What's New

Axiom Software

Version 2018.1

KaufmanHall

axiom software

Contents

Introduction

Kaufman Hall is pleased to announce the release of **Axiom Software Version 2018.1**. In this release, we introduce a new, streamlined way to report on data in Axiom Software: the Report Builder for web reports, a fully web-enabled reporting solution. This release also continues to grow our flexible toolset for Axiom forms, by adding new components for reporting data and enhanced styling options.

Enhancements in this release include:

- The new Report Builder for web reports introduces a new paradigm for reporting:
  - Fully web-enabled: create, edit, and view reports all in the browser
  - Easier and more streamlined process of report creation requires less training and no spreadsheets
  - Display Axiom data using a powerful data grid with built-in features such as grouping, filtering, drilling, and sorting
  - Display key performance indicators using the data-driven KPI Panel component
  - Use refresh variables to allow users to filter data on-the-fly
- Axiom forms have been enhanced with several new features to support robust data display and input:
  - Data Grid component to directly and efficiently query data into the form, and provide built-in grid features like sorting and filtering
  - Data-driven KPI Panel component to easily display and format key performance indicators
  - New styling approach for Formatted Grid components, allowing precise control over formatting for each row and column
  - New lightweight auto-submit option for Formatted Grid components, to enable updating formulas in the grid for user inputs without triggering a full form update
- A new kind of reference table—the KPI table—can be easily created in the Web Client Table Manager and used to provide data to KPI Panel components
- A new set of filtering enhancements includes the ability to save filters for easy reuse in reports and other areas, a new refresh variable to create and apply advanced filter statements, and a new Axiom function to launch the Filter Wizard from a cell in the spreadsheet

This *What's New* document provides information on all new features and enhancements in this release. Reviewing this document should give you a basic understanding of how these new features work, and what benefits they may provide to your organization. For full details on any new feature, please see the Axiom Software Help files or the PDF guides.

IMPORTANT: Before upgrading to version 2018.1, make sure you have reviewed the separate *Release Notes* document to understand any important technical changes and upgrade considerations in this release.

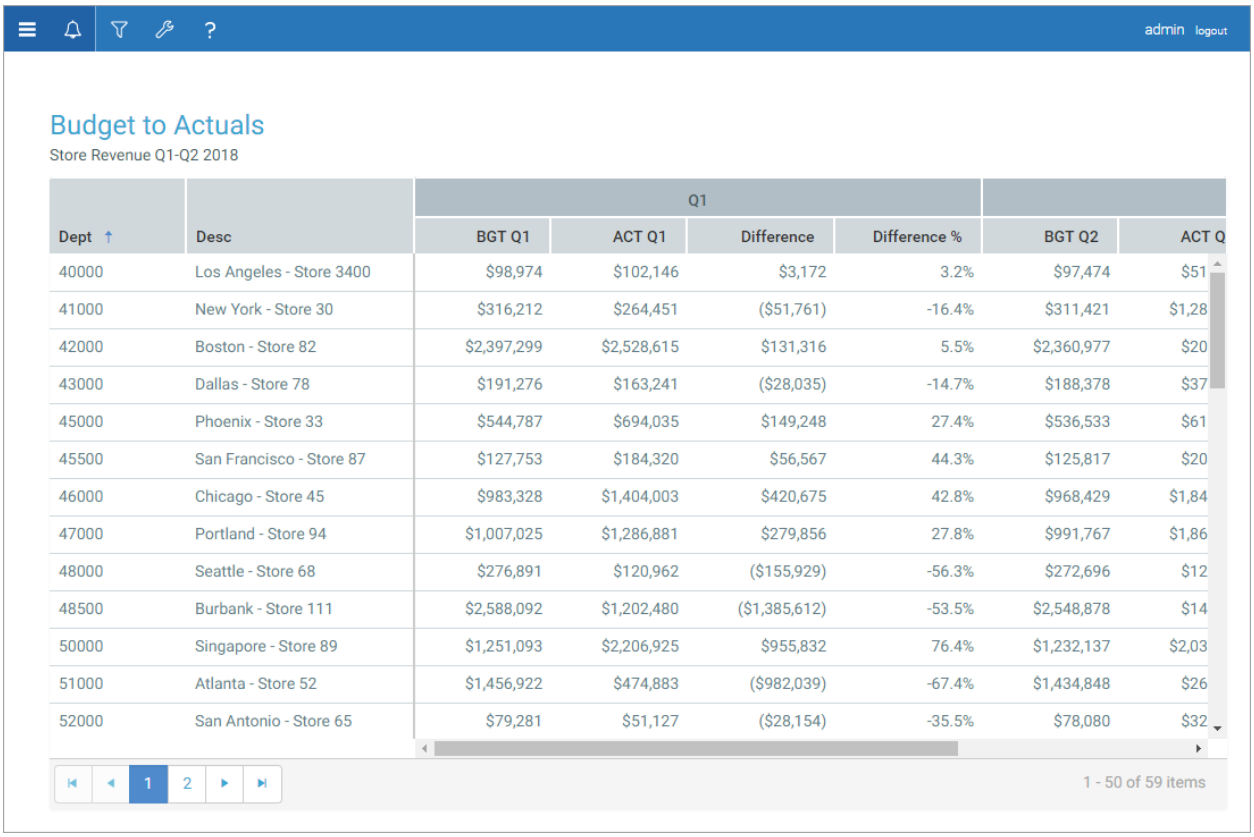
Report Builder for Web Reports

Version 2018.1 introduces a new way to report on data in Axiom Software—the Report Builder, a fully web-enabled reporting option. The Report Builder streamlines and simplifies the report creation process, so that reports can be created more quickly and require less training and technical requirements.

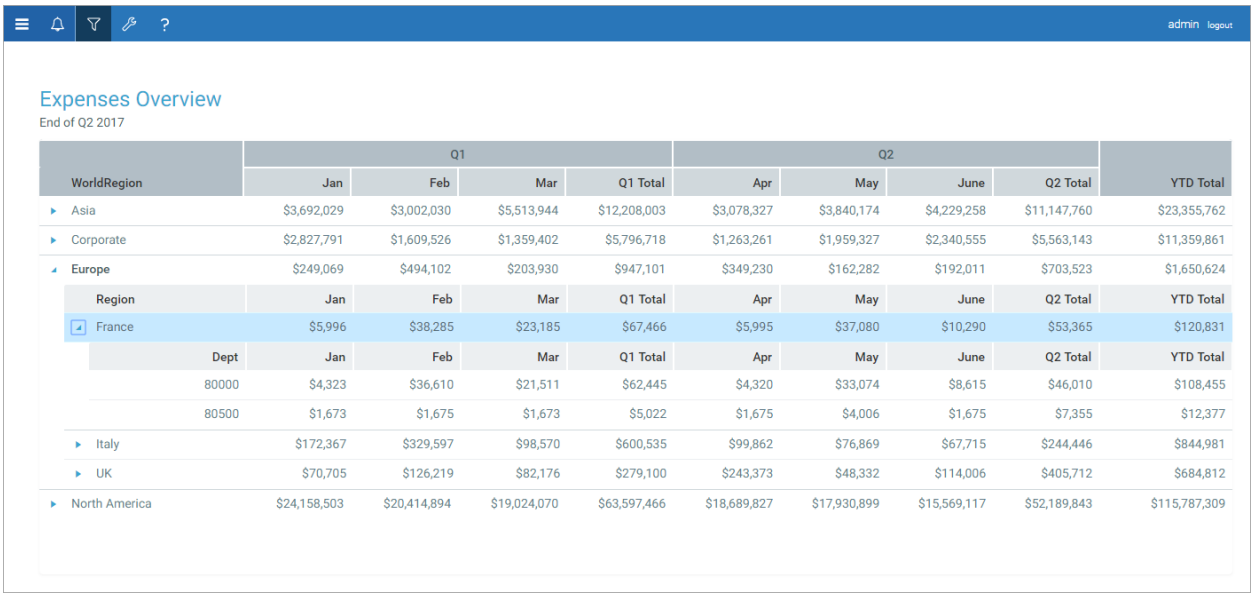
Using the Report Builder, web reports can be created and viewed entirely in the Web Client browser, without requiring any spreadsheet design and without requiring the Desktop Client (Excel or Windows). This greatly expands the available environments for report creation, since the only requirement is a supported browser.

This release is "phase one" of the Report Builder, providing the basic web report framework and functionality. We plan to expand this feature in future releases to include more report creation tools, enhanced reporting features, and continued improvements to ease-of-use.

In phase one, web reports can contain title text, hyperlinks, data grids, and KPIs. Using the default templates, web reports look similar to the following examples:

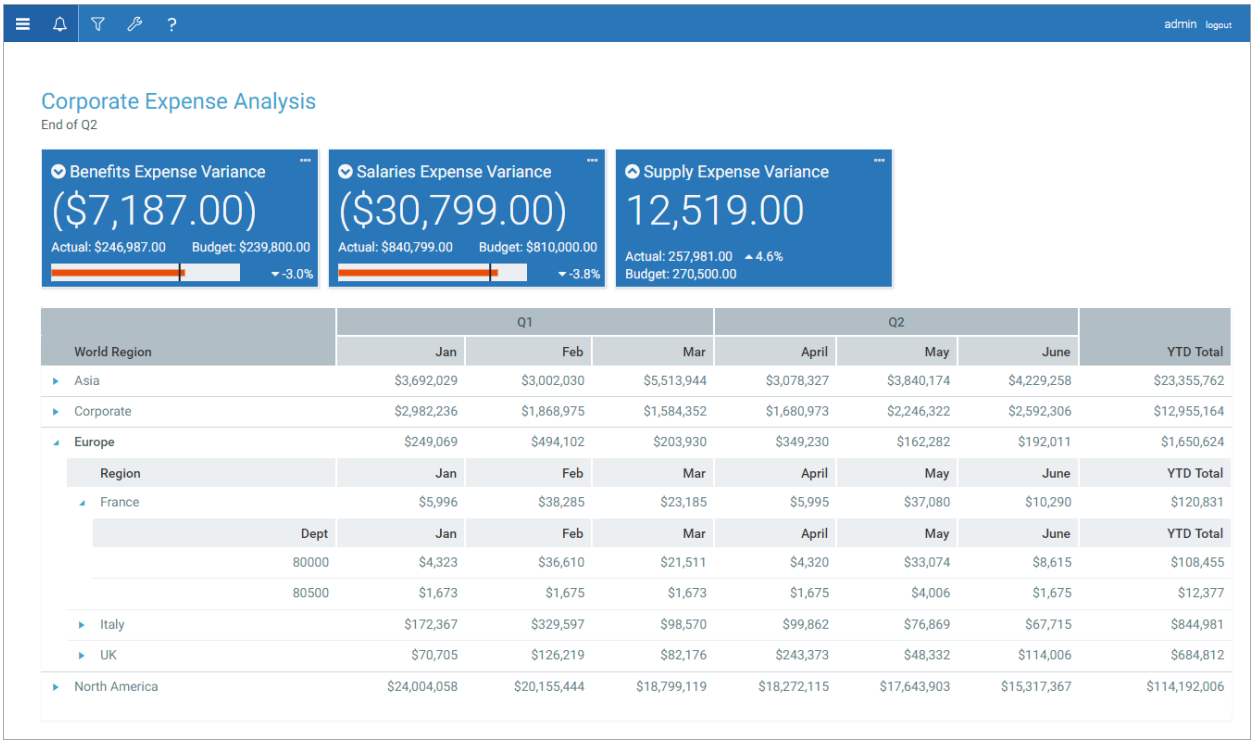


Example report with ungrouped data grid



	Q1				Q2				
WorldRegion	Jan	Feb	Mar	Q1 Total	Apr	May	June	Q2 Total	YTD Total
Asia	$3,692,029	$3,002,030	$5,513,944	$12,208,003	$3,078,327	$3,840,174	$4,229,258	$11,147,760	$23,355,762
Corporate	$2,827,791	$1,609,526	$1,359,402	$5,796,718	$1,263,261	$1,959,327	$2,340,555	$5,563,143	$11,359,861
Europe	$249,069	$494,102	$203,930	$947,101	$349,230	$162,282	$192,011	$703,523	$1,650,624
Region	Jan	Feb	Mar	Q1 Total	Apr	May	June	Q2 Total	YTD Total
France	$5,996	$38,285	$23,185	$67,466	$5,995	$37,080	$10,290	$53,365	$120,831
Dept	Jan	Feb	Mar	Q1 Total	Apr	May	June	Q2 Total	YTD Total
80000	$4,323	$36,610	$21,511	$62,445	$4,320	$33,074	$8,615	$46,010	$108,455
80500	$1,673	$1,675	$1,673	$5,022	$1,675	$4,006	$1,675	$7,355	$12,377
Italy	$172,367	$329,597	$98,570	$600,535	$99,862	$76,869	$67,715	$244,446	$844,981
UK	$70,705	$126,219	$82,176	$279,100	$243,373	$48,332	$114,006	$405,712	$684,812
North America	$24,158,503	$20,414,894	$19,024,070	$63,597,466	$18,689,827	$17,930,899	$15,569,117	$52,189,843	$115,787,309

Example report with grouped data grid



	Q1			Q2			
World Region	Jan	Feb	Mar	April	May	June	YTD Total
Asia	$3,692,029	$3,002,030	$5,513,944	$3,078,327	$3,840,174	$4,229,258	$23,355,762
Corporate	$2,982,236	$1,868,975	$1,584,352	$1,680,973	$2,246,322	$2,592,306	$12,955,164
Europe	$249,069	$494,102	$203,930	$349,230	$162,282	$192,011	$1,650,624
Region	Jan	Feb	Mar	April	May	June	YTD Total
France	$5,996	$38,285	$23,185	$5,995	$37,080	$10,290	$120,831
Dept	Jan	Feb	Mar	April	May	June	YTD Total
80000	$4,323	$36,610	$21,511	$4,320	$33,074	$8,615	$108,455
80500	$1,673	$1,675	$1,673	$1,675	$4,006	$1,675	$12,377
Italy	$172,367	$329,597	$98,570	$99,862	$76,869	$67,715	$844,981
UK	$70,705	$126,219	$82,176	$243,373	$48,332	$114,006	$684,812
North America	$24,004,058	$20,155,444	$18,799,119	$18,272,115	$17,643,903	$15,317,367	$114,192,006

Example report with KPIs and grouped data grid

Web reports are similar to Axiom forms and use the same components as building blocks, but are created and managed differently:

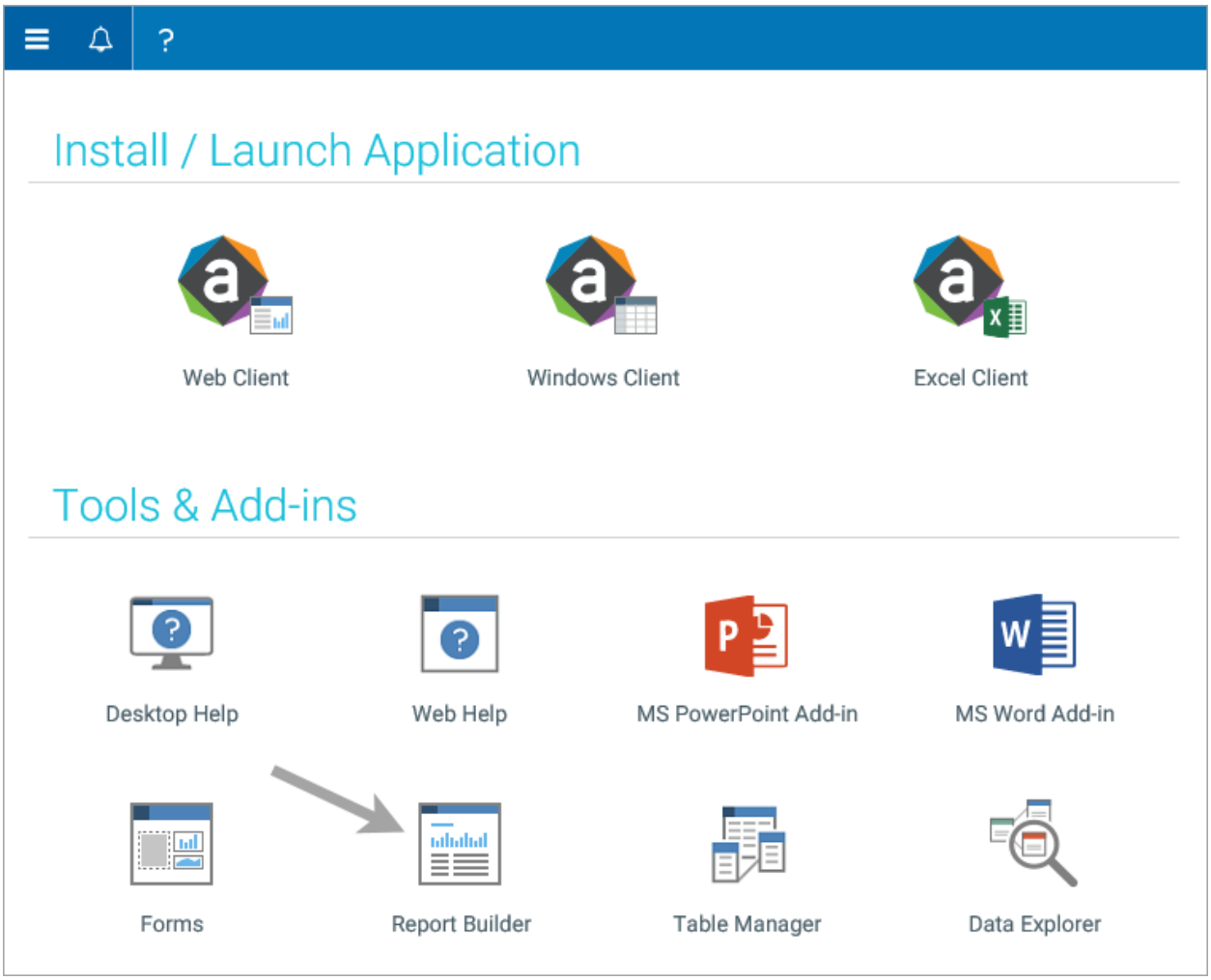
- Axiom forms are created in the Desktop Client (Excel or Windows), using a source spreadsheet file as the basis for querying data and designing the form web page. Once the form has been created,

end users can access it using just a browser—the Desktop Client is no longer required. The settings in the spreadsheet are used to render the form web page. Forms are very flexible and can support a wide variety of needs, but the setup and design requires a high level of Axiom Software knowledge and can be complicated.

- Web reports are created in the Web Client browser, with no spreadsheet dependencies. The report setup is saved as an XML file, which is then used to render the report web page. Web reports support fewer options than forms, but are easier to create (though a basic knowledge of the system's data structures is still required).

Create and view web reports

The Axiom Web Client has a new area dedicated to creating and viewing web reports. You can access this area using the Report Builder icon on the launch page:



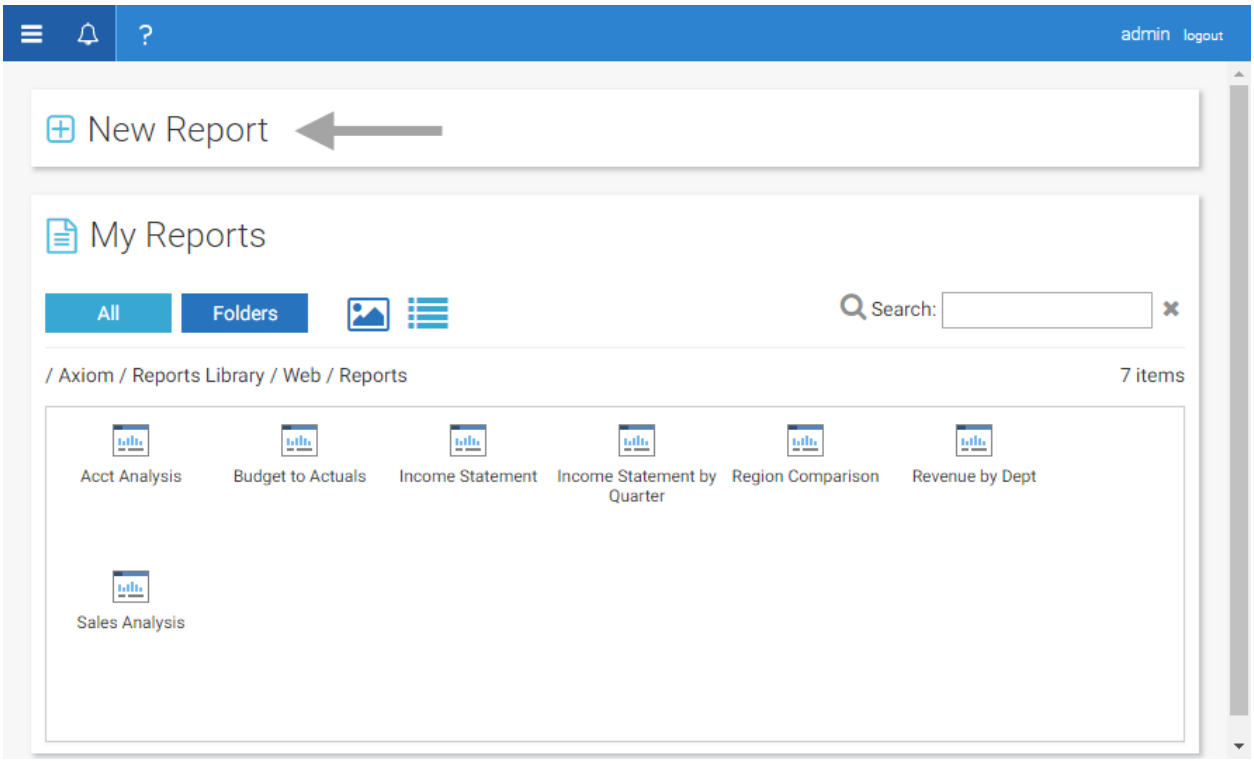
Report Builder icon on launch page

Alternatively, you can go to directly to the Report Builder page as follows:

Example On-Premise URL	`http://ServerName/Axiom/Reports` Where *ServerName* is the name of the Axiom Application Server, and Axiom is the default name of the virtual directory.
Example Cloud System URL	`https://CustomerName.axiom.cloud/Reports` Where *CustomerName* is the name of your cloud service system.

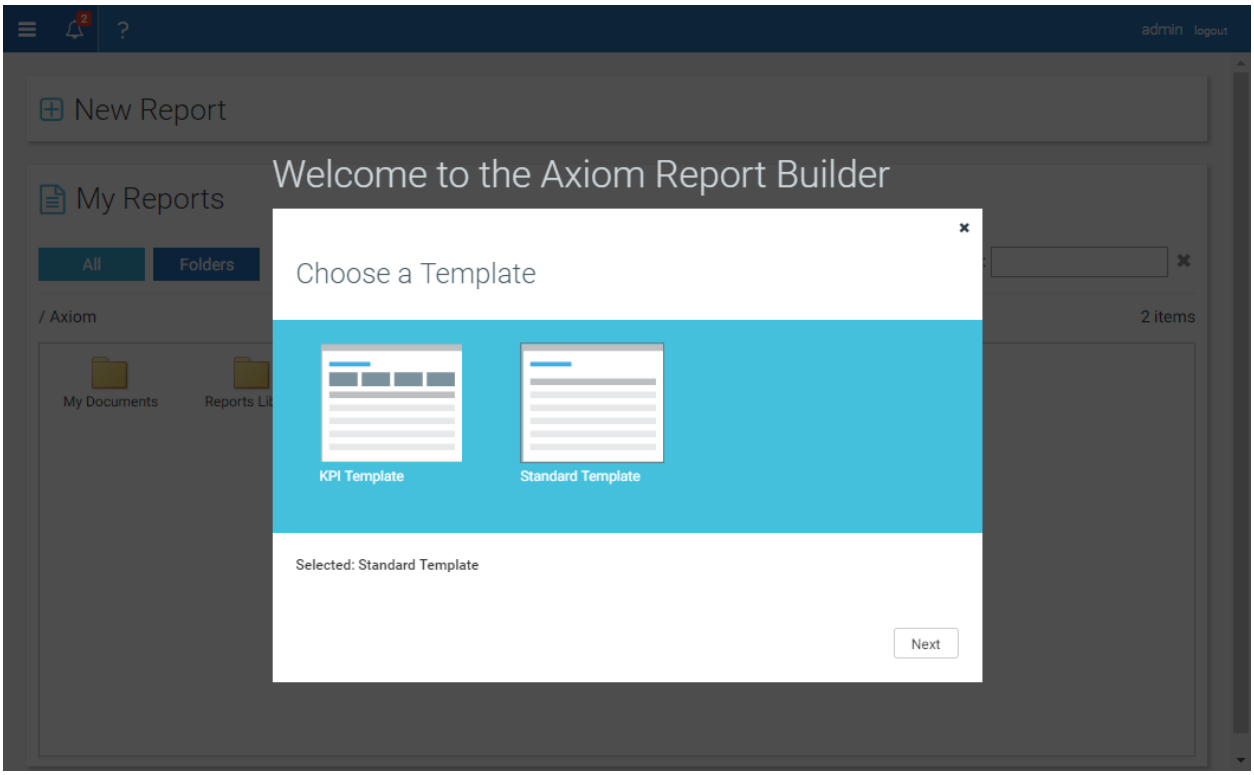
Creating web reports

Using the Report Builder page, you can create new web reports by clicking **New Report** at the top of the page.



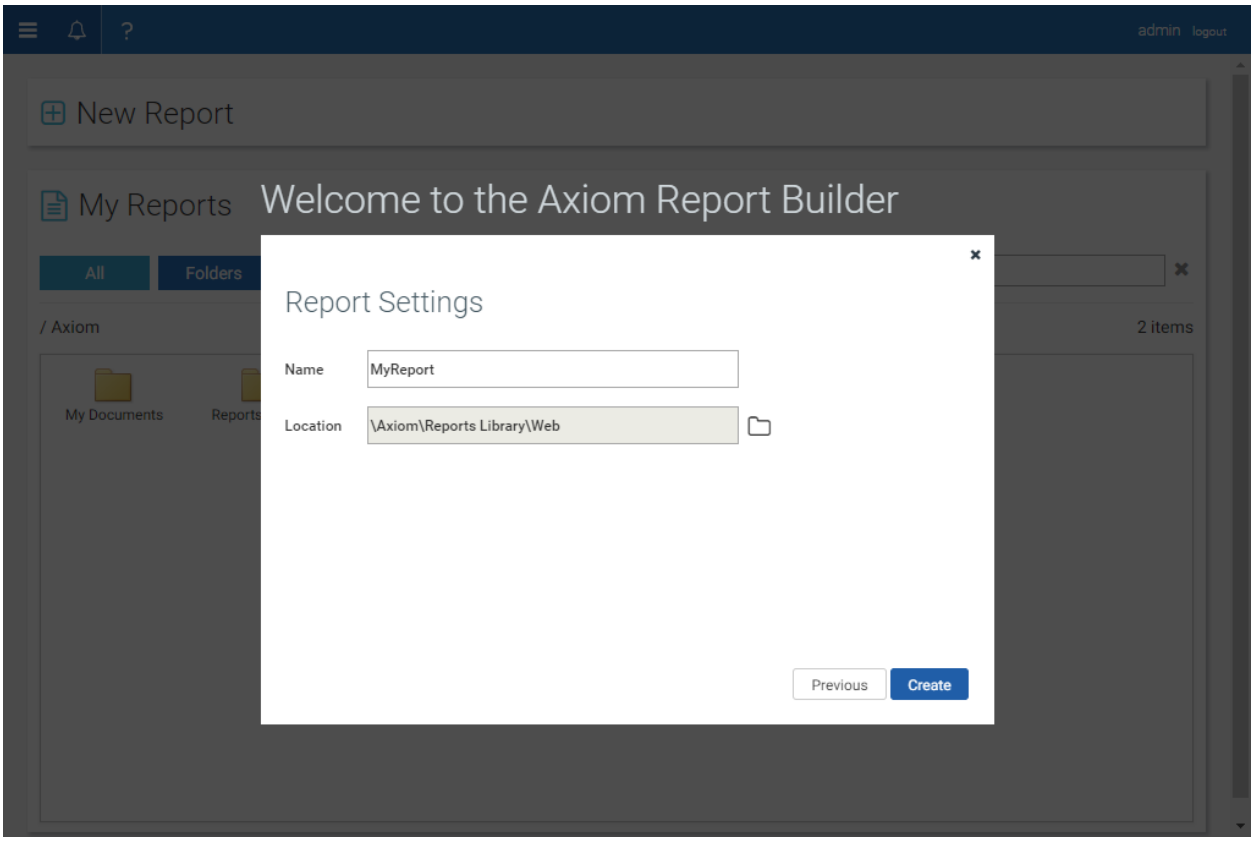
New Report action on Report Builder page

When creating a new report, the first step is to select a template. The template determines the layout of the report and the available components that you can configure. Currently, the Axiom Software platform provides two templates: a standard template to display data in a data grid, and a KPI template to display KPIs plus a data grid. You can create your own templates, and various Axiom Software products may provide additional templates.



