# System Administration Guide

Axiom Software

Version 2018.1

KaufmanHall

axiom software

Kaufman Hall
Axiom Software
10260 SW Greenburg Road, Suite 710
Portland, Oregon 97223
877.691.9969 (Toll-Free)
503.977.0234 (Local)
www.kaufmanhall.com

# Table of Contents


# Introduction

This guide discusses system administration for Axiom Software, including data and file management.

## Intended audience

This guide is intended for administrators who are responsible for the setup and ongoing administration of Axiom Software.

## What is covered in this guide?

This guide covers the following aspects of system administration:

- Table administration
- Importing / exporting data
- Auditing configuration and review
- File management
- Customizing the Axiom user interface for users and roles

## What is not covered in this guide?

The following related topics are not covered in this guide:

- File group administration. For more information on creating and managing file groups, see the *File Group Administration Guide*.
- Security administration. For more information on creating users and roles and configuring security settings, see the *Security Guide*.
- Scheduler. For more information on setting up Scheduler and scheduling tasks for automated execution, see the *Scheduler Guide*.
- Process Management. For more information on defining and administering processes, see the *Process Management Guide*.
- Axiom file setup. For more information on querying and saving data via Axiom files, and other file setup options, see the *Axiom File Setup Guide*.

All documentation for Axiom Software can also be accessed using the Axiom Software Help files.

## Axiom Software Client versions

This guide discusses functionality that is available in the Axiom Excel Client and the Axiom Windows Client. Screenshots of features may show either client. The Axiom Software functionality is virtually identical and applies to both clients.


# Table Administration

This section discusses how to administer database tables for the Axiom Software system. Tables are used to hold all data in Axiom Software, including imported data and planning data saved from plan files. Tables also define the dimensions and grouping relationships in your system.

## About tables and columns

This section contains essential information to understand tables and columns in Axiom Software.

### About table classifications

Each table in Axiom Software is classified as one of the following:

- Data table
- Reference table (including picklist tables and KPI tables)
- Document Reference table

This topic discusses the properties and general design considerations for each table classification.

#### Data tables

Data tables hold imported or calculated financial data (or both). Data tables are the primary targets of save-to-database processes in plan files, and are the primary sources of data for reports.

Data tables can have multiple key columns. For example, the key columns for a data table might be DEPT and ACCT. Every combination of these key columns defines a unique row of data in the table. In this case two rows of data in the database cannot share the same account and department code.

Key columns in data tables typically have lookup relationships with reference tables that define the valid values for the key columns. For example, if a key column in the data table is ACCT, that column would have a lookup to the ACCT column in the ACCT reference table to define the set of valid accounts. However, key columns are not required to have a lookup relationship. If a key column does not have an assigned lookup column, then the entries for that column are not validated during an import or a save-to-database.

By default, non-admin users do not have access to the data in data tables. If you want non-admin users to be able to query data in a data table, they must be granted access to the table in Security. Table types may be used to group related tables and manage security access as a group.

### Reference tables

Reference tables hold lists of information to be referenced elsewhere. Reference tables can be used for many purposes, from lists of planning codes to import mapping tables.

Reference tables also define grouping relationships for security filters and reporting. For example, a department table might also have columns that assign each department to a region, country, or VP. By defining a filter criteria statement based on a grouping column, you can restrict access to data or define summing levels for a data query.

Reference tables can be designated as the plan code table for a file group. In this case, each code or record in the table has an associated plan file, and the table can contain additional columns to drive file group processes.

Reference tables can have only one key column. Each entry in that key column must be unique.

By default, non-admin users do not have access to the data in reference tables. If you want non-admin users to be able to query data in a reference table, they must be granted access to the table in Security. If the table holds common data that all users need to be able to access, such as ACCT and DEPT tables, you can use the Everyone role to grant access to all users.

Reference tables support two special sub-classes of tables, intended for specific purposes. These special tables generally follow the same rules as reference tables, but with some additional requirements and restrictions.

- **Picklist tables:** Picklist tables hold lists of items from which users can make selections. For example, you might want users to select from a list of defined categories or status codes, and these lists can be defined as picklist tables. For more information on using picklist tables, see About picklist tables.
- **KPI tables:** KPI tables hold key performance indicator values (KPIs), for display within reports. The KPI Panel component in Axiom forms and web reports can display KPI values held in KPI tables. For more information on using KPI tables, see About KPI tables.

### Document reference tables

Document reference tables are single-key tables that are controlled using a source spreadsheet file. Each document reference table is linked to a specific managed Axiom file, where the table and its data are managed. Traditionally, document reference tables are used to maintain file group driver tables, and can also be used for other data if desired.

Save Type 3 is used to create a document reference table in the database, based on the structure of the source sheet. Each time the Axiom file is saved, the associated document reference table is re-created

based on the structure and data within the Axiom file. For this reason, document reference tables are not created or edited from the **Tables** menu (although you can launch the source file from that location, if you have the appropriate rights).

By default, all users have security permission to query the data in document reference tables, via the Everyone role.

Generally speaking, document reference tables are being de-emphasized in favor of using regular reference tables and picklist tables, especially in Axiom Software packaged products. Although document reference tables are still supported, it is recommended to use other solutions when possible. Document reference tables have many limitations as compared to the other table classifications. For example, document reference tables do not support the full range of table and column properties, and cannot belong to table types. Columns in document reference tables cannot look up to other tables, and other tables cannot look up to document reference tables.

## About key columns

Every table in the database must have at least one key column. Reference tables and document reference tables can have only one key column, whereas data tables can have multiple key columns. When you are defining a column for a table, you must specify whether it is a key column.

Key columns define unique records of data in the table. If a table has one key column, then each value in that key column must be unique and defines a unique record in the table. If a table has multiple key columns, then each combination of values in those key columns defines a unique record in the table.

For example, if you have a reference table DEPT, the key column is DEPT, and that column contains each unique department code. The remaining columns in the DEPT table contain information that describes each unique department code, such as the department name, or the region the department belongs to, or the manager of that department.

Data tables typically have at least two key columns, such as DEPT and ACCT. Each combination of a department and an account defines a unique record in the data table. This allows you to plan for each account at a department level. The specific key columns vary depending on the level of planning. For example, you might forecast at the division and account level, or perform capital planning at the project and account level. When defining key columns for data tables, consider whether the data in that column must define a unique record of data in the database, in relationship with the other key columns in the table.

Your plan may require additional key columns in a data table. For example, you might want to track item-level detail for certain accounts, such as Travel. You could have a Detail sheet in your plan files where each manager lists the specific detailed items that make up the Travel total, and assigns each one a Detail code (such as Detail_01). In this case, the Detail column needs to be a key column, so that each Dept / Acct / Detail combination is a unique record in the database. If Detail is not a key column, the save process would either fail due to a summing error, or the detail data would be collapsed within the Dept / Acct combination. For example:

| DEPT | ACCT | Detail | M1 | M2 |
|---|---|---|---|---|
| 100 | 3000 | Detail_01 | 0 | 1000 |
| 100 | 3000 | Detail_02 | 1000 | 0 |

If Detail is not a key column, the save process would fail, because the database can only contain one row of data for the combination of DEPT 100 / ACCT 3000, and Axiom Software cannot sum the contents of the Detail column.

If Detail *is* a key column, then the save process would be successful, saving one row of data for DEPT 100 / ACCT 3000 / Detail_01, and another row of data for DEPT 100 / ACCT 3000 / Detail_02.

Key columns can only be the following data types: Identity, Integer (all types), String, or Date. Other data types are not valid for key columns.

### Key columns and lookup columns

A key column can be assigned a lookup column, so that the values in the key column are validated against a master list in another table. For example, the key column GL2018.ACCT would be assigned a lookup column of ACCT.ACCT. However, key columns do not have to use a lookup column, if a master list does not make sense for the data that the key column will hold.

Whether a key column has a lookup column or not, key columns cannot contain blank values.

### Descriptions for key values

When defining reference tables, each value in the key column typically has a unique description. For example, if the key column is DEPT.DEPT, each department code in that column has an associated description. In almost all cases, whenever the department codes are presented to users you want the description to display as well, because most users need the descriptions to find the departments they are looking for (or just to reinforce that they have the correct department).

To pair the key column with its descriptions, you create another column in the table to hold the descriptions and then enable the **Describes Key** property for that column. Often this column is named "Description" as well, but it does not have to be. When Axiom Software displays the key values to end users in dialogs or in other built-in contexts such as drilling, the descriptions will be automatically included.

You can designate multiple columns as description columns, however, you should consider how this content will display to end users when multiple columns are included along with the key value. Too much information may be confusing and difficult to read in certain contexts.

### Aggregating duplicate keys during data save

Because each combination of key codes defines a unique record of data in the database, if two rows to be saved have the same key codes, then by default Axiom Software attempts to aggregate the data in these rows when saving to the database.

**NOTE:** This default aggregation behavior only applies to data tables. If duplicate key codes are found when saving to a reference table or a document reference table, an error always results.

For example, imagine that you have the following rows in a plan file:

| DEPT | ACCT | M1 | M2 |
|---|---|---|---|
| 100 | 2000 | 50 | 150 |
| 100 | 2000 | 100 | 200 |

When a save is performed, the duplicate keys are aggregated by summing the numeric fields, resulting in the following record of data in the database:

| DEPT | ACCT | M1 | M2 |
|---|---|---|---|
| 100 | 2000 | 150 | 350 |

When string columns are used in a data table, you must take care not to create a situation where the string column must be aggregated during a save-to-database. For example:

| DEPT | ACCT | M1 | M2 | Comment |
|---|---|---|---|---|
| 100 | 2000 | 50 | 150 | Software release event |
| 100 | 2000 | 100 | 200 | Party planning funds |

In this case, the save process will return an error, because it cannot sum the contents of the Comment column. If both Comment fields contained the same string—for example, if they both said "Software release event"—then the save would process successfully because it could reconcile the contents of the Comment column.

If aggregation applies, non-key fields are treated as follows:

- Numeric fields are summed.
- Integer fields (all types) are summed, unless the column is a validated column, in which case the values must match).
- String, Date, DateTime, and Boolean fields must match.

Aggregation applies by default when saving data to a data table using Save Type 1 or when using an import utility. In both cases, the aggregation can be disabled if desired (meaning that the data must not contain duplicate keys or else an error will result). When using Open Table in Spreadsheet, aggregation is not allowed in any case and duplicate keys will result in an error.

### Alternate key columns

Tables can have one or more *alternate key columns*. Alternate key columns are columns that are required to have unique values. These columns can be used to uniquely identify each row in the table, even though they are not formally designated as key columns. All tables except document reference tables can have alternate key columns.

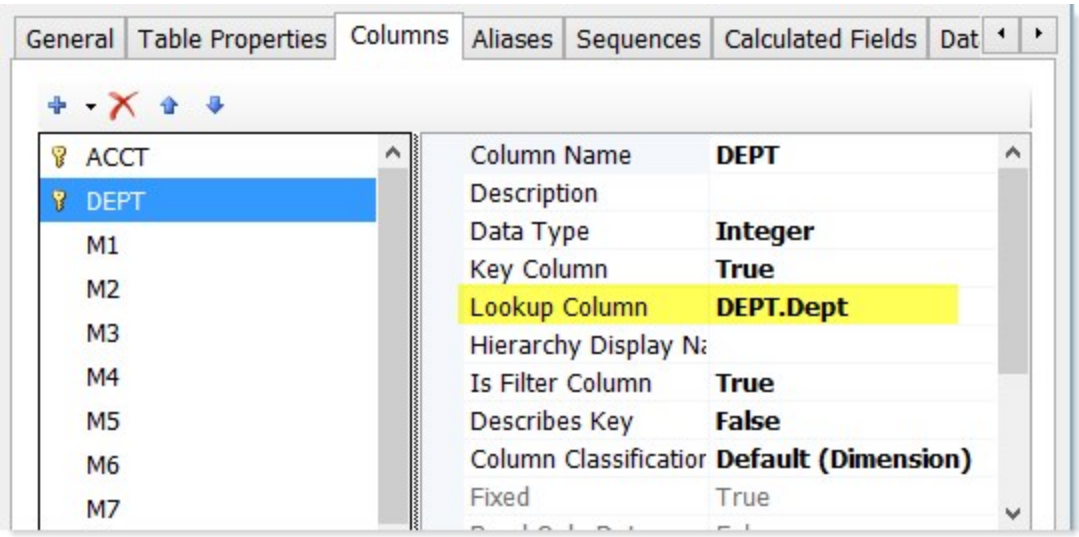
Currently, the primary purpose of alternate key columns is when using the alternate key mapping feature of Save Type 1. This feature allows you to save data by using an alternate key to look up the actual key of a table. However, the alternate key feature can be used for any column where unique values are required.

When the Axiom Software documentation uses the term *key column*, this refers to primary key columns only (columns where **Key Column** is set to **True**). You should not assume that a requirement or behavior described in the documentation for key columns also applies to alternate key columns, unless alternate key columns are explicitly mentioned.

## About lookup columns

Certain columns in tables need to be restricted to a specific set of valid values. These columns are known as *validated columns*, because before any data can be saved to the column, the data is first validated against a designated *lookup column*. The lookup column defines the set of allowed values for the validated column.

For example, if the table GL2018 has a column DEPT, then you need to restrict the values in that column to the set of valid departments for your organization. This list of valid departments is defined in the separate DEPT reference table, in the key column (for example: DEPT.DEPT). You can enforce this validation by assigning DEPT.DEPT as the lookup column for GL2018.DEPT.




*Lookup Column assignment in the Edit Table dialog*

The GL2018.DEPT column is now a *validated column* (also known as a *linked column*), and the DEPT.DEPT column is a lookup column. The DEPT reference table is also referred to as a *lookup table* for the GL2018 table.

GL Data Table

| DEPT | ACCT | M1 | M2 | M3 |
|---|---|---|---|---|
| 20000 | 6000 | 2644.23 | 1173.38 | 5046.89 |
| 20000 | 6200 | 0.00 | 23000.00 | 18400.00 |
| 20000 | 6300 | 6120.00 | 3187.50 | 6948.75 |
| 20000 | 6400 | 1565.53 | 2316.27 | 2284.36 |
| 20000 | 6500 | 11729.61 | 23459.14 | 11729.57 |
| 20000 | 6600 | 0.00 | 0.00 | 0.00 |

Validated Column

DEPT Reference Table

| Dept | Description | Template | WorldRegion | Country |
|---|---|---|---|---|
| 20000 | Corporate | Master Budget Template | Corporate | USA |
| 21000 | Corporate Administration | Master Budget Template | Corporate | USA |
| 22000 | Information Technologies | Master Budget Template | Corporate | USA |
| 23000 | Purchasing & Materials Mgmt | Master Budget Template | Corporate | USA |
| 24000 | Business Development | Master Budget Template | Corporate | USA |
| 25000 | Finance | Master Budget Template | Corporate | USA |
| 26000 | Portfolio Management | Master Budget Template | Corporate | USA |
| 27000 | Human Resources | Master Budget Template | Corporate | USA |
| 28000 | Facilities | Master Budget Template | Corporate | USA |
| 29000 | Legal | Master Budget Template | North America | USA |
| 40000 | Los Angeles - Store 34 | Master Budget Template | North America | USA |
| 40500 | West Coast Distribution | Master Budget Template | North America | USA |

Lookup Column

*Lookup column relationship*

Note the following about lookup columns and validated columns:

- Only key columns of reference tables can be lookup columns.
- A column can be the assigned lookup column for any number of validated columns. For example, multiple DEPT columns in multiple tables can all use DEPT.DEPT as a lookup column.
- A validated column must have the same data type as its assigned lookup column. There is one exception—if the lookup column is an Identity column, then the validated column must be an Integer column. If the columns are string columns, then they must also have the same string length and Unicode status (enabled or disabled).
- Lookup columns and validated columns are not required to have the same column name, though often it is a good idea to use the same name to avoid confusion and to reinforce the relationship between the columns.

Both key and non-key columns can be validated columns. For example, you could have a reference table DEPT, with a non-key column named DEPTMGR that contains the manager name of each department. If you wanted to restrict that column to a set of valid manager names, you could create another reference

table in the database named MGR, and then assign MGR.MGR as the lookup column for DEPT.DEPTMGR.

You can assign a lookup column when creating or editing a column. If the table already contains data when you make the lookup column assignment, then the assignment will only be allowed if the existing data can be validated against the lookup column.

Once a lookup relationship has been created, the validated column and the lookup column become interdependent. You cannot save data to the validated column unless it matches a valid value in the lookup column. Also, you cannot delete a value from the lookup column if it is used in the validated column.

### Validated columns and default values

When you make a column a validated column, you should decide whether to also set the default value of the column to a valid value in the lookup column. If the default value is a valid value for the column, then you have the option to omit the validated column from save processes and/or leave the column blank in save processes, and the column will use the default value automatically.

Alternatively, you can decide to not set a valid default value for the column, which means that the column must always be included in save processes and must always have a defined value. You might do this if you always want the column to have a specified or calculated value for the save process as opposed to using a default value.

For example, imagine that you have a validated column with a lookup to a column that contains category codes. If it is acceptable for this validated column to use a default category if no category is set by the save process, then you would set the default value of the validated column to the desired default value. This default value would likely be a pseudo-code such as 99, meaning no assigned category. On the other hand, if you always want the category to be explicitly selected or calculated for the save process, then you might want to leave the default value at an invalid value, so that the save process will fail if the validated column is missing or blank.

### Self-referencing lookup columns

You can set up "self-referencing" lookup columns within a reference table. This means that the validated column and the lookup column are within the same table. For example, the column DEPT.DeptMap can use DEPT.DEPT as a lookup column, both within the DEPT table.

This can be useful when setting up mapping for plan files. You may be mapping certain departments into a "parent" department for planning purposes, and you manage those mappings in the DeptMap column. You want to make sure that only valid department numbers are used in this column, so you assign it a lookup column of DEPT.DEPT. If you do this, keep in mind that you cannot have blank values or entries like "Not Mapped" in the DeptMap column—all rows must be completed and all entries must be valid department numbers.

This approach also allows you to force Axiom to use values from the "parent" record when mapping. For example, consider the following setup for the ACCT table:

| ACCT | Description | BgtCombineDuringQuery | BgtSectionInFile | CMAssign |
|---|---|---|---|---|
| 1000 | Main Rev | 1000 | Fixed | Fixed CM |
| 1100 | Other Rev 1 | 1000 | Fixed | |
| 1200 | Other Rev 2 | 1000 | Variable | Variable CM |

Imagine that you are mapping accounts 1100 and 1200 into account 1000 for planning purposes. Under normal circumstances you might have an Axiom query data range filter of `ACCT.CMAssign='Fixed'` to bring accounts into that section, and the calc method assignment for the query would be `Acct.CMAssign`.

However, this approach will not work as expected when mapping, because when summing the three records together the query will choose the string values from BgtSectionInFile and CMAssign seemingly at random. It might select Variable as the section instead of Fixed, and it might select the Variable CM or the blank cell from the CMAssign column (the latter case meaning the default calc method would be used). This is not the desired result—what you want is a way to force it to use the values from the "parent" account, which in this case is account 1000.

The way around this is to configure BgtCombineDuringQuery to use ACCT.ACCT as a lookup column. Then in the Axiom query setup, you can use the "multiple-level" column syntax, which will enforce the use of the values for the "parent" account. This is the same syntax that is used when there are actually multiple tables involved in the lookup chain. Effectively, this causes the account table to be joined back in after the records are summed, and then use the values from the BgtCombineDuringQuery column to determine the account.

In the case of the data filter, you would specify `ACCT.BgtCombineDuringQuery.BgtSectionInFile='Fixed'`, and in the case of the calc method assignments, you would specify `ACCT.BgtCombineDuringQuery.CMAssign`. You would also use this syntax as appropriate in the field definition for the query, for example to bring in the appropriate account description: `ACCT.BgtCombineDuringQuery.Description`.

## Design considerations for validated columns

- If the data in a table is zeroed as part of a Save Type 1 process (using the **Zero on save** feature) or when using **Zero/Delete Table Data**, the values in validated columns are left as is, whether the validated column is a key or a non-key.
- If a column is assigned a lookup column, then the values in that validated column can never be blank or "null." All lookup columns are key columns, and key columns cannot be blank. Therefore any blank value in the validated column cannot be validated against the lookup column.

## About picklist tables

Picklist tables hold lists of items from which users can make selections. For example, you might want users to select from a list of defined categories or status codes, and these lists can be defined as picklist tables. Picklist tables are a special type of reference table, specially designed to be used as picklists.

Picklist tables have the following advantages over using regular reference tables for the same purpose:

- Picklist tables can be easily created and edited using the Web Client Table Manager. This makes it easier for non-technical users to manage picklist tables and their values.
- Refresh variables provide special support for selecting values from picklist tables. This feature makes it easy to present users with custom text values for selection, but save those selections back to the database using standard integer codes.
- File groups have a special picklist variable type, allowing enhanced control over how the picklist can be used within the file group.

Additionally, some systems may have been using document reference tables to maintain picklist-style data, in order to leverage the easy-to-use spreadsheet interface for editing picklist values. Picklist tables also provide an easy-to-use interface, and additionally picklist tables can be used as lookup tables for other columns (whereas document reference tables cannot).

### Picklist column structure

Picklist tables have a predefined structure of three columns. All of these columns are created by default when the picklist table is created, and cannot be deleted.

| Name | Column Properties | Description |
|---|---|---|
| Code | Integer<br>Key Column | Required. Defines a numeric code for each record. Each code must be a unique positive integer.<br>When users select items from a picklist, items are returned and stored using this numeric code. |
| Value | String (50)<br>Alternate Key Column | Required. Defines a text value for each record. Each text value must be unique.<br>Users select items from the picklist using these values. |
| Description | String (1023) | Optional. Defines a description or explanatory text for each record.<br>When users select items from a picklist, you can optionally display this text to help users choose the correct values. |

The main column properties of picklist columns cannot be modified, such as column name, data type, key column status, etc. Only minor column properties such as read-only status can be modified.

## Picklist table editor

Picklist tables have a special editor in the Web Client, designed to make it easy for authorized users to create new picklist tables and to edit picklist values. For more information, see Managing picklist tables.

Picklist tables can also be created and edited within the Desktop Client if needed, using the normal table creation and editing features.

When using the Web Client editor, only the three required columns of Code, Value, and Description are visible and editable. Although it is possible to add other columns to a picklist table using the Desktop Client, these columns have limited usefulness and must be populated using Open Table in Spreadsheet or Save Type 1 instead of using the Web Client editor.

## Using picklist tables

Picklist tables have a set structure that combines an integer code key column with a string value column. You can define the string value as needed while maintaining the standard integer code. When users select items from the picklist, you can display the text values but save back the selection as the integer code.

In order to display picklists to users, you can use any selection feature that is available to reference tables. In additional, several features provide special support for picklist tables:

- ComboBox and Grid refresh variables support special syntax to specify a picklist table as the source of data, to make it easy to set up and use refresh variables with picklist tables.
- File group picklist variables support optional properties to store useful information about the use of the picklist table in the file group. You can associate the picklist with a column in the plan code table, flag the picklist as required, and conditionally enable the picklist based on other related values. Using the functions GetFileGroupVariableProperty and GetFileGroupVariableEnablement, you can return this information into a template or other file, and then use this information to dynamically configure features in the template (such as custom save validation for a required picklist).

## Picklist table restrictions and limitations

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

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

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

Generally speaking, whenever a feature says a reference table is valid for use, a picklist table can be used as well. Any exceptions are noted in the specific feature documentation.

### Picklist table security

By default, non-admin users do not have access to the data in picklist tables. When a new picklist is created, you must configure read-only or read / write access to the table as needed (unless the picklist is assigned to a table type to inherit permissions).

Note the following special considerations for picklist table security:

- Because picklist tables are comprised of simple lists and do not normally contain sensitive information, you may want to configure read-only access on the Everyone role to simplify security. Make sure to enable **Specify custom write access** and then leave the filter blank so that users do not have read / write access, because all users with read / write access can modify picklist values using the Web Client Table Manager. Read / write access should be restricted to a small subset of users who define and administer the picklist values.
- Picklist tables can belong to picklist table types, for purposes of easily granting full read or write access to a set of related tables. You might create different table types to hold picklist tables for different types of planning (Capital, Budget) or for different facilities or entities. You can then grant access at the table type level, to be inherited by all tables in the table type.
- A special security permission, Administer Picklists, is available to give users rights to create and manage picklists in the Web Client Table Manager. Users with this permission can do the following:
    - Create new picklist tables as needed. Note that these picklists must be assigned to table types that the user has access to, so that the user will have permission to the new table once it is created.
    - Modify picklist table properties and delete picklist tables. The user must have at least read-only access to the table in order to perform these actions.
    - These special permissions only apply to the Web Client Table Manager. Administer Picklists users cannot create or manage picklist tables in the Desktop Client.
- Any user with read / write access to the table can edit picklist table values using the Web Client Table Manager. This is different behavior from other tables, where having read / write access to a table does not automatically give the user an interface to view and edit the table values. Because of this difference, you should not give a user read / write access to a picklist table unless you want them to have free rein to edit the values in the table.

## About KPI tables

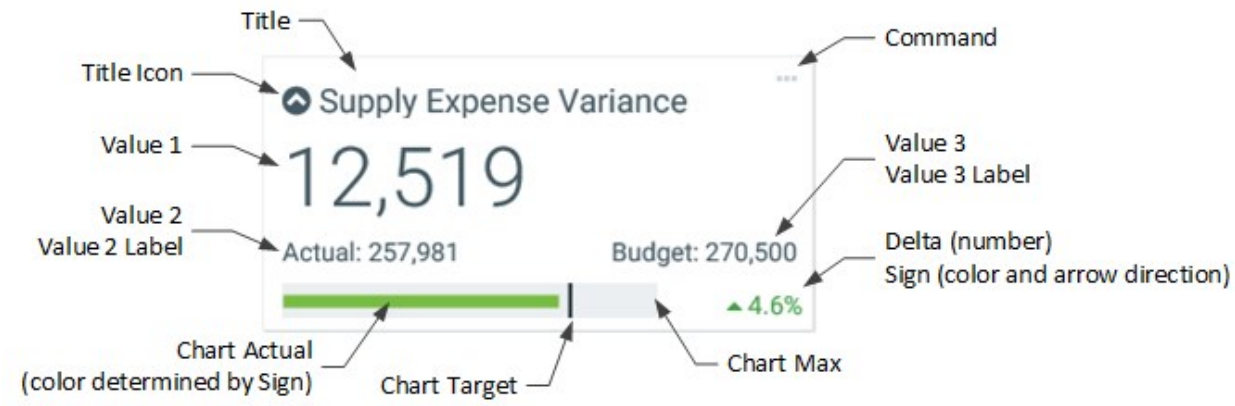
KPI tables store key performance indicators (KPIs) for your organization. These KPIs can be referenced in web reports and in Axiom forms using the KPI Panel component.

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

**NOTE:** This feature is only available to systems running Axiom Software version 2018.01.54 or higher.

## Column structure for KPI Tables

KPI tables have a predefined column structure that corresponds to the values used by the KPI Panel component. The following diagram shows how these values are displayed in each KPI box:




All of the following columns are created by default when the KPI table is created, and cannot be deleted. All string columns are non-Unicode.

| Name | Column Properties | Description |
|---|---|---|
| Name | String (100)<br>Key Column | Defines a code that uniquely identifies each KPI in the table.<br>The code can be anything you want, as long as it is unique. For example, you might use a meaningful code format like KPI_ExpenseVar or KPI_ExpenseVar_D42000 (where KPIs are stored by department). |
| Title | String (250) | The title of the KPI, displayed at the top of the KPI box. The title typically indicates the type of KPI, such as Supplies Expense Variance or Customer Retention. Note that if the title is too long to fit on one line, it will not wrap to a second line. Instead, the text displays with an ellipses, and the user can read the full title text in the tooltip. |

| Name | Column Properties | Description |
|---|---|---|
| TitleIcon | String (64) | Optional. The name of an icon to display in the KPI title. Enter any valid icon name, such as `fa-dollar`. The icon names are the same as the symbol names available for use in Axiom form components such as Formatted Grids. If specified, the symbol displays in the far left of the title, before the title text.<br><br>To look up valid icon names, you can use the symbol chooser available for Formatted Grid components. In a form-enabled file, right-click a cell and then choose **Insert Formatted Grid Tag > Symbol**. Then use the **[...]** button to the right of the **Symbol** box to open the **Choose Symbol** dialog. You can hover your cursor over an icon to see its name. |
| Value1 | Numeric | The primary value to highlight for the KPI. This value displays in large, bold font directly underneath the title. This is the value that you want to draw the most attention to.<br><br>The number format of this value is determined by the value specified in the NumericType column. |
| Value2 | Numeric | Optional. A secondary value to show for the KPI. Value 2 displays in smaller font to the left side of the KPI box, directly above the bullet chart. A label can be defined for this value, using the Value2Label column.<br><br>The secondary values of Value 2 and Value 3 can be used to provide supporting information for the primary value or to provide additional information. For example, if the primary value is a variance, then Value 2 and Value 3 might display the actual and budget numbers used to calculate that variance. Or if the primary value is the actual number, then Value 2 and Value 3 might display the variance and the budget number to provide more context for the actual number. The values displayed are entirely user-definable.<br><br>The number format of this value is determined by the value specified in the NumericType column. |

| Name | Column Properties | Description |
|---|---|---|
| Value2Label | String (100) | Optional. The label for Value 2. The label precedes the value and displays with a colon, such as "Actuals: *Value 2*". The label should explain what Value 2 represents. |
| Value3 | Numeric | Optional. A secondary value to show for the KPI. Value 3 displays in smaller font to the right side of the KPI box, directly above the Delta value. A label can be defined for this value, using the Value3Label column. The value displays using the number format defined for the cell.<br><br>See the description of Value2 for more information.<br><br>The number format of this value is determined by the value specified in the NumericType column. |
| Value3Label | String (100) | Optional. The label for Value 3. The label precedes the value and displays with a colon, such as "Budget: *Value 3*". The label should explain what Value 3 represents. |
| ChartTarget | Numeric | Optional. A value that defines the target line for the bullet chart. This value can be omitted if it is not needed on the chart, or if you are defining a KPI with no chart. |
| ChartActual | Numeric | Optional. A value that defines the actual line for the bullet chart. This value is displayed using a red or green bar, where the color is determined by the Sign value. This value can be omitted if you are defining a KPI with no chart. |
| ChartMax | Numeric | Optional. The maximum value of the bullet chart. The chart target and actual values are represented in relation to this maximum value. This value is required if you want to display a chart for the KPI. If you don't want to display a chart, omit this value.<br><br>For example, if the actual value is 100 and the maximum value is 1000, then the actual bar will only take up 1/10th of the bullet chart. But if the maximum value is 150, then the actual bar will take up 2/3rds of the bullet chart. |

| Name | Column Properties | Description |
|---|---|---|
| Delta | Numeric | Optional. A value that illustrates the positive or negative measure of the KPI. This value can be omitted if not needed. If it is omitted and the KPI has a bullet chart, then the chart extends to fill the bottom area of the KPI box.<br><br>The value is displayed to the right of the bullet chart, in either red or green (as determined by the Sign value). The value displays using the number format defined for the cell.<br><br>The Delta value can be used to show a variance percent or a raw difference value. It can also be used to show the change in value since the last time the primary KPI value was measured. |
| Sign | String (8) | Optional. Specifies whether the primary KPI value is trending up (positive) or down (negative). Enter either `Up` or `Down`. If omitted, Down is assumed.<br>• If Up, then an up-arrow displays in front of the Delta value. The Delta value and the actual bar of the bullet chart display in green.<br>• If Down, then a down-arrow displays in front of the Delta value. The Delta value and the actual bar of the bullet chart display in red. |
| Style | String (32) | Optional. Specifies a color style to set the background color of the KPI box. The following Axiom color styles are supported (specify one per KPI): S1, S6, A11, A51, P5, P6, P7, P9, P10. When using darker background colors, the text in the KPI automatically adjusts to white. |
| Tooltip | String (250) | Optional. Defines a tooltip to display when a user hovers over the command button in the KPI box. The tooltip should tell the user what action will occur when the user clicks the button. If no command is defined, the button does not display and the tooltip is ignored. |

| Name | Column Properties | Description |
|---|---|---|
| Command | String (1023) | Optional. Specifies an action to execute when the user clicks the **...** button in the top right corner of the KPI box. If a KPI does not have a defined command, no button displays on the box.<br><br>The action can be any of the following:<br>• A URL (starting with HTTP/S) to open a web page, Axiom form, or web report.<br>• A document shortcut (`document://filepath`) to a file in the Axiom Software file system.<br>• A command from the Command Library. Multiple command strings can be used, separated by commas. You can use any command that is supported for use in Axiom forms, though some commands may not make sense to execute from a KPI Panel.<br><br>The command strings use the same syntax supported by the Button tag for Formatted Grid components. If the command string is invalid, no error displays and no action occurs when the button is clicked.<br><br>Commands from the Command Library are only supported when the KPI table is used by a KPI Panel in an Axiom form. Web reports do not support use of commands, but will honor URLs and document shortcuts.<br><br>To create a command string, you can use **Axiom Wizards > Command Wizard** on the right-click menu in any Axiom file. |

| Name | Column Properties | Description |
|---|---|---|
| NumericType | String (16) | Specifies the numeric type of the values in Value1, Value2, and Value3 columns:<br>• Number<br>• Currency<br>• Decimal<br>This determines the format of these values within the KPI box. If blank, Number is assumed. The **Numeric Type** defined on the individual value columns is not used. |
| DeltaNumericType | String(16) | Specifies the numeric type of the value in the Delta column:<br>• Number<br>• Currency<br>• Decimal<br>This determines the format of the Delta value within the KPI box. If blank, Number is assumed. The **Numeric Type** defined on the Delta column is not used. |
| Hidden | Boolean | Specifies whether the KPI row is omitted from KPI Panel components (True/False). The default value is False.<br>You can set this to True in order to temporarily hide a KPI, or to archive a KPI without deleting it. |
| CalculatedDateTime | DateTime | Optional. Specifies the date/time of the KPI calculation. If defined, this value displays on the tooltip for the KPI title.<br>This is not an automatically calculated value, because the applicable date/time of the KPI may be different than the date/time the value was saved to the table. If you want to use this value, it must be explicitly defined and saved to the KPI table along with the other values. |

The main column properties of KPI columns cannot be modified, such as column name, data type, key column status, etc. Only minor column properties such as read-only status can be modified.

## Creating KPI tables

KPI tables have a special editor in the Web Client, designed to make it easy for table administrators to create new KPI tables.

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

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

## Using KPI tables

KPI tables have a set structure that provides the necessary KPI properties for a KPI Panel component. You can configure the component to use a KPI table, and then that component is automatically populated with the KPIs in the table.

KPI tables can be used KPI Panel components in Axiom forms and in web reports.

- In web reports, KPI Panel components are required to use a KPI table in order to provide KPI data to the component.
- In Axiom forms, KPI Panel components can use either a KPI table or a data source defined within the spreadsheet.

When configuring a KPI Panel component, you can optionally specify a filter to determine the KPIs that display in the component. For example, if you have added a Dept or Entity column to the KPI table, you can filter the component to only show KPIs relating to a particular department or entity. Any KPI with **Hidden** set to **True** is automatically hidden from all KPI Panel components.

## KPI table restrictions and limitations

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

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

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

Generally speaking, whenever a feature says a reference table is valid for use, a KPI table can be used as well. Any exceptions are noted in the specific feature documentation.

## KPI table security

By default, non-admin users do not have access to the data in KPI tables. When a new KPI table is created, you must configure read-only or read / write access to the table as needed (unless the KPI table is assigned to a table type to inherit permissions).

Note the following considerations for KPI table security:

- KPI tables can belong to table types, for purposes of easily granting full read or write access to a set of related tables. You can then grant access at the table type level, to be inherited by all tables in the table type. KPI tables can belong to any table type that uses the reference table classification; there is no special table type classification for KPI tables.
- KPI tables can be created by administrators or by users with the **Administer Tables** permission. There is no special permission specifically for administering KPI tables. Once a KPI table is created, standard table and table type permissions apply.

## About table index schemes

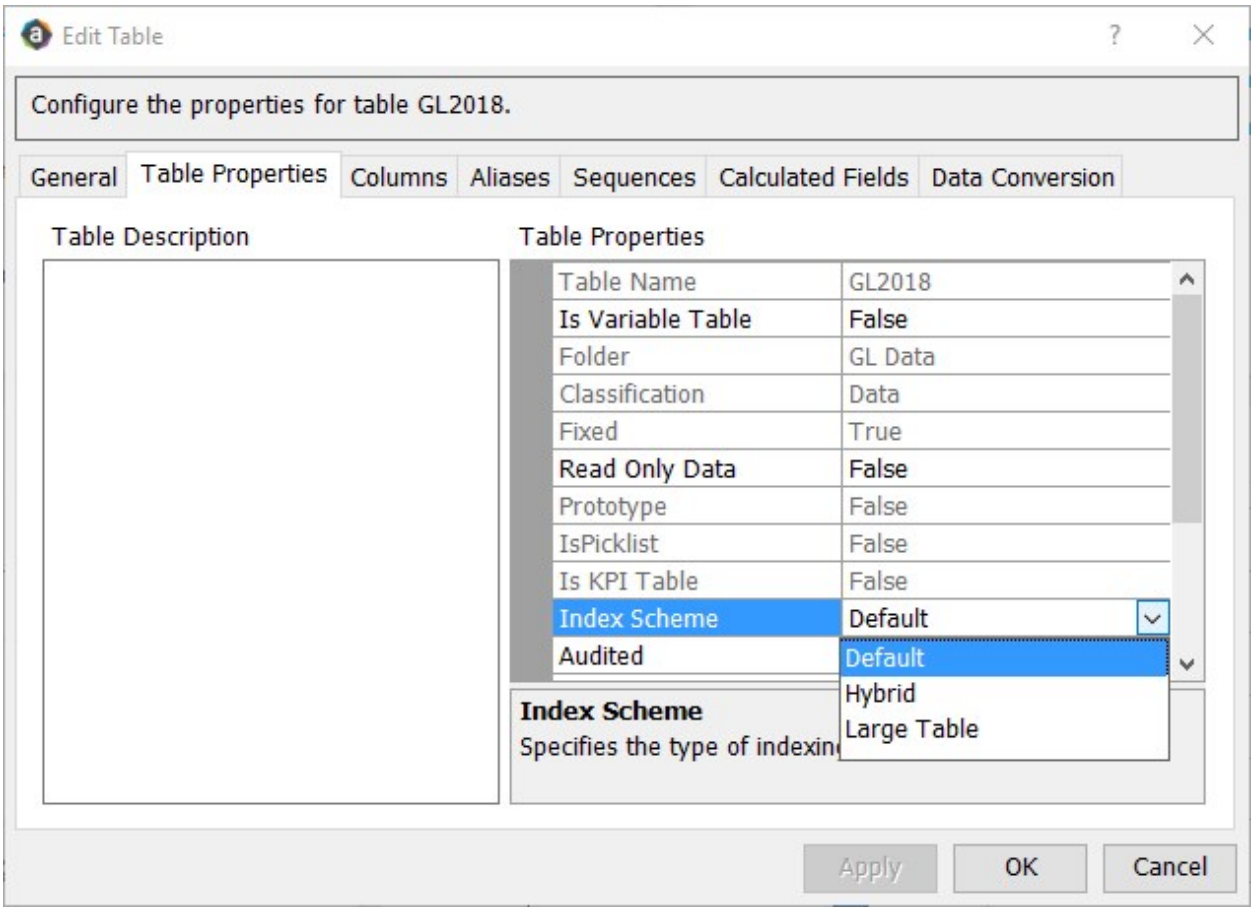
All data and reference tables are assigned to a table index scheme that determines how each table is indexed within the Axiom Software database. In certain situations, using a specialized index scheme can improve performance when querying data from the table or saving data to the table.

**IMPORTANT:** The table index scheme is an advanced feature that requires a good understanding of data structures and database management. You should only change the index scheme as instructed by your implementation consultant or Kaufman Hall Software Support. This feature is only available to systems running Axiom Software version 2018.01.54 or higher.

Axiom Software provides the following index schemes:

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

Index schemes are specified in the table properties. You can set the index scheme when creating a table, or when editing a table. However, if you want to use the Large Table index scheme, this must be specified when the table is created. After a table is created, it is not possible to change Default or Hybrid to Large Table, or vice versa. If necessary, you can clone the table and change the index scheme as part of the clone.




*Index Scheme in the table properties*

**NOTE:** In previous releases, the Large Table index scheme was managed using a separate table classification of Large Data tables. When upgrading to TBD or higher, all Large Data tables are converted to Data tables using the Large Table index scheme. The table should behave the same before and after the upgrade, the only change is in how the table is classified.

## System requirements

The Default index scheme can be used with any supported version of Microsoft SQL Server. The other schemes have greater system requirements:

- Hybrid: Requires SQL Server 2014 or higher
- Large Table: Requires SQL Server 2016 or higher

Before enabling these features, you should make sure that your system meets these requirements by consulting with your database administrator for on-premise systems, or with Kaufman Hall Software Support for Cloud Service systems.

## Table structure restrictions when using Large Table or Hybrid indexing

The following table structure restrictions apply to the Large Table and Hybrid index schemes.

### Hybrid

When using the Hybrid index scheme, the table cannot contain a String column that is set to maximum (unlimited) width.

### LargeTable

When using the Large Table index scheme, the following restrictions apply:

- Once a large table has data, it is not possible to add or remove columns in the table. The overall column structure is fixed.
- Identity columns are not allowed in large tables.
- Large tables do not support column aliases, column sequences, calculated fields, or data conversions.

#### Data restrictions when using Large Table indexing

Tables that use the Large Table index scheme are restricted in how data can be saved to the table. Only the following features can be used:

- Import utilities
- Copy Table Data

Save Type 1 and Open Table in Spreadsheet cannot be used to save data to large tables. Open Table in Spreadsheet can only be used as read-only. Generally speaking, once the data is imported into the large table, it should not be necessary to further manipulate it within the table.

Zeroing large table data using the **Zero/Delete Table Data** feature is not allowed. The option is suppressed if the currently selected table uses the LargeTable index scheme. However, deleting large table data using Zero/Delete Table Data is allowed.

The data in large tables cannot be audited. The table property **Audited** defaults to **False** and cannot be changed. Generally speaking, large tables should have limited auditing needs because the data is imported from other sources and not manipulated within the large table itself.

Additionally, the RecordModifiedBy and RecordModifiedDTM columns are not available for large tables. Instead, a single column of ActivityID is available to record the associated audit activity. However, currently there is no easy way to look up an audit activity by ID number.

## Handling time-based data in tables

Time is not usually defined as a dimension in Axiom Software. Instead, time-based data is stored in series of sequential columns in data tables.

For example, if you want to calculate twelve months of budget data, that data is stored in twelve columns named something like NYB1 through NYB12 (where "NYB" is a convention for "Next Year's Budget"). The columns can be literally named that, or you can name the columns something like M1 through M12, and use alias names to point to the appropriate columns for next year's budget. Alias

names allow for some additional flexibility—for example, when you move to a new year of planning, you can edit the aliases so that the P1 through P12 columns are now associated with the alias names for CYB (Current Year's Budget), and the NYB alias names now point to a different table.

Time is handled this way so that data is stored in exactly the same way that it is presented in plan files within the system. Plan files are typically structured so that each column represents a month (or other periodicity) of plan data. Using the same structure for tables optimizes the save-to-database process for plan files, and also provides the following benefits:

- Data is optimized for reporting, since time is usually presented in columns in variance reports. Since the general structure is the same, Axiom Software does not have to reorganize the data to bring it into the report, which could slow performance.

  Although this is the most common way to present data in reports, it is by no means the only way. You can present dimensions side-by-side in columns (such as one column for each department or region), including stacked variance calculations for each dimension element, and you can present time vertically. Any data can be presented in any format.

- All data related to a specific record is stored on the same row. Axiom Software can save string values in certain columns and decimal values in others—all related to one line of “the plan.” If time was a dimension instead of sequential columns, then all data would be in one tall and thin table with multiple rows per account/department combination. All information related to one line of a plan would be scattered in the table and dissociated.

- Because data is stored in the same way that it is presented in Axiom files, it is easier for finance users to understand the database structure and view data directly when needed.

- Column aliases can be used so that the source column for a relative time period (like "last year's actuals") can change over time, allowing for dynamic reporting.

- Storing time data in columns makes it easy to "lock down" time-specific data as needed—the column can be set to read-only.

Columns that hold time-based data are created in the same way as other database columns; however, you can use the **Add Multiple Columns** feature to create all of the columns at once. To define the columns as a sequential "set" of data, you define a sequence for the columns. This allows you to create calculated fields for the sequence, such as YTD (year-to-date) and TOT (total).

Each table can have one or multiple "sets" of time-based data columns. For example, you can have a table that contains three years of data, with three sets of sequential columns representing each year. Or you can have one table for each year, with each table containing one set of sequential columns.

## Populating tables with data

The following features are available to populate tables with data.

- **Import utilities**
  You can import data from a source file or a database into an Axiom Software table. This method is typically used to import historical data into data tables.
- **Open Table in Spreadsheet**
  You can open a table in a spreadsheet, and add rows of data to the table. This method is typically used to populate reference tables, although imports can be used for this purpose as well.
- **Save-to-database processes defined in Axiom files (Save Type 1)**
  You can save data to a table from an Axiom file, such as a plan file or a report. This method is typically used to save planning data to data tables, or to perform data utilities such as allocations.
- **Copy Table Data**
  Using this utility, you can copy data from any data or reference table to another data or reference table.

The following exceptions and limitations apply:

- Picklist tables can also be populated using the picklist editor in the Web Client Table Manager.
- Tables using the Large Table index scheme can only be populated using import utilities or Copy Table Data.
- Document reference tables can only be populated by saving data from their source Axiom files, using Save Type 3.

## Table and column naming requirements

When naming tables and table columns (including alias names), follow these requirements:

- Table and column names can use standard ASCII alphanumeric characters (a-z, 0-9) and the underscore character. No spaces and no other special characters are allowed.
- The first character in a table or column name cannot be a number or an underscore. For example, you can have a table named `Plan_09` but you cannot have a table named `09_Plan`.
- Columns cannot be named `Col#`, where # is a number, such as Col1, Col2, etc. Other column name constructions with text and numbers are allowed—for example, BUD1 is valid.
- Database reserved words should be avoided whenever possible.
- Table names and column names are limited to 50 characters by default. Remember that table and column names will be used in Axiom queries and Axiom functions to query data, so they should be short yet descriptive.

  **NOTE:** Systems can be configured to allow longer table and column names if desired. See System Configuration Settings.

Table and column naming requirements are enforced when a new table or column is saved.

# Managing tables

This section explains how to create, edit, and delete tables in Axiom Software. You can create new tables from scratch, or you can clone existing tables.

Tables are managed using the Table Library within Axiom Explorer. You can view the list of tables by folder, by table, or by table type. By default, tables are displayed by folder. If you want to change the display, right-click the Table Library folder and then use the **View** menu to select the desired view.

**NOTE:** Document reference tables can only be created and edited from within an Axiom file. If you want to create a document reference table, you must configure an Axiom file to use Save Type 3. For more information, see the *Axiom File Setup Guide*.

## Creating a table

You can create new tables as needed. Only administrators and users with the **Administer Tables** permission can create tables in the Desktop Client.

Using the **Create Table** wizard in the Desktop Client, you can create the following kinds of tables:

- Data tables
- Reference tables (including picklist tables but excluding KPI tables)

Other kinds of tables are created differently. Document reference tables can only be created using Save Type 3 in Axiom spreadsheet files. KPI tables can only be created using the Web Client Table Manager. For more information on the different kinds of tables, see About table classifications.

**NOTE:** Although it is possible to create picklist tables in the Desktop Client, the Web Client Table Manager is the intended area for picklist table management. Additionally, users with the **Administer Picklists** security permission can only use the Web Client Table Manager to create picklist tables, edit picklist table properties, and delete picklists.

**To create a table:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Add Table**.

   The **Create Table** wizard opens.

   **NOTE:** If you are using an Axiom packaged product, you can access this feature from the **Admin** tab. Click **System Browser** to open Axiom Explorer, then right-click the **Table Library** and select **New > Table**.

2. Define the general characteristics of the table, and then click **Next**.

| Item | Description |
| --- | --- |
| Table Name | Type a name for the table.<br><br>This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. The table name must be unique. See Table and column naming requirements. |
| Table Type | Select a table type. You can select from existing table types, or click **New Type** to create a new table type. If you select **None**, then the table will have no table type.<br><br>Table types are used to create groups of related tables. You can then define security settings for the group instead of by individual table.<br><br>The new table must contain any required columns for the table type. For more information, see Table Types. |
| Folder | Specify the folder for the table, to determine where the table displays in the Table Library.<br><br>You can type the folder name (for example: GL Data), or you can click the **Set Folder** button to view the Table Library structure and select a folder.<br><br>For more information on the Table Library, see Managing the Table Library. |
| Table Classification | Select a table classification:<br><br>• **Reference Table**: Use for tables that define reference information, such as dimension tables or mapping tables.<br>• **Picklist Table**: Use for simple tables that hold lists of values from which users can make selections.<br>• **Data Table**: Use for tables that will hold historical data or plan data.<br><br>If you have assigned the table to a table type, then the table classification is automatically selected based on the table type and cannot be changed.<br><br>For more information on table classes, see About table classifications. |

| Item | Description |
| --- | --- |
| Add description column | If the table is a reference table, then by default Axiom Software will create a column to hold descriptions for each key code. This column is named Description and is a string column with a length of 50. (You can change these defaults later if desired.)<br><br>If you do not want a description column, clear this check box. However, it is recommended to have a description column to display descriptions for the items in the table throughout the software.<br><br>This option does not display if the table is a data table or a picklist table. (Picklist tables do have a Description column, but it is required and has a different size.) |

3. If the selected table type has required columns, a dialog opens to display a list of those columns. Click **OK** to add the required columns and continue creating the table.

   You can click **Cancel**, but you cannot continue creating the table unless the required columns are added.

4. Define the following table properties, and then click **Next**.

| Item | Description |
| --- | --- |
| Table Description | Optional. Type a description for the table. |
| Table Properties | Optional. In the **Table Properties** grid, edit table settings as desired. If you want to use the default settings, you can skip this step. For more information on the available settings, see Table properties. |

5. Define columns for the table, and then click **Next**.

   If you added required columns for the table type in step 3, then those columns are listed here. The table must contain all required columns for the specified table type. If the table is missing a required column, then when you save the table you will be prompted to add it.

   Otherwise, the Columns screen starts out as follows, depending on the table classification:

   - **Data**: One "starter" column is listed, named KeyColumn. You can rename and edit this column as needed
   - **Reference**: Two columns are listed—the key column and the description column. By default, the key column has the same name as the table (for example, the DEPT table has a key column named DEPT). You can rename and edit these columns as needed.
   - **Picklist**: The three required columns of Code, Value, and Description are present.

You can add more columns as needed. For more information on creating table columns, see Managing table columns.

**NOTE:** You can click **Finish** at this point and in any of the following screens to create the table. All of the remaining screens of the wizard are optional.

6. Optional. Define aliases for table columns.

   When querying information from the database, you can use alias names instead of column names. For more information, see Column Aliases.

7. Optional. Define column sequences for the table.

   Column sequences are used to group and order related columns. For more information, see Column Sequences.

8. Optional. Define calculated fields for the table.

   Calculated fields can be used to automatically calculate values such as Total and YTD. For more information, see Calculated Fields.

9. Optional. Configure data conversions for the table. Only applies to Data tables.

   Data conversions can be used to convert data on-the-fly, such as for currency conversions. For more information, see Data Conversions.

   **NOTE:** Although the Data Conversion screen is present in the Create Table wizard, the table must be saved before data conversion can be fully configured. If you want to use data conversion with this table, you should finish creating the table, then edit the table later to add the conversion settings.

10. Click **Finish**.

After you create a new table, you may need to configure security access for the table. If the table belongs to a table type, then it automatically inherits the security permissions set for the table type. Otherwise, no user has security permissions to access the new table.

**NOTE:** If a user with Administer Tables permission creates a new table and does not assign it to a table type, then that user will not have access to data in the newly created table. An administrator must configure access to the table in security.

## Cloning a table

When you clone a table, all of the existing table settings and structure can be copied to a new table. You also have the option to do the following as part of the process:

- Copy table data
- Copy table security

Only administrators and users with the **Administer Tables** permission can clone a table.

**NOTE:** Document reference tables cannot be cloned, because they are controlled by the source file. Instead, you could copy the source file, modify the source file as needed (including specifying a new table name), and then execute a save-to-database to create the new table.

**To clone a table:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Manage Tables**.

   The **Axiom Explorer** dialog opens, with the focus on the Table Library.

   **NOTE:** If you are using an Axiom packaged product, you can access this feature from the **Admin** tab. Click **System Browser** to open Axiom Explorer, then navigate to the **Table Library**.

2. In the **Table Library** folder, navigate to the table that you want to clone, then right-click that table and click **Clone**.

   You can view the list of tables by folder, by table, or by table type. By default, tables are displayed by folder. If you want to change the display, right-click the Table Library folder and then use the **View** menu to select the desired view.

   The **Clone Table** wizard opens, with all of the settings inherited from the cloned table.

3. On the first screen of the wizard, do the following:

   - In the **Table Name** box, define a name for the new table. By default, the table name is "Copyof*ClonedTable*".
   - Decide whether you want to copy the current table's security settings to the new table. By default, security settings will be copied to the new table. If you do not want to copy security, then clear the check box for **Copy Table Security** at the bottom of the screen.

     Keep in mind that if the new table does not belong to a table type, then no non-admin users will be able to access the table once it is copied, unless you copy the current table's security. If you do not want to copy security, then you may need to edit security settings after creating the table.

4. Change any other table properties as desired for the new table. You can click **Previous** and **Next** to move through the wizard screens. When you are finished editing settings, click **Finish**.

   A confirmation dialog displays the name of the new table and the columns to be created.

5. If you want to copy data from the original table to the new table, then at the bottom of the confirmation dialog, select **Open Copy Data** dialog. Click **OK** to continue.

   The new table is created. If you did not select to copy data, the process is complete. If you did select to copy data, then the **Copy Table Data** dialog opens, and is automatically populated with

the settings to copy data from the cloned table to the new table.

6. Edit any of the copy data settings as desired, and then click **Copy**.

## Editing table properties

You can edit existing tables to change certain table properties, and to add, edit, or delete columns and other table features (such as aliases, sequences, and calculated fields). The following restrictions apply:

- The table name cannot be edited. If you really need to change a table name, you can clone the table and give it a new name, and copy the table data to the new table.
- The table classification can only be changed if the table does not belong to a table type, and the structure of the table is compatible with the new classification. For example, you cannot change a data table to a reference table if it has multiple key columns. Additionally, it is not possible to change the classification of KPI tables.
- Once a table has been created, it is not possible to change the index scheme to Large Table (or from Large Table to something else).
- If you want to change the table type of a table to a different table type, the table must be compatible with the new table type. You can always change a table type to **None**.

**NOTES:**

- Table current periods are not set in the table properties. If you want to set the current period for an individual table, see Setting the current period for a table.
- Table properties cannot be edited for document reference tables. Document reference tables are controlled by their source documents.

The ability to edit table properties is limited to administrators and to users with the following security permissions: the global **Administer Tables** permission or the **Allow changing table structure** permission for individual tables. Users with the Administer Picklists permission can edit picklist table properties, but only when using the Web Client Table Manager.

**To edit table properties:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Manage Tables**.

   The **Axiom Explorer** dialog opens, with the focus on the Table Library.

   **NOTE:** If you are using an Axiom packaged product, you can access this feature from the **Admin** tab. Click **System Browser** to open Axiom Explorer, then navigate to the **Table Library**.

2. In the **Table Library** folder, navigate to the table that you want to edit.

You can view the list of tables by folder, by table, or by table type. By default, tables are displayed by folder. If you want to change the display, right-click the Table Library folder and then use the **View** menu to select the desired view.

3. Right-click the table and then click **Edit table structure**.
4. In the **Edit Table** dialog, edit the table as desired.
    - On the **Table Properties** tab, you can edit the description and several optional properties. See Table properties.
    - On the **Columns** tab, you can add or delete columns, and edit column properties. See Managing table columns.
    - On the **Aliases** tab, you can add, edit, or delete aliases. See Managing column aliases.
    - On the **Sequences** tab, you can add, edit, or delete sequences. See Managing column sequences.
    - On the **Calculated Fields** tab, you can add, edit, or delete calculated fields. See Managing calculated fields.
    - On the **Data Conversion** tab, you can enable and configure data conversion, or disable conversion. See Configuring conversions for a table.
5. Click **Save**.

**TIP:** You can also access the **Edit Table** dialog through the Table Library listed on the **Tables** menu. Navigate to the table that you want to edit, then choose **Edit table structure**.

## Deleting a table

Deleting a table removes that table and its data from the database. This action cannot be undone.

**NOTES:**

- You cannot delete a table if another table is dependent on that table. If the table is a lookup source for another table, you must first edit the other table to point to a different lookup source.
- You can delete a document reference table (Save Type 3) if it is no longer needed. However, unless the source file is also deleted or configured differently, the table will be recreated the next time a save-to-database is executed in the file.
- If the table is the target of a drill-through definition, that drill-through definition will also be deleted when the table is deleted. The software does not provide a warning before performing this deletion. If the drill-through definition should not be deleted, then you must modify the definition to point to a different table before deleting the original table.

Only administrators and users with the **Administer Tables** permission can delete a table in the Desktop Client. Users with the Administer Picklist permission can delete picklist tables, but only by using the Web Client Table Manager.

**To delete a table:**

1. On the Axiom tab, in the Administration group, click **Tables > Table Administration > Manage Tables**.

   The **Axiom Explorer** dialog opens, with the focus on the Table Library.

   **NOTE:** If you are using an Axiom packaged product, you can access this feature from the **Admin** tab. Click **System Browser** to open Axiom Explorer, then navigate to the **Table Library**.

2. In the **Table Library** folder, navigate to the table that you want to delete, then right-click that table and click **Delete**.

You are prompted to confirm that you want to delete the table.

## Table properties

This topic is a reference for all the properties that can be defined for tables. The Editable column indicates whether that property is editable after the table has been created. Table properties are defined on the **General** tab and the **Table Properties** tab of the **Edit Table** dialog.

For information on column settings, see Column properties.

| Property | Editable | Description |
|---|---|---|
| Table Description | Yes | The description of the table. The table description displays in wizards and other dialogs to help users understand what type of data the table holds. |
| Table Name | No | The name of the table. This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. The table name must be unique.<br><br>**NOTE:** The table name can only be changed by using Save Type 4. If you change the table name of an existing table, keep in mind that this name change is not applied throughout the system—any references to the old table name will now be invalid. Generally speaking, this should only be done to a new table that has not yet been referenced throughout the system. For more information, see Managing tables using Save Type 4. |

| Property | Editable | Description |
|---|---|---|
| Preferred Name | Sometimes | This setting only applies to systems with installed product packages, and is only present if **Is Variable Table** is set to True. It defines an alternate display name for the table. This setting is intended to be configured during system implementation as needed. You should not change this setting afterward without consulting Kaufman Hall Software Support. |
| Table Type | Yes | The table type assigned to the table, for security filter purposes. This can be a defined table type or **None**.<br><br>You can change the table type from **None** to a specific table type, and vice versa. However, if you want to change from one table type to another table type, the following restrictions apply:<br>• If the new table type has a different table classification, the table must be eligible to be converted to that classification.<br>• Any required columns for the new table type must be added to the table (if not already present). |
| Is Variable Table | No | This setting only applies to systems with installed product packages, and cannot be modified. This setting is controlled at the product level. |
| Folder | Yes | The virtual folder location where the table resides in the Table Library. |
| Classification | Sometimes | The class of table, either **Data** or **Reference**. This property is displayed on both the General tab and the Table Properties tab, but it can only be edited on the **General** tab.<br><br>**NOTE:** Picklist tables and KPI tables are reference tables; they are not separate classifications.<br><br>The table classification can only be changed if:<br>• The table is not assigned to a table type. If the table is assigned to a table type, then it must match the classification of the table type.<br>• The structure of the table is compatible with the new classification. For example, a data table cannot be changed to a reference table if it has more than one key column.<br>It is not possible to change an existing table to a picklist table or a KPI table. If you have an existing reference table that you want to convert to a picklist table or a KPI table, it is recommended to create a new table and then copy the relevant data. |

| Property | Editable | Description |
|---|---|---|
| Fixed | N/A | This table property refers to a deprecated feature. Enabling the setting has no impact in the system. |
| Read-Only Data | Yes | Specifies whether the table data is read-only. All tables default to **False**.<br><br>If this is set to **True**, then the data in the table cannot be changed by Axiom Software processes such as save-to-database or **Open Table in Spreadsheet**.<br><br>You can still edit table properties, change the table current period, and edit table aliases, sequences, hierarchies, and calculated fields. |
| IsPicklist | No | Specifies whether the table is a picklist table. This is set when the picklist table is created and cannot be changed.<br><br>**NOTE:** It is possible to convert a picklist table to a regular reference table by using the **General** tab of the **Edit Table** dialog to change the classification to **Reference Table**. This will convert **IsPicklist** to **False**. If the picklist table belongs to a picklist table type, you must remove it from the table type first. |
| Is KPI Table | No | Specifies whether the table is a KPI table. This is set when the table is created and cannot be changed. |
| Index Scheme | Sometimes | Specifies the index scheme used by the table. This should be left at **Default** unless you are advised to use a different scheme by your implementation consultant or by Kaufman Hall Software Support. For more information, see About table index schemes.<br><br>Once the index scheme is set for a table, it can only be changed as follows:<br>• Default can be changed to Hybrid, and vice versa. In this case, the new indexes will not be created until the next time the System Index Maintenance job is run by Scheduler. By default, the job runs nightly.<br>• Large Table can only be set when the table is first created. Afterward, it is not possible to change Default or Hybrid to Large Table, or vice versa. If necessary, you can clone the table and change the index scheme as part of the clone. |

| Property | Editable | Description |
|---|---|---|
| Audited | Yes | Specifies whether changes to the table data are tracked in the audit database. By default, this is set to **True**, which means data changes are tracked. Change this to **False** if you do not need to be able to audit the granular data changes to the table.<br><br>For more information on table auditing, see Auditing changes to table data.<br><br>**NOTE:** Auditing cannot be enabled for tables using the **Large Table** index scheme. |
| Current Period | No | Displays the current period for the table. If the table uses the system current period, that value is displayed. If the table has its own current period, that value is displayed.<br><br>The current period cannot be edited from the table properties. See Setting the current period for a table. |

### Table properties for document reference tables

Because document reference tables can only be managed within a spreadsheet using Save Type 3, it is not possible to directly edit the table properties. Many of the properties described above do not apply to document reference tables. For example, document reference tables cannot belong to table types, and do not have a current period. Most of the remaining properties are set for the document reference tables by default and cannot be changed.

The table name and folder for a document reference table are defined in the Save Type 3 settings within the source file. All tables created using Save Type 3 are automatically classified as document reference tables. Additionally, document reference tables are always audited, unless the in-memory table feature is enabled for your system. Auditing is automatically disabled on document reference tables if they are being stored in-memory.

# Managing table columns

When you are creating or editing a data or reference table, you can manage the columns for the table. Each table must have at least one key column.

The table's current columns are listed in the left-hand side of the **Edit Table** or **Create Table** dialog. When you select a column, its properties are listed on the right-hand side of the dialog.

## Adding a column to a table

Columns can be added to tables when the table is created, or later by editing the table structure. The ability to add new columns to a table is limited to administrators and to users with the following security

permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**NOTES:**

- If the table is a Picklist table, keep in mind that any additional columns beyond the required columns of Code, Value, and Description will not be visible or editable in the Web Client picklist editor.
- If the table uses the Large Table index scheme, columns can only be added to the table if the table does not contain any data. Once the table has data, the overall column structure is fixed.

**To add a column to a table:**

1. On the **Columns** tab of the **Edit Table** dialog, click **Add Column** ✚.

   If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

   The new column is added to the bottom of the column list.

2. In the column properties area on the right-hand side of the dialog, complete the following settings (at minimum):

   - In the **Column Name** box, enter a name for the column. This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. See Table and column naming requirements.
   - In the **Data Type** box, select a data type. If the data type is **String**, enter a maximum string length in the **Max String Length** field.
   - In the **Key Column** box, select **True** if you want this column to be a key column for the table. Otherwise, leave the default of **False**. See About key columns.
   - If the values in this column should be validated, complete the **Lookup Column** setting by selecting the name of the table and column that contains the master list of values. For example, if you are defining an ACCT column in a data table, the lookup column would be set to `ACCT.ACCT`. For more information, see About lookup columns.

     The lookup column must point to a key column in an existing reference table.

   For more information on all column properties, see Column properties.

3. To save the new column, click **Apply** (or **OK** if you are finished editing the table).

### Other options for adding table columns

The previous procedure details how to add a single column to a table. Alternatively, you can add columns using the following tools:

- Use **Add Multiple Columns** to add multiple related columns in a batch process—for example, to add a set of time-based data columns. See Adding multiple related columns.
- Use **Add Columns from External Table** to copy the column structure of another table. See Adding columns from an external table.

## Adding multiple related columns

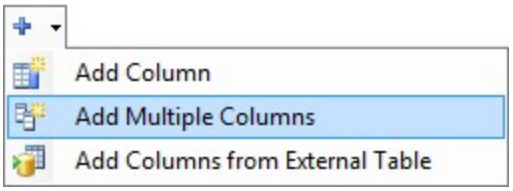
You can add multiple related columns at once by using the **Add Multiple Columns** feature. This feature is primarily intended to add a series of columns to hold time period data. For example, you might have columns NYB1 through NYB12 to hold twelve months of budget data. (The columns could be literally named that, or more likely you would name the columns something like P1-P12, and then define NYB1-NYB12 as alias names.)

To initially create a set of multiple related columns, the columns must share a naming convention and use the same column properties. Once the columns are created, they have no programmatic relationship to each other. If you want to change the name or other properties for the set of columns, they must be edited individually.

The ability to add columns to a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To add multiple columns:**

1. On the **Columns** tab of the **Edit Table** dialog, click the down arrow to the right of the **Add Column** button, and then select **Add Multiple Columns**.

   If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

   The **Add Multiple Columns** dialog displays.

2. Complete the following settings:

| Field | Description |
|---|---|
| Column Prefix | Enter a name to use for all columns in the set. This name is combined with a number to create each individual column. |
| Start Number | Enter the start number for the set of columns. |
| End Number | Enter the end number for the set of columns. |
| Data Type | Select a data type. |
| Max Length | If the data type is String, enter the maximum length of the string column. |
| Fixed | If this check box is selected, users cannot use **Open Table in Spreadsheet** to delete the columns or change column settings. |

For example, if the column prefix is P, the start number is 1, and the end number is 12, this process will create 12 columns named P1-P12.

3. Click **OK**.

   The new columns are added to the **Edit Table** (or **Create Table**) dialog.

4. Click **Apply** or **OK** to save your changes.

Once the columns have been created, you can edit them as normal on an individual-column basis. For example, you can add a description to each column.

## Adding columns from an external table

You can copy the column structure from a table in another database to a table in the current Axiom Software system. This feature works as follows:

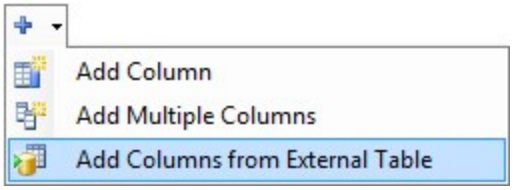
- You specify a connection string to the external table and the table name.
- Only the column structure is copied, not the column data.
- All columns from the external table are copied. You can delete any unneeded columns.

The ability to add columns to a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**NOTE:** This feature is not supported for use in the Axiom Software cloud service.

**To add columns from an external table:**

1. On the **Columns** tab of the **Edit Table** dialog, click the down arrow to the right of the **Add Columns** button, and then click **Add Columns from External Table**.




If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

The **Select External Table** dialog opens.

2. In the **Connection String** box, type the connection string to the external database.

   After entering the connection string, you can click **Test** to test whether Axiom Software can connect to the external database.

3. In the **Source Table Name** box, type the name of the external table.

   The table name must be qualified with the database name. View names can also be used. For example: AxiomEPM.dbo.vw_acct.

4. Click **OK**.

   You are returned to the **Edit Table** dialog (or the **Create Table** dialog). The columns from the external table have been added to the current table.

5. Modify the added columns as desired, or delete any unneeded columns.

6. Click **Apply** or **OK** to save your changes.

## Editing column properties

You can edit column properties at any time. All column properties are editable, however, certain restrictions apply:

- The data type of a column can only be changed in certain circumstances. The current contents of the column must be valid in the context of the new data type, or else the change cannot be made.
- The length of a string column can be increased, but it cannot be decreased.
- A column can be assigned to a new lookup column at any time, but only if all of its data is valid in the context of the new lookup column. If Axiom Software detects any invalid data, then the change will not be saved and you will be informed of the invalid data.
- If you change the name of a column, this may impact Axiom files and system settings that reference the column. However, if you are referencing the column via an alias name, then changing the column name will not affect the reference.
- If a table belongs to a table type, and a column is a required column for the table type, then only minor column properties can be edited. Generally speaking, the column properties must match the column properties defined for the required column in the table type.

- For picklist tables and KPI tables, only minor column properties can be edited for the required columns.
- If the column is controlled by an Axiom packaged product, the column is locked and cannot be edited (with the exception of changing the read-only status).

The ability to edit column properties is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To edit column properties:**

1. On the **Columns** tab of the **Edit Table** dialog, select the column that you want to edit.

   If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

   When you select a column in the left-hand pane, the properties for the column display in the right-hand pane.

2. Edit the column properties as desired.

   For more information on all column properties, see Column properties.

3. Click **Apply** or **OK** to save your changes.

## Ordering columns in a table

Column order affects the following functionality:

- **Table editing.** The order of columns in the list determines the order of columns when viewing or editing the table in **Open Table in Spreadsheet**.
- **Default sort order.** For data tables, the order of the key columns defines the default sort order.
- **Plan file lists.** For reference tables that are specified as plan code tables, the order of the columns determines the display order of the columns in file group dialogs.

The ability to order columns in a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To change the order of columns:**

1. In the **Columns** tab of the **Edit Table** dialog, select the column or columns that you want to move.

   If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

2. Click the **Up** or **Down** arrows until the column is at the desired location.

**NOTE:** Key columns are always listed before non-key columns. However, you can change the order of the key columns. Also, columns designated as description columns (property of **Describes Key**) always display after the key(s).

3. Click **Apply** or **OK** to save your changes.

## Deleting a column from a table

You can delete columns from a table with the following restrictions:

- Key columns can only be deleted if the table has no data. Otherwise, deleting a key column is prevented because it could result in non-unique data.
- Required columns for the table's table type cannot be deleted.
- Columns that belong to a sequence cannot be deleted. You must edit or delete the sequence first.
- If the table is a picklist table or a KPI table, the required columns cannot be deleted.
- If the table uses the Large Table index scheme and the table has data, then no existing columns can be deleted. The overall column structure is fixed once the table has data.
- If the column is controlled by an Axiom packaged product, the column is locked and cannot be deleted.

**NOTE:** If you delete a column that has an alias, the associated alias will also be deleted.

The ability to delete columns from a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To delete a column:**

1. On the **Columns** tab of the **Edit Table** dialog, select the column that you want to delete.

   If you are creating a new table, you can manage columns on the **Edit Columns** screen of the **Create Table** wizard.

2. Click **Delete Column**  .

   The column is removed from the list.

3. Click **Apply** or **OK** to save your changes.

## Column properties

This topic is a reference for all properties that can be defined for table columns in data tables and reference tables (including picklist and KPI tables). Almost all of these properties can be edited after column creation, unless the column is restricted due to being a table type required column or a

picklist/KPI required column. Column properties are defined on the **Columns** tab of the **Edit Table** dialog.

**NOTE:** If the column is controlled by an Axiom packaged product, then the column is locked and cannot be edited. The only column property that can be changed by the customer is **Read Only Data**.

| Property | Description |
|---|---|
| Column Name | The name of the column. This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. See Table and column naming requirements. |
| Preferred Name | This setting only applies to systems with installed product packages, and is only present if Is Variable Column is set to True. It defines an alternate display name for the column. This setting is intended to be configured during system implementation as needed. You should not change this setting afterward without consulting Kaufman Hall Software Support. |
| Description | Optional. A description of the column. |

| Property | Description |
| --- | --- |
| Data Type | Defines the type of data in the column. Available data types are:<br>• **Boolean**: Column contains TRUE/FALSE values.<br>• **Date**: Column contains string data that can be translated to valid SQL Date objects.<br>• **DateTime**: Column contains string data that can be translated to valid SQL DateTime objects.<br>• **Identity**: Column contains automatically-generated, sequential ID numbers.<br>• **Integer**: Column contains whole numbers only (no decimals). Equivalent to the SQL data type bigint, however, numbers should be limited to no more than 15 digits due to Excel limitations. It is recommended to use one of the smaller integer data types if appropriate for the intended data.<br>• **Integer 32**: Column contains whole numbers only (no decimals). Equivalent to the SQL data type int. Numbers are limited to 32 bits (-2,147,483,648 to 2,147,483,647).<br>• **Integer 16**: Column contains whole numbers only (no decimals). Equivalent to the SQL data type smallint. Numbers are limited to 16 bits (-32,768 to 32,767).<br>• **Numeric**: Column contains whole numbers or decimals. Holds up to 15 digits total.<br>• **String**: Column contains string (text) data. Numbers and dates can be saved as string data if appropriate, but then cannot be used in calculations.<br>Older systems may have a deprecated data type of **Decimal**. This data type exists for backward-compatibility only. New systems should use Numeric instead. If you have a Decimal column, any feature that applies to Numeric columns applies to Decimal columns.<br>For more information on identity columns, see About identity columns. |

| Property | Description |
| --- | --- |
| Numeric Type | Defines the type of numeric data held in the column, for formatting purposes when column values are displayed in certain areas of the software. Only applies to Integer columns (all types) and Numeric columns. Available options are:<br>• **Number** (default): Column values are treated as plain numbers.<br>• **Currency**: Column values are treated as currency values.<br>• **Percent**: Column values are treated as percentage values.<br>• **Date**: Column values are treated as date values.<br>• **DateTime**: Column values are treated as date time values.<br>**NOTE:** Date and DateTime are placeholder values for future features. Currently these options do not have any effect.<br><br>For more information where this formatting is applied and how each option is treated, see Using numeric type for column formatting. |
| Max String Length | Defines the maximum length of a string column in characters. Only applies to String columns.<br><br>The default length is 50. You should set the length to the smallest size necessary to accommodate the anticipated contents of the column. Setting the string length unnecessarily long can severely impact system performance. You can specify a length from 1 to 4000.<br><br>**NOTE:** It is possible to set the column to an unlimited length by leaving the **Max String Length** field blank. (In **Open Table in Spreadsheet**, this setting displays as **Unlimited**.) This should only be done if absolutely necessary. Make sure that you understand the potential ramifications of having an unlimited length column. Key columns cannot be set to unlimited length. |
| Unicode | Specifies whether the string column is Unicode-compliant. Only applies to String columns.<br>• By default, this is False, which means the column is limited to storing extended ASCII characters. This is equivalent to the SQL varchar data type. The database collation determines the specific set of valid characters.<br>• If True, then the column can store any Unicode data. This is equivalent to the SQL nvarchar data type.<br>If you do not need full Unicode support for a column, leaving this option disabled can save space in the database. |

| Property | Description |
| --- | --- |
| Key Column | Specifies whether the column is a key column for the table (**True**/**False**). See About key columns.<br><br>Key columns can only be the following data types: Identity, Integer (all types), String, or Date. Other data types are not valid for key columns.<br><br>Reference tables can only have one key column. Data tables can have as many key columns as needed. |
| Alternate Key Column | Specifies whether the column is an alternate key column for the table (**True**/**False**). See the discussion of alternate key columns in About key columns.<br><br>A table can have as many alternate key columns as needed. Alternate key columns can be validated columns (with an assigned lookup column) or not validated. The same data types that can be key columns are eligible to be alternate key columns.<br><br>**NOTE:** Once a column has been designated as an alternate key column, the System.IndexMaintenance job in Scheduler must be run before any data can be saved to the table. This job is used to create the constraint that enforces unique values in the column. By default this job is configured to run nightly. It can be run manually if you need to save data before the next scheduled execution. |

| Property | Description |
| --- | --- |
| Lookup Column | Optional. Specifies another column that defines the valid values for this column. The specified lookup column must be a key column in an existing reference table, and it must have the same data type as the current column. (Exception: If the lookup column is an Identity column, then the current column must be an Integer column.) If the column is a string column, the string length and Unicode status must also match.<br><br>If a database column has a lookup column, it is known as a *validated column*. See About lookup columns.<br><br>Any column can be a validated column, except for identity columns.<br><br>**NOTES:**<br>• You can assign a lookup column to a column that already contains data, but only if all of the data in the column is valid in relation to the lookup column (empty or "null" values are not valid). When you attempt to do this, Axiom Software will first validate the data. If the validation is successful, the lookup column assignment will be accepted. If the validation is not successful, a warning will display and you will need to correct the data before you can assign the lookup column.<br>• If you are creating a new table, and you want to create a "self-referencing" lookup (where the assigned lookup column is the key column of the current table), then you must save the table first and then go back and edit the column to add the lookup assignment. Until the table is saved, its key column will not display in the drop-down list. |

| Property | Description |
| --- | --- |
| Configured Partition Scheme | Optional. The table partition scheme assigned to the column. This property only applies to key columns. This property does not apply to identity key columns or string key columns with Unicode enabled.<br><br>The configured partition scheme may or may not be currently applied to the table. The **Current Partition Scheme** property indicates the currently applied partition scheme. When you save the column properties, Axiom Software will attempt to apply the configured partition scheme if the configured and current partition schemes do not match. If the table contains data that exceeds a configured limit (2 million rows by default), then Axiom Software cannot apply the partition scheme as part of saving the column properties, and will instead give you the option to process the partition change using Scheduler. If you say Yes, then the table will be processed using the System.RebuildPartitionTables Scheduler job. If you say No, then the configured partition scheme will be saved in the table properties but the partition scheme will not be applied to the table.<br><br>If a partition scheme is configured and applied to a table, then the table is partitioned within the database according to the boundary values defined for the scheme. Partitioning large tables may improve performance when querying the table and filtering the query based on the partition key column.<br><br>Table partitions are primarily defined and configured by product developers and system implementers, and delivered with product packages. If you have a custom system and believe that you may have a need for table partitions, please contact Kaufman Hall Software Support for more information. |
| Current Partition Scheme | The currently applied partition scheme for the table. This is a read-only, system-controlled property. If this value does not match the **Configured Partition Scheme**, then Axiom Software will attempt to apply the configured partition scheme as the current partition scheme when you save the column properties. |
| Is Cascade Delete | Specifies whether cascading deletion of data is supported for this table, based on the lookup relationship for this column. By default, this property is set to **False**. This property only applies when the column has an assigned lookup column, and that lookup column is an identity column.<br><br>If this property is set to **True**, then when data is deleted from the lookup table, any corresponding data will automatically be deleted from this table. For example, if identity record 47 is deleted from the lookup table, then all data relating to record 47 in this table will also be deleted.<br><br>If you want to use the Delete Plan Files command to delete on-demand plan files, then you should set this property to **True**. Otherwise, plan files cannot be deleted if linked tables contain data associated with those plan files. |

| Property | Description |
| --- | --- |
| Is Variable Column | This setting only applies to systems with installed product packages, and cannot be modified. This setting is controlled at the product level. |
| Hierarchy Display Name | Defines a "friendly" display name for the column. The display name may be helpful to clarify the contents of the column to users in various areas of the software.<br><br>The display name is used as follows:<br>• When the column is included in a hierarchy and displayed in the hierarchy view of the Quick Filter, Filter Wizard, or in the drill-down menu.<br>• When users are prompted to select a value for the column as part of creating a new plan file for an on-demand file group.<br>• When the column is included in a data set for use in the Data Explorer. |
| Is Filter Column | Specifies whether the column displays in the Filter Wizard. By default, this is **True**, which means the column is available to be selected in the Filter Wizard advanced view.<br><br>If **False**, then the column is hidden in the Filter Wizard. The column is still a valid filter column for existing or manually created filters, it simply does not display in the Filter Wizard. |
| Hyperlink Label | Specifies whether the column contains hyperlink data. Only applies to String columns.<br><br>When text is placed here, the column is identified as a hyperlink column and behaves as follows:<br>• The column can contain URL strings starting with http or https, or file paths in the Axiom file system.<br>• When this column is queried via Axiom query, the contents will be automatically converted to either a Hyperlink function (for URL strings) or to a GetDocument function (for Axiom file paths). The Hyperlink Label is used as the display text. |

| Property | Description |
| --- | --- |
| Describes Key | Specifies whether the column contains descriptions for the key column values. For example, most tables have a column named **Description** which holds a description for each key code.<br><br>If **True**, then the column is treated like a description column, which means that it will be included automatically in most dialogs as a description for the key. You can specify multiple columns as description columns if appropriate.<br><br>All other non-description columns in the table should be left at the default of **False**.<br><br>**NOTE:** If the table is a reference table, and no column has been designated as the description column, then a warning will display before the table is created. It is recommended to designate a description column for reference tables. |
| Column Classification | Specifies the classification of the columns for purposes of data presentation in the Data Explorer. Columns are automatically assigned a classification by default; you can override this classification if appropriate. Select one of the following:<br>• **Default**: The column uses the default classification for this type of column, either Dimension or Value.<br>• **Dimension**: The column is a dimension or grouping. These types of columns are typically used to group data in reports, or to display information about dimensional records.<br>• **Value**: The column holds data such as actuals, budget, payroll, or other data. This type of data is typically summed based on the dimension or grouping level of the report. |
| Fixed | This column property refers to a deprecated feature. Enabling the setting has no impact in the system. |
| Read Only Data | Specifies whether data in the column can be edited. By default, this is set to **False**, which means that the column data can be edited.<br><br>Key columns, and columns in document reference tables are always set to **False** and cannot be changed. Other columns can be set to **True**.<br><br>If a column is set to **True**, then the data in that column cannot be edited by Axiom Software processes such as **Copy Table Data**, **Open Table in Spreadsheet**, or save-to-database. If new rows are inserted, the column will have the default value.<br><br>Columns designated as read-only can be deleted. |

| Property | Description |
|---|---|
| Default Value | Defines the default value for the column. When a new column is created, this value is set to the default value defined for this column's data type in the system configuration settings. You can change this value for this column to any valid default value for the data type, including:<br>• A string literal in single quotation marks, such as '' or 'n/a' or '_'<br>• A number, such as 0 or 1<br>• A SQL function, such as GetDate() or GetUTCDate()—the result is applied as the default value<br>• Blank for null values<br>For more information, see About default column values. |

# Column design considerations

This section details various column design considerations relating to certain specialized data types and other column properties.

## About default column values

Each column in a table has a defined default value. This default value is applied in cases where no value is specified for the column.

The default value for a column is determined by the **Default Value** field in the column properties.

- When a column is created, the Default Value field is initially populated with the default value specified for the column's data type. These values are defined in the system configuration settings per data type, and can be changed for your installation.
- If desired, you can change the default value for a particular column by editing the Default Value field for that column.

