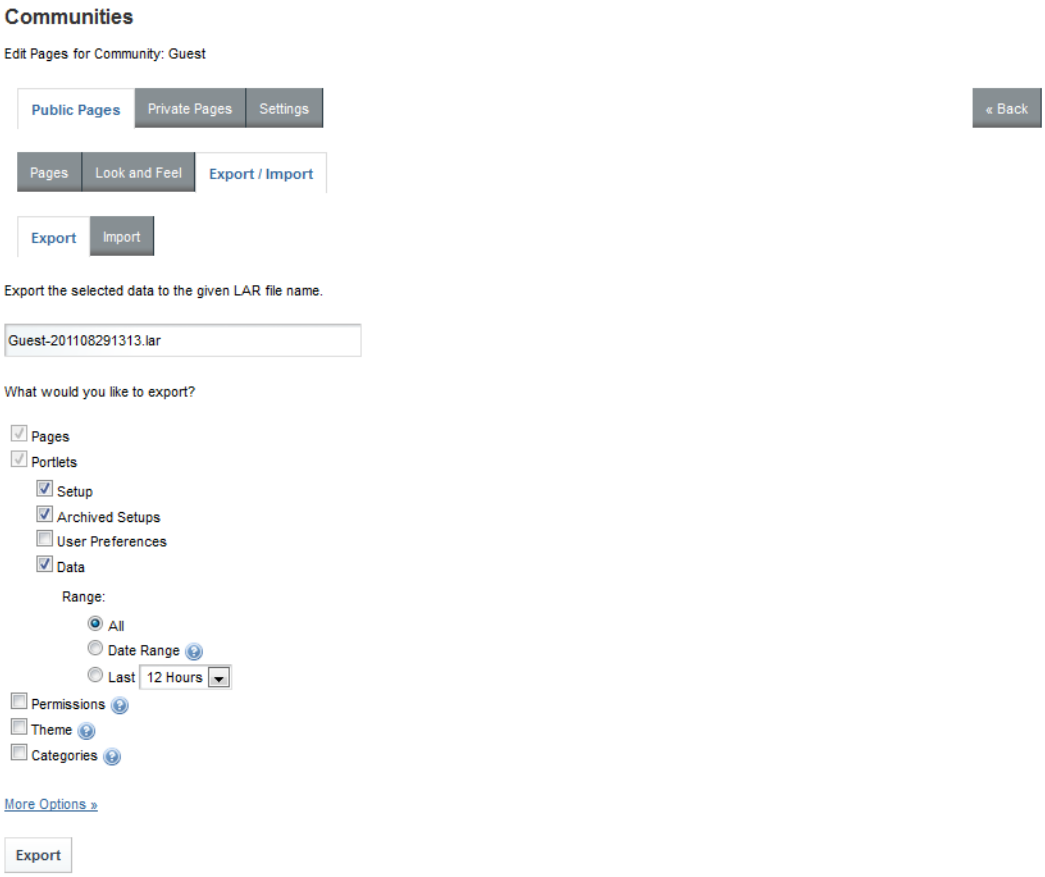
The export option is available from both the **Manage Pages** option on the Dock, and the **Communities** component on the **Control Panel**. We recommend using the option on the **Communities** component.

To export pages from Latitude Studio:

1. To display the **Manage Pages** options from the **Communities** component on the **Control Panel**:
   a) From the Dock menu, select **Control Panel**.
   b) On the **Control Panel** menu, click **Communities**.
   c) On the **Communities** page, for the community you want to export pages from, click the **Actions** button, then click **Manage Pages**.

   If you have not set up any other communities, then the only community is the **Guest** community.

2. On the **Communities** page for the selected community, click the **Export/Import** tab.
3. Click the **Export** tab. The **Export** tab contains the options for exporting the pages.



4. In the field, set the name of the LAR file to export the pages to.
5. In most cases, you can use the default settings for the export.

   Note that if you choose to export permissions, remember that you need to have the same users and user groups on the destination environment as on the source environment.

6. Click **Export**.

   You are prompted to save the resulting LAR file.

Importing pages into Latitude Studio

Ensuring that imported pages will work properly

When importing pages into Latitude Studio, make sure that both environments are based on the same version of Liferay.

In addition, to ensure that your imported pages will work correctly on the new environment, make sure that:

- You import the pages into the same community as you exported them from.

  If you import a page into a different community, then any links between pages or deep links from external sites may not work.

- The data sources used by the page components are also configured on the destination environment.

  For those data sources, the attribute group configuration also needs to be the same.

- If you are importing permissions, the same users are configured on the destination environment.

Completing the import

The import option is available from both the **Manage Pages** option on the Dock, and from the **Communities** component on the **Control Panel**. We recommend using the option on the **Communities** component.

Note that the import can only add new pages to a destination environment. It cannot update or replace existing pages. If the destination environment contains a page with the same name as a page in the LAR file, then that page will not be updated.

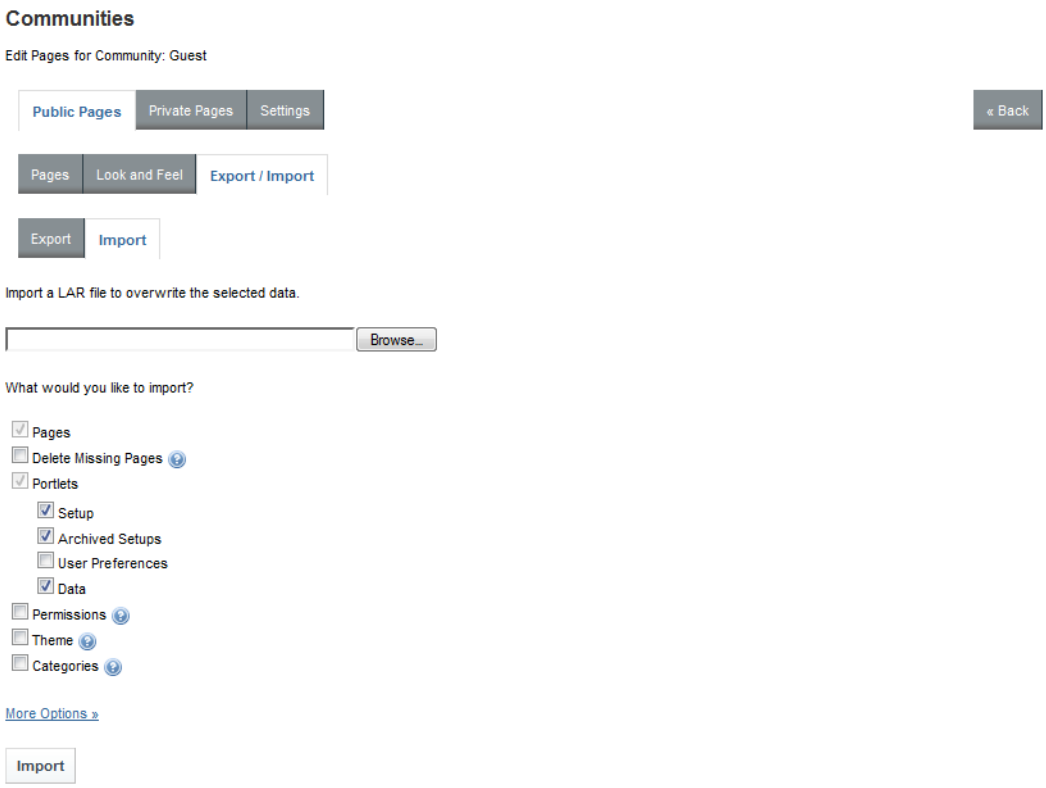
Before you import the LAR file, delete any duplicate pages that you want to replace with pages from the LAR file.

To import pages from a LAR file into Latitude Studio:

1. To display the **Manage Pages** options from the **Communities** component:
   a) From the Dock menu, select **Control Panel**.
   b) On the **Control Panel**, click **Communities**.
   c) On the **Communities** page, for the community you want to import pages into, click the **Actions** button, then click **Manage Pages**.

      If you have not set up any other communities, then the only community is the **Guest** community.

2. On the **Communities** page for the selected community, click the **Export/Import** tab.
3. Click the **Import** tab. The **Import** tab contains the options for importing pages from a LAR file.



4. To search for and select the file to import, click the **Browse** button.
5. To delete any pages on the destination environment that do not exist in the LAR file, check the **Delete Missing Pages** checkbox.

   For example, a LAR file contains the pages Welcome, Dashboard, and Search. The destination environment contains a page called Charts.

   If the **Delete Missing Pages** checkbox is checked, then when the LAR file is imported, the Charts page would be removed from the destination environment. The destination environment would only contain Welcome, Dashboard, and Search.

   If the checkbox is not checked, then when the LAR file is imported, the destination environment would contain Welcome, Dashboard, Search, and Charts.

6. After selecting the import options, to complete the import, click the **Import** button.

Part 5

Using and Configuring Latitude Components

Chapter 19

Results Components

These components provide a detailed view of records for the current refinement.

Data Explorer

About the Data Explorer component

The **Data Explorer** component displays the list of records for the current refinement. Each record contains a complete set of attribute-value pairs.

The **Data Explorer** component is designed to be a useful tool for power users. For example, they may use the component to verify newly loaded data.

Using the Data Explorer component

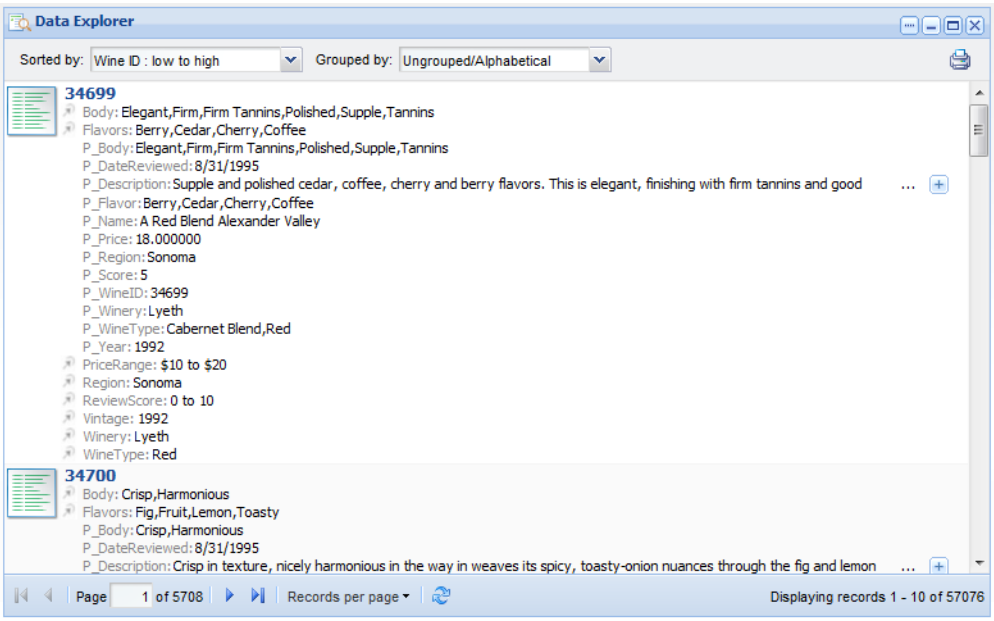
Power users use the **Data Explorer** component to view the attribute values for each record.

About the Data Explorer display

On the **Data Explorer** component, each record is displayed as the record ID followed by a complete list of attribute/value pairs.

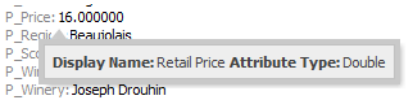
The attributes may be displayed:

- In alphabetical order
- By data type
- Within their attribute groups



In the list, the managed attributes are indicated by an icon in front of the attribute name.

The **Data Explorer** component shows the attribute key and value. To see the display name and data type for an attribute, hover the mouse over the attribute key.

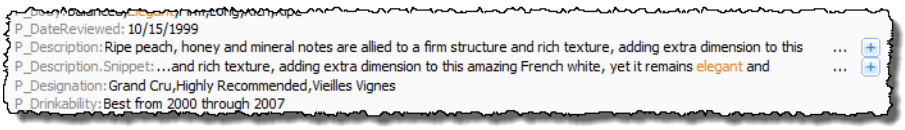


Records included in the Data Explorer component

The **Data Explorer** component reflects the current refinement state of the data.

So for example, if you have used a **Guided Navigation** component or a search to refine the data, the **Data Explorer** component only displays the matching records.

When users perform a keyword search, if snippeting is enabled for an attribute, then if the attribute value includes the search term, the search snippet for that attribute is displayed. The snippet displays the portion of the attribute value that contains the search term.



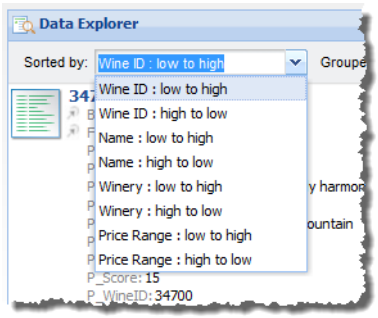
For details on configuring snippeting for searches, see the *Latitude Developer's Guide*.

Sorting the records in the Data Explorer

The **Data Explorer** can be configured to allow users to sort the list.

If users can sort the list, then at the top of the component is a **Sorted by** drop-down list.

The drop-down list contains the options for sorting the list. Each option contains the attribute to use to sort the list, and the direction in which to sort. The current sort order is highlighted.



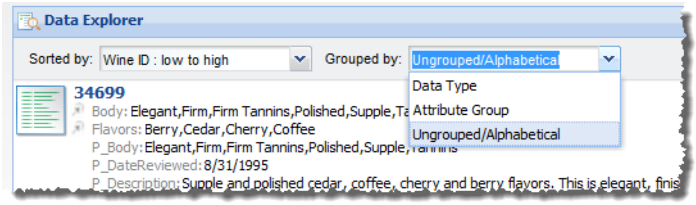
To sort the list, select a sort option.

When users use a search to refine the data, the list automatically is sorted by Search Relevance. Users can then select a different sort option from the drop-down list.

Selecting the grouping for the attributes

The **Data Explorer** component is configured with a default grouping for the attributes. End users can then select a different grouping.

To change how the attributes are grouped within each record, from the **Grouped by** drop-down list, select the new grouping option.



The options are:

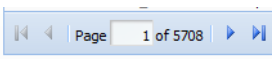
Data Type	In this mode, the attributes are grouped by the data type of the attribute value. For example, all integer values are in one group, and all string values are in another group.
Attribute Group	In this mode, the attributes are displayed within their attribute groups.
Ungrouped/Alphabetical	In this mode, the attributes are displayed in alphabetical order by attribute name. They are not grouped.

Navigating through the list

The **Data Explorer** component can be configured to allow users to navigate through the list of results.

The component is configured with a default number of records to display on a page.

If pagination is allowed, then pagination tools are displayed at the bottom of the component.



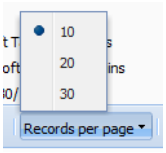
Users can use these tools to:

- Navigate to the next or previous page
- Navigate to the first or last page
- Jump to a specific page. To go to a specific page, type the page number in the field, then press **Enter**.

Setting the number of records to display per page

If the **Data Explorer** component allows navigation, then the component can include a **Records per page** button next to the pagination tools.

When users click the **Records per page** button, the list of available options is displayed. The currently selected option is highlighted.



To change the number of records per page, select the number.

Displaying the details for a Data Explorer record

Each record in the **Data Explorer** component includes a hyperlink that can be used to display the details for that record.

The hyperlink is the record ID at the top of the record.

When users click the link, a **Record Details** component is populated with the details for that record. The **Record Details** component may be on the same page as the **Data Explorer** component, or may be on a different page.

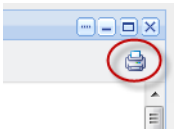
Printing the Data Explorer list

Users can print the current content of a **Data Explorer** component.

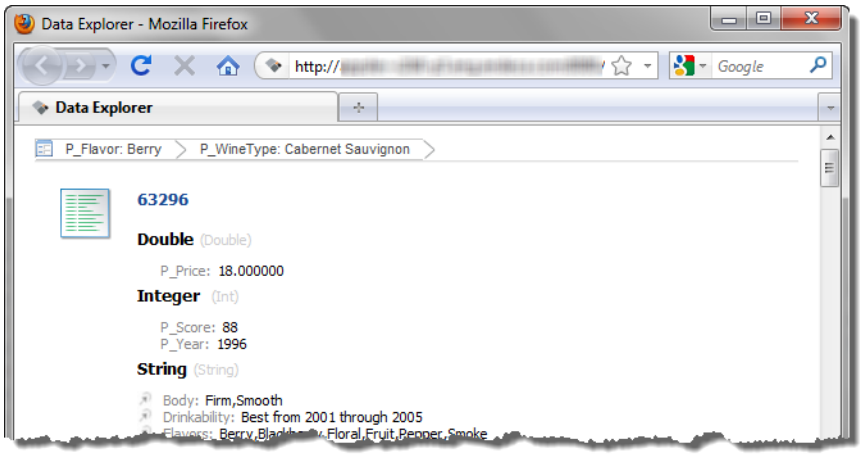
The print option only prints the records for the current refinement.

To print the **Data Explorer** list:

1. Click the print icon, located at the top right of the component.



The **Data Explorer** list is displayed in a separate browser window. Above the list are the current refinement selections.



The print dialog box also is displayed.

2. Select the printing options, and then complete the printing process.

Configuring a Data Explorer component

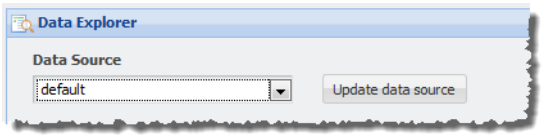
For a **Data Explorer** component, power users determine whether users can navigate through and change the sort order of the records in the list.

Selecting the data source for the Data Explorer component

The **Data Explorer** component requires a data source. When the component is first added, it displays the records for the default data source.

To select a different data source:

1. From the **Data Source** drop-down list, select the data source to use.

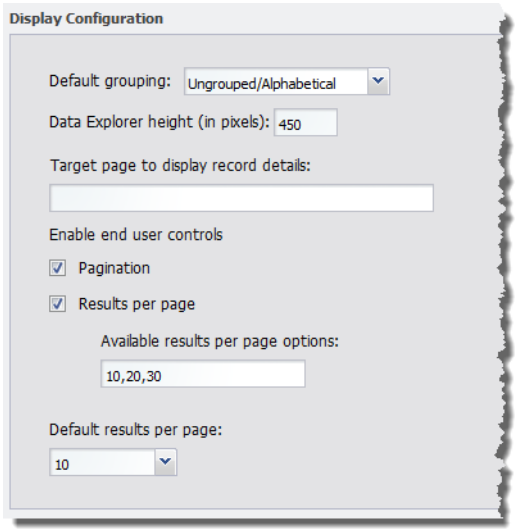


2. Click the **Update data source** button.

   The **Available Attributes** list is updated to reflect the selected data source.

Configuring the Data Explorer display and pagination options

From the edit view of a **Data Explorer** component, power users can select the default grouping for the attributes. They also can configure whether to enable pagination, and if pagination is enabled, whether to allow users to select the number of results per page.



Under **Display Configuration**, to configure the default grouping, pagination, and navigation options for a **Data Explorer** component:

1. From the **Default grouping** drop-down list, select the default grouping for the attributes. You can either:
   - Display the attributes in alphabetical order
   - Group the attributes by data type
   - Display the attributes within their attribute groups
2. To specify a height for the component, in the **Data Explorer height (in pixels)** field, type the height value in pixels.
3. In the **Target page to display record details** field, type the name of the page to display when users click the record hyperlink.

   The selected page must contain a **Record Details** component that uses the same data source.

   If you do not provide a page name, then the user stays on the current page.
4. To display the pagination bar to allow users to navigate through the entire list, check the **Pagination** checkbox. The box is checked by default.

   If the box is not checked, then users cannot navigate through the list. The component only displays a number of records equal to the value of the **Default results per page** field. To see other records, the user must further refine the data.
5. If the **Pagination** checkbox is checked, then to allow users to select the number of records to display per page, check the **Results per page** checkbox.

   The available values for the user to select from are configured in the **Available results per page options** field.
6. If the **Results per page** checkbox is checked:
   a) In the **Available results per page options** field, type a comma-separated list of available values for the number of results to display per page.

      These values are used both to populate the **Default results per page** drop-down list below the field, and the **Records per page** options for users.

      By default, the available options are 10, 20, and 30 records per page.

b) From the **Default results per page** drop-down list, select the default number of records to display per page.

The available options are configured in the **Available results per page options** field.

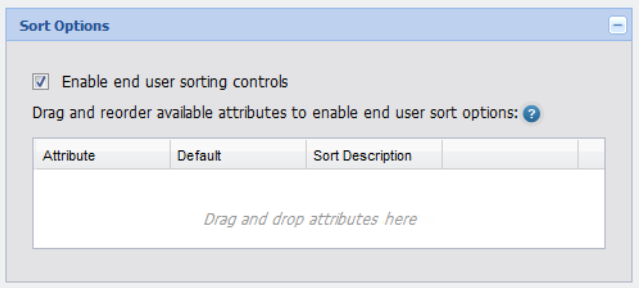
7. If the **Results per page** checkbox is not checked, then in the **Default results per page** field, type the number of records to display per page.

Configuring the sorting options for a Data Explorer component

From the edit view of a **Data Explorer** component, power users can configure whether users can change the sort order for the list.

To configure the sorting options for the component:

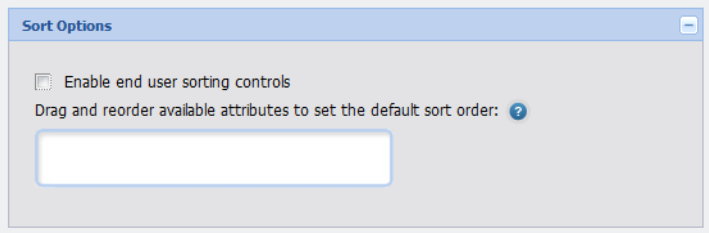
1. Under **Sort Options**:



- To allow users to set the sort order, check the **Enable end user sorting controls** checkbox. The box is checked by default.
- To not allow users to control the sort order, uncheck the checkbox.

The **Sort Options** section is updated to reflect your selection.

2. If the **Sorting** checkbox is not checked, then users cannot change the list sort order. The **Sort Options** section contains a single field.

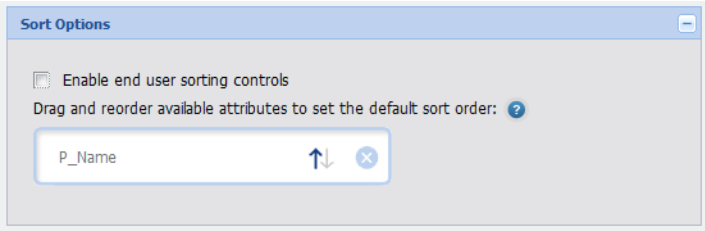


To determine how the results list is sorted, drag an attribute from the **Available Attributes** list to the **Sort Options** field.

In the **Available Attributes** list, you can use the filter field to find the attribute you want to use.

After you drop the attribute, to determine the sort order:

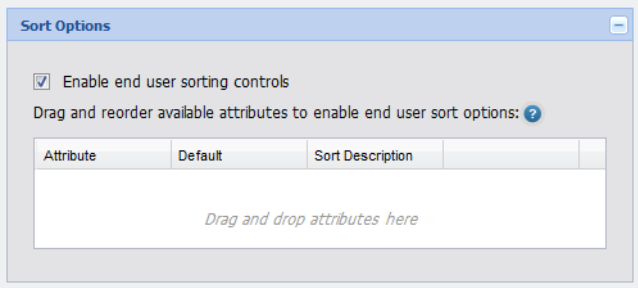
- To sort by the selected attribute in ascending order, click the up arrow. This is the default.
- To sort by the selected attribute in descending order, click the down arrow.



To remove the attribute, click the delete icon.

To replace the attribute with a different attribute, drag a different attribute from the **Available Attributes**, and then drop it on top of the currently selected attribute.

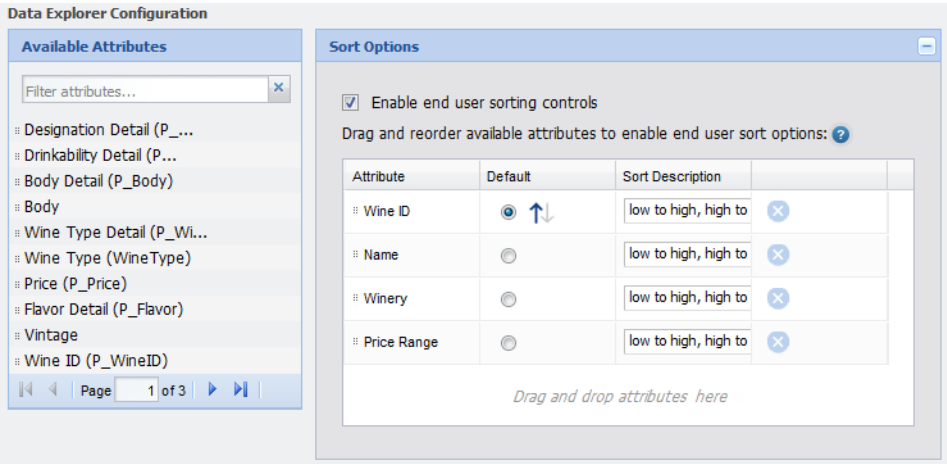
3. If the **Sorting** checkbox is checked, then the **Sort Options** section allows you to provide a list of attributes that users can use to sort the list.



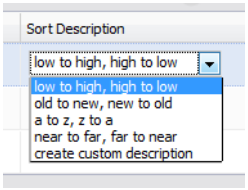
To add an attribute, drag the attribute from the **Available Attributes** to the **Sort Options** table.

In the **Available Attributes** list, you can use the filter field to find the attribute you want to use.

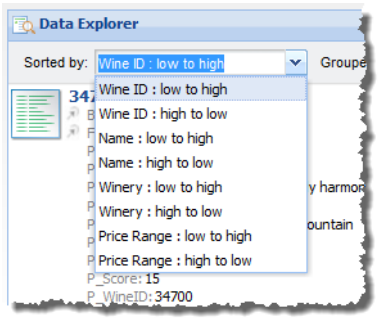
4. To determine the order of the attributes in the **Sorted by** drop-down list, drag each row up or down in the list.



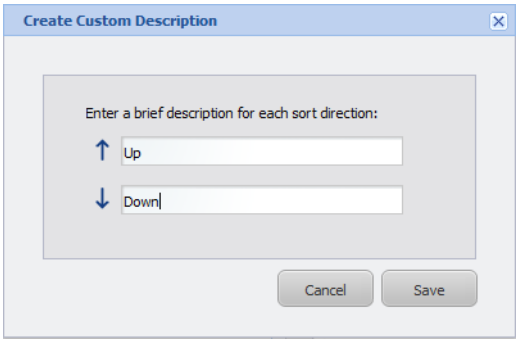
5. For the attribute you want to use for the default sort order:
   a) Click the **Default** radio button next to the attribute.
   b) To sort by that attribute in ascending order, click the up arrow.
   c) To sort by that attribute in descending order, click the down arrow.
6. From the **Sort Description** drop-down list, select the text that displays next to the attribute in the **Sorted by** drop-down list.



The **Sorted by** list includes one entry for sorting in ascending order, and one for sorting in descending order.



To create your own descriptions for ascending and descending order, from the **Sort Description** drop-down list, select **create custom description**. On the **Create Custom Description** dialog box:



a) In the first field, marked with the up arrow, type the text to use to indicate ascending order.
b) In the second field, marked with the down arrow, type the text to use to indicate descending order.
c) Click the **Save** button.

7. To remove an attribute from the list, click the delete icon for that attribute.

Saving changes to the Data Explorer configuration

In order for the **Data Explorer** configuration to take effect, you must save the configuration changes.

From the edit view of the **Data explorer** component:

1. To save the changes to the configuration, click **Save Preferences**.

   A message displays indicating that the changes were saved successfully.

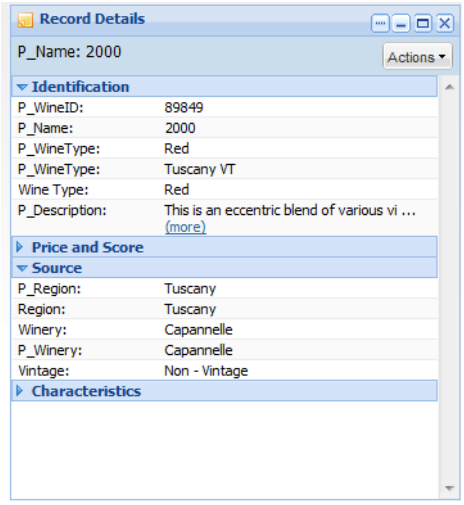
2. To exit the edit view, click **Return to Full Page**.

Record Details

About the Record Details component

The **Record Details** component displays a list of attribute values for a record selected from another component.

Power users determines the attribute groups to include, and the order in which to display the groups and the attributes within each group.



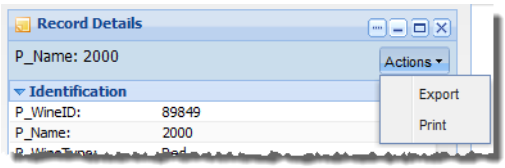
The **Record Details** component can only be used if there is another component with the same data source that allows users to display record details.

Using a Record Details component

The **Record Details** component displays a list of attribute values for a record selected from another component. The attributes are displayed within their attribute groups.

The component can also include an **Actions** menu with options for end users to:

- Export the record details to a spreadsheet. The details are exported as a single row.
- Print the record details



Configuring a Record Details component

For a **Record Details** component, power users can configure the data source, the available actions, and the list of attribute groups.

Selecting the data source for the Record Details component

The **Record Details** component requires a backing data source. When the component is first added, it is bound to the default data source.

From the edit view of the **Record Details** component, to bind a different data source to the component:

1. From the data source drop-down list, select the data source.
2. Click **Update data source**.

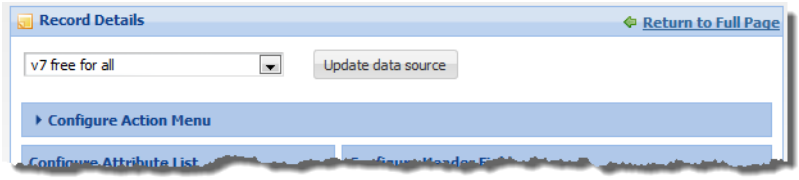
Configuring the available actions for a Record Details component

The **Record Details** component can include an **Actions** menu to allow end users to perform an action on the selected record details. Power users configure whether the menu is available, and the options to include.

The available actions are:

Export	Allows end users to export the record details to a spreadsheet.
Print	Allows end users to print the record details.

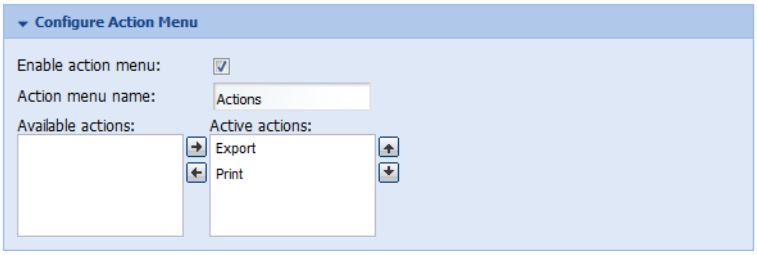
On the edit view of the **Record Details** component, you use the **Configure Action Menu** section to configure the **Actions** menu. By default, the section is collapsed, with only the heading displayed.



To configure the **Actions** menu:

1. To open the **Configure Action Menu** control, click the arrow in front of the section label.

   The fields for the **Configuration Action Menu** section are displayed.



2. To make the **Actions** menu available to end users, check the **Enable action menu** checkbox.

   If you uncheck the box, the remaining fields are disabled.

3. In the **Action menu name** field, type the name to display for the menu.

   The default name is **Actions**.

4. To enable an action for users, click the action in the **Available actions** list, then click the right arrow button.

   The action is moved to the **Active actions** list.

5. To change the display order of an action in the **Actions** menu, click the action in the **Active actions** list, then click the up or down arrow button.

   The action is moved up or down in the list.

6. To disable an action for users, click the action in the **Active actions** list, then click the left arrow button.

   The action is moved to the **Available actions** list.

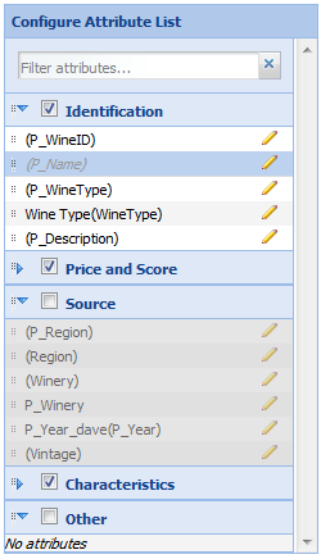
7. To save your changes, click **Save Preferences**.
8. To exit the edit view, click **Return to Full Page**.

Selecting the attribute groups to display on a Record Details component

For a **Record Details** component, power users select the attribute groups to display. By default, all of the attribute groups are displayed, and the groups and attributes are in the default sort order for the data source.

From the edit view of a **Record Details** component, to select the attribute groups and determine the group display order:

1. Under **Configure Attribute List**, to remove an attribute group from the display, uncheck its checkbox.



   To restore a hidden group, check its checkbox.

   To find a specific attribute, so that you can see which group it is in, type the attribute name in the **Filter Attributes** field. As you type, the list is filtered to only display the matching attributes.

2. To change the display order of the attribute groups, drag the group to the new location in the list.

   You can collapse the attribute groups to make them easier to work with. To expand or collapse a group, click the group name.

3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

Formatting the attribute values displayed on a Record Details component

For each attribute displayed on a **Record Details** component, end users can format the displayed value. They can select a different format, and configure specific options for the selected format type.

From the edit view of a **Record Details** component, to format the displayed attribute values:

1. Under **Configure Attribute List**, click the edit icon for the attribute.

   The **Edit Attribute Display** dialog is displayed. The default format for the attribute is based on the attribute's data type.

2. From the **Format** drop-down list, select the format to use for the attribute value, and then configure the options for the selected type.

   If you select a different format, make sure that it is an appropriate format for the value being displayed.

