What's New
Axiom Software
Version 2019.1

KaufmanHall
AXIOM

Contents

Introduction

Kaufman Hall is pleased to announce the release of **Axiom Software Version 2019.1**. This release features enhancements to various areas of the software, such as:

- Enhancements to the **Data Grid component**, to accommodate a wider range of use cases:
  - Option to suppress zero-data rows
  - Ability to include calculations on the total row
  - Create if-then-else conditions for icon display
- Enhancements to the **Fixed Report component**, including the same enhancements listed for the Data Grid component, as well as additional options to reverse the sign on variance calculations and to hide empty sections
- New **Import** feature to import multiple files from a folder using a single import utility, and general improvements to the Import Wizard dialog
- Updated branding (logos and colors) throughout the application

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.

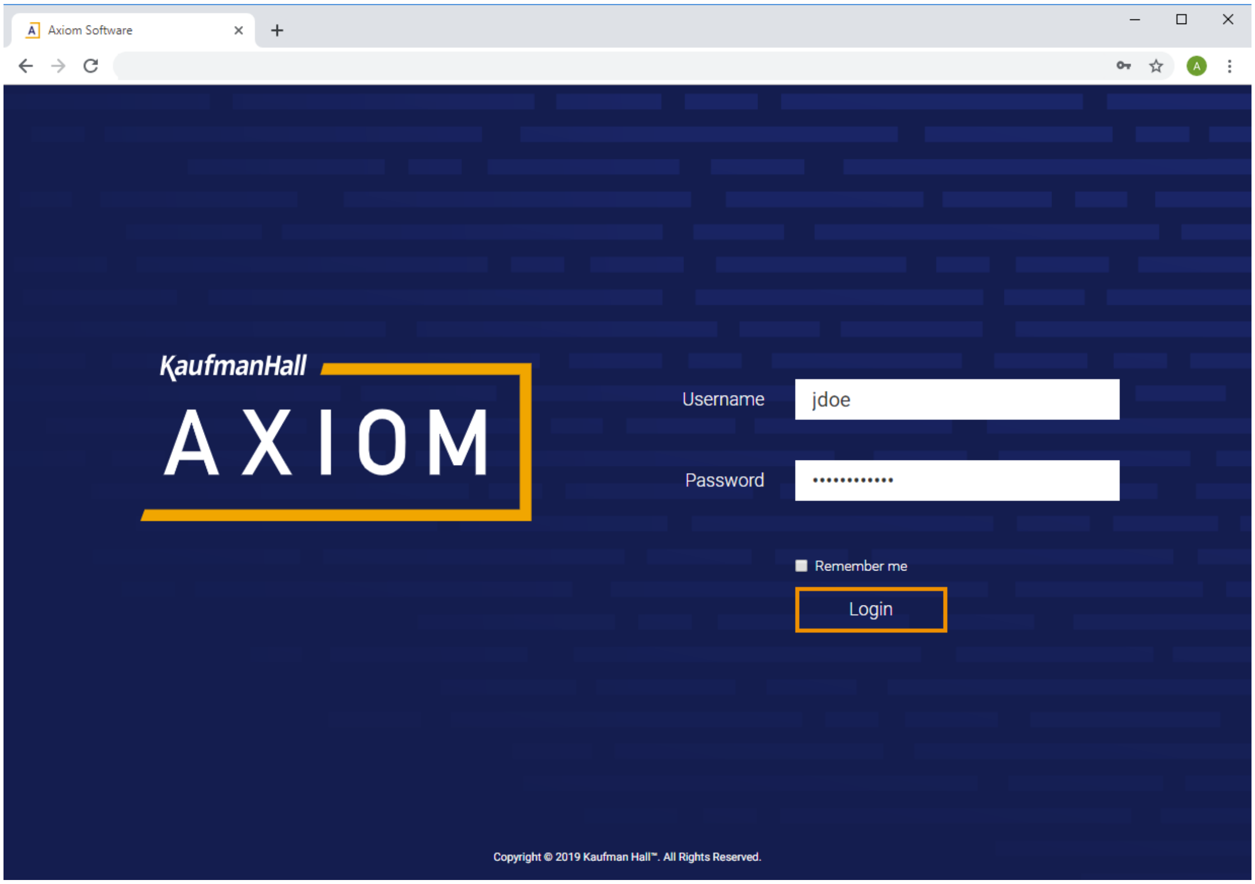
IMPORTANT: Before upgrading to version 2019.1, make sure you have reviewed the separate *Release Notes* document to understand any important technical changes and upgrade considerations in this release.

Branding updates

Version 2019.1 features updated logos, images, and colors for Axiom Software. These changes are most prominent when logging into the system, and when using the Web Client.

Login screens

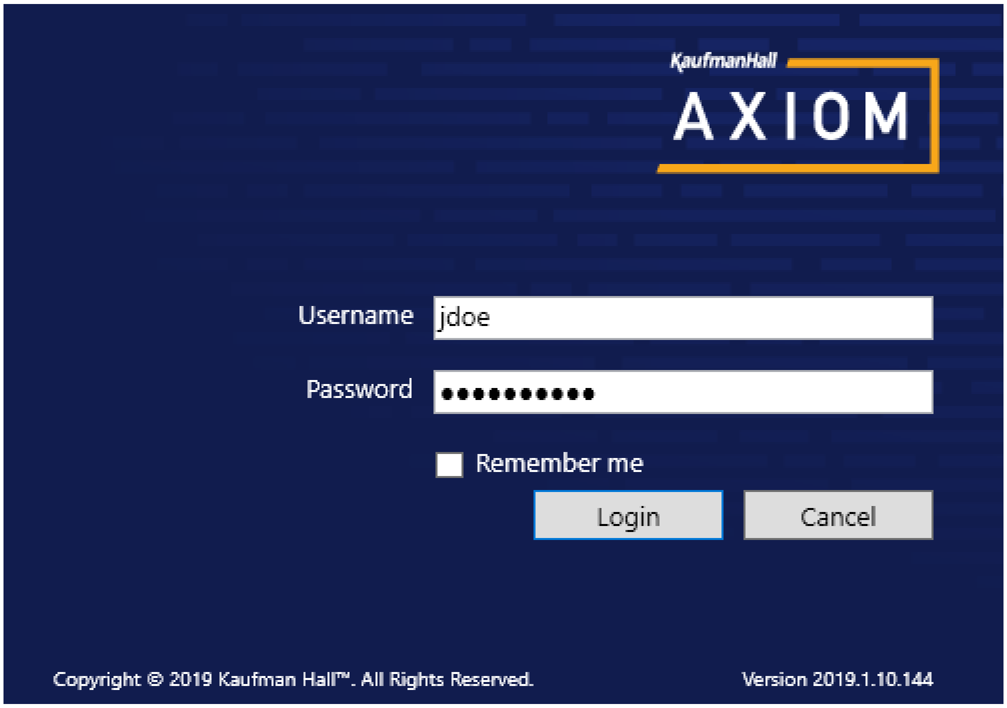
The login screens for both the Web Client and the Desktop Client have been updated with new branding.



Username: jdoe

Password: ••••••••••••

Updated Web Client login page



Username: jdoe

Password: ●●●●●●●●●●

□ Remember me

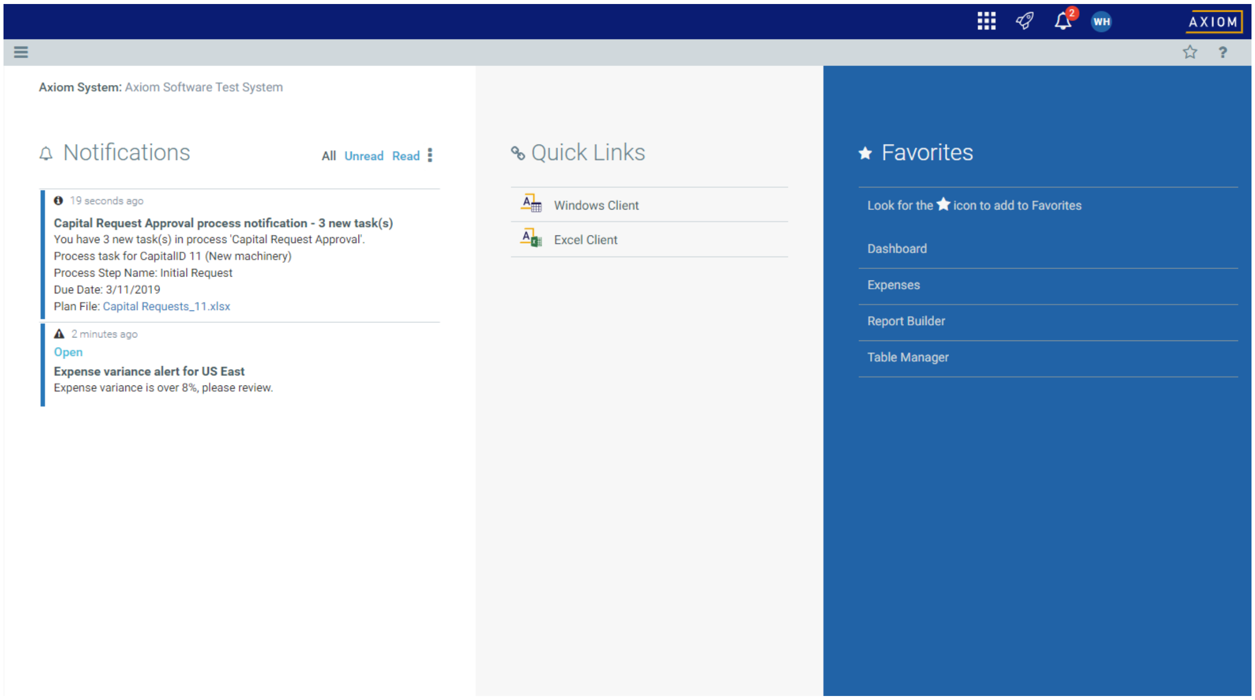
Copyright © 2019 Kaufman Hall™. All Rights Reserved.

Version 2019.1.10.144

Updated Desktop Client login dialog

▶ Web Client container

The bar that displays across the top of pages in the Web Client has been updated to match the new branding. The container is present as you navigate throughout the Web Client.



Axiom System: Axiom Software Test System

19 seconds ago

Capital Request Approval process notification - 3 new task(s)
You have 3 new task(s) in process 'Capital Request Approval'.
Process task for CapitalID 11 (New machinery)
Process Step Name: Initial Request
Due Date: 3/11/2019
Plan File: Capital Requests_11.xlsx

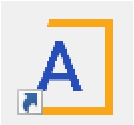
2 minutes ago
Open
Expense variance alert for US East
Expense variance is over 8%, please review.

Look for the ★ icon to add to Favorites

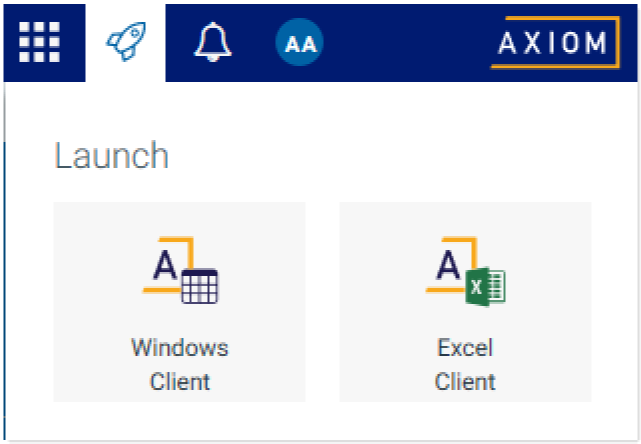
Example Web Client home page showing updated container

Additional changes

The Axiom Software "A" logo has been updated to match the new software logo. This change affects desktop shortcuts, the About box, and various other areas of the application where the logo was used.



Updated desktop shortcut



Updated logos in Launch panel

Axiom forms

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

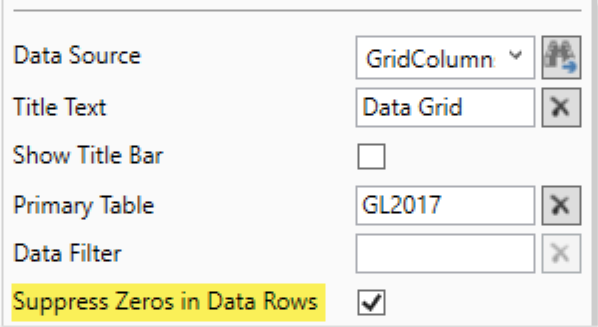
Data Grid enhancements

Several enhancements were made to the Data Grid component, to streamline the component setup and to handle a wider variety of reporting needs.

