What's New
Axiom Software
Version 2018.4

KaufmanHall
axiom software

Contents

Introduction

Kaufman Hall is pleased to announce the release of **Axiom Software Version 2018.4**. This release features enhancements to various areas of the software, such as:

- **Axiom forms:**
  - New component to create a fixed-row report that queries data directly from the database
  - Expanded functionality for the Data Grid component, including an optional total row, the ability to select multiple rows, and the ability to report on plan file process information
  - Standard icons in the task bar to trigger a save-to-database or generate a PDF
- **Report Builder:** Ability to dynamically change years in a web report using a Year refresh variable, and support for PDF generation
- **Axiom queries:** Process multiple queries in parallel instead of sequentially, by using batch query processing
- **Scheduler:** Monitor the Scheduler job queue and run jobs on demand from a browser, using the Web Client
- **OData API:** Access Axiom Software data in custom solutions using the standard Open Data Protocol

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.

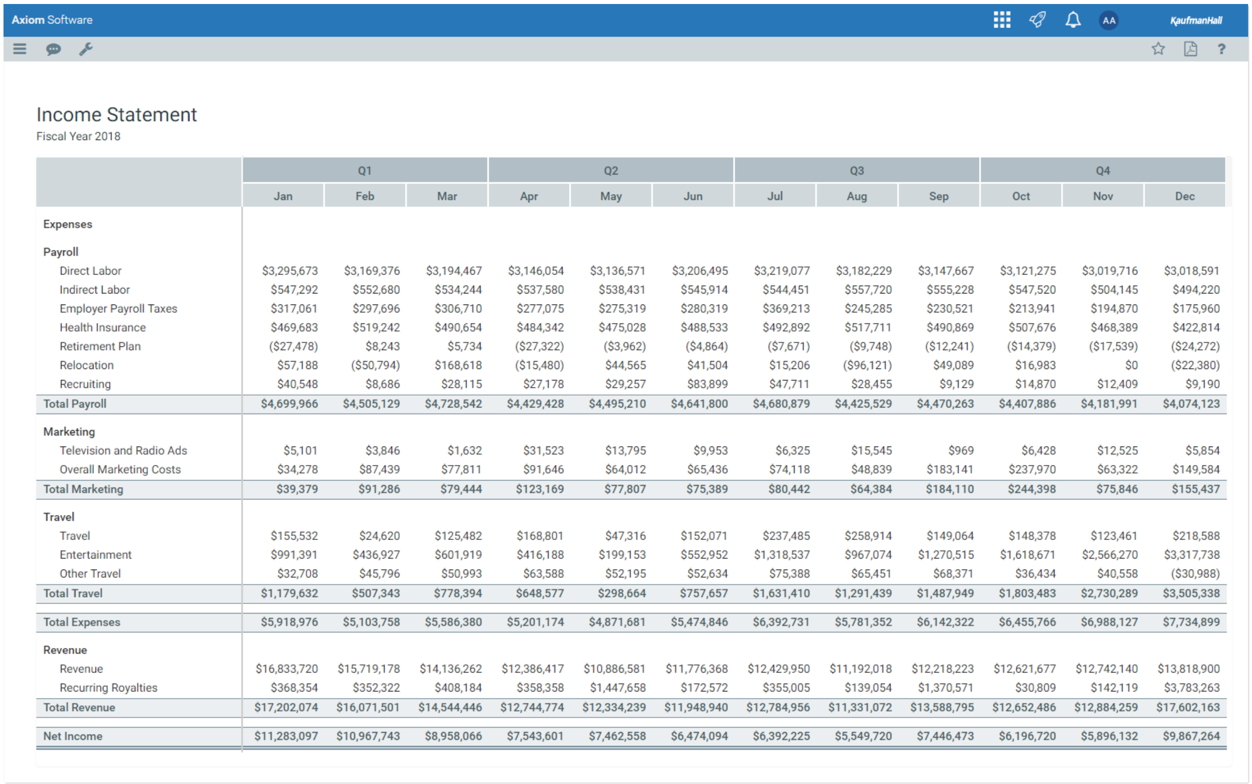
IMPORTANT: Before upgrading to version 2018.4, make sure you have reviewed the separate *Release Notes* document to understand any important technical changes and upgrade considerations in this release.

Axiom forms

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

New component to create fixed-row reports

Using the new Fixed Report component, you can query data from the Axiom Software database and display that data in a fixed-row report structure, within an Axiom form. The Fixed Report component can be used to display an income statement or any other report that uses a fixed format, including headers, subheaders, subtotals, and totals.



Income Statement

Fiscal Year 2018

	Q1			Q2			Q3			Q4		
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Expenses												
Payroll												
Direct Labor	$3,295,673	$3,169,376	$3,194,467	$3,146,054	$3,136,571	$3,206,495	$3,219,077	$3,182,229	$3,147,667	$3,121,275	$3,019,716	$3,018,591
Indirect Labor	$547,292	$552,680	$534,244	$537,580	$538,431	$545,914	$544,451	$557,720	$555,228	$547,520	$504,145	$494,220
Employer Payroll Taxes	$317,061	$297,696	$306,710	$277,075	$275,319	$280,319	$369,213	$245,285	$230,521	$213,941	$194,870	$175,960
Health Insurance	$469,683	$519,242	$490,654	$484,342	$475,028	$488,533	$492,892	$517,711	$490,869	$507,676	$468,389	$422,814
Retirement Plan	($27,478)	$8,243	$5,734	($27,322)	($3,962)	($4,864)	($7,671)	($9,748)	($12,241)	($14,379)	($17,539)	($24,272)
Relocation	$57,188	($50,794)	$168,618	($15,480)	$44,565	$41,504	$15,206	($96,121)	$49,089	$16,983	$0	($22,380)
Recruiting	$40,548	$8,686	$28,115	$27,178	$29,257	$83,899	$47,711	$28,455	$9,129	$14,870	$12,409	$9,190
Total Payroll	**$4,699,966**	**$4,505,129**	**$4,728,542**	**$4,429,428**	**$4,495,210**	**$4,641,800**	**$4,680,879**	**$4,425,529**	**$4,470,263**	**$4,407,886**	**$4,181,991**	**$4,074,123**
Marketing												
Television and Radio Ads	$5,101	$3,846	$1,632	$31,523	$13,795	$9,953	$6,325	$15,545	$969	$6,428	$12,525	$5,854
Overall Marketing Costs	$34,278	$87,439	$77,811	$91,646	$64,012	$65,436	$74,118	$48,839	$183,141	$237,970	$63,322	$149,584
Total Marketing	**$39,379**	**$91,286**	**$79,444**	**$123,169**	**$77,807**	**$75,389**	**$80,442**	**$64,384**	**$184,110**	**$244,398**	**$75,846**	**$155,437**
Travel												
Travel	$155,532	$24,620	$125,482	$168,801	$47,316	$152,071	$237,485	$258,914	$149,064	$148,378	$123,461	$218,588
Entertainment	$991,391	$436,927	$601,919	$416,188	$199,153	$552,952	$1,318,537	$967,074	$1,270,515	$1,618,671	$2,566,270	$3,317,738
Other Travel	$32,708	$45,796	$50,993	$63,588	$52,195	$52,634	$75,388	$65,451	$68,371	$36,434	$40,558	($30,988)
Total Travel	**$1,179,632**	**$507,343**	**$778,394**	**$648,577**	**$298,664**	**$757,657**	**$1,631,410**	**$1,291,439**	**$1,487,949**	**$1,803,483**	**$2,730,289**	**$3,505,338**
Total Expenses	**$5,918,976**	**$5,103,758**	**$5,586,380**	**$5,201,174**	**$4,871,681**	**$5,474,846**	**$6,392,731**	**$5,781,352**	**$6,142,322**	**$6,455,766**	**$6,988,127**	**$7,734,899**
Revenue												
Revenue	$16,833,720	$15,719,178	$14,136,262	$12,386,417	$10,886,581	$11,776,368	$12,429,950	$11,192,018	$12,218,223	$12,621,677	$12,742,140	$13,818,900
Recurring Royalties	$368,354	$352,322	$408,184	$358,358	$1,447,658	$172,572	$355,005	$139,054	$1,370,571	$30,809	$142,119	$3,783,263
Total Revenue	**$17,202,074**	**$16,071,501**	**$14,544,446**	**$12,744,774**	**$12,334,239**	**$11,948,940**	**$12,784,956**	**$11,331,072**	**$13,588,795**	**$12,652,486**	**$12,884,259**	**$17,602,163**
Net Income	**$11,283,097**	**$10,967,743**	**$8,958,066**	**$7,543,601**	**$7,462,558**	**$6,474,094**	**$6,392,225**	**$5,549,720**	**$7,446,473**	**$6,196,720**	**$5,896,132**	**$9,867,264**

Example form with Fixed Report component

The Fixed Report component queries the data directly from the database, using the primary table defined in the component properties, and the column and row information defined in the associated data sources. The resulting data is not returned into the spreadsheet source file; it is only returned into the form. This provides a more efficient and performant method of displaying data in an Axiom form, as compared to querying data into the spreadsheet source file and then tagging it for display in a Formatted Grid component.

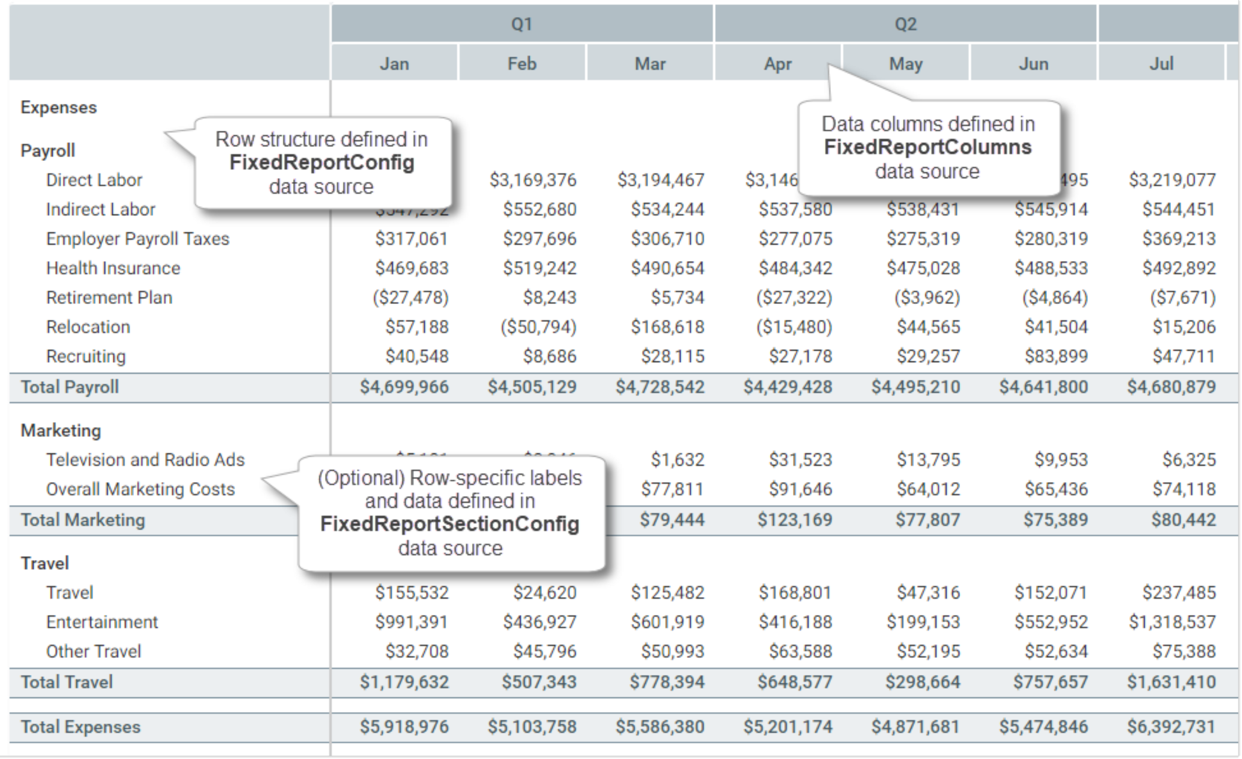
The setup and behavior of the Fixed Report component is similar to the existing Data Grid component, and supports many of the same features—such as built-in data drilling, selected rows, export to spreadsheet, and icons. Form designers who are already familiar with the Data Grid component can leverage their existing knowledge to quickly learn this new component.

This section provides an overview of how to setup and use this new component. For more detailed information, see the following topic in Axiom Software Help: *Fixed Report component* (AX1740).

▶ Overview of fixed report creation

Defining a Fixed Report component is a multiple-step process that requires the following:

- Creation of a **FixedReportConfig** data source in the spreadsheet to define the row structure of the report, including headers, subheaders, data row sections, subtotals, and totals. This process may include creation of one or more optional **FixedReportSectionConfig** data sources, to individually define the rows in a section, as opposed to dynamically generating the rows based on the values in a specified table column.
- Creation of a **FixedReportColumns** data source in the spreadsheet to define the columns to display in the report, as well as various display properties for those columns.
- Placement and configuration of a Fixed Report component on the Axiom form canvas. The primary table for the query and the overall data filter are defined in the component properties.



	Q1			Q2			
	Jan	Feb	Mar	Apr	May	Jun	Jul
Expenses							
Payroll							
Direct Labor		$3,169,376	$3,194,467	$3,146		495	$3,219,077
Indirect Labor	$547,292	$552,680	$534,244	$537,580	$538,431	$545,914	$544,451
Employer Payroll Taxes	$317,061	$297,696	$306,710	$277,075	$275,319	$280,319	$369,213
Health Insurance	$469,683	$519,242	$490,654	$484,342	$475,028	$488,533	$492,892
Retirement Plan	($27,478)	$8,243	$5,734	($27,322)	($3,962)	($4,864)	($7,671)
Relocation	$57,188	($50,794)	$168,618	($15,480)	$44,565	$41,504	$15,206
Recruiting	$40,548	$8,686	$28,115	$27,178	$29,257	$83,899	$47,711
Total Payroll	$4,699,966	$4,505,129	$4,728,542	$4,429,428	$4,495,210	$4,641,800	$4,680,879
Marketing							
Television and Radio Ads			$1,632	$31,523	$13,795	$9,953	$6,325
Overall Marketing Costs			$77,811	$91,646	$64,012	$65,436	$74,118
Total Marketing			$79,444	$123,169	$77,807	$75,389	$80,442
Travel							
Travel	$155,532	$24,620	$125,482	$168,801	$47,316	$152,071	$237,485
Entertainment	$991,391	$436,927	$601,919	$416,188	$199,153	$552,952	$1,318,537
Other Travel	$32,708	$45,796	$50,993	$63,588	$52,195	$52,634	$75,388
Total Travel	$1,179,632	$507,343	$778,394	$648,577	$298,664	$757,657	$1,631,410
Total Expenses	$5,918,976	$5,103,758	$5,586,380	$5,201,174	$4,871,681	$5,474,846	$6,392,731

Row structure defined in **FixedReportConfig** data source
Data columns defined in **FixedReportColumns** data source
(Optional) Row-specific labels and data defined in **FixedReportSectionConfig** data source

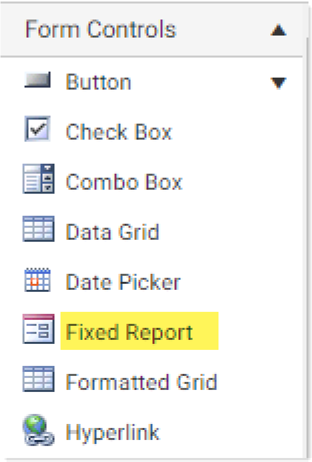
How the sections in a Fixed Report component correspond to its various data sources

This setup is very similar to the Data Grid component setup:

- The FixedReportColumns data source serves the same basic purpose as the DataGridColumns data source, and uses many of the same properties.
- The component properties of the Fixed Report component are very similar to the properties of the Data Grid component, including specifying the primary table and optional filter for the data query, and configuring report options such as row selection, drilling, and export to spreadsheet. The main difference is that instead of specifying a single data source for the Fixed Report component, you must specify two: the FixedReportColumns and FixedReportConfig data sources.
- Where the Data Grid component uses a simple row structure based on the sum by level, the Fixed Report component uses a separate FixedReportConfig data source to define each section of the report and its contents. Essentially, the sum by level is defined per section—or even per row, when using the FixedReportSectionConfig data source to define the data on a row-by-row basis.