   The options are:

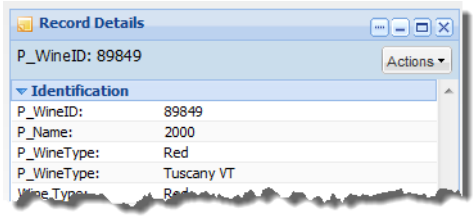
Integer	Indicates that the value is an integer. For integer values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value
String	Indicates that the value is a text string. For string values, you can configure: • Whether to change the capitalization of the text string • The number of characters after which to truncate the value
Currency	Indicates that the value is a currency value. For currency values, you can configure: • The character to use to separate the thousands • The character to use as the decimal point • The number of decimal places to display • The currency symbol to use • Whether to display the currency symbol in front of the value (**prefix**) or behind the value (**suffix**)
Decimal	Indicates to display the value as a decimal value. For decimal values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value • The character to use as the decimal point • The number of decimal places to display
Date	Indicates that the value is a date. For a date value, you can configure the format to use. The options are: • **American style (MM/DD/YYYY)**. For example, for October 20, 2010, the American style date would be 10/20/2010.

- **European style (DD/MM/YYYY)**. For example, for October 20, 2010, the European style date would be 20/10/2010.

3. To save the changes, click **OK**.

Selecting the attribute value to display in the Record Details heading

The heading of the **Record Details** component contains an attribute value. By default, the record ID is displayed.

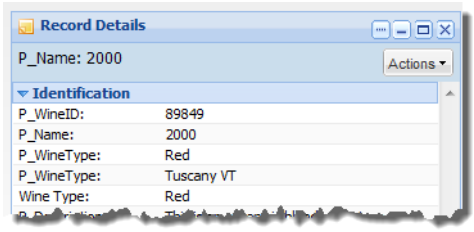


From the edit view of the **Record Details** component, to select the attribute value to display in the header:

1. To select the attribute to use, either:
   - In the **Header Field** text box, type the name of the attribute. As you type, the component verifies that the text entered is a valid attribute name.
   - From the **Attribute List**, drag the attribute into the **Header Field** text box.

2. To save your changes, click **Save Preferences**.
3. To exit the edit view, click **Return to Full Page**.

The header field of the **Record Details** table is updated to display the selected attribute value.

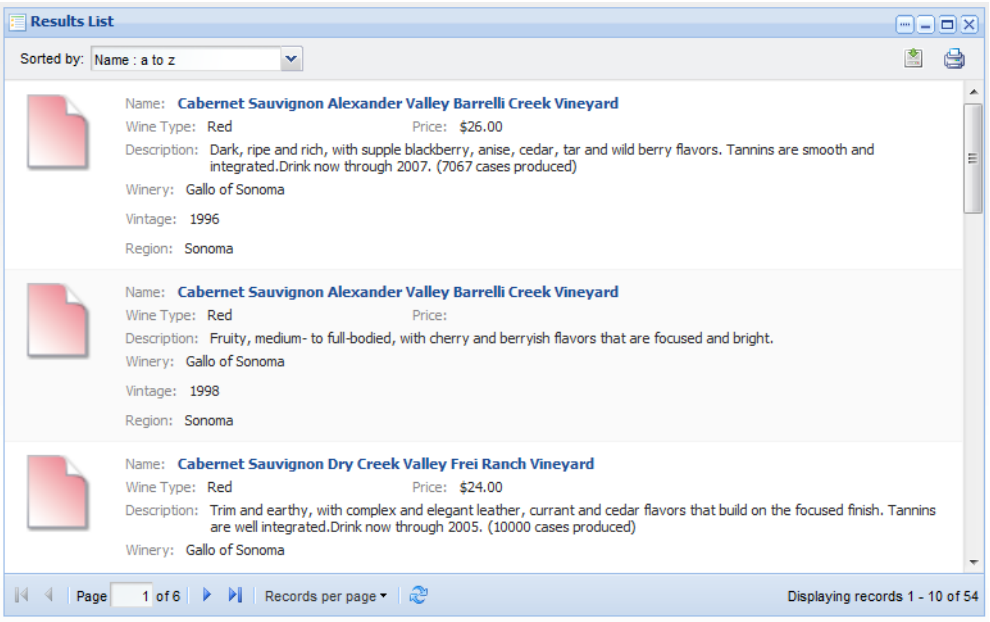


Results List

About the Results List component

The **Results List** component displays a list of records in a list format similar to regular web search results.

In the list, each record includes a selected set of attributes.

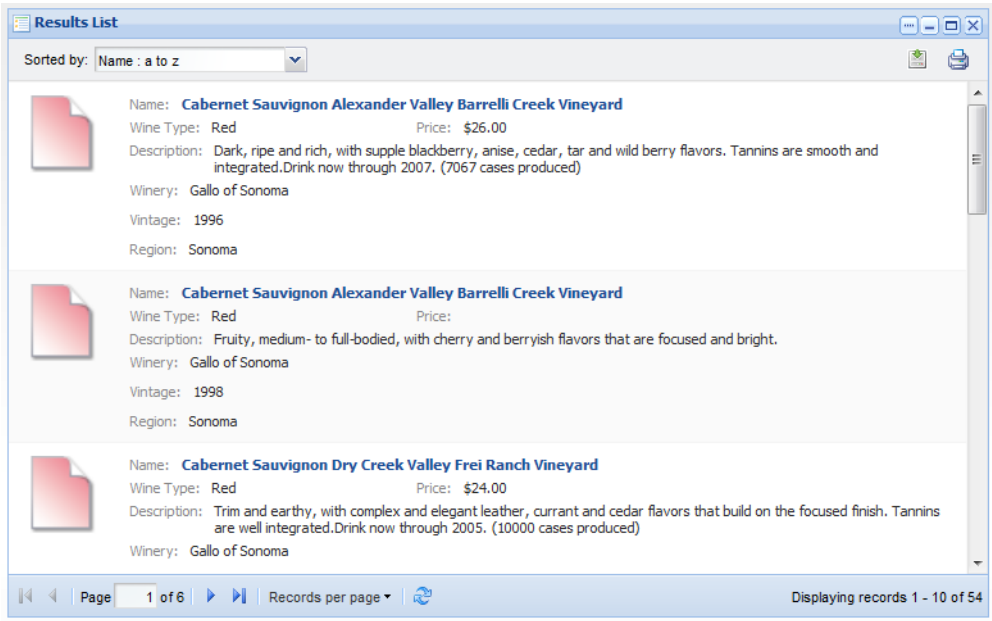


Using the Results List component

End users use the **Results List** component to view record information.

About the Results List display

On the **Results List** component, the list looks very much like the search results from a standard web search.

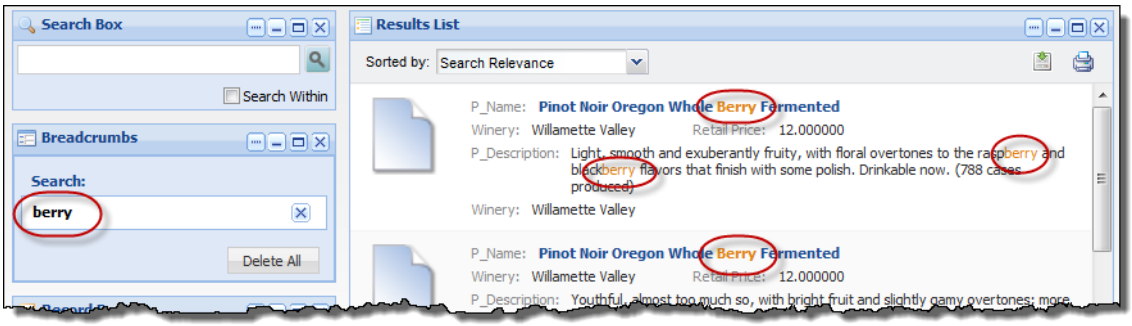


Records included in the Results List component

The **Results List** component reflects the current refinement state of the data.

So for example if an end user has used a **Guided Navigation** component or a search to refine the data, the **Results List** component only displays the matching records.

When end users use a search to refine the data, the component can be configured to highlight the search text in the attribute values.



If an attribute is configured to enable snippeting, then the search snippet is displayed instead of the full attribute value. The snippet displays the portion of the attribute value that contains the search term.

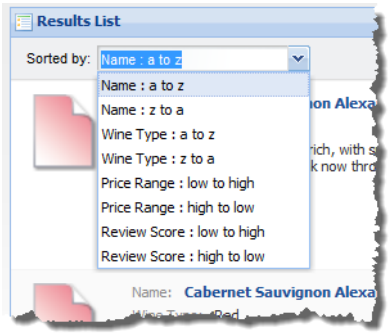
For details on configuring snippeting for searches, see the *Latitude Developer's Guide*.

Sorting the records in the Results List

The **Results List** can be configured to allow end users to sort the list.

If end users are allowed to sort the results list, then at the top of the component is a **Sorted by** drop-down list.

The drop-down list contains the options for sorting the list. Each option contains the attribute to use to sort the list, and the direction in which to sort. The current sort order is highlighted.



To sort the list, end users select a sort option.

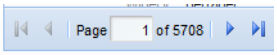
When end users use a search to refine the data, the list automatically is sorted by Search Relevance. End users can then select a different sort option from the drop-down list.

Navigating through the list

The **Results List** component can be configured to allow end users to navigate through the list of results.

The **Results List** component is configured with a default number of records to display on a page.

If the power user has configured the component to allow pagination, then pagination tools are displayed at the bottom of the component.



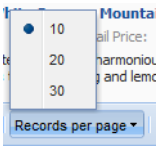
End users can use these tools to:

- Navigate to the next or previous page
- Navigate to the first or last page
- Jump to a specific page. To go to a specific page, type the page number in the field, then press **Enter**.

Setting the number of records to display per page

If the **Results List** component allows navigation, then the component can include a **Records per page** button next to the pagination tools.

When you click the **Records per page** button, the list of available options is displayed. The currently selected option is highlighted.

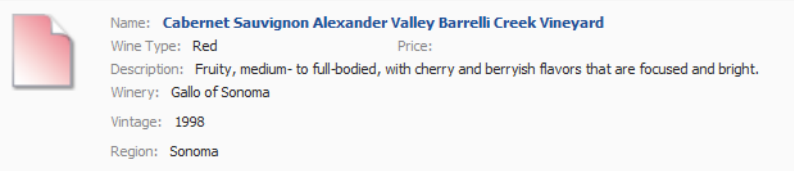


To change the number of records per page, select the number.

Displaying the details for a Results List record

Each record in the **Results List** component can include a hyperlink that can be used to display the details for that record.

The hyperlink, if included, is from the attribute at the top of the list item.



When end users click the link, a **Record Details** component is populated with the details for that record. The **Record Details** component may be on the same page as the **Results List** component, or may be on a different page.

Using attribute values to refine the Results List data

In the **Results List** component, the attribute values can be configured to allow end users to refine by that value.

If the attribute value is configured to be enabled for refinement, then when end users click the attribute value, the data on the page is refined by that value.

The selected value is added to any **Breadcrumbs** component on that page.

Using attribute values to navigate from the Results List

In the **Results List** component, the attribute values can be configured to allow end users to display a different Web page or open a file.

If that attribute value is configured to allow navigation, then when end users click the attribute value, the specified URL or file is displayed.

The URL may be configured to be specific to an attribute value.

Exporting the results list

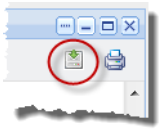
End users can export the current content of a **Results List** component to a spreadsheet file.

The export only includes:

- The records for the current refinement
- The attributes displayed in the list

To export the list from the **Results List** component:

1. Click the export icon, located at the top right of the component.



2. When the export is complete, you are prompted to save or open the spreadsheet file.

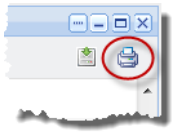
Printing the results list

End users can print the current content of a **Results List** component.

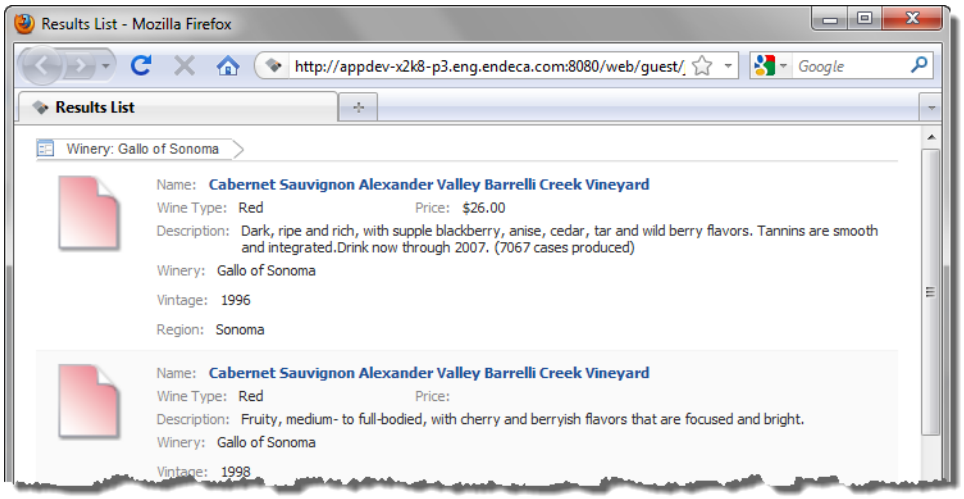
The print option only prints the records for the current refinement.

To print the **Results List**:

1. Click the print icon, located at the top right of the component.



The results list is displayed in a separate browser window. Above the list are the current refinement selections.



The print dialog box also is displayed.

2. Select the printing options, and then complete the printing process.

Configuring a Results List component

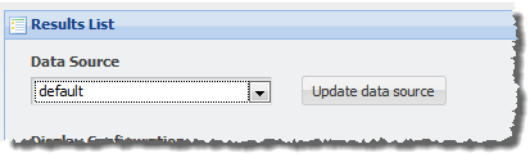
When the **Results List**component is first added, it does not display any records. Power users determine how the component is displayed and how end users can interact with each displayed attribute.

Selecting the data source for a Results List component

Each **Results List** component requires a data source.

From the edit view of a **Results List** component, to select the data source:

1. From the **Data Source** drop-down list, select the data source to use.

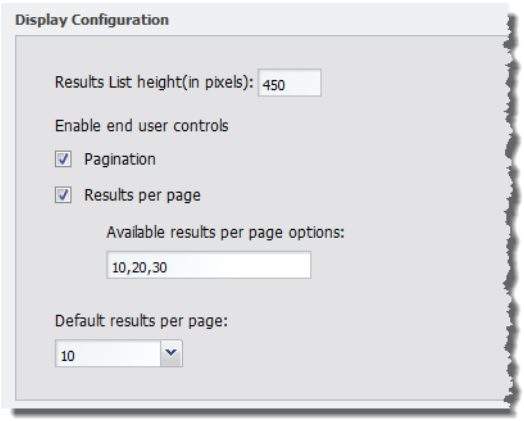


2. Click the **Update data source** button.

   The **Available Attributes** list is updated to reflect the selected data source.

Configuring pagination options for a Results List component

From the edit view of a **Results List** component, power users can configure whether to enable pagination, and if pagination is enabled, whether to allow end users to select the number of results per page.



Under **Display Configuration**, to configure the pagination and navigation options for a **Results List** component:

1. In the **Results List height(in pixels)** field, type the height in pixels for the component.
2. To display the pagination bar in order to allow end users to navigate through the entire list, check the **Pagination** checkbox. The box is checked by default.

   If the box is not checked, then end users cannot navigate through the list. The component only displays a number of records equal to the value of the **Default results per page** field. To see other records, the end user must further refine the data.

3. If the **Pagination** checkbox is checked, then to allow end users to select the number of records to display per page, check the **Results per page** checkbox.

   The available values for the end user to select from are configured in the **Available results per page options** field.

4. In the **Available results per page options** field, type a comma-separated list of available values for the number of results to display per page.

   These values are used both to populate the **Default results per page** drop-down list below the field, and the **Records per page** options for end users.

   By default, the available options are 10, 20, and 30 records per page.

5. If the **Results per page** checkbox is checked, then from the **Default results per page** drop-down list, select the default number of records to display per page.

   The available options are configured in the **Available results per page options** field.

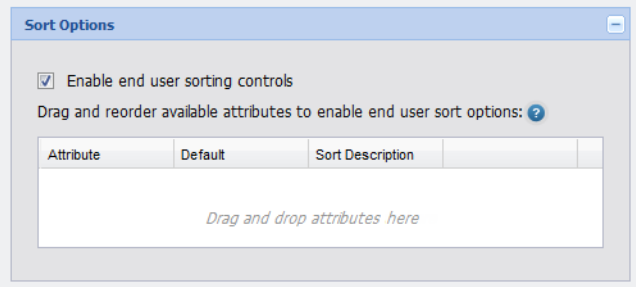
6. If the **Results per page** checkbox is not checked, then in the **Default results per page** field, type the number of records to display per page.

Configuring the sorting options for a Results List component

From the edit view of a **Results List** component, power users can configure whether end users can change the sort order for the list.

To configure the sorting options for the component:

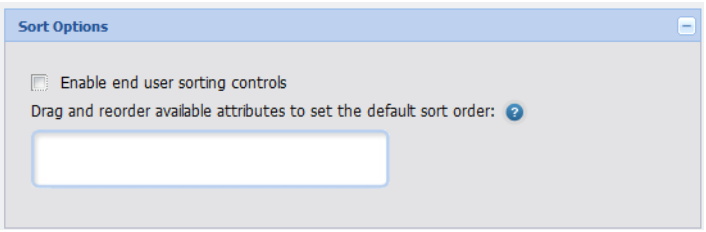
1. On the edit view, under **Sort Options**:



   - To allow end users to set the sort order, check the **Enable end user sorting controls** checkbox. The box is checked by default.
   - To not allow end users to control the sort order, uncheck the checkbox.

   The **Sort Options** section is updated to reflect your selection.

2. If the **Sorting** checkbox is not checked, then end users cannot change the list sort order. The **Sort Options** section contains a single field.

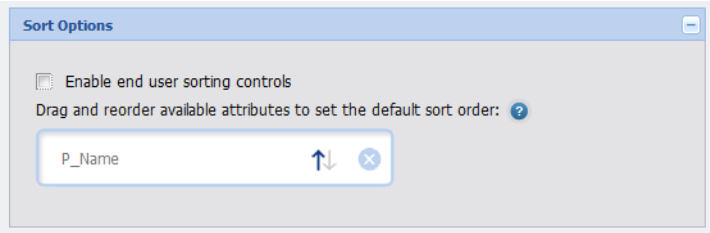


   To determine how the results list is sorted, drag an attribute from the **Available Attributes** list to the **Sort Options** field.

   In the **Available Attributes** list, you can use the filter field to find the attribute you want to use.

   After you drop the attribute, to determine the sort order:

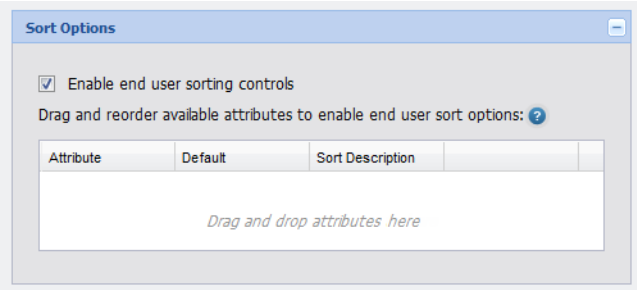
   - To sort by the selected attribute in ascending order, click the up arrow. This is the default.
   - To sort by the selected attribute in descending order, click the down arrow.



To remove the attribute, click the delete icon.

To replace the attribute with a different attribute, drag a different attribute from the **Available Attributes**, and then drop it on top of the currently selected attribute.

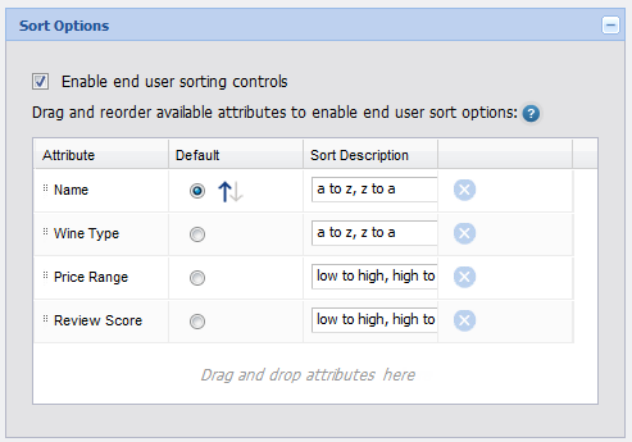
3. If the **Sorting** checkbox is checked, then the **Sort Options** section is updated to allow you to provide a list of attributes end users can use to sort the list.



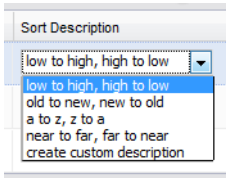
To add an attribute, drag the attribute from the **Attributes List** to the **Sort Options** table.

In the **Available Attributes** list, you can use the filter field to find the attribute you want to use.

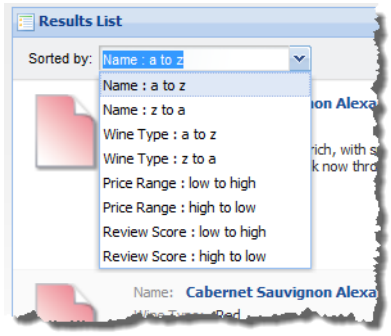
4. To determine the order of the attributes in the **Sorted by** drop-down list, drag each row up and down in the list.



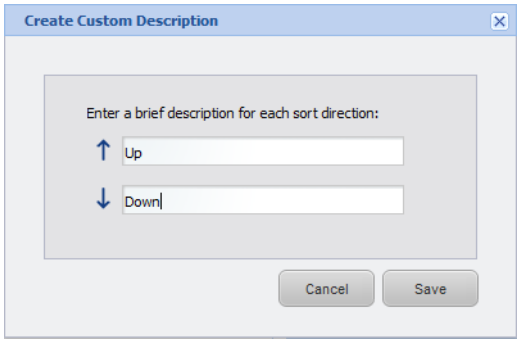
5. For the attribute you want to use for the default sort order:
   a) Click the **Default** radio button next to the attribute.
   b) To sort by that attribute in ascending order, click the up arrow.
   c) To sort by that attribute in descending order, click the down arrow.
6. From the **Sort Description** drop-down list, select the text that displays next to the attribute in the **Sorted by** list.



For each attribute, the **Sorted by** drop-down list includes one entry for sorting in ascending order, and one for sorting in descending order.



To create your own descriptions for ascending and descending order, from the **Sort Description** drop-down list, select **create custom description**. On the **Create Custom Description** dialog box:

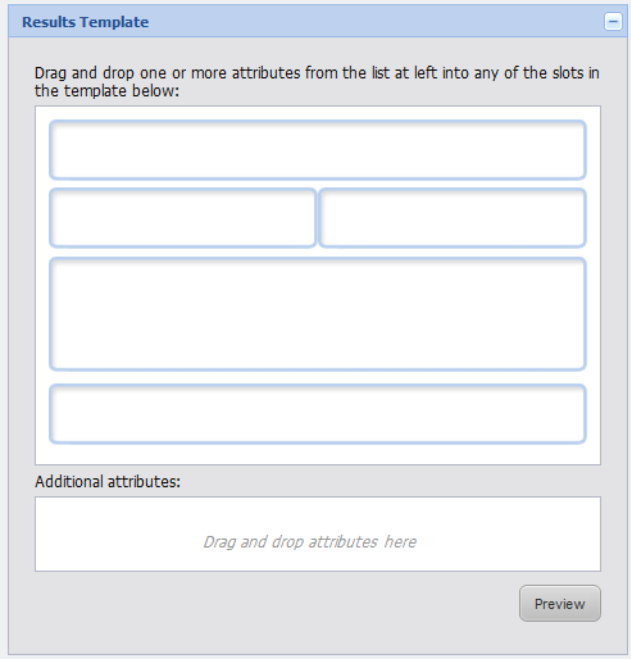


a) In the first field, marked with the up arrow, type the text to use to indicate ascending order.
b) In the second field, marked with the down arrow, type the text to use to indicate descending order.
c) Click **Save**.

7. To remove an attribute from the list, click the delete icon for that attribute.

About the Results Template for the Results List

For the **Results List** component, you use the **Results Template** section of the edit view to configure the attributes to include for each record.



Each block on the top area of the template represents a single attribute. You can also drag any number of attributes into the additional attributes area at the bottom of the template.

Each attribute can have an associated action, either to

- Link to the record detail
- Refine the data set
- Link to a resource such as a Web page or file. These resources may be specific to the value of a selected attribute.

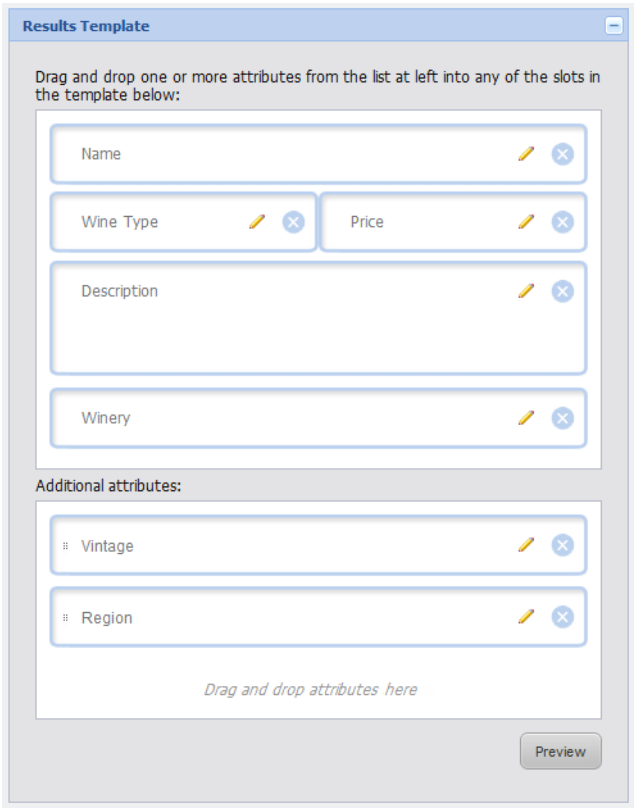
Selecting the attributes to display for each Results List record

For each block on the **Results Template** section of the **Results List** component edit view, you select an attribute for which to display the value.

To select the attributes to display:

1. To add an attribute to the template, drag the attribute from the **Available Attributes**.

   While the top area of the template is fixed, for the lower set of additional attributes, you can add any number of attributes. Each new attribute is added to the bottom of the list. You can then drag the attributes to set the display order.

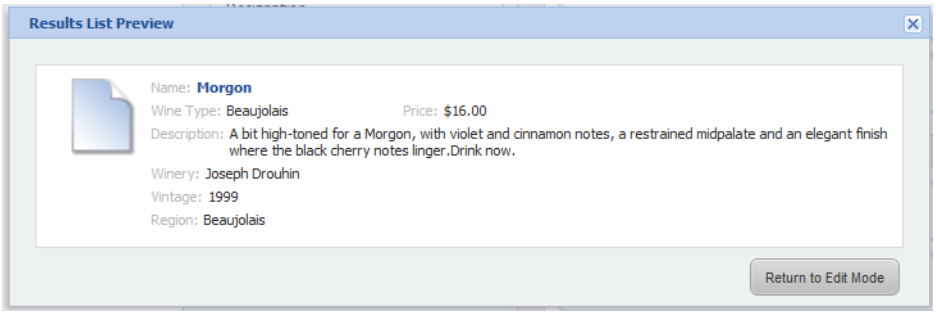


2. To replace an attribute in the top area of the template with a different attribute, drop the new attribute into the block.

   For the additional attributes, you cannot replace an attribute directly. All new attributes are added to the end of the list.

3. To clear a top block entirely, or remove a block from the additional attributes, click the delete icon for that block.
4. To preview how the record will look on the end user display, click the **Preview** button.

   The **Results List Preview** dialog is displayed.



   To close the preview, click **Return to Edit Mode**.

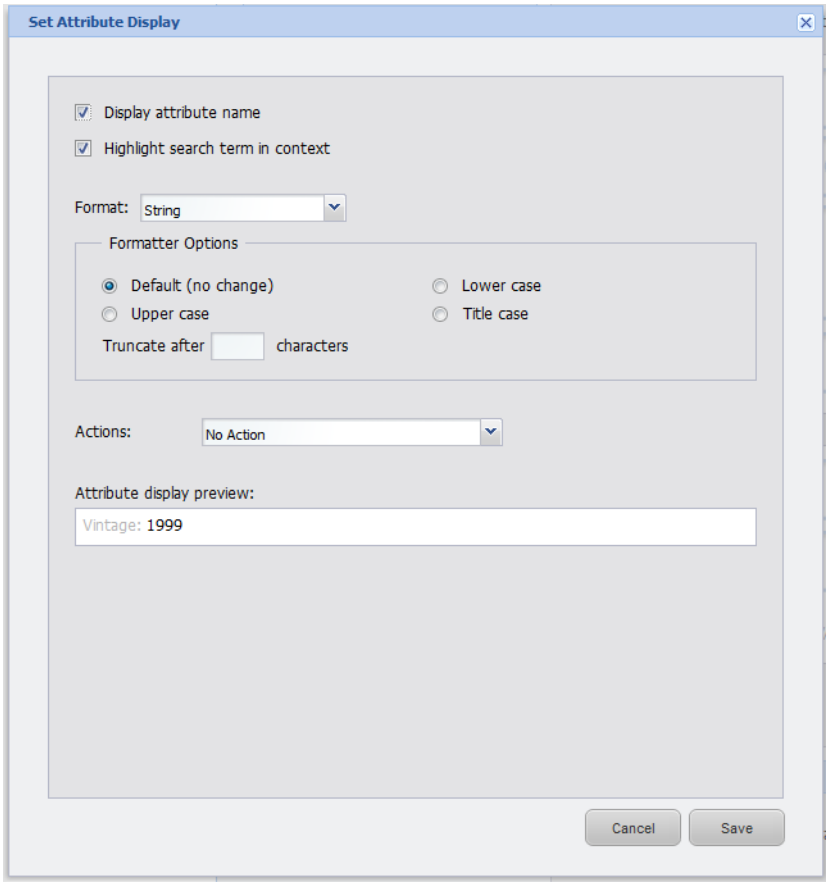
Configuring the attributes in a Results List record

For each attribute selected to display for a **Results List** component, power users configure the display and behavior.

From the **Results Template** section of the edit view, to configure an attribute:

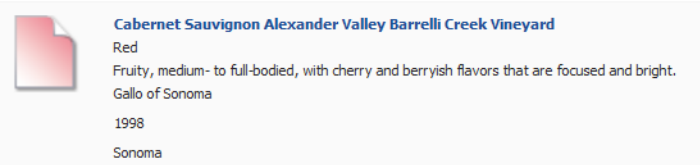
1. Click the edit icon for that block.

   The **Set Attribute Display** dialog box displays.



2. To display the attribute name as well as the value, check the **Display attribute name** checkbox. The box is checked by default.

   To only display the value, uncheck the box. Here is an example of a record where all of the attribute names are turned off.



3. When end users type search text into a **Search Box**, to highlight the search text in the attribute value, check the **Highlight search term in context** checkbox.

   The box is checked by default.

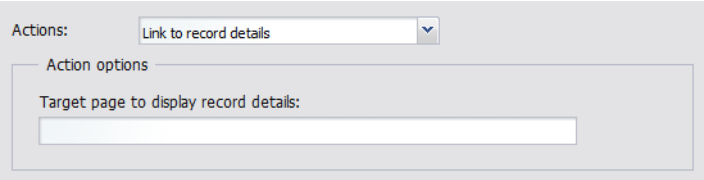
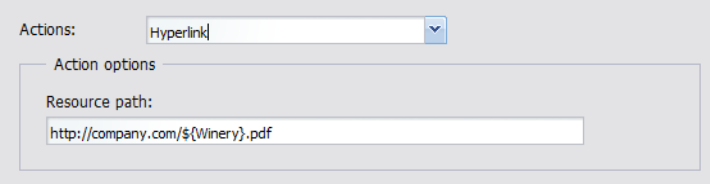
4. From the **Format** drop-down list, select the format to use to display the attribute value. The default format is based on the attribute's data type.

   The options are:

Integer	Indicates that the value is an integer. For an integer value, you can configure: • The character to use to separate the thousands • Whether to display a % sign after the value
Currency	Indicates that the value is a currency value. For a currency value, you can configure: • The character to use to separate the thousands • The character to use for the decimal • The number of decimal places to include • The currency symbol to use • Whether to display the currency symbol in front of the value (**prefix**) or after the value (**suffix**)
Decimal	Indicates that the value is a decimal number. For a decimal value, you can configure: • The character to use to separate the thousands • Whether to display a % sign after the value • The character to use for the decimal • The number of decimal places to display
String	Indicates that the value is a text string. For a string value, you can configure the capitalization. You can either: • Keep the string as it is • Convert the string to all upper case • Convert the string to all lower case • Convert the string to title case (first letter of each word is capitalized) You also can provide a number of characters after which to truncate the value. For example, for a long description, you may want to only display the first 200 characters.
Date	Indicates that the value is a date. For a date value, you can configure the format to use. The options are: • **American style (MM/DD/YYYY)**. For example, for October 20, 2010, the American style date would be 10/20/2010. • **European style (DD/MM/YYYY)**. For example, for October 20, 2010, the European style date would be 20/10/2010.

5. From the **Actions** drop-down list, select the option to indicate whether the attribute value is clickable, and if so, what happens when end users click the value. The options are:

No Action	If you select this option, then the attribute value is not a hyperlink.

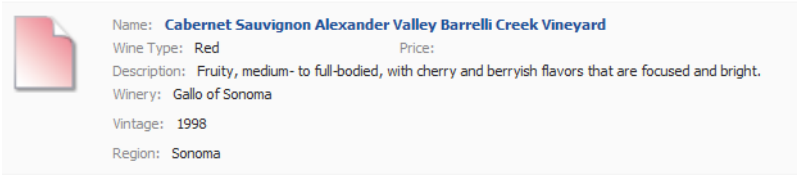
Link to record details	This option is only available for the top block of the template. If you select this option, then when end users click the attribute value, the **Record Details** component is populated with the details for that record. When you select the option, the **Target page to display record details** field is displayed.  In the field, type the name of the page to display when end users click the record hyperlink. The selected page must contain a **Record Details** component. The component must use the same data source as the **Results List** component. If you do not provide a page name, then the end user stays on the current page.
Enable drill down by refinement on this field	This option only displays if the attribute can be used for navigation. It is not available for the top block of the record. If you select this option, then when end users click the attribute value, the data set, including the results list, is refined to only show records with that value. The selected value is added to the **Breadcrumbs** component on that page.
Hyperlink	If you select this option, then the attribute value becomes a link, for example: • A link to another Web page • A link to a file such as a document or spreadsheet You can create different content to represent different values of an attribute. The URL or file name must then include the attribute value. When you select this option, the **Resource path** field is displayed.  In the field, type the path to the page or file. If you have created different content for different values of an attribute, then in the URL, use `${<attribute name>}` to represent the attribute value, where *<attribute name>* is the name of the attribute.