### Configurable default values for columns by data type

The following system configuration settings control the default values for columns based on data type:

| Setting | Description |
|---|---|
| DefaultColumnValue_Boolean | Defines the default value for Boolean columns. By default, this is set to 0 (FALSE). You can change this to 1 for TRUE, or blank for null. However, it is not recommended to use null as the default for Boolean columns unless you have a particular use case that requires it. |
| DefaultColumnValue_Date | Defines the default value for Date columns. By default, this is blank (null). You can change this to any valid default value. |

| Setting | Description |
|---|---|
| DefaultColumnValue_DateTime | Defines the default value for DateTime columns. By default, this is blank (null). You can change this to any valid default value. |
| DefaultColumnValue_Number | Defines the default value for columns that hold numbers—Integer (all types) and Numeric columns. By default, this is set to 0. You can change this to any valid default value, including blank for null. However, it is not recommended to use null as the default for number-based columns unless you have a particular use case that requires it. |
| DefaultColumnValue_String | Defines the default value for String columns. By default, this is set to '' (an empty string). You can change this to any valid default value. Single quote marks are required to indicate the value is a string.<br><br>**NOTE:** Although it is possible to set the String column default to blank for null, Axiom Software does not differentiate between null values and empty string values within the spreadsheet environment. Therefore it is recommended to leave the default of empty string if you want the default to be empty (blank). |

If any of these values are changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### Setting the default value for a particular column

When you create a new column, the **Default Value** field is set to the default value defined for the appropriate data type. For example, if you create a new Integer column, the Default Value field is set to 0, which is the value set in the **DefaultColumnValue_Number** system configuration setting.

This inheritance from the system configuration settings only occurs when the column is created. If DefaultColumnValue_Number is later changed to a different value, this will not affect any existing columns, only new columns.

If desired, you can change the Default Value on a per column basis. For example, imagine that this new Integer column tracks the status of items based on an integer value from 1 to 5. If you want all new records to start out at 1 by default, then you can set the Default Value for this column to 1.

**NOTE:** For document reference tables, there is no means to "override" the system column value defaults. Document reference tables always use the default values as defined in the system configuration settings per data type.

### How the default value is applied to records in the table

The default value is applied in the following circumstances:

- When a new column is added to a table and that table has existing data, the default value is used for all existing records. If you are adding a new validated column to a table, then the default value of the validated column must be set to a valid value in the lookup column.
- When a new record is added to a table and the column is *not* included in the save, the default value is used for that column. If the omitted column is a validated column, then the default value of the validated column must be set to a valid value in the lookup column.
- When the column is included in a save (whether for a new record or an existing record) and the value to be saved is null (a blank cell in the spreadsheet), the default value is used for that column. If the null value is for a validated column, then the default value of the validated column must be set to a valid value in the lookup column.

  **NOTE:** If the column is a non-validated String column, then the blank cell is interpreted as '' (empty string) instead of null, so the empty string is saved.

### Allowing null values in a column

If you want to allow null values in a column, then the default value for the column must be null. This applies to all data types except String columns.

If the default value is not null, then there is no way to save null values to the column. For example, the system default for Boolean columns is 0 (False). If you save a blank value to a Boolean column that uses this default—for example, by clearing the cell in Open Table in Spreadsheet, or by using Save Type 1 with a blank cell for that column—this results in the default value of 0 (False) being applied to the column. However, if the default value for the column is blank (null), then saving the blank value to the column causes the default value of null to be applied to the column.

As mentioned previously, this behavior does not apply to String columns. If you save a blank value to a String column, the blank cell is interpreted as '' (empty string) instead of null, so the empty string is saved. If the String column uses the system default of empty string, then there is no effective difference between saving the empty string and applying the default value. But if the String column uses a different default value, such as '_' or "N/A", then the result of the save will be an empty string rather than applying the default.

**NOTES:**

- If a column is a key column, an alternate key column, or a validated column, then null values or empty string values cannot be saved to that column, regardless of its default value.
- Boolean columns should not have the default value set to null unless you need to track three Boolean states—True, False, and unset (null). If the column usage depends on every record having either a True or False value, then the default value must be either True (1) or False (0).
- Number columns should not have the default value set to null if the column is part of a sequence that is used by a calculated column. Null values in the sequence will cause the calculated column to not calculate as expected.
- If you change the default value of a DateTime column to something other than null, then you will not be able to save null values to that column—every record must have an assigned date. You should only do this if the field requires a date.
- If you are adding a new column to a table that contains many records, this process can take some time as the default value must be populated into every record. In some cases it may be worthwhile to speed up this process by adding the new column to the table with a temporary default of null, then go back afterward and set the default value to the intended default value. However if you do this, you must be careful to later update the table to replace these null values with the intended values for each record. For example, if the column is a Numeric column and you expect records to contain 0 by default, then these records should be updated to actually contain zeros instead of null values (otherwise calculations such as average will not work as expected).

## About hyperlink columns

You can associate a hyperlink with each record in a table, so that users can click the hyperlink to view supporting information about the record—such as an image, or a specifications sheet, or an invoice. For example, you might have a table where the key column is a part number. This table could contain a hyperlink column that links to the unique specifications sheet for each part.

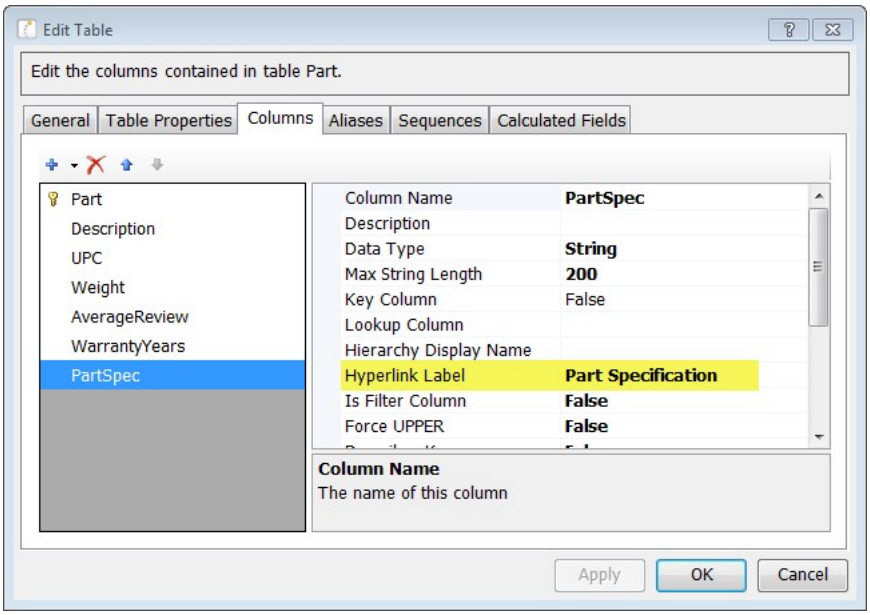
Hyperlink columns contain URL strings or file paths that Axiom Software automatically converts to clickable links when the column is queried using an Axiom query. Although it is possible to store this data in regular string columns and then manually convert it in the spreadsheet, using a designated hyperlink column has several advantages beyond simply eliminating this manual step. For example:

- Because the system performs the conversion, the data will be automatically converted to hyperlinks when the system builds the Axiom query—such as when performing drill-through drilling.
- When you set up an Axiom query to return data from a hyperlink column, you can optionally choose to override the default conversion and instead return the "raw" data. This provides greater flexibility in how you can use the data.

## Creating a hyperlink column

Hyperlink is not a specific data type for columns. Instead you can configure a string column to be a hyperlink column as follows:

- Set the **Max String Length** to an appropriate size for the hyperlink data that you intend to store.
- In the **Hyperlink Label** field, enter the label that you want to associate with the hyperlinks. This label can be merely descriptive, such as "Part specification," or it can include instruction to the end user, such as "Click to view part specification." When hyperlinks are generated by Axiom Software, the hyperlink label is used as the display text for the hyperlink.




*Example string column configured as a hyperlink column*

## Populating a hyperlink column

Once a column has been designated as a hyperlink column, the contents of the column must be in any of the following formats:

- URL strings starting with http or https. For example: `http://partserver/partspecs/part1.pdf`
- External file paths using the file protocol. For example: `file://\\partserver\partspecs\part1.pdf`
- File paths in the Axiom file system. For example: `\Axiom\Reports Library\PartSpecs\part1.pdf`

**NOTE:** External file paths are only supported for use in the Excel Client. Hyperlink functions to file paths do not work in the Windows Client.

If a value in the column does not match one of the valid formats, no validation error will occur when saving the data. However, when the data is queried it will not resolve into a valid link.

The hyperlink column can be populated via any of the normal methods, such as imports, Open Table in Spreadsheet, or Save Type 1. All three formats can be used in the same column if desired; each record will be treated individually.

### Querying a hyperlink column

When a hyperlink column is queried via an Axiom query, the contents of the column are automatically resolved as clickable hyperlinks. This occurs as follows:

- If the column contents start with file, http, or https, then a Hyperlink formula is returned.
- All other column contents are returned using a GetDocument formula.

In all cases, the column's defined Hyperlink Label text is used as the display text for the formula.

It is also possible to override the automatic conversion of the column contents and instead return the raw values as they are stored in the database. For more information on returning values from a hyperlink column using an Axiom query, see the *Axiom File Setup Guide*.

GetData functions always return the raw value stored in the column. There is no option to automatically convert the results. However, it is a simple matter to reference the GetData result in a Hyperlink function or in a GetDocument function.

## About identity columns

Columns that use the Identity data type contain automatically-generated, unique ID numbers. The primary use for this column type is to support the option to add plan files "on demand" for a file group. For example, when a user selects to create a plan file for an on-demand file group, a new record is added to the plan code table and a unique ID number is automatically generated. However, identity columns can be used for other purposes as needed.

The ID numbers for identity columns start at 1 and increment using whole numbers. You cannot specify a different starting value for the identity column, or any other ID format.

**NOTE:** The automatically-generated ID numbers are not necessarily sequential. There are many conditions that may cause numbers to be "skipped," such as encountering an error when attempting to create a new plan file, or restarting the SQL service. Skipped numbers should be expected during normal system use and do not indicate any issue with your system.

### Column linking and identity columns

Identity columns cannot have assigned lookup columns. Because the numbers in the identity column are automatically-generated, they cannot be validated against another list.

However, identity columns can serve as the lookup column for columns in other tables, if you want to limit the other column to only using values from the identity column. The column linked to the identity column must use the Integer data type.

### Editing identity columns

Identity columns are read-only, except when performing processes that cause an ID to be automatically generated. This means that new rows of data can be saved to the table, but you cannot manually specify an ID number or edit existing values in the identity column.

In the primary use case, values are added to the identity key column by use of the "add file" feature for on-demand file groups. However, you can also create new identity records by using Save Type 1 or Open Table in Spreadsheet. The behavior is different for the different save types:

- When using Save Type 1, you must leave the identity key column value blank and disable aggregation for the save. For each row in which the identity column is left blank, a new record will be added to the table, and the identity value will be automatically generated as normal.
- When using Open Table in Spreadsheet, you must enter a valid integer value into the identity key column, and that value must not match any existing values in the column. When the table is saved, the entered value is ignored and instead a new identity value will be automatically generated for the new record. You cannot leave the identity column blank in this environment, as an error will result.

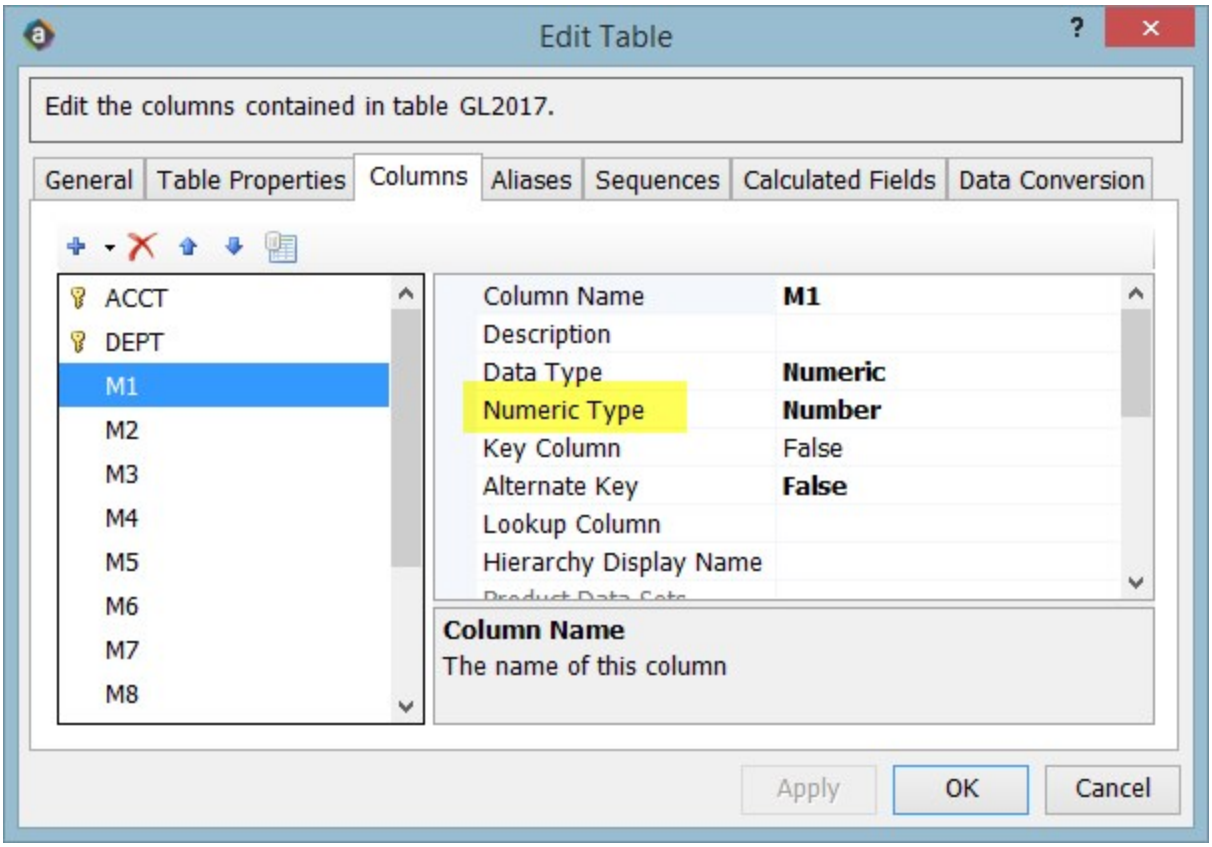
Data rows with identity columns can be deleted as normal using features such as Open Table in Spreadsheet or Zero/Delete Table Data. If an ID number is deleted from a table, it is not reused. However if the identity column is the key column of a plan code table for an on-demand file group, then you should use the **Delete Plan Files** command to delete the record instead. This feature deletes the record, the associated plan file, and any associated data in other tables.

**NOTE:** When using an identity column with an on-demand file group, the **Show On List** column can be used as an alternative to deleting an ID record. If **Show On List** is set to **False**, the associated plan file will no longer display in file group dialogs, but the record still remains in the table.

## Using numeric type for column formatting

You can use the **Numeric Type** property on a numeric column to specify what type of data that column holds. This property is used by Axiom Software to determine the formatting of the column values in certain areas of the software.

The numeric type is only available for the following column data types: Integer (all types) and Numeric. By default, the numeric type is set to **Number** for all applicable columns. You can change the setting for any column when creating or editing the table.




**NOTE:** Calculated fields are always treated as Currency type, regardless of the specified numeric type on the columns in the sequence for the calculated field.

## Numeric types

The following types are available:

| Type | Description | Formatting Example (1234) |
|---|---|---|
| Number | Column values are treated as plain numbers. If the data type includes decimals, two decimal places are included. | 1234<br>1234.00 |

| Type | Description | Formatting Example (1234) |
|---|---|---|
| Currency | Column values are treated as currency values. By default, currency values are displayed with a currency symbol and with a thousands separator. Decimals are not included by default.<br><br>By default, the symbol is determined by the current locale. However, the symbol can be explicitly defined by using the system configuration setting **SystemCurrencySymbol**. For information on changing system configuration settings, see System Configuration Settings. | $1,234 |
| Percent | Column values are treated as percentage values and displayed with a percent sign. Decimals are not included by default. | 123400% |
| Date | Column values are treated as date values. This is a placeholder for future functionality, intended to display valid date serial numbers as dates. Currently this option has no effect. | N/A |
| DateTime | Column values are treated as date/time values. This is a placeholder for future functionality, intended to display valid date serial numbers as date/time values. Currently this option has no effect. | N/A |

### Impacted areas of Axiom Software

Currently, the numeric type is only honored in a few areas of Axiom Software:

- File group dialogs that use the configured Display Columns for the file group, such as Open Plan Files and Process Plan Files
- The Process Status dialog (as configured by the Process Columns for the file group)
- The Plan File Directory page for file groups (as configured by the Web Configuration settings for the file group)
- Columns in the Data Grid component (for either Axiom forms or web reports)

These areas look up the numeric type of the configured columns and display in the appropriate format. When configuring a column to display in these areas, you can also optionally customize the formatting using an Excel format string.

**NOTE:** The numeric type has no effect on column values that are brought into a spreadsheet, such as via an Axiom query or a GetData function. The spreadsheet cell formatting determines the display of numbers in the spreadsheet.

## Storing dates in table columns

When storing dates in a table column, you should use either the **Date** or **DateTime** data type, depending on whether you require the additional time component.

- Date stores only the date value, such as 12/1/2010.
- DateTime stores both the date and the time, such as 12/1/2010 1:00AM.

It is not possible to omit the time component from a DateTime value in the database. If the date value does not contain a time when it is saved to the database, then the default time of 12:00 AM is applied to the value. Therefore if you want to store only the date, you should use the Date data type.

Generally speaking, date values should not be stored in string columns unless it is absolutely necessary (for example, if you require the column to contain a mix of date values and other strings). When the date is stored as a string, it will not be treated as a date in the database. This means, for example, that you cannot use filters such as DateColumn > 1/20/2015 because greater than / less than filters are not valid for string columns.

### Date display in a spreadsheet

When using an Axiom query or a GetData function to bring a date into a spreadsheet, the cell format determines how the value displays. Ideally, you should explicitly format the cell according to how you want the date to display. If you do not, then the spreadsheet may assume the format for you. For example:

- If you leave the cell format as General in the calc method for an Axiom query, the spreadsheet auto-formats Date and DateTime values using your default date format when the data is inserted into the query's data range.
- If you leave the cell format as General for a GetData function, this format is left as is, meaning the Date and DateTime values display as their numeric equivalents. In this case, the cell format must be changed to a date format if you want the value to display as a date.

## Considerations when storing numbers as strings

In general, it is not recommended to use a String column to hold numeric data. If data is truly numeric, then it should be stored using a Numeric column or an Integer column (all types) to ensure the data is treated appropriately when it is queried into spreadsheets and saved to the database from spreadsheets.

However, sometimes storing numeric data as strings is required when the numbers are codes such as 001. If 001 is saved as a number, the leading zeroes will be removed and only the number 1 will be saved. If 001 is saved as a string, then the leading zeroes can be retained as part of the string.

When storing numeric codes as strings, you should make sure that the data is alphanumeric instead of numeric only. If the source codes are numeric only, then when the data is brought into a spreadsheet it may be erroneously treated as a number instead of a string. For example, if you query the string value 001 in an Axiom query but the target cell is not formatted as text, then the spreadsheet will "guess" that

the value is a number and automatically convert it to 1. When viewing data this may be a minor inconvenience, but if the data is also saved to the database from the spreadsheet then the string value will now be saved back as 1.

Converting the numeric codes to alphanumeric codes eliminates this issue. You can adopt a convention such as prefixing all values with a particular letter, such as A001 instead of just 001. The presence of the letter means that the spreadsheet will always treat the value as a string.

# Managing picklist tables

Picklist tables define lists of values from which users can make selections. Using the Picklist tab of the Web Client Table Manager, you can create, edit, and delete picklist tables.

**To access picklist tables in the Web Client Table Manager:**

1. Go to the Table Manager area of the Web Client.

   **Example On-Premise URL**

   `http://ServerName/Axiom/TableManager`

   Where *ServerName* is the name of the Axiom Application Server, and Axiom is the default name of the virtual directory.

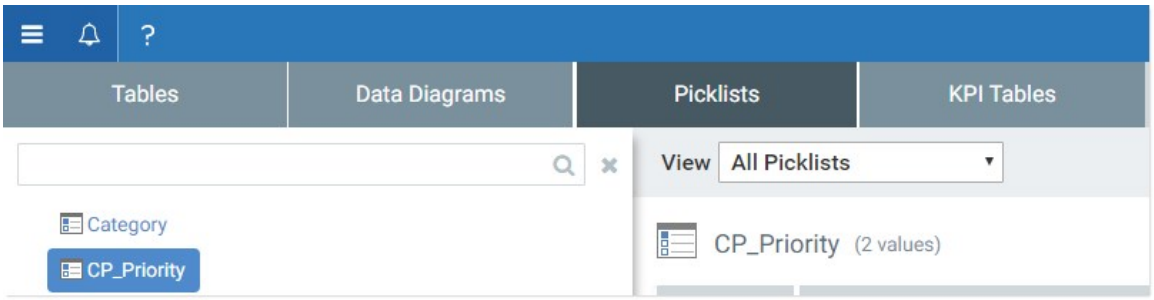
   **Example Cloud System URL**

   `https://CustomerName.axiom.cloud/TableManager`

   Where *CustomerName* is the name of your cloud service system.

   Alternatively, you can go to the Axiom Software launch page and click the **Table Manager** icon.

2. Select the **Picklists** tab.



## Browsing picklist tables

In the **Picklists** tab, the left-hand pane displays the list of picklist tables in your system that you have rights to see. You can view this list using the following **View** options:

- **All Picklists**: The list displays picklist tables in alphabetical order.
- **By Table Folders**: The list displays picklist tables in a treeview by table folders.
- **By Picklist Table Types**: The list displays tables in a treeview by picklist table types. Picklist tables that are not assigned to a table type are organized under a **(No Type)** node at the top of the treeview.

If you click on a table folder or a table type, then the tables in that folder or table type display in the right-hand pane. You can toggle this display between Icon View and Detail View.

You can also use the search box at the top of the left-hand pane to find a table by name. To clear the search results and return to the full list, click the X icon to the right of the search box.

You can select a picklist table in the left-hand pane to see its defined values in the right-hand pane, and to access the management options for the table. All users who have at least read-only access to the table can view the defined values. If defined in security, table filters are honored to limit the values displayed.

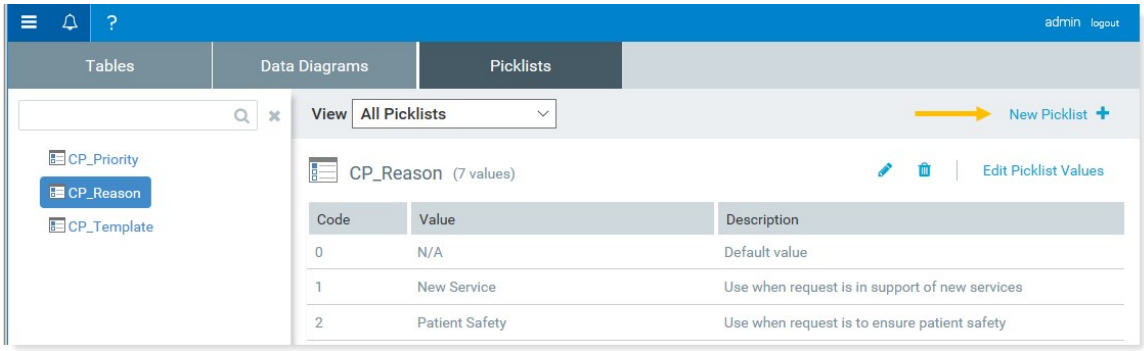
**NOTE:** The Web Client Table Manager can be accessed by any Axiom Software user. By default, the list of tables displayed in the Table Manager is limited to the tables where the user has at least read access. However, if the user is an administrator or has the **Administer Tables** permission, all tables are displayed.

## Creating a picklist table

You can create new picklist tables as needed. Only administrators and users with one of the following security permissions can create new picklists: **Administer Tables** or **Administer Picklists**.

**To create a new picklist:**

1. At the top right of the Picklists page, click **New Picklist**.



2. In the **Create New Picklist** dialog, complete the following fields:

| Item | Description |
|---|---|
| Picklist Table Name | Enter the name of the table. The table name can only contain letters, numbers, and underscores, and must start with a letter. Spaces are not allowed. |
| Description | Optional. Enter a description of the table. This description is visible when users view the table in the Picklists area of the Web Client Table Manager, and also in the Table Management area of the Desktop Client. |
| Table Folder | Select a folder for the table. You can select any folder that has already been created in the Desktop Client. For example, your system may have a dedicated folder for picklist tables, or picklist tables may be organized by product-specific folders.<br><br>It is not possible to create a new table folder from the Web Client. If you want to create a new folder, you must use the Desktop Client. Only administrators or users with the Administer Tables permission can create a new table folder. For more information, see Managing the Table Library. |
| Table Type | Optional. Select a table type that you want the table to belong to. You can select any picklist table types that have already been created in the Desktop Client.<br><br>Table types are useful to assign security permissions to picklist tables as a group. For example, if three different picklist tables belong to the same table type, you can grant access at the table type level instead of needing to grant access to all three tables individually.<br><br>It is not possible to create a new table type from the Web Client. If you want to create a new table type, you must use the Desktop Client. Only administrators or users with the Administer Tables permission can create a new table type. For more information, see Table Types.<br><br>**IMPORTANT:** If you are not an administrator, then you must select a table type where you have been granted security permissions, so that the newly created picklist table can inherit these security permissions from the table type. Otherwise, you cannot create the table because you will not have access to the table after it is created. |

3. Click **Create Picklist**.

The new picklist table is created, and the **Edit Picklist Values** dialog opens. For more information on editing picklist values, see the following section.

The picklist table is created using the three required picklist columns of Code, Value, and Description. It is not possible to delete or rename these required columns. It is also not possible to add any additional columns in the Web Client Table Manager.

## Editing picklist values

You can add, edit, and delete the values in a picklist table. Any user with read/write security permission to the table can edit the table values in the Web Client Table Manager.

All picklist tables contain the following columns. When editing picklist values, each record must have a defined code and value.

| Name | Column Properties | Description |
|---|---|---|
| Code | Integer<br>Key Column | Required. Defines a numeric code for each record. Each code must be a unique positive integer.<br>When users select items from a picklist, items are returned and stored using this numeric code. |
| Value | String (50)<br>Alternate Key Column | Required. Defines a text value for each record. Each text value must be unique.<br>Users select items from the picklist using these values. |
| Description | String (1023) | Optional. Defines a description or explanatory text for each record.<br>When users select items from a picklist, you can optionally display this text to help users choose the correct values. |

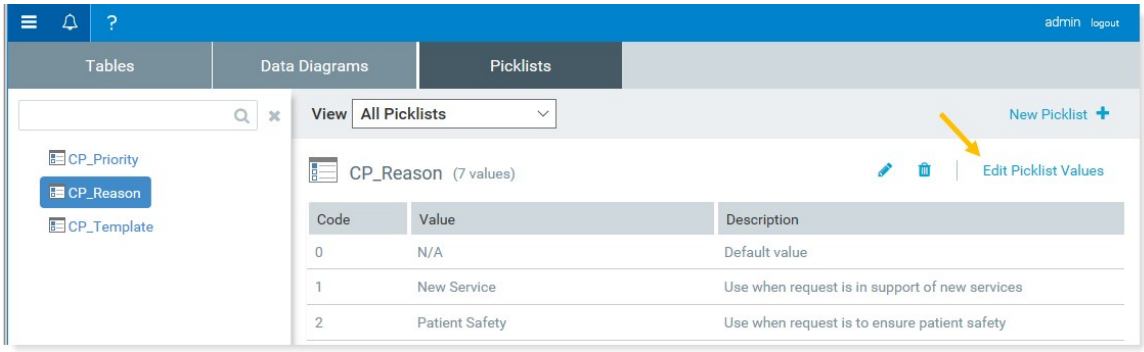
**NOTE:** If the picklist table has additional columns beyond the required columns, those columns do not display in the Web Client editor and cannot be edited. If you add new records to the table, the default value is used for any additional columns.

**To edit picklist values:**

1. On the Picklists page, select the picklist table in the left-hand pane.

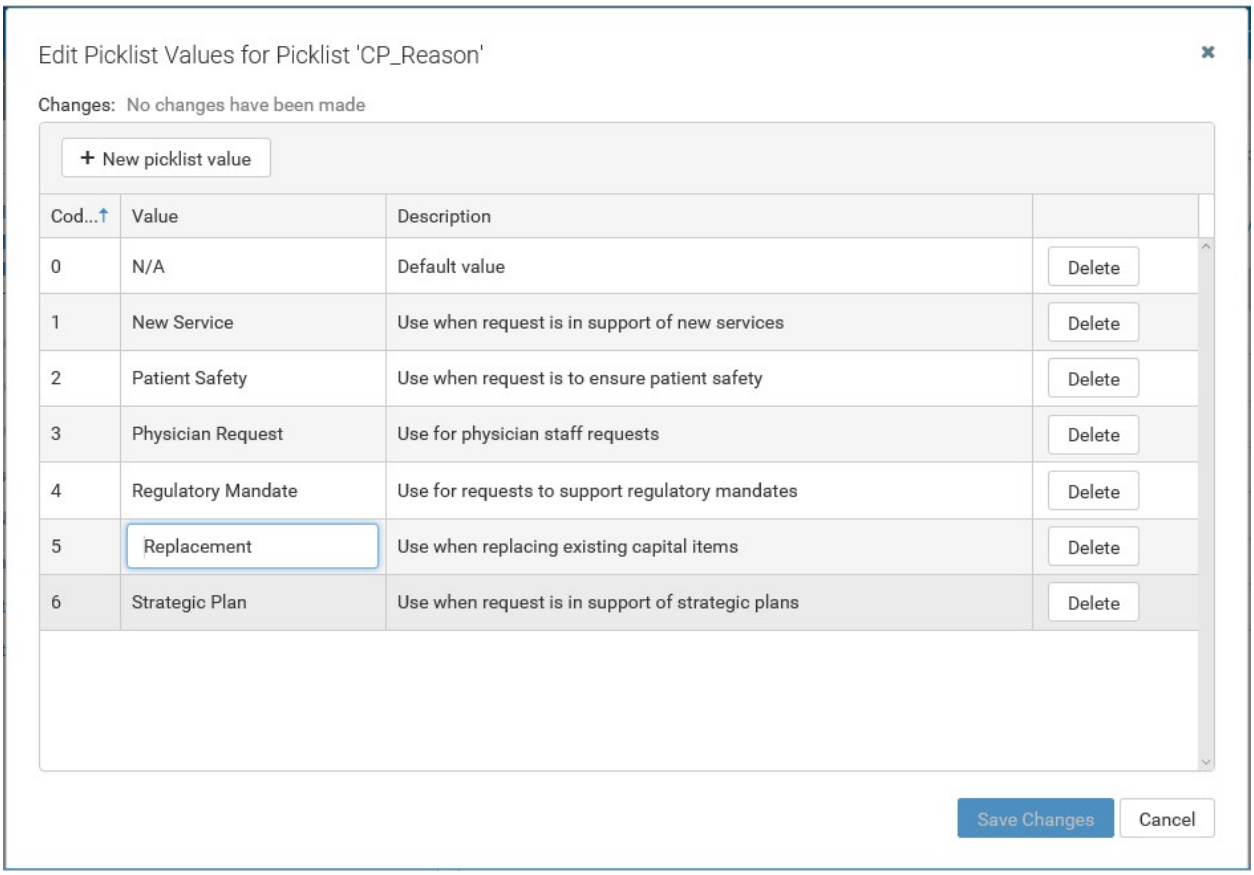
   You can change the **View** options or use the **Search** box to help find the picklist that you are looking for.

2. In the right-hand pane where the current picklist values display, click **Edit Picklist Values** at the top right of the page.




The **Edit Picklist Values** dialog opens. Using this dialog, you can edit the picklist values as needed. When you are finished, click **Save Changes**.

- **Adding new values:** To add a new value, click **New picklist value** at the top of the editor. This adds a new record to the table. You can then modify the values in each column as needed.
- **Editing existing values:** To edit an existing value, click on the cell that you want to modify. The cell becomes editable, and you can type your desired change. For codes, you can type a number or you can use the up and down arrows to make the current number larger or smaller.
- **Deleting existing values**: To delete an existing value, click the **Delete** button on the record that you want to delete.




*Example picklist value editor*

**NOTES:**

- All tables start with a row 0 that defines the default value of the table. You can modify the value and description of this row, but you cannot delete it or change the integer code. However, when setting up table columns to look up to this picklist table, you can optionally configure those columns to use a different value in the table as the default value.
- Code and value must be unique within the table. If a newly added or edited code or value matches an existing code or value, you will be informed of the duplication and prevented from saving the change.
- The integer codes do not have to be contiguous. You can set the code to any valid integer. When creating a new item in the table, by default the code is incremented by 1.
- If any validated columns in the system look up to this picklist table, then you will be prevented from deleting any record that is present in those validated columns. You will also be prevented from editing existing codes if the edit would effectively delete a code that is referenced in another table. For example, if code 5 is referenced in a validated column, then you cannot change code 5 to code 6 if that means the picklist table no longer contains a code 5.

## Editing picklist table properties

You can edit the main picklist table properties as needed. Only administrators and users with one of the following security permissions can edit picklist table properties: **Administer Tables**, **Administer Picklists**, or **Allow changing table structure** (table-specific permission).

**To edit picklist table properties:**

1. On the Picklists page, select the picklist table in the left-hand pane.

   You can change the **View** options or use the **Search** box to help find the picklist that you are looking for.

2. In the right-hand pane where the current picklist values display, click the pencil icon at the top of the page.

3. In the **Edit Picklist** dialog, edit any of the following properties as needed, then click **Save**.
   - Description
   - Table Folder
   - Table Type

   **NOTE:** If you change the assigned table type of the table, this may affect security permissions for the table. The table will no longer inherit security permissions from the original table type, and will now inherit permissions from the new table type.

If you need to modify a table property that is not displayed in the Web Client Table Manager, then you must use the Desktop Client to edit the table properties. For example, if you want to make the picklist table read-only, you must use the Desktop Client.

Similarly if you need to modify column properties for a picklist table, you must use the Desktop Client. The main column properties cannot be edited (such as column name, data type, etc.), but it is possible to make minor property edits such as defining a description or making a column read-only. You can also add additional columns, though those columns will not display in the Web Client.

## Deleting picklist tables

Deleting a picklist table removes that table and its data from the database. This action cannot be undone.

You can delete a picklist table if it is not referenced by other tables in the system. Only administrators and users with one of the following security permissions can delete picklist tables: **Administer Tables** or **Administer Picklists**.

**To delete a picklist table:**

1. On the Picklists page, select the picklist table in the left-hand pane.

   You can change the **View** options or use the **Search** box to help find the picklist that you are looking for.

2. In the right-hand pane where the current picklist values display, click the trashcan icon at the top of the page.
3. On the **Confirm Delete** prompt, click **OK**.

The picklist table is deleted.


# Additional Table Setup

This section details information about additional features for tables and table columns, including:

- Table types
- Column aliases
- Column sequences
- Calculated fields
- Data conversions

## Table Types

Table types are used to group related tables for security and reporting purposes. Using table types, you can:

- Define access rights in security for all tables in the table type
- Define query and sheet filters on Axiom files that apply to all tables in the table type

Table types are optional. You do not need to assign tables to a table type in order to apply security or query filters, but in certain cases it may be easier to do so.

### About table types

Tables can be assigned to a table type in order to group related tables and control access to the tables as a group. For each table type, users can be granted full access or filtered access to the tables in the table type.

You can create any number of table types to meet your needs. When you create a table type, you specify the table classification for the table type, such as Data or Reference. All tables assigned to the table type must be the same table classification.

Each table type can have a set of required columns that are common to each table that belongs to the table type—for example, ACCT and DEPT columns for GL actuals tables. Tables in the table type can have other key columns, and other non-key columns, in addition to the common columns for the table type. However, the table type filters in Security should only use the common key columns.

Table types are optional. You do not need to assign tables to a table type in order to apply security or query filters, but in certain cases it may be easier to do so.

## Required columns for table types

Table types can have any number of required columns, key or non-key. Table types can also have no required columns. However, if you want to define security filters for the table type, it is recommended to define required columns as follows:

- The table type should have at least one required key column. Table type filters should only use key columns, and the key columns must be common to all tables in the table type.
- The required key column must use a lookup column. For example, if the required key column is ACCT, it should look up to ACCT.ACCT. In order for the filter to work across all tables in the table type, it must be constructed using a common reference table.

If you define required columns for a table type, then any security filter for the table type must be based on those required columns. This ensures that any filter defined for the table type will be valid on all tables within the table type.

If you do not define required columns for a table type, then it is not possible for Axiom Software to validate the security filter for use with all tables in the table type. In this case, it is the system administrator's responsibility to verify that any security filter defined is valid on all tables within the table type.

## Security and table types

In security, you can define access rights at the table type level and/or the individual table level. If a table belongs to a table type, that table inherits the security permissions set at the table type level. In most cases, you will want to assign related tables to table types and set permissions at the table type level to streamline the process of defining security settings.

Because most data tables can be easily grouped into table types with shared key columns, it makes sense to use table types and define access rights at the table type level. For example, imagine that your GL actuals data is stored in individual tables by year (GL2017, GL2018, etc.). Most likely, you want to apply the same security filters to each table. Rather than defining the filters for each individual table, you can assign the tables to a GL table type, and then set the filters once. Additionally, as you create new GL actuals tables and assign them to the GL table type, the existing security will automatically be applied.

Reference tables (including picklist tables and KPI tables) can also be assigned to table types if desired, however, these tables do not have shared key columns. Therefore, if you want to set security permissions at the table type level for reference tables, this permission must be limited to full read and/or write access—filters cannot be used. If you need to use security filters with reference tables, then those tables should not belong to a table type.

Table types do not apply to document reference tables. By default, all users have access to the data in a document reference table, via the Everyone role. If you need to change this access, you would do so at the individual table level.

The following table summarizes how table types apply to different table classifications:

| Classification | Assign to Table Type? | Security |
|---|---|---|
| Data | Yes. In the majority of cases you will want to group related data tables into a table type, to streamline the process of defining security settings, including filters that use shared key columns. However, it is not required to assign data tables to a table type if it is not necessary for certain tables. | By default, users do not have access to the data in data tables. You must grant access to users and/or roles on a per table type and/or table basis. |
| Reference (including picklist and KPI tables) | Maybe. If you have multiple related reference tables where you can set full read and/or write access at the table type level, using table types can streamline the process of defining security settings for those tables. But if security filters are needed, then there is no benefit to using a table type with those tables, because the filters can only be set per table. | By default, users do not have access to the data in reference tables. You must grant access to users and/or roles on a per table type and/or table basis. |
| Document Reference | No. Document reference tables cannot be assigned to a table type. | By default, all users have access to the data in document reference tables, via the Everyone role. If you do not want all users to have access to this data, you must modify the Everyone role to remove access, and then grant access to the specific users and roles that need it. |

### Sheet filters and table types

Table types are also useful for defining sheet filters in Axiom files. Sheet filters can be used to filter all of the data queries on a particular sheet. Instead of setting a filter on each individual Axiom query or GetData function, you can set the filter at the sheet level.

**NOTE:** Table types can only be used with sheet filters when all tables in the table type share common validated key columns. Effectively this means that table type sheet filters are only for use with table types that group data tables.

Sheet filters can be set at the individual table level, or at the table type level. If you set a filter at the table type level, the filter applies to all tables in the table type. This is useful if you are:

- Querying data from several tables, such as to compare actuals to plan data, or to compare data from different years. In this case you can define one sheet filter that applies to all data tables in the query, assuming that all the tables belong to the same table type.
- Using aliases to query data. In this case you may be updating the aliases from year to year so that they point to different tables. If the tables belong to the same table type, then you can define one sheet filter that continues to apply as you change which data table the aliases point to.

For example, imagine that you have a report that queries data from the GL2017 and GL2018 tables, and you want to filter the report by a specific region using a sheet filter. If the tables do not belong to a table type, then you need to define the filter twice, once for each table. If the tables belong to the same table type, you can define the filter once, at the table type level.

## Managing table types

Table types can be created as needed when you are creating a new table, or you can create and manage table types using Axiom Explorer.

**To view and manage table types:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Manage Tables**.

   The **Axiom Explorer** dialog opens, with the focus on the Table Library.

   **NOTE:** If you are using an Axiom packaged product, you can access this feature from the **Admin** tab. Click **System Browser** to open Axiom Explorer, then navigate to the **Table Library**.

2. In the left-hand pane of Axiom Explorer, right-click the Table Library folder, and then select **View > By Table Type**.

The Table Library now displays the tables by table type instead of by folder. From here you can add, edit, and delete table types. You can also view the tables that belong to each table type.

### Adding a table type

New table types can be added as needed. Before creating a new table type, you should have a good idea of the following:

- The required columns for the table type (if any).
- The tables that will belong to the table type, and their classification.

Table types can be used to define security access for groups of tables. When a table type is first created, all non-admin users default to "no access" to the table type. After creating the table type and assigning tables to it, you should define access rights to the table data as necessary. You can set access rights at both the table type level and at the individual table level.

Only administrators and users with the **Administer Tables** permission can create a new table type.

**To add a new table type:**

1. With the Table Library in Table Type View, right-click the **Table Library** folder and then click **New Table Type**.

   The **New Table Type** dialog opens.

2. Complete the following table type settings, and then click **Next**.

| Item | Description |
|---|---|
| Table Type Name | Type a name for the table type.<br>For example, you might use the name "GL" for tables that hold general ledger history and budget data. |
| Table Classification | Select one of the following:<br>• **Reference Table**: Use for tables that define reference information, such as dimension tables or mapping tables.<br>• **Picklist Table**: Use for simple tables that hold lists of values from which users can make selections.<br>• **Data Table**: Use for tables that will hold historical data or plan data.<br>**NOTE:** KPI tables can belong to reference table types. There is not a separate table type class for KPI tables.<br>When you create a new table and assign it to this table type, the table classification will be automatically selected and cannot be changed. |
| Table Type Description | Type a description for the table type. |

3. Define the required columns for the table type as needed.

   All tables that are assigned to this table type must have these columns. You should define all key columns that are common to this table type, so that they can be used to define security filters.

   You can **Add** or **Delete** columns using the buttons on the left-hand pane. For each column, you can define a column name and the data type, specify whether the column is a key column, and specify a lookup column. For more information on these column properties, see Column properties.

**NOTES:**

- It is not required to define required columns for the table type, but it is recommended if you want to define security filters for the table type. See About table types.
- If the table type is for Picklist tables, you do not need to define the built-in columns such as Code and Value as required columns. This requirement is automatically enforced for picklist tables. In most cases, additional required columns are not used with picklist table types.

4. Click **OK** to create the table type.

The new table type is now available to be assigned to new tables.

**TIP:** You can also create a table type on-the-fly when you are creating a new table. To do so, click the **New Table Type** button on the first screen of the **Create Table** wizard.

## Editing a table type

Generally, once a table type has been defined, you should only make edits to it if no tables are assigned to it. The exception is the table type description, which can be changed at any time.

Although it is possible to add or remove required columns, or change the properties of required columns, this may cause issues with security filters if tables are already assigned to the table type. When you edit a table type, the changes are not applied to existing tables in the table type. If you later attempt to edit the structure of a table in the table type, an error will occur if Axiom Software detects that the table is out of sync with the table type.

For example, you may have a table type with required key columns of DEPT and ACCT. If you edit the table type to change the column name to Account, then any new tables added to the table type will have a key column named Account, but the existing tables in the table type will have a key column named ACCT. Security filters for the table type may no longer be valid because the tables in the table type no longer share common key columns. (In this particular example you could fix the issue by editing the existing tables to change the column name to Account, so that all of the tables in the table type once again share the common key column.)

Only administrators and users with the **Administer Tables** permission can edit a table type.

**To edit a table type:**

1. With the Table Library in Table Type View, locate the table type that you want to edit. Right-click the table type and then select **Edit Table Type**.
2. In the **Edit Table Type** dialog, make changes as desired.
3. Click **Apply** or **OK** to save your changes.

### Deleting a table type

A table type cannot be deleted if any tables are assigned to that table type. Only administrators and users with the **Administer Tables** permission can delete a table type.

**To delete a table type:**

1. With the Table Library in Table Type View, locate the table type that you want to delete. Right-click the table type and then select **Delete**.
2. At the confirmation prompt, click **Yes** to delete the table type.

The table type is deleted.

# Column Aliases

Column aliases allow you to define a standard set of names for reporting—such as CYA for Current Year's Actuals, and NYB for Next Year's Budget—and then dynamically point those names to whichever columns currently hold that data. Instead of updating all your reports, you simply update the aliases, and the reports now return the appropriate data.

## About column aliases

Each column in a table can be assigned one or more alias names. When you want to bring data from that column into an Axiom file, you can use the alias name instead of the literal column name.

Using aliases, you can point reports to new data columns without needing to change the column names in the file—instead, you can assign the alias name to a new column. Any column, including calculated fields, can be assigned to an alias name.

For example, you might want to use columns NYB1 through NYB12 of the Plan2019 table to hold next year's budget data. If you reference the columns directly in reports, then when you start a new year of planning, you would need to update the reports to point to the new columns in the new data table (Plan2020). Instead, you can define NYB1 through NYB12 as alias names for the Plan2019 table, and then reference the aliases in reports. When you are ready to start a new year of planning, you can move the aliases from the Plan2019 table to the Plan2020 table. Generally it is easier and less error prone to update the aliases for the data tables, rather than making sure you have updated every single column reference in reports.

An alias name can only be assigned to one column at a time in an Axiom Software system. For example, if the alias name of NYB1 is used in table Plan2019, you cannot also use the alias name NYB1 in any other table in that system. However, you can move the alias from one column / table to another.

The primary use case for aliases is in reports. Generally reports are designed to be used on an ongoing basis, year after year, while the templates, drivers, and plan files in a file group are designed for a specific year or cycle of planning. Therefore in most cases, you should use literal column names in file groups, so that they always point to the correct tables for their intended planning cycle.

### Evaluating column names as a table column or an alias

If all alias names are unique, and do not share names with any literal column names in any tables, then every time an alias name is used, it will point to the source column for the alias. This is the recommended way to set up alias and column names. For example, if you know you plan to set up NYB1-NYB12 as alias names, then you should not use those names for any literal database columns. Instead, the database columns can be named something like P1-P12 (periods) or M1-M12 (months).

However, Axiom Software does not enforce unique naming between aliases and columns. If there is duplication between literal column names and alias names, Axiom Software determines which column to use as follows:

1. If the name exists as a literal column in the target table, then that table column is used instead of the alias. For example, if the primary table for an Axiom query is Plan2019, and that table contains a column named NYB1, and an alias is defined in the system that is also named NYB1, Axiom Software will use Plan2019.NYB1 instead of the alias.
2. If the name does not exist as a literal column in the target table, but it does exist as a literal column in a lookup reference table, then that table column is used instead of the alias.
3. If 1 and 2 do not apply, then the alias column is used.

## Managing column aliases

Each column in a table (including calculated fields) can be assigned a column alias, which can be used instead of the column name in data queries and save-to-database operations for Axiom files. You can then easily change the query or save-to-database, by editing the alias to point to a different column.

You can create, edit, and delete aliases when creating or editing a table. Column aliases can be used with reference tables and data tables.

**NOTE:** Alternatively, you can use Save Type 4 to create and delete aliases via a spreadsheet. This may be an easier method if you have many aliases to maintain, especially if you are "rolling over" aliases to point to different columns and tables when starting a new planning cycle. For more information, see Managing column aliases using Save Type 4.

### Adding an alias

Aliases can be added to a table at any time. The ability to add aliases to a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**IMPORTANT:** It is not recommended to duplicate alias names and literal column names. For example, if you know you are going to create alias names NYB1-NYB12, do not use those names for literal columns in a data table. Instead, name the literal columns something like P1-12 or M1-12 (for periods or months). If aliases and table columns share the same name, and an entry could be resolved as either the alias or the table column, Axiom Software gives preference to the table column.

**To add an alias to a table:**

1. From the **Aliases** tab of the **Edit Table** dialog, click **Add Alias**  .

   If you are creating a new table, you can manage aliases on the **Manage Column Aliases** screen of the **Create Table** wizard.

   The **New Alias** dialog opens.

2. In the **Alias Name** box, type a name for the alias.

   This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. See Table and column naming requirements.

   Each alias name must be unique for the Axiom Software system.

3. From the **Column** list, select the column to assign to the alias.

   When this alias name is used in an Axiom function or an Axiom query, it will return the data from the selected column.

4. If desired, in the **Description** box, type a description.

5. Click **OK** to close the **New Alias** dialog.

6. To save the new alias, click **Apply** (or **OK** if you are finished editing the table).

## Editing an alias

Alias properties can be edited at any time. For example, you may need to edit an alias to point to a different column as part of your planning and reporting cycle, so that reports that use the alias will bring in data from the new column.

**NOTE:** When editing an alias using the **Edit Table** dialog, you can only assign it to columns within the same table. If you want to assign the alias to a column in a different table, you can delete the alias and then re-create it in the new table. If you are "rolling over" data for a new planning cycle, and you have many aliases that you need to move to different table columns, it will most likely be easier to use Save Type 4 instead. See Managing column aliases using Save Type 4.

The ability to edit aliases is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To edit an alias:**

1. On the **Aliases** tab of the **Edit Table** dialog, select the alias that you want to edit.

   If you are creating a new table, you can manage aliases on the **Manage Column Aliases** screen of the **Create Table** wizard.

2. Click **Edit Alias**  .

3. In the **Edit Alias** dialog, edit the alias properties as desired, and then click **OK**.

4. Click **Apply** or **OK** to save your changes.

### Deleting an alias

You can delete an alias if it is no longer needed, or if you want to assign the alias name to a column in a different table. The ability to delete an alias is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To delete an alias:**

1. On the **Aliases** tab of the **Edit Table** dialog, select the alias that you want to delete.

   If you are creating a new table, you can manage aliases on the **Manage Column Aliases** screen of the **Create Table** wizard.

2. Click **Delete Alias**  .

   The alias is removed from the list.

3. Click **Apply** or **OK** to save your changes.

# Column Sequences

A table can have one or more column sequences. A sequence is a group of related columns, such as M1 through M12 for twelve months of budget data. Sequences are used to set up other table features, such as calculated fields.

## About column sequences

A sequence is a set of one or more related columns. The sequence defines which columns belong to the sequence, and their order.

The purpose of column sequences is to support other table-related features in Axiom Software. Sequences are used with the following features:

- **Calculated fields.** Each calculated field is associated with a column sequence, to determine the applicable columns for the calculation.
- **Drill-Through definitions.** Each drill-through definition is associated with one or more column sequences, to determine the set of data that can be drilled.
- **Data conversions.** Data conversions use mapped column sequences to match columns in the data table with the appropriate rate in the conversion table.

Before configuring any of these other features, you must first define the necessary column sequences. In most cases the same sequence can be used for all three features.

For example, you may have a sequence named "Months," which contains columns M1-M12 of a table. The same sequence can be used to:

- Define calculated fields for the table such as YTD, CUR, and TOT.
- Configure drill-through definitions so that you can drill through the data in each column to the associated transactions held in another table.
- Configure data conversions so that you can report on the data in each column using a conversion rate, such as for currency conversions.

## Managing column sequences

Tables can have one or more column sequences. You can create, edit, and delete sequences when creating or editing a table.

Column sequences can be used with data tables and reference tables (except picklist tables or KPI tables).

### Adding a sequence

Column sequences can be added to a table at any time. The ability to add column sequences is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To add a sequence:**

1. From the **Sequences** tab of the **Edit Table** dialog, click **Add Sequence**  .

   If you are creating a new table, you can manage sequences on the **Manage Column Sequences** screen of the **Create Table** wizard.

   The **New Column Sequence** dialog opens.

2. In the **Sequence Name** box, type a name for the sequence.

   The sequence name can be anything you like. It is only used when setting up other table features; end users do not see or interact with sequence names.

The sequence name must be unique per table. However, multiple tables can use the same sequence name. For example, the GL2017 and the GL2018 tables can both have a sequence named "M" (for months).

3. If the sequence is for a partial year of data, edit the **Start Period** to reflect the starting period of the data.

   In most cases, the start period is 1 and does not need to be changed. However, if the sequence is for a partial year of data, you can specify the start period for the sequence.

   **Example**
   Imagine that you want to define a sequence for last year's data, but the table only contains columns for the last two months of the prior year. If you define those two months as a sequence, and leave the start period as 1, then Axiom Software will interpret the two columns as the first two months of the year rather than the last two, which would make some calculated fields incorrect. To correctly identify the two columns as the last two months of the year, you can set the **Start Period** to 11.

4. In the **Available Columns** box, select the columns to be included in the sequence, then click **Add** to move them to the **Columns in Sequence** box.

   You can use the SHIFT and CTRL keys to select multiple columns. If a column is added in error, use the **Remove** button to remove it from the **Columns in Sequence** box. You can also click **Undo Changes** to undo any changes made to the column list in the current session.

5. In the **Columns in Sequence** box, adjust the order and weight of the columns as necessary.

   The columns will be evaluated in the order listed. For example, if the sequence is used for a YTD calculation (year-to-date), and the current period is 4, the first four columns in the list would be included in that calculation. Use the **Up** and **Down** buttons to adjust the order as needed.

   The column **Weight** is used for weighted average calculations, such as the "daily weighted average" calculation for financial institutions (in which case the weight would be the number of days in that period). If you are not using weighted averages for this sequence, leave the weight at 1 for each column.

6. Click **OK** to close the **New Column Sequence** dialog.

7. To save the new sequence, click **Apply** (or **OK** if you are finished editing the table).

### Editing a sequence

Column sequence properties can be edited at any time. Keep in mind that changes to a sequence may impact associated calculated fields, drill-through definitions, and data conversion.

The ability to edit column sequences is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To edit a sequence:**

1. On the **Sequences** tab of the **Edit Table** dialog, select the sequence that you want to edit.

   If you are creating a new table, you can manage sequences on the **Manage Column Sequences** screen of the **Create Table** wizard.

2. Click **Edit Sequence**  .

3. In the **Edit Column Sequence** dialog, edit the sequence properties as desired, and then click **OK**.

4. In the **Edit Table** dialog, click **Apply** or **OK** to save your changes.

### Deleting a sequence

You can delete a column sequence from a table. If a sequence has associated calculated fields, those fields will also be deleted when you delete the sequence. If a sequence has associated drill-through definitions, then drill-through drilling will no longer be available for the columns in that sequence.

**NOTES:**

- You will be prompted to confirm the deletion before it occurs, but the confirmation prompt does not identify whether the sequence has any associated calculated fields. Before deleting the sequence, check the **Calculated Fields** tab for the table to be sure that no calculated fields use the sequence.
- You cannot delete a sequence if it is mapped as part of a data conversion.

The ability to delete column sequences is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To delete a sequence:**

1. On the **Sequences** tab of the **Edit Table** dialog, select the sequence that you want to delete.

   If you are creating a new table, you can manage sequences on the **Manage Column Sequences** screen of the **Create Table** wizard.

2. Click **Delete Sequence**  .

3. At the confirmation prompt, click **Yes**.

   The sequence is removed from the list.

4. Click **Apply** or **OK** to save your changes.

## Sequence properties

This topic is a reference for the properties available for column sequences. Column sequences are defined on the **Sequences** tab of the **Edit Table** dialog.

| Property | Description |
| --- | --- |
| Sequence Name | The name of the sequence.<br><br>The sequence name can be anything you like. It is only used when setting up other table features; end users do not see or interact with sequence names.<br><br>The sequence name must be unique per table. However, multiple tables can use the same sequence name. For example, the GL2017 and the GL2018 tables can both have a sequence named "M" (for months). |
| Start Period | The starting period of the sequence.<br><br>In most cases, the starting period is 1 and does not need to be changed. However, if the sequence is for a partial year of data (such as data that starts with period 7), you can specify the start period for the sequence. |
| Columns in Sequence | The columns that are included in the sequence.<br><br>• To add columns, select columns in the **Available Columns** list and then click **Add**.<br>• To remove columns, select columns in the **Columns in Sequence** list and then click **Remove**.<br><br>The columns will be evaluated in the order listed. For example, if the sequence is used for a YTD calculation (year-to-date), and the current period is 4, the first four columns in the list would be included in that calculation. Use the **Up** and **Down** buttons to adjust the order as needed.<br><br>The column **Weight** is used for weighted average calculations, such as the "daily weighted average" calculation for financial institutions (in which case the weight would be the number of days in that period). If you are not using weighted averages for this sequence, leave the weight at 1 for each column. |

# Calculated Fields

Using calculated fields, you can dynamically return data such as year-to-date, total, or current period. Each table can have one or more calculated fields.

## About calculated fields

Each table can have a set of calculated fields, such as total, year-to-date, and quarter-to-date. These fields are calculated automatically by Axiom Software.

To add calculated fields to a table, you must do the following:

- Define a sequence of columns to be used as the basis of the calculation.
- Specify the calculation to be performed.

For example, a table may have twelve columns of budget data representing one year of planning (NYB1 to NYB12). These twelve columns can be defined as a sequence named M (for months). You can then define a calculated field named TOT for sequence M, that uses calculation type TotalToPeriod12. Axiom Software will maintain a database field named TOT which, when queried, returns a total value for the specified sequence. This field can be referenced in Axiom queries and Axiom functions to return a total value, instead of needing to bring in the twelve columns and manually calculating the value.

**NOTE:** If using aliases, then the database columns should be named something like P1-P12, with alias names of NYB1-NYB12 pointing to those columns. You can also define alias names for the calculated fields. For example, the calculated field can be named TOT, with an alias of NYB_TOT.

Calculated fields are calculated on demand, whenever a query is made to the field. Calculated field values are not pre-calculated and stored. There is no need to "recalculate" the fields after data or system changes.

### Referencing calculated fields

For the most part, calculated fields behave just like other columns in a table. You can query calculated field data by using the calculated field name (for example, `GL2018.TOT` or just `TOT` if the table is assumed). You can also define alias names for calculated fields, and then reference them using the alias names.

In areas where they can be used, calculated fields display in lists of table columns, such as in the Axiom Sheet Assistant. However, they do not display in areas where you can only work with literal table columns, such as when using **Open Table in Spreadsheet**, or when copying table data.

### Calculation types

Each calculated field is assigned a calculation type that specifies the calculation to be performed. Axiom Software provides a number of default calculation types that can be modified as needed for industry-specific or organization-specific requirements. Additional calculation types can also be added.

Modifying and creating calculation types is an advanced function that should only be performed by implementation consultants, Axiom Software support, or advanced users.

## Managing calculated fields

Each data or reference table can have one or more calculated fields. Calculated fields are used to automatically calculate values for a sequence of data columns in a table, such as TOT (total) or YTD (year-to-date).

You can create, edit, and delete calculated fields when creating or editing a table. Calculated fields can be used with data tables and reference tables (except picklist tables or KPI tables).

## Adding a calculated field

Before creating a calculated field, you must have defined at least one column sequence for the table. The sequence is used to define the set of columns for the calculation. See Column Sequences.

The ability to add a calculated field to a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

1. From the **Calculated Fields** tab of the **Edit Table** dialog, click **Add Calculated Field**.

   If you are creating a new table, you can manage calculated fields on the **Manage Calculated Fields** screen of the **Create Table** wizard.

   The **New Calculated Field** dialog opens.

2. In the **Calculated Field Name** box, type a name for the field.

   This name is used in Axiom queries and Axiom functions to query data, so it should be descriptive and short. For example: YTD.

3. From the **Calculation Type** list, select the calculation to be used for the calculated field, such as **YearToDate**.

4. From the **Column Sequence** list, select the sequence to be used for the calculation.

5. Click **OK** to close the **New Calculated Field** dialog.

6. To save the new field, click **Apply** (or **OK** if you are finished editing the table).

## Editing a calculated field

Calculated fields for tables can be edited at any time. Keep in mind the following:

- If you change the name, this may cause errors in existing Axiom queries and functions that reference the old name. However, if an alias is set up for the calculated field, the alias is updated automatically for the new calculated field name, and any files that reference the alias will still work as expected.
- If you change the calculation type or the associated column sequence, the values of the calculated field may change.

The ability to edit calculated fields is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To edit a calculated field:**

1. On the **Calculated Fields** tab of the **Edit Table** dialog, select the calculated field that you want to edit.

   If you are creating a new table, you can manage calculated fields on the **Manage Calculated Fields** screen of the **Create Table** wizard.

2. Click **Edit Calculated Field**  .
3. In the **Edit Calculated Field** dialog, edit the field properties as desired, and then click **OK**.
4. In the **Edit Table** dialog, click **Apply** or **OK** to save your changes.

### Deleting a calculated field

Before deleting a calculated field, you should be sure that it is not in use. Deleting a calculated field that is in use will cause errors in existing Axiom queries and functions that reference the field.

If an alias is set up for the calculated field, that alias is also deleted when the calculated field is deleted. If you want to continue using the alias but instead point it to a different column, you should edit the alias before deleting the calculated field.

The ability to delete calculated fields is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To delete a calculated field:**

1. On the **Calculated Fields** tab of the **Edit Table** dialog, select the calculated field that you want to delete.

   If you are creating a new table, you can manage calculated fields on the **Manage Calculated Fields** screen of the **Create Table** wizard.

2. Click **Delete Calculated Field**  .

   The calculated field is removed from the list.

3. Click **Apply** or **OK** to save your changes.

# Data Conversions

You can set up data conversions for a table, such as to perform currency conversions. For example, the data in a table may be stored using U.S. dollars, but you want to report on it in Canadian dollars or British pounds.

Data is converted on-the-fly, when the query request (Axiom query or GetData function) includes a conversion target. The converted data is not physically stored in the database. Data conversions are similar to calculated fields in this sense.

This section discusses how to set up data conversions for a table. For more information on querying converted data, see the *Axiom File Setup Guide*.

## About data conversions

You can use data conversions to take the data in a table and convert it using a multiplication factor. For example, data conversions can be used to perform currency conversions.

Data conversions are set up as follows:

- You create a *conversion table* that holds the multiplication factors for the data conversion. This table must be set up in a specific manner in order to enable the conversion. For example, this table must have:
    - Two key columns to specify the "from" and "to" conversion. For example, you may be converting USD to Euro.
    - One key column to specify the conversion type. For example, if you are setting up currency conversions, you may calculate some values using an average monthly rate, and some values using an end-of-month rate.
    - One key column to specify the scenario. For example, if you are forecasting using currency conversions, you might have a "best case" and "worst case" scenario for the forecasted rates.
    - Any number of columns to hold the conversion rates. For example, if you are converting the data using a monthly rate, you would have 12 columns, one for each month.

    For more information on the conversion table requirements, see Conversion table requirements.
- The conversion table and the data table must each have a defined sequence to map the data columns to the appropriate multiplication factors. For example, the data in column M1 of the data table is multiplied by the rate in column M1 of the conversion table. The sequences must each have the same number of columns that will correspond directly.
- You enable the conversion by setting up the required fields on the **Conversion** tab for the data table. As part of this process, you point the data table to the relevant conversion table, and specify other conversion properties such as the default "from" value.

Once data conversions have been set up for a table, you can query the converted data by specifying a conversion target on a GetData function or on an Axiom query.

When the data is queried, the data is converted from the "from" value to the "to" value, using the relevant multiplication factor in the conversion table. The multiplication factor for any particular query is determined as follows:

- The "from" value is obtained from the conversion configuration for the data table being queried. The "from" value is either a fixed value defined in the configuration, or looked up from a specified grouping column.
- The "to" value is specified within the GetData function or in the Axiom query settings, as the conversion target. The conversion target can be a fixed value, or looked up from a specified grouping column.

- The "type" value is obtained from the conversion configuration for the data table being queried. The type value is either a fixed value defined in the configuration, or looked up from a specified grouping column. The type can also be overridden on a per Axiom query basis (but not for GetData functions).
- The "scenario" value is obtained from the conversion configuration for the data table being queried. The scenario value is a fixed value defined in the configuration. The scenario can also be overridden on a per Axiom query basis (but not for GetData functions).
- The rate for the conversion is obtained based on the column being queried. If the column is the 5th column in the data table sequence, then the conversion rate is defined in the 5th column of the conversion table sequence.

### Data conversions and calculated fields

Data conversions are applied to calculated fields, as long as the calculated field is based on the applicable sequence in the data table. To arrive at the converted calculated field value, the data conversion is first applied to the relevant columns for the calculation, and then the calculation is applied.

For example, if you have a calculated field for YTD and the current period is 3, then the relevant columns for the calculation are M1, M2, and M3. Axiom Software first applies the data conversion to these columns, and then performs the YTD calculation.

### Security and conversion tables

Users can query data using data conversions without needing to have security access to the conversion table. Only those users who need to edit the conversion table directly must have security access to the conversion table.

For example, you may create a table type named Conversions and assign all conversion tables to that table type. End users can have no access to this table type, and they will still be able to query converted data. However, users who need to be able to edit the conversion tables must be granted full or filtered access.

### Calculations for data conversions

The only calculation available for data conversions is straight multiplication. The data in the data table is multiplied by the rate in the conversion table, and the result is the converted data. No other calculation can be used.

### Drilling and data conversions

Data conversion is applied to drill-down data as follows:

- For Axiom queries, the data conversion settings for the original query are applied to the drill-down data.
- For GetData rows, if all GetData functions in the row use the same conversion target, then that target is applied to the drill-down data. If the GetData functions use different conversion targets, or if some of the GetData functions do not use data conversion, then data conversion is not applied to the drill-down data.

Data conversion is not applied to drill-through data. Data is presented as it exists in the target detail table.

## Conversion table requirements

This section details the requirements for conversion tables. Conversion tables hold the rates for data conversions on data tables. Before setting up conversions for a data table, you must first manually create the associated conversion table.

### Table structure

Conversion tables must be data tables, and they must contain the columns detailed in the following table.

**NOTE:** The key columns can be named anything you like. The example names below are used in the documentation to discuss the conversion table and its columns.

| Column | Key Column? | Data Type | Description |
|---|---|---|---|
| From | Yes | String | Contains identification codes for the "from" or starting point for the conversion.<br><br>For example, you may want to convert United States dollars to Canadian dollars. In this case, the starting value is United States dollars, and the "from" code may be USD. |
| To | Yes | String | Contains identification codes for the "to" or ending point for the conversion.<br><br>For example, you may want to convert United States dollars to Canadian dollars. In this case, the ending value is Canadian dollars, and the "to" code may be CAD. |

| Column | Key Column? | Data Type | Description |
|---|---|---|---|
| Type | Yes | String | Contains identification codes for different conversion types. Each conversion type can use a different rate.<br><br>For example, you may maintain two sets of conversion rates for currency conversions. One rate is the average of all the conversion rates for a particular month, whereas another rate uses the conversion rate as of the end of the month. Therefore you would have two conversion types, AVG and EOM. |
| Scenario | Yes | String | Contains identification codes for different conversion scenarios. Each scenario can have a different rate.<br><br>For example, if you want to apply conversion rates to forecasted data, you may have several different scenarios with different rates, such as Baseline, BestCase, and WorstCase.<br><br>You could also choose to store scenarios in different conversion tables rather than use different scenario codes within the same table. However, the column must still exist in each conversion table. |
| <Rate Columns> | No | Numeric or Integer | These columns contain the rates for the conversions, on a per period basis. The table should have one column for each period of data that you want to convert.<br><br>For example, if you will be converting 12 months of data, you would have one rate column for each month. These columns should be named something like M1-M12 or P1-P12, and must be assigned to a sequence. See the following section on the conversion sequence for more information. |

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | | **Data Type** | | *String* | *String* | *String* | *String* | *Numeric* | *Numeric* | *Numeric* | *Numeric* | *Numeric* |
| 5 | | **String Length** | | *5* | *5* | *10* | *25* | | | | | |
| 6 | | **Description** | | | | | | | | | | |
| 8 | | Delete Row | | **FromCur** | **ToCur** | **RateType** | **RateScenario** | **M1** | **M2** | **M3** | **M4** | **M5** |
| 9 | | | | CAD | USD | AVG | Actual | 0.95 | 0.96 | 0.96 | 0.97 | 0.97 |
| 10 | | | | CAD | USD | EOM | Actual | 0.94 | 0.95 | 0.96 | 0.96 | 0.98 |
| 11 | | | | GBP | USD | AVG | Actual | 1.20 | 1.30 | 1.40 | 1.20 | 1.10 |
| 12 | | | | USD | CAD | AVG | Actual | 1.05 | 1.02 | 1.00 | 0.95 | 0.97 |

*Example conversion table*

The conversion table must contain all four key columns, even if you do not need to use all four keys. For example, you may not need multiple rates or multiple scenarios. In this case you must still create the column, but you would place the same code in all rows. In the example screenshot, only one scenario applies, so the same code ("Actual") is used in all rows.

### Conversion sequence

The rate columns in the conversion table must be assigned to a sequence. This sequence must correspond to a sequence of the same length within the data table where you want to enable conversions.

For example, imagine that you have a data table GL2018, with 12 columns of data (M1-M12) that belong to a sequence named Months. In order to configure conversions for this table, you must create a conversion table with 12 rate columns (M1-M12), and assign those columns to a sequence (say, MonthlyConversionRates).

When you configure the conversion, you associate the Months sequence in the data table to the MonthlyConversionRates sequence in the conversion table. Column M1 of the Months sequence would be converted using the rates in M1 of the MonthlyConversionRates sequence.

### Rates

You can populate the rates in the rate columns manually using Open Table in Spreadsheet, or you can set up an import. For example, you might have a monthly import that brings in the actual currency conversion rates for the prior month.

The rates in the conversion table are entirely user-defined; Axiom Software does not provide any pre-populated rates for activities such as currency conversions. If you want to use different types of conversions such as AVG and EOM, your organization must obtain or calculate these rates and enter them into the conversion table.

**NOTES:**

- Rate columns must not be left blank for any record. If a rate column is left blank, data conversions for that period will return zero values.
- When converted data is queried, if the "from" value is the same as the "to" value (for example, USD to USD), and no record with that configuration exists in the table, then the applied rate is 1. In all other cases, if the conversion keys cannot be found in the table, the query will return zero values.

## Configuring conversions for a table

You can set up data conversions for a data table, such as to perform currency conversions. To do this, you enable the table for conversions using the **Conversions** tab of the table properties.

Once a table has been set up for conversions, then you can query converted data using GetData functions or Axiom queries. The data is converted on-the-fly when queried and does not reside in the database.

Before enabling conversions for a table, you must have already done the following:

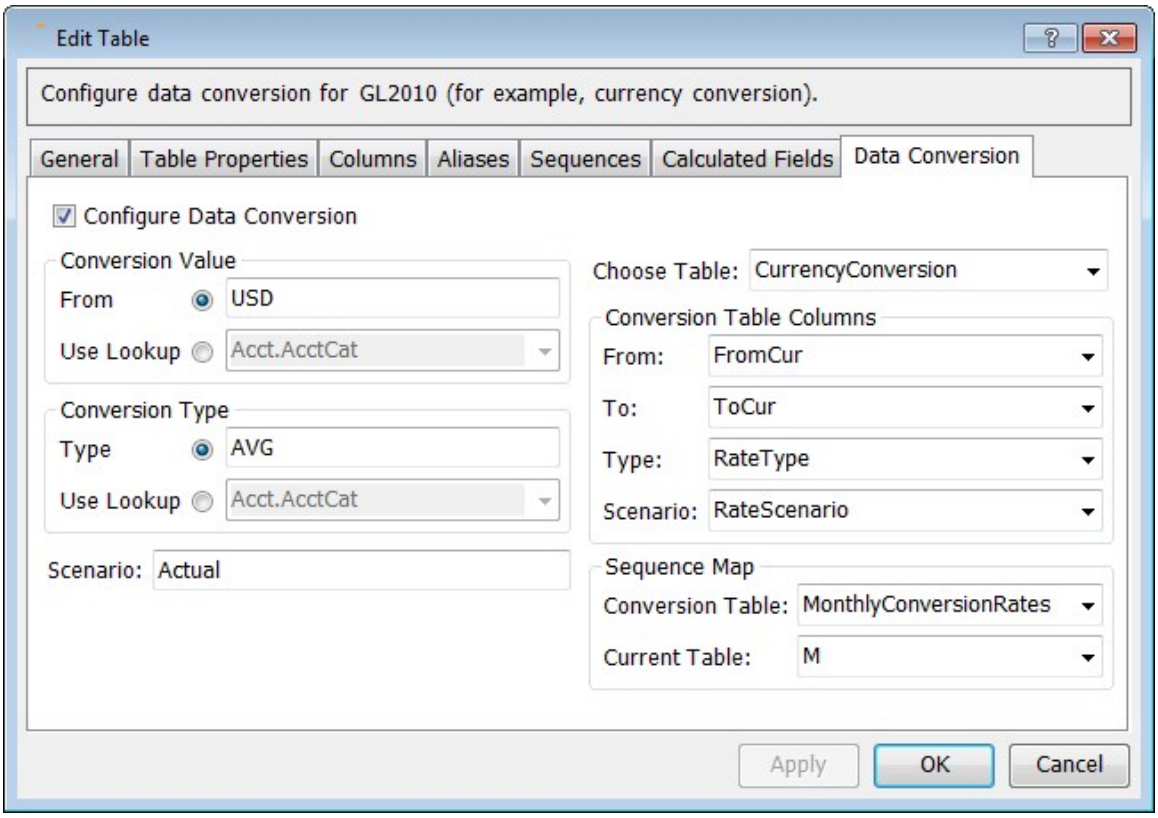
- Manually created the conversion table that stores the rates for the data conversions. For more information, see Conversion table requirements.
- Created a sequence for the columns that you want to be able to query as converted data. This may be the same column sequence that you are already using for calculated fields. For more information, see Column Sequences.
- If you want to use a grouping column to define the "from" and/or "type" values for the conversion, then you must have already set up that column on the relevant reference table. If the column does not yet exist, you can configure a fixed value for now, and then edit the configuration once the column has been created.

The ability to configure conversions on a table is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Allow changing table structure** permission for individual tables.

**To configure conversions for a table:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables**. In the Table Library on the menu, navigate to the table that you want to edit, and then select **Edit table structure**.
2. In the **Edit Table** dialog, select the **Data Conversion** tab.

   This tab is only available if the table classification is Data, and the index scheme is not Large Table.




*Example Data Conversion tab*

3. Select the **Configure Data Conversion** check box.

   This enables the conversion options so that you can set up the conversion. Later, if you want to disable conversions for this table, you can clear the check box.

4. From the **Choose Table** list, select the table that contains the rates for this conversion.

   This list is limited to tables that meet the structural requirements of conversion tables (four key columns).

5. In the **Conversion Table Columns** section, map the key columns of the conversion table according to their purpose.

| Item | Description |
|---|---|
| From | Select the column that contains the "from" or "source" codes for the conversion. |
| To | Select the column that contains the "to" or "target" codes for the conversion. |
| Type | Select the column that contains the type codes for the conversion. |
| Scenario | Select the column that contains the scenario codes for the conversion. |

6. In the **Sequence Map** section, select the associated sequences for the conversion.

| Item | Description |
|---|---|
| Conversion Table | Select the sequence that contains the conversion rates in the specified conversion table. This is the table specified in the **Choose Table** field. |
| Current Table | Select the sequence in the current table on which you want to convert data. |

   The sequences must contain the same number of literal columns or effective columns. "Effective columns" takes into account the start period of the sequence. For example, if the sequence has 6 columns but the start period is 7, the sequence has 12 effective columns because the sequence covers periods 7-12. When you query the "first" column of the data table sequence, the data will be converted using the rate in column 7 of the conversion table sequence.

7. In the **Conversion Value** section, select one of the following to specify the "from" or "source" value for the conversion. The "from" value is configured up front and applies to all conversion queries made to this data table. The "to" value is specified when you make a query via a GetData function or an Axiom query.
   - **From**: Select this option if every record in the data table uses the same start value. Enter the appropriate "from" value in the box. The value must be present in the specified **From** column.
   - **Use Lookup**: Select this option if different records in the data table use different start values. Select the appropriate column that holds the "from" value assignments. This column must be located in a lookup reference table. When a conversion query is made to the table, the appropriate "from" value will be looked up from the specified column.

   For example, if all of the departments in this data table use United States dollars (USD) as the base currency, you can select **From** and then specify `USD` directly. However, imagine that some departments use USD as the base currency, but other departments use Canadian dollars (CAD). In this case you would set up a column on the DEPT table, such as DEPT.Currency, and assign each department to either USD or CAD. You would then select **Use Lookup** and specify the DEPT.Currency column.

8. In the **Conversion Type** section, select one of the following to specify the conversion type.
   - **Type**: Select this option if every record in the data table uses the same conversion type. Enter the appropriate "type" value in the box. The value must be present in the specified **Type** column.
   - **Use Lookup**: Select this option if different records in the data table use different conversion types. Select the appropriate column that holds the type assignments. This column must be located in a lookup reference table. When a conversion query is made to the table, the appropriate type will be looked up from the specified column.

For example, if all of the accounts in this data table use end-of-month (EOM) as the conversion type, you can select **Type** and then specify `EOM` directly. However, imagine that some accounts should be converted using EOM, but other accounts should be converted using an average rate (AVG). In this case you would set up a column on the ACCT table, such as ACCT.ConvType, and assign each account to either EOM or AVG. You would then select **Use Lookup** and specify the ACCT.ConvType column.

**NOTE:** The conversion type can be overridden when querying data via an Axiom query. When using GetData functions, the type will always be the type specified here.

9. In the **Scenario** box, enter the name of the scenario to use for the conversions. The value must be present in the specified **Scenario** column. This value is required, even if all of the records in the conversion table use the same scenario value.

   **NOTE:** The scenario can be overridden when querying data via an Axiom query. When using GetData functions, the scenario will always be the scenario specified here.

10. Click **Apply** or **OK** to save the conversion configuration.

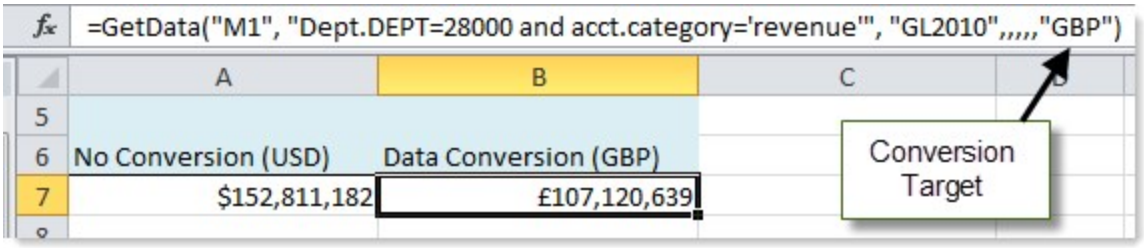
Users can now query converted data for this table.

### Disabling data conversions

If you later want to disable the ability to query converted data for this table, edit the **Conversions** tab to clear the **Configure Data Conversion** check box. Keep in mind that doing this will clear all configured conversion settings.

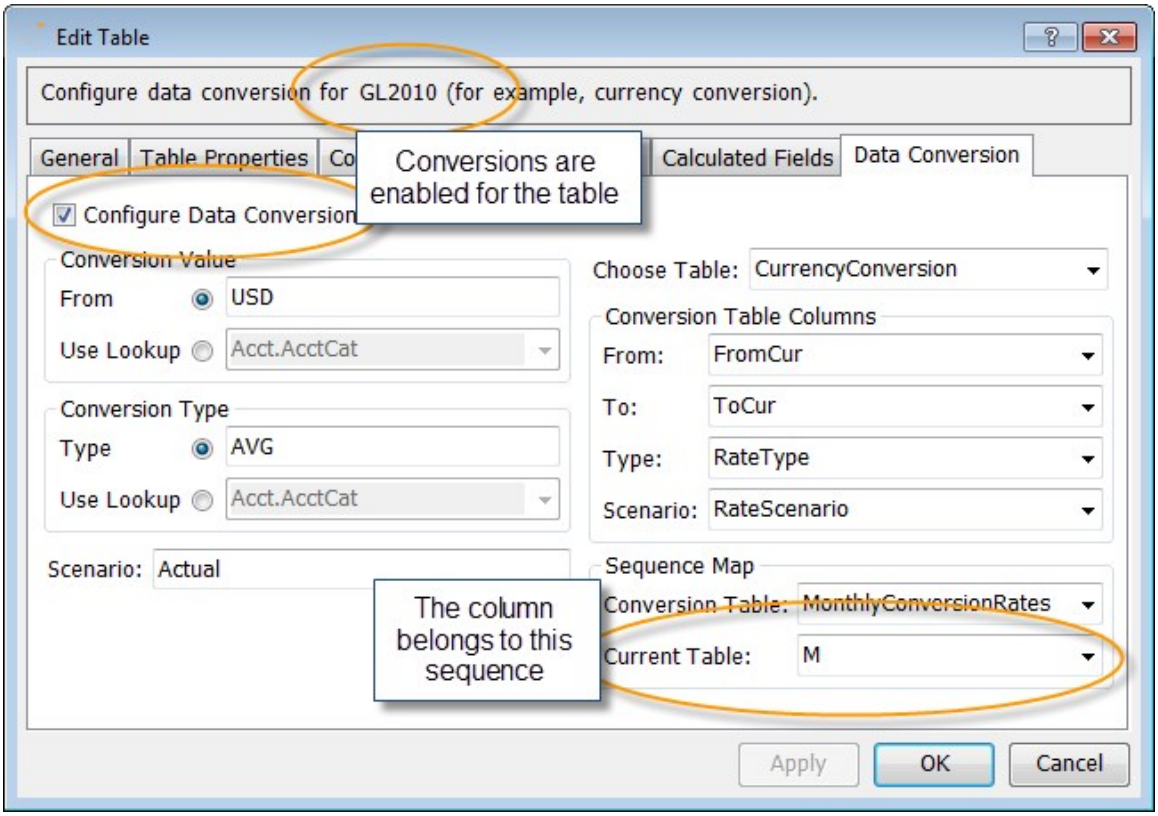
## Example data conversion

The following screenshot shows an example of data being converted using a GetData function. In this example, there are two GetData functions, one returning the "base" non-converted data, and one returning the converted data. The functions are exactly the same except that the second function is using an optional parameter to specify a conversion target.




The base data is $152,811,182 and the converted data is £107,120,639. The following explains how this converted value was obtained:

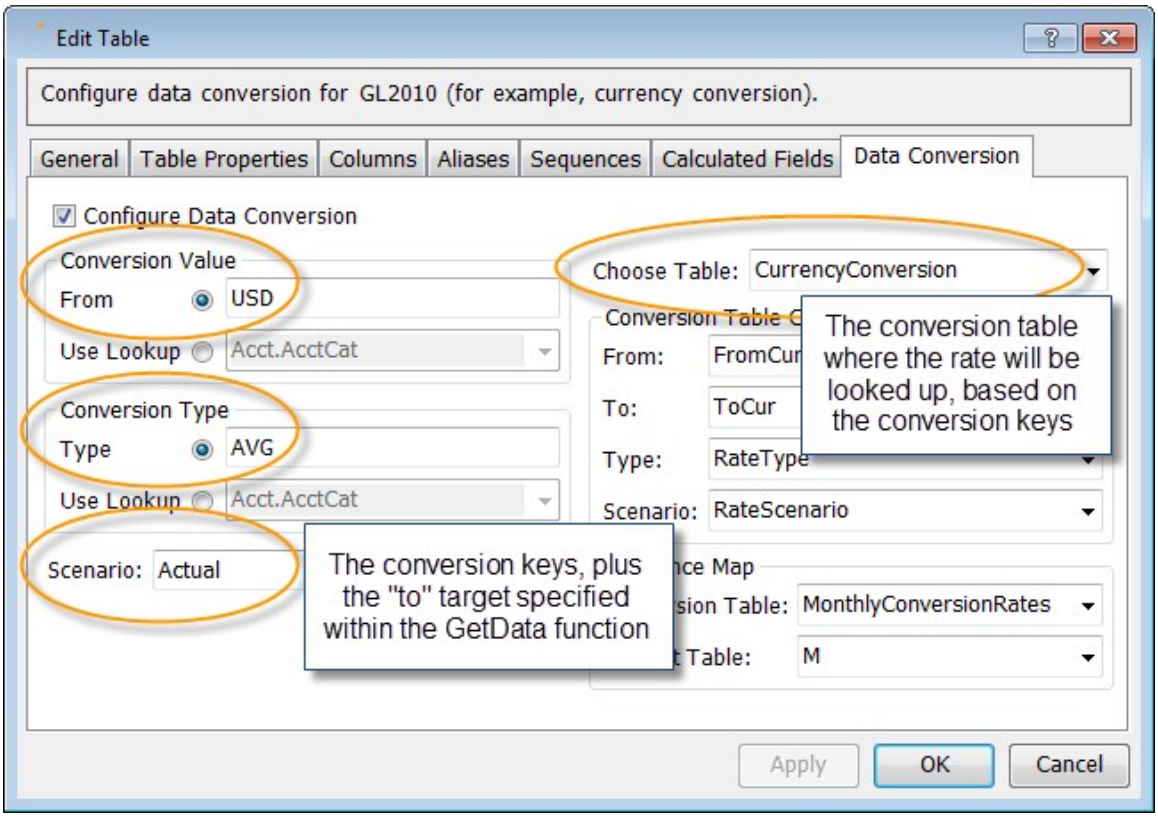
1. Axiom Software checks that the specified table is configured for conversions, and checks that the specified column (M1) is part of the mapped sequence for that conversion.