NOTE: These enhancements only apply to the Data Grid component for Axiom forms. The Data Grid component for web reports does not have access to these enhancements.

Suppress zero-data rows in Data Grids

You can now configure a Data Grid component to hide rows that contain all zero data. This works in a similar manner as the corresponding feature for Axiom queries.



Data Source	GridColumn
Title Text	Data Grid
Show Title Bar	☐
Primary Table	GL2017
Data Filter	
Suppress Zeros in Data Rows	☑

New option to suppress zero-data rows in data grids

If **Suppress Zeros in Data Rows** is enabled in the component properties, then rows that contain all zero data are hidden in the report.

Non-key columns that meet both of the following criteria are evaluated to determine whether a row should be hidden:

- The column data type is Integer (all types) or Numeric.
- The column is from the primary table or an additional data table.

If the primary table is a data table, Integer and Numeric columns on lookup reference tables are ignored—meaning these columns may have values, but the row is still suppressed if all applicable data table columns have zero values. There is one exception: reference table columns are considered if the column classification is Values and the numeric type is Currency.

Calculated columns defined in the data grid are not evaluated for this purpose and do not prevent a row from being hidden.

If the component uses a HierarchicalGrid data source, this option does not display in the component properties. Instead, this option can be enabled separately for each grouping level within the data source, using the `[SuppressZeroRows]` tag.

Improved total row behavior for Data Grids

The behavior of the total row for the Data Grid component has been enhanced to simplify the setup, and to better support calculations. The change also better aligns the total behavior of the Data Grid component with the behavior of the Fixed Report component.

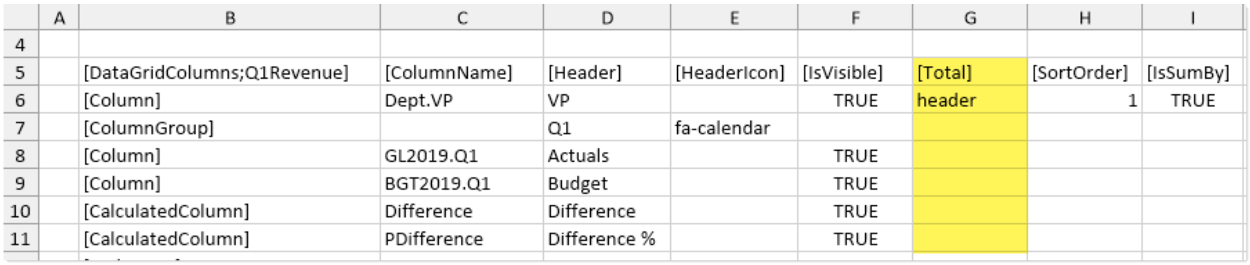
The new total behavior is designed to automatically include all relevant columns and calculations in the total row, and to automatically apply the appropriate aggregation or calculation.

- All numeric "data" columns and calculations are now automatically included in the total row by default. It is no longer necessary to flag these columns for inclusion.
- In previous releases, you had to specify the aggregation type to use in the total row for each column. This is no longer necessary. Table columns shown in the total row automatically use the same aggregation applied to the column values (such as Sum or Count). Note that certain aggregations are not supported in the total row, such as DistinctCount, so any columns using unsupported aggregations are automatically excluded from the total row.
- In previous releases, it was not possible to apply the calculation used in a calculated column to the total row. Now, calculated columns automatically apply the same calculation to the values in the total row, which means that calculations like percent difference now display the expected results in the total row.

Going forward, the process to display a total row in the grid is much easier:

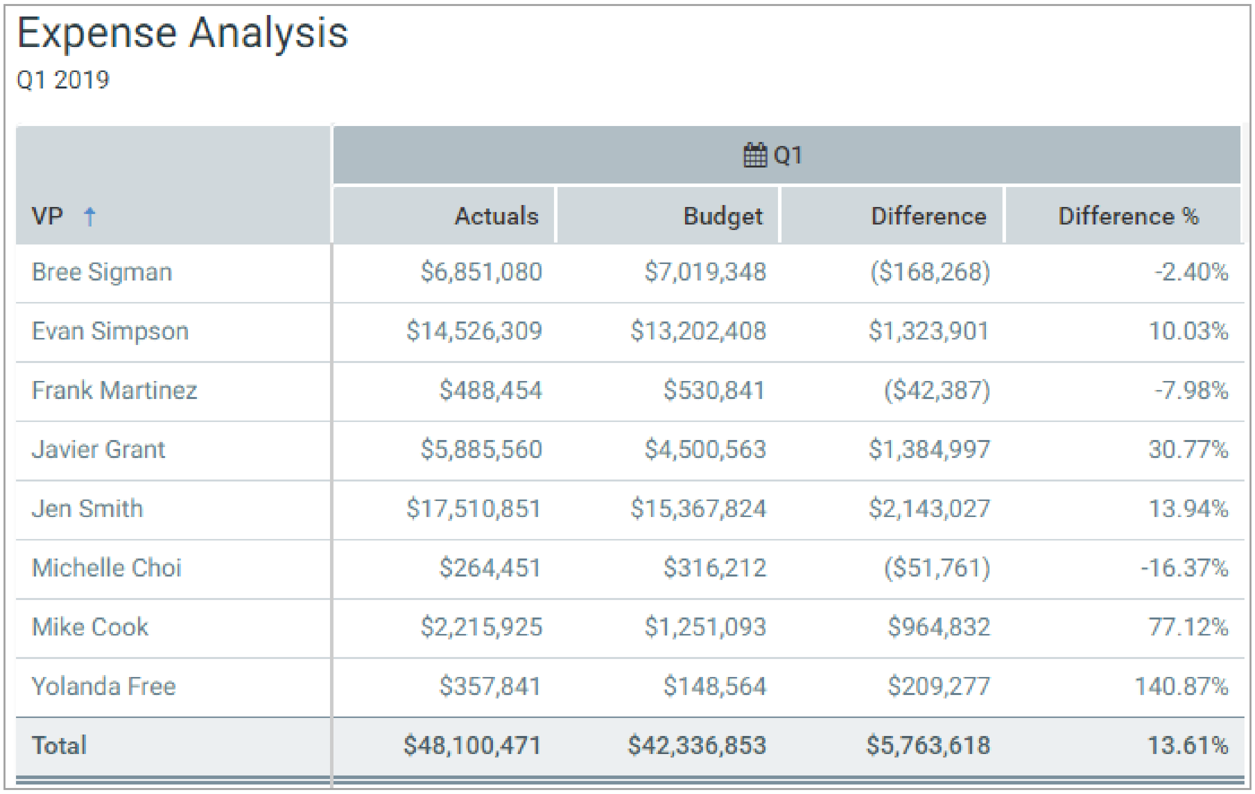
- Enable **Include Total Row** in the component properties.
- Use the **Total Row Header** field in the component properties to define a label for the total row, such as "Total" or "Total Expenses".
- Place the keyword `Header` in the `[Total]` column of the DataGridColumns data source, in the row corresponding to the column where you want the total row label to display. Typically this is the column that holds the "sum by" values, or its description column.
- In the majority of cases, there is no need to further populate the `[Total]` column with `True` or `False`, because all relevant columns are automatically included in the total row. However, in the rare case that you do not want to display a particular column, you can enter `False` for that column.

The following example shows a DataGridColumns data source with a new `[Total]` column:



	A	B	C	D	E	F	G	H	I
4									
5		[DataGridColumns;Q1Revenue]	[ColumnName]	[Header]	[HeaderIcon]	[IsVisible]	[Total]	[SortOrder]	[IsSumBy]
6		[Column]	Dept.VP	VP		TRUE	header	1	TRUE
7		[ColumnGroup]		Q1	fa-calendar				
8		[Column]	GL2019.Q1	Actuals		TRUE			
9		[Column]	BGT2019.Q1	Budget		TRUE			
10		[CalculatedColumn]	Difference	Difference		TRUE			
11		[CalculatedColumn]	PDifference	Difference %		TRUE			

This grid will render as follows in the form. Notice that all relevant columns are included in the total row by default, and calculations are applied to the total row.



Expense Analysis

Q1 2019

	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	$488,454	$530,841	($42,387)	-7.98%
Javier Grant	$5,885,560	$4,500,563	$1,384,997	30.77%
Jen Smith	$17,510,851	$15,367,824	$2,143,027	13.94%
Michelle Choi	$264,451	$316,212	($51,761)	-16.37%
Mike Cook	$2,215,925	$1,251,093	$964,832	77.12%
Yolanda Free	$357,841	$148,564	$209,277	140.87%
Total	**$48,100,471**	**$42,336,853**	**$5,763,618**	**13.61%**

The [Total] tag is automatically included in all newly created DataGridColumns data sources. For existing data sources, you must manually add it.

Backward-compatibility considerations

In previous releases, the Data Grid component used a [TotalRow] tag to configure the total row. No columns were included by default. In order to flag a column for inclusion, you were required to specify the aggregation to apply to the total row.

Existing Data Grid components that use the legacy [TotalRow] tag are treated as follows:

- The Header keyword is still honored in the [TotalRow] column to display the total row label.

- Any column with a non-blank value in the [TotalRow] column is included in the total row, using the new behavior where the current aggregation or calculation is automatically applied.

Generally speaking, this means that any column that you had previously configured to display in the total row will continue to display in the total row. Although it is possible that a table column may now use a different aggregation in the total row than it was previously configured to use, this should be an extremely rare or non-existent occurrence. Note that DistinctCount aggregation is no longer supported for use in the total row, but since this is a fringe use case, the change is unlikely to affect any existing grids. Calculated columns included in the total row will now automatically use their defined calculation instead of a specified aggregation.

Going forward, you should convert any existing grids with a total row to use the new [Total] column. This can be done very easily:

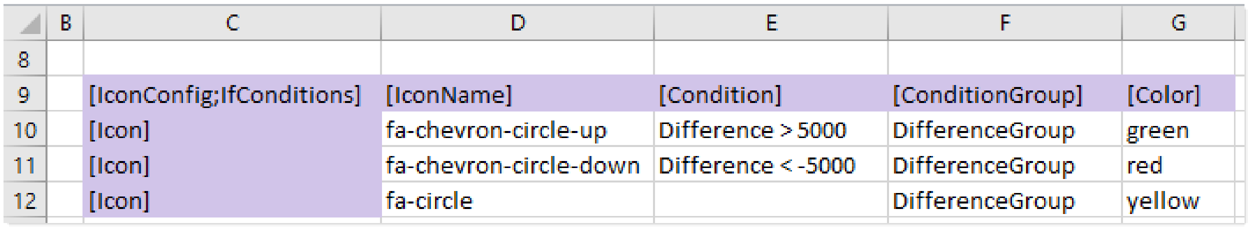
- Rename the [TotalRow] column in the DataGridColumns data source to be just [Total].
- Clear out all values in the column except the Header keyword.

In most cases, this will result in the desired total behavior. If you view the grid and determine that you want to exclude a column that has been automatically included, you can enter False into the [Total] column.

Display icons in Data Grids using if-then-else conditions

When using an IconConfig data source to display conditional icons in a Data Grid component, you can now define condition groups to be evaluated as a set of if-then-else conditions.