For example, if you have created PDF files with summary information about each winery, with the file name being the winery name, then the path would be something like `http://company.com/resources/${Winery}.pdf`.

6. To save the attribute configuration, click **Save**.

Configuring the images to display next to each Results List record

For a **Results List** component, power users can configure an image to display next to each record.



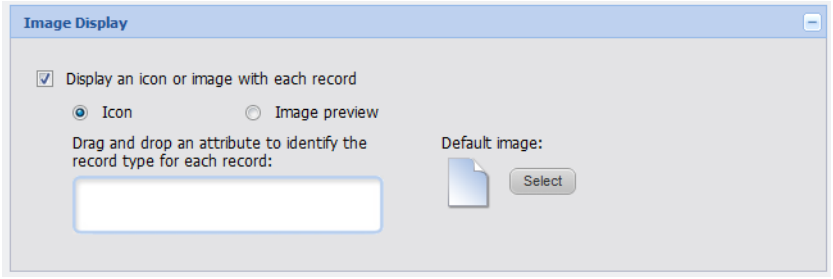
You can have the same image for each record, or can display different images to represent different values of a selected managed attribute.

You can select a standard image from the image gallery, or create your own images. If you create your own images:

- The image format must be either JPEG/JPG, Bitmap, GIF, or PNG.
- To have separate images for each value of a specific attribute:
  - The files must be located in the same directory.
  - There must be a file for each possible attribute value.
  - Each file must use the attribute value as the file name. For example, to have separate images for each type of wine, you might create a directory containing the images `Red.jpg`, `White.jpg`, and `Sparkling.jpg`.

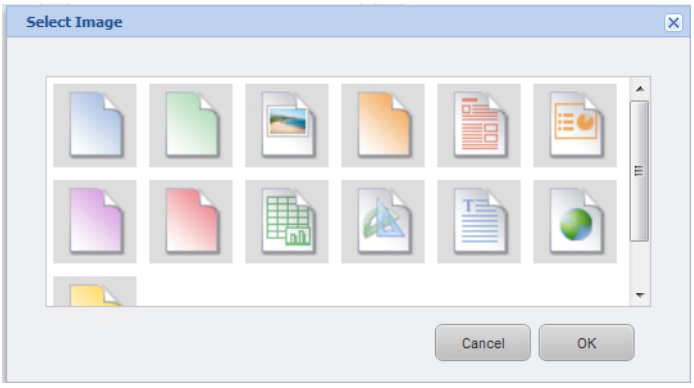
In the **Image Display** section of the edit view, to configure the images:

1. To display an image next to each record, check the **Display an icon or image with each record** checkbox. The box is checked by default.



   To not display an image, uncheck the box. If you uncheck the box, the rest of the image settings are hidden.

2. If you are displaying an image next to each record, then the **Default image** determines the default image to display. To select a different default image:
   a) Click the **Select** button.
   b) On the **Select Image** dialog, click the image you want to use as the default image.



c) Click **OK**.

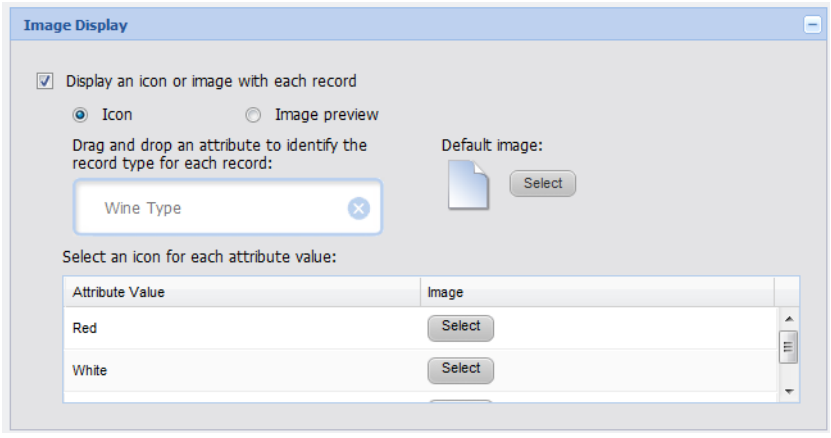
For images other than the default image, you can either display:

- An icon for each value of a selected managed attribute. The selected attribute cannot have more than 15 values.
- An image from a URL you provide

3. To configure an icon to display for each value of a selected managed attribute:
   a) Click the **Icon** radio button.
   b) From the **Available Attributes** list, drag the attribute you want to use to select the icons.

Note: To make it easier to drag the attribute, you can collapse the **Sort Options** and **Results Template** sections.

When you drop the attribute, the list of available attribute values is displayed.

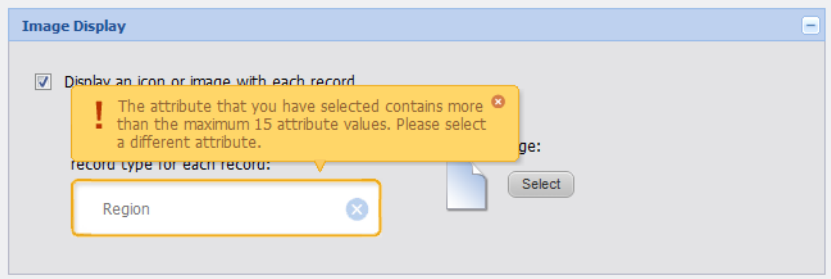


For a hierarchical attribute, only the top level of values is displayed. On the end user view, records use the image for the top level value.

For example, for a WineType attribute, the top level values might be Red, White, and Sparkling.

If a record has a WineType value of Merlot (Red --> Merlot), that record would use the image selected for the Red value.

If the attribute is not a managed attribute, or has more than 15 values, an error message is displayed.

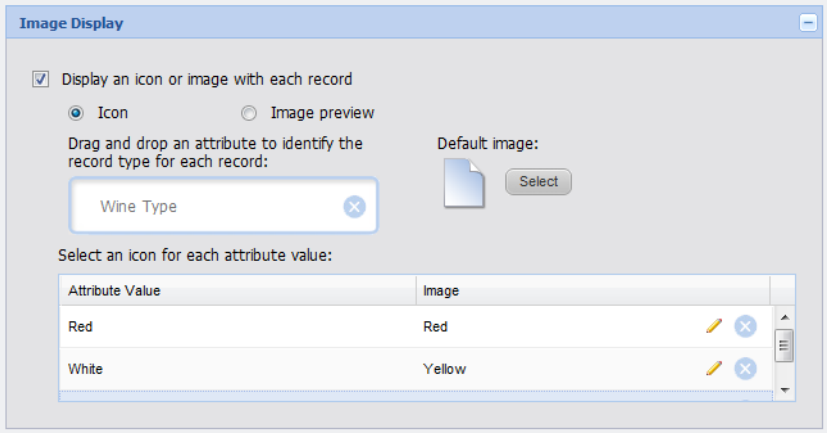


When you close the error, the attribute is removed.

c) For each attribute value, to select the image to use, click the **Select** button.

d) On the **Select Image** dialog box, click the image you want to use, then click **OK**.

The list is updated with the name of the selected image.

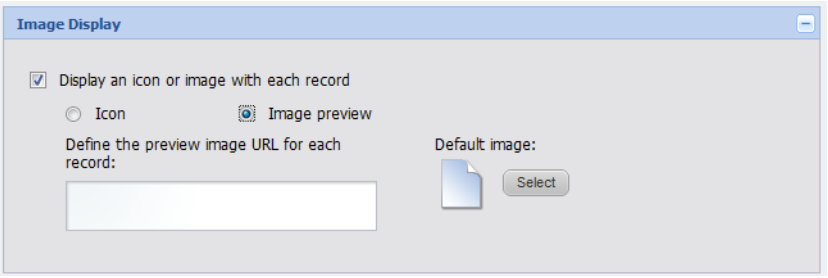


e) To clear a selected image, and display the default image for that attribute value, click the delete icon.

f) To select a different image for an attribute value, click the edit icon.

4. To display your own images:

a) Click the **Image preview** radio button.



b) In the field, type the URL to the image file.

If you have created a set of image files to represent different attribute values, then for the file name, specify `${<attribute name>}.<file extension>` , where:

- *<attribute name>* is the name of the attribute.
- *<file extension>* is the file type for the image files.

For example, if you have created a set of jpg image files representing the different values of the WineType attribute, then the image URL would be something like `http://company.com/images/${WineType}.jpg`.

Saving changes to the Results List configuration

In order for the **Results List** configuration to take effect, you must save the configuration changes.

From the edit view of the **Results List** component:

1. To save the changes to the configuration, click **Save Preferences**.

   A message displays indicating that the changes were saved successfully.

2. To exit the edit view, click **Return to Full Page**.

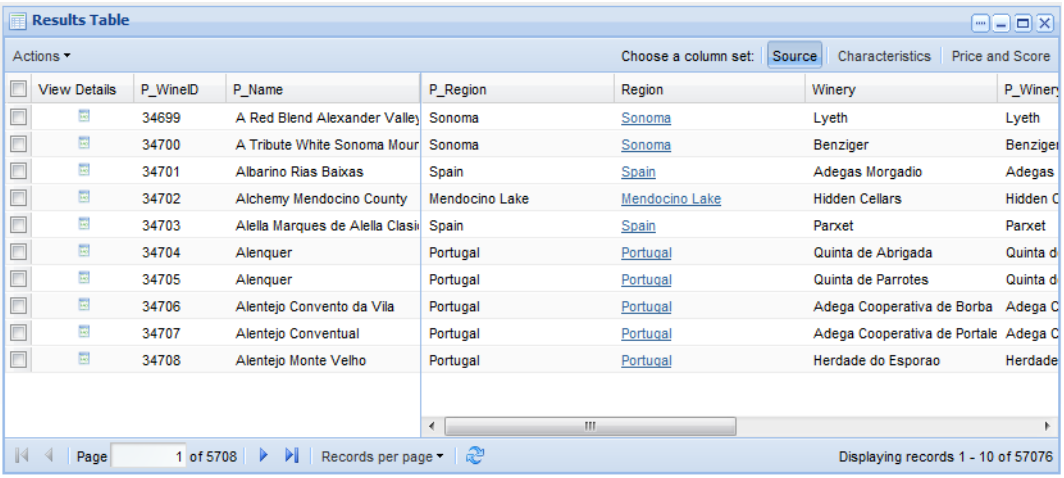
Results Table

About the Results Table component

The **Results Table** component displays a set of data in a table format.

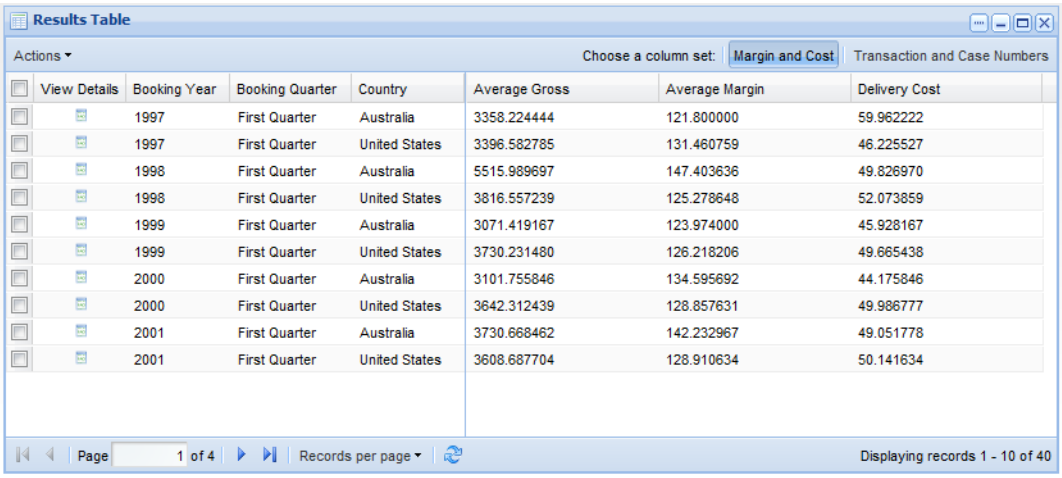
The data displayed in the **Results Table** component is either:

- A list of records from the data set. Each row represents a single record. The columns contain the attribute values for the record.



- A set of generated metrics calculated from an LQL query. For example, the table could display the average prices, profit margin, and delivery time for each region and year.

  Each row represents a unique combination of the attributes used to group the metrics values. The columns contain the grouping attribute values and the corresponding metric values.



Using Results Table

End users can page through and sort the results. They also may be able to use the results to refine the data or display related content.

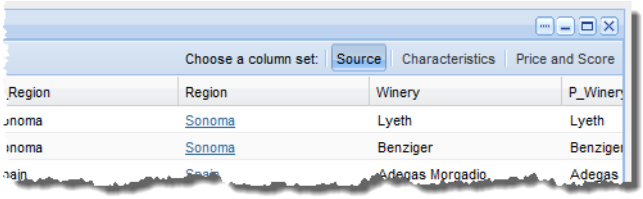
Selecting the set of columns to display

The **Results Table** results may be grouped into multiple column sets, with only one set of columns displayed at a time.

For a list of records, each column set is an attribute group. For a list of metrics, power users define the column sets.

The **Results Table** also may be configured with locked columns that are always displayed to the left of the table.

If the **Results Table** contains multiple column sets, then the column set names are displayed at the top of the table. The currently selected set is highlighted.



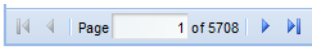
To display the columns in a column set, click the column set name.

The columns for the selected column set are displayed to the right of any locked columns for the results.

If there is only one column set, then the column set name is not displayed, and the columns are always displayed.

Navigating through the list of results

The **Results Table** component is configured with a default number of records to display on a page. If the power user has configured the component to allow pagination, then pagination tools are displayed at the bottom of the component.



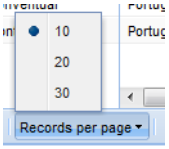
End users can use these tools to:

- Navigate to the next or previous page
- Navigate to the first or last page
- Jump to a specific page. To go to a specific page, type the page number in the field, then press **Enter**.

Selecting the number of results to display per page

The **Results Table** component can include a **Records per page** button, which is used to select the number of records to display per page.

When users click the **Records per page** button, the list of available options is displayed. The currently selected option is marked.



To change the number of records per page, select the number.

Sorting the Results Table results

The **Results Table** may be configured to allow end users to sort the list of results.

To sort the list, click the column heading for the column you want to sort by. The results are sorted in ascending order.

To sort the results in descending order, click the column heading again.

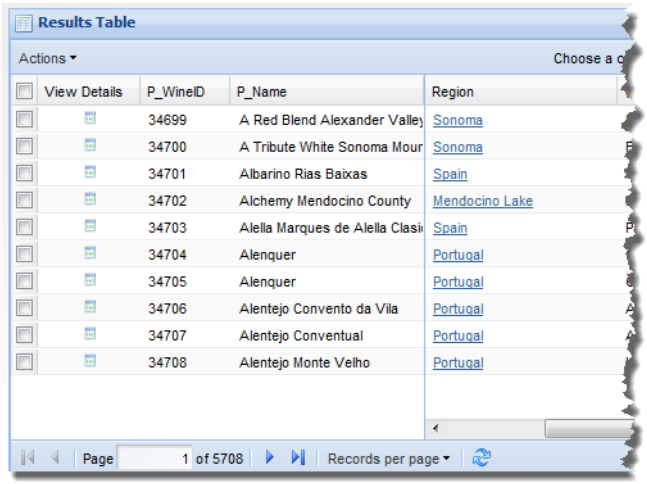
Types of links available from a Results Table component

A **Results Table** component can include links to record details or other types of information.

The table can include hyperlinks to:

- Refine the current data. When you use an attribute value to refine the data, it is added to the **Breadcrumbs** component.
- Populate a **Record Details** component with the values for a selected row
- Display related information

The links can be either from a displayed value or in a separate action column. Action columns can contain either an icon or a text link.



When you hover over a link, a tooltip may display to indicate the available action.

Exporting results from a Results Table component

The **Results Table Actions** menu can include an option to export results from the table to a CSV file.

To export results from a **Results Table**:

1. Check the check box next to each row you want to export.

   To export all of the rows on the current page, check the check box in the column heading.

   If you do not check any of the rows, then the entire list is exported.
2. From the **Actions** menu, select **Export**.
3. You are prompted to view or save the file.

Printing the current page of Results Table results

The **Results Table** component includes an option to print the current page of the results. If other actions are available, then the **Print** option is in the **Actions** menu. If no other actions are available, then a **Print** button is displayed in place of the **Actions** menu.

To print a page of the **Results Table** results:

1. Navigate to the page you want to print.
2. From the **Actions** menu, select **Print**.

   If there is no **Actions** menu, then click the **Print** button.

   The results list is displayed in a separate browser window. Above the list are the current refinement selections.

   The print dialog box also is displayed.
3. Select the printing options, and then complete the printing process.

Configuring a Results Table component

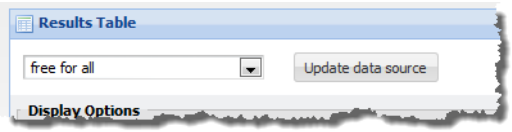
When the **Results Table** is first added, it does not display any data. Power users configure the data to include and the available options for working with the results.

Selecting the data source to use for a Results Table

The **Results Table** component requires a backing data source. By default, the component uses the default data source.

From the component edit view, to bind a different data source to the component:

1. From the data source drop-down list, select the data source.



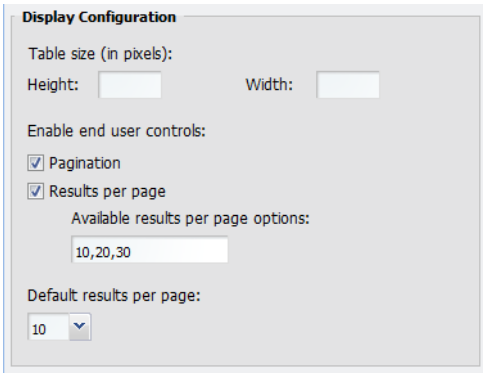
2. Click **Update data source**.

   A message displays indicating that the component now uses the new data source.

Configuring Results Table size and pagination options

For a **Results Table** component, power users can set the size of the component, and whether to allow end users to page through the results and set the number of results per page.

In the **Display Configuration** section of the edit view, to configure the component size and pagination options:



1. To set the size of the component:
   a) In the **Height** field, type the height in pixels of the component.
   b) In the **Width** field, type the width in pixels of the component.
2. To allow end users to page through the results, check the **Pagination** checkbox. The box is checked by default.

   If the box is not checked, then end users cannot navigate through the list. The component only displays a number of records equal to the value of the **Default results per page** field. To see other records, end users must further refine the data.
3. To allow end users to set the number of results per page, check the **Results per Page** checkbox.

4. If the **Results per Page** checkbox is checked, then:
   a) In the **Available results per page options** field, type a comma-separated list of the available options for the number of results per page.

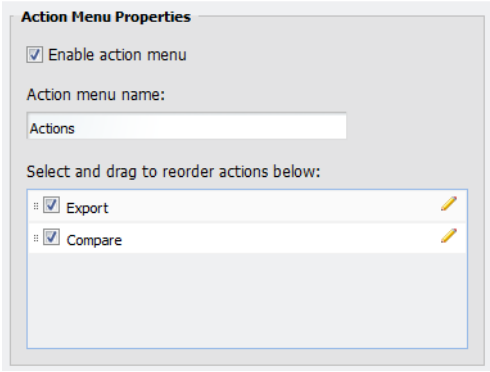
      These values are used both for the **Default results per page** drop-down list below this field, and the **Results per page** drop-down list used by end users.

   b) From the **Default results per page** drop-down list, select the number of results to display per page by default.
5. If the **Results per Page** checkbox is not checked, then in the **Defaults results per page** field, type the number of results to display per page.
6. To save the configuration changes, click **Save Preferences**.
7. To exit the edit view, click **Return to Full Page**.

Selecting the available actions for a Results Table component

By default, the **Results Table** component includes an **Actions** menu with a **Print** option. You also can configure the **Actions** menu to allow end users to export the results to a spreadsheet, and use a **Compare** component to compare selected records.

On the **Results Table** edit view, in the **Action Menu Properties** section, to configure the **Actions** menu:



1. To include the **Actions** menu on the component, check the **Enable action menu** checkbox.

   If the box is not checked, then on the edit view, the rest of the **Action Menu Properties** fields are disabled. On the end user view, the **Print** button is displayed instead of the **Actions** menu.

2. If the **Actions** menu is enabled, then in the **Action menu name** field, type the label to use for the **Actions** menu.

   The default is **Actions**.

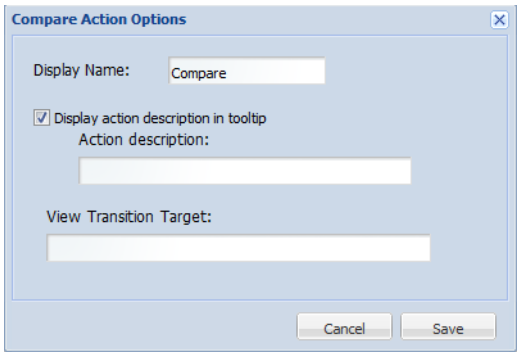
3. In the list of available actions, check the checkbox next to each action to include in the menu.

   To set the order of the actions, drag each action to the correct location in the list.

   Note that the **Print** option is always available, and is always the first option. You can only control the display of the **Compare** and **Export** options.

4. To configure each action:
   a) Click the edit icon.

      The action configuration dialog is displayed.



b) In the **Display Name** field, type the label to use for this action in the **Actions** menu.
c) To display a tooltip for the action, check the **Display action description in tooltip** checkbox.

   In the **Action description** field, type the text to display in the tooltip.

d) For the **Compare** action only, in the **View Transition Target** field, type the name of the page to send users to when they select the **Compare** action.

   The page must contain a **Compare** component that uses the same data source.

   If you do not provide a page name, the end user stays on the current page.

e) To save the changes to the action, click **Save**.

5. To save the configuration changes, click **Save Preferences**.
6. To exit the edit view, click **Return to Full Page**.

Using LQL to generate the Results Table results

By default, the **Results Table** component displays a list of records. You can also configure a **Results Table** component to display a set of metrics generated from an LQL query.

On the edit view of a **Results Table** component, to use an LQL query to generate the **Results Table** results:

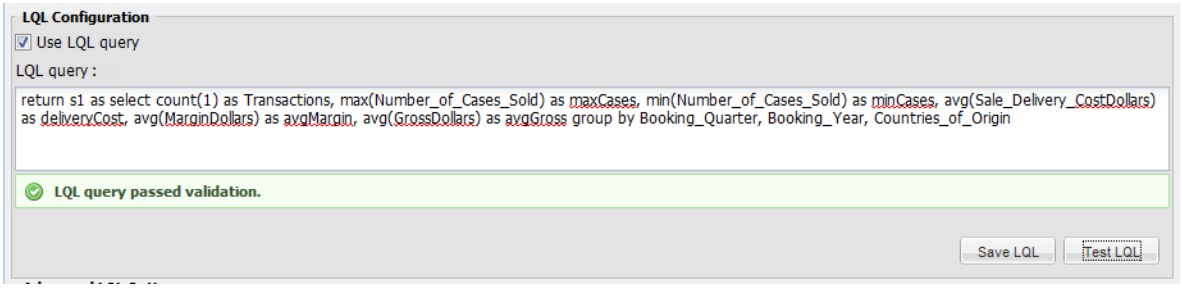
1. Under **LQL Configuration**, check the **Use LQL query** checkbox.

   The **LQL query** text area and **Advanced LQL Options** section are displayed.

2. In the **LQL query** field, type the LQL query.
3. After entering the query, to validate it, click **Test LQL**.

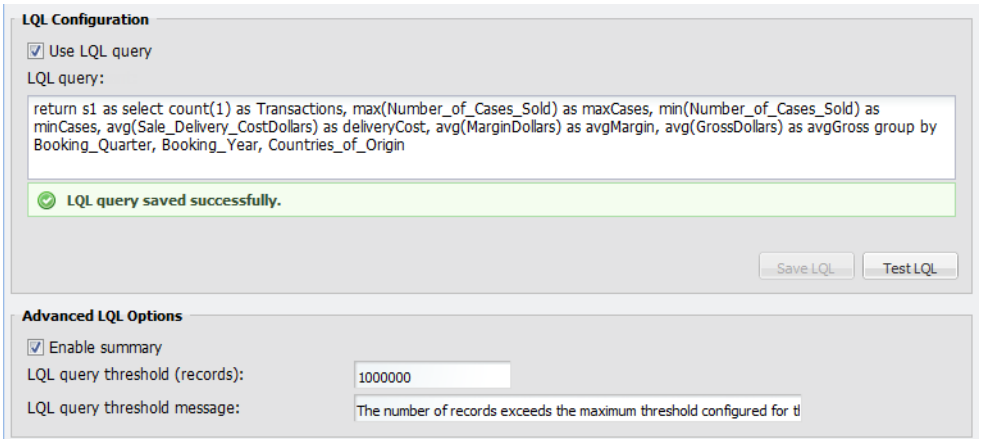
   If there are no errors, then:

   - The **Save LQL** button is enabled.



- In the **Column Group Configuration** section, the list of attribute groups is replaced with the list of metrics and group-by attributes, contained in a single **Default Group** column set.

4. To save the LQL query, click **Save LQL**.
5. Under **Advanced LQL Options**:

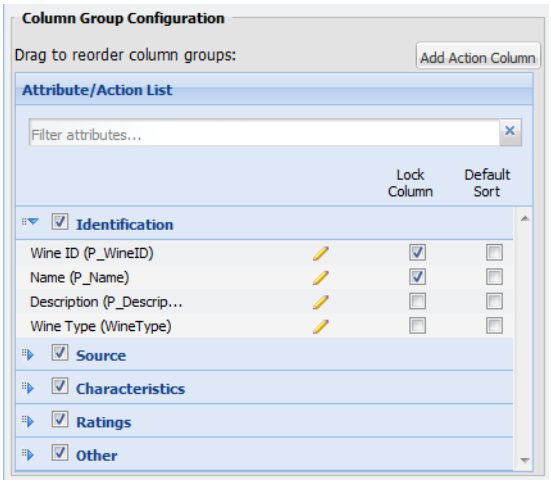


Enable summary	To display a summary row at the bottom of the table, check the checkbox. For each column, you then configure whether to display a summary value in the row.
LQL query threshold (records)	In the field, type the maximum number of base records to be processed by the LQL query. If the number of records processed is greater than this number, then the end user is prompted to refine the data.
LQL query threshold message	In the field, type the message to display to end users when the number of records processed is greater than the maximum.

6. To save the configuration changes, click **Save Preferences**.
7. To exit the edit view, click **Return to Full Page**.

Configuring the column sets to display for record-based results

If you are not using LQL to populate the table, then you use the **Attribute/Action List** to select the attribute groups to include on the component. Each displayed attribute group becomes a column set.



To configure the attribute groups to display:

1. To include an attribute group, check the checkbox for the group.
2. For the displayed attribute groups, for each attribute:
   - To display the attribute value as a locked column, check the **Lock Column** checkbox.

     The attribute is copied to the **Locked Columns** list. Locked columns are not displayed in their attribute group. They are only displayed in the locked column section to the left of the table.

     Note: You should not have more than 2-3 locked columns. Having too many locked columns can make the table less readable for end users.

   - To use the attribute value in the default sort order, check the **Default sort** checkbox.

     The attribute is copied to the **Sorting Controls** list.

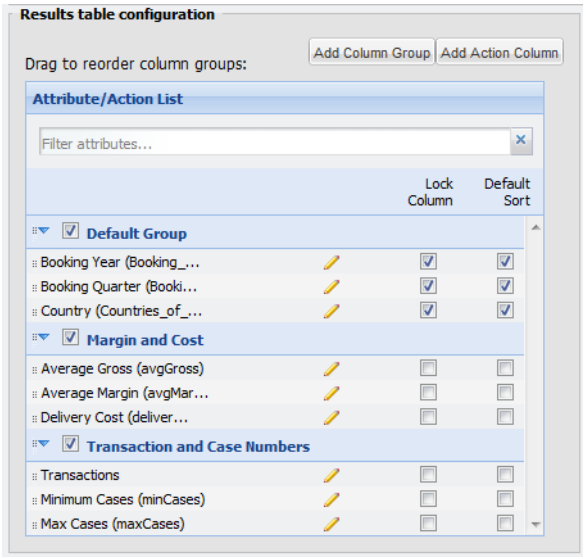
   To search for a specific attribute, in the **Filter attributes** field, begin typing the attribute name. As you type, the list is filtered to only display the matching attributes.
3. To set the display order of the attribute groups, drag the group to the correct location in the list.

   You can expand and collapse the groups to make them easier to work with. To expand or collapse a group, click the group name.
4. To save the changes, click **Save Preferences**.
5. To exit the edit view, click **Return to Full Page**.

Configuring the column sets to display for LQL-based results

If you are using an LQL query to populate the table, then when you first load the LQL query, the **Attribute/Action List** is populated by a single column set called **Default Group**. In the **Default Group** are the metrics and grouping attributes from the LQL query.

For results generated from an LQL query, you can create and rename column sets. You can then move the metrics into your column sets.



To configure the column sets:

1. To add a new column set, click **Add Column Group**.
   An empty column set is added to the list.
2. To rename a column set:
   a) Double-click the set name.

      The name displays in an editable field.

   b) Type the new name in the field, then press **Enter**.
3. To include a column set in the display, check its checkbox. To remove a column set, uncheck the checkbox.
4. To change the display order of a column set, drag the set to the new location in the list.
5. To remove an empty column set, click its delete icon.

   You cannot remove a column set that contains columns.
6. To move a metric into a different column set, drag it into the column set.
7. To change the display order of the columns within a column set, drag each column to the new location in the list.
8. To display the metric value as a locked column, check its **Lock Column** checkbox.

   The column is copied to the **Locked Columns** list. Locked columns are not displayed in their column set.

   They are only displayed in the locked column section to the left of the table.

   Note: You should not have more than 2-3 locked columns. Having too many locked columns can make the table less readable for end users.
9. To use the metric in the default sort order for the results, check the **Default sort** checkbox.

   The column is copied to the **Sorting Controls** list.

10. To save the changes, click **Save Preferences**.
11. To exit the edit view, click **Return to Full Page**.

Formatting and selecting actions for Results Table column values

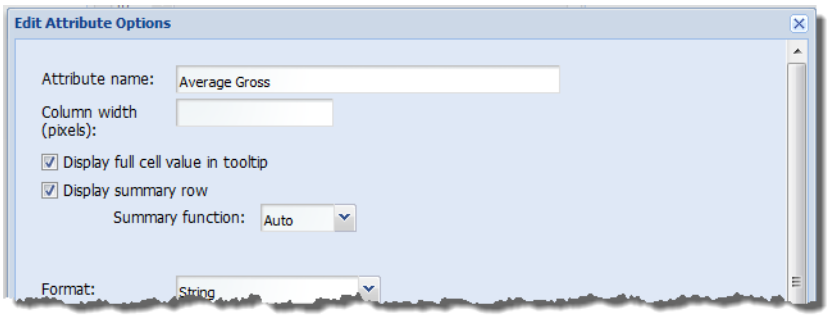
For each column in a **Results Table**, power users can configure the format used to display the value, and whether to allow users to click the value in order to refine data or display related information. For an LQL-based table, power users also configure the column heading and summary options.

From the edit view of a **Results Table** component, to configure a column:

1. Click the edit icon for the column.

   The **Edit Attribute Options** dialog is displayed.

2. For columns in an LQL query, in the **Attribute name** field, type the label to use for the **Results Table** column.



   For attribute values in a records-based table, the display name is used, and you cannot change the column heading.

3. In the **Column width (pixels)** field, type the default width of the column. End users can then adjust the width.
4. When end users hover the mouse over a value, to display the full text of the value in a tooltip, check the **Display full cell value in tooltip** checkbox.

   This is particularly useful for longer text strings such as descriptions.

5. For LQL-based tables, if you have enabled the summary row, then for a generated metric, to display the summary value in the summary row:
   a) Check the **Display summary row** checkbox.
   b) From the **Summary function** drop-down list, select the function used to generate the value in the summary row.

   Note that you cannot display a summary value for a group-by attribute.

6. From the **Format** drop-down list, select the format to use to display the column value.

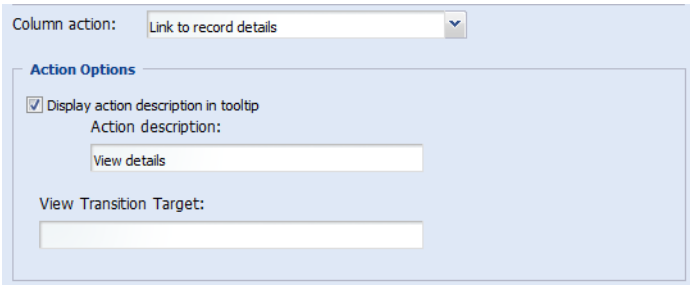
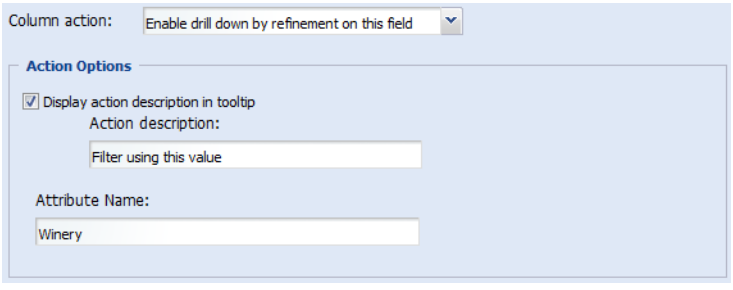
For attributes, the default value is based on the attribute data type. Make sure to select a format that makes sense for the data being displayed.