Example web report template selection

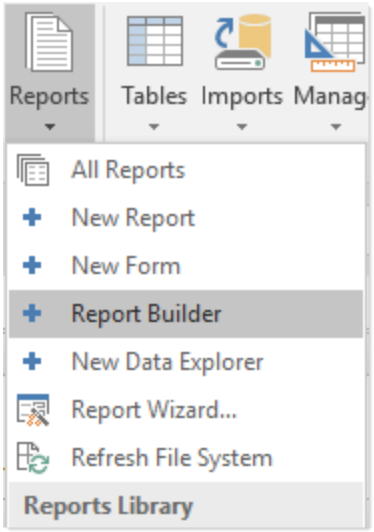
After selecting a template, you can then define a name and folder location for the web report. Web reports are stored in the Reports Library like spreadsheet reports and form-enabled reports, and use the same file security. Web reports use the AWR file type (Axiom Web Report).



Example web report name and location

After clicking the **Create** button, the new report automatically opens in the Report Builder, which is the web environment for configuring reports. For more information, see Using the Report Builder.

You can also initiate creation of new web reports from the Desktop Client (Excel or Windows), using the Reports menu. When you click **Report Builder**, the previously described creation wizard opens in your browser.



Report Builder option on Reports menu in Desktop Client

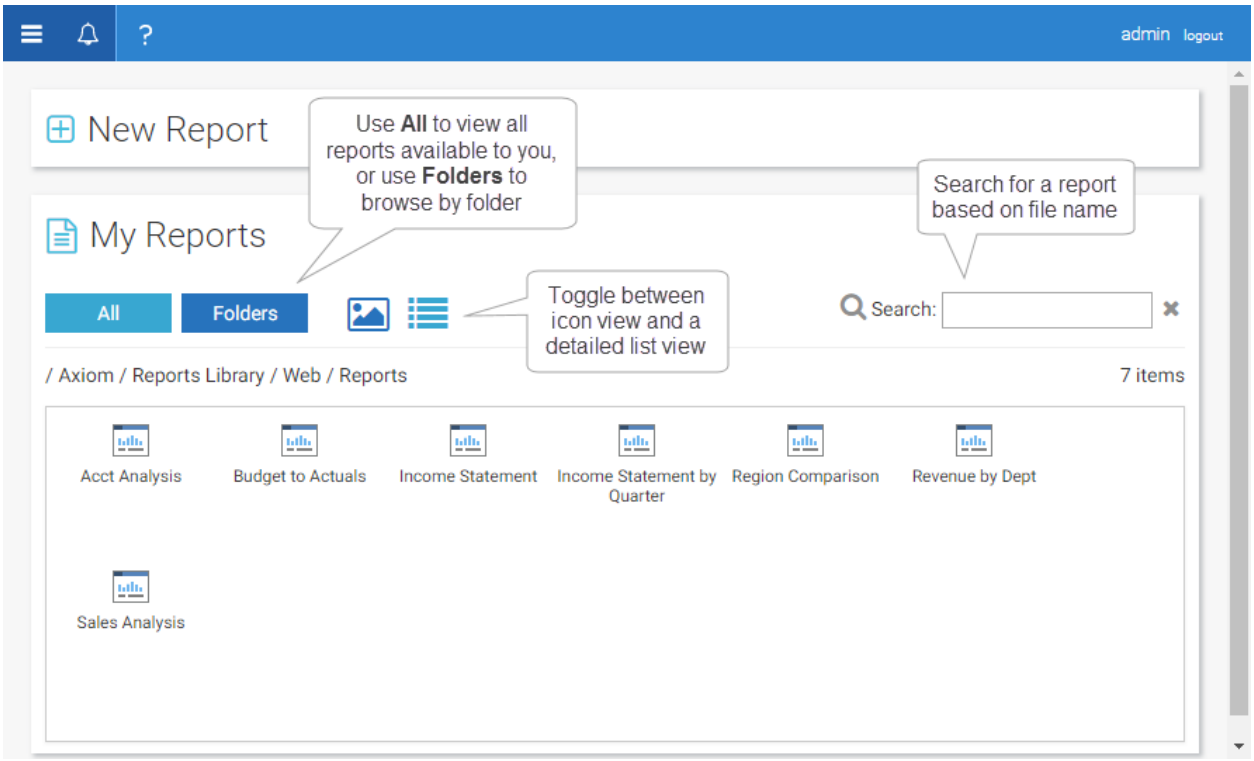
The Report Builder action can also be added to a custom task pane or ribbon tab, using the **Create New Report** command in the Command Library.

The ability to create a new web report is available to any user who has read/write access to at least one location in the Reports Library, or access to My Documents. If a user does not have any place to save a new report, the action is hidden.

Web report templates are created using Axiom forms. You can set up an Axiom form using certain supported components, and then save the form to the Axiom system template area to make it available when creating new web reports. For more information, see the following topic in Axiom Software Help: *Creating web report templates* (AX1701).

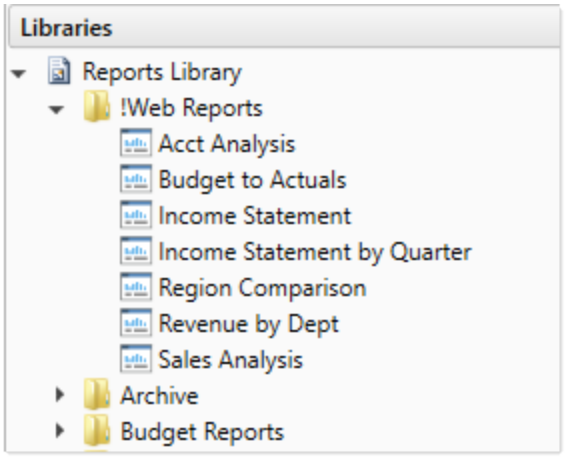
Viewing web reports

Using the **Report Builder** page of the Web Client, you can browse all web reports that you have access to. When you click on a report name, the report opens in the browser. You can then view and interact with the report, or open the Report Builder to make further changes to the report (if you have read/write access).



My Reports area on Report Builder page

In the Desktop Client (Excel or Windows), you can open web reports from the **Reports** menu, or directly from the Reports Library. When you open a web report from the Reports Library in the Desktop Client, it opens in the Web Client browser.



Web reports in the Desktop Client

You can also open web reports from custom task panes and ribbon tabs using document shortcuts or folder shortcuts, just like any other report. Web reports can also be used as the Home page for Axiom Software.

Once the web report is open, you can review the data in the report and also do the following (depending on the report configuration):

- Use built-in data grid tools such as the ability to sort the grid by any column, move between pages of data, and filter the displayed data in the grid by any column where filtering has been enabled
- If the grid shows grouped data, expand and collapse sections of data in the grid
- Use special grid features such as the ability to drill down rows of data or export grid data to a spreadsheet
- Launch supporting information for a KPI by clicking the action button in the top right corner of a KPI box
- Filter the report data on demand, using predefined refresh variables in the Filters panel

Using the Report Builder

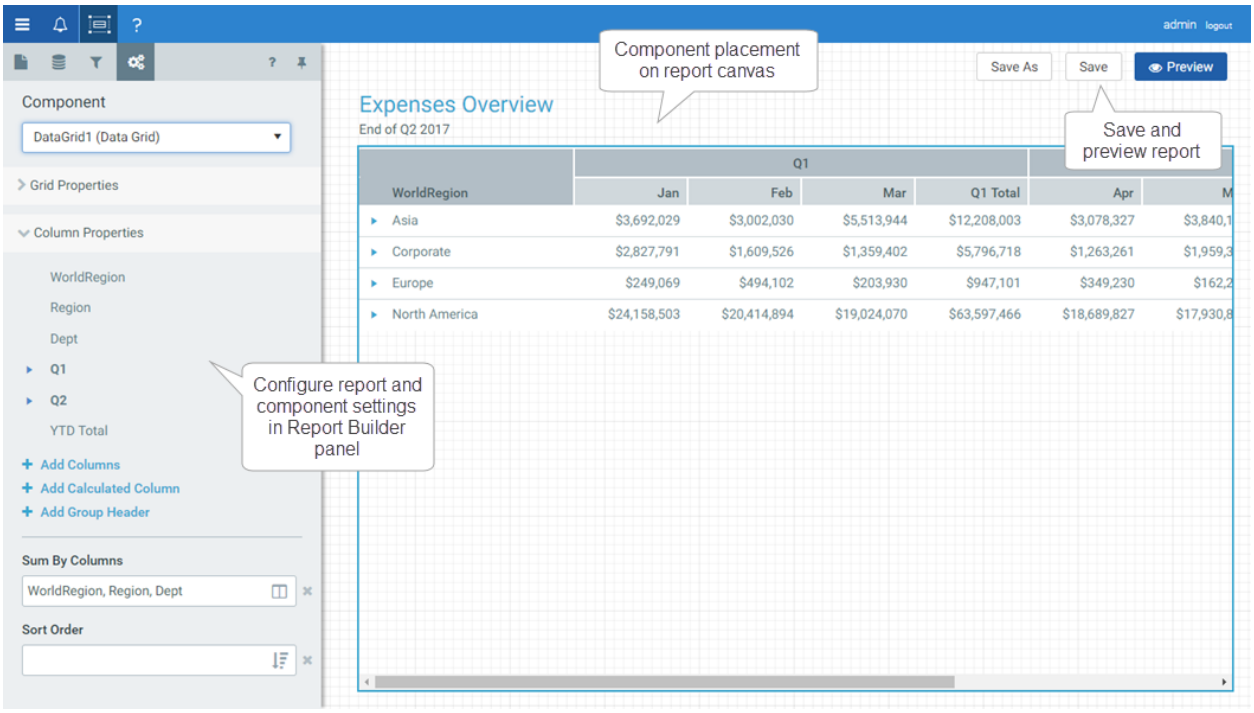
Using the Report Builder, you can configure all aspects of a web report directly in the browser. When you create a new web report, it automatically opens in the Report Builder. If you open an existing report and have rights to edit it, you can select **Tools > Edit Web Report** to open the report in the Report Builder.

This document provides a summary of the Report Builder environment. For more information, see the following topic in Axiom Software Help: *Using the Report Builder* (AX1676).

Report Builder overview

The Report Builder consists of the following:

- A configuration panel docked along the left-hand side of the screen, to configure various aspects of the report
- A report canvas area showing the placement of the components in the report
- A secondary toolbar displayed at the top right of the canvas, to save and preview the file



Example Report Builder

The Report Builder configuration panel is organized into separate tabs to define different properties for the web report. You can switch between tabs by clicking the icons at the top of the panel.

Report Builder Tabs	Icon	Description
Report Settings		Define general settings for the report, such as a title and an associated file group.
Data Source		Define one or more data sources for the report. Data sources determine the data available to report components.
Refresh Variables		Define one or more refresh variables for the web report, to allow users to filter the report data on demand.
Components		Configure the components in the report, such as a Label component for the report title, and a Data Grid component to display data.

When starting a new report, there are two properties that should be configured first: File Group Context and the Data Source.

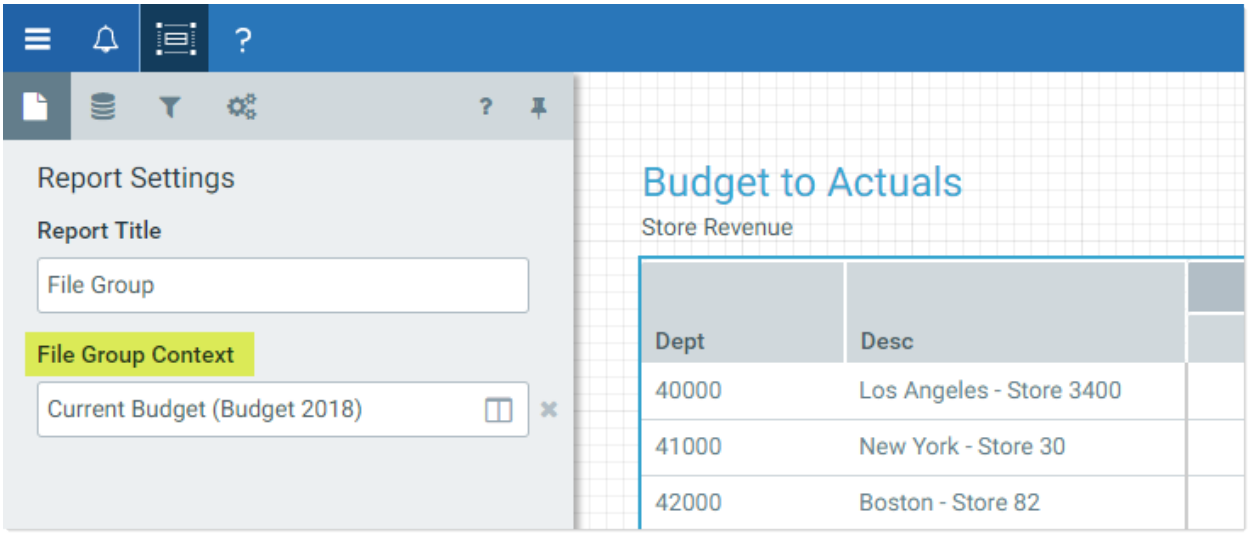
File Group Context

You can optionally associate the web report with a file group, in order to leverage file group table variables in the report. When a file group is selected (using an alias or a file group name), it impacts the report as follows:

- When choosing a table or column for critical areas of the report, the selection dialogs are limited to showing the tables associated with the file group by default.
- When you select a table or column, it is stored using the table variable name. This enables the report to update dynamically when the target of the file group alias changes.

For example, you can select the file group alias Current Budget and set up the web report to reference tables that are associated with the current target of the alias (Budget 2018). Later, you can roll over the file group to create Budget 2019 and update the Current Budget alias to point to the new file group. The web report will automatically update to reference the tables used by the Budget 2019 file group.

The **File Group Context** is defined on the **Report Settings** tab. If you do not need to use a file group context with the report, you can leave this blank.



Example Report Settings tab with File Group Context

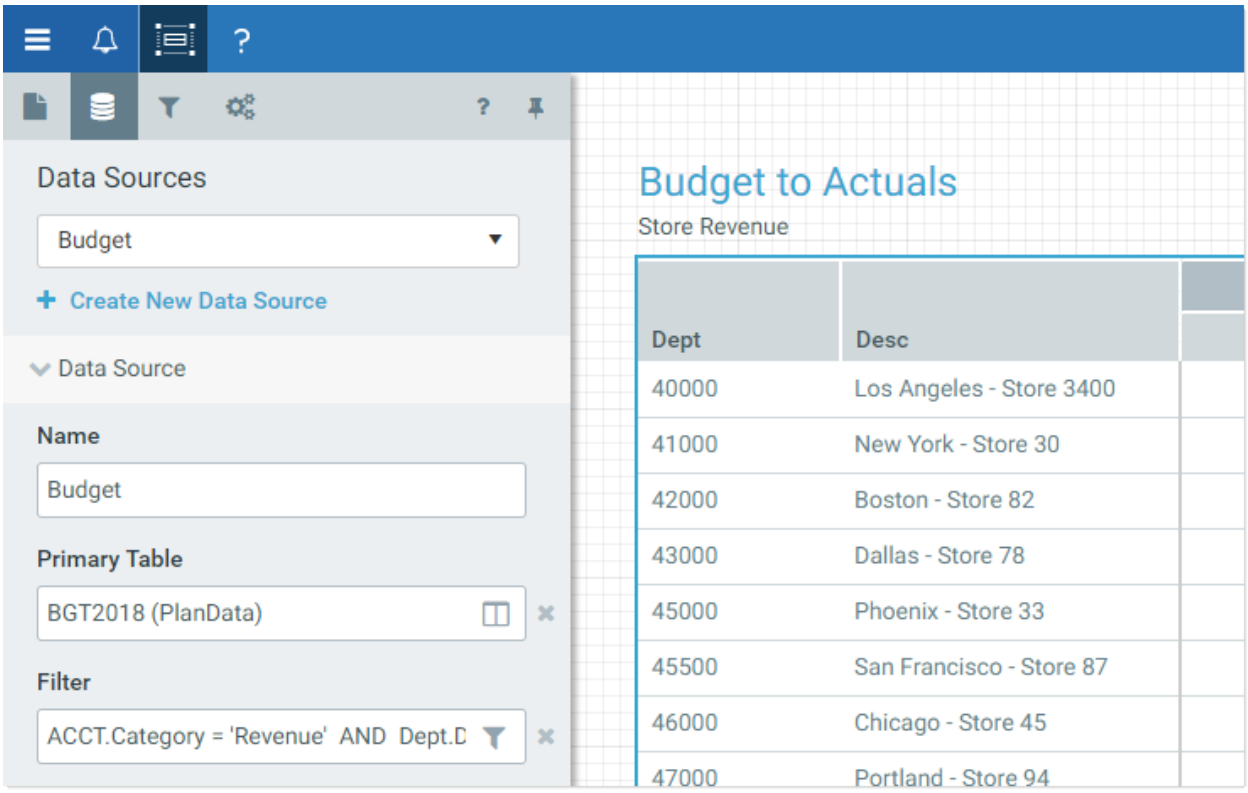
For more information about using a file group context with a web report, see the following topic in Axiom Software Help: *Associating a web report with a file group* (AX1699).

Data Source

Each web report with a data grid must have at least one defined data source. Data sources determine the data available to the Data Grid component. To define a data source, you specify a primary table and an optional filter:

- **Primary Table:** The primary table is the focus of the data query and determines the valid tables and columns that can be used by components assigned to the data source. For example, you can specify BGT2018 as the primary table and assign the data source to a Data Grid component. When configuring the data grid, you can include columns from any table that can be joined to BGT2018.
- **Filter:** You can optionally define a filter that applies to all components that use the data source.

Data sources are defined on the **Data Source** tab of the Report Builder. In this example, the report uses a file group context, so the selected primary table has been stored using the file group variable (displayed in parentheses after the table name).



Example Data Source tab with Budget data source

Web report components

Web reports consist of various components that you can configure as needed using the Report Builder. The component configuration determines the data and text to display in the report.

Currently, you cannot add new components to web reports, you can only configure the components provided by the template. Additionally, you cannot move or resize the existing components. In future releases, we plan to enhance the Report Builder to allow adding, deleting, moving, and resizing components.

The following components can be used in web reports. These components are the same as the corresponding components in Axiom forms, but with limited functionality and easier, streamlined configuration.

- **Data Grid:** The Data Grid component is the primary means of displaying data in a web report. You specify the columns to include, as well as other grid options such as filtering, drilling, and grouping data.
- **KPI Panel**: Display a series of key performance indicators in a web report.
- **Label**: Display titles, subtitles, and other text in a web report. Only the label text and the tooltip can be configured.
- **Hyperlink**: Link to other web reports, Axiom forms, and web pages. Only the display text and URL can be configured.
- **Panel:** Panels can be present in web report templates to group and position other components, but cannot be further configured in the web report and are invisible to the report creator.

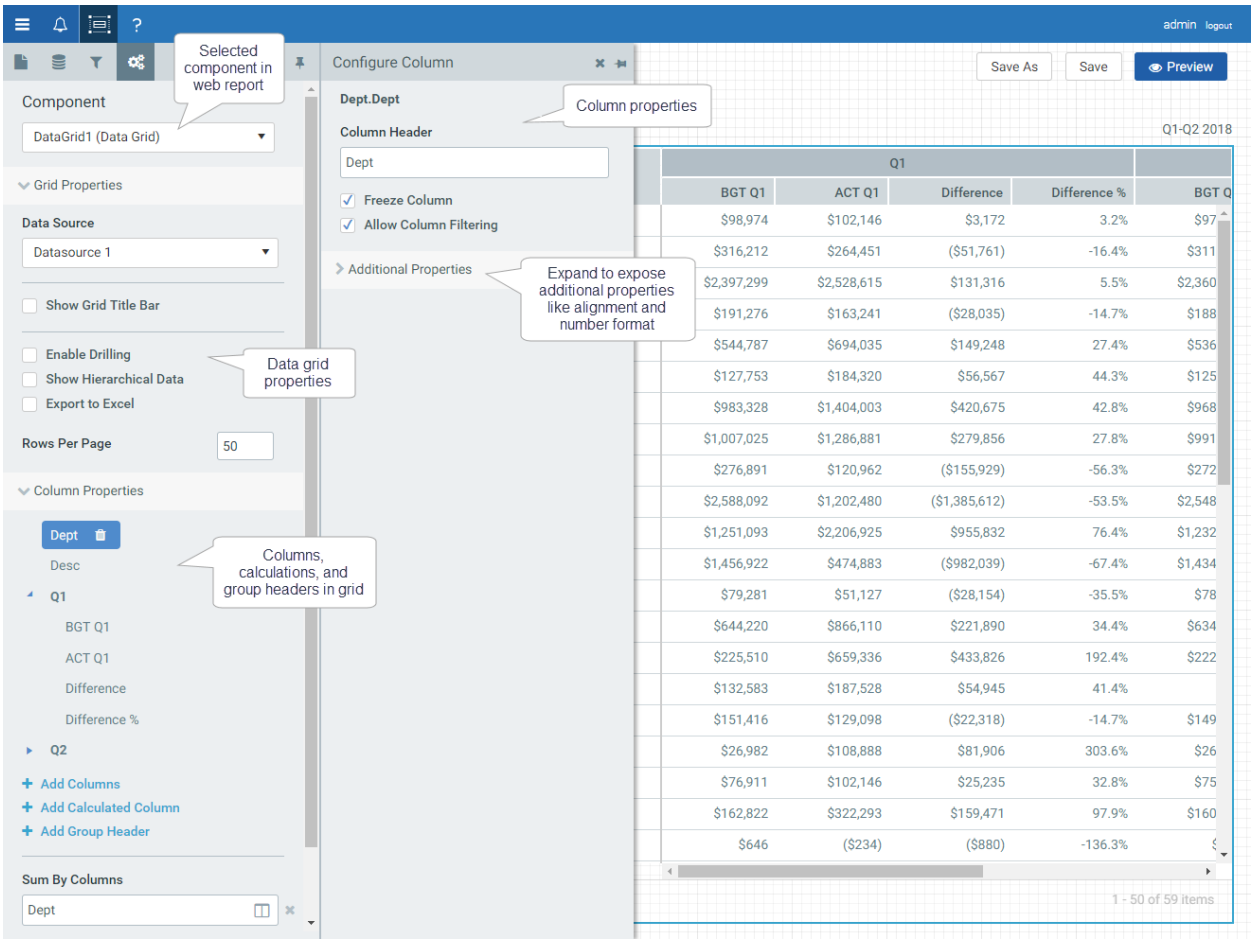
Component properties are configured on the **Components** tab of the Report Builder. You can select the component that you want to configure on the report canvas, or use the drop-down list at the top of the panel.

Data Grid component

The primary focus of web report configuration is the Data Grid component. Using the Report Builder, you can configure the following properties for a data grid:

- Specify the data source for the grid
- Add various table columns and calculations to display data in the grid
- Specify the sum level and sort level for the grid
- Define group header text, to span header text over certain groups of columns
- Configure various display properties for columns like alignment, number format, and frozen columns
- Configure various column features like end user filtering
- Enable various grid options like data drilling and grouping data by dimensions

The following screenshot shows an example configuration for a Data Grid component:



Example Data Grid configuration for web reports

The Data Grid component is a new component in Axiom forms as well. Although the process to set up a grid is different in both environments, the basic grid properties and the grid behavior for end users are the same. For more information on the Data Grid component, see Display queried data within a controlled data grid.

The following is a brief summary of some of the grid's special features in web reports:

- **Drilling**: You can enable drilling for the grid and specify one or more hierarchies to define the drilling levels. Users can then drill any row in the grid to see the data at a different level of detail. The drilling behavior is the same as described for Axiom forms, except that it is not supported to use a DrillLevels data source to define custom drilling levels in web reports.
- **Grouped hierarchical data**: You can display data in the grid grouped by hierarchical dimensions. When enabled, the data is grouped by the specified sum by columns, in the specified order. For example, if the sum by columns are Dept.WorldRegion, Dept.Region, Dept.Dept, the data is grouped first by WorldRegion, then by Region, then by Department. Users can expand each grouping to see the levels underneath. The grouping behavior is the same as described for Axiom forms.

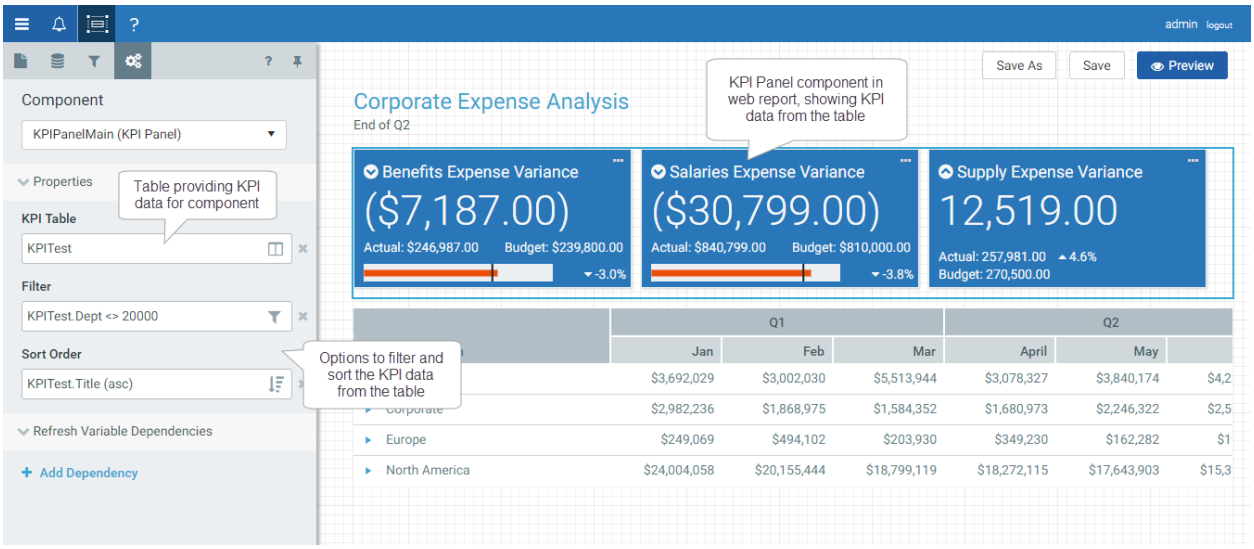
- **Export to Excel**: You can enable export for the grid, so that users can export the grid data to an Excel spreadsheet. If enabled, an **Export to Excel** button displays over the top right of the data grid.

Other features of the Data Grid component, such as the ability to select a row and the ability to display icons in the grid, are not currently available in web reports, but may be supported in future releases.

For more information on configuring a Data Grid component in web reports, see the following topic in Axiom Software Help: *Data Grid component for web reports* (AX1678).

KPI Panel component

Using the KPI Panel component, you can display one or more key performance indicators (KPIs) in a series of KPI boxes. To determine the KPI data to display in the component, you configure the component to reference a designated KPI table. The KPI data in this table is automatically formatted and positioned within each box, and the boxes are automatically positioned within the component.



Example KPI Panel configuration for web reports

The KPI Panel component is a new component in Axiom forms as well. Although the process to set up the component is different in both environments, the basic properties and behavior are the same. For more information on the KPI Panel component, see Display KPIs within an Axiom form. Note the following differences:

- Licensing does not apply to the KPI Panel component in web reports. It can be used regardless of your Axiom forms license level.
- In web reports, the command on each KPI can only be used to launch URLs or document shortcuts. It cannot execute a command from the Command Library.

The ability to create KPI tables is also a new feature in this release, to support the KPI Panel component in both Axiom forms and web reports. For more information, see New KPI tables.

For more information on configuring a KPI Panel component in web reports, see the following topic in Axiom Software Help: *KPI Panel component for web reports* (AX1717).