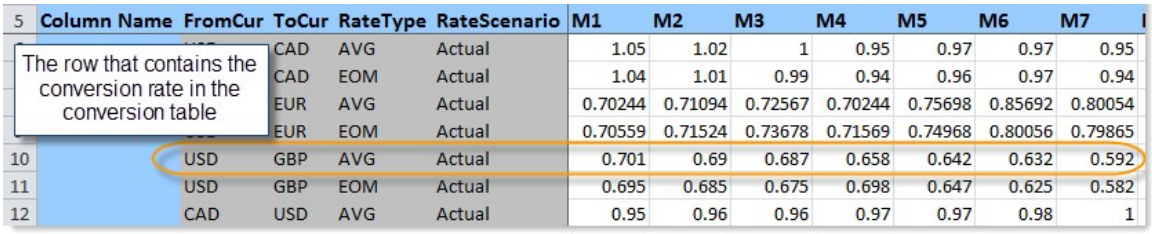

2. Axiom Software determines the keys for the conversion (From, To, Type, Scenario). The specific key combination will determine which row to use in the conversion table.

    - The "To" is determined from the GetData function, from the ConversionTarget parameter ("GBP"). (If this were an Axiom query instead of a GetData function, then the target is specified in the Axiom query settings on the Control Sheet.)
    - The "From" is determined from the conversion settings. In this case the "from" value is specified as a fixed value, "USD". If a column had been specified instead, the value would have been looked up from that column.
    - The "Type" is determined from the conversion settings. In this case the "type" value is specified as a fixed value, "AVG". If a column had been specified instead, the value would have been looked up from that column.
    - The "Scenario" is determined from the conversion settings ("Actual").

    Therefore the keys for this conversion are USD, GBP, AVG, Actual.

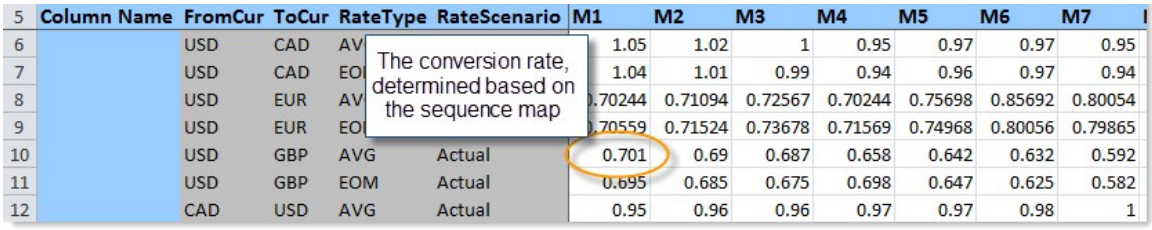



3. Axiom Software identifies the record in the conversion table that holds the conversion rate, using the conversion keys.




| 5 | Column Name | FromCur | ToCur | RateType | RateScenario | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | CAD | AVG | Actual | 1.05 | 1.02 | 1 | 0.95 | 0.97 | 0.97 | 0.95 |
| | | | CAD | EOM | Actual | 1.04 | 1.01 | 0.99 | 0.94 | 0.96 | 0.97 | 0.94 |
| | | | EUR | AVG | Actual | 0.70244 | 0.71094 | 0.72567 | 0.70244 | 0.75698 | 0.85692 | 0.80054 |
| | | | EUR | EOM | Actual | 0.70559 | 0.71524 | 0.73678 | 0.71569 | 0.74968 | 0.80056 | 0.79865 |
| 10 | | USD | GBP | AVG | Actual | 0.701 | 0.69 | 0.687 | 0.658 | 0.642 | 0.632 | 0.592 |
| 11 | | USD | GBP | EOM | Actual | 0.695 | 0.685 | 0.675 | 0.698 | 0.647 | 0.625 | 0.582 |
| 12 | | CAD | USD | AVG | Actual | 0.95 | 0.96 | 0.96 | 0.97 | 0.97 | 0.98 | 1 |

4. Axiom Software uses the sequence map in the conversion configuration to determine which rate to use. The GetData function queries column M1 of the GL2010 table, which is the first column of the M sequence. This sequence is mapped to the sequence MonthlyConversionRates in the CurrencyConversion table, so the conversion rate is taken from the first column of that sequence, M1.




| 5 | Column Name | FromCur | ToCur | RateType | RateScenario | M1 | M2 | M3 | M4 | M5 | M6 | M7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 6 | | USD | CAD | AV | | 1.05 | 1.02 | 1 | 0.95 | 0.97 | 0.97 | 0.95 |
| 7 | | USD | CAD | EO | | 1.04 | 1.01 | 0.99 | 0.94 | 0.96 | 0.97 | 0.94 |
| 8 | | USD | EUR | AV | | 0.70244 | 0.71094 | 0.72567 | 0.70244 | 0.75698 | 0.85692 | 0.80054 |
| 9 | | USD | EUR | EO | | 0.70559 | 0.71524 | 0.73678 | 0.71569 | 0.74968 | 0.80056 | 0.79865 |
| 10 | | USD | GBP | AVG | Actual | 0.701 | 0.69 | 0.687 | 0.658 | 0.642 | 0.632 | 0.592 |
| 11 | | USD | GBP | EOM | Actual | 0.695 | 0.685 | 0.675 | 0.698 | 0.647 | 0.625 | 0.582 |
| 12 | | CAD | USD | AVG | Actual | 0.95 | 0.96 | 0.96 | 0.97 | 0.97 | 0.98 | 1 |

5. Axiom Software performs the math on the data table to arrive at the converted value: 152,811,182 x .701 = 107,120,639.


# Table Tools

This section details the tools and utilities available to manage the tables in your system.

## Viewing and editing tables using Open Table in Spreadsheet

Axiom Software supports a spreadsheet interface for viewing and editing the data within a table, known as **Open Table in Spreadsheet**. This feature is primarily intended for populating reference tables, but it can also be used to edit data tables if necessary. A read-only version is also available, so that you can view the table data without locking the table for editing.

**NOTE:** The **Open Table in Spreadsheet** feature does not apply to document reference tables. For document reference tables, a menu item is available named **Open Source Document**, which opens the source spreadsheet for the table (for example, the source driver file).

### Opening a table

You can open a data or reference table within a spreadsheet interface, in order to view and/or edit the table data.

The ability to use Open Table in Spreadsheet is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Open Table in Spreadsheet** permission for individual tables (set to **Read** or **Read/Write**).

**NOTE:** Axiom Software enforces a system-wide limit on the number of data rows that can be brought into the spreadsheet. By default, the limit is 50,000 rows, but the limit can be changed for a system using the system configuration setting **MaxTableRowsToOpenInSpreadsheet**. For more information, see System Configuration Settings.

**To open a table for viewing or editing:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables**.
2. In the Table Library section at the bottom of the **Tables** menu, navigate to the table that you want

to open in a spreadsheet, then click one of the following:

- **Open in Spreadsheet**: Edit the table data in a spreadsheet interface.
- **Open in Spreadsheet (Read-Only)**: View the table data in a spreadsheet interface. This may be the only option available if the table is configured as read-only, or if your security permissions restrict you to read-only access. Additionally, tables using the LargeTable index scheme can only be opened as read-only.

**TIP:** You can also open a table in a spreadsheet when working with the Table Library in Axiom Explorer.

The **Open Table in Spreadsheet** dialog opens. By default, the table where you launched the dialog is automatically selected as the table to open. If desired, you can change the table to open by clicking the **Choose Table** button to the right of the **Table** box.

If you want to view all rows of the table, with no special features, then you can click **OK** at this point to open the table. Otherwise, you can use the options in the dialog to filter the data to be displayed, or to apply special features.

3. Optional. If you want to filter the data to be displayed in the spreadsheet, you can use the following options:

| Option | Description |
|---|---|
| Columns to Return | Select one of the following:<br>• **All columns** (default): Displays all user-defined columns of the table.<br>• **Selected columns**: Select this option if you want to limit the data to particular columns. Click the column icon to the right to choose the columns to display.<br>Key columns are always included in the spreadsheet. You can select any other set of columns to display by checking or clearing the column check boxes.<br>Additionally, you can select one or both of the following options. By default these options are not selected.<br>• **Include audit columns**: Selecting this option will display the following audit columns: RecordModifiedBy, RecordModifiedDTM.<br>• **Include calculated columns**: Selecting this option will display all calculated columns defined for the table.<br>Audit columns and calculated columns are always read-only. |

| Option | Description |
| --- | --- |
| Rows to Return | By default, **All rows** of the table (up to the maximum row limit) will be displayed. To limit the data to a specific number of rows, select **Limit to** and then type the number of rows to return. If you specify 0 rows, then the column headers will be brought into the spreadsheet with no data.<br><br>**NOTE:** If limited to a specific number of rows, the rows returned are a random set within the eligible data. |
| Data Filter | To limit the data by using a filter criteria statement, type the criteria statement into the filter box, or click the **Filter Wizard** button . |

You can use any combination of these settings to filter the data. If a row limit and a data filter are both defined, then the data filter is first applied to determine the total eligible rows, and then the row limit is applied.

**NOTE:** If you have filtered write access to a table and you want to open the table for editing, the records that you bring into the spreadsheet must match your write filter. If your read filter is the same as your write filter, then this will happen by default. However, if your read filter is greater than your write filter, then you must apply a filter to **Open Table in Spreadsheet** that is equal to or less than your write filter, or else you will not be able to save changed data.

4. Optional. In the **Options** section, select any special features to apply to the table:

| Option | Description |
| --- | --- |
| Apply Autofilter | Applies Microsoft Excel's AutoFilter feature to the spreadsheet. You can use this to sort and filter the data in the spreadsheet. This option is selected by default. |
| Open Read-Only | Opens the table read-only in the spreadsheet. You should select this option if all you need to do is view the data, to prevent the table from being locked for editing.<br><br>If you only have read-only access to the table, then this option is selected by default and cannot be cleared.<br><br>This option is checked by default if you opened the table using the **Open in Spreadsheet (Read-Only)** menu item. |

| Option | Description |
| --- | --- |
| Sort Data | Specifies whether data is sorted as part of the database query to the table. This option is selected by default, which means that data is sorted by the first key column in the table.<br><br>If desired, you can uncheck this option to bring unsorted data into the spreadsheet. This will improve performance when querying large numbers of records. Data can then be sorted in the spreadsheet using the AutoFilter feature.<br><br>**NOTE:** If the table has 100,000 records or more, this option is unchecked by default to speed the data query. |
| Limit to Unused Items | Filters the table so that only items that are not used in any associated data tables are displayed.<br><br>This option is only available for reference tables. For more information, see Identifying unused items. |

5. Click **OK** to open the table.

The table opens within a temporary spreadsheet. The spreadsheet is highly structured and is configured in the correct manner to save data back to the database. Do not unfreeze panes and edit the spreadsheet structure.

Freeze panes are automatically set to include the key column(s) and description column(s) for the table. However, Axiom Software will never freeze more than half of the available screen area to preserve the ability to scroll. So depending on the current size of the window and the number of columns to freeze, some key and description columns may not be included in the frozen panes.

The Table task pane is available in the Axiom Assistant area. You can use this task pane to refresh the spreadsheet with current data and to change any Open Table in Spreadsheet settings.

If you opened the table using **Open in Spreadsheet (Read-Only)**, then the text **(R/O)** displays on the file tab. You can refresh the data in the spreadsheet, but you cannot save any edits.

**TIP:** Once you have opened a table in a spreadsheet, you can save that table as a favorite (using the right-click option on the file tab) and the favorite will retain your current filter settings. Also, the associated entry in your **Recent** items list will remember the filtering options that you last used to open the table.

## Editing a table

You can edit table data when a table is opened using **Open Table in Spreadsheet**.

**NOTES:**

- If the table is open as read-only, then you cannot save any changes to the table.
- If you have filtered write access to a table and you want to open the table for editing, the records that you bring into the spreadsheet must match your write filter. If your read filter is the same as your write filter, then this will happen by default. However, if your read filter is greater than your write filter, then you must apply a filter to **Open Table in Spreadsheet** that is equal to or less than your write filter, or else you will not be able to save changed data.

### Adding or editing data

To add or edit data in the table, type the data into the appropriate cells. You can also use Excel features to copy and paste data.

If you are editing a column that has an assigned lookup column, then you can double-click cells in that column to bring up a list of valid values. You can select a value from the list to place it into the cell. This helps to ensure that you do not accidentally enter an invalid value in a validated column. You can tell which columns are enabled for this double-click behavior by the presence of a comment in the column header.

Any data entered into a cell must match the data type for that column (for example, integer or string). If the contents of a cell do not match the column's specified data type, an error will result when the data is saved. See Column properties for more information on column data types.

If all of the data in a column is grayed out, then that column is read-only and its data cannot be edited. The column also has an Excel comment stamped on its name, explaining that it is read-only. You are not prevented from changing data within the column, but the changes will not be saved.

**NOTE:** If the comment for a read-only column appears to be "stuck" in a visible state, you can correct the issue by changing your Excel comment settings. Go to **File > Options > Advanced**, and then in the **Display** section select **Indicators only, and comments on hover** for the option **For cells with comments, show**.

The ability to save data using Open Table in Spreadsheet is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Open Table in Spreadsheet** permission for individual tables (set to **Read/Write**). Non-administrators must also have write access to the table (full or filtered) in order to save data.

### Adding or deleting rows

- To add a row, type the data into the appropriate cells. In all cases, the values in the key column(s) must be unique. Data is not summed when using Open Table in Spreadsheet.
- To delete a row, place your cursor in Column B, click the arrow button, and then select **Delete Row**. The text "Delete Row" appears in Column A. When the changes are saved to the database, any rows marked with the delete tag will be deleted.

  **NOTE:** Simply deleting a row from the spreadsheet does not delete the row from the database. The row must be flagged with the delete tag in order to be deleted.

The ability to save data using Open Table in Spreadsheet is limited to administrators and to users with the following security permissions: either the global **Administer Tables** permission, or the **Open Table in Spreadsheet** permission for individual tables (set to **Read/Write**). Non-administrators must also have write access to the table (full or filtered) in order to save data.

### Saving table changes to the database

If you have made changes to the table data or structure using Open Table in Spreadsheet, you can save those changes back to the source table in the Axiom Software database.

**To save the table changes to the database:**

- On the **Axiom** tab, in the **File Options** group, click **Save**.

If any errors occur during the save, they are displayed in the **Save Errors** task pane. This pane identifies the type of error and directs you to the location of the error. Any errors must be resolved before the save can proceed.

If no errors occur, the data is saved and a confirmation message displays. The confirmation message includes information about the number of records saved (or deleted). Click **Show Details** to see this information if desired.

After changes are saved, the spreadsheet is automatically refreshed.

## Table task pane

Once you have opened a table using Open Table in Spreadsheet, the Table task pane is available in the Axiom Assistant area.

By default, this pane displays the settings for the currently active table opened in a spreadsheet, including any filter settings or other special options.

You can use this pane to change any of the Open Table in Spreadsheet settings, for either the current table or to open a new table. When you refresh the spreadsheet, the new settings are applied and current data is brought into the spreadsheet.

**NOTES:**

- Refreshing the spreadsheet essentially deletes the existing spreadsheet and replaces it with a new spreadsheet. If any changes have been made to the spreadsheet (and the spreadsheet is open read/write), then you will be prompted to save your changes before refreshing.
- If you use the file tab to save the table as a favorite, the current Open Table in Spreadsheet settings will be saved in the favorite. When you subsequently open the table from Favorites, the Open Table in Spreadsheet dialog will be pre-populated with the favorite settings, which you can then edit if desired.

The following options are available in the task pane:

| Option | Description |
|---|---|
| Table | The table to open in a spreadsheet. The number of rows in the selected table is displayed underneath the table name.<br><br>To specify a table, you can either type the table name, or use the **Choose Table** button to the right of the box.<br><br>Keep in mind that changing the table will *replace* the current table in the spreadsheet; it does not open an additional spreadsheet. If you want to open multiple tables at the same time, you must use the Table menu or Axiom Explorer to open the table, not the Table task pane. |
| Columns to Return | Specify the columns to return in the spreadsheet:<br>• **All columns** (default): The spreadsheet displays all columns in the table.<br>• **Selected columns**: Select the columns that you want to display by clicking the column icon to the right. Key columns are always included in the spreadsheet. You can select any other user-defined columns to display by checking or clearing the column check boxes.<br><br>Additional options:<br>• **Include audit columns**: The spreadsheet displays audit columns in addition to the user-defined columns. This includes RecordModifiedBy and RecordModifiedDTM. Audit columns are read-only.<br>• **Include calculated columns**: The spreadsheet displays calculated columns in addition to the user-defined columns. Calculated columns are read-only. |

| Option | Description |
| --- | --- |
| Rows to Return | Optional. Specify the rows to return in the spreadsheet:<br>• **All rows**: The spreadsheet displays all rows of the table, up to the maximum row limit. By default, the limit is 50,000 rows. This limit can be changed in the system configuration settings.<br>• **Limit to**: Type the desired number of rows that you want to display in the spreadsheet. The rows returned are a random set within the eligible data. If you specify 0 rows, then the column headers will be brought into the spreadsheet with no data. |
| Data Filter | Optional. Specify a filter criteria statement to limit the data to return in the spreadsheet.<br>You can type the filter or you can use the Filter Wizard  . The filter must use the current table or a lookup table.<br>If a row limit and a data filter are both defined, then the data filter is first applied to determine the total eligible rows, and then the row limit is applied.<br>**NOTE:** If you have filtered write access to a table and you want to open the table for editing, the records that you bring into the spreadsheet must match your write filter. If your read filter is the same as your write filter, then this will happen by default. However, if your read filter is greater than your write filter, then you must apply a filter to **Open Table in Spreadsheet** that is equal to or less than your write filter, or else you will not be able to save changed data. |
| Apply Autofilter | Applies Microsoft Excel's AutoFilter feature to the spreadsheet. You can use this to sort and filter the data in the spreadsheet. |
| Open Read-Only | Opens the table read-only in the spreadsheet. You should select this option if all you need to do is view the data, to prevent the table from being locked for editing.<br>If you only have read-only access to the table, then this option is selected by default and cannot be cleared. |
| Sort Data | Specifies whether data is sorted as part of the database query to the table. This option is selected by default, which means that data is sorted by the first key column in the table.<br>If desired, you can uncheck this option to bring unsorted data into the spreadsheet. This will improve performance when querying large numbers of records. Data can then be sorted in the spreadsheet using the AutoFilter feature.<br>**NOTE:** If the table has 100,000 records or more, this option is unchecked by default to speed the data query. |

| Option | Description |
|---|---|
| Limit to Unused Items | Filters the table so that only items that are not used in any associated data tables are displayed.<br><br>This option is only available for reference tables. For more information, see Identifying unused items. |

After making any changes to the Table options, click the **Refresh Spreadsheet** button at the bottom of the task pane.

# Identifying unused items

Reference tables can contain master lists of dimensional items to be used in data tables, such as lists of departments or accounts. Periodically you may want to review the lists to see if any items are not used in the system, for troubleshooting or to remove obsolete items.

You can see which items in a reference table are unused by using the **Limit to unused items** feature. This feature opens the table within a spreadsheet, and automatically applies a filter so that the only items shown are items that are not used in any associated data tables. To determine which items are unused, Axiom Software checks all data tables that have a lookup relationship to the reference table.

This feature is only available for reference tables. Keep in mind that the feature is only useful if the reference table is the lookup source for at least one data table. If you use this feature on a non-lookup reference table—for example, on a table that defines an import mapping relationship—then every item in the table will be identified as unused because the table is not linked to any data tables.

**To identify the unused items in a dimensional reference table:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables**.
2. In the **Table Library** section at the bottom of the **Tables** menu, navigate to the table, and then click **Limit to unused items**.

   **TIP:** You can also select this option within the Open Table in Spreadsheet dialog.

3. In the Open Table in Spreadsheet dialog, select any optional features, such as **Apply Autofilter** and **Open Read-Only**.
4. Click **OK**.

The selected table opens in a spreadsheet, displaying only the items that are not used in any associated data tables. The tab title for the spreadsheet is ***Tablename*** **(unused)**, to differentiate the file from any "normal" Open Table in Spreadsheet instances you may have open.

If, after reviewing the list, you determine that a particular item is obsolete and should be deleted, you can delete the item within the current spreadsheet. Place your cursor in column A of the applicable row, and then select **Delete Row** from the drop-down list. When you save, the item will be deleted.

# Copying table data

You can copy data from one table to another, or within the same table. If you are copying between tables, the source and destination tables must share common key columns.

The destination table must already contain columns to hold the copied data. **Copy Table Data** does not create columns as part of the copy operation.

You can copy all rows of data, or you can specify an optional filter to limit the data to be copied.

**NOTE:** By default, the Copy Table Data utility is limited to copying no more than 250,000 rows of data (via the **MaximumRowsAllowedThroughCopyTableData** system configuration setting). If you need to copy more rows than that, we recommend creating an import utility (to import from the source table to the target table) and then scheduling that import to run via Scheduler.

**To copy table data:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Copy Table Data**.

   The **Copy Table Data** dialog opens. The top section of the dialog specifies the source and destination tables for the copy operation. The middle section can be used to define an optional filter. The bottom section specifies which columns are to be copied from the source table, and which columns in the destination table will receive the data.

2. For **Source Table**, specify the source table from which to copy data. To select a table, click the **Choose table** icon or double-click in the box to open the **Select Table** dialog.

   By default, the dialog is filtered to only show data and reference tables, but you can copy data from a document reference table if desired. Select the **Doc. Reference** check box to see these tables.

3. For **Destination Table**, specify the target table for the copied data. To select a table, click the **Choose table** icon or double-click in the box to open the **Select Table** dialog.

   **NOTE:** Document reference tables and read-only tables cannot be specified as the destination table.

4. If desired, in the **Filter** box, specify a filter to limit the data to be copied.

   For example, the filter `Dept=200` would copy only the data for department 200. If no filter is defined, then all data rows are copied for the selected columns.

You can type in the filter or use the Filter Wizard. The selections in the Filter Wizard are automatically limited to the relevant tables—the table being copied, and any lookup tables for that table.

To check that the filter uses valid syntax, click **Validate filter**.

5. In the **Column mapping** section, specify the source columns to be copied by selecting the check box to the left of the column name.

   **NOTE:** Calculated fields are listed in the source columns. If desired, you can copy from a calculated field to a "normal" column. This will take the current calculated values in the calculated field and copy them as static values to the destination column.

   You can select the check boxes for individual columns, or you can use one of the following:

   - Click **Select physical columns** to select only the literal columns in the table, excluding any calculated columns.
   - Click the check box in the column header to select all columns, including any calculated columns.

   All key columns must be included in the copy operation and are selected by default.

6. In the **Column mapping** section, specify the destination column for each selected source column.

   You can manually select the destination columns by using the drop-down list for each column, or you can click **Auto-generate destination column mapping** to have Axiom Software "guess" the appropriate destination columns based on column names and data types.

   In most cases it will be easiest to auto-generate the mappings and then make any edits as necessary.

   You only need to specify a destination column for each source column to be copied. If a destination column is specified but the check box is not selected for the source column, then that entry will be ignored.

7. Click **Copy**.

   The **Copy** button is not available until all required settings have been made and no invalid settings exist. If the settings contain errors, an error message displays at the bottom of the dialog.

The copy operation begins. Axiom Software displays a confirmation message when it is complete, or an error message if the operation experienced errors.

The **Copy Table Data** dialog remains open with the current settings until you close it. If the operation experiences errors, you can change the settings and try again.

**TIP:** You can also copy data by right-clicking a table in Axiom Explorer (or the Explorer task pane) and choosing either **Copy data from this table** or **Copy data to this table**. The table will automatically be set as either the source table or the destination table as appropriate.

# Zeroing or deleting table data

You can zero or delete the data in a table by using the **Zero/Delete Table Data** utility. For example, you might use this utility to clear data before starting a new cycle of planning, or to clear testing data.

When you zero data in a table, the existing rows are retained, but the non-key column values are changed as follows:

- Numeric columns and Integer columns (all types) are changed to zero.
- String columns are changed to blank values.
- Date, DateTime, and Boolean columns are changed to their default values.

When you delete data in a table, the entire row is deleted from the table.

You can choose to zero or delete all rows in the table, or you can specify a filter to zero or delete a subset of rows. Additionally, when zeroing data, you can zero all columns or only selected columns.

Copy Table Data is only available to administrators or to users with the **Administer Tables** permission. For non-admin users, the user's write access permissions to the table apply.

**To zero or delete data from a table:**

1. On the **Axiom** tab, in the **Administration** group, select **Tables > Table Administration > Zero/Delete Table Data**.

   The **Zero/Delete Table Data** dialog opens.

2. For **Source table**, specify the table where you want to zero or delete data. To select a table, click the **Choose table** icon or double-click in the box to open the **Select Table** dialog.

   Only data tables and reference tables are listed. Document reference tables are controlled by their source documents and therefore are not available to be edited via this utility.

   **NOTE:** If a table is set to read only, then it is not listed here. You cannot zero data or delete rows in a table that has been set to read only.

3. In the **Filter** box, specify the set of data that you want to zero or delete:
   - If you want to zero or delete *all* rows in the table, then leave the **Filter** box blank.
   - If you want to zero or delete a subset of rows, then define a filter in the **Filter** box. For example, the filter `Dept=200` would zero or delete only the data for department 200.

     You can type in the filter or use the Filter Wizard. The selections in the Filter Wizard are automatically limited to the relevant tables—the table being zeroed or deleted, and any lookup tables for that table.

     To check that the filter uses valid syntax, click **Validate filter**.

4. In the **Processing Action** section, specify whether to zero or delete data:

- **Zero selected columns in rows**: This action zeroes rows in the table, limited by the filter (if specified). If this option is selected, then you must also specify which columns in the table to zero.
- **Delete rows**: This action deletes rows in the table, limited by the filter (if specified).

**NOTE:** If the source table uses the Large Table index scheme, then it is not supported to zero data in that table. The only available option is to delete rows.

5. If you selected to zero data, then specify which columns in the table to zero:
   - To zero all columns, select the check box in the header to select all columns.
   - To zero only selected columns, select the check boxes for the desired columns.

   Key columns, alternate key columns, and validated columns are never zeroed and are not available to be selected.

   **NOTE:** If a column in the table is an identity column or has been set to read-only, then that column cannot be zeroed and is not available to be selected. The text "(R/O)" displays next to the column name.

6. Click **Zero** or **Delete** (depending on which action you are performing).
7. Axiom Software calculates how many rows in the table will be affected, and then displays a message listing the number of rows and asking if you want to proceed. Click **Yes** to continue.

   If the number of rows is not as expected, you can click **No** to return to the dialog and change your settings.

   **NOTE:** If you are deleting all of the rows in a table, the table will be truncated, which is a faster deletion process. If you are deleting a subset of rows in the table, and the number of rows to delete exceeds 50,000, then you will be warned that the deletion process may take some time.

The specified data is deleted or zeroed.

**TIP:** You can also delete or zero data by right-clicking the table in Axiom Explorer (or the Explorer task pane) and then choosing Zero/Delete Table Data. The table will automatically be set as the source table.

# Setting the current period

The *current period* in Axiom Software is used to drive calculated fields for database tables. For example, if the current period is 6, then a year-to-date (YTD) calculation would calculate a total through period 6.

You can set the current period at a system-wide level and at a table level. By default, tables use the system current period, but this can be overridden on a per table basis. If a table has a defined current

period, then the calculated fields for that table will use the table current period instead of the system current period.

For example, most of the tables in your system may contain 12 periods, but you have one table that contains 26 periods. In that case the current period must be set at the table level for the 26-period table, otherwise the calculated fields for that table will not calculate as desired.

The current period only applies to data tables. Reference tables and document reference tables do not have a current period.

**NOTE:** File groups can also have a current period, to be used for planning purposes. This planning current period is typically stored in a driver file for the file group, and does not impact calculated fields in data tables.

## Setting the system current period

You can change the system current period at any time. When you change the system current period, all calculated fields that use the current period (such as year-to-date) are updated for the new current period.

Tables that have a current period set at the individual table level are not impacted by changes to the system current period.

**NOTE:** Alternatively, you can use Save Type 4 to change the system current period from within a spreadsheet. For more information, see Editing system configuration settings using Save Type 4.

**To change the system current period:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > System Current Period / Year**.

   The **System Current Period** dialog opens. The current value for the current period is displayed in the **System Current Period** section.

2. In the **System Current Period** section, in the **New Value** box, type the new value for the current period.

3. Click **OK**.

The system current period is changed. Queries to calculated fields will reflect the new current period.

## Setting the current period for a table

By default, tables use the system current period. If necessary, you can set a different current period for an individual table, and the calculated fields in that table will use that table's current period.

For example, you might have one table that uses 26 periods, while all other tables use 12 periods. In that case, you would need to set the current period for the 26-period table individually.

**NOTE:** Alternatively, you can use Save Type 4 to change the current period for tables from within a spreadsheet. This may be an easier method if you have many tables to update, or if you need to do it frequently. For more information, see Managing tables using Save Type 4.

**To set the current period for a table:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Table Current Periods**.

   The **Table Current Periods** dialog opens. The list of tables is grouped by table type. To view or edit the settings for a table, expand the table type grouping.

   The dialog only lists table types that are configured as data table types. Tables without a table type are also listed.

2. If the table is not already configured to use a table current period, clear the **Uses System** check box for the table.

   If this box is checked, the table uses the system current period. If this box is cleared, the table uses the current period specified in this dialog.

3. To change the current period for a table, place your cursor in the **Current Period** column, and then type the new period.

   **NOTE:** Typing a new period into the **Current Period** column will automatically clear the **Uses System** check box, if it is not already cleared for that table.

4. Click **Apply** to save the change (or **OK** if you are finished editing the current period).

The table current period is changed. Queries to calculated fields will reflect the new current period.

If you ever want to change the table to use the system current period again, select the **Uses System** check box for the table.

# Setting the current year

You can specify a current year for the system. Typically this represents the current year of actual data that you are reporting on in the system (not the current planning year).

The current year can be referenced using the GetPeriod function. Using this function, you could:

- Display the current year in report or plan headers.
- Use the current year to drive processes in Axiom files, such as when using Save Type 4 to modify aliases.

You can also reference the current year in import utilities using the built-in import variable {SystemCurrentYear}.

The current year is not used programmatically and is optional. If you do not need to use the current year setting, then you do not need to maintain it.

**To set the current year:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > System Current Period / Year**.

   The **System Current Period** dialog opens. The current value for the current year is displayed in the **System Current Year** section.

2. In the **System Current Year** section, in the **New Value** box, type the new value for the current year.

3. Click **OK**.

The system current year is changed.

# Managing the Table Library

For ease of access, tables are organized into a series of virtual folders known as the Table Library. For example, you might have the following table folders configured in your system:

- GL data
- Plan data
- Drivers
- Dimensions
- Mapping tables

You can have as many folders and sub-folders as you need and name them anything you like. Axiom Software does not assume or require any particular folder names. You can change existing folder names as desired, as long as you update any impacted Save Type 3 settings (Save Type 3 specifies a target folder in the Table Library).

**To manage the folders in the Table Library:**

- On the **Axiom** tab, in the **Administration** group, select **Tables > Table Administration > Manage Tables**.

Axiom Explorer opens, with the focus on the Table Library. The following actions are available for table folders:

| Action | Description |
|---|---|
| Adding folders | To add a top-level folder, right-click **Table Library** and then select **New Folder**.<br>To add a sub-folder, right-click the parent folder. |

| Action | Description |
|---|---|
| Renaming folders | To rename a folder, right-click the folder and click **Rename**. The folder name becomes highlighted and editable. Type the new name. |
| Deleting folders | To delete a folder, right-click the folder and click **Delete**.<br><br>**NOTE:** If any tables are currently in the folder, then the **Delete** option is not available and does not display on the menu. You must first delete or move all of the tables in the folder. |
| Moving tables between folders | To move a table between folders, drag and drop the table to the new folder.<br><br>Note that if you want to change the folder location for a document reference table, you must edit the Control Sheet of the source file. |

# Data diagrams

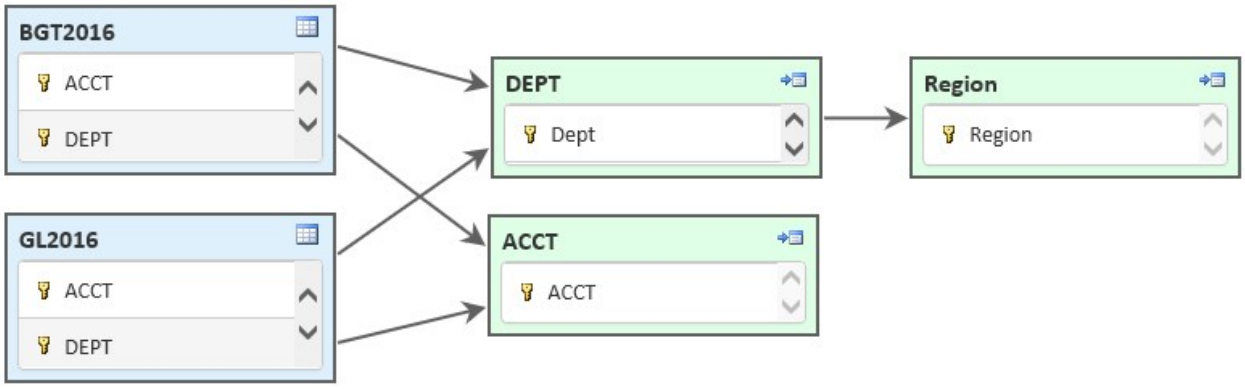
Data diagrams provide a way to visually explore and document the relationships between your tables in Axiom Software. It is important to understand these relationships when designing data queries and performing other setup activities.

For example, you may want to know information such as:

- Which tables look up to the DEPT table?
- Which tables does the GL2019 table look up to?
- Which lookup columns do the tables GL2019 and Plan2019 have in common?
- Does a relationship exist between the GL2019 table and the Region table?

It can be time-consuming to answer these questions by inspecting the individual table properties directly, and difficult to visualize the relationships—especially when dealing with multiple levels of lookups. By creating a data diagram, you can find the answers to these questions by dragging and dropping the tables you want to explore. Relationships between tables are visually marked with arrows so that you can see the direction of the relationship and also trace the connection through multiple tables if applicable.

For example, in the following diagram we can see that the data tables BGT2016 and GL2016 have lookup relationships to the reference tables ACCT and DEPT. The arrows show the direction of the relationship—for example, BGT2016 looks up to DEPT. Tracing the arrows through the DEPT table, we can also see that both data tables have a multiple-level lookup to the reference table Region, because the DEPT table has a lookup relationship to the Region table. This multiple-level lookup means that the Region table can be used when querying these data tables.




In addition to the visual representation of these relationships, the diagram can be explored further to better understand the specific columns involved in each lookup. For example, you can hover over a connection arrow to see the validated column and the lookup column represented by that arrow.

Diagrams can be used by table administrators as a hands-on tool to investigate table relationships, or the diagrams can be saved and shared with other users as a way to communicate and document table relationships.

The data diagram feature is available in the Axiom Software Web Client only. Although you can see saved data diagrams in the Data Diagrams Library in the Axiom Software Excel Client and Web Client and open the diagrams from that location, all viewing and editing of diagrams must be performed in the Web Client.

## Managing data diagrams

Using the Data Diagrams area of the Web Client Table Manager, you can create, edit, and view data diagrams.

**To access diagrams in the Web Client Table Manager:**

1. Go to the Table Manager area of the Web Client.

   **Example On-Premise URL**

   `http://ServerName/Axiom/TableManager`

   Where *ServerName* is the name of the Axiom Application Server, and Axiom is the default name of the virtual directory.

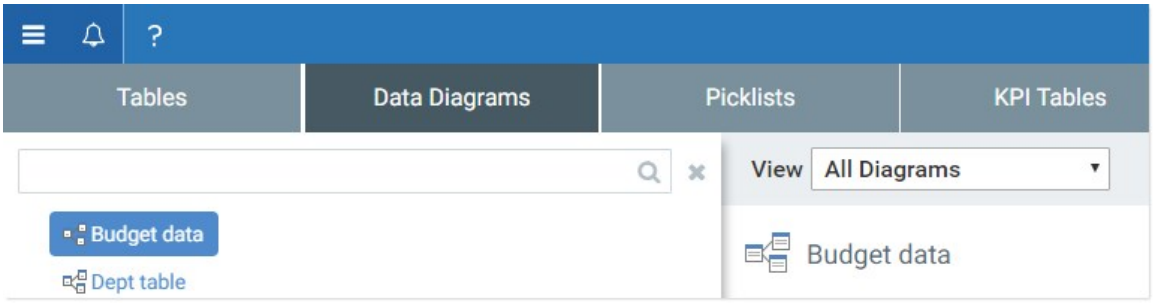
   **Example Cloud System URL**

   `https://CustomerName.axiom.cloud/TableManager`

   Where *CustomerName* is the name of your cloud service system.

   Alternatively, you can go to the Axiom Software launch page and click the **Table Manager** icon.

2. Select the **Data Diagrams** tab.




The left-hand pane displays the list of data diagrams in your system that you have rights to see. You can view this list using the following **View** options:

- **All Diagrams**: The list displays diagrams in alphabetical order.
- **By Diagram Folders**: The list displays diagrams in a treeview by folders.

If you click on a folder, then the diagrams in that folder display in the right-hand pane. You can toggle this display between Icon View and Detail View.

You can also use the search box at the top of the left-hand pane to find a diagram by name. To clear the search results and return to the full list, click the X icon to the right of the search box.

You can select a diagram in the left-hand pane to see the diagram rendered in the right-hand pane.

## Creating data diagrams

In order to create a data diagram, you must have a place to save it. This means you must have **Read/Write** access to a folder in the Data Diagrams Library, or access to the My Documents folder.

1. At the top right of the Data Diagrams page, click **New Diagram**.
2. In the New Data Diagram page, create the diagram by dragging and dropping tables from the left-hand treeview into the right-hand canvas area. Adjust the format and alignment of the tables as desired.

   For more information about the features available to design the data diagram, see Using the data diagram editor.
3. If you want to save the diagram for future use, click **Save**. In the **Save As** dialog:
   - Select the folder where you want to save the diagram. You can save the diagram to any folder in the Data Diagrams Library where you have read/write permission. If you have access to the My Documents folder, you can also save it there.
   - In the **File name** field, type the file name. The file will be saved as an ADD file type (Axiom Data Diagram).
   - Click **Save**.

The data diagram is saved to the designated folder.

### Editing data diagrams

In order to edit a data diagram, you must have read/write permission to the file.

1. In the Data Diagrams page, select the diagram in the left-hand pane.

   You can change the **View** options or use the **Search** box to help find the diagram that you are looking for.

2. The diagram opens in the right-hand pane as read-only. To open the diagram editor, click the pencil icon at the top right of the page.

   > **TIP:** You can also open diagrams for editing from within the Desktop Client. Use the Explorer task pane or the Axiom Explorer dialog to find the diagram, then double-click it to open. The diagram opens in the Web Client, just as if you browsed to it within the Web Client Table Manager and opened it for editing.

3. Edit the diagram as desired. For more information about the features available to design the data diagram, see Using the data diagram editor.

4. Click **Save**.

### Deleting data diagrams

In order to delete a data diagram, you must have read/write permission to the parent folder. Data diagrams cannot be deleted in the Web Client; they must be deleted in the Excel Client or Windows Client.

1. On the **Axiom** tab of the Excel Client or Windows Client, in the **Administration** group, click **Manage > Axiom Explorer**.

   > **TIP:** You can also use the Explorer task pane to delete a data diagram.

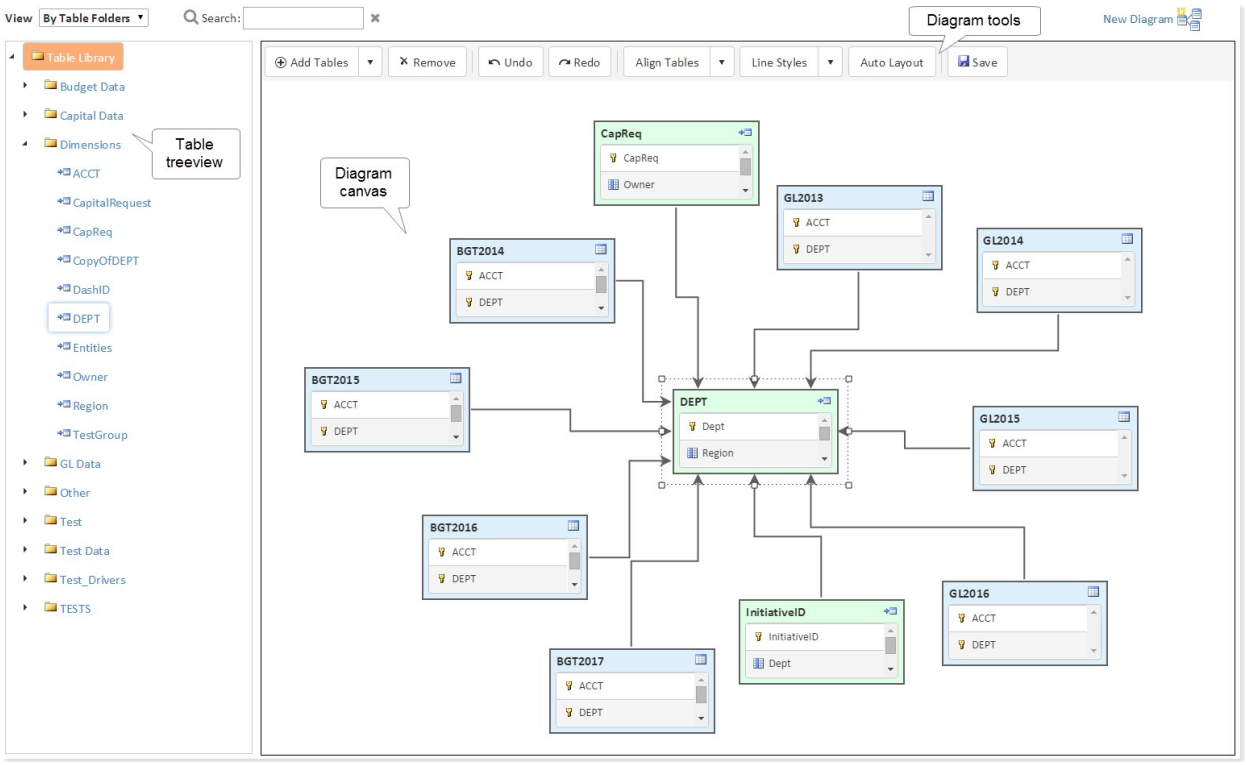
2. Navigate to the **Data Diagrams Library**, and then locate the diagram that you want to delete.

3. Right-click the diagram and then select **Delete**.

## Using the data diagram editor

The data diagram editor provides the tools necessary to create or edit a data diagram. When you click **New Diagram** or **Edit Diagram** in the **Table Manager** header, the diagram opens in the editor.

The left-hand side of the editor displays the available tables in a treeview. The right-hand side of the editor is the diagram canvas. To start a new diagram, drag and drop a table from the treeview onto the canvas. Once a table has been placed on the canvas, you can use various features to add new tables and format the diagram.




*Data diagram editor*

## Adding tables to the diagram

You can add tables to the diagram canvas by dragging and dropping tables from the treeview, or by using the **Add Tables** button.

When using the treeview, you can drag and drop tables one at a time. To find a table, you can:

- Browse the treeview by table, folder, or table type. Use the **View** list in the header to change the display of the treeview.
- Use the **Search** box in the header. The Search box finds tables based on table name. For example, enter `BGT` into the Search box to find tables with "BGT" somewhere in the table name.

The **Add Tables** button adds tables that are related to an existing table on the diagram canvas. Select the table for which you want to add related tables, then click the down arrow to the right of the **Add Tables** button. Select one of the following options:

- **Tables referenced by *SelectedTable***: Adds tables that the selected table looks up to. This shows you the lookup tables for validated columns in the selected table. This option is available for data tables or reference tables with defined lookups.

- **Tables that reference *SelectedTable***: Adds tables that look up to the selected table. This shows you which tables use the selected table as the lookup source for a validated column. This option is only available for reference tables, because only reference tables can serve as the lookup column for a validated column.
- **Add all tables related to *SelectedTable***: Adds tables that the selected table looks up to, and tables that look up to the selected table (as if you selected both of the other options). This option is only available for reference tables, because only reference tables can both be a lookup and have lookups.

Whether you use the treeview or the Add Tables button, the available tables are limited by the tables that you have security permission to see. You must have at least read access to a table in order for the table to be visible in the treeview or added by Add Tables. Therefore users who do not have access to all tables may not be able to create a diagram that encompasses all relationships for a given table. (However, when viewing a saved diagram, users can see any table that was already placed on the diagram.)

For example, imagine that you start a diagram by dragging and dropping the DEPT reference table to the canvas. You can add any additional tables by dragging and dropping them from the treeview, or you can specifically add tables that are related to the DEPT table as follows:

- If you want to see which tables use DEPT as a lookup table, then you can click **Add Tables > Tables that reference DEPT**. This will add tables such as GL2018 and Plan2018 to the diagram, assuming these tables have a Dept column that is validated against `DEPT.Dept`.
- If you want to see which tables DEPT looks up to, then you can click **Add Tables > Tables referenced by DEPT**. This will add tables such as Region to the diagram, assuming the Region column in the DEPT table is validated against `Region.Region`.
- If you want to see all tables with relationships to the DEPT table, then you can click **Add Tables > Add all tables related to DEPT**. This adds all tables described in the previous two bullets.

To remove a table from the diagram canvas, select the table and then click **Remove**. You can select and remove multiple tables at a time by using the CTRL key.

### Formatting table boxes in the diagram

In the diagram, tables display as boxes with the table name in the top left corner, and the table classification indicated by the icon in the top right corner. You can adjust the size of the table box by selecting the table and then dragging a selection handle until the table box is the desired size.

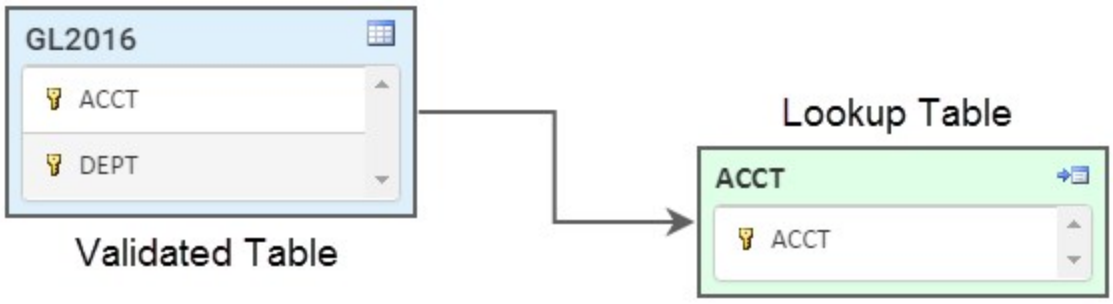
The inside of each table box details the key columns of the table, as well as any columns that look up to other tables. All other columns in the table are not shown. By default, reference tables and document reference tables are sized to show the single key column only, and data tables are sized to show two key columns. You can resize the table box to be taller if you want to show all included columns in the diagram. If the table box is too small to see all of the included columns, you can use the scroll bar to scroll the column list (note that this is difficult to do in the editor because it selects the table as if you want to move it, but it works when viewing a saved diagram as read-only).

The color of the box indicates the table classification. The color cannot be changed.

- **Blue**: Data table
- **Green**: Reference table
- **Orange**: Document reference table

### Formatting relationship lines between tables

Whenever a table is placed on the diagram canvas, relationship lines are automatically drawn between that table and any other table on the canvas that the table has a relationship with. The line flows from the validated table to the lookup table, with an arrowhead pointing to the lookup table.




You can hover your cursor over the line to see the names of the two columns that make up the connection.

Relationship lines are automatically anchored on both table boxes:

- The anchor points are automatically determined based on the relative positions of the connected boxes; so as you move a table, the anchor points may change. By default, lines are anchored in the center of the side that the line is connected to.
- If a table has multiple relationship arrows anchoring to the same side of the table box, then the anchor points are equally distributed along the side.
- It is not possible to manually adjust anchor points, but you can adjust the positioning of the affected tables to see if there is an automatic configuration that you prefer.

Relationship lines can be straight or cascading. Cascading lines have right angles (the previous example shows a cascading line).

**To change the line style for the diagram:**

1. Click the down arrow to the right of the **Line Styles** button.
2. Select **Cascading** or **Straight**. Only one option is available at a time; the currently applied style is grayed out.

The line style applies to all lines in the diagram. It is not possible to mix line styles in the same diagram.

**NOTE:** Document reference tables cannot link to other tables and therefore will never have relationship lines on a diagram. However, you can still add these tables to diagrams because they may be functionally related to other tables even though they are not technically related. For example, you may want to create a diagram that shows all tables used by file group Budget 2019, which would include one or more document reference tables (driver tables).

### Positioning tables on the diagram

The diagram editor has various features to assist you in positioning tables on the diagram canvas:

- **Moving tables**: To move a table, select the table and then drag and drop it to the desired location on the diagram canvas. You can also select multiple tables (using CTRL or Shift) and then move those tables as a group.
- **Aligning tables**: To align multiple tables, first select the table that you want to use as the "anchor" for the alignment. Then, use the CTRL or Shift key to select one or more additional tables that you want to align with the first table. Once all tables are selected, click the down arrow to the right of the **Align** button, and then choose the alignment point: **Left**, **Right**, **Top**, **Bottom**, **Center Horizontally**, or **Center Vertically**.

  For example, if you click **Align Tables > Left**, then all selected tables will be aligned with the left side of the "anchor" table.
- **Automatic layout**: Click **Auto Layout** to automatically distribute tables on the canvas, using an internal algorithm. Once the auto-layout has been applied once, continuing to click the button will make small adjustments to the layout. If a table is added, removed, or moved, clicking the button will re-distribute the layout based on the current number of tables and relative positions of each.

If you make a layout change and then decide that you do not like it, you can use the **Undo** button to restore the diagram to its prior state.

You can zoom the canvas and pan the diagram contents on the canvas, but these changes are not saved in the diagram. When a saved diagram is opened, it starts at the default zoom and pan position. For more information, see Viewing data diagrams.

## Viewing data diagrams

Data diagrams can be shared with users to document and communicate table relationships in Axiom Software. If you have access to a diagram, you can open the diagram and explore the relationships presented in the diagram.

For example, report writers may be given a diagram of the tables they are reporting on, so that they understand how the tables are joined and which lookup tables can be used to filter or sum the query.

In order to view a data diagram, you must have at least read-only access to the diagram, as defined in security.

Diagrams can be opened from either the Web Client or the Desktop Client. Regardless of where the file is opened, the diagram always opens in the Web Client for viewing.

### Opening a data diagram using the Desktop Client

You can open a data diagram from the Data Diagrams Library in the Desktop Client (Excel Client or Windows Client).

1. On the **Axiom** tab of the Desktop Client, in the **Administration** group, click **Manage > Axiom Explorer**.

   **TIP:** You can also use the Explorer task pane to open a data diagram.

2. Navigate to the **Data Diagrams Library**, and then double-click the diagram that you want to open.

The diagram opens in the Web Client using your default browser.

Note that if you have read/write access to the diagram, the diagram opens in the editor by default. If you do not intend to edit the diagram, you may prefer to open it as read-only, so that you can explore the diagram without accidentally moving items around or making other changes. Currently, the only way to view a diagram as read-only when you have read/write permission is to view it from the Data Diagrams treeview in the Web Client.

### Opening a data diagram using the Web Client

You can open a data diagram from the Table Manager in the Web Client.

1. Go to the Table Manager area of the Web Client.

   **Example On-Premise URL**

   `http://ServerName/Axiom/TableManager`

   Where *ServerName* is the name of the Axiom Application Server, and Axiom is the default name of the virtual directory.

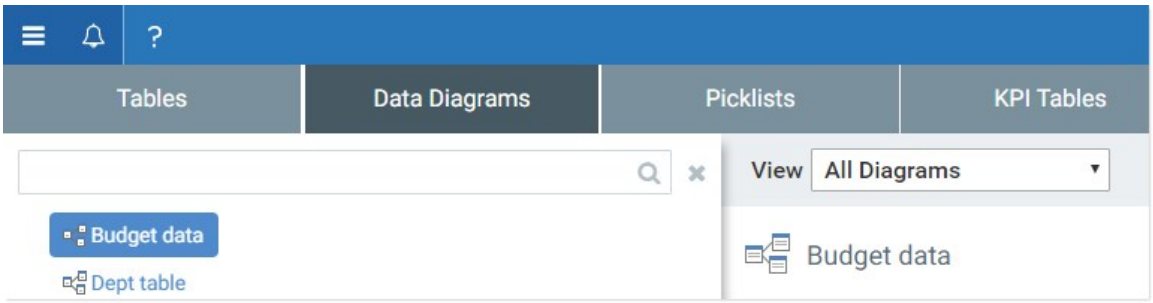
   **Example Cloud System URL**

   `https://CustomerName.axiom.cloud/TableManager`

   Where *CustomerName* is the name of your cloud service system.

   Alternatively, you can go to the Axiom Software launch page and click the **Table Manager** icon.

2. Select the **Data Diagrams** tab.



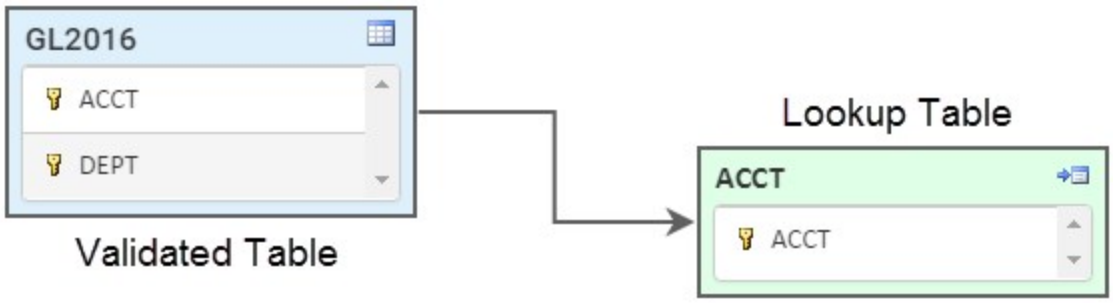

3. In left-hand pane, select the diagram that you want to view.

   You can change the **View** options or use the **Search** box to help find the diagram that you are looking for.

The diagram opens as read-only in the diagram viewer. You can explore the diagram in this view.

## Exploring a data diagram

Data diagrams consist of tables and the connections between those tables (via lookup column relationships). Tables with lookup relationships are connected with a line. An arrowhead points at the lookup table to indicate the direction of the relationship (TableX looks up to TableY).



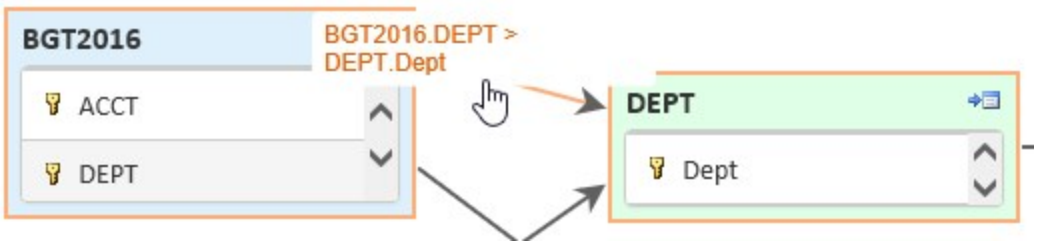

If two tables on the diagram have a lookup relationship, that relationship is always marked with a line. However, the tables in the diagram may have relationships with other tables that are not on the diagram. The diagram designer has chosen which tables to show.

Each table displays in a box with the table name in the top left corner, and an icon representing the table classification in the top right corner. The table classification is also represented by color: blue for data tables, green for reference tables, and orange for document reference tables. When viewing a diagram you can see all tables that were saved in the diagram, including tables that you do not have security permission to access.

Each table displays a list of its key columns and any other columns with lookup relationships. If the table box is too small to see all of the included columns, you can use the scroll bar to scroll the column list.

You can use the following features to further explore the data diagram:

- Hover your cursor over a relationship line to see the names of the two columns that make up the connection. The first column is the validated column and the second column is the lookup column. For example:




- While hovering your cursor over the diagram, use the scroll wheel on your mouse to zoom in or zoom out.
- To pan the diagram, hold the CTRL key and move your mouse in the direction that you want to pan.


# Dimensions and Hierarchies

Reference tables are often used to define the *dimensions* of your system. Dimensions are the levels at which planning and reporting activities are performed in your system—such as department and account.

Reference tables can have one or more column hierarchies, which define hierarchical and rollup relationships for dimensions and their groupings. For example, you can have a Geography hierarchy where departments roll up into regions, and regions roll up into countries. These hierarchies are used throughout the system to define data filters and to define drilling paths.

Using the Dimension Manager, you can create, edit, and delete column hierarchies, as well as edit the hierarchy assignments for members of the dimension.

## About hierarchies

Column hierarchies define grouping and rollup relationships for the data in Axiom Software. Hierarchies are used as follows:

- In the Filter Wizard, to assist in defining simple filters throughout the system, including the Quick Filter feature.
- For drill-down drilling, to specify the level of detail for drilling.

Hierarchies are not required, but it is recommended to create one or more hierarchies for common rollup relationships. In most cases it is much easier and intuitive for users to create filters and drill using hierarchies.

For example, you can set up hierarchies on the DEPT reference table for Geography (Country > Region > DEPT) and Management (VP > Manager > DEPT). Users can quickly identify the grouping level that they want to use for drilling or for a filter.

In Axiom Software hierarchies are defined by selecting multiple grouping columns to belong to the hierarchy and then specifying the order of those columns. The columns must already be defined in the table in order to add them to a hierarchy. The values in those columns then define the hierarchy assignment for each individual data element in the table. For example, Dept 25000 is assigned to country USA and region West based on the entries in the Country and Region columns of the Geography hierarchy.

Only reference tables can have column hierarchies. You can define any number of hierarchies for a reference table using any columns in the table. The same column can be used in multiple hierarchies.

### Hierarchy order

When you define a hierarchy, you add one or more columns and you specify an order for those columns. The column at the top is the highest level of the hierarchy, and the column at the bottom is the lowest level.

**NOTE:** You should always place the key column at the bottom of the hierarchy. This helps illustrate the complete rollup path from top to bottom, and makes it easier for the user to select the key column for drilling and filtering.

The column relationships in a hierarchy are not forced to be exclusive. For example, if you have a hierarchy where managers roll up into VPs, each individual manager can roll up into multiple VPs like so:

| DEPT | Manager | VP |
|---|---|---|
| 200 | Smith | Jones |
| 300 | Smith | McDougal |

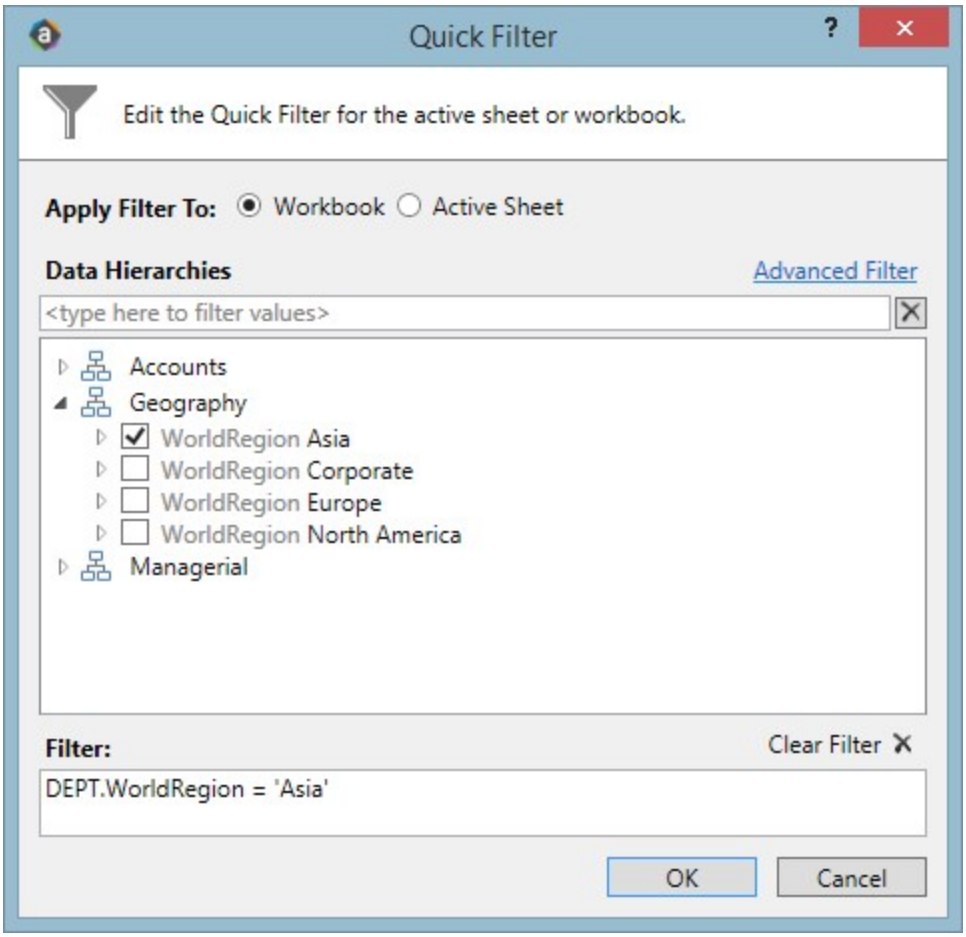
However, the order of the hierarchy is enforced—meaning that you can drill down to lower levels of a hierarchy, but you cannot drill "up" to higher levels.

Imagine that you have the following Geography hierarchy: Country > Region > DEPT. If the source report displays data at the Region level, then Country will not be available for drilling. You cannot drill "up" to a higher level, but in this case you can drill down to the DEPT level.

If the source report displays data at the DEPT level, then you cannot drill based on any DEPT hierarchy, because you are already at the lowest level of data for DEPT. However, you may be able to drill based on other key columns and their hierarchies.

### Hierarchy display names

When hierarchies are displayed in the filter view, the column name is displayed in front of the value as shown in the following screenshot. In this example, WorldRegion is a column in the hierarchy.




If desired, you can define an alternate display name for the column, to be used when the column name is displayed in a hierarchy. For example, you could define a display name of "World Region" or "Global Division." This might be most useful when the literal column name is not very user-friendly.

The display name is defined in the column properties of the table, using the property **Hierarchy Display Name**. You can define the hierarchy display name by editing the table properties, or when editing the hierarchy (using the **Rename** button).

The Hierarchy Display Name is used in the hierarchy view of the Filter Wizard, and when hierarchies are displayed for drill-down drilling. It is also used in the Data Explorer if the column is part of a data set.

### Excluding elements from hierarchy display

If necessary, you can define a filter for a hierarchy to control which dimension or grouping elements display in that hierarchy. A **Filter** field is available when creating or editing the hierarchy for the table.

This filter is for display purposes only—to simplify the interface for end users by only displaying relevant hierarchy selections. The hierarchy filter does *not* impact the data query. If a dimension or grouping has data and belongs to the selected hierarchy level, then that data will still be returned by the query, regardless of whether the dimension or grouping displays in the hierarchy.

For example, you might have certain "closed accounts" that you want to exclude from the hierarchy because they are no longer being actively used. You could exclude an account by defining a hierarchy filter such as `Acct.Acct<>4300`, or `Acct.Status='Active'` (assuming you are using a grouping column to designate active versus inactive accounts).

In the above example, account 4300 would not display in the hierarchy, and therefore would not be available for user selection. However, if the account has data, and belongs to a hierarchy grouping such as account category Overhead, then the account's data would still be included when the user selects Overhead from the hierarchy.

# Managing hierarchies

Each reference table can have one or more hierarchies to define grouping and rollup relationships for that dimension. You can manage hierarchies and their assignments using the Dimension Manager. To access the Dimension Manager:

- From the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Dimension Manager**.

This dialog lists all existing hierarchies in the system. You can change the **View by** setting to view the list either by table name or hierarchy name. You can also use the **Filter** box to search for a particular hierarchy.

If you select a hierarchy in the left-hand pane, you can view the column relationship for that hierarchy in the left hand pane. You can edit or delete the hierarchy by using the buttons in the toolbar, or you can edit the hierarchy assignments in the associated table by clicking the link to **Edit hierarchy assignments**.

This topic discusses how to create, edit, and delete hierarchies. For more information on editing hierarchy assignments, see Editing hierarchy assignments.

Only administrators and users with the **Administer Tables** permission can access the Dimension Manager.

**NOTES:**

- Picklist tables and KPI tables cannot have hierarchies.
- If a table has more than 50,000 rows, then the elements of the hierarchy will not display in the Filter Wizard or the Quick Filter dialog. The hierarchy name will still be present, but if expanded a message will display to the user explaining that the hierarchy has too many rows. You can adjust this limit using the **MaxHierarchyRowSize** system configuration setting, however, keep in mind that viewing hierarchies with large numbers of rows may cause out-of-memory errors on client machines.

## Creating a hierarchy

Hierarchies are based on columns in the reference table. The columns must already exist in order to create the hierarchy.

1. In the toolbar of the **Dimension Manager** dialog, click **New** ✚.

   **TIP:** If the table for which you want to create a hierarchy is already listed in the left-hand pane of the dialog, you can select that table and then click **Add to *TableName***. This will automatically select the table for the hierarchy, meaning you can skip the following table selection step.

2. In the **Select Table** dialog, select the reference table for which you want to create a hierarchy. To help find the table you are looking for, you can use the **View by** setting to change how the list of tables is displayed, and you can use the **Filter** box to find a particular table name.

   The **Hierarchy Editor** dialog opens.

3. In the **Hierarchy Name** box, type a name for the hierarchy. Each hierarchy name must be unique for the table.

   Use a name that will be intuitive to end users. For example, you might name the hierarchy "By Geography" or "By Region" if the hierarchy defines a geographical drilling relationship.

4. Optional. In the **Hierarchy Filter** box, define a filter to determine which items are displayed in the hierarchy. Use the Filter Wizard to build a filter.

   For example, you may have "closed accounts" that you want to exclude from the hierarchy because they are no longer being actively used. Remember that anything that matches the filter will be displayed, so if you want to exclude certain accounts, the filter must use not equals (such as `acct.acct<>4300`). Or you might maintain a grouping code for the account status, so the filter could be `acct.status='active'` (where closed accounts are flagged as "inactive").

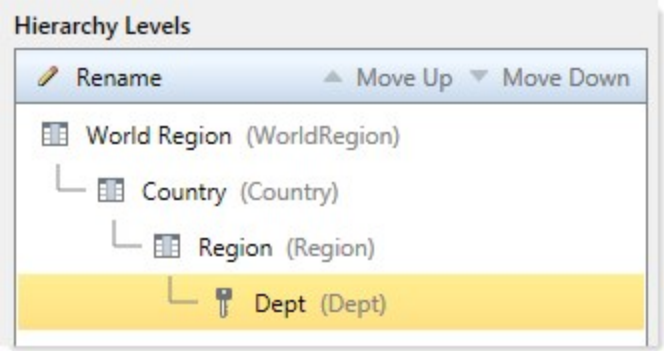
   **NOTE:** This is a display-only filter—it does not affect the data query. If a dimension or grouping element has associated data and belongs to a selected grouping, then that data will be included regardless of whether the element displays in the hierarchy.

5. Specify the columns to include in the hierarchy.

   - In the **Grouping Columns** box, select each column that you want to add to the hierarchy and then click **Add**. You must do this one column at a time.
   - In the **Hierarchy Levels** box, use the **Move Up** and **Move Down** buttons to specify the order of the columns in the hierarchy. The top column in the list is the highest level of the hierarchy, and the bottom column in the list is the lowest level.
   - If a column is added the hierarchy by mistake, you can select the column in the right-hand pane and click **Remove**.

**Example**

For a geographical hierarchy, the column list might appear like so:




For example, Dept 2000 might belong to the region "US West," which rolls up to the country "United States," which rolls up to the world region "North America." DEPT is the lowest level of the hierarchy, and WorldRegion is the highest.

**NOTE:** Although it is not required, it is recommended to add the table's key column to hierarchies for ease of drilling to the lowest level. If you add the key column (for example, DEPT), that column can only be the lowest level of the hierarchy.

6. Optional. For the columns in the hierarchy, you can define or edit the Hierarchy Display Name for that column. To do this, select a column name in the right-hand pane and then click **Rename**.

   If defined, the Hierarchy Display Name will be used when the hierarchy is displayed within the Filter Wizard or on the drill-down menu. It is also used when displaying column names in the Data Explorer. For example, if you have a column named "WorldRegion" (with no space), you can define a more user-friendly display name of "World Region" (with a space). If you edit the name here, the Hierarchy Display Name field for the column will be updated in the table properties.

7. Click **OK** to save.

The new hierarchy is now available for drilling and for use in the Filter Wizard.

## Editing a hierarchy

You can edit the hierarchy properties at any time. Hierarchy changes are immediately available to users.

**NOTE:** If you want to edit the individual hierarchy assignments in the table instead of the hierarchy definition, then you can either edit the table directly using Open Table in Spreadsheet, or you can use the Dimension Editor. For more information, see Editing hierarchy assignments.

1. In the left-hand pane of the Dimension Manager, select the hierarchy that you want to edit, and then click **Edit** .

   You can use the **View by** drop-down and the **Filter** box to help find the hierarchy that you are looking for.

2. In the **Hierarchy Editor** dialog, edit the hierarchy settings as desired, and then click **OK**.

### Deleting a hierarchy

You can delete a hierarchy at any time. Deleted hierarchies will no longer be available to users.

- In the left-hand pane of the Dimension Manager, select the hierarchy that you want to delete, and then click **Delete** .

  You can use the **View by** drop-down and the **Filter** box to help find the hierarchy that you are looking for.

# Editing hierarchy assignments

Using the Dimension Editor, you can edit the hierarchy assignments for the individual items in a table. For example, if a hierarchy has a structure of VP > Manager > Dept, then each individual department in the DEPT table must have an assigned VP name and an assigned Manager name to determine its place in the hierarchy. While these assignments can also be edited using Open Table in Spreadsheet, the Dimension Editor provides an alternate interface that presents the hierarchy assignments in a treeview and allows drag-and-drop editing.

**Example**

Using the Dimension Editor may be easier and more intuitive for making certain kinds of edits to hierarchy assignments. For example:

- Imagine that one of your VPs has left the company and is being replaced by a new VP. In Open Table in Spreadsheet, you would need to find all instances of the old name in the VP column and then replace it with the new name. In the Dimension Editor, you can simply rename the appropriate VP node in the hierarchy to reflect the new name. When you save the change in the Dimension Editor, Axiom Software will update the VP column in the DEPT table as appropriate for this change.
- Imagine that one of your VP positions is being reorganized so that half of the managers who used to report to that VP are now reporting to a new VP position. In the Dimension Editor, you can first create a new VP node for the new position. Then you can expand the VP node for the reorganized position, and then drag-and-drop items as appropriate from that VP to the new VP. When you save the change in the Dimension Editor, Axiom Software will update the VP column in the DEPT table for the changed assignments. Although you could also make this change in Open Table in Spreadsheet, it is probably easier to visualize the changes in the Dimension Editor and it requires fewer steps to make all changes.

The Dimension Editor is accessible from the Dimension Manager:

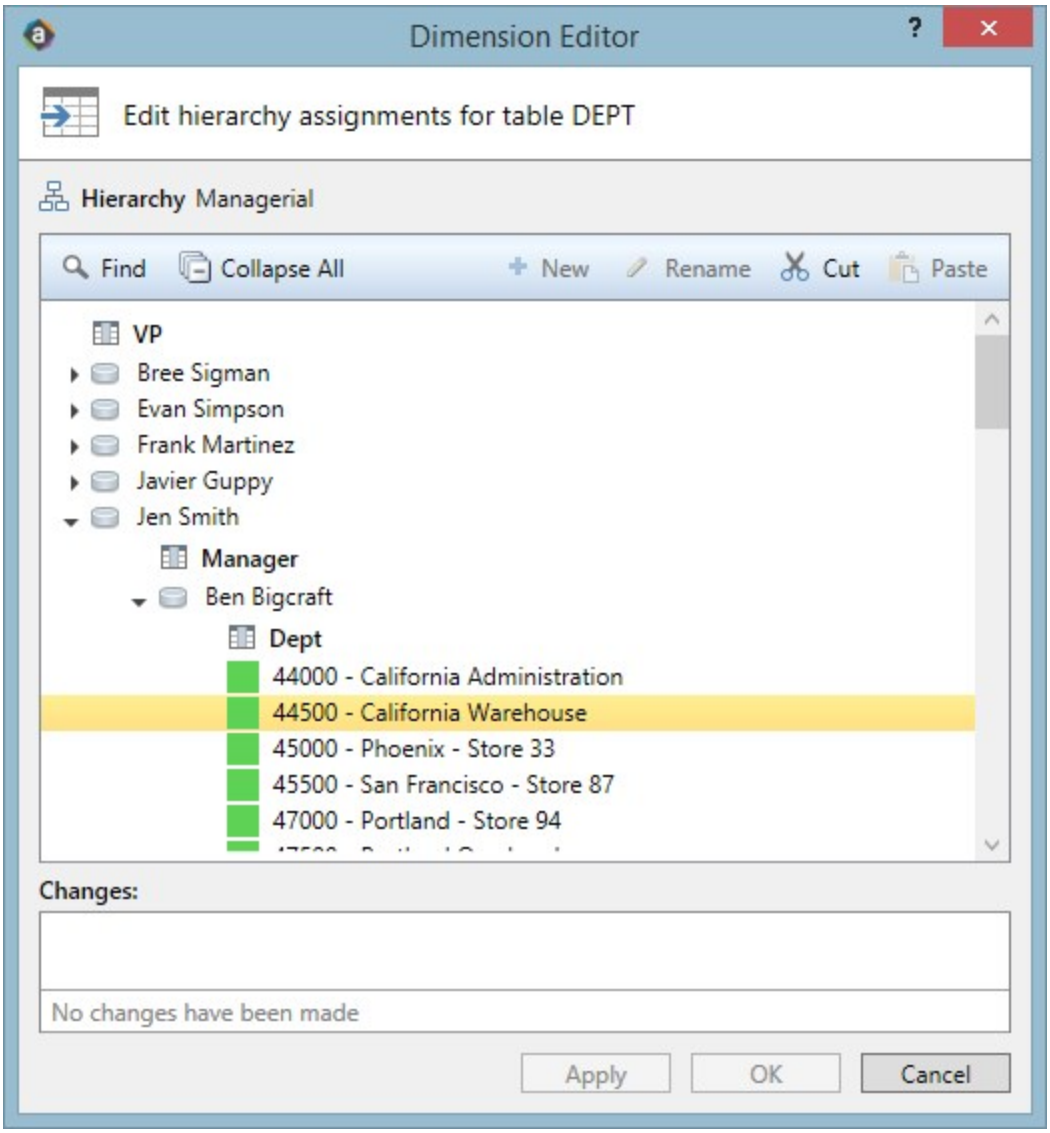
1. From the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Dimension Manager**.
2. In the left-hand pane of the Dimension Manager, select the hierarchy where you want to edit the hierarchy assignments. You can use the **View by** drop-down and the **Filter** box to help find the hierarchy that you are looking for.
3. In the right-hand pane of the Dimension Manager, click **Edit hierarchy assignments**.

Only administrators and users with the **Administer Tables** permission can access the Dimension Manager.

**NOTE:** Because changes made in the Dimension Editor will be applied to the underlying table when saved, opening the Dimension Editor locks the table to the current user (in the same way that opening the table using Open Table in Spreadsheet locks the table). Similarly, if the table is already locked to another user when you attempt to open the Dimension Editor, then you will be informed that the table is locked and given the option to open the Dimension Editor read-only (to view assignments only).

## Dimension Editor overview

The Dimension Editor displays the hierarchy assignments from the underlying table in a treeview. In the following example, the hierarchy starts by displaying the top-level column of VP and the entries in that column (in this case, various VP names). Expanding one of the VP nodes displays the entries for the next level column of Manager that are assigned to that particular VP. Lastly, expanding a Manager node displays the lowest level of the hierarchy, the individual departments assigned to that VP > Manager path.




Another way to visualize the hierarchy assignments is to look at the lowest level of the hierarchy and trace its path upwards to see the column entries for a particular department. If you look at Dept 44000 in the Dimension Editor, you can see that the DEPT table for that department would look as follows (for the columns included in this hierarchy):

| Dept | Manager | VP |
|---|---|---|
| 44000 | Ben Bigcraft | Jen Smith |

As you work in the Dimension Editor, if you change anything that would impact the hierarchy column assignments for a particular dimension element, the corresponding entries in the underlying table will be updated when you save the changes. The following examples illustrate what will be updated in the DEPT table when changes are saved:

- Imagine that Ben Bigcraft leaves the company and Mike Jones is hired as his replacement. If you rename the **Ben Bigcraft** node to **Mike Jones**, then the Manager column of the DEPT table will be changed to "Mike Jones" for every department that is currently assigned as follows: VP Jen Smith > Manager Ben Bigcraft. For example, Dept 44000 will look like this after changes are saved:

  | Dept | Manager | VP |
  |---|---|---|
  | 44000 | Mike Jones | Jen Smith |

  Keep in mind that Ben Bigcraft may also be assigned as a Manager underneath different VPs—the hierarchy entries are not required to be exclusive. To be sure that you have changed all applicable entries, you can use the **Find** functionality to search for "Ben Bigcraft".

- Imagine that the company is reorganized and now Ben Bigcraft reports to a different VP, but the departments he is responsible for remain the same. If you drag and drop the **Ben Bigcraft** node underneath VP **Bree Sigman**, then the VP column of the DEPT table will be changed to "Bree Sigman" for every department that is currently assigned as follows: VP Jen Smith > Manager Ben Bigcraft. For example, Dept 44000 will look like this after changes are saved:

  | Dept | Manager | VP |
  |---|---|---|
  | 44000 | Ben Bigcraft | Bree Sigman |

  Nodes can only be dragged and dropped to the same level in the hierarchy. In this example, Manager nodes can only be dragged and dropped to the Manager level at other locations in the hierarchy. They cannot be moved to a different level such as VP, because then the departments would not have entries for all levels in the hierarchy.

- Imagine that the company is reorganized and now department 44000 reports to a different VP > Manager path. If you drag and drop department **44000** so that it is now located under VP **Frank Martinez** and Manager **Mark Wahl**, then the VP and Manager columns of the DEPT table will be changed for Dept 44000 as follows:

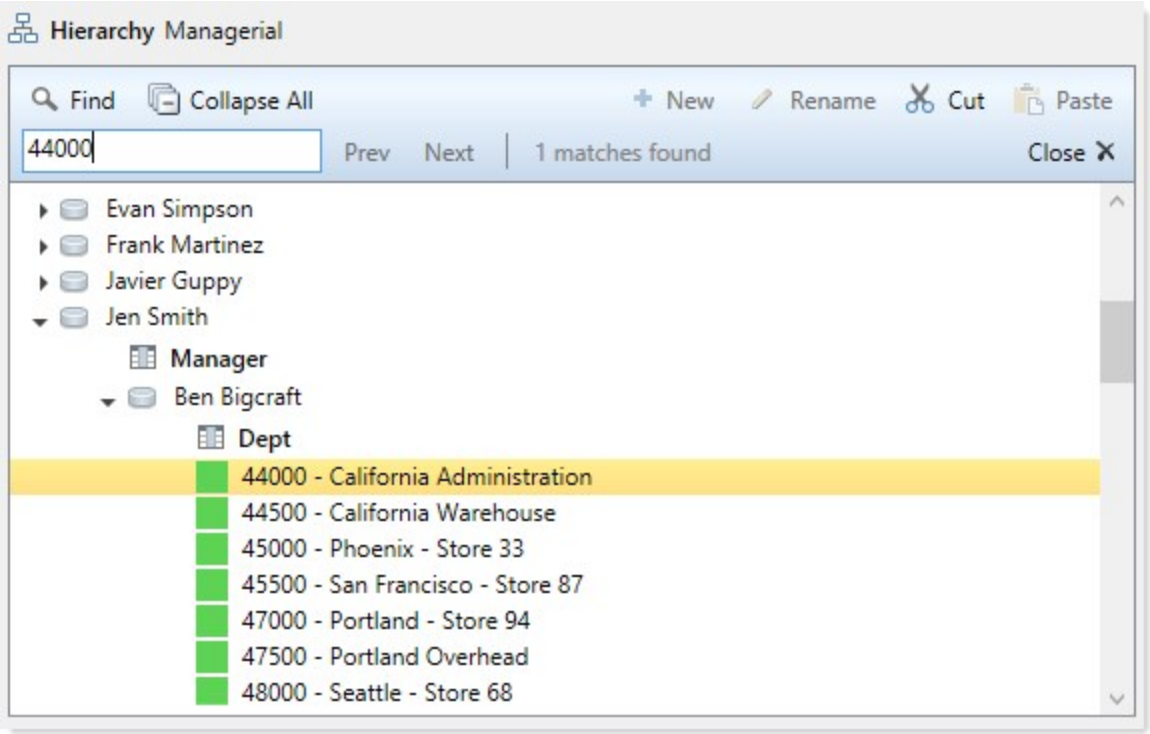
  | Dept | Manager | VP |
  |---|---|---|
  | 44000 | Mark Wahl | Frank Martinez |

These are just a few examples of the types of changes you can make in the Dimension Editor, and the corresponding impact on the underlying table. You can also make changes such as creating new entries (or "nodes") within a column and assigning items to those nodes, and even deleting dimension elements from the table.

## Moving items to new locations in the hierarchy

Using the Dimension Editor, you can edit the hierarchy assignments by dragging and dropping dimension elements and hierarchy nodes to different locations in the hierarchy treeview, at the same level. For example, if a node is at the second level of a hierarchy, you can drag and drop it underneath a different first-level node, but you cannot "promote" it to the first level or "demote" it to the third level. All items underneath the node will be moved along with it to the new location.

To locate a hierarchy node or individual dimension element in the hierarchy, click **Find** and then type in the value that you are looking for. If the value matches several items, you can use **Prev** and **Next** to move to the next or previous item. When you are done searching, click **Close** to hide the Find toolbar.