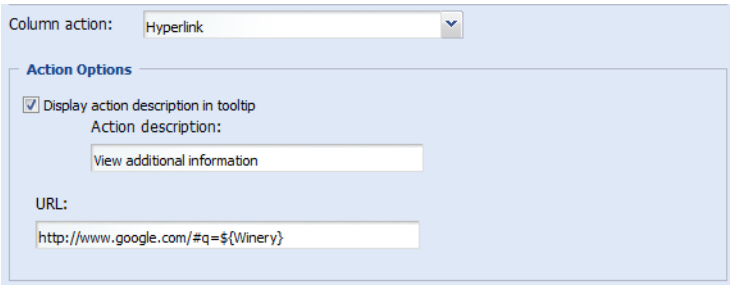
When you select a format, the **Formatter options** section is updated to display the fields to use to configure that format type. The available formats are:

Currency	Indicates that the value is a currency value. For a currency value, you can configure: • The character to use to separate the thousands • The character to use for the decimal point • The number of decimal places to include • The currency symbol to use • Whether to display the currency symbol in front of the value (**prefix**) or after the value (**suffix**)
String	Indicates that the value is a text string. For a string value, you can configure the capitalization. You can either: • Keep the string as it is • Convert the string to all upper case • Convert the string to all lower case • Convert the string to title case (first letter of each word is capitalized) You also can provide a number of characters after which to truncate the value. For example, for a long description, you may want to only display the first 200 characters.
Integer	Indicates that the value is an integer. For an integer value, you can configure: • The character to use to separate the thousands • Whether to display a % sign after the value
Decimal	Indicates that the value is a decimal value. For a decimal value, you can configure: • The character to use to separate the thousands • Whether to display a % sign after the value • The character to use for the decimal • The number of decimal places to display
Date	Indicates that the value is a date. For a date value, you can configure the format to use. The options are: • **American style (MM/DD/YYYY)**. For example, for October 20, 2010, the American style date would be 10/20/2010. • **European style (DD/MM/YYYY)**. For example, for October 20, 2010, the European style date would be 20/10/2010.

7. From the **Column action** drop-down list, select the action to perform when users click the column value.

   When you select an option, the **Action options** section is updated to display the fields to use to configure that action type. The available actions are:

None	Indicates that the value is not a hyperlink. This is the default value.
Link to record details	Indicates that when end users click the icon or link, a **Record Details** component is populated with the values for that row. For records-based tables, the attributes are displayed in their attribute groups. For LQL-based tables: • The group-by attributes are displayed in their attribute groups. • The generated metrics are displayed in the **Other** group.  For the **Link to record details** action, you can configure: • Whether to display a tooltip when the end user hovers the mouse over the value • The text to display in the tooltip • The page on which the **Record Details** component is located. If you do not specify a page, the end user stays on the current page.
Enable drill down by refinement on this field	Indicates that when end users click the value, the value is used to refine the data. For an LQL-based table, this option can only be used for the grouping attributes. It cannot be used for the generated metrics.  For the **Enable drill down by refinement on this field** action, you can configure:

	• Whether to display a tooltip when the end user hovers the mouse over the value • The text to display in the tooltip You also must provide the name of the attribute. Make sure the attribute specified here is the attribute displayed in the column.
Hyperlink	Indicates that the value is a hyperlink to another page or file.  For the **Hyperlink** action, you can configure: • Whether to display a tooltip when the end user hovers the mouse over the value • The text to display in the tooltip • The URL for the hyperlink If you have created different resources for different values of an attribute, then in the URL, use `${<attribute name>}` to represent the attribute value, where *<attribute name>* is the name of the attribute. For example, if you have created PDF files with summary information about each winery, with the file name being the winery name, then the path would be something like `http://company.com/resources/${Winery}.pdf`.

Note that if you have also configured the column to display the full value in a tooltip, when end users hover over the value, both tooltips are displayed.

8. To save the column configuration, click **Save**.

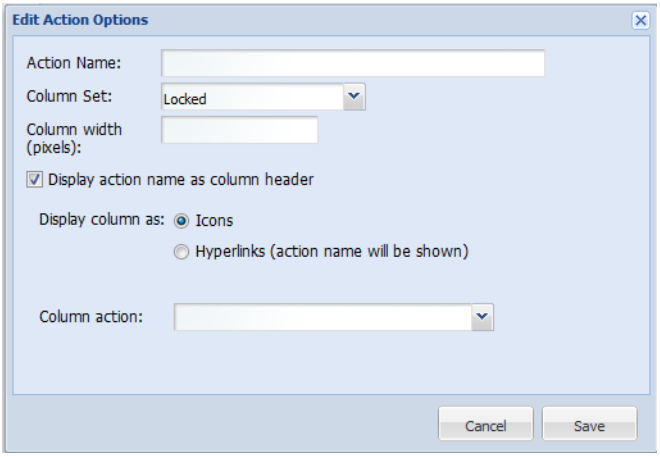
Adding action columns to the Results Table

In addition to the attribute values and metrics, a **Results Table** component can contain action columns. An action column contains a link to refine the data, display record details, or display other information.

From the edit view of a **Results Table** component, to add an action column:

1. Click the **Add Action Column** button.

   The **New action column** dialog is displayed.



2. In the **Action name** field, type the name of the action.

   The action name may be used as both the column heading and the text of the action hyperlink.

3. From the **Column group** drop-down list, select the column set to add the action to.

   By default, the action column is added to the **Locked** column set, to ensure that the action is always available to end users.

4. In the **Column width** field, type the width in pixels of the action column.
5. To display the action name in the column heading, check the **Display action name as column header** checkbox.

   If the box is not checked, then the column heading is empty.

   By default, the box is not checked.

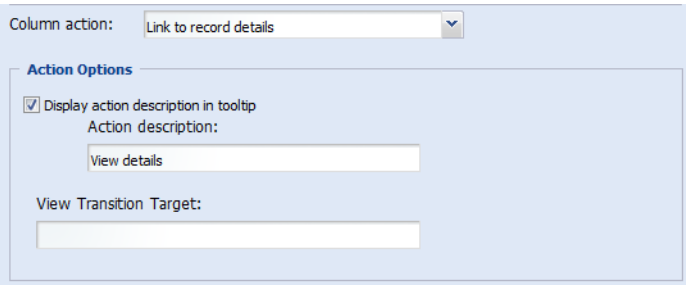
6. Under **Display column as**, click the radio button to indicate how to display the action.

   - To have end users click an icon, click the **Icons** radio button.
   - To have end users click a text link, click the **Hyperlinks** radio button.

7. Under **Column actions**, from the **Action** drop-down list, select the action.

   When you select an action, the **Column actions** section is updated to display the fields to use to configure that action type. The available actions are:

Link to record details	Indicates that when end users click the icon or link, a **Record Details** component is populated with the values for that row. For records-based tables, the **Record Details** component contains the standard list of attribute groups. For LQL-based tables, the **Record Details** component contains all of the column sets from the table.

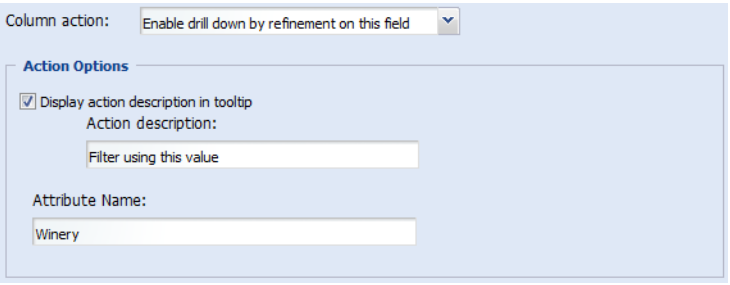


For the **Link to record details** action, you can configure:

- Whether to display a tooltip when the end user hovers the mouse over the column
- The text to display in the tooltip
- The page on which the **Record Details** component is located. If you do not specify a page, the end user stays on the current page.

Enable drill down by refinement on this field

Indicates that when end users click the icon or link, the value for the selected attribute is used to refine the data.



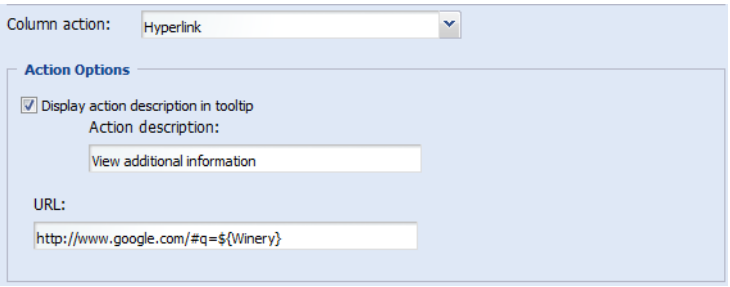
For the **Enable drill down by refinement on this field** action, you can configure:

- Whether to display a tooltip when the end user hovers the mouse over the value
- The text to display in the tooltip

You also must provide the name of the attribute.

Hyperlink

Indicates that the action is a hyperlink to another page or file.



For the **Hyperlink** action, you can configure:

- Whether to display a tooltip when the end user hovers the mouse over the value

	• The text to display in the tooltip • The target URL for the hyperlink If you have created different resources for different values of an attribute, then in the URL, use `${<attribute name>}` to represent the attribute value, where *<attribute name>* is the name of the attribute. For example, if you have created PDF files with summary information about each winery, with the file name being the winery name, then the path would be something like `http://company.com/resources/${Winery}.pdf`.

8. To save the new action, click **Save**.

   The action column is added to the selected column set. It is highlighted to indicate that it is an action and not a displayed value. If you have added the action column to the locked columns, then:

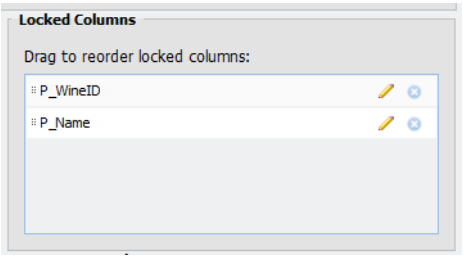
   - For a records-based table, the column is added to the **Other** attribute group.
   - For an LQL-based table, the column is added to the **Default Group**.

9. To remove the action from the table, in the **Attribute/Action List**, click the delete icon for the action.
10. To save the changes, click **Save Preferences**.
11. To exit the edit view, click **Return to Full Page**.

Configuring the locked columns for a Results Table component

Locked columns display to the left of the table, and are always visible. On the **Results Table** edit view, in the column set list, when you check the **Lock Column** checkbox for a column, it is copied to the **Locked Columns** list.

Note: You should not have more than 2-3 locked columns. Having too many locked columns can make the table less readable for end users.



For each locked column, the **Locked Columns** list displays an edit icon to allow you to format the value. This is the same edit function as in the column sets list.

In the **Locked Columns** list, to configure the locked columns:

1. To configure the display order of the locked columns, drag each column to the new location in the list.

2. To remove a column from the locked columns list, either:
   - In the column groups list, uncheck the **Lock Column** checkbox.
   - In the **Locked Column Settings** list, click the delete icon for the column.

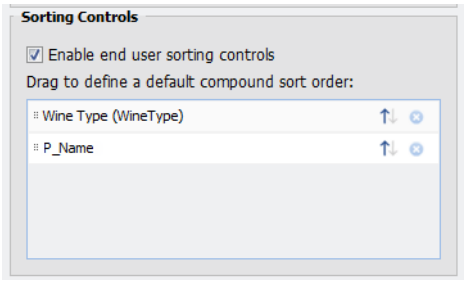
   When you remove a column from the locked columns, it is then displayed in its original column set.
3. To save the changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

Configuring the sorting options for a Results Table component

For the **Results Table**, power users configure a default sort order for the results. The default sort order can include more than one column.

Power users also can allow end users to sort the list by any of the columns.

In the column sets list, to add a column to the default sort order, check the **Default sort** checkbox. You cannot use action columns to sort the results. When you add a column to the default sort order, the column is copied to the **Sorting Controls** list.



From the **Sorting Controls** section of the **Results Table** edit view, to configure the sorting options:

1. To allow end users to sort the results, check the **Enable end-user sorting controls** checkbox.
2. In the list of columns:
   a) To change the order in which to use the columns for sorting, drag each column to the correct location in the list.

      For example, you may want to sort first by year, then by region, or vice versa.

   b) For each column, to sort in ascending order, click the up arrow icon. To sort in descending order, click the down arrow icon.
   c) To remove a column from the default sort order, click the delete icon for that column.

      You can also uncheck the **Default** sort checkbox in the **Attribute/Action List**.
3. To save the changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

Chapter 20

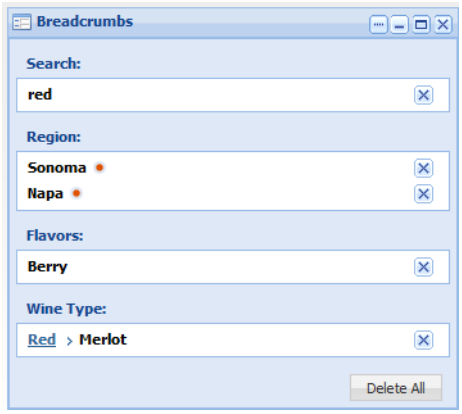
Filtering Components

Filtering components allow you to search, navigate, and filter your data.

Breadcrumbs

About the Breadcrumbs component

The **Breadcrumbs** component summarizes in a vertical stack the values the end user has selected to filter the data.



The component includes entries for:

- Guided Navigation selections
- Keyword searches
- Range filters

It also includes attribute value refinements for when end users:

- Click an alert message on an **Alerts** component
- Click an attribute value on a **Results List** or **Results Table** component
- Click a metric value on a **Chart** component
- Click an attribute value on a **Tag Cloud** component

The **Breadcrumbs** component only displays refinements that the end user selects. It does not include any record filters applied to the data source itself. For example, if the data source is configured so that end users can only ever see data for red wines, the red wine type is not displayed as a breadcrumb.

The **Breadcrumbs** component requires a backing data source. In order for the component to be useful, there also must be at least one component that is bound to the same data source and that allows end users to select refinements.

Using Breadcrumbs

End users use the **Breadcrumbs** component to view and edit their refinement selections.

How the refinements are displayed

For attribute value selections, the breadcrumb displays the attribute name as a heading.

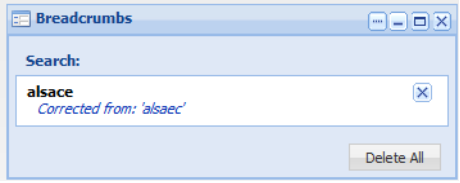
For negative refinements (refining data to show records that do NOT have the selected value), there is a red dot next to the attribute value.

If multiple values for an attribute are selected, they are displayed in sequential order. Depending on the number of records selected, end users can expand and collapse the list.

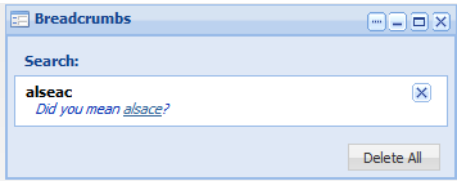
For a hierarchical attribute value, the breadcrumb may include all of the ancestor values.

For search term values, the heading indicates that the value is a search term.

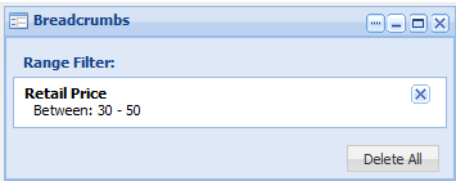
If the search function made an automatic spelling correction, it is included in the breadcrumb:



The breadcrumb may also include a "did you mean" option to suggest a search term that closely matches the specified term:



For range filters, the breadcrumb contains the attribute name and the selected range of values. The heading indicates that it is a range filter.



Removing refinements

From the **Breadcrumbs** component, to remove a single refinement, click the delete button for that refinement. For attributes that have multiple values selected, end users can either remove a single value or all of the values.

For hierarchical attributes, to filter by an ancestor value, click the ancestor value.

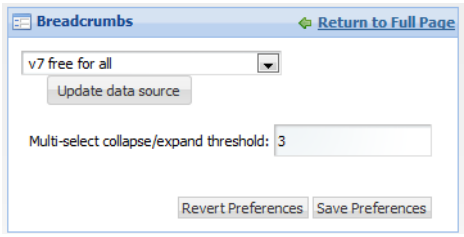
To remove all of the refinements, click the **Delete All** button.

Configuring a Breadcrumbs component

For a **Breadcrumbs** component, power users can configure the data source and whether to expand a list of multi-select attribute values.

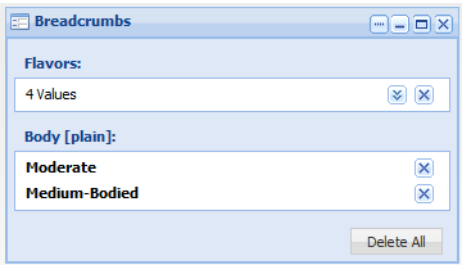
To configure a **Breadcrumbs** component:

1. On the **Breadcrumbs** component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.

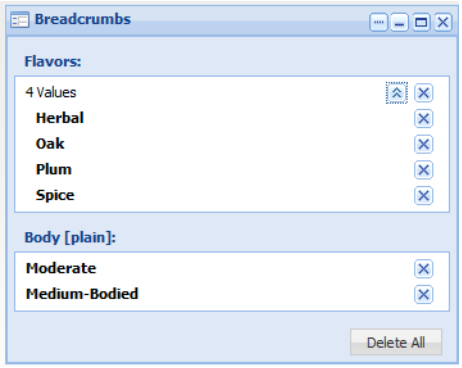


2. In the **Multi-select collapse/expand threshold** field, set the number of attribute values after which the list can be collapsed.

   When end users select multiple values for an attribute, if they select more than this number, then on the **Breadcrumbs** component, the list of selected values is initially collapsed.



End users can then use the expand/collapse button to display or hide the full list.

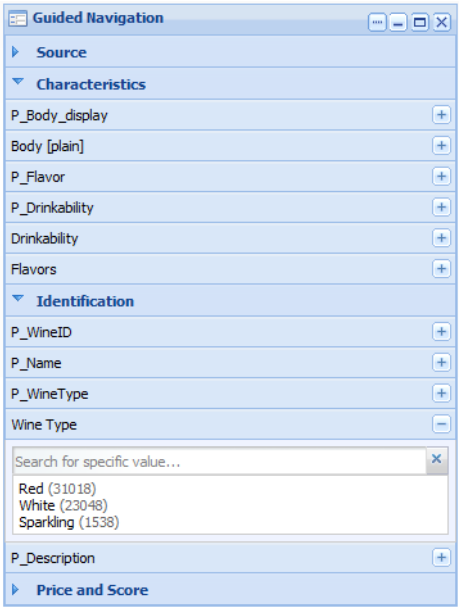


3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

Guided Navigation

About the Guided Navigation component

The **Guided Navigation** component provides Endeca Guided Navigation® functionality, allowing users to use attribute values to filter data.



The component displays the values for selected attributes. If base refinement statistics are enabled in the MDEX Engine, then the component also displays the number of matching records for a selected value.

The component does not show contracted or implicit attribute values. It also does not show attribute values that lead to a dead end (no matching records).

From the **Guided Navigation** component, end users select values in order to refine the current data to only include records with those values. For some attributes, end users can select multiple values. They also may be able to do negative refinement, to only include records that do NOT have a selected value.

The component does not support:

- Multiple selection for hierarchical attributes
- Tree view controls
- Aggregated statistics

To allow end users to view and clear their selected **Guided Navigation** refinements, you must include a **Breadcrumbs** component that uses the same data source.

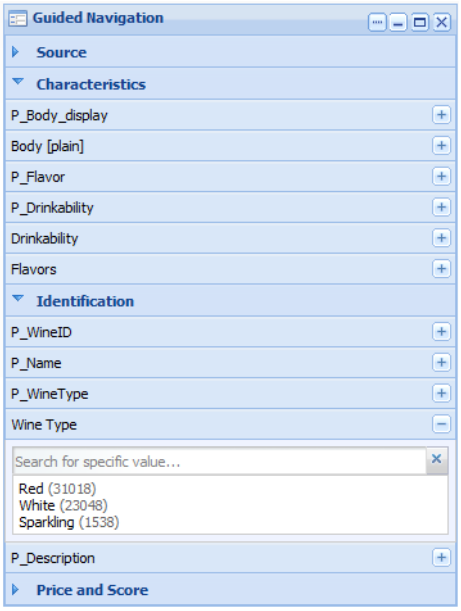
Using Guided Navigation

End users use the **Guided Navigation** component to refine the displayed data.

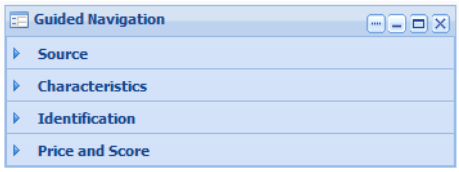
Displaying the available attribute values

The **Guided Navigation** component displays the attribute groups that end users can use to filter the data.

The values for each attribute may be displayed or hidden by default.

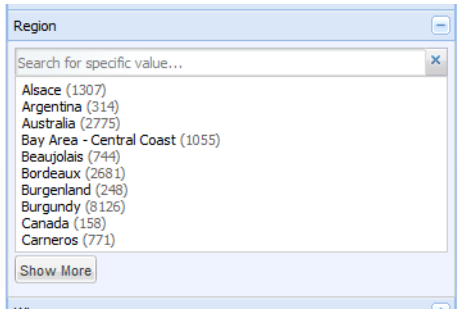


To show or hide the list of attributes for a group, click the group name.



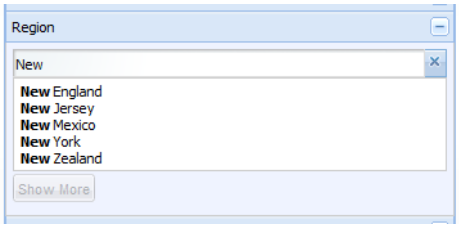
To display or hide the available values for an attribute, click the attribute name, or use the **+** and **-** buttons for the attribute.

If the number of values is too large, then a **Show More** button is displayed:



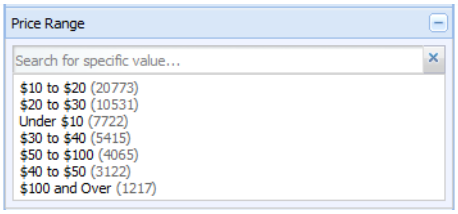
To display the remaining values, up to the configured total maximum, click the button.

To search for a specific value for an attribute, begin typing the value in the attribute search box. As you type, the matching values are displayed.



Selecting a single, non-hierarchical attribute value

For attributes that are not hierarchical, and do not allow multiple selections, the list of values is displayed.

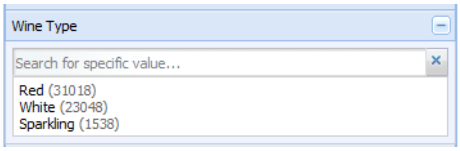


When you click a value:

- The data is filtered to only display records with that value.
- The selected value is added to the **Breadcrumbs** component.
- The attribute is removed from the **Guided Navigation** component.

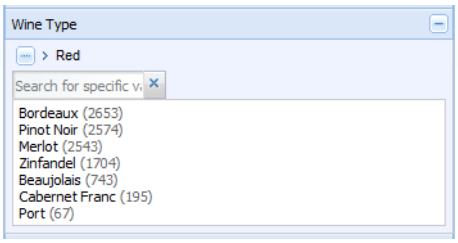
Selecting hierarchical attribute values

For a hierarchical attribute, only **Guided Navigation** component initially displays the top level of the hierarchy.



When you click an attribute value:

- The data is filtered to only display records with that value.
- The selected value is added to the **Breadcrumbs** component.
- On the **Guided Navigation** component, the child values for the selected value are displayed, with the parent value displayed above the list.



To return to the top level of the hierarchy, click the **...** button.

When you click a child attribute value, the selected value is added to the existing breadcrumb for that attribute. If there is another level of child values, those values are displayed on the **Guided Navigation** component.

When you click a value that does not have any children, the attribute is removed from the **Guided Navigation** component.

Selecting a negative refinement for an attribute value

Attributes also can be configured to allow negative refinement. Negative refinement means that the data is refined to only include records that do NOT have the selected attribute value.

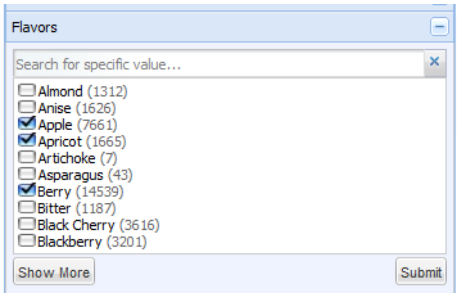
If an attribute allows negative refinement, then when you move the mouse over the value, a green dot (on the left) and red dot (on the right) are displayed next to the value.

To refine the data to only include records that have that value, click the green dot.

To refine the data to only include records that do NOT have that value, click the red dot. When you do a negative refinement, then a red dot also is displayed next to the breadcrumb.

Selecting multiple values for an attribute

If users can select multiple values for an attribute, then a checkbox displays next to each available value.



To select values for filtering, check the checkbox next to each value, then click the **Submit** button. When you filter by multiple values:

- Depending on the attribute configuration, the data is filtered to include records that either:
  - Have any one or more of the selected values
  - Have all of the selected values
- The selected values are added to the **Breadcrumbs** component.
- The selected values are removed from the **Guided Navigation** component.

End users can still select the other remaining values. If the user selects all of the values, then the attribute is removed from the **Guided Navigation** component.

Effect of filtering on the available attribute values

When you filter by a selected attribute value, it affects the available values for other attributes.

For example, a wine data set includes attribute for both the price range and the region. If you filter the list to only show wines with a price range of $30-40 dollars, then any region that does not have any wines in that price range is removed from the list of available region values.

The **Guided Navigation** component also is affected by filtering performed from other components. For example, when end users perform a search, the available attribute values on the **Guided Navigation** component are restricted to those associated with the matching records from the search.

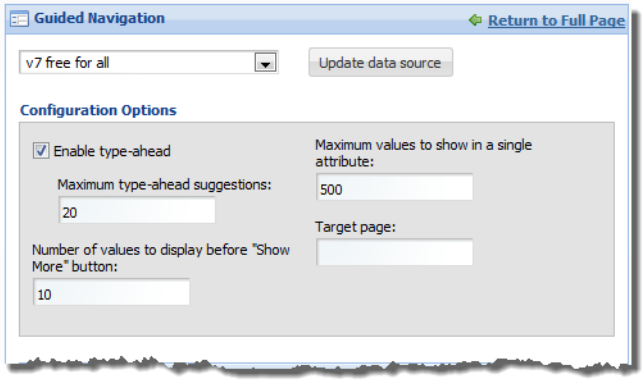
Setting the Guided Navigation data source and configuration options

At the top of the **Guided Navigation** component edit view are fields for configuring the data source and the general configuration options for the component.

Before configuring type-ahead suggestions, attribute search wildcarding must be enabled in the MDEX Engine.

To set the **Guided Navigation** data source and the configuration options:

1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.
2. Under **Configuration Options**:



Enable type-ahead	To enable the type-ahead search box for attribute values, check the checkbox. This feature is useful for attributes that have a large number of values. Before enabling this feature, make sure that your attributes have been configured to be wildcard searchable.
Maximum type-ahead suggestions	In the field, type the maximum total number of type-ahead suggestions the MDEX Engine will offer. For example, if you set this value to 10, then when end users use the type-ahead search, only the first 10 matching values are displayed.
Maximum values to show in a single attribute	In the field, type the maximum number of values per attribute to be displayed. The default value is 500. If the number of values is greater than this number, then end users can use the type-ahead search to find a specific value. Limiting the number of values to display can help prevent end users from being overwhelmed by an excessive number of results. **Note:** This threshold is in place for performance and usability reasons, to manage the end user's experience when attributes have very long lists of attribute values. This

	threshold can be used in conjunction with dynamic ranking and type-ahead search in the **Guided Navigation** interface, to allow users to get access to specific values by filtering for them.
Target page	In the field, type the name of the page to display when end users select a value. The selected value is passed to the components on that page. If you do not specify a target page, then the end user stays on the current page.
Number of values to display before "Show More" button	In the field, type the number of attribute values to display before displaying the **Show More** button. For example, if this value is set to 20, then when end users first select an attribute, only 20 values are displayed. If there are more than 20 values, then the **Show More** button is displayed. When end users click **Show More**, the remaining values are displayed, up to the maximum values setting.

3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

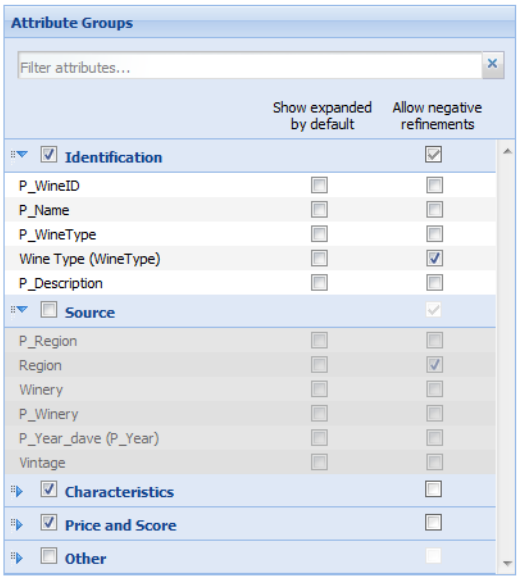
Configuring the attribute list for Guided Navigation

On the edit view of the **Guided Navigation** component, the **Attribute Groups** list determines the attribute groups that are displayed to end users.

To configure the attribute groups shown in the **Guided Navigation** component:

1. In the **Attribute Groups** list, to display an attribute group to end users, check the group checkbox. By default, the component displays all of the groups.

   To hide a group from end users, uncheck its checkbox.



2. To change the display order of the groups, drag the group to the new location in the list.

   You can expand and collapse the groups to make them easier to manage. To expand or collapse a group, click the group name.

3. In the displayed groups, for each attribute, you can configure:

Show expanded by default	To automatically expand the list of values for the attribute, check the checkbox. By default, the box is not checked. End users can always collapse or expand the list as needed.
Allow negative refinements	To allow negative refinements for the attribute, check the checkbox for that attribute. By default, the box is not checked. To enable or disable negative refinements for all of the attributes in a group, check or uncheck the group-level checkbox. If negative refinements are allowed, then end users can use the **Guided Navigation** component to exclude records that have a particular value.

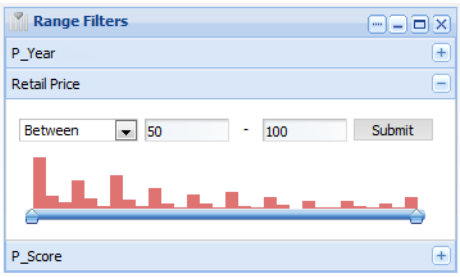
   To search for a specific attribute to configure, begin typing the name into the **Filter attributes** search box. As you type, the list is filtered to show the matching attributes.

4. To save your changes, click **Save Preferences**.
5. To exit the edit view, click **Return to Full Page**.

Range Filters

About the Range Filters component

The **Range Filters** component allows power users to create a set of filters. Each filter is for a specific attribute. End users can then use these filters to refine the displayed data to only include records that have attribute values within a specified range.



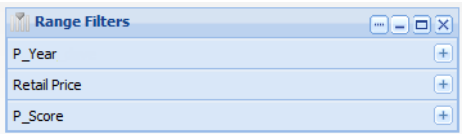
The **Range Filters** component can only be used for numeric and date attributes.

It cannot be used with managed attribute values.

Using Range Filters

End users can use the **Range Filters** component to select a range of values for which to display data.

When end users first view the page, the **Range Filters** component displays the list of configured range filters. Each filter is for a specific date or numeric attribute.



End users can expand one range filter at a time. To expand a range filter, either:

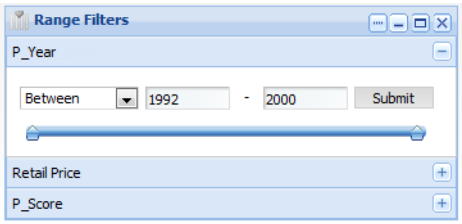
- Click the **+** button for that range filter
- Click the attribute name

When end users expand a range filter, if another filter is currently expanded, it collapses automatically. To collapse the expanded range filter manually, click the attribute name or **-** button for the filter.

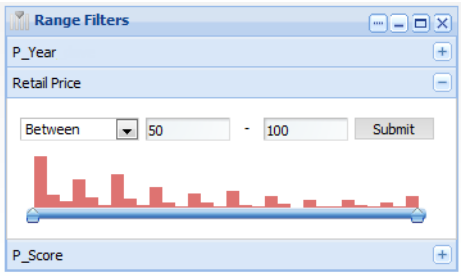
Each filter displays the current minimum and maximum values for the selected attribute. The value is either:

- A specific value configured by the power user
- The actual minimum/maximum value for the current refinement

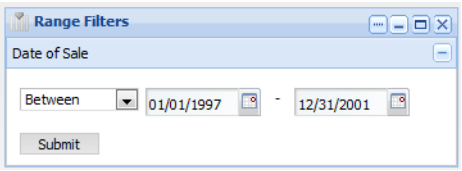
For numeric values, below the values is a slider bar:



A numeric filter also may be configured to include a histogram, which shows how the attribute values are distributed within the range:

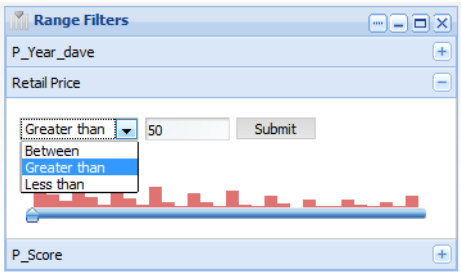


For date filters, there is no slider or histogram:



To use a numeric filter to refine the data:

1. From the drop-down list, select whether to specify a range:
   - **Between** two values
   - **Greater than** a selected value
   - **Less than** a selected value

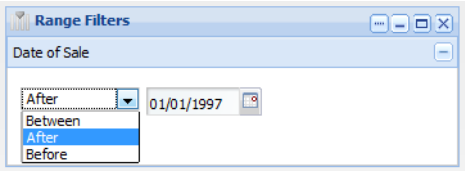


2. To specify the values to use, either:
   - Type the values in the field or fields
   - Use the slider bar to select the values
3. Click **Submit**.

To use a date filter to refine the data:

1. From the drop-down list, select whether to specify a range:
   - **Between** two dates

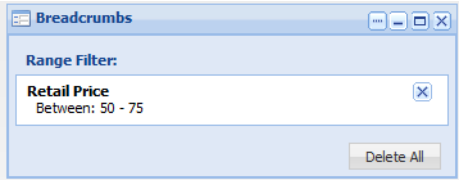
- **After** a selected date
- **Before** a selected date



2. To specify the values to use, either:
   - Type the dates in the field or fields
   - Use the date picker to select the date
3. Click **Submit**.

The data is refined to only include records with the specified value. Because the filter for an attribute can only be applied once, the filter is removed from the **Range Filters** component.