The new property [ConditionGroup] is used to assign a set of icon conditions to the same group. This property is added by default to all new IconConfig data sources. If you have an existing data source, you can add the property manually. For example, a condition group could be defined as follows:

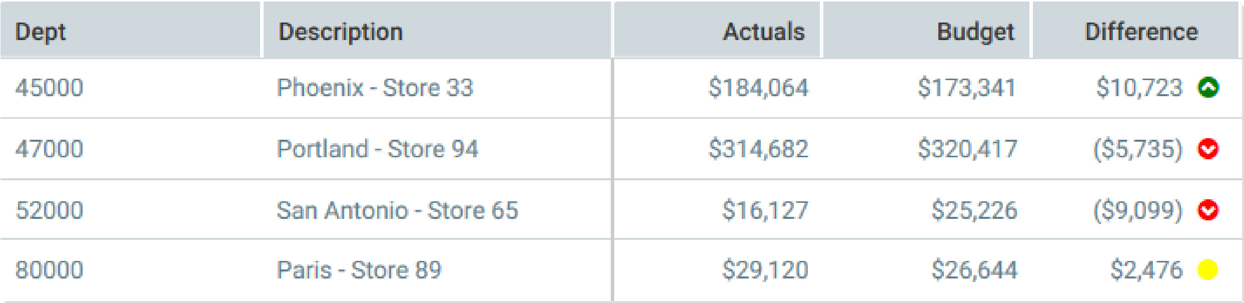


	B	C	D	E	F	G
8						
9		[IconConfig;IfConditions]	[IconName]	[Condition]	[ConditionGroup]	[Color]
10		[Icon]	fa-chevron-circle-up	Difference > 5000	DifferenceGroup	green
11		[Icon]	fa-chevron-circle-down	Difference < -5000	DifferenceGroup	red
12		[Icon]	fa-circle		DifferenceGroup	yellow

Example condition group in an IconConfig data source

This condition group shows an icon based on the Difference value, where Difference is the name of a calculated column defined for the grid. If the first condition is true for a row in the grid, then the row uses the fa-chevron-circle-up icon, and no further conditions in the group are evaluated. If the first condition is not true, then the second condition is evaluated, and so on. The row with the blank condition serves as a catch-all "else" statement, so that all rows in the grid will match one of the icons in this group. (Note that if you want to use this kind of catch-all row, it must be the last row in the group. Rows with a blank condition automatically end the group.)

The following example shows how this grid would be rendered using the icon conditions:



Dept	Description	Actuals	Budget	Difference
45000	Phoenix - Store 33	$184,064	$173,341	$10,723 ▲
47000	Portland - Store 94	$314,682	$320,417	($5,735) ▼
52000	San Antonio - Store 65	$16,127	$25,226	($9,099) ▼
80000	Paris - Store 89	$29,120	$26,644	$2,476 ●

Example data grid showing the if-then-else icon conditions

Fixed Report enhancements

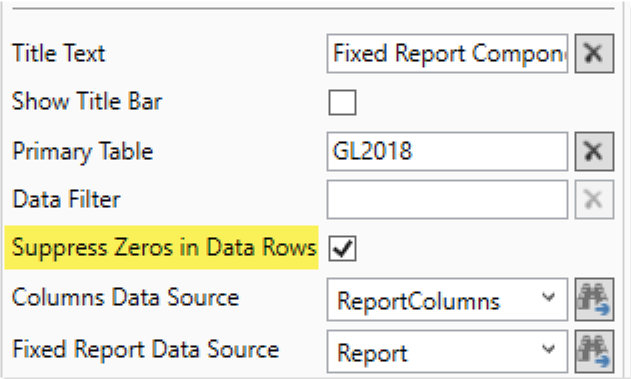
Various enhancements were made to the Fixed Report component for Axiom forms, so that it can handle a wider variety of reporting needs.

Suppress zero-data rows and empty sections in Fixed Reports

The Fixed Report component was enhanced with two new options to suppress unnecessary rows in the report—the ability to suppress zero-data rows, and the ability to hide empty data row sections.

Suppress zero-data rows

You can now configure a Fixed Report component to hide rows that contain all zero data. This works in a similar manner as the corresponding feature for Axiom queries. If **Suppress Zeros in Data Rows** is enabled in the component properties, then rows that contain all zero data are hidden in the report.



Title Text	Fixed Report Compon
Show Title Bar	□
Primary Table	GL2018
Data Filter	
Suppress Zeros in Data Rows	☑
Columns Data Source	ReportColumns
Fixed Report Data Source	Report

New option to suppress zeros in fixed reports

Non-key columns that meet both of the following criteria are evaluated to determine whether a row should be hidden:

- The column data type is Integer (all types) or Numeric.
- The column is from the primary table or an additional data table.

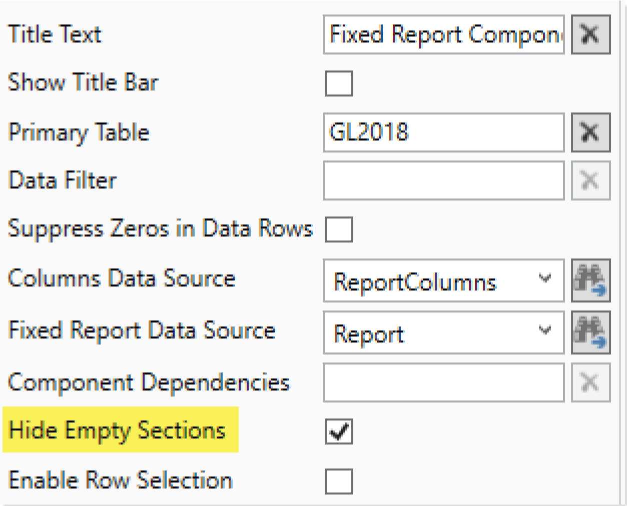
If the primary table is a data table, Integer and Numeric columns on lookup reference tables are ignored—meaning these columns may have values, but the row is still suppressed if all applicable data table columns have zero values. There is one exception: reference table columns are considered if the column classification is Values and the numeric type is Currency.

Calculated columns defined in the fixed report are not evaluated for this purpose and do not prevent a row from being hidden.

If all rows in a particular section are zero-data rows, the associated section header and subtotal row (if present) are not automatically hidden. You can enable the separate option **Hide Empty Sections** to hide these items (see the following section for more information).

Hide empty data sections

You can now configure a Fixed Report component to hide empty data sections. If **Hide Empty Sections** is enabled, then sections with no data rows are automatically hidden, including their associated section header and section subtotal row (if present). Additionally, if all of the data row sections referenced by a separate subtotal or total row are hidden, then the subtotal or total row is automatically hidden as well.



Title Text	Fixed Report Compon
Show Title Bar	□
Primary Table	GL2018
Data Filter	
Suppress Zeros in Data Rows	□
Columns Data Source	ReportColumns
Fixed Report Data Source	Report
Component Dependencies	
Hide Empty Sections	☑
Enable Row Selection	□

New option to hide empty sections in fixed reports

The primary reason to enable this option is if the report data can be dynamically changed using refresh variables (or by using interactive form components). If some sections of the report may become empty due to user inputs, you can choose to hide these empty sections.

Whether a data row section is considered to be empty depends on how the row data is defined:

- If a data row section uses a Table.Column to dynamically generate the rows, then the section is empty if no matching rows are returned from the database.

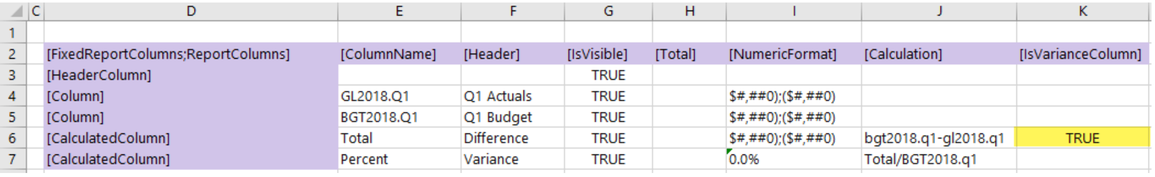
- If a data row section uses a FixedReportSectionConfig data source to define individual rows, then the section is never empty by default. If no matching data is returned from the database, the defined rows display with zeros. However, if **Suppress Zeros in Data Rows** is also enabled, then the section is considered empty if all of the defined rows are hidden due to zero suppression.

Sign reversal for variance calculations

You can now perform sign reversal on certain calculated columns in a Fixed Report component, on a section by section basis. This is intended for cases where the column contains a variance calculation, and the variance needs to be shown differently for different sections in the report. For example, when comparing budget to actuals in a report, you may want revenue sections to show positive variance (Actuals-Budget) while expense sections show negative variance (Budget-Actuals).

Sign reversal can be enabled using new columns in the FixedReportColumns data source and the FixedReportConfig data source:

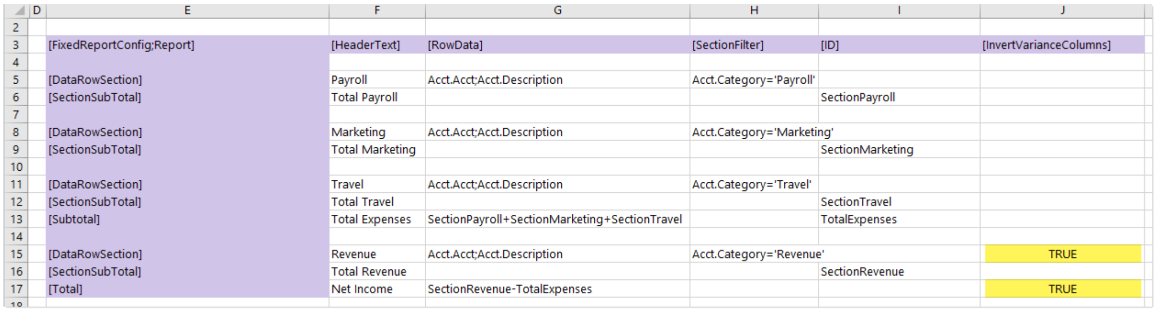
- For calculated columns where you want to enable sign reversal, enter `True` into the `[IsVarianceColumn]` field of the FixedReportColumns data source.



	C	D	E	F	G	H	I	J	K
1									
2		[FixedReportColumns;ReportColumns]	[ColumnName]	[Header]	[IsVisible]	[Total]	[NumericFormat]	[Calculation]	[IsVarianceColumn]
3		[HeaderColumn]			TRUE				
4		[Column]	GL2018.Q1	Q1 Actuals	TRUE		$#,##0);($#,##0)		
5		[Column]	BGT2018.Q1	Q1 Budget	TRUE		$#,##0);($#,##0)		
6		[CalculatedColumn]	Total	Difference	TRUE		$#,##0);($#,##0)	bgt2018.q1-gl2018.q1	TRUE
7		[CalculatedColumn]	Percent	Variance	TRUE		0.0%	Total/BGT2018.q1	

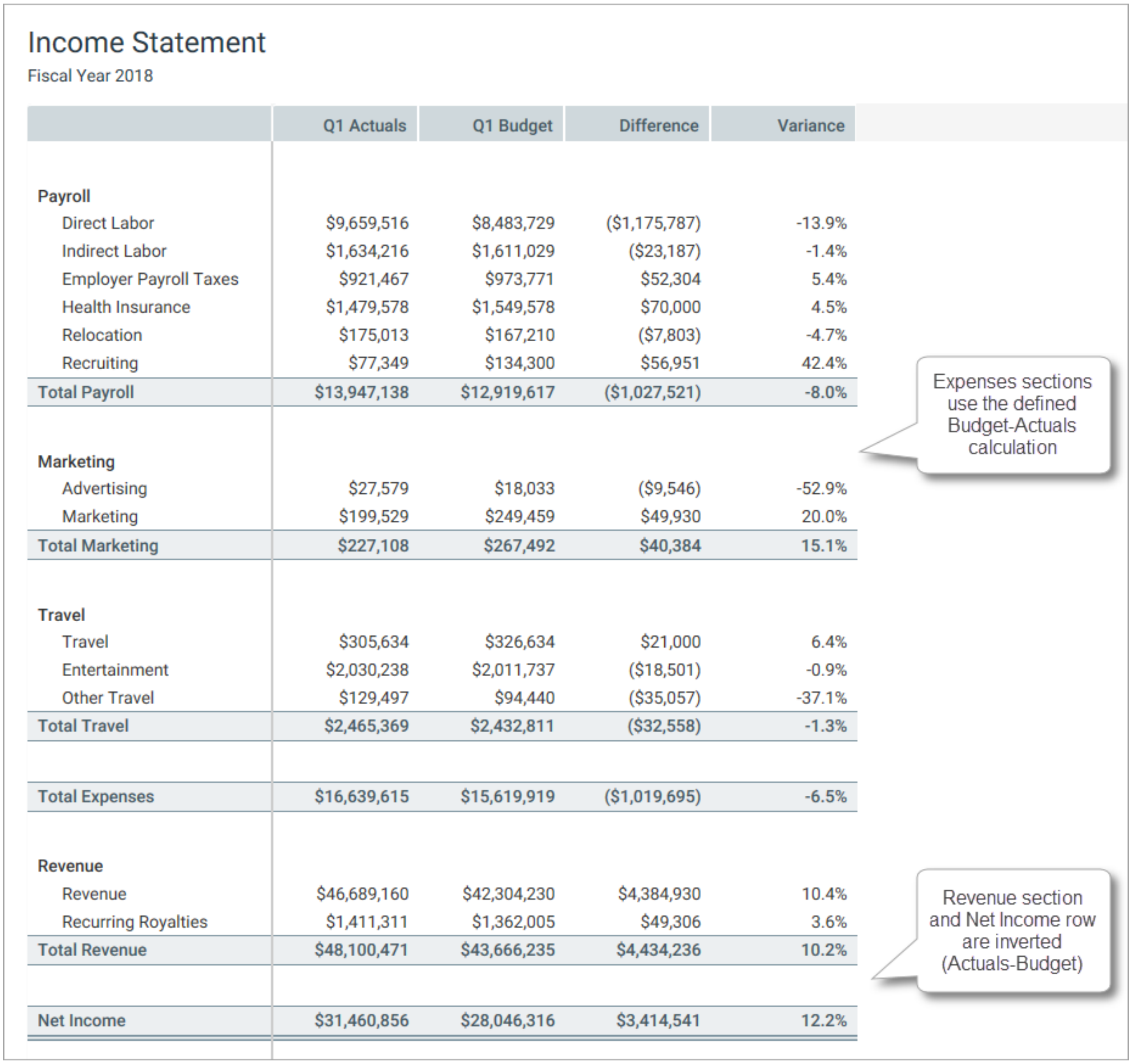
- Then for each section where you want the sign to be reversed, enter `True` into the `[InvertVarianceColumns]` field of the FixedReportConfig data source. You can enable this option on `[DataRowSection]` rows to reverse the sign on all data rows in the section, including the associated section subtotal row (if present). If you want to reverse the sign on separate `[Subtotal]` or `[Total]` rows, you must separately enable the option on those rows.