Filter data in web reports

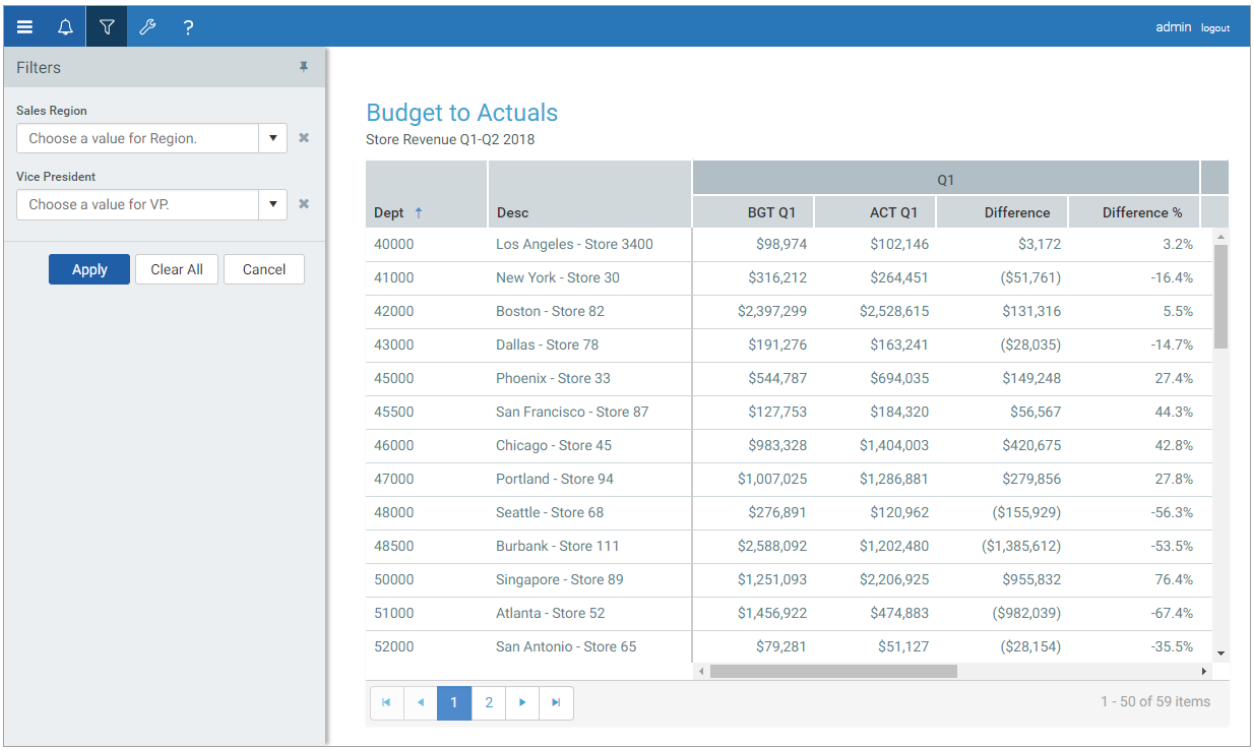
You can use refresh variables with web reports, to allow users to change the data in the report by selecting values for the refresh variables. For example, you may want the user to be able to dynamically filter the report by region. You can set up a refresh variable that allows users to select a region name. When the variable is applied to the report, the relevant data queries are filtered by the selected region.

The basic setup process for refresh variables in web reports is as follows:

- You define the refresh variables that you want to make available to users. Refresh variables have different types that control what users can select and how they select it.
- You create "dependencies" between refresh variables and data sources in the web report. The dependency tells the web report that the specified data source should be filtered by the variable value. All components that use the data source will then be updated to show the filtered data.

To continue the region example, imagine that your web report contains a Data Grid component that uses the GL2018 data source. In order for that data grid to be filtered by the user's selected region, you must create a dependency between the Region refresh variable and the GL2018 data source. When the user selects a value for region and applies that value to the report, the GL2018 data source is automatically filtered by that region, which in turn impacts the data shown in the data grid.

If a web report has defined refresh variables, those refresh variables display to web report users in the Filters panel (just like when using refresh variables for Axiom forms). The user can open the Filters panel, define values for the variables, and then apply the variables to impact the data shown in the report.



Example Filters panel for a web report

This section provides an overview of using refresh variables in web reports. For more information, see the following topic in Axiom Software Help: *Configuring refresh variables for web reports* (AX1687).

Differences when using refresh variables in web reports

The refresh variables available in web reports are essentially the same as the refresh variables available to spreadsheet Axiom files and Axiom forms. However, only a few variable types are currently supported for use in web reports, and there are some functionality limitations. For example:

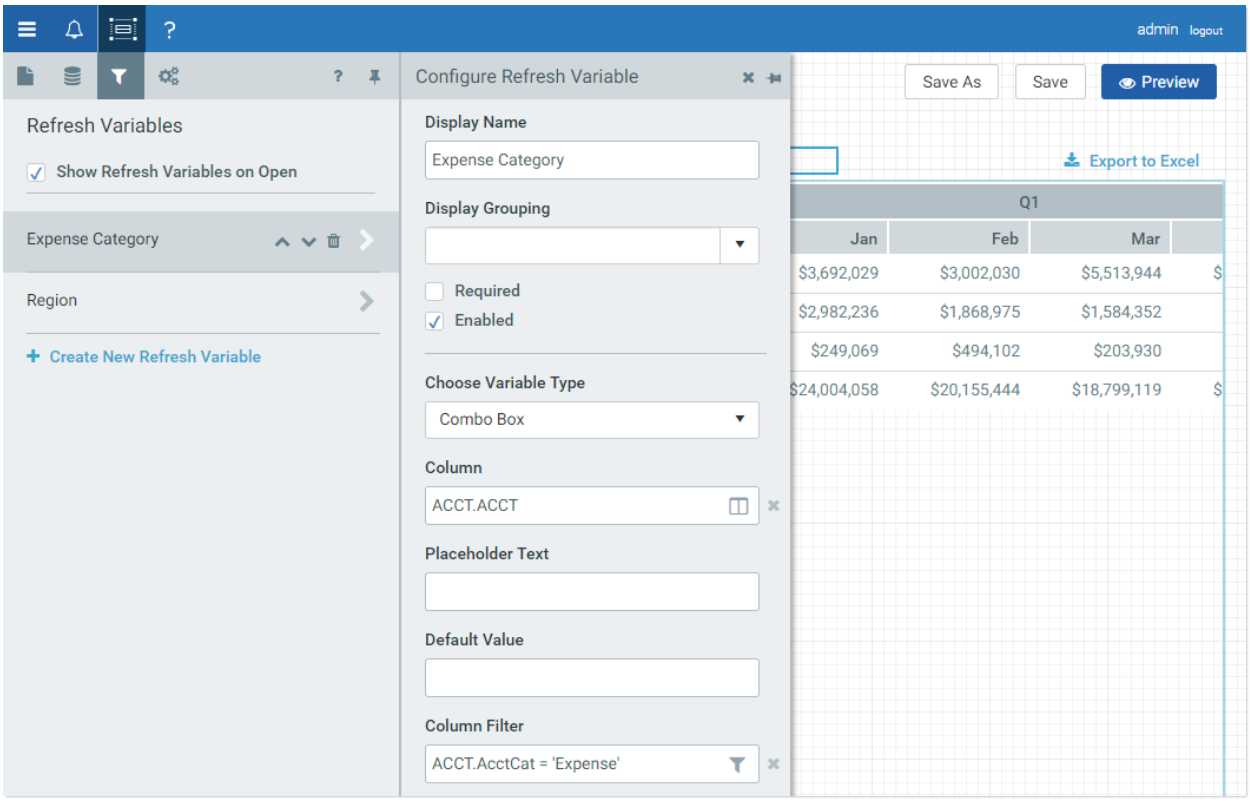
- Only Combo Box, Grid (multi-select), and Hierarchy Filter refresh variables can be used in web reports.
- The variables must use a table column or a hierarchy as the data source.
- The variables cannot be dependent on other refresh variables.

When you use refresh variables in spreadsheet Axiom files or Axiom forms, the variable setup is very manual but also very flexible. When a user selects a value for the refresh variable, that value is written back to the `[RefreshVariables]` data source. It is the responsibility of the file designer to configure the file to do something based on that selected value, but the value can be used for almost any purpose. For example, the value could be used to filter an Axiom query, or change the visible columns, or hide a component.

When you use refresh variables in web reports, the variable set up is much simpler, but the variable can only be used to impact data queries in report components. The advantage to this setup is that once a dependency is created between the variable and a data source, there is nothing more that the file designer needs to do to make it work. The data source will automatically be filtered by the variable. The disadvantage to this setup is that refresh variables cannot currently be used for other purposes, such as to hide components or change the visible columns in a grid. In future releases, we plan to expand the ways in which refresh variables can be employed in web reports.

Defining refresh variables for web reports

Refresh variables are defined on the Refresh Variables tab of the Report Builder. Here you can create, edit, and delete refresh variables for the web report. You can also specify whether the Filters panel should automatically be active when the report is opened.



Example Refresh Variables tab in Report Builder

The variable properties work the same way as when configuring refresh variables for other Axiom file types (though as mentioned, not all variable properties are available for use in web reports). The general variable properties like the Display Name are defined at the top of the panel, followed by the type-specific properties.

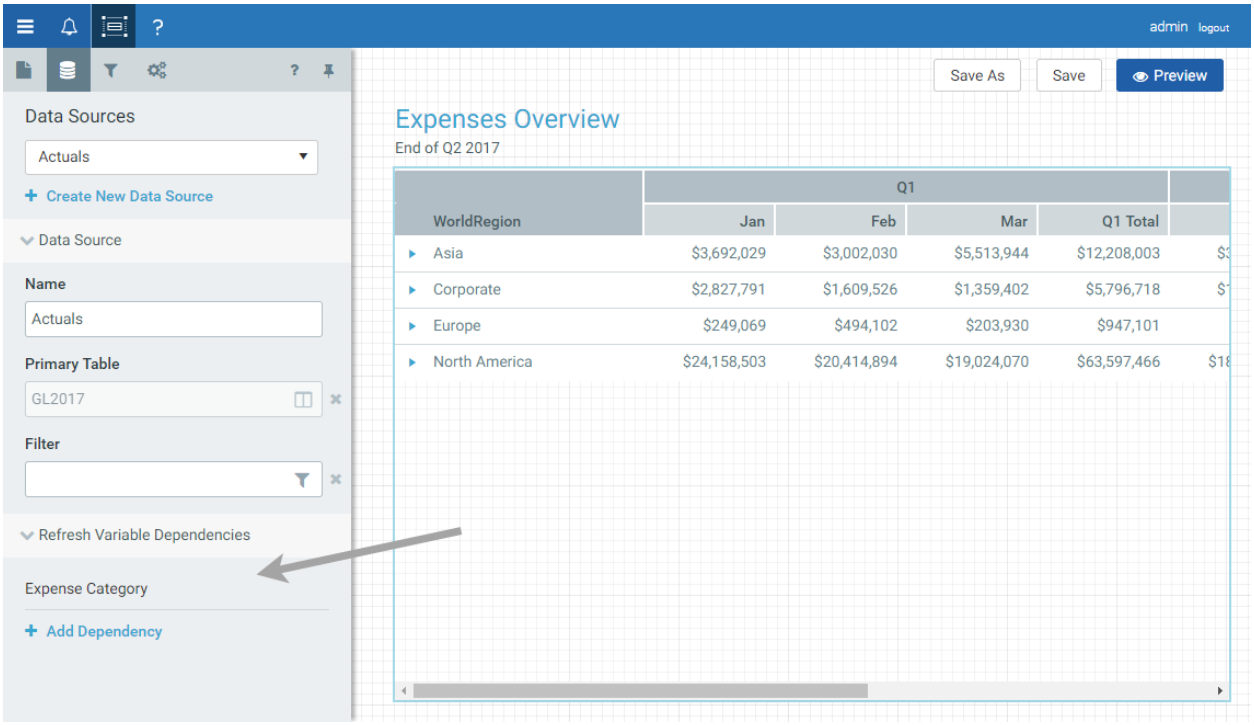
Defining refresh variable dependencies

Once you define a refresh variable for a web report, you must also define a dependency for the variable. The dependency creates an association between the refresh variable and a specific data source, so that the data source and its associated components are automatically filtered by the selected variable value.

Variable dependencies are defined on the **Data Source** tab of the Report Builder. To define dependencies, you first select the desired data source and then use **Add Dependency** to select the refresh variable that you want to associate with the data source.

NOTE: KPI Panel components can also be filtered by refresh variables, but because KPI Panel components do not use a data source, the dependency is defined directly on the component instead of on a data source. For Data Grid components, the dependency must be defined on the data source.

In the following example, the Actuals data source is dependent on the Expense Category variable. When a user applies a value for the Expense Category variable, the data source is automatically filtered by that value, affecting any Data Grid components that use the data source.



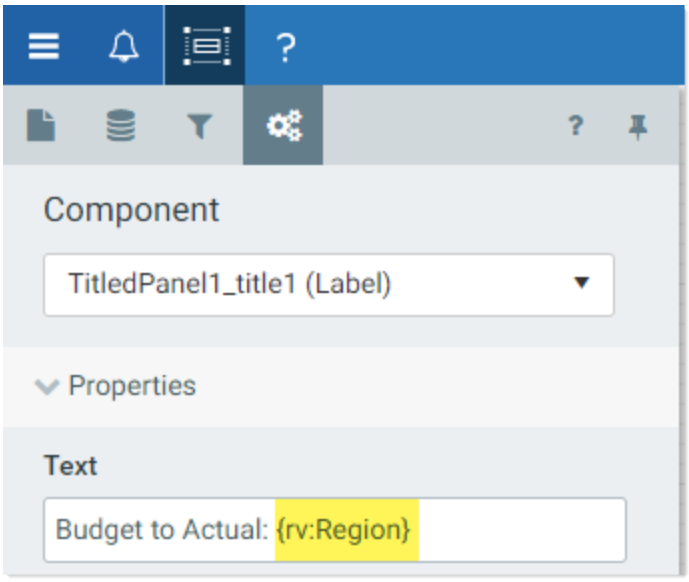
Example data source dependent on refresh variable

Referencing the current value of a variable

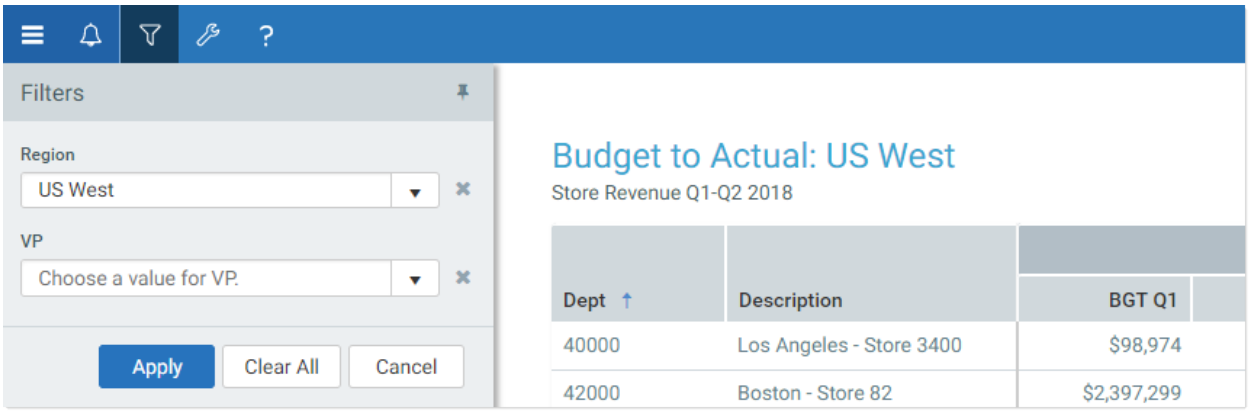
You can reference the current value of a refresh variable in Label components. This can be useful to create header text that updates dynamically based on the currently selected value. To reference the current value of a refresh variable, use the following syntax:

```
{rv:VariableName}
```

For example, if the variable name is Region, the label could look as follows:



When the label is rendered, the `{rv}` syntax is replaced with the currently selected value for the variable. In the following example, the selected value for the variable is "US West".



When the file is first opened, if the refresh variable has a defined default value, this is used as the currently selected value.

Related enhancements

The following additional enhancements were made to support the Report Builder and web reports.

Generate URL to web report

A new Axiom function of GetWebReportDocumentURL is available to generate a URL to a particular web report. For example:

```
=GetWebReportDocumentUrl("\Axiom\Reports Library\!Web Reports\Budget to Actuals.awr")
```

The resulting URL can then be used in Axiom forms or other web reports to link between web-enabled reports.

Updated iPad app

A new version of the Axiom Software iPad app is available to provide support for viewing web reports in the app. Web reports stored in the Reports Library are available in the Report Forms section of the app, and can be opened, viewed, and edited from this location (based on user permission to the files).

If you are using the older version of the app, it is necessary to update the app to the latest version if you want to view web reports.

Axiom forms

This section details the new features and enhancements made to Axiom forms.

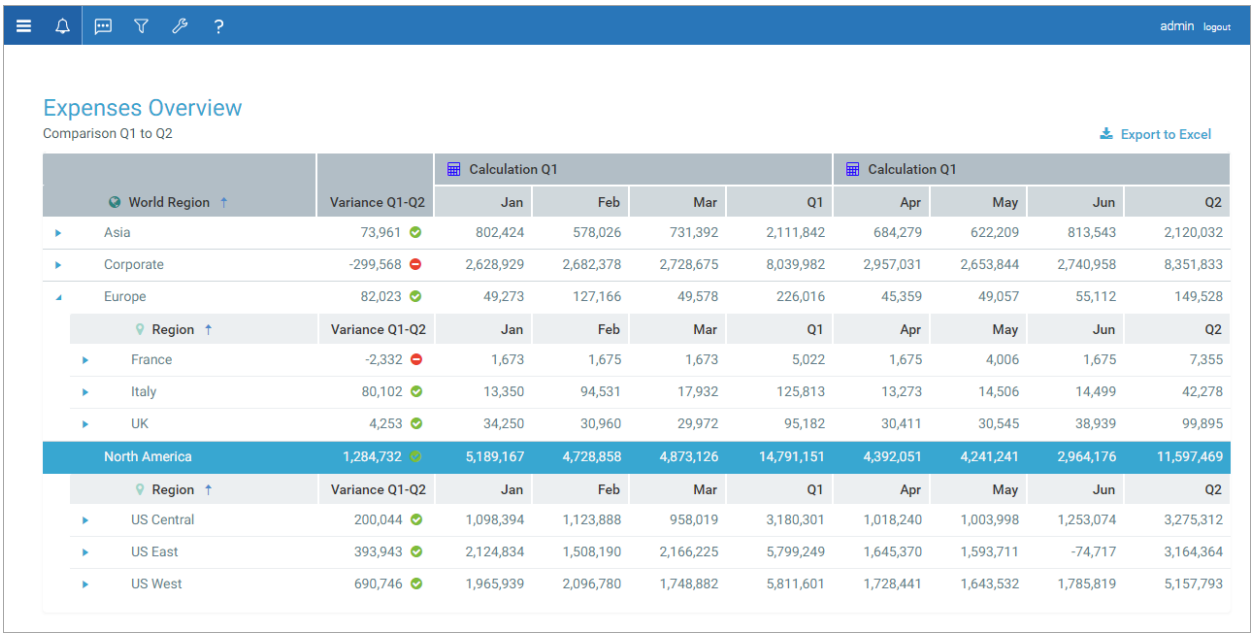
Display queried data within a controlled data grid

Using the new Data Grid component, you can directly query data from the Axiom Software database and display that data in a grid within an Axiom form. This component is intended to be used to display reporting data. It provides various features to make it easy for users to view the data, such as:

- Built-in sorting and filtering tools
- Ability to display data in expandable / collapsible groupings
- Configurable data paging (for example, show 50 rows per page)
- Ability to drill through data using hierarchies or custom drill levels

To configure the data grid, you specify a primary table and the columns that you want to display in the grid. The grid then queries the data from the database directly and displays it in the grid. You can also display calculations in the grid, such as to display the difference between two columns.

In addition to grid data, you can display icons in the grid. The icons can be for display purposes—such as to indicate an up or down trend at-a-glance—or they can be used to trigger certain actions. The icons can link to web pages or other web-enabled Axiom files, or they can perform certain actions such as opening a dialog panel.



Example Data Grid component in an Axiom form

The Data Grid component in Axiom forms is the same underlying component used in web reports. Although the general grid properties are the same, the setup is different due to the differing environments. The Axiom forms environment also supports a few additional features that are not yet available in web reports, such as icons. However, once the grid is rendered, the way that users interact with the grid is the same in both environments, providing a consistent user experience.

The following sections provide a summary of the setup and features of the new Data Grid component. For more information, see the following topic in Axiom Software Help: *Data Grid component* (AX1624).

Comparison of Data Grid to Formatted Grid

The Data Grid component has several differences as compared to the existing Formatted Grid component, to make it the primary choice for displaying reporting data:

- **Data query.** When using the Data Grid component, the grid queries data directly from the Axiom Software database and returns it into the form web page. The data is never brought into the source spreadsheet for the form. This provides a more performant data query option as compared to the Formatted Grid component, where you must first query data into the spreadsheet, and then tag the spreadsheet rows for inclusion in the grid.
- **Controlled display.** The Data Grid component uses standardized grid formatting with only a few formatting options. This makes the grid much easier to set up (no need to tag rows and columns with styles), and provides a consistent user interface across all forms. It also enables the built-in grid options such as sorting, filtering, and grouping.

- **Controlled data updates.** The Data Grid component does not refresh data on every form update. This behavior is intended to improve performance and also to persist the user's selections in the grid, such as expanded groupings, user-defined sorting and filtering, the current page of results, etc. If you want the data grid to query the database again and refresh the data as part of a form update, you must configure the grid to be dependent on another form component. Then when that component changes, the data in the grid is refreshed. Otherwise, the only time a data grid refreshes its data is when refresh variables are applied to the form.

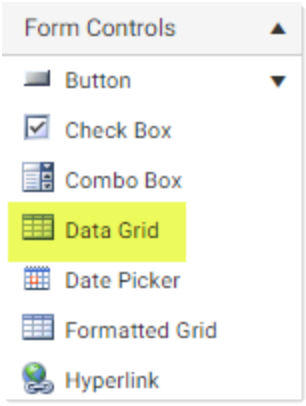
The Data Grid component is for data display only. It is not possible to edit the data in the data grid. The Formatted Grid component is still the primary way to support data edits in a grid format.

Data Grid creation

To use a Data Grid component in an Axiom form, you must add the component to the canvas and create a corresponding data source to define the grid columns.

Adding a Data Grid component to the form canvas

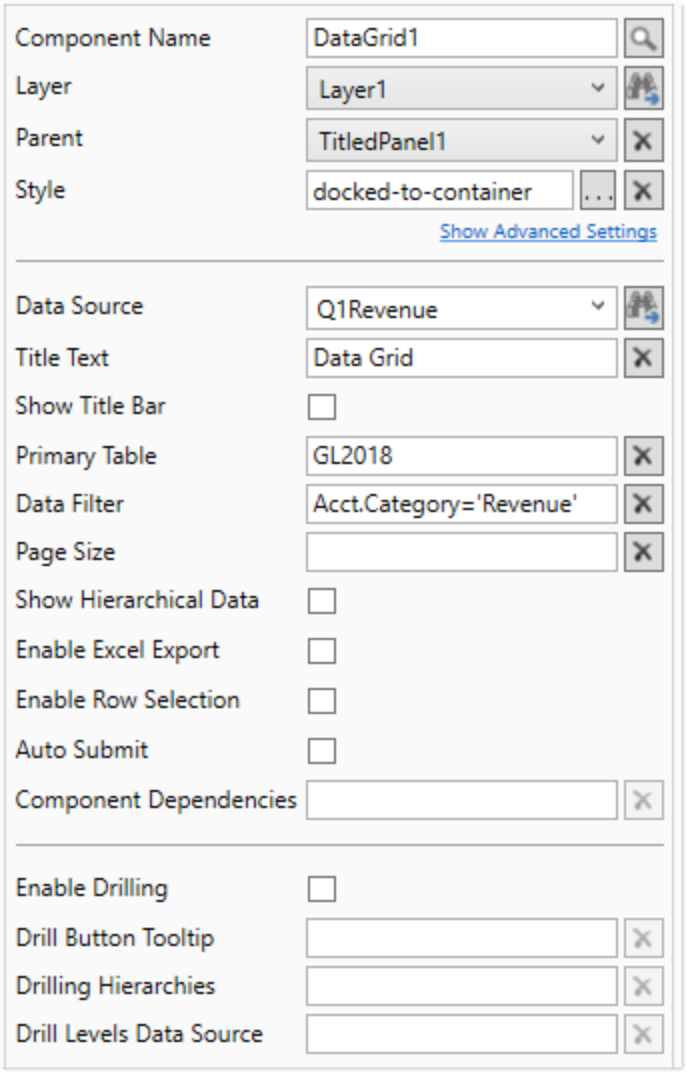
The Data Grid component is available in the **Form Controls** section of the component sidebar.



New Data Grid component in Form Designer

When configuring the component, you specify the following:

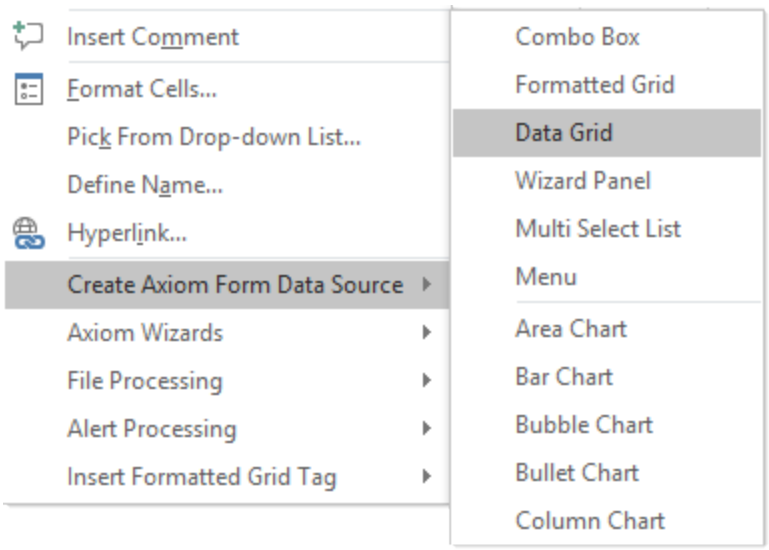
- The data source for the component, to define the columns to display in the grid and their display settings
- The primary table, to determine which columns are valid for inclusion in the grid
- An optional data filter to limit the data returned by the query and displayed in the grid
- The number of rows to display on each page of the grid
- Whether you want to show data grouped by expandable / collapsible hierarchical levels
- Whether you want users to be able to select rows in the grid, in order to drive other form contents (such as to show detailed information about the selected row in a secondary panel)
- Whether you want users to be able to export grid contents to an Excel spreadsheet
- Whether the grid is dependent on one or more other form components, to refresh the data in the grid based on changes made to those components
- Whether drilling is enabled, and if so various properties to determine the drilling options



Example Data Grid component properties

Defining the DataGridColumns data source

The `[DataGridColumns]` data source defines the columns to display in the grid and their display properties. To create a `[DataGridColumns]` data source, use the **Create Axiom Form Data Source** wizard on the right-click menu:



New data source in right-click wizard

The `[DataGridColumns]` data source uses three different types of rows. The primary row type is `[Column]`, to define a database column to display in the grid. For each column, you can specify properties such as:

- The column name, such as Dept.Dept or GL2018.M1
- The display name for the column, to display in the grid header
- Whether the column is a "sum by" column of the grid, to determine the summation level of the rows (same concept as Axiom query "sum by" columns)
- Whether the data is initially sorted by the column and whether the sort is ascending or descending
- Whether the column is frozen for scrolling purposes
- Whether users can perform ad hoc filtering using the column values
- Display properties such as the column width, numeric format, and alignment
- Optional data properties for the column, such as alternate aggregation or a column filter (same concept as Axiom query column filters)

The `[DataGridColumns]` data source also supports a row type of `[CalculatedColumn]`, to display custom calculations in the grid. Instead of specifying a column name, you specify a calculation. For example, to display the percent difference between budget and actual, you would enter a calculation such as `(GL2018.M1-BGT2018.M1)/BGT2018.M1`. Calculations can use various math operators and can reference any column name, calculated field name, or column alias name. The columns referenced in calculations do not have to be otherwise included in the grid, but they do have to be valid in the context of the primary table.

Lastly, you can specify `[ColumnGroup]` and `[EndGroup]` rows to define groups of columns and header text to display above the group.

The following screenshot shows an example `[DataGridColumns]` data source with some columns, some calculated columns, and a column group.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
4														
5		[DataGridColumns;Q1Revenue]	[ColumnName]	[Header]	[HeaderIcon]	[IsVisible]	[SortOrder]	[IsSumBy]	[Width]	[NumericFormat]	[IsFilterable]	[IsFrozen]	[Calculation]	[HeaderAlignment [A
6		[Column]	Dept.VP	VP		TRUE	1	TRUE	175			TRUE		
7		[ColumnGroup]		Q1	fa-calendar									
8		[Column]	GL2018.Q1	Actuals		TRUE			125	$#,##0_);($#,##0)				
9		[Column]	BGT2018.Q1	Budget		TRUE			125	$#,##0_);($#,##0)				
10		[CalculatedColumn]	Difference	Difference		TRUE			125	$#,##0_);($#,##0)	TRUE		GL2018.Q1-BGT2018.Q1	
11		[CalculatedColumn]	PDifference	Difference %		TRUE			125	0.00%	TRUE		(GL2018.Q1-BGT2018.Q1)/BGT2018.Q1	

Example DataGridColumns data source

The resulting data grid would look as follows in the rendered form:

Expense Analysis

Q1 2018

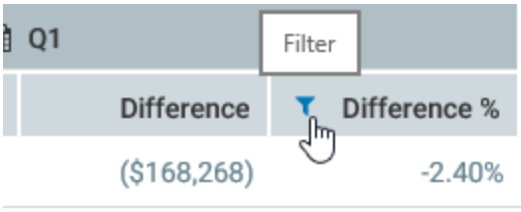
	Q1			
VP ↑	Actuals	Budget	Difference	Difference %
Bree Sigman	$6,851,080	$7,019,348	($168,268)	-2.40%
Evan Simpson	$14,526,309	$13,202,408	$1,323,901	10.03%
Frank Martinez	$837,295	$661,170	$176,125	26.64%
Javier Guppy	$7,828,034	$5,751,656	$2,076,378	36.10%
Jen Smith	$17,510,851	$15,367,824	$2,143,027	13.94%
Michelle Choi	$264,451	$316,212	($51,761)	-16.37%

Example data grid

While viewing the grid, users can:

- Sort the grid by any column by clicking on the column header. Each click toggles between ascending sort, descending sort, and no sort.
- Change the column width by clicking and dragging, or change column order by dragging and dropping.