If there is a **Breadcrumbs** component for the same data source on the page, then the selected filter is added to it.



When you remove the filter from the **Breadcrumbs** component, it is added back to the **Range Filters** component.

Configuring a Range Filters component

For a **Range Filters** component, power users can configure the list of available filters. For each filter, they choose the attribute, the initial range of values, and for numeric filters, whether to include a histogram.

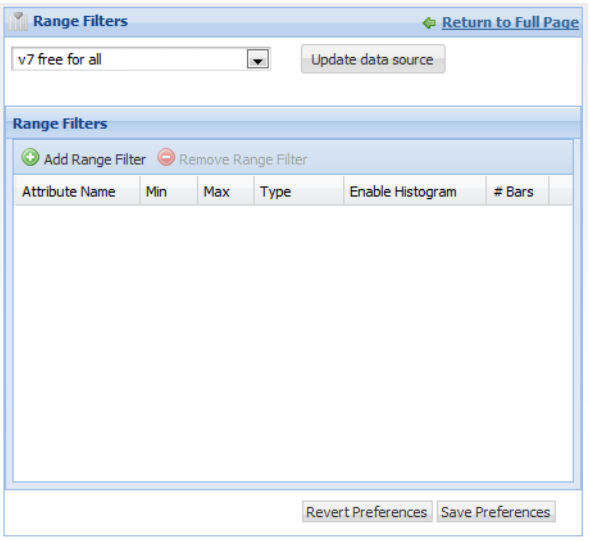
Before you can add a new range filter, the attribute already must exist as a date or numeric standard attribute in the MDEX Engine.

When you first add a **Range Filters** component, it displays a message indicating that additional configuration is needed.

To configure the **Range Filters** component:

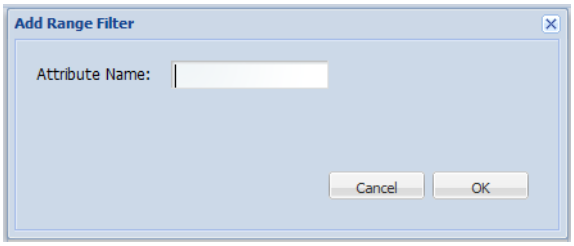
1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.

   The list of filters is cleared.



2. To add a new filter:
   a) Click **Add Range Filter**.

      The **Add Range Filter** dialog is displayed.



   b) In the **Attribute Name** field, type the name of the attribute.

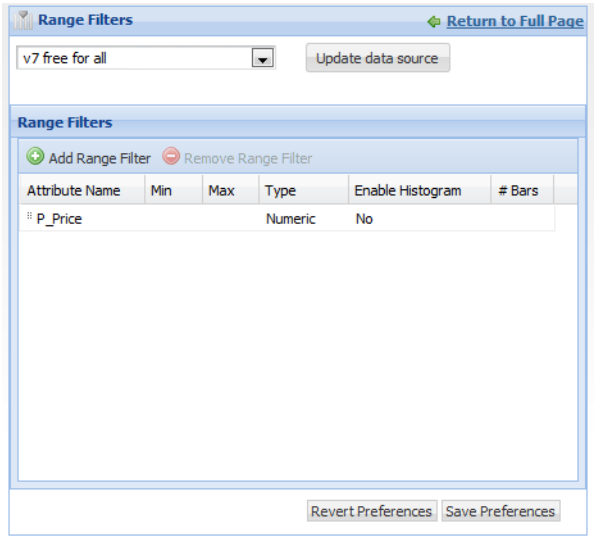
      The attribute must be either a date or a numeric attribute, and cannot be a managed attribute.

      You must know the exact name of the attribute. Attribute names are case sensitive.

      If you are unsure of an attribute's name, you can use the **Attribute Settings** component to find it.

   c) Click **OK**.

      The range filter is added to the list.



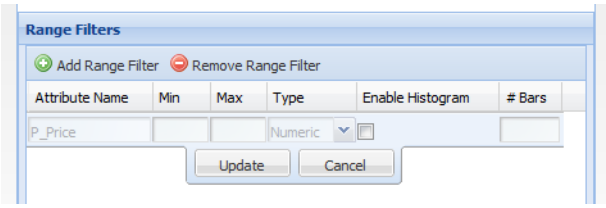
Note that the new filter is not actually saved until you save the component.

When the new filter is added:

- There are no minimum and maximum values specified.
- The type is based on the attribute type.
- The filter does not include a histogram.

3. To edit a filter:
   a) Double-click the filter row.

   The fields for that row are enabled for editing, except for the **Attribute Name** and **Type**. You cannot change those fields.



   b) To update the filter settings:

Min	In the field, enter the minimum value to allow for the range filter. If you do not provide a minimum value, then the filter displays the minimum value for the current refinement. For date filters, the **Min** value must be given in milliseconds-since-epoch format.
Max	In the field, enter the maximum value to show for the range filter. If you do not provide a maximum value, then the filter displays the maximum value for the current refinement. For date filters, the **Max** value must be given in milliseconds-since-epoch format.

Enable Histogram	For a numeric filter, to display a histogram for the filter, check the checkbox. If the box is not checked, then the slider bar displays without the histogram. For date filters, this checkbox cannot be checked.
#Bars	For numeric filters, if the **Enable Histogram** checkbox is checked, then in the **#Bars** field, type the number of bars to display in the histogram. The default is 30. The more bars that are displayed, the more precise the histogram.

c) To confirm the changes, click **Update**.

Note that any changes to a filter are not actually saved until you save the component.

4. To remove a filter:
   a) Click the filter row.
   b) Click **Remove Range Filter**.

   The row is removed from the table. Note that the filter is not actually removed until you save the component.
5. To change the order in which a filter is displayed, drag and drop the row to the new location in the list.
6. To save the changes to the component, click **Save Preferences**.
7. To exit the edit view, click **Return to Full Page**.

Search Box

About the Search Box component

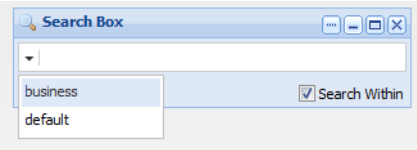
The **Search Box** component provides a search function for a Latitude Studio application.

It allows end users to submit keyword searches. The **Search Box** component also can be configured to provide type-ahead suggestions listing attribute values that match the typed text.

Using Search Box

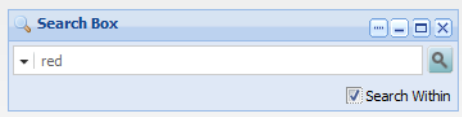
End users use the **Search Box** component to find records that contain a specified search term.

If multiple search configurations are available, end users can first select the search configuration that they want to use. The search configuration determines the data source, how to determine a matching record, and whether to support type-ahead.



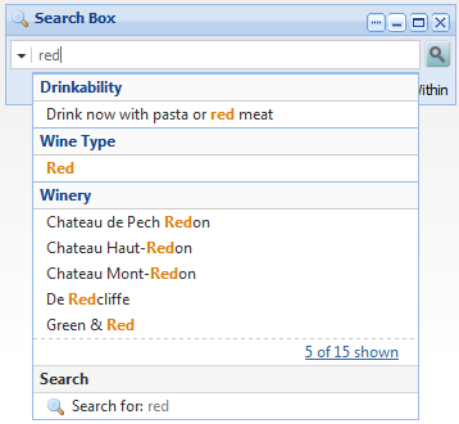
After selecting the search configuration, to do a basic search, end users:

1. In the field, type the search term.
2. To only search within the currently displayed data, make sure the **Search Within** checkbox is checked.

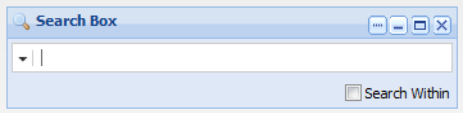


3. Click the search icon.

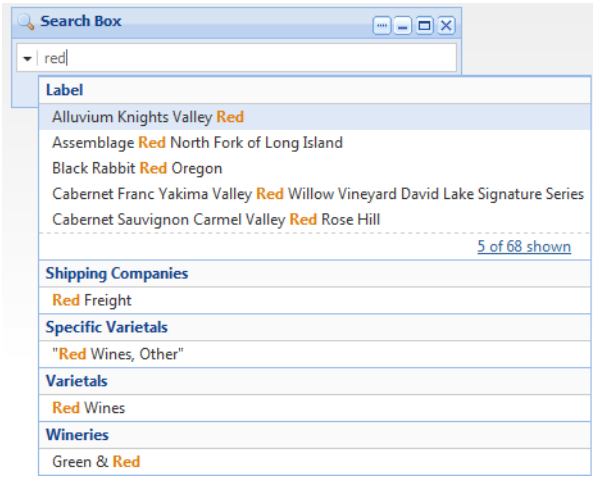
If type-ahead is supported, then as end users type the search term, a list of attribute values containing the search terms is displayed. To filter using an attribute value, end users click the value. Below the matching attributes is an option to just do a search. This option performs the same function as the search icon.



If the data source for the selected search configuration does not have a search interface, then only type-ahead is supported. The search icon does not display.



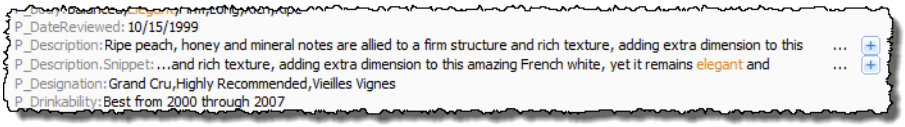
For type-ahead only searches, as end users type, the list of matching attribute values is displayed. There is no option to do a regular search.



The first value is selected automatically. To search by this value, end users can press **Enter**.

When users complete the search, the data is refined to only include records with the matching terms. Only components with the same data source are affected.

For regular text searches, on the **Data Explorer** and **Results List** components, the search terms may be highlighted. For attributes that support snippeting, the search snippet is displayed. The snippet displays the portion of the attribute value that contains the search terms.



For details on configuring snippeting for searches, see the *Latitude Developer's Guide*.

The **Breadcrumbs** component also is updated to add the search filter. If the **Search Within** checkbox was checked, then the breadcrumb is added to the list of breadcrumbs. If the box was not checked, the breadcrumb replaces the current list.

- If the end user performed a regular search, then the breadcrumb is a search breadcrumb.

- If the end user selected an attribute value, then the breadcrumb is an attribute value breadcrumb.

The selected value also is removed from the available values on the **Guided Navigation** component.

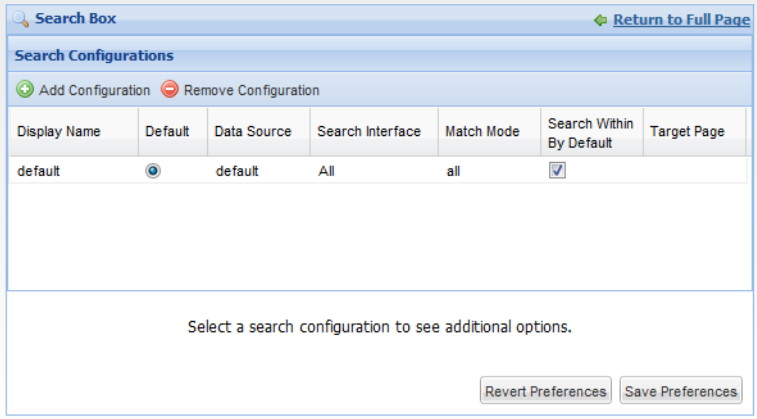
Configuring the Search Box component

For the **Search Box** component, power users create a list of available search configurations.

Managing the list of search configurations

At the top of the **Search Box** edit view is the list of search configurations.

For a new **Search Box** component, there is a single default configuration that uses the default data source:



To manage the list of search configurations:

1. To add a configuration to the list, click **Add Configuration**.

   A new row is added to the list.

   You can then click each column to display the field or drop-down list to set the configuration value.

2. To edit an existing configuration, click the configuration row in the list.

   You can then click each column to display the field or drop-down list to set the configuration value.

3. For each configuration:
   a) In the **Display Name** field, type the name of the search configuration.

      Each configuration must have a unique display name.

      The display name displays in the drop-down list on the **Search Box** component.

   b) From the **Data Source** drop-down list, select the data source to use for the configuration.

      The **Search Interface** drop-down list will only include search interfaces for the selected data source.

      If the selected data source does not have any search interfaces defined, then the **Search Interface** and **Match Mode** columns are disabled. End users will only be able to find attribute values. They will not be able to do a regular search.

   c) From the **Search Interface** drop-down list, select the search interface to use for the search configuration.

The search interface limits the end user's search, and allows you to control record search behavior for groupings of some number of attributes and ranking strategies.

d) From the **Match Mode** drop-down list, select a match mode for the search configuration. The match mode options are:

all	For this mode, a record only is returned if it contains all of the search terms. For example, if the user types `California red`, then only records containing both "California" and "red" are included. This is the default mode.
any	For this mode, a record is returned if it contains any of the search terms. For example, if the user types `California red`, then records that contain either "California" or "red" are included. The records do not have to include both terms.
partial	For this mode, a record is returned based on the partial search rules for the selected search interface. Each search interface can be configured with either: • A "Match at Least" rule, indicating the minimum number of matching terms • An "Omit at Most" rule, indicating the maximum number of terms that be not found in the record So if the search interface has a "Match at Least" rule, then records are only returned if they match at least that number of the search terms. For example, if the user types `California red berry sweet`, and the search interface "Match at Least" rule is 2, then only records with at least two of those terms are included. If the search interface has an "Omit at Most" rule, then records are only returned if they aren't missing more than that number of the search terms. For example, if the user types `California red berry sweet`, and the search interface "Omit at most" rule is 1, then only records with at least three of those terms are included.
partialmax	This mode is similar to the partial mode, except that the search stops when it finds records that contain the largest number of matching values. So it first looks for records that match all of the search terms. If it finds any, it stops looking, and returns those records. If it does not find any records with all of the search terms, it next looks for records that have all but one of the search terms. If it finds any, it then stops looking and returns those records. The "Match at Least" and "Omit at Most" rules still apply. The system will not search for records with fewer than the "Match at Least" rule, and will not remove more terms than the "Omit at Most" rule.

allany	For this mode, the search first looks for records that have all of the terms. If it finds any, it stops looking and returns those records. If none of the records have all of the search terms, then the search looks for records that have any of the search terms.
allpartial	For this mode, the search first looks for records that have all of the terms. If it finds any, it stops looking and returns those records. If none of the records have all of the search terms, then the search changes to partial mode.
boolean	This mode allows end users to do a Boolean search. The Boolean search supports the following operators: • AND • OR • NOT • NEAR • ONEAR

e) Use the **Search Within By Default** checkbox to determine whether to check the **Search Within** checkbox by default on the end user view.
f) In the **Target Page** field, type the name of the page on which to apply the search.

   For example, you could place the search box on one page, and a results table on a different page.

   If you do not provide a page name, then the end user stays on the current page.

4. To remove a configuration, click the configuration row, then click **Remove Configuration**.
5. To select the default configuration for the component, click the **Default** radio button for that configuration.
6. To change the sort order of the list, click the heading of the column you want to sort by.

   The sort order on the edit view also determines the order of the search configurations in the drop-down list on the **Search Box** component.

7. To save the changes to the list, click **Save Preferences**.
8. To exit the edit view, click **Return to Full Page**.

Configuring Search Box type-ahead suggestions

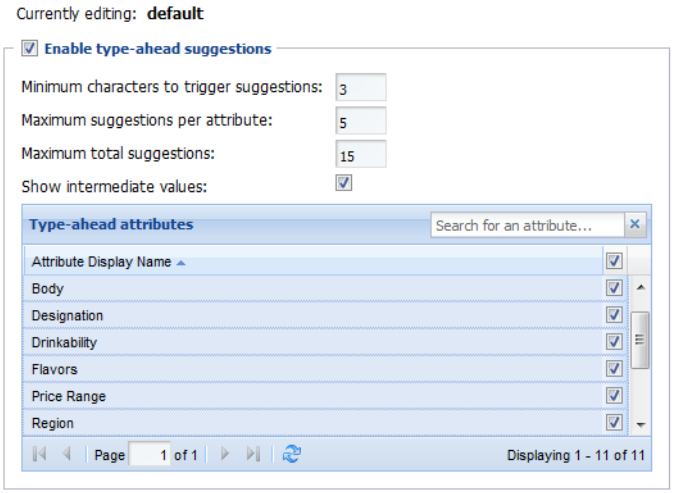
Each search configuration for a **Search Box** component can be configured to allow type-ahead suggestions. When type-ahead is enabled, the component displays attribute values that match the entered text while the end user types.

In order to use type-ahead suggestions, attribute search wildcarding must be enabled in the MDEX Engine.

On the edit view for the **Search Box** component, to configure type-ahead suggestions for a search configuration:

1. In the list of search configurations, click the configuration you want to edit.

   The type-ahead suggestions section for that configuration is displayed below the list.



2. To enable the type-ahead suggestions feature, check the **Enable type-ahead suggestions** checkbox.

   For a new configuration, the box is checked by default. If you uncheck the box, then the remaining fields are hidden.

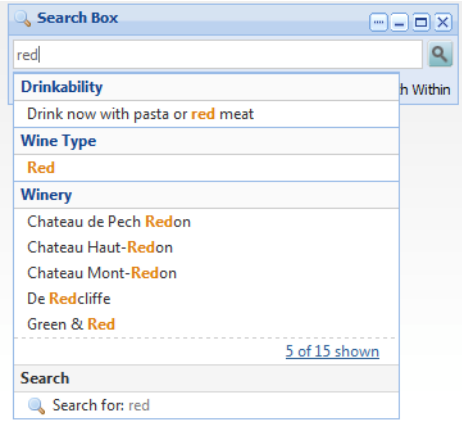
   If the data source for the configuration does not have any search interfaces defined, then the checkbox is checked and locked.

3. In the **Minimum characters to trigger suggestions** field, set the minimum number of characters the user must type before type-ahead suggestions are offered.
4. In the **Maximum suggestions per attribute** field, set the maximum number of type-ahead suggestions the MDEX Engine will offer for each attribute.

   If the number of values is greater than this number, than a link displays to allow the end user to see the other values.

5. In the **Maximum total suggestions** field, set the maximum total number of type-ahead suggestions the MDEX Engine will offer.

   So for example, if the maximum suggestions is 5, and the maximum total is 15, then if the number of values is greater than 5, the first 5 are shown, with a link to display the remaining values up to 15.



6. To display the full path for hierarchical attributes, check **Show intermediate values**.

   For example, if the user types `merlot` into the file, and the attribute value Merlot is a child of the attribute value Red, then:

   - If the box is checked, the type-ahead value is displayed as **Red > Merlot**.
   - If the box is not checked, the value is displayed as **Merlot**.

7. In the **Type-ahead attributes** list, check the checkbox next to each attribute for which you want to display type-ahead suggestions.

   To find a specific attribute, use the search filter field.

8. To save your changes, click **Save Preferences**.
9. To exit the edit view, click **Return to Full Page**.

Chapter 21

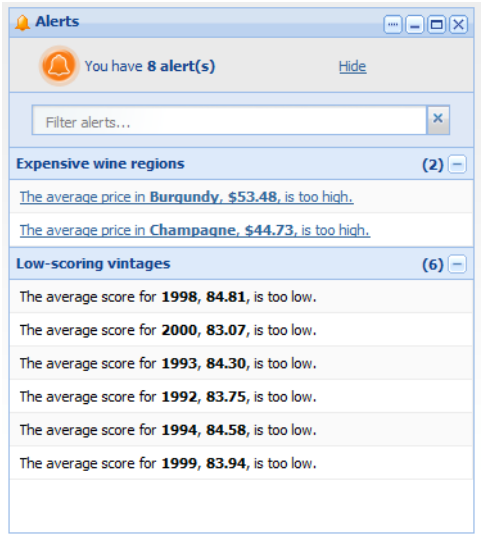
Data Visualization Components

These components provide a more detailed view of your data.

Alerts

About the Alerts component

The **Alerts** component displays alert messages to the end user. The alert messages are displayed based on values returned by LQL queries.

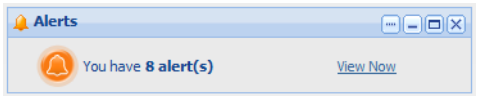


For example, alerts can be used to flag values that fall outside of a specific range.

Using the Alerts component

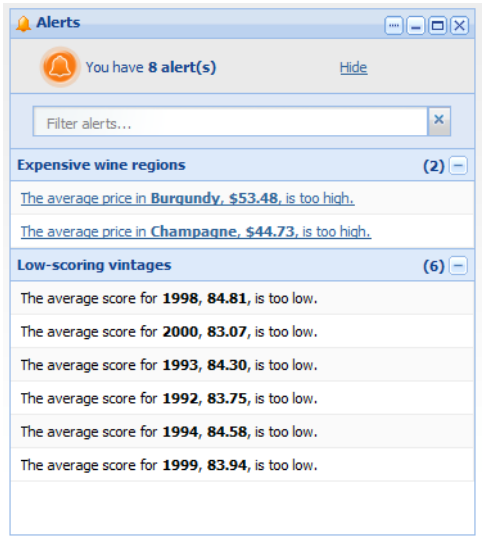
End users can view and filter alerts, and may be able to use alerts to refine the data set.

When end users first log in, if there are alerts available, the **Alerts** component displays with a message indicating the number of alerts.



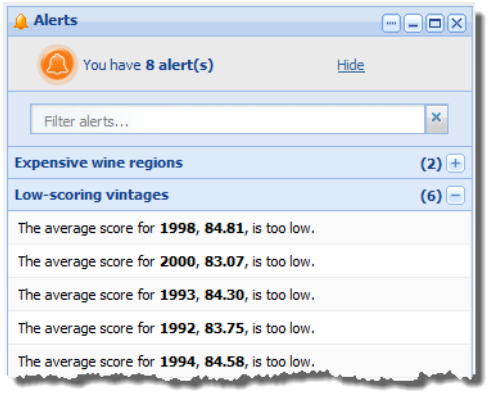
To display the list of the alerts, click **View Now**.

The alerts are divided into groups.

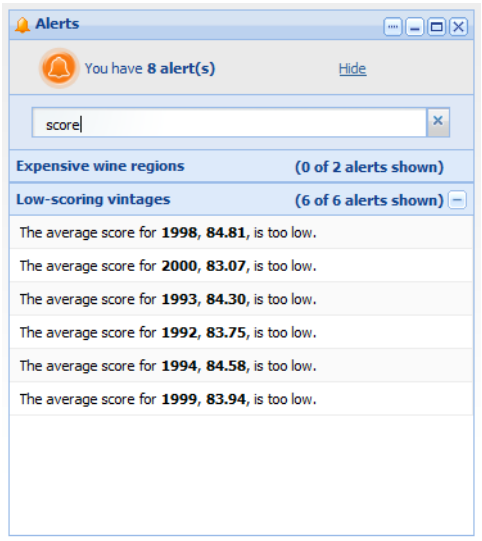


Each group may be configured to be expanded or collapsed by default. To expand or collapse the group, either:

- Click the **+** or **-** button to the right of the group.
- Click the group title bar.



The **Alerts** component also can be configured to include a search box at the top of the component. If the search box is available, then as you type text into the box, the alerts list is filtered to only include alert messages that include the typed text.



If the alert is hyperlinked, then when you click the alert, the data is refined using the attribute values associated with the message. You can be redirected to a different page to view the refined data, or the refinement can be done on the current page. The attribute value is added to the **Breadcrumbs** component for the target page.

For example, if an alert is displayed because the average price of wines in the Champagne region is over $50, then when end users click the alert, the data is refined to only Champagne wines. Because the average price is a calculated value, and not an attribute value, it cannot be used to refine the data.

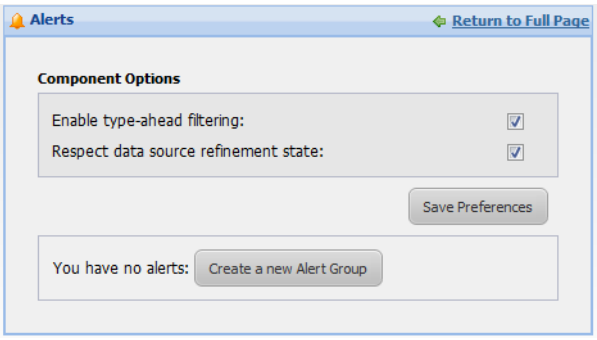
Configuring an Alerts component

The configuration for the **Alerts** component includes both general component options and the alert groups.

Setting the component options

On the edit view for the **Alerts** component, you can configure settings for filtering and data refinement.

For a new component, the edit view looks like:



To configure the component options:

1. Configure the checkboxes as follows:

Enable type-ahead filtering	If this checkbox is checked, then when end users display the list of alerts, there is a search box at the top to allow them to filter the list. If the box is not checked, then the search box is not displayed. This box is checked by default.
Respect data source refinement state	If this checkbox is checked, then the LQL query for the alert is only performed on the data for the current refinement. If the box is not checked, then the LQL query is performed on the entire data set. This box is checked by default.

2. To save the changes, click the **Save Preferences** button.

   Note that the **Save Preferences** button does not affect the alert group configuration.

Creating an alert group

For an **Alerts** component, you must create at least one alert group.

About alert groups

Alerts are displayed within alert groups.

Each alert group requires an LQL query to retrieve a set of matching records. The records are grouped by one or more attributes.

If the query does not return any records, then there are no alerts for that alert group, and the group does not display to end users.

If there are matching records, then there is a separate alert for each attribute value for the group-by attribute. For example, if an alert is being displayed when the average price of wine within a single region is greater than $50, then there is a separate alert for each region.

Adding an alert group

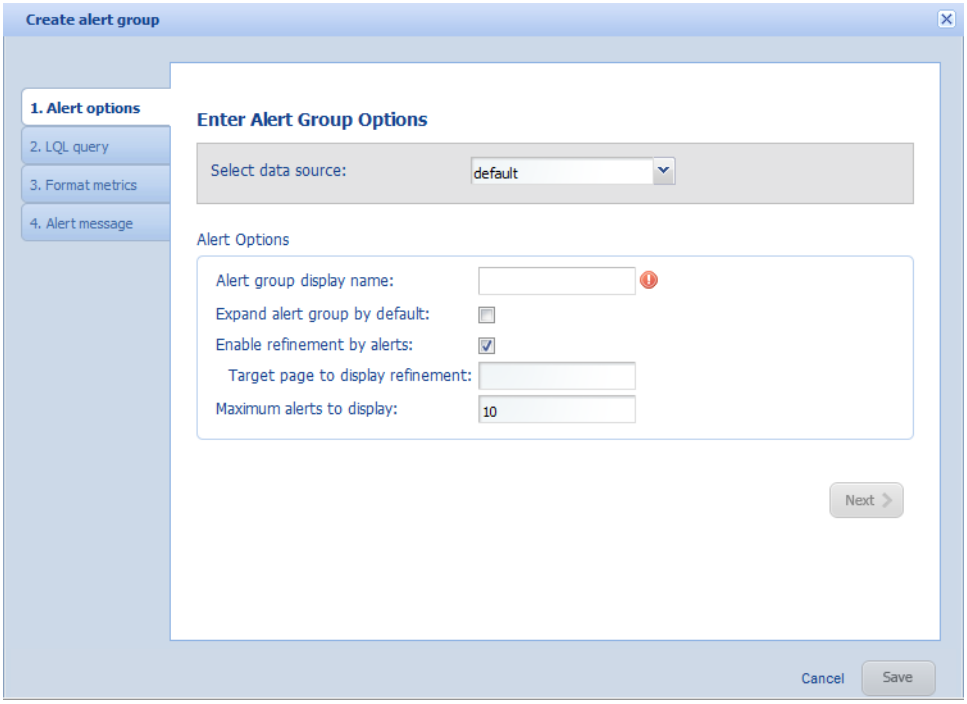
When you first add an **Alerts** component, it has no alert groups.

To add a new alert group, click the **Create a new Alert Group** button.

The **Create Alert Group** dialog is displayed.

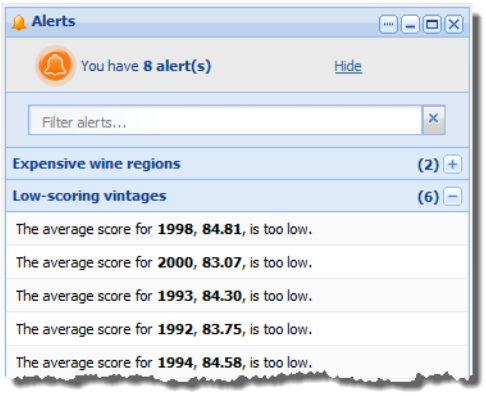
Configuring the alert options for an alert group

The **Alert options** tab for an alert group contains general configuration options for the alert group.



To configure the alert options:

1. From the **Select data source** drop-down list, select the data source to use for the LQL query.
2. In the **Alert Options** section:

Alert group display name	Required. In the field, type the name of the alert group. This is the name that displays to the end user.
Expand alert group by default	If this checkbox is checked, then when users display the alerts list, the alert group is automatically expanded to show the individual alerts. If the box is not checked, then the alert group is closed.  By default, this box is not checked.

Enable refinement by alerts	If this checkbox is checked, then the alert messages are hyperlinked. When end users click the alert message, the data is refined to only show the records that the alert applies to. By default, this checkbox is checked. If you uncheck the box, then the **Target page to display refinement** field is disabled.
Target page to display refinement	If you are allowing end users to use the alert message to refine the data, then in this field, type the name of the page to display when users click the alert message. If you do not provide a page name, then the end user stays on the same page.
Maximum alerts to display	Type the maximum number of alerts to display in the alert group. If the number of alerts is greater than this number, then end users cannot see the remaining alerts. The default value is 10.

3. To display the **LQL query** tab, click **Next**.

Configuring the LQL query for an alert group
On the **LQL query** tab, you configure the LQL query for the alert group. If any records match the query, then alerts are displayed.

To configure the LQL query:

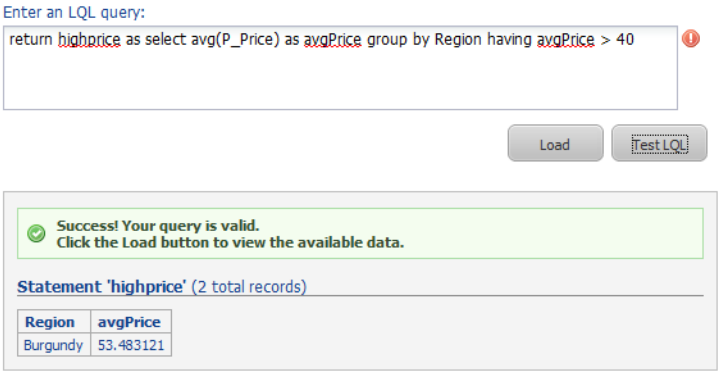
1. In the text area, enter the LQL query.

Note: When entering the query, remember that all attribute names, including names of derived attributes, must be NCName-compliant. They cannot contain spaces or special characters.

2. After you enter the query, to validate the query, click **Test LQL**.

   If the query is not valid, then an error message is displayed.

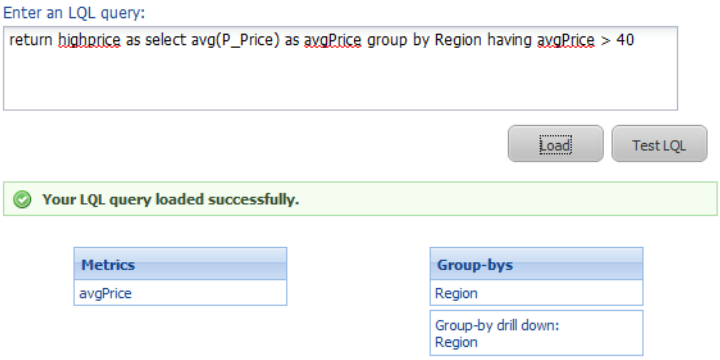
   If the query is valid, then a "success" message is displayed, and the **Load** button is enabled.



3. Click the **Load** button.

   The dialog is updated to display:

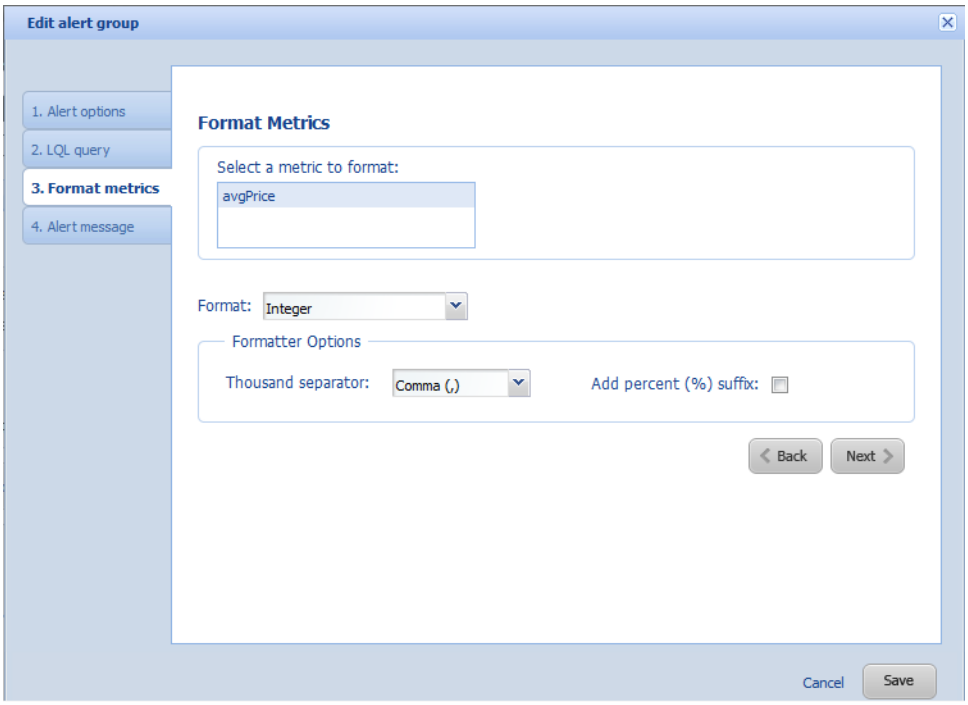
   - The metrics item(s) for the query. These are the values you are using for the comparison to determine whether to display an alert.
   - The group-by items for the query. This determines the number of alerts that display for the group. Below the full list of group-by items is the list of items that can be used for refinement.



4. To display the **Format metrics** tab, click **Next**.

Configuring the display of metrics values for an alert group

The alert message can include the metrics values as dynamic values. You use the **Format metrics** tab to configure how to display those values in the alert message.