As an alternative to dragging and dropping, you can also use **Cut** and **Paste**. This may be useful for situations where you are not sure of the new location in the hierarchy so you need to search for it. For example, imagine you want to move Dept 44000 to Manager Susan Thompson but you are not sure where that manager is located in the hierarchy. You can first cut Dept 44000 from its current location. Then you can use Find to locate Susan Thompson. Once located, you can use Paste to place Dept 44000 in its new location.

**NOTE:** Hierarchy assignments are not required to be exclusive in Axiom Software. For example, if you have a manager value of Jane Doe, that manager might roll up to VP Susan Thompson for some departments, and VP Frank Martinez for other departments. If you move the Jane Doe node that is located under VP Susan Thompson to a different VP, only those departments will be assigned to the new VP; the Jane Doe assignments under VP Frank Martinez will be unchanged. If you are unsure whether a value exists at multiple locations of the hierarchy, you can search for that value to check.

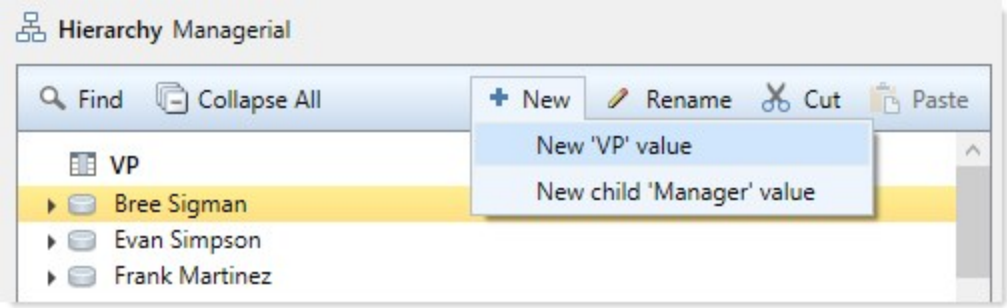
### Creating new hierarchy values

Using the Dimension Editor, you can create new values for hierarchy columns. For example, if VP is a column in the hierarchy, you can create a new VP value.

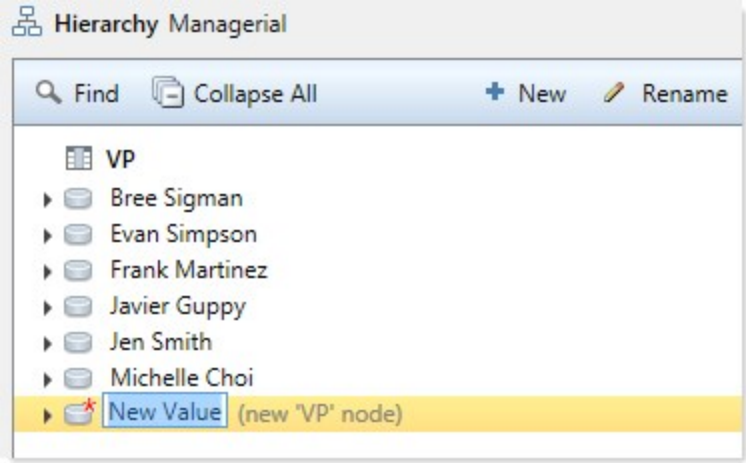
Depending on the situation, you might create a new value by renaming an existing value, or by adding an entirely new value. For example, if a VP leaves the company and is being replaced by a new VP, you can simply rename the existing VP value to the new value. This will adjust the hierarchy so that everything that used to be assigned to the old VP is now assigned to the new VP.

On the other hand, if you are adding a new VP position and all the existing VP names are still valid, you would want to add an entirely new value to the VP column. Once you have created the new value, you would then move existing items in the hierarchy to the new VP name as appropriate.

- **Adding a new value:** To add a new value, highlight an item in the hierarchy that is at the same level where you want to add an item. Click **New** ✚ and select **New *ColumnName* value**, where *ColumnName* is the name of the level for which you are adding a new item. You also have the option of adding items for levels underneath the current level.



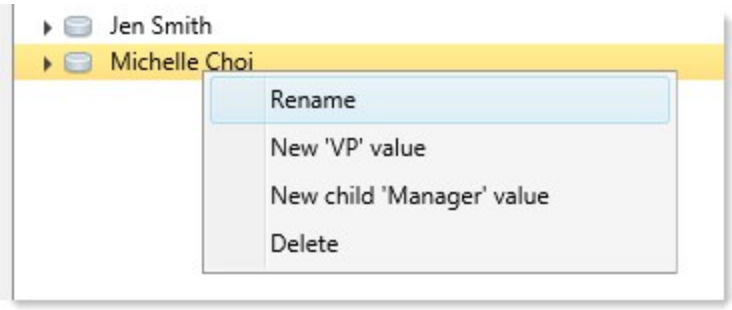
This adds a new node at the current level. You can type in the name for the new node, and then drag and drop items to this node as appropriate.




**NOTE:** New values are only saved if at least one dimension element is assigned to that value. The values displayed in the Dimension Editor represent entries in the hierarchy columns within the underlying table. If a value does not have a place to be written in the table, it will not save.

You can also add new values by using the right-click menu in the hierarchy treeview.

- **Renaming an existing value:** To rename an existing value, right-click the item that you want to rename, and then select **Rename**.




The current value becomes editable and you can type in the new value. All assignments *in that current path of the hierarchy* will be updated to belong to the new value.

**NOTE:** Hierarchy assignments are not required to be exclusive in Axiom Software. For example, if you have a manager value of Jane Doe, that manager might roll up to VP Susan Thompson for some departments, and VP Frank Martinez for other departments. If you rename the Jane Doe node that is located under VP Susan Thompson, then only those departments will be assigned to the new name; the Jane Doe assignments under VP Frank Martinez will be unchanged. If you are unsure whether a value exists at multiple locations of the hierarchy, you can search for that value to check.

Only hierarchy column values can be added or renamed in the Dimension Editor—meaning, values in grouping columns assigned to the hierarchy such as VP or Manager. The individual dimension values (such as departments) cannot be added or renamed using this dialog.

### Deleting items from the hierarchy

Using the Dimension Editor, you can delete hierarchy nodes or individual dimension elements.

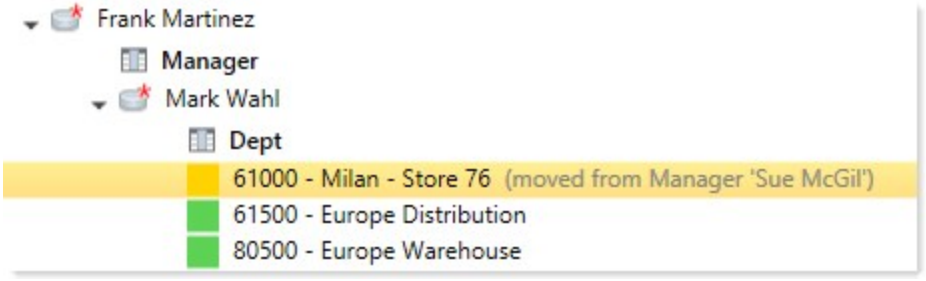
**IMPORTANT:** Deleting an item from the hierarchy deletes the corresponding dimension elements (such as departments) from the underlying table (such as DEPT). If you delete a hierarchy node, all dimension elements under that node will be deleted. This action should be used with extreme caution and only when you are absolutely certain that you want to delete these dimension elements.

To delete an item, right-click it and select **Delete**. A warning message lists the number of rows that will be deleted from the underlying table and prompts you to confirm that you want to delete the item. If you click OK to this warning message, keep in mind that you can still opt to Undo the change or Cancel out of the dialog without saving. However, once you save the change (by clicking Apply or OK for the Dimension Editor), then the dimension elements will be irrevocably deleted from the table if they are eligible to be deleted. Dimension elements will not be deleted if they are referenced by other tables.

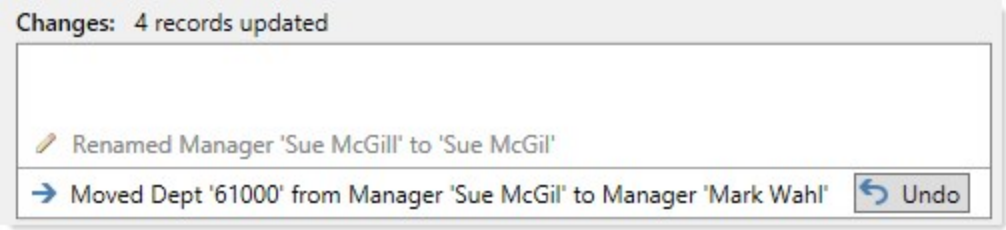
### Reviewing and undoing changes before saving

As you make changes in the Dimension Editor, these changes are tracked as follows:

- In the treeview, if a change has been made to a hierarchy node or an individual dimension element, a description of the change displays next to the item in parentheses. If the changed item is a dimension element, the icon changes from green to yellow to indicate the changed status. Also, if a change has occurred anywhere underneath a hierarchy node, a red asterisk displays on the node icon.



- A running list of all unsaved changes is displayed at the bottom of the Dimension Editor dialog, in the **Changes** section. The change listed at the bottom is the most recent change. The change directly above it is the next most recent change, and so on. Additionally, the running total displayed over the Changes box shows you how many records in the underlying table will be updated as a result of these changes.




Any change made in the Dimension Editor can be reversed ("undone") as long as it has not yet been saved. Changes must be undone one at a time, starting with the most recent change. Once changes have been saved by clicking Apply or OK, they cannot be undone.

To undo the most recent change, click **Undo**. Once that change has been reversed, the Undo button is now available for the next change in the list. You can continue reversing changes until you have returned the hierarchy assignments to the desired state. Of course, if you want to reverse all changes that you have made in the Dimension Editor, you can simply click **Cancel** to exit the dialog without saving.

### Saving the changes to the table

When you are finished making changes to the hierarchy assignments, you can save the changes to the underlying table for the hierarchy. The table that will be affected by the changes is listed at the top of the dialog.

Click **Apply** to save the changes and leave the Dimension Editor open, or click **OK** if you want to save the changes and are finished editing.

All changes listed in the **Changes** box will be applied to the table. After the save is complete, the treeview in the Dimension Editor will be updated to reflect the current state of the hierarchy assignments.


# Drill-Through Definitions

If you want users to be able to "drill through" the data in one table to the detailed sub-GL data in another table, then you must set up drill-through definitions.

A drill-through definition does the following:

- Specifies the target table that holds the detailed data
- Specifies the number of drilling periods (columns) for the definition
- Defines the filters to determine the appropriate data to return when drilling a specific period
- Maps the drilling periods to one or more column sequences in data tables

For example, if a report is displaying actuals data from the GL2018 table, you want users to be able to "drill through" to the transactions data held in the GLDetail table. To do this, you need to set up a drill-through definition for the GLDetail table that maps the data in that table to the appropriate columns in the GL2018 table.

This section discusses how to manage drill-through definitions for a system. For more information on how users use the definitions to drill data, see the *User Guide*. For more information on how drill-through drilling works and file design considerations, see the *Axiom File Setup Guide*.

## About drill-through definitions

Drill-through definitions map the data in a specified target table to a column sequence in another data table. This allows users to "drill through" column data in the data table to the detailed data in the target table.

For example, imagine a report that displays actuals data from the GL2018 table. The user wants to see the detailed transactional data for the January 2018 value (column M1 from the GL2018 table). When the user selects to drill through that value, Axiom Software needs to know:

**What table contains the associated detailed data for this value?**

For example, you may have a table named GLDetail that holds the detailed transactional data for current year's actuals. Axiom Software needs to know that when you drill through data in the GL2018 table, the target table for this drill is GLDetail.

You may have one table that always holds the detailed data for the current year, or you may have multiple tables to hold detailed data per year (for example, GLDetail2018, GLDetail2017). You may also have multiple tables to hold different types of detailed data (for example, GLDetail, APDetail, RevDetail).

**What filter needs to be applied to the target table to result in the associated detailed data for this value?**

For actuals data and planning data, time is represented in the data table by using *column sequences*. For example, the GL2018 table may have a column sequence of M1-M12, representing 12 months of actuals data (aliases CYA1-CYA12).

However, data tables that hold detailed data typically define time as a "dimension," using a key column such as YearMo (year and month of the data). In order to drill from an actuals / plan data table to a detailed data table, Axiom Software needs to be able to map each column in the column sequence to the appropriate keys of the YearMo column.

For example, the first column of the sequence, M1, corresponds to YearMo values of 201801. This is represented as a filter applied to the detailed table: `YearMo=201801`.

## Components of a drill-through definition

Each drill-through definition consists of the following:

- The number of periods (columns) in the definition. This determines the number of filters to define, and the valid column sequences that you can map to. If you want to map to a 12 column sequence, then the definition must be 12 columns.
- The target table for the definition. This is the table that data will be queried from when drilling through data.
- The column filters to be applied when drilling. These column filters will be matched up to the column sequences when drilling. For example, the first column in the sequence matches up with the first column filter, and so on.
- The column sequences to define the source data eligible for drilling. In most cases each drill-through definition will be associated with only one column sequence. For example, if the target table GLDetail contains transaction data for the current year, then it will probably only map to the sequence M1-M2 of the GL2018 table. However, if it makes sense for other sequences in the same table or in other tables to map to the same set of data, then you can do so.
- The columns of the target table to display when drilling.

### Requirements of the target table

There are no specific requirements for the structure of the target table, though typically this table will share certain key columns with the source tables for the mapped sequences, such as Acct and Dept.

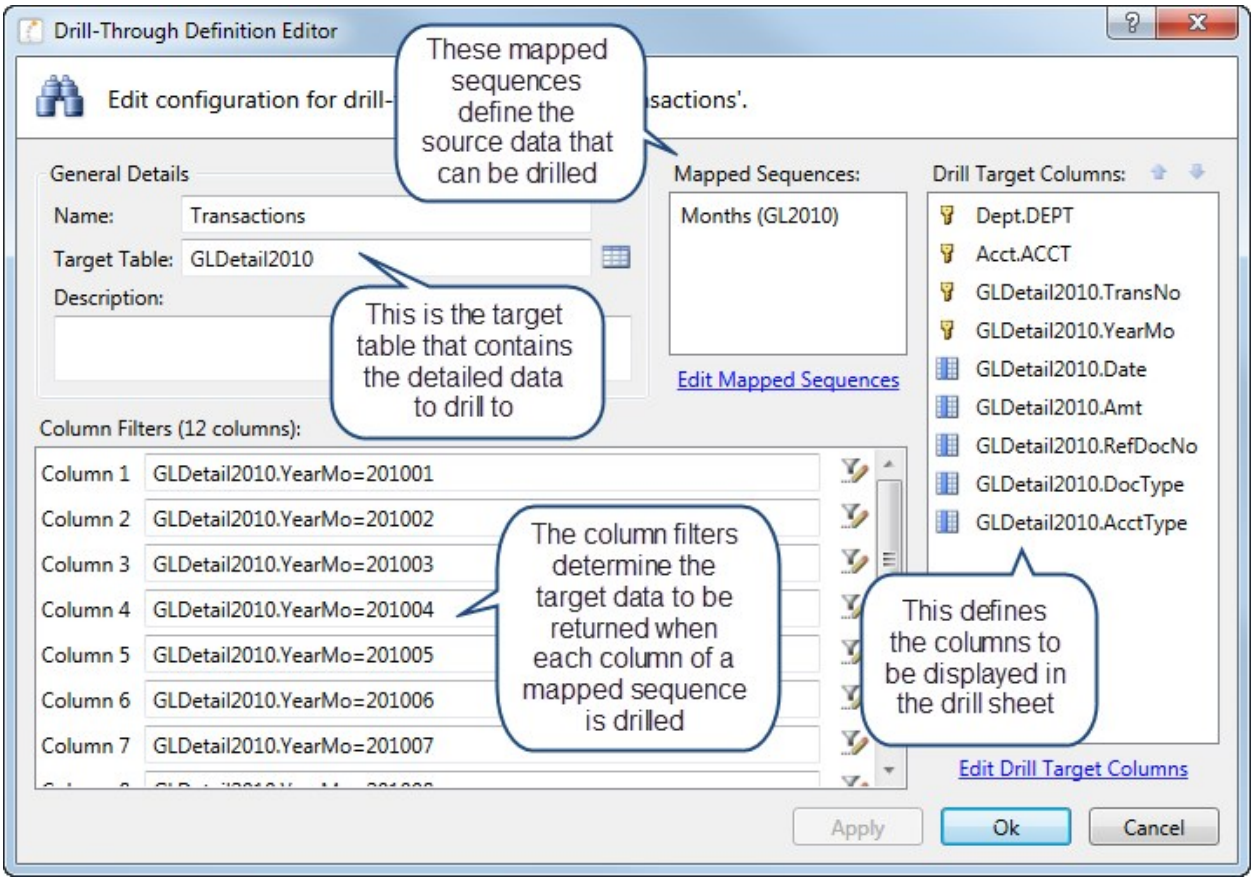
What determines a valid drill-through context has more to do with the setup of the current report rather than the structure of the source and target tables. When a drill through is initiated from a report, a filter is applied to the drill-through table that includes the following:

- The dimensionality of the current row being drilled (determined by the sum by of the Axiom query or the filter for the GetData function)
- Any sheet filters defined for the sheet
- Any filters applied to the Axiom query, such as the data filter or column filters (if drilling an Axiom query)

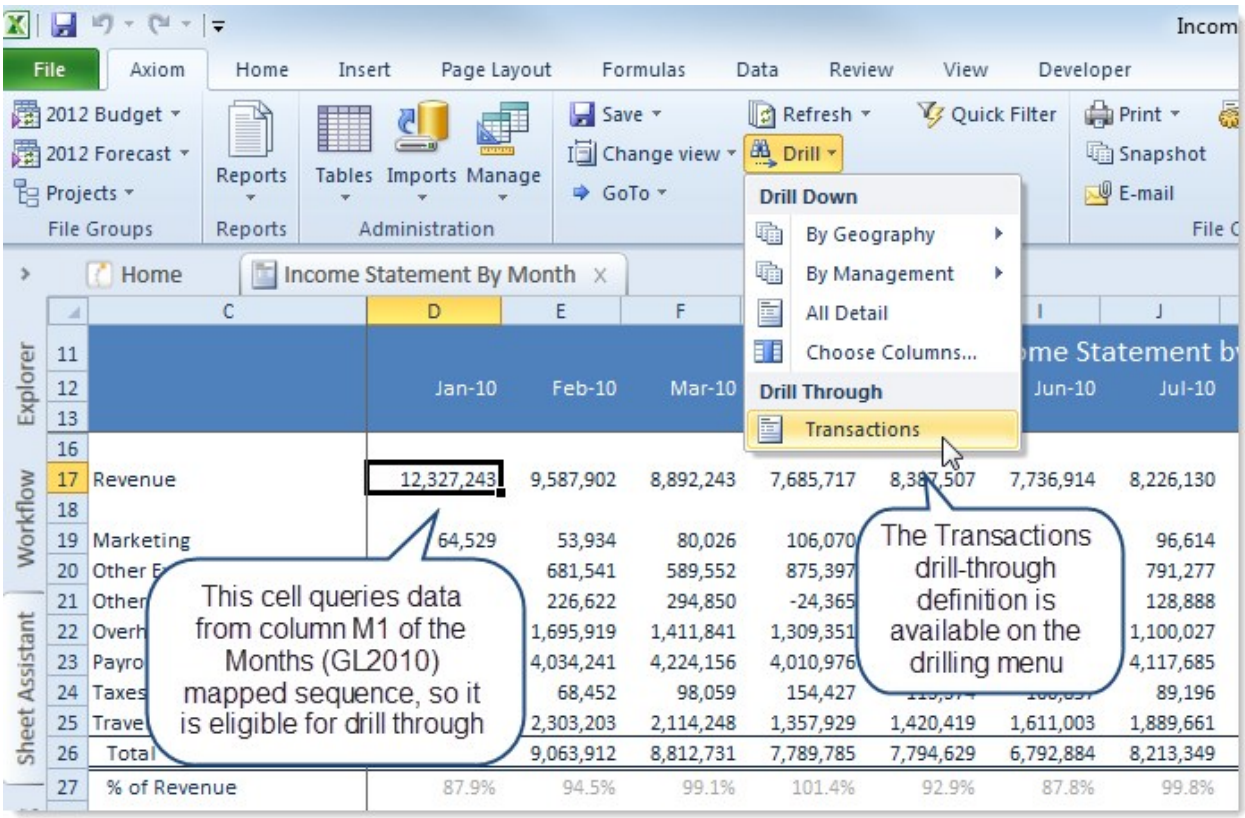
If any of these areas use columns that cannot be applied as filters to the target table, then the drill will fail with an invalid filter context. Therefore, the more similarities between the source and target tables (especially having common key columns with lookups), the greater the likelihood that the current dimensionality and filters applied in the report will apply to the target table. However it is also possible that the source and target tables could share only one common column, and as long as the dimensionality and filters of the report only use that common column, the drill-through context will be valid.

### Drill-through example

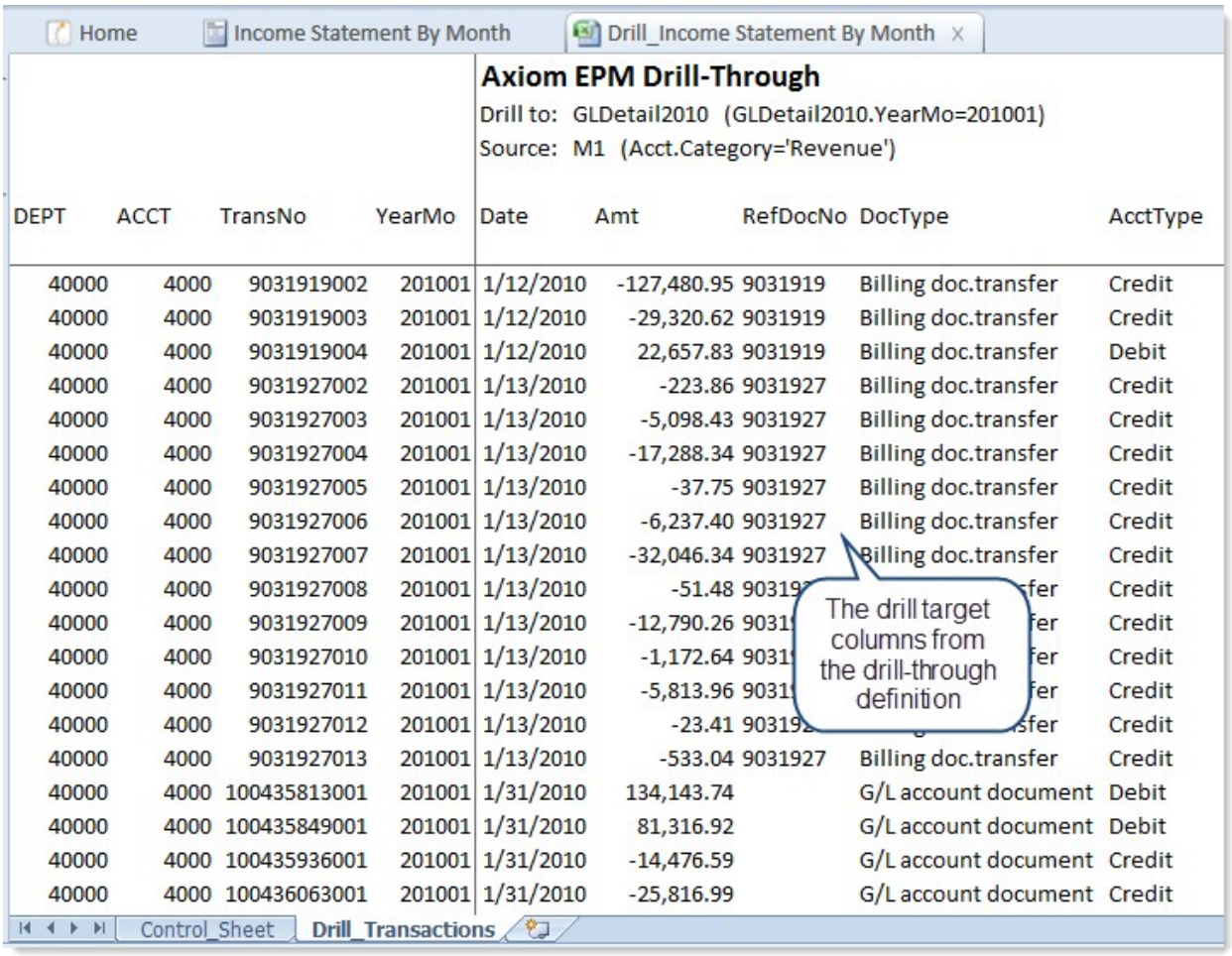
The following shows an example drill-through definition, an example file eligible to be drilled, and the example drill-through results.




*Example drill-through definition*




*Example report to be drilled*




*Example resulting drill sheet*

# Managing drill-through definitions

If you want to enable drill-through drilling in Axiom files, then you must define a drill-through definition for each target table / source sequence that you want to be able to drill.

For example, if you want to be able to drill through CYA data (current year actuals) to detailed transactional data, then you must define a drill-through definition for the GLDetail table that maps to the CYA column sequence (M1-M12 in the GL2018 table).

You can create, edit, and delete drill-through definitions by using the **Drill-Through Manager**. Only administrators or users with the **Administer Tables** security permission can access this dialog.

## Adding a drill-through definition

New drill-through definitions can be added at any time. Users can drill through data for the associated sequence as soon as the definition has been saved.

Before you can create a drill-through definition, the following components must already be in place:

- The desired target table must already be created in the database.
- The desired source table must already be created in the database, and the columns that you want to drill must be part of a column sequence defined on that table. In most cases, these column sequences already exist for other purposes (such as for calculated fields), so you do not need to create them specially for drill-through definitions.

Only administrators and users with the **Administer Tables** permission can add a drill-through definition.

**To create a new drill-through definition:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Drill-Through Manager**.
2. In the **Drill-Through Manager** dialog, click **New** ✚.
3. In the **Number of Columns** dialog, type the number of columns for this drill-through definition, and then click **OK**.

   This number must correspond to the number of columns in the sequences that you want to map to. For example, if this definition is intended to allow users to drill CYA data (current year actuals), and that sequence is 12 columns, then the definition must have 12 columns.

   **NOTE:** Once you specify the number of columns for the definition, it cannot be changed.

   The **Drill-Through Definition Editor** opens.
4. In the **General Details** section, complete the following items:

| Item | Description |
|---|---|
| Name | The name of the drill-through definition. This name will display to users on the **Drill** menu, so it should be brief and intuitive.<br><br>For example, if the definition defines a drill to transaction-level detail, the name could be "Transactions" or "Transaction Detail". |
| Target Table | The name of the table that provides the drill-through data.<br><br>For example, actuals data may be stored in a table named GL2018, and you want to be able to drill that table to the transaction data held in the GLDetail table. In this case GLDetail is the target table. |
| Description | Optional. A description for the drill-through definition.<br><br>The description only displays in the Drill-Through Manager. |

5. In the **Mapped Sequences** section, specify the associated column sequences for this definition. These sequences define the columns that users can drill in Axiom files.

**NOTE:** Although this section allows you to specify multiple sequences, in most cases each drill-through definition will only have one selected sequence. Each drill-through definition has one set of column filters to determine the appropriate drill data in the target table, so if you do specify multiple sequences, they must be able to use the same column filters. This will only apply in certain rare cases.

a. Click **Edit Mapped Sequences**.

b. In the **Select Sequences** dialog, add the column sequences that you want to map to this definition.

   For example, if this drill-through definition is to enable drilling on actuals data from the GL2018 table, you would select the 12-month sequence defined for that table (most likely named "M" or "Months").

   To find the desired column sequences, you can filter the list by typing into the filter box, or you can change the view to display tables by table name, table type, or folder.

   This dialog displays any table that has a column sequence that matches the number of columns for the drill-through definition. For example, if the drill-through definition has 12 columns, you can select any column sequence that has 12 literal columns or 12 effective columns.

   "Effective columns" takes into account the start period of the sequence. For example, if the sequence has 6 columns but the start period is 7, that sequence has 12 effective columns because the sequence covers periods 7-12. When you drill on the "first" column of the sequence, it will return data based on the column 7 filter in the drill definition.

   You can select multiple sequences, but each sequence must be able to use the same column filter to determine the appropriate drill data in the target table.

c. Click **OK** to return to the **Drill-Through Definition Editor**.

6. In the **Column Filters** section, specify the drill filter to associate with each column in the mapped sequence.

   This is the filter to apply to the target table when drilling on a column from the mapped sequence. For example, if you are drilling on column 1 of the mapped sequence (M1 or CYA1), the filter to apply to the target table would be something like: `YearMo=201801`.

   You must specify a filter for each column. You can type the filter into the filter box, or you can use the filter wizard  .

   **IMPORTANT:** The column filters determine the data that will be returned when performing "drill-through" drilling. It is important that the column filters correspond appropriately to the columns in the mapped sequences, or else drill results will not be as you expect.

7. In the **Drill Target Columns** section, specify the columns that you want to display on the drill

sheet.

a. Click **Edit Drill Target Columns**.

b. In the **Select Columns** dialog, add the columns that you want to display on the drill sheet.

   You can select any column from the target table, and any column from a lookup table for the target table. For example, if the target table is GLDetail and uses key columns of DEPT and ACCT, then you can select any column from the GLDetail table, the DEPT table, and the ACCT table.

c. Click **OK** to return to the **Drill-Through Definition Editor**.

d. In the **Drill Target Columns** list, change the order of columns as desired.

   Columns will display in drill sheets in the order listed here. To move a column, select the column in the list and then use the arrows to move it up or down.

8. Click **OK** to save the definition.

Users can drill-through using the definition as soon as it is saved.

## Cloning an existing drill-through definition

You can clone an existing drill-through definition to create a new definition, instead of creating the definition from scratch. Only administrators and users with the **Administer Tables** permission can clone a drill-through definition.

**To clone a drill-through definition:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Drill-Through Manager**.
2. In the **Drill-Through Manager** dialog, select the definition that you want to clone, and then click **Clone**.

   The **Drill-Through Definition Editor** dialog opens, containing a copy of the cloned definition.
3. Edit the settings as desired for the new definition, and then click **OK**.

   For more information on the available settings, see Drill-through definition properties.

## Editing a drill-through definition

Drill-through definitions can be edited at any time. Changes take effect immediately, meaning that any new drills initiated after the changes are saved will use the new settings.

**NOTE:** The number of columns for a drill-through definition cannot be changed. All other properties can be edited.

Only administrators and users with the **Administer Tables** permission can edit a drill-through definition.

**To edit a drill-through definition:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Drill-Through Manager**.
2. In the **Drill-Through Manager** dialog, select the definition that you want to edit, and then click **Edit**.
3. In the **Drill-Through Definition Editor** dialog, edit the definition settings as desired.

   For more information on the available settings, see Drill-through definition properties.

   If you change the target table for the definition:

   - Any columns in the **Drill Target Columns** section will be cleared. You must select new columns that relate to the new target table.
   - Column filters are retained and validated against the new table. If the column filter is not valid in the context of the new table, the filters are highlighted in red and must be changed.
4. Click **Apply** or **OK** to save your changes.

### Deleting a drill-through definition

You can delete a drill-through definition at any time. If the definition is deleted, that drill-through path will no longer be available to users in Axiom files.

Only administrators and users with the **Administer Tables** permission can delete a drill-through definition.

**To delete a drill-through definition:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Drill-Through Manager**.
2. In the **Drill-Through Manager** dialog, select the definition that you want to delete, and then click **Delete**.
3. At the confirmation prompt, click **Yes**.

The drill-through definition is deleted.

# Drill-through definition properties

The following properties are available for drill-through definitions in the **Drill-Through Definition Editor** dialog.

### General Details

This section defines the basic details for the drill-through definition.

| Item | Description |
|---|---|
| Name | The name of the drill-through definition. This name will display on the drill menu, so it should be brief and intuitive.<br><br>For example, if the definition defines a drill to transaction-level detail, the name could be "Transactions" or "Transaction Detail". |
| Target Table | The name of the table that provides the drill-through data. You can select any data table.<br><br>For example, actuals data may be stored in a table named GL2018, and you want to be able to drill that table to the transaction data held in the GLDetail table. In this case GLDetail is the target table.<br><br>If you change the target table for an existing definition:<br>• Any columns in the **Drill Target Columns** section will be cleared. You must select new columns that relate to the new target table.<br>• Column filters are retained and validated against the new table. If the column filter is not valid in the context of the new table, the filters are highlighted in red and must be changed. |
| Description | Optional. A description for the drill-through definition.<br><br>The description only displays in the Drill-Through Manager. |

## Mapped Sequences

This section specifies the column sequences that are mapped to this drill-through definition. These are the columns that users will be able to drill in Axiom files.

**NOTE:** Although this section allows you to specify multiple sequences, in most cases each drill-through definition will only have one selected sequence. Each drill-through definition has one set of column filters to determine the appropriate drill data in the target table, so if you do specify multiple sequences, they must be able to use the same column filters. This will only apply in certain rare cases.

To add or remove a sequence, click **Edit Mapped Sequences** to open the **Select Sequences** dialog. This dialog displays any table that has a column sequence that matches the number of columns for the drill-through definition. For example, if the drill-through definition has 12 columns, you can select any column sequence that has 12 literal columns or 12 effective columns.

"Effective columns" takes into account the start period of the sequence. For example, if the sequence has 6 columns but the start period is 7, the sequence has 12 effective columns because the sequence covers periods 7-12. When you drill on the "first" column of the sequence, it will return data based on the column 7 filter in the drill definition.

To find the desired column sequence, you can filter the list by typing into the filter box, or you can change the view to display tables by table, table type, or folder.

### Column Filters

This section defines the drill filters to associate with each column in the mapped sequence.

This filter is applied to the target table when drilling on a column from the mapped sequence. If you drill the first column of the sequence, the filter specified for Column 1 is applied to the drill query. For example, if the mapped sequence is "Months" from table GL2018 (columns M1-M12), then the filter that you want applied when column M1 is drilled is something like `YearMo=201801`.

You must specify a filter for each column. You can type the filter into the filter box, or you can use the filter wizard  .

**IMPORTANT:** The column filters determine the data that will be returned when performing "drill-through" drilling. It is important that the column filters correspond appropriately to the columns in the mapped sequences, or else drill results will not be as you expect.

### Drill Target Columns

This section defines the columns to be displayed on the drill sheet when this drill is executed.

To add or remove a column, click **Edit Drill Target Columns** to open the **Select Columns** dialog.

You can select any column from the target table, and any column from a lookup table for the target table. For example, if the target table is GLDetail and uses key columns of DEPT and ACCT, then you can select any column from the GLDetail table, the DEPT table, and the ACCT table.

Columns will display in drill sheets in the order listed in the **Drill Target Columns** list. To move a column, select the column in the list and then use the arrows to move it up or down.


# Data Explorer Setup

Using the Data Explorer, users can analyze and report on data using an intuitive drag-and-drop interface. Users can explore data and see results immediately, without the overhead of designing and formatting a more formal report.

Certain setup tasks must be performed in order to make the Data Explorer available to users and to expose selected "slices" of data. These setup tasks are discussed in this section. For more information on how the Data Explorer works, see the *User Guide*.

## Exposing table data for use in the Data Explorer

The Data Explorer is intended to be used for quick and easy data analysis. Therefore, instead of exposing all data in your system at once (which may be overwhelming for users to sift through and find what they want), the Data Explorer uses the concept of *data sets* to define a distinct set of data available for analysis. Administrators must define one or more data sets to enable use of the Data Explorer.

Data sets are based on table columns. When you create a data set, you pick columns from one or more tables. Users can then drag and drop those columns as desired within the Data Explorer. Any data outside of those columns is not available in the Data Explorer while the user is working with that data set.

In order to expose data for use in the Data Explorer, you must do the following:

- Review the column classification for any column that you plan to include in a data set, and adjust as needed.
- Define one or more data sets to make available for use in the Data Explorer.
- (Optional) Define intuitive display names for columns in data sets so that users can more easily understand what is in each column.

These activities are performed using table management features in the Excel Client or the Windows Client.

## Defining column classifications

Each column in a table is assigned a classification. This classification is used to determine whether the data in a column is considered to be values or dimensions:

- **Value**: Value columns contain reportable data such as budget or actuals data. For example, most columns in a data table such as GL2018 are value columns, such as M1 through M12, as well as calculated columns such as CUR, TOT, and YTD.
- **Dimension**: Dimension columns define the levels at which you want to group and display the value data. For example, the key columns in a data table such as GL2018 are dimension columns (such as Acct and Dept). Columns in lookup reference tables such as Dept and Acct are almost always dimension columns as well. In addition to the Acct and Dept key columns, this would include columns such as AcctCategory, Company, Region, VP, etc.

When using the Data Explorer, value columns define the data values to be reported on, and dimension columns define the row groupings (and optionally column groupings). For example, you can drag a dimension column to the rows or the columns of the Data Explorer, but you can't drag it to the values. Similarly you can drag a value column to the values of Data Explorer but not to the rows or columns. Filters can be defined using either value or dimension columns.

All columns are assigned a classification by default, but you can override this classification as needed. The default classifications are as follows:

- Numeric columns are classified as Value by default.
- Integer columns (all types) are classified as Dimension in reference tables. In data tables, non-key, non-validated Integer columns are classified as Value, otherwise they are classified as Dimension.
- All other columns are classified as Dimension.

**NOTE:** All columns in document reference tables are assigned Dimension as the classification. This cannot be changed.

If the default classification is not appropriate for a particular column in a data or reference table, you can change it in the column properties. To edit column properties in a table, you must either be an administrator or have the **Administer Tables** permission (as well as access to the table).

**To change the classification for a column:**

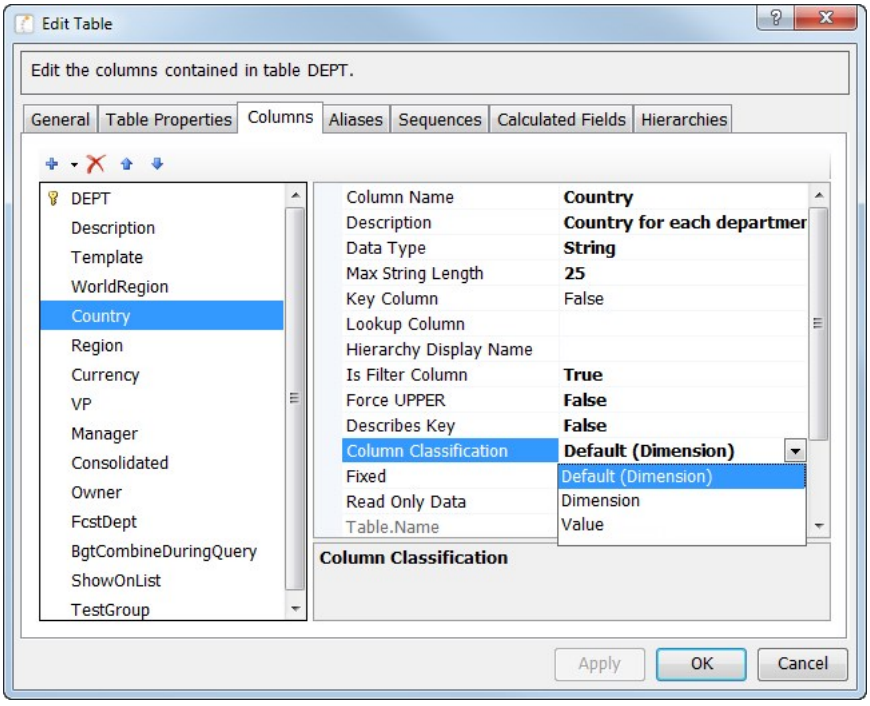
1. On the **Axiom** tab, in the **Administration** group, click **Tables** and then navigate in the Table Library to the desired table. Select **Edit Table Structure** for that table.

   **TIP:** If you are in the Axiom Explorer table view, you can right-click a table to edit the table structure.

2. In the **Edit Table** dialog, click the **Columns** tab.
3. In the left-hand pane of the Columns tab, select the column that you want to change. Then in the

right-hand pane, scroll down the list of properties until you locate **Column Classification**.

4. Select one of the following options for Column Classification:
   - **Default**: The column will use the default classification for this type of column. The default classification for the current column is listed in parenthesis.
   - **Dimension**: The column is considered to be a dimension column, regardless of its default classification.
   - **Value**: The column is considered to be a value column, regardless of its default classification.




5. Click **OK** to save.

## Column display names

By default, the columns in the data set display in the Data Explorer using their actual column names. If desired, you can define more intuitive display names that will be used instead of the column names. This may make it easier for non-technical users to identify the data that they want to see. The user can still see the Table.Column name in the tooltip when they hover over a display name.

To define a display name, use the Hierarchy Display Name property for the column. This display name is also used by hierarchies (such as in the Quick Filter and Filter Wizard). Make sure that the defined display name is appropriate for all places where it will be used.

To edit column properties in a table, you must either be an administrator or have the **Administer Tables** permission (as well as access to the table).

**To define the display name for a column:**

1. On the Axiom tab, in the **Administration** group, click **Tables** and then navigate in the Table Library to the desired table. Select **Edit Table Structure** for that table.

   **TIP:** If you are in the Axiom Explorer table view, you can right-click a table to edit the table structure.

2. In the **Edit Table** dialog, click the **Columns** tab.
3. In the left-hand pane of the Columns tab, select the column that you want to change. Then in the right-hand pane, scroll down the list of properties until you locate **Hierarchy Display Name**.
4. Type the desired display name into the **Hierarchy Display Name** field.
5. Click **OK** to save.

### Defining data sets

Data sets define collections of table columns that are available for drag and drop analysis within the Data Explorer. When users create new Data Explorers, they first select which data set they want to use. They can then report on any data within the value columns in that data set, and they can group and display the data using any dimension columns in that data set.

When defining a data set, you should consider the following:

- What is a useful and distinct set of data that users may want to perform ad hoc analysis on? Data sets are most useful when they are focused on a particular area of interest to help users answer a specific set of questions. Including all of your data in a data set may make the Data Explorer difficult to use.
- Once you have decided on the data, which specific columns should be included in the data set? Generally speaking, you want to limit the data set to only the necessary columns so that users can quickly and easily identify what they want to see and how they want to see it, without sifting through endless lists of columns.
- What name should the data set have, so that the contents and purpose of the data set will be obvious to users? For example, instead of naming the data set "Budget Columns", you might want to name it something like "Analyze Budget Data".

To define data sets, you use the Data Set Manager. This tool is available from the **Tables > Table Administration** menu, and is only available to administrators or users with the **Administer Tables** permission. For more information, see Data Sets for Data Explorer.

# Data Sets for Data Explorer

Data sets are used by the Data Explorer to determine which columns are available for drag-and-drop analysis. In this context, the selected data set defines the data available to the Data Explorer at any one time.

When you define a Data Explorer data set, you select the columns to include and you define a name to describe that collection of columns.

Only administrators or users with the **Administer Tables** security permission can manage Data Explorer data sets.

**NOTE:** There is a second type of data set known as *product data sets*, used to define the columns that belong to a particular product. Product data sets can only be defined from within the **Edit Table** dialog, and do not display when working with Data Explorer data sets.

## Data set design considerations

Note the following design considerations for data sets:

- For dimension-classified columns with lookup relationships, the Data Explorer always uses the lookup column for the underlying database query, not the validated column. For example, if GL2013.Acct links to Acct.Acct, Acct.Acct will always be used. Therefore when selecting columns to include in the data set, you should include either the validated column *or* the lookup column (since using either will give you the same results)—but not both. Including both columns may cause confusion for users as to which column to use. If multiple data tables are included in the data set, you should always include the lookup column instead of the validated columns.
- It is recommended to limit the data set to only the specific columns that users will report and filter on. Keep the list of columns as streamlined and focused as possible, so that users can easily explore the data and find the columns they are looking for. Do not select all columns in a table unless you truly need all of those columns.

## Adding a data set

You can create as many data sets for Data Explorer as you want. Data sets can be created from scratch, or you can clone an existing data set.

**To create a new Data Explorer data set:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Data Set Manager**.
2. In the **Data Set Manager** dialog, click **New** ✚.
3. At the top of the **Data Set Editor** dialog, define the following:
    - **Name**: The name should be brief and intuitive. Users will select these names when creating Data Explorer files, to define the set of data available in the Data Explorer.
    - **Description**: The description is for data set managers only, to explain the purpose of the data set. The description does not display to end users.
4. At the bottom of the **Data Set Editor** dialog, select the columns to include in the data set.

You can select all of the columns in a particular table by selecting the check box next to the table name, or you can expand the table name and select individual columns in that table.

If a column has a defined **Hierarchy Display Name**, that name displays in parentheses after the column name. The display name will be used in the Data Explorer.

To make it easier to find specific columns that you want to select, you can filter the list of tables by using the table classification check boxes at the top of the list, or by typing filter text into the box above the list.

5. Click **OK** to save.

**To clone an existing Data Explorer data set:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Data Set Manager**.
2. In the **Data Set Manager** dialog, select the data set that you want to copy, and then click **Clone**.

   The **Data Set Editor** dialog opens, populated with the settings from the cloned data set. The name of the data set is incremented by a number to prevent duplicate data set names.
3. Edit the data set properties as desired.
4. Click **OK** to save.

## Editing a data set

Data sets for Data Explorer can be edited at any time. Any changes made to a data set will not be reflected in any open Data Explorer files using that data set. Changes will take effect the next time a user opens that file.

If you remove a column from a data set, and a Data Explorer file has been saved that uses the column as part of its data query configuration, then the next time that Data Explorer is opened the column will be flagged as no longer part of the data set (which may result in the query no longer being useful).

**To edit a Data Explorer data set:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Data Set Manager**.
2. In the **Data Set Manager** dialog, select the data set that you want to edit, and then click **Edit**.
3. In the **Data Set Editor** dialog, edit the data set properties as desired.
4. Click **OK** to save.

## Deleting a data set

Data sets can be deleted at any time. If any Data Explorer files use the deleted data set, those Data Explorer files can still be opened but will only show data for the columns that were configured in the

query settings when the Data Explorer was last saved. In other words, the Data Explorer will still show data for the currently configured Rows, Values, and Columns, but any columns that were not part of the query when the Data Explorer was saved will no longer be available.

**To delete a Data Explorer data set:**

1. On the **Axiom** tab, in the **Administration** group, click **Tables > Table Administration > Data Set Manager**.
2. In the **Data Set Manager** dialog, select the data set that you want to delete, and then click **Delete** ✕.

   You are prompted to confirm that you want to delete the data set.
3. Click **OK** to delete.


# Importing Data

From the **Imports** menu, you can create and edit imports, and execute imports.

## About imports

Axiom Software provides robust data import functionality. This topic explains some of the key concepts and requirements for importing data.

### Import sources

You can import data into Axiom Software from the following sources:

- Directly from another database
- Delimited files
- Excel files (XLS or XLSX)

You can also import from one table in Axiom Software to another table in Axiom Software.

### The import process

When data is queried from the source file or database, it is first placed in a temporary table known as the *temptable*. You can then perform actions on the temptable before the data is saved to the destination table, such as mapping or data transformations. Use the reserved term `{temptable}` whenever you want to refer to this temporary table in SQL statements.

When an import is executed, the following processes occur:

1. If any variables are defined on the **Variables** tab, the user is prompted to select values for these variables. The selected variable values are then substituted for the variable names within the import settings.

   When running the import via Scheduler, you must specify values for the import variables within the Scheduler task, or use job variables that will populate the import variable values when the job is executed.

2. The import creates the temptable by querying data from the source database or by gathering data from the specified file. The settings on the **Mapping** tab are used to determine the structure of the temptable. You can insert additional columns into the temptable (meaning columns that were not

in the source file or table) by adding them as work column mappings.

3. Any transforms defined on the **Transforms** tab are processed, in the order listed. Transforms can edit the temptable directly, and they can reference information held in other tables in the Axiom Software database. Transforms can also set values for transform variables, which can then be used in subsequent transform steps and in certain import settings.
4. The temptable data is validated and then saved to the destination table, based on the destination column settings on the **Mapping** tab. If a column in the temptable is not mapped, then that data is not saved.

### Import save behavior

The save-to-database process for imports is performed as follows:

- If the destination table has any validated columns (columns that have an assigned lookup column), then the temptable data is validated against these lookup columns before saving. If a data row contains an invalid value, that data row is invalid and cannot be saved. The behavior of the import depends on whether **Ignore lookup and key errors** is enabled.
  - By default, if any invalid data rows are found, then the import will abort and no data will be saved.
  - If instead the optional setting **Ignore lookup and key errors** is enabled, then the save-to-database process will ignore the invalid data rows and only valid data rows will be saved.
- By default, temptable data is aggregated before the save is performed. This means that duplicate rows (rows with the same key column values) will be treated as follows:
  - Columns holding numeric data will be summed, unless the destination table is a reference table, in which case the maximum value will be used.
  - For all other column types, the maximum value will be used.

  If instead the optional setting **Aggregate rows on final save** option is disabled, then no aggregation is performed on the temptable data. If any duplicate rows are found, then the import will abort and no data will be saved.
- Blanks are not allowed in key columns. If a data row in the temptable contains a blank key value, that data row is invalid and cannot be saved. The behavior of the import depends on whether **Ignore lookup and key errors** is enabled.

# Managing imports

Imports can be created, edited, or deleted by using the **Imports** menu in the **Administration** group of the **Axiom** tab. You can also manage imports using the Explorer task pane or Axiom Explorer.

**NOTE:** Only administrators and users with the **Administer Imports** security permission can create imports. Other users may be granted access to execute and/or edit specific imports, as defined on the **Files** tab of Security.

For details on executing an import, see Executing import utilities.

## Creating an import

1. On the **Axiom** tab, in the **Administration** group, click **Imports > Create New Import**.
2. In the **Create New Import** dialog, select one of the following and then click **OK**.
   - **Create from scratch** (default): Create a new import starting with blank import settings.
   - **Create from existing**: Create a new import by copying an existing import. If you select this option, then select the import that you want to copy from the list in the bottom of the dialog.
3. In the **Import Wizard** dialog, complete the settings on each tab as appropriate.

   For details on specific import settings, see Import Wizard.

   If you chose to copy an existing import, that import's settings are copied into the Import Wizard, and the import is named "Copy of *ImportName*." Edit these settings as appropriate for the new import.

   You can move between tabs in any order, however, before you can save the import, all required settings must be completed and no invalid settings must be present. If errors exist, an error message displays at the bottom of the dialog; you can click the error link to be taken to the tab with the error.
4. When you are finished completing the settings and no errors exist, click **OK** to save the import.
5. In the **Save As** dialog, navigate to the folder where you want to save the import, then click **Save**. By default, the import will be saved to the root of the Imports Library. You can create a new sub-folder from this dialog if desired (and if you have the appropriate permissions).

Once an import has been created, it becomes available on the **Imports** menu to be executed, edited, or deleted. Imports are listed in alphabetical order based on the import name.

## Editing an import

To edit the settings for an existing import, click **Imports > *ImportName* > Edit**.

The Import Wizard opens. You can change any import settings as desired, including the name of the import which displays on the **Imports** menu.

**NOTE:** When editing an import, you can use the **Save As** button to save the current import as a new import file. If you have read-only rights to the import, then the **Save As** button is the only save option available.

## Deleting an import

To delete an import, click **Imports > *ImportName* > Delete**.

# Executing import utilities

When you execute an import utility, data is queried from the source database or file, import steps are processed, and the resulting data is saved to the destination table. For more information on what occurs when an import is executed, see About imports.

**NOTES:**

- By default, only administrators can execute import utilities. Non-admin users can be granted permission to execute specific imports, as defined on the **Files** tab of Security.
- You can also use Scheduler to execute import utilities, using the **Import ETL Package** task.

**To execute an import:**

1. On the **Axiom** tab, in the **Administration** group, click **Imports** > ***ImportName*** > **Execute**.
2. In the **Execute Import** dialog, review the import description for any important information about the import, and then adjust the **Import Options** as desired.

   For more information about the import options, see Execute tab. If you are testing the import, you may want to enable either or both of **Allow pauses** and **Preview only**.

   **IMPORTANT:** When an import is executed in preview mode, all of the steps in the **Transforms** tab are performed, including any steps that modify tables other than the temptable. If you do not want these steps to occur as part of a preview, you should edit the import to disable the step on the **Transforms** tab.

3. Click **Execute**.
4. If the import uses variables, the **Variables** dialog opens for you to define values for the variables. For each variable, you can select a value from the drop-down list. Some variables may also allow you to type in a user-defined value.
5. If the import uses a source file, you may be prompted to specify the location of the source file. In the **Open** dialog, navigate to the file that you want to use and then click **Open**.

   In this case, a copy of the specified file is uploaded to the application server for processing. Once the import is complete, the temporary copy of the file is deleted.

As the import is processed, status messages display in the **Execution log** at the bottom of the dialog. If an error occurs, the error message displays in the log and the import is stopped. If necessary, you can copy and paste the text in the execution log; for example, to send the error to Axiom Software support. Right-click inside the log and choose **Select All** and then **Copy**.

**TIP:** If an import experiences an error and you need more information on the error, try running the import in preview mode. The error messages in preview mode may contain more detail.

If you want to stop the import while it is executing, click **Stop**. The import stops after completing the step in process. You cannot restart the import at the same spot—when you click **Execute**, the import starts over from the beginning.

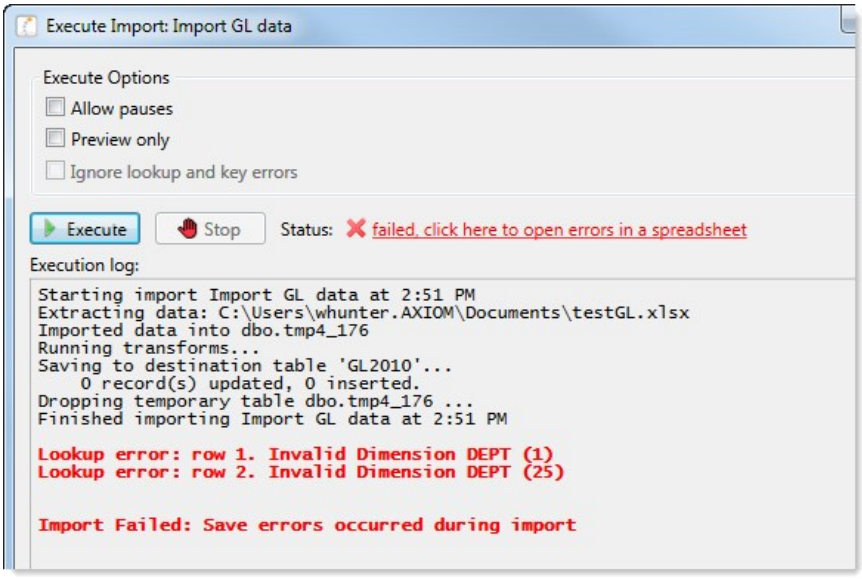
When the import is complete, click **OK** to close the dialog.

### Import validation errors

If the import experiences import validation errors, then those errors will be logged to a separate CSV file in addition to being displayed in the Execution log within the dialog. You can open this file separately to see exactly which rows of data were invalid within the context of the import data. This error file includes the following:

- Lookup validation errors from Axiom Software's built-in validation against lookup columns.
- Validation errors from any **Custom Data Validation** steps in the transforms.
- Key validation errors such as blank keys or duplicate keys.

You can open the file from the **Execute Import** dialog by clicking the link in the Status area. The status will be either "failed" or "warning," followed by the text "click here to open errors in a spreadsheet." The status type depends on whether the option to **Ignore lookup and key errors** is selected.




The CSV file contains the import data, followed by one or more validation columns. Validation columns are labeled as follows:

- Lookup and key validation messages are in a column named ***LookupColumnName* Lookup Error**. For example: "Acct Lookup Error" when looking up against the ACCT column.
- Custom Data Validation messages are in a column named **AXTRANSFORM_*StepNumber***, where *StepNumber* is the number of the associated transformation step. For example: "AXTRANSFORM_5" when the associated transform is step 5 in the list.

The error CSV files are placed in a system-maintained folder in the Imports Library named Import Errors. Access to the error files is automatically managed based on access to the import. You can access the error file directly later by using the Explorer task pane or Axiom Explorer.

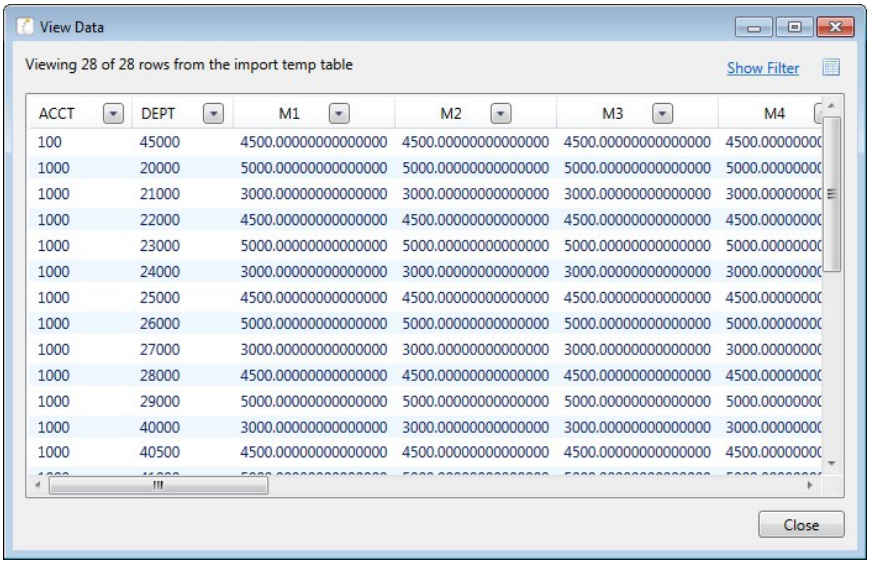
Each execution of an import that results in a lookup error will generate a unique error file (differentiated by a date/time stamp). These error files are not automatically deleted; you must manually delete them when you are finished investigating the error.

# Troubleshooting imports using pause

To assist in troubleshooting imports, you can pause the import after a transform and display the temptable data as it exists after that step. You can then review the data to see if it appears as you expect.

The **View Data** dialog displays after executing a transform if both of the following are true:

- The **Pause** check box is selected for the transform on the **Transforms** tab.
- The **Allow Pauses** option is selected when executing the import.




*Example View Data dialog*

You can view and filter the data in the View Data dialog to troubleshoot the import. When you are done viewing the data, click **Close** to return to the import dialog. At this point the import is still paused. You can click **Continue** to continue the import, or **Stop** to stop it.

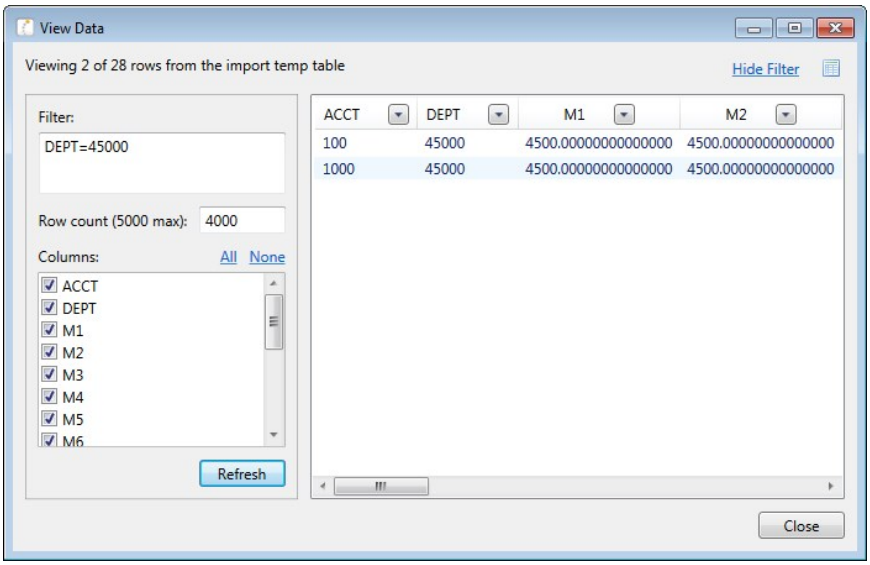
## Filtering the data in the View Data dialog

By default, the View Data dialog shows the first 500 rows of data in the temptable. This is controlled by the ETLMaxRows system configuration setting.

If desired, you can filter the data to make it easier to find specific records you might be looking for. To do this, click **Show Filter** in the top right corner. This enables the following filter options for the dialog:

| Item | Description |
|---|---|
| Filter | Type in a filter to limit the data shown in the dialog. The filter must use column-only syntax, using a column name in the temptable. For example:<br>`Dept=45000`<br>Where Dept is a column in the temptable. |
| Row count | Type in a number to limit the data to a certain number of rows. The maximum number of rows that can be shown is 5000. |
| Columns | Select or clear columns to limit the temptable columns that display in the dialog. You can use the **All** or **None** options to select or clear all columns at once.<br>If you clear a column, then that column cannot be used in the filter.<br>**NOTE:** At least one column must be selected to display. If no columns are selected, then all columns will display. |

After modifying the filter options, click **Refresh** to filter the data in the dialog using your selections. To clear your filter options and return to the default dialog, click **Hide Filter**.




*Example View Data dialog with filter options*

# Using variables in imports

You can use two different types of variables in imports:

- **Import variables:** Import variables are defined on the **Variables** tab of the Import Wizard. Import variables can be used throughout the import settings (though not in *all* settings; see the documentation for each setting to see if variables are supported there). When the import is

executed, the user is prompted to define values for the variables.

- **Transform variables:** Transform variables are defined on the **Transforms** tab of the Import Wizard. Transform variables can only be used in transform statements, and as destination columns. Transform variables are associated with a specific SQL statement that results in a single value.

The values for import variables are defined by the user who is executing the import, at the start of the import, before any other import steps are processed. Therefore import variables are a good fit for actions such as:

- Selecting the appropriate source file based on user input.
- Selecting the appropriate destination table based on user input.

On the other hand, values for transform variables can only be determined as a result of a SQL statement, and are defined near the end of the import, after the temptable has been created. Transform variables are a good fit for situations where actions need to be driven dynamically based on the contents of the imported data, not by a user selection.

### Variable syntax

To use a variable in the import, enter the variable name into one of the supported areas of the import settings, enclosed in curly brackets {}. For example, if the variable name is "column", you would enter `{column}`.

**NOTE:** If the variable defines the destination table, then you must place the variable in double curly brackets when you use it in a SQL statement, so that the eventual table name value is enclosed in curly brackets as expected. For example, if you have a variable named "destinationtable", you would reference that variable as `{{destinationtable}}`. That way, when the `{destinationtable}` value is defined, it will resolve as `{GL2018}`.

### System variables

In addition to the user-defined variables, you can reference system variables in imports. The following variables are supported:

| System Variable | Description | Can Be Used In |
|---|---|---|
| `{CurrentPeriod}` | The current period as defined for the destination table (if not set, then this is the system current period). | All import locations that support variables, except the destination table. |
| `{CurrentUserDomain}` | The domain name of the user running the import. Returns blank for users who do not have a defined domain. | All import locations that support variables. |

| System Variable | Description | Can Be Used In |
|---|---|---|
| {CurrentUserEmail} | The email address of the user running the import. | All import locations that support variables. |
| {CurrentUserLogin} | The login name of the user running the import. | All import locations that support variables. |
| {SourceFileName} | The name of the source file for the import. | Import transformation steps only. |
| {SystemCurrentYear} | The current year as defined for the system. | All import locations that support variables. |
| {SystemCurrentPeriod} | The system current period. | All import locations that support variables. |
| {TempTable} | The temporary table where imported data is placed before saving to the destination table. | All import locations that support variables. |
| {*TableName*} | Any user-defined table created in the Axiom Software system. For example, {ACCT}, {DEPT}, {GL2018}, {Plan2018}. | All import locations that support variables. |

# Import Wizard

Using the Import Wizard, you can create new imports and edit existing imports. Import settings are organized by tab. See the individual topics on each tab for more information.

The import **Name** is defined at the top of the Import Wizard, above the individual settings tabs. The import name is how you access and execute the import from the **Imports** menu, and in other areas such as the **Import ETL Package** task in Scheduler.

As you complete the import settings, the Import Wizard performs error checking for missing required settings and invalid settings. If an error is detected, an error message displays in the bottom of the dialog. You can click the link to be taken to the tab that contains the error. Only one error is displayed at a time; once you resolve the current error message, a new message may appear.

# Source tab

The **Source** tab of the Import Wizard specifies the source of the data to be imported. The contents of the tab depend on the **Import source** selection at the top of the tab.

| If you are importing data from... | Use one of the following options... |
|---|---|
| An external database (direct read) | • **SQL Server**<br>• **Oracle**<br>• **OLEDB**<br>• **ODBC**<br>• **Intacct** |
| A file | • **Delimited File**<br>• **Excel File** |
| A table in the current Axiom Software database | • **Current Axiom system database**<br>• **Current Axiom audit database**<br>• **Current Contract Management database** |

**NOTE:** The OLEDB source type can also be used to import a file. If you want to import from a file that is not one of Axiom Software's supported file types, you may be able to use the OLEDB source type. The database connection strings can be configured to pull data from various file types.

The following sections detail the requirements and import settings for each import source.

## Import source: Delimited file

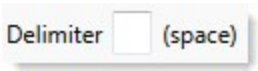
You can import data from a delimited file. The delimited file must meet the following criteria:

- Delimiters can be any character. You specify the delimiting character in the import settings.
- The first row of the file can optionally contain column header names.
- Numeric values cannot be in scientific notation or formatted with extraneous characters such as currency signs or parentheses.

The **Source** tab of the Import Wizard uses the following settings when importing from a delimited file.

| Item | Description |
|---|---|
| Source | Select **Delimited File** to import data from a delimited text file, such as CSV or TXT files. The first row can optionally contain header names. |

| Item | Description |
| --- | --- |
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Cloud Service, and you are not using the "prompt for file" option (see the **File Location** setting), then you must specify a remote data connection so that the cloud service can read the file located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |
| File Location | The name and location of the source file. Select one of the following options:<br>• **Always use this file**: Select this option if the file name and location is always known. In the **File** box, enter the path and file name. You can click the folder icon to navigate to the file.<br>The file path must be a UNC path (i.e. \\servername\foldername\filename). If you enter a mapped drive, it will automatically be converted to a UNC path.<br>The location of this file must be accessible to the Axiom Application Server. When you specify a file, Axiom Software will validate whether the application server can access the file, and will display an error if not.<br>• **Prompt for file during execution**: Select this option if you want the user to be able to specify the file when running the import. If desired, in the **Folder** box, you can specify a folder location. When the user is prompted to select a file, it will open to this folder by default. The user can still browse to a different location.<br>In this case, a copy of the specified file is uploaded to the application server for processing. Once the import is complete, the temporary copy of the file is deleted.<br>**NOTE:** Files greater than 100 MB cannot be uploaded using the "prompt" option. While it is possible to increase this limit, it is not recommended. Instead, you should use **Always use this file** if you need to import a file larger than 100 MB. Please contact Kaufman Hall Software Support if you need assistance with a large file.<br><br>Variables can be used to specify the file name or location. See Using variables in imports. |
| First row has column names | Select this option if the first row of the file contains column names. If the first row of the file contains data, leave this option unchecked. |

| Item | Description |
|---|---|
| Import file has multi-line values | Select this option if the import file has data where a field value splits across rows (within the text qualifier). For example:<br>`DEPT, ACCT, LOCATION, A1, A2, A3`<br>`100,4000, East Region, 123, 456, 789`<br>`100,4100,"West`<br>`Region",111,222,333`<br>If this option is selected, then the split value will be read as a single import value. |
| Delimiter | In the box to the right of the option, enter the delimiting character used in the source file. For example, if the delimiter is a comma, enter a comma into the box.<br>  <br>If the delimiting character is a space or a tab, place your cursor in the box and press the space bar or the tab key. The character will be indicated in parentheses to the right of the box (since the character is not visible in this case).<br>  |
| Text Qualifier | By default, the text qualifier is double quotation marks ("). If desired, you can enter a different character as the text qualifier, or you can clear the field if you do not want to use a text qualifier.<br>The text qualifier is used when values in the source file may contain the delimiting character. For example, if the delimiting character is a comma, but the source file contains values such as full names that also contain a comma (for example: "Doe, Jane"). In this case, the comma within the quotation marks is considered part of the field value instead of starting a new field. |

## Import source: Excel file

You can import data from an Excel file. The Excel file must meet the following criteria:

- The file format must be XLS or XLSX. XLSM files cannot be imported.
- The first row can optionally contain header names.
- Each column in the Excel file translates to a column in the destination table. Each row in the file translates to a data record in the table. Blank columns and rows are ignored.

- The data in the spreadsheet must match the designated data type for the destination column. For example, if numeric values in the spreadsheet are prefixed with a quotation mark, then Excel considers those values to be strings instead of numbers. This will cause an error if attempting to import these string values into a numeric column.

The **Source** tab of the Import Wizard uses the following settings when importing from an Excel file.

| Item | Description |
|---|---|
| Source | Select **Excel File** to import data from an Excel file. |
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Cloud Service, and you are not using the "prompt for file" option (see the **File Location** setting), then you must specify a remote data connection so that the cloud service can read the file located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |
| Sheet name | The sheet in the Excel file to import. Leave this blank to use the first sheet in the file. Only one sheet can be imported.<br><br>Variables can be used to specify the sheet name. See Using variables in imports. |

| Item | Description |
| --- | --- |
| File Location | The name and location of the source file. Select one of the following options:<br>• **Always use this file**: Select this option if the file name and location is always known. In the **File** box, enter the path and file name. You can click the folder icon to navigate to the file.<br>The file path must be a UNC path (i.e. \\servername\foldername\filename). If you enter a mapped drive, it will automatically be converted to a UNC path.<br>The location of this file must be accessible to the Axiom Application Server. When you specify a file, Axiom Software will validate whether the application server can access the file, and will display an error if not.<br>• **Prompt for file during execution**: Select this option if you want the user to be able to specify the file when running the import. If desired, in the **Folder** box, you can specify a folder location. When the user is prompted to select a file, it will open to this folder by default. The user can still browse to a different location.<br>In this case, a copy of the specified file is uploaded to the application server for processing. Once the import is complete, the temporary copy of the file is deleted.<br>**NOTE:** Files greater than 100 MB cannot be uploaded using the "prompt" option. While it is possible to increase this limit, it is not recommended. Instead, you should use **Always use this file** if you need to import a file larger than 100 MB. Please contact Kaufman Hall Software Support if you need assistance with a large file.<br>Variables can be used to specify the file name or location. See Using variables in imports. |
| First row has column names | Select this option if the first row of the file contains column names. If the first row of the file contains data, leave this option unchecked. |

## Import source: SQL Server

The **Source** tab of the Import Wizard uses the following settings when importing data from a SQL Server database.

| Item | Description |
| --- | --- |
| Source | Select **SQL Server** to read data directly from a SQL Server database. |

| Item | Description |
|---|---|
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Software cloud service, then you must specify a remote data connection so that the cloud service can connect to the database server located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |

## Connection

Complete the following connection information for the import source. Once you have completed the connection settings, click the **Test connection** button to test the connection. The **Status** updates to show either a success message or an error message.

**NOTE:** This information is only used to connect to the source and extract the data into the import temptable. Transformation steps are not performed using these credentials.

| Item | Description |
|---|---|
| Server | The name of the SQL Server. |
| Database | The name of the database. |
| User | The user name to use to connect to the specified server and database. The user credentials must be for a SQL Server account; network domain credentials cannot be used. |
| Password | The password to use to connect to the specified server and database.<br><br>**NOTE:** The password must be re-entered whenever any of the other connection properties are changed. |

## SQL Select Statement

The SQL SELECT statement defines the data query to the source database, resulting in the set of data to be imported to the *temptable*. You can then perform mapping and transformations on the data before importing into the destination table.

In the **SQL Select Statement** box, enter any valid SQL statement to define the data query. You can click the **SQL editor** button to open the **Edit SQL** dialog. This dialog provides a text editor for entering and reviewing large SQL statements, and several tools to check the statement. For more information, see Creating the SQL SELECT statement.

Variables can be used in the SELECT statement. See Using variables in imports.

## Import source: Oracle

The **Source** tab of the Import Wizard uses the following settings when importing data from an Oracle database.

| Field | Description |
|---|---|
| Source | Select **Oracle** to read data directly from an Oracle database.<br><br>**NOTE:** The Oracle Data Access Connection software (ODAC) must be installed on the Axiom Software application server in order to use this import option. If you want to import directly from an Oracle database without installing this software on the application server, you can use the OLEDB import source instead. |
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Software cloud service, then you must specify a remote data connection so that the cloud service can connect to the database server located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |

### Connection

Complete the following connection information for the import source.

**NOTE:** This information is only used to connect to the source and extract the data into the import temptable. Transformation steps are not performed using these credentials.

| Field | Description |
|---|---|
| Server | The connection parameters for the Oracle server. You can obtain this information from the Oracle TNS Names entry. See the following section for more information. |
| User | The user name to use to connect to the database. |
| Password | The password to use to connect to the database.<br><br>**NOTE:** The password must be re-entered whenever any of the other connection properties are changed. |

Once you have completed the connection settings, click the **Test connection** button to test the connection. The **Status** updates to show either a success message or an error message.

### Obtaining the Oracle connection parameters from a TNS Names entry

A typical TNS Names entry for Oracle looks like the following:

```
SERVER=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=MyHOSTNAME)
(PORT=MyPORT))(CONNECT_DATA=(SERVICE_NAME=MyOracleServiceID)))
```

Axiom requires this information in the following format:

```
MyHOSTNAME:MyPORT/MyOracleServiceID
```

Where:

- *MyHostName* is the name of the Oracle server machine
- *MyPort* is the port number that the server is listening on, typically 1521
- *MyOracleServiceID* is the name of the Oracle service running on the host machine

### SQL Select Statement

The SQL SELECT statement defines the data query to the source database, resulting in the set of data to be imported to the *temptable*. You can then perform mapping and transformations on the data before importing into the destination table.

In the **SQL Select Statement** box, enter any valid SQL statement to define the data query. You can click the **SQL editor** button  to open the **Edit SQL** dialog. This dialog provides a text editor for entering and reviewing large SQL statements, and several tools to check the statement. For more information, see Creating the SQL SELECT statement.

Variables can be used in the SELECT statement. See Using variables in imports.

## Import source: OLEDB

Use the **OLEDB** option on the Import Wizard's **Source** tab to connect to any database or file that supports OLEDB, with the following exceptions:

- If the database is a SQL Server or Oracle database, you can use the database-specific options instead. However, you can use OLEDB to connect to these database types if desired.
- If the file is an Excel file, OLEDB cannot be used. Use the **Excel File** option instead.

The **Source** tab of the Import Wizard uses the following settings when importing data using an OLEDB connection.

| Item | Description |
|---|---|
| Source | Select **OLEDB** to read data directly from a database or a file using an OLEDB connection. |

| Item | Description |
|---|---|
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Software cloud service, then you must specify a remote data connection so that the cloud service can connect to the database server located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |

## Connection

The connection string identifies the name and location of the database or file to connect to, including any necessary validation information. In the **Connection string** box, enter the connection string to the source database or file. Any valid SQL connection string can be used. The connection string cannot contain spaces.

**NOTE:** This information is only used to connect to the source and extract the data into the import temptable. Transformation steps are not performed using these credentials.

Once you have completed the connection settings, click the **Test connection** button  to test the connection. The **Status** updates to show either a success message or an error message.

**NOTE:** If the connection string contains a password, that password must be re-entered whenever any of the other connection properties are changed.

A good resource for connection strings is http://www.connectionstrings.com/. Some examples of common connection strings are shown below:

| Source | Sample string |
|---|---|
| CSV | Server=.\SQLExpress;Provider=MSDASQL;Driver={Microsoft Text Driver (*.txt; *.csv)};UID=test;PWD=test!123;Database=AxiomFinancial |
| SQL Server 2005, trusted connection | Data Source=myServerAddress;Initial Catalog=myDataBase;Integrated Security=SSPI; |
| Oracle with TNS | Data Source=TORCL;User Id=myUsername;Password=myPassword; |
| SQLOLEDB (standard) | Provider=sqloledb;Data Source=myServerAddress;Initial Catalog=myDataBase;User Id=myUsername;Password=myPassword; |
| SQLOLEDB (trusted) | Provider=sqloledb;Data Source=myServerAddress;Initial Catalog=myDataBase;Integrated Security=SSPI; |

| Source | Sample string |
|---|---|
| SQLOLEDB (server instance) | Provider=sqloledb;Data Source=myServerName\theInstanceName;Initial Catalog=myDataBase;Integrated Security=SSPI; |
| AS400 | Provider=IBMDA400;Data Source=MY_SYSTEM_NAME;User Id=myUsername;Password=myPassword; |

### SQL Select Statement

The SQL SELECT statement defines the data query to the source database, resulting in the set of data to be imported to the *temptable*. You can then perform mapping and transformations on the data before importing into the destination table.

In the **SQL Select Statement** box, enter any valid SQL statement to define the data query. You can click the **SQL editor** button  to open the **Edit SQL** dialog. This dialog provides a text editor for entering and reviewing large SQL statements, and several tools to check the statement. For more information, see Creating the SQL SELECT statement.

Variables can be used in the SELECT statement. See Using variables in imports.

**NOTE:** The syntax of the SQL statement cannot be validated when using OLEDB as the source.

## Import source: ODBC

Use the **ODBC** option on the Import Wizard's **Source** tab to connect to any database that supports Open Database Connectivity. Generally speaking, this option should only be used if no other option is available to connect to your desired database. If you are connecting to a SQL Server or Oracle database, the database-specific options should be used instead.

The **Source** tab of the Import Wizard uses the following settings when importing data using an ODBC connection.

| Item | Description |
|---|---|
| Import source | Select **ODBC** to read data directly from a database using an ODBC connection. |
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom Software cloud service, then you must specify a remote data connection so that the cloud service can connect to the database server located on your network.<br><br>You can select from any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |

### Connection

The connection string identifies the name and location of the database to connect to, including any necessary authentication credentials. The connection string requirements and syntax vary depending on the source database you are attempting to connect to. Consult the documentation from your database vendor to determine an appropriate ODBC connection string for this purpose.

**NOTE:** This information is only used to connect to the source and extract the data into the import temptable. Transformation steps are not performed using these credentials.

Once you have completed the connection settings, click the **Test connection** button to test the connection. The **Status** updates to show either a success message or an error message.

**NOTE:** If the connection string contains a password, that password must be re-entered whenever any of the other connection properties are changed.

### SQL Select Statement

The SQL SELECT statement defines the data query to the source database, resulting in the set of data to be imported to the *temptable*. You can then perform mapping and transformations on the data before importing into the destination table.

In the **SQL Select Statement** box, enter any valid SQL statement to define the data query. You can click the **SQL editor** button to open the **Edit SQL** dialog. This dialog provides a text editor for entering and reviewing large SQL statements, and several tools to check the statement. For more information, see Creating the SQL SELECT statement.

Variables can be used in the SELECT statement. See Using variables in imports.

**NOTE:** The syntax of the SQL statement cannot be validated when using ODBC as the source.

### ODBC driver

Use of ODBC requires an ODBC driver to be installed on the following servers:

- For on-premise systems, the driver must be installed on the Axiom Software Application Server.
- For cloud service systems, the driver must be installed on the local server that is hosting the Axiom Software Cloud Integration Service.

The ODBC driver is specific to your source database. If you want to use ODBC with a particular database, that database vendor should provide or recommend an ODBC driver for use with that database.

## Import source: Intacct

**NOTE:** This import source is only available if you have licensed Intacct integration.

The **Source** tab of the Import Wizard uses the following settings when importing data from Intacct.

| Item | Description |
|---|---|
| Source | Select **Intacct** to read data directly from Intacct. |

### Connection

Complete the following connection information for the import source.

**NOTE:** This information is only used to connect to the source and extract the data into the import temptable. Transformation steps are not performed using these credentials.

| Item | Description |
|---|---|
| User ID | The user name to use to connect to Intacct. |
| Company ID | The company ID to use to connect to Intacct. |
| Password | The password to use to connect to Intacct. |
| Table Name | The Intacct table from which to read data. |
| Filter | A filter to limit the data to be read from the table.<br><br>The following SQL operators are supported in the filter: <, >, >=, <=, =, LIKE, NOT LIKE, IN, NOT IN. When doing NULL comparisons, use IS NOT NULL or IS NULL. Compound filters using AND and OR are supported, but joins are not supported. If the value you are filtering on contains an apostrophe, add a backslash before it to escape the apostrophe (for example: `contactname = 'Erik\'s Deli'`). |
| Column Names | A comma-delimited list of columns to be read from the table. Leave this blank to return all columns. |

See the Intacct Developer API Reference for more information on Intacct tables that can be queried, and any special requirements and considerations for each table. Tables can be queried if they support the ReadByQuery API method.

## Import source: Current Axiom database

Use one of the **Current Axiom database** options to import data from the current Axiom database. This feature might be used to automate the process of copying data from one table to another, or to transform the data within a table.

**NOTE:** Only administrators can create or edit an import that uses the current Axiom database as the source. Non-administrators do not see this option when creating or editing an import, and cannot open any existing imports that are configured to use this option. However, non-administrators can be given the right to execute an import that uses the Axiom database.

When you use one of these options, Axiom Software automatically creates the necessary connection string in the background when the import is performed. If the import is copied to another system or if the current database is restored to a different location, the new context is used to create the connection string.

You can select either of the following options:

- **Current Axiom system database**: Import data from a table in the system database.
- **Current Axiom audit database**: Import data from a table in the corresponding audit database.

**NOTE:** Systems with an Axiom Contract Management license can also import data from the **Current Contract Management database**. See the separate product documentation for more information on importing contract management data.

### SQL Select Statement

The SQL SELECT statement defines the data query to the source database, resulting in the set of data to be imported to the *temptable*. You can then perform mapping and transformations on the data before importing into the destination table.

In the **SQL Select Statement** box, enter any valid SQL statement to define the data query. You can click the **SQL editor** button to open the **Edit SQL** dialog. This dialog provides a text editor for entering and reviewing large SQL statements, and several tools to check the statement. For more information, see Creating the SQL SELECT statement.

Variables can be used in the SELECT statement. See Using variables in imports.

## Creating the SQL SELECT statement

If the import source is SQL Server, Oracle, OLEDB, or one of the current Axiom database options, then you must define a SQL SELECT statement to query the source database, resulting in the set of data to be imported to the *temptable*. You can use the **Edit SQL** dialog to create and test the SELECT statement.

To open the dialog:

- On the **Source** tab of the **Import Wizard**, click the browse button (**...**) to the right of the **SQL Select Statement** box.

The **Edit SQL** dialog provides a text editor for the statement, and also several tools to help create and test the statement.

### Creating the statement

You can type the statement into the text editor, or copy and paste from another source.

You can use the **Choose Table** tool to automatically generate a SQL statement that selects all columns in a specified table. You can then edit the statement to meet the specific data needs. To do this:

1. Click the **Choose source table to create SQL** button .
2. In the **Choose Table** dialog, select the table for which to generate the SQL statement, and then click **OK**.

   The **Choose Table** dialog lists all tables in the SQL Server database specified on the **Source** tab, including views.

The generated SELECT statement is placed in the text editor. Any existing text in the editor is overwritten.

### Testing the statement

**NOTE:** If the SQL statement uses variables, then these validation features are not available. Validation features are also not available if the source is OLEDB.

To validate the syntax of the SQL statement, click the **Check SQL syntax** button . Axiom Software sends the statement to your database server to see if the statement can be parsed, resulting in either a success message or an error message.

To view a set of sample records, click the **View data** button . Axiom Software queries the database and returns the first 100 rows in the **View Data** dialog. You can review this data to help determine if the SELECT statement is returning the desired set of data.

Within the **View Data** dialog, you can sort, filter, and group the sample data using standard Axiom grid functionality. If desired, you can click the **Export data to .CSV file** button to export the data to a CSV file .