- Filter the grid by values in a column, if `[IsFilterable]` is enabled for the column. Users can hover over the column header to see the filter icon, and then click on the icon to set the filter options.



- Select a row in the grid, if **Enable Row Selection** is enabled for the component. For more information, see Select rows in the grid.
- Expand and collapse grouped sections of data, if **Show Hierarchical Data** is enabled for the component. For more information, see Display grouped data in the grid.
- Move between pages of data by using the paging controls at the bottom of the grid.

All of these state changes are preserved for the current session when the form is updated, unless the Data Grid is forced to refresh based on a specified component dependency, or when refresh variables are applied.

Display grouped data in the grid

You can show data in the grid grouped by hierarchical dimensions. Instead of flat rows of data, data is shown in expandable / collapsible groups based on the sum by levels defined for the query.

To show grouped data in the grid:

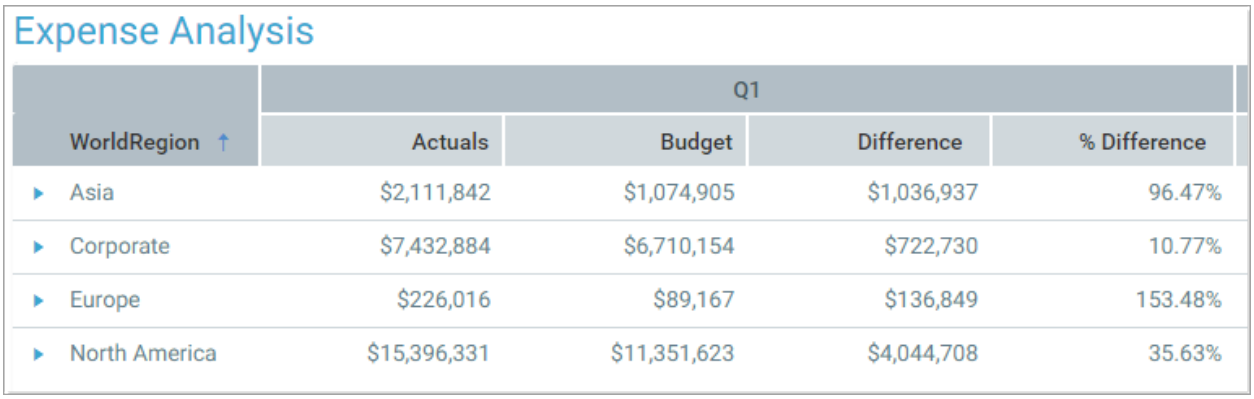
- In the component properties, enable **Show Hierarchical Data**.
- In the data source, add the appropriate reference table columns and enable them as sum by columns. The top-level grouping must be listed first, followed by the next level, and so on. There must be at least two sum by columns defined in the data source to create a grouping.

The following example data source shows sum by columns of `Dept.WorldRegion`, `Dept.Region`, and `Dept.Dept` (in that order). This means that WorldRegion is the top-level grouping, and Dept is the lowest level.

	A	B	C	D	E	F	G	H
4								
5		[DataGridColumns;Q1Revenue]	[ColumnName]	[Header]	[IsVisible]	[SortOrder]	[SortDirection]	[IsSumBy]
6		[Column]	Dept.WorldRegion	WorldRegion	TRUE	1		TRUE
7		[Column]	Dept.Region	Region	TRUE	2		TRUE
8		[Column]	Dept.Dept	Dept	TRUE	3		TRUE

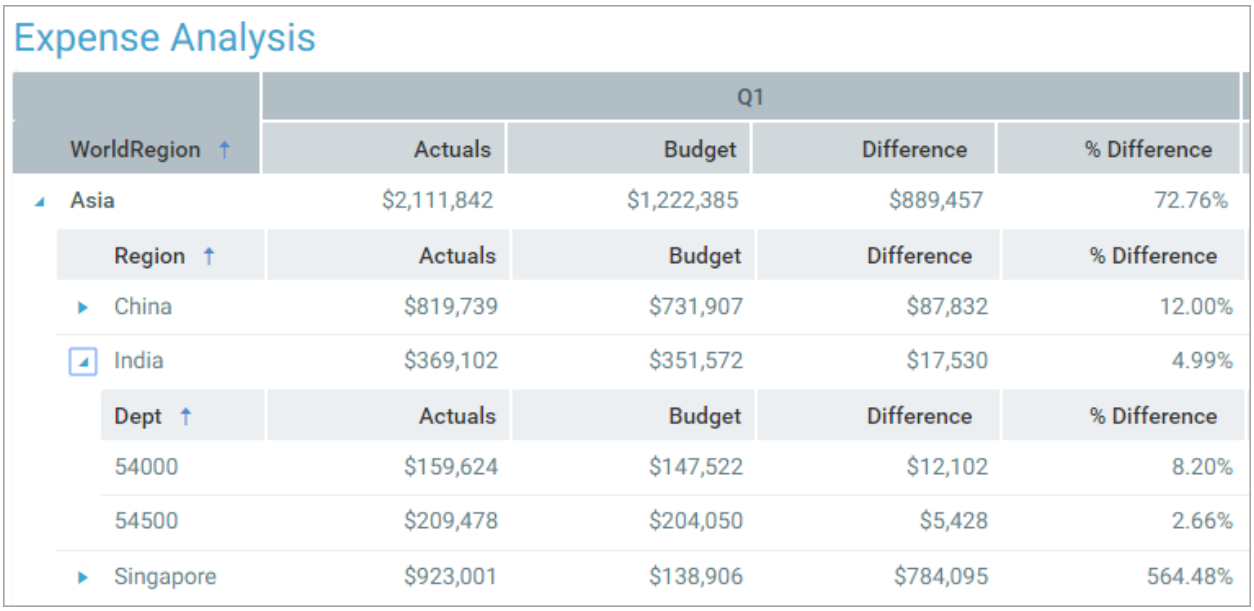
Example data source with multiple sum by columns

When this grid is rendered, it will display as follows, with data initially grouped by world regions.



Example data grid with Show Hierarchical Data enabled

You can expand a world region to see the region data underneath it, and then expand a region to see the department data for that region.



Example expanded grouping to show next dimension level

The other columns in the data source are the same for all levels. For example, if the column GL2018.Q1 is included in the grid, then you will see that data at the world region level, then the region level, then the department level.

Select rows in the grid

If **Enable Row Selection** is enabled for the Data Grid component, users can select a row in the grid. The current dimension values for the selected row are submitted back to the form source file, and written to the `[SelectedRowValue]` column of the `[DataGridColumns]` data source. The current dimension values are determined by the sum by columns specified in the data source. Each row in the grid has a unique value for each sum by column.

If you want the Axiom form to respond to the currently selected row, then you can set up the file so that another component references one or more of the selected dimension values, and changes based on those values.

NOTE: The ability to select rows is only available when using the Data Grid in Axiom forms. Row selection is not available in web reports.

A data grid could display data summed by department, where the sum by column is Dept.Dept.

Dept	Desc	Months	
		Jan	Feb
20000	Corporate	$376,388	$389,632
21000	Corporate Administration	$118,402	$156,274
22000	Information Technologies	$502,486	$521,661
23000	Purchasing & Materials Mgmt	$141,996	$187,932
24000	Business Development	$32,125	$31,618

Example data grid with a selected row

If a user selects the row for Dept 22000, then that value is submitted back to the source file and written to the `[SelectedRowValue]` column in the `[DataGridColumns]` data source, in the row that defines the Dept.Dept column.

	A	B	C	D	E	F	G
4							
5		[DataGridColumns;Report]	[ColumnName]	[Header]	[IsVisible]	[IsSumBy]	[SelectedRowValue]
6		[Column]	Dept.Dept	Dept	TRUE	TRUE	22000
7		[Column]	Dept.Description	Desc	TRUE		

Dimension value written back to data source

If instead the grid has two sum by columns, such as Dept.Dept and Acct.Acct, then each row in the grid has a unique combination of Dept and Acct. If a user selects the row for Dept 22000 and Acct 5400, then both of those values are written back to the data source, in the rows that define the Dept.Dept column and the Acct.Acct column.

	A	B	C	D	E	F	G
4							
5		[DataGridColumns;Report]	[ColumnName]	[Header]	[IsVisible]	[IsSumBy]	[SelectedRowValue]
6		[Column]	Dept.Dept	Dept	TRUE	TRUE	22000
7		[Column]	Dept.Description	Desc	TRUE		
8		[Column]	Acct.Acct	Acct	TRUE	TRUE	5400
9		[Column]	Acct.Description	Desc	TRUE		

Multiple dimension values written back to data source

There are a number of ways that the form could respond to the selected value in the grid. For example, you might want to display detailed information about the selected row in another grid or in a chart. The grid or chart would need to be set up with formulas that look to the appropriate cells of the `[SelectedRowValue]` column, so that the data in the grid or chart changes based on the currently selected row.

The selected row behavior can also be used when the grid data is grouped by dimensions. In this case, the dimension values for the selected row depend on its grouping level in the grid. Dimension values are written back for the selected grouping level and all grouping levels above it, but not for any grouping levels below it.

Drill rows in the grid

You can enable drilling for Data Grid components and configure the grid so that certain drilling selections are available to users. If drilling is enabled, users can drill any row in the grid by clicking on a drill icon that displays on each row. Users can select a drilling level from among the available selections, and then the drilling results are presented in a separate web page. Users can continue to drill the drilling results if desired, or return to the original grid and drill again from there.

To enable drilling, select **Enable Drilling** and then configure the rest of the drilling properties as desired. To define the drilling options that display to users, you can use hierarchies, or you can define a DrillLevels data source.

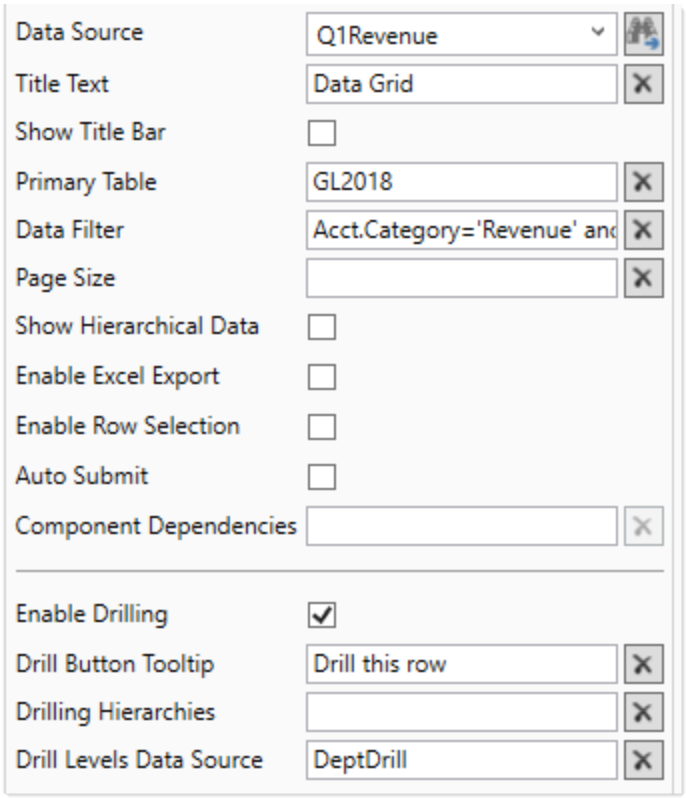
- **Drilling Hierarchies**: You can specify one or more hierarchies to define the drilling options. Users can then select any column in the hierarchy to drill to that level. This property supports the same syntax used by Formatted Grid components to define hierarchies for drilling—for example, `Dept` to use all hierarchies on the Dept table, or `Dept:Geography` to use that specific hierarchy.
- **Drill Levels Data Source**: You can define a `[DrillLevels]` data source and enter the name of that data source in this property. Users can then select any drilling option listed in that data source. The `[DrillLevels]` data source is the same data source used by Formatted Grid components to display custom drilling options and works the same way.

The drilling options presented to users are determined as follows:

- When drilling the data grid, the Drill Levels Data Source is always used if defined, otherwise the Drilling Hierarchies are used. If Drilling Hierarchies is blank, all relevant hierarchies are used (based on the primary table).

- When drilling the drill results, the Drilling Hierarchies are always used (the Drill Levels Data Source is ignored if set). If Drilling Hierarchies is blank, all relevant hierarchies are used (based on the primary table).

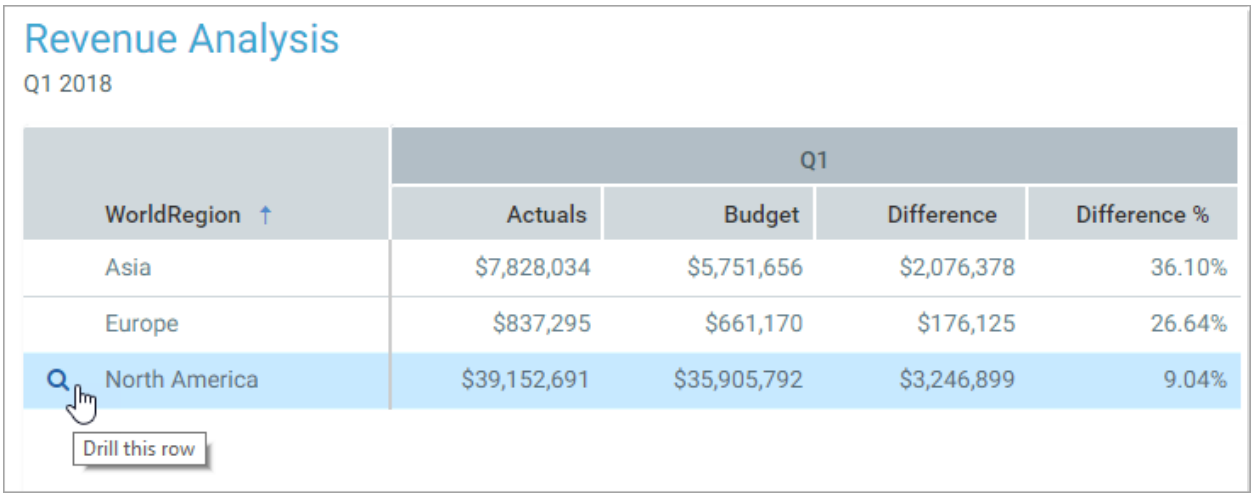
For example, imagine the Data Grid component is configured as follows:



Example component properties with drilling enabled

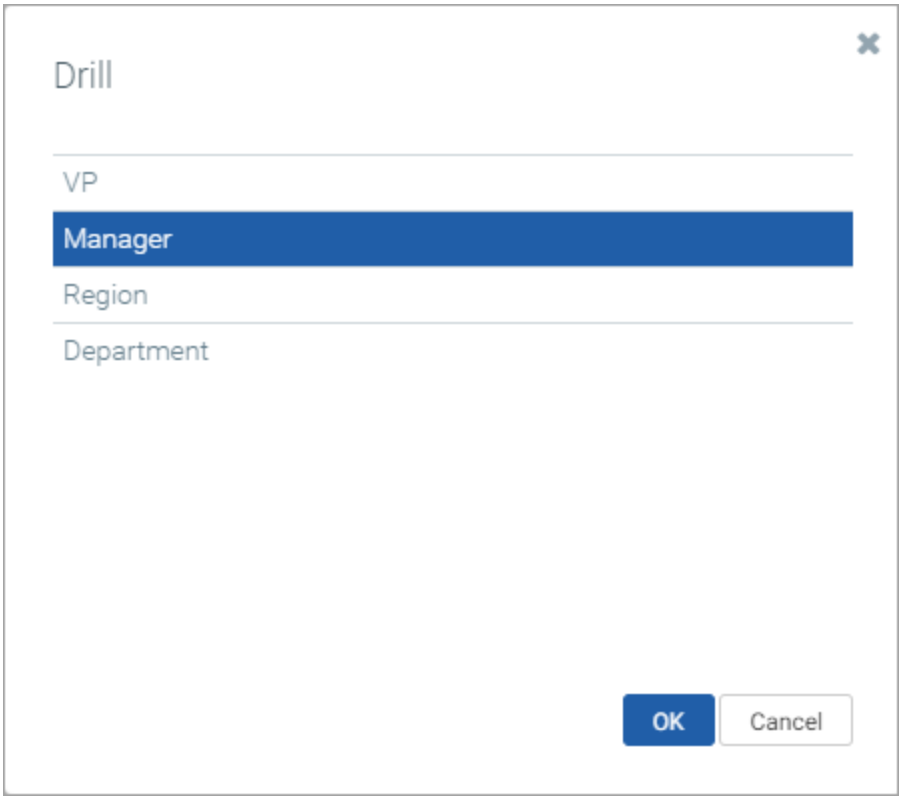
- The primary table is GL2018, which means than any hierarchies or drilling levels must be valid in the context of this table.
- **Enable Drilling** has been enabled for the component.
- The **Drilling Hierarchies** property is left blank, but a data source name is defined for **Drill Levels Data Source**. This means that the data source will be used to define the drilling options for the data grid instead of using a hierarchy.

When this file is viewed as a form, the first column in the data grid is now the drill action column. When a user hovers their cursor in that column, they can see a drill icon (a magnifying glass) for the current row. The user can click on the icon to initiate a drill for that row.



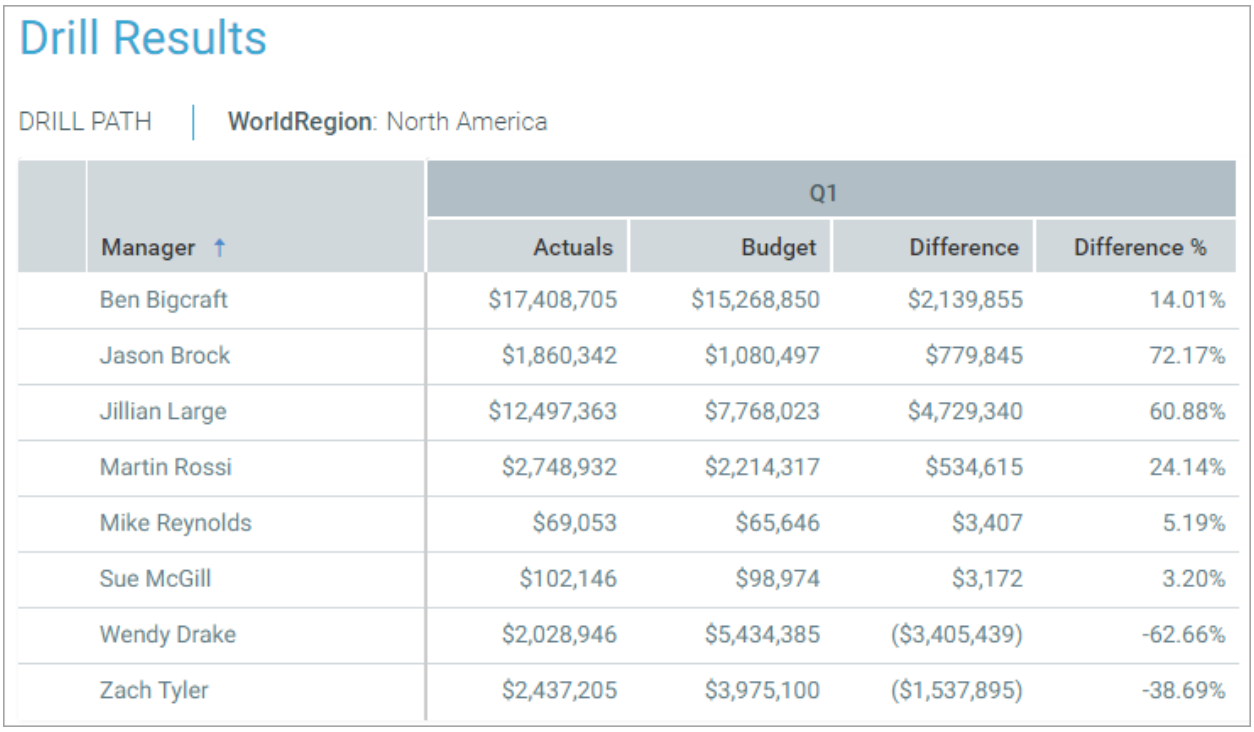
User hovers cursor over first column to see drill icon

Once the user has initiated the drill, a dialog opens to display the available drilling levels. In this example, these are the drilling levels defined in the `[DrillLevels]` data source named DeptDrill. (If instead one or more hierarchy names had been specified in the Data Grid properties, the dialog would prompt users to select from the hierarchy levels.)



User is prompted to select drilling level

After the user selects a drill level (Manager in this case), a new browser tab opens to display the drill results. In this example, the data for the North America row is now shown at the Manager level.



Drill Results

DRILL PATH | **WorldRegion**: North America

Manager	Q1 Actuals	Q1 Budget	Q1 Difference	Q1 Difference %
Ben Bigcraft	$17,408,705	$15,268,850	$2,139,855	14.01%
Jason Brock	$1,860,342	$1,080,497	$779,845	72.17%
Jillian Large	$12,497,363	$7,768,023	$4,729,340	60.88%
Martin Rossi	$2,748,932	$2,214,317	$534,615	24.14%
Mike Reynolds	$69,053	$65,646	$3,407	5.19%
Sue McGill	$102,146	$98,974	$3,172	3.20%
Wendy Drake	$2,028,946	$5,434,385	($3,405,439)	-62.66%
Zach Tyler	$2,437,205	$3,975,100	($1,537,895)	-38.69%

Drill results display in new tab

The drill results automatically include all columns from the original data grid except for the sum by columns, which are replaced by the selected drill level. The current drilling level is displayed at the top of the page.

If desired, the user can further drill on the drill results. When drilling the drill results, the Drill Levels Data Source is ignored and instead the Drilling Hierarchies are used to determine the drill options. Because Drilling Hierarchies was left blank in our grid configuration, the user is presented with all relevant hierarchies for the primary table. When drilling the drill results, the results are presented in the same page, overwriting the current results.

For more information on configuring drilling for a data grid in Axiom forms, see the following topic in Axiom Software Help: *Setting up drilling for Data Grid components* (AX1691).

NOTE: Data grid drilling is available in both Axiom forms and web reports. However, in web reports, the drilling options must be configured using hierarchies. It is not possible to define custom drill levels in web reports.

Display icons in the grid and trigger actions

You can display icons in a Data Grid component, and optionally use icons as "buttons" for users to trigger actions relating to the current row. Icons can be used as follows:

- Use the `[HeaderIcon]` property to display an icon in the column header, next to the header text. This is for display only; actions cannot be triggered from the header icon.

- Use the `[Icon]` property to display one or more icons in a column (either by themselves or along with a database column or calculation). The icons can be display only—such as to conditionally display a "good" or "bad" indicator next to a number—or used as buttons to trigger an action.
- Use the `[HoverActions]` property to display one or more icons in a column, where the icons only display when a user hovers over a cell in the column. Hover icons can only be used as buttons to trigger an action.

The icons are the same symbols that are available for use in Formatted Grid components, buttons, and labels. To use an icon in the grid, you put the icon name into the appropriate property. For example, you can enter `fa-globe` in `[HeaderIcon]` to display a globe icon next to the header text. You can use the symbol chooser dialog for other features to look up the appropriate names—for example, right-click a cell and choose **Insert Formatted Grid Tag > Symbol** to easily access the symbol chooser dialog.

For `[HeaderIcon]` and `[Icon]`, you can list a single icon name along with an optional color, such as `fa-globe;blue`. For colors, you can use standard web color names, hexadecimal color codes, or the color style codes in the new Axiom2018 skin. In this case, the icon is always present.

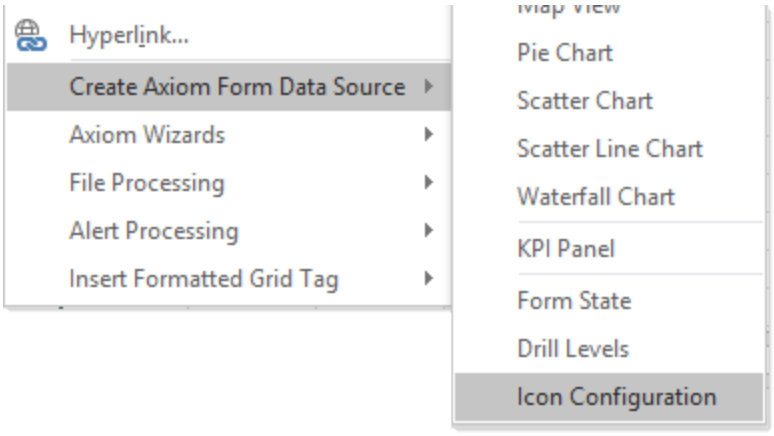
For `[HoverActions]`, you must define a separate `[IconConfig]` data source that defines one or more icons along with the assigned action and optional condition. You can also optionally use an `[IconConfig]` data source with the `[Icon]` setting, if you want to display multiple icons or define conditions for the icons.

This topic provides a summary of using icons in a Data Grid component. For more information, see the following topic in Axiom Software Help: *Using the IconConfig data source with Data Grid components* (AX1697).

NOTE: Icons are only supported when using the Data Grid in Axiom forms. Icons are not available in web reports.

Defining an icon data source

The `[IconConfig]` data source defines the icons to display in the grid, and their display properties. To create an `[IconConfig]` data source, use the **Create Axiom Form Data Source** wizard on the right-click menu:



New data source in right-click wizard

For each icon listed in the data source, you can specify properties such as:

- The icon name, color, and tooltip
- The action to perform when a user clicks the icon, as well as an optional confirmation message
- A condition to determine whether the icon is present on a particular row

The following screenshot shows an example icon data source to define conditional markers for a calculated column:

	M	N	O	P	Q	R	S	T
58								
59		[IconConfig;Variance]	[Color]	[Tooltip]	[IconName]	[Condition]	[Action]	[Confirmation]
60		[Icon]	A31	Variance down	fa-minus-circle	VarianceQ1toQ2<0		
61		[Icon]	A41	No variance	fa-exclamation-triangle	VarianceQ1toQ2=0		
62		[Icon]	A51	Variance up	fa-check-circle	VarianceQ1toQ2>0		

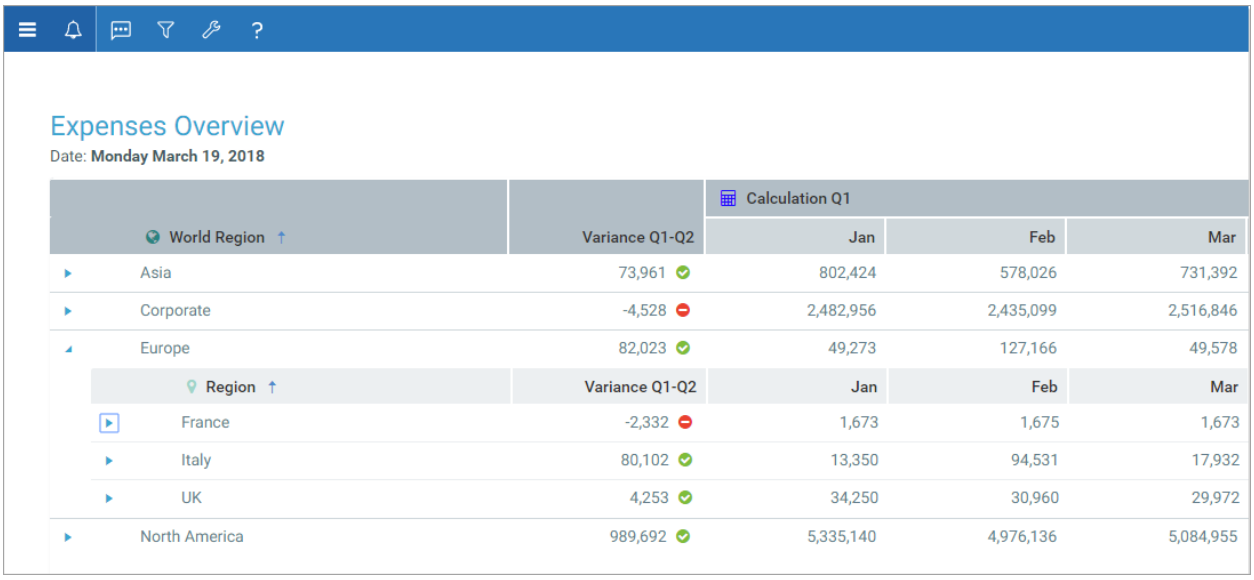
Example IconConfig data source with conditions

This icon data source can then be referenced in the `[Icon]` column of the `[DataGridColumns]` data source. In this example, it is placed on the VarianceQ1toQ2 calculated column.