At the top of the tab is the list of metrics values. To configure the display for a specific value:

1. Click the metrics value you want to format.
2. From the **Format** drop-down list, select the type of value.

   When you select a format, the **Formatter Options** section is updated to contain the relevant fields for the selected value type. The options are:

Integer	Indicates that the value is an integer. For integer values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value
Currency	Indicates that the value is a currency value. For currency values, you can configure: • The character to use to separate the thousands • The character to use as the decimal point • The number of decimal places to display • The currency symbol to use • Whether to display the currency symbol in front of the value (**prefix**) or behind the value (**suffix**)
Decimal	Indicates to display the value as a decimal value. For decimal values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value • The character to use as the decimal point

	• The number of decimal places to display
String	Indicates that the value is a text string. For string values, you can configure the capitalization. The options are to: • Display the value without changing the capitalization • Convert the entire value to lower case • Convert the entire value to upper case • Display the value in title case - the first letter in each word is capitalized

3. After configuring the format, to display the **Alert message** tab, click **Next**.

Configuring the alert message for an alert group
You use the **Alert message** tab to configure the text of the alert message for the alert group.

To configure the alert message for an alert group:

1. In the text field, type the text of the message.
2. To add a metric or group-by value, you can either:
   - Drag the value from the **Metrics** or **Group-bys** list
   - Type the value manually. When typing the value manually, remember to enclose the value in braces. For example, `{AvgPrice}`.

When you provide the alert message, if the alert group is complete, the **Save** button changes to a **Done** button.

Saving changes to an alert group
The alert group configuration is saved from the alert group dialog.

To save an alert group:

From the alert group dialog box, either:

- Click the **Save** button
- On the **Alert message** tab, click the **Done** button

If the alert group is complete, then it is displayed on the preferences panel under the **Alert Groups** heading. Complete alert groups can be displayed to end users on the **Alerts** component.

If the alert group is not complete (for example, if it doesn't have a valid LQL query or an alert message), it is displayed under the **Incomplete Alert Groups** heading, and is not displayed to end users.

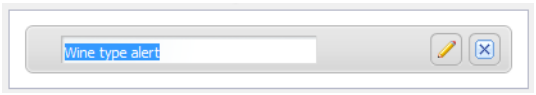
Editing an alert group

After creating an alert group, you can make changes to the group configuration.

To simply edit the display name of the alert group, then:

1. On the **Alerts** component edit view, double-click the group name.

   The group name becomes editable.



2. Type the new name, then press **Enter**.

To change the order in which the alert groups are displayed, drag each alert group to the correct location in the list.

To make more significant changes to an alert group:

1. On the **Alerts** component edit view, click the edit icon for the alert group.

   The **Edit alert group** dialog is displayed.

2. Make the required changes.

   If you change the selected data source, then you must retest and reload the LQL query.

   Latitude Studio preserves as much of the alert group configuration as possible. For example, if the updated query uses the same metrics value, then the metrics formatting is preserved.

3. To save the changes, click **Save** or **Done**.

Deleting an alert group

You can also delete alert groups from an **Alerts** component.

To delete an alert group from an **Alerts** component:

1. On the **Alerts** component edit view, click the delete icon for the alert.

   A message displays to confirm whether you want to delete the group.

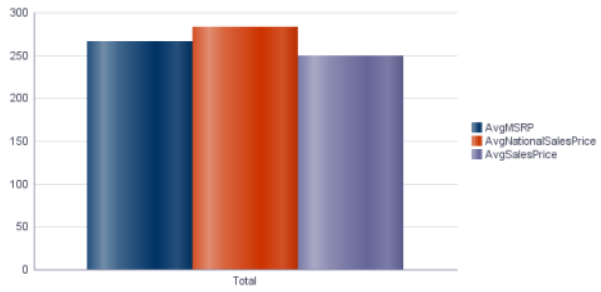
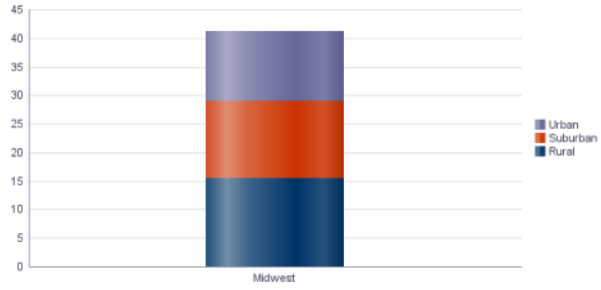
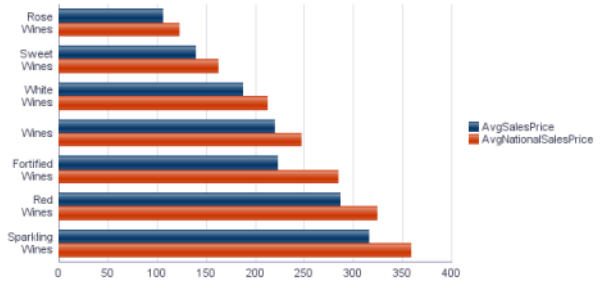
2. On the confirmation prompt:
   - To delete the alert group, click **Yes**. You cannot undo this change.
   - To not delete the group, click **No**.

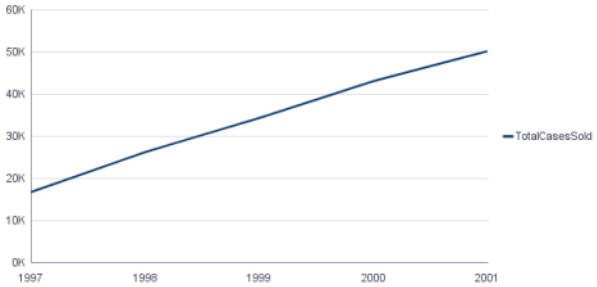
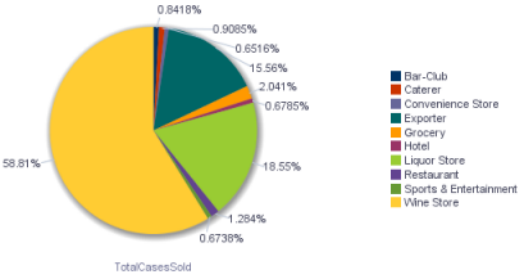
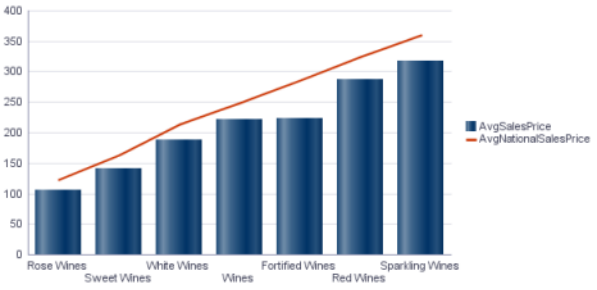
Chart

About the Chart component

The **Chart** component allows end users to view LQL-based charts of the current data.

It supports the following chart styles:

Vertical Bar	
Stacked Vertical Bar	
Horizontal Bar	

Line	
Pie	
Line Bar Combo	

The **Chart** component can contain multiple individual charts. End users select which chart they want to view.

End users can mouse over a chart to reveal numeric data, and click a chart element to filter results by the attribute values displayed in the chart.

End users also may be able to select a different metric or grouping option to display on the chart.

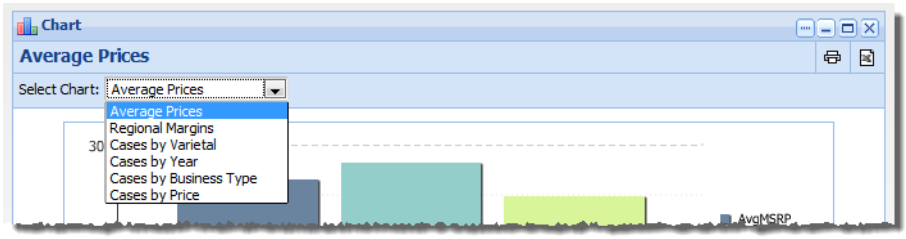
The **Chart** component also requires the use of LQL.

Using Chart

End users can modify the display of the **Chart** component, and use the chart elements to refine the data.

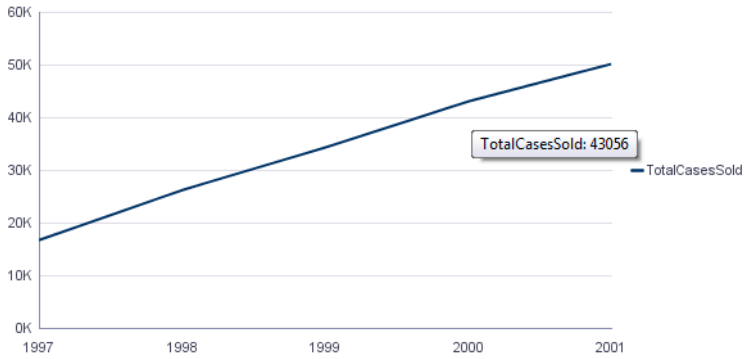
Selecting the chart to display

If the **Chart** component contains more than one chart, then at the top of the component is a drop-down list of the charts. To display a chart, end users select it from the list.



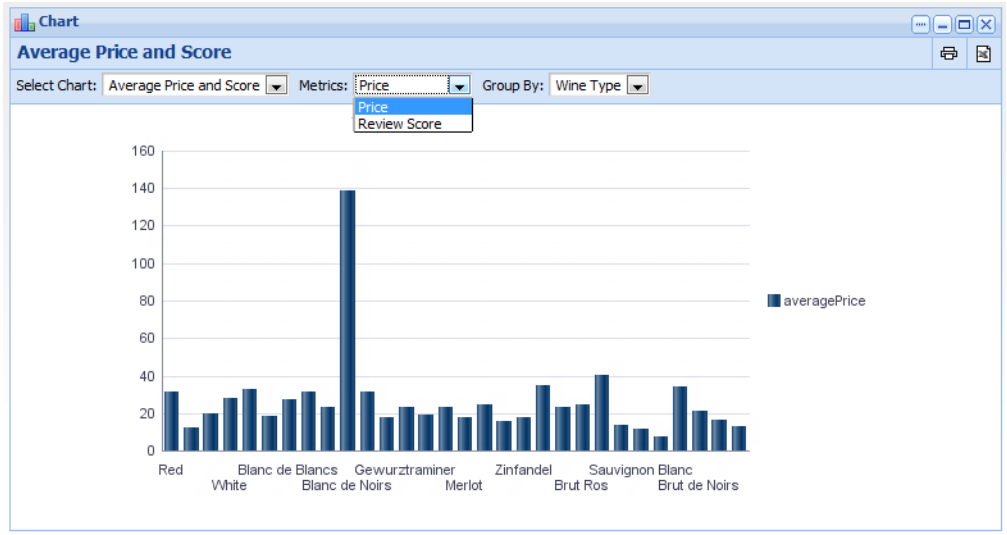
Displaying related data for a chart element

For each chart, end users can mouse over chart elements to see the related data.



Changing the display of a parametric chart

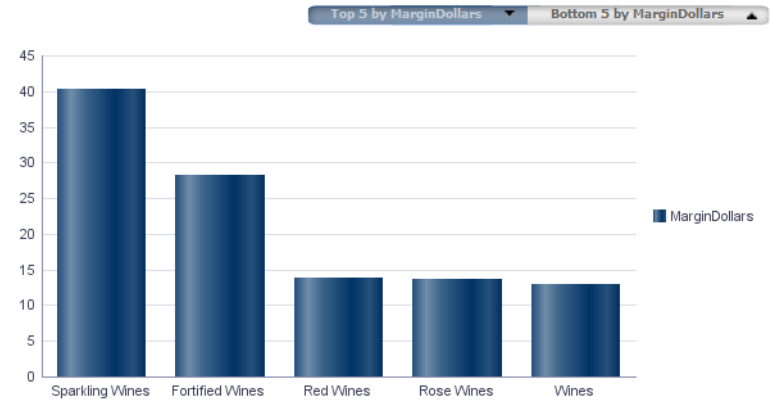
Power users also can create parametric charts, which allow end users to modify the chart display by selecting values from the drop-down lists at the top of the chart.



Using the "top n" function to change the display

If there are a large number of values, the chart may only display the top or bottom "n" number of values. For example, the chart may only show the top 10 sales figures or the 5 lowest performing regions.

In that case, the chart may include a toggle to switch between showing the top and bottom set of values.



Using the chart to refine data

When end users click a chart element, the data is refined by the group-by attribute value for the selected chart element.

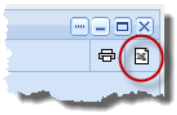
The attribute value is added to any **Breadcrumbs** component that uses the same data source.

If the chart is configured to cascade the group-by attributes, then when end users refine the data based on a group-by attribute, the chart may be updated to use a different group-by attribute.

Exporting a chart to a spreadsheet

End users can export the chart data to a spreadsheet.

To export the chart, end users click the export icon at the top right of the component.



They are then prompted to save or open the spreadsheet file.

Note that if the **Chart** component contains multiple charts, only the currently displayed chart is exported.

Printing a chart

End users also can print a chart.

To print a chart, end users click the print icon at the top right of the component.

A printable version of the component is displayed in a separate browser window, along with the print dialog.

End users select the print options, then print the chart.

Note that if the **Chart** component contains multiple charts, only the currently displayed chart is printed.

Configuring the Chart component

For the **Chart** component, power users can configure the data source, component dimensions, and the chart or charts to display.

Note: The **Chart** component includes online help, which mostly provides tips on working with LQL queries. The **Help** option is available from the same drop-down menu that contains the **Preferences** option.

Configuring the data source and chart options

The top sections of the edit view allow power users to configure the data source and general display options for the **Chart** component as a whole.

To configure the data source and display options:

1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.

   Note that the same data source is used for all of the charts created for the **Chart** component.

2. Under **Chart Options**, set the display options for the chart:

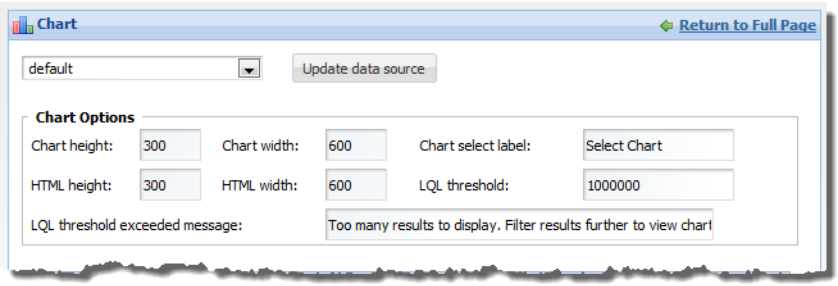


Chart height	In the field, enter the display height in pixels of the chart.
Chart width	In the field, enter the display width in pixels of the chart.
HTML height	In the field, enter the height in pixels of the HTML page in which to display the chart. For the best display results, this should be the same as the chart height.
HTML width	In the field, enter the width in pixels of the HTML page in which the chart will be displayed. For the best display results, this should be the same as the chart width.
Chart select label	In the field, enter the label to use for the drop-down list from which end users select the chart to display. This is only used if the component contains more than one chart.

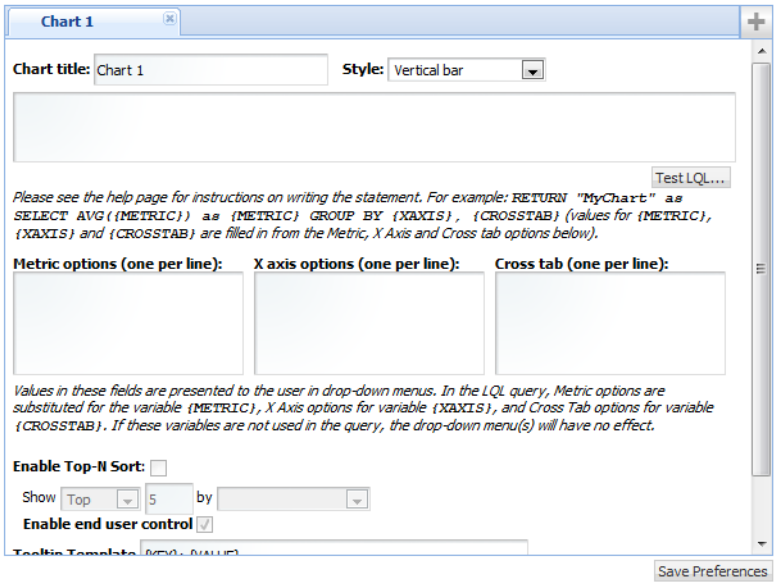
LQL threshold	In the field, type the maximum number of LQL results the query can return. If the number of results is larger than this number, then the message in the **LQL threshold exceeded message** field is displayed.
LQL threshold exceeded message	In the field, type the message to display if the number of query results exceed the value specified in the **LQL threshold** field.

Configuring charts for a Chart component

From the edit view of the **Chart** component, power users can create one or more charts to include on the component. For each chart, the power user selects the chart style, and provides an LQL query to generate the chart data.

Adding and removing charts

On the edit view of the **Chart** component, each tab represents a chart.



To add and remove charts for a **Chart** component:

1. To add a chart to the **Chart** component, click the **+** button.

   To remove a chart from the component, click the delete button at the top right corner of the chart tab.

2. In the **Chart title** field, enter the title for the chart.

   For end users, the title is displayed at the top of the chart. If the component contains multiple charts, it is displayed in the chart drop-down list.

   In the edit view, the title also is used as the chart tab label.

3. From the **Style** drop-down list, select the type of chart.

After creating and naming the chart, you then configure the LQL query, the metrics to include, and the display options.

Providing the chart LQL query and selecting metrics

Each chart requires an LQL query to generate the values used on the chart. For a parametric chart, instead of referring to specific attributes, the query contains placeholder values, and you provide a set of available attributes.

On the chart tab of the **Chart** edit view, to provide the LQL query and select the metrics:

1. To create a non-parametric chart:
   a) In the text area, enter the LQL query on which the chart will be based. For example:

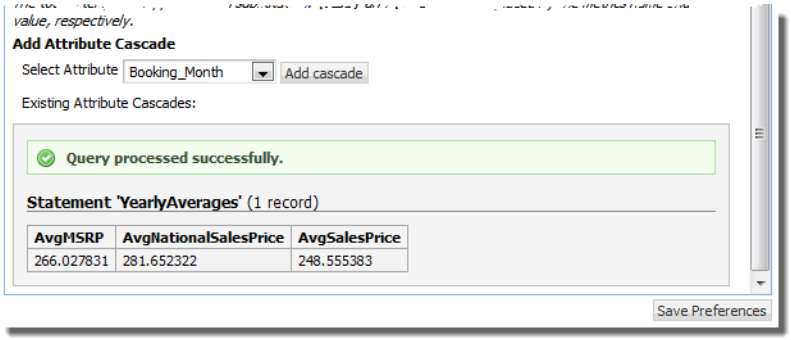
```
RETURN YearlyAverages AS SELECT
    AVG(Product_MSRPDollars) AS AvgMSRP,
    AVG(Product_Avg_National_Sales_PriceDollars) AS AvgNationalSale¬
sPrice,
    AVG(Unit_Sale_PriceDollars) AS AvgSalesPrice
GROUP
```

   Note that for non-parametric charts, the metrics labels are defined using the AS clause in the LQL query. The labels must be NCName-compliant (no spaces or special characters).

   b) To test the validity of the query, click the **Test LQL query** button.

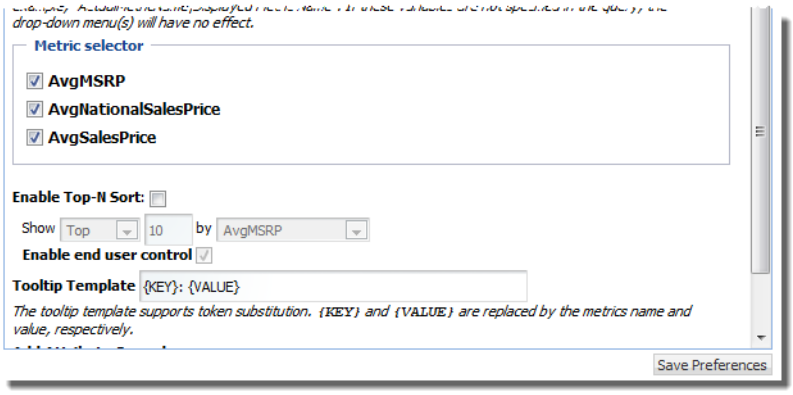
   If the query is not valid, then an error message is displayed.

   If the query is valid, then a "success" message is displayed, as well as the metric values from the query.



   c) To save the new or updated LQL query, click **Save Preferences**.

   The **Metric selector** section is displayed, listing all of the metrics from the current LQL query.



   d) By default, the chart displays all of the metrics. From the **Metric selector** section, to remove a metric from the display, uncheck its checkbox.

For example, your LQL query may include a metric to control the display order of the chart elements. However, you may not want the chart to display this value.

e) To save any changes to the configuration, click **Save Preferences** again.

2. To create a parametric chart, where end users can use drop-down lists to change the chart display:
   a) In the LQL query, replace the attribute names with the following placeholder values:
      - `{METRIC}` is replaced at runtime by a user-specified value from the **Metrics Options** list.

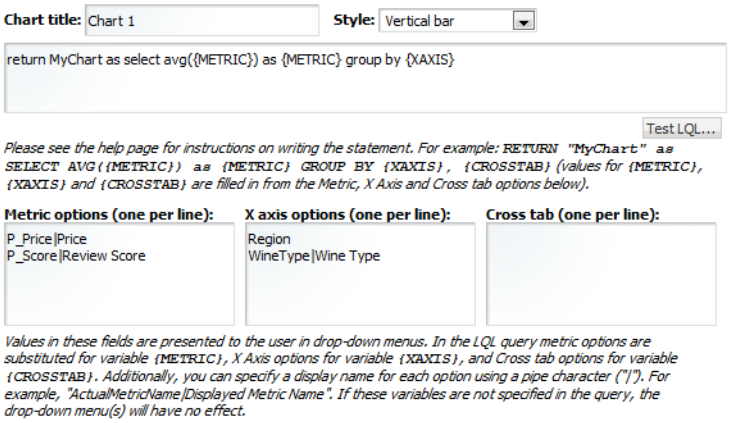
        If you use the `{METRIC}` token, you must use it for both the field to be returned and the "AS" alias for that field. The `{METRIC}` token is an exception to the rule that the metric name must be NCName-compliant.
      - `{XAXIS}` is replaced at runtime by a user-specified value from the **X Axis Options** list.
      - `{CROSSTAB}` is replaced at runtime by a user-specified value from the **Cross tab** list.

        Cross tabs are grouped aggregations such as cross-tabulated totals over one or more attributes.

      For example:

      ```
      RETURN MyChart AS SELECT AVG({METRIC}) as {METRIC} GROUP BY {XAXIS}
      ```

   b) In the **Metric options**, **X Axis options**, and **Cross tab** fields, enter the lists of attributes you want to use.



These attributes become the available options in the drop-down lists used to control the chart display.

For each attribute, you must provide the attribute key (not the display name). You also can provide a display name to use in the drop-down list on the end user view. The format is:

```
AttributeKey|Display Name
```

For example:

```
Trans_Country|Country
```

   c) To save the configuration, click **Save Preferences**.

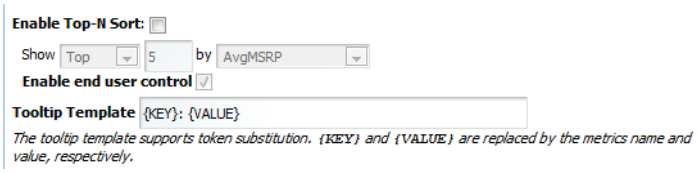
3. To exit the edit view, click **Return to Full Page**.

Enabling top "n" sort for a chart

By default, the chart shows all of the values generated from the LQL query. You can also configure a chart to only display a top or bottom set of values.

To only display the top or bottom "n" values on the chart:

1. Check the **Enable Top-N Sort** checkbox.



2. From the **Show** drop-down list:
   - To only display the top values, select **Top**.
   - To only display the bottom values, select **Bottom**.
3. In the field, enter the number of values to display.

   The default is 10.
4. From the **by** drop-down list, select the metric to use to sort the displayed values.
5. To allow end users to toggle between the top and bottom values, check the **Enable end user control** checkbox.
6. To save your changes, click **Save Preferences**.
7. To exit the edit view, click **Return to Full Page**.

Configuring the content of metric tooltips

When end users mouse over a metric value in a chart, a tooltip is displayed. The tooltip can include both the value and the metric name.

To configure the content of the tooltip:

1. In the **Tooltip Template** field, type the tooltip content.

   In the value, use `{KEY}` to represent the metric name, and `{VALUE}` to represent the value.

   The default value is `{KEY}: {VALUE}`, indicating to display the metric name followed by a colon and then the metric value. For example, "AveragePrice: 200".
2. To save changes to the configuration, click **Save Preferences**.
3. To exit the edit view, click **Return to Full Page**.

Configuring cascading for group-bys

When end users refine the data so that a chart displays only one metric value, you can configure the chart to cascade to a different group-by attribute. A cascade can contain multiple levels.

When there are no more attributes to cascade to, the chart changes back to the original group-by.

For example, a chart by default displays sales totals grouped by country. The chart also is configured so that Country cascades to State, which in turn cascades to Supplier.

With this configuration:

1. When users refine the data to only show records for the United States, the chart displays sales totals grouped by state (within the United States).

2. If users then refine the data to only show records for California, the chart displays sales totals grouped by supplier (within California).
3. If users then refine the data by a specific supplier, the chart changes back to displaying the sales total for the United States (for the selected supplier and within California).

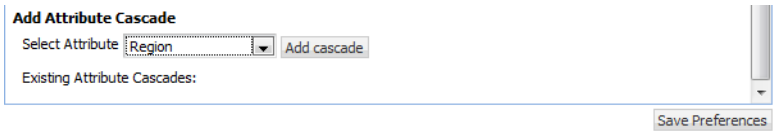
You can define multiple cascades for a chart. For example, if a chart has multiple group-bys, you can define a separate cascade for each one. Another use case for multiple cascades is a parametric chart. You can define a separate cascade for each group-by or cross tab option.

For each chart, you can configure the cascading for the group-bys.

Note: The attribute cascading feature does not work for attributes configured with the multi-or selection type. If you add a multi-or attribute to a cascade, the cascading stops at that attribute.

On the chart tab, to create a single cascade:

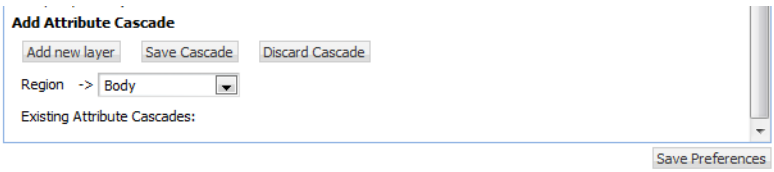
1. Under **Add Attribute Cascade**, from the **Select Attribute** drop-down list, select the attribute at the top of the cascade.



   This usually will be either a group-by from an LQL query or one of the group-by or cross tab options from a parametric chart.

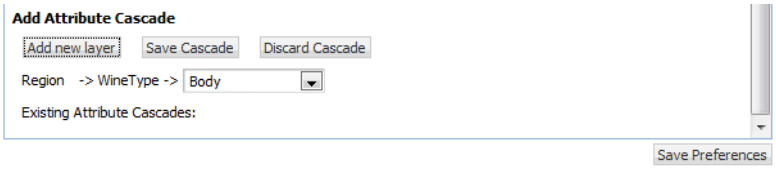
2. Click **Add Cascade**.

   A new cascade is started, with the selected attribute at the top. A drop-down list is displayed for you to select the next attribute in the cascade.



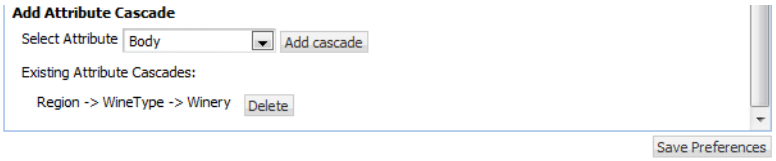
3. To add another layer to the cascade, click the **Add new layer** button.

   A drop-down is added to the end of the cascade.



4. To clear the settings for the cascade you're working on and start over, click **Discard Cascade**.
5. When you've finished adding layers to the cascade, to save the cascade, click **Save Cascade**.

   The cascade is added to the **Existing Attribute Cascades** list.



6. To delete a saved cascade, click its **Delete** button.

Chart component examples

Here are examples of each chart style. Each example includes both the LQL query and the resulting chart display.

Vertical bar chart

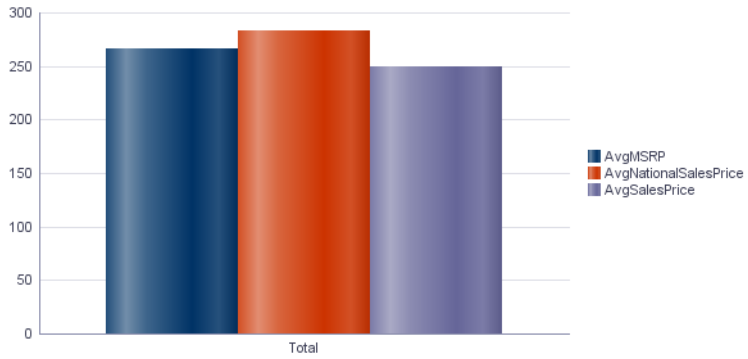
Here is an example of a vertical bar chart.

The LQL query for this example uses a zero `GROUP BY` statement with three metrics:

```
RETURN YearlyAverages AS SELECT
    AVG(Product_MSRPDollars) AS AvgMSRP,
    AVG(Product_Avg_National_Sales_PriceDollars) AS AvgNationalSalesPrice,

    AVG(Unit_Sale_PriceDollars) AS AvgSalesPrice
GROUP
```

The chart displays "AvgMSRP", "AvgNationalSalesPrice", and "AvgSalesPrice" metrics:



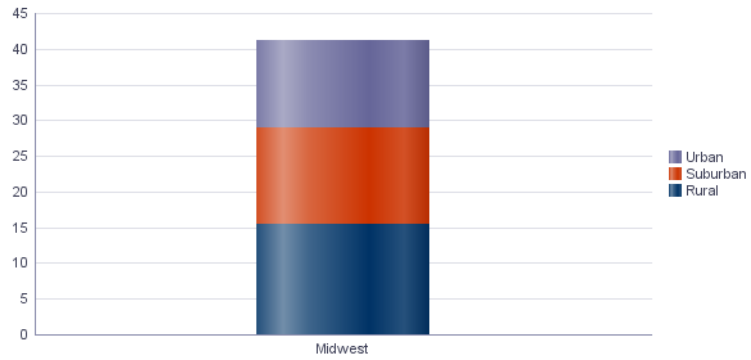
Stacked vertical bar chart

Here is an example of a stacked vertical bar chart.

The LQL query for this example has two `GROUP BY` statements. One metric is summarized by a specified categorization, and the second `GROUP BY` is specified as the cross tab.

```
RETURN RegionalMargin AS SELECT
   SUM(GrossDollars * MarginPercentage) / SUM(GrossDollars) AS DollarWeighte¬
dAvgMargin
GROUP BY Regions, Demographics
```

The chart summarizes the "DollarWeightedAvgMargin" metric for each Region (which is the first `GROUP BY` and therefore the primary category) and presents a cross-tabulation with each Demographic (which is the second `GROUP BY` and therefore the cross tab):



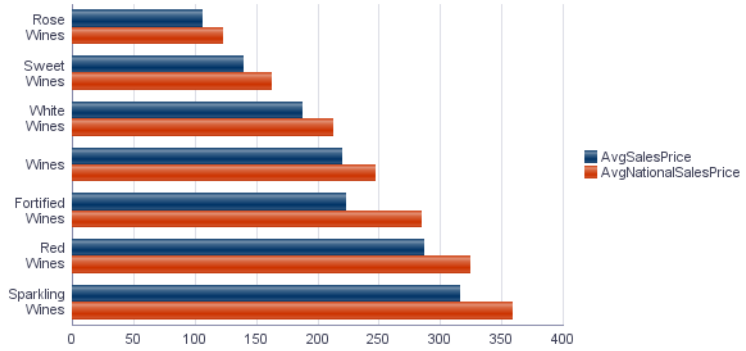
Horizontal bar chart

Here is an example of a horizontal bar chart.

The LQL query for this example uses a single `GROUP BY` statement with two metrics:

```
RETURN NumCases AS SELECT
    AVG(Unit_Sale_PriceDollars) AS AvgSalesPrice,
    AVG(Product_Avg_National_Sales_PriceDollars) AS AvgNationalSalesPrice
GROUP BY Varietals ORDER BY AvgNationalSalesPrice
```

The chart compares the "AvgSalesPrice" metric to the "AvgNationalSalesPrice" for each varietal:



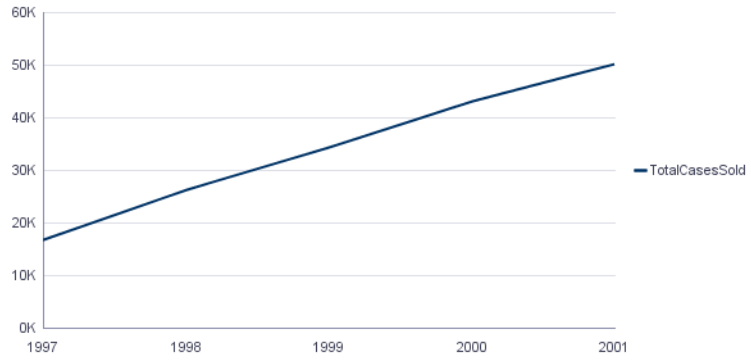
Line chart

Here is an example of a line chart.

The LQL query for this example has one `GROUP BY` and a single metric:

```
RETURN NumCases AS SELECT
    SUM(Number_of_Cases_Sold) AS TotalCasesSold
GROUP BY Booking_Year
```

The chart summarizes the "TotalCasesSold" metric for each Booking_Year:



Pie chart

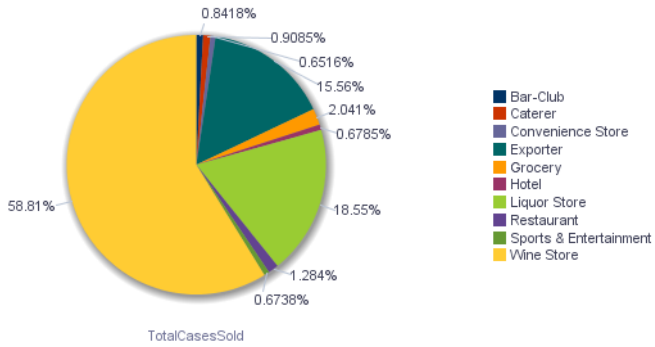
Here is an example of a pie chart.