Adding a Fixed Report component to the form canvas

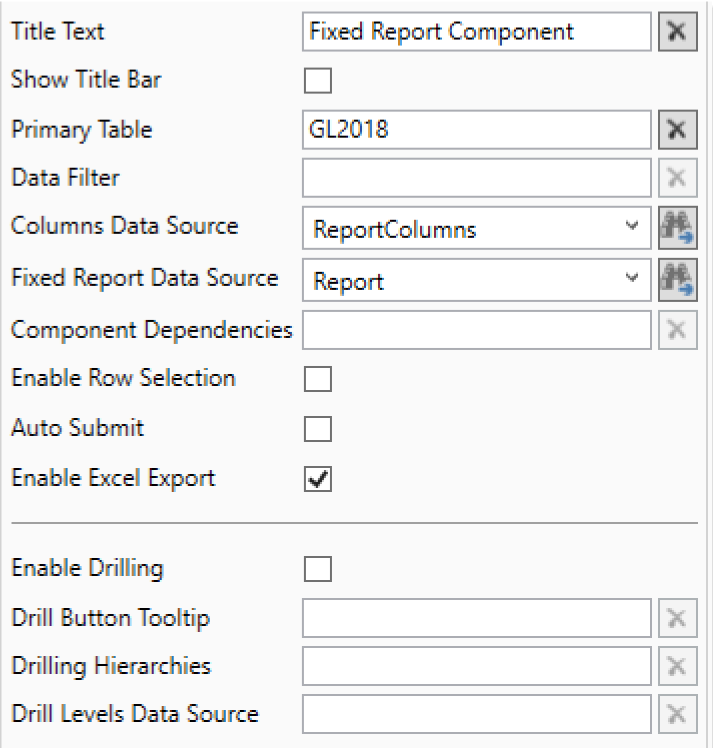
The Fixed Report component is available in the Form Controls section of the component sidebar:



New Fixed Report component in Form Designer

When configuring the component, you specify the following:

- The data sources for the component, to define the row structure of the report and the columns to display
- The primary table, to determine which columns are valid for inclusion in the report
- An optional data filter to limit the data returned by the query and displayed in the report
- Whether you want users to be able to select rows in the report, 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 report contents to an Excel spreadsheet
- Whether the report is dependent on one or more other form components, to refresh the data in the report based on changes made to those components
- Whether drilling is enabled, and if so various properties to determine the drilling options

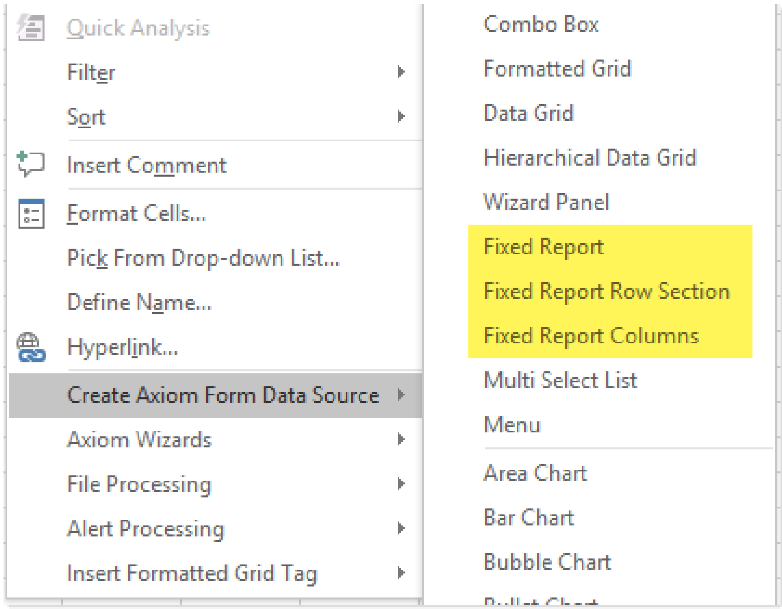


Title Text	Fixed Report Component
Show Title Bar	□
Primary Table	GL2018
Data Filter	
Columns Data Source	ReportColumns
Fixed Report Data Source	Report
Component Dependencies	
Enable Row Selection	□
Auto Submit	□
Enable Excel Export	☑
Enable Drilling	□
Drill Button Tooltip	
Drilling Hierarchies	
Drill Levels Data Source	

Example Data Grid component properties

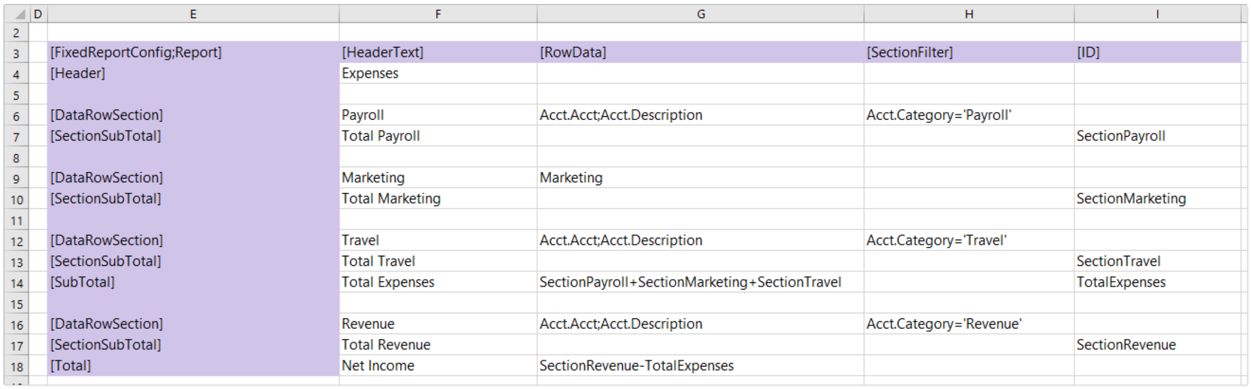
Defining the data sources for the Fixed Report component

The Fixed Report component uses three different data sources to define the report contents. Two of the data sources are required, and the third is optional. All three data sources can be added to a sheet using the right-click option **Create Axiom Form Data Source**:



New data sources for Fixed Report components

The first required data source, FixedReportConfig, defines the row structure of the report. (This corresponds to the **Fixed Report** option on the right-click menu.) This data source defines the headers, data row sections, subtotal rows, and total rows of the report. The following screenshot shows the FixedReportConfig data source used to generate the example fixed report shown earlier in the section.

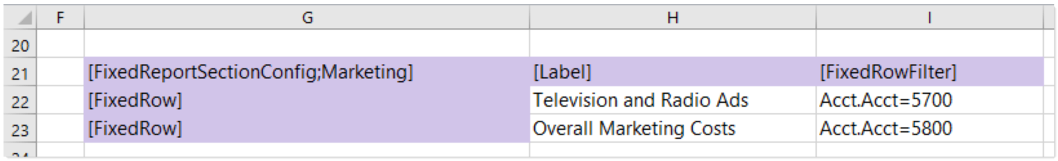


	D	E	F	G	H	I
2						
3		[FixedReportConfig;Report]	[HeaderText]	[RowData]	[SectionFilter]	[ID]
4		[Header]	Expenses			
5						
6		[DataRowSection]	Payroll	Acct.Acct;Acct.Description	Acct.Category='Payroll'	
7		[SectionSubTotal]	Total Payroll			SectionPayroll
8						
9		[DataRowSection]	Marketing	Marketing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

Example FixedReportConfig data source

Note the following:

- Data row sections can be defined by placing either a Table.Column name or a data source name in the `[RowData]` column.
  - If a column name is specified, the rows are dynamically generated based on the values in that column, limited by the `[SectionFilter]` if defined. For example, the Payroll section in this example uses column `Acct.Acct` to define row data, but only for accounts matching `Acct.Category='Payroll'`. So that section contains a row for each payroll account. The `Acct.Description` appended to the column name causes the description to display in the report instead of the account code.
  - If instead a data source name is specified, such as for the Marketing section in this example, then the rows for that section are defined in the named FixedReportSectionConfig data source. (This corresponds to the **Fixed Report Row Section** option on the right-click menu.) Using this optional data source, you can precisely specify the data to include in on each row of the section.

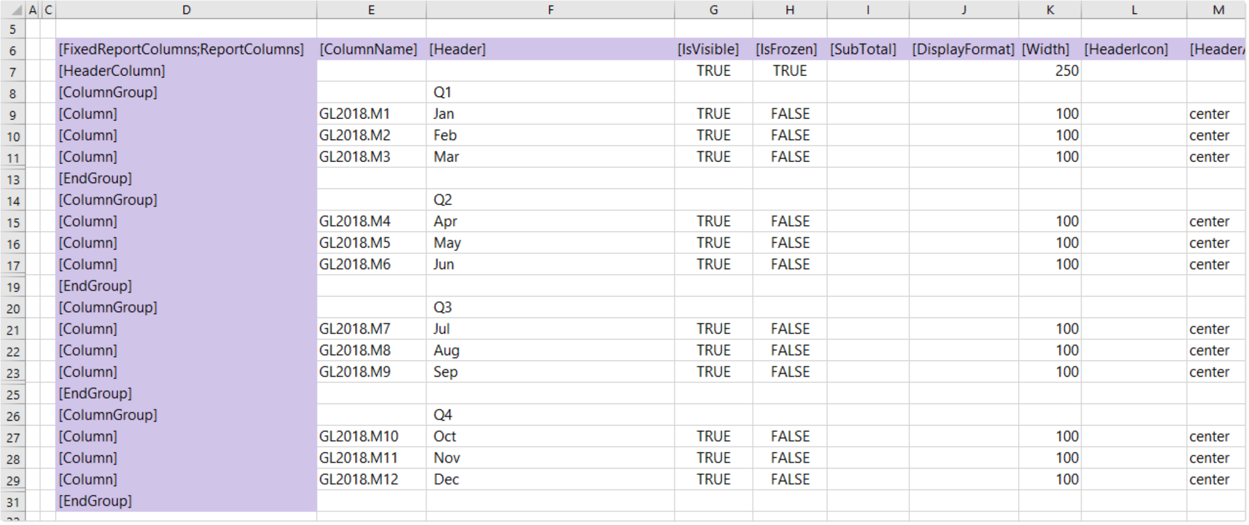


	F	G	H	I
20				
21		[FixedReportSectionConfig;Marketing]	[Label]	[FixedRowFilter]
22		[FixedRow]	Television and Radio Ads	Acct.Acct=5700
23		[FixedRow]	Overall Marketing Costs	Acct.Acct=5800

Example FixedReportSectionConfig data source

- Each section has a defined ID, which can be used in subtotal and total rows to define which sections to total and the calculation to apply. However, the section subtotal rows don't need to reference an ID, because they immediately follow the section they are meant to subtotal.

The second required data source, FixedReportColumns, defines the columns to display in the report. (This corresponds to the **Fixed Report Columns** option on the right-click menu.). This data source is very similar to the DataGridColumns data source, and uses the same row tags such as `[Column]` and `[CalculatedColumn]`.



	A	C	D	E	F	G	H	I	J	K	L	M
5												
6			[FixedReportColumns;ReportColumns]	[ColumnName]	[Header]	[IsVisible]	[IsFrozen]	[SubTotal]	[DisplayFormat]	[Width]	[HeaderIcon]	[Header/
7			[HeaderColumn]			TRUE	TRUE			250		
8			[ColumnGroup]		Q1							
9			[Column]	GL2018.M1	Jan	TRUE	FALSE			100		center
10			[Column]	GL2018.M2	Feb	TRUE	FALSE			100		center
11			[Column]	GL2018.M3	Mar	TRUE	FALSE			100		center
13			[EndGroup]									
14			[ColumnGroup]		Q2							
15			[Column]	GL2018.M4	Apr	TRUE	FALSE			100		center
16			[Column]	GL2018.M5	May	TRUE	FALSE			100		center
17			[Column]	GL2018.M6	Jun	TRUE	FALSE			100		center
19			[EndGroup]									
20			[ColumnGroup]		Q3							
21			[Column]	GL2018.M7	Jul	TRUE	FALSE			100		center
22			[Column]	GL2018.M8	Aug	TRUE	FALSE			100		center
23			[Column]	GL2018.M9	Sep	TRUE	FALSE			100		center
25			[EndGroup]									
26			[ColumnGroup]		Q4							
27			[Column]	GL2018.M10	Oct	TRUE	FALSE			100		center
28			[Column]	GL2018.M11	Nov	TRUE	FALSE			100		center
29			[Column]	GL2018.M12	Dec	TRUE	FALSE			100		center
31			[EndGroup]									

Example FixedReportColumns data source

This data source also uses a special row tag named `[HeaderColumn]`, which defines the column properties for the column that contains the row labels (as defined in the FixedReportConfig and FixedReportSectionConfig data sources). In our example report, this is the column that contains the account descriptions, as well as the general header text and total/subtotal labels.

Fixed report features

The Fixed Report component supports a variety of additional features. Most of these features are set up and work in the same way as the corresponding features for the existing Data Grid component.

- **Drilling:** You can enable drilling for the report. Users can drill down any data row in the report, to see the data at a different level of detail. You can use hierarchies to define drilling options, or define custom drilling options using a DrillLevels data source.
- **Icons and commands:** You can display icons in the report. The icons can be used simply as informational signals, or they can be used to trigger a command or open a designated URL. The icons can be persistent in the report, or they can display on hover only. Conditions can be defined for the icons, so that the icons only display when certain conditions are met.
- **Excel export:** You can enable the ability to export the report contents to an Excel spreadsheet.
- **Selected rows:** You can enable the ability for users to select a row in the report. This selected row can then be used to drive other contents of the form, such as to display detailed information about the current row in another panel.

Like the Data Grid component, the Fixed Report component does not refresh data on every form update. This behavior is intended to improve performance, so that data is only queried when needed. If you want the report to query the database again and refresh the data as part of a form update, you must configure the report to be dependent on another form component (using the **Component Dependencies** field in the component properties). Then when that component changes, the data in the report is refreshed. Otherwise, the only time a fixed report refreshes its data is when refresh variables are applied to the form.

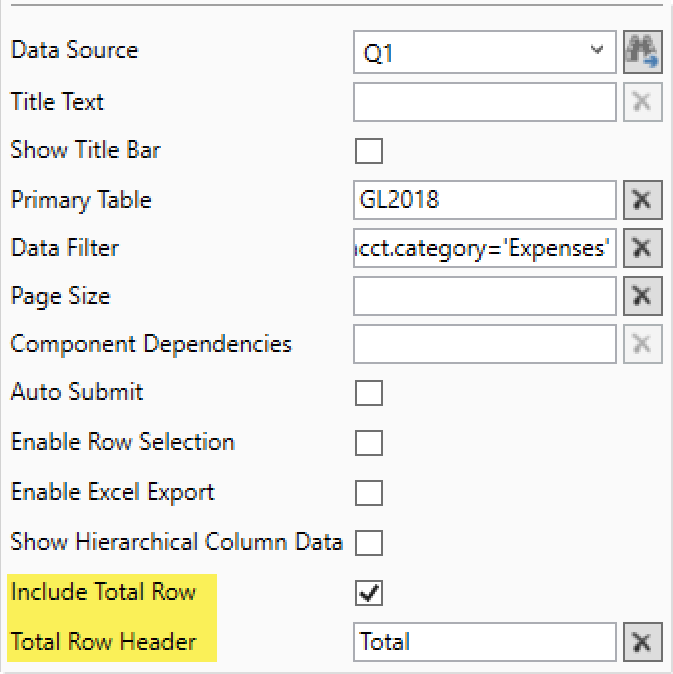
The Fixed Report component does not support user inputs, and uses standard report formatting that cannot be modified. If user inputs or customized formatting is required, a Formatted Grid component should be used instead.

Data Grid enhancements

Several enhancements were made to the Data Grid component for Axiom forms, to expand the available use cases for the grids.

Display a total row

You can now display a total row in the data grid, to see the column totals. The component properties have a new option to enable the total row, and to define a label for it.

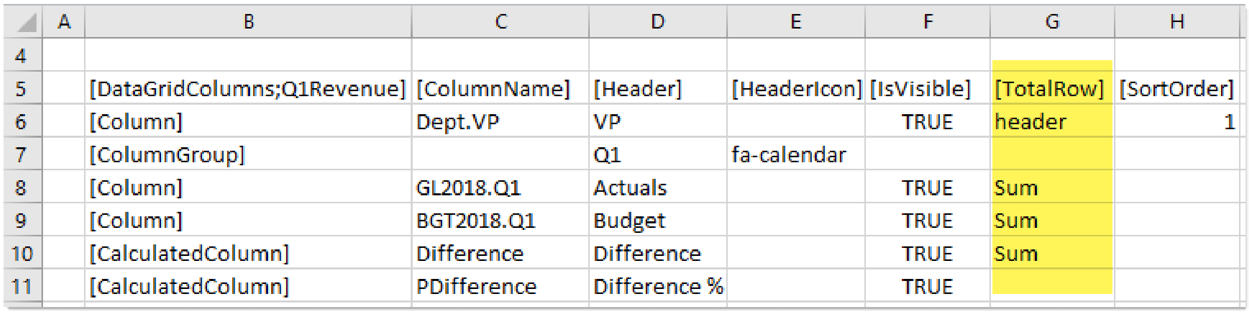


Data Source	Q1
Title Text	
Show Title Bar	☐
Primary Table	GL2018
Data Filter	cct.category='Expenses'
Page Size	
Component Dependencies	
Auto Submit	☐
Enable Row Selection	☐
Enable Excel Export	☐
Show Hierarchical Column Data	☐
Include Total Row	☑
Total Row Header	Total

New component properties for the total row

Once the total row is enabled, you can configure it using the new column `[TotalRow]` in the DataGridColumns data source. This column specifies which data grid columns should be included in the total row, and what type of aggregation should be used for the total. This column is included by default when adding new DataGridColumns data sources to a file. For existing data sources, you can add it manually.