	K	L	M	N	O	P
10						
11		[DataGridColumns;GridColumnsConfigMain]	[ColumnName]	[Icon]	[Header]	[HeaderIcon]
12		[Column]	Dept.WorldRegion		World Region	fa-globe;A12
13		[Column]	Dept.Region		Region	fa-map-marker;A10
14		[Column]	Dept.Description	DepartmentAction2	Department	
15		[Column]	Acct.Description	AccountIcon	Account	
39		[CalculatedColumn]	VarianceQ1toQ2	Variance	Variance Q1-Q2	
40		[ColumnGroup]			Calculation Q1	fa-calculator;blue
41		[Column]	GL2017.M1		Jan	
42		[Column]	GL2017.M2		Feb	
43		[Column]	GL2017.M3		Mar	
44		[CalculatedColumn]	CalculatedQ1		Q1	
45		[EndGroup]				

Example DataGridColumns data source referencing the IconConfig data source

When the grid is rendered, the conditions are evaluated per row to determine whether the icon displays. In this case, rows with variances above 0 display with a green check and variances greater than zero display with a red minus.



Expenses Overview

Date: **Monday March 19, 2018**

World Region	Variance Q1-Q2	Calculation Q1 Jan	Feb	Mar
Asia	73,961	802,424	578,026	731,392
Corporate	-4,528	2,482,956	2,435,099	2,516,846
Europe	82,023	49,273	127,166	49,578

Region	Variance Q1-Q2	Jan	Feb	Mar
France	-2,332	1,673	1,675	1,673
Italy	80,102	13,350	94,531	17,932
UK	4,253	34,250	30,960	29,972

North America	989,692	5,335,140	4,976,136	5,084,955

Example data grid with conditional icons

Using actions with icons

You can use the `[IconConfig]` data source to perform an action when an icon is clicked. This is optional for fixed-display icons in the `[Icon]` property, but required for hover icons in the `[HoverActions]` property.

The `[Action]` column in the icon data source can contain either of the following:

- A URL to open when the icon is clicked. The URL must use full `http:` syntax.

  You can optionally use the variable `{value}` in the URL, in order to reference the current cell value in the URL. For example, if the icons are displayed in the same column as department codes, the `{value}` variable will be replaced with the department code for the current cell.

- A command to execute when the icon is clicked. For example, you can use the Dialog Panel command to open a dialog with more information about the current row.

  You can use any command that is available for use in forms, though, some commands may not make sense to execute in this context and may not work as expected. To create the command statement, right-click the cell and select **Axiom Wizards > Command Wizard**. You can then use the Shortcut Target to select a command from the Command Library and configure its shortcut properties, just like you would for a Button component.

	Q	R	S	T	U	V	W
16							
17		[IconConfig;RegionActions]	[Tooltip]	[IconName]	[Action]	[Confirmation]	[Condition]
18		[Icon]	Region Information	fa-map-o	command://ShowFormDialogPanel?DialogPanel=DialogPanelWorldMap		

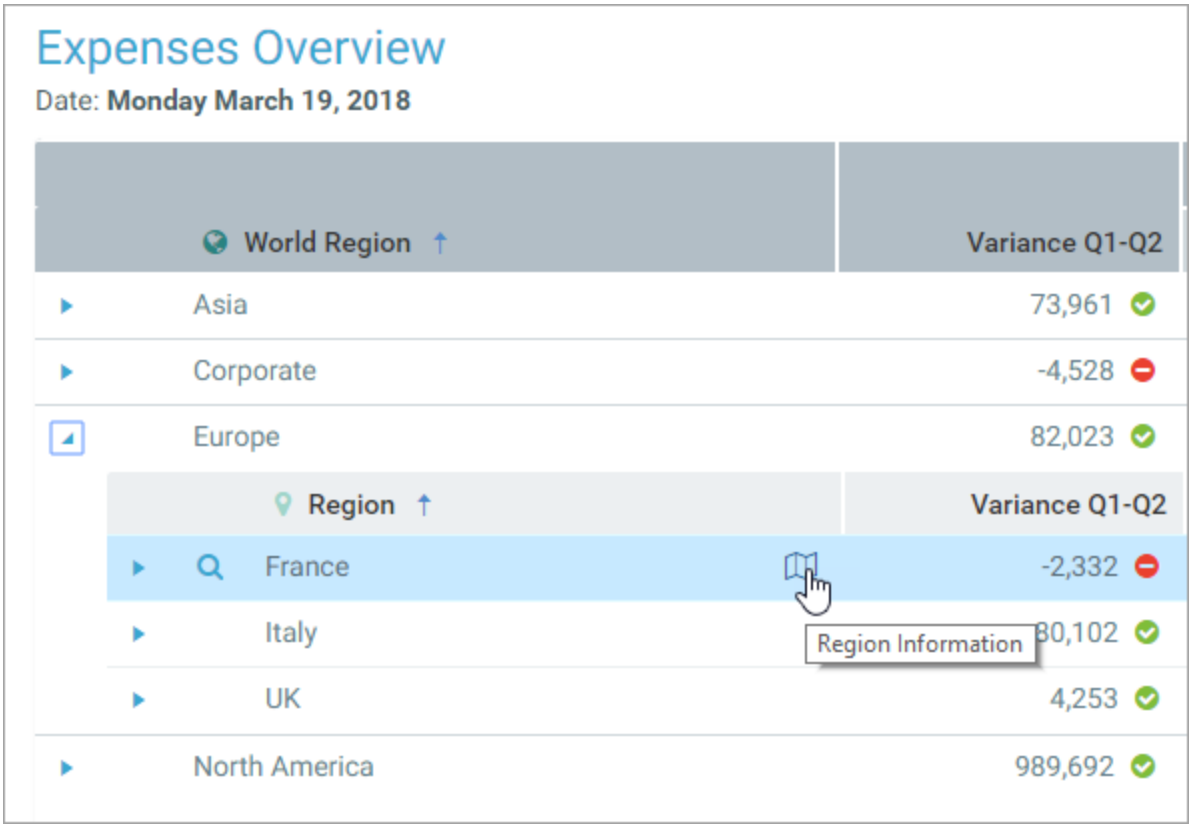
Example IconConfig data source with an action

If the action is a command, you can optionally reference any dimension value for the current row, by using the `[ActionRowValue]` column in the `[DataGridColumns]` data source.

When the user clicks on an icon to perform an action, the dimension values for the current row are written back to the `[ActionRowValue]` column. This works just like the `[SelectedRowValue]` column discussed previously, except in this case the values are written back when the icon is clicked, instead of when the row is selected.

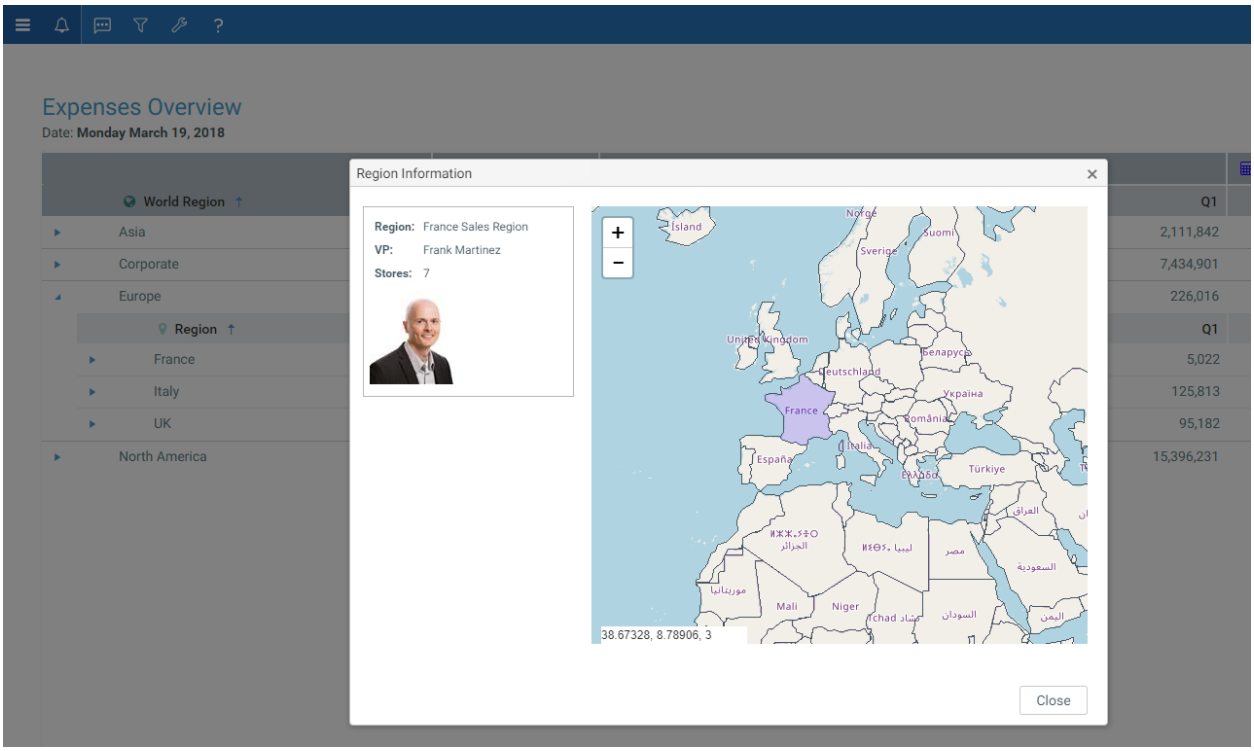
For example, imagine that you want to use the Dialog Panel command to open a dialog that shows more information about the current row. The child components of the Dialog Panel and the relevant data queries in the form can reference the `[ActionRowValue]`, so that the labels and data in the dialog are dynamically updated for the current row.

In this example, the RegionActions data source is assigned to the Region column in the data grid. When a user hovers over the region cell, the icon displays.



User hovers over region to see action icon

When the user clicks the icon, the ShowDialogPanel command is run, opening a dialog to show more information about the current row. The components and data queries in the dialog panel are referencing the `[ActionRowValue]` for Region (in this case, France) to show the relevant information.

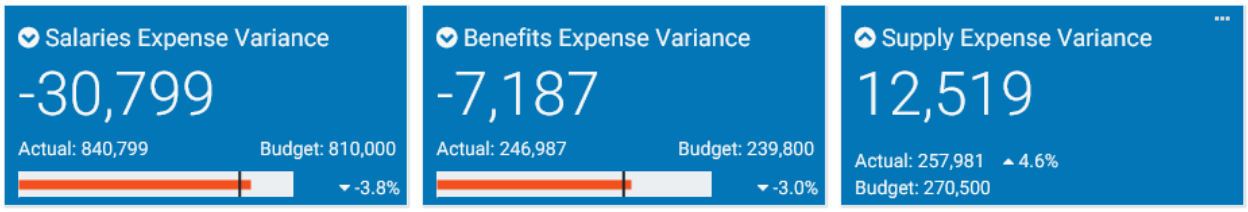


Dialog panel opens after clicking hover icon

Display KPIs within an Axiom form

Using the new KPI Panel component, you can display one or more key performance indicators (KPIs) in a series of KPI boxes. Once you configure the component to read KPI data from either a KPI table or a data source, this data is automatically formatted and positioned within each box, and the boxes are automatically positioned in the form based on the size of the overall panel.

KPIs consist of a primary value, several optional supporting values, an optional up/down indicator, and an optional bullet chart. The following screenshot shows an example of what the KPIs look like with and without the bullet chart.



Example KPI panels

The KPI can also include an optional action, which can be launched using the **...** button in the top right corner of the box (shown on the 3rd KPI box in the example above). The action can be used to open a designated web page or other Axiom form, or to perform a command.

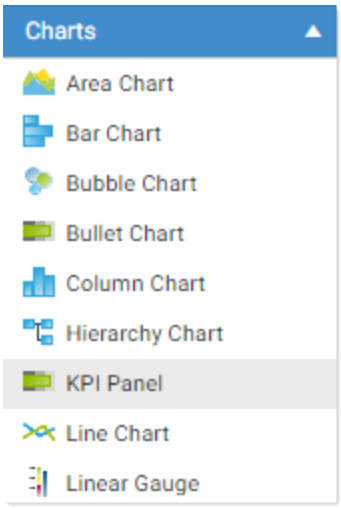
NOTE: The KPI Panel is considered a dashboard component in Axiom forms. Your license must include **Full Forms** functionality in order to create forms using the KPI Panel component.

The KPI Panel component in Axiom forms is the same underlying component used in web reports. Although the general properties are the same, the setup is different due to the differing environments. The Axiom forms environment also supports a few additional features that are not available in web reports, such as the ability to execute commands from the Command Library. However, the display of the KPIs is the same in both environments, providing a consistent user experience.

For more information on configuring a KPI Panel component in Axiom forms, see the following topic in Axiom Software Help: *KPI Panel component* (AX1670).

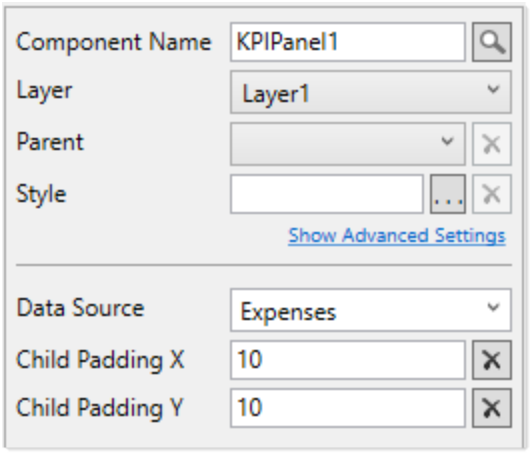
Adding a KPI Panel to the form canvas

The KPI Panel component is available in the Charts section of the component sidebar.



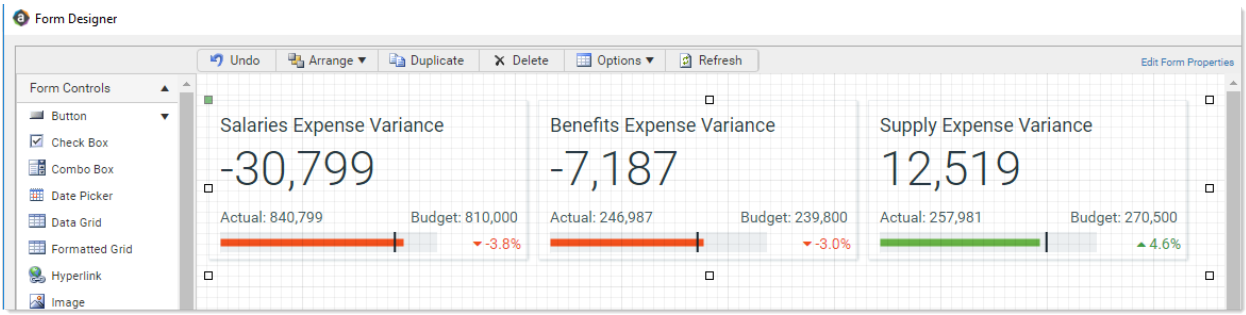
New KPI Panel component in Form Designer

When configuring the component, you must choose whether to use a KPI table or a data source in the spreadsheet to provide the KPI data. You can also optionally specify the X and Y padding, which is the vertical and horizontal padding between KPI boxes. The following screenshot shows an example configuration when using a Data Source:



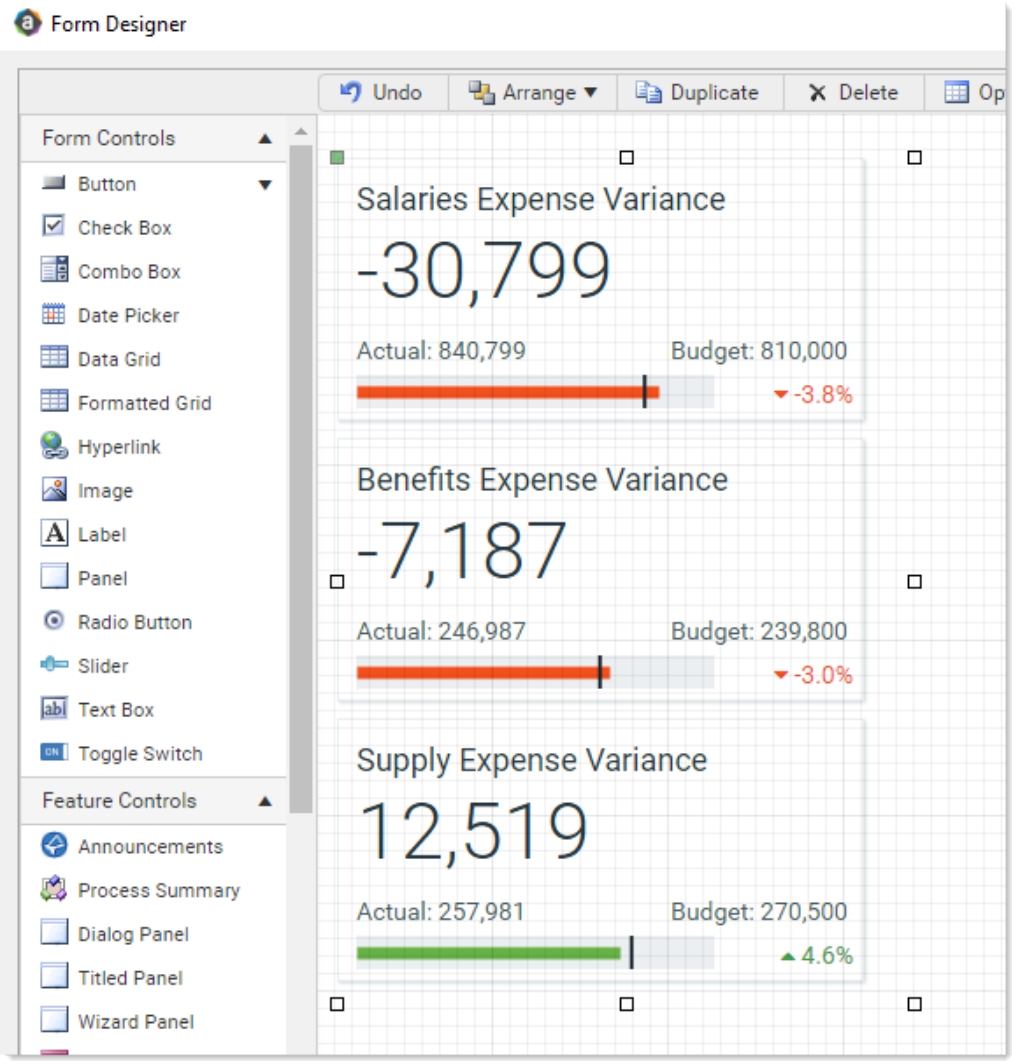
Example KPI Panel properties using a data source

The KPI Panel is a specialized panel component using flow behavior, so that the KPI boxes from the data source automatically position themselves on the form. This means that you must size the component as appropriate on the form canvas, so that the KPI boxes flow as desired. For example, if you want the KPI boxes to flow across the top of the form, then the KPI Panel component should be sized wide (perhaps docked horizontally) so that the boxes fit side-by-side.



Example KPI Panel sized wide to flow horizontally

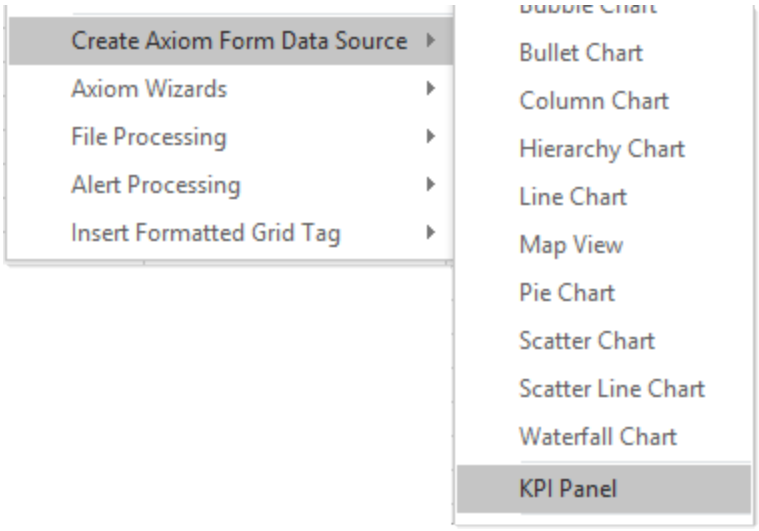
If instead you want the KPI boxes to stack vertically along the side of the form, the KPI Panel component should be sized tall (perhaps docked vertically) and thin so that the boxes stack on top of each other.



Example KPI Panel sized tall and thin to flow vertically

Using a KPI data source

If you configure the component to use a data source, you must create a KPISource data source in the spreadsheet. The data source defines the data for each KPI that you want to display in the panel. To create a KPISource data source, use the **Create Axiom Form Data Source** wizard on the right-click menu:



New data source in right-click wizard

Within the data source, each row defines a KPI "box" to display in the KPI Panel component. For each row, you can specify the following:

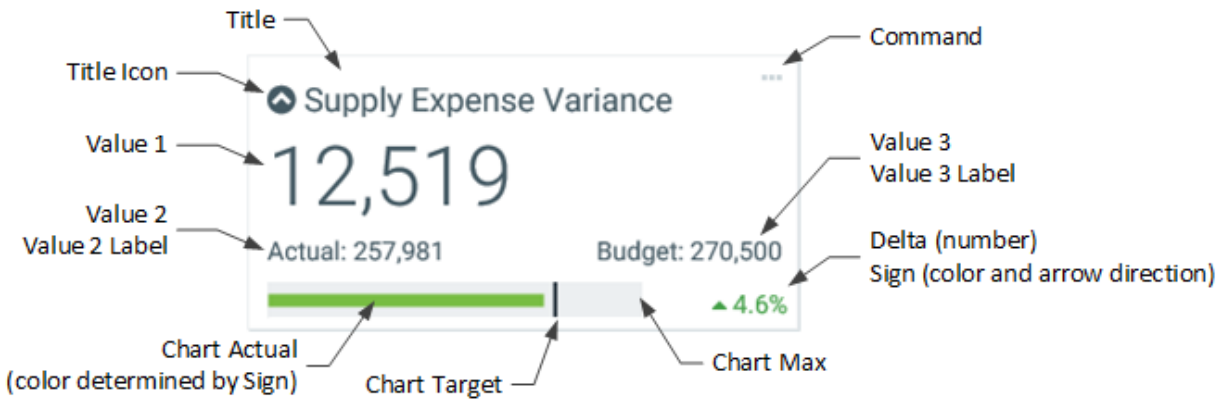
- The title of the KPI and an optional symbol to display in the title
- The primary KPI value to display in the box (Value1)
- Two optional secondary values relating to the KPI (Value2 and Value3), and labels for those values
- Target, actual, and max values for the optional bullet chart
- An optional delta value and corresponding up/down sign, to indicate the trend of the KPI
- An optional color code, to set the background color of the KPI box
- An optional command string, URL, or document shortcut, to perform an action by clicking the button on the KPI box

The following screenshot shows an example data source with three KPIs. The third KPI does not have a chart.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
2																
3		[KPISource;Expenses]	[Title]	[TitleIcon]	[Value1]	[Value2]	[Value2Label]	[Value3]	[Value3Label]	[ChartTarget]	[ChartActual]	[ChartMax]	[Delta]	[Sign]	[Style]	[Command]
4		[KPI]	Salaries Expense Variance	fa-chevron-circle-down	-30,799	840,799	Actual	810,000	Budget	810,000	840,799	1,000,000	-3.8%	Down	P7	
5		[KPI]	Benefits Expense Variance	fa-chevron-circle-down	-7,187	246,987	Actual	239,800	Budget	239,800	246,987	350,000	-3.0%	Down	P7	
6		[KPI]	Supply Expense Variance	fa-chevron-circle-up	12,519	257,981	Actual	270,500	Budget				4.6%	Up	P7	http://whqa1/A

Example KPISource data source

The following diagram shows how these data source values translate to the KPI box:



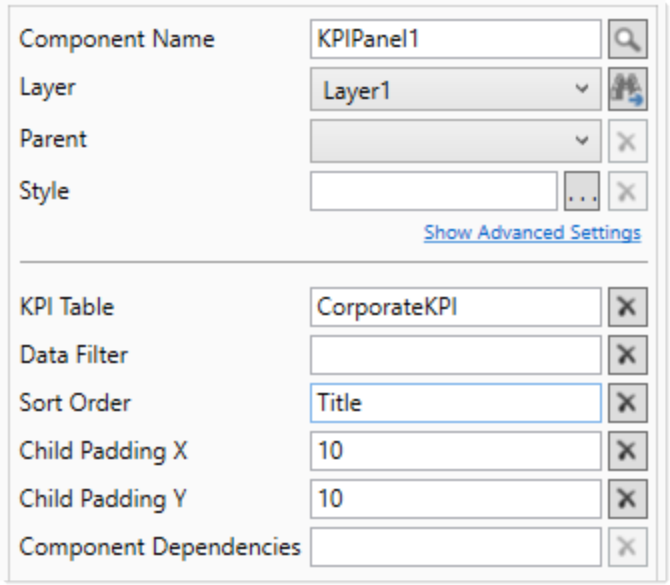
Once it has been added to the sheet, you can populate the data source manually, or you can use the Data Source Assistant. However, the Data Source Assistant does not currently provide any helper tools to populate the various properties. It simply provides an alternate way of viewing and editing the cells in the data source.

NOTE: The KPIDataSource only applies when using the KPI Panel component in an Axiom form. In web reports, KPI Panel components must use a KPI table.

Using a KPI table

As an alternative to using a data source, you can configure the component to use a KPI table. A KPI table is a special type of reference table, with a predefined data structure that matches the properties used by the KPI Panel component. For example, the table has columns of Title, TitleIcon, Value1, etc. For more information on creating a KPI table, see New KPI tables.

When using a KPI table, the component configuration looks like the following screenshot:



Example KPI Panel properties using a table

You can optionally specify a data filter to limit the KPIs shown in the component, as well as an optional sort order. When the component is rendered in the form, Axiom Software queries the specified table and then displays the KPI data from the table in the component.

When the KPI Panel uses a table, it uses special update behavior similar to the Data Grid component. The component does not re-query data from the table on each form update. If you need the component to refresh its data, you can use the **Component Dependencies** field to specify one or more components that the KPI Panel is dependent on. When those components trigger a form update, the KPI Panel will refresh its data from the database.

Performing an action from the KPI

The Command column in the KPISource data source (or in the KPI table) can be used to associate an action with the KPI box. If the Command column has content, a **...** button displays in the top right corner of the KPI box. Form users can click this button to perform the associated action.

The Command column can contain any of the following:

- A command string to perform a command from the Command Library. For example, you can use the new Show Form Dialog Panel command to launch a dialog panel to show more information about the current KPI.
- A URL to a web site or to an Axiom form. The web page will open within the current window.
- A document shortcut to an Axiom spreadsheet file or an Axiom form.

NOTE: Commands from the Command Library are only supported for use in Axiom forms. In web reports, KPI Panel components can only launch URLs or document shortcuts.

Formatting enhancements to Axiom forms and Formatted Grids

Version 2018.1 introduces a new default skin for Axiom forms, named Axiom2018. This skin features the following changes:

- Removal of themes as a styling concept, to simplify form formatting and setup
- New row and column style syntax that is easier to use and allows precise control over formatting
- Updated colors and other minor styling changes, to match the latest Axiom Software designs

The Axiom2018 skin is intended to replace the current Axiom skin, and it is the default skin for new forms going forward. Existing forms will retain their currently assigned skin on upgrade, so that you can update the form as needed before switching over to the Axiom2018 skin. Some forms can be migrated directly with minor (or no) adjustments, but forms that use Formatted Grids must be explicitly updated for the new styles as part of the migration.