The LQL query for this example contains one `GROUP BY` and a single metric:

```
RETURN NumCases AS SELECT
    SUM(Number_of_Cases_Sold) AS TotalCasesSold
GROUP BY Business_Types
```

Important: The pie chart type only supports a single `GROUP BY` statement. If you have multiple `GROUP BY`s in your statements, the chart will not display or execute drill-down as expected.

The chart summarizes the "TotalCasesSold" metric for each value of Business_Type:



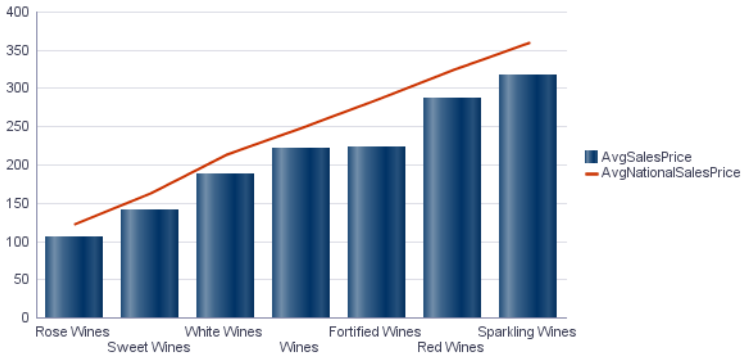
Line bar combo chart

Here is an example of a line bar combo chart.

The LQL query for this example contains one `GROUP BY` statement and two metrics:

```
RETURN NumCases AS SELECT
    AVG(Unit_Sale_PriceDollars) AS AvgSalesPrice,
    AVG(Product_Avg_National_Sales_PriceDollars) AS AvgNationalSalesPrice
GROUP BY Varietals ORDER BY AvgNationalSalesPrice
```

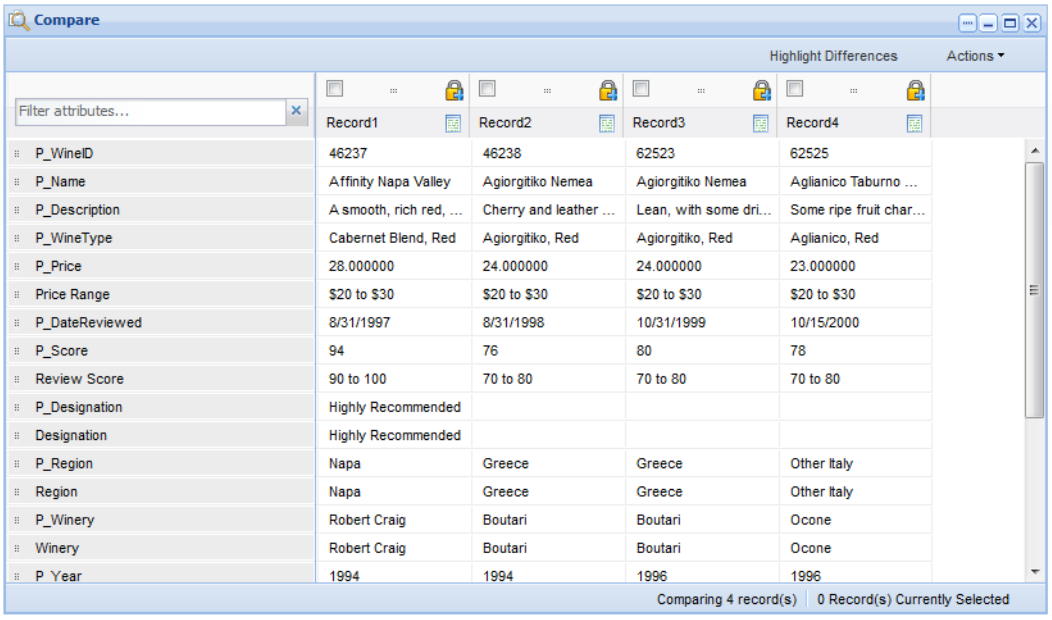
The chart compares the "AvgSalesPrice" metric to the "AvgNationalSalesPrice" for each varietal:



Compare

About the Compare component

The **Compare** component allows end users to view two or more records from a **Results Table** component side-by-side in order to analyze their differences and similarities.



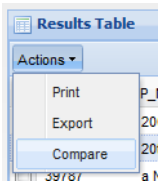
Using Compare

End users can use the **Compare** component to compare the attribute values for selected **Results Table** records. They can highlight differences among the records, and display record details for each record.

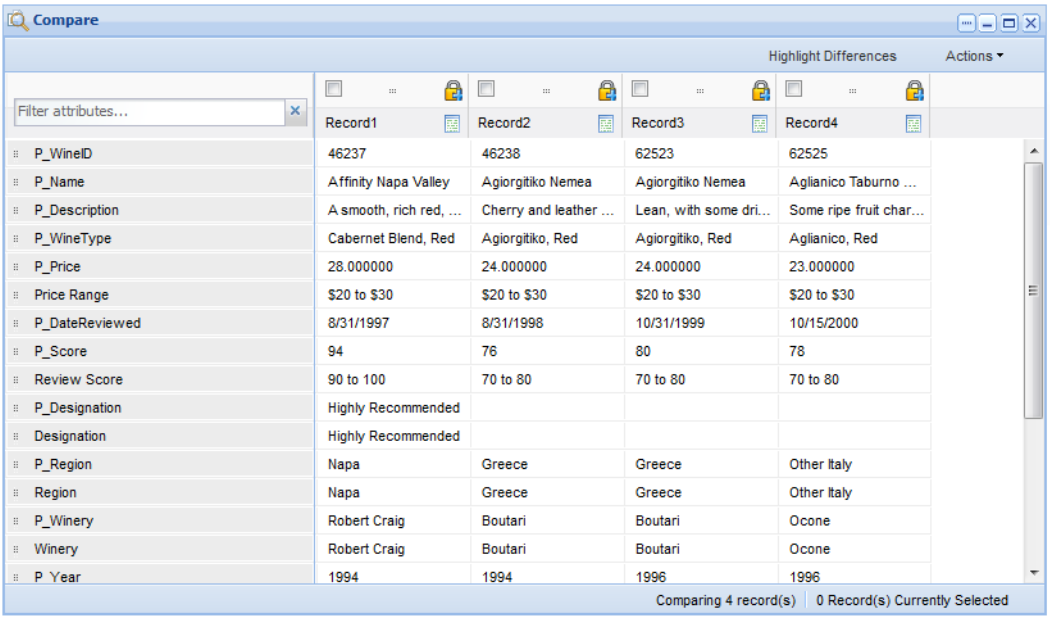
Selecting the records to compare

On the **Results Table** component, to select the records to compare using the **Compare** component:

1. Check the checkbox for each record you want to compare.
2. From the **Actions** menu, select **Compare**.

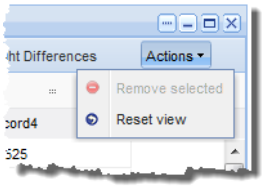


The selected records are displayed on the **Compare** component.



Depending on how the component was configured, the differences among the records may be highlighted.

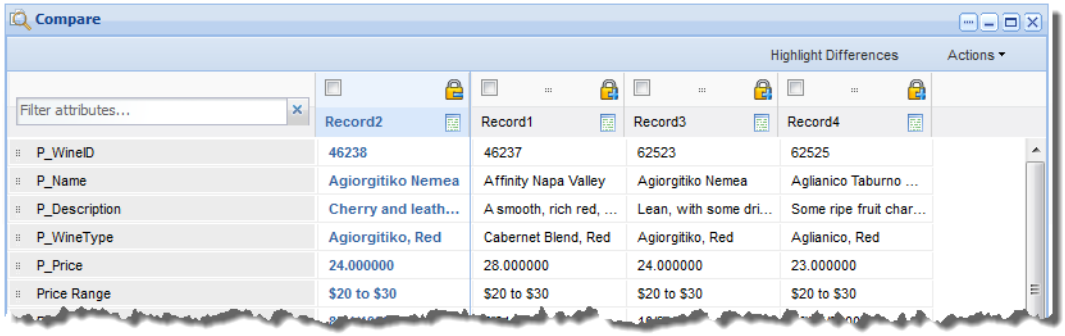
To restore the default display for the component, from the **Actions** menu, select **Reset view**.



Determining the display order of the records

To select a record to be the baseline record, in the record heading, click the lock icon.

The selected record becomes the first column in the list, and cannot be moved. The lock icon is changed to indicate that the record is a baseline record. If another record was previously selected as the baseline, it becomes a non-baseline record.



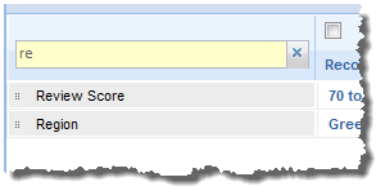
To remove the designation as a baseline record, click the lock icon again. The record column remains at the left of the table, but can now be moved.

You can drag non-baseline records left or right in the display, to allow you to do a side-by-side comparison of selected records.

Managing the displayed attributes

End users also can drag and drop attributes to move them up or down in the display.

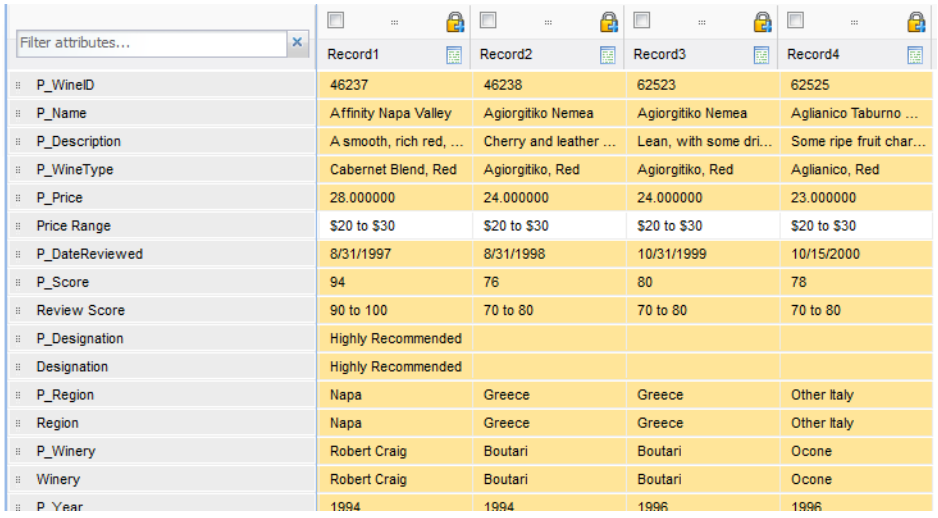
To search for a specific attribute, begin typing the attribute name in the **Filter attributes** field. As you type, the matching attributes are displayed.



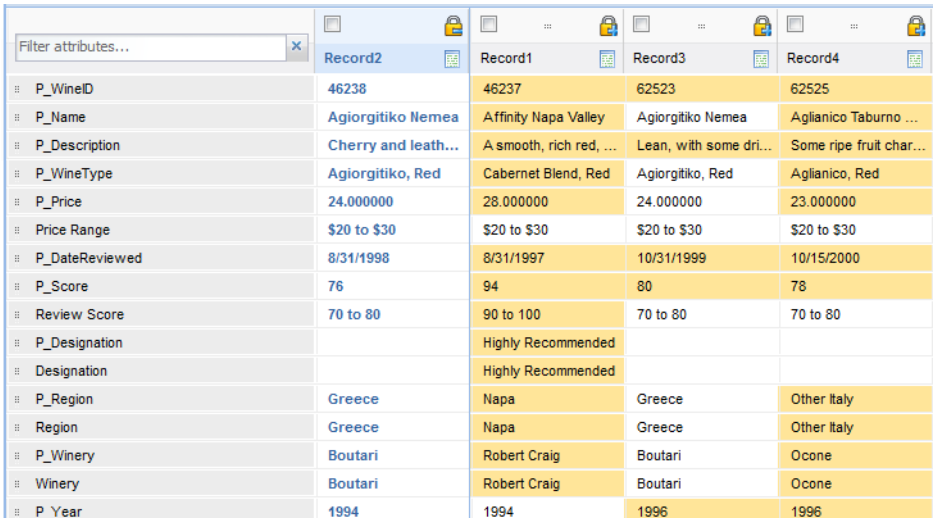
Highlighting differences between the records

To highlight the differences in attribute values, click **Highlight Differences**.

If the end user hasn't selected a baseline record, then the component highlights attribute values that are not the same across all of the selected records.



If there is a baseline record, then in the non-baseline records, the component highlights attribute values that are different from the baseline record:



Filter attributes...	Record2	Record1	Record3	Record4
P_WineID	46238	46237	62523	62525
P_Name	Agiorgitiko Nemea	Affinity Napa Valley	Agiorgitiko Nemea	Aglianico Taburno ...
P_Description	Cherry and leath...	A smooth, rich red, ...	Lean, with some dri...	Some ripe fruit char...
P_WineType	Agiorgitiko, Red	Cabernet Blend, Red	Agiorgitiko, Red	Aglianico, Red
P_Price	24.000000	28.000000	24.000000	23.000000
Price Range	$20 to $30	$20 to $30	$20 to $30	$20 to $30
P_DateReviewed	8/31/1998	8/31/1997	10/31/1999	10/15/2000
P_Score	76	94	80	78
Review Score	70 to 80	90 to 100	70 to 80	70 to 80
P_Designation		Highly Recommended		
Designation		Highly Recommended		
P_Region	Greece	Napa	Greece	Other Italy
Region	Greece	Napa	Greece	Other Italy
P_Winery	Boutari	Robert Craig	Boutari	Ocone
Winery	Boutari	Robert Craig	Boutari	Ocone
P_Year	1994	1994	1996	1996

When the highlighting is displayed, to remove the highlighting, click **Hide Highlights**.

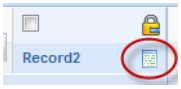
Removing records from the display

To remove records from the **Compare** component:

1. Check the checkbox for each record to remove.
2. From the **Actions** menu, select **Remove selected**.

Displaying the record details for a record

To display the details for a record, click the record details icon in the column heading for that record.



The **Record Details** component for that record is displayed.

Configuring a Compare component

For a **Compare** component, power users select the data source and the attribute values to display. They also set the display options for the table.

Selecting the data source for a Compare component

The **Compare** component requires a backing data source.

When you first add a **Compare** component, it uses the default data source.

On the component edit view, to bind the component to a different data source:

1. From the drop-down list, select the data source.
2. Click **Update data source**.

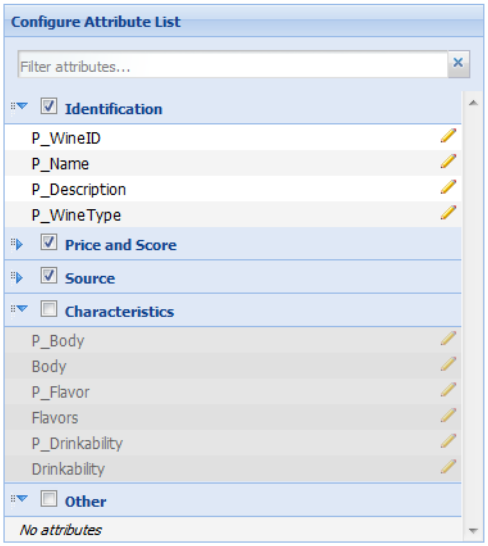
Selecting the attributes to display on a Compare component

For a **Compare** component, power users select the attribute groups for which to display attributes, and the sort order to use when the component is first displayed. By default, the attributes for all of the groups are displayed in the default sort order for the data source.

Note: If an attribute belongs to more than one of the selected groups, it is only displayed once on the end user view.

From the edit view of a **Compare** component, to configure the attribute groups for which to display attributes:

1. Under **Configure Attribute List**, to remove the attributes for an attribute group from the display, uncheck its checkbox.



   To restore the attributes for a hidden attribute group, check its checkbox.

   To find a specific attribute, so that you can see which group it is in, type the attribute name in the **Filter Attributes** field. As you type, the list is filtered to only display the matching attributes.

2. The default display order of the attributes is based on the order of the groups on the edit view. To change the default display order of the attributes, drag each group to its new location in the list.

   You can collapse and expand the attribute groups to make them easier to work with.

3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

Formatting the attribute values displayed on a Compare component

For each attribute displayed on a **Compare** component, end users can format the displayed value. They can select a different format, and configure specific options for the selected format type.

From the edit view of a **Compare** component, to format the displayed attribute values:

1. Under **Configure Attribute List**, click the edit icon for the attribute.

   The **Formatter** dialog is displayed. The default format for the attribute is based on the attribute's data type.

2. From the **Format** drop-down list, select the format to use for the attribute value, and then configure the options for the selected type.

   If you select a different format, make sure that it is an appropriate format for the value being displayed.

   The options are:

Integer	Indicates that the value is an integer. For integer values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value
String	Indicates that the value is a text string. For string values, you can configure: • Whether to change the capitalization of the text string • The number of characters after which to truncate the value
Currency	Indicates that the value is a currency value. For currency values, you can configure: • The character to use to separate the thousands • The character to use as the decimal point • The number of decimal places to display • The currency symbol to use • Whether to display the currency symbol in front of the value (**prefix**) or behind the value (**suffix**)
Decimal	Indicates to display the value as a decimal value. For decimal values, you can configure: • The character to use to separate the thousands • Whether to display the percent symbol after the value • The character to use as the decimal point • The number of decimal places to display
Date	Indicates that the value is a date. For a date value, you can configure the format to use. The options are: • **American style (MM/DD/YYYY)**. For example, for October 20, 2010, the American style date would be 10/20/2010. • **European style (DD/MM/YYYY)**. For example, for October 20, 2010, the European style date would be 20/10/2010.

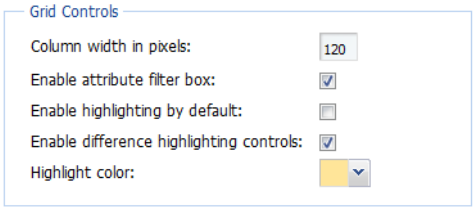
3. To save the changes, click **Save**.

Configuring the display options for a Compare table

For a **Compare** component, end users can configure how the table is displayed, and whether end users can search for attributes or control the highlighting.

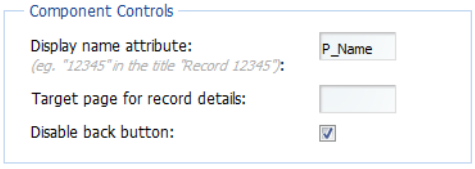
From the edit view of a **Compare** component, to configure the table display and options:

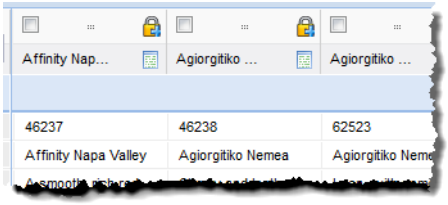
1. Under **Grid Controls**:



Column width in pixels	In the field, type the width in pixels of the record columns. Each column uses the same width.
Enable attribute filter box	If this checkbox is checked, then on the **Compare** component, the attribute search filter is displayed, to allow end users to search for specific attribute values. If the box is not checked, then the attribute search filter is not displayed, and end users cannot search for attribute values. By default, the box is checked.
Enable highlighting by default	If this checkbox is checked, then when the **Compare** component is first populated, it automatically highlights the differences among all of the selected records. If the box is not checked, then the differences are not highlighted automatically. By default, the box is not checked.
Enable difference highlighting controls	If this checkbox is checked, then on the **Compare** component, the **Highlight Differences**/**Hide Highlights** button is displayed, to allow end users to show or hide the highlighting. If this box is not checked, then the button is not displayed, and end users cannot show or hide the highlighting. By default, the box is checked.
Highlight color	From the drop-down list, select the color to use for the highlighting.

2. In the **Component Controls** panel:



Display name attribute	In the field, type the name of the attribute to use to identify the record in the column heading.  The attribute should be one that is unique for each record, such as a name or ID. If you do not specify an attribute, or if none of the records have a value for the selected attribute, then the records are labeled as Record1, Record2, etc.
Target page for record details	In the field, type the page on which to display the record details. The page must contain a **Record Details** component that uses the same data source as the **Compare** component. If you do not specify a page, then the current page is used.
Disable back button	If this checkbox is checked, then when the **Compare** component is on a different page from the **Results Table** component, there is no **Back** button to allow end users to navigate back to the page containing the **Results Table**. If the checkbox is not checked, then the **Back** button is displayed. Note that if the **Compare** and **Results Table** components are on the same page, the **Back** button is not displayed. By default, this box is checked.

3. To save your changes, click **Save Preferences**.
4. To exit the edit view, click **Return to Full Page**.

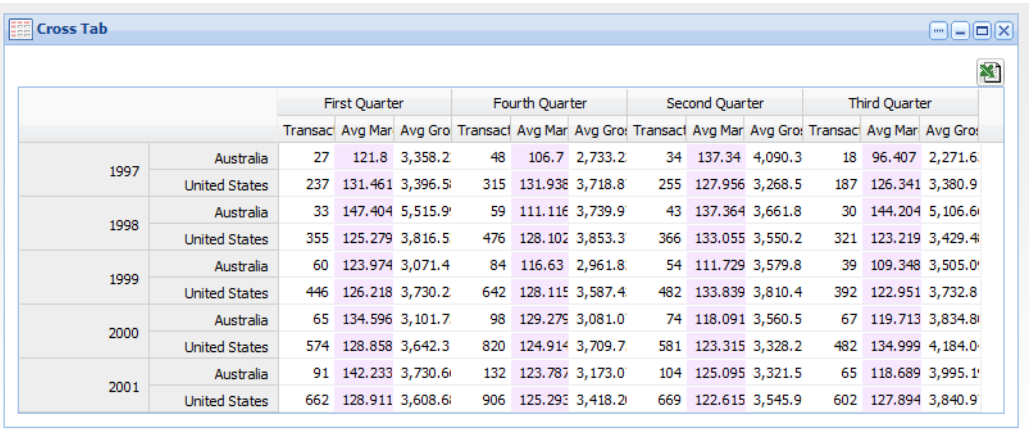
Cross Tab

About the Cross Tab component

The **Cross Tab** component generates a table that allows end 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 value of the specified 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, the first (top left) summary cell contains the number of transactions in Australia for the first quarter of 1997.

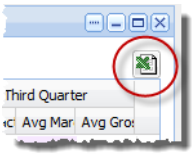


Using Cross Tab

End users can view the cross tab result table, but cannot edit or modify it.

The component can be configured to be updated automatically when end users refine the data.

End users also can export the contents of the **Cross Tab** component to a spreadsheet. To export the data, click the export icon at the top right of the component.



Configuring a Cross Tab component

For a **Cross Tab** component, power users provide the LQL query and configure how the table is displayed.

Configuring the data source and LQL query for a Cross Tab component

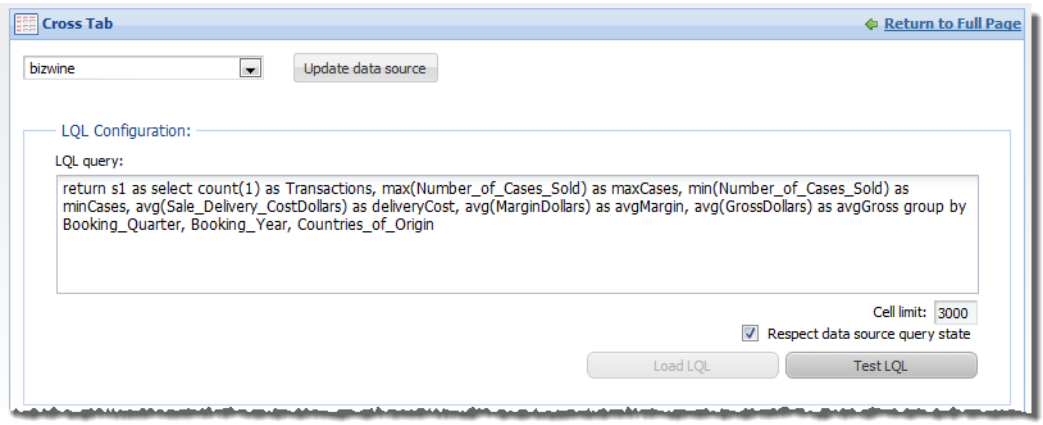
For a **Cross Tab** component, power users can select a different data source, and must provide an LQL query.

To configure the data source and LQL query for the component:

1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.

   Note that if you change the data source for an existing component, the rest of the component configuration, including the LQL query, is cleared.

2. In the **LQL query** field, enter an LQL query to return the cross-tab values.



Note: When entering your LQL query, remember that all attribute names, including the names of derived attributes, must be NCName-compliant. They cannot contain spaces or special characters.

When selecting the attributes to group by, make sure to select attributes that have a limited number of values. If you group by an attribute that has too many values, your **Cross Tab** table may become too large to be useful.

3. In the **Cell Limit** field, set the maximum number of cells to include in the **Cross Tab** results table.

   The default value is 3000.

   In general, you should not make this value much larger than the default. A table with many thousands of cells is time-consuming to render and difficult for end users to read.

4. To update the table automatically when end users refine the data, check the **Respect data source query state** box.

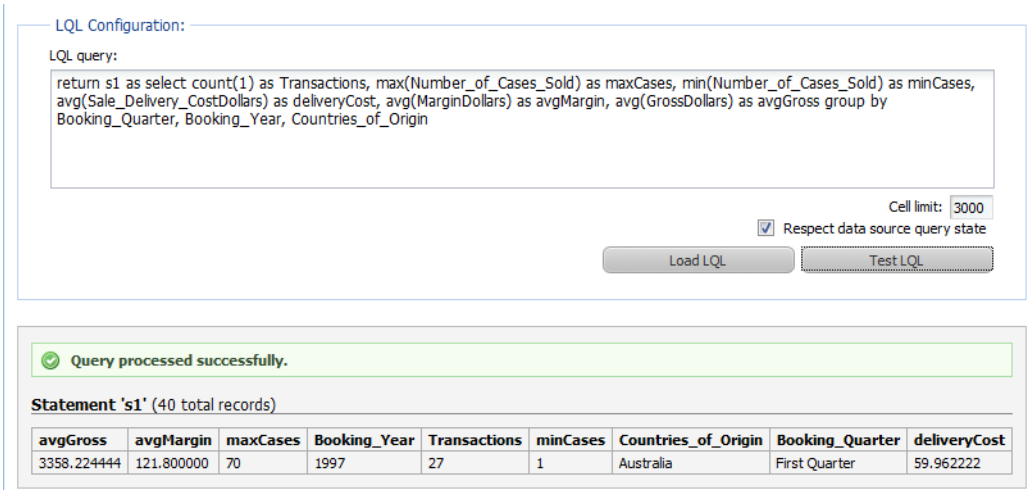
   The box is checked by default.

   When the box is unchecked, the component is independent of the data source query state, and the metric values remain constant.

5. To validate the LQL query, click the **Test LQL** button.

   If the LQL query is valid, then:

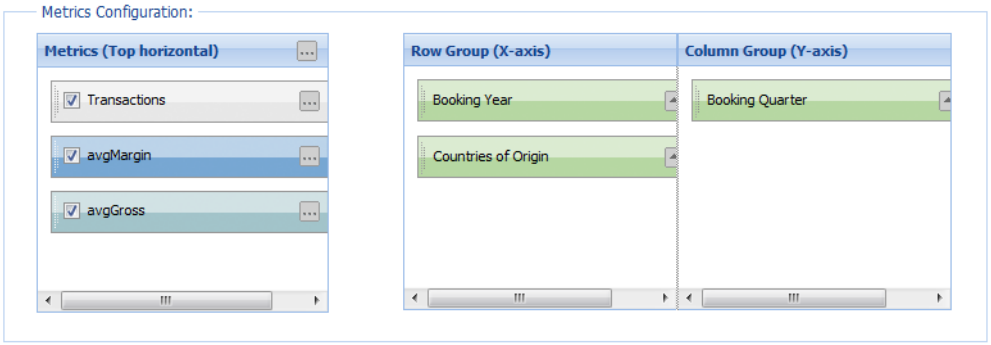
   - The **Load LQL** button is enabled.
   - A sample result is displayed.



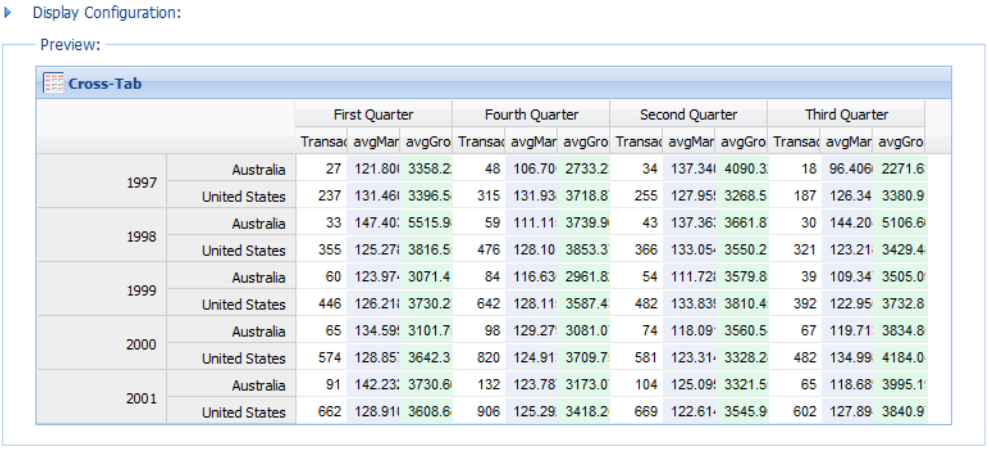
6. To load the LQL query, click the **Load LQL** button.

When you load the query:

- The **Metrics Configuration** section is populated with the metrics and grouping fields from the query. The section provides options to configure the display of the metrics and groupings.

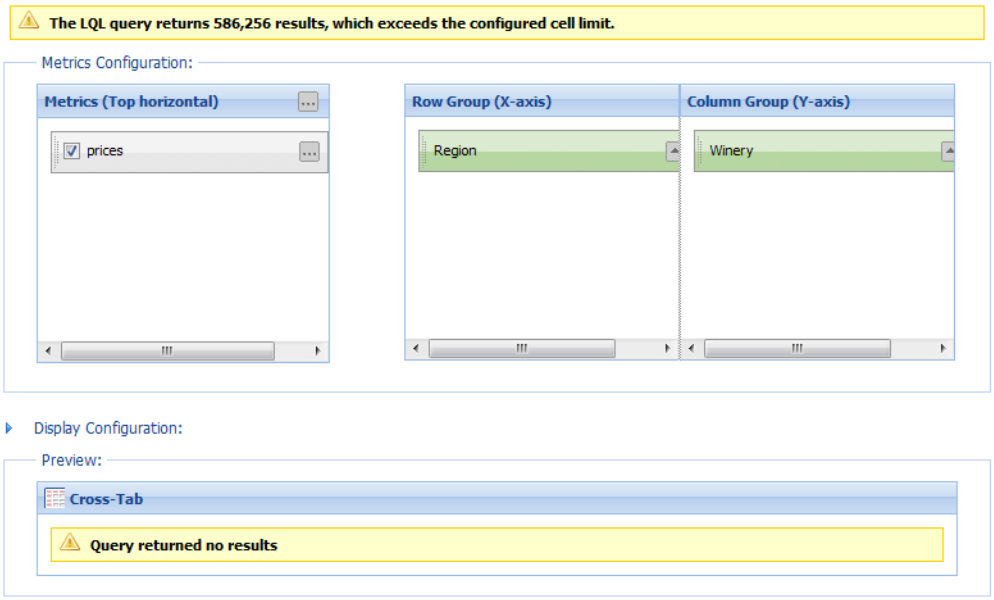


- The preview table at the bottom of the edit view shows how the component will display to end users.



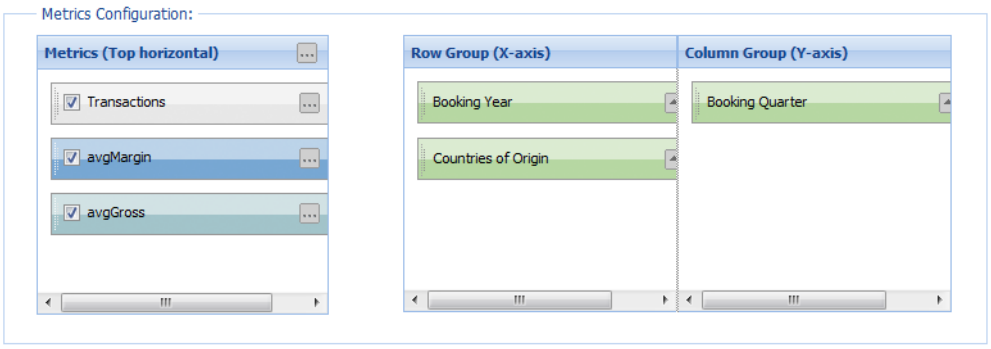
If you do not need to configure the table layout or dimensions, then to exit the edit view, click **Return to Full Page**.

If the number of results is larger than the value of **Cell limit**, then when you load the LQL query, the preview is not displayed. Error messages indicate that the results need to be filtered.



Configuring the Cross Tab table layout

The **Metrics Configuration** section of the **Cross Tab** component edit view allows power users to control the layout of the table and the format of the metrics values.



In the **Metrics Configuration** section:

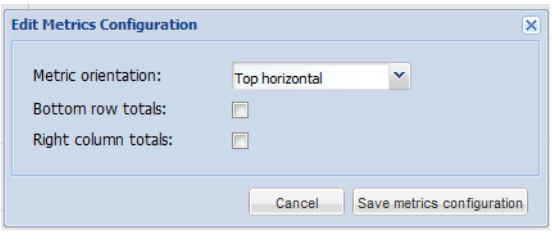
- The **Metrics** list contains the metrics from the LQL query. By default, the metrics labels are displayed in the bottom header row on the table.
- The **Row Group** list contains the group by fields that are displayed as the header columns to the left of the table.
- The **Column Group** list contains the group by fields that are displayed as the header rows above the metrics labels.

As you make changes to the configuration, to see their effects on the table, check the preview table at the bottom of the edit view.

To configure the layout of the **Cross Tab** table:

1. In the **Metrics** list, to configure the general display rules for the metrics:
   a) Click the **...** button in the top right corner.