When you are finished reviewing the data, click **Close** to return to the **Edit SQL** dialog.

## Import source file considerations

If the import source is a delimited file or an Excel file, then you must consider how the file is specified. There are two options: **Prompt for file location** and **Always use this file**.

### Prompt for file location

| | |
|---|---|
| **File permissions** | The file must be accessible by the user's file system permissions. |
| **Data flow** | • A copy of the file is streamed from the client machine to the application server.<br>• Import from the application server to the database server.<br>• The copy of the file on the application server is deleted after the import is complete. |
| **Ramifications / Limitations** | • Slower performance (file is copied multiple times)<br>• Only available when running the import interactively<br>• Limited to files less than 100MB |

### Always use this file

| | |
|---|---|
| **File permissions** | • File must be accessible by the application server's permissions<br>• File path must be a UNC path, not a mapped drive (meaning \\servername\foldername, not I:\foldername) |
| **Data flow** | The file is streamed from the file source to the database server for import. |
| **Ramifications / Limitations** | • Faster performance<br>• No practical limit to the file size<br>• Can be used in Scheduler |

# Variables tab

In the **Variables** tab of the Import Wizard, you can define variables to be used in the import. Import variables can be used in the following import settings:

- Source and destination columns (**Mapping** tab)
- Destination table (**Mapping** tab)
- Transform statements or functions (**Transforms** tab)
- SQL SELECT statement to the source table (**Source** tab)
- Source file (**Source** tab)
- Sheet name (**Source** tab)

When the import is executed, before any import steps are processed, Axiom Software checks the **Variables** tab for variables. If variables are defined (regardless of whether they are used in the import), the **Variables** dialog is presented to the user. Once the user has specified a value for each variable, the import begins processing.

For example, you might have database tables that have the year incorporated into the table name. You could define a variable for year, and configure the destination table to use the variable. Data would then be imported into the appropriate table, depending upon the year selected by the user.

To use a variable in the import, enter the variable name into one of the supported areas of the import settings, enclosed in curly brackets {}. For example, if the variable name is "column", you would enter `{column}`.

**NOTE:** If the variable defines the destination table, then you must place the variable in double curly brackets when you use it in a SQL statement, so that the eventual table name value is enclosed in curly brackets as expected. For example, if you have a variable named "destinationtable", you would reference that variable as `{{destinationtable}}`. That way, when the `{destinationtable}` value is defined, it will resolve as `{GL2018}`.

## Managing import variables

This tab has two sections. In the top section, you can create user-defined variables for the import. The bottom section displays the built-in system variables that can be used in the import, such as *temptable*. This **Built-in Variables** section is for reference only.

- To add a variable, click **Add variable** ✚. The new variable row is added below the row that is currently selected.
- To edit a variable, type the changes into the grid.

  If you change the name of a variable, you must update any references to that variable in the import to use the new name.
- To delete a variable, select the row that you want to delete and then click **Remove variable** ✕.

  Make sure the variable is not used in the import before deleting it. If the import references a variable that is not defined, an error will result when executing the import.

The following settings are defined for variables:

| Item | Description |
|---|---|
| Prompt for value even if variable only contains one choice | This setting applies to all defined variables for the import, and determines the system behavior when a variable has only one choice.<br>• If enabled (default), then users will always be prompted to select variable values, even if a variable has only one defined choice.<br>• If disabled, then users will not be prompted to select variable values for variables that have only one defined choice. |

| Item | Description |
| --- | --- |
| Name | The name of the variable.<br><br>Import variable names cannot use the same name as transform variable names, and vice versa. Every variable name within the import must be unique.<br><br>**NOTE:** Import variables cannot use the same names as tables defined in your system. This is because the syntax for referencing tables is the same as the syntax for referencing variables. A validation message will display in the Import Wizard if a variable name matches a table name. If a table is later created with the same name as an existing variable, then subsequent executions of the import will fail with an error identifying the table name / variable name duplication. |
| Choices | The set of valid choices for the variable, separated by semi-colons.<br><br>You can leave the choices blank if there is not a defined set of values for the variable. When the import is executed, the user can type in a value for the variable (if **Allow Free Input** is enabled).<br><br>You can also specify a column or columns in the database by entering fully qualified Table.Column syntax. The user will be presented with a drop-down list of all values (up to 500) in the specified column. For example, enter DEPT.Region if you want the user to select from the list of regions in the DEPT table.<br><br>You can specify multiple database columns, separated by semi-colons. For example, DEPT.VP; DEPT.Mgr. The column values will be collected into a single list. |
| Description | Optional description text that displays in the **Variables** dialog. This dialog displays when the import is executed, to prompt the user to specify a value for the variable.<br><br>Description text displays as follows above the variable selection drop-down list:<br><br>*VariableName*:*DescriptionText* |
| Allow Free Input | Specifies whether users can type in their own values for the variable, or if they are restricted to the list of choices.<br><br>• If enabled (default), then users can either select from the list of choices, or type in their own value. Keep in mind that the user-defined value may not be valid in the context of the import.<br>• If disabled, then users can only select from the defined list of choices. |

# Mapping tab

The **Mapping** tab of the Import Wizard maps the import data to the destination table. In this tab, you define:

- The columns to be created in the *temptable*, including any "work columns" to be used for interim calculations only.
- The destination table for the imported data.
- The destination columns for the imported data.

Data is first imported from the source file or table into the temptable. Each entry in the **Temp Table Column** field becomes a column in the temptable. After performing any mappings or calculations on the temptable (as defined in the transforms), data is imported from the temptable to the destination table. The entries in the **Destination Column** field determine whether a column of data is imported to the destination table, and to which column in the destination table.

This tab has two sections. The top section is for mapping imported data columns from the source file or table, and the bottom section is for work columns.

Variables can be used in the **Destination table** field, the **Temp Table Column** field, and the **Destination Column** field. See Using variables in imports.

## Imported column mappings

The top section of the **Mapping** tab must contain a row for each column of data to be imported.

The columns must be listed in the order of the columns in the source. For example, the first column of the source file must be row 1 in the mapping grid, the second column must be row 2, and so on.

You can click **Auto-generate temp table and destination columns** to auto-generate an entry for each column in the source table or file. The import source settings must be complete and accurate to do this, and you must have specified the destination table. Axiom Software will attempt to auto-populate the data types and destination columns for each column of data in the source table or file. After auto-generating, review all entries to be sure they are correct.

**NOTE:** If the import source is set to **Prompt for file during execution**, then when you click **Auto-generate**, you will be prompted to select a file to use for the auto-generated mappings.

You can also define columns manually, or edit the settings after auto-generating:

- To add a row, click **Add imported column mapping** . The new row is added below the row that is currently selected.
- To edit a row, type changes into the grid.
- To delete a row, select the row that you want to delete and then click **Remove mapping** .
- To change the order of rows, select the row that you want to move and then click the arrow icons to move it up or down.

**NOTE:** If you perform any action that changes the existing order of rows, this may cause data to be imported incorrectly. After making manual adjustments to the grid, check to make sure that each mapping row in the grid matches up with the appropriate source column.

### Mapping settings

| Item | Description |
|---|---|
| Destination table | The destination table for the imported data.<br><br>You can also use an import variable if you want the destination table to be determined by a variable. |
| Source Column | The number of the corresponding source column in the import source. The first row in the grid corresponds with the first column in the source, and so on.<br><br>These numbers cannot be edited. If you auto-generate the mappings, the name of the source column will display next to the number for reference. Names are only displayed when the import source is a database or a file with a header row. |
| Temp Table Column | The name of the column to create in the temptable to hold the imported data.<br><br>The column name does not have to match the name of the column in the source. The data is imported in column order, not by name.<br><br>Import variables can be used to define temptable column names.<br><br>**NOTE:** Temptable column names must follow the same rules as normal table columns. See Table and column naming requirements. Keep in mind that if you auto-generate the column mappings, the temptable column names are based on the headers in the source. These headers may contain spaces or other invalid naming conventions that should be manually corrected. |

| Item | Description |
| --- | --- |
| Type | The data type of the column. For more information on the available column data types, see Column properties.<br><br>If the data type is **String**, you must also specify the maximum length of the string field. This entry should match the string length of the destination column so that data is saved appropriately.<br><br>To specify the string length, click the browse button (**...**) to the right of the field. In the **Edit String Length** dialog, type the string length (from 1 to 4000). The string length displays in parenthesis after the data type. For example: **String (200)**.<br><br>The type must match the type of the destination column. The type is automatically selected when you use the auto-generate feature or when you manually select a destination column. You only need to manually select a type if the column is only for the temptable and does not have a destination column.<br><br>**NOTES:**<br>• To create a string column with unlimited size, leave the string field blank. You should only do this if you understand the ramifications. See Column properties.<br>• When importing numeric values, the number of digits in the import source cannot exceed the number of digits allowed by the data type.<br>• If a string column has a destination column, the column in the temptable automatically matches the Unicode status of the destination column (True/False). However, if the string column does not have a destination column, Unicode is assumed as True. |

| Item | Description |
| --- | --- |
| Nulls | Determines how blank values in the import source are brought into the temptable column.<br>• If disabled (the default behavior), then blank values are brought into the temptable column as the default value for the specified destination column. If the column is unmapped, then the system default value as defined for the column type is used.<br>• If enabled, then blank values are brought into the temptable column as null values.<br>This setting can impact transform statements that look for "blank" or undefined values in the temptable. You should set this as appropriate depending on whether you want to check for null values or the default value for the affected column.<br>The presence of null values in the temptable does not necessarily mean that the imported data will contain null values. When the temptable values are imported into the destination table, the default values for the destination columns are always applied to any null values at that point. However, if the default value for a destination column is null, then the null values will be retained. |
| Destination Column | The name of the column in the destination table where you want this data to be saved.<br>You can type the name of the column directly, or use the drop-down list to select a column name. The data type of each column is displayed in the drop-down list for reference. The data type of the destination column must match the data type specified in the mapping grid.<br>If this column is not intended to be saved to the destination table, select **<not mapped>**. "Not mapped" is the default if you do not explicitly select a destination column.<br>Import variables or transform variables can be used to define destination column names. If you are using a variable for a destination column and you want to set the column to not mapped, the value of the variable must be blank for import variables and empty string ('') for transform variables). |

### Work column mappings

You can use work columns in the temptable to perform calculations and mapping on the data before saving it to the destination table. Any column defined in the **Work column mappings** section will be created in the temptable. If a destination column is specified, the data will be saved to the destination table.

- To add a row, click **Add work column mapping** ✚.
- To edit a row, type changes into the grid.
- To delete a row, select the row that you want to delete and then click **Remove mapping** ✕.
- To change the order of rows, select the row that you want to move and then click the arrow icons to move it up or down.

#### Mapping settings

| Item | Description |
|---|---|
| Temp Table Column | The name of the work column.<br><br>Import variables can be used to define temptable column names.<br><br>Temptable column names must follow the same rules as normal table columns. See Table and column naming requirements. |
| Type | The data type of the work column. For more information on the available column data types, see Column properties.<br><br>See the Type entry in the previous table for more information. |
| Null | Determines the starting values for work columns in the temptable (before transforms are applied).<br>• If disabled (the default behavior), then the work column starts with the default value for the specified destination column. If the column is unmapped, then the system default value as defined for the column type is used.<br>• If enabled, then the work column starts with null values.<br><br>This setting can impact transform statements that look for "blank" or undefined values in the temptable. You should set this as appropriate depending on whether you want to check for null values or the default value for the affected column.<br><br>The presence of null values in the temptable does not necessarily mean that the imported data will contain null values. When the temptable values are imported into the destination table, the default values for the destination columns are always applied to any null values at that point. However, if the default value for a destination column is null, then the null values will be retained. |

| Item | Description |
| --- | --- |
| Destination Column | The name of the column in the destination table where you want this data to be saved. If this column is used only for calculations in the temptable and is not intended to be saved to the destination table, select **<not mapped>**.<br><br>See the Destination Column entry in the previous table for more information. |

### Remarks

- If you select to auto-generate the column mappings and either the import source or the destination table uses variables, then the **Variables** dialog opens so that you can specify variable values to use to generate the mappings.
- If the destination table uses a variable and you are manually creating column mappings, then the selections in the destination column drop-down lists are based on the first choice listed for the variable.

# Transforms tab

The **Transforms** tab of the Import Wizard contains a set of statements to perform actions on the data in the temptable, in order to transform the data before saving it to the destination table.

When the import is executed, the transforms are processed after import variables have been selected and after the data query has been made to the source table or file. Transforms are processed in the order listed in the tab.

Transforms can use SQL statements or built-in Axiom transform functions.

### Managing transforms

- To add a transform, click **Add transform** ✚. The new transform is added below the row that is currently selected.
- To duplicate a transform, select the row that you want to duplicate and then click **Duplicate selected transform**.
- To edit a transform, make changes directly in the grid.
- To delete a transform, select the row that you want to delete, and then click **Remove transform** ✕.
- To change the order of transforms, select the row that you want to move and then use the arrow icons to move it up or down.

The following settings are defined for each transform:

| Field | Description |
|---|---|
| Number | The ordinal number assigned to each transform, to determine processing order. This setting is managed by Axiom Software. When you change the order of rows, Axiom Software automatically changes the order number. |
| Transform | A SQL statement, or a built-in Axiom import function.<br><br>To define the transform, double-click the field or click the browse button (**...**) in the field. The **Edit Transform** dialog opens so that you define the transform. See Defining transform statements. |
| Description | Displays the description of the transform. This is for reference only. You can define this description when defining the transform. |
| Target Variable | Displays the name of the target variable for the SQL transform.<br><br>For SQL statements, target variables are defined when editing the statement. Target variables do not apply to import functions. See Defining transform statements. |
|  | Select the **Pause** check box if you want the import to pause and display the temptable after processing this transform. This option only applies if the user selects **Allow pauses** when executing the import. For more information, see Troubleshooting imports using pause. |
|  | Select the **Disable** check box to disable the transform. When the import is processed, this transform will be skipped. |

### Transform notes

If the import contains a transform that zeroes old data before importing the new data, the **Pre-Save Validate** function should be used before the zero step. This allows you to identify any data issues before deleting existing data. If invalid data is found, the import is aborted and the zero step will not be processed.

## Defining transform statements

For each transform listed on the **Transforms** tab of the **Import Wizard**, you must use the **Edit Transform** dialog to define the associated SQL statement or transform function. To open the dialog, click the **...** button in a transform field.

At the top of the **Edit Transform** dialog, specify whether the transform is a **SQL** statement, or a **Built-in Function**. By default, **SQL** is selected.

### Defining a SQL statement transform

If the transform is a SQL statement, complete the following:

| Field | Description |
| --- | --- |
| Enter a SQL statement | Type or copy and paste the SQL statement into the text editor. To validate the syntax of the SQL statement, click the **Check SQL syntax** button  .<br><br>**NOTE:** When referencing Axiom Software tables in a SQL statement, place the table name in curly brackets (like you do with variable names). For example, to reference the table DEPT, use `{DEPT}`.<br><br>The SQL statement can use import variables and/or transform variables (that were defined in a previous transform statement). See Using variables in imports. If the statement includes variables, then the ability to check the SQL syntax is not available.<br><br>Variables can be used in calculations in the SQL statement, for example, to calculate the value of a second variable based on the first variable.<br><br>**NOTE:** If the variable defines the destination table, then you must place the variable in double curly brackets when you use it in a SQL statement, so that the eventual table name value is enclosed in curly brackets as expected. For example, if you have a variable named "destinationtable", you would reference that variable as `{{destinationtable}}`. That way, when the `{destinationtable}` value is defined, it will resolve as `{GL2018}`.<br><br>For some examples of common SQL transforms, see Example SQL statements for transforms. |
| Target Variable Name | If you want to define a target variable for the transform, type the name of the variable.<br><br>The SQL statement can be any query that results in a single value. If the query returns multiple values, the target variable uses the first value in the return set.<br><br>The target variable can be used in subsequent transforms, or as destination columns in the **Mapping** tab. To use the variable in the import, enter the variable name into one of the supported areas of the import settings, enclosed in curly brackets {}. For example, if the variable name is Column, you would enter `{column}`.<br><br>Transform variable names cannot use the same name as import variable names, and vice versa. Every variable name within the import must be unique.<br><br>Transform variables should not use the same name as an Axiom Software table that you plan to reference in the import, because the syntax for table names is the same as for variables (both are placed in curly brackets). If a variable name and a table share the same name, any references will be interpreted as the variable, not the table. |

| Field | Description |
| --- | --- |
| Description | If desired, type a description for the transform.<br><br>The description may be useful when viewing the list of transforms on the Transforms tab, to easily identify specific transforms. |

### Using a transform function

If the transform is a built-in function, select the function that you want to use from the **Function** list, and then complete the parameters for the function.

- Add new dimension elements
- Custom data validation
- Delete rows
- Pre-save validate
- Update temp column

You can edit the description for the transform if desired. By default, the standard description of the function is used.

**NOTE:** If you select a different function, the current description will be overwritten with the standard description of the new function. If you go back to the original function before saving the transform, the original description will be preserved.

## Add new dimension elements during an import

To add new dimension elements during an import, create a transform step that uses the built-in function **Add new dimension elements**. You can use this function to add new accounts, departments, or other dimensions as part of the import.

This function is intended to be used in cases where the import data may contain new dimension elements that have not yet been added to the relevant reference tables in Axiom Software, and the organization wants these records to be added automatically as part of the import. Other organizations may prefer to prevent these records from importing and instead add the new dimension elements manually, in which case this function should not be used.

If new dimension elements are added by this function, then after the import is complete a system administrator will most likely need to edit the reference table in order to fill in grouping columns for the new element. This is why the function supports notifying one or more users of the added dimension elements. Keep in mind that it is possible for invalid dimension elements to be added when using this function, if the source data for the import is not correct.

### Function parameters

This function uses the following parameters:

| Parameter | Description |
| --- | --- |
| Table | The name of the dimension table to update. This can only be reference tables that have a single-level lookup relationship with the destination table for the import.<br><br>For example, if you are importing data into GL2018, and that table has columns Acct and Dept which have lookup relationships with the Acct and Dept tables, then you can add new dimension elements to the Acct and Dept tables.<br><br>When this transform step is performed, the data in the temptable will be validated against this dimension table. If any dimension elements are found that do not match the dimension table, those new dimension elements will be added to the dimension table. |
| Column defaults | Optional. Values to use for grouping columns in the target dimension table when new dimension elements are added. You can add as many column=value pairs as needed, separated by commas. See the discussion following the table for more information on the specific syntax.<br><br>Any columns in the target dimension table that are not listed here will use the column's default value (as defined in the column properties) when the new record is added.<br><br>If the target dimension table contains any validated columns , then you must do one of the following:<br>• List the validated column in this parameter with a valid value. That valid value will be used when the new records are added to the table.<br>OR<br>• Make sure the validated column has a valid default value in its column properties. If the validated column is not listed in this parameter, then its default value from the column properties will be used when the new records are added to the table.<br><br>If the validated column is not assigned a valid value using one of these options, then an error will occur when this transform is processed and the import cannot continue. |
| Email notification | Users and or roles to notify via email when new dimension elements are added to the table. Enter a list of one or more user and role names, separated by commas. |
| Task pane notification | Users and or roles to notify via the Notifications task pane when new dimension elements are added to the table. Enter a list of one or more user and role names, separated by commas. |

### Defining column defaults

You can populate a grouping column using a fixed default value, or by using a value from a column in the import temptable.

- To use a fixed default value, use the following syntax:

  *GroupingColumnName='StringValue',GroupingColumnName=NumericValue, etc.*

  Where *GroupingColumnName* is the grouping column in the target dimension table. If the grouping column is a string column, then the value must be placed in single quotation marks, just like when writing a filter statement.

- To use a value from a column in the import temptable, use the following syntax:

  *GroupingColumnName*={temptable}.*TempTableColumnName, etc.*

  For example, if the temptable contains a column named Desc that you want to use to populate the Description column in the target dimension table, you would enter: `Description={temptable}.Desc`

**NOTE:** If the default values are populated from a column in the import temptable, those values must be the same for all instances of a particular dimension element. If the same dimension element has multiple rows in the temptable with different values in the specified temptable column, then the maximum value will be placed in the grouping column for that dimension element.

### Example

To add new departments when importing GL actuals data, the function parameters could look as follows:

| | |
|---|---|
| **Table:** | Dept |
| **Column defaults:** | Description={temptable}.Desc, DeptStatus='New' |
| **Email notification:** | sysadmins |
| **Task pane notification**: | jdoe |

When the import is run, any departments that do not already exist in the DEPT table will be added to that table. The Description column in the DEPT table will be populated with the value from the Desc column in the import temptable, and the DeptStatus column in the DEPT table will be populated with the string "New". An email notification of the added departments will be sent to users in the role sysadmins, and a task pane notification of the added departments will display for user jdoe.

All other columns in the DEPT table will use the default value defined for the column in the column properties.

## Deleting rows of data as part of an import

To delete rows from a target table during an import, create a transform step that uses the built-in function **Delete rows**. You can use this function to "clear" a table before importing new data.

**NOTE:** If you are only updating specific columns in the destination table, then you may want to use a SQL step to zero data rather than using the delete function.

### Function parameters

| Parameter | Description |
|---|---|
| Table | The name of the table from which to delete rows. |
| Filter | Optional. A filter used to identify rows to delete. If omitted, all rows in the table are deleted.<br><br>The filter can be against the specified table or against a lookup reference table. Standard Axiom filter criteria syntax applies.<br><br>**NOTE:** The temptable cannot be referenced in the filter. |

For example, if the table is GL2018, then the filter could be something like:

```
GL2018.YrMo='{YrMo}'
```

Where the value of YrMo is defined by an import variable.

```
Dept.Region='North'
```

Where the GL2018 table has a column Dept that looks up to the Dept table.

## Updating data in the temptable based on another column

To update data in the temptable based on another column, create a transform step that uses the built-in function **Update temp column**. This function updates a column in the temptable from a specified column in another table, using the given match key. For example, this function would typically be used to look up the credit reversal sign from the Account table.

### Function parameters

| Parameter | Description |
|---|---|
| Temp table column | The name of the column in the temptable to be updated. |
| Source column | The name of the column to use to update the temptable. Fully qualified Table.Column syntax must be used. |
| Match columns | The columns to use to match data to update, in the format *TempColName* = *SourceColName*. Separate multiple match sets with commas.<br><br>For example: `ACCT = ACCT, DEPT = DEPT` |

| Parameter | Description |
| --- | --- |
| Temp table filter | Optional. A filter used in the SQL WHERE clause against the temptable, to identify rows to update. If omitted, all matching rows are updated.<br><br>Fully qualified Table.Column syntax must be used. The temptable must be represented as a variable.<br><br>For example: `{temptable}.ACCT > 20000`<br><br>**NOTE:** If a temp table column uses a database reserved word (such as "Key") then in the filter you must place that column in double quotation marks and use all upper-case letters. Use of database reserved words should be avoided whenever possible. |

## Validating data to be imported before the save

To validate data against lookup columns *before* performing the save for the import, create a transform step that uses the built-in function **Pre-save validation**. This function takes no parameters.

This is the same validation process that occurs automatically as part of the actual save for the data import. By performing the validation before the save, you can check for invalid data and abort the process before performing irrevocable data changes in subsequent transform steps (such as deleting old data in the destination table). If instead you want to perform a different custom validation, see Performing custom data validation for an import.

If any destination columns in the import are validated columns, this function validates the data in the temptable against the lookup columns. If exceptions are found, an error message is written to the import log and the import is aborted.

**NOTE:** If this function is used in an import, then the option to **Ignore lookup and key errors** does not apply. If invalid data is found in the pre-save validation step, then the import is automatically aborted and does not proceed to the data save.

## Performing custom data validation for an import

To exclude rows of data from an import based on a custom criteria, create a transform step that uses the built-in function **Custom data validation**. Any data in the temptable that matches the specified filter will be excluded from the save.

This custom validation is separate from the built-in lookup validation that occurs as part of the data save (or by use of the Pre-Save Validate function).

**Function parameters**

| Parameter | Description |
|---|---|
| Filter | A filter used in the SQL WHERE clause against the temp table. Any rows that match this filter are *excluded* from the save, and will be reported in the import error log with any other validation errors. |
| Failure Message | An error message to display next to records that are excluded due to this data validation step.<br><br>Errors are logged in a column named **AXTRANSFORM_*StepNumber***, where *StepNumber* is the number of the associated transformation step. For example: "AXTRANSFORM_5" when the associated transform is step 5 in the list. |

For example, you might want to check to make sure that data exists in a particular column of the temptable before importing that record. If the necessary data is missing, then that record will be excluded from the import. In this case the function settings would be something like the following:

**Filter:** Benchmark=0

**Failure Message:** The Benchmark data is missing

Where Benchmark is the column in the temptable that must have a value in order to import the record.

## Example SQL statements for transforms

The following example SQL statements can be used to perform common transforms for imports.

### Converting data types

**Convert a number to a string to populate into a string column**

```
UPDATE {temptable} SET RATESTRING = CONVERT(NVARCHAR, RATENUM) WHERE
RATENUM IS NOT NULL
```

If Unicode is disabled for the string column, then convert to varchar instead.

**Convert a string to a number to populate into a numeric column**

```
UPDATE {temptable} SET RATENUM = CONVERT (BIGINT, RATESTRING) WHERE
RATESTRING IS NOT NULL
```

or

```
UPDATE {temptable} SET ACCT=CAST (tAcct as BigInt)
```

If you are using Integer 32 or Integer 16, then convert to int or smallint instead.

**Convert a datetime field to a concatenated string (yearmo)**

```
UPDATE {temptable} SET YEARMO = YEAR(DateTimeField) * 100 + MONTH
(DateTimeField)
```

**Convert a datetime field to a concatenated string (yearmonthday)**

```
UPDATE {temptable} SET YEARMONTHDAY = (YEAR(DateTimeField) * 100 + MONTH
(DateTimeField)) * 100 + DAY(DateTimeField)
```

**Other conversion statements**

```
CONVERT(STRING(xx), RATENUM)

CONVERT(BIGINT, RATESTRING)

CONVERT(DECIMAL(28,14), RATESTRING)
```

## Math transformations

**Convert values to a negative number if a credit column exists in the temptable**

```
UPDATE {temptable} SET M1 = - M1, M2 = -M2, M3 = -M3, M4 = -M4, M5 = -M5,
M6 = - M6, M7 = -M7, M8 = -M8, M9 = -M9, M10 = -M10, M11 = -M11, M12 = -m12
WHERE Credit = 'C'
```

**Perform math on a field if another field in the temptable contains a value**

```
UPDATE {TempTable} SET M1 = M1 * Rate WHERE Rate <> ''
```

**Round a value to 2 decimals and replace the value**

```
UPDATE {temptable} SET RATE = ROUND(RATE, 2)
```

**Divide two integer numbers and keep the decimal**

Remember that an integer divided by an integer returns only an integer (example: 5/7 = 0). If you want to capture the decimal remainder, you must cast the integer values as decimal:

```
CAST(INT1 AS DECIMAL(28,14)) / CAST(INT2 AS DECIMAL(28,14))
```

## Pivoting data

**Pivot incoming data with respect to time**

```
UPDATE {temptable} SET
  M1 = case when (TheMonth=1) then Amt else 0 end,
  M2 = case when (TheMonth=2) then Amt else 0 end,
  ETC.
```

**Pivot data from columns to rows (using a monthly variable)**

```
Update {temptable} set Amt =
  Case
    When ({VarMonth}=1) then M1
    When ({VarMonth}=2) then M2
    ETC.
    Else 0
  End
```

## General temptable transformations

**Perform a find and replace in a data column to detect the # character and replace it with nothing**

```
UPDATE {temptable} SET AcctDesc = REPLACE(AcctDesc,'#','')
```

**Place zeros in a field rather than null values**

```
UPDATE {TempTable} SET Rate = 0 WHERE Rate IS NULL
```

**Insert new records into the temptable by summarizing transaction detail records**

```
INSERT INTO {temptable} (DEPT,ACCT,TRANSID,M1,M2,M3,DELETE) SELECT
DEPT,ACCT,'Summarized',SUM(M1),SUM(M2),SUM(M3),'DeleteFlag' FROM
{temptable} GROUP BY DEPT,ACCT
```

**Add an identity or row number to each record, using the system column AxReference**

```
UPDATE {temptable} SET MYROWNUMBER={temptable}.AxReference
```

**Delete records from the temptable with a flag set**

```
DELETE from {temptable} where FLAG = 'DeleteFlag'
```

**Delete records from the temptable where the dimension combination already exists in the destination table**

This example might be used to load only new transactional records and leave existing destination records untouched.

```
DELETE FROM {temptable} WHERE {temptable}.DEPT IN (Select DEPT from
{TRANSACTIONTABLE} group by DEPT) and {temptable}.TRANSID IN (Select
TRANSID from {TRANSACTIONTABLE} group by TRANSID)
```

**Concatenate strings together with a hyphen in between**

```
UPDATE {temptable} SET DESCRIPTION = DESC1 + ' - ' + DESC2
```

**Fill an entire column with the same value**

```
UPDATE {temptable} SET RATE = 0
```

**Copy the left 5 characters into a new column**

```
UPDATE {temptable} SET ShortDesc = LEFT(LongDesc,5)
```

**Subtract 1 character from the end of a field and copy the remaining text to a new column**

```
UPDATE {temptable} SET ShortDesc = LEFT(LongDesc, (LEN(LongDesc) - 1))
```

**Aggregate balance sheet records**

```
UPDATE {temptable} SET M1=BegBal+M1
UPDATE {temptable} SET M2=M2+M1
UPDATE {temptable} SET M3=M3+M2
UPDATE {temptable} SET M4=M4+M3
UPDATE {temptable} SET M5=M5+M4
UPDATE {temptable} SET M6=M6+M5
UPDATE {temptable} SET M7=M7+M6
UPDATE {temptable} SET M8=M8+M7
UPDATE {temptable} SET M9=M9+M8
UPDATE {temptable} SET M10=M10+M9
UPDATE {temptable} SET M11=M11+M10
UPDATE {temptable} SET M12=M12+M11
```

**Import into only the current period using a variable**

On the Mapping tab, for each column where you want to use a variable, set the destination column as {M1}...{M12} instead of the normal M1 … M12.

On the Transforms tab, create 12 statements—one for each data column that you set up with a variable. The SQL for each statement is as follows (each statement will have a different current period value and a matching column name):

```
SELECT RESULT =
  CASE
    WHEN {CurrentPeriod} = 1 THEN 'M1'
    ELSE ''
  END
```

In the **Target Variable Name** section of the transform, type the name of each variable. In the text above, the variable name would be M1. The next statement would be M2, then M3, and so on. This will set the variable value to the result of the SQL statement. Columns that do not match the current period will be set to '', which means they will not be imported to the destination table.

**Abort the import if no data exists to import**

```
IF (SELECT COUNT(*) FROM {temptable}) = 0

RAISERROR ('Temptable was empty, aborting import', 11, 0)
```

### Updating a table other than the temptable

**Update a reference table with new elements that currently exist in the temptable, but not in the reference table**

Whenever possible, the new built-in function **Add new dimension elements** should be used for this instead of a SQL statement. See Add new dimension elements during an import. If this function cannot be used, then use a SQL statement like the following:

```
INSERT INTO {ACCT}(ACCT, Description) Select ACCT, 'Exception from Import'
FROM {temptable} WHERE NOT EXISTS (SELECT ACCT FROM {ACCT} WHERE
{temptable}.ACCT = {ACCT}.ACCT GROUP BY ACCT) GROUP BY ACCT
```

**Zero out columns in a destination based on a flag in a reference table**

```
UPDATE {{destinationtable}} set M1=0,M2=0,M3=0,M4=0,
M5=0,M6=0,M7=0,M8=0,M9=0,M10=0,M11=0,M12=0 FROM {{destinationtable}} INNER
JOIN {ACCT} ON {ACCT}.ACCT={{destinationtable}}.ACCT WHERE
{ACCT}.ZEROME='1'
```

This example assumes that {destinationtable} is an import variable that resolves to a table name.

**Delete large numbers of records from a table**

```
DECLARE @RowsDeleted INTEGER
SET @RowsDeleted = 1
WHILE (@RowsDeleted &gt; 0)
  BEGIN
    DELETE TOP (10000) FROM MyTable [WHERE .....]
    SET @RowsDeleted = @@ROWCOUNT
  END
```

The WHERE clause in the DELETE step is optional.

# Execute tab

Using the **Execute** tab, you can define execution settings and run the import.

**NOTE:** This tab is not visible to non-admin users who do not have **Execute** permissions for the import.

### Execute Options

The following options affect the execution of the import.

| Option | Description |
|---|---|
| Allow pauses | Specifies whether pauses are honored during processing.<br>• If enabled, then Axiom Software honors the pause settings on the **Transforms** tab. If a step has **Pause** enabled, then the import pauses after performing the step, and displays the temptable in the **View Data** dialog. When the dialog is closed, the import continues to the next step. See Troubleshooting imports using pause.<br>• If disabled (the default setting), then the import will continue without pausing, regardless of whether any steps are flagged with **Pause**.<br>This option is only available to users with read/write access to the import. This option only applies to the current execution of the import; it is not saved in the import settings. |
| Preview only | Specifies whether the import will be run in "preview mode."<br>• If enabled, then all steps of the import are performed *except* for the final step where data is saved to the destination table. This option is for testing purposes, so that you can test the import settings without actually saving data.<br>• If disabled (the default setting), then the full import is run and data is saved to the destination table.<br>**IMPORTANT:** When an import is run in preview mode, all of the steps in the **Transforms** tab are performed, including any steps that modify tables other than the temptable. If you do not want these steps to occur as part of a preview, then you should disable the step on the **Transforms** tab before executing the preview.<br>This option only applies to the current execution of the import; it is not saved in the import settings. |

| Option | Description |
| --- | --- |
| Ignore lookup and key errors | Specifies the import behavior if lookup and key errors exist in the temptable data before saving. These errors include invalid lookup data, blank keys, and duplicate keys (if saving to a reference table or if **Aggregate rows on final save** is disabled).<br>• If enabled, then any rows with lookup and key errors are ignored, and only valid rows are saved to the destination table. Once the import is complete, an error log is provided to detail the invalid rows. For more information, see Import validation errors.<br>This option allows you to continue to save valid data even if invalid data is present. You can then investigate the invalid data, make corrections, and reimport.<br>• If disabled (the default setting), then the import is aborted if any lookup and key errors exist in the temptable data. No data is saved to the destination table.<br>This option is only available to users with read/write access to the import. This option is saved in the import settings and will apply to future executions of the import by default.<br>**NOTES:**<br>• If this option is enabled and the import is executed by Scheduler, the execution status of the job is set to Partial Success if any errors are found. This will result in an email notification if the job is set to notify only on error.<br>• This option does not apply if a **Pre-Save Validation** function is used in the import's transform steps. If this function is used, then the import is aborted if any validation errors are found and no further import steps are processed. |

| Option | Description |
| --- | --- |
| Aggregate rows on final save | Specifies whether duplicate rows are aggregated during the final save to the destination table. Duplicate rows are rows that have the same key column values.<br>• If enabled (the default setting), then duplicate rows are aggregated before saving data to the destination table. This aggregation process may take some time for large imports.<br>• If disabled, then the temptable data is not aggregated before saving data to the destination table. If any duplicate rows exist, the import is aborted and no data is saved to the destination table. You can optionally use the **Ignore lookup and key errors** option to instead exclude the invalid rows and only import valid rows.<br>Disabling this option improves import performance for large imports where aggregation is not necessary.<br>This option is only available to users with read/write access to the import. This option is saved in the import settings and will apply to future executions of the import by default.<br>This option only applies when saving data to a data table. If you are saving data to a reference table, rows are never aggregated and duplicate keys are not allowed. |

### Description

You can optionally enter a description for the import. The description can be used to document the purpose of the import and/or to detail important import instructions. The description is limited to 2000 characters.

### Executing the import

Click **Execute** to run the import. If you want to stop the execution, click **Stop**.

The **Execution log** displays the process steps of the import as it executes. If errors occur, they are displayed in the log.

For more information, see Executing import utilities.


# Exporting Data

Using an export utility, you can export data from an Axiom table to an external database or to a delimited file.

When you create an export, you specify a customer-defined table in Axiom Software from which to export data. You can export all of the data in the table, or you can limit the data by selecting columns to export or by defining a data filter. Then, you define the destination properties, such as connection information for the target database and table, or the name and location of the destination file.

When an export is executed, Axiom Software extracts the data from the source table in the Axiom database, and then creates or updates the destination table in the target database or creates the destination file.

**NOTES:**

- The ability to export data to a delimited file is only supported for cloud systems using a remote data connection. As an alternative, on-premise systems can use the File Processing feature instead.
- The File Processing feature also supports the ability to create a delimited export file, using the **Export to File** processing option. Instead of extracting data directly from a table, the desired data is first populated into an Axiom report, and then extracted from that report. The data can be manipulated within the report as needed before it is extracted.
- Export utilities do not have any means of transforming data as part of the export; however this can be accomplished in conjunction with an import utility. For more information, see Transforming export data.

## Transforming export data

Export utilities do not provide any method for manipulating or transforming the data to be exported—data is exported as is. If you need to transform data before exporting, you can do the following:

- Set up an import utility that imports data from an Axiom table to another Axiom table (the source table for the export). This import can transform the data as needed.
- Set up an export utility that exports data from the second table (the destination table for the

import) to the target external database or delimited export file.

- Set up a Scheduler job that performs the import first, followed by the export. You can then either schedule this job for recurring execution, or you can run the job as needed.

For example, imagine that you want to export data from the BGT2019 table to another system in your organization, and that this data needs to be transformed in some way before it can be exported to the target database. You would first create an import utility that imports data from BGT2019 to another table named something like BGT2019_Export. This import utility would perform all the necessary data transformations. You would then create an export utility that exports data from BGT2019_Export to the target database or export file.

To ensure that dependent utilities are run in sequence, you can create a Scheduler job that runs the import utility before running the export utility. When setting up the import task in the job, make sure that the following setting is *disabled*: **If this task fails, continue executing subsequent tasks**. This way if the import task fails, the export task will not be performed.

# Managing exports

Export utilities can be created, edited, or deleted using the Export Library.

**To access the Export Library:**

- On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**. The Export Library is available in the Libraries section.

  You can also use the Explorer task pane to access the Exports Library.

**NOTE:** Only administrators and users with the **Administer Exports** security permission can create exports. Other users may be granted access to execute and/or edit specific exports.

For details on executing an import, see Executing import utilities.

## Creating an export

1. Right-click the **Export Library**, and then click **New > Export Package**.
2. In the **Create New Export** dialog, select one of the following and then click **OK**.
   - **Create from scratch** (default): Create a new export starting with blank export settings.
   - **Create from existing**: Create a new export by copying an existing export. If you select this option, then select the export that you want to copy from the list in the bottom of the dialog. All settings for the existing export will be copied to the new export as a starting point.

   The Export Wizard dialog opens. You can move between tabs in any order, however, before you can save the export, all required settings must be completed and no invalid settings must be

present. If errors exist, an error message displays at the bottom of the dialog; you can click the error link to be taken to the tab with the error.

3. At the top of the dialog, enter a **Name** for the export.

   The export name is how users access and execute the export. If this export is a copy of an existing export, the default name is "Copy of *ExportName*".

4. In the **Source** tab, specify the source table for the export and then complete the source settings. For more information, see Configuring the export source data.

   **NOTE:** Export utilities do not have any means of transforming the source data as part of the export; however this can be accomplished in conjunction with an import utility. For more information, see Transforming export data.

5. In the **Destination** tab, specify the destination type (database or delimited file) and then complete the destination settings. For more information, see Configuring the export destination.

6. In the **Execute** tab, configure additional export options as appropriate. In this tab, you can define a general description for the export and configure settings relating to the destination table. For more information, see Configuring additional export options.

7. When you are finished completing the settings and no errors exist, click **OK** to save the export.

   **NOTE:** By default, new exports are saved to the root of the Exports Library. If you want to save an export to a sub-folder instead, click **Save As** to save the export instead of clicking OK. This brings up the Save As dialog, where you can select a folder location to save.

### Editing an export

To edit the settings for an existing export, right-click the export in the Exports Library and then click **Edit**.

The Export Wizard opens. You can change any export settings as desired.

**NOTE:** When editing an export, you can use the **Save As** button to save the current export as a new export file. If you have read-only rights to the export, then the **Save As** button is the only save option available.

### Deleting an export

To delete an export, right-click the export in the Exports Library and then click **Delete**.

# Configuring the export source data

Use the **Source** tab of the Export Wizard to specify the data to be exported. You can export data from any user-defined table in the system.

| Item | Description |
|---|---|
| Source table | The Axiom table that contains the data to be exported. You can select any table in your Table Library.<br><br>To specify a table, click the table icon to the right of the entry box. In the **Choose Table** dialog, select the table to use as the source table and then click **OK**. You can change the view and filter the list to find the table you are looking for. |
| Filter | Optional. A data filter to specify the rows in the table to export. If no filter is applied, then all rows will be available for export.<br><br>**NOTE:** If the user executing the export has a security filter on the source table, that filter will be honored in addition to any filter defined here.<br><br>You can type in a filter criteria statement, or you can use the Filter Wizard to create one. Once you have specified a filter, you can click the validate icon to validate the filter and return the current row count for the filter. |
| Choose columns to export | The columns of the table to export. By default, all columns are selected for export.<br><br>If desired, you can clear the check box for specific columns if you do not want to export the data in those columns. Key columns must be included in the export and cannot be changed.<br><br>The column list includes all regular columns, all calculated fields, and the system information columns for modified by / date. |

# Configuring the export destination

Use the **Destination** tab of the Export Wizard to specify the export destination and configure the necessary destination properties. You can export data to the following destinations:

- External database (SQL Server or Oracle)
- Delimited file

The destination properties differ depending on the selected destination type.

## Exporting data to an external database

Complete the following settings to export data to a target table in an external database.

| Item | Description |
|---|---|
| Remote Destination | Select either **SQL Server** or **Oracle**. |

| Item | Description |
|---|---|
| Remote Data Connection | If your Axiom Software system is hosted on the Axiom cloud service, then you must specify a remote data connection in order to connect with the external database.<br><br>You can select any remote data connection that has been set up in Scheduler. If no remote data connections have been defined in your system, then this setting does not apply and will not display. |
| Server | • For SQL Server, enter the server name for the external database.<br>• For Oracle, enter the connection parameters for the external database. You can obtain these parameters from the Oracle TNS Names entry. For more information, see the similar discussion for setting up Oracle as an import source (Import source: Oracle). |
| Database | Enter the name of the external database. This setting only applies if the database is SQL Server. |
| User | Enter the user name to use to connect to the external database. For SQL Server, the user credentials must be for a SQL Server account; network domain credentials cannot be used.<br><br>**NOTE:** The user credentials must have the appropriate permissions to perform the table actions for the export in that database. |
| Password | Enter the password to use to connect to the external database.<br><br>**NOTE:** The password must be re-entered whenever any of the other connection properties are changed. |
| Target Table | Specify the name of the table to create or update in the destination database. If desired, you can use the Choose Table button to the right of the box to look up existing table names in the destination database.<br><br>You can update an existing table or create a new table. The behavior is determined by the setting **Drop and Create Destination Table** on the **Execute** tab (see Configuring additional export options). Note that if "drop and create" is not enabled for the export, then the table must already exist in the destination database because the export will not create it. You can either manually create the table in the destination database before running the export, or you can run the export once with "drop and create" enabled to initially create the table, then disable the option for future executions to update the existing table. |

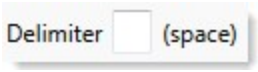
Once the connection information is completed, you can click **Test connection** to test the connection information for the external database. The **Status** updates to show either a success message or an error message.

**NOTE:** The test connection requires the user to have the highest level of permissions that could potentially be used by the export (permission to drop and create tables). If the specified user does not have this level of permissions, the test connection will fail. However, if "drop and create" is not enabled for the export, then the export may succeed even though the test connection failed.

### Exporting data to a delimited file

Complete the following settings to export data to a delimited file.

| Item | Description |
|---|---|
| Remote Destination | Select **Delimited File** to export data to a delimited text file. |
| Remote Data Connection | Select a remote data connection to use for the export. You can use any remote data connection that has been set up in Scheduler.<br><br>A remote data connection is required in order to export data to a delimited file. If your Axiom Software installation is not a cloud service system and therefore you do not have any remote data connections, then you must use the File Processing feature instead to generate export files. |
| Folder path | Enter the folder path in which save the destination file. You can type the path or click the folder icon to navigate to the desired path.<br><br>The folder path must be a UNC path (i.e. \\servername\foldername\filename). If you enter a mapped drive, it will automatically be converted to a UNC path. The folder path must be accessible to the Axiom Cloud Integration Service. |
| File name | Enter the name of the delimited file for the export to create.<br><br>By default, comma-delimited files are saved as CSV format, and all other delimiters are saved as TXT format. If you want a comma-delimited file to be saved as TXT format instead, then include the file extension on the file name. |
| First row has column names | Select this option if you want the first row of the delimited file to contain column names. |

| Item | Description |
| --- | --- |
| Delimiter | In the box to the right of the option, enter the delimiting character to use in the export file. For example, to delimit by a pipe character, enter a pipe character into the box. By default, the delimiter is set to comma.<br>  <br>If you want to use a space or tab as the delimiting character, place your cursor in the box and press the space bar or the tab key. The character will be indicated in parentheses to the right of the box (since the character is not visible in this case).<br>  |
| Text Qualifier | By default, the character used as the text qualifier is double quotation marks ("). If desired, you can enter a different character as the text qualifier, or you can clear the field if you do not want to use a text qualifier.<br>The text qualifier is used when values in the data may contain the delimiting character. For example, when the delimiting character is a comma, but the data contains values such as full names with a comma (for example: "Doe, Jane"). When the delimited file is created by the export, the value with the comma will be wrapped in quotation marks so that the comma is not read as a delimiter. |
| Compress | Select this option if you want the export file to be compressed. If enabled, the export generates a ZIP file that contains the export file. |

# Configuring additional export options

The **Execute** tab is used to execute the import, and also contains some options to control the behavior of the export. For more information on executing an export, see Executing export utilities.

The Execute tab contains the following options:

| Item | Description |
| --- | --- |
| Drop and Create Destination Table | Specifies whether the destination table is created or updated. Only applies when the export destination is a table.<br>• If enabled (default), the destination table will be created as part of the export. If the destination table already exists in the destination database, it will first be dropped (deleted), and then re-created with the new data from the export. Any existing data in the table will be lost.<br>• If disabled, the existing destination table will be updated with the new data from the export. The destination table must already exist in the destination database; the export will not create the table. Any existing data in the table may or may not be retained, depending on whether the option **Truncate Destination Table** is enabled. |
| Truncate Destination Table | Specifies whether the existing rows in the destination table are deleted before updating the table with new data from the export. Only applies when the export destination is a table, and only if the export is configured to update an existing table (**Drop and Create Destination Table** is disabled).<br>• If enabled, all existing rows in the destination table will be deleted before the table is updated with new data.<br>• If disabled (default), then all existing rows in the destination table are preserved, and new data rows are appended to the table.<br>**NOTE:** New data rows are added "as is", without validating for duplicate keys or other constraints. This option should only be used if you are confident that new data will not conflict with existing data. |
| Description | Optional. A description for the export. The description can be used to document the purpose of the export and/or to detail important export instructions. The description is limited to 2000 characters. |

# Executing export utilities

When you execute an export utility, data is queried from the source table in Axiom Software according to the export configuration settings, and then the resulting data is saved to the destination table or the destination file. If the export destination is a table, the table is either created or updated, depending on the export configuration.

**NOTES:**

- By default, only administrators can execute export utilities. Non-admin users can be granted permission to execute specific imports, as defined on the **Files** tab of Security.
- If the user executing the export has a security filter for the source table, that filter will be honored to determine the available data to export.
- You can also use Scheduler to execute export utilities, using the **Export ETL Package** task.

**To execute an import:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**. You can also use the Explorer task pane.
2. In the **Export Library**, double-click the export that you want to execute.
    - If your security permissions only allow you to execute the export, then the export begins running immediately with no further interaction.
    - If your security permissions allow you to view or edit the export settings, then the **Export Wizard** opens.
3. If you have permission to edit the export, you can review the options on the Export tab and modify them as needed. For more information on these options, see Configuring additional export options.
4. On the **Export** tab of the Export Wizard, click the **Execute** button to start the export.

As the export is processed, status messages display in the **Execution log** at the bottom of the dialog. If an error occurs, the error message displays in the log and the export is stopped. If necessary, you can copy and paste the text in the execution log; for example, to send the error to Axiom Software support. Right-click inside the log and choose **Select All** and then **Copy**.

If you want to stop the export while it is executing, click **Stop**. When the export is complete, click **OK** to close the dialog.


# Customizing the User Interface

You can customize the Axiom Software user interface by creating custom task panes and custom ribbon tabs, and then assigning them to users and roles. Many of the built-in task panes and ribbon tabs can also be customized, including the main Axiom ribbon tab.

Generally speaking, the creation of custom task panes and ribbon tabs is an administrator function. The intent is for administrators or other power users to create the custom task panes and ribbon tabs as needed for their end users, and then assign these items as startup files in Axiom Software Security.

## About custom task panes and ribbon tabs

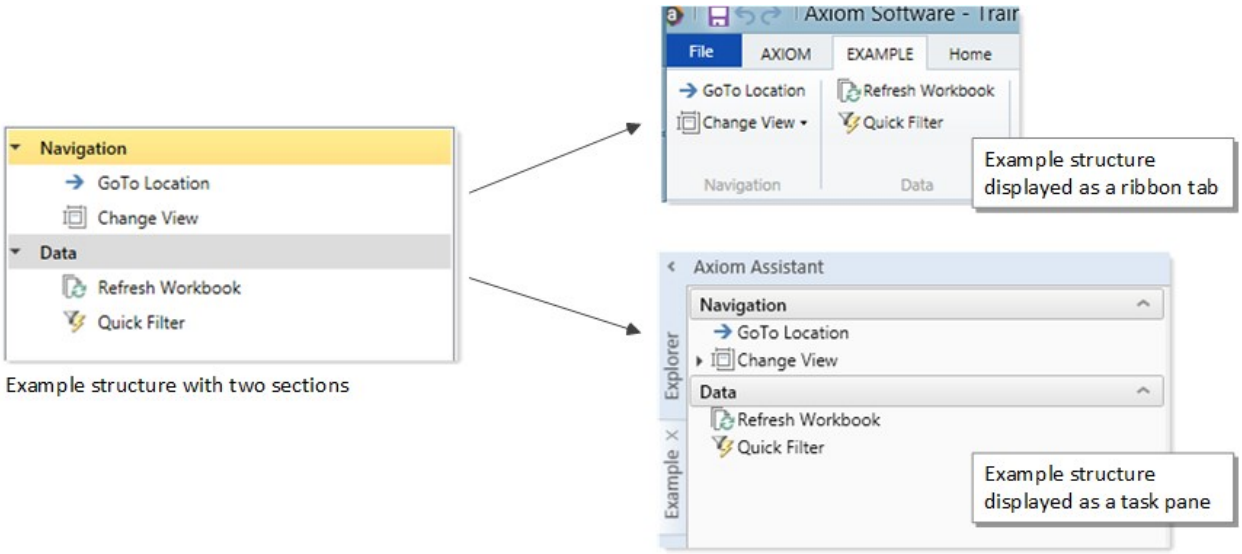
Using custom task panes and ribbon tabs, you can provide your users with a user interface designed especially for them.

### Defining custom task panes and ribbon tabs

Task panes and ribbon tabs are stored as AXL files in the Task Panes Library and in the Ribbon Tabs Library respectively. Task panes and ribbon tabs are created in a very similar manner. Both use a multi-level structure, with top-level "sections" that can contain one or more child items.

- In the task pane environment, the top-level section is displayed as a gray header bar which can be collapsed and expanded to access the child items in the section.
- In the ribbon environment, the top-level section is displayed as a ribbon group, and the child items in that section display as buttons in the group.

The following example shows how the same basic structure is rendered as a ribbon tab and as a task pane.




Task panes and ribbon tabs can contain the following items:

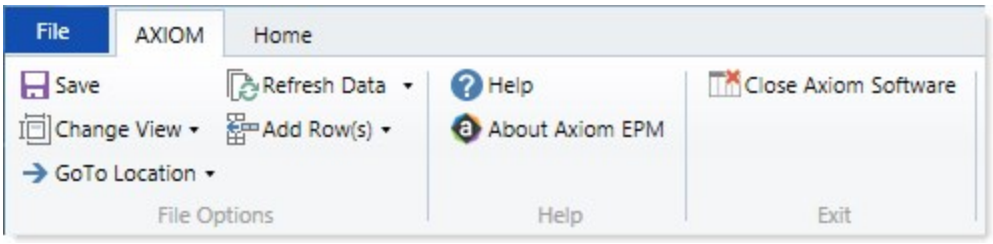
- Links to file groups, files, and folders in Axiom Software, so that users can open these items from the task pane or ribbon.
- Various Axiom Software feature commands from the Command Library, so that users can launch the feature from the task pane or ribbon.
- Plain text items for header and group titles, and for instructional text (such as in a task pane that provides instruction for a process).

Security permissions apply to all items in task panes and ribbon tabs. If a file or feature is placed in a task pane or ribbon tab, but a particular user does not have the appropriate security permissions for that item, then that user will not be able to use that item. The item will either be hidden or it will display as grayed out, depending on whether the **Show restricted item** option is enabled for that item in the task pane or ribbon tab.

## Common uses of custom task panes and ribbon tabs

The most common use of custom task panes and ribbon tabs is to provide users with a custom user interface. For example, you may want to streamline the user interface so that end users only see the specific Axiom features that they need to use. You can do this by customizing the "built-in" Axiom ribbon tab, and/or by creating and assigning additional custom ribbon tabs and task panes to users and roles.

For example, some customers might want to create a very simple version of the Axiom ribbon tab for their end users, something like the following:



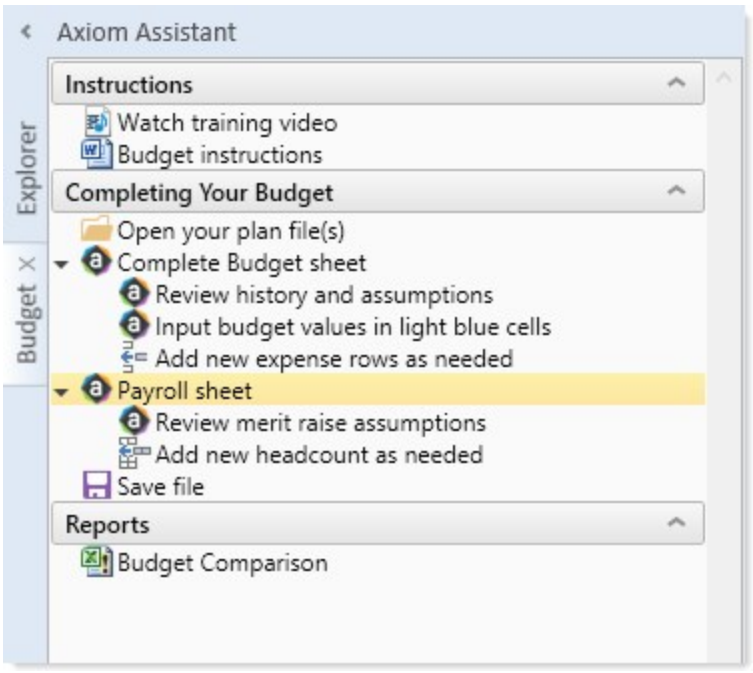

End user roles could be assigned this simple ribbon tab, while administrators and power users continue to use the default Axiom ribbon tab. Notice in this example that no file groups are listed on the ribbon and there's no Reports button—this sort of design would depend on users accessing their plan files and reports via links on the Home page, or by using the Explorer task pane or a custom task pane. This example is intentionally very simple to illustrate the full range of possibilities between the default ribbon tab and a minimal one.

Other typical uses include:

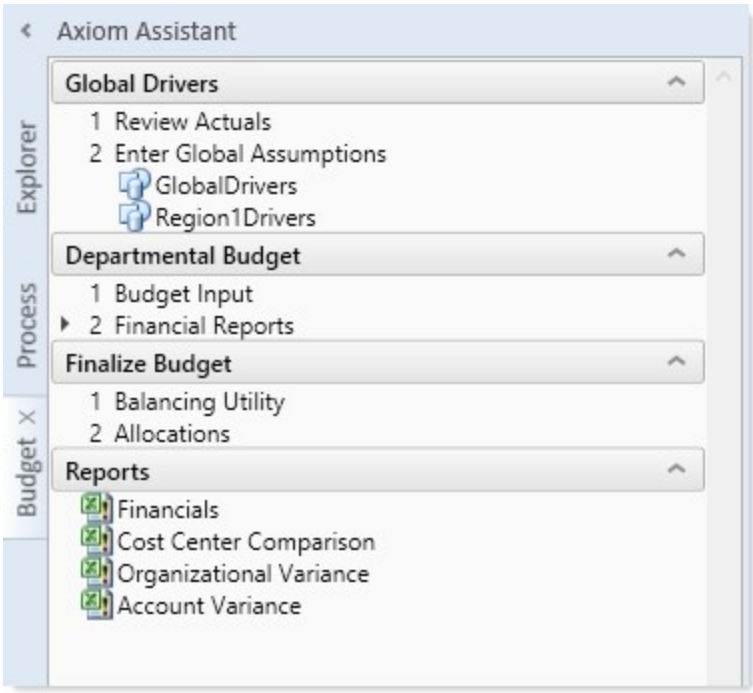
- Providing easy access to certain files, such as a list of frequently-used reports. This list could be presented in a reporting task pane so that users do not need to navigate the Reports Library to access these reports.
- Providing user instruction for various processes. For example, you could provide a task pane that walks the user through the steps needed to open their plan file, edit their plan file, and then submit their plan file for review.

  **NOTE:** Custom task panes do not provide full process management; they are intended more as a casual guide for users. If you want to define a process with enforced task order and ownership, including an audit trail for when a task was completed and by whom, then you should use the Process Management feature.

The following is an example task pane that could be made for end users to complete their budgets. Although it would be possible to do this as a ribbon tab, use of text instruction for a process is better suited for a task pane.




The next example task pane would be intended for administrators and other finance "power users." It collects all of major files and utilities that these users need to manage the budget process.



### Assigning custom task panes and ribbon tabs to users and roles

The primary method of using custom ribbon tabs and task panes is to assign them as startup files in Axiom security to certain users and roles. When a custom ribbon tab or task pane is assigned as a startup file, then it automatically opens when the user starts Axiom Software.

Axiom Software provides certain "built-in" custom task panes and ribbon tabs that are automatically assigned to the Everyone role by default, such as the Axiom ribbon tab or the Explorer task pane. You can use these built-in items as is, or you can customize them as desired and change their security configuration. For more information, see Built-in task panes and ribbon tabs.

For task panes only, you can also choose to give users and roles direct access to certain task pane files in the Task Panes Library (by granting permission on the Files tab of Axiom security). These users can then open these task panes "on demand" as needed. This is a good option for task panes that only need to be used occasionally rather than all the time. Ribbon tabs cannot be used in this way because the contents of the ribbon are determined at startup. It is not possible to open a ribbon tab in the ribbon "on demand".

For more information, see Assigning ribbon tabs to users and Assigning task panes to users.

# Task panes

You can use custom task panes to provide users with a custom user interface in the Excel Client or the Windows Client. For an overview of how the Axiom user interface can be customized, see About custom task panes and ribbon tabs.

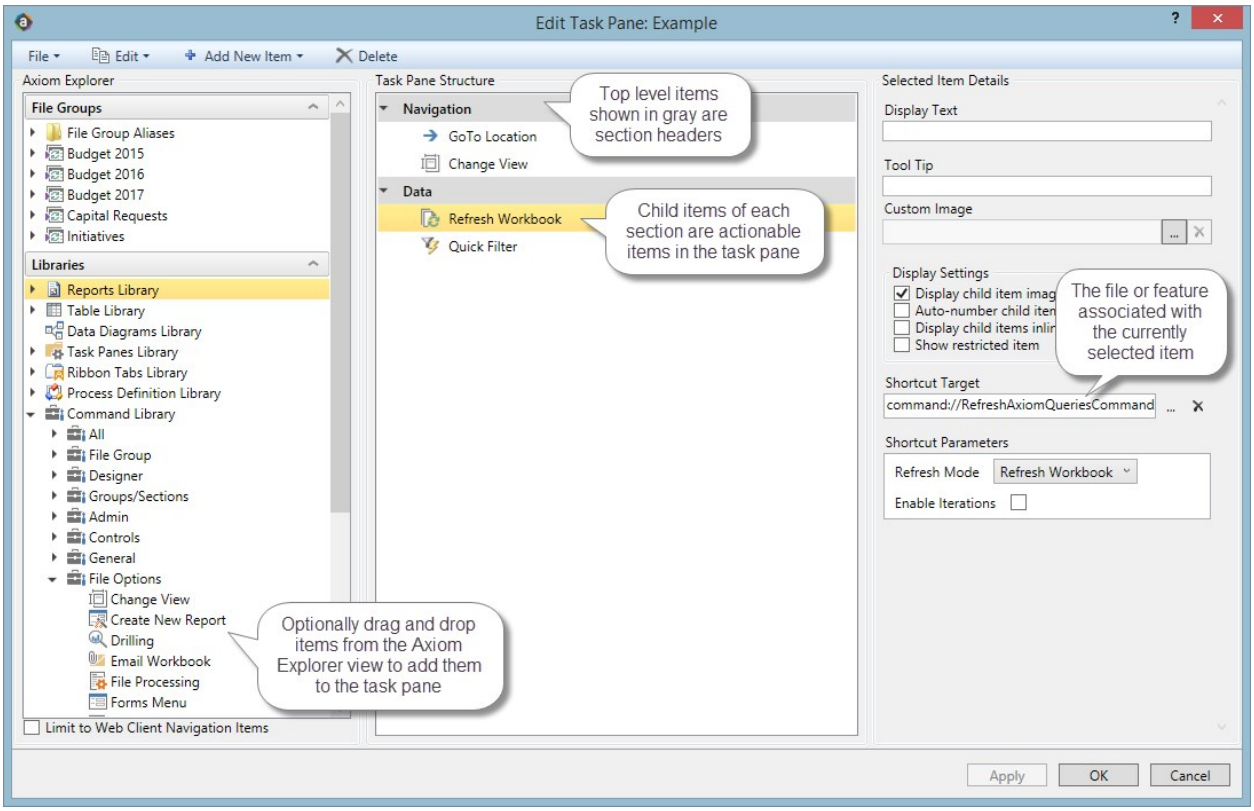
This section explains how to create, edit, and delete custom task panes, and then assign them to end users. Task panes are managed using the Task Panes Library within Axiom Explorer. Task pane files display using the task pane icon  .

In addition to creating your own custom task panes, you can customize the built-in task panes provided by Axiom Software, such as the Explorer task pane, the Workflow task pane, and the Process task pane. For more information, see Built-in task panes and ribbon tabs.

**NOTE:** It is also possible to use an Axiom form as a task pane. In this case, the Axiom form displays within the task pane area and is considered a "dialog" for behavior purposes (such as for storing and sharing "form state" values). For more information, see the *Axiom Forms and Dashboards Guide*.

## Task pane editor

You can create and edit custom task panes using the **Edit Task Pane** dialog. This dialog has three sections as shown in the following example screenshot.




*Task pane editor*

The middle section, labeled **Task Pane Structure**, is where you define the sections and individual items to display in the task pane. To add new items to this structure, you can either:

- Use the **Add New Item** button in the toolbar to add new items. You can choose to add a new child item to the currently selected item, or add a new item at the same level, or add a new section-level item.
- Use the right-click menu to add new items. This has the same options as the Add New Item button.
- Drag and drop an item from the **Axiom Explorer** pane (the left-hand section of the dialog) to the desired location in the task pane structure. This will automatically link the item in the task pane structure to the appropriate feature or file (based on what you dragged and dropped).

When you select an item in the task pane structure, the properties for that item display in the right-hand section of the dialog, labeled **Selected Item Details**. Here you can define display text and other display properties for each item, and assign the associated feature or file for the item using the Shortcut Target. (When you drag and drop and item from the Axiom Explorer pane to the task pane structure, the shortcut target is automatically assigned.) For more information on the available properties, see Task pane properties.

You can move items in the task pane structure by dragging and dropping them to various locations, or by using **Copy** or **Cut** and then **Paste**. To remove an item, select it and then click Delete. Any child items of the item you deleted will also be deleted.

The task pane editor is not a full WYSIWYG editor. After creating a task pane, you should make sure to open and review the task pane in the Axiom Assistant area to ensure that it displays as you intended.

**TIP:** If you need to edit an existing task pane, you can first open the task pane in the Axiom Assistant area and then open the task pane file for editing. As you make changes in the editor, you can click **Apply** to see the change reflected in the currently open task pane.

## Managing task panes

Task panes can be created, edited, or deleted using **Manage > Task Panes**, in the **Administration** group of the **Axiom** tab. You can also manage task panes from the Task Panes Library in the Explorer task pane.

### Creating a task pane

You can create task panes for various purposes and display them to users within the Axiom Assistant area. Once a task pane is created, it can be assigned to users and roles to open automatically when the user logs into the system, or to be available to open on demand.

Task panes consist of one or more section headers and child items within the section. The section headers display as gray bars that can be used to expand or collapse the section. Child items can be used to display instructional text or to link to files and features.

**NOTE:** Only administrators and users with the **Administer Task Panes** security permission can create new task panes.

**To create a new task pane:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Task Panes**.

   The **Axiom Explorer** dialog opens, with the focus on the Task Panes Library. You can also access this library using the Explorer task pane.

   **TIP:** If you already have a custom task pane open in the Axiom Assistant area, you can right-click the tab of the task pane and select **Create New Task Pane**.

2. Right-click the **Task Panes Library**, and then select **New > Task Pane**.

   The Edit Task Pane dialog opens.

3. New task pane files start with a single section header, named Section1. Edit the name of this section using the **Section name** box in the right-hand side of the dialog.

You can configure other appearance details for this section, such as to define an optional tool tip, or to enable auto-numbering or image display for this section's child items. For more details on the available options, see Task pane properties.

4. Add child items to the section as needed. You can add child items by doing any of the following:

   - With the section name selected, click **Add New Item > Child of selected item**.
   - With a child item selected, click **Add New Item** and then either **Before selected item** or **After selected item**.
   - Drag and drop an item from the Axiom Explorer pane in the left-hand side of the screen to the desired place in the task pane structure. A black line shows you where the item will be placed.

   You can configure each item as desired. An item can be just display text for instructional purposes, or you can link the item to an Axiom file or feature. If you dragged and dropped an item from the Axiom Explorer pane into the task pane structure, then the item is automatically linked to that file or feature. Otherwise, use the **Shortcut Target** setting to assign the item to a file or feature.

   For more information on the types of files and features that can be included on task panes and how they work, see Linking to Axiom files and features in a task pane or ribbon tab. For more details on available appearance properties, see Task pane properties.

   **NOTE:** If an item has child items, then it cannot also have a shortcut target. When users interact with the task pane, double-clicking an item with children expands and collapses the item, so there is no way for the user to launch the shortcut target in this case.

   There are a variety of ways to configure task panes. For more details on some common design goals and avoiding potential issues, see Task pane design considerations.

5. Add more sections and child items as needed. You can add new sections by using **Add New Item > New top level section**. Arrange these items as desired by dragging or dropping items within the structure, or by using the **Edit** button to cut, copy, and paste.

6. Click **OK** to save.

   The **Save As** dialog opens to the Task Panes Library. Give the file a name and optionally a description, and save it to the desired location.

   **NOTE:** The name of the task pane file determines the default name of the task pane tab within the Axiom Assistant area. For example if the file is named Reports, then when it is opened as a task pane it will have a tab name of Reports. Make sure the file name is brief and intuitive.

After you have created a new task pane, you should open it to make sure all items display and behave as expected. If the task pane will be assigned to end users, it is a good idea to log in as a representative end user and test it.

## Editing a task pane

You can edit a task pane at any time, regardless of whether it is currently open in the Axiom Assistant area. Task panes within the Axiom Assistant area are always open read-only and do not impact editing availability. (If a message tells you that a task pane file is locked, this means that another user has opened the task pane in the editor, not that the task pane is open in the Axiom Assistant area.)

If you are editing a task pane, the changes will be visible in your session right away, but other users will not see the changes until the next time the task pane is opened. This means that if a task pane is configured to open on startup, users will not see the changes until the next time they launch the system.

**To edit a task pane:**

1. Do one of the following to open the task pane for editing:
   - If the task pane is already open in the Axiom Assistant area, right-click the tab and then select **Edit Task Pane**.

     > **TIP:** One advantage of having the task pane open while editing is that you can click **Apply** to immediately see the results of your edits, without needing to close the editing dialog.

   - From the **Axiom** tab, in the **Administration** group, click **Manage > Task Panes**. In the **Task Panes Library**, navigate to the task pane that you want to edit, then right-click it and select **Edit**. You can also perform this action from the Explorer task pane.
2. In the **Edit Task Panes** dialog, edit the task pane as desired. For more information on the available task pane properties, see Task pane properties.
3. Click **Apply** or **OK** to save the task pane. If you would rather save this task pane as a new file, you can click **File > Save As** from the dialog toolbar.

## Deleting a task pane

You can delete a task pane at any time. However, if the task pane is currently open in the Axiom Assistant area within *your* session, you must first close it before the file can be deleted. It does not matter if the task pane is opened in other users' sessions—this does not prevent deletion.

If the task pane is a startup task pane that is configured to be non-closeable, then you must first remove or modify the startup configuration so that you will be able to close it. After you have modified the applicable startup configuration in Security, exit the system and then log in again to apply the new startup configuration.

If any other users currently have the task pane open when you delete the file, the task pane will remain useable in that session until the user closes the task pane or exits the system. If the deleted task pane was configured to open on startup, no special steps are required to remove it from user configurations—the next time users log in, the task pane will simply no longer exist and therefore will be ignored by the startup configuration. However, it is a good idea to remove the obsolete configuration from Security so that user configurations match what you expect to be displayed.

**To delete a task pane:**

1. From the **Axiom** tab, in the **Administration** group, click **Manage > Task Panes**. You can also delete task panes using the Explorer task pane.
2. In the **Task Panes Library**, navigate to the task pane that you want to delete, then right-click and select **Delete**.

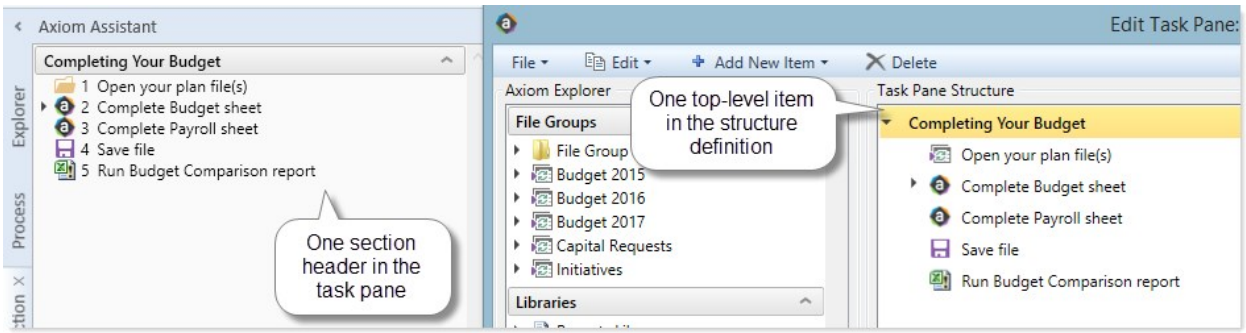
## Task pane design considerations

Keep in mind the following design considerations when creating task panes.

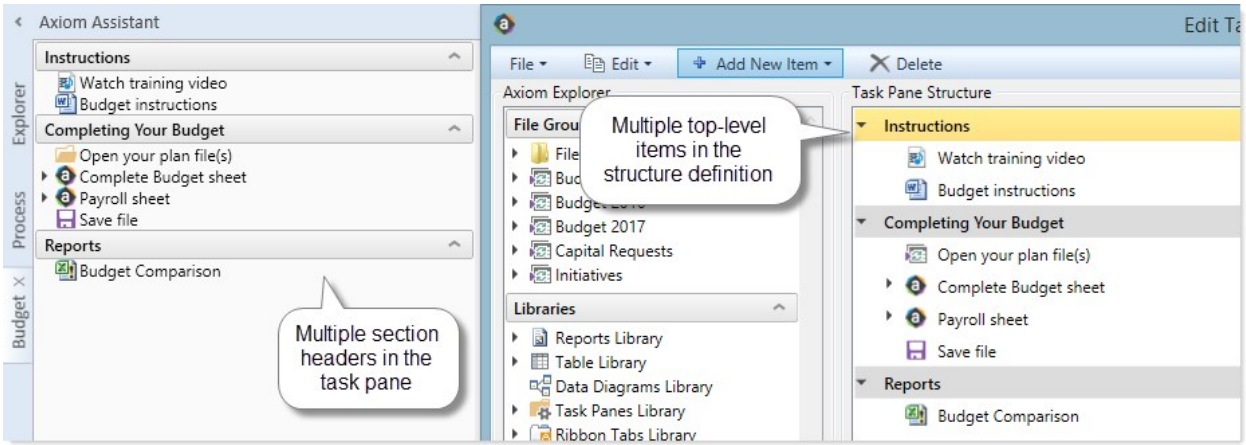
### Creating task panes with multiple sections

All top-level items in a task pane are automatically section headers. These items display using a gray header bar, and can be expanded and collapsed (by default, they display expanded).

If the task pane only needs one section, then the task pane should be created so that all items are children of a single top-level item. The following screenshot shows an open task pane with one header side-by-side with the associated view in the editor.




If the task pane needs multiple sections, then you can add multiple top-level items, and then add child items to each section as needed. The following screenshot shows an open task pane with multiple headers side-by-side with the associated view in the editor.

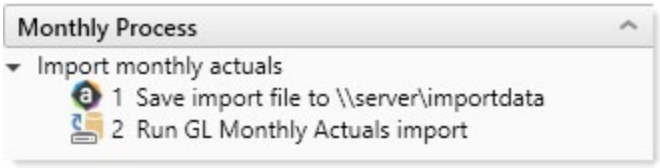


### Shortcuts on items with children

Shortcuts cannot be defined on items in a task pane that have child items. When an item has child items, double-clicking the parent item expands or collapses the item—therefore, there is no way to launch the associated shortcut for that item.

The task pane editor does not prevent saving a task pane with this configuration, however, it does display a validation message informing you that an item cannot have both a shortcut and child items.

If you want to link to a file or feature and include associated instructional text, the shortcut should be defined in the child items, not on the parent item. For example:




In this example, the parent item serves as a statement of what to do, followed by instructional text and then the link to the import.

### Forcing display of an item

By default, if an item in a task pane links to a file or feature, and the user viewing the task pane does not have access to that file or feature, then the item does not display in the task pane for that user. This allows you to create task panes that are dynamic by user—you can include a full set of items in the task pane and users will only see the items that apply to them.

Items in a task pane are also hidden if they do not apply to the current context. For example, an item that links to the **Add Rows** command is hidden by default if the current file is a report, because that command cannot be used in reports.

However in some cases you may want all items in a task pane to display, regardless of whether a user has rights to the items, and regardless of whether the item applies to the current context. For example, you may be creating a task pane that documents a particular process, and all of the process steps are numbered. If a user does not have rights to the file linked in step 5, then step 5 will be omitted from the task pane. Although the auto-numbering will be adjusted for the omitted item, you may prefer to include the item in the task pane anyway for one or more of the following reasons:

- The omitted step may be required, and the user needs to know that it is there and that they cannot perform it as is. Perhaps the user needs to wait for someone else to perform that step, or maybe the security settings have been made in error and the user should have access to that file.
- The omitted step may be optional, but you still want the user to see its place in the overall process. Users with the appropriate rights can perform the step, and users without rights can simply move on to the next step.

To force an item to display in a task pane regardless of security settings or current context, enable **Show restricted item** in the Display Settings for the item. This must be selected on a per item basis.

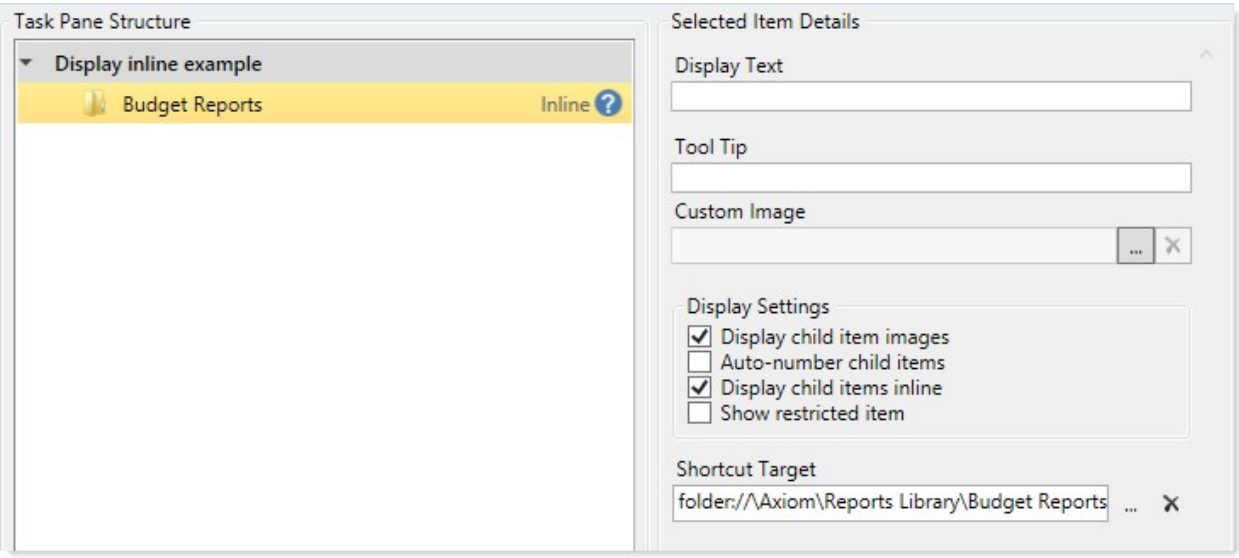
If this option is selected and a user does not have access to the linked item, or the item is not applicable to the current context, then the item is grayed out in the task pane.

### Displaying child items inline

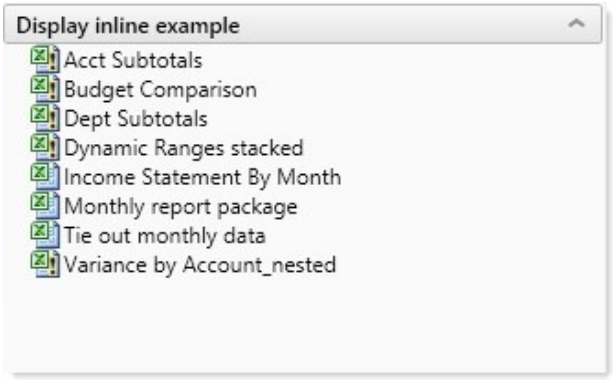
If an item has child items, you can choose to display those items inline (at the level of the parent item), omitting the display of the parent item. To do this, you must enable **Display child items inline** for the parent item.

This behavior applies even when the child items are not defined in the task pane structure, but are dynamically generated as part of the parent item. For example, if you link a task pane item to a report folder, the folder is the parent and the files in the folder are the child items, even though the child items aren't technically defined in the task pane structure.

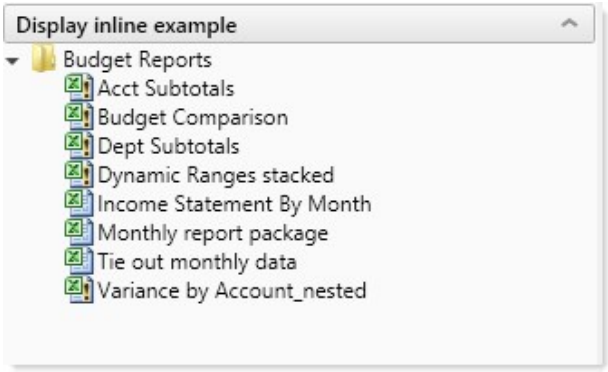
In this example task pane structure, the task pane item links to a report folder, and the item is configured to display child items inline:




The following screenshots show the difference in how this item displays, depending on whether **Display child items inline** is enabled.




*Display child items inline: Enabled*

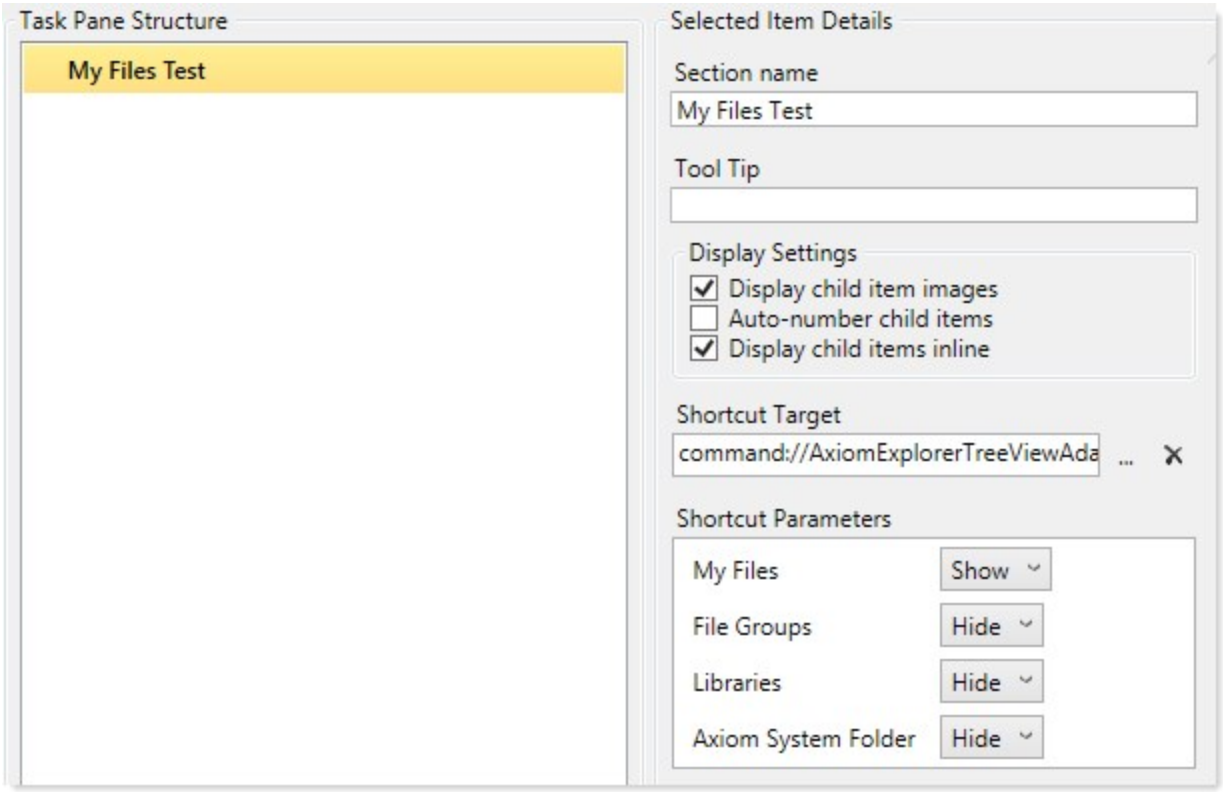



*Display child items inline: Disabled*

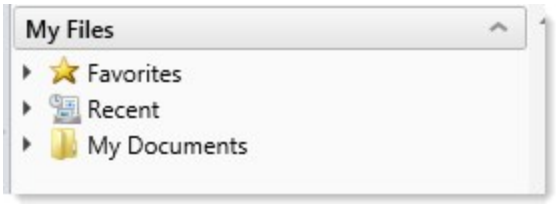
In the first example, the parent item does not display, and instead the child items display at the same level of the parent item. In the second example, the child items display as normal under the parent item.