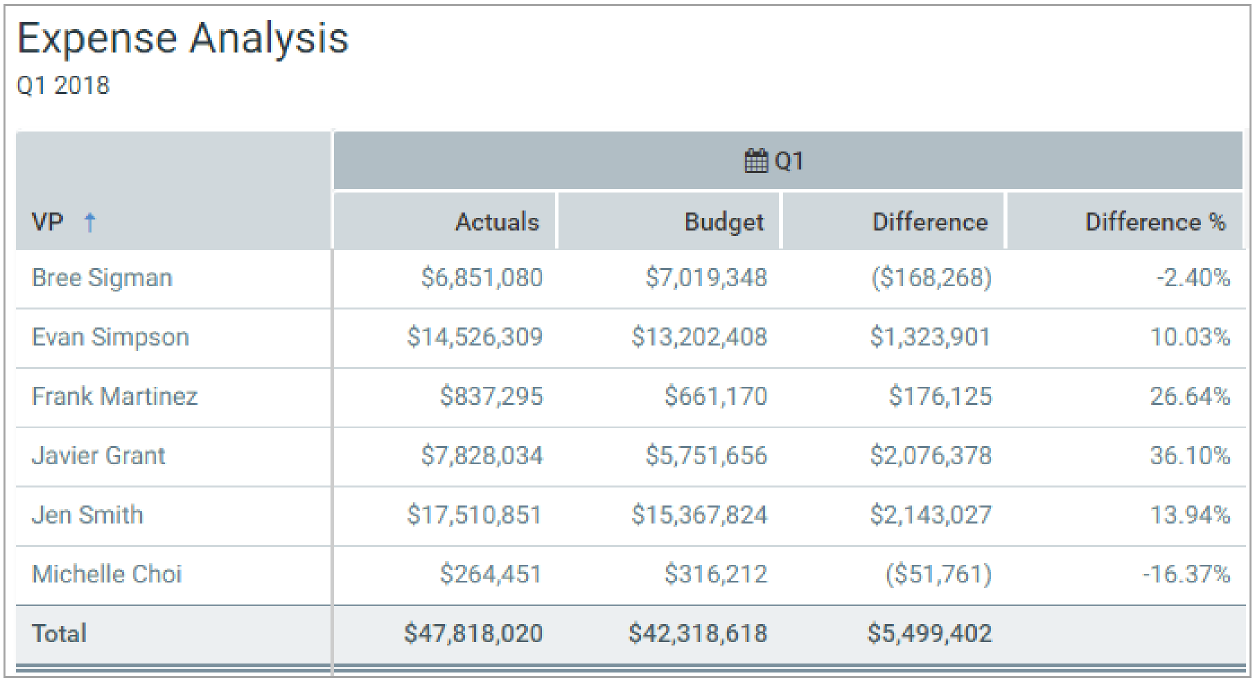
The following example shows a completed `[TotalRow]` column that places the total row label in the sum by column, and sums three of the data columns.



	A	B	C	D	E	F	G	H
4								
5		[DataGridColumns;Q1Revenue]	[ColumnName]	[Header]	[HeaderIcon]	[IsVisible]	[TotalRow]	[SortOrder]
6		[Column]	Dept.VP	VP		TRUE	header	1
7		[ColumnGroup]		Q1	fa-calendar			
8		[Column]	GL2018.Q1	Actuals		TRUE	Sum	
9		[Column]	BGT2018.Q1	Budget		TRUE	Sum	
10		[CalculatedColumn]	Difference	Difference		TRUE	Sum	
11		[CalculatedColumn]	PDifference	Difference %		TRUE		

Example DataGridColumns data source with configured total row

The resulting data grid contains a total row as follows:



Expense Analysis

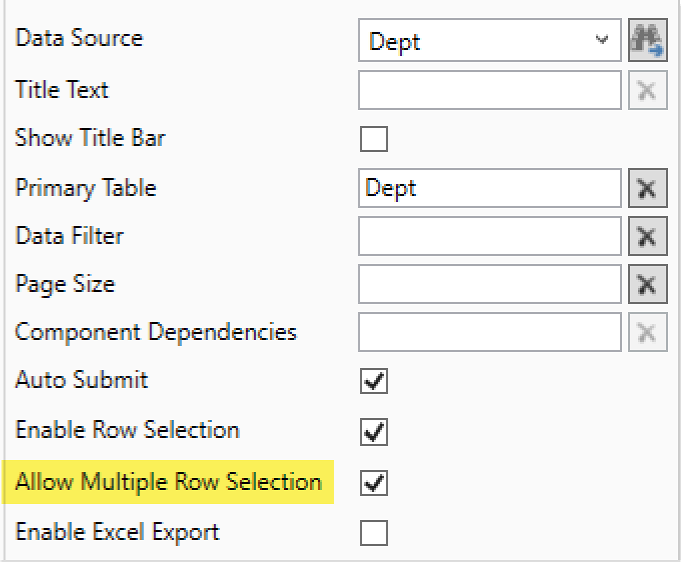
Q1 2018

	📅 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 Grant	$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%
Total	**$47,818,020**	**$42,318,618**	**$5,499,402**	

Example data grid with total row

Select multiple rows

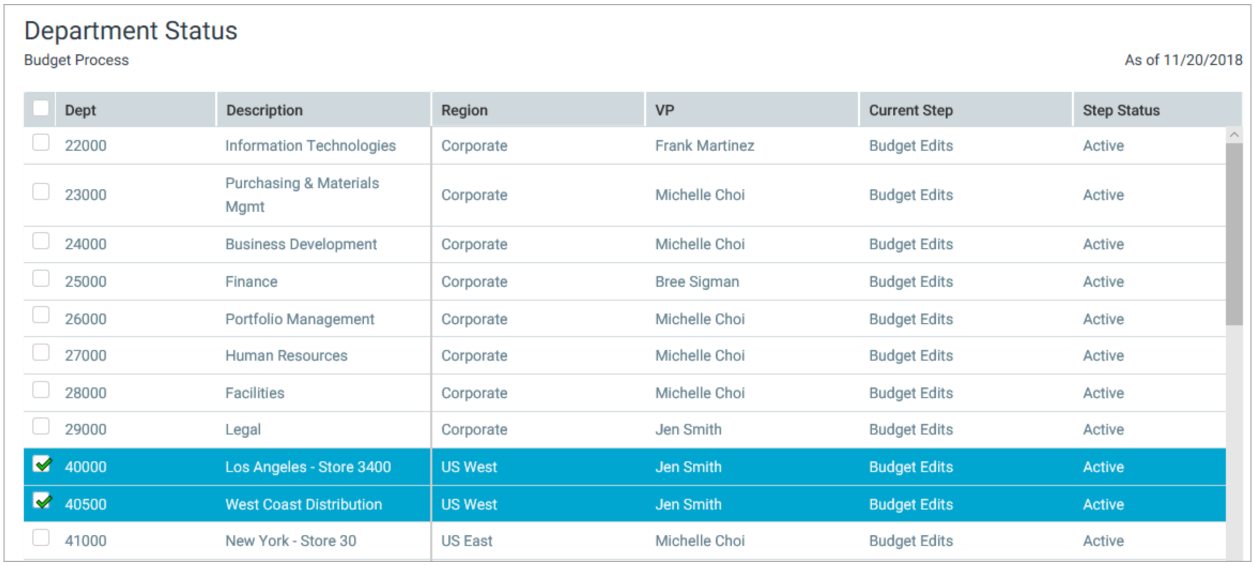
You can now select multiple rows in a data grid instead of just a single row. When **Enable Row Selection** is enabled in the component properties, a second option becomes available named **Allow Multiple Row Selection**.



Data Source	Dept
Title Text	
Show Title Bar	☐
Primary Table	Dept
Data Filter	
Page Size	
Component Dependencies	
Auto Submit	☑
Enable Row Selection	☑
Allow Multiple Row Selection	☑
Enable Excel Export	☐

New option to enable multiple row selection

If this option is enabled, then all rows in the grid display with a check box, so that users can select multiple rows.



Department Status

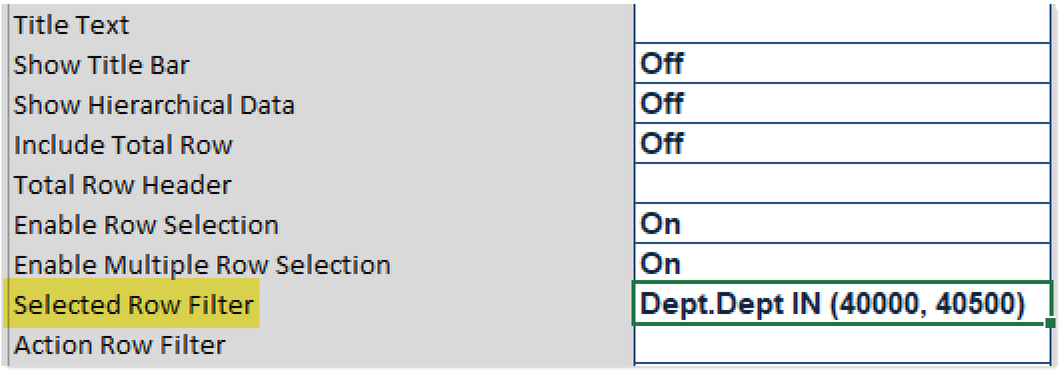
Budget Process

As of 11/20/2018

☐	Dept	Description	Region	VP	Current Step	Step Status
☐	22000	Information Technologies	Corporate	Frank Martinez	Budget Edits	Active
☐	23000	Purchasing & Materials Mgmt	Corporate	Michelle Choi	Budget Edits	Active
☐	24000	Business Development	Corporate	Michelle Choi	Budget Edits	Active
☐	25000	Finance	Corporate	Bree Sigman	Budget Edits	Active
☐	26000	Portfolio Management	Corporate	Michelle Choi	Budget Edits	Active
☐	27000	Human Resources	Corporate	Michelle Choi	Budget Edits	Active
☐	28000	Facilities	Corporate	Michelle Choi	Budget Edits	Active
☐	29000	Legal	Corporate	Jen Smith	Budget Edits	Active
☑	40000	Los Angeles - Store 3400	US West	Jen Smith	Budget Edits	Active
☑	40500	West Coast Distribution	US West	Jen Smith	Budget Edits	Active
☐	41000	New York - Store 30	US East	Michelle Choi	Budget Edits	Active

Example grid with multiple selection enabled

When using multiple-row selection, selected row values are not written back to the `[SelectedRowValue]` column of the data source. However, a filter statement representing the selected rows is written back to the **Selected Row Filter** field on the Form Control Sheet. You can use this filter to impact the form in some way, such as to filter another data query.



Title Text	
Show Title Bar	Off
Show Hierarchical Data	Off
Include Total Row	Off
Total Row Header	
Enable Row Selection	On
Enable Multiple Row Selection	On
Selected Row Filter	Dept.Dept IN (40000, 40500)
Action Row Filter	

Example filter statement for selected rows

Include process columns

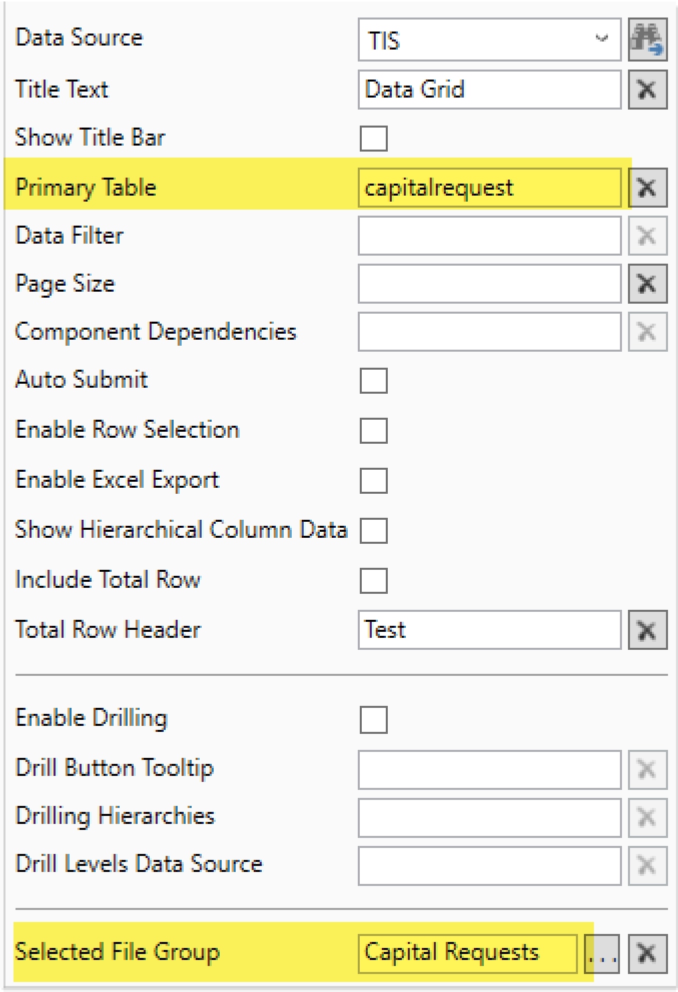
You can now include process columns in a Data Grid component, in order to report on information for a plan file process. For example, you can show the current step of a plan file, or the time spent in a step. The basics of this feature work the same way as the existing feature to include process columns in an Axiom query.

In order to use process columns, the component properties must be set up as follows:

- A new field is available in the component properties to specify a **Selected File Group**. The component automatically uses the process designated as the **Plan File Process** for the selected file group (in the file group properties).

- Set the **Primary Table** of the component to one of the following:
  - The plan code table of the selected file group.
  - A data table with a lookup to the plan code table.

  For example, if the plan file process is for the Budget 2018 file group, and the plan code table for that file group is the Dept table, then the primary table can be Dept or a data table that looks up to Dept (such as BGT2018). If you use a data table, then the data returned will only be for plan codes that are referenced within that table.



Data Source	TIS
Title Text	Data Grid
Show Title Bar	□
Primary Table	capitalrequest
Data Filter	
Page Size	
Component Dependencies	
Auto Submit	□
Enable Row Selection	□
Enable Excel Export	□
Show Hierarchical Column Data	□
Include Total Row	□
Total Row Header	Test
Enable Drilling	□
Drill Button Tooltip	
Drilling Hierarchies	
Drill Levels Data Source	
Selected File Group	Capital Requests

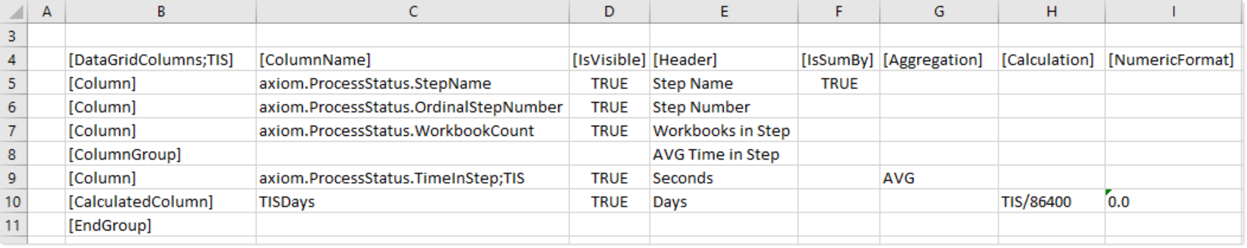
Example component properties configured to use process columns

Once these requirements are met, you can use process columns from the system tables Axiom.ProcessInstance and Axiom.ProcessStatus in the data grid as follows:

- Any of the process columns can be used as `[Column]` rows in the data source, so that the columns display in the grid. The columns can also be designated as sum by columns for the grid, and used to sort the grid.
- Process columns can be used in the data filter for the grid. You can also enable `[IsFilterable]` for the columns.
- Process columns that return numeric data can be used in `[CalculatedColumn]` rows. Practically speaking, this is only useful with the Axiom.ProcessStatus.TimeinStep column—for example, to use a calculation to convert the seconds to hours or days.

All other grid features, including grouping, drilling, and row selection, can all be used when the grid contains process columns.

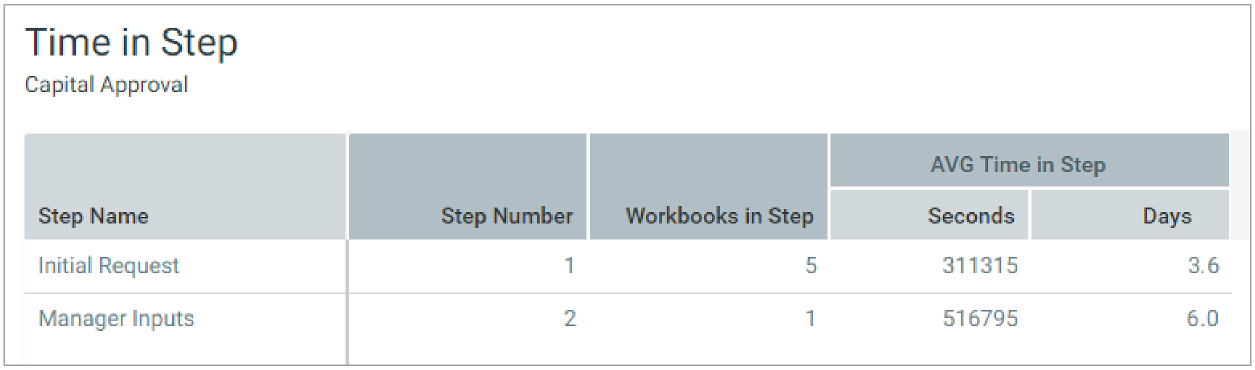
The following screenshot shows an example DataGridColumns data source that is set up to show time-in-step information for a process. The grid uses several process columns and also includes a calculated column to convert the time-in-step from seconds to days.