      The **Edit Metrics Configuration** dialog is displayed.



   b) From the **Metric orientation** drop-down list, select where to display the metrics labels:
      - To display the metrics labels across the top of the table, select **Top horizontal**. This is the default value.
      - To display the metrics labels to the left of the table, select **Left vertical**.

   c) To add a row to the bottom of the table containing the totals for each column, check the **Bottom row totals** checkbox.

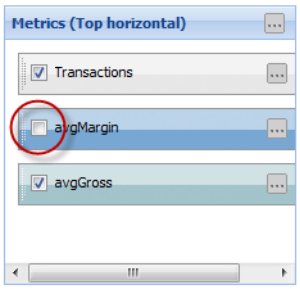
      This box is unchecked by default.

   d) To add a column to the right of the table containing the totals for each row, check the **Right column totals** check box.

      This box is unchecked by default.

Note: Totals are computed using the same aggregating function as the original metric. For example, metrics computed using the average function (AVG) display an overall average for the total. Because totals may be confusing in cases where the ARB function is used, you may not want to display totals in those cases.

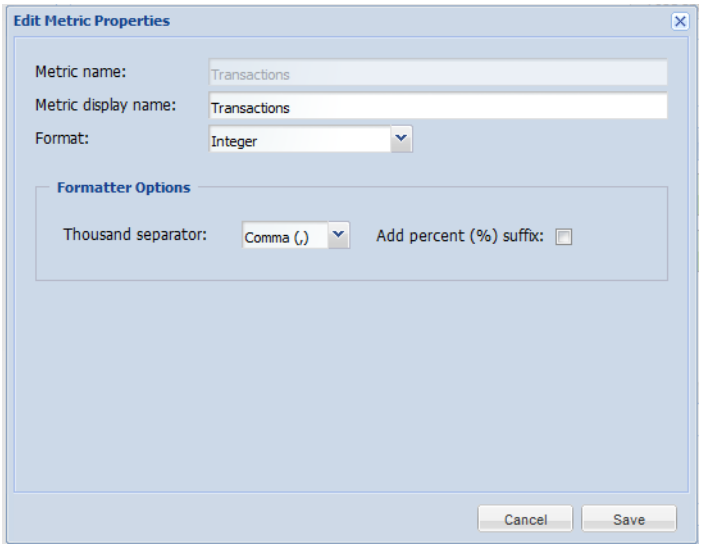
2. In the **Metrics** list, to determine the order in which to display the metrics, drag and drop the metrics within the list.
3. In the **Metrics** list, to remove a metric from the table, uncheck its checkbox.



   You would most likely use this option to hide a metric that you used to generate the metric you wanted to display. Otherwise, if you do not want to display a metric, you should remove it from the LQL query instead of unchecking the checkbox.

4. In the **Metrics** list, to configure the display of an individual metric:
   a) Click the . . . button on the metric label.

      The **Edit Metric Properties** dialog is displayed.



b) In the **Metric display name** field, type the label to display for the metric row or column heading.

c) From the **Format** drop-down list, select the format to use to display the metric.

When you select the format, the **Formatter Options** section is updated to display the available fields for that format.

The format options are:

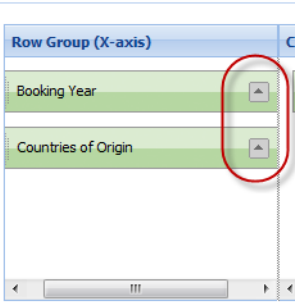
Integer	Displays the metrics value as an integer. This is the default value. For an integer value, you can configure: • The character to use for the thousands separator • Whether to display a percent sign after the value
Currency	Displays the metrics value as a currency value. For a currency value, you can configure: • The character to use for the thousands separator • The character to use for the decimal point • The number of decimal places to display • The currency symbol to display • Whether to display the currency symbol before the value (**prefix**) or after the value (**suffix**)
Decimal	Displays the metrics value as a decimal value. For a decimal value, you can configure: • The character to use for the thousands separator • Whether to display a percent sign after the value • The character to use for the decimal point • The number of decimal places to display

d) To save the changes, click **Save**.

5. In the **Row Group** and **Column Group** sections, to configure the group by values:
   a) To control where the group by value displays, drag it to the appropriate list:
      - To display it as a row heading, drag it to the **Row Group** list.
      - To display it as a column heading, drag it to the **Column Group** list.
   b) To determine the order in which the group by values display, drag the values within the list.

      In the **Row Group** list, the top value is in the leftmost heading column.

      In the **Column Group** list, the top value is in the top heading row.
   c) To control the order in which the values display, use the toggle next to the value name.

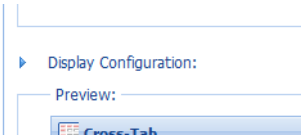


By default, the values display in ascending order. To display the values in descending order, click the icon.

Configuring the table dimensions for a Cross Tab component

The **Display Configuration** section of the **Cross Tab** component edit view allows you to more precisely control the dimensions (in pixels) of the table and its window.

When you first display the edit view, the fields are hidden. By default, the fields are empty, and the table is displayed based on the width of the layout column.



To configure specific window and table dimensions:

1. To display the fields, click the arrow icon to the left of the heading.
2. In the **Display Configuration** fields:

Window width (px)	In the field, type the width in pixels of the **Cross Tab** window. If there is no value in the field, then the table is adjusted for the best display for the width of the layout column.

	If you do provide a value, the metrics column widths are adjusted to fit within the specified width.
Window height (px)	In the field, type the height in pixels of the **Cross Tab** window. If there is no value in the field, then the table is as high as needed to display all of the rows. Note that if the height you specify is less than the height needed to display all of the rows, the table is truncated.
Table width (px)	In the field, type the width in pixels of the metrics area of the table. If there is no value in the field, then the table is adjusted for the best display based on the width of the window.
Left column header width (px)	In the field, type the width in pixels of the left column header area of the table. If there is no value in the field, then the table is adjusted for the best display based on the width of the window and table.

3. After configuring the dimensions, to save the changes, click **Save display configuration**.

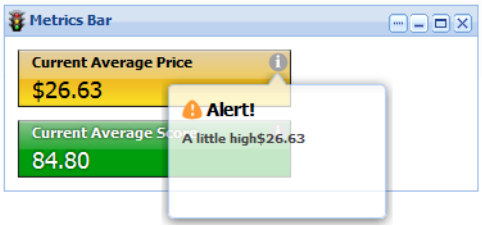
Metrics Bar

About the Metrics Bar component

The **Metrics Bar** component allows users to quickly view metrics that summarize various aspects of the underlying data.

The **Metrics Bar** component displays metric values that are based on LQL queries written by the power user.

The power user can change the display format of each returned metric value. They also can set up alerts to notify end users when a metric has reached a certain value or range of values:

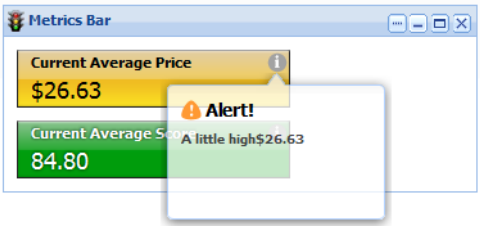


Using Metrics Bar

End users can view the **Metrics Bar** component and display additional information.

On each metric box, end users can view the metric values.

They also can mouse over the **i** symbol to view any alerts or threshold violation indicators.



Configuring a Metrics Bar component

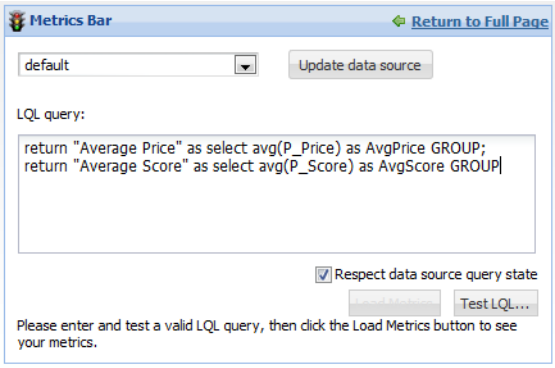
For a **Metrics Bar** component, power users configure the metrics to display, the format for each metric, and the alert messages to associate with the metric values.

Providing the LQL query for the Metrics Bar

The **Metrics Bar** component requires an LQL query in order to generate the metrics it uses.

To provide the LQL query for the **Metrics Bar** component:

1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.
2. In the **LQL query** field, enter your LQL query.



Note: When entering your LQL query, remember that all attribute names, including names of derived attributes, must be NCName-compliant. They cannot contain spaces or special characters.

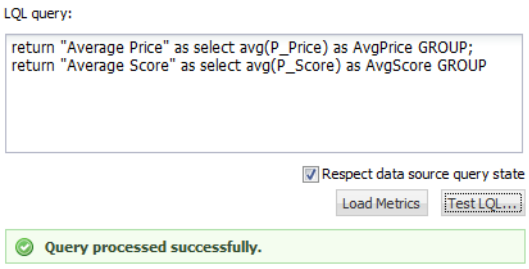
3. The **Respect data source query state** box determines whether the **Metrics Bar** values are for the currently displayed data only.

   If this box is checked, then when the data source query state changes, the metric values are updated. For example, if an end user uses the **Guided Navigation** component to refine the data, then the **Metrics Bar** values are updated to reflect the currently displayed data only.

   If the box is unchecked, the **Metrics Bar** component is independent of the data source query state. The metric values remain constant regardless of the data source query state.

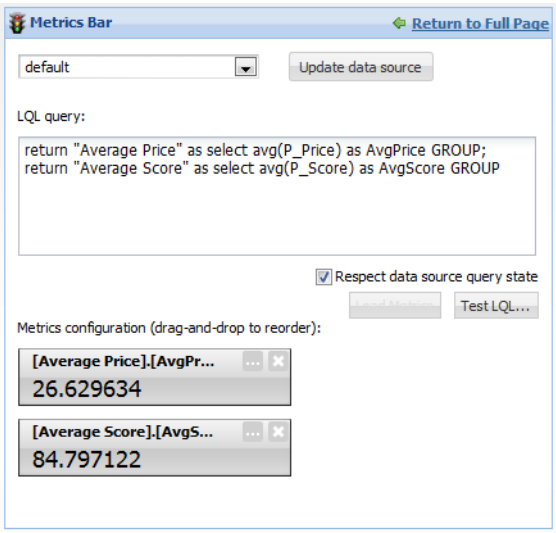
4. To validate the LQL query, click the **Test LQL query** button.

   If the LQL query is valid, the **Load Metrics** button is enabled.



5. Click **Load Metrics**.

   For each metric in the LQL query, a metric box is added to the edit view.



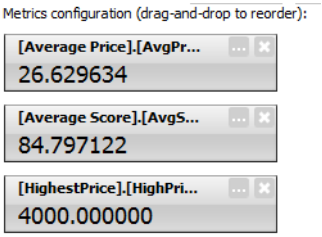
After you select your metrics, you can:

- Hide metrics that you do not want to display
- Configure the format for each displayed metric value
- Configure alert messages based on the metric values

Hiding and showing metrics on the Metrics Bar

You can hide metrics that you do not want to display on the **Metrics Bar**, and also restore them if needed.

When you first load the LQL query for the **Metrics Bar** component, a metric box is created for each metric from the query.



However, you may not want to display all of these metrics. For example, some of the metrics may simply be used to generate the metric you actually want to display.

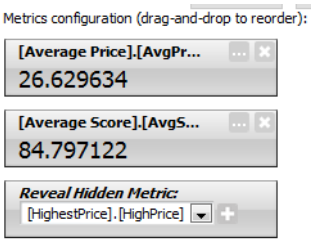
To hide and restore metrics on the **Metrics Bar**:

1. To hide a metric, click the x icon in top right corner of the metric box.

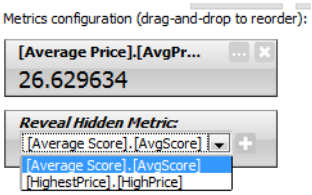
   The metric box for that metric is removed.

   If this is the first metric you are hiding, a box is added for the hidden values.

   The metric is added to the **Reveal Hidden Metric** drop-down list on the hidden metric values box.



2. To restore a hidden metric:
   a) From the **Reveal Hidden Metric** drop-down list, select the metric to restore.



   b) Click the **+** icon.

      The metric is removed from the **Reveal Hidden Metric** drop-down list.

      The metric box is added for that metric.

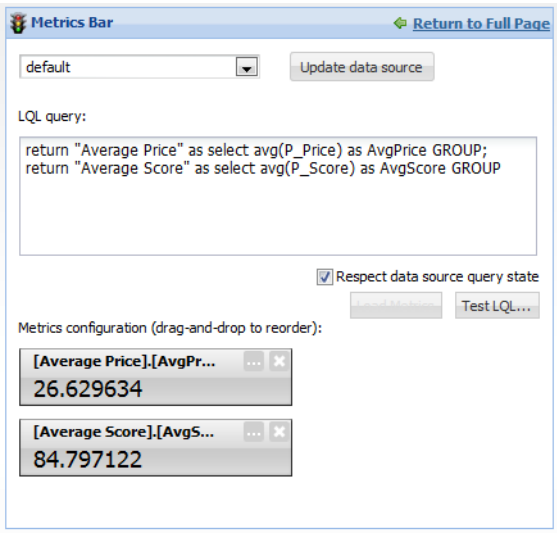
      If there are no other hidden metrics, then the hidden metric values box is removed.

Configuring the metric boxes for a Metrics Bar

After you select the metrics to display for the **Metrics Bar**, you then configure how to display each metric.

On the edit view of the **Metrics Bar** component, each metric to display is represented by a metrics box. When you first add the metrics:

- The metrics boxes are grey.
- Each box label is based on the LQL query.
- There are no alerts associated with the metrics.

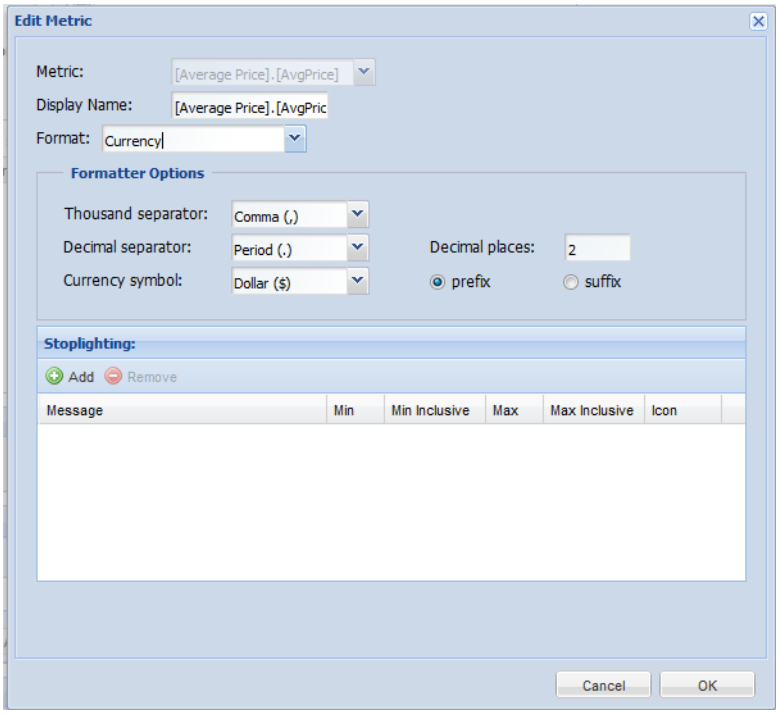


To configure the order in which to display the metrics, drag and drop the metric boxes.

For each metric, to configure how the value and box are displayed:

1. On the metrics box, click the **...** button.

   The **Edit Metric** dialog is displayed.



2. In the **Display Name** field, type the display name for the metric.

   This is the label that displays at the top of the metric box.

   By default, the display name is based on the LQL query.

3. From the **Format** drop-down list, select the format to use to display the metric value.

When you select a format, the **Formatting Options** section is updated to contain the fields for configuring the selected format.

The options are:

Integer	For integer values, you can configure: • The character to use to separate the thousands • Whether to display a percent sign (%) after the value.
Currency	For currency values, you can configure: • The character to use to separate the thousands • The character to use for the decimal • The number of decimal places to display • The currency symbol to display • Whether to display the currency symbol before the value (**prefix**) or after the value (**suffix**)
Decimal	For decimal values, you can configure: • The character to use to separate the thousands • Whether to display a percent sign (%) after the value. • The character to use for the decimal • The number of decimal places to display
String	For string values, you can configure whether to change the capitalization of the text. The options are to: • Keep the string as it is retrieved from the database • Convert the entire string to lower case • Convert the entire string to upper case • Convert the string to title case (first word of each letter is capitalized)

4. For numeric values (integer, currency, and decimal values), you can configure alert messages to display based on the current value of the metric. String values cannot have alert messages.

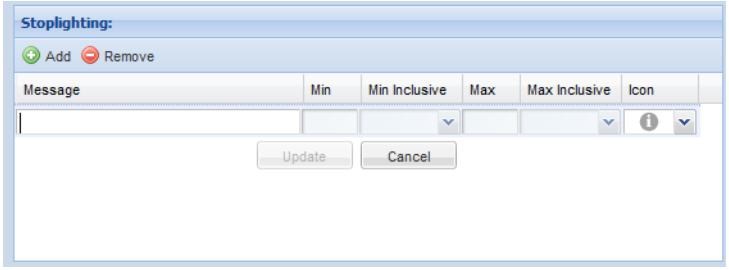
   The alert messages are displayed when the end user hovers the mouse over the **i** icon on the metric box.

   As part of this configuration, you also determine the color of the metric box.

   In the **Stoplighting** section, to configure the alert messages:

   a) To add an alert message, click **Add**.

      The next row of fields is enabled.

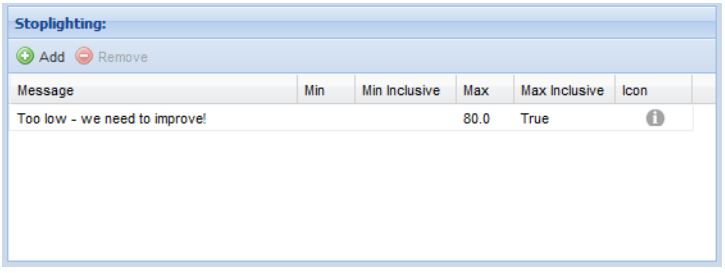


b) To configure the message:

Message	In the **Message** field, type the text of the message to display.
Min	In the **Min** field, type the minimum metric value for which to display this message. For example, to only display the message if the value is greater than 85, type 85 in the field.
Min Inclusive	If you have provided a minimum value, then from the **Min Inclusive** drop-down list, select whether the provided value is inclusive. If you select **True**, then the alert message displays if the value is greater than or equal to the provided value. If you select **False**, then the alert message displays only if the value is greater than the provided value.
Max	In the **Max** field, type the maximum metric value for which to display the message. For example, to only display the message if the value is less than 100, type 100 in the field.
Max Inclusive	If you have provided a maximum value, then from the **Max Inclusive** drop-down list, select whether the provided value is inclusive. If you select **True**, then the alert message displays if the value is less than or equal to the provided value. If you select **False**, then the alert message displays only if the value is less than the provided value.
Icon	From the **Icon** drop-down list, select the icon representing the color of the metric box when the metric value meets the minimum/maximum value criteria.

c) After filling out the fields, to save the alert message, click **Update**.

The alert message is added to the list.



d) To edit an existing alert message, double-click the message row. The fields are enabled to allow you to edit the message.

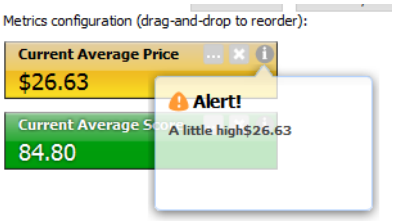
After making the edits, to save the changes, click **Update**.

e) To remove an alert message, click the message, then click **Remove**.

5. To save the metric display configuration, click **Save**.

   On the edit view, the metric box is updated to reflect the new display name, and to use the appropriate color based on the metric value.

   To test the alert message, hover the mouse over the **i** icon.



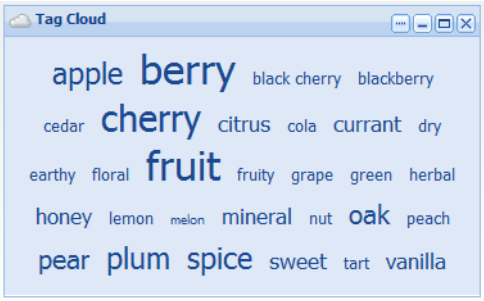
6. To exit the edit view, and return to the end user view, click **Return to Full Page**.

Tag Cloud

About the Tag Cloud component

The **Tag Cloud** component provides a visual representation of the frequency or relevance of text values for a selected attribute within the current data set.

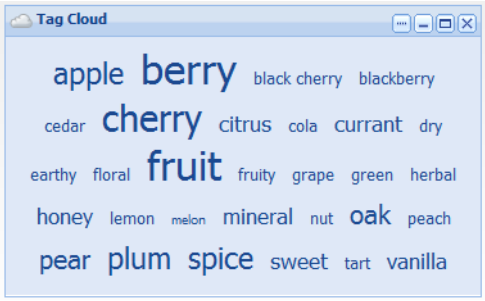
On the component, the font size indicates the relative significance of each value. For multi-select attributes, end users also may be able to refine the data by a selected value.



To provide a meaningful summary, the **Tag Cloud** component must be associated with an attribute that consists of numerous textual attribute values.

Using Tag Cloud

On the **Tag Cloud** component, end users can see the distribution of values for a selected attribute within the current data set. Terms in a larger font occur more frequently or are more relevant.

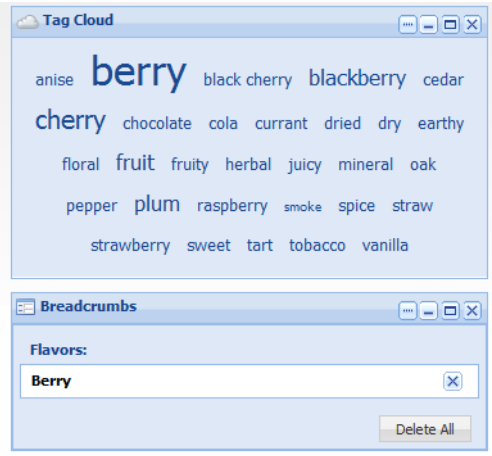


Note: For hierarchical attributes, the tag cloud only shows the frequency of the actual displayed value, not of any child values in the hierarchy.

For example, for a Wine Type attribute, the value Red may have the children Bordeaux and Merlot. On the Tag Cloud, if the value Red is displayed, it only reflects records that have the value of Red. It does not include records tagged with Bordeaux or Merlot.

If the values are for a multi-select attribute, then end users may be able to use the values to refine the data. If the component supports refinement, then when end users click an attribute value:

- The data is refined to only include records with that attribute value.
- The selected attribute value is added to the **Breadcrumbs** component.
- The tag cloud is updated to reflect the distribution of the attribute values in the matching records.

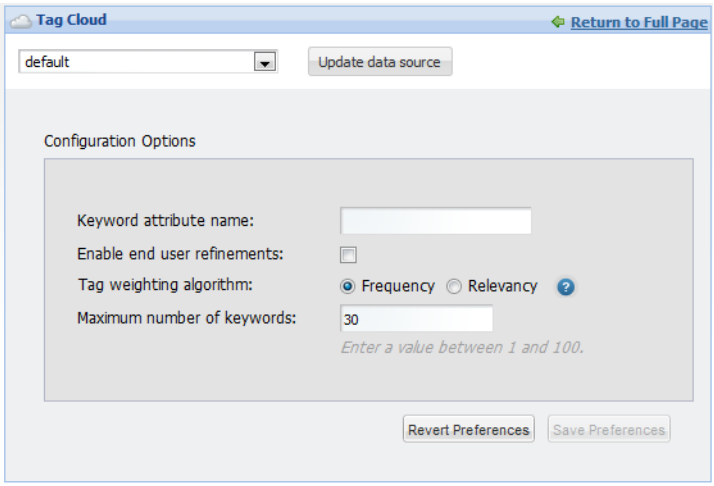


Configuring a Tag Cloud component

For a **Tag Cloud** component, power users select the data source and attribute to use. For multi-select attributes, they can also enable the refinement option.

To configure a **Tag Cloud** component:

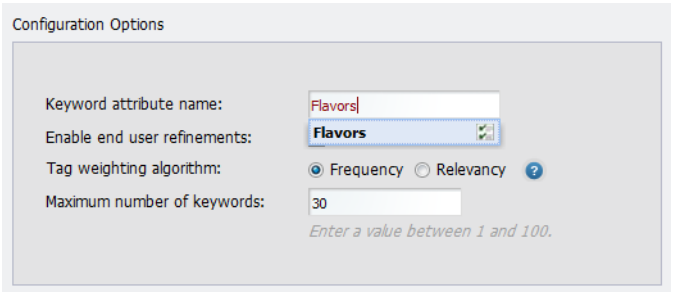
1. On the component edit view, to bind a different data source to the component, select the data source from the drop-down list, then click **Update data source**.



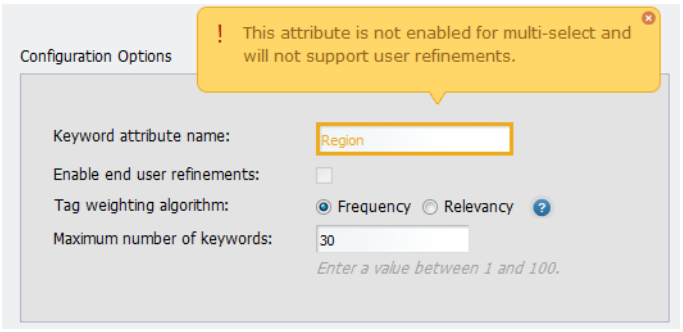
2. In the **Keyword attribute name** field, enter the name of the attribute to use.

   As you type, the matching attribute names are displayed.

   In order to support refinement, the attribute must be enabled for multi-select. If you enter a multi-select attribute, then in the type-ahead area, Latitude Studio shows a double-check mark icon next to the attribute name.



   If you enter an attribute that is not enabled for multi-select, then when you try to save the configuration, a warning message is displayed.



   For non-multi-select attributes, the tag cloud still shows the distribution of text values, but end users cannot use those values to refine the data.

3. If the attribute allows multi-select, then the **Enable end user refinements** box is checked, indicating that end users can use the attribute values to refine the data.

   To not allow the values to be used for refinement, uncheck the box.

   If the attribute is not enabled for multi-select, then you cannot check the checkbox.

4. Under **Tag weighting algorithm**, click the radio button next to the option to use to determine the significance of a tag.

   The options are:

Frequency	If you select this option is selected, then the tags are sized to represent the number of records associated with the related attribute value. The terms associated with the largest number of records are displayed in the largest font size.
Relevancy	If this option is selected, then the tags are sized to represent the number of records associated with the tag. They then are weighted relative to their uniqueness in the matching record set. Terms that appear throughout the index are less relevant, and appear smaller, than those that appear only in the matching records. This strategy is recommended if certain terms appear on most records in the index, to ensure that these common terms are displayed less prominently in the tag cloud.

5. In the **Maximum number of keywords** field, specify the maximum number of attribute values to display.

   The default is 30.

6. To save your changes, click **Save Preferences**.
7. To exit the edit view, click **Return to Full Page**.

Chapter 22

Personalization Components

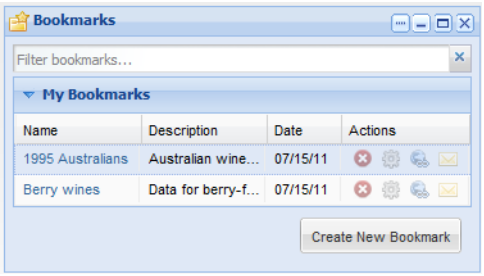
Personalization components allow end users to customize their Latitude Studio application.

Bookmarks

About the Bookmarks component

Bookmarks allows end users to save a given navigation and component state so that they can share it or return to it at a later time.

Once a bookmark has been created, end users can click it to return to that state. Bookmarks are listed in the order they were created, along with an accompanying description and date.



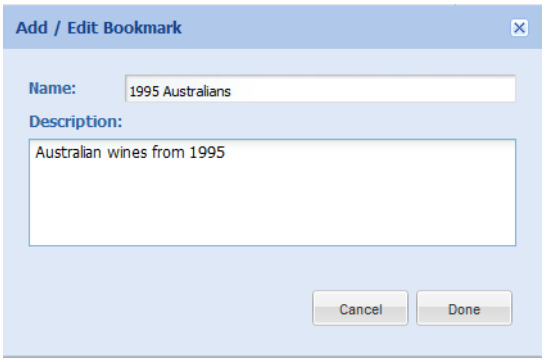
Note: The **Bookmarks** component does not require a backing data source. It stores and retrieves data from the underlying Liferay database.

Using Bookmarks

The **Bookmarks** component allows the end user to create, edit, delete, and share bookmarks.

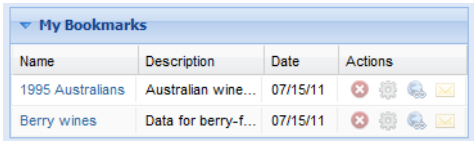
To add a new bookmark:

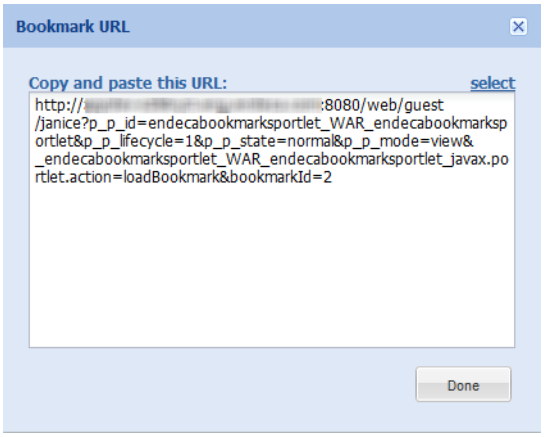
1. Click **Create new bookmark**.
2. On the **Add / Edit Bookmark** dialog box, fill in the **Name** and **Description**.

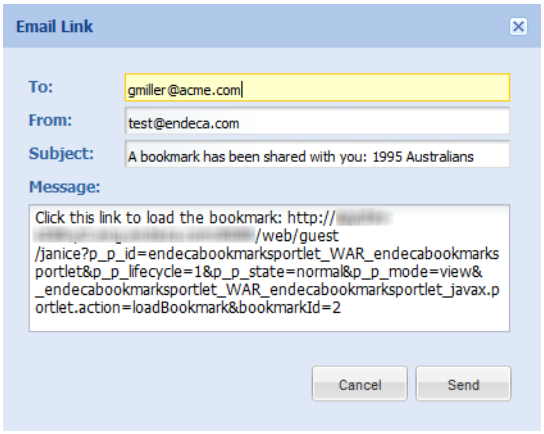


3. Click **Done**.

For each bookmark, the following actions are available:



	Delete	Removes the selected bookmark from the list.
	Edit	Displays the **Add / Edit Bookmark** dialog box, from which you can change the bookmark name or description.
	Create link	Captures and copies the URL of a bookmark. When you click the icon, the **Bookmark URL** dialog box is displayed:  To select the URL in order to copy and base it, click the **select** link.
	Email	Used to email the URL of a bookmark to another user. When you click the icon, the **Email link** dialog box is displayed, pre-populated with everything except the **To:** field:



After filling in the recipient information, to send the URL, click **Send**.

Note: In order to use the email feature, the Latitude Studio administrator must set up the email server, as described in the *Latitude Administrator's Guide*.

Bookmark data saved for each Latitude component

For each Latitude component, the **Bookmarks** component saves the following information about the component state.

There is no bookmark data for **Control Panel** components.

Component	Persisted States	Comments
Alerts	Expanded/collapsed view of alert groups	For the **Alerts** component, the bookmark saves whether the list of alert groups is displayed. If the alert groups are displayed, then for each alert group, the bookmark also saves whether the list of alerts is expanded.
Bookmark	None	Bookmarks do not save any information about the **Bookmarks** component itself.
Breadcrumbs	Expanded multi-select managed attributes	When users select multiple values for an attribute, the **Breadcrumbs** component may group those values in a collapse/expand control. The bookmarks stores whether each collapse/expand control is expanded or collapsed.
Chart	Metric, x-axis, and cross tab drop-down lists	Power users can configure parametric charts, where they provide multiple options for metrics, x-axis, and cross tab. The user's selections from the **Metric**, **Group By (X-Axis)**, and **Cross Tab** drop-down lists are persisted with a bookmark.

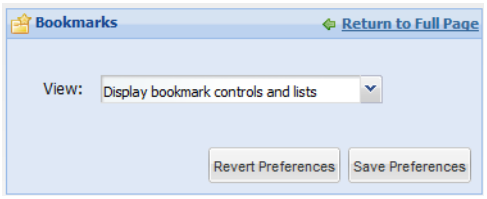
Component	Persisted States	Comments
Compare	**Attributes:** persists the order of attributes and the expand and collapse state of attribute groups. **Records:** persists the order of records and their highlighted state	
Cross Tab	None	
Data Explorer	None	
Guided Navigation	Expanded attributes and collapsed attribute groups	If an attribute is expanded to display the available values, then it is expanded when the bookmark is loaded. Attribute groups are expanded by default. If a group is collapsed when a bookmark is saved, then it also is collapsed when the bookmark is loaded.
Metrics Bar	None	
Range Filters	None	
Record Details	None	
Results List	None	
Results Table	Sort state and number of records per page	If you have sorted on a column and then create a bookmark, the sort is applied when the bookmark is loaded. If you have selected the number of records per page and then create a bookmark, the results table displays that number of records per page when you load the bookmark.
Search Box	None	
Tabbed Component Container	The tab in focus	If you create a bookmark when on a tab other than the first one, that tab is reloaded properly when the bookmark is loaded.
Tag Cloud	None	

Configuring a Bookmarks component

For a **Bookmarks** component, power users select the display view, which controls whether end users can create bookmarks and view the list of bookmarks.

To configure a **Bookmarks** component:

1. On the component edit view, from the **View** drop-down list, select the display mode.



The options are:

Display Bookmark Controls only	End users can create new bookmarks, but cannot see the list of saved bookmarks.
Display Bookmark Lists only	End users can display the list of existing bookmarks, but cannot create a new bookmark.
Display Bookmark Controls and Lists	End users can both create new bookmarks and display the list of existing bookmarks. This is the default.

2. To save your changes, click **Save**.
3. To exit the edit view, click **Return to Full Page**.

Index

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W