  In this example, the variance calculation is defined as Budget-Actuals, so the sign reversal should be performed on the Revenue section and on the Net Income total row. If instead the variance calculation was defined as Actuals-Budget, then the sign reversal should be performed on the Expenses sections and their separate subtotal.



	D	E	F	G	H	I	J
2							
3		[FixedReportConfig;Report]	[HeaderText]	[RowData]	[SectionFilter]	[ID]	[InvertVarianceColumns]
4							
5		[DataRowSection]	Payroll	Acct.Acct;Acct.Description	Acct.Category='Payroll'		
6		[SectionSubTotal]	Total Payroll			SectionPayroll	
7							
8		[DataRowSection]	Marketing	Acct.Acct;Acct.Description	Acct.Category='Marketing'		
9		[SectionSubTotal]	Total Marketing			SectionMarketing	
10							
11		[DataRowSection]	Travel	Acct.Acct;Acct.Description	Acct.Category='Travel'		
12		[SectionSubTotal]	Total Travel			SectionTravel	
13		[Subtotal]	Total Expenses	SectionPayroll+SectionMarketing+SectionTravel		TotalExpenses	
14							
15		[DataRowSection]	Revenue	Acct.Acct;Acct.Description	Acct.Category='Revenue'		TRUE
16		[SectionSubTotal]	Total Revenue			SectionRevenue	
17		[Total]	Net Income	SectionRevenue-TotalExpenses			TRUE

The Fixed Report component in this example looks as follows:



Income Statement

Fiscal Year 2018

	Q1 Actuals	Q1 Budget	Difference	Variance
Payroll				
Direct Labor	$9,659,516	$8,483,729	($1,175,787)	-13.9%
Indirect Labor	$1,634,216	$1,611,029	($23,187)	-1.4%
Employer Payroll Taxes	$921,467	$973,771	$52,304	5.4%
Health Insurance	$1,479,578	$1,549,578	$70,000	4.5%
Relocation	$175,013	$167,210	($7,803)	-4.7%
Recruiting	$77,349	$134,300	$56,951	42.4%
Total Payroll	**$13,947,138**	**$12,919,617**	**($1,027,521)**	**-8.0%**
Marketing				
Advertising	$27,579	$18,033	($9,546)	-52.9%
Marketing	$199,529	$249,459	$49,930	20.0%
Total Marketing	**$227,108**	**$267,492**	**$40,384**	**15.1%**
Travel				
Travel	$305,634	$326,634	$21,000	6.4%
Entertainment	$2,030,238	$2,011,737	($18,501)	-0.9%
Other Travel	$129,497	$94,440	($35,057)	-37.1%
Total Travel	**$2,465,369**	**$2,432,811**	**($32,558)**	**-1.3%**
Total Expenses	**$16,639,615**	**$15,619,919**	**($1,019,695)**	**-6.5%**
Revenue				
Revenue	$46,689,160	$42,304,230	$4,384,930	10.4%
Recurring Royalties	$1,411,311	$1,362,005	$49,306	3.6%
Total Revenue	**$48,100,471**	**$43,666,235**	**$4,434,236**	**10.2%**
Net Income	**$31,460,856**	**$28,046,316**	**$3,414,541**	**12.2%**

Expenses sections use the defined Budget-Actuals calculation

Revenue section and Net Income row are inverted (Actuals-Budget)

Improved total and subtotal row behavior

The behavior of subtotal and total rows for the Fixed Report component has been updated to better support calculations, and to align with the new behavior of the Data Grid component.

The new total behavior and configuration for the Fixed Report component is very similar to previous releases, because fixed reports were already designed to automatically include all relevant columns in subtotal and total rows by default. The important change is that calculated columns defined in the report now apply the same calculation to values in the subtotal or total row, instead of simply adding or subtracting the calculated column values. Previously you may have excluded calculated columns from subtotal and total rows because they did not display the correct values—going forward, this should no longer be necessary.

The Fixed Report component now uses a `[Total]` tag to configure subtotal and total rows, instead of the previous `[Subtotal]` tag, so that it uses the same tag name as the Data Grid component. The new tag is automatically included in all newly created FixedReportColumns data sources. For existing data sources, you must manually add it.

When using the new `[Total]` tag, in most cases you do not need to do anything in order to display the desired columns in subtotal and total rows. If a column is automatically included that you do not want to show, you can enter `False` for that column.

▶ Backward-compatibility considerations

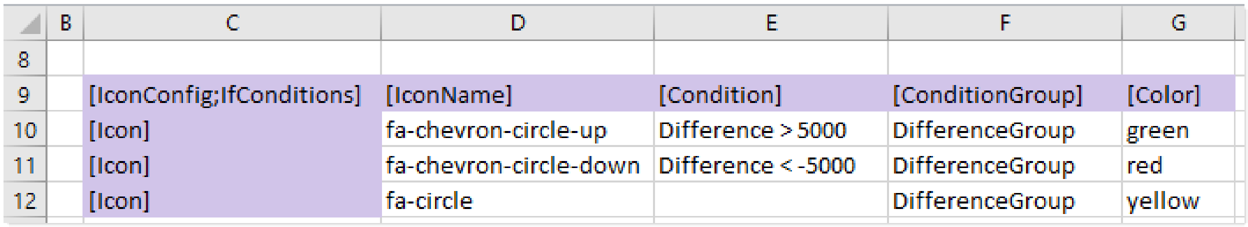
Existing Fixed Report components that use the legacy `[SubTotal]` tag will continue to work in almost the same way as before. Relevant columns are included in subtotal and total rows unless they are set to `False`. Calculated columns included in subtotal and total rows will now use the new behavior of applying the calculation to the total row.

Going forward, you should convert any existing fixed reports to use the new tag by renaming the existing `[Subtotal]` tag in the FixedReportColumns data source to `[Total]`. No other changes are needed, though you may now want to include calculated columns that you had previously excluded.

Display icons in Fixed Reports using if-then-else conditions

When using an IconConfig data source to display conditional icons in a Fixed Report component, you can now define condition groups to be evaluated as a set of if-then-else conditions.

The new property `[ConditionGroup]` is used to assign a set of icon conditions to the same group. This property is added by default to all new IconConfig data sources. If you have an existing data source, you can add the property manually. For example, a condition group could be defined as follows:

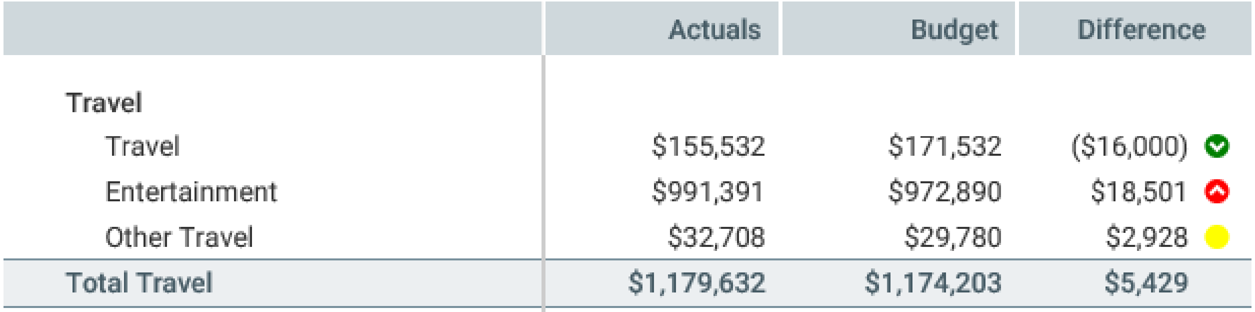


	B	C	D	E	F	G
8						
9		[IconConfig;IfConditions]	[IconName]	[Condition]	[ConditionGroup]	[Color]
10		[Icon]	fa-chevron-circle-up	Difference > 5000	DifferenceGroup	green
11		[Icon]	fa-chevron-circle-down	Difference < -5000	DifferenceGroup	red
12		[Icon]	fa-circle		DifferenceGroup	yellow

Example condition group in an IconConfig data source

This condition group shows an icon based on the Difference value, where Difference is the name of a calculated column defined for the report. If the first condition is true for a row in the report, then the row uses the fa-chevron-circle-up icon, and no further conditions in the group are evaluated. If the first condition is not true, then the second condition is evaluated, and so on. The row with the blank condition serves as a catch-all "else" statement, so that all rows in the report will match one of the icons in this group. (Note that if you want to use this kind of catch-all row, it must be the last row in the group. Rows with a blank condition automatically end the group.)

The following example shows how a section of this report would be rendered using the icon conditions:



	Actuals	Budget	Difference
Travel			
Travel	$155,532	$171,532	($16,000) ✓
Entertainment	$991,391	$972,890	$18,501 ▲
Other Travel	$32,708	$29,780	$2,928 ●
Total Travel	$1,179,632	$1,174,203	$5,429

Example fixed report section showing the if-then-else icon conditions

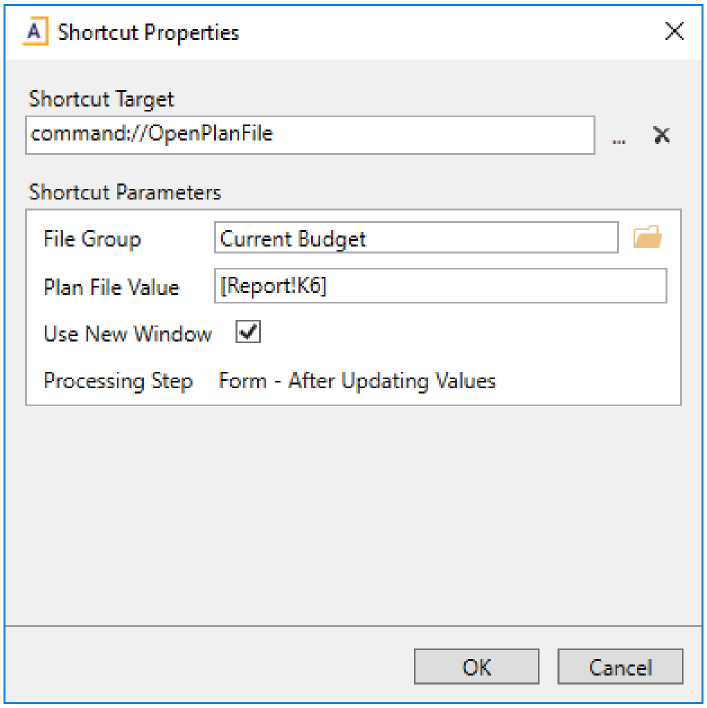
Additional Axiom form enhancements

New command to open plan files from an Axiom form

A new command is available to open a designated plan file from within an Axiom form. To use the **Open Plan File** command, you specify the file group that the plan file belongs to, and then you specify the plan file value.

- The file group can be a file group name, a file group alias name, or the current file group (if the form belongs to a file group).
- The plan file value can be obtained from a cell in the source file, using a bracketed cell reference. The plan file value corresponds to a code in the plan code table for the file group, such as department codes if the plan code table is Dept.