	A	B	C	D	E	F	G	H	I
3									
4		[DataGridColumns;TIS]	[ColumnName]	[IsVisible]	[Header]	[IsSumBy]	[Aggregation]	[Calculation]	[NumericFormat]
5		[Column]	axiom.ProcessStatus.StepName	TRUE	Step Name	TRUE			
6		[Column]	axiom.ProcessStatus.OrdinalStepNumber	TRUE	Step Number				
7		[Column]	axiom.ProcessStatus.WorkbookCount	TRUE	Workbooks in Step				
8		[ColumnGroup]			AVG Time in Step				
9		[Column]	axiom.ProcessStatus.TimeInStep;TIS	TRUE	Seconds		AVG		
10		[CalculatedColumn]	TISDays	TRUE	Days			TIS/86400	0.0
11		[EndGroup]							

Example data source with process columns

When the form is rendered, this example grid looks as follows:



Time in Step

Capital Approval

			AVG Time in Step	
Step Name	Step Number	Workbooks in Step	Seconds	Days
Initial Request	1	5	311315	3.6
Manager Inputs	2	1	516795	6.0

Example data grid with process columns

For more information on the columns that can be used, see the following topic in Axiom Software Help: *Including process columns in a report* (AX1747).

Optimized filter behavior for Data Grids using the HierarchicalGrid data source

When you expand a grouping in a hierarchical data grid, a filter is applied to the child grid so that it only shows data relating to the sum by level of the row that was expanded. In previous releases, if the child grid had multiple lookup paths to the parent sum by level, an ambiguous column error would occur when trying to filter the child grid.

Going forward, if the child grid has multiple lookup paths to the parent sum by level, Axiom Software will attempt to select a "preferred" lookup path using the following priority order:

1. Single-level lookup path (if there is only one)
2. Key column lookup path (if there is only one)
3. Shortest lookup path length (if there is only one at the shortest length)
4. If no single lookup path can be found that matches these rules, then the ambiguous column error occurs.

Additional Data Grid enhancements

- When defining calculations for `[CalculatedColumn]` rows, you can now reference the names of previous calculated columns in the calculation expression. For example, if a calculated column row is named Difference, you can create a second calculated column row named Difference_Percent that uses a calculation of `Difference/BGT2018.Q4`. The result of the Difference column is used in the calculation.

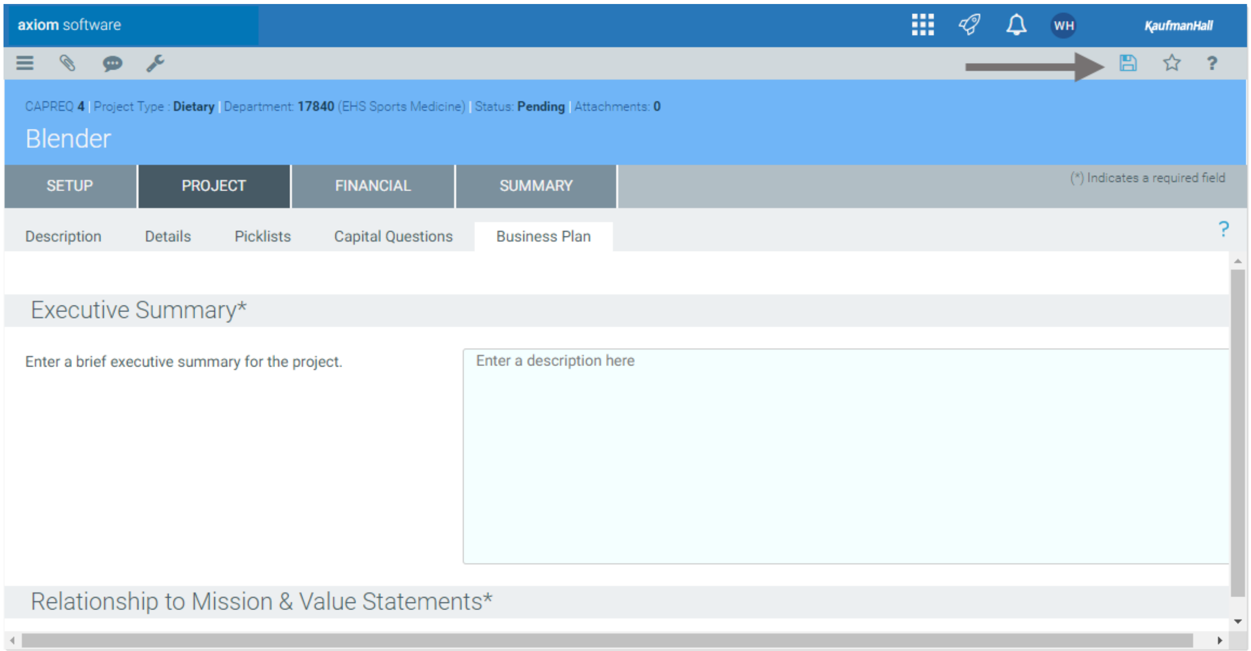
- When exporting grid contents to Excel, the number format defined for the column is now preserved. If the column has no number format, the column uses default formatting based on the column's data type (and numeric type, if applicable).

- The style **titledpanel-body** is now available for the Data Grid component (and the new Fixed Report component), so that those components can be easily used with the Titled Panel component. You can add the Titled Panel component to the canvas, delete the Formatted Grid component, then add a Data Grid / Fixed Report component and configure it to use the style. Remember that you must clear the existing position and size settings for the component in order to use the settings defined by the style.

- Data Grid components now support the ability to print to PDF. The grid data is automatically extended to include all rows in the PDF. For additional design considerations, see the following topic in Axiom Software Help: *Data Grid component for Axiom forms* (AX1624).

Save data using standard icon in Task Bar

Axiom forms now have a standard save icon that can be used to save data from the form. This icon displays in the right side of the gray Task Bar across the top of the form.



CAPREQ 4 | Project Type: Dietary | Department: 17840 (EHS Sports Medicine) | Status: Pending | Attachments: 0

Blender

(*) Indicates a required field

Executive Summary*

Enter a brief executive summary for the project.

Enter a description here

Relationship to Mission & Value Statements*

Example form with save icon in Task Bar

This icon provides a standardized, consistent way for users to save data from a form. This consistency makes it easier for users to understand when saves are available and how to trigger the save. Although you can still use individual components to trigger a save-to-database, we recommend converting older forms to use the new save icon instead for consistency.

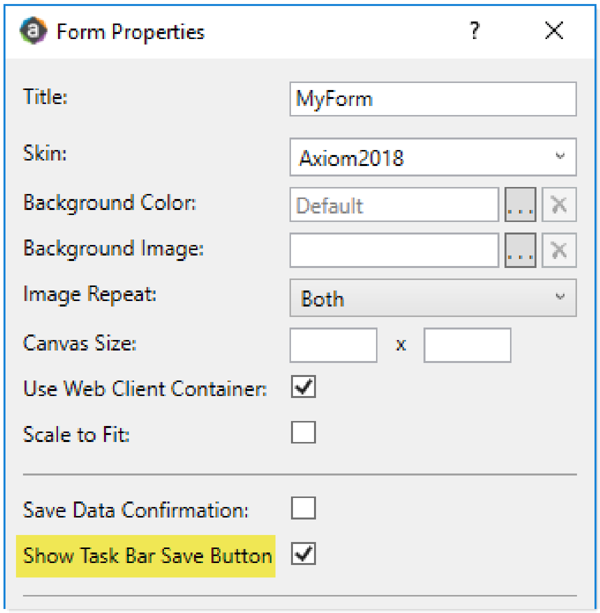
The new save icon is enabled in all newly created forms by default. This means:

- The save icon displays in the task bar when a user views the form.
- If no save-to-database processes are enabled in the form, or if the current user does not have **Allow Save Data** rights for the form, then the save icon is inactive. The icon displays as faded and gray.
- If a save-to-database process is enabled in the form, and the current user has **Allow Save Data** rights, then the save icon is active and can be used to save data to the database. The icon displays in a blue color.

In some cases, you may want to disable the save icon for a form so that it does not show in the task bar. For example:

- If the form is for reporting only and never saves data, the save icon is irrelevant and can be hidden.
- If you have configured a component to **Save on Submit** as an alternative to using the save icon, you may want to hide the save icon to avoid having two different ways to trigger a save for the form.

You can enable or disable the save icon using the **Show Task Bar Save Button** option in the Form Properties dialog. From the top of the **Form Assistant** task pane or the **Form Designer** dialog, click **Edit form properties**.



Title: MyForm

Skin: Axiom2018

Background Color: Default

Background Image:

Image Repeat: Both

Canvas Size: x

Use Web Client Container: ☑

Scale to Fit: □

Save Data Confirmation: □

Show Task Bar Save Button ☑

New option to enable save icon in form Properties

▶ Converting older forms to use the save icon

The save icon is disabled by default for any Axiom forms that were created prior to version 2018.4, to preserve backward-compatibility for those forms. We recommend converting these older forms to use the save icon unless there is a functional reason not to.

For example, if it is necessary to perform other commands at the same time as the save-to-database, then that form should continue to use a Save on Submit button because the save icon cannot currently perform commands. In most other cases, the form should be converted to using the save icon, so that users have a consistent way to execute save-to-database processes from Axiom forms.

To determine whether a form can be converted:

- Find the **Save on Submit** component(s) for the form. In almost all cases this will be a Button component, but it is also possible to configure certain stand-alone interactive components to enable Save on Submit (such as CheckBox, ComboBox, and so on). Button tags in Formatted Grid components can also be configured to Save on Submit.

- If the Save on Submit component is a Button component (or a Button tag), review its configuration. The following configurations *cannot* be handled using the save icon:

  - The button performs one or more commands that need to be executed along with the save-to-database.

  - The button uses the **Dialog Panel Action** behavior, and you need the save-to-database to be executed at the same time as the dialog action (such as the **OK** action).

  - The button has a defined confirmation message that you need to display before executing the save-to-database.

  For all three of these configurations, you should continue to use the button as is, and do not convert the form to enable the save icon on the task bar.

- If the Save on Submit component is not a button, then review the form to determine if it is necessary to trigger the save when the interactive component is used, or if it would be acceptable to instead trigger it using the save icon. If it is necessary to trigger the save using the interactive component, then you should not convert the form.

To convert an older form to use the save icon:

1. Remove the old method of triggering the save-to-database using one of the following approaches:

   - If the only purpose of the old Save on Submit component was to trigger the save-to-database, delete the component from the form.

     OR

   - Clear the **Save on Submit** check box for the component so that it no longer triggers a save-to-database. You would only do this if you want to retain the component in the form, in order to use it for some other purpose.

You can also leave the old Save on Submit component as is, *and* enable the save icon in the task bar, but it may be confusing for users to have two different ways to trigger the save-to-database.

2. From the top of the **Form Assistant** task pane or the **Form Designer** dialog, click **Edit form properties**.

3. In the **Form Properties** dialog, select the check box for **Show Task Bar Save Button**.

   For older forms, you must use the Form Properties dialog to configure this setting, because the setting is not yet present in the Form Control Sheet. Editing the setting in the Form Properties dialog will add it to the Form Control Sheet.

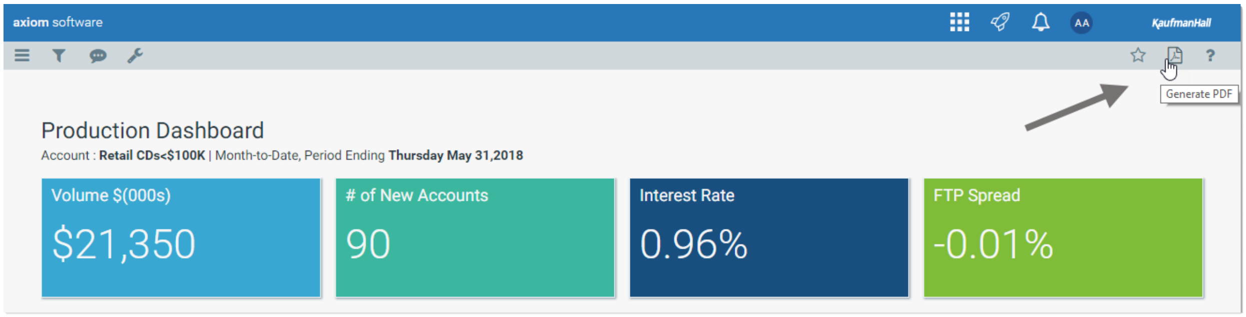
4. Make any other changes to the form that may be necessary to maintain functionality. Keep in mind the following design considerations:

   - If you were dynamically controlling the ability to save data by using a formula in the **Enabled** property of the Save on Submit component, this approach is not available for the save icon in the task bar. Instead, you can use a formula to dynamically control whether the save-to-database is enabled or not on the default Control Sheet—for example, by dynamically toggling **Save Type 1 Enabled** from **Off** to **On**.

   - If anything in the form was dependent on the Save on Submit component being the **Triggering Component**, this should be changed so that formulas now check for **Axiom.FormSave** as the triggering component. This reserved term indicates that the save icon in the task bar is the triggering component.

Generate PDF using standard icon in Task Bar

Axiom forms now have a standard PDF icon that can be used to generate a PDF of the form. This icon displays in the right side of the gray Task Bar across the top of the form.

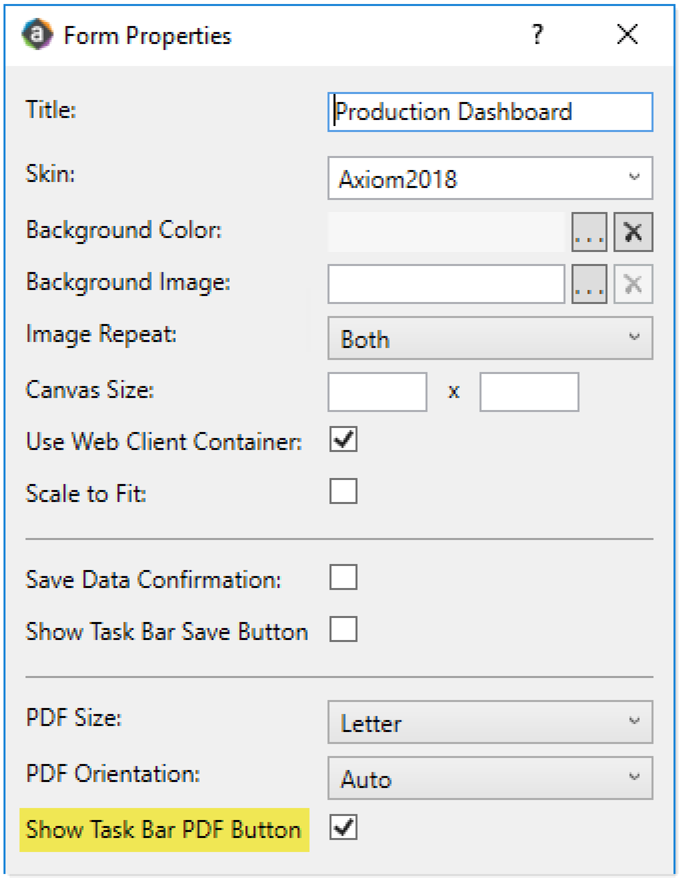


Example form with PDF icon in Task Bar

The icon provides a standardized, consistent way for users to generate a PDF of the form. This consistency makes it easier for users to understand when a form has been optimized for PDF, and how to create the PDF. Although you can still use individual Button components to generate a PDF, we recommend converting older forms to use the new PDF icon instead for consistency.

NOTE: The ability to generate a PDF is still available on the Tools menu for all forms, but this approach is not obvious to end users. If you intend for users to generate PDFs of the form, you should enable the icon for ease-of-use.

By default, the PDF icon is disabled in Axiom forms. You must enable it if you want the icon to display for a particular form. The idea is that the icon should only be made available for Axiom forms where the PDF output has been configured and tested to result in nicely formatted and usable output. To enable the icon, use the **Form Properties** dialog and select **Show Task Bar PDF Button**.



