

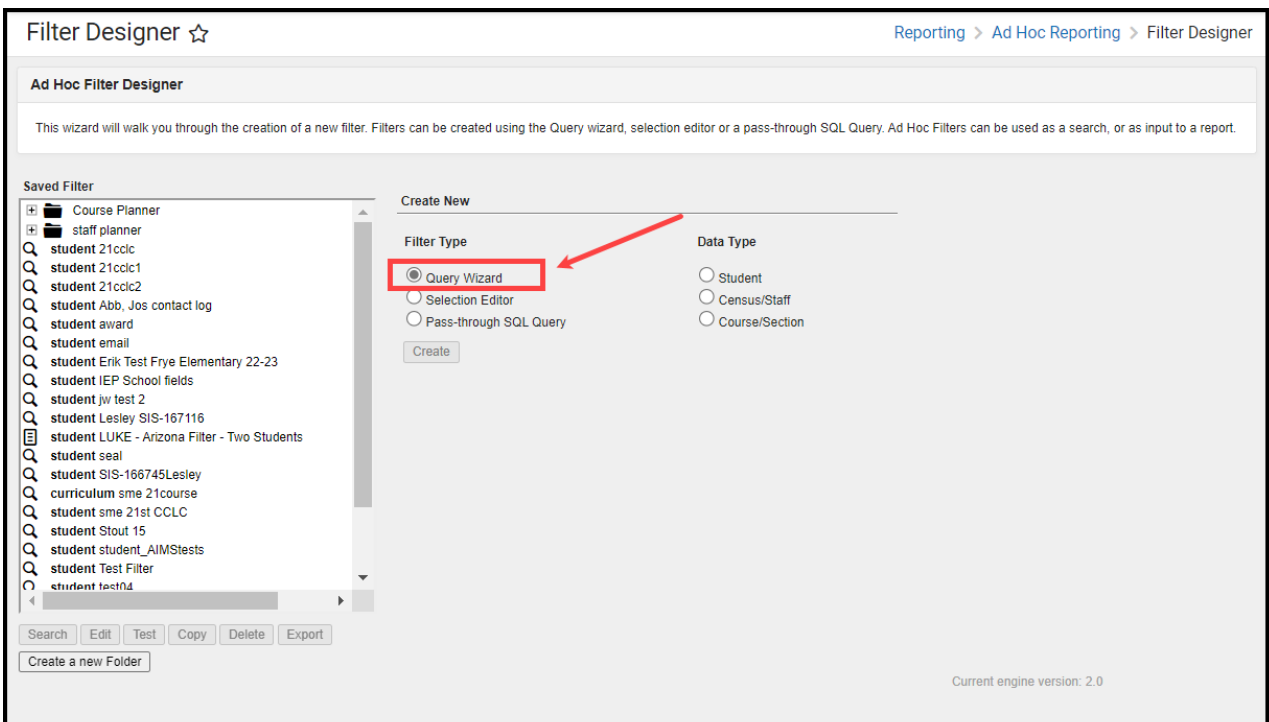
Ad Hoc Filter Designer

Last Modified on 12/04/2024 9:26 am CST

[Query Wizard Features](#) | [Create a Filter](#) | [Manage Filters](#)

Tool Search: Filter Designer

In the Query Wizard, elements are organized in a straightforward pattern, so it is easy to select the elements needed. Filters can be designed with student information, census/staff information or course/section information. Queries for students and course/section data pulls results from the calendar selected in the Campus toolbar. Census/Staff data pulls results from the entire Campus database, regardless of the calendar selected.



Users need at least **Read** rights to the Filter Designer tool and at least **Add** rights for Query Wizard Filters in order to properly use this tool.

For more information about Tool Rights and how they function, see the [Tool Rights](#) article.

Unless using the [Data Warehouse](#), queries should be created in such a way to avoid large results. Generating large queries may cause performance issues.

An ad hoc row limit is set on the database at 5 million rows. Any query that returns more than this is shortened. A warning message displays when this occurs.

When generating large queries and the Ad hoc Row Limit is met:

- Select fewer fields to include in the query

- Add more filters (see [Functions](#)) to reduce the number of records
- Use direct SQL access

Filters including GPA fields may task the server. It is recommended that these queries be generated after normal school hours.

Filters built in the the Filter Designer display in HTML format. The HTML output allows for column sorting, filtering, grouping, and exporting to Excel or PDF.