In the following example, the plan file value is being read from `[Report!K6]`. If this cell contained the value 24000, the command would open the plan file for Dept 24000 in the file group that the Current Budget alias points to.



Shortcut Target
command://OpenPlanFile

Shortcut Parameters

File Group	Current Budget
Plan File Value	[Report!K6]
Use New Window	☑
Processing Step	Form - After Updating Values

Example Open Plan File command for use in Axiom forms

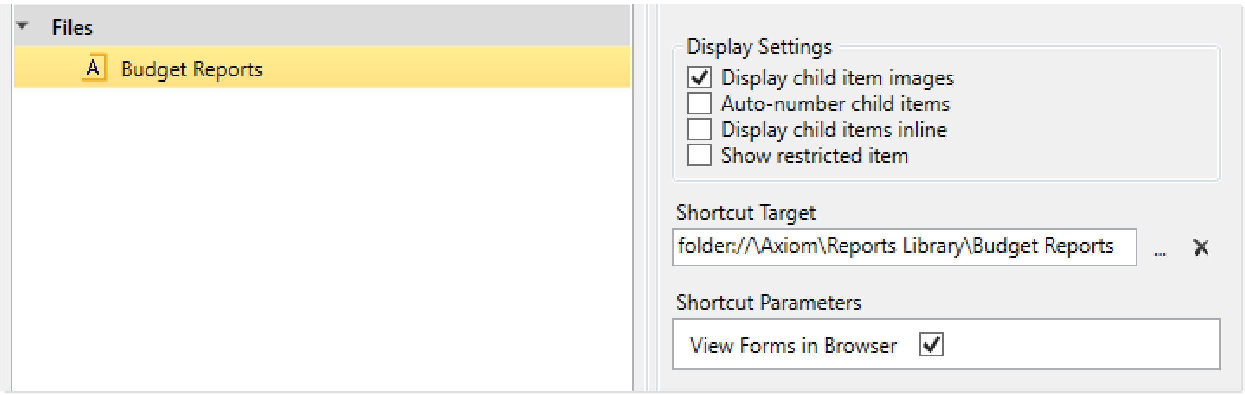
Although there are other features that can be used to open plan files from within an Axiom form, this new option provides the following advantages:

- It can open both form-enabled plan files and spreadsheet plan files.
- It is easier to set up than many options, requiring only two basic items of information.
- It can be used in components where it is not possible to dynamically generate a URL per row, such as Data Grid components and Fixed Report components.

▶ Open Axiom forms from a folder in a task pane

When creating a custom task pane for use in the Desktop Client, you can link a task pane item to a folder in order to dynamically show all files in that folder. If the folder contains form-enabled files, you can now specify that those files should open as Axiom forms instead of opening the source spreadsheet file.

Folders now have a shortcut parameter named **View Forms in Browser**. If this option is enabled, then any form-enabled files in the folder will open as Axiom forms.



Files

Budget Reports

Display Settings

☑ Display child item images
□ Auto-number child items
□ Display child items inline
□ Show restricted item

Shortcut Target

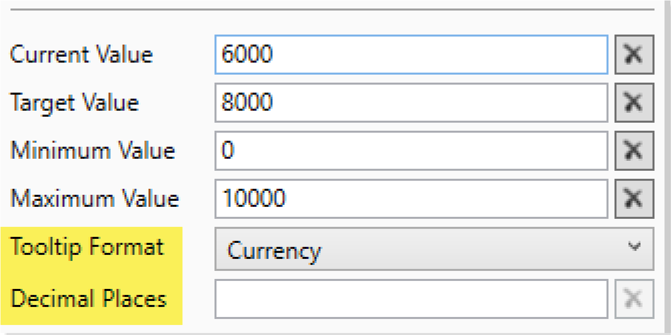
folder://\Axiom\Reports Library\Budget Reports

Shortcut Parameters

View Forms in Browser ☑

Set the display format for Bullet Chart values

The Bullet Chart component now supports the ability to set the display format of the chart values to either **Number**, **Currency**, or **Percent**. Additionally, you can set the number of **Decimal Places** to display. These settings impact the display of chart values in the tooltip and in the axis labels (if visible).



Current Value	6000
Target Value	8000
Minimum Value	0
Maximum Value	10000
Tooltip Format	Currency
Decimal Places	

The default display format is Number with 0 decimal places, which is the same as the previous default display format.

Configuration of legacy form features

A couple of legacy form features are now hidden in the user interface, so that these features can only be configured manually on the Form Control Sheet. This change was made to streamline the user interface to remove features that are inapplicable to most users, and to prevent confusion over when these features should be used.

- The option **Grid Formatting** for Formatted Grid components is no longer visible in the Form Assistant and Form Designer. Newly created grids should always use the default thematic formatting. It is not necessary to edit this setting except in very rare cases when supporting legacy forms that use the deprecated spreadsheet-formatting option.

- The option **Use Web Client Container** is no longer visible in the Form Properties dialog. Newly created forms should always use the container, since many newer features require the presence of the container to work correctly. It is not necessary to change this setting except in very rare cases when supporting legacy forms.

This is a display change only—supported functionality for these legacy features remains as is. Both settings still exist, and can still be edited manually on the Form Control Sheet as necessary.

▶ Miscellaneous

- The default width of calculated columns in Data Grid components (Axiom forms and web reports) and Fixed Report components is now 120 pixels, which is the same default width as Numeric columns.
- Checkbox tags in Formatted Grid components now honor the Text Alignment property of row and column styles. By default, if no alignment is specified, the checkboxes remain left-aligned.
- If a child embedded form contains a component with special update behavior—such as Data Grid, Fixed Report, and KPI Panel components—the component now honors the `[ForceRefresh]` option in the Menu data source to determine whether the component refreshes when the child form is reloaded via the menu.
- Tooltips for running totals and totals in Waterfall Charts now use the number format of the previous XValue.

Imports

This section details the new features and enhancements for import utilities.

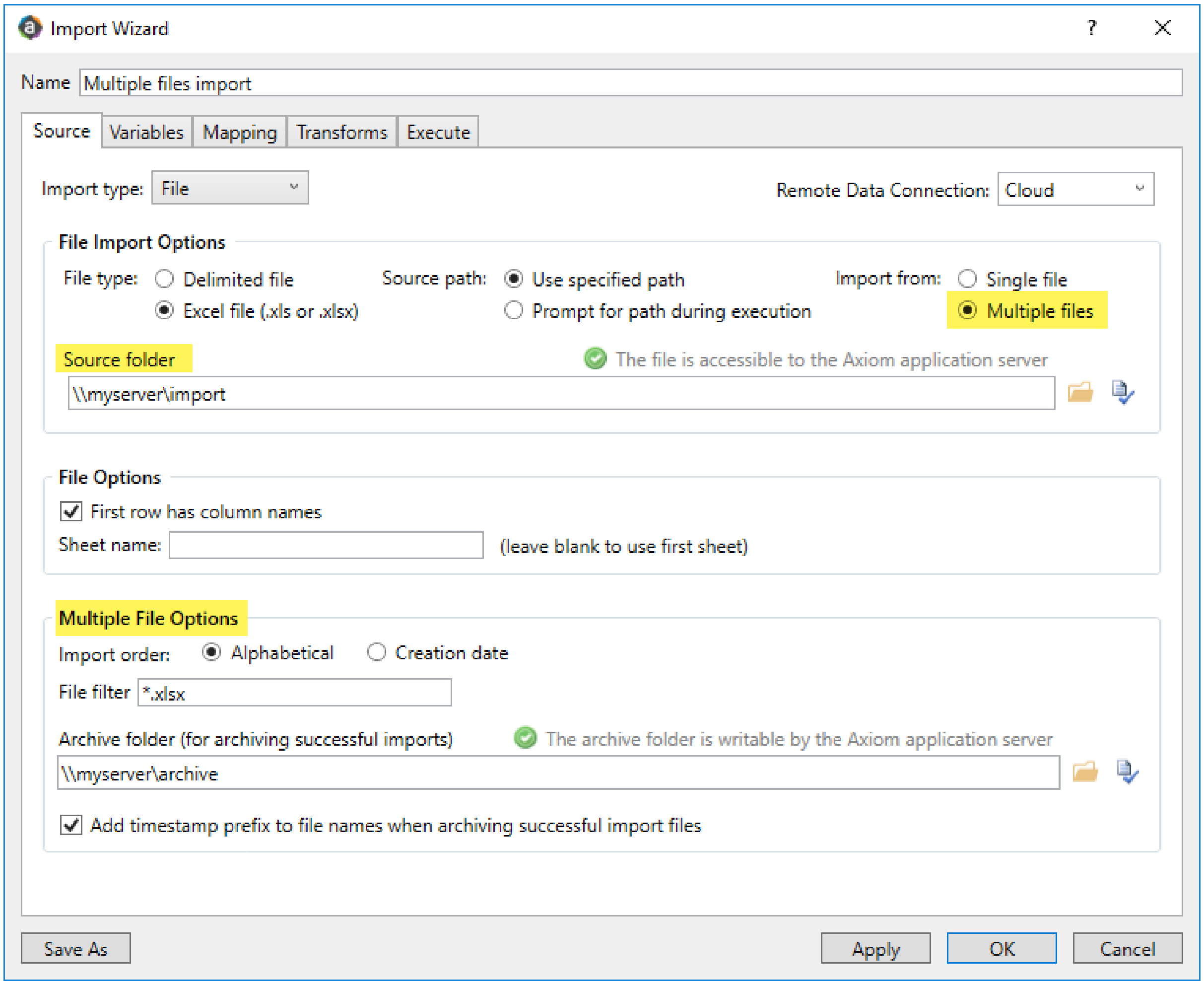
Import data from multiple files

You can now set up an import utility to import data from multiple files. You specify a source folder for the import and an optional file filter, and when the import is executed, all matching files in the folder are imported. This approach is useful when you have multiple import files that share the same file structure and destination table.

This feature imports multiple files sequentially. Each file is fully processed by the import—including saving data to the database—before moving on to the next file. Data from multiple files is not batched together before saving.

▶ Import setup

The Source tab of the Import Wizard contains new settings to support importing multiple files. When the import source is a file (either delimited or Excel), you can now specify whether you are importing a single file or multiple files.



Name: Multiple files import

Import type: File

Remote Data Connection: Cloud

File Import Options

File type: ○ Delimited file ● Excel file (.xls or .xlsx)

Source path: ● Use specified path ○ Prompt for path during execution

Import from: ○ Single file ● Multiple files

Source folder ✓ The file is accessible to the Axiom application server

\\myserver\import

File Options

☑ First row has column names

Sheet name: (leave blank to use first sheet)

Multiple File Options

Import order: ● Alphabetical ○ Creation date

File filter *.xlsx

Archive folder (for archiving successful imports) ✓ The archive folder is writable by the Axiom application server

\\myserver\archive

☑ Add timestamp prefix to file names when archiving successful import files

New options to import multiple files from a source folder

When importing multiple files, the path to the import source is now a source folder instead of a specific file path. All of the files that you want to be imported must reside within this source folder. (You also still have the option to configure the import to prompt the user to select the folder, instead of specifying the folder within the import properties.)

In the **Multiple File Options** section, you can also specify the following:

- The order in which files are imported, alphabetical or by creation date
- An optional file filter to determine which files in the folder are imported (blank means all files)
- An archive folder to store files that are successfully imported
- Whether to add a timestamp to archived files or not

On the **Mapping** tab, you specify one set of mapping columns that must apply to all files to be imported. When automatically generating the mappings, Axiom Software will use the first matching file that it finds in the source folder, based on the specified import order.

▶ Import behavior

When importing multiple files, the import behaves as follows:

- When the import begins, Axiom Software inspects the source folder and compiles a list of all files found in that folder. If a file filter is specified, the list is limited to only those files that match the filter.
- The files in the list are then processed sequentially, in the configured order (either alphabetical by file name, or by creation date). For each file, the following occurs.
    - Data is copied from the source file into the temptable.
    - Transform statements are run.
    - Data is saved to the destination table.
    - The source file is moved from the source folder to the archive folder.

  Each file must complete this import process successfully before moving on to the next file.
- If any file fails to import successfully, the import stops and the import status is set to Error. No further files are imported. Currently, it is not possible to configure the import to ignore the error and continue processing other files.
- If all files in the list import successfully, the import stops and the import status is set to Success.

