QlikView 11 Feature Sheet
Comparative Analysis

What is included?

Over the last couple of QlikView releases we've introduced some new ways to expand how you analyze and search your data to lead you to enhanced business discovery. Set Analysis and Associative Search are just two of the features introduced that expand the insights you gain with your data. With QlikView 11 we are delivering more features that further QlikView's advanced analytic capabilities.

With the new comparative analysis feature in QlikView 11, developers can create a QlikView app that has graphs, tables, or sheets that are based on different selection sets. Business users can then compare multiple selections of data inside one document, one chart, or one object, to yield new insights into patterns of use, opportunities and threats, and relative performance.

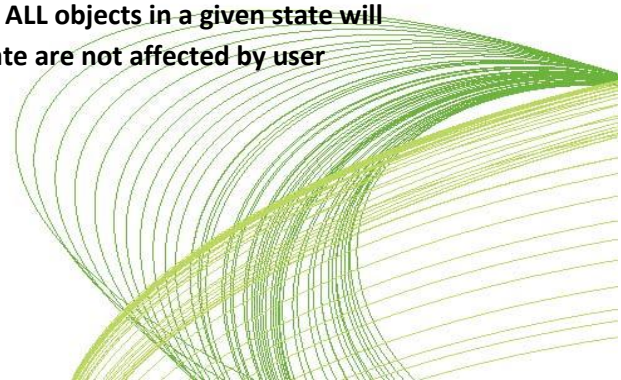
What Does This Feature Bring to QlikView?

When analyzing information in QlikView, business users can derive new insights, and achieve faster time to insight, with the ability define two or more different data sets and compare them side by side to identify issues and opportunities in the data.

QlikView's central associative analysis works by allowing users to make selections against their data and to see not only the classic value of the various KPI for the selected values, but also to gain insight into what is related (and not related) to their selections. These insights allow simple comparison of inside/outside viewpoints. But a common need within QlikView applications is to compare and contrast selections of like data – baskets of products, groups of customers/suppliers/users. Today's QlikView developer can achieve this but typically at a cost – duplicating data models, extensive macro code, complex user interactions. Comparative analysis brings a native way for QlikViews to compare like data sets, setting up the possibility of complex side by side exploration, and ad hoc element groupings.

How does it work?

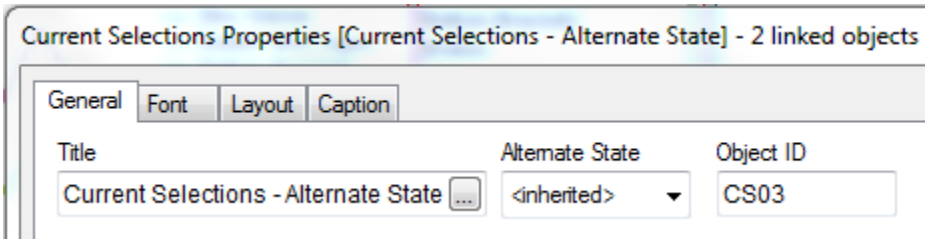
QlikView has allowed users to detach an object from the underlying data model for quite some time. When an object is detached that object no longer responds to user selections. The user can then re-attach the object and it instantly updates to reflect the current state of the selections. Comparative analysis is similar to this. The QlikView developer can create multiple 'alternate states' within a QlikView document and then apply these states to specific objects within the document. The end-user can create copies of these objects (server objects) and then put those objects into different states. **The most important difference between detach and an alternate state is that ALL objects in a given state will respond to user selections made in that state. Objects in a given state are not affected by user selections in the other states.**



How do I configure it?

Comparative analysis is enabled by a QlikView developer when the developer creates at least one Alternate State from the General tab of the Document Properties dialog box. The QlikView developer can create any number of Alternate States within a QlikView document. The QlikView developer provides a name for each state created. These names are then referenced by screen objects or chart expressions. There are two special state names: default and inherited. The usage of these two states is described below.

Screen objects can be placed into specific states by using the General tab of the object's property dialog. After alternate states functionality is enabled in the QlikView document the QlikView developer will be able to assign states to various objects. See the screen shot below.



There are two states that are always available; <default state> and <inherited>. The <default state> represents how QlikView has always operated. This is the state where most QlikView usage occurs and will occur. This state is represented by '$'. Objects can inherit states from higher level objects. This becomes important with the use of container objects as the overall container object can be set to a state and the objects in the container can inherit the container's state or be in different states from the container. Additionally, the document and sheets can be put into alternate states. This means that states are inherited as such: Document → Sheet →Sheet Objects. The document, sheets and all screen objects are in the <inherited> state unless overridden by the QlikView developer.

Business Cases

- Group Analysis
- Ad Hoc Entity Groupings
- Brushing

Upgrade Considerations

This is a new feature that is being introduced with QlikView 11. All existing set, aggregation functions, operations such as set reference, detach/attach built in earlier versions of QlikView, will continue to work without modification in QlikView 11. Implementations that have modeled the capabilities of Comparative analysis using macros, bookmarks, extended data models etc should consider the ability to

rework their implementation using alternate states. However it is not required to remove/change this code simply to move your documents ‘as is’ to QlikView 11.