Form Properties

Title:	Production Dashboard
Skin:	Axiom2018
Background Color:	
Background Image:	
Image Repeat:	Both
Canvas Size:	x
Use Web Client Container:	☑
Scale to Fit:	□
Save Data Confirmation:	□
Show Task Bar Save Button	□
PDF Size:	Letter
PDF Orientation:	Auto
Show Task Bar PDF Button	☑

New option to enable PDF icon in Form Properties

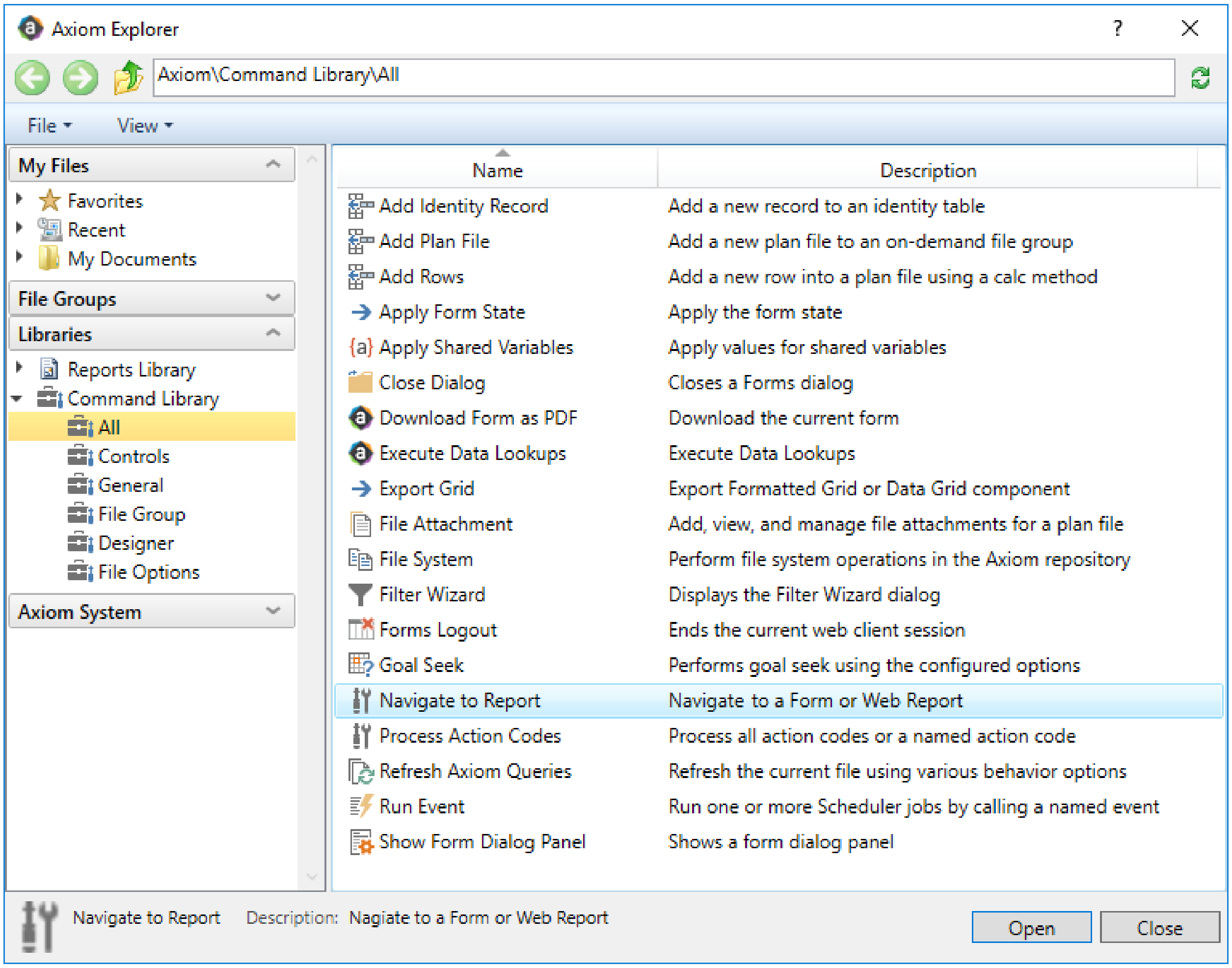
For older forms, you must use the Form Properties dialog to configure this setting, because the setting is not yet present in the Form Control Sheet. Editing the setting in the Form Properties dialog will add it to the Form Control Sheet.

If you have an older form with a Button component that uses the **Download Form as PDF** command, you can simply remove this button and instead enable the icon. Both the button and the icon use the same approach to generate the PDF, so the behavior will be the same. Since it was not possible to combine the Download Form as PDF command with any other commands or form update behavior, there should be no other design considerations.

New command to open files from a form

A new command is available to open reports and other files from within an Axiom form. Although it is currently possible to do this using generated URLs and document shortcuts, the command offers additional flexibility and can be used with features such as icon actions in Data Grid components.

The new command, Navigate to Report, is available in the Command Library when configuring commands in an Axiom form:



Name	Description
Add Identity Record	Add a new record to an identity table
Add Plan File	Add a new plan file to an on-demand file group
Add Rows	Add a new row into a plan file using a calc method
Apply Form State	Apply the form state
Apply Shared Variables	Apply values for shared variables
Close Dialog	Closes a Forms dialog
Download Form as PDF	Download the current form
Execute Data Lookups	Execute Data Lookups
Export Grid	Export Formatted Grid or Data Grid component
File Attachment	Add, view, and manage file attachments for a plan file
File System	Perform file system operations in the Axiom repository
Filter Wizard	Displays the Filter Wizard dialog
Forms Logout	Ends the current web client session
Goal Seek	Performs goal seek using the configured options
Navigate to Report	Navigate to a Form or Web Report
Process Action Codes	Process all action codes or a named action code
Refresh Axiom Queries	Refresh the current file using various behavior options
Run Event	Run one or more Scheduler jobs by calling a named event
Show Form Dialog Panel	Shows a form dialog panel

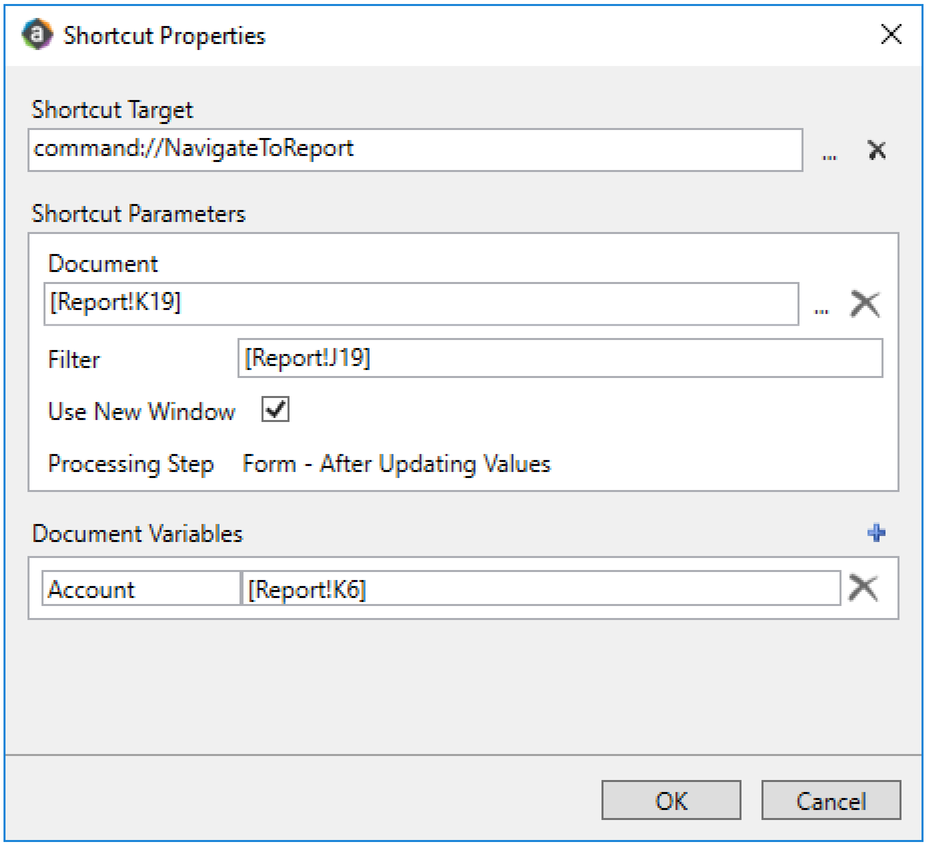
Navigate to Report Description: Nagiate to a Form or Web Report

New command in Command Library

When configuring the command, you can specify the following:

- The document to open—any Axiom form or web report
- A filter to apply to the document (Axiom forms only)
- Whether to open the document in a new window
- One or more document variables to pass to the document (Axiom forms only)

The main advantage to using the command is that most of the properties can optionally use bracketed cell references, so that the value is read from the specified cell instead of "hard-coded" into the command. In the following example, the document path, the filter statement, and the value of the Account variable are all read from cell references in the file. Using cell references allows these properties to change dynamically.



Shortcut Target: command://NavigateToReport

Shortcut Parameters

Document: [Report!K19]

Filter: [Report!J19]

Use New Window ☑

Processing Step: Form - After Updating Values

Document Variables

Account: [Report!K6]

Shortcut properties for Navigate to Report command

This new command means that you can now dynamically open different documents and apply different filters and/or document variables when using icon actions with Data Grid and Fixed Report components. Previously this was not possible, because if you constructed a URL or document shortcut using a formula, the formula would not update when the icon in the grid was clicked. However, the bracketed cell references in the command will update and pass the current cell values to the command.

Additional Axiom form enhancements

▶ Select a value from Scatter Charts

Charts that use the Scatter data source now support selecting a value in the chart. This includes the Scatter, ScatterLine, and Bubble Charts.

To enable selecting a value, you must add an [ID] column to the Scatter data source, and then populate that column with unique ids for each value. When the user selects a value from the chart, the corresponding id is written to the **Selected ID** property for the chart on the Form Control Sheet. You can then use that selected value to impact the form in some way.

The component properties for the affected chart types now also have an **Auto Submit** property, which should be enabled if you want the user's value to be submitted on selection.

▶ Clear selected value for charts

Users can now easily clear the selected value for a chart, by clicking on the selected value again. This applies to all chart components that support selecting a value, including:

- Chart components that use the XYChart data source (such as Column and Bar Charts)
- Chart components that use the Scatter data source (such as Scatter and Bubble Charts)
- Pie Chart
- Hierarchy Chart
- Map View
- KPI Panel

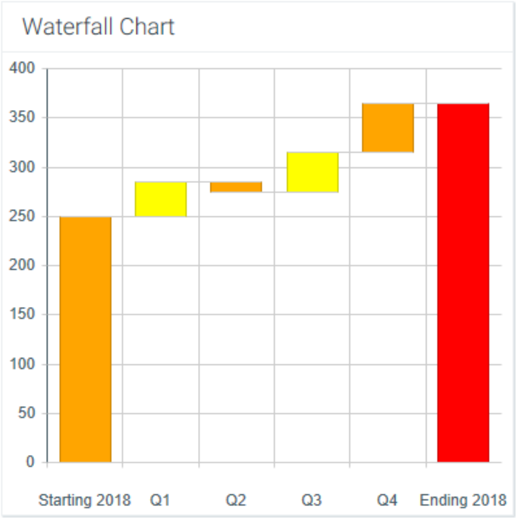
For example, if a user clicks a slice in a pie chart to select it, they can now click the same slice again to clear that selection. The chart now has no selected value.

IMPORTANT: In previous releases, users could not have cleared the selected value for the chart (unless the form was designed with a manual solution to do so, such as using action codes). Therefore, if you designed a form with a default selected value for a chart, then the chart would always have a selected value and would not need to handle the case where the chart has no selected value. Going forward, users can now clear the selected value so the form must be designed to handle this use case.

▶ Waterfall Chart enhancements

You can now use multiple colors in a Waterfall Chart component. When the chart series is a waterfall series (regular or horizontal), the [Color] tag can accept a comma-delimited list of colors—either color names or hexadecimal codes. The bars in the waterfall chart will cycle through the specified colors—including totals and running totals, unless different colors have been specified for those bars.

In the following example, the regular waterfall bars cycle through orange and yellow, and then the total bar uses red.



Example waterfall chart using multiple colors

Additionally, the tooltips for total and running total columns now use the number format as defined by the **Primary Y-Axis Format** setting in the component properties. You should set this to the same format used by the column values in the chart.

Report Builder

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

Use dynamic years in a web report

You can now set up a web report so that the data in the report dynamically changes based on a user's selected year. To do this:

- Set up the report using year-based tables, such as GL2018, BGT2019, and so on.
- Create a Year refresh variable and configure it as needed, so that users can select years as appropriate for the report setup and available data.

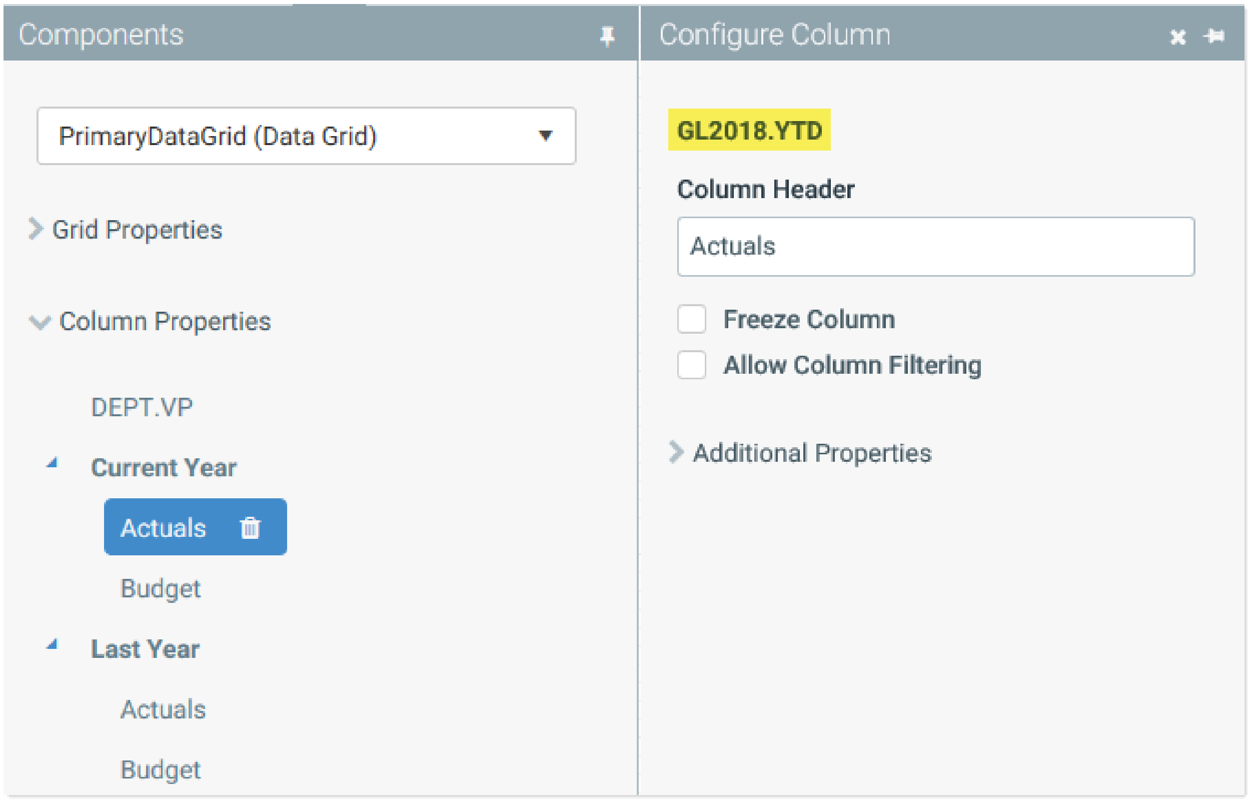
The Year refresh variable is a new variable type that is available in web reports only. Users can select a year in order to adjust the "base year" of the report to the selected year. The report is then refreshed with data relative to that year.

For example, imagine that a report shows 2018 actuals, with a comparison to the prior year (2017). If the user wants to see how last year compared to the year before that, they can use the year variable to change the base year to 2017. Now the report shows 2017 actuals with a comparison to the prior year (2016).

▶ Setting up the report for dynamic years

The basic report setup is as normal, except that data components must use year-based tables if you want the data to change based on the selected year. A year-based table is simply a table with the year in the name, such as GL2018.

For example, you could set up a Data Grid component as shown in the following screenshot. This report shows current year actuals and budget (GL2018.YTD and BGT2018.YTD) as compared to last year's actuals and budget (GL2017.YTD and BGT2017.YTD). The report could also have calculated columns to show the difference and percent difference.



GL2018.YTD

Column Header: Actuals

□ Freeze Column

□ Allow Column Filtering

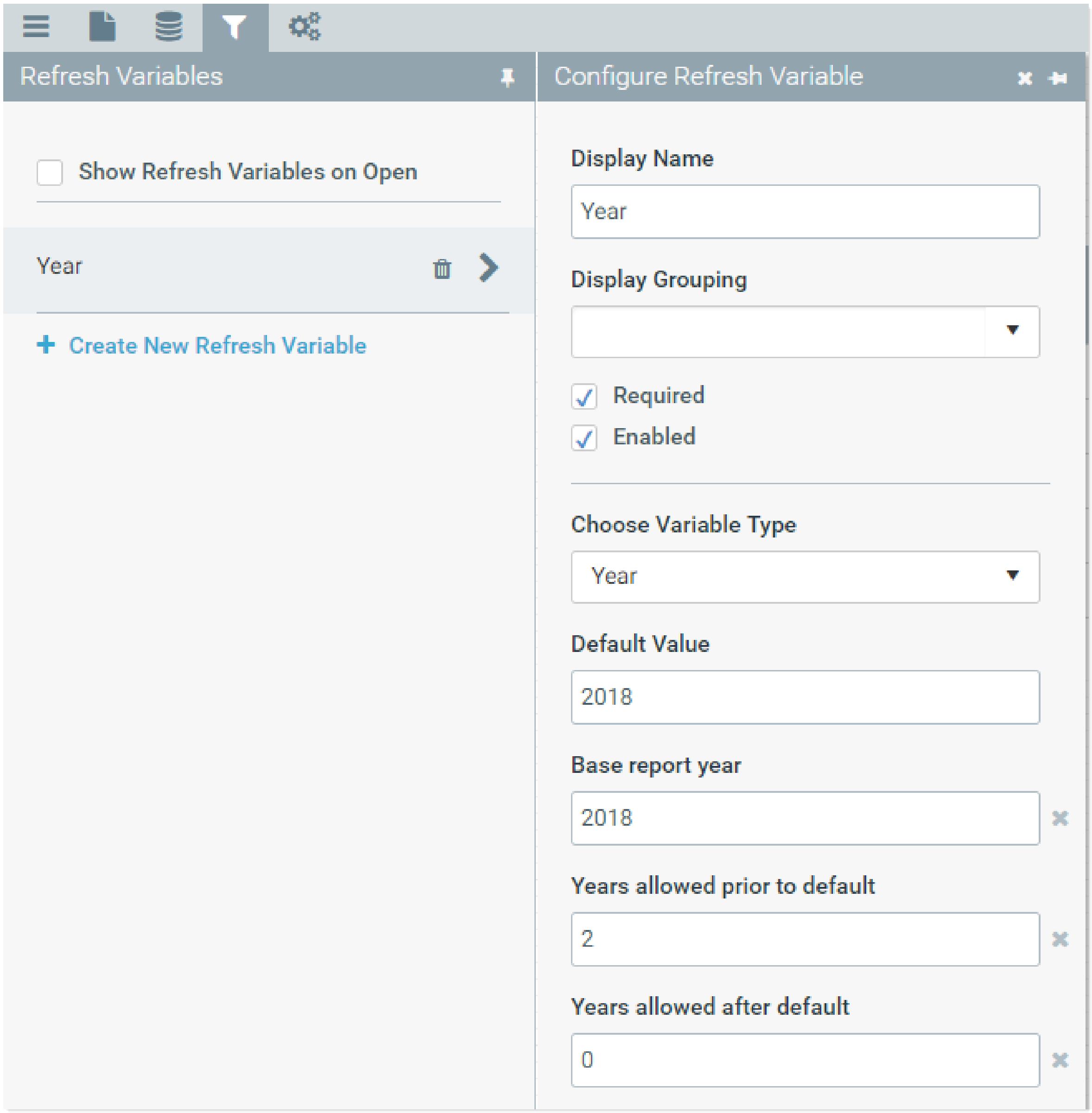
In this example, 2018 is the base year for the report, because the report is built for 2018 and uses columns from 2018 tables (and adjacent year tables). When the Year refresh variable is used to change the year, all table and column references are adjusted based on the difference between the base year and the selected year.