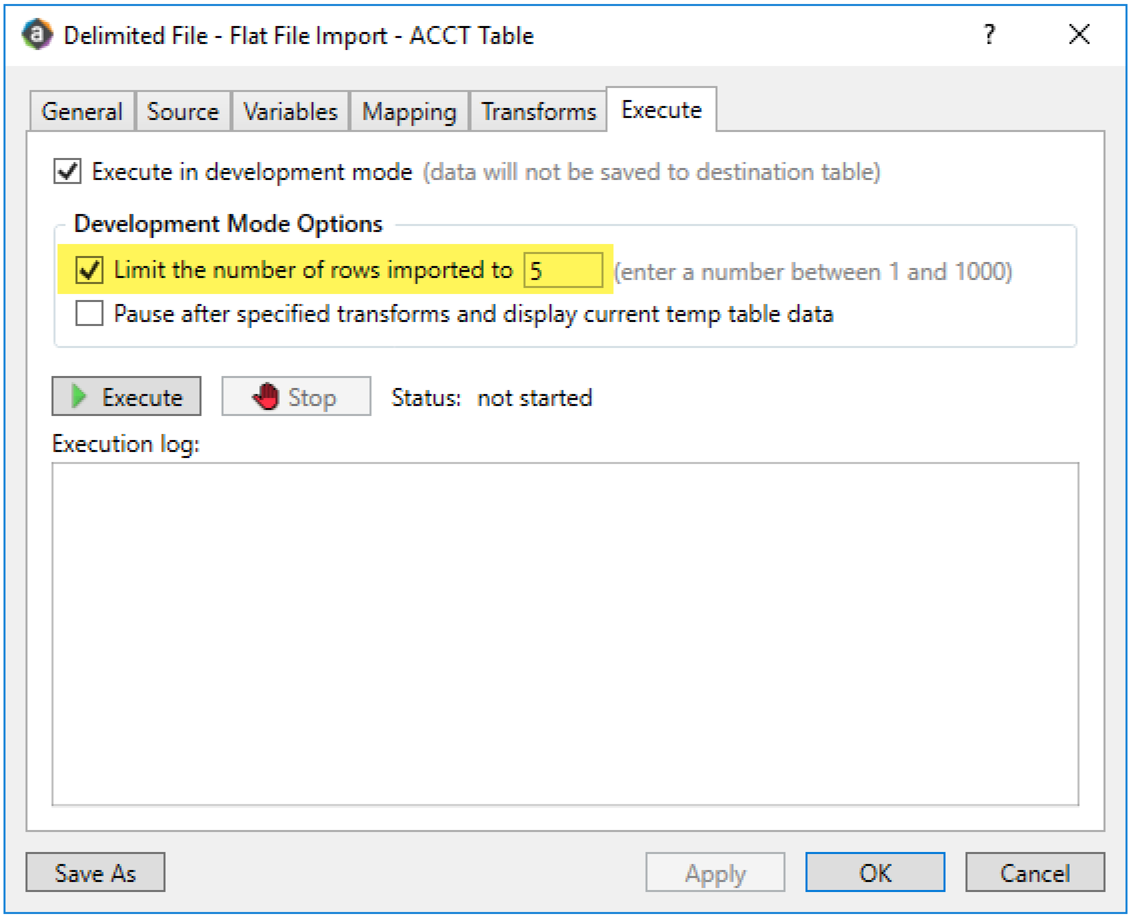
Limit rows when testing a delimited file import

When testing or troubleshooting a delimited file import, you can now limit the rows to process. This can help speed up the iterative testing process.

As part of these changes, the option to run an import as "preview only" has been renamed to "development mode". When you run an import in development mode, the behavior is the same as before—all steps of the import are performed, except for the final step that saves data to the destination table. Development mode is intended to help you test new imports and troubleshoot issues with existing imports.

When you enable **Execute in development mode** on the **Execute** tab, the **Development Mode Options** section becomes available:

- The "allow pauses" option has been renamed to **Pause after specified transforms and display current temp table data**.
- The "limit rows" option is only available when the import source is a delimited file. To use this option, enter a number of rows into the box to the right of **Limit the number of rows imported to**.



☑ Execute in development mode (data will not be saved to destination table)

Development Mode Options

☑ Limit the number of rows imported to 5 (enter a number between 1 and 1000)

□ Pause after specified transforms and display current temp table data

Status: not started

Execution log:

Example Execute tab for a delimited file import

By default, the delimited import is limited to 1000 rows when running in development mode. You can enter any number from 1 to 1000 to limit the number of rows to process.

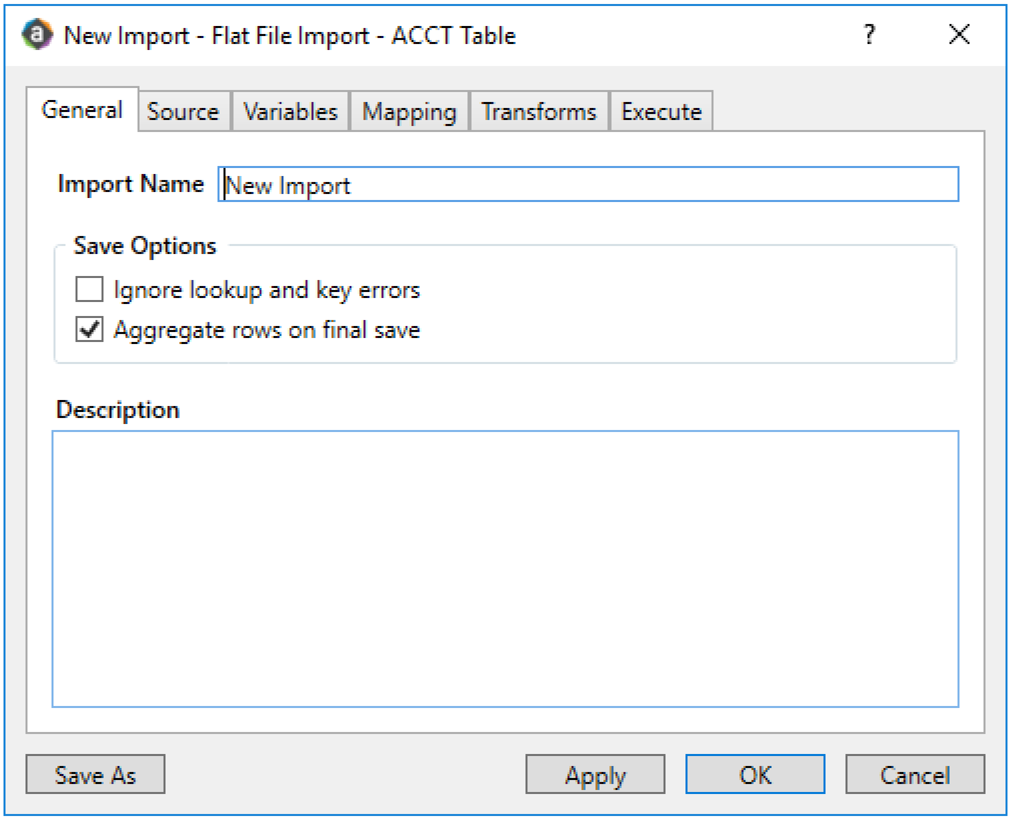
Reorganized Import Wizard and Export Wizard

As part of the enhancements made to imports, several areas of the Import Wizard were reorganized to streamline the choices in the user interface and to logically group related settings. Some of these changes also impact the Export Wizard.

When you upgrade, your existing import and export configurations will be retained and displayed within the reorganized dialogs. These are display changes only—no action is necessary to update existing assets.

New General tab for Import Wizard and Export Wizard

The Import Wizard has a new tab named **General**, to hold general settings such as the name, description, and options that affect the overall process. The **Name** field was moved from the top of the dialog to this new tab. Additionally, all general settings that affect how the import is performed have been moved from the **Execute** tab to this new tab.



Import Name New Import

Save Options

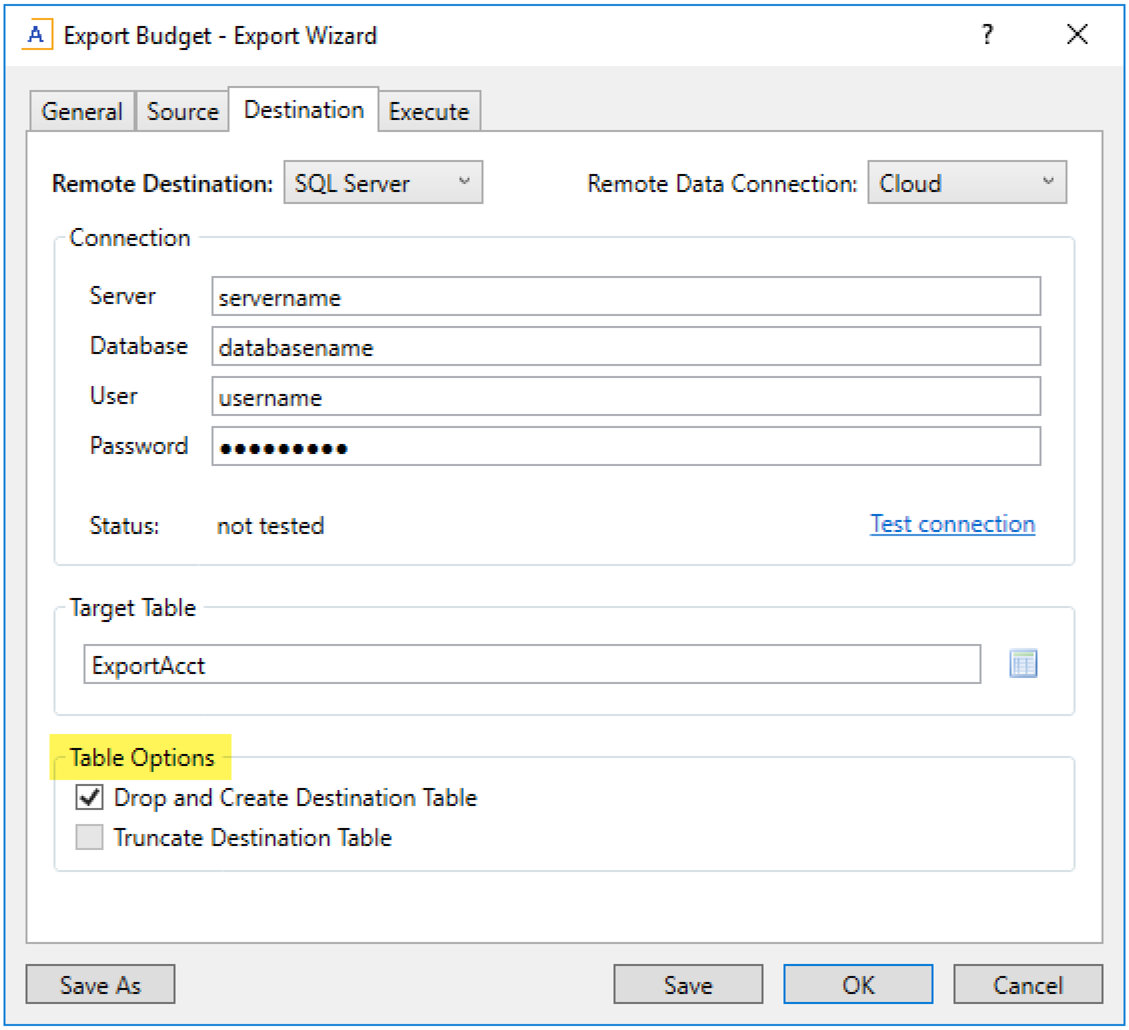
☐ Ignore lookup and key errors

☑ Aggregate rows on final save

Description

Example General tab for the Import Wizard

The Export Wizard also has the new General tab to define the name and description. However, the options that were previously on the Execute tab of the Export Wizard only apply when exporting data to a destination table, so these options have been moved to the **Destination** tab.



Remote Destination: SQL Server

Remote Data Connection: Cloud

Connection

Server servername

Database databasename

User username

Password ●●●●●●●●●●

Status: not tested

Target Table

ExportAcct

Table Options

☑ Drop and Create Destination Table

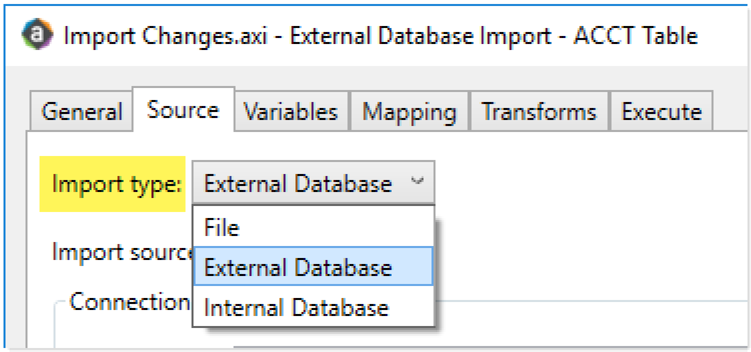
☐ Truncate Destination Table

Example Destination tab for the Export Wizard

The Execute tab now only provides the ability to execute the import or export. For imports, this also includes the ability to execute the import in development mode (formerly known as preview mode).