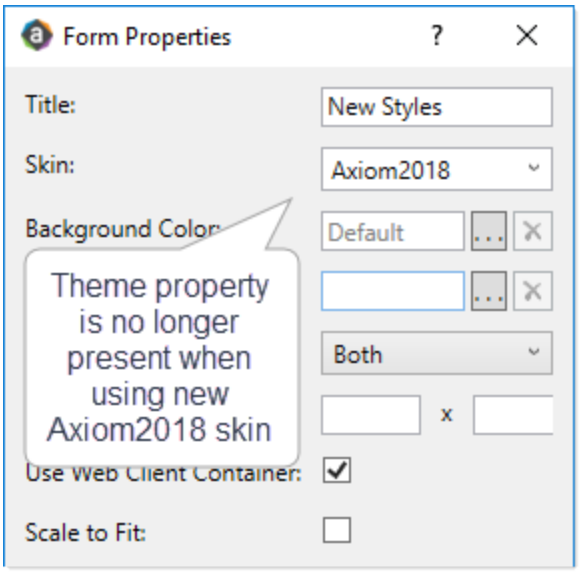
Note that it is no longer possible to leave the Skin property in a form blank to assume the system default skin (as defined in the WebClientSkin system configuration setting). The Skin property is now automatically populated with the system default skin when the form is created. For more information, see Upgrade behavior.

Removal of themes

In previous releases, Axiom forms had both a skin and a theme. The theme was intended to align with the general purpose of the form, such as whether it was a Report or a Worksheet. The theme controlled which styles were available to the components, and also could impact the formatting of "global" styles.

Themes have been deprecated going forward because user feedback indicated that it was difficult to tell which theme to use, and difficult to manage different styles for different themes (particularly with Formatted Grid components). In response to this feedback we have eliminated the themes and created a new style structure that directly defines formatting properties rather than using semantic styles. However, this new approach only applies to the new Axiom2018 skin. When working with older forms, you must continue to use themes until you migrate the form to the new skin.

When using the Axiom2018 skin, the **Theme** options are hidden from the Form Designer and the Form Assistant because they no longer apply. If you migrate an existing form to use Axiom2018, any existing theme assignments will be ignored.



Form Properties using new Axiom2018 skin

New row and column style syntax for Formatted Grids

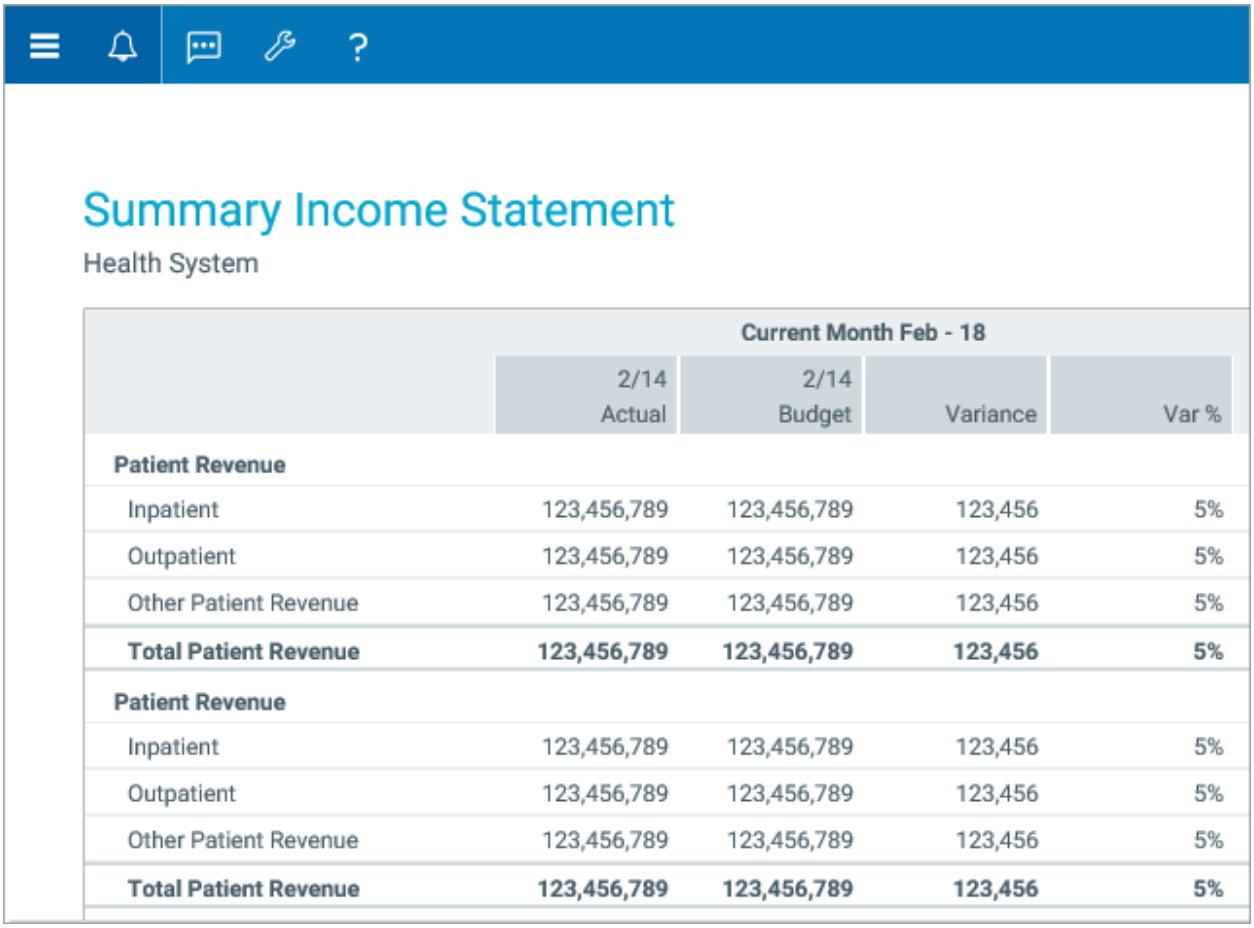
The Axiom2018 skin introduces new style syntax for rows and columns in Formatted Grid components. When using this syntax, you specify formatting properties such as font size, bold text, and background colors directly, instead of selecting from semantic style options (such as "subtitle"). The new syntax provides a more straightforward approach to formatting rows and columns in the grid.

For example, if you want a row to be bold with a top and bottom border, then you can apply a style such as **row, bold, brd-tb-2**. When reviewing the style, you know exactly what formatting will be applied to the row. The following screenshot shows a grid using the new tag syntax:

	C	D	E	F	G	H	I	J	K
2									
3	[Grid;NewTags]	[rowstyle]	[fixed]	[fixed]		[Column]	[Column]	[Column]	[Column]
4	[columnwidth]		10px	200px		80px-120px	80px-120px	80px-120px	80px-100px
5	[fixed]	row, bg-s2				[Format;Columns	Current Month Feb - 18		
6	[columnstyle]		col			colhead, right	colhead, right	colhead, right	colhead, right
7	[fixed]	row, bg-s2, row-height-18				2/14	2/14		
8	[fixed]	row, bg-s2, row-height-18, pad-b-4				Actual	Budget	Variance	Var %
9	[columnstyle]		col			col	col	col	col
10	[row]	row, bold, pad-t-4, row-height-18, pad-B-4		[Format;TargetCell=+1;]	Patient Revenue				
11	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Inpatient	123,456,789	123,456,789	123,456	5%
12	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Outpatient	123,456,789	123,456,789	123,456	5%
13	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Other Patient Rev	123,456,789	123,456,789	123,456	5%
14	[row]	row, bold, brd-tb-2		[Format;ColumnStyle=col, inc	Total Patient Reve	123,456,789	123,456,789	123,456	5%
15									
16	[row]	row, bold, pad-t-4, row-height-18, pad-B-4		[Format;TargetCell=+1;]	Patient Revenue				
17	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Inpatient	123,456,789	123,456,789	123,456	5%
18	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Outpatient	123,456,789	123,456,789	123,456	5%
19	[row]	row, brd-tb, brd-color-s2		[Format;ColumnStyle=col, inc	Other Patient Rev	123,456,789	123,456,789	123,456	5%
20	[row]	row, bold, brd-tb-2		[Format;ColumnStyle=col, inc	Total Patient Reve	123,456,789	123,456,789	123,456	5%

Example Grid data source using new style syntax

This grid renders as follows in the form:



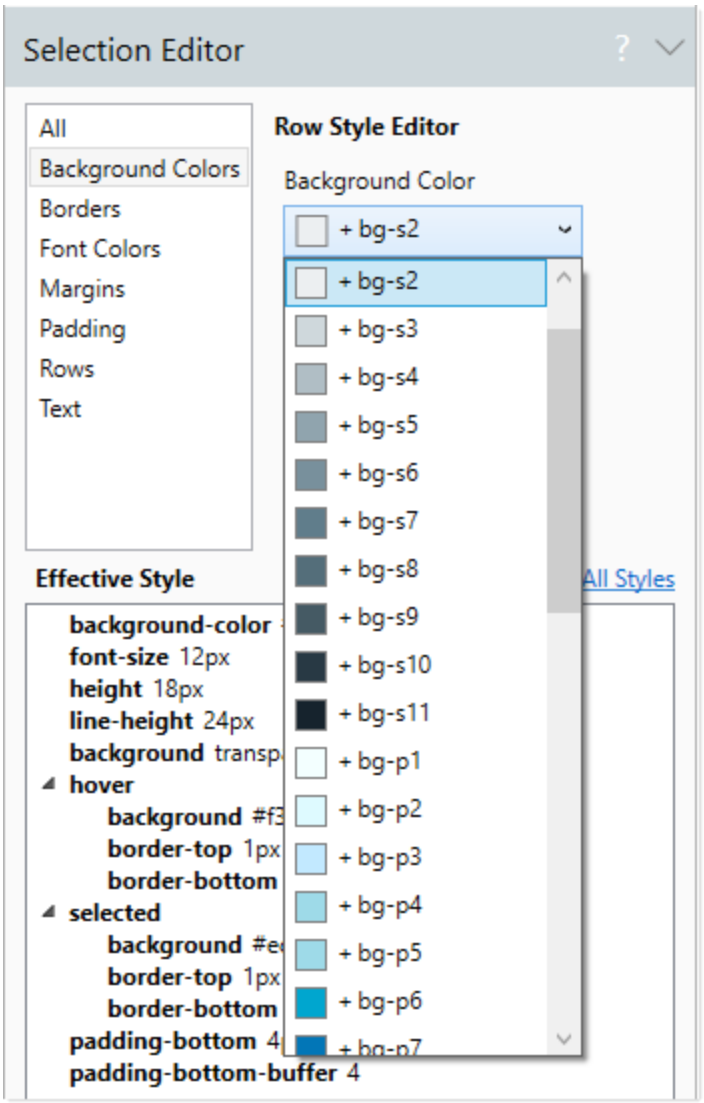
Summary Income Statement

Health System

	Current Month Feb - 18			
	2/14 Actual	2/14 Budget	Variance	Var %
Patient Revenue				
Inpatient	123,456,789	123,456,789	123,456	5%
Outpatient	123,456,789	123,456,789	123,456	5%
Other Patient Revenue	123,456,789	123,456,789	123,456	5%
Total Patient Revenue	**123,456,789**	**123,456,789**	**123,456**	**5%**
Patient Revenue				
Inpatient	123,456,789	123,456,789	123,456	5%
Outpatient	123,456,789	123,456,789	123,456	5%
Other Patient Revenue	123,456,789	123,456,789	123,456	5%
Total Patient Revenue	**123,456,789**	**123,456,789**	**123,456**	**5%**

Example formatting applied to grid based on styles

The new syntax provides many more options for row and column styles, and the Data Source Assistant has been updated to make it easier to review and apply these options. The Selection Editor section is now organized into categories such as Borders and Background Colors. Additionally, some style options include visual helpers, such as the color blocks that display next to the color styles.



Data Source Assistant updated for new row and column styles

A few predefined semantic styles are still available, though these styles can also now be created by combining the appropriate formatting styles. For example, **colhead** is still available to style column headers. For a full list of all the new style options, see the following topic in Axiom Software Help: *Thematic grid style reference* (AX1671).

When using the Axiom2018 skin, it is no longer necessary to specify a base style if the base style is the default **row** or **col** style. This means that the following style combinations are treated the same:

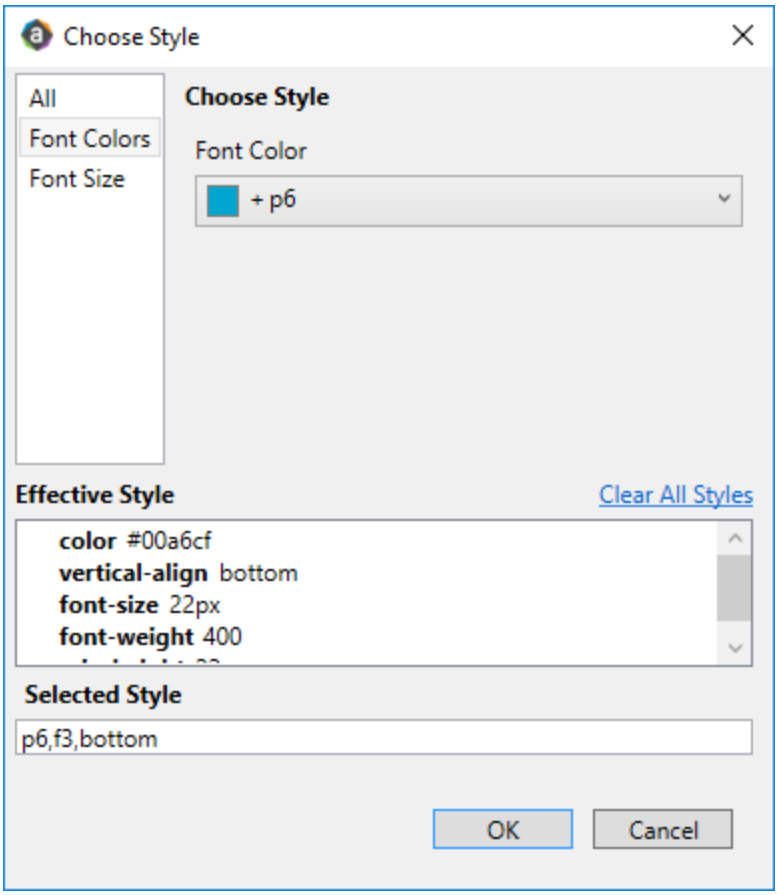
- **row,bold,brd-tb-2**: This combines the **row** base style plus two add-on styles.
- **bold,brd-tb-2**: This combines two add-on styles. The **row** base style is assumed if omitted, and if no other base style is listed.

Additionally, style names are no longer case-sensitive, when using any skin. Although the Data Source Assistant still shows all styles in lower-case, style names will be recognized regardless of case.

Related style changes

Although the primary component affected by these changes is the Formatted Grid component, a few other components were affected.

- **Label components**: The **h1-h5** styles are no longer available in the Axiom2018 skin. Instead, the font size and font color can now be specified directly, using the same style codes used by thematic grids.



Updated Choose Style dialog for Label components

- **Wizard Panel components**: The **wizardpanel-content** style is no longer available. Instead, **docked-to-container** is used. The Wizard theme is no longer necessary; the necessary design elements for the wizard are provided directly by the Axiom2018 skin.

Additionally, the color used in the Web Client toolbar (the Web Client Container) was updated to use a slightly different shade of blue in the Axiom2018 skin. If you have some forms that use the legacy Axiom skin and other forms that use the new Axiom2018 skin, you will notice a color change in the toolbar when switching from form to form.

Migrating an existing form to use the Axiom2018 skin

If you have existing forms that you want to migrate to the Axiom2018 skin, some manual adjustment in the form will likely be required after changing the skin in the **Form Properties**. Remember: when

upgrading, all of your existing forms will retain their currently assigned skin, so this only applies if you decide to manually change an existing form to use the new skin.

At minimum, you need to do the following after changing the skin:

- If the form uses thematic Formatted Grid components, you must update the row and column styles in the grids to use the new styles supported by the Axiom2018 skin. The previously assigned row and column styles will not be recognized by the Axiom2018 skin. Until the styles are updated, the grid will display as if it has no formatting.
- If the form uses h1-h5 styles for Label components, these styles are not supported by the Axiom2018 skin. You must update the labels to use the new font size and color styles.
- If the form uses a Wizard Panel component, you must change the style on the grid or panels that provide the wizard contents from **wizardpanel-content** to **docked-to-container**.

All other component-level styles from the legacy skin should be recognized by the Axiom2018 skin and continue to display as expected. However, it is a good idea to thoroughly review the form after changing the skin to detect any small formatting changes that may impact component display, and adjust the form accordingly.

The older the form, the greater the chance that additional manual adjustments will be required after changing the skin to Axiom2018. Older forms may use out-of-date designs that do not take advantage of the latest enhancements, and some of these out-of-date designs may impact the display. It is a good idea to review the entire form as part of the migration and identify any areas that could benefit from adopting new features.

Lastly, if the form is old enough that it is still using a spreadsheet-formatted grid instead of a thematic grid, the grid should be updated as part of the migration. For more information, see the following topic in Axiom Software Help: *Migrating spreadsheet-formatted grids to thematic grids* (AX1513).

Upgrade behavior

If you have existing forms where the **Skin** property is blank, these forms are updated as part of the upgrade to 2018.1, so that the forms use the same skin before and after the upgrade. The upgrade behavior depends on the **WebClientSkin** system configuration setting, which determines the default skin for your system.

- If the WebClientSkin setting for your system is blank (meaning the system default skin was Axiom), the forms are updated to use the Axiom skin.
- If the WebClientSkin setting for your system has a defined value, the forms are updated to use that defined value. For example, if the value was Uniform, the forms are updated to use Uniform.

Any forms that already have a defined value in the Skin property are left as is. None of your existing forms will use the Axiom2018 skin unless you manually migrate them after the upgrade.

When upgrading to version 2018.1, the WebClientSkin property is handled as follows:

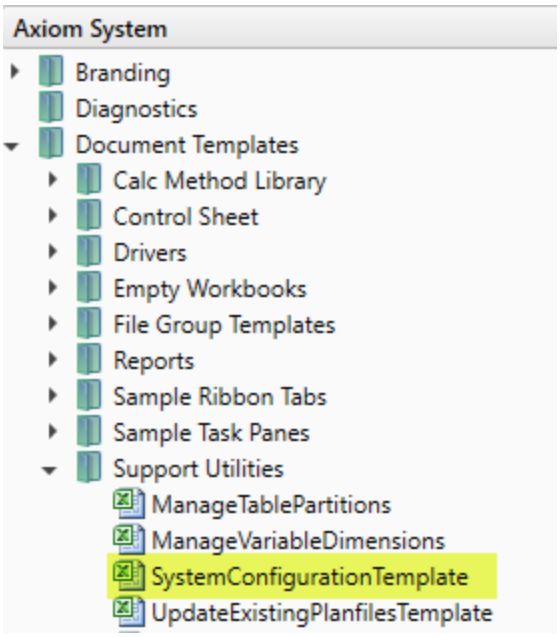
- If the WebClientSkin property for your system is blank, it is changed to Axiom2018. This change occurs after your existing forms have been updated. Any new forms created after the upgrade will

use Axiom2018 by default.

- If the WebClientSkin property for your system has a defined value, that value is left as is. Any new forms created after the upgrade will continue to use the legacy skin.

Before running the upgrade, you may want to check your current WebClientSkin value, so that you know what the upgrade behavior will be. And after running the upgrade, you may need to change the value to whatever skin you want to use for new forms going forward.

Kaufman Hall Software Support provides access to a utility that can be used to query and modify the system configuration settings. Only administrators can access this file and save changes to the configuration. This file is located in the Axiom file system at `\Axiom\Axiom System\Document Templates\Support Utilities\SystemConfigurationTemplate.xlsx`. It is recommended to copy the file into the Reports Library, where you can customize it for your use as needed.



Support template to modify system configuration settings

NOTE: Your system may already have a custom utility that is designed to query and edit the system configuration settings. If so, you can use that utility. The support utility is made available for customers who do not already have a utility.

The support utility is already designed with an Axiom query to read the current system configuration settings, and a Save Type 4 to save changes back.

- To view the current WebClientSkin setting, open the file and click **Refresh**. Scroll down until you find WebClientSkin, and check the value in the **Configured** column. For example, most systems will look like the following example before the upgrade, with both the Default and Configured as blank (meaning Axiom):

	D	E	F	G	H
7		SystemConfiguration - AQ Example			
8			Default	Configured	Modify
109		VBA_RunMacroDisabled	TRUE	TRUE	
110		WebClientSkin			
111		WebPageTabItemBackground	#FFFFFF	#FFFFFF	

- If you need to change the WebClientSkin, type the desired value into the Modify column, and then save. For example, most systems will look like the following example after the upgrade, with Default and Configured set to Axiom2018. If you need to continue using a different skin for new forms (such as Axiom or Uniform), you can type that skin name into the Modify column.

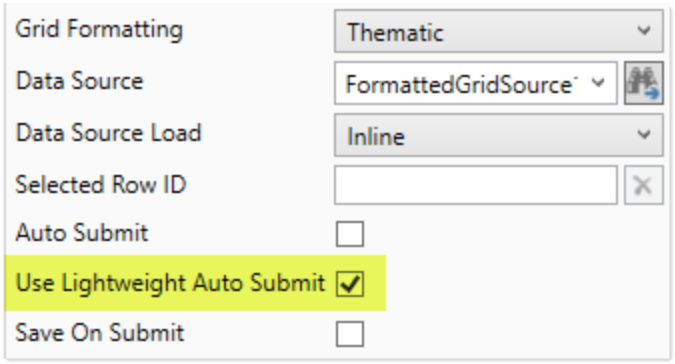
	D	E	F	G	H
7		SystemConfiguration - AQ Example			
8			Default	Configured	Modify
110		VBA_RunMacroDisabled	TRUE	TRUE	
111		WebClientSkin	Axiom2018	Axiom2018	Axiom
112		WebPageTabItemBackground	#FFFFFF	#FFFFFF	

If you need further assistance to view or edit the system configuration settings, contact Kaufman Hall Software Support.

Lightweight option to update Formatted Grids for user input

When a Formatted Grid component is being used to input numbers, often the grid also contains formulas that show the sum or other calculations based on these inputs. In previous releases, if you wanted the formulas to update immediately after the user inputs a number, the grid had to be set to auto submit and the full form update cycle would occur, forcing the entire grid to redraw.

A new option, **Use Lightweight Auto Submit**, is now available to enable targeted updates to the grid for this use case.



New submit option for Formatted Grid components

When enabled, the following behavior applies:

- When a user edits an unlocked cell or a text box, only that new value is submitted back to the form source document on the server.

- The source document is calculated so that any formulas referencing the changed cell are updated. No data refresh occurs, just a calculation.
- If any of the following cell types in the grid have modified values after the calculation, the form web page is updated to display these new values: simple locked or unlocked cells, cells with Format tags, and cells with TextArea tags.

No other form values are submitted, and no other form components are updated. The regular form update process does not occur.

NOTE: This option only applies to thematic grids. Spreadsheet-formatted grids cannot use this option.

The lightweight option can be enabled by itself or in conjunction with auto submit. If enabled, the lightweight option overrides the normal auto-submit behavior for the affected cells, but all other cells in the grid can continue to use the regular auto-submit behavior or not as desired.

Additional Axiom form enhancements

Control row height for thematic grids

A new tag of `[RowHeight]` has been added to the Grid data source, to provide more options to control the row height of thematic formatted grids. Row heights in thematic grids are now set as follows:

- Each row uses the height set by the row style by default. All base row styles include a specified row height (including the default row style that is used when no base style is specified). Additionally, you can apply add-on styles to set the row height, such as **row-height-18**.
- If a row height is set in the `[RowHeight]` column of the Grid data source, this row height overrides the style. So if the style sets a row height of 12 pixels, but you enter 30 pixels into the `[RowHeight]` column, then the row height for that row is 30 pixels.

  Generally speaking, this column is only intended for cases where you need to override the row style or set the row to a height that is not supported by the available styles. It is not necessary to enter a row height for every row.

This new tag is added by default when creating a new Grid data source. It can be left blank or removed from the data source if not needed. The tag has no effect on spreadsheet-formatted grids.

The following entries are valid in the `[RowHeight]` column:

- **Value in pixels**: For example, you can enter `30` or `30px` to set the height to 30 pixels.
- **Value in percent**: Percent values are applied to the height defined by the row style. For example, if the row style is 12 pixels, you can enter `200%` to size the row twice the height of the row style (24 pixels).
- **Auto**: You can enter the keyword `auto` to indicate that the row uses the height defined in the row style. This is for cases where you want to populate all cells of the `[RowHeight]` column for clarity, but you want some rows to use the height defined in the style.
- **Blank**: The row uses the height defined in the row style.

Command enhancements

- The **Export Grid** command can now be used with both types of grid components: Formatted Grids and Data Grids. For Data Grid components, this provides an alternative to using the built-in **Enable Excel Export** option in the grid properties. When using the built-in option, the export button is always positioned in the same place, at the top right corner of the grid. You can use the command instead if you want to position the button elsewhere.
- A new **Show Form Dialog Panel** command is available, so that Dialog Panel components can be launched from non-button components that support commands, such as the KPI Panel component or when using icon actions in a Data Grid component. The command uses a single parameter of the Dialog Panel component name. The behavior of the command is the same as when using the Dialog Panel Action button behavior.
- When launching the Filter Wizard from an Axiom form using the Filter Wizard command, the **Preview** box is now editable. In previous versions, you had to move the filter from the Preview box to the Filter box before it became editable.

Miscellaneous enhancements

When using a Form Help component to display custom help text for a form, the current help code now displays in small font in the top right of the help panel.

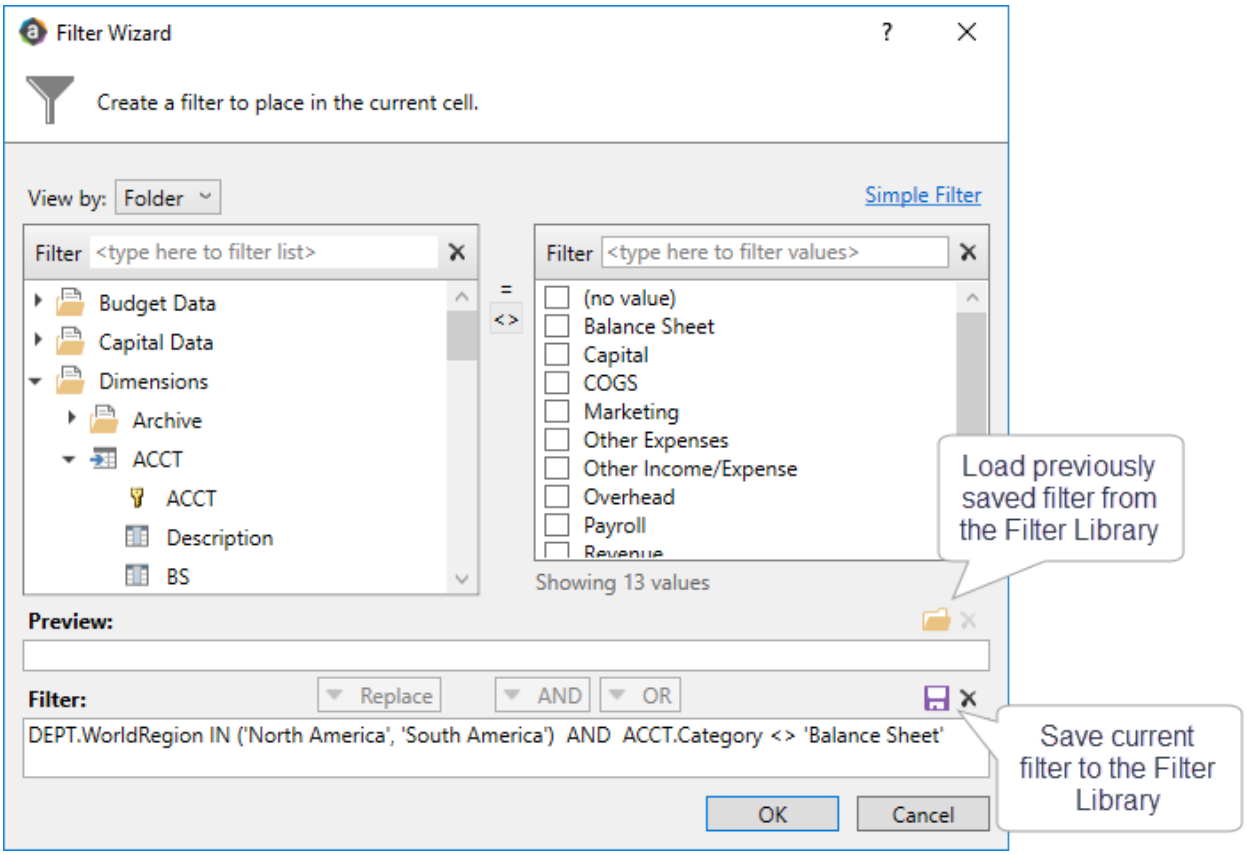
Filters

This release introduces a set of related features to support saving filters for reuse throughout the system.

Save filters to the Filter Library for reuse

You can now save filter criteria statements to the new Filter Library, so that commonly-used filters are stored in a centralized location. Once a filter is saved to the Filter Library, you can reuse it in various areas of the software, such as in reports or security.