For example, if the user selects 2017 as the new value for the refresh variable, that selected year is one year less than the original base year of 2018. All references to year-based tables in the report are adjusted to be one year less. The reference to GL2018.YTD is adjusted to GL2017.YTD, and the reference to GL2017.YTD is adjusted to GL2016.YTD.

References to year-based tables are adjusted this way throughout the report. This includes the primary table for the report, table columns used in data grids, calculated columns used in data grids, and so on.

Creating the Year refresh variable

In order to change the year dynamically, use the new Year refresh variable type.



□ Show Refresh Variables on Open

Year

Display Name: Year

Display Grouping:

☑ Required

☑ Enabled

Choose Variable Type: Year

Default Value: 2018

Base report year: 2018

Years allowed prior to default: 2

Years allowed after default: 0

Example Year variable type

When creating this variable, you define the following:

- **Base Report Year**: The year the report is built for, based on the tables and columns included in the report (as discussed in the previous section). This setting is required in order to set the baseline to adjust the year-based references in the report.

- **Default Value**: The year that you want the report to use by default when it is opened. When the report is first created, the default value and the base report year will likely be the same. In our example, we want the report to use 2018 by default.

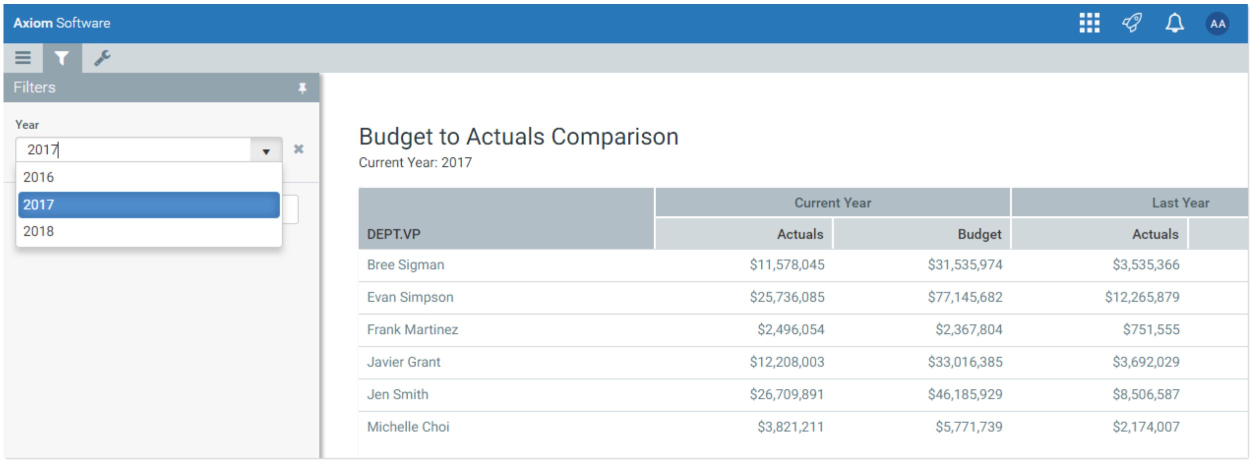
  However, over time, the default value can change while the base report year remains the same. When 2019 starts, we can change the default value of the report to 2019 so that the report is refreshed based on 2019 data (adjusting references 1 year up from the base year of 2018). However, the base year of 2018 must remain the same, because the tables and columns referenced in the report have not changed.

  If the report uses a file group context, you can leave the default value blank to use the file group year by default. Otherwise, the default value is required.

- **Years allowed prior to / after default**: These two settings determine the years available for the user to select. If the prior-allowed years is set to 2 and the default year is 2018, then users can select 2018, 2017, and 2016. If the after-allowed years is set to 1, then users can also select 2019. These settings are optional and are interpreted as 0 if left blank.

  You should set these fields as appropriate depending on the purpose of the report and the available data. If you allow users to select a year which results in invalid tables, an error results. For example, if the report displays actuals data and you allow users to select years in the future, your system may not have actuals tables for those future years yet (and even if it does, those tables likely do not have any data).

When the user views the web report, they can now use the Year refresh variable to change the base year and refresh the file with applicable data.



Budget to Actuals Comparison

Current Year: 2017

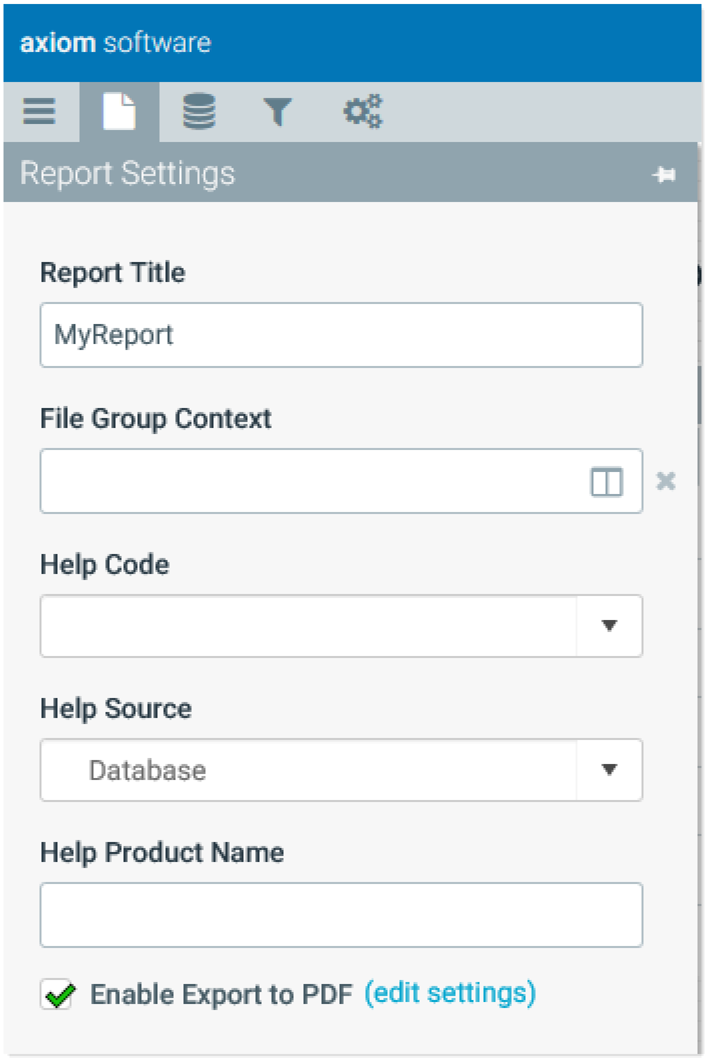
	Current Year		Last Year
DEPT.VP	Actuals	Budget	Actuals
Bree Sigman	$11,578,045	$31,535,974	$3,535,366
Evan Simpson	$25,736,085	$77,145,682	$12,265,879
Frank Martinez	$2,496,054	$2,367,804	$751,555
Javier Grant	$12,208,003	$33,016,385	$3,692,029
Jen Smith	$26,709,891	$46,185,929	$8,506,587
Michelle Choi	$3,821,211	$5,771,739	$2,174,007

Example Year variable in Filter panel

Generate a PDF copy of a web report

Web reports now support the ability to generate a PDF of the report, for printing purposes. This option can be enabled or disabled on a per report basis.

A new option is available on the **Report Settings** tab of the Report Builder, named **Enable Export to PDF**. Select this check box if you want users to be able to generate a PDF of the report.



Report Settings

Report Title

MyReport

File Group Context

Help Code

Help Source

Database

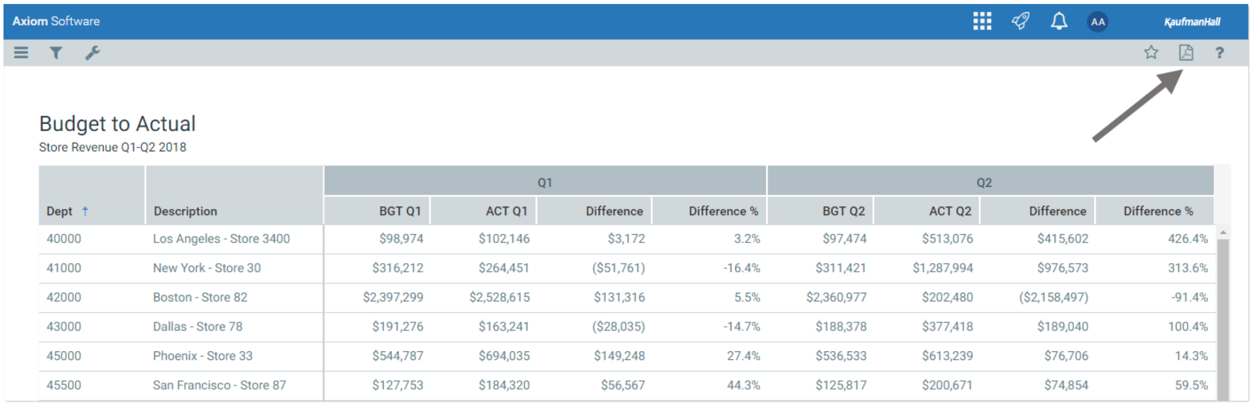
Help Product Name

☑ Enable Export to PDF (edit settings)

New option to enable PDF generation

The default PDF page size is Letter, and the default orientation is Portrait. If you want to change these options, click the **(edit settings)** link to open the **PDF Settings** dialog.

If PDF generation is enabled, then a PDF icon displays in the right side of the gray Task Bar across the top of the page. Users can click this button to generate a PDF of the web report. (This is the same standard PDF icon that is newly available for Axiom forms.)



Budget to Actual

Store Revenue Q1-Q2 2018

		Q1				Q2			
Dept ↑	Description	BGT Q1	ACT Q1	Difference	Difference %	BGT Q2	ACT Q2	Difference	Difference %
40000	Los Angeles - Store 3400	$98,974	$102,146	$3,172	3.2%	$97,474	$513,076	$415,602	426.4%
41000	New York - Store 30	$316,212	$264,451	($51,761)	-16.4%	$311,421	$1,287,994	$976,573	313.6%
42000	Boston - Store 82	$2,397,299	$2,528,615	$131,316	5.5%	$2,360,977	$202,480	($2,158,497)	-91.4%
43000	Dallas - Store 78	$191,276	$163,241	($28,035)	-14.7%	$188,378	$377,418	$189,040	100.4%
45000	Phoenix - Store 33	$544,787	$694,035	$149,248	27.4%	$536,533	$613,239	$76,706	14.3%
45500	San Francisco - Store 87	$127,753	$184,320	$56,567	44.3%	$125,817	$200,671	$74,854	59.5%

Example web report with PDF icon

When you generate a PDF of a web report, your browser will prompt you to save a copy of the PDF. The PDF does not open in a new tab, as it may when generating a PDF of an Axiom form.

For information regarding how Data Grid and KPI Panel components will render in the PDF, see the following topic in Axiom Software Help: *Configuring a web report for printing to PDF* (AX1672).

Axiom files

This section details the new features and enhancements that impact general Axiom file design.

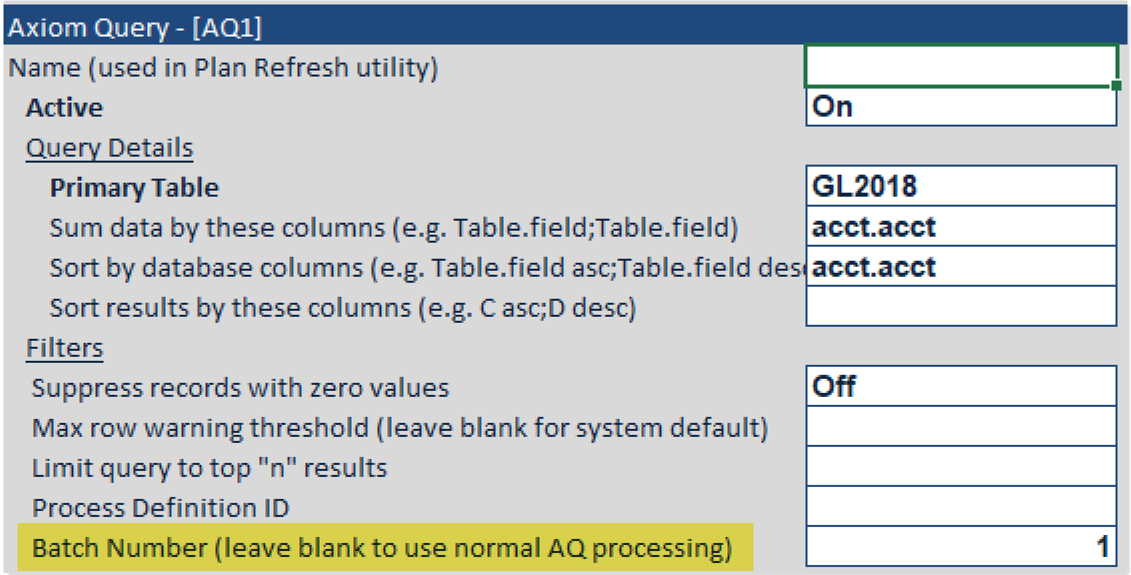
Batch processing for Axiom queries

By default, Axiom queries are processed in sequential order, one by one. Axiom Software walks through the sheets defined on the Control Sheet from left to right, and then processes the queries on each sheet in ascending order (AQ1 first, then AQ2, and so on).

You can now optionally define batches of Axiom queries within a sheet, so that the queries within the batch are processed in parallel instead of sequentially. You can assign two or more queries to the same batch if those queries are not dependent on each other and therefore can be run at the same time. Batching queries to be run in parallel can improve file performance, especially when the file has many queries, or has several process-intensive queries (such as queries that return a lot of data, or that use large calc methods).

In order to process a query within a batch, you assign the query a batch number. All queries on a sheet that have the same batch number will be processed in parallel instead of sequentially.

To assign a batch number to a query, a new field is available on the Control Sheet in the **Query Details** section, named **Batch Number**:



Axiom Query - [AQ1]	
Name (used in Plan Refresh utility)	
Active	On
Query Details	
Primary Table	GL2018
Sum data by these columns (e.g. Table.field;Table.field)	acct.acct
Sort by database columns (e.g. Table.field asc;Table.field des	acct.acct
Sort results by these columns (e.g. C asc;D desc)	
Filters	
Suppress records with zero values	Off
Max row warning threshold (leave blank for system default)	
Limit query to top "n" results	
Process Definition ID	
Batch Number (leave blank to use normal AQ processing)	1

New Batch Number setting for an Axiom query

When batch numbers are defined for Axiom queries, query processing works as follows:

- Each sheet is still processed in order. It is not possible to configure queries on separate sheets to be processed in the same batch.
- For each sheet, queries with an assigned batch number are processed first. Batches are processed in ascending order (batch 1 first, then batch 2, and so on). All queries with the same batch number are processed in parallel.
- After all batches have been processed for a sheet, the remaining Axiom queries on that sheet are processed, in ascending order.

For more information on the requirements and limitations, see the following topic in Axiom Software Help: *Batch processing for Axiom queries* (AX1743).

> **NOTE:** This feature was originally delivered in a late patch of 2018.3. However, it is included in these release notes because 2018.4 provides the official rollout of the feature to all customers.

Additional Axiom file enhancements

▶ Return name of remote data connection using a function

The GetSystemInfo function supports a new keyword to return the name of the default remote data connection. For example:

```
=GetSystemInfo("DefaultRemoteDataConnection")
```

This returns the name MyRDC if that is the name of your default remote data connection.

The default remote data connection is determined as follows:

- If your system has multiple defined connections, then the default is determined alphabetically among the connections that are not enabled for authentication.
- If all of the connections are enabled for authentication, then the default is simply determined alphabetically.

▶ Select month or year from Calendar refresh variable (Web Client only)

You can now prompt users to select a month or a year using a Calendar refresh variable, in the Web Client only. Use the `[DisplayFormat]` property of the Calendar refresh variable to specify the selection type of either `Date`, `Month`, or `Year`. By default, Date is used if this property is omitted or blank, for backward-compatibility.