▶ Updated Source tab for Import Wizard

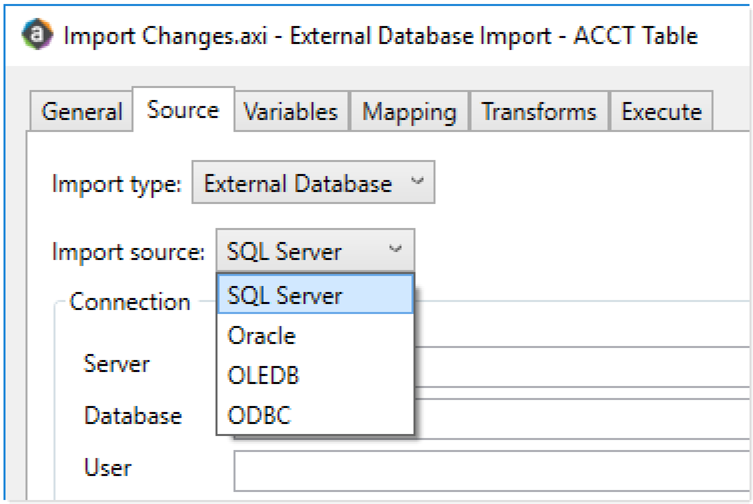
The Source tab of the Import Wizard has been reorganized to streamline the choices in the user interface. Instead of selecting from a long list of specific import sources, you now select the general **Import Type**.



Import type: External Database

New option to select the general type of import

Once you select the import type, you then select the specific import source. For example, if the type is External Database, you can select SQL Server, Oracle, and so on. The options available in the Source tab depend on the selected type and source.



Import type: External Database

Import source: SQL Server

Example import source selection based on the import type

The previous import source options are now organized under the following import types:

- **File**: Delimited or Excel files
- **External Database**: SQL Server, Oracle, OLEDB, or ODBC
- **Internal Database**: Current Axiom system database or audit database
- **Intacct**: Intacct systems (only available with license)

Viewing the description in execute-only mode

Import and export utilities can be opened in execute-only mode. This mode displays the contents of the Execute tab, but hides all other settings.

The Execute dialog now contains a link that can be used to view the import or export description, since this description is no longer located on the Execute tab.

Additional import enhancements

When an import is configured to use a remote data connection, that connection also affects the following:

- **Mapping** tab: The remote data connection is used to obtain the source file when auto-generating column mappings. This makes it easier to set up mappings for Cloud Service systems when configuring the import.
- **Source** tab: When browsing for a file or folder path, by default the file explorer dialog shows the folder structure of the server where the Axiom Cloud Integration Service is installed. This makes it easier to specify paths that reside on that server.

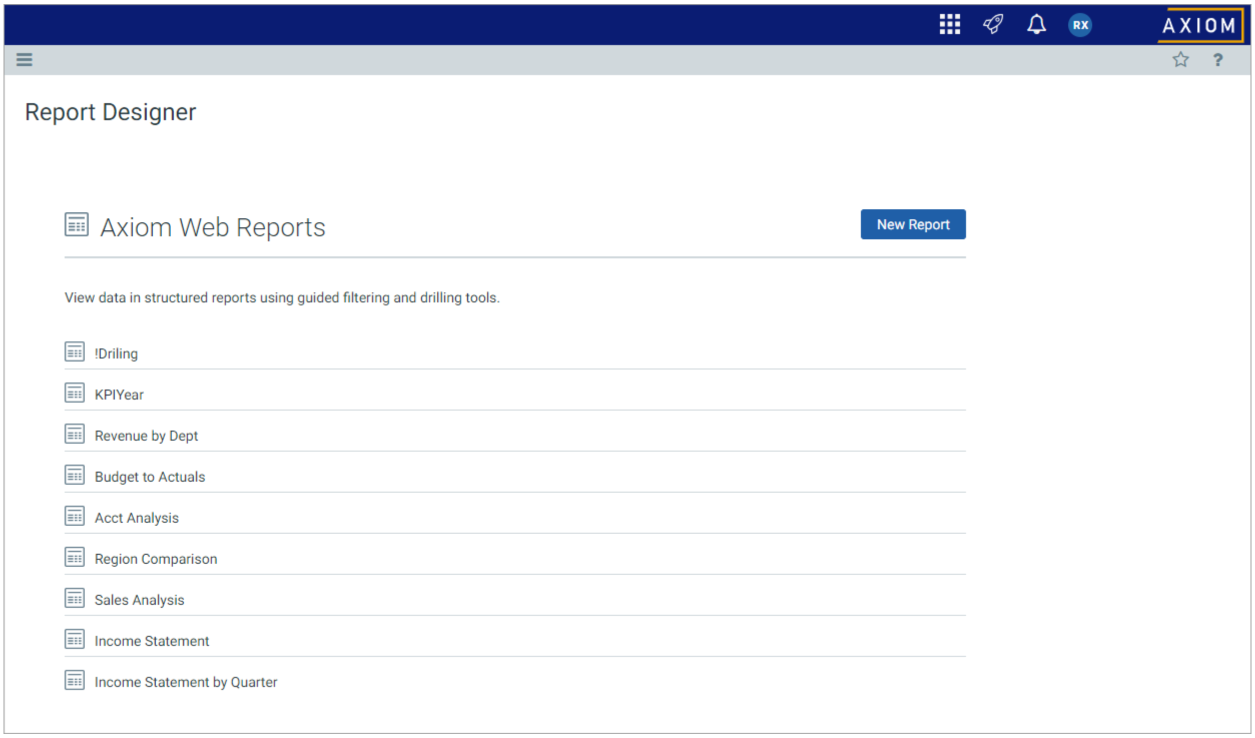
Web reports

This section details the new features and enhancements made to web reports.

Revised landing page to open and create web reports

The former Report Builder page used to open and create web reports has been revised and renamed. These changes were made as part of our phased introduction of Axiom Intelligence reporting, which is currently being rolled out to a limited set of healthcare products. In future releases, we plan to further redesign the report landing pages in the Web Client to provide easier access to files and general ease-of-use improvements.

The Report Builder page is now known as the **Report Designer**. Additionally, this page has been streamlined for improved display and to make it easier to quickly create or open reports. Most users only have access to a handful of web reports which now display prominently, without the unnecessary folder overhead.



Report Designer

Axiom Web Reports

View data in structured reports using guided filtering and drilling tools.

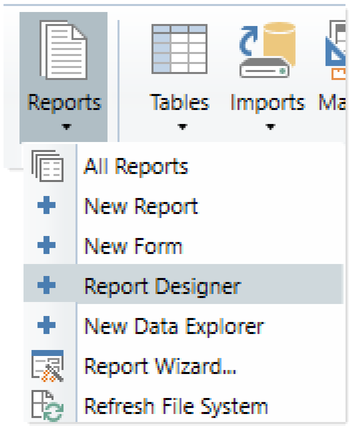
- !Driling
- KPIYear
- Revenue by Dept
- Budget to Actuals
- Acct Analysis
- Region Comparison
- Sales Analysis
- Income Statement
- Income Statement by Quarter

Revised page to open and create web reports

Reports are displayed in last-modified order, with the most recently created or modified reports on top. You can hover your cursor over a report to see when the report was last modified and by whom.

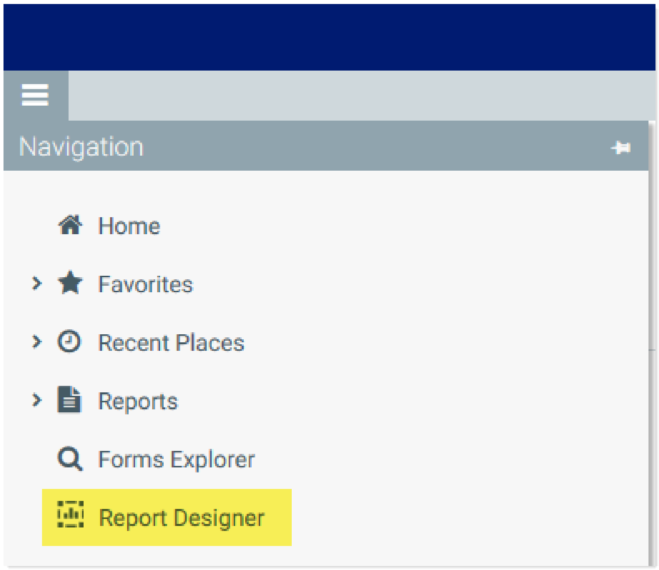
The Report Designer page is located at the same URL, and can still be accessed using all of the same methods as before:

- In the Desktop Client, click **Reports > Report Designer** to open this page. The name of this menu item is automatically updated when using the default Axiom ribbon tab, or when using a custom ribbon tab with default display text.



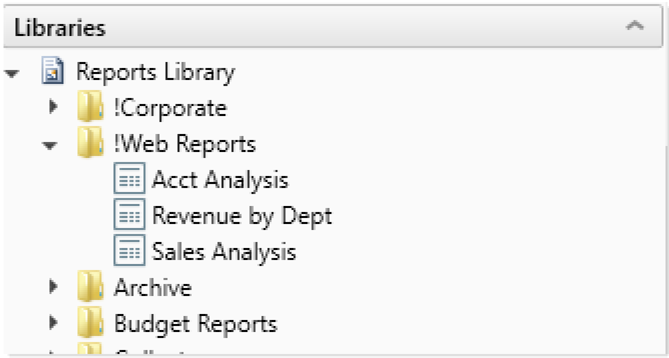
NOTE: In previous releases, clicking the Report Builder menu item opened the page and automatically started the report creation wizard. Starting in 2019.1, clicking Report Designer simply opens the page. You can then click **New Report** to start the report creation wizard if desired.

- In the Web Client, click **Report Designer** or **Report Builder** on the Navigation panel to open this page.



The name of this menu item is only updated automatically in newly created systems. Also, if your system has installed products, the product update may provide an updated Navigation panel. In all other cases, the menu item will remain as Report Builder until it is manually updated, though it will still go to the correct place regardless of the name. See the following section for information on how to update your Navigation panel for this new name.

In addition to the redesigned page, note that the web reports icon has been changed. This change also applies to the Desktop Client:

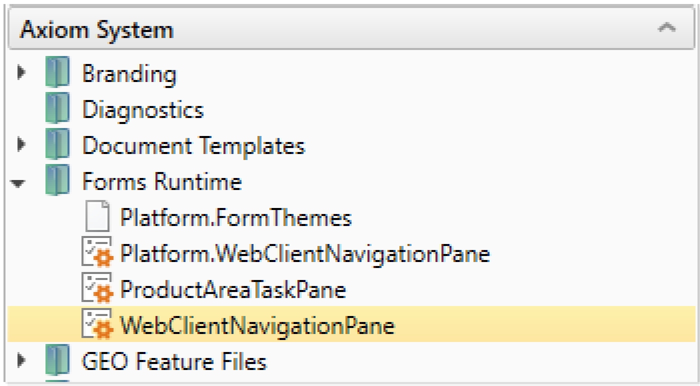


Libraries
Reports Library
!Corporate
!Web Reports
Acct Analysis
Revenue by Dept
Sales Analysis
Archive
Budget Reports

New web reports icon in Reports Library

Updating the Navigation panel in your system

The contents of the Navigation panel are controlled by the `WebClientNavigationPane.axl` file, located in the following folder: `\Axiom\Axiom System\Forms Runtime`. This is a custom task pane file that gets rendered in the Navigation panel in the Web Client.



Axiom System
Branding
Diagnostics
Document Templates
Forms Runtime
Platform.FormThemes
Platform.WebClientNavigationPane
ProductAreaTaskPane
WebClientNavigationPane
GEO Feature Files

Because the contents of the Navigation panel are customizable, Axiom Software does not automatically overwrite this file when updates are made. Instead, the most current version of the file is placed in this folder as `Platform.WebClientNavigationPane.axl`. In brand new systems, both files are the same, but in existing systems only `Platform.WebClientNavigationPane.axl` has the latest updates.