This setting is also useful when including pre-built feature controls in a custom task pane, such as the Axiom Explorer Tree View. If you link a task pane item to the control, then by default the control will display as a child item underneath the parent item. If instead you want the item to display directly within the task pane, omitting the parent item, then select **Display child items inline**.

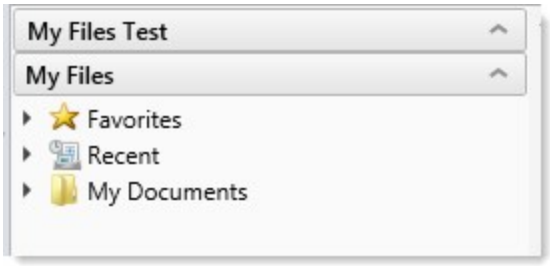
For example, the following task pane item is linked to the Axiom Explorer Tree View control, configured to show only the My Files section and configured to display child items inline.



The following screenshots show the difference in how this item displays, depending on whether **Display child items inline** is enabled.




*Display child items inline: Enabled*




*Display child items inline: Disabled*

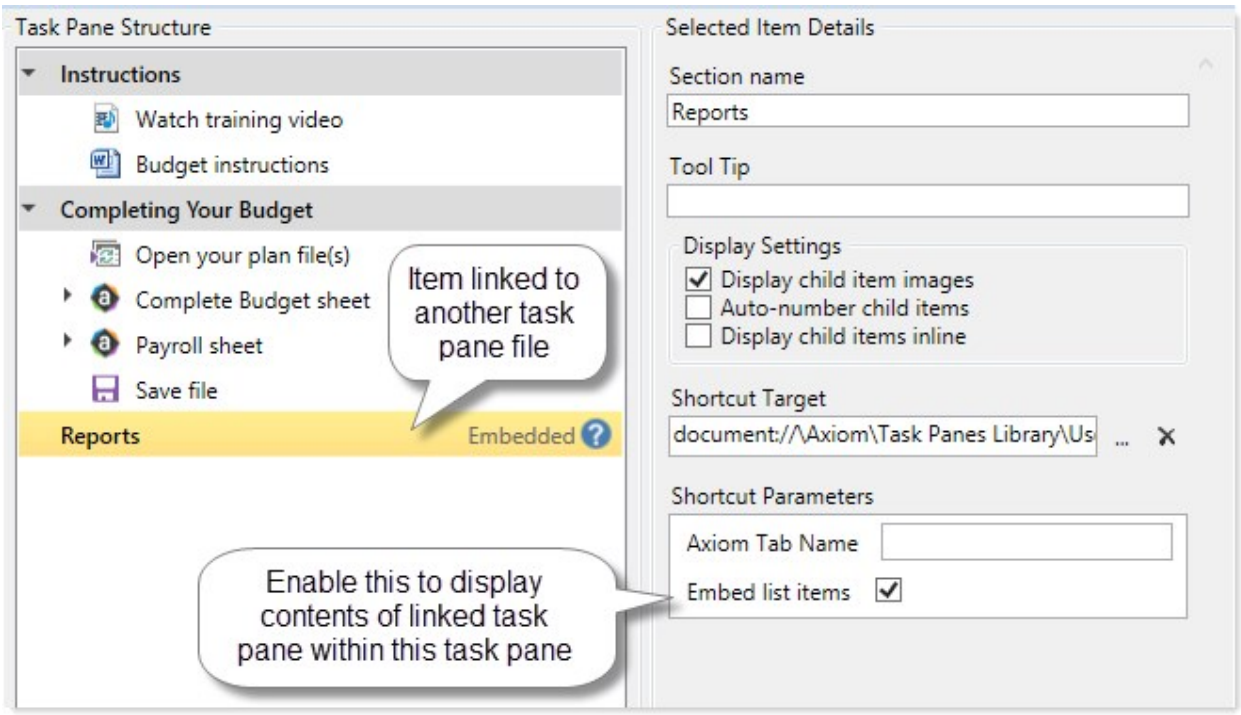
In the first example, the My Files control is displayed directly within the task pane, omitting the parent item. This is most often the preferred way to display a pre-built feature control. In the second example, the My Files control is displayed underneath the parent item, which results in a duplicate header.

### Embedding a task pane within another task pane

It is possible to create "nested" task panes, where the content from one task pane displays within another task pane. This is accomplished by using the shortcut property **Embed list items** when linking to the task pane, and then optionally also enabling **Display child items inline** for the linked item.

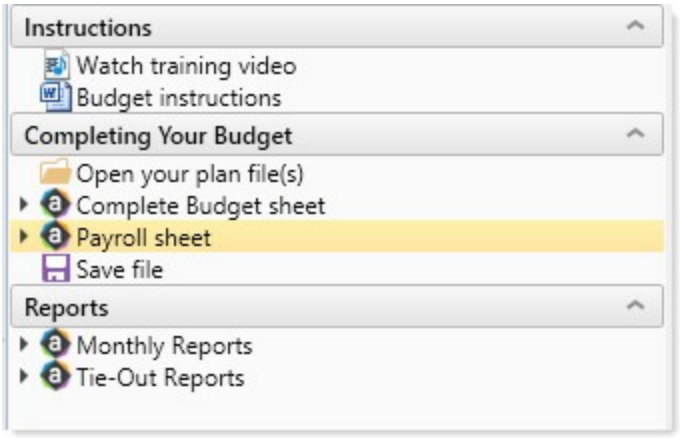
**NOTE:** If **Embed list items** is enabled, then the contents of the embedded task pane are treated as if they are part of the current task pane. Axiom Software does not make a separate security check to see if the user has permission to access the embedded file—the user's permission to the current file is all that matters. However, security permissions will be applied to the *contents* of the embedded task pane as normal.

For example, imagine that you have a task pane, and you want to create a second task pane that has the same content as the first task pane plus some additional items. If you link to the first task pane from the second task pane, by default the first task pane will display as a file within the second task pane, and users can double-click the file to open the first task pane. But if you instead enable **Embed list items**, then the contents of the first task pane will display embedded within the second task pane.

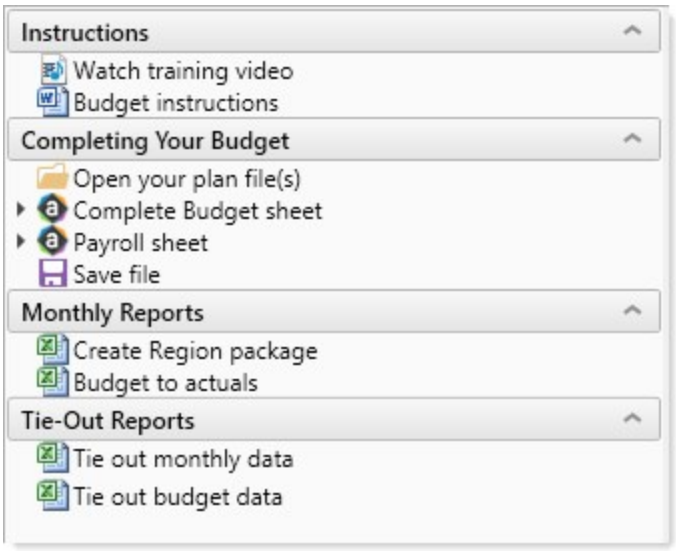



*Definition example for nested task panes*

At this point the items display as a collapsible node, with the name of the first task pane as the "parent" item and the contents of the first task pane as child items.




If instead you want the contents of the first task pane to display inline within the second task pane (as if they were defined directly within the second task pane), then you must also enable **Display child items inline** for the linked item in the second task pane.




**NOTE:** If you embed a task pane in another task pane, and you plan to assign the "parent" task pane to users as a startup file, keep in mind that users must have explicit permission to the embedded task pane in order for it to display in the startup task pane. The implicit permission that results from startup file assignment only applies to the parent task pane.

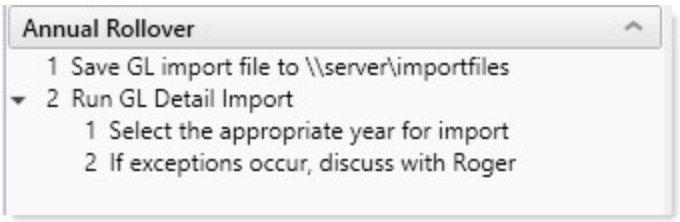
## Task pane properties

The following properties are available for items in a task pane. If you are editing one of the built-in task panes (such as Explorer), see the following for more information on that task pane: Built-in task panes and ribbon tabs.

| Item | Description |
|---|---|
| Display Text / Section Name | The text to display for the item. If the item has a defined shortcut target to an Axiom file or feature, then you can leave this box blank, and the display text will be the name of the file or feature.<br><br>If the item is a top-level item, then this property is called Section Name, to indicate that the item will be displayed as a section header in the task pane. Section headers are gray title bars that can be expanded and collapsed. |
| Tool Tip | The text to display as a tool tip when a user hovers their cursor over the item. This text can be helpful to explain the purpose of the item within the context of the list.<br><br>If no tool tip text is defined, then the tooltip displays the full path of the shortcut target (if defined). |

| Item | Description |
| --- | --- |
| Custom Image | The image to display for the item on the task pane. Click the **Browse** button **[...]** to select an image that is stored in the Reports Library. If the image is not already saved in the Reports Library, you can right-click a folder and select **Import** to import the image (if you have the appropriate rights to do so).<br><br>The image must be at least 32x32 pixels, and the file format must be JPG or PNG. Larger images will be resized to fit. The image should be square because the aspect ratio will be retained when resizing.<br><br>If no custom image is defined, then the default Axiom image for the shortcut target is used. If the item has no shortcut target, the Axiom "dots" icon is used. In either case (custom image or default image), an image will only display next to an item if the parent item has **Display child images** enabled.<br><br>**NOTES:**<br>• Users must have permission to the image file in order to see it rendered in the form. It is recommended to create a dedicated Images folder in the Reports Library and store all images in this location. You can grant access to this folder using the Everyone role, or you can create subfolders and grant access to users and roles as needed.<br>• The next time you open this file after saving, the path to the image will be automatically converted into a system-managed document shortcut (you can tell the difference by the presence of a `_tid` parameter on the end of the shortcut). This is to make the file reference "repairable" in cases where the file is renamed or moved. Note that if the path is a result of a formula instead of directly within the cell, then the conversion will not occur and the file reference will not be repairable.<br><br>This option is not available for top-level section items. |

### Display Settings

| Item | Description |
|---|---|
| Display child item images | Select this option if you want icons to display on the child items for the current item. This option is selected at the parent item level because all child items must have the same icon settings (enabled or disabled), so that the child items will align appropriately in the task pane.<br><br>If enabled, then each child item will display using the associated icon for its shortcut target. For example, if the target is a report file, then the relevant report file icon will display next to the item. If an item does not have a shortcut target, or if the target does not have a specific icon, then the default Axiom Software "dots" icon is used.<br><br>If disabled, then no icons will display on child items. |
| Auto-number child items | Select this option if you want to apply auto-numbering to child items for the current item. If selected, all child items will be auto-numbered by whole numbers (1, 2, 3...).<br><br>**NOTE:** Multi-level auto-numbering is not supported. If auto-numbering is applied at two subsequent levels, the numbering will appear as follows:<br><br>  |
| Display child items inline | Select this option if you want all of this item's child items to display inline, omitting this item.<br><br>For example, this might be used in the following circumstances:<br><br>• When using certain items from the Command Library as targets, to hide the "command line" and only show the contents of the command.<br>• When linking to a folder, to hide the "folder" node and instead display the resulting files inline.<br>• When linking to a task pane, to hide the file name and instead show the contents of the linked task pane directly within this task pane. The shortcut parameters for the linked task pane must also be configured to enable **Embed list items**.<br><br>**NOTE:** If this option is selected for an item that does not have any child items (either literal child items in the structure or child items inherent to the shortcut target), then the item will be hidden in the task pane. |

| Item | Description |
| --- | --- |
| Show restricted item | Select this option if you want this item to always display in the task pane, regardless of the user's security permissions and regardless of the current context.<br><br>By default if a task pane item is linked to a file or feature that a user does not have access to, then that item is hidden from the user. There are times when that may not be the desired behavior. For example, if the task pane details a set of process steps, you may want the user to be able to see the entire process from start to finish, even if they do not have rights to one or more particular items referenced in the process. For more information, see Task pane design considerations.<br><br>If this option is selected, and the user does not have rights to the linked file or feature (or the item is not applicable to the current context), then the item will display as grayed out and the file or feature cannot be launched.<br><br>This option only applies to child items; it does not apply to top-level sections. By default, if a user has no access to any items in a section, the entire section will not display. |

**TIP:** As you change appearance options, you can click **Apply** to view the result of the change within the Axiom Assistant area, without needing to close the **Edit Task Pane** dialog. This only works if the task pane was opened before editing.

### Shortcut settings

| Item | Description |
|---|---|
| Shortcut Target | The target file or feature for the item. When a user double-clicks on the item in the task pane, the file or feature will open.<br><br>Click the **Browse** button [...] to select the desired file or feature. You can select any item that you can navigate to within Axiom Explorer. For more information on what types of files and features can be linked to task pane items and their behavior within the task pane, see Linking to Axiom files and features in a task pane or ribbon tab.<br><br>Once a target is selected, Axiom Software places the necessary syntax to open the target within the **Shortcut Target** box. It is best not to manually edit this text, as you may inadvertently render the syntax invalid. If you want to select a different target, use the Browse button to select the target rather than editing the target syntax.<br><br>If you no longer want the item to link to a shortcut target, click the **Delete** button ✕ to remove the shortcut.<br><br>**NOTE:** If an item has child items, then that item cannot have a shortcut target. When users work with the task pane, double-clicking an item with children expands or collapses the item, so there is no way for the user to launch the shortcut target. |
| Shortcut Parameters | Most shortcut targets have one or more associated shortcut parameters that impact the behavior of the linked file or feature when it is opened from the task pane. The available parameters depend on the shortcut target—for example, report files have different parameters than tables.<br><br>For more information on available shortcut parameters per shortcut target type, see:<br><br>• Shortcuts<br><br>You can link to various files and other items in Axiom Software using shortcuts. For more information, see Axiom Software Help: **Reference > Shortcuts** (or search on code **AX1169** to bring up the Shortcuts section).<br><br>• Command Library<br><br>You can link to various Axiom Software features using commands in the Command Library. For more information, see Axiom Software Help: **Reference > Command Library** (or search on code **AX2685** to bring up the Command Library section). |

### Task pane editor settings

The option **Limit to Web Client Navigation Items** is located underneath the Axiom Explorer section of the dialog. This option should only be used when you are creating a task pane that defines navigation links for the Web Client container. If enabled, it filters the dialog so that it only displays options that are relevant to Web Client navigation links.

## Assigning task panes to users

You can assign task panes to users and roles using the Startup tab in Axiom Software Security. When a user starts Axiom Software, their assigned startup task panes will automatically open.

Alternatively, you can also grant permission to access one or more task panes in the Task Pane Library directly, at the user or role level. This allows users to open task panes on demand.

### Assigning as startup files

To configure a task pane to open on system startup for a particular user or role, use the **Startup** tab of Security. Any task panes listed on the Startup tab will open automatically when the user starts the Excel Client or the Windows Client. Task panes do not apply to the Web Client. For more information on assigning startup task panes, see the *Security Guide*.

When task panes are assigned as startup files, users do *not* otherwise need to be granted access to these files. The files will open automatically at startup.

When assigning the startup file, you may want to use the Shortcut Parameters to designate the task pane as non-closeable. This ensures that the task pane will always be available to the user. This option is provided so that users do not accidentally close the task pane and then have no way to reopen it (because they do not otherwise have permissions to the file).

In most cases, you will configure startup task panes on a per role basis. For example, you might have a Budget User role, and you want all users in this role to see a simple task pane tab that presents the handful of items that they need to access. Alternatively, if all users in a system need to see a particular task pane on startup, you can configure it for the Everyone role.

Startup task panes display in the Axiom Assistant area when opened, in the following order:

1. Task panes assigned to the Everyone role.
2. Task panes assigned to other roles that the user belongs to (multiple roles are sorted in alphabetical order).
3. Task panes assigned to the user.

Keep in mind that the visibility of a startup task pane may depend on whether the task pane has any content to display. For example, the built-in Process task pane is assigned to the Everyone role by default and therefore it opens at startup for all users, but if a user does not have any process tasks then the task pane does not display for that user.

### Access to the Task Panes Library

If desired, you can grant users read-only access to files in the Task Panes Library, using the **Files** tab of Security. This allows users to access the library from the Explorer task pane, and then open task panes on demand. You can also link to task panes from within a task pane or a ribbon tab, and in this case the user must have read-only access to the task pane in order to open it using the link.

You might do this if you have some task panes that users only need to access occasionally, when performing certain tasks. In this case it may not make sense to open these task panes at startup. When users want to open one of these task panes, they can double-click on it in the Task Pane Library, and it will open as a task pane in the Axiom Assistant area. They can then close the task pane when they are finished with it, by clicking on the **X** button on the task pane tab.

In most cases, end users do not need to edit task panes and therefore do not need read/write access to these files. However, if you have non-administrator users who need to create and edit task panes, those users must be granted the **Administer Task Panes** permission as well as read/write access at a file or folder level in the Task Panes Library.

## Associating a task pane with a particular file

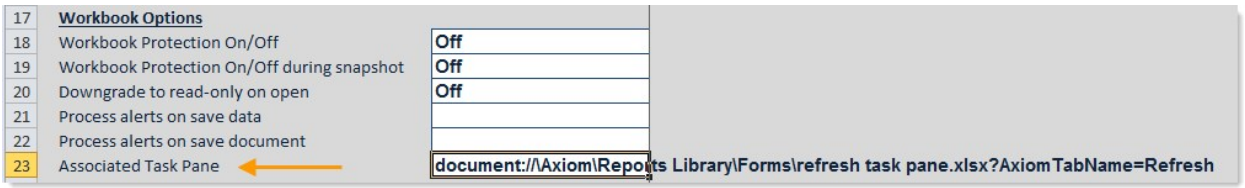
You can associate a task pane with a particular file, so that the task pane automatically opens when that file is opened, and remains available for use while the file remains open.

For example, you might want to allow users to change refresh options for a report within a task pane instead of using a refresh dialog. Or you might design a task pane to assist users in completing their plan files, and you want that task pane to be automatically available when users are working in a plan file.

The associated task pane can be a custom task pane in the Task Panes Library, or it can be an Axiom form that has been designed for use as a task pane. For more information about designing an Axiom form for use as a task pane, see the *Axiom Forms and Dashboards Guide*.

### Specifying the associated task pane for a file

To associate a task pane with a file, use the **Associated Task Pane** setting within the file. This setting is located on the Control Sheet, in the **Workbook Options** section:




Specify the full path and file name to one of the following:

- A custom task pane file in the Task Panes Library (AXL file)
- A form-enabled file, where the form has been designed for use as a task pane

For example:

```
\Axiom\Task Panes Library\MyTaskPane.axl

\Axiom\Reports Library\Forms\MyTaskPane.xlsx
```

To easily obtain the full path for a file, navigate to that file in the Explorer task pane, then right-click and select **Copy document path to clipboard**. You can then paste the value into the cell.

Alternatively, you can use document shortcut syntax to specify the task pane file. When using a document shortcut, you can add a parameter to specify an alternate tab name for the task pane. If specified, this tab name will be used instead of the file name. For example:

```
document://\Axiom\Task Panes Library\MyTaskPane.axl?AxiomTabName=MyTab
```

To create the document shortcut syntax, take the normal file path and then add the text `document://` to the front of it. To use the optional tab name parameter, append the text `?AxiomTabName=`*Name* to the end of the shortcut.

The next time you open the document after saving, the entry will be automatically converted into a system-managed document shortcut (you can tell the difference by the presence of a `_tid` parameter on the end of the shortcut). If you need to change the entry to point to a different document, or to change the tab name, simply enter the path or document shortcut as you would have originally, and it will be converted again when you save the file.

**NOTE:** The user must have at least read-only permission to the associated task pane file in order to open it. If the user does not have access to the file then the task pane will not open and no error will display. If desired, when configuring security access to the task pane you can clear the **Show in Explorer** option so that the task pane does not display in the Explorer task pane or other "Explorer views."

If the Axiom file is form-enabled, then the associated task pane does not apply when the file is open as an Axiom form, with one exception: if the associated task pane is being used to define navigation links for the form, those links will display in the Navigation panel of the Web Client container. For more information on using an associated task pane to define navigation links for an Axiom form, see the *Axiom Forms and Dashboards Guide*.

### Task pane behavior

When a task pane is associated with an Axiom spreadsheet file, the following behavior occurs:

- The task pane is opened automatically when a user opens the file, and the task pane is made active—meaning the task pane is selected and visible in the Axiom Assistant area. The tab name of the task pane is the file name, unless an alternate tab name has been specified using document reference syntax (see the previous section).
- The task pane is non-closeable. As long as the file is open, the task pane will remain open. The task

pane will automatically close when the file is closed.

- The task pane is linked to the file. If the user moves to a different file, the task pane will hide, and if the user moves back to the file the task pane will become visible again.

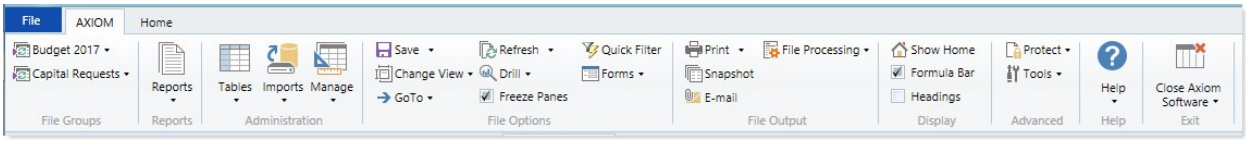
The same task pane can be associated with multiple files. If a user has multiple files open with the same associated task pane, the behavior depends on whether the task pane is a custom task pane or an Axiom form.

- **Axiom form**: Each file has its own separate instance of the task pane (though only a single instance displays at any one time, depending on which file is the currently active file). Each instance maintains its own state for the active document.
- **Custom task pane**: One instance of the task pane is opened. This single instance remains linked to all associated files, but does not maintain a separate state for each file.

# Ribbon tabs

You can use ribbon tabs to provide users with a custom user interface on the ribbon in the Excel Client or the Windows Client. For an overview of how the Axiom user interface can be customized, see About custom task panes and ribbon tabs.

The *ribbon* is the name of the multi-tabbed toolbar area across the top of the Axiom Software application. In the Excel Client this area includes the full Excel ribbon. You might also think of this area as the Axiom Software menu.



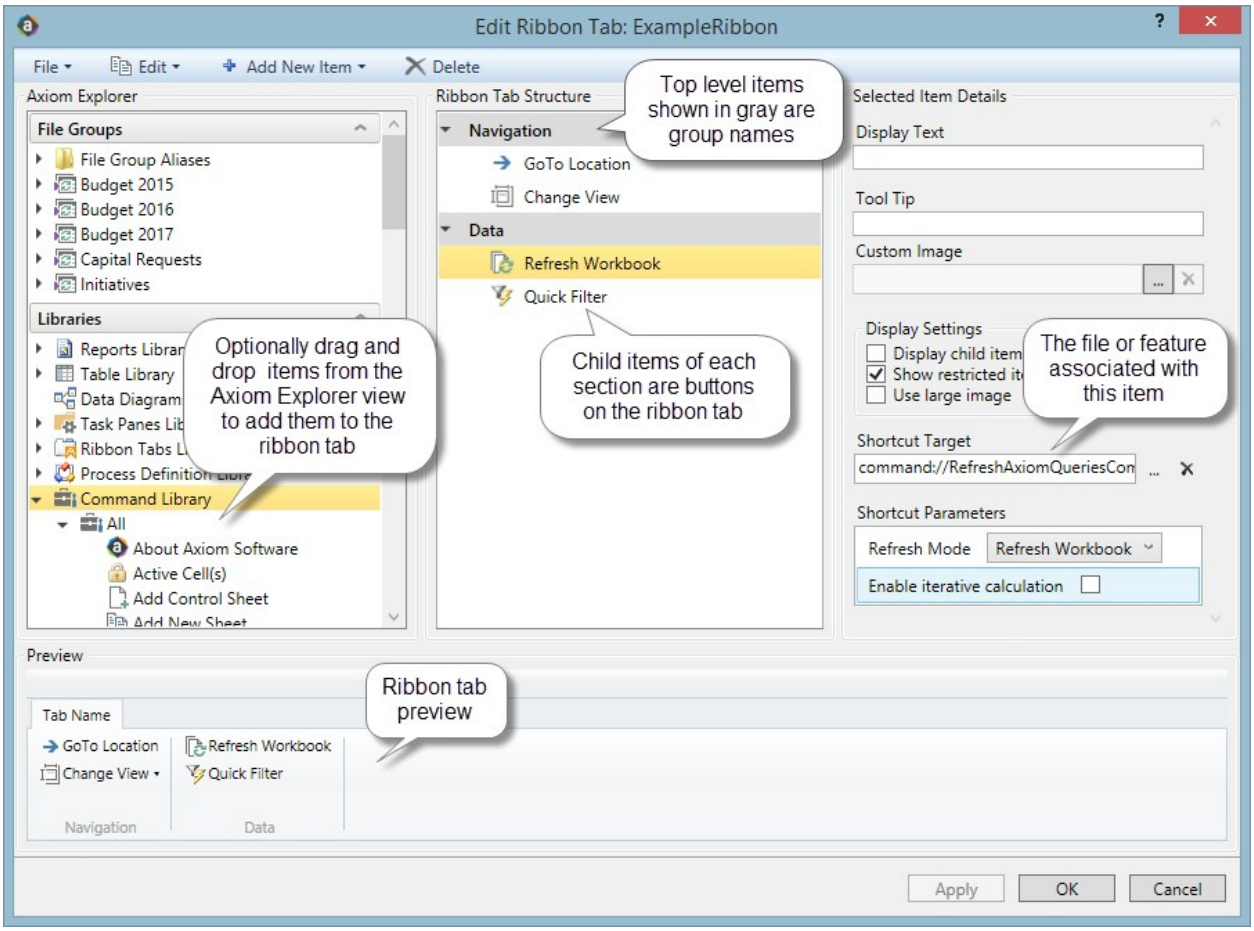

*Example ribbon area in the Windows client*

This section explains how to create, edit, and delete custom ribbon tabs, and then assign them to end users. Ribbon tabs are managed using the Ribbon Tabs Library within Axiom Explorer. Ribbon tab files display using the ribbon icon .

In addition to creating your own custom ribbon tabs, you can customize the built-in ribbon tabs provided by Axiom Software—the main Axiom ribbon tab and the Axiom Designer ribbon tab. For more information, see Built-in task panes and ribbon tabs.

## Ribbon editor

You can create and edit custom ribbon tabs using the **Edit Ribbon Tab** dialog. This dialog has four sections as shown in the following example screenshot.




*Ribbon editor*

The middle section, labeled **Ribbon Structure**, is where you define the groups and individual items to display on the ribbon. To add new items to this structure, you can either:

- Use the **Add New Item** button in the toolbar to add new items. You can choose to add a new child item to the currently selected item, or add a new item at the same level, or add a new group-level item.
- Use the right-click menu to add new items. This has the same options as the Add New Item button.
- Drag and drop an item from the **Axiom Explorer** pane (the left-hand section of the dialog) to the desired location in the ribbon structure. This will automatically link the item in the ribbon structure to the appropriate feature or file (based on what you dragged and dropped).

When you select an item in the ribbon structure, the properties for that item display in the right-hand section of the dialog, labeled **Selected Item Details**. Here you can define display text and other display properties for each item, and assign the associated feature or file for the item using the Shortcut Target. (When you drag and drop and item from the Axiom Explorer pane to the ribbon structure, the shortcut target is automatically assigned.) For more information on the available properties, see Ribbon tab properties.

You can move items in the ribbon structure by dragging and dropping them to various locations, or by using **Copy** or **Cut** and then **Paste**. To remove an item, select it and then click Delete. Any child items of the item you deleted will also be deleted.

The **Preview** section at the bottom of the dialog displays a preview of how the current ribbon structure will look when it is assigned as a startup file and opened in the ribbon. The preview assumes that all items in the ribbon are available to the user. When the ribbon is actually assigned to a user and opened at startup, the display of each item will be dependent on the current file context and on the user's security permissions. The currently selected item in the ribbon structure is also highlighted in the ribbon, so that you can easily find particular items in the preview.

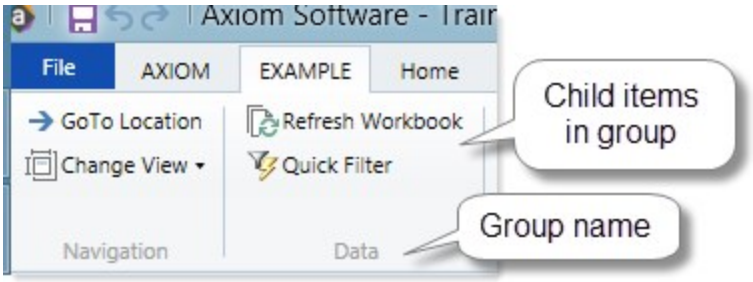
## Managing ribbon tabs

Ribbon tabs can be created, edited, or deleted using **Manage > Ribbon Tabs**, in the **Administration** group of the **Axiom** tab. You can also manage ribbon tabs from the Ribbon Tabs Library in the Explorer task pane.

### Creating a ribbon tab

You can create ribbon tabs for various purposes and display them to users in the application ribbon. Once a ribbon tab is created, it can be assigned to users and roles to open automatically when the user logs into the system.

Ribbon tabs are created by defining one or more groups, and placing one or more child items in each group. The top-level group item is text only and cannot link to any file or feature. Child items in the group can link to any eligible file or feature. The following screenshot illustrates how groups and child items display in the ribbon tab:




As you create the ribbon tab in the editor, a preview of the tab displays in the bottom of the dialog. You can use this preview to see what your ribbon tab will look like when displayed in the ribbon.

**NOTE:** Only administrators and users with the **Administer Task Panes** security permission can create new ribbon tabs.

**To create a new ribbon tab:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Ribbon Tabs**.

   The **Axiom Explorer** dialog opens, with the focus on the Ribbon Tabs Library. You can also use the Explorer task pane for this purpose.

2. In the **Libraries** section, right click the **Ribbon Tabs Library**, and then select **New > Ribbon Tab**.

   The Edit Ribbon Tab dialog opens.

3. New ribbon tab files start with a single group, named Group1. Edit the name of this section using the **Group name** box in the right-hand side of the dialog.

   All top-level items in the ribbon tab are groups. In most cases, the only valid property for a group is the group name.

4. Add child items to the group as needed. You can add child items by doing any of the following:

   - With the group name selected, click **Add New Item > Child of selected item**.
   - With a child item selected, click **Add New Item** and then either **Before selected item** or **After selected item**.
   - Drag and drop an item from the Axiom Explorer pane in the left hand side of the screen to the desired place in the ribbon structure. A black line shows you where the item will be placed.

   You can configure each item as desired. In most cases, you want each item in the ribbon tab to link to an Axiom file or feature. If you dragged and dropped an item from the Axiom Explorer pane into the ribbon tab, then the item is automatically linked to that file or feature. Otherwise, use the **Shortcut Target** setting to assign the item to a file or feature.

   For more information on the types of files and features that can be included in ribbon tabs and how they work, see Linking to Axiom files and features in a task pane or ribbon tab. For more details on available appearance properties, see Ribbon tab properties.

   There are a variety of ways to configure ribbon tabs. For more details on some common design goals and avoiding potential issues, see Ribbon tab design considerations.

5. Add additional groups and child items as needed. You can add new groups by using **Add New Item > New top level group**. Arrange these items as desired by dragging or dropping items within the structure, or by using the **Edit** button to cut, copy, and paste.

6. Check the ribbon **Preview** at the bottom of the dialog, to make sure the groups and items display in the ribbon as you expect.

7. Click **OK** to save.

   The **Save As** dialog opens to the Ribbon Tabs Library. Give the file a name and optionally a description, and save it to the desired location.

**NOTE:** The name of the file determines the default name of the tab in the ribbon. Make sure the file name is brief and intuitive. When assigning the ribbon tab to users in security, you can override the file name and define an alternate tab name using the **Axiom Tab Name** shortcut parameter.

After you have created a ribbon tab, the only way to test it in the ribbon is to assign it to yourself as a startup file in security, and then close and reopen Axiom Software. If the ribbon tab is for a particular user or user group, then an administrator should also test the ribbon tab by assigning it to the appropriate user or group and then logging in as a representative user. See Assigning ribbon tabs to users.

## Editing a ribbon tab

You can edit a ribbon tab at any time, regardless of whether it is currently open in the ribbon. Ribbon tabs currently open in the ribbon do not impact editing availability. If a message tells you that a ribbon tab file is locked, this means that another user has opened the ribbon tab in the editor, not that the ribbon tab is open in the ribbon.

Ribbon tabs are only loaded at startup. If you edit a ribbon tab, you will not see the changes to that ribbon tab until you close and reopen the system (assuming the ribbon tab is configured to open at startup).

**To edit a ribbon tab:**

1. From the **Axiom** tab, in the **Administration** group, click **Manage > Ribbon Tabs**.

   You can also access the Ribbon Tabs Library from the Explorer task pane.

2. In the **Ribbon Tabs Library**, navigate to the ribbon tab that you want to edit, then double-click it to open.
3. In the **Edit Ribbon Tab** dialog, edit the ribbon tab as desired.
4. Click **Apply** or **OK** to save the ribbon tab. If you would rather save this ribbon tab as a new file, you can click **File > Save As** from the dialog toolbar.

## Deleting a ribbon tab

You can delete a ribbon tab at any time. It does not matter if the ribbon tab is currently open on the ribbon in your session or any other user's session.

If a ribbon tab is currently open in a user's session and that ribbon tab file is deleted, the ribbon tab will remain useable in that session until the user exits the system. If the deleted ribbon tab was configured to open on startup, no special steps are required to remove it from user configurations—the next time users log in, the ribbon tab will simply no longer exist and therefore will be ignored by the startup configuration. However, it is a good idea to remove the obsolete configuration from Security so that user configurations match what you expect to be displayed.

**To delete a ribbon tab:**

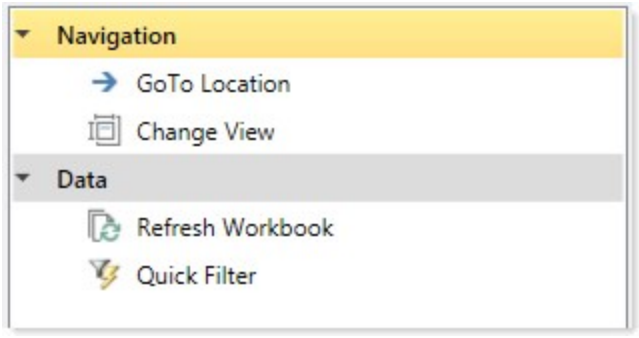
1. From the **Axiom** tab, in the **Administration** group, click **Manage > Ribbon Tabs**. You can also delete ribbon tabs using the Explorer task pane.
2. In the **Ribbon Tabs Library**, navigate to the ribbon tab that you want to delete, then right-click and select **Delete**.

## Ribbon tab design considerations

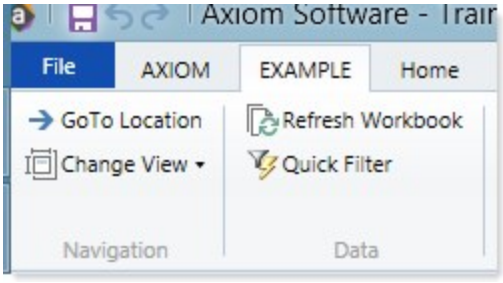
Keep in mind the following design considerations when creating ribbon tabs.

### Creating ribbon tabs with multiple groups

All top-level items in a ribbon tab define groups within the ribbon. Each group can have one or more child items that display as buttons in the group.




*Sample ribbon tab definition with two groups*




*Display of tab in the ribbon*

Most ribbon tabs will have multiple groups. Groups are used to organize the ribbon into logical sections and help users find the buttons they are looking for. If all buttons in the ribbon tab are within a single group, this may make the ribbon tab difficult to use.

In most cases, the only valid property for a top-level item is the display text to define the group name. Top-level items in a ribbon tab cannot be assigned to a shortcut target and do not use any of the display options. However, there are two exceptions:

- The Excel Command Button command can be placed at the top level, to display an entire Excel group in the ribbon tab.
- An Axiom Managed Group command can be placed at the top level, to include the entire contents of an Axiom group in the ribbon tab.

Both of these exceptions are discussed later in this design considerations topic.

### Using Axiom Managed Groups in a ribbon tab

Axiom Managed Groups are special commands that allow you to include the entire contents of a particular ribbon group in a custom ribbon tab. The contents of each group are based on the default Axiom ribbon tab that is delivered with the application.

The advantage of using an Axiom Managed Group is that the group will remain in sync with the default Axiom ribbon tab as you upgrade the software. So if a new feature is added to the File Output group in a particular release, and you are using the Axiom Managed Group command for File Output in a custom ribbon tab, then the new feature will automatically be included in the custom ribbon tab. If instead you have customized the File Output group and are no longer using the Axiom Managed Group command, then you would need to manually add the new feature to the custom ribbon tab if you wanted it to display.

The built-in Axiom ribbon tab (`AxiomMain.AXL`) is made using Axiom Managed Groups. If you want to create a new custom ribbon tab that uses these groups, it is easiest to copy this file and then edit the copy as desired. However, if you want to manually add an Axiom Managed Group to a custom ribbon tab, you can do so as follows:

- In the **Axiom Explorer** pane of the **Edit Ribbon Tab** dialog, navigate to the **Groups/Sections** folder of the **Command Library**.
- Drag and drop the desired group from the Command Library to the **Ribbon Structure** pane, as a top-level group item.

It is not possible to add a new "blank" group item and then link it to the group command, because by default the Shortcut Target field is not available for group items. You must either drag and drop the command as a top-level item directly, or create a new child item and link it to the command, then move the child item up to the top level.

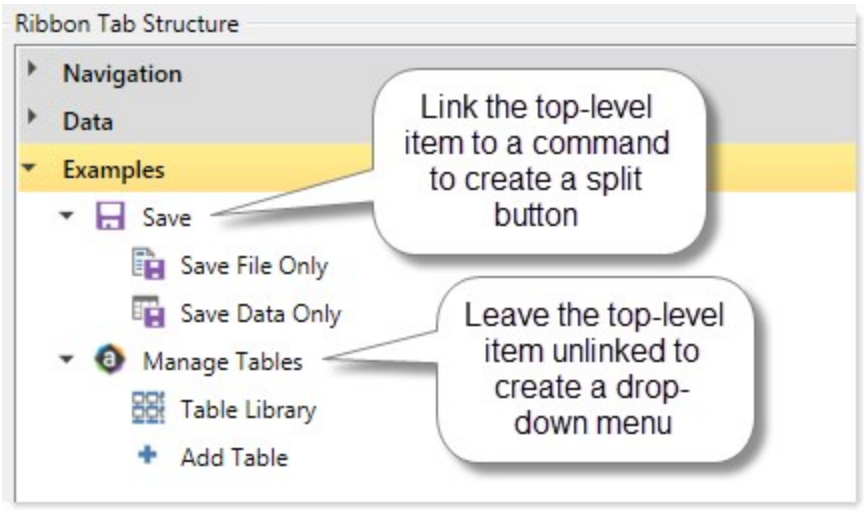
If a ribbon tab already contains an Axiom Managed Group, and you want to customize the contents of that group to remove or add features (or otherwise modify the configuration of individual items in the group), then you can "convert" the group to its individual commands. To do so, right-click the group and then select **Convert group to individual items**. The group will be replaced with all of the individual items in that group, so that you can modify the group as desired. However, keep in mind that converted groups are no longer system-managed and will not automatically update for future changes.

### Creating drop-down menus and "split buttons" in a ribbon tab

Non-group items in a ribbon tab can also have child items. If the user clicks on the "parent" item in the ribbon tab, a drop-down menu allows the user to click one of the child items. This will either perform the configured action for the child item, or open a further sub-menu if the item has child items.

Some features that you can link to in a ribbon tab are automatically drop-down menus. For example, if you use the **Axiom Function** command and select a function group in the shortcut parameters, then clicking this button will automatically display a drop-down menu of the functions in that group. In this case you should not place any child items under this item, as this will cause the default menu to not display as expected.

It is also possible to configure "split buttons" in ribbon tabs. A split button is a button that performs a default command, but also has a secondary drop-down menu where users can select related commands. For example, the Refresh button on the default Axiom tab is a split button. To create a split button, you assign a shortcut target to the "parent" item, and then also assign child items to that item. This configuration will automatically render as a split button in the ribbon tab.

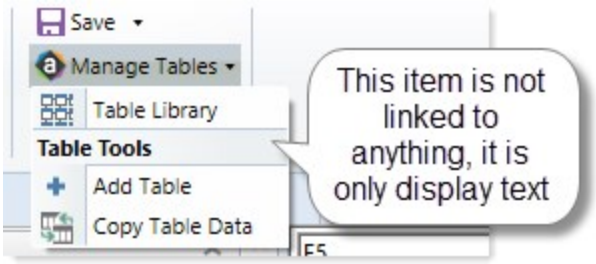



*Example ribbon tab structure*

When this ribbon definition is rendered as a ribbon tab, the Examples group will have two buttons:

- **Save button**: Because the Save parent item is linked to a command, users can click on the Save button to perform a save, or they can use the secondary drop-down menu to save the file only or data only. This would be similar to the Save button on the default Axiom tab.
- **Manage Tables button**: Because the Manage Tables parent item is *not* linked to a command, clicking that button does not perform an action. Instead it brings up a drop-down menu where users can open the Table Library or add a table.

If the split button or drop-down menu has several items in it, you may want to use separator text to organize the menu. You can do this by adding an item where you want the separator text to display, and then only defining display text for that item. This non-linked item will automatically display as separator text in the drop-down menu. For example:




## Hiding items in the ribbon tab

By default, the option to **Show restricted item** is selected for all child items in a ribbon tab. This means that the item will always display in the ribbon tab, but it will be grayed out and unavailable if the user does not have permission to use the feature, or if the feature does not apply to the current context.

In most cases, it is recommended to leave this option selected. It may be disorienting to users to have options in the ribbon tab appear and disappear if it is not obvious why this behavior is occurring. Users may wonder why a particular command no longer displays on the ribbon tab, without realizing that the command does not apply to the current context. However, in certain cases it is best to disable this option—such as when placing file groups and file group categories on the ribbon, so that users do not see file groups that they do not have access to.

Leaving this option selected for most items in a ribbon tab also avoids the potential issue of displaying entirely blank ribbon tabs. Depending on the commands in your ribbon tab, it is possible to end up in situations where none of the commands apply to the current context, and therefore the entire ribbon tab is blank. For example, if your home page is an Axiom form instead of a spreadsheet file, then none of the commands that can be normally used in a spreadsheet file apply to this context, and they will not display on the ribbon tab unless **Show restricted items** is selected.

**NOTE:** This option is not available for Axiom Managed Groups; the visibility of commands in the group is system-managed.

## Including Excel features on an Axiom ribbon tab

You can include Excel features on an Axiom ribbon tab, using the **Excel Command Button** in the Command Library. You can enter Excel command IDs to incorporate virtually any Excel feature on the ribbon. This feature is only supported for use in the Excel Client; the Windows Client will ignore any item set to the Excel Command Button.

It is also possible to include entire Excel command groups on the ribbon. To do this:

- From the Axiom Explorer pane in the Edit Ribbon Tab dialog, locate the Excel Command Button command in the Command Library, and then drag and drop it into the ribbon tab structure as a top-level item.

- Use the shortcut parameters to specify the appropriate **Excel Control ID** for the group that you want to display. The **Excel Control Type** must be set to **Group**.

  Group codes cannot be looked up in the Excel Options dialog like button codes. You will need to download the appropriate resources from Microsoft to look up the desired group codes, as discussed in the Excel Command Button topic referenced above.

- The group node cannot have any child items. Excel will automatically populate the group node with the features that normally display in the specified group.

### Displaying child items inline

If an item has child items, you can choose to display those items inline (at the level of the parent item), omitting the display of the parent item. To do this, you must enable **Display child items inline** for the parent item.

For example, you may want to create a drop-down menu with a couple of feature commands at the top, followed by the contents of a particular folder (similar to the Reports menu). To do this, you can add the folder to the ribbon tab structure, and then enable **Display child items inline** for that item. When the ribbon is rendered to users, the folder item will be omitted and instead the individual files in the folder will dynamically display in the drop-down menu.

The inline behavior applies whether the child items are defined in the ribbon tab structure, or if the child items will be generated dynamically based on the parent item. In the previous example, the individual files aren't defined in the ribbon tab structure, they are generated dynamically based on the folder parent item. However, keep in mind that if the child items are displayed directly on the ribbon instead of in a drop-down list, then they will not update dynamically as changes are made to the folder in that session.

## Ribbon tab properties

The following properties are available for items in a ribbon tab. If you are editing one of the built-in ribbon tabs (Axiom and Axiom Designer), see the following for more information on those tabs: Built-in task panes and ribbon tabs.

The available settings for each item in a ribbon tab depends on whether it is a top-level group item, or a child item in a group.

### Top-level group items

Each top-level item in the ribbon structure is a group item. Group items are identified in the ribbon structure with a gray bar.

Generally speaking, group items define the label for each "group" or section of items in the ribbon tab. Most group items only have a single property of Group Name. However in a few cases groups can have assigned Shortcut Targets and can be used to include entire groups of content in the ribbon.

| Item | Description |
| --- | --- |
| Group Name | The label to display for the group in the ribbon tab.<br><br>If the group has an assigned Shortcut Target, then you can leave this box blank and the group name will be determined by the shortcut target.<br><br>Otherwise, if no group name is defined then the group will have no label in the ribbon tab. The only indicator of a separate group will be the divider lines between groups. |
| Use large image | This option only applies if the group item is an Axiom Managed Group. If this option is selected, then all items that belong to the managed group will display using the large versions of their associated icons, instead of the small versions. By default, all items display using the small versions of their associated icons.<br><br>**NOTE:** Using large icons can quickly fill up the ribbon tab. It is recommended to use small icons for managed groups unless you know there will be adequate space on the ribbon tab for the large icons. |
| Shortcut Target | The target command in the Command Library to use for the group item. This is only permitted in two circumstances:<br><br>• You can drag and drop the **Excel Command Button** command as a top-level item and configure it to display a predefined Excel group. This is only supported for use in the Axiom Excel Client; the Axiom Windows Client will ignore the group and hide it in the ribbon tab.<br>• You can drag and drop an Axiom Managed Group command as a top-level item, to automatically display all items of that group in the ribbon tab.<br><br>**NOTE:** It is not possible to add a "blank" group item to the ribbon structure and then link it to the desired shortcut target, because the Shortcut Target field does not display by default for group items. Instead, you must drag and drop an eligible command from the Axiom Explorer pane to the Ribbon Structure pane. (Or, you can add a child item to the ribbon structure, link it to the desired command, and then move that child item up to the group level. This is only permitted if the command is eligible to be used at the group level.) |
| Shortcut Parameters | This section only applies if the group item uses the Excel Command Button. You can use the shortcut parameters to specify which Excel group should display on the ribbon tab. |

### Child items

Each child item within a group defines a feature or file to display in the ribbon tab.

| Item | Description |
|---|---|
| Display Text | The text to display for the item in the ribbon tab. If the item has a defined shortcut target to an Axiom file or feature, then you can leave this box blank, and the display text will be the name of the file or feature. |
| Tool Tip | The text to display as a tool tip when a user hovers their cursor over the item. This text can be helpful to explain the purpose of the item within the context of the ribbon tab. If no tool tip text is defined, then the tooltip displays the full path of the shortcut target (if defined). |
| Custom Image | The image to display for the item on the ribbon tab. Click the **Browse** button **[...]** to select an image that is stored in the Reports Library. If the image is not already saved in the Reports Library, you can right-click a folder and select **Import** to import the image (if you have the appropriate rights to do so).<br><br>The image must be at least 32x32 pixels, and the file format must be JPG or PNG. Larger images will be resized to fit. The image should be square because the aspect ratio will be retained when resizing.<br><br>**NOTES:**<br>• Users must have permission to the image file in order to see it rendered in the form. It is recommended to create a dedicated Images folder in the Reports Library and store all images in this location. You can grant access to this folder using the Everyone role, or you can create subfolders and grant access to users and roles as needed.<br>• The next time you open this file after saving, the path to the image will be automatically converted into a system-managed document shortcut (you can tell the difference by the presence of a `_tid` parameter on the end of the shortcut). This is to make the file reference "repairable" in cases where the file is renamed or moved. Note that if the path is a result of a formula instead of directly within the cell, then the conversion will not occur and the file reference will not be repairable.<br><br>If no custom image is defined, then the default Axiom image for the shortcut target will be used.<br><br>This option does not apply if the shortcut target for the item is a command that displays with a check box (such as Freeze Panes). |

### Display settings

| Item | Description |
|---|---|
| Display child items inline | Select this option if you want all of this item's child items to display inline, omitting this item. For example, this might be used when linking to a folder, to hide the "folder" node and instead display the resulting files inline.<br><br>**NOTE:** If this option is selected for an item that does not have any child items, then the item will be hidden in the ribbon tab. |
| Show restricted item | Select this option if you want this item to always display in the ribbon tab, regardless of the user's security permissions and regardless of the current context. This option is selected by default. If the item cannot be used by the user, then it will display as grayed out.<br><br>If this option is disabled, then the item will not display on the ribbon tab if it cannot be used. It is common to disable this option when placing file groups on a ribbon tab, so that users only see the file groups that they have security permission to access. |
| Use large image | Select this option if you want the item to display using the large version of its associated icon, instead of the small version. By default, all items display using the small version of the associated icon.<br><br>This option does not apply if the shortcut target for the item is a command that displays with a check box (such as Freeze Panes), or if the item is rendered in a drop-down list on the ribbon. Only items that display directly on the ribbon as buttons are eligible to use large icons. |

### Shortcut settings

| Item | Description |
|---|---|
| Shortcut Target | The target file or feature for the item. Click the **Browse** button [...] to select the desired file or feature. You can select any item that you can navigate to within Axiom Explorer, including using commands in the Command Library. For more information, see Linking to Axiom files and features in a task pane or ribbon tab.<br><br>Alternatively, you can drag and drop items from the Axiom Explorer pane in the editor directly to the ribbon tab structure, to both create a new item and link it to that file or feature.<br><br>Once a shortcut target is defined, Axiom Software places the necessary syntax to open the target within the **Shortcut Target** box. It is best not to manually edit this text, as you may inadvertently render the syntax invalid. If you want to select a different target, use the Browse button to select the target rather than editing the target syntax. If you no longer want the item to link to a shortcut target, click the Delete button ✕ to remove the shortcut.<br><br>**NOTE:** All child items in the ribbon tab structure should be assigned a shortcut target, with one exception. If you are configuring an item as a split button or a drop-down menu, and you want to place a separator in the drop-down menu, then you would leave the shortcut target for the item blank and just define display text. Otherwise, child items without an assigned shortcut target will not display in the ribbon. |
| Shortcut Parameters | Most shortcut targets have one or more associated shortcut parameters that impact the behavior of the linked file or feature when it is opened from the ribbon tab. The available parameters depend on the shortcut target—for example, report files have different parameters than tables.<br><br>For more information, see the following:<br><br>• Shortcuts<br><br>You can link to various files and other items in Axiom Software using shortcuts. For more information, see Axiom Software Help: **Reference > Shortcuts** (or search on code **AX1169** to bring up the Shortcuts section).<br><br>• Command Library<br><br>You can link to various Axiom Software features using commands in the Command Library. For more information, see Axiom Software Help: **Reference > Command Library** (or search on code **AX2685** to bring up the Command Library section). |

# Assigning ribbon tabs to users

You can assign ribbon tabs to users and roles using the Startup tab in Axiom Software Security. When a user starts Axiom Software, their assigned startup ribbon tabs will automatically open.

Ribbon tabs can only be opened at startup. It is not possible to allow users to open ribbon tabs "on demand". Generally speaking, the only reason to grant a user file access to ribbon tabs in the Ribbon Tabs Library is if the user needs to create and edit ribbon tabs.

## Assigning as startup files

To configure a ribbon tab to open on system startup for a particular user or role, use the **Startup** tab of Security. Any ribbon tabs listed on the Startup tab will open automatically when the user starts the Excel Client or the Windows Client. Ribbon tabs do not apply to the Web Client. For more information on assigning startup ribbon tabs, see the *Security Guide*.

When ribbon tabs are assigned as startup files, users do *not* otherwise need to be granted access to these files. The files will open automatically at startup. Keep in mind that the visibility of the ribbon tab depends on its configuration. Using the shortcut properties for the startup file, you can configure the ribbon tab so that it only displays if certain conditions are met, such as:

- Whether the user is an administrator
- Whether the user has access to the Sheet Assistant
- Whether a particular type of file is open

For example, a ribbon tab can be associated with report file types. Therefore, although the ribbon tab is technically opened at startup, it does not actually display to the user until a report file is opened.

In most cases, you will assign startup ribbon tabs on a per role basis. For example, you might have a Budget User role, and you want all users in this role to see a simple ribbon tab that presents the handful of items that they need to access. Alternatively, if all users in a system need to see a particular ribbon tab on startup, you can configure it for the Everyone role.

Startup ribbon tabs will be opened in the following order:

1. Ribbon tabs assigned to the Everyone role.
2. Ribbon tabs assigned to other roles that the user belongs to (multiple roles are sorted in alphabetical order).
3. Ribbon tabs assigned to the user.

All custom ribbon tabs display before (to the left of) any Excel ribbon tabs. In the case of the Windows Client, custom ribbon tabs display before the Home tab. Once a ribbon tab is opened, it cannot be closed by the user (though it may become hidden depending on its configuration and the current context).

### Access to the Ribbon Tabs Library

It is not possible to open a ribbon tab "on demand" from the Ribbon Tabs Library. If a user double-clicks on a ribbon tab file, it will open in the ribbon tab editor, not in the ribbon. The only way to access a ribbon tab is to assign it as a startup file and then restart the application. Therefore, there is no reason to give a user file permission to ribbon tab files in the Ribbon Tabs Library unless the user needs to create and edit ribbon tabs.

If you have non-administrator users who need to create and edit ribbon tabs, those users must be granted the **Administer Task Panes** permission as well as read/write access at a file or folder level in the Ribbon Tabs Library.

# Converting task panes and ribbon tabs

You can convert an existing task pane file into a ribbon tab, and vice versa. The general structure of the task pane or ribbon tab definition is applicable to both formats.

When you convert a task pane or a ribbon tab to the other format, a copy of the original file is created and that new file is converted. The original file remains intact. For example, if you convert a task pane named "Reports," a new ribbon tab file is created within the Ribbon Tabs Library named "Reports". You can then rename the new file as needed.

Task panes and ribbon tabs can use settings that don't apply to the other format, and can link to files and features that aren't supported in the other format. Generally speaking, if this occurs the inapplicable setting or unsupported feature will simply be ignored in the new format after conversion. However, you should test the file in its new format to make sure that everything displays and works as intended.

**To convert a task pane or a ribbon tab:**

1. In Axiom Explorer or the Explorer task pane, locate the task pane file or ribbon tab file that you want to convert.
2. Right-click the file, and then select either **Convert to ribbon** or **Convert to task pane**.

A copy is created of the original file, and that copy is converted to the specified format. The copy is placed in the corresponding library. For example if you convert a ribbon tab to a task pane, the copy is created in the Task Panes Library.

### Considerations when converting a task pane to a ribbon

The following task pane settings are inapplicable to ribbon tabs and will be ignored in that format:

- Display child item images
- Auto-number child items

Some task pane shortcuts, such as the built-in feature controls for Axiom Explorer and Process Management, are not supported in the ribbon format. These items will be ignored when the ribbon is displayed.

Some designs that are well-suited for the task pane format are not well suited for the ribbon format, although they will be technically converted. For example, if the task pane depicts a numbered process, it will be converted to the ribbon but the numbers will be lost, and the process flow will not be as clear as it would be in a task pane.

### Considerations when converting a ribbon to a task pane

The following ribbon settings are inapplicable to task panes:

- Use large image (option will be ignored in the task pane)
- Shortcut targets defined on items with children (users will be unable to launch the shortcut target)

Task panes do not support the concept of a "split button" where you can click on the parent button to perform a particular action, or you can open a sub-menu to perform other actions. In the task pane format, if an item has child items then double-clicking the parent item will only expand or collapse the child items; no action will be performed. If a shortcut target is assigned to a parent item in a task pane, this target will be ignored when the task pane is in use. Within the task pane editor, this configuration will result in a validation error.

# Linking to Axiom files and features in a task pane or ribbon tab

Items in a custom task pane or a ribbon tab can be linked to an Axiom file or feature. To link an item to a file or feature, you use the **Shortcut Target** property for the item. Depending on the shortcut target, there may also be **Shortcut Parameters** to further configure the behavior of the shortcut.

**To define a shortcut for an item in a task pane or a ribbon tab:**

1. In the **Edit Task Pane** dialog or the **Edit Ribbon Tab** dialog, select the item that you want to link to a file or feature.
2. Click the **Browse** button [**...**] to the right of the **Shortcut Target** box. Then, in the Axiom Explorer dialog, navigate to the item that you want to link to.

   You can link to any folder, file, or other item that is displayed in Axiom Explorer. Additionally, the Command Library is available when defining task panes and ribbon tabs, so that you can link to Axiom features. For more information, see the following reference material:

   - Shortcuts

     You can link to various files and other items in Axiom Software using shortcuts. For more information, see Axiom Software Help: **Reference > Shortcuts** (or search on code **AX1169** to bring up the Shortcuts section).

- Command Library

  You can link to various Axiom Software features using commands in the Command Library. For more information, see Axiom Software Help: **Reference > Command Library** (or search on code **AX2685** to bring up the Command Library section).

**NOTE:** You can also link items to files and features by using the Axiom Explorer pane within the editor. If you drag and drop an item from the Axiom Explorer pane to the task pane or ribbon tab definition, then a new item will automatically be created and that item will be linked to the appropriate shortcut target.

3. Configure the available **Shortcut Parameters** for the item, if applicable.

   Some items have additional options that impact how the item behaves in the task pane or the ribbon tab. The available options are specific to the item type.

4. Define the **Display Text** for the item as desired.

   By default, the item will display using the name of the shortcut target—for example, the file name if the target is a file. If you want to specify a different name, then enter different text into the **Display Text** box.

If you assign a shortcut target to an item and then later you want to remove the shortcut, click the **Delete** button to the right of the box. You can also click the Browse button again to select a different shortcut target.

## Visibility of linked items

The visibility of a linked item in a task pane or a ribbon tab depends on the following:

- Whether the user has rights to access the linked file or feature. In most cases, if a user does not have permission to an item, the item does not display in the task pane or ribbon tab. For example, if the item links to a report that the user does not have access to, the item does not display because the user cannot open the report.
- Whether the linked feature applies to the currently active file. In most cases, if the item is a file-related feature but it does not apply to the currently active file type, then the item does not display in the ribbon tab. For example, if the item links to the Add Rows command, but the currently active file is a report, then the item does not display because calc method libraries cannot be used in report files.
- Whether the linked item is valid. If the shortcut target of an item is invalid, then the item does not display in the task pane or ribbon tab because it cannot be used. A user with the appropriate permissions must go to the editor to see and fix the invalid item. For example, if the item links to a report but the report is later deleted, the item does not display because it is invalid.

- Whether **Show restricted item** is enabled for the item. If enabled, then the item will always display, regardless of any of the previously-listed factors. If the item cannot be used for any reason, then it displays as grayed out. You might enable this option if you want the items in the task pane or ribbon tab to remain constant for all users and all contexts.

**NOTE:** The visibility rules above are general guidelines. Certain commands have different behavior. See the help topic for the specific command for more information on the visibility rules for that command.

The default state of **Show restricted item** depends on whether you are creating a task pane or a ribbon tab. In task panes, the option is disabled by default, meaning that items will be hidden if the user cannot use them. In ribbon tabs, the option is enabled by default, meaning that items will display as grayed out if the user cannot use them. For more information, see Task pane design considerations and Ribbon tab design considerations.

# Built-in task panes and ribbon tabs

In addition to the custom task panes and ribbon tabs that you can create for your system, Axiom Software provides certain built-in task panes and ribbon tabs.

Some of these built-in items are entirely system-controlled, such as the Axiom ribbon tab, the Sheet Assistant task pane, or the File Processing task pane. These items cannot be customized.

Other built-in items are provided by default, but can be customized if desired. You can edit these items, modify their display behavior, or disable them entirely if desired. These customizable items are:

- Axiom Designer ribbon tab
- Explorer task pane
- Process task pane
- Workflow task pane (see Axiom Software Help for more information)

## Customizing the Axiom ribbon tab

Axiom Software provides a built-in Axiom ribbon tab to provide access to Axiom functionality. By default, this tab is assigned as a startup file for the Everyone role in security, and is configured to display to all users in all circumstances. If desired you can do any of the following:

- Customize the contents of the Axiom Designer ribbon tab
- Create different versions of the tab and assign them to different users and roles
- Disable the tab for certain users and roles

**IMPORTANT:** If you choose to customize the Axiom ribbon tab, it is possible that you could inadvertently create an environment where no users can access a certain feature, because that feature is no longer available on any menu. You should take care to ensure that administrators always have access to the full set of Axiom features. If you are customizing a version of the Axiom ribbon tab that is meant for administrators, and you choose to remove a feature from that tab, you should make sure that feature is available to administrators in some other way—either in another custom ribbon tab or task pane that is available for administrator use, or in the Axiom right-click menu.

### Default behavior

When a new system is created, or when an older system is upgraded, a copy of the Axiom ribbon tab is placed in the root of the Ribbon Tabs Library, using the file name `AxiomMain.AXL`. This copy is listed as a startup ribbon tab for the Everyone role.

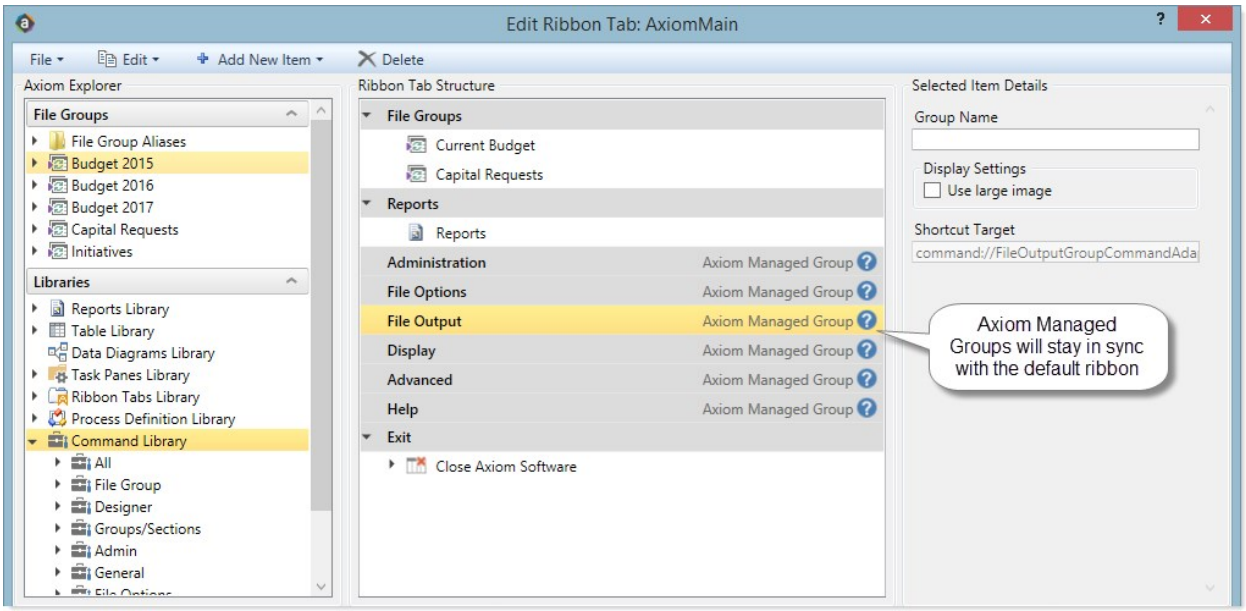
**NOTE:** You can rename the file and/or move it into a subfolder without breaking the existing startup configuration. For example, you may want to create a Startup subfolder in the Ribbon Tabs Library to hold all of the ribbon tabs that are configured to open at startup.

This custom ribbon tab replaces the old system-controlled Axiom ribbon tab. Using a custom ribbon tab for the Axiom tab allows clients to fully customize the Axiom menu as desired. If you do not want to customize the Axiom menu, then you can leave the ribbon tab as is.

### Customizing the contents of the Axiom tab

You can customize the contents of the Axiom ribbon tab by modifying the `AxiomMain.AXL` file, or by copying the file and creating different versions for different user audiences. You can add features to the tab, reorganize the tab, or remove features from the tab. For more information on creating and editing ribbon tabs in general, see Ribbon tabs.

The built-in Axiom ribbon tab is designed using Axiom Managed Group commands. These group commands correspond to the groups (sections) on the default Axiom ribbon, such as Administration, File Options, and File Output. When a group command is used, all features that are part of that group are automatically included on the ribbon.




**NOTE:** A few groups in the default Axiom ribbon tab are not built using group commands. The **File Groups** group is always customized per installation and therefore does not have a defined set of commands. The **Reports** group and the **Exit** group do not have group commands because there is only one button in each group.

The benefit of using managed group commands is that the contents of the group will remain in sync with the default Axiom ribbon. This means that if Axiom Software adds a new feature to the **File Options** group in a future release, then any custom ribbon tabs (or task panes) that use the managed File Options group will automatically update to display that new feature. If instead you decide to customize your ribbon tab so that the File Options group consists of individual commands instead of the managed group command, then your ribbon tab would not automatically update for the new feature. You would have to manually edit your ribbon tab to add it.

Keep in mind the following when customizing the contents of the Axiom ribbon tab:

- It is recommended to use Axiom Managed Group commands whenever it makes sense to do so, because this will make it easier to access new features in the future.
- Use caution before removing entire groups from the ribbon tab. Remember that if a user does not have any access to any items in a particular group, then that group will not display on the ribbon.

  For example, it is not necessary to remove the Administration group from a ribbon tab if you do not want end users to see it, because by default it will not display to those users anyway. But if you choose to remove it, then any end users who do have special security permissions to certain administration features will now have no way to access those features. You should either leave the Administration group on the ribbon and rely on security permissions to hide or show it as necessary, or make sure that you identify those end users who need special access to certain features and assign them a different version of the ribbon tab that contains those features.

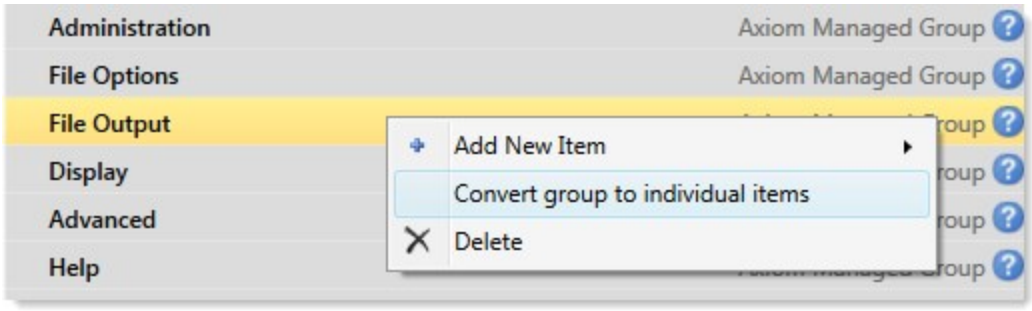
- If you absolutely need to customize the contents of a group, then the easiest way to begin is to "convert" the group to its individual items. You can convert a group by right-clicking it and selecting **Convert group to individual items**. This conversion process removes the group command from the ribbon and replaces it with a collection of individual commands that are configured to match the way those commands display in the group. You can then remove individual commands from the group, or change the configuration of these commands, or add different commands to the group.

## Example customization

Many clients have requested the ability to create different menus for administrators versus end users. You can do this by creating different versions of the Axiom ribbon tab and then assigning them to different users and roles.

Imagine that you do not want your end users to have access to the Snapshot feature on the default Axiom ribbon tab. You can do the following:

- Create a copy of `AxiomMain.AXL` and name it something like `AxiomEndUser.AXL`.
- Modify `AxiomEndUser.AXL` to remove the Snapshot feature:
  - Right-click the **File Output** group, then select **Convert group to individual items**.




  - A warning message informs you that the group will no longer be system-managed. After clicking **OK** to continue, there is now a regular top-level item with a Group Name of File Output, followed by child items for each feature in that group.

    - In the converted File Output group, right-click the items for **Snapshot** and **E-mail** and then click **Delete**. (The E-mail Workbook feature also allows taking a snapshot, so you would need to remove both items if you wanted to completely remove the feature from the ribbon.)

- Modify Axiom security so that end users are now assigned the `AxiomEndUser.AXL` ribbon tab as a startup file instead of `AxiomMain.AXL`.

  The way that you do this will depend on how your security is set up. If all of your end users belong to certain roles, then you can assign `AxiomEndUser.AXL` to those roles. You can then modify the configuration of the Everyone role so that `AxiomMain.AXL` only displays to administrators. Or, you could remove `AxiomMain.AXL` from the Everyone role entirely, and instead assign it to an administrator role or individual administrator users.

When you assign a ribbon tab as a startup file, remember to set the **Axiom Tab Name**. If you do not define the Axiom Tab Name, then the tab will use the file name (in this case, AxiomEndUser).

### Restoring the default Axiom ribbon tab

If you modify the default `AxiomMain.AXL` file and then later you decide that you want to restore the file to its original configuration, a copy of the default file is always available in `\Axiom\Axiom System\Document Templates\Sample Ribbon Tabs`.

Keep in mind that if you delete the copy in the Ribbon Tabs Library and then replace it with a copy from the sample folder, you will need to reconfigure your security settings to point to the new file (because startup shortcuts are based on document ID, not on file name and location). Alternatively you can import the default file over the copy in the Ribbon Tabs Library. To do this in Axiom Explorer:

- Right-click the default file in the Sample Task Panes folder, and select **Export** to save it to your local file system.
- Right-click the Task Panes Library and select **Import Files** to import the file from your local file system. This will overwrite the existing file if they both have the same name, and retain the document ID.

**IMPORTANT:** If your system is accidentally put in a state where an administrator cannot access necessary system features using the Axiom ribbon tab (or if the Axiom ribbon tab does not display at all for an administrator), administrators can correct the situation by accessing Axiom Explorer from the File menu in the Windows Client: **File > Launch Axiom Explorer**. From here an administrator can perform actions such as editing their assigned ribbon tab to add the necessary features, or they can create a temporary task pane to provide access to features such as Security (for example, if the administrator does not currently have an assigned ribbon tab and cannot access Security to assign one).

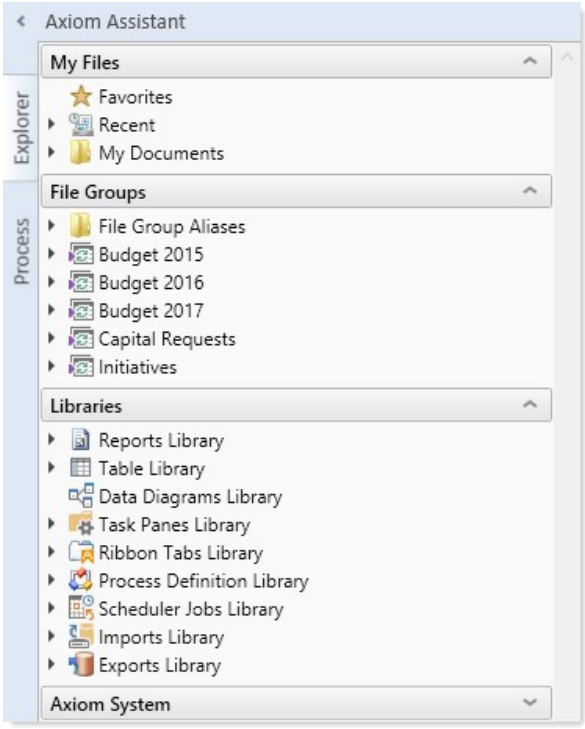
### Disabling the Axiom tab for certain users and roles

In rare cases, clients may want to disable the Axiom tab entirely for certain users or roles. Typically this would only be done if you intend to provide access to all necessary functionality using a custom task pane instead of a ribbon tab.

To do this, you would remove `AxiomMain.AXL` from the Everyone role and instead assign it only to the users and roles that need it (or you could adjust the configuration on the Everyone role so that the ribbon tab only displays to administrators). Any users and roles who are not assigned a ribbon tab would need to be assigned a custom task pane that is set up with all of the functionality they need.

## Configuring the Explorer task pane

Axiom Software provides a built-in Explorer task pane to provide users with quick access to their favorites, recent items, and all Axiom files that they have rights to access. By default, all users are given access to this task pane. If desired, you can disable use of this task pane entirely, or restrict access to certain sets of users.



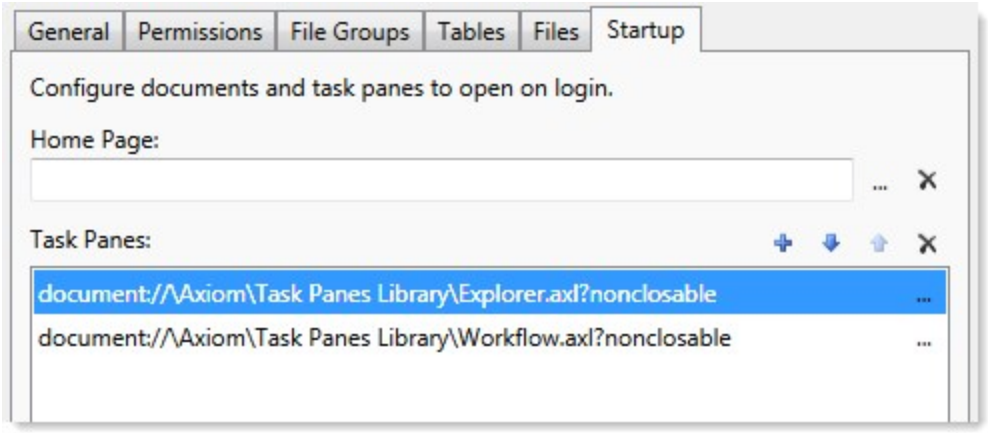

*Example Explorer task pane for an administrator*


*Example Explorer task pane for an end user*

**NOTE:** Access to the Explorer task pane is managed using the startup file settings in Security. The **Administer Axiom Explorer** permission does not apply to the task pane—it only controls access to the full dialog accessible from **Manage > Axiom Explorer**.

## Default behavior

When a new system is created, a copy of the Explorer task pane is placed in the root of the Task Panes Library (file name: `Explorer.AXL`). This copy is listed as a startup task pane for the Everyone role, and is configured to prevent closing.




This means that the Explorer task pane is automatically opened when the system is started, for all users. Users cannot close the task pane, so it is always available. Users do not otherwise have access to the task pane (meaning, no access is granted to the file itself on the **Files** tab of Security).

**NOTE:** You can rename the `Explorer.AXL` file and/or move it to a subfolder without breaking the existing startup configuration. You can also edit the shortcut parameters for the startup configuration to use an alternate tab name, or to allow users to close the task pane if desired.

## Disabling the Explorer task pane

If you do not want to use the Explorer task pane at all in your system, then you can edit the startup settings for the Everyone role in Security, to remove the Explorer task pane from the list of startup task panes. Once it is removed from the Everyone role, then the task pane will not display for anyone unless you later manually add it to other roles or users.

If you want to delete the file entirely so that any users with access to the Task Panes Library cannot see it, you can do so. However, you cannot delete the file if it is open in your current session. You must first edit the settings for the Everyone role as previously described, and then you must exit the system and log in again. Now that the Explorer task pane is not open in your session, you can delete the file from the Task Panes Library.

If you deleted the `Explorer.AXL` file and then later decide that you want to use it, a copy of the file is always available in `\Axiom\Axiom System\Document Templates\Sample Task Panes`. Make a copy of this file and save it to the Task Panes Library, and then configure access in Security as desired.

### Modifying access to the Explorer task pane

If you want to use the Explorer task pane, but you do not want all users to have access to it, then you can modify the Security settings as desired.

- First, edit the startup settings for the Everyone role to remove the Explorer task pane from the list of startup task panes.
- Then, add the Explorer task pane to the startup settings for the applicable roles or users. Select the **Non-closeable** check box if you want users to always have access to this task pane. If you allow the task pane to be closed, remember that users without access to the Task Panes Library will not be able to reopen the task pane within the current session if they close it.

### Customizing the Explorer task pane

The Explorer task pane uses a single item, the **Axiom Explorer Tree View** from the Command Library. This control replicates the Axiom Explorer treeview within a task pane. While it is possible to manually re-create the Explorer look and feel in a task pane by linking to each individual component, using the control provides advantages such as:

- **Access to relevant features.** When using the Explorer control, users with the appropriate permissions gain access to item-specific commands via the right-click menu. For example, users can right-click a file to open it as read-only, or they can right-click a table to edit the table structure. If instead an item is linked on a task pane directly, then the item-specific right-click commands are not available.
- **Automatic updates.** The Explorer control renders the Axiom Explorer treeview as it currently exists. If future upgrades introduce new libraries or a changed folder structure, the control will remain in sync.

It is possible to make a few minor customizations to the Explorer task pane. If desired, you can use shortcut parameters to hide any of the major sections: My Files, File Groups, Libraries, and Axiom System Folder. Keep in mind that security permissions are always applied, so users without rights to a particular section or item will not see it in the task pane. You only need to hide a section if you do not want it to display for *any* users, within the task pane.

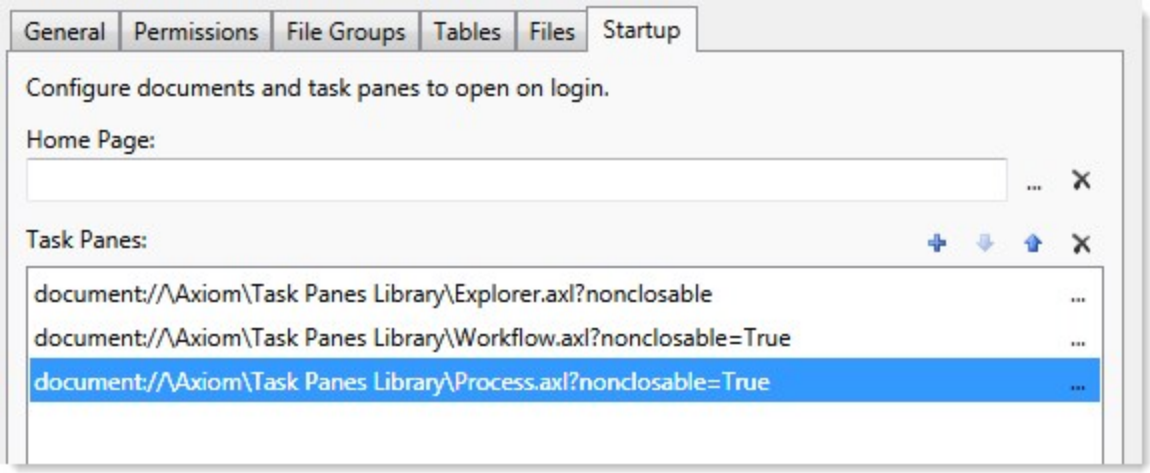
## Configuring the Process task pane

Axiom Software provides a built-in Process task pane to allow the assigned owners of process steps to easily complete their assigned tasks, and to allow administrators to monitor active processes. This is applicable if you are using the Process Management feature to define and manage processes.

By default all users are given access to this task pane, however, the task pane will only display to non-admin users if the user is the owner of an active step in a process. The task pane will display to administrators if any process is made active. If you never define or activate any processes in your system, or if a user never has any active tasks in a process, then the task pane will never display.

## Default behavior

When a new system is created, a copy of the Process task pane is placed in the root of the Task Panes Library (file name: `Process.AXL`). This copy is listed as a startup task pane for the Everyone role, and is configured to prevent closing.




This means that the Process task pane will dynamically display depending on whether any Process Management processes are ever made active. If they are, then administrators will always see the task pane if there are any active processes, and non-admin users will only see the task pane when they are the assigned owner of an active step in a process. Users do not otherwise have access to the task pane (meaning, no access is granted to the file itself on the **Files** tab of Security).

**NOTE:** You can rename the `Process.AXL` file and/or move it to a subfolder without breaking the existing startup configuration.

## Disabling the Process task pane

If you do not want to use the Process task pane at all in your system, then you can edit the startup settings for the Everyone role in Security, to remove the Process task pane from the list of startup task panes. Also, if you want to delete the file entirely so that any users with access to the Task Panes Library cannot see it, you can do that too. However, remember that users will only see the task pane when it is relevant to them, so we recommend leaving the configuration as is in case you change your mind in the future and decide to use Process Management.

If you deleted the `Process.AXL` file and then later decide that you want to use it, a copy of the file is always available in `\Axiom\Axiom System\Document Templates\Sample Task Panes`. Make a copy of this file and save it to the Task Panes Library, and then configure access in Security as desired.

## Customizing the Process task pane

The Process task pane uses a single item, the **User Process View** from the Command Library. This command does not have any shortcut parameters to configure its behavior.

If desired, you can include this control in a custom task pane along with other items as desired, and use that custom task pane instead of the built-in task pane. If you do this, you should remove the built-in task pane from the Everyone role as described previously, and instead configure security for your custom task pane as needed.

## Configuring the Axiom Designer ribbon tab

Axiom Software provides a built-in Axiom Designer ribbon tab to provide various tools to assist in creating and testing files. By default, this tab is assigned as a startup file for the Everyone role in security, and configured to only display to administrators. If desired you can do any of the following:

- Customize the contents of the Axiom Designer ribbon tab
- Modify the configuration of the startup file on the Everyone role so that more users have access to the ribbon tab
- Remove the startup file from the everyone role and then reassign it to other users and roles as desired

### Default behavior

When a new system is created, or when an older system is upgraded, a copy of the Axiom Designer ribbon tab is placed in the root of the Ribbon Tabs Library (file name: `AxiomDesigner.AXL`). This copy is listed as a startup ribbon tab for the Everyone role, and is configured as follows:

- **Requires Admin** is enabled. The Axiom Designer tab will only display to users who are administrators.
- **Requires Sheet Assistant** is enabled. The Axiom Designer tab will only display when the active file is an Axiom spreadsheet file and the Sheet Assistant is also present.

**NOTE:** You can rename the file and/or move it into a subfolder without breaking the existing startup configuration.

### Changing the startup configuration

If you want to expose the Axiom Designer ribbon tab to other users, you can change the startup configuration for the file. For example:

- You can leave the file on the Everyone role, but modify the shortcut parameters to remove the administrator-only requirement. You could allow all users to see it, or you could only require Sheet Assistant permission so that only "file designers" would see the ribbon tab. For more information on the available shortcut parameters, see the Security Guide.
- You can remove the file from the Everyone role, but instead assign it to different users and/or roles. For example you might have a particular role for report writers or other power users, and you could assign it to that role.