The way that filters are saved and reused is via the Filter Wizard dialog. Once you have created a filter in the Filter Wizard that you would like to store for reuse, you can save it to the library from the dialog. To reuse a filter, you can open the Filter Wizard and load a saved filter from the library, then apply that filter to the current context of the Filter Wizard. These features are only available in Advanced Filter view.



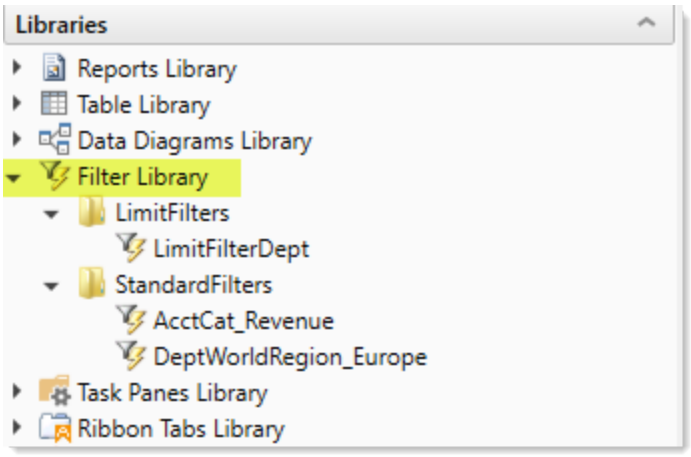
Updated Filter Wizard with save and load buttons

For example:

- If you are in a report that uses the new AdvancedFilter refresh variable, you can use that Filter Wizard dialog to load a saved filter instead of creating a new one. When you apply the refresh variables, that filter is now applied to the report.
- If you are configuring security for plan files in a particular file group, you can launch the Filter Wizard to load a saved filter, and then apply that filter to the file group.

NOTE: When you load a filter, the filter criteria statement from the Filter Library is copied into the Filter Wizard so that you can use it in your current task. The filter loaded in the Filter Wizard is no longer associated with the saved filter in the Filter Library. The Filter Library does not enable centralized maintenance of filters; it is just a convenience so that you do not have to re-create the filter from scratch every time that you want to use it.

You can configure user access to the Filter Library like other libraries in Axiom Software. Users can have read-only access to certain filters in order to load and use the filters, or they can have read/write access to edit filters and save new filters.



Example Filter Library in Axiom Explorer

There are two types of filters that you can save to the Filter Library:

- Standard filter criteria statements (for use throughout the system)
- Limit query statements (for use in the advanced setting of Axiom queries: **Limit query data based on another table**)

Standard filter criteria statements can be created and saved using any Filter Wizard in the application. Because of the specialized use of limit query statements, limit query statements can only be created and saved using certain features: AdvancedFilter refresh variables, ShowFilterWizardDialog function, and the Filter Wizard command for Axiom forms. In this case, you must specifically configure the feature to create a limit query statement, and set up an Axiom query in the file to use the resulting statement.

New refresh variable for advanced filtering

The new AdvancedFilter refresh variable can be used to prompt users to create a filter criteria statement using the Filter Wizard. The resulting filter can then be used to filter the report data.

The AdvancedFilter refresh variable can create two different kinds of filters, depending on the variable configuration:

- Standard filter criteria statements, for use in any report filter such as sheet filters and Axiom query data filters
- Limit query statements, for use in the advanced setting of Axiom queries: **Limit query data based on another table**

Variable setup

The AdvancedFilter refresh variable is defined like other refresh variables in the `[RefreshVariables]` data source. The setup for the variable depends on whether you want to create a standard filter criteria statement or a limit query statement.

- The `[VariableType]` property must be set to **AdvancedFilter**.
- A new property, `[PrimaryTable]`, is available to define the primary table for the Filter Wizard. Instead of showing all tables, the Filter Wizard only shows tables that can be used to filter the primary table. This streamlines the list of tables and helps to ensure that the resulting filter is valid for use in your intended context.
- A new property, `[LimitColumn]`, is available to define the limit column for a limit query statement. If this property is used, the Filter Wizard launches in a special mode to create a limit query statement. If blank or omitted, the Filter Wizard launches in normal mode to create a standard filter criteria statement based on the primary table.

  If the limit column uses fully qualified Table.Column syntax, then the primary table can be left blank because the primary table is the limit column table. If only a column name is specified as the limit column, then the primary table must be defined.
- If you are using the variable to create a standard filter criteria statement, you can optionally use the existing `[UseAsQuickFilter]` property to apply the resulting filter as a Quick Filter. In this case, it is not required to set up the report to reference the filter because it will be applied automatically.

When inserting a new `[RefreshVariables]` data source into a file, the data source now includes the new property columns for the AdvancedFilter variable. If you have an existing data source, you can manually add the columns or you can use the Data Source Assistant.

For more information on defining a AdvancedFilter variable, see the following topic in Axiom Software Help: *AdvancedFilter refresh variables* (AX1680).

The AdvancedFilter variable can be used in the Desktop Client and the Web Client, for spreadsheet Axiom files and Axiom forms. It is not currently available in web reports.

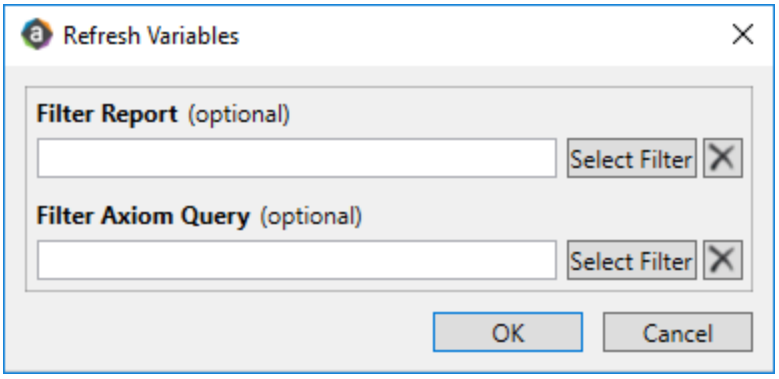
Variable behavior

The following screenshot shows example AdvancedFilter variables defined in the `[RefreshVariables]` data source.

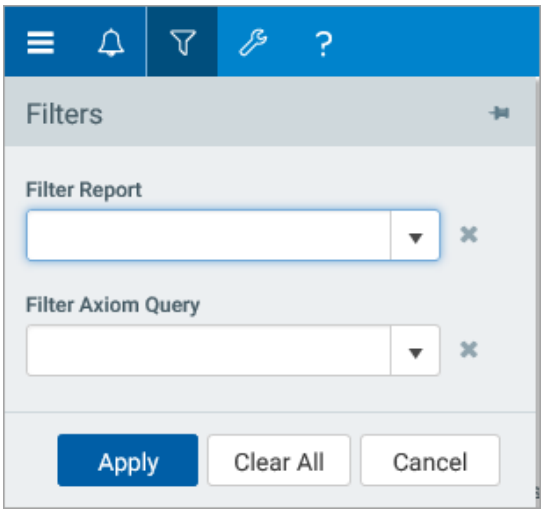
	B	C	D	E	F	G	H	I	J	K	L
19											
20		[RefreshVariables]	[Name]	[DisplayName]	[VariableType]	[IsRequired]	[IsEnabled]	[SelectedValue]	[PrimaryTable]	[LimitColumn]	[UseAsQuickFilter]
21		[Variable]	Standard	Filter Report	AdvancedFilter	False	True		GL2018		TRUE
22		[Variable]	Limit	Filter Axiom Query	AdvancedFilter	False	True			Dept.Dept	

The first variable, Standard, is being used to define a standard filter criteria statement, and then apply it as a Quick Filter. The second variable, Limit, is being used to create a limit query statement.

When the user refreshes the file (Desktop Client) or opens the Filters panel (Web Client), both variables display as a box with a button to open the Filter Wizard.

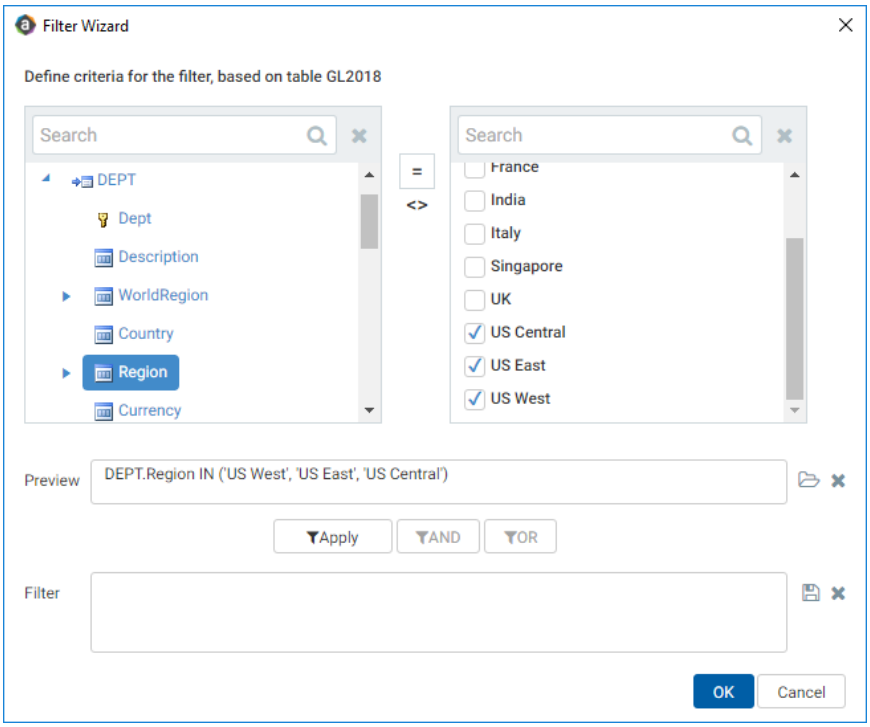


Desktop Client example



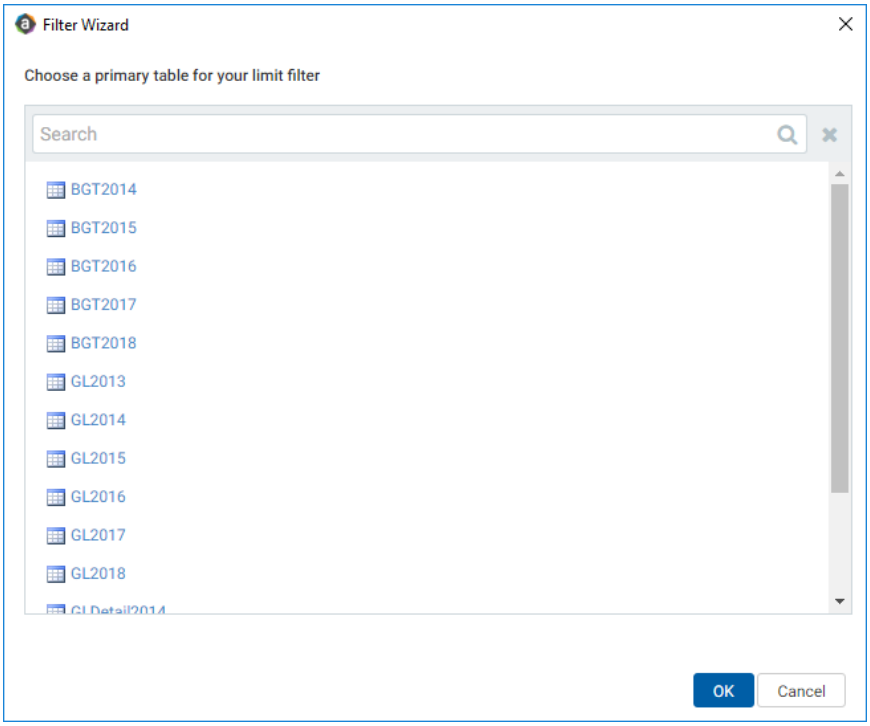
Web Client example

When the user interacts with the Standard variable, they see the typical Filter Wizard "advanced view," but with the list of tables filtered by the primary table. They can create a filter criteria statement as normal and then apply it to the form. This dialog is the same in the Desktop Client and the Web Client.



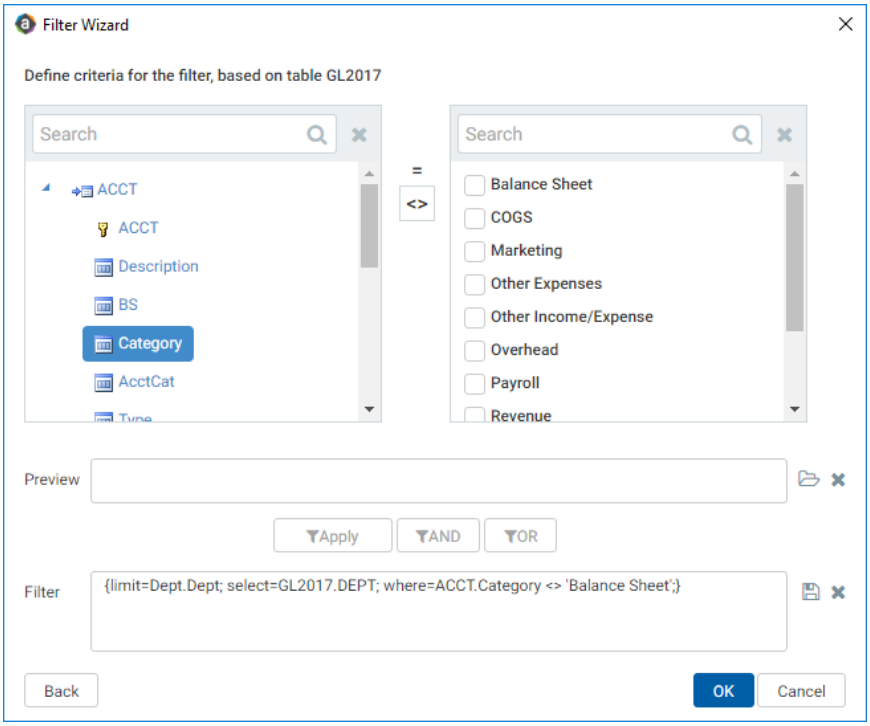
Filter Wizard to create standard filter

When the user interacts with the Limit variable, they see a special two-step wizard to create a limit query statement. In the first screen, the user selects a table to use to limit the data in the Axiom query.



First screen to create limit query statement

In the second screen, the user defines a filter criteria statement to apply to that table. Once the filter criteria statement has been created, all of the information is automatically combined into a limit query statement that can be used in the Axiom query setting **Limit query data based on another table**. The example screenshot below shows the automatically created statement after the user has created the filter.



Second screen to create limit query statement

Double-click cell to open Filter Wizard dialog

A new Axiom function is available to launch the Filter Wizard by double-clicking a cell in a spreadsheet Axiom file. The function can be used to create two different kinds of filters, depending on the function parameters:

- Standard filter criteria statements, for use anywhere in the system
- Limit query statements, for use in the advanced setting of Axiom queries: **Limit query data based on another table**

This function launches the same Filter Wizard dialog as described for the AdvancedFilter refresh variable.

Function syntax

The function uses the following syntax:

```
ShowFilterWizardDialog("DisplayText", "TargetCell", "PrimaryTableName",
"LimitColumn", "DialogTitle")
```

Parameter	Description
DisplayText	The text to display in the cell.
TargetCell	The target cell in which to place the filter. You can enter any of the following: • A standard cell reference, with or without the sheet name. For example: "B2" or "Variables!B2". • A column reference, such as "H" or "AZ". The filter will be placed in that column, within the same row as the cell with the function. • A relative column reference, using the format +N or -N. The filter will be placed N columns to the right or left, within the same row as the function. For example, "+1" places the filter one column to the right of the cell with the function, and "-1" places the filter one column to the left of the cell with the function.
PrimaryTableName	Optional. Complete this parameter as follows, depending on the objective: • **Create a filter using any table**: Omit both the PrimaryTableName and the LimitColumn if you want to open the Filter Wizard dialog showing all available tables. The user can create a filter using any table. • **Create a filter based on a primary table**: Specify a PrimaryTableName if you want to open the Filter Wizard dialog and only show tables that are valid to filter the primary table. The user can create a filter based on the primary table or on a lookup reference table. This is the recommended approach if the file is designed to apply the filter to a specific data query, so that you can be sure the filter will be valid in the context of the query. • **Create a limit query statement**: Specify a fully qualified LimitColumn name if you want to use the Filter Wizard to create a limit query statement for an Axiom query. In this case you can leave the PrimaryTableName blank.
LimitColumn	Optional. Complete this parameter if you want to use the Filter Wizard to create a limit query statement for an Axiom query. Otherwise, leave this parameter blank to create a standard filter criteria statement. This works the same way as described for the AdvancedFilter refresh variable.
DialogTitle	Optional. Specifies custom title text for the Filter Wizard dialog. If omitted, the standard title of "Filter Wizard" is used.

Examples

```
=ShowFilterWizardDialog("Filter Data","A10")
```

This example allows the user to create a filter using any customer table in the system. The filter will be placed in cell A10 on the current sheet.

```
=ShowFilterWizardDialog("Filter Data","H","GL2018")
```

This example allows the user to create a filter that is valid for use against the GL2018 primary table. The wizard only shows the GL2018 table as well as any lookup reference tables (such as Dept and Acct). The

filter will be placed in column H, in the same row as the cell containing the function.

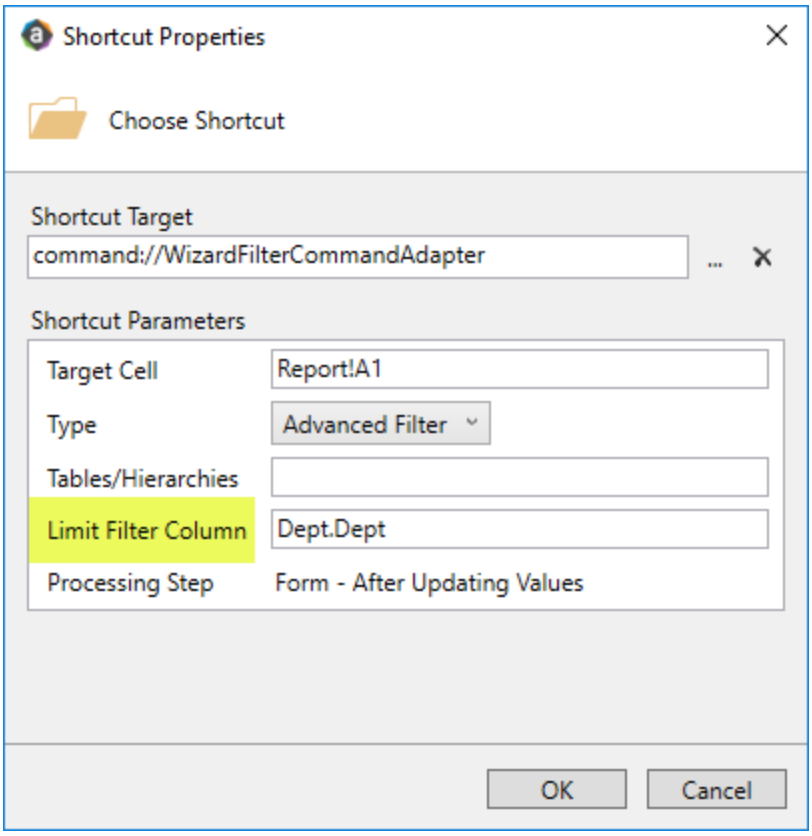
```
=ShowFilterWizardDialog("Limit Query","A10",,"Dept.Dept","Limit Query")
```

This example allows the user to create a limit query statement for an Axiom query, to limit the data in the Dept column based on another table. This configuration launches the same two-step wizard as described for the AdvancedFilter refresh variable, when creating a limit query.

Additional filtering enhancements

Create limit query statements using the Filter Wizard command

The Filter Wizard command for Axiom forms now supports the ability to create limit query statements, in addition to its existing functionality to create standard filter criteria statements. A new optional field, **Limit Filter Column**, can be used to configure the command to create a limit query statement. The behavior of the Filter Wizard dialog is the same as when using the AdvancedFilter refresh variable to create a limit query statement.



New Limit Filter Column option

Reference tables in the Filter Wizard

When the tables shown in the Filter Wizard are limited based on a reference table, any relevant lookup tables are now only shown underneath the primary reference table. In previous versions, lookup tables

were shown by themselves in addition to underneath the primary reference table, which means it was possible to generate a filter using a lookup table that would not have been valid in all contexts. While it is still possible to construct a filter using lookup tables, the new approach forces the filter to start with the primary reference table, which is the desired syntax.

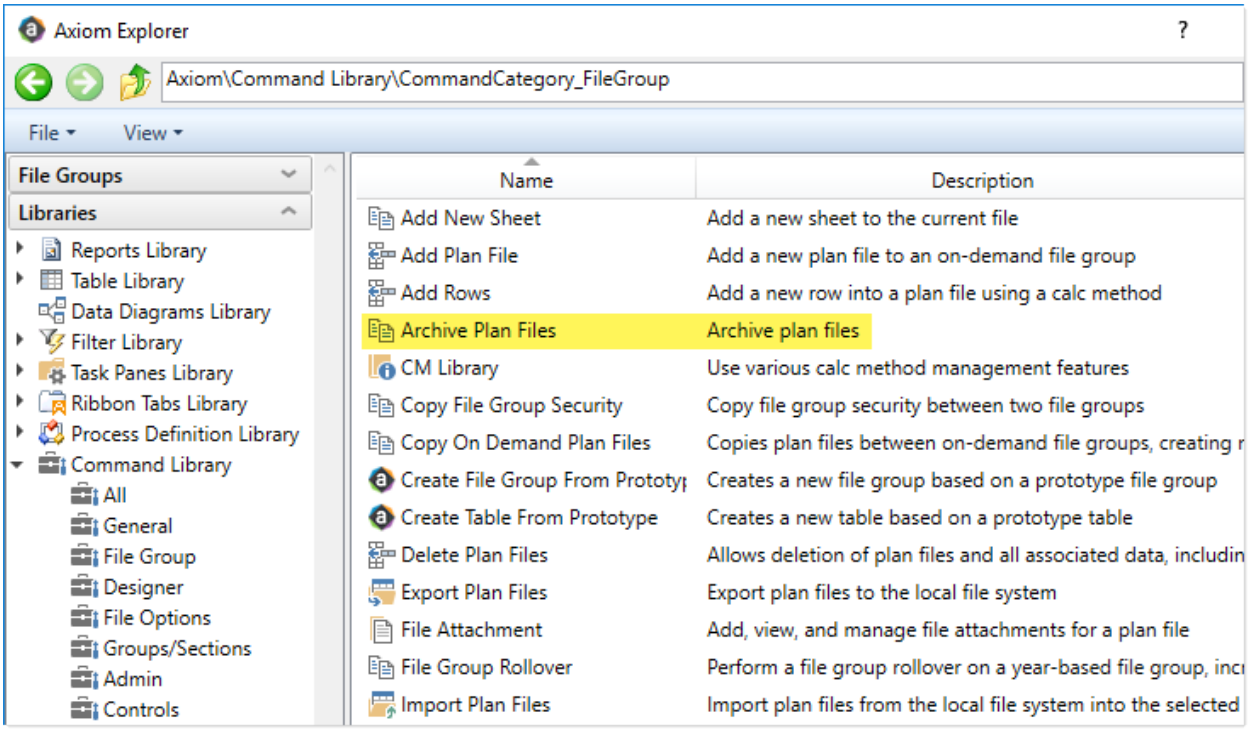
File groups

This section details the new features and enhancements made to file groups.

Archive plan files

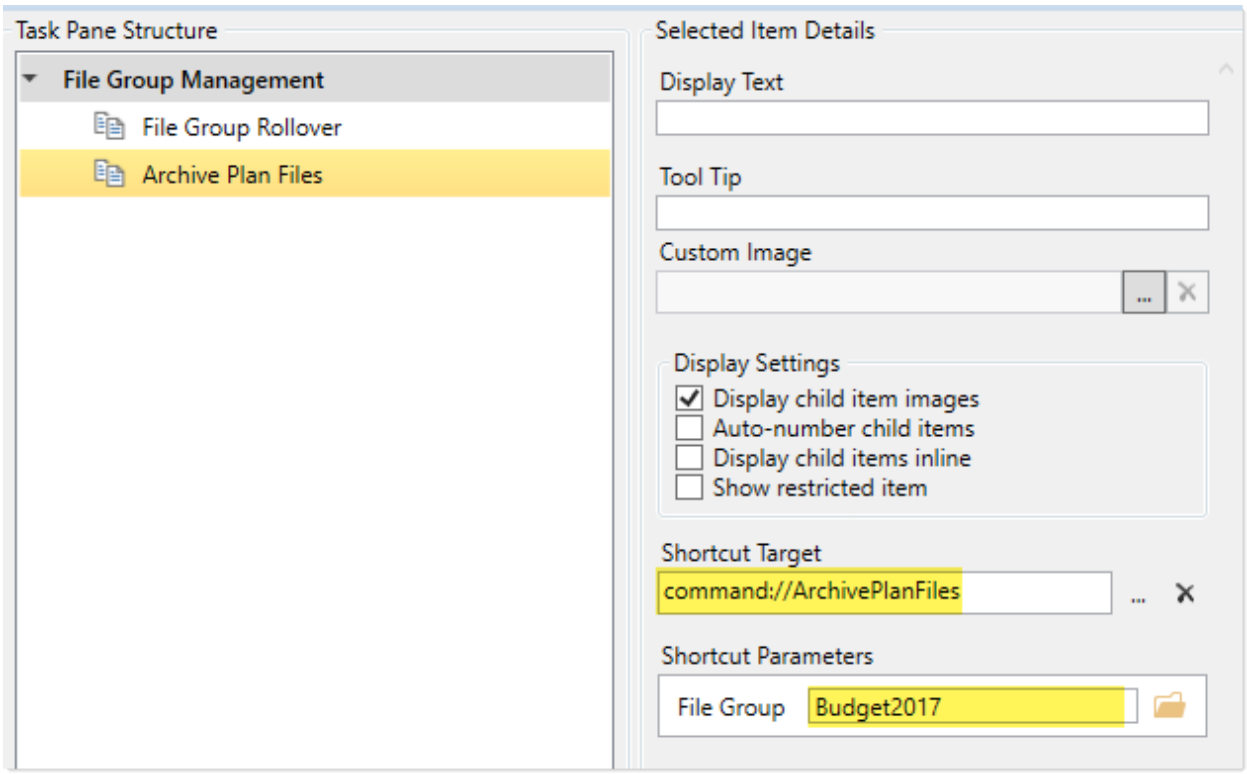
A new feature is available to convert "live" plan files in a file group to static snapshots, for viewing purposes only. This feature is intended to be used in cases where planning is finished for the file group, but you still want users to be able to view the finalized plan files. However, you do not want the plan files to be updated with new data or save data to the database.

This feature is provided using a new command in the Command Library named **Archive Plan Files**. If you want to use the feature, you can add the command to a custom task pane or custom ribbon tab. The feature is not available on the default file group menu.



New command in Command Library

The only parameter to configure on the command is to specify the file group where you want to archive plan files. You can select any file group that uses standard spreadsheet plan files. The archive process will not work on file groups that use virtual plan files or form-enabled plan files.



Example configured command in custom task pane

When the command is executed, all plan files in the file group are processed and converted into static files. All formulas are converted to values, all control sheets are removed, and tags for various processes such as action codes are disabled. The end result is a file that can be opened and viewed as is, but is not updated with data and does not save data to the database. For more details on what occurs during the archive process, see the following topic in Axiom Software Help: *Archive Plan Files command* (AX1719).

Before the plan files are converted, a restore point is created so that the original plan files can be restored if the command is executed in error. However, except for this temporary safeguard, the archive process is not reversible. Plan files should only be archived if you know they are no longer needed for active planning.

Additional file group enhancements

The following additional enhancements were made to file groups.

File group scenario subfolders for tables

When creating a file group scenario, tables created for the scenario are now placed in scenario-specific subfolders. For example, if BGT2019 is located in `\Budget\Data`, then BGT2019_v1 is located in `\Budget\Data\V1` (where V1 is the scenario name). This change does not have any impact on scenario functionality—it is simply for organizational purposes, to make it easier to locate the tables in the Table Library.

File Group Rollover enhancements

The following enhancements were made to the existing File Group Rollover command. This command can be used in a custom task pane or ribbon tab to help automate a rollover process.

- A confirmation message now displays before executing the File Group Rollover command, so that the user has the opportunity to continue or cancel before executing the rollover.
- The File Group Rollover command now supports the option to update a file group alias as part of the rollover. Once the new file group is created, the specified alias is updated to point to the new file group.
- The File Group Rollover command now supports executing a specified utility after creating the new file group. If a report utility is specified in the **Copy Data Utility** property, that utility will be executed via Scheduler once the new file group has been created. Active Axiom queries are run in the utility file and a save-to-database is executed.