If your organization has not customized `WebClientNavigationPane.axl`, then you can update your system as follows:

1. Delete `WebClientNavigationPane.axl`.
2. Copy `Platform.WebClientNavigationPane.axl`.
3. Rename the copy to `WebClientNavigationPane.axl`.

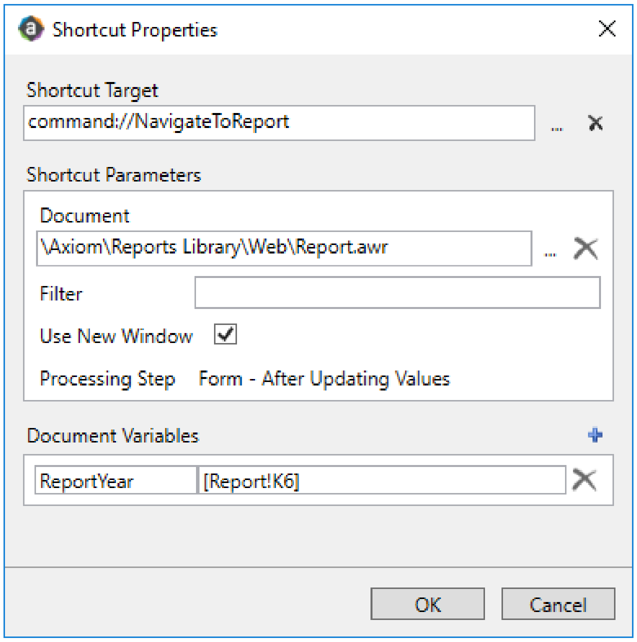
If your organization has customized `WebClientNavigationPane.axl`, then you can simply edit this file as needed to update the display text for any item that uses the former **Navigation - Report Builder** command. There is a new command named **Navigation - Report Designer**, but since both items go to the same place, it is not necessary to update the actual command unless you want to.

Only administrators can access the Axiom System area and modify the system files discussed in this section.

Passing a year value to a web report

When you open a web report from an Axiom form, you can now pass a default value for the Year refresh variable, so that the report is refreshed for the specified year.

The easiest way to do this is to use the Navigate to Report command with the reserved document variable name **ReportYear**. In the following example, the Navigate to Report command is configured to open a web report and pass a specified value for ReportYear. The value for ReportYear uses a bracketed cell reference to read the value from the form source file, so that the value can dynamically change based on inputs made to the form.



Shortcut Target: command://NavigateToReport

Shortcut Parameters

Document: \Axiom\Reports Library\Web\Report.awr

Filter:

Use New Window ☑

Processing Step: Form - After Updating Values

Document Variables

ReportYear	[Report!K6]

Example Navigate to Report command that passes a value for ReportYear

When this command is executed, the specified web report is opened, and the value for the Year refresh variable is set to the value passed from the form. It does not matter what the name of the Year refresh variable is—since each web report can only have one Year refresh variable, the ReportYear document variable is automatically associated with it.

Alternatively, you can manually append the ReportYear parameter to a generated URL for a web report, using the syntax `&ReportYear=`*`year`*. You can use the Axiom function GetWebReportDocumentURL to generate the URL, and then use a formula to append the parameter. For example:

```
=GetWebReportdocumentURL("\Axiom\Reports
Library\Web\Report.awr")&"&ReportYear=2018"
```

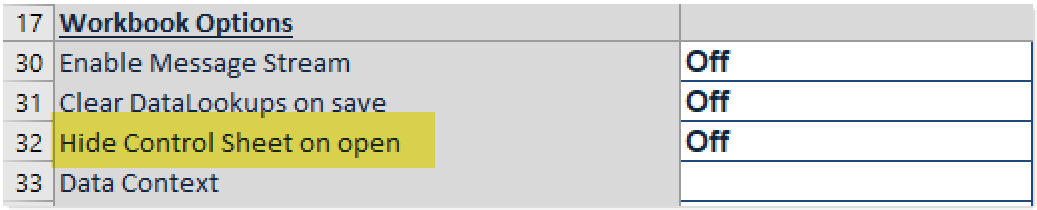
This example formula generates a URL such as:

```
https://mycompany.axiomepmcloud.com/reports?docref=9SCqQ5RkUXM7$wDGZcP0g
Hl-OSvgIU6of$t-GAeS4ujBRzGg0LBjXPBtivoiYJ$qa3Xa$EmT39se7o5mSQAm-A_
&reportyear=2018
```

Additional enhancements

Control Sheets no longer hidden by default for Sheet Assistant users

The file-level option **Hide Control Sheet on open** determines whether the Control Sheet is hidden by default for all users when the file is opened. Going forward, the default setting for this option is **Off**. This applies to all newly created spreadsheet files, and any existing files where the setting is blank or not present.



17	Workbook Options	
30	Enable Message Stream	Off
31	Clear DataLookups on save	Off
32	Hide Control Sheet on open	Off
33	Data Context	

When this option is disabled, the visibility of the Control Sheet depends on whether the current user has the **Allow Sheet Assistant** permission for the file (as specified on the **Files** tab of security). This default behavior works as follows:

- If you are an administrator or you have the **Allow Sheet Assistant** permission, *and* you open the document as read/write, then the Control Sheet is visible. This behavior is intended to provide report writers with easy access to configuration settings on the Control Sheet.
- Otherwise, the Control Sheet is hidden. In this case, users with **Allow Sheet Assistant** permission can unhide the Control Sheet by double-clicking a field name on the Sheet Assistant. And as always, any user with **Allow Unprotect** permission can manually unhide the sheet.

This default behavior means that in most cases, there is no longer any need to enable **Hide Control Sheet on open**. However, if you do have a situation where you need the Control Sheet to always hide, for all users, then you can set this option to **On**.

For existing files, this behavior change only applies to files where the **Hide Control Sheet on open** option is blank or not present (because it will now default to Off instead of On). Additionally, only administrators and users with access to the Sheet Assistant will notice any change in behavior. In most cases it is beneficial and desired to show the Control Sheet by default to these users. End users who do not have access to the Sheet Assistant will continue to not see the Control Sheet, regardless of how this option is configured.

If you have an existing file where **Hide Control Sheet on open** is *not* set to **On**, but the Control Sheet is still hiding for all users on open, this may mean that the file was saved using the previous default behavior (or that the Control Sheet was manually hidden using Excel functionality). In this case, you can open the file, unhide the Control Sheet, and then save the file. The next time the file is opened, it should now use the default behavior where Sheet Assistant access controls the visibility of the Control Sheet.

> **NOTE:** This behavior does not apply to plan files. The Control Sheet is always hidden in plan files for all users. It is rarely necessary to access the Control Sheet in plan files, because any configuration changes for plan files need to be made in the template.

▶ Configure visibility for dynamic items in a task pane

Using the existing **Configure Visibility** feature for task panes, you can hide certain items in a task pane without needing to delete the items. For example, a task pane might have items that are only relevant at certain times of year. You can hide these items when they are not currently needed, and then show them again later.

The Configure Visibility feature now supports configuring visibility for dynamically generated items in a task pane, such as when a task pane item links to a report folder. When the task pane is viewed, the folder item is dynamically populated with all the files in that folder. In previous releases, the Configure Visibility dialog would have only allowed hiding or showing the entire folder. Now, you can optionally hide or show the individual files in the folder.

▶ Use variables in default values for Copy On Demand Plan Files Scheduler task

When configuring column default values for the Copy On Demand Plan Files Scheduler task, you can now use variables in this section. You can use bracketed column names and/or file group variables, just like you can when configuring the Copy On Demand Plan Files command in a task pane. Additionally, you can use Scheduler job variables.

▶ Restriction enforced on post-query sum by for Axiom queries

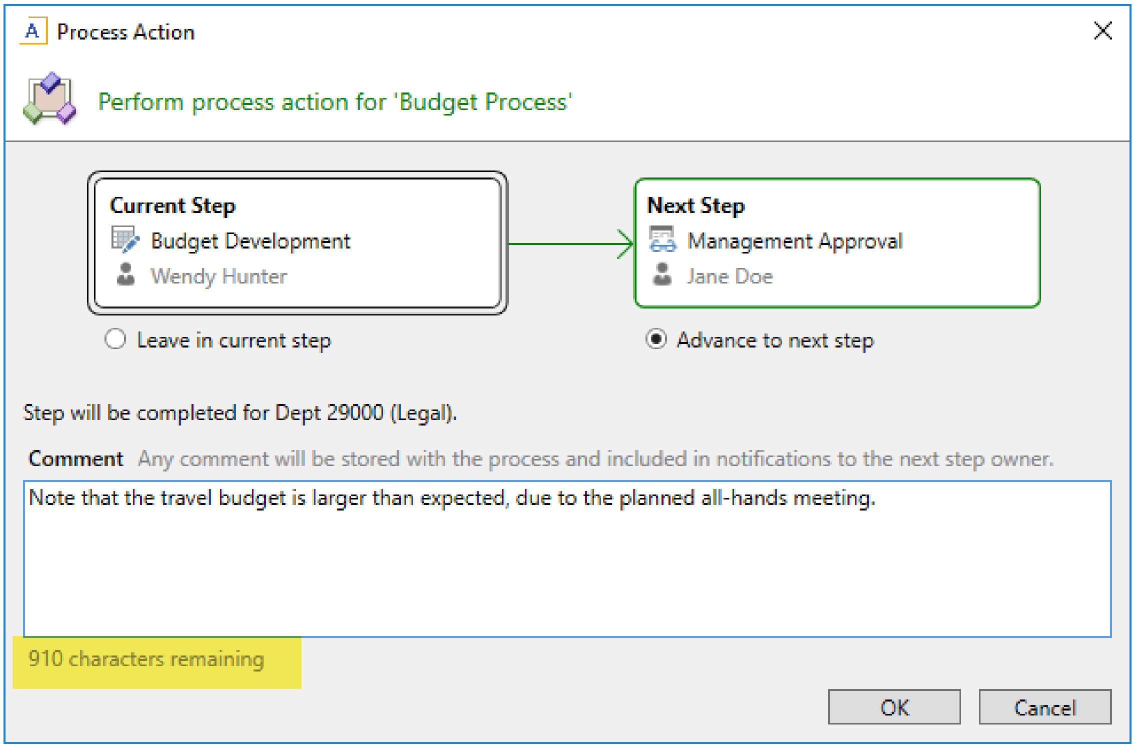
Axiom queries have an advanced setting that allows you to set a sum by level that applies after the initial database query is made. The purpose of this setting is to support grouping by segments, when using another advanced feature to organize Axiom query data into ranked segments. Although this was the only intended use case, the setting was not restricted and allowed entries other than the segment column.

Going forward, the **Post Query Sum By** setting for Axiom queries now only honors segment column names. If a non-segment column name is used in this setting, it is ignored and the query does not error. Segment column names are defined using special syntax in the Axiom query field definition, as described in the following topic in Axiom Software Help: *Organizing Axiom query data into ranked segments* (AX1117).

This change should not affect any clients, since the setting still works as expected for its intended use case, and we are unaware of any other needed use case. Additionally, because it is an advanced setting that is only present on the Control Sheet, it is unlikely to have been used by accident. However, if you identify a report that no longer works as expected because a non-segment column is used in the Post Query Sum By, please contact Kaufman Hall Software Support for assistance.

Other changes

- When completing a process task and defining a comment, the **Process Action** dialog now displays the remaining allowed characters for the comment. Comments are limited to 1000 characters.



Perform process action for 'Budget Process'

Current Step: Budget Development, Wendy Hunter

Next Step: Management Approval, Jane Doe

○ Leave in current step

● Advance to next step

Step will be completed for Dept 29000 (Legal).

Comment Any comment will be stored with the process and included in notifications to the next step owner.

Note that the travel budget is larger than expected, due to the planned all-hands meeting.

910 characters remaining

- Various minor display improvements were made to the Scheduler area of the Web Client, such as improved spacing between elements, removal of duplicated title text, and better alignment of job results.

- The **Document History** dialog has been updated to hide the **Open** button for non-spreadsheet-based files. This change was made because the ability to open prior versions in Axiom Software was originally designed for spreadsheet-based files, and does not work as expected in all cases for non-spreadsheet-based files. Other file types can still be exported if you need to view or restore them. Additionally, the **Export** button is now available to users with read/write permission to the file, for non-spreadsheet-based files.