### Customizing the contents

If desired you can customize the contents of the Axiom Designer ribbon, by modifying the `AxiomDesigner.AXL` file. You can add additional features to the ribbon tab, reorganize the tab, or remove existing features. You could also copy the file and create different versions of the tab, and assign those versions to different users and roles. For more information on creating and editing ribbon tabs, see Ribbon tabs.

If you modify the default `AxiomDesigner.AXL` file and then later you decide that you want to restore the file to its original configuration, a copy of the default file is always available in `\Axiom\Axiom System\Document Templates\Sample Ribbon Tabs`. Keep in mind that if you delete the copy in the Ribbon Tabs Library and then replace it with a copy from the sample folder, then you will need to reconfigure your security settings to point to the new file (because the startup shortcuts are based on document ID, not on file name and location). You can instead import the default file over the copy in the Ribbon Tabs Library; this will retain the document ID.


# Alerts

Using alerts, you can define a condition to monitor and then notify one or more users if that condition is met. For example, you may want to alert certain users if their budget-to-actuals variance for a particular month exceeds a specified amount.

Alerts are defined within Axiom files, on a dedicated Alert Control Sheet (Control_Alert). To define an alert condition, you specify properties such as:

- The condition to be evaluated to trigger the alert.
- The recipients for the alert notification if it is triggered.
- The message to be displayed to alert recipients.
- The method of alert delivery (email and/or task pane).
- An optional link to the source file or another file to provide the user with more information about the alert.

Once alerts have been defined, you can process them in a variety of ways. You can set up alerts within a report and then process alerts for that report periodically using File Processing or by using the Process Document List task in Scheduler. Or you can set up alerts within a plan file, and then those alerts will be processed when the plan file is run using Process Plan Files.

If an alert condition evaluates to True when alerts are processed, then the specified recipients receive an alert notification. This notification is displayed in the Notification task pane and/or delivered via email, depending on the alert settings. The user can read the notification, navigate to the specified supporting file, and delete the notification when they are finished reviewing it. Alternatively, you can query alert notifications using the Axiom.Notifications system table and display them in a file, such as the Home file, or in an Axiom form.

## Creating alerts

Alerts can be created within any Axiom file, by adding an Alert Control Sheet to that file.

- Alerts should be added to a report file if you want to report upon some data and then alert based on that data. Report alerts can be processed at any time by using File Processing (to leverage multipass processing) or by using the Process Document List task in Scheduler.

- Alerts should be added to a template (for use in plan files) if you want to alert based upon data within a particular plan file, and you want the alerts to be processed when running Process Plan Files.

Alerts are based on a condition defined within the file. Before starting to create alerts, you should take some time to consider exactly what condition you want to track, and what data you need to evaluate that condition.

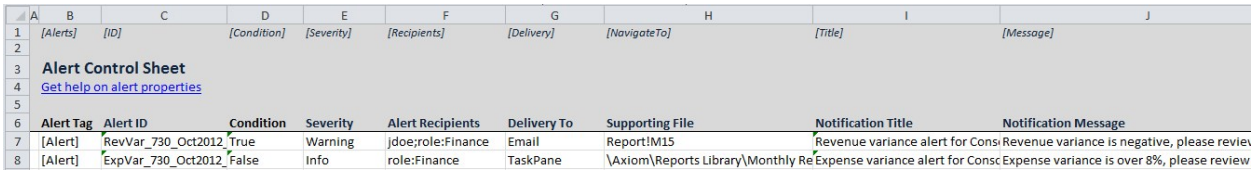
Only administrators can enable a file for alerts. Once the Alert Control Sheet has been added to a file, any user with access to the file and access to the Alert Control Sheet can define alerts.

**To enable alerts for a file:**

- On the **Axiom Designer** tab, in the **Developer** group, click **Tools > Add a Control Sheet > Alerts**.

The Alert Control Sheet is added to the file as **Control_Alert**. All alerts for the file must be defined on this sheet. By default, the sheet is set up with a header row for the alert tags, and a couple of sample alerts (which you can delete or leave for reference as desired).

You can define alerts by adding rows to this sheet. Each row should start with an `[Alert]` tag (with brackets), and then complete each alert property (without brackets). For more information, see Alert properties.



| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | *[Alerts]* | *[ID]* | *[Condition]* | *[Severity]* | *[Recipients]* | *[Delivery]* | *[NavigateTo]* | *[Title]* | *[Message]* |
| 2 | | | | | | | | | | |
| 3 | | **Alert Control Sheet** | | | | | | | | |
| 4 | | Get help on alert properties | | | | | | | | |
| 5 | | | | | | | | | | |
| 6 | | **Alert Tag** | **Alert ID** | **Condition** | **Severity** | **Alert Recipients** | **Delivery To** | **Supporting File** | **Notification Title** | **Notification Message** |
| 7 | | [Alert] | RevVar_730_Oct2012_ | True | Warning | jdoe;role:Finance | Email | Report!M15 | Revenue variance alert for Cons | Revenue variance is negative, please revie |
| 8 | | [Alert] | ExpVar_730_Oct2012_ | False | Info | role:Finance | TaskPane | \Axiom\Reports Library\Monthly Re | Expense variance alert for Consc | Expense variance is over 8%, please review |

The default format of the Alert Control Sheet is provided for your convenience. However, you can set up this sheet however you like; you do not have to use this format. For example, you may want to define alerts using an Axiom query instead of "hard-coding" them into the default grid. If so, then you can set up an Axiom query on the Alert Control Sheet. As long as each column has the appropriate alert header tag (as shown in row 1 above), and each alert row is flagged with an [Alert] tag, then alerts can be placed anywhere on the sheet. However, all alerts must be defined on the Alert Control Sheet—you cannot define them on other sheets in the file.

**NOTE:** Simply defining the alerts within the file does not cause any alerting actions to take place. It does not matter if a particular condition currently resolves to True or False within the file; the condition status only matters when the alerts are processed.

## Setup considerations for File Processing

If you plan to process alerts using multipass File Processing, then many alert settings should be set up using dynamic formulas. This includes the alert ID, the recipients, the notification message and the notification title. For example, if you will be processing by department, then you definitely want to include the department number for the current pass in the ID, title, and notification, and you probably

want to return different user or role names depending on the department. You can use GetCurrentValue () in these formulas.

If you specify the current file as the supporting document—the [NavigateTo] location—the multipass context will be automatically applied when the user opens the file from the alert notification. For example, if the alert was generated for the Dept 4200 pass, then when the user opens the file from that notification (whether from the task pane or the email link), the file will automatically be filtered for Dept 4200, and GetCurrentValue functions will return values as if it were currently processing that department. This allows the user to see the file using the same context as when the alert was generated.

**NOTE:** The filter will only be applied when the file is refreshed. You should configure the relevant Axiom queries to refresh on open if you want the user to see the data as soon as they open the file, without needing to click the Refresh button.

These features only apply when the alerts are processed using File Processing. If you process alerts in any other way, then GetCurrentValue will not return values during processing, and no filters will be applied when the file is opened from the notification. Generally, if you set up alerts to leverage File Processing features, then those alerts should always be processed using File Processing and not using any other alert processing option.

## Alert properties

Each alert defined on the Alert Control Sheet has the following alert properties.

The Alert Control Sheet is only visible to administrators or to users with both read/write permission to the file and **Allow Unprotect** permission. Otherwise, it is hidden by default.

**NOTE:** If you plan to process alerts using File Processing, then you can use GetCurrentValue to dynamically incorporate the current pass into certain alert settings, such as the alert ID, or the notification title.

| Item<br>[Column Tag] | Description |
|---|---|
| Alert Tag<br>[Alert] | The alert tag for the alert. Each alert defined in the sheet must be identified by an `[Alert]` tag.<br>You can also use this tag to specify whether a particular alert is "active" or not. For example, you can set up the tag using a formula so that the cell either displays `[Alert]` or is blank based on some condition. This allows you to temporarily "disable" some alerts and exclude them from processing without needing to delete the entire alert definition. |

| Item<br>[Column Tag] | Description |
| --- | --- |
| Alert ID<br>[ID] | The identifying code for the alert. This code is used to determine whether an alert is considered "new" or "existing" at the time of processing. If an existing alert in the database has the same code, then the alert is ignored even if the condition is True.<br>In most cases, you will want to set up a dynamic alert ID that incorporates the desired alerting "interval." For example, if the file will be processed daily but you only want users to be alerted of a particular condition once per week at most, then the ID should incorporate some kind of indicator of the current week. You may also want to incorporate the document ID into the alert ID, so that there is no chance of duplicating alert IDs generated from another file.<br>For more information about defining the alert ID and determining the alerting interval, see Controlling the frequency of alert notifications.<br>The alert ID is only used for alert management purposes; it does not display to the user. The ID is limited to 255 characters. |
| Condition<br>[Condition] | The condition to be evaluated when the alert is processed. This condition must evaluate to True or False.<br>You can define any condition using an IF statement. For example, if you want to alert users when a particular variance exceeds a certain threshold, you can set up the file to query the necessary data and then calculate the variance, and then create an IF function that returns True when the variance exceeds the threshold, and False when it does not. For example:<br>`=IF(Report!G23 >= 0.1, "True","False")` |
| Severity<br>[Severity] | The severity of the alert. When an alert notification is displayed in the Notifications task pane (or the Notifications panel in the Web Client), an icon indicates the severity.<br>• `Info`  <br>• `Warning`  <br>• `Error` <br>The severity is provided to give the user a visual indication of the importance of the alert. The example icons shown above are from the Desktop Client. The Web Client uses similar icons but with different styling. |

| Item<br>[Column Tag] | Description |
|---|---|
| Alert Recipients<br>[Recipients] | The Axiom Software users who should be notified if the alert condition resolves to True.<br>You can specify one or more user login names or role names. Separate multiple names with semicolons.<br>Role names must be prefixed with `role:` (for example, role:Finance). If a role is specified, the alert notification is displayed to all users who belong to the role at the time the alert was generated. |
| Deliver To<br>[Delivery] | Optional. Specify how the alert notification should be delivered:<br>• `Email`: Deliver the alert notification via email, using the email address defined for the recipients within Axiom security.<br>• `TaskPane` (default): Deliver the alert notification via the Notifications task pane. These notifications will also display in the Notifications panel of the Web Client container, for web users.<br>If left blank, the notification will be delivered via the Notifications task pane.<br>If you want the notification to be delivered via email *and* the Notifications task pane, enter both options separated by a semicolon like so: `Email;TaskPane` |
| Supporting File<br>[NavigateTo] | Optional. Specify a supporting file that a user can open to get more information about the alert condition. See the discussion following this table for more information on how to specify the file.<br>If this setting is left blank, then the alert notification will not contain any link to open a file (whether the notification is displayed in the task pane or delivered via email). |
| View As Form<br>[ViewAsForm] | Optional. Specify whether the supporting file should be opened as an Axiom form in the Desktop Client, if the supporting file is form-enabled.<br>If True, the file will be opened as a form; if False, the source spreadsheet file will be opened. If nothing is specified (either the column is blank or the column is not present) then True is assumed as the default.<br>This setting is ignored if the supporting file is not form-enabled. It is also ignored if the link is opened from the Notifications panel in the Web Client. From the Web Client, if the file is form-enabled then it is always opened as a form (unless a cell address is specified in the Supporting File property, in which case the source spreadsheet file will be opened so that the designated cell can be shown). |

| Item [Column Tag] | Description |
|---|---|
| Notification Title [Title] | The title of the alert notification. The title should be brief yet give the user an idea as to the content of the alert. For example: "Budget Variance Alert".<br>The title is limited to 255 characters. |
| Notification Message [Message] | The message for the alert notification. Ideally, the message should explain to the user what the alert condition is, why they are being notified, and what action (if any) the user should take as a result of the alert.<br>The message is limited to 2000 characters. |

### Specifying the supporting file for the alert

If you want the user to be able to open a supporting file that contains more information about the alert condition, you can specify either of the following:

- A location in the current file, such as `Report!G23`. When the user receives the alert, they can click a link to open the file and be taken to the specified location.
- A file path to a different file. When the user receives the alert, they can click a link to open the file and be taken to the specified location (if applicable).

  The full Axiom file path must be specified, such as: `\Axiom\Reports Library\Monthly Reports\Variance_Report.xlsx`. If desired, you can specify a location in the file using the following syntax:

  *FilePath;Location*

  For example: `\Axiom\Reports Library\Monthly Reports\Variance_Report.xlsx;Report!D25`

  **TIP:** You can right-click a file in Axiom Explorer or the Explorer task pane and choose **Copy document path to clipboard**, then paste the path into the alert definition.

In both cases, the user must have at least read-only rights to the file in order to open it from the alert notification.

**NOTES:**

- If the supporting file references a location in the current file and you process alerts using File Processing, then when the user clicks the link in the notification to open the file, the multipass filter will automatically be applied to the file. This allows the user to view the file using the same context that generated the alert. Axiom queries should be set to refresh on open in this case, so that the filter is applied when the file is opened. Note that this behavior does not apply if the link is opened from the Notifications panel in the Web Client.
- If the supporting document is opened as an Axiom form in the Desktop Client, the cell location within the file does not apply and will be ignored if specified.

## Design considerations for sending alert notifications via email

### Email content

If you decide to send the alert notification via email, you should be aware of how the notification message from the alert definition will display in the email. The email body is structured as follows:

> The following alert was generated in Axiom Software:
>
> `<Message text from the alert definition>`
>
> More information is available at `<CellLocation>` in `<SupportingFileName>`:
> `<Hyperlink to Supporting File>`

You can see an example default email here.

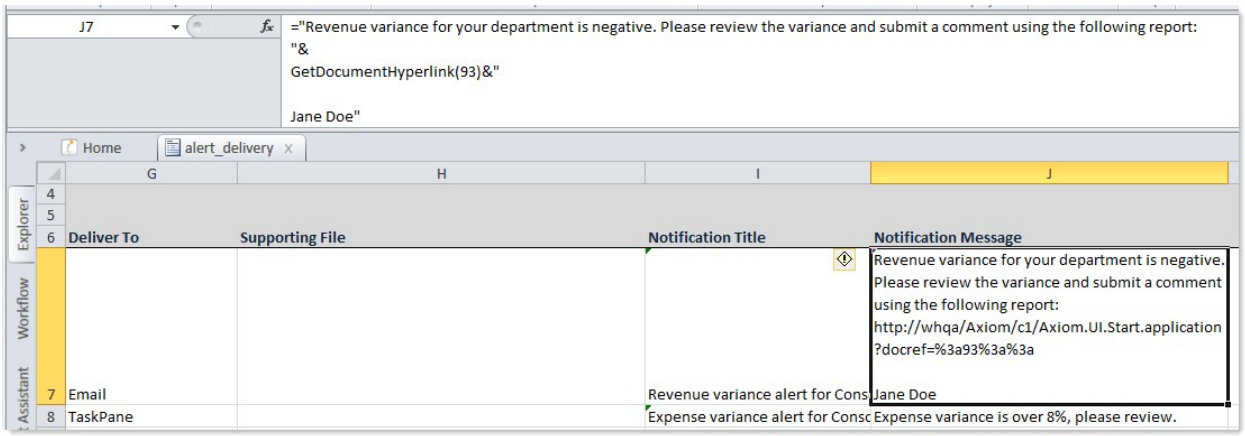
The beginning line that starts "The following alert..." is always included. This line cannot be removed.

The "More information..." hyperlink is only included if a supporting file is indicated in the alert definition, in the [NavigateTo] column. If this field is left blank for a particular alert, then no automatically generated hyperlink will be included in the email. However, you can include a manually generated hyperlink in the email body by placing it within the defined message text (see next section for an example).

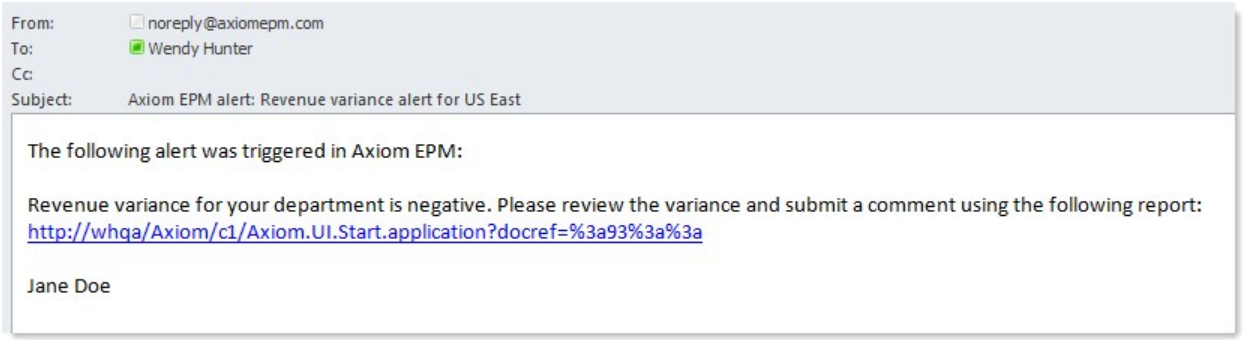
### Line breaks

When defining the body text for the alert notification, you can create line breaks in the cell by pressing ALT+ENTER. For example, you may want to define body text with two separate paragraphs, or with a separate line for a signature.

The following screenshot shows an example alert definition. The Notification Message cell has several line breaks created using ALT+ENTER. Line breaks are used to place the hyperlink and the signature on new lines. In this example the text is wrapped in a formula so that a hyperlink can be manually inserted within the body text using GetDocumentHyperlink (instead of specifying a supporting file in the alert definition).




When alerts are processed and a notification email is sent, the email will look similar to the following:




**NOTE:** Placing a hyperlink in the message text will only work when the alert notification is delivered via email. In the Notifications task pane, the hyperlink text will not resolve to a clickable hyperlink.

# Processing alerts

Once alerts have been defined within a file, they can be processed in several different ways, depending on the type of file. When you process alerts, each alert condition in the file is evaluated to be True or False. If True, then an alert notification is delivered to the designed recipients using email and/or the Notifications task pane.

**NOTE:** At the time of processing, the alert ID is used to determine whether a particular alert is "new" or "existing." If the ID matches an existing alert ID in the database, then no action is taken for that alert. For more information on this process, see Controlling the frequency of alert notifications.

Alerts can be processed in the following ways:

- **File Processing**: You can set up alerts in a file and then use File Processing to process the alerts using multipass processing.
- **Process Document List**: You can set up alerts in a report file (or a driver file) and then use the Process Document List task in Scheduler to process the alerts.
- **Process Plan Files**: If alerts are defined within plan files, these alerts are processed automatically when the file is processed using Process Plan Files.
- **Process Alerts**: You can process alerts directly within the file, by using the **Process Alerts** action.
- **Save**: You can configure a file so that alerts are processed automatically as part of a save.

You should perform alert processing as frequently as you need the alert condition to be evaluated. For example, maybe you are checking budget to actual variances for the current period, and you only need to check the condition once per month, after the actuals data for the current period has been imported into the system. It is not necessary to process these alerts more frequently, because the data will not change over the course of the month.

On the other hand, maybe you are monitoring budget data for the current planning cycle, while users are actively working in their budgets. In this case you want to process alerts more frequently, because the data is continually changing, and you want to alert users of a problem condition in their budgets as soon as possible. You might choose to process these alerts weekly, daily, or even hourly.

## Processing alerts using File Processing

File Processing supports a processing type of **Alerts**. The only action performed by this processing type is to refresh the file using the current pass filter, and then process the alerts defined on the Alert Control Sheet.

You can process alerts using File Processing manually within the file, or schedule it using the File Processing task for Scheduler.

The advantage of using File Processing to process alerts is that you can define an alert once and then process it multiple times at different levels using multipass processing. For example, imagine that you want to alert department managers if their department exceeds a variance threshold for a particular area. You could set up a report that brings in data for each department and calculates the variance, and then define an alert for each individual department value. However, it would be much more streamlined to use file processing to process the report by department. In this case you would set up a report that brings in the consolidated data and then define one dynamic alert against that value. Then you would set up the report for multipass processing by department. For each pass, it will automatically apply the department-specific filter for the data and then process the alert against that data. The alert settings would need to dynamically change for each pass—for example you could set up the alert ID, message title, etc., using GetCurrentValue() so that the current department number is reflected for each pass of alerts.

When an alert is generated using multipass File Processing, the current pass information is stored within the alert notification. If the supporting file for the alert—the `[NavigateTo]` location—is set to the

current file, then the multipass context will be automatically applied when the user opens the file from the alert notification. For example, if the alert was generated for the Dept 4200 pass, then when the user opens the file from that alert notification, the file will automatically be filtered for Dept 4200, and GetCurrentValue functions will return values as if it were currently processing that department. This allows the user to see the file using the same context as when the alert was generated.

**NOTES:**

- The multipass filter will only be applied when the file is refreshed. You should configure the relevant Axiom queries to refresh on open if you want the user to see the data as soon as they open the file, without needing to click the Refresh button.
- This multipass behavior does not apply if the link is opened from the Notifications panel in the Web Client. It only applies when the link is opened from the Notifications task pane in the Desktop Client, or from the email notification.

### Processing alerts using Process Document List

The Process Document List task in Scheduler allows you to schedule processing of a report file or a driver file. To process alerts as part of this task, select the **Process alerts in selected workbooks** option. All formulas in the file will be recalculated before alerts are processed.

You can configure the remaining options as desired. For example, if your alerts are dependent on an Axiom query within the file, then you should select **Perform all enabled Axiom Queries in selected workbooks**. If you want to perform a save-to-database and/or save the file after alerts are processed, you can enable those options as well.

### Processing alerts using Process Plan Files

If alerts are defined within a plan file, then those alerts will be processed any time the file is processed using Process Plan Files, whether it is run manually from the ribbon menu or by using the Scheduler task. There is no way to disable this processing. Alerts are processed after formulas are recalculated and after any Axiom queries are run, based on the Process Plan Files configuration.

### Processing alerts using Process Alerts

You can process alerts directly within the file, without using file processing:

- On the **Axiom** tab, in the **Advanced** group, click **Process Alerts**.

This approach is primarily intended as an alert testing tool, and not as a regular method of processing alerts. When you first set up alerts within a file, you should use this feature to test the alerts by sending them to yourself or to a designated testing group. After confirming that the alerts work as expected, you should finalize the alert setup and then process them using the primary means listed above.

When you use Process Alerts, the file is recalculated but Axiom queries are not refreshed. If your conditions depend on data from an Axiom query, you may want to perform a refresh before processing the alerts.

This option is only available to administrators and to users with the following permissions to the file: Read/Write and Unprotect.

### Processing alerts on save

You can configure a file so that alerts are processed automatically when a save is performed: either a save-to-database, or saving the document. For example, you might have alerts set up in a plan file to track certain conditions in the plan file, and you want those alerts to be processed automatically on save.

To enable alert processing on save, use *one* of the following options on the default Control Sheet for the file, in the **Workbook Options** section:

- **Process alerts on save data**
- **Process alerts on save document**

You can choose one behavior or the other, but not both—if both are enabled then only the save-to-database processing will occur. Once you have decided which behavior you want to enable, you can select one of the following options for that behavior:

- **Process**: When alerts are processed, the normal alert processing results dialog is shown. This is typically only used when the alert author is testing the alert setup for the file.
- **ProcessSilently**: Alerts are processed silently in the background; the user performing the save will not be aware of the alert processing (unless, of course, the user receives an alert as a result of the processing). This is the intended setting for production-ready files.

If you want to use save processing in a form-enabled file, then you must use the "save data" option, because the file is not saved when in the Axiom forms environment. Also, in the Axiom forms environment both **Process** and **ProcessSilently** will behave in the same way (processing silently), because Axiom forms do not support the ability to display the alert processing results dialog.

# Controlling the frequency of alert notifications

When setting up alerts, it is important to consider how often you want a user to be notified about a particular condition.

The primary determiner of alert frequency is, of course, how often the alerts are processed. If a particular file is processed for alerts just once per month, then at most users can receive this alert once per month. However, processing frequency is only part of the equation—you also need to consider whether a user may have already been alerted about a particular condition, and whether they should be alerted again at this particular processing interval.

Imagine a scenario where a file is processed for alerts once per day or even per hour. For example, users may be actively working on budgets, and you have certain alert conditions that you want to monitor for the budget data. Because users may be saving their budgets throughout the day and changing the budget data, the alerts need to be processed frequently. So imagine that an alert condition resolves to

True when this file is processed at 2:00 AM on Monday. Do you want users to receive the same alert again if this condition is still True when the file is processed again the next day, or the next hour?

The answer may vary depending on the type of conditions you are monitoring with alerts, and your alerting preferences. You may want users to be constantly alerted until they correct the issue, even that means they are receiving an alert daily or even hourly. On the other hand, you might prefer that users be alerted of a particular condition once per week at most. In this case you can think of the alerting "interval" as weekly, even though the processing frequency might be daily or hourly.

To control the alerting interval, you can use the alert ID to determine whether an alert is considered "new" or "existing" when it is processed. New alerts will be added to the database and displayed to the designated recipients. Existing alerts will be ignored.

Alert processing is handled as follows:

- When alerts are processed, Axiom Software evaluates all active alert conditions found on the Alert Control Sheet of the file. If a condition evaluates to False, then no action is taken. If the condition evaluates to True, then the alert is eligible for notification.
- Axiom Software then compares all newly eligible alerts to existing alerts in the database, using the alert ID (as defined on the Alert Control Sheet) as the comparison point. If the ID for a newly eligible alert matches the ID for an existing alert in the database, then no action is taken for that alert. If no ID match is found, then the new alert is added to the database and the alert notification is delivered to the designated alert recipients.
- Alerts remain in the Axiom Software database until they are purged using the Scheduler System Data Purge task. This task purges alerts older than a specified number of days (by default, 60 days). It is important to understand that if a user deletes an alert notification from their Notification task pane, this does *not* delete the associated alert from the database; it simply removes the notification from the task pane.

For example, if you want an alert to be sent every time alerts are processed (assuming the condition evaluates to True), then you could set up the alert ID so that it uses the current date/time stamp as part of the ID (Variance_11202012_0930). If you want the alert to be sent once per month at most, then you could set up the alert ID to use the current month as part of the ID (Variance_June2012).

Keep in mind that the alert ID is compared against all existing alerts in the database, regardless of their source. If another file also has alerts set up with an ID format of Variance_MonthYear, it is possible that an alert ID from one file would match an existing alert ID generated from another file. If this is a concern, then you may want to set up your alert IDs to incorporate the current document ID, so that alerts are unique per document.

### Deleting alerts using the System Data Purge task

As discussed above, alerts remain in the system until they are purged by the System Data Purge task in Scheduler. It is important to understand that this functionality is primarily intended as a database management tool to purge old data from the database, and not as a means of alert control.

In theory, if you know that the System Data Purge task is configured to purge all alerts older than 15 days, and you have a file that you want to process for alerts on a monthly basis, then you would not need to worry about making the alert ID dynamic per month because there should not be any alerts left over in the database from the prior month. However, it is best to use the dynamic alert ID method so that you can explicitly control the alerting interval. Keep in mind that the purge task is limited to deleting a specific number of records each time it is run, so depending on how often your purge task is run and how much old data is present in the database, old alerts might not get purged until later than expected.

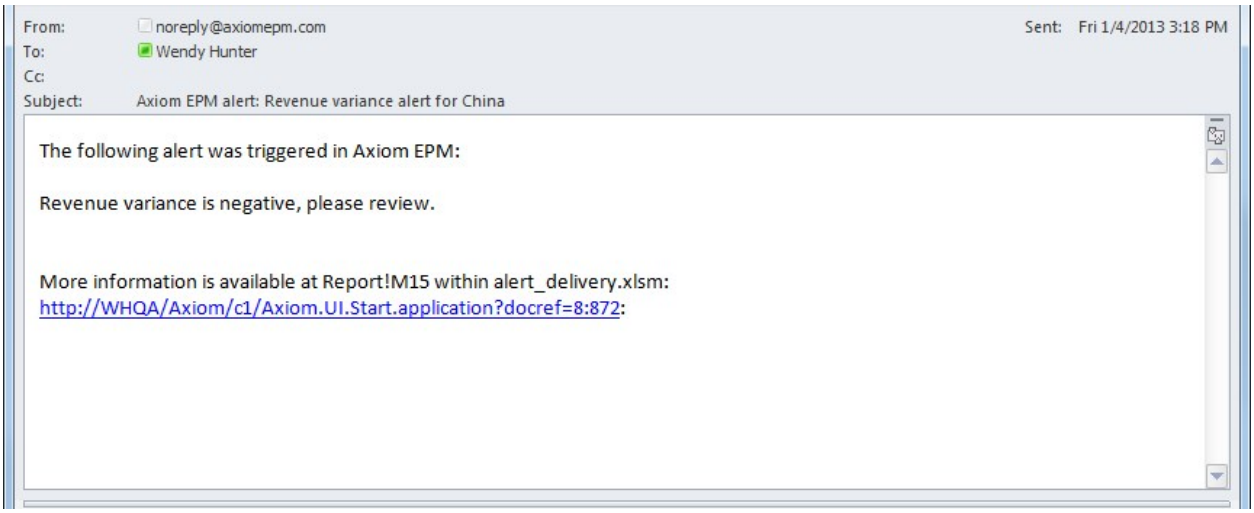
Also, keep in mind the user experience when configuring the purge task. When alerts are purged from the database, they will disappear from the user's Notification task pane with no explanation (assuming that is how the alerts were delivered). Ideally, the purge time frame should be set so that most users will have already deleted the alert from their task pane by the time the alert record is purged from the database. Setting a shorter time frame may cause user confusion.

# Viewing alert notifications

When a new alert is added to the database as a result of alert processing, an alert notification is delivered to all designated alert recipients, using email and/or the Notifications task pane.

## Email notification

If the alert notification is delivered via email, the subject of the email is "Axiom Software alert: *<Alert Title>*". The alert message is placed within the body text. The email also contains a hyperlink to the supporting file if one is specified in the alert definition.

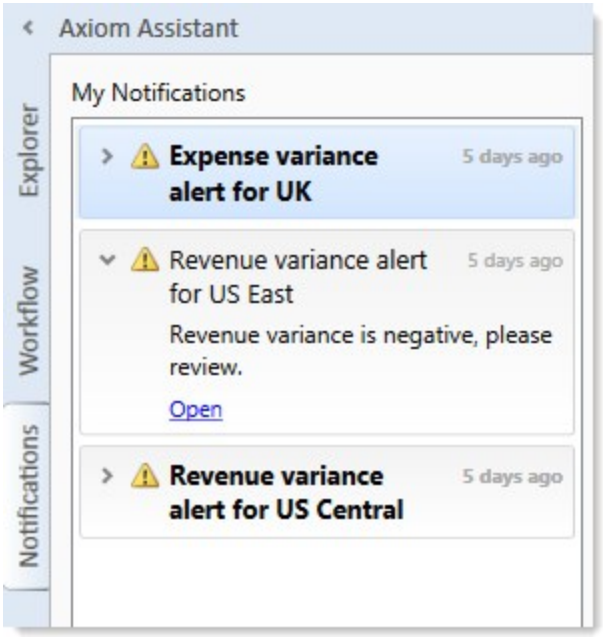



*Example email alert notification*

## Notifications task pane

Alert notifications are displayed in the Notifications task pane in the order they are received, with the most recent alerts at the top.

By default, notifications are collapsed so that only the severity icon and the alert title are visible. Users can expand the notification to read the alert message and to navigate to the supporting document for the alert, if defined.




*Example Notifications task pane*

If a user receives a new alert notification during the current session or if unread alerts are present when the user logs in, the Notifications tab will flash orange and will remain orange until the tab is clicked. New notifications display in bold text until they are read.

Using the right-click menu, users can mark notifications as read (or unread), and can delete notifications from the task pane. If a user has rights to access the source file for an alert, they can also navigate to the alert definition in that file.

Notifications do not have an expiration date. A notification will continue to display in the user's Notifications task pane until the user deletes it, or until the notification record is purged from the database using the System Data Purge task in Scheduler. Keep in mind that if a user deletes a notification, this simply removes the notification from the user's task pane, it does not delete the notification record from the database.

The Notifications task pane is system-controlled. Users cannot manually open and close it, and it is not available for customization. The task pane behaves as follows:

- The task pane displays on startup if the user has an active notification (read or unread). If the task pane contains unread notifications, the task pane is active and the tab flashes. The task pane continues to display for the duration of the current session, even if the user deletes all active notifications from the task pane.

- The task pane does not display on startup if the user has no active notifications. If the user receives a new notification mid-session, the task pane automatically opens, becomes active, and the tab flashes.

**NOTE:** In the Web Client, users can view alert notifications using a similar approach, in the Notifications panel of the Web Client container.

### Alternative alert viewing methods

In addition to the Notifications task pane, you can display alert notifications to users by querying the Axiom.Notifications system table. This is intended for environments where you need custom display of notifications.

# Updating alert notification status using Save Type 4

Using Save Type 4, you can update alert notification status (read or not read) by using save-to-database within a spreadsheet. This is intended to be used in cases where alert notifications are being viewed using a custom solution, rather than by using the built-in Notifications task pane (or the Notifications panel in the Web Client container).

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the notification properties.
- Row tags in the save-to-database control column, to flag rows to be saved.

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB; Axiom.Notifications; CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]` |
| Column tags | `NotificationID`<br>`IsRead` |

**NOTE:** Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.

### Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.Notifications]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.Notifications;CustomSaveTag=MarkRead]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the alert notification properties in the control row

Within the control row for the save-to-database process, specify the columns that define the alert notification properties. These properties can be placed in any column.

| Column Tag | Description |
|---|---|
| `NotificationID` | The database ID of the alert notification. This is different than the AlertID, which is defined in the alert properties. |
| `IsRead` | The read status of the alert notification, either True or False. |

The column tags can be placed to either the right or the left of the SaveStructure2DB tag. Both columns are required. Only the IsRead status can be changed; the NotificationID is to identify the notification to modify.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

**NOTE:** These are the only columns of the Axiom.Notifications table that are supported for use with Save Type 4. Any other columns listed in the save-to-database row will be ignored by the save process.

### Flagging the rows to be saved

Within the control column for the save-to-database process, mark each row that you want to be saved with a `[Save]` tag. This is the only valid action for the table save.

If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.Notifications; CustomSaveTag=MarkRead]` then you would place the tag `[MarkRead]` in the rows that you wanted to be saved.

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

**NOTE:** The row tag can be placed within a formula if desired.

### Populating the table properties in the spreadsheet

In the property columns, enter the alert notification properties for each notification that you want to update. It is assumed that this will be done by using an Axiom query or GetData functions to the Axiom.Notifications table to return the relevant properties.

For example, imagine that you want to query in the current alert notifications for a user, and then display those notifications in an Axiom form. The user can then check a box to show they have read the notification. The read notifications are then hidden in subsequent refreshes and the next time the file is opened. You would:

- Create an Axiom query to the Axiom.Notifications table to bring in the notifications for the current user. This filtering happens automatically when the table is queried; you don't need to define a data filter to restrict the results to the current user.
- Configure Save Type 4 to save back the current IsRead status for the notifications, and set up the form to support saving to the database.
- Configure a formatted grid to display the results of the query. The formatted grid could also contain a check box that would toggle the IsRead value from False to True.
- Omit notifications that have been read by either filtering the Axiom query (IsRead=False), or by configuring the [Row] tags for the formatted grid to be dynamic based on the IsRead status.

The Axiom form user can select the check box in the grid to indicate that they have read a notification. The grid could be set to auto-submit and save on submit to immediately update once a user has made this selection, or you could trigger the update and save using a Button component.


# File Management

This section contains information about file management features in Axiom Software. All managed files are stored in the Axiom Software database.

## Managed and non-managed files

Axiom *managed files* are files that are stored in the Axiom Software database. By default, Axiom Software stores all Axiom files in the database—plan files, templates, reports, and driver files—as well as other supporting files such as the default Start file and calc method libraries.

Access to managed files is controlled by Axiom Software security. Managed files are also available to Axiom Software utilities, such as the **Process Plan Files** utility.

A file does not have to be managed in order to use certain Axiom Software functionality. For example, you can use Axiom functions in regular Excel spreadsheets, and they will return data as long as the file is opened within Axiom Software. If the non-managed file has a Control Sheet, you can also use Axiom file functionality such as Axiom queries.

You can use non-managed files to *query* data from Axiom Software, but you cannot use non-managed files to *save* data to the database. If you were to save data to the database from a non-managed file, then there would be no record of how the data in the database was calculated. Therefore, saving data from non-managed files is not supported.

All files related to file groups must be managed files. Report files can be managed or non-managed; however, we recommend keeping all reports as managed files so that Axiom Software security can be applied.

### ▶ Saving a managed file as a non-managed file

You can save a managed file locally (to your computer or a network drive) by clicking the down arrow on the **Save** button and then clicking **Save As (Local File)**. The file is then considered a non-managed file.

The most common reason to do this is to move files between Axiom Software systems. You can save the managed file locally, and then import it into another system.

### Saving a non-managed file as a managed file

You can save any file to the Reports Library and make it a managed file. If the file is an Excel file, you can open the file in Axiom Software, then click the down arrow on the **Save** button and click **Save As (Repository)**. If the file is not an Excel file, you must use Axiom Explorer to import the file to the Reports Library.

Any file that is saved in the Reports Library can be opened from the **Reports** menu. If the file is not a file type that can be opened within Excel (such as a Word file), the appropriate program is launched (if installed on the computer).

You can save files to other folders in the Axiom Software file system using Axiom Explorer, but these folders are more structured. For example, the **File Groups** folder should only contain valid file group files. Imagine that you have an existing Excel file that already has all of the information that you want to use for a driver sheet. You could take that file, add a Control Sheet, configure it as appropriate, then save it to a **Drivers** folder in Axiom Explorer.

Only advanced users should save files to locations other than the Reports Library.

# Managing files using Axiom Explorer

Using Axiom Explorer, administrators can view and manage the files in the Axiom Software file system. You can access files, import and export files, delete files, view file history, and manage folder structures for various Axiom libraries.

Axiom system files and *managed files* are stored in the Axiom Software database instead of a Windows file system, and therefore are not accessible by applications such as Windows File Explorer. Axiom Explorer must be used to manage these files.

In addition to basic file management, other administration features such as file group administration and table administration are performed in Axiom Explorer. For more information on performing these activities, see the documentation for each particular feature. This topic discusses file management features only.

**To access Axiom Explorer:**

- On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**.

**NOTE:** For information on each folder in the Axiom Software file system and what it contains, see Axiom Software Help: **Reference > Axiom Software file system**.

Axiom Software provides a streamlined version of the Axiom Explorer dialog using the built-in Explorer task pane. Most file management features that are available in Axiom Explorer are also available in this task pane.

## Access to Axiom Explorer

By default, only administrators have access to Axiom Explorer. Non-administrators can be granted access by enabling the **Administer Axiom Explorer** permission.

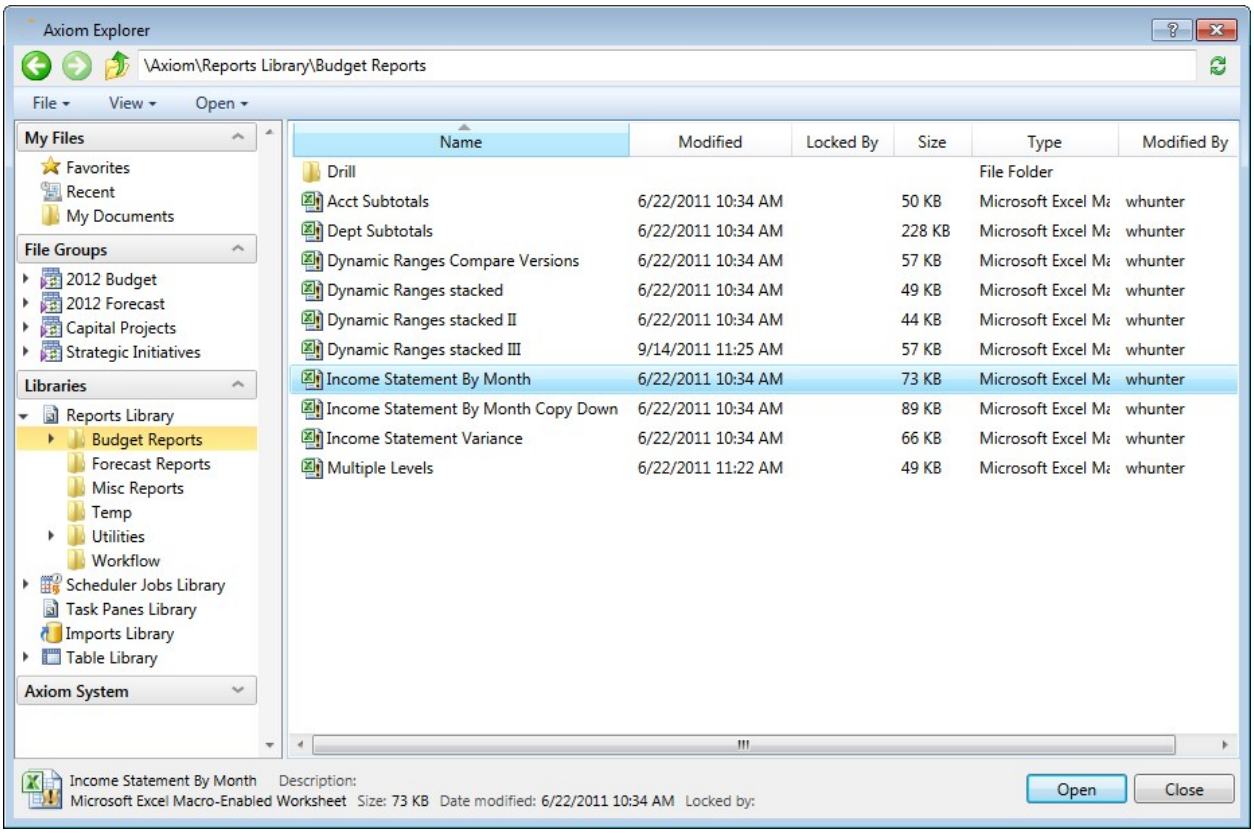
If a non-administrator has access to Axiom Explorer, what they can see in the dialog and what actions they can perform are controlled by their other relevant security permissions. For example, if a user does not have any access to imports, then they will not see the Imports Library. If a user has read-only access to a folder in the Reports Library, then they will see that folder and can open reports, but they will not have access to commands such as creating new folders and deleting files.

**NOTE:** The **Axiom System** section contains system files for Axiom Software and therefore is only available to administrators. Other users cannot be granted access to this section.

The Explorer task pane is a streamlined version of Axiom Explorer. Essentially, the task pane displays the treeview from the left-hand pane of Axiom Explorer, which allows users to open files and perform actions available on the right-click menu. The same rules apply to the Explorer task pane in determining what files and features a user has rights to access within the treeview. However, by default, all users have rights to the Explorer task pane, via Startup settings defined on the Everyone role (these settings can be changed if desired, or the task pane disabled altogether). The **Administer Axiom Explorer** permission is *not* required for non-administrator users to see the Explorer task pane.

## Axiom Explorer overview

In the Axiom Explorer dialog, the virtual folder directory is displayed in the left-hand pane, and the sub-folders and files within a selected folder are listed in the right-hand pane.




*Example Axiom Explorer*

When you select a folder or file, information about that item displays in the bottom of the dialog, such as the file type and size, the date the file was last modified, and the file or folder description. If the file is currently in use, the user who has locked the file is also listed.

As you navigate through the directory, the current location is displayed in a box at the top of the window. You can also type a location into the box to jump directly to that location.

To refresh the file list, click the **Refresh** button in the toolbar.

## File groups and Axiom Explorer

File groups have special treatment in Axiom Explorer. Only administrators can access all of the underlying folders and files for a file group.

If a non-administrator has access to Axiom Explorer, that user will not see the Plan Files folder or any plan files, even if the user otherwise has access to the plan files. Non-administrators must use Open Plan Files (or similar related software features) to open their plan files. The Plan File Attachments folder is also hidden.

Non-administrator users can see the following folders only if they have been granted access to the folders (or files within them) on the **Files** tab of the **Security Management** dialog: Drivers, Templates, Utilities, Process Definitions.

## Managing folders

You can perform the following actions by right-clicking a folder. You must have read/write permissions to the folder (as defined in Security) in order to perform any of these actions.

| Folder Action | Description |
|---|---|
| New > Folder | Create a new sub-folder in the selected folder. Once the folder has been created, use **Rename** to name it. You can also define a description for the folder.<br><br>Note that the Reports Library on the Reports menu only displays folder structures up to four levels deep. Folder structures greater than four levels can only be accessed using **Reports > All Reports** (or Axiom Explorer).<br><br>You must have read/write access to the parent folder in order to create a sub-folder.<br><br>**NOTE:** Not all folders in the Axiom file system support creating sub-folders. Generally speaking, if access to the folder is controlled using the Files tab in Security, then users with read/write access can create a sub-folder there. |
| Rename | Rename the selected folder. You can also press the F2 key to rename a selected folder.<br><br>You must have read/write access to the folder in order to rename it.<br><br>**NOTE:** Not all folders in the Axiom file system support being renamed. If the folder is system-controlled and Axiom Software expects the folder to have a particular name, then this option will not be available. |

| Folder Action | Description |
|---|---|
| Import Files | Import files into the selected folder. The imported files will be saved in the Axiom Software database. You must have read/write access to the folder in order to import files.<br><br>**NOTE:** If you import a file with the same name as an existing file in the folder, that file will replace the existing file and retain its document ID.<br><br>Certain folders have additional file type and user permission requirements. For example:<br>• Folders that are dedicated to specialty Axiom file types (such as AXL for task pane files) can only accept imported files of that type.<br>• Report folders can accept imported files of virtually any file type.<br>• Plan file folders can only accept imported XLSX or XLSM files that have the same name as an existing plan file in the folder.<br>• If you have an XLS file that you want to import into Axiom Software, it is recommended to convert it to XLSX or XLSM first. XLS files are not fully supported by Axiom Software. |
| Delete | Delete the selected folder. This option is only available for user-created folders.<br><br>A folder cannot be deleted if it contains any files or sub-folders. You must first delete or move the files within the folder before the **Delete** command becomes available on the menu.<br><br>You must have read/write access to the folder in order to delete it. |

You can move folders by dragging and dropping. Only one folder can be moved at a time—drag and drop does not support multiple selections. Folders can only be moved within the Reports Library and the Scheduler Jobs Library. Other folders cannot be moved to new locations.

## Managing files

You can perform the following actions by right-clicking a file:

| File Action | Description |
|---|---|
| Open | Open the selected file. You can select multiple files and open them simultaneously. |
| Open (as other user) | Open the selected file, using the security permissions of a different user. This option is only available to administrators and only for Data Explorer files, so that administrators can test the file setup. A dialog box allows you to select the user to view the file as. |
| Open Read Only | Open the selected file as read-only. |

| File Action | Description |
| --- | --- |
| Open Without Refresh | Open the selected file without running any Axiom queries that are set to **Refresh on Open**. This option also ignores the **Downgrade to read-only on open** setting.<br><br>This option is only available to administrators. |
| Open in Browser | Open the selected file in a browser. This is only available for form-enabled files. |
| Open in Browser (as other user) | Open the selected file in a browser, using the security permissions of a different user. This option is only available to administrators and only for form-enabled files, so that administrators can test the file setup. A dialog box allows you to select the user to view the file as. |
| Break Selected Locks | Break the file lock for the selected file or files. This option is only available to administrators, and only if the selected files are locked by other users. |
| Document history | View the history for the selected document. For more information, see Viewing document history.<br><br>You must have read/write access to the file in order to view its history. |
| Cut<br><br>Copy | Cut or copy the selected file, to be pasted to a location in the Axiom Explorer. After cutting or copying, navigate to the location where you want to paste the file. Right-click in the blank area of the file list (the right-hand pane) and then click **Paste**.<br><br>You can select multiple files and copy them simultaneously.<br><br>You must have read/write access to the file's folder in order to use cut. |
| Rename | Rename the selected file. You must have read/write access to the file's folder in order to rename a file. |
| Delete | Delete the selected file. You must have read/write access to the file's folder in order to delete a file. |
| Export | Exports a copy of the selected file to your local computer or a network file location. You can select multiple files and export them simultaneously.<br><br>When the export copy is created, workbook and worksheet protections are applied and "refresh on open" Axiom queries are run. The behavior is as if you had opened the document in Axiom Software and then used **Save As (Local)**.<br><br>If you are an administrator and you want to save a local copy of a file without these protections and data, then you must use **Export Without Refresh**. |

| File Action | Description |
|---|---|
| Export Without Refresh | Exports a copy of the document as it exists in the database. This option is only available to administrators.<br><br>When using this option, Axiom Software exports the file without performing the various "open document" protocols such as applying protection and running "refresh on open" Axiom queries. |
| Copy shortcut to clipboard | Creates a hyperlink to the selected file and copies it to the clipboard. You can then paste the hyperlink into a file or an email (or otherwise provide it to a user).<br><br>The structure of the hyperlink (and the behavior when using it) is the same as described for hyperlinks created by the GetDocumentHyperlink function. |
| Copy document path to clipboard | Copies the full path of the document to the clipboard. You can then paste the path somewhere—for example, for use with alert setup, or within certain Axiom functions that use a document path. |
| Add to Favorites | Add the selected file to your favorites list. |
| Package Details | View the package details for the file, including the feature the file belongs to, the package version, and the date the feature was applied.<br><br>This action is only available in the Axiom Explorer dialog, and only applies if the file belongs to a product package. If the file does not belong to a product package, then no package details will be found. |

You can move files by dragging and dropping. Only one file can be moved at a time—drag and drop does not support multiple selections.

## Defining file and folder descriptions

Each file and folder can have a description. Descriptions are displayed in the bottom of the Axiom Explorer window when a file or folder is selected, and in some cases display elsewhere in the system, such as when selecting a report to open.

**To create or edit a file or folder description:**

1. Select the file or folder in Axiom Explorer.
2. At the bottom of the dialog, in the file/folder properties section, click on the space to the right of the **Description** label. (When you hover your cursor over this area, the description box becomes outlined.)
3. Type the description into the box. If you want to clear a description, highlight the existing text and then press the Delete key.

# Importing and exporting files

You can import external files into the Axiom file system, and you can export Axiom files from the file system to a local folder or network share. This can be done using the Axiom Explorer dialog or the Explorer task pane.

## Importing files

To import an external file into a folder in the Axiom file system, you must be an administrator or have read/write access to the folder. This is the same level of permissions required to save a new file within a folder.

Most folders in Axiom Software are limited to holding files of a certain type. For example, the Scheduler Jobs Library can only hold Scheduler jobs, and the Imports Library can only hold import utilities. If you attempt to import a file into a folder and that file type is not allowed in that folder, the import will be prevented.

A few folders do not have file type restrictions. For example, the Reports Library can contain files such as Word files, PowerPoint files, video files, and image files in addition to Axiom reports. These files can be referenced throughout the system—such as a logo file that is referenced by various reports and forms.

**NOTES:**

- If you import a file that has the same name as a file that is already in the folder, then the imported file will replace the current file and retain the current document ID. This is important for cases where you want the imported file to be recognized as the "same" file, so that shortcuts and other references to the file remain intact.

  If you have read/write access to a particular Axiom file but not its folder, then you cannot use import to overwrite the file. Instead, you can open the external file within Axiom Software, and then use **Save As (Repository)** to overwrite the existing file.

- When you import a spreadsheet file into the Axiom file system, any passwords relating to workbook/worksheet protection will be removed from the file. This is to put the imported file in a state where it can be opened by any Axiom Software client. If the file is an Axiom file with protection enabled on the Control Sheet, that protection will be reapplied when the file is opened (using the system-applied password as appropriate).

  If Axiom Software is unable to remove the protection-related passwords from the file (for example, if there is a file-level password on the file), then the file cannot be imported.

**To import a file into the Axiom file system:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**.

   Importing files can also be done in the Explorer task pane.

2. Navigate to the folder where you want to import the file, then right-click and select **Import Files**.

3. In the **Select File to Import** dialog, select the file to import from your local drive or a network file share, and then click **Open**. You can also select multiple files to import.
4. If a file with the same name already exists in the folder, you are prompted to confirm that you want to overwrite the existing file. Click **Yes** if you want to overwrite the existing file, retaining the same document ID.

   **NOTE:** When overwriting an existing file, you must also have read/write access to that file.

The selected file is imported to the folder.

## Exporting files

If a file is visible to you in Axiom Explorer or the Explorer task pane, you can export it. The most common reason to export an Axiom file is to import it into another Axiom Software system.

Any file in the Axiom file system can be exported using the **Export** command. For Axiom spreadsheet files (such as reports), there are two export options:

- Non-administrator users can use **Export**, which exports the file in the same state that it would appear to the user in Axiom Software. This means that workbook/worksheet protection is applied to the exported file (if enabled on the Control Sheet), and "refresh on open" Axiom queries are run. Effectively, this is the same behavior as if the user had opened the file in Axiom Software, then used **Save As (Local)** to save a copy of the file outside of Axiom Software.
- Administrator users have access to a second option called **Export Without Refresh**. This can be used to export the file as it exists in the database—without applying protection, and without running "refresh on open" queries.

**To export a file from the Axiom file system:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**.

   Exporting files can also be done in the Explorer task pane.
2. Navigate to the file that you want to export, then right-click and select **Export** or **Export Without Refresh**.

   In the Axiom Explorer dialog only, you can select multiple items to export. Multiple selection is not available in the Explorer task pane.
3. In the **Export Repository File** dialog, navigate to the local folder or network file share where you want to save the exported copy, and then click **Save**.

The exported copy is saved to the designated location.

# Viewing document history

Axiom Software automatically creates document versions when a file is modified, and stores those versions in the database for a configured amount of time. These historical versions can be viewed using Axiom Explorer, and can be restored as the current version if needed.

New document versions are automatically created when a document is modified within a user's lockout session, once per lockout. For example, if a user opens a plan file as read/write and then saves that document, the new version becomes the current version of the document, and the prior version is "archived" in the audit database. Whether the user saves the document once or multiple times within the lockout session, only one version is created. However, if the user closes the document and then re-opens it, creating a new lockout session, another version will be created if the user saves the document.

Administrators have access to an additional means of creating document versions of Axiom spreadsheet files, using the **Create Restore Point** feature on the Axiom Designer tab. Using this feature saves the file and creates a new document version, within the same lockout session.

The available document history at any particular time depends on how your System Data Purge Scheduler job is configured. This task purges audit history as of a specified number of days. For example, if the task is configured to purge audit data older than 30 days, then you cannot view or restore any document versions older than 30 days.

**NOTE:** If you need to restore plan files, you can use the Restore Plan Files feature instead. This feature can be used to restore any number of plan files as of a specified point in time. For more information, see the File Group Administration Guide.

## Viewing available document versions

You can view the list of available document versions using Axiom Explorer, and open these versions.

**To view the history for a document:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Axiom Explorer**.

   You can also view document history using the Explorer task pane.

2. Navigate to the desired document, then right-click it and select **Document History**.

The **Document History** dialog opens, listing all of the available versions of the selected document (including the current version). The version numbering starts at 1 and increments by +1 for each subsequent version, so that the version with the greatest number is the most recent version.

If you want to view a prior document version, select the version and then do either of the following:

- Click **Open** to open the version in Axiom Software. This opens the selected version as an unmanaged copy of the document. The name of the unmanaged file is *`Filename - version #`*. (Note that if you open the current version of the document from the Document History dialog, it opens normally, just as if you had opened it from Axiom menus. The version behavior only

applies when opening a historical version.)

If you want to save a copy of this version, you must use **Save As** to save the file locally or back to the Axiom Software file system as a new name. If you just click **Save**, then the file is saved within the Axiom temporary folder and will not be retained.

- Click **Export** to export a copy of the selected version. This is equivalent to using **Export Without Refresh**—meaning, protection will not be applied, and "refresh on open" Axiom queries will not be run. You can save the exported copy to your local computer or a network file share.

This process assumes that you just want to view the document or save a copy of it, not replace the currently active version with it. If you want to restore a prior version over the currently active version, see the next section.

## Restoring document versions

The easiest way to restore a document version is by using the **Restore Prior Version** button on the Axiom Designer ribbon tab. By default this feature is only available to administrators, but it can be used by any user with read/write access to the document if it is included on a custom ribbon tab that they have access to. This feature is only available to Axiom spreadsheet files.

**To restore a prior document version:**

1. Open the document, then go to the **Axiom Designer** tab.
2. Click **Restore Prior Version** to view a drop-down list of available versions, then click the prior version that you want to restore.
3. Axiom Software prompts you to confirm that you want to open the selected prior version in place of the current version. Click **Yes** to continue.
4. If the currently open version of the file has unsaved changes, then Axiom Software prompts you to save those changes before the prior version is opened. Click **Yes** or **No** depending on whether those changes need to be saved.

   The current document is closed and the prior version of the document is opened as the current file.
5. To save this prior version as the current version, click **Save**.

If it is not possible to use Restore Prior Version, then you can:

- Open the prior version, then use **Save As** to save the file over the current version.

  OR
- Export the prior version, then use **Import Files** to import the file over the current version.

# Recovering deleted documents

If a document has been deleted in error, you can recover it. The recovery process restores the last active version of the file.

By default, deleted files are recovered to the folder location that they were deleted from. If this location no longer exists, then you can still recover the file, but it will be placed in the following location: `\Axiom\Reports Library\Recovered Files` (regardless of whether the recovered file is a report). You can then move the file to the desired location.

The ability to recover a deleted file is only available if the following conditions are met:

- The deleted file has not yet been purged from the audit database.

  This is determined by the configuration of your System Data Purge Scheduler job. This job purges audit history as of a specified number of days. For example, if the job is configured to purge audit data older than 15 days, that means that deleted files will be available for recovery for 15 days and then they will be purged from the database.

- Another file with the same name does not already exist in the recovery location.

  For example, imagine that the report `\Axiom\Reports Library\Budget Reports\IncomeStatement.xlsx` is deleted. Then, a new report is saved to the same location, and it is given the name `IncomeStatement.xlsx`. The deleted file cannot be recovered, because there is already a file with the same name in the recovery location. In this case you can rename or move the new file, which will allow the deleted file to be recovered.

When a deleted file is recovered, it retains the same document ID that it had before it was deleted. This means that any broken shortcuts to the deleted file should automatically work again once the deleted file is recovered.

Only administrators can recover deleted files.

**To recover a deleted file:**

1. On the **Axiom** tab, in the **Administration** group, click **Manage > Recover Deleted Files**.

   The **Recover Deleted Files** dialog opens. This dialog lists all of the deleted files that are currently saved in the audit database. Each file is listed by name, the date it was deleted, the user who deleted it, and the folder location it was deleted from.

2. Select the file that you want to recover, and then click **OK**. You can optionally use the filter box to find the file that you are looking for. The filter matches by document name.

3. You are prompted to confirm that you want to recover the selected file. Click **OK** to continue.

A confirmation message displays when the file has been recovered successfully. If the document could not be recovered, an error message will explain why.

**TIP:** If you do not see the recovered file in the file system, refresh the file system using **Reports > Refresh file system**.

# Managing locked items

Administrators can use **Manage Locked Items** to view a list of locked files and tables, and to remove locks if necessary. To access this dialog:

- On the **Axiom** tab, in the **Administration** group, click **Manage > Locked Items**.

The **Axiom Explorer** dialog opens, with the focus on the **Locks** area. This area lists all currently locked files and tables. The following information is provided:

- The name of the item
- Whether the item is a document or a table
- The user who locked the item
- The date the item was locked

**NOTE:** Only administrators, subsystem administrators, and users with the **Administer Locked Items** security permission can access this dialog and remove locks on items. Subsystem administrators are limited to managing locked items for users in their subsystem. Users with the Administer Locked Items security permission are limited to managing locked items that they have some level of access to.

## How do files and tables get locked?

Axiom Software stores files in the Axiom Software database instead of on a file server. When a user opens a file, a temporary copy is placed on the user's local hard drive, and the file in the database is locked to prevent other users from making changes to the file. If a different, non-admin user attempts to open the file, that user is informed that the file is locked and can open the file as read-only. The lock is removed when the locking user closes the file.

Tables become locked when a user opens the table using **Open Table in Spreadsheet**, and remain locked until the user closes the spreadsheet. (All other operations that modify tables lock the table immediately prior to the save, and release the lock immediately afterward, so this dialog does not apply.)

Situations may occur where a file or a table remains locked to a user unintentionally. For example, the user may have exited Axiom Software improperly or experienced a system crash, therefore preventing the lock from being removed as part of the normal application closing processes. Or, the user may have disconnected from an active Terminal Server session, and can no longer access that session. In these situations, administrators can remove the lock so that the file or table can be edited again by other users.

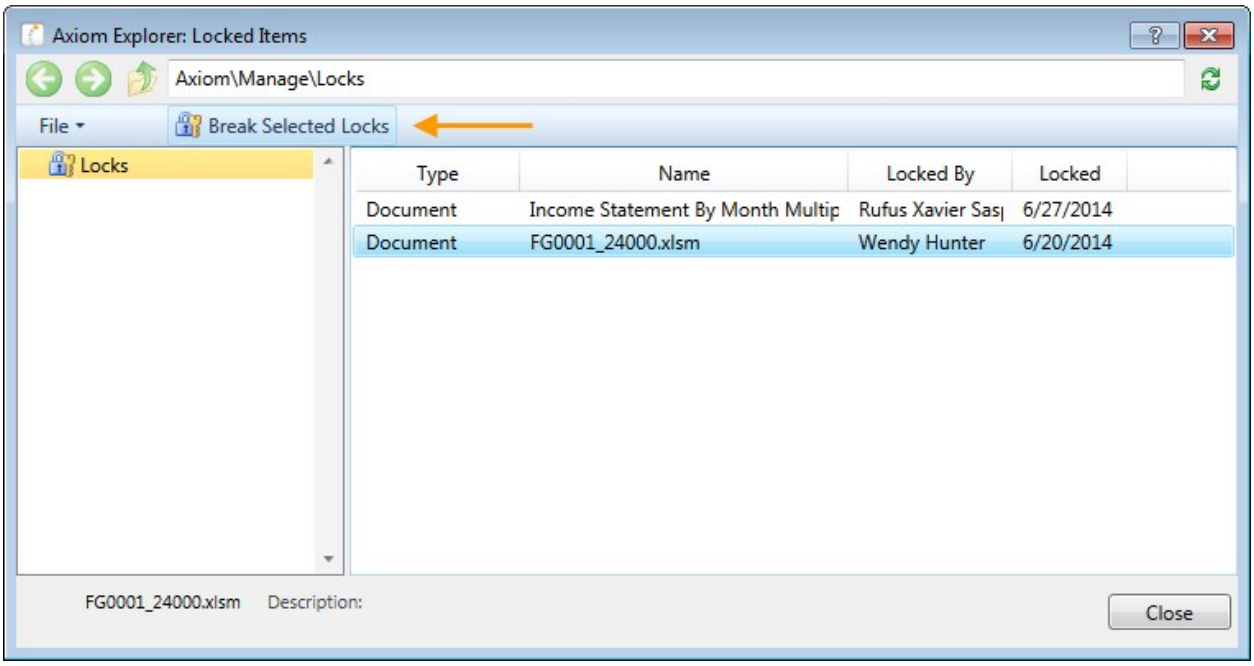
**NOTES:**

- Only *managed files* are locked by Axiom Software. If a file is non-managed (not stored within the Axiom Software database), then the lock status of the file is not tracked by Axiom Software, and the normal file locking behavior of the operating system applies.
- If a file is locked to a user, that user can always re-acquire and release the lock by opening the file again. The only time an administrator needs to manually break the lock is if other users need to access the document.

## Removing locks

If necessary, you can remove the lock on a file or table. To do so:

- Select one or more items in the list, and then click **Break Selected Locks**.




You are prompted to confirm that you want to remove the lock. If you click **Yes**, the lock is removed from the item, and any user with the appropriate rights can open the item as read/write.

**IMPORTANT:** If the user is actually in the document when you break the lock, then that user may not be able to save their changes. If the document does not become locked to any other user in the meantime, then when the original user clicks **Save** they will re-acquire the lock and can save their changes. However, if another user has locked the file, then the original user will be unable to save.

**NOTE:** You can also break locks as you are browsing throughout Axiom Explorer, by using the right-click context menu on a locked file.

# File locking

At any one time, only one user can have an Axiom managed file open for editing.

When a user opens a file as read/write, the file is "locked" to that user. Any other non-admin users that attempt to open the file can only open it as read-only. When the user closes the file, the lock is removed, and any other user with the appropriate level of rights can open it for editing.

**NOTES:**

- File lockout behavior explicitly applies to read/write permissions, not to save data rights. It is possible to grant a user read-only permission together with the ability to save data. This is typically only for situations where the file operates as a temporary input form. If a user has read-only and save data rights, then when that user opens the file, lockout does not apply. Another user can still open the file as read/write.
- File lockout behavior does not apply to non-managed Axiom files. If the file is not stored within the Axiom Software database, then the normal file locking behavior of the operating system applies.
- Opening a file as an Axiom form does not lock the file.

## Administrators and file locking

Users with administrator rights have the option to remove another user's lock on a file. Typically, this is only done when a system crash or other technical issue prevents the file lockout from being removed normally. In this case, administrators can use **Managed Locked Items** to remove the lock on a file.

Administrators can also acquire another user's lock on a file. When an administrator attempts to open a file that another user has open as read/write, the administrator is given the option to open the file as read-only or to acquire the lock. If the administrator acquires the lock, then the file is opened as read/write for the administrator, and the original user no longer has the lock. This may prevent the other user from saving changes to the file, so generally you should not take another user's lock unless you are absolutely certain the other user does not need it.

The original user may be able to save the file if: the document lock is once again available when the user attempts to save, and, no other user has saved changes to the file and therefore incremented the document version. For example, if the administrator takes the user's lock accidentally, then immediately closes the file, and then the original user saves before any other user takes the document lock, then the original user re-acquires the lock and is able to save.

## Users with the same file open in multiple sessions

If a user has a file opened in one Axiom Software session, the same user can open the same file in another session, both as read/write. For example, a user can open the same file in the Excel Client and the Windows Client simultaneously. In this situation, the first-saved file "wins" the lockout. If the version in the Excel Client is saved first, then that file becomes locked to the user, and the version in the Windows Client cannot be saved.

# Refreshing the file system

The Axiom Software file system is the virtual folder directory used to enable easy navigation of managed Axiom files and other system support files.

If files are added by another user, they may not immediately be seen in Axiom Software menus. You can manually refresh the file system to update any stale lists of files.

**To refresh the file system:**

- On the **Axiom** tab, in the **Reports** group, click **Reports > Refresh File System**.

This action is for rare cases when the file system was not refreshed naturally. Most areas that list files from the file system include a built-in refresh to get the latest information, such as when opening Axiom Explorer.

# Saving additional report templates

You can save additional report templates to the Axiom Software file system. When a user creates a new report, they can select any of the report templates to use as a starting point.

The ability to save additional report templates is primarily for report design purposes—for example, to have a report with pre-set fonts, colors, logos, etc. Report templates should be saved without a Control Sheet, so that when the template is used to create a new report, the latest Control Sheet is automatically applied.

Alternatively, you can save a template with a Control Sheet, in order to store pre-set Control Sheet configurations. However, in this case the user will most likely need to upgrade the Control Sheet to use the latest features.

**To save an additional report template:**

1. Design the report sheets as desired within the file that you want to save as a report template. When you are finished with the file, save it to any accessible location.

   You can start with an Axiom report, or you can use a normal Excel file. If the file is an Axiom file, you should remove the Control Sheet before saving the file as a template.

2. On the **Axiom** tab, in the **Administration** group, select **Manage > Axiom Explorer**.
3. In Axiom Explorer, navigate to `\Axiom\Axiom System\Document Templates\Reports`.
4. Right-click the `Reports` folder and then select **Import Files**.
5. In the **Select File to Import** dialog, navigate to the file, and then click **Open**.

The file is saved to the Reports template folder. When a user creates a new report, this file will be one of the options to use as the report template.

**NOTE:** It is also possible to save multiple driver templates, so that users can choose the template to

use when creating a new driver file for a file group. To do this, use the same process detailed previously, but save the file to `\Axiom\Axiom System\Document Templates\Drivers`.

# User access to My Documents

Axiom Software provides a My Documents folder which administrators can use to store their own personal reports, or as a temporary holding area for reports that are "in progress." This folder displays in the My Files area of Axiom Explorer, along with favorites and recent items.

By default, only administrators have access to this folder. In most cases, end users do not need an area like this because they are not creating new files, and exposing the folder to them may cause confusion.

However, if you want to give certain power users access to this folder, you can grant them the **User Documents Folder Access** permission in Security. If a user has this permission, then a My Documents folder is created for them, within their user folder. These user folders are located in the Axiom Software file system at: `\Axiom\Axiom System\User Folders`. Each user has a folder that contains their favorites, recent items, and My Documents (if applicable). The folders are named using the user's database ID, but they display in Axiom Explorer with the user name as follows: ~*username*. These folders are what display to each individual user as their My Files area.

If a user has access to a My Documents folder, and then later the permission is removed, the existing folder will be hidden from the user in Explorer screens but the folder will remain in the file system. You can delete this folder in Axiom Explorer if desired.

**NOTE:** Non-administrator users will not be able to access the Sheet Assistant or use other file-level permissions (such as the ability to unprotect) on files saved to My Documents. Currently there is no way to grant file-level permissions to this area, so only administrators have full access to file features.

# Setting up home pages for Axiom Software

All Axiom Software clients can have a designated Home file.

- In the Desktop Client (Excel or Windows), the Home file opens automatically when a user starts the application. This file is displayed using the tab name Home, and it can be opened or closed using the **Home** check box in the **Display** group of the Axiom ribbon.
- In the Web Client, a designated Home file can be shown when a user navigates to the Axiom Forms area.

One or more Home files may be provided to you as part of your Axiom Software implementation. There may be one default home file located in the `\Axiom\Axiom System\StartUp\Home` directory, or there may be one or multiple home files located in the Reports Library and assigned via Security.

### Using the default Home file in the Startup folders

By default, the system file `Home.xlsx` is opened on startup for all users. This file can be customized for your system as desired. You can also optionally use different Home files on a per client basis.

The default Home file is located in the `\Axiom\Axiom System\Startup\Home` folder. It is used as follows:

| Folder | Description |
|---|---|
| `\Startup\Home` | The Home file in this folder is opened when the system is launched, if:<br>• The user does not have a specified **Home Page** in Security.<br>• The applicable client-specific Home folder is empty. |
| `\Startup\Home\Web Client` | The Home file in this folder is opened when the system is launched, if:<br>• The user does not have a specified **Home Page** in Security.<br>• The Web Client is being used.<br>This folder does not contain a file by default. If you want to use a Web Client-specific Home file, you must copy or import a file to this location. Any file saved to this location must be form-enabled, or else it will be ignored. The file will automatically open as a form in the Web Client environment. |
| `\Startup\Home\Excel Client` | The Home file in this folder is opened when the system is launched, if:<br>• The user does not have a specified **Home Page** in Security.<br>• The Excel Client is being used.<br>This folder does not contain a file by default. If you want to use an Excel Client-specific Home file, you must copy or import a file to this location. |
| `\Startup\Home\Windows Client` | The Home file in this folder is opened when the system is launched, if:<br>• The user does not have a specified **Home Page** in Security.<br>• The Windows Client is being used.<br>This folder does not contain a file by default. If you want to use a Windows Client-specific Home file, you must copy or import a file to this location. |

**NOTES:**

- Only one file can be used as the Home file in each of the `\Startup\Home` folders. If any of the Home folders contain multiple files, the file with the smallest document ID is used.
- If you want to use a form-enabled Home file in the Desktop Client, you must assign the file via Security. If you place a form-enabled file in the `\Startup\Home` folders, it will be opened as a spreadsheet file instead of as a form. (The exception to this is the Web Client folder, where the file must be form-enabled and only opens as a form.)

### Assigning alternate Home files in Security

If desired, you can assign alternate Home files on a per user or role basis. This setting is called **Home Page** and it is defined on the **Startup** tab of Security Management. If a user is assigned an alternate home page, this file replaces the default Home file for that user.

For more information on assigning an alternate home file in Security, see the *Security Guide*.

## Designing Home files for Axiom Software

Home files are designed using Axiom reports. They can be regular spreadsheet reports, or form-enabled reports. Form-enabled reports can be used as Home files in all Axiom Software clients.

Your organization may use one Home file for all users, or you may use multiple Home files that are designed for different roles.

### Using spreadsheet reports as Home files

You can use almost any Axiom Software feature in a spreadsheet Home file. For example, you can use Axiom queries and other query methods in the Home file to show current data that refreshes when the file is opened.

The primary goal of the Home file should be to communicate information, not to perform tasks. The Home file can be graphical and use text to communicate information about the planning process to your end users. Some features, such as save-to-database, cannot be performed in spreadsheet Home files.

The default Control Sheet is automatically hidden for any file that is used as the home page. You do not need to manually hide this sheet when designing a spreadsheet Axiom report to be used as a Home file.

**NOTE:** If a spreadsheet Home file has refresh variables, the refresh variables cannot be displayed when the file is opened. If the file is configured with **Refresh Forms Run Behavior** of **OnManualRefreshAndOpen** or **OnOpenOnly**, the variables will not display and the refresh-on-open query will not be run.

## Using Axiom forms as Home files

Many clients use Axiom forms as Home files, because the web presentation is well-suited to the purpose of a home page, regardless of which client you are using. Web pages can present summary information in a more attractive and user-friendly way than a spreadsheet.

Additionally, Axiom forms provide pre-built support for certain information that is commonly included in Home files, such as:

- Announcements
- Current process tasks
- Process summary

Although it is possible to present this information in spreadsheet Home files, it requires developing a custom solution. Axiom forms support standardized, configurable components that are specifically designed to present this information.

All Axiom form features are available to forms used as Home files, with one exception. When using an Axiom form as a Home file in the Desktop Client, the Web Client container is not available in that environment. However, you can still use refresh variables if desired—the variables will display in a Filters task pane separate from the Web Client container.

When using an Axiom form in the Web Client, keep in mind that the Home file is displayed instead of the built-in browse page for Axiom forms. Therefore, all necessary form navigation should either be incorporated in the Home file itself, or included in the navigation panel of the Web Client container.

## Editing Home files

You can edit Home files just like any other Axiom report. If you are using the default Home file in the Startup folder, only administrators can edit that file. If you are using custom Home files located in the Reports Library and assigned via Security, access to those files is controlled using normal file security.

If the Home file that you want to edit is currently open as your Home file in the Desktop Client, then you must first close the Home file so that you can open it with read/write permissions.

1. Click the **X** button on the **Home** file tab to close the Home file (or right-click the file tab and click **Close**). Note that you must have at least one other file open before you can close the Home file (otherwise Axiom Software will close if no files are currently open).
2. Open the file using Axiom Explorer, and edit it as desired. Once you have finished your edits, save and close the file.
3. You can now re-open the file as the Home file by clicking **Show Home** in the Axiom ribbon.

Because you have reopened the Home file, you will see your edits immediately. Other users will see the changes the next time that they log in (or if they close and then reopen the Home file within their current session).

**NOTE:** If the Home file has been configured as non-closeable in Security, then you will not be able to close it. In this case, you must use Save As to save the Home file with a different name, then make your edits in that file. To replace the existing file with your new file, you should export the new file, then rename it locally to have the same name as the original file, then import it over the original file. This process will retain the document ID of the original file.

# Using additional startup files

By default, the Axiom Software Home page opens automatically when the Desktop Client is started. If desired, you can designate additional files to open automatically when the application is started. There are two ways to assign additional startup files: by using Security settings, or by using the system Startup folder.

Any files designated as startup files are always opened as read-only at startup, regardless of the user's access permissions to the file (if any). The user does not need to be granted access permissions to the file.

Additional startup files can be any of the following file types:

- Axiom task panes
- Axiom ribbon tabs
- Axiom report files (regular or form-enabled)
- Regular Excel spreadsheets

## Using Security to assign additional startup files

In Security Management, on the **Startup** tab, you can specify task panes, ribbon tabs, and other files to open when Axiom Software is started, on a per user and per role basis. This is the most flexible and most common method of assigning additional startup files.

When using security settings to designate additional startup files, the files must be located in either the Reports Library, the Task Panes Library, or the Ribbon Tabs Library.

For more information on assigning additional startup files in Security, see the *Security Guide*.

## Using the system Startup folder

You can place files in the `\Axiom\Axiom System\Startup` folder structure, and those files will be opened on startup, for all users. If desired, you can define different files to open based on which Axiom client is used.

| Folder | Description |
| --- | --- |
| \Startup | Files in this folder are opened when the Desktop Client is started (either Excel Client or Windows Client). |
| \Startup\Excel Client | Files in this folder are opened when the Excel Client is started. |
| \Startup\Windows Client | Files in this folder are opened when the Windows Client is started. |
| \Startup\Hidden | Files in this folder are opened as hidden files on startup. If used, these hidden files are intended to support background system processes and are not exposed to users. This is primarily a legacy feature that only remains available to support backward-compatibility. |

You can use Axiom Explorer to copy or import files into these folders. Only administrators have access to these folders.

**NOTES:**

- If a user has any assigned "other documents" on the **Startup** tab of Security, then the Startup folders are ignored. Assigning an alternate home page does not prevent use of the Startup folders.
- Ribbon tabs are not supported for use in the Startup folders. If you want to open a ribbon tab at startup, it must be assigned via Security.
- Any form-enabled files placed in the Startup folders will be opened as spreadsheet files, not as forms. If you want to open a form-enabled file as an additional startup file, it must be assigned via Security.


# Auditing

Axiom Software provides robust auditing capabilities so that you can view a detailed record of system activities and changes made within the system. This audit coverage is provided by the following features:

- **Axiom Audit Manager**: Using the Axiom Audit Manager, administrators can view audited activities for the system. Activities can be filtered by date, type, user, session, document, and table.
- **Audit Tables**: Changes to table data are maintained in corresponding audit tables. By reporting on these audit tables, you can see what changes were made to the data, when and by whom.
- **Document History**: Changes to documents are maintained using document versions stored in the database. You can view prior versions, restore prior versions, and recover deleted documents. For more information on these features, see File Management.

All auditing data is stored in the audit database. By default, the audit database is a separate database that is paired with your system database. Alternatively, the audit database can be implemented as an "embedded" database within the system database (this configuration is primarily for Axiom cloud service systems).

## Viewing the audit log

Axiom Software keeps an audit log of important activities performed in the system, so that you can see what was done in the system, when it was done, and by whom. The types of activities that are logged include:

- Document access
- Refreshing a file
- Data saves via various system features
- Execution of key utilities such as Process Plan Files
- Scheduler job processing
- Changes to key system components such as security, file groups, and tables

The audit log is only accessible by administrators and users with the **Browse Audit History** security permission.

**To view the audit log in the Excel Client or Windows Client:**

- From the **Axiom** tab, in the **Administration** group, click **Manage > Auditing History**.

The audit log is web-based and always opens in the browser (the Web Client), even if it is launched from the Desktop Client.

**To view the audit log in the Web Client:**

Go to the Auditing area of the Web Client:

| | |
|---|---|
| **Example On-Premise URL** | `http://ServerName/Axiom/AuditManager`<br>Where *ServerName* is the name of the Axiom Application Server at your organization. |
| **Example Cloud System URL** | `https://CustomerName.axiom.cloud/AuditManager`<br>Where *CustomerName* is the name of your cloud service system. |

Alternatively, you can go to the Axiom Software launch page and click the **Audit Manager** icon.

## Activities view

When using the **Activities** view, audit data is organized by activity. By default the dialog displays the last 100 activities performed in the system, with the most recent activity listed at the top.

You can filter the activity data using the filter settings at the top of the screen. You can filter by any combination of dates (From and To), by activity type, by user, by document, and by table. For dates, you can click the **Today** or **Yesterday** links to automatically set the date filters as appropriate for those days. To clear a filter, click the **X** button to the right of filter box.

In addition to the information shown in the activity grid, you can get more information as follows:

- **Activity Type**: When you select an activity in the list, the summary details for that activity display at the bottom of the screen. You can also click the activity type to be taken to the full details for the activity, including child activities.
- **User Session**: Click this item to be taken to the User Session view, filtered by the selected session and showing the activities performed during that session.
- **Children**: Click this item to view the child activities for the selected activity. Child activities are activities performed as part of the larger parent activity (such as several individual tasks performed as part of a Scheduler job).

## User Sessions view

When using the **User Sessions** view, audit data is organized by session. By default the dialog displays the last 100 sessions for the system, with the most recent login listed at the top. A *user session* encapsulates all activity for a user from the time the user logs into the system via a particular client, to the time the user logs out.

You can filter the session data using the filter settings at the top of the screen. You can filter by any combination of dates (From and To), by user, and by client type. For dates, you can click the **Today** or

**Yesterday** links to automatically set the date filters as appropriate for those days. To clear a filter, click the **X** button to the right of filter box.

To see the activities performed during a particular user session, click the **View** link for that session. This will open the Activities view, filtered to show only the activities for the selected session.

**NOTES:**

- The **Domain** is only noted when the user logs in using Windows User Authentication.
- The **Impersonated By** column is used to track when an administrator logs in as another user, using the "Log in as" feature in Security. In this case the user of the session is the "log in as" user, and the "impersonated by" user is the administrator user.

### Treatment of Scheduler jobs

Each time the Axiom Scheduler Service executes a job, it creates a new session to perform the tasks of that job. The client type of the session is Scheduler. By default, audit data resulting from Scheduler client sessions is excluded from the Audit Manager results unless you select the check box to **Include scheduler clients** at the top of the screen. Scheduler sessions are excluded by default due to the frequent execution of system jobs such as the SMTP job—this data can easily fill up the results.

The user of each Scheduler session is the user identity under which the job is being run. Regular jobs will be run as either the owner of the job or the requester (for jobs triggered by event handlers). If the job is a system job, then the job is run with the user name of "Scheduler Service System".

If a user opens the Scheduler dialog and starts a job using the Run Now option, the activity for that job will be logged in a separate Scheduler session, not as part of the user's current session. The user's current session will show activities such as accessing the Scheduler dialog and opening the job, but not executing the job.

### Availability of audit data

Audit data is available based on the purge settings configured in your System Data Purge Scheduler job. This job purges audit data as of a specific number of days. For example, if the job is configured to purge data older than 15 days, then you can only view 15 days' worth of audit data in the Audit Manager. For more information, see Purging audit data.

# Auditing changes to table data

Axiom Software can track the specific data changes made to each record in a particular table. This is a more granular level of auditing than what is available in the activity-based audit log, which tracks when actions are performed on a table.

## Enabling auditing for a table

The table classification impacts whether auditing is enabled by default and whether it can be configured:

- **Data**: Auditing is enabled by default. It can be disabled if it is not needed.
- **Reference**: Auditing is enabled by default. It can be disabled if it is not needed. This includes picklist tables and KPI tables.
- **Document Reference**: Auditing is enabled by default and cannot be disabled. However, if the in-memory table feature has been enabled for document reference tables, then the in-memory tables are not audited.

**NOTE:** If a table uses the Large Table index scheme, auditing cannot be enabled for that table.

If auditing status can be configured for a table, then it can be changed by modifying the **Auditing** setting in the table properties. If enabled, then the table has a corresponding audit table in the audit database. The audit table name is AU_*TableName*.

Generally speaking, auditing should only be enabled for a table if it is necessary to track the granular data changes to that table. Table data auditing can impact performance for import utilities, if the import involves large amounts of updated or deleted data for the destination table. Table auditing typically does not significantly impact the performance of other data saves within Axiom (such as using Save Type 1 or Open Table in Spreadsheet), although there are a few exceptions (such as saving extremely large amounts of data using File Processing with Save at End).

## How table auditing works

If auditing is enabled for a table, then the corresponding audit table tracks when a record in the source table is updated or deleted. It does not track when new records are inserted. The audit table also does not track structure changes to the table, or other activities such as when a table was queried—these types of activities are tracked in the main audit log (viewable using the Axiom Audit Manager).