Miscellaneous enhancements

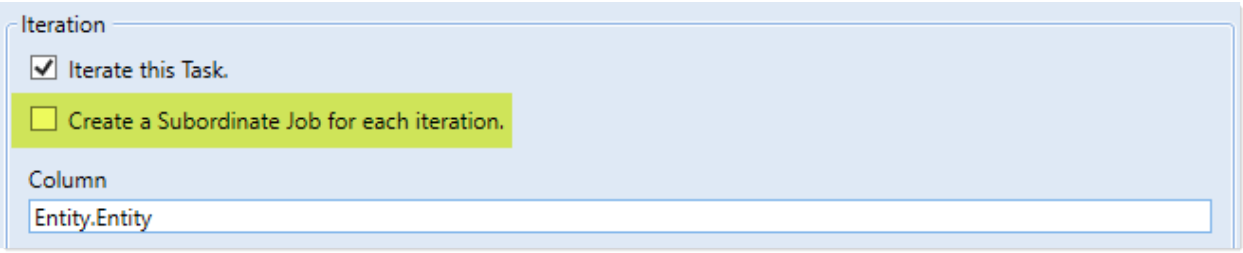
- In the Apply Calc Method Changes feature, the text for the default update behavior has been changed from **Replace entire calc method** to **Update entire calc method**. This is a display-only change that does not impact functionality.
- When Process Plan Files or Apply Calc Method Changes sends a system-generated notification of results, the notification title now includes the system name.

Scheduler

A couple of enhancements were made to existing features in Scheduler.

Execute iterations concurrently

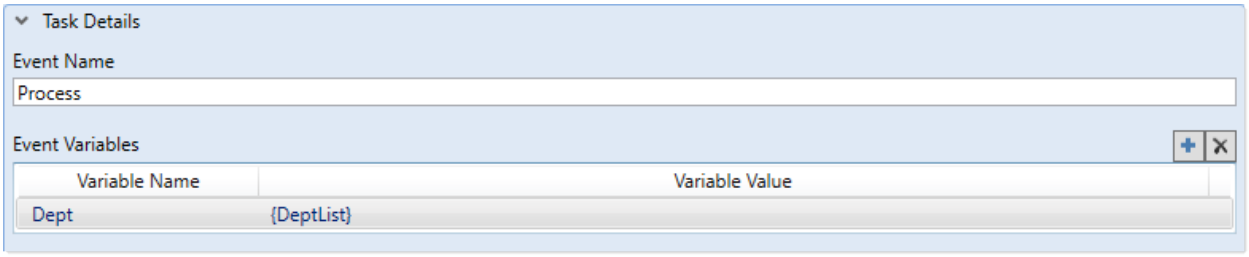
If you are using iterative task processing and your iterations do not need to be executed in a specific order, you can now process them concurrently by enabling **Create Subordinate Job for each iteration**. When this is enabled, each iteration is executed as a separate sub-job instead of as a series of sequential tasks. Multiple sub-jobs are processed concurrently, which improves the overall performance.



Pass variables when using Raise Event

When using the Raise Event Scheduler task to trigger another Scheduler job for execution, you can now define variables in the Raise Event task and pass those variables to the triggered jobs.

This job has a new section, **Event Variables**, where you can define one or more variable names and values. Job variables can be used in the value.



When the Raise Event task triggers one or more jobs to be executed based on the event name, the event variables and their values are passed into the triggered jobs. If the jobs are configured to use these variables, the variables can impact how the jobs are executed.

Tables

This section details the new features and enhancements made to table management.

New KPI tables

KPI tables are special reference tables that store key performance indicators (KPIs) for your organization. These KPIs can be referenced in web reports and in Axiom forms using the new KPI Panel component.

KPI tables have a predefined structure that maps to the properties used by the KPI Panel component. Once you assign the table to a KPI Panel component, the component automatically formats and positions the KPI data in the table into a series of KPI boxes. You can apply an optional filter to limit the KPIs shown in a particular component.

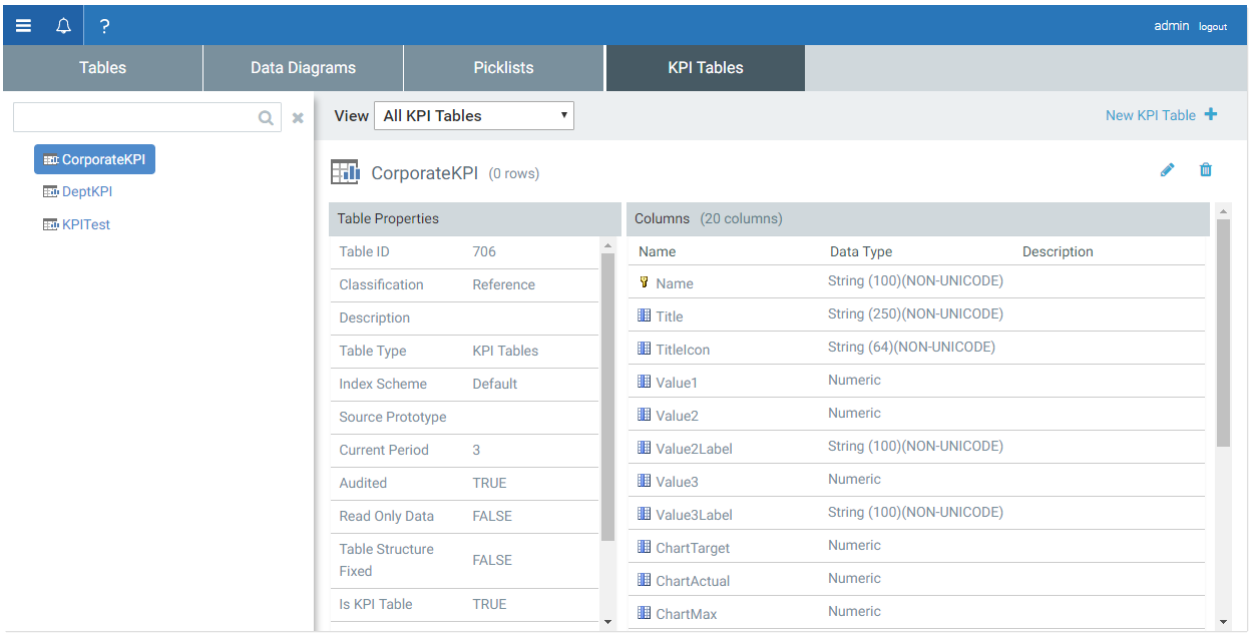
For example, you might create a save-to-database report that calculates the KPI data and saves it to a designated KPI table, and schedule the report to run weekly, monthly, or quarterly. The updated KPI data would then automatically display in any KPI Panel that references the table.

Creating KPI tables

The new KPI Tables area of the Web Client Table Manager provides an easy way to create new KPI tables. To access this area, click the Table Manager icon on the Axiom Software launch page, or go directly to the following URL:

```
<URL_to_Axiom>/TableManager
```

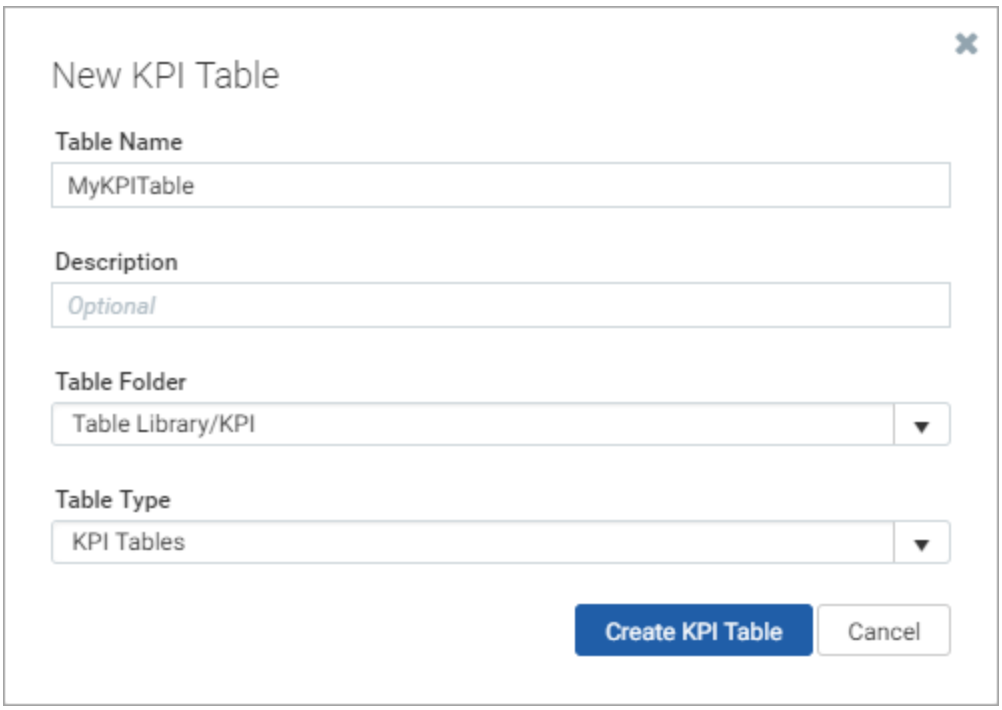
On the Table Manager page, select the **KPI Tables** tab at the top of the screen.



Example KPI Tables area of Web Client Table Manager

Using this page, you can browse existing KPI tables and create new ones. Only administrators and users with the **Administer Tables** permission can create new KPI Tables.

To create a new table, click **New KPI Table** at the top right of the page. In the **New KPI Table** dialog, complete the basic table properties such as the table name, folder, and optional table type. Note that KPI tables can belong to any reference table type.



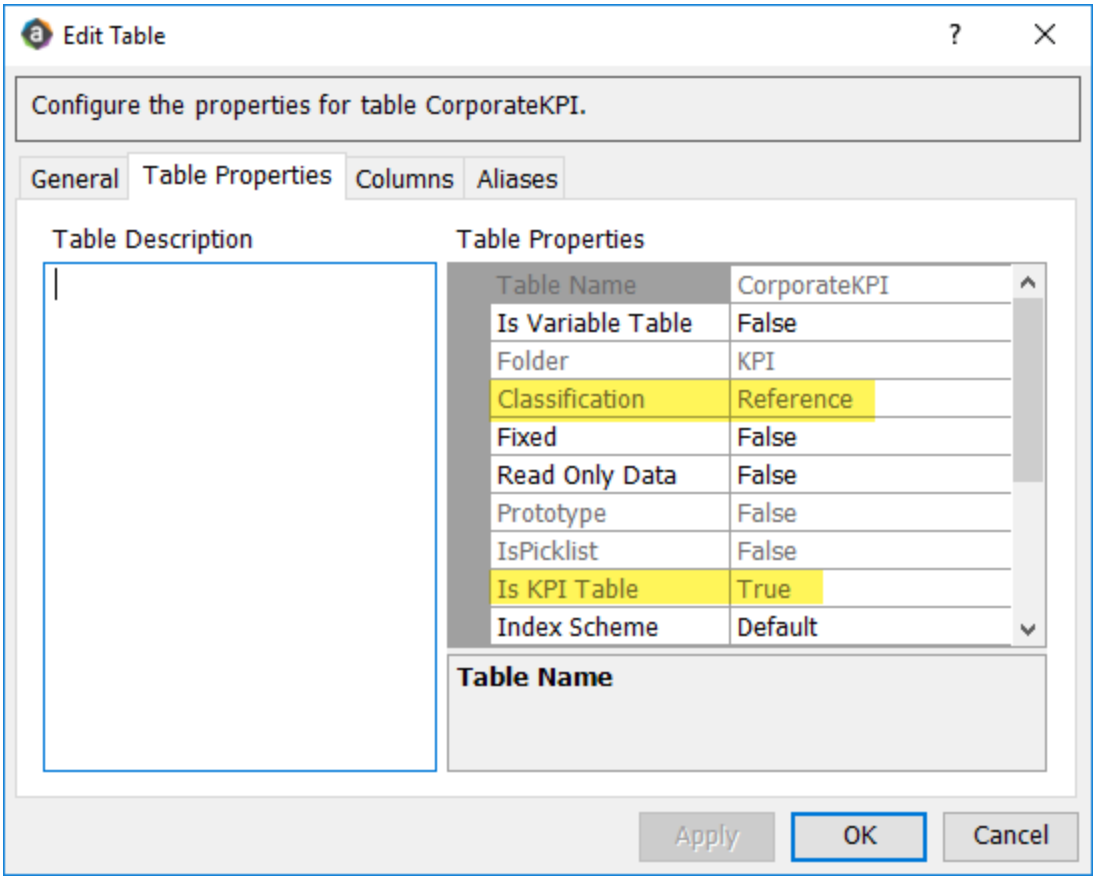
The KPI table is automatically created with all of the columns needed to populate the KPI Panel component. For example, the table has columns such as Title, TitleIcon, Value1, Value2, Value2Label, and so on. The columns roughly correspond to the KPISource data source columns, with a few differences. For more information on the required columns in the KPI Table, see the following topic in Axiom Software Help: *About KPI Tables* (AX1712).

Currently, KPI tables can only be created in the Web Client. Once a KPI table has been created, additional columns can be added as needed using the standard **Edit Table** feature in the Desktop Client.

To populate a KPI table with KPI values, you can use any of the standard features for populating tables, such as Save Type 1 and Open Table in Spreadsheet.

KPI table restrictions and limitations

KPI tables are reference tables; they do not belong to a new table classification. When you create a new KPI table, it is created as a reference table with the **Is KPI Table** property set to **True**. This flags the table as a KPI table and controls its structure and use.



KPI tables follow the same general rules of reference tables, with some additional restrictions. In addition to the pre-set column structure, the following limitations and restrictions apply to KPI tables:

- KPI tables cannot be plan code tables.
- KPI tables cannot have column hierarchies, column sequences, or calculated fields.

When filtering a list of tables by table classification, KPI tables will display along with the other reference tables, unless they are being omitted from the list as invalid selections.

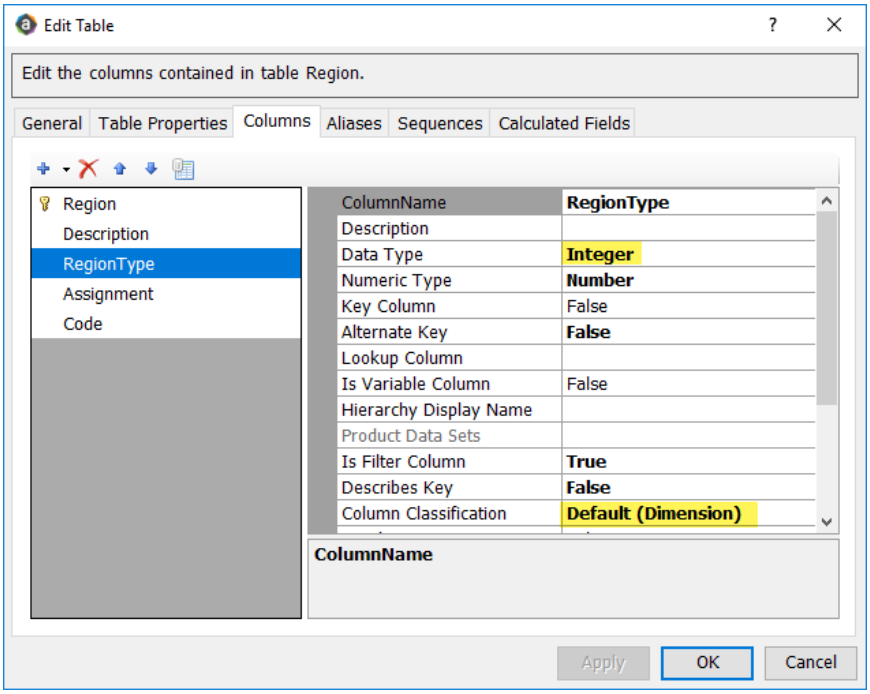
Changes to column classification behavior

A few changes were made to table columns, to impact how those columns are treated by data queries.

Default column classification for Integer columns

The **Column Classification** property for table columns is used to classify columns as **Dimension** or **Value**. This classification is used by the Data Explorer feature to determine whether a column is available to be used as row or column grouping (Dimension) or as a reportable value column (Value). Columns have a default classification based on the table classification and column data type, which you can override on a per column basis as needed.

Starting in this version, the default classification for Integer columns in reference tables is now Dimension instead of Value. This impacts existing columns that are set to **Default**, and new columns created after upgrade.



This change was made because Integer columns in reference tables are more likely to hold dimension or grouping codes rather than data values.

In previous releases, the column classification was only used by the Data Explorer. If you have columns like this in existing Data Explorer data sets, this impacts how the columns are used in the Data Explorer. If you want an Integer column to continue to be classified as Value within the Data Explorer, you can edit the column classification to explicitly set it to Value instead of using the default behavior.

In 2018.1, we have introduced new data query behavior that references the column classification to determine the aggregation type for certain columns. See the following section for more information.

Default aggregation behavior for certain columns now based on column classification

The default aggregation behavior for Integer columns in reference tables is now based on the **Column Classification** property. This change is intended to improve the default aggregation behavior to automatically provide the desired behavior in the majority of cases. This applies to Axiom queries, GetData functions, and the new Data Grid component.

IMPORTANT: As of patch 2018.1.23 this change is now controlled by a system configuration setting, so that the change does not affect customers on upgrade.

Aggregation refers to how the data query treats multiple values from a column, to result in a single value for each row returned by the query. For example, the query may return the sum of the values, or the maximum or minimum value, or an average value. Data queries use default aggregation behavior to present the data in the most useful way—for example, to sum data columns and return the maximum value for dimension columns. The default aggregation behavior for a column can be overridden in data queries as needed.

In previous versions, non-key, non-validated Integer columns (all types) were aggregated in data queries as follows:

- If the primary table of the query was a reference table, the column was summed.
- If the primary table of the query was a data table, the maximum value was returned.

Going forward, the aggregation behavior for non-key, non-validated Integer columns will be determined based on the column classification, regardless of whether the primary table is a reference table or a data table:

- If the column classification is **Value**, the column is summed.
- If the column classification is **Dimension**, the maximum value is returned.

The default classification for non-key, non-validated Integer columns is Value in data tables and Dimension in reference tables. The default classification for Numeric columns is always Value. In the majority of cases, this default classification will result in the desired aggregation behavior. If a particular column should be aggregated differently, you can change the classification to either Value or Dimension instead of using the default behavior. And if you need a column to aggregate differently in a particular context, you can always override the default aggregation behavior using optional features such as AxAggregate for Axiom queries.

Remember, this new behavior does not apply to Integer key columns or validated columns. These columns always return the maximum value by default.

As of patch 2018.1.23, this change is controlled by the system configuration setting **UseLegacyColumnAggregation**. This setting is **True** by default, which means the previous behavior applies, and customers will not see any change in data queries after the upgrade. We plan to make some further enhancements to the new aggregation behavior in upcoming releases, so we recommend leaving the system configuration setting enabled until these changes are complete.

If you installed an earlier version of 2018.1 and then upgrade to patch 2018.1.23 or higher, the aggregation behavior will be restored to the prior behavior. This means that queries will behave the same way that they did before the upgrade, regardless of the column classification. If you changed the column classification of any columns, this will not affect anything at this time other than the Data Explorer (as described in the previous upgrade consideration), and presumably the change is appropriate for the Data Explorer as well.

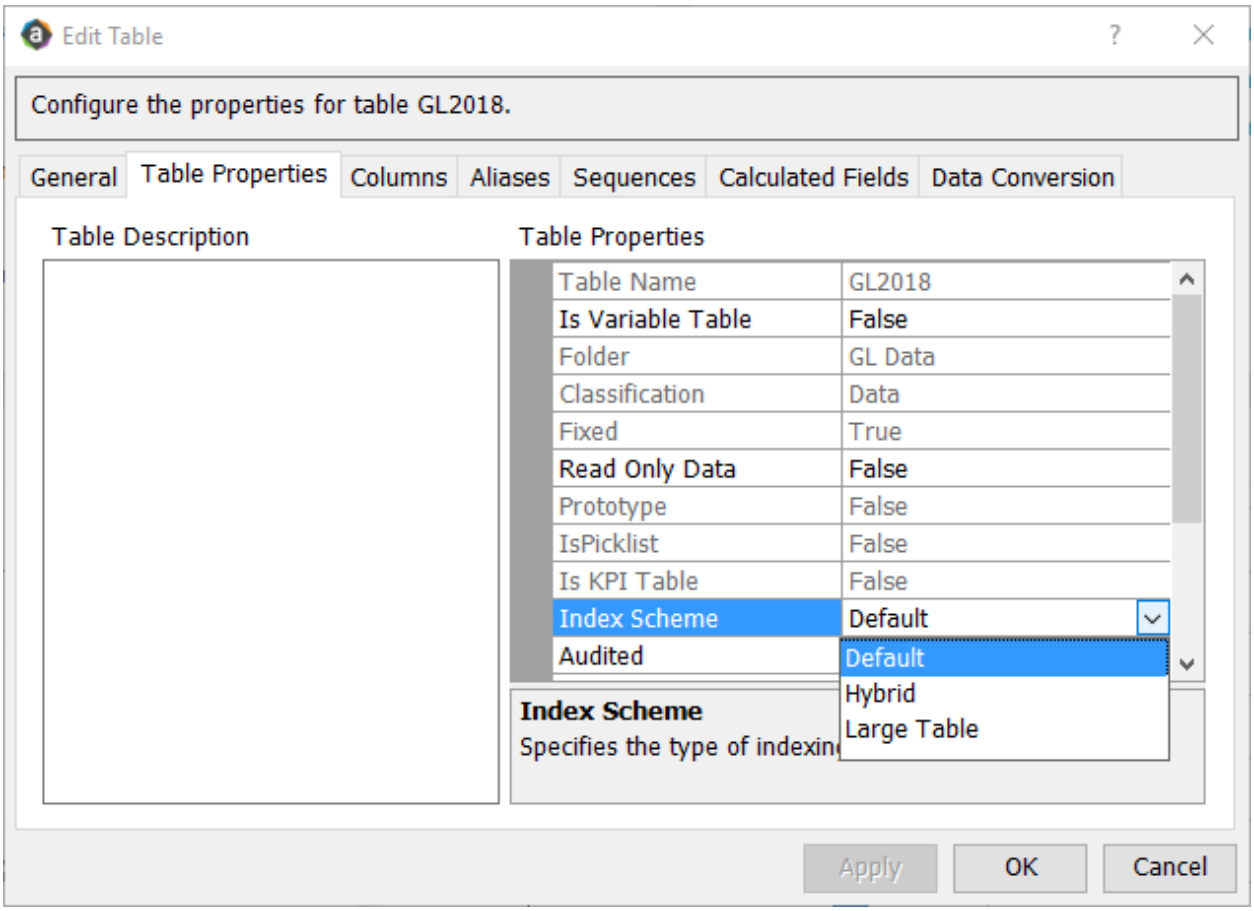
Configurable index scheme for tables

Tables in Axiom Software now have a configurable index scheme that determines how the table is indexed in the database. In certain situations, using a specialized index scheme can improve performance when querying data from the table or saving data to the table.

IMPORTANT: The table index scheme is an advanced feature that requires a good understanding of data structures and data management. You should only change the index scheme as instructed by your implementation consultant or Kaufman Hall Software Support.

Data tables and reference tables can use the following index schemes:

- **Default**: This option applies the default index scheme suitable for most tables. Tables use a clustered row store index.
- **Hybrid**: This option combines the default index scheme with a non-clustered column store index. This option may improve performance on larger tables when the **Large Table** scheme cannot be used due to its restrictions.
- **Large Table**: This option applies an index scheme suitable for tables that contain very large sets of data—from hundreds of millions to billions of rows. Tables use a clustered column store index. Use of this option introduces a set of restrictions on how the table and its data can be managed.



New Index Scheme option for tables

In previous versions, Axiom Software supported a Large Data table classification that was intended for tables that contain very large sets of data. While the concept of "large tables" is still needed, we have changed how these tables are specified, to provide more flexibility in how they can be used. As a result, the Large Data table classification no longer exists. Instead, large tables are specified by setting the new **Index Scheme** table property to **Large Table**. Removing the separate table classification means that regular data tables and large tables can now belong to the same table type, which may streamline security configuration for these tables. Additionally, reference tables can now be configured as large tables.

Tables using the Large Table index scheme have the same restrictions that previously applied to the Large Data table classification. Table data can only be updated by using import utilities or Copy Table Data. Once the table contains data, the table structure cannot be changed. Additionally, certain table features like aliases, sequences, and calculated fields cannot be defined for large tables.

The following additional related enhancements were made, to query the index scheme of a table:

- The column IndexScheme was added to Axiom.Tables.
- The keyword IndexScheme was added to the GetTableInfo function.

Upgrade behavior

When you upgrade, existing tables are updated as follows:

- Any tables that were using the Large Data table classification are converted to the Data table classification, and the index scheme is set to Large Data. Additionally, any Large Data table types are converted to Data table types. After upgrade, you may decide that the former Large Data table types are no longer needed because you can now move the large tables into other existing Data table types.
- All other tables are set to Default index scheme.

In both cases, the tables behave the same before and after the upgrade.

In rare cases, you may have tables where a custom non-clustered column store index was applied to the table, either by Kaufman Hall Software Support or as part of an installed product package. In this case, the table is automatically set to use the Hybrid index scheme, and the index becomes system-managed instead of custom.

Additional enhancements

Limitations removed from Windows Client spreadsheet engine

Some Excel-compatibility limitations were removed from the spreadsheet engine used by the Windows Client. This engine is also used by the Axiom Application Server when rendering Axiom forms, and by the Axiom Scheduler Server when using the "Web engine" for spreadsheet processing.

- The Aggregate function is now supported.
- New functions introduced in Excel 2013 are now supported (see list), with the exception of FORMULATEXT.
- Conditional formatting features compatible with Excel 2016 are now supported. This means that the previous limitations on using features such as icon sets, overlapping rules, and more than 3 rules no longer apply.

This change applies to both 2017.3 and 2018.1.

Column validation for Save Type 4

When executing Save Type 4, Axiom Software now validates the entries in the SaveStructure2DB control row to ensure that all entries are valid columns for saving. If an invalid column name is detected, an error occurs and the save process stops before any changes are saved to the target table. This change was made for consistency with other save processes and also to assist in file creation and troubleshooting (to quickly identify errors in the control row).

In previous versions, Save Type 4 ignored any invalid entries in the SaveStructure2DB control row. Therefore it is possible that you may have Save Type 4 utilities that worked in previous versions but will now fail with an error (due to accidental invalid entries in the control row). If this error occurs, locate the invalid column entry identified in the error message and remove it from the control row.

Help now hosted on centralized web site

The Axiom Software Help files are now hosted on a centralized web site instead of installed on your application server. When you access help from within the software, you are automatically granted rights to access the site.

This change provides several advantages, including the ability for Kaufman Hall to update the help files as needed to provide you with the most current information for your software version. It also reduces the size of the application server installation.

As part of this change, the help has been updated to use a new format with a more modern user interface.

Axiom query enhancements

- There is no longer any restriction on using the same dimension in the sum by and as a column filter. In previous releases, this configuration would cause errors in some environments, such as when drilling Formatted Grid components.
- When the primary table of an Axiom query is a reference table, and the field definition contains columns from linked data tables, the data table columns can now use column filters.
- The Choose Column helpers in the Sheet Assistant for the Axiom query **Sum By** and **Data Sort** settings now have up and down arrows so that you can re-order selected columns.

Improvements to data save processes

Axiom Software now automatically applies columnstore indexes on temporary tables if possible (as determined by your SQL Server version), to improve performance when saving data to the database.

As a result of this change, a few advanced system configuration settings have been removed in this release:

- **UseInMemoryTempTable**: Testing has determined that this option does not noticeably improve performance, and it has been removed.
- **CreateClusteredColumnStoreIndexOnTemporaryTables**: This option is now obsolete and has been removed.
- **CreateNonClusteredColumnStoreIndexOnTemporaryTables**: This option is now obsolete and has been removed.

Additionally, testing has indicated that setting **UpdateStatisticsDuringSave** to **True** is unnecessary and could potentially impact performance in some systems. We have changed the default setting of this option to **False**. On upgrade, any existing configured setting for this option will be removed, so that all systems now use the default of False. However, the option is being retained so that it can be enabled if it is determined to be helpful in certain rare cases.

These are all advanced system configuration settings that do not impact most installations and would have only been modified while working with Kaufman Hall Software Support.

Miscellaneous enhancements

- When using Save Type 4 to Axiom.Roles to create or modify a security role, you can now set the assigned subsystem for the role using the **AssignedSubsystem** column.
- When using an export utility to create a delimited file, you can now optionally compress the export file. The export file is saved to the destination folder as a ZIP file.
- The **Upgrade Control Sheet** option no longer displays on the Sheet Assistant if the Axiom file is opened as read-only. Upgrading the Control Sheet requires read/write access to the file.
- Additional audit details were added for imports, to make it easier to troubleshoot import issues.