If you specify Month or Year, then the refresh variable displays a month/year selector or a year selector instead of the calendar date selector. This works the same way as the Month and Year options for the Date Picker component for Axiom forms. The Month and Year options are only valid for use in the Web Client. If the refresh variable is rendered in the Desktop Client, it will always be a calendar date selector.

The Month and Year options only impact the selector that is presented to the user. The selected value is always written back as an Excel date serial number. For example, if the user selects the year 2018 from the year selector, the value is written back as 1/1/2018. You may need to use functions such as MONTH or YEAR to extract the desired information from the full date. Similarly, if you want to specify a minimum or maximum value for a Calendar refresh variable that uses Month or Year, you must enter the values as full dates and not as month or year strings.

▶ Enhancements to Axiom Queries that use process columns

Axiom queries can use columns from the Axiom.ProcessInstance and Axiom.ProcessStatus tables, to report on plan file processes. This existing feature has been enhanced as follows:

- The primary table for the Axiom query can now be a data table that looks up to the plan code table of the file group for the process. In previous releases, the primary table could only be the plan code table.
- Any of the process columns can now be used to filter or sort the query.

File Processing

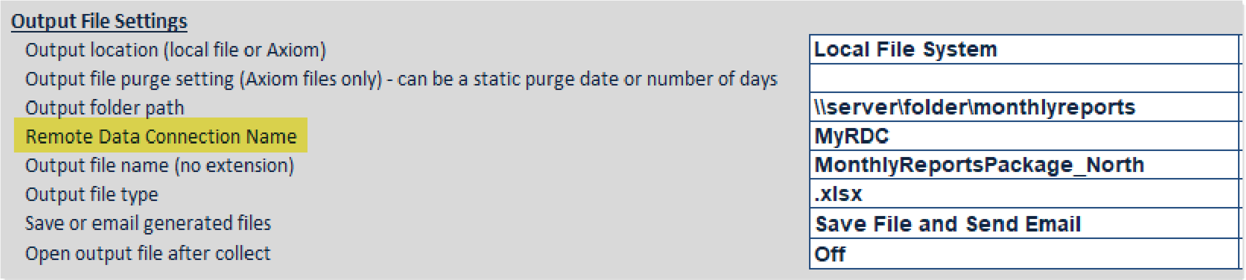
The following enhancements were made to the File Collect option of File Processing.

Use a Remote Data Connection with File Collect

Remote data connections can now be used with File Collect. This allows Cloud Service systems to do either or both of the following:

- Collect files from a local folder location, instead of needing to collect files from the Axiom repository.
- Save the output file to a local folder location, instead of needing to save the output file to the Axiom repository.

The File Collect Configuration sheet has a new field to specify the name of a remote data connection. You can enter the name of any remote data connection defined in your environment. Although this field is located in the Output File Settings section, the remote data connection specified here will also be used to generate and access the file list as applicable.



Output File Settings

Output location (local file or Axiom)	Local File System
Output file purge setting (Axiom files only) - can be a static purge date or number of days	
Output folder path	\\server\folder\monthlyreports
Remote Data Connection Name	MyRDC
Output file name (no extension)	MonthlyReportsPackage_North
Output file type	.xlsx
Save or email generated files	Save File and Send Email
Open output file after collect	Off

New field to specify a remote data connection for File Collect

By default, this setting uses the function `GetSystemInfo("DefaultRemoteDataConnection")`, so that the field is automatically populated with the name of your default remote data connection. If you do not have any remote data connections, the field is blank. If you want to use a different remote data connection, you can simply delete the function and enter the desired name.

The **Remote Data Connection Name** field is only present on new File Collect Configuration sheets. Currently, there is no way to upgrade an existing configuration sheet. If you need to use the Remote Data Connection Name field in an existing file, you can use the File Processing task pane to add a new File Collect Configuration sheet, and then manually copy the configuration from the existing sheet to the new sheet.

Excel not required to create PDFs via File Collect

The Excel Client is no longer required to create PDFs when using the File Collect option of file processing. File Collect can now generate a PDF output file when executing in the Windows Client directly, or when executing via Scheduler.

Scheduler

This section details the new features and enhancements made to Scheduler.

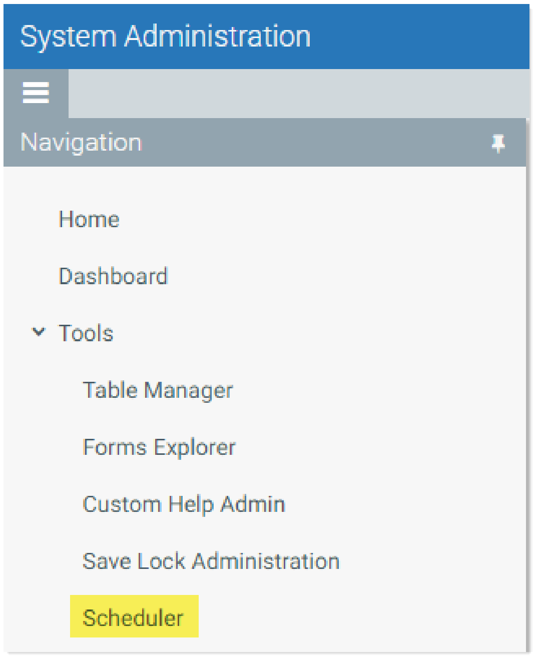
Scheduler in the Web Client

You can now monitor the Scheduler job queue and run jobs on demand using the Web Client. The Web Client Scheduler is intended to support our browser-based products, and also provides an alternative Scheduler interface for all customers.

Using the Web Client Scheduler, you can:

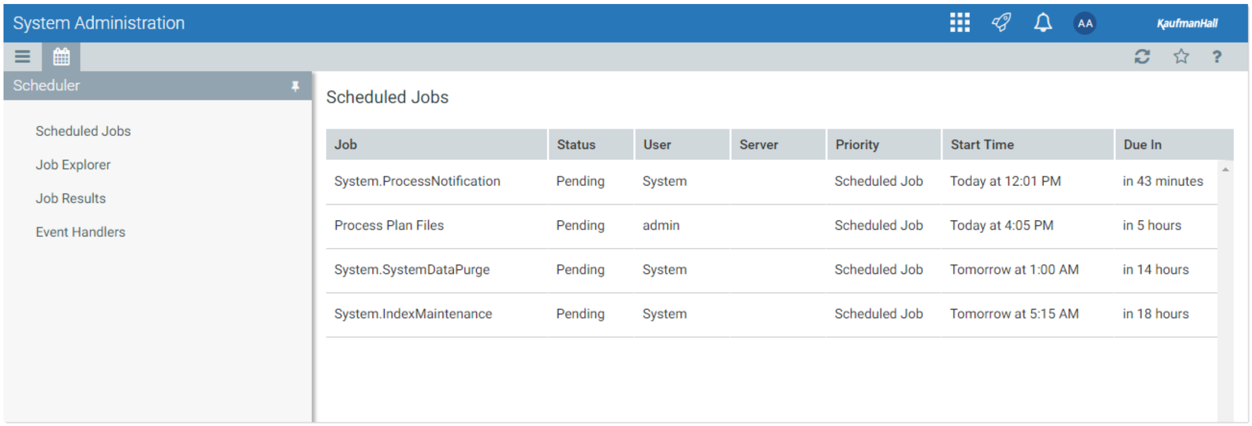
- View the current job schedule
- View job results from previous executions
- Run jobs manually as needed
- Remove jobs from the schedule
- View job properties

Scheduler is available in the **System Administration** area of the Web Client, under **Tools > Scheduler**. Only administrators and users with the Scheduled Job User permission can access Scheduler.



New Scheduler item on System Administration menu

Within the Scheduler area, you can use the Scheduler panel to change the current view—such as to view Scheduled Jobs or Job Results.



Scheduled Jobs

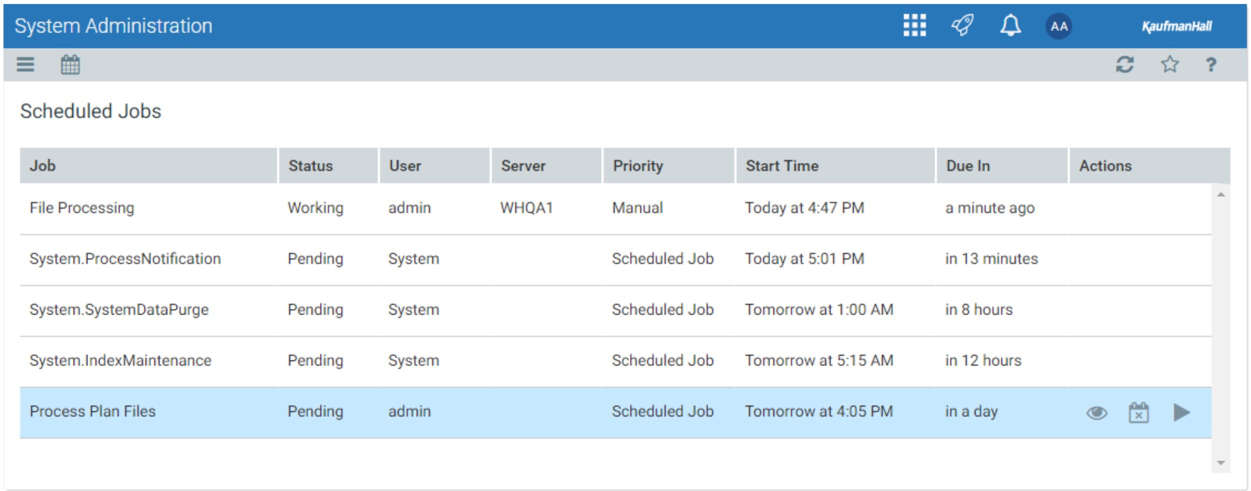
Job	Status	User	Server	Priority	Start Time	Due In
System.ProcessNotification	Pending	System		Scheduled Job	Today at 12:01 PM	in 43 minutes
Process Plan Files	Pending	admin		Scheduled Job	Today at 4:05 PM	in 5 hours
System.SystemDataPurge	Pending	System		Scheduled Job	Tomorrow at 1:00 AM	in 14 hours
System.IndexMaintenance	Pending	System		Scheduled Job	Tomorrow at 5:15 AM	in 18 hours

New Scheduler area, showing Scheduled Jobs view

Currently, all Scheduler jobs are read-only in the Web Client. If you need to create a new job or edit an existing job, you must still use Scheduler in the Desktop Client.

Managing the job schedule

In the **Scheduled Jobs** view, you can view the status of all jobs that are currently on the schedule. You can use the **Actions** column in the right side of the grid to view the job details, remove the job from the schedule, or run the job now.



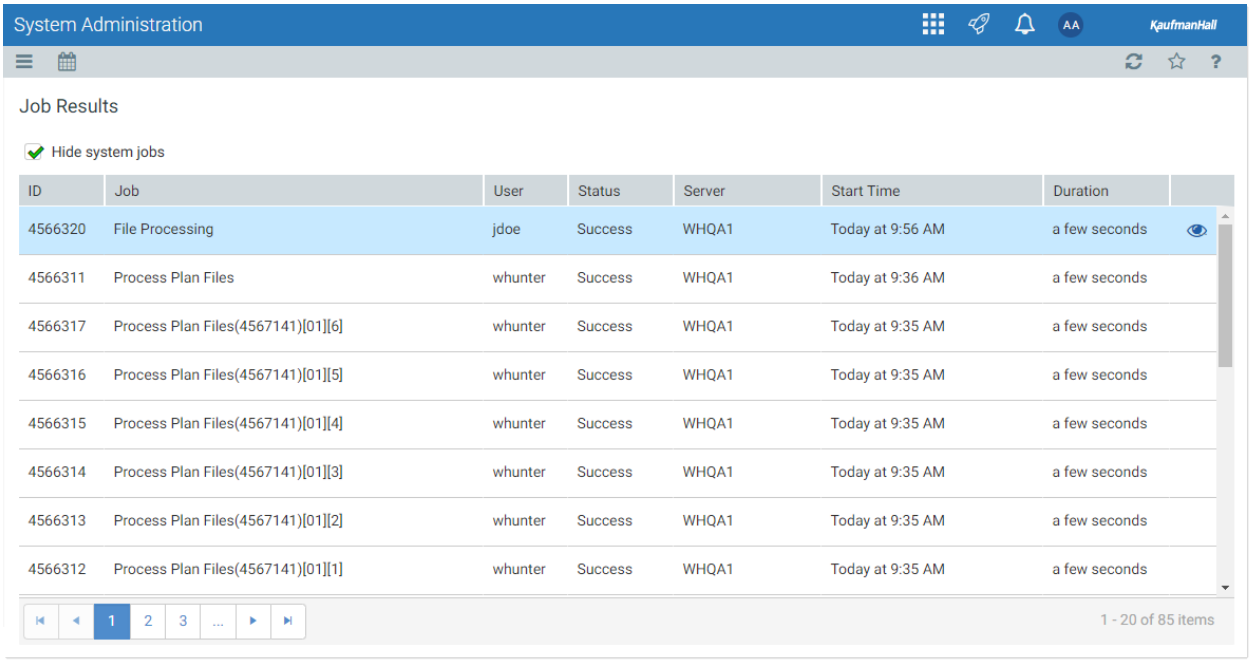
Scheduled Jobs

Job	Status	User	Server	Priority	Start Time	Due In	Actions
File Processing	Working	admin	WHQA1	Manual	Today at 4:47 PM	a minute ago	
System.ProcessNotification	Pending	System		Scheduled Job	Today at 5:01 PM	in 13 minutes	
System.SystemDataPurge	Pending	System		Scheduled Job	Tomorrow at 1:00 AM	in 8 hours	
System.IndexMaintenance	Pending	System		Scheduled Job	Tomorrow at 5:15 AM	in 12 hours	
Process Plan Files	Pending	admin		Scheduled Job	Tomorrow at 4:05 PM	in a day	👁 📅 ▶

Example Scheduled Jobs view

Viewing job results

In the **Job Results** view, you can view the results of jobs that have been executed. For each job, you can see when it was run, and whether it completed successfully or had errors.



Job Results

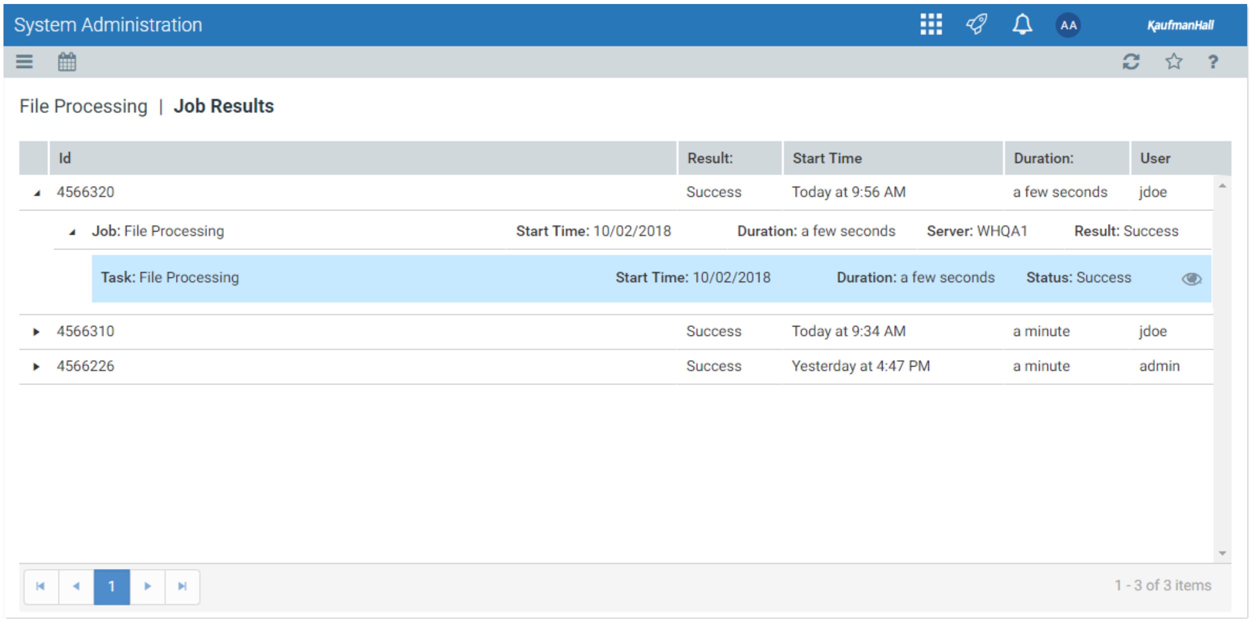
☑ Hide system jobs

ID	Job	User	Status	Server	Start Time	Duration
4566320	File Processing	jdoe	Success	WHQA1	Today at 9:56 AM	a few seconds
4566311	Process Plan Files	whunter	Success	WHQA1	Today at 9:36 AM	a few seconds
4566317	Process Plan Files(4567141)[01][6]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds
4566316	Process Plan Files(4567141)[01][5]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds
4566315	Process Plan Files(4567141)[01][4]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds
4566314	Process Plan Files(4567141)[01][3]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds
4566313	Process Plan Files(4567141)[01][2]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds
4566312	Process Plan Files(4567141)[01][1]	whunter	Success	WHQA1	Today at 9:35 AM	a few seconds

1 - 20 of 85 items

Example Job Results view

To view detailed results for a particular job execution, hover your cursor over the job and then click the View icon in the far right column. This opens the job properties to the **Job Results** section, with the corresponding execution ID expanded. You can further expand the job results to see the specific tasks that were executed.