For example, if a save-to-database is performed and an existing record is updated in a table, this will be logged as follows:

- The Axiom Audit Manager logs that the save-to-database occurred, as well as details such as the user who executed the save and what table was affected. This level of auditing is always available and cannot be disabled, though differing methods of save-to-database may produce different levels of detail.
- If auditing is enabled at the table level, then the specific data change is logged in the corresponding audit table. This data change can then be viewed in the Axiom Audit Manager, or by querying the audit table directly (such as by using an Axiom query in a report).

  However, if auditing is not enabled at the table level, then the specific data change is not logged and cannot be viewed or reported on. You will only know that a save-to-database occurred.

The available audit history at any particular time depends on how your System Data Purge Scheduler job is configured. For more information, see Purging audit data.

## Reporting on audited table data

You can use an Axiom query to report upon changes made to table data, for tables that have auditing enabled. These "audit tables" store information on the updated or deleted records in a table.

Any user with at least read-only permission to a particular table has the same level of permission to its corresponding audit table.

### Audit table information

If a table is configured to allow auditing, then the audit version of the table is named as follows:

AU_*TableName*

For example, if the table is named DEPT, then the corresponding audit table is named AU_DEPT. This is the name that you would specify as the primary table for the Axiom query.

The audit table contains the same columns as the original table. While the original table only has one record per unique key combination, the audit table may have multiple records for a particular key combination, representing each time that key combination was modified. These "duplicate" rows are differentiated using the following columns:

- RecordModifiedBy: The user who modified the record.
- RecordModifiedDTM: The date/time the record was modified.
- AuditModifiedDTM: The date/time the record was added to the audit table.

For example, imagine a new record is added to a table on 1/1/2018 by user jdoe. Initially this record is not reflected in the audit table because audit tables do not track insertions. This record is then modified on 1/15/2018 by user jsmith. Now both tables have a record as follows:

Source table - this reflects the record as it exists currently

| ACCT | DEPT | <Other columns> | RecordModifiedBy | RecordModifiedDTM |
|---|---|---|---|---|
| 1000 | 40000 | ..... | jsmith | 1/15/2018 12:00:00 |

Audit table - this reflects the prior version of the record

| ACCT | DEPT | <Other columns> | RecordModifiedBy | RecordModifiedDTM | AuditModifiedDTM |
|---|---|---|---|---|---|
| 1000 | 40000 | ..... | jdoe | 1/1/2018 12:00:00 | 1/15/2018 12:00:00 |

If the record is modified again in the source table, then another record will be added to the audit table to capture the prior version of the record.

### Axiom query design considerations

Keep in mind the following when querying an audit table using an Axiom query:

- **Sum By**: The grouping level for the Axiom query should be set to the unique keys in the table, plus RecordModifiedBy and RecordModifiedDTM. This will return unique rows for each record/user/change combination. For example, if the key columns of the table are ACCT and DEPT, the grouping should be set to:

  ```
  Acct, Dept, RecordModifiedBy, RecordModifiedDTM
  ```

  You can also use AuditModifiedDTM in place of the RecordModified columns (this is the default sum level if you do not specify one). This column tracks when the record was added to the audit table.

- **Filter**: If you want to filter the data returned by the query, it is best to use filters that apply to the overall query to the database (such as the Axiom query data filter), rather than filters that apply at the sheet level (such as data range filters). The audit table is likely many times larger than the source table, so server-side filtering is recommended to improve performance.

- **Deleted rows and columns**: If a column has been deleted in the source table, that column still exists in the audit table but you cannot query it using an Axiom query. However, if a row has been deleted in the source table, that deleted row can be returned from the audit table by an Axiom query.

- **Security**: Your *current* security read filter for the source table will be applied to any queries to the audit table.

- **Outer joins**: The audit table query can only use the audit table and any relevant lookup tables. Outer joins to other data tables / audit tables are not supported.

### Using the Report Wizard to create an audit table report

You can use the Report Wizard to automatically create an audit report for a particular table.

The report shows all current records in the table, as well as the available history for each record. The report uses nested Axiom queries, where the first query is for the selected table (example: ACCT) and the second query is to the audit table (example: AU_ACCT).

**NOTE:** Because this report is based on the current records in the table, deleted records are not included. If you want to view deleted records, then you must manually create a query directly to the audit table instead of using the Report Wizard.

# Purging audit data

Over time, the audit data builds up in the audit database for the system. Audit data should be purged periodically to help maintain overall system performance.

You can purge audit data by using the System Data Purge Scheduler job. This is a system job that is created and run automatically within each system.

To purge audit data, edit the system job to define a number of days of audit data to keep. Each time the job runs, it will purge audit data that is older than the specified number of days. By default, the job is set to 15 days. There are two settings that control purging audit data:

- **Number of days to keep system history**: This controls purging audit data for system activity, and document versions.
- **Number of days to keep table history**: This controls purging audit data for table auditing.

To access the system job, go to **Administration > Manage > Scheduler**. On the **Scheduled Jobs** tab, double-click the **System.SystemDataPurge** job to open it and edit the settings. For more information on the task settings, see the *Scheduler Guide*.

# Reporting on system information

You can use an Axiom query or GetData functions to return system information within an Axiom file. For example, you can query the following:

- Table metadata (tables, columns, calculated fields, aliases, hierarchies, table types)
- Security information (users, roles, permissions)
- File group information (file groups, variables, plan file attachments, aliases)
- Process information (process events, process tasks, process steps)
- Other system information (configuration settings, alert notifications, file access, etc.)

For example, you can write reports that return the following:

- A list of all users that have been granted administrator rights for the system.
- A list of all users that belong to each security role.
- A list of all tables defined in the system, including their descriptions, classifications, table types, and component columns.
- A list of all aliases defined in the system, and the tables and columns they point to.

This information is held in various system tables. A system table is a table maintained by Axiom Software, as opposed to the customer-defined tables defined in the Table Library. System tables are named using the convention Axiom.*TableName*—for example, Axiom.Aliases is the system table that holds information on column aliases.

To query system information using an Axiom query, you use the appropriate Axiom.*TableName* as the **Primary Table**, and then use columns from that table in the field definition. The same system tables and columns can also be used within a GetData function.

For more information on setting up an Axiom query to return system information, see **Axiom Software Help: Axiom queries > Additional Axiom query options > Using an Axiom query to return system information**.

# System Configuration Settings

The Axiom Software database stores several system configuration settings that can be edited if needed. The settings are defined per system.

These settings can be edited for a system in one of the following ways:

- By using the Axiom Software Manager, using the **Installation Manager > Configure System Properties** feature. For assistance with the Software Manager, please see the *Installation Guide* or contact Kaufman Hall Software Support.
- By using Save Type 4 in an Axiom file.
- By using the System Configuration page in the Application Server Administration area of the Web Client. Only certain settings can be modified here, mostly relating to the launch page or the login page.

After making any changes to the system configuration settings, you must reset the cache for the Axiom Application Server before the changes will take effect. The Software Manager will attempt to reset the cache automatically when you save changes to the system configuration settings. However, if you need to do this manually, you can use the **Reset Server Cache** feature in the Software Manager, or on the administration web page for Axiom Software.

The following system configuration settings are available:

### ADONetBatchSize

| | |
|---|---|
| **Default True** | 100 |
| **Description** | Specifies the batch size for update statements. This should only be changed on the advice of Kaufman Hall Software Support. |

### AllowAmbiguousAlternateAggregationAndColumnFilterFieldDefinitions

**Default value** False

**Description** Specifies whether unsupported combinations of alternate aggregations and column filters can be run in an Axiom query. By default, this is False, which means that the Axiom query will error if an unsupported condition is found. The True case is only intended for situations where an existing report uses these unsupported combinations and the customer wants to continue to be able to run the report, understanding the limitation.

### AllowBlanksToBeDataRowSeparators InAQDataUpdate

**Default value** True

**Description** Specifies whether blank rows cause duplicate records in an Axiom query data range to be treated as contiguous or not (True/False). This is for purposes of determining update behavior for duplicate records. By default, blank rows are treated as data separators and each duplicate row will be updated.

If disabled, then blank rows are ignored and the duplicate records are treated as contiguous, which means only the first row in the contiguous block will be updated.

### AllowShowExcel

**Default value** True

**Description** Specifies whether the icon to install / launch the Axiom Excel Client shows on the Axiom Software launch page (True/False). By default the icon is available.

If disabled, the icon will be hidden and therefore users cannot launch the Excel Client from the launch page. Users can still launch the client via other means if it is installed (such as a desktop shortcut or a dedicated URL).

This setting can be modified using the System Configuration page in the Web Client.

### AllowShowPowerPoint

**Default value** True

**Description** Specifies whether the icon to install / launch the PowerPoint Add-In shows on the Axiom Software launch page (True/False). By default the icon is available.

If disabled, the icon will be hidden and therefore users cannot install or launch the PowerPoint add-in from the launch page. Users can still install and launch the add-in via other means (such as by opening a PowerPoint file saved within the Axiom Software database).

This setting can be modified using the System Configuration page in the Web Client.

### AllowShowWord

**Default value** True

**Description** Specifies whether the icon to install / launch the Word Add-In shows on the Axiom Software launch page (True/False). By default the icon is available.

If disabled, the icon will be hidden and therefore users cannot install or launch the Word add-in from the launch page. Users can still install and launch the add-in via other means (such as by opening a Word file saved within the Axiom Software database).

This setting can be modified using the System Configuration page in the Web Client.

### AuthenticationDomainSelectionListRequired

**Default value** False

**Description** Specifies whether the authentication domain selection list always displays on the login screen (True/False). By default, the selection list only displays when it is necessary (meaning only when there are duplicate user names defined in Axiom that use different domains). If you want the domain selection list to always show, then set this to True.

### AutoCastFloatColumnsDuringAggregation

**Default value** False

**Description** Specifies treatment of numeric columns when data is aggregated by a query to the database. This should only be changed on the advice of Kaufman Hall Software Support.

### AutoUpdateProcessTaskPane

**Default value** True

**Description** Specifies whether the Process task pane automatically refreshes in response to process updates that affect the user (True/False). By default the task pane automatically refreshes.

If disabled, then the Process task pane does not automatically refresh. Instead, a message displays at the top of the task pane to inform the user that the process has updates and the user should refresh the task pane to see them. The user can click the message to refresh.

This setting should only be changed to False if system performance is being affected by the automatic process updates, which should be rare.

### AxiomSignalRServerAddress

**Default value** <Blank>

**Description** The URI of the server used to control IPC traffic for Axiom Software. By default, when no value is specified here, the Axiom Application Server for the installation is used.

This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### ClickOnceClientNameExcel

**Default value** Axiom EPM Excel Client

**Description** The name of the shortcut for the Excel Client when using ClickOnce installations. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### ClickOnceClientNameWindows

**Default value** Axiom EPM Windows Client

**Description** The name of the shortcut for the Windows Client when using ClickOnce installations. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### ClickOnceShortcutLocation

**Default value** DesktopAndStartMenu

**Description** The location where Axiom shortcuts will be placed when using ClickOnce installations. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

Available options are: **DesktopAndStartMenu**, **DesktopOnly**, **StartMenuOnly**.

### ClickOnceShortcutType

**Default value** CurrentClient

**Description** Specifies whether shortcuts are created when using ClickOnce installations, and if so what type. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

Available options are: **None**, **Generic**, **CurrentClient**, **Both**. See the *Installation Guide* for more information on what these options mean.

### CreateKeyIndexesOnTemporaryTables

**Default value** True

**Description** Specifies whether indexes are created for key columns of temporary tables (True/False). This should only be disabled on the advice of Kaufman Hall Software Support.

**NOTE:** By default, Axiom Software attempts to create columnstore indexes on temporary tables (attempting clustered first, then non-clustered). This setting is only honored if the system is not able to create the columnstore indexes.

### CreateNonKeyIndexesOnTemporaryTables

**Default value** True

**Description** Specifies whether indexes are created for non-key columns of the import temptable. This should only be disabled on the advice of Kaufman Hall Software Support.

**NOTE:** By default, Axiom Software attempts to create columnstore indexes on temporary tables (attempting clustered first, then non-clustered). This setting is only honored if the system is not able to create the columnstore indexes.

### CriticalErrorRegularExpression

**Default value** Object reference not set

**Description** Defines error text to match on for purposes of flagging critical errors in server logs. By default, any error messages containing this text are considered critical. This should only be changed on the advice of Kaufman Hall Software Support.

### DatabaseIsCaseSensitive

**Default value** False

**Description** Specifies whether case-sensitivity should be recognized for SQL Server databases. This should only be enabled if your SQL Server database collation is case-sensitive.

### DefaultColumnValue_Boolean

**Default value** 0 (meaning False)

**Description** Defines the default value for Boolean columns. You can change this to 1 for True, or blank for null. It is not recommended to use null as the default for Boolean columns unless you have a particular use case that requires it.

If changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### DefaultColumnValue_Date

**Default value** <Blank> (meaning null)

**Description** Defines the default value for Date columns. You can change this to any valid default value for Date columns.

If changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### DefaultColumnValue_DateTime

**Default value** <Blank> (meaning null)

**Description** Defines the default value for DateTime columns. You can change this to any valid default value for DateTime columns.

If changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### DefaultColumnValue_Number

**Default value** 0

**Description** Defines the default value for Integer columns (all types) and Numeric columns. You can change this to any valid default value, including blank for null. It is not recommended to use null as the default for number-based columns unless you have a particular use case that requires it.

If changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### DefaultColumnValue_String

**Default value** ''

**Description** Defines the default value for String columns. You can change this to any valid string value. Single quote marks are required.

> **NOTE:** Although it is possible to set the String column default to blank for null, Axiom Software does not differentiate between null values and empty string values in String columns within the spreadsheet environment. Therefore it is recommended to leave the default of empty string if you want the default to be null (blank).

If changed, the new value only applies to new columns created going forward. Existing columns will continue to use the default value that was set when they were created.

### DefaultEmailClientName

**Default value** Outlook

**Description** Specifies a custom name for the default email client, as displayed when using the **Email Workbook** feature.

If desired, you can change this name to a different email client used by your organization, or you can clear out the setting and leave it blank. If blank, then the text "Default E-Mail Client" will display in the dialog for this option.

The email client option is only available to users who are running the Axiom Desktop Client on their client machines, using the ClickOnce installation type. When using the shared client, emails must be sent using the Axiom server.

### DefaultFilterType

**Default value** Includes

**Description** Specifies the default comparison method for filtering "simple view" pick lists, such as when using Choose Data Element. Available options are: **BeginsWith** or **Includes**.

### DefaultGetDataNoValueErrorMessage

**Default value** <Blank>

**Description** Specifies the return value when the parameters of a GetData function use valid syntax but the function does not return any data. You can change this to any text that you want to display in this situation. This return value can be overridden within each individual GetData function.

### DefaultGetDataSQLErrorMessage

**Default value** #ERR

**Description** Specifies the return value when the parameters of a GetData function use invalid syntax, resulting in a SQL error. You can change this to any text that you want to display in this situation. This return value can be overridden within each individual GetData function.

### DisableHashBytesCheck

**Default value** False

**Description** Specifies whether the hashbytes check is disabled. This should only be changed on the advice of Kaufman Hall Software Support.

### EnableFreeThreadingInUserFunctions

**Default value** True

**Description** Specifies locking behavior for Axiom functions. This should only be disabled on the advice of Kaufman Hall Software Support.

### EnableLegacyAQMode

**Default value** False

**Description** Specifies whether to use the legacy method for creating database queries for Axiom queries (True/False). This is for backward-compatibility only and should never be manually enabled.

If enabled, then the database queries for Axiom queries are created using the rules in place prior to version 5.1. When you upgrade from a pre-5.1 system to the current version, your system will be automatically set to this mode if the upgrade script detects that your system has the potential for a certain type of query ambiguity (where a specified lookup column potentially exists on more than one lookup table). In order to disable legacy mode, you must modify any affected files to use fully qualified syntax to specify the intended lookup table. Please contact Kaufman Hall Software Support for more information.

### EnableLegacyWorkflowEngine

**Default value** False

**Description** Specifies whether the legacy Workflow feature is visible in the system (True/False). By default, the Workflow feature is hidden and cannot be used. Plan file processes in Process Management should be used to manage plan files through a set of defined steps.

If enabled, then the Workflow feature is visible and can be used. This should only be enabled in older systems with existing workflows, where customers have not yet had the opportunity to convert to using plan file processes. When older systems are upgraded to version 2016.2 or later, this setting is automatically set to True if the system has existing workflows.

### EnablePasswordPolicy

**Default value** True

**Description** Specifies whether password rules are enforced (True/False). By default, passwords must meet the rules defined in the PasswordRegularExpression setting. If disabled, then any password is valid.

**NOTE:** If enabled, password rules are only enforced when defining new passwords. If the system contains existing passwords that not meet the rules when this setting is enabled, these existing passwords will continue to be valid.

### ETLMaxRows

**Default value** 500

**Description** Defines the maximum number of temp table rows that will be displayed to the user when running an import using the Allow Pauses option. This setting should only be changed on the recommendation of Kaufman Hall Software Support. The value cannot be higher than 5,000.

### ExcelAddinStartupTimeout

**Default value** 00:01:00 (meaning one minute)

**Description** Defines the timeout value when starting up the Excel Client and the Windows Client. If the client does not start within this length of time, the startup routine is canceled and an error message displays to the user. This should only be changed on the advice of Kaufman Hall Software Support.

### ExcelWorksheetPasswordProtectionEnabled

**Default value** True

**Description** Specifies whether or not worksheet protection is applied with a password (True/False). By default, when a sheet is designated as protected via the Control Sheet, the protection requires a password to remove it manually (Axiom-controlled features can be used to remove the protection without entering a password).

If disabled, then the sheet is still protected but the protection can be removed manually without a password.

### ExtendedDBCommandTimeout

**Default value** 216000

**Description** Defines the timeout value, in seconds, before terminating an attempt to execute a SQL command. This should only be changed on the advice of Kaufman Hall Software Support.

### FileProcessingSaveDataDefaultBatchSize

**Default value** 7000

**Description** Defines the number of records to process at a time when performing a save-to-database using file processing. This setting should only be changed on the advice of Kaufman Hall Software Support.

### ImportSaveBatchSize

**Default value** 1000000

**Description** Defines the number of rows to save in each batch when importing data. This setting applies when inserting or updating rows in the destination table. This setting should only be changed on the advice of Kaufman Hall Software Support.

### IncludeNonKeysColumnsOnTemporaryTable Index

**Default value** True

**Description** Specifies whether non-key columns are included in the non-clustered index on temporary tables. This setting should only be changed on the advice of Kaufman Hall Software Support.

### IntacctPageSize

**Default value** 500

**Description** Defines the number of records to be processed per batch when importing data from Intacct. The value can be set from 100 to 1000. This setting should only be changed on the advice of Kaufman Hall Software Support.

### InvalidPasswordErrorMessage

**Default value** <Blank>

**Description** Defines the message to display when creating a new password if that password does not meet the rules. This should be left blank to use the default message which details the default password rules. If the rules have been customized with the assistance of Kaufman Hall Software Support, then you can define a custom message that details these rules.

### IsPackageIntegrationServer

**Default value** False

**Description** This setting relates to an internal Axiom Software tool and is not for customer use.

### LDAPAllowedSuffixes

**Default value** <Blank>

**Description** Defines the allowed suffixes for LDAP Authentication, used if LDAP Authentication is enabled. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### LDAPAuthEnabled

**Default value** False

**Description** Specifies whether LDAP Authentication is enabled for the system (True/False). If enabled, then the LDAPConnectionString must be specified. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### LDAPConnectionString

**Default value** <Blank>

**Description** Defines the connection string to the LDAP server, used if LDAP Authentication is enabled. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### LegacyWorkflowAdminMode

**Default value** False

**Description** Specifies whether admin are treated using pre-7.2 behavior for purposes of determining workflow stage owners when the assignment is through a role. This is for backward-compatibility only and should not be manually enabled.

By default, administrators are treated the same way as other users for this purpose. If enabled, then administrators will automatically be stage owners if they are members of the assigned role, regardless of whether they have a permission set with **Interacts with Process Management** enabled.

### MapF9ToAxiom

**Default value** True

**Description** Specifies whether F9 performs an Axiom refresh in addition to calculating the workbook. By default, F9 will refresh all sheets in the workbook, and Shift+F9 will refresh the current sheet.

If disabled, then the native F9 behavior of the Microsoft Excel version or the Windows Client applies.

### MaxAQRows

**Default value** 10000

**Description** Defines the maximum number of rows to return for an Axiom query. This setting is intended to prevent users from accidentally bringing in an extremely large number of rows, potentially resulting in excessive processing times or system errors. If the number of rows in the query exceeds the MaxAQRows setting, the user is warned of this and asked if they want to continue.

**NOTES:**

- The only user options are to cancel the query or continue with the full number of rows. This setting defines the threshold for the warning, it does not limit the query to the specified number of rows.
- The warning only applies when users manually refresh Axiom queries. If the query is processed by Scheduler, then processing will fail if the query exceeds the threshold.

This setting can be overridden on a per query basis, on the Control Sheet.

### MaxChooseValueRows

**Default value** 1000

**Description** Defines the maximum number of rows that can be displayed in the Choose Value dialog. This dialog is used by several features to present a list of values for user selection, such as the GetDataElement function and refresh variables. The limit only applies when the dialog is showing the "full grid view"—meaning, the dialog shows more columns than just the key column and description.

### MaxConsecutiveCutnPasteWithoutWaiting

**Default value** 50

**Description** Defines how many consecutive cut and paste actions can occur before the system stops and pauses for a second. This setting has to do with system-controlled cut and paste actions such as when creating snapshot copies. Too many consecutive actions may cause certain environments to experience memory quota errors. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxExcelCellsToProcessInAQBatch

**Default value** 10000

**Description** Defines the maximum number of cells to process at a time when refreshing a vertical Axiom query. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxExcelCellsToProcessInSingleArray

**Default value** 100000

**Description** Defines the number of cells in a workbook that are processed at a time when saving to the database. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxFileAttachmentSizeKB

**Default value** 10000

**Description** Defines the maximum size for a single file attachment. This limit is enforced when users upload attachments for plan files.

When uploading multiple attachments at a time in the Web Client, the total size of all attachments must be less than 50MB. This can be configured separately if needed by using the maxRequestLength property in the web.config file for the Axiom Application Server.

### MaxHierarchyRowSize

**Default value** 50000

**Description** Defines the maximum row size for a hierarchy. If a hierarchy is created for a table with greater than 50,000 rows, then the hierarchy name will display in the Filter Wizard and Quick Filter, but when expanded a message will display to the user explaining that the hierarchy has too many rows and therefore the individual elements cannot be displayed.

If needed you can change this setting to a larger row size, however, large numbers of rows in a hierarchy may cause out-of-memory errors on the client.

### MaximumErrorsToReturnFromSaveOperations

**Default value** 10000

**Description** Defines the maximum number of errors to return when a save-to-database operation occurs. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaximumRowsAllowedThroughCopyTableData

**Default value** 250000

**Description** Defines the maximum number of rows that can be copied when using the Copy Table Data feature. Copying tables with larger numbers of rows may cause performance issues. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxLoginAttempts

**Default value** 0 (meaning unlimited)

**Description** Defines the maximum number of login attempts before a user is locked out. By default, users have unlimited login attempts. You can change this to a set number of login attempts to comply with your organization's security policies.

### MaxRecursionCount

**Default value** 10

**Description** Defines the maximum number of recursive calculations for GetData functions. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxSaveDataRowsInClientMemory

**Default value** 34000

**Description** Defines the maximum number of save data rows that are stored in memory on the client. If the maximum is reached, the data rows are sent to the server and cleared from the client, and then the save process continues. The intent of this setting is to prevent out of memory errors on the client when saving very large sets of data. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxSaveDataToOutputSheetRowsInClient Memory

**Default value** 250000

**Description** Defines the maximum number of save data rows that are stored in memory on the client, for purposes of placing them on an output sheet in the file. This setting only applies when using multipass file processing with the **Save to Output Sheet** option. If Axiom Software determines that the total number of rows for the process will exceed this limit, a message is presented to the user and the process is stopped. This setting should only be changed on the advice of Kaufman Hall Software Support.

### MaxSystemCurrentPeriod

**Default value** 12

**Description** Defines the maximum number that the system current period can be set to. Individual tables can be set to higher current periods if necessary. For example, most tables may use 12 periods, but you may have a table with payroll data that uses 26 periods.

### MaxTableColumnNameLength

**Default value** 0 (meaning use system default)

**Description** Defines the maximum length of table and column names. The system default is 50 characters. You can specify a different maximum size if needed. However, these names will be used in Axiom queries and Axiom functions to query data, so they should be descriptive and short.

### MaxTableRowsToOpenInSpreadsheet

**Default value** 50000

**Description** Defines the maximum number of records to be returned within Open Table in Spreadsheet. This can be set to a larger number if desired, however, returning larger amounts may result in significant performance issues.

### NeverDisableQueryAggregation

**Default value** True

**Description** Controls SQL query aggregation behavior in Axiom Software. This setting should only be changed on the advice of Kaufman Hall Software Support.

### NumberOfPrincipalFiltersToValidateForBulkSave

**Default value** 5

**Description** Specifies the number of filters that are validated when saving user permissions in bulk via Open Security in Spreadsheet. This setting should only be changed on the advice of Kaufman Hall Software Support.

### OpenIDAuthenticationEnabled

**Default value** <Blank>

**Description** Specifies whether OpenID Authentication is enabled for the system (True/False). This option is initially configured during the Axiom Application Server installation, and can be subsequently changed by repairing the installation.

### PackageIntegrationServerURL

**Default value** 50000

**Description** This setting relates to an internal Axiom Software tool and is not for customer use.

### PackageManagerDefaultValuesThreshold

**Default value** 0

**Description** This setting relates to an internal Axiom Software tool and is not for customer use.

### PasswordRegularExpression

**Default value** <Custom string used to define default rules>

**Description** Defines the password rules to be enforced if **EnablePasswordPolicy** is set to **True**. The built-in rules are as follows:

- Must be at least 8 characters long
- Must contain at least 1 upper-case letter and at least 1 lower-case letter
- Must contain at least 1 non-alphabetic character (a number or a symbol)

Axiom Software does not provide functionality for customers to change these rules. If you need to enforce password rules and use different rules, please contact Kaufman Hall Software Support for assistance.

### PlatformUpdateURL

**Default value** <URL to software support portal>

**Description** Defines the URL to check for platform updates. This value should not be manually changed except as directed by Kaufman Hall Software Support.

### ProductUpdateURL

**Default value** <Blank>

**Description** Defines the URL to check for product updates. This value should not be manually changed except as directed by Kaufman Hall Software Support.

### ReleaseJobLocksOnMIAScheduler

**Default value** True

**Description** Specifies whether Axiom Software can break a Scheduler Server's lock on a job if that server reports as MIA (True/False). This should only be disabled on the advice of Kaufman Hall Software Support.

### RemoteDataConnectionBatchSize

**Default value** 10000

**Description** Defines the number of rows to be sent in each batch between the Axiom Application Server and the Axiom Cloud Integration Service. This setting should only be changed on the advice of Kaufman Hall Software Support.

### RestrictSaveAsLocalForManagedDocs

**Default value** False

**Description** Specifies whether **Save As (Local)** is available for managed documents (True/False). By default, Save As (Local) is available for eligible documents.

If you want to remove this option for Axiom managed documents, change this setting to True. When a managed document is active, the Save As (Local) option will no longer appear on the **Save** menu of the Axiom ribbon, or when right-clicking the file tab in the navigation pane.

Administrators can still save a file locally by using the **Export** option in Axiom Explorer.

### SAMLAuthenticationEnabled

**Default value** False

**Description** Specifies whether SAML Authentication is enabled for the system (True/False). If enabled, then the SAMLUserNameHeader must be specified. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed by repairing the installation.

### SAMLUsernameHeader

**Default value** HTTP_EPPN

**Description** Specifies the user name header for SAML when SAML Authentication is enabled. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### SAMLUsernameSuffixPreserved

**Default value** False

**Description** Specifies whether user names defined in Axiom include an @suffix for the SAML identity provider (True/False). By default, it is assumed that Axiom user names do not include a suffix. Set this to True if Axiom user names include a suffix. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### SaveRetryCountOnDeadlock

**Default value** 3

**Description** Specifies how many times the system will attempt to save data to the database after encountering a deadlock error. This setting should only be changed on the advice of Kaufman Hall Software Support.

### Scheduler_AutomationEngine

**Default value** WebPreferred

**Description** Specifies when the Web Engine should be used for Scheduler job processing:

- **WebAlways**: The Web Engine is always used for Scheduler job processing.
- **WebPreferred**: The Web Engine is the preferred spreadsheet processing engine, but Excel can optionally be chosen for specific tasks. The Web Engine will be selected for all spreadsheet tasks by default (except for those tasks which require Excel).
- **WebOptional**: Excel is the preferred spreadsheet engine, but the Web Engine can optionally be chosen for specific tasks. Excel will be selected for all spreadsheet tasks by default.

**NOTE:** Using Excel on the Scheduler server is no longer officially supported. This configuration setting refers to an obsolete feature.

### Scheduler_FromEmailAddress

**Default value** noreply@axiomepm.com

**Description** Defines the "From" email address to use for any email notifications that do not have a separate specified "From" email address (for example, process management email notifications).

This email address must be specified in order to enable these email notifications. If no email address is specified, the SMTP Email Delivery task will experience an error when attempting to deliver the notifications.

**NOTE:** For installations that are using subsystems, the system variable `{Scheduler.FromEmailAddress}` may resolve to a subsystem administrator email address instead of the Scheduler "from" email address.

### SchedulingBehaviorTimezone

**Default value** <Blank>

**Description** Specifies the time zone used when determining the next execution of a scheduled job. By default, if this setting is left blank, the local time zone is used. This setting is primarily for use in Axiom cloud service systems and will be configured by Kaufman Hall Software Support.

### ShowRememberMe

**Default value** True

**Description** Specifies whether the Remember Me option is present on login screens (True/False). By default, this option is available. If disabled, then Remember Me will be hidden on login screens, meaning that users must enter their credentials each time they launch Axiom Software.

If you disable this option after users have already used Remember Me, their current settings will not be cleared. Each user's credentials will continue to be remembered until they clear the credentials by logging out. Credentials are stored separately for the Web Client versus the Desktop Client, so if a user has saved credentials for both environments then they must log out of both. For the Desktop Client, the only way to log out is to use the Word or PowerPoint add-in, because the Excel and Windows Clients do not have a log out feature.

This setting can be modified using the System Configuration page in the Web Client.

### ShowWarningIfMissingAQTags

**Default value** True

**Description** Specifies whether missing `[aq#]` tags for active Axiom queries trigger a warning on refresh (True/False).

The False setting is intended for backward-compatibility only. If warnings are being triggered in many existing files, you can disable this setting until you are able to correct the Axiom query configuration within the affected files. If the [aq#] tag was deliberately hidden using a formula as a means of dynamically disabling a query, the file should be updated to instead use the Active setting for the query. Please contact Kaufman Hall Software Support for assistance as needed.

### ShowWarningOnGetDataProcessingFailure

**Default value** True

**Description** Specifies whether the circular reference warning displays when Axiom Software detects a failure in processing GetData functions. If disabled, then no warning message displays. This setting should only be changed on the advice of Kaufman Hall Software Support.

### SignalRClientInvalidationLagTime

**Default value** 1000

**Description** Specifies the time zone used when determining the next execution of a scheduled job. By default, if this setting is left blank, the local time zone is used. This setting is primarily for use in Axiom cloud service systems and will be configured by Kaufman Hall Software Support.

### SubsystemsEnabled

**Default value** False

**Description** Specifies whether the security subsystem feature is enabled (True/False). By default, security subsystem features do not appear in the user interface, and subsystems cannot be created.

If enabled, then the security subsystem features become available in the user interface.

**NOTE:** If you set this option to True, create subsystems and make user assignments, and then set it back to False, the existing subsystem restrictions will still apply to assigned users. This option simply hides the feature in the user interface; it does not stop any existing subsystems from being enforced. If you do not want to use subsystems anymore, you should remove all user assignments and delete the subsystems before disabling the feature.

### SuspendDrawingWhileScreenUpdatingOff

**Default value** False

**Description** This configuration setting is used to assist in troubleshooting screen drawing issues. This setting should only be changed on the advice of Kaufman Hall Software Support.

### SystemCurrencySymbol

**Default value** <Blank>

**Description** Defines the currency symbol used by columns that are configured as Currency numeric type. If left blank, the currency symbol for the current locale is used.

The Numeric Type is defined in the column properties. Currently, this symbol is only used by file group display columns.

### SystemCurrentPeriod

**Default value** 1

**Description** Defines the current period for the system. To change the current period, you can use the user interface in the software (**Administration > Tables > Table Administration > System Current Period / Year**).

### SystemCurrentYear

**Default value** 1

**Description** Defines the current year for the system. To change the current year, you can use the user interface in the software (**Administration > Tables > Table Administration > System Current Period / Year**).

### SystemName

**Default value** Axiom EPM

**Description** Defines the name of the system. You can change this name as desired.

This setting can be modified using the System Configuration page in the Web Client.

### TableRepartitioningRowLimit

**Default value** 2000000

**Description** Specifies the row limit for repartitioning a table as part of saving the table properties. If the table data exceeds this row limit, then the table will not be repartitioned and instead you will be prompted to process the table repartitioning using Scheduler.

### TiePlanFileSaveToWorkflowTaskSubmit

**Default value** True

**Description** Specifies whether users can complete tasks for plan file processes when saving plan files (True/False). By default, when a user saves a file for which they have an active process task, the user is given the option to complete the task as part of the save. The "complete on save" option only applies when the user opens the file as read/write.

If disabled, then no process prompt displays on save. This setting also applies to the legacy workflow feature.

### TranslateFrenchKeyboardDecimalSeparator

**Default value** False

**Description** Specifies whether special decimal translation behavior applies to the Web Client and Windows Client (True/False). If enabled, then pressing the decimal key on a numeric keypad is interpreted as a comma when the following are true: the current locale's number format is comma, and the number format for the cell is one of Number, General, Percent, Scientific, or Currency. Note that the number format only applies to the Web Client; in the Windows Client, the behavior applies regardless of the number format.

### UpdateStatisticsDuringSave

**Default value** False

**Description** Specifies whether statistics are rebuilt as part of save-to-database processes (True/False). This applies to both the temporary table (if saving more than 10,000 rows) and the destination table. This should only be enabled on the advice of Kaufman Hall Software Support.

### UseLegacyColumnAggregation

**Default value** True

**Description** Specifies whether to use the legacy default aggregation behavior for Integer and Numeric columns in data queries. We recommend leaving the default setting and using the legacy behavior until the new behavior is finalized in an upcoming release. The new behavior leverages the column classification to determine the aggregation type for the column.

### UseLegacySQLParser

**Default value** False

**Description** Specifies whether to use the legacy SQL parser (True/False). This should only be enabled on the advice of Kaufman Hall Software Support.

### UseRestrictedWebModeForNonAdmins

**Default value** True

**Description** Specifies whether to use a special restricted mode in the Windows Client for non-admin users (True/False). If enabled, then certain file features are explicitly tied to read-write rights. For more information please contact Kaufman Hall Software Support.

This setting exists for backward-compatibility only and should not be manually enabled in systems where it is not already enabled.

### UseStickSessionServerHash

**Default value** False

**Description** Specifies how sticky sessions are handled for Cloud Service systems. This setting should only be changed on the advice of Kaufman Hall Software Support.

### VBA_RunMacroDisabled

**Default value** True

**Description** Specifies whether VBA custom solutions can be run while in the Axiom Excel Client (True/False). By default, the system does not check for VBA custom solutions, and AxiomVBA.xlam is not loaded when the system is run.

If set to False, then the system uses the designated event handlers to check for VBA custom solutions, and runs them if found. This setting should only be changed to False if your system is specifically designed to use VBA custom solutions. Enabling custom solutions may unnecessarily impact system performance for systems that do not use VBA.

### WebClientSkin

**Default value** <Blank>

**Description** Specifies the default skin to be applied to Axiom forms. If the **Skin** property for a form is blank, the default skin defined here is used.

By default, this setting is blank, which means the default **Axiom** skin is used. If desired, you can specify any skin that is available to be selected for an Axiom form, including any custom skin defined for your organization. For the list of available skin names, check the **Skin** property on an Axiom form.

If you have defined a specific skin name and then want to return to using the default Axiom skin, clear this setting so that it is blank again.

### WebPageTabItemBackground

**Default value** #FFFFFF

**Description** Specifies the background color of web tabs and form dialogs while the "loading" spinner displays. You can specify a different color using a hexadecimal code.

### WindowsAuthAllowedDomains

**Default value** <Blank>

**Description** Specifies the allowed domains for Windows Authentication, if enabled. Separate multiple domain names with a comma. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

This setting can be modified using the System Configuration page in the Web Client.

### WindowsAuthEnabled

**Default value** True

**Description** Specifies whether Windows Authentication is enabled for the system (True/False). By default this is enabled, which means that the allowed domains must be specified using WindowsAuthAllowedDomains. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

If disabled, then Windows Authentication is not used.

### WindowsAuthUserSyncEnabled

**Default value** False

**Description** Specifies whether Active Directory Synchronization is enabled for the system (True/False), allowing users to be imported from and synchronized with Active Directory. This option is only applicable if Windows User Authentication is enabled for the system. This option is initially configured during the Axiom Application Server installation, and can be subsequently changed using the system configuration settings or by repairing the installation.

### WorkflowChunkingSize

**Default value** 5000

**Description** Defines the number of workflow tasks and instances to be returned from the server to the client at a time. This data is chunked to avoid using up too much memory per request. This setting should only be changed on the advice of Kaufman Hall Software Support.

# Editing system configuration settings using Save Type 4

Using Save Type 4, you can edit system configuration settings by using save-to-database within a spreadsheet, instead of using the Software Manager.

When the system configuration settings are changed using Save Type 4, the application server cache is also reset, so that you will see the effects of most changes immediately. For changes that affect the ribbon, you must close Axiom Software and log in again to see the change.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the configuration properties.
- Row tags in the save-to-database control column, to determine the action to take on the system configuration setting (save or delete).

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.SystemConfiguration;CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]`<br>`[Delete]` |
| Column tags | `DatabaseCode`<br>`ConfiguredValue` |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator.
- An example template with a Save Type 4 to Axiom.SystemConfiguration is available in `\Axiom\Axiom System\Document Templates\Support Utilities`. You can copy this template to your Reports Library and adapt it for your system as needed.

## Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.SystemConfiguration]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.SystemConfiguration;CustomSaveTag=SaveConfig]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the system configuration properties in the control row

Within the control row for the save-to-database process, specify the columns that define the system configuration properties. You can place these properties in any column:

| Column Tag | Description |
|---|---|
| `DatabaseCode` | The database code of the system configuration setting.<br><br>The database code is not always the same as the name displayed for the configuration setting in the Software Manager. The easiest way to get the exact database code for the desired setting is to use an Axiom query to bring in the name and database code for each setting. The table to be queried is Axiom.SystemConfiguration. |
| `ConfiguredValue` | The value of the system configuration setting.<br><br>The value must match the allowed values for the particular configuration setting (such as True or False for a Boolean setting).<br><br>For more information on the allowed values for a setting, see the *System Administration Guide*. You can also use an Axiom query to bring in the data type of each setting from the Axiom.SystemConfiguration table. |

The column tags can be placed to either the right or the left of the SaveStructure2DB tag. Both of the column tags are required and must be placed somewhere in the save-to-database control row.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to save or delete

Within the control column for the save-to-database process, mark each row that you want to be processed with the appropriate row tag:

| Row Tag | Description |
|---|---|
| `[Save]` | Saves the system configuration setting with the specified value.<br><br>If the value is invalid for a particular setting, then the save will not occur.<br><br>If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.SystemConfiguration; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved. |
| `[Delete]` | Deletes the current user-defined value for the configuration setting, to restore the system default value.<br><br>In this case it does not matter what value is specified in the ConfiguredValue column; whatever current value exists in the database will be deleted, so that the system uses the system default value. |

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

When the save-to-database occurs, delete actions are processed first, followed by save actions.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Populating the system configuration properties in the spreadsheet

In the property columns, enter the database code and the configured value to be updated.

You can manually type the values, but the easiest method is to use an Axiom query to populate the spreadsheet with the configuration settings. Once you have the current list as a starting point, you can adjust them as necessary, and then save changes back to the database. This is also the best way to ensure that you have used the correct database code and the correct data type for the configured value.

In the following example, the Axiom.SystemConfiguration table was queried to bring in the list of configuration settings (rows 3 and 4 are the Axiom query field definition and calc method respectively). We then changed the configured value for the SystemCurrentPeriod and flagged that row to save. For the Save Type 4 process, Row 6 is the save-to-database control row and column I is the control column.

| | name | defaultvalue | datatype | databasecode | configuredvalue | |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | databasecode | configuredvalue | [savestructure2db;axiom.systemconfiguration] |

| Display Name | Default Value | Data Type | Database Code | Configured Value | Update |
|---|---|---|---|---|---|
| AxiomOnly | FALSE | Boolean | AXIOMONLY | FALSE | |
| UseLargeFileGroupIcons | FALSE | Boolean | LARGEMODELICONS | FALSE | |
| MaxMemoryChunk | 7000 | Integer | MAXMEMORYCHUNK | 7000 | |
| MaxExcelCellsToProcessInSingleArra | 100000 | Integer | MAXEXCELCELLSTOPROCESSINSINGLEARRAY | 100000 | |
| WorkbookPassword | 9225FdA/iQLB+Z( | String | WORKBOOKPASSWORD | 9225FdA/iQLB+ZeKYYYzqA== | |
| WorksheetPassword | vFZpLHMAxK8p5 | String | WORKSHEETPASSWORD | vFZpLHMAxK8p51HJC6750w== | |
| DesktopAuthenticationEnabled | FALSE | Boolean | DESKTOPAUTHENTICATIONENABLED | TRUE | |
| MaxSystemCurrentPeriod | 12 | Integer | MAXSYSTEMCURRENTPERIOD | 12 | |
| MaxSaveDataRowsInClientMemory | 34000 | Integer | MAXSAVEDATAROWSINCLIENTMEMORY | 34000 | |
| MaxSaveDataToOutputSheetRowsIn( | 250000 | Integer | MAXSAVEDATATOOUTPUTSHEETROWSINCLIENTMEMORY | 10000 | |
| SystemCurrentPeriod | 1 | Integer | SYSTEMCURRENTPERIOD | 6 | [save] |
| UserDirectorySyncEnabled | FALSE | Boolean | USERDIRECTORYSYNCENABLED | FALSE | |
| RefreshButtonLimitToActiveSheet | FALSE | Boolean | REFRESHBUTTON_LIMITTOACTIVESHEET | FALSE | |
| Scheduler_FromEmailAddress | Axiom | String | SCHEDULER_FROMEMAILADDRESS | axiom@axiomepm.com | |
| Scheduler_FileWatcherFolder | | String | SCHEDULER_FILEWATCHERFOLDER | | |
| Scheduler_UseWebEngineAsDefault | FALSE | Boolean | SCHEDULER_USEWEBENGINEASDEFAULT | TRUE | |
| VBA_RunMacroDisabled | FALSE | Boolean | VBA_RUNMACRODISABLED | TRUE | |
| EnableOfflineMode | FALSE | Boolean | ENABLE_OFFLINEMODE | TRUE | |
| MaxTableColumnNameLength | 27 | Integer | MAXTABLECOLUMNNAMELENGTH | 27 | |
| FileGroupCurrentYearPeriodUsed | FALSE | Boolean | FILEGROUPCURRENTYEARPERIODUSED | FALSE | |
| UseSubQueryTempTables | FALSE | Boolean | USESUBQUERYTEMPTABLES | FALSE | |
| MaxAQRows | 10000 | Integer | MAXAQROWS | 10000 | |
| RestrictSaveAsLocalForManagedDocs | FALSE | Boolean | RESTRICTSAVEASLOCALFORMANAGEDDOCS | FALSE | |
| TiePlanFileSaveToWorkflowTaskSubi | FALSE | Boolean | TIEPLANFILESAVETOWORKFLOWTASKSUBMIT | TRUE | |
| SystemCurrentYear | 1 | Integer | SYSTEMCURRENTYEAR | 1 | |
| GetDataBatchSize | 0 | Integer | GETDATA_PROCESSBATCH_BATCHSIZE | 0 | |
| SuppressAutoExpansionOfWorkflow | FALSE | Boolean | SUPPRESSAUTOEXPANSIONOFWORKFLOWTASKPANE | TRUE | |
| SubsystemsEnabled | FALSE | Boolean | SUBSYSTEMSENABLED | TRUE | |

If you only ever needed to change the system current period, then you would not need to bring in all this other information—you could manually set up a save process for that single setting. But if you wanted to have a way to manage all of your configuration settings outside of the Software Manager, you could use a setup like this, and then limit access to this file to only your system administrators.


# Save Type 4 for Tables

Save Type 4 can be used to modify certain table properties from within a spreadsheet. This allows you to change properties based on queries, calculations, and inputs in a spreadsheet, rather than using the software interface. Additionally, Save Type 4 utilities can be scheduled for processing using Scheduler's Process Document List task.

## Managing tables using Save Type 4

Using Save Type 4, you can create new tables and/or edit table properties by using save-to-database within a spreadsheet, rather than using the table administration tools. This may be a more convenient approach if you have many tables to create or edit, or if you need to update the tables frequently.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the table properties.
- Row tags in the save-to-database control column, to flag rows to be saved.

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.Tables;CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]` |
| Column tags | Any supported table property exposed from Axiom.Tables. See Defining the table properties in the control row. |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- Save Type 4 does not apply to document reference tables. Document reference tables must be created and modified using their source document.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.

### Placing the primary save-to-database tag in the sheet

To define the location of the save-to-database control row and control column, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.Tables]
```

The row containing SaveStructure2DB becomes the control row, and the column containing SaveStructure2DB becomes the control column.

You can also optionally use the custom save tag. For example:

```
[SaveStructure2DB;Axiom.Tables;CustomSaveTag=SaveTable]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the table properties in the control row

Within the control row for the save-to-database process, specify the columns that define the table properties. The column tags can be placed in any column, to either the right or the left of the SaveStructure2DB tag.

At minimum, you must include the following properties on the save row to create a new table. All other table properties will use the default value if omitted from the save.

- TableName
- FolderPath (will be created if it does not currently exist)
- TableClassification (Data or Reference)

**NOTE:** It is not possible to create new picklist tables or new KPI tables using Save Type 4.

When updating an existing table, you must include the TableID and the TableName in the save to identify the table to update. You can modify any of the following table properties:

- Description
- TableClassification (table must be eligible for conversion to the target table classification)
- TableType (table must meet the requirements of the target table type)
- FolderPath
- IndexScheme (can change between Default and Hybrid, but not to or from LargeTable)
- IsDataAudited
- IsDataReadOnly
- CurrentPeriod

**NOTE:** It is possible to change the name of a table using Save Type 4. You must include the TableID to identify the table to update, and modify the CustomerTableName field to change the name. Keep in mind that this will not update any references to the table in other tables and throughout the system. This is an advanced feature that should be used with caution.

It is also possible to configure data conversion for a table using Save Type 4, using the following table properties. Saving valid data conversion properties will cause data conversion to become enabled for the table.

- RateTableID
- RateTableFromColumnDefinitionID
- RateTableToColumnDefinitionID
- RateTableTypeColumnDefinitionID
- RateTableScenarioColumnDefinitionID
- DefaultCurrency (meaning Conversion Value From)
- DefaultCurrencyColumnID (meaning Conversion Value Use Lookup)
- ConversionType
- ConversionTypeColumnID
- Scenario

**IMPORTANT:** To complete the data conversion configuration, you must also use Save Type 4 to the Axiom.ColumnSequences table, to set the RateSequenceID on the applicable sequence in the table you are configuring. The RateSequenceID for the table sequence must be set to the sequence ID of the sequence on the data conversion table (the table specified as the RateTableID). This maps the two sequences together for data conversion. For example, if the sequence ID for the sequence on the data conversion table is 130, then the RateSequenceID for the sequence on the table must be set to 130.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging rows to be saved

Within the control column for the save-to-database process, mark each row that you want to be saved with a `[Save]` tag. This is the only valid tag for the table save.

If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.Tables; CustomSaveTag=TableSave]` then you would place the tag `[TableSave]` in the rows that you wanted to be saved.

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

**NOTE:** The row tag can be placed within a formula if desired.

### Populating the table properties in the spreadsheet

In the property columns, enter the table name and the other relevant values for each table that you want to create or update. You can manually type the table properties within the spreadsheet, or you can use an Axiom query to Axiom.Tables to populate the spreadsheet with the properties for existing tables. Once you have the current list as a starting point, you can adjust them as necessary, and then save the changes back to the database.

For example, imagine that you want to "move" a table from one system to another. One way to do this is to:

- Create an Axiom query in the original system to return the table properties and column properties for the table.
- Export the file out of that system and then import it to the new system.
- Within the new system, set up Save Type 4 to save the new table and then save the new columns.
- Make any additional edits to the table manually as needed, such as creating aliases, sequences, and calculated fields.

### Considerations when creating new tables using Save Type 4

When creating a new table using Save Type 4, keep in mind that you are creating the table properties only, you are not creating any table columns. To fully create a new table, you would need two save passes:

- A first save pass to Axiom.Tables to create the new table.
- A second save pass to Axiom.Columns to create the columns in the table.

If you try to save both at the same time, the save to Axiom.Columns will not result in an error, it just won't create any columns because it does not recognize the new table at this point. You must execute the save twice before the columns can be created.

# Managing columns using Save Type 4

Using Save Type 4, you can create, edit, or delete columns in a table by using save-to-database within a spreadsheet, instead of using the **Edit Table** dialog. This may be a more convenient approach when you want to impact many columns at one time.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns to update.
- Row tags in the save-to-database control column, to determine the action to take on the column.

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.Columns;CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]`<br>`[Delete]` |
| Column tags | Any available column property exposed from Axiom.Columns. See Defining the column properties in the save-to-database control row for a list of supported column properties. |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.
- If a column is controlled by an Axiom packaged product, then the column is locked and cannot be deleted or edited via Save Type 4 (with the exception of the read-only status).

## Defining the save-to-database control row and control column

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.Columns]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.Columns;CustomSaveTag=SaveColumn]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the column properties in the save-to-database control row

Within the control row for the save-to-database process, specify the columns that define the column properties by entering the reserved column names from Axiom.Columns. The column tags can be placed to either the right or the left of the SaveStructure2DB tag.

The following properties can be used:

| Column Tag | Description |
|---|---|
| `TableName` | The name of the table. |
| `ColumnName` | The name of the column. |
| `ColumnId` | The database ID of the column.<br><br>**NOTE:** This item should be left blank when creating a new column. |
| `Description` | The optional description of the column. |
| `DataType` | The data type of the column. See Column properties for a list of available column data types. |
| `StringDataMaxLength` | The maximum allowed length of the string, if the column is a String column. Defaults to 20 when creating a new column if not specified. |
| `IsUnicode` | Whether the column is Unicode-compliant. Defaults to False when creating a new column if not specified. |
| `NumericType` | The numeric type of the column, for display formatting purposes. Only applies to Integer (all types) and Numeric columns. Defaults to Number when creating a new column if not specified. |
| `IsKeyColumn` | Whether the column is a key column (True/False). Defaults to False when creating a new column if not specified. |
| `IsUniqueKeyColumn` | Whether the column is an alternate key column (True/False). Defaults to False when creating a new column if not specified. |
| `IsDescription` | Whether the column is a description column for the key column (True/False). Defaults to False when creating a new column if not specified. |
| `ParentColumnId` | The database ID of the assigned lookup column for the column. Leave blank if no lookup column applies. |

| Column Tag | Description |
|---|---|
| `IsDataReadOnly` | Whether the column is read-only (True/False). Defaults to False when creating a new column if not specified. |
| `IsFixedStructure` | Whether the column properties can be modified using Open Table in Spreadsheet (True/False - True means it *cannot*). Defaults to False when creating a new column if not specified. |
| `SortOrder` | The order of the column in relation to other columns. The first column is 0, the next column is 1, and so on.<br><br>Use caution when creating a new column or modifying the sort order for an existing column. Ideally, every column should be assigned a unique number for the sort order. If multiple columns are assigned the same number, then the database will determine the order of those columns. By default if sort order is omitted on a new column, it will be assigned a sort order of 0.<br><br>New columns should be assigned to one number higher than the last number used in the table (assuming you want the column to be placed at the end of the sort order). If you need to reorder the sort order, then you must manually reassign the number of every impacted column; the Save Type 4 process does not have a way of automatically updating these numbers.<br><br>For example, if you have columns with a sort order of 0 to 4, and you want to create a new column to place at a sort order of 3, then you will also have to update the existing column 3 to 4, and the existing column 4 to 5. |
| `IsFilterColumn` | Whether the column displays in the Filter Wizard (True/False). Defaults to True when creating a new column if not specified. |
| `HierarchyDisplayName` | Optional alternative name to display in hierarchy views. |
| `HyperlinkLabel` | Flags the column as containing hyperlink data (web URLs or Axiom file paths). Label is used when auto-generating the hyperlink via Axiom query.<br><br>Only applies to string columns. Leave blank if column does not contain hyperlink data. |

| Column Tag | Description |
|---|---|
| `DefaultValue` | Defines the default value for the column. Note that blank is a valid value and means that you want the default value to be null. If you are creating a new column and you want it to use the system default for the applicable data type instead, then you should not include the DefaultValue property in the save.<br><br>If you want to change this value to a string literal (such as 'N/A') for a String column, the leading quotation mark must be escaped by another quotation mark. For example, the actual value in the cell must be `''n/a'`. This causes Excel to treat the leading quotation mark as part of the actual value; otherwise it is treated as a text marker and stripped from the value (this will cause an error when saving). |

TableName, ColumnName, and Data Type are required; all other fields can be included or not as desired.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to be saved or deleted

Within the control column for the save-to-database process, mark each row that you want to be processed with the appropriate row tag:

| Row Tag | Description |
|---|---|
| `[Save]` | Creates or updates the column with the properties specified in the spreadsheet.<br><br>**NOTE:** If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.Columns; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved. |
| `[Delete]` | Deletes the column, based on the specified ColumnID (or ColumnName if no ColumnID is specified). |

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

When the save-to-database occurs, delete actions are processed first, followed by save actions.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Populating the column properties in the spreadsheet

You can manually type the column properties within the spreadsheet, or you can use an Axiom query to populate the spreadsheet with the columns that are currently defined in the database. Once you have the current list of columns as a starting point, you can adjust them as necessary, and then save the changes back to the database.

For example, imagine that you want to update columns for the Dept table. You can:

- Create an Axiom query to bring in the current columns for that table, by querying Axiom.Columns and filtering the results to just that table.
- Set up the save-to-database tags so that the fields to be saved line up with the information brought in by the Axiom query.
- If you want to edit the properties of existing columns, you can change the properties and then place a [Save] tag in the save-to-database control column. If you want to delete an existing column, you would use a [Delete] tag.
- If you want to create a new column for that table, you can enter the column properties (leaving ColumnID blank), then place a [Save] tag in the save-to-database control column.

When you perform the save-to-database, the columns will be created, updated, or deleted as appropriate.

# Managing column aliases using Save Type 4

Using Save Type 4, you can create, edit, or delete column aliases by using save-to-database within a spreadsheet, instead of using the **Edit Table** dialog. This may be a more convenient approach when "rolling over" many aliases from one table to another.

One major advantage of using Save Type 4 is that you can move column aliases directly from one table to another, without needing to delete the alias first. When you edit aliases using the **Edit Table** dialog, you must delete and then re-create the alias if you want to move it to a different table.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the alias properties.
- Row tags in the save-to-database control column, to determine the action to take on the alias (create or delete).

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.Aliases;CustomSaveTag=Name]` |
| Row tags | `[Save]`<br>`[Delete]` |
| Column tags | `Alias`<br>`Description`<br>`TableName`<br>`ColumnName` |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.

## Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.Aliases]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.Aliases;CustomSaveTag=SaveAlias]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

## Defining the alias properties in the control row

Within the control row for the save-to-database process, specify the columns that define the alias properties. You can place these properties in any column:

| Column Tag | Description |
|---|---|
| `Alias` | The name of the alias. |
| `Description` | The description of the alias. |
| `TableName` | The table that contains the source column for the alias. |
| `ColumnName` | The source column for the alias. |

The column names can be placed to either the right or the left of the SaveStructure2DB tag. All of the columns except Description are required and must be placed somewhere in the save-to-database control row. If the description column is omitted entirely, or if the description is left blank for a particular row, any existing description for the alias will be deleted when the save occurs.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to be saved or deleted

Within the control column for the save-to-database process, mark each row that you want to be processed with the appropriate row tag:

| Row Tag | Description |
|---|---|
| `[Save]` | Saves the alias with the properties specified in the spreadsheet.<br>• If no alias previously existed with this specified alias name, a new alias is created.<br>• If the alias previously existed, it is updated with the properties specified in the spreadsheet. If the alias previously pointed to a different table and/or column, the alias is changed to point to the new table and/or column. You do not have to delete the alias if you want to use it in a different table (as you would if using the **Edit Table** dialog).<br>**NOTE:** If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.Aliases; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved. |

| Row Tag | Description |
| --- | --- |
| [Delete] | Deletes the alias.<br><br>The delete action is only necessary if the alias is not needed anymore, and you no longer want the alias name to exist in the system. If you have an existing alias that you want to point to a new table and/or column, you do not have to delete it first and then recreate it; only a single [Save] action is necessary to update the alias.<br><br>If an alias in the spreadsheet is marked with delete, but no matches are found in the database for that alias name, no error occurs during the save. |

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

When the save-to-database occurs, delete actions are processed first, followed by save actions.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Populating the alias properties in the spreadsheet

You can manually type the alias properties within the spreadsheet, or you can use an Axiom query to populate the spreadsheet with the alias names that are currently defined in the database. Once you have the current list of alias names as a starting point, you can adjust them as necessary, and then save the changes back to the database.

For example, imagine that you have 12 aliases, NYB1-NYB12, which are currently assigned to columns M1-M12 in table Plan2011. You are getting ready to start a new year of planning, so you want to reassign those aliases to columns M1-M12 in table Plan2012. You can:

- Create an Axiom query to bring in the current alias properties. You would set up the query so that the entries in the query's field definition row line up with the appropriate alias properties in the save-to-database control row (or you can use the same row for both purposes).
- Modify the [TableName] column to change Plan2011 to Plan2012. In this example, all other alias properties can remain the same.
- In the save-to-database control column, place a [Save] tag in each row.

When you perform the save-to-database, the aliases will be updated to point to the new table.

**TIP:** After you use the Axiom query to bring in the current alias names as a starting point, you may want to disable the query to avoid accidentally refreshing the file and overwriting any changes you have made to the alias properties. You could also change the refresh behavior for the query to insert and update (instead of rebuild).

## Examples

The following example shows the save-to-database tags and several aliases. When a save is performed, the aliases marked with `[Delete]` will be deleted, and then all of the aliases marked with `[Save]` will be saved.

| | D | E | F | G | H | I |
|---|---|---|---|---|---|---|
| 4 | | | | | | |
| 5 | | Alias | Description | TableName | ColumnName | [SaveStructure2DB;Axiom.Aliases] |
| 6 | | | | | | |
| 7 | | NYB1 | Next Year Budget - Month 1 | Plan2012 | M1 | [save] |
| 8 | | NYB2 | Next Year Budget - Month 2 | Plan2012 | M2 | [save] |
| 9 | | NYB3 | Next Year Budget - Month 3 | Plan2012 | M3 | [save] |
| 10 | | NYB4 | Next Year Budget - Month 4 | Plan2012 | M4 | [save] |
| 11 | | NYB5 | Next Year Budget - Month 5 | Plan2012 | M5 | [save] |
| 12 | | NYB6 | Next Year Budget - Month 6 | Plan2012 | M6 | [save] |
| 13 | | NYB7 | Next Year Budget - Month 7 | Plan2012 | M7 | [save] |
| 14 | | NYB8 | Next Year Budget - Month 8 | Plan2012 | M8 | [save] |
| 15 | | NYB9 | Next Year Budget - Month 9 | Plan2012 | M9 | [save] |
| 16 | | NYB10 | Next Year Budget - Month 10 | Plan2012 | M10 | [save] |
| 17 | | NYB11 | Next Year Budget - Month 11 | Plan2012 | M11 | [save] |
| 18 | | NYB12 | Next Year Budget - Month 12 | Plan2012 | M12 | [save] |
| 19 | | NYB_TOT | Next Year Budget - Total | Plan2012 | TOT | [delete] |
| 20 | | NYB_YTD | Next Year Budget - YTD | Plan2012 | YTD | [delete] |
| 21 | | NYBTOT | Next Year Budget - Total | Plan2012 | TOT | [save] |
| 22 | | NYBYTD | Next Year Budget - YTD | Plan2012 | YTD | [save] |

The purpose of this example would be two-fold:

- To move the NYB aliases from the Plan2011 table to the Plan2012 table.
- To delete the aliases that use an underscore in their name, and recreate them with no underscore.

The next example shows how an Axiom query could be used to bring in the current alias names as a starting point. Row 5 can double as both the field definition row for the query, and as the control row for the save-to-database operation (the query will ignore the save tag in row 5).

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 4 | | | | | | | |
| 5 | | | Alias | Description | TableName | ColumnName | [SaveStructure2DB;Axiom.Aliases] |
| 6 | | | | | | | |
| 7 | [aq1] | | | | | | |
| 8 | NYB1 | | NYB1 | Next Year Budget - Month 1 | Plan2011 | M1 | |
| 9 | NYB2 | | NYB2 | Next Year Budget - Month 2 | Plan2011 | M2 | |
| 10 | NYB3 | | NYB3 | Next Year Budget - Month 3 | Plan2011 | M3 | |
| 11 | NYB4 | | NYB4 | Next Year Budget - Month 4 | Plan2011 | M4 | |
| 12 | NYB5 | | NYB5 | Next Year Budget - Month 5 | Plan2011 | M5 | |
| 13 | NYB6 | | NYB6 | Next Year Budget - Month 6 | Plan2011 | M6 | |
| 14 | NYB7 | | NYB7 | Next Year Budget - Month 7 | Plan2011 | M7 | |
| 15 | NYB8 | | NYB8 | Next Year Budget - Month 8 | Plan2011 | M8 | |
| 16 | NYB9 | | NYB9 | Next Year Budget - Month 9 | Plan2011 | M9 | |
| 17 | NYB10 | | NYB10 | Next Year Budget - Month 10 | Plan2011 | M10 | |
| 18 | NYB11 | | NYB11 | Next Year Budget - Month 11 | Plan2011 | M11 | |
| 19 | NYB12 | | NYB12 | Next Year Budget - Month 12 | Plan2011 | M12 | |
| 20 | NYB_TOT | | NYB_TOT | Next Year Budget - Total | Plan2011 | TOT | |
| 21 | NYB_YTD | | NYB_YTD | Next Year Budget - YTD | Plan2011 | YTD | |
| 22 | [stop] | | | | | | |
| 23 | | | | | | | |

In this example, the query was configured to bring in all alias names for the Plan2011 table. If we want to move all of these aliases to the Plan2012 table, we can edit the TableName column to say Plan2012, add `[Save]` tags to the save-to-database control column, and then save. (You could also place the `[Save]` tag in your in-sheet calc method to automatically populate the column during the Axiom query refresh. However, after the query is run you must delete the tag from the in-sheet calc method, or else Axiom Software will try to save that row.)

# Managing column sequences using Save Type 4

Using Save Type 4, you can create, edit, or delete column sequences by using save-to-database within a spreadsheet, instead of using the **Edit Table** dialog.

Column sequences are managed using two different Save Type 4 processes. Axiom.ColumnSequences defines the sequence "container," such as the sequence name, table name, and ID. Axiom.ColumnSequenceItems defines the individual columns assigned to the sequence and their order. If you are editing an existing sequence, you can save to one or the other as appropriate for the edits you want to make. If you are creating a new sequence, then you must perform both saves using two separate passes. For more information, see Considerations when creating new column sequences using Save Type 4.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the sequence properties.
- Row tags in the save-to-database control column, to determine the sequence rows in the spreadsheet to save or delete.

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.ColumnSequences; CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]`<br>`[Delete]` |
| Column tags | `TableName`<br>`SequenceID`<br>`SequenceName`<br>`StartPeriod`<br>`RateSequenceID` |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.

## Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.ColumnSequences]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.ColumnSequences;CustomSaveTag=SaveSequence]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the column sequence properties in the control row

Within the control row for the save-to-database process, specify the columns that define the sequence properties. You can place these properties in any column:

| Column Tag | Description |
|---|---|
| `TableName` | The name of the table for the sequence. |
| `SequenceID` | The ID of the sequence. |
| `SequenceName` | The name of the sequence. |
| `StartPeriod` | The starting period of the sequence. |
| `RateSequenceID` | The ID of the corresponding mapped sequence for data conversion. Should only be used if you need to set up data conversion for the table that the sequence belongs to. If you want to clear an existing value, set RateSequenceID to 0 instead of making it blank.<br><br>This property applies to the sequence on the table where data conversion is being configured, and should be set to the ID of the corresponding sequence on the data conversion table. |

The column tags can be placed to either the right or the left of the SaveStructure2DB tag. TableName and SequenceName are required. If StartPeriod is omitted when creating a new sequence, the default value is 1.

SequenceID should only be included if you want to edit the name of an existing sequence; in this case the ID must be used to identify the existing sequence to update. Note that the sequence ID changes when the sequence name is changed or when the table is saved, so all other saves to Axiom.ColumnSequences or Axiom.ColumnSequenceItems should use SequenceName instead of SequenceID to identify the sequence.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to save

Within the control column for the save-to-database process, mark each row that you want to be processed with the appropriate row tag:

| Row Tag | Description |
|---|---|
| `[Save]` | Creates or updates the column sequence with the properties specified in the spreadsheet.<br><br>**NOTE:** If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.ColumnSequences; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved. |
| [Delete] | Deletes the column sequence, including any associated column sequence items. |

When the save-to-database occurs, delete actions are processed first, followed by save actions.

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Considerations when creating new column sequences using Save Type 4

The most common use case for saving to Axiom.ColumnSequences is to create new sequences—for example, as part of an automated solution that uses Save Type 4 to create new tables. Editing existing sequences is much less common since sequences do not have many properties and these properties generally do not change once they have been defined.

When creating a new sequence using Save Type 4, keep in mind that you are creating the basic sequence properties only, you are not defining any of the columns that belong to the sequence. To fully create a new sequence, you would need two save passes:

- A first save pass to Axiom.ColumnSequences to create the new sequence.
- A second save pass to Axiom.ColumnSequenceItems to define the columns that belong to the sequence and their order.

If you try to save both at the same time, the save to Axiom.ColumnSequenceItems will fail because it does not recognize the new sequence at this point. You must execute the save twice before the sequence items can be created.

# Managing column sequence items using Save Type 4

Using Save Type 4, you can create and edit column sequence items by using save-to-database within a spreadsheet, instead of using the **Edit Table** dialog. For each column that is assigned to a sequence,

there is a corresponding sequence item that defines the properties of that column within the sequence, such as its order within the sequence.

Column sequences are managed using two different Save Type 4 processes. Axiom.ColumnSequences defines the sequence "container," such as the sequence name, table name, and ID. Axiom.ColumnSequenceItems defines the individual columns assigned to the sequence and their order. If you are editing an existing sequence, you can save to one or the other as appropriate for the edits you want to make. If you are creating a new sequence, then you must perform both saves using two separate passes. For more information, see Considerations when creating new column sequence items using Save Type 4.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the sequence item properties.
- Row tags in the save-to-database control column, to identify the sequence item rows in the spreadsheet to save.

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.ColumnSequenceItems; CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]` |
| Column tags | `TableName`<br>`SequenceName`<br>`ColumnName`<br>`Period`<br>`WeightFactor` |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.

## Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.ColumnSequenceItems]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.ColumnSequenceItems;CustomSaveTag=SaveSequence]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the sequence item properties in the control row

Within the control row for the save-to-database process, specify the columns that define the sequence item properties. You can place these properties in any column:

| Column Tag | Description |
|---|---|
| `TableName` | The name of the table for the sequence. |
| `SequenceName` | The name of the sequence. |
| `ColumnName` | The name of the table column assigned to the sequence item. |
| `Period` | The period of the sequence item, meaning its order in the sequence. |
| `WeightFactor` | The weight factor of the sequence item. |

The column tags can be placed to either the right or the left of the SaveStructure2DB tag.

The following columns are always required: TableName, SequenceName, and ColumnName. When creating a new sequence item, Period is also required. If WeightFactor is omitted when creating a new sequence, the default value is 1.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to save

Within the control column for the save-to-database process, column, mark each row that you want to be saved with a `[Save]` tag. This is the only valid tag for the column sequence item save.

If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.ColumnSequenceItems; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved.

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Considerations when creating new column sequence items using Save Type 4

The most common use case for saving to Axiom.ColumnSequenceItems is when creating new sequences—for example, as part of an automated solution that uses Save Type 4 to create new tables. Editing existing sequence items is much less common because the sequence item properties typically do not change once they have been defined.

To fully create a new sequence, you would need two save passes:

- A first save pass to Axiom.ColumnSequences to create the new sequence.
- A second save pass to Axiom.ColumnSequenceItems to define the columns that belong to the sequence and their order.

If you try to save both at the same time, the save to Axiom.ColumnSequenceItems will fail because it does not recognize the new sequence at this point. You must execute the save twice before the sequence items can be created.

# Managing calculated fields using Save Type 4

Using Save Type 4, you can create, edit, or delete calculated fields by using save-to-database within a spreadsheet, instead of using the **Edit Table** dialog.

Save Type 4 depends on the placement of save-to-database tags within the sheet. There are three components:

- The primary SaveStructure2DB tag, which defines the locations of the save-to-database control row and control column, and specifies the desired operation.
- Column tags in the save-to-database control row, to specify the columns which hold the calculated field properties.
- Row tags in the save-to-database control column, to determine the action to take on the calculated field (save or delete).

Save-to-database tag summary

| Tag Type | Tag Syntax |
|---|---|
| Primary tag | `[SaveStructure2DB;Axiom.CalculatedFields; CustomSaveTag=`*`Name`*`]` |
| Row tags | `[Save]`<br>`[Delete]` |

| Tag Type | Tag Syntax |
|---|---|
| Column tags | `TableName` |
| | `ColumnName` |
| | `ColumnID` |
| | `SequenceName` |
| | `CalculationTypeName` |

**NOTES:**

- Save Type 4 must be enabled for the sheet on the file's Control Sheet in order for the save process to occur.
- The user performing the save must be an administrator or have the **Administer Tables** security permission.

### Placing the primary save-to-database tag in the sheet

To define the save-to-database process, place the following tag in any cell in the sheet, within the first 500 rows:

```
[SaveStructure2DB;Axiom.CalculatedFields]
```

The row containing this tag becomes the control row for the process, and the column containing this tag becomes the control column for the process.

You can also optionally use the custom save tag parameter. For example:

```
[SaveStructure2DB;Axiom.CalculatedFields;CustomSaveTag=SaveField]
```

**NOTES:**

- The primary SaveStructure2DB tag must be located in the first 500 rows of the sheet.
- The SaveStructure2DB tag can be placed within a formula, as long as the starting bracket and identifying tag are present as a whole within the formula.

### Defining the calculated field properties in the control row

Within the control row for the save-to-database process, specify the columns that define the calculated field properties. These properties can be placed in any column:

| Column Tag | Description |
|---|---|
| `TableName` | The name of the table for the calculated field. |
| `ColumnName` | The name of the calculated field. |

| Column Tag | Description |
|---|---|
| `ColumnID` | The database ID of the calculated field.<br><br>**NOTE:** This item should be left blank when creating a new field. |
| `SequenceName` | The name of the sequence for the calculated field. |
| `CalculationTypeName` | The name of the calculation type for the calculated field. |

The column tags can be placed to either the right or the left of the SaveStructure2DB tag. All of the column tags except ColumnID are required, and must be placed somewhere in the save-to-database control row. ColumnID is only required if you want to change the column name.

The control row must be dedicated to containing only valid column names for the Save Type 4 operation to the target table. Any invalid entries in the control row will cause an error when saving.

### Flagging the rows to be saved or deleted

Within the control column for the save-to-database process, mark each row that you want to be processed with the appropriate row tag:

| Row Tag | Description |
|---|---|
| `[Save]` | Creates or updates the calculated field with the properties specified in the spreadsheet.<br><br>**NOTE:** If you have defined a custom save tag in the SaveStructure2DB tag, then you must mark the rows with that tag instead of the default tag. For example, if your primary tag is `[SaveStructure2DB; Axiom.CalculatedFields; CustomSaveTag=MySave]` then you would place the tag `[MySave]` in the rows that you wanted to be saved. |
| `[Delete]` | Deletes the calculated field. |

Only rows that are marked with a valid tag are processed; all other rows are ignored, even if there is content in the property columns. If a row contains a valid tag but no content exists in the property columns, a save error will occur.

When the save-to-database occurs, delete actions are processed first, followed by save actions.

**NOTE:** The row tag can be placed within a formula if desired. For example, you might want to use a formula to determine whether a particular row should be saved or deleted.

### Populating the calculated field properties in the spreadsheet

You can manually type the calculated field properties within the spreadsheet, or you can use an Axiom query to populate the spreadsheet with the fields that are currently defined in the database. Once you

have the current list of calculated fields as a starting point, you can adjust them as necessary, and then save the changes back to the database.

For example, imagine that you want to update calculated fields for the GL2015 table. You can:

- Create an Axiom query to bring in the current calculated fields for that table, by querying Axiom.CalculatedFields and filtering the results to just that table.
- Set up the save-to-database tags so that the fields to be saved line up with the information brought in by the Axiom query.
- If you want to edit the properties of existing calculated fields, you can change the properties and then place a `[Save]` tag in the save-to-database control column. If you want to delete an existing calculated field, you would use a `[Delete]` tag.
- If you want to create a new calculated field for that table, you can enter the field properties (leaving ColumnID blank), then place a `[Save]` tag in the save-to-database control column.

When you perform the save-to-database, the calculated fields will be created, updated, or deleted as appropriate.

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | ColumnName | TableName | SequenceName | CalculationTypeName | ColumnID | | [SaveStructure2DB;Axiom.CalculatedFields] |
| 3 | | ColumnName | TableName | SequenceName | CalculationTypeName | ColumnID | | |
| 4 | | | | | | | | |
| 5 | [aq1] | | | | | | | |
| 6 | GL2015.TOT | TOT | GL2015 | S1 | TotalToPeriod12 | 1685 | | |
| 7 | GL2015.YTD1 | YTD1 | GL2015 | S1 | TotalToPeriod1 | 1686 | | |
| 8 | GL2015.YTD2 | YTD2 | GL2015 | S1 | TotalToPeriod2 | 1687 | | |
| 9 | GL2015.YTD3 | YTD3 | GL2015 | S1 | TotalToPeriod3 | 1688 | | |
| 10 | GL2015.YTD4 | YTD4 | GL2015 | S1 | TotalToPeriod4 | 1689 | | |
| 11 | GL2015.YTD5 | YTD5 | GL2015 | S1 | TotalToPeriod5 | 1690 | | |
| 12 | GL2015.YTD6 | YTD6 | GL2015 | S1 | TotalToPeriod6 | 1691 | | |
| 13 | GL2015.YTD7 | YTD7 | GL2015 | S1 | TotalToPeriod7 | 1692 | | |
| 14 | GL2015.YTD8 | YTD8 | GL2015 | S1 | TotalToPeriod8 | 1693 | | |
| 15 | GL2015.YTD9 | YTD9 | GL2015 | S1 | TotalToPeriod9 | 1694 | | |
| 16 | GL2015.YTD10 | YTD10 | GL2015 | S1 | TotalToPeriod10 | 1695 | | |
| 17 | GL2015.YTD11 | YTD11 | GL2015 | S1 | TotalToPeriod11 | 1696 | | |
| 18 | GL2015.YTD12 | YTD12 | GL2015 | S1 | TotalToPeriod12 | 1697 | | |
| 19 | GL2015.Q1 | Q1 | GL2015 | S1 | Q1Total | 1698 | | |
| 20 | GL2015.Q2 | Q2 | GL2015 | S1 | Q2Total | 1699 | | |
| 21 | GL2015.Q3 | Q3 | GL2015 | S1 | Q3Total | 1700 | | |
| 22 | GL2015.Q4 | Q4 | GL2015 | S1 | Q4Total | 1701 | | |
| 23 | GL2015.CUR | CUR | GL2015 | S1 | CurrentPeriod | 1702 | | [delete] |
| 24 | GL2015.YTD | YTD | GL2015 | S1 | YearToDate | 1911 | | |
| 25 | [stop] | | | | | | | |
| 26 | | NewColumn | GL2015 | S1 | YearToDate | | | [save] |
| 27 | | | | | | | | |

*Example Save Type 4 to Axiom.CalculatedFields*

In this example an Axiom query is used to bring in the calculated fields for the GL2015 table (with the field definition on row 3). The save-to-database control row is row 2, and the save-to-database control column is column H. The save is currently set up to delete the existing calculated field listed in row 23, and to create a new calculated field defined in row 26.


# Reference

## Filter criteria syntax

Several areas of Axiom Software use criteria statements to define a set of data. The syntax for these criteria statement is as follows:

```
Table.Column='Value'
```

- *Table* is the name of the database table.
- *Column* is the name of the column in the database table.
- *Value* is the value in the column.

If the column is String, Date, or DateTime, the value must be placed in single quotation marks as shown above. If the column is Numeric, Integer (all types), Identity, or Boolean, then the quotation marks are omitted.

For example:

- To filter data by regions, the filter criteria statement might be: `DEPT.Region='North'`. This would limit data to only those departments that are assigned to region North in the Region column.
- To filter data by a single department, the filter criteria statement might be: `DEPT.Dept=100`. This would limit data to only department 100.

If the table portion of the syntax is omitted, then the table is assumed based on the current context. For example, if the filter is used in an Axiom query, then the primary table for the Axiom query is assumed. If the current context supports *column-only syntax*, and the specified column is a validated key column, then the lookup table is assumed.

### Operators

The criteria statement operator can be =, >,<,<>,<=,>=. Greater than or less than statements can only be used with numeric values. For example:

```
ACCT.Acct>1000
```

SQL IN and LIKE syntax can also be used. For example:

```
DEPT.Region IN ('North','South')
```

### Compound criteria statements

You can use `AND` and `OR` to combine multiple criteria statements. If you are creating long compound criteria statements with multiple ANDs or ORs, you can use parentheses to group statements and eliminate ambiguity. For example:

```
(DEPT.Region='North' OR DEPT.Region='South') AND (ACCT.Acct=100 OR
ACCT.Acct=200)
```

**NOTES:**

- When filtering on multiple values in the same column, you must use OR to join the statements, not AND. In the example above, if the statement was instead `DEPT.Region='North' AND DEPT.Region='South'`, that statement would return no data because no single department belongs to both the North and South regions. When you use OR, the statement will return departments that belong to either the North or the South regions.
- Alternatively, you can use the SQL IN syntax to create a compound statement for values in the same column. For example, the statement `DEPT.Region='North' OR DEPT.Region='South'` can also be written as `DEPT.Region IN ('North','South')`. The Filter Wizard uses IN syntax by default.

### Using criteria statements in functions

If you are using a criteria statement in a function, such as GetData, you must place the entire criteria statement in double quotation marks. For example:

```
=GetData("Bud1","DEPT.Region='North'","GL1")
```

You can also place the criteria statement in a cell and then use a cell reference in the function. In this case, you do not need to use double quotation marks in the function, unless you are concatenating text and cell reference contents within the function.

### Referencing blank values in filters

If a string column contains a blank value, you may want to create a filter that includes or excludes records with these blank values. For SQL Server, the blank value is stored as an empty string. This empty string is indicated with empty quotation marks in the filter. For example: `ACCT.CMAssign=''` or `ACCT.CMAssign<>''`

If you use the Filter Wizard to construct the filter, it will automatically use the appropriate syntax.

### Referencing values with apostrophes in filters

If a string column contains a value with an apostrophe (such as O'Connor), then that apostrophe must be escaped with another apostrophe so that it is not read as the closing apostrophe for the filter criteria statement. For example:

`Dept.VP='O'Connor'`

Invalid. This construction does not work because Axiom Software reads it as Dept.VP='O' and then does not know what to do with the rest of the text.

`Dept.VP='O''Connor'`

Valid. The extra apostrophe tells Axiom Software that the apostrophe is part of the string value and is not the closing apostrophe.

**NOTE:** This syntax must use two apostrophe characters in sequence and *not* a double quotation mark. If you create the filter using the Filter Wizard, Axiom Software will construct the appropriate syntax for you.

### Referencing Date or DateTime values in filters

If your locale uses a date format where the first value is the day, filters using that date or date-time value will not process correctly. Instead, the date or date-time value must be in standard format. Standard format is `YYYY-MM-DDTHH:MM:SS` for DateTime and `YYYY-MM-DD` for Date.

If you use the Filter Wizard to construct the filter, it will automatically convert the date or date-time value to the appropriate syntax.

# Using formulas with Axiom feature tags

Axiom Software supports a number of features that depend on using reserved tags within a spreadsheet. For example, save-to-database operations, action codes, sheet views, and data sources for Axiom forms all use these feature tags.

Generally speaking, feature tags have the following components:

- A "primary tag" that enables use of the feature within the sheet. This tag defines the control row and control column for the feature, and may also contain additional parameters that control options for the feature.
- Various column tags to be placed in the control row, to define the operation of the feature.
- Various row tags to be placed in the control column, to define the operation of the feature.

For example, `[Save2DB;TableName]` is the primary tag for the Save Type 1 save-to-database feature. The "Save2DB" part of the tag identifies the feature, and the rest of the tag defines parameters for the feature (there are other optional parameters not shown here). The column names that determine the destination columns for the save are the column tags, and the `[Save]` tag that flags rows to be saved are the row tags.

### Using formulas with primary tags

You can use formulas to create primary tags instead of "hard-coding" the tag within the cell. However, the initial bracket and the identifying text must be present as a whole in the formula. You cannot

concatenate this part of the tag or reference another cell that contains this tag. For example:

**Valid** `="[Save2DB;"&C13&"]"`

This is valid because the initial bracket and the identifying text “Save2DB” are present as a whole in the formula. Axiom Software looks for the text "[Save2DB" in order to determine if the feature exists in the sheet.

**Invalid** `="["&"Save2DB;" &C13&"]"`

This is invalid because the bracket and identifying text are the result of a concatenation.

**Invalid** `=P4&C13`

This is invalid because the bracket and identifying text are not contained within the formula. This formula might resolve to what looks like a valid tag within the spreadsheet (depending on what is in P4 and C13), but it will not be found by Axiom Software.

Once Axiom Software has identified that a cell contains a primary tag, the formula result is evaluated as normal to determine if the feature is actually applied. For example, if the formula contains "[Save2DB" then Axiom Software recognizes it as a candidate for processing. But if the formula uses an IF statement and therefore the actual result of the formula in the file is a blank cell or other text such as "No Save," then that cell will be skipped for processing.

If the primary tag takes parameters, the parameters can be created by concatenation or by cell reference, without restrictions. All of the following examples are valid:

```
[Save2DB;Plan2018]
=”[Save2DB;”&C13&”]”
=”[Save2DB;”&”Plan”&”2018]”
```

## Using formulas with column and row tags

Once a primary tag has been found in the sheet and identified for processing, its associated column and row tags are fully evaluated. You can use any kind of formula construction to create these tags.

For example, the row tag `[Save]` can be created by concatenation or by cell reference, without restrictions. All of the following examples are valid:

```
[Save]
=”[“&”Save”&”]”
=C24
```

# Index

## D



## E



## F

## H



## I



## K



## L



## M



## N



## O

## P



## R



## S

## T



## U

## V



## Z