File Processing | **Job Results**

Id	Result:	Start Time	Duration:	User
4566320	Success	Today at 9:56 AM	a few seconds	jdoe
4566310	Success	Today at 9:34 AM	a minute	jdoe
4566226	Success	Yesterday at 4:47 PM	a minute	admin

Job: File Processing **Start Time:** 10/02/2018 **Duration:** a few seconds **Server:** WHQA1 **Result:** Success

Task: File Processing **Start Time:** 10/02/2018 **Duration:** a few seconds **Status:** Success

1 - 3 of 3 items

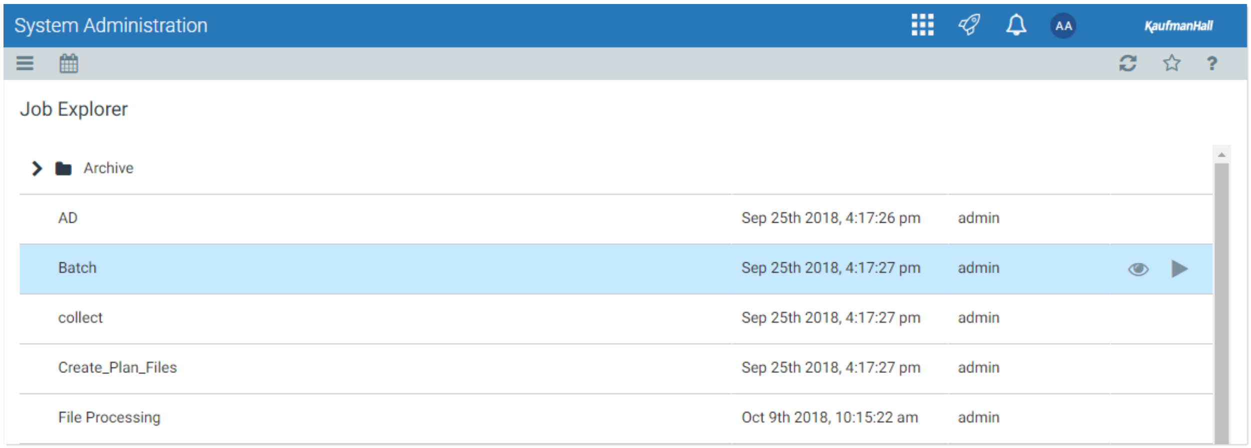
Example Job Results detail showing tasks executed

To view the detailed task results, hover your cursor over the task and then click the View icon in the far right column. This opens a dialog to display the results for that task. For example, for a file processing task, the detailed results would contain information such as the processing type and the number of passes, and the output that was created at the end of the process.

You can also view job results by opening a particular job directly and then browsing the results for that job. In many cases this may be easier than trying to find the particular execution of the job in the job results.

Viewing and running jobs

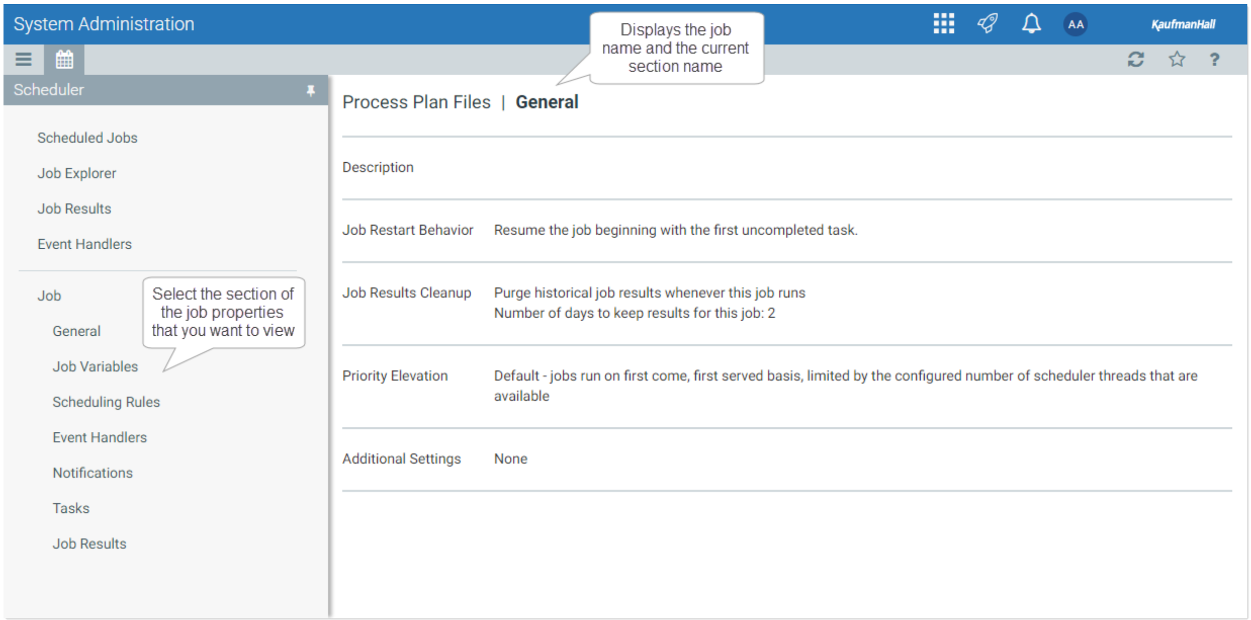
In the Job Explorer view, you can view specific jobs and run jobs manually. To perform these actions, hover your cursor over a job and then use the icons in the far right column.



Archive		
AD	Sep 25th 2018, 4:17:26 pm	admin
Batch	Sep 25th 2018, 4:17:27 pm	admin
collect	Sep 25th 2018, 4:17:27 pm	admin
Create_Plan_Files	Sep 25th 2018, 4:17:27 pm	admin
File Processing	Oct 9th 2018, 10:15:22 am	admin

Example Job Explorer view

If you view a job, the Scheduler panel updates to show the viewable sections of the job. You can switch between sections by selecting section names in the panel. By default, the **General** section is shown.



Displays the job name and the current section name

Select the section of the job properties that you want to view

Process Plan Files | **General**

Description

Job Restart Behavior: Resume the job beginning with the first uncompleted task.

Job Results Cleanup: Purge historical job results whenever this job runs
Number of days to keep results for this job: 2

Priority Elevation: Default - jobs run on first come, first served basis, limited by the configured number of scheduler threads that are available

Additional Settings: None

Example job properties

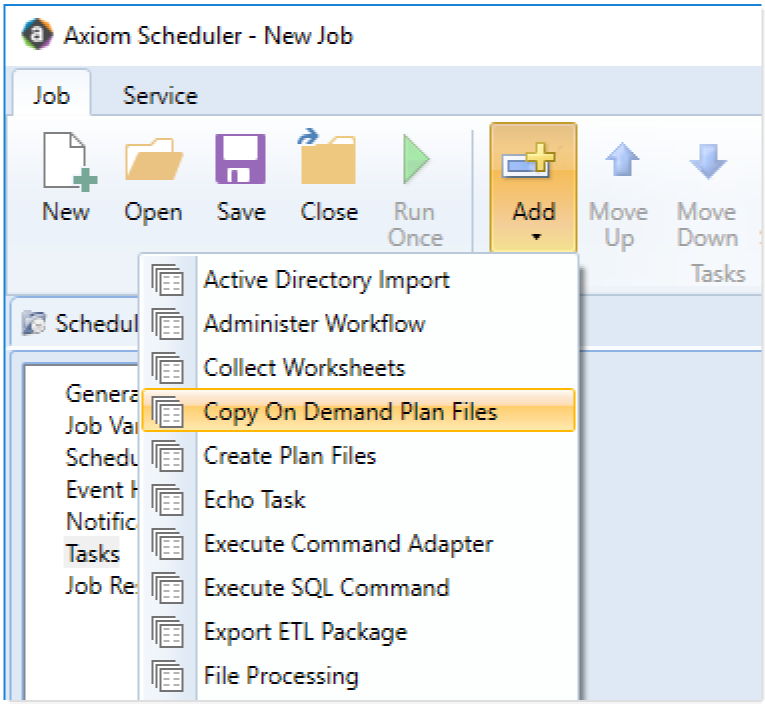
New task to copy plan files

You can now use Scheduler to copy on demand plan files from one file group to another. The primary purpose of this new feature is to support copying plan files from within an Axiom form, by triggering a Scheduler job using the RunEvent command.

The ability to copy plan files is typically used to support specialized solutions where plan files must be moved from one file group to another (to continue planning for them in a different file group), or copied from one file group to another (to be used as the "starter" for a new similar request in the new file group).

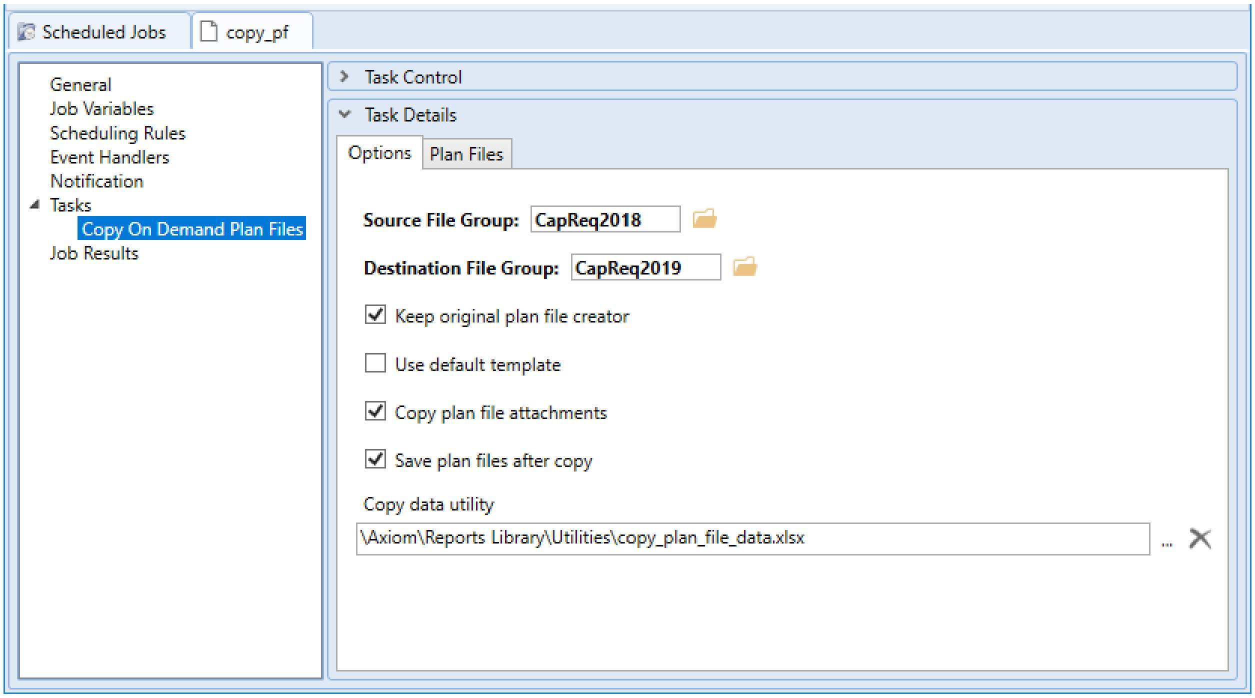
NOTE: Copying plan files from one file group to another is an advanced feature. It is up to the solution designer to ensure that the copied plan files will work as intended in the new file group. This requires advanced knowledge of file groups, file design, and data structures.

A new Scheduler task type is available, named **Copy On Demand Plan Files**.



Adding new copy task in Scheduler

The task performs the same basic operation as the existing **Copy On Demand Plan Files** command that can be used in task panes and ribbon tabs, and has the same basic settings. You must specify the source file group and the target file group, configure various options that impact the copy operation, and specify the plan files to copy.



Source File Group: CapReq2018

Destination File Group: CapReq2019

☑ Keep original plan file creator

□ Use default template

☑ Copy plan file attachments

☑ Save plan files after copy

Copy data utility

\Axiom\Reports Library\Utilities\copy_plan_file_data.xlsx

Settings for new copy task

Because in most cases the Scheduler task will be triggered for execution using RunEvent (such as from an Axiom form), all of the copy task settings can be overridden using a set of system variables. For example, the system variables CopyDataSourceFileGroupID and CopyDataTargetFileGroupID can be used to override the source and target file groups. For more information, see the following topic in Axiom Software Help: *Copy On Demand Plan Files task* (AX1752).

Additional enhancements

▶ OData API

Version 2018.4 introduces support for a data API using the Open Data Protocol (OData). The OData API can be used in custom solutions to access data from the Axiom Software database.

The Open Data Protocol is a standard data access protocol for the web. Use of this standard protocol makes it easier for customers to leverage existing expertise and develop solutions, and easier for Kaufman Hall to develop and support the API. Information on this standard can be found at https://www.odata.org (external link).

All requests to the OData API must be authenticated using a key generated by a valid Axiom Software user name and password. Each OData session should start with a POST request for the key. This key must then be included in an Authorization header in each subsequent GET request for data. All data returned for the session honors the security settings of the user identity used to generate the key.

Request data for the API is returned in JSON format. Each row of data returned from the table is a JSON object with *columnname*: *value* pairs. Standard OData syntax can be used to specify the table and columns to return, filter and order the request data, and so on.

For more information, including example authentication and data requests, see the following topic in Axiom Software Help: *OData API* (AX1749).

▶ Defining allowed email domains for Axiom Software

You can now optionally define a list of allowed email domains for Axiom Software, so that emails are only sent if the recipients are in the allowed domains. This feature is intended for organizations that want to block sending email from Axiom Software to recipients outside of the organization.

To define a list of allowed email domains, use the **EmailDistributionAllowedDomainList** system configuration setting. By default this setting is blank, which means that Axiom Software will attempt to send email to any domain. You can list one or more domains, separated by commas. For example:

```
@kaufmanhall.com
```

Emails can only be sent to recipients in the @kaufmanhall.com domain.

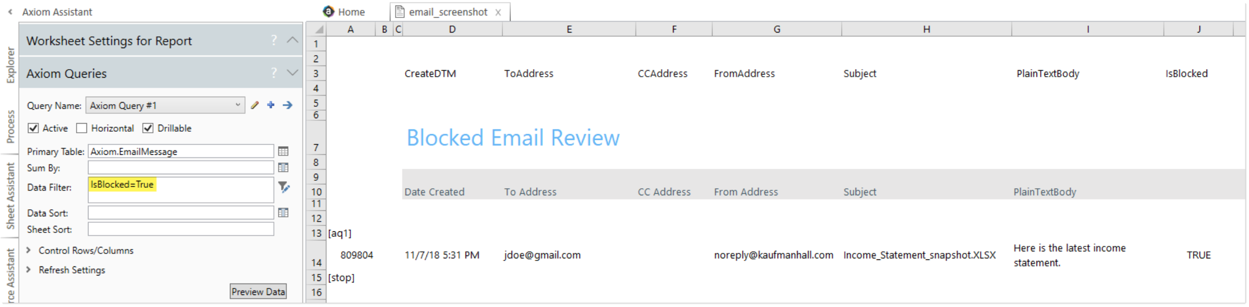
```
@kaufmanhall.com, @axiomepm.com
```

Emails can be sent to recipients in either the @kaufmanhall.com domain or the @axiomepm.com domain.

If an email has a recipient that does not belong to one of the listed domains—in any of the To, CC, or BCC fields—then the email is flagged as blocked. Blocked emails are prevented from being processed by the **System.SMTPEmailDelivery** job in Scheduler. If an email is blocked, it is not sent to any recipients, even if some recipients belong to allowed domains.

To modify the system configuration settings, you can use the Software Manager, or use a Save Type 4 report to the **Axiom.SystemConfiguration** table. You can find an template for this type of report at `\Axiom\Axiom System\Document Templates\Support Utilities\SystemConfigurationTemplate.xlsx`. Only administrators can access this template and change the system configuration settings.

If you have defined a list of allowed email domains, then you may want to review the blocked emails to see which recipients are being blocked. This can be useful for auditing purposes, and also to verify that your list of allowed domains is complete and correct. In order to do this, you can create an Axiom query to the new **Axiom.EmailMessage** system table. This table contains the recent email messages that have been generated by Axiom Software. The Axiom query can bring in the email recipients, the email subject and body text, and also whether the email was blocked. If you only want to view the blocked emails, you can filter the query by the **IsBlocked** column. For more information on the columns in this table, see the following topic in Axiom Software Help: *Axiom.EmailMessage*.



Query Name: Axiom Query #1

☑ Active □ Horizontal ☑ Drillable

Primary Table: Axiom.EmailMessage

Sum By:

Data Filter: IsBlocked=True

Data Sort:

Sheet Sort:

	A	B	C	D	E	F	G	H	I	J
3				CreateDTM	ToAddress	CCAddress	FromAddress	Subject	PlainTextBody	IsBlocked
7				Blocked Email Review						
10				Date Created	To Address	CC Address	From Address	Subject	PlainTextBody	
13	[aq1]									
14	809804			11/7/18 5:31 PM	jdoe@gmail.com		noreply@kaufmanhall.com	Income_Statement_snapshot.XLSX	Here is the latest income statement.	TRUE
15	[stop]									

Example report using Axiom.EmailMessage to review blocked emails

Once an email is flagged as blocked, there is no automated way to re-send it. Adding the blocked email domain to the allowed list will allow future emails to be sent to that domain, but it will not cause previously blocked emails to be sent.