The screenshot shows a table titled "sme dems Total Records: 23" with a link to "Simple HTML table". Below the link are "Export to Excel" and "Export to PDF" buttons. The table has columns: STUDENT.LASTNAME, STUDENT.FIRSTNAME, STUDENT.GENDER, and STUDENT.BIRTHDATE. A filter menu is open over the STUDENT.FIRSTNAME column, showing options: Sort Ascending, Sort Descending, Columns, and Filter. The Filter option is selected, and a sub-menu is open showing "Show items with value that:" with a dropdown menu set to "Benjamin". Below this, there is an "And" dropdown, "Is equal to" dropdown, and "Filter" and "Clear" buttons.

HTML Filter Display

To view the output in a simple HTML table, click the link at the top of the output. This displays the output without the ability to sort, group and organize the columns.

Query Wizard functionality allows users to easily create Ad hoc filters by organizing elements in a straightforward manner. Query Wizard filters are dynamic and always pull current information from the database based on the fields and filter options selected.

When using Custom Tab fields within Ad Hoc Query Wizard, all students are included in the results even if the student does not have a record within the custom dated tab. To exclude students without records for fields from a custom tab that is Table or List Element tab type, set the statusDate Operator to **IS NOT NULL**. When pulling in fields from a custom tab that is a Table or List Element tab type, Ad Hoc logic outputs every possible combination based on a specific date and time. The Table Tab Type stores specific times. The List Element Tab type always stores 12:00 AM. See the [Custom](#) article for more information.

Query Wizard Features

[Short and Long Filter Descriptions](#) | [Filter Operators](#) | [Logical Expressions](#) | [Functions](#) | [Output Formatting](#) | [Grouping and Aggregation Descriptions](#)

Short and Long Filter Descriptions

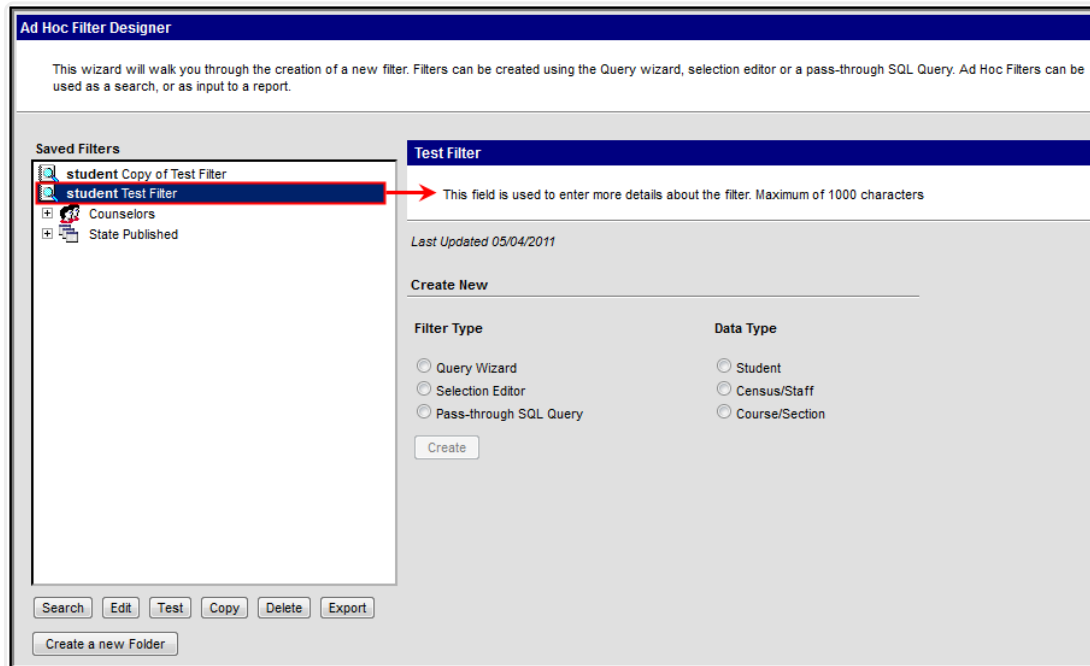
This provides additional information and context about the filter. It's displayed when a user selects that filter from the Saved Filters list and when the filter is being modified.

▶ [Click here to expand...](#)

To attach short and/or long descriptions to a filter, enter this information within the **Short Description** and **Long Description** text fields. To access the **Long Description** text box, select the (+) icon. Once the filter itself is saved, all descriptions entered are saved.

Once a filter has a long description entered and saved, this information displays on the Filter Designer editor when the filter is selected in the **Saved Filters** window. This is useful when determining what filter to use as well as communicating any important information about the filter prior to editing or making modifications. If a short description has been entered, this information displays when the cursor hovers over the filter within the **Saved Filters** window.

Both the Short and Long Descriptions display when a saved filter is edited/modified. Although the Long Description field appears locked, it can be modified by selecting the (+) icon.



Filter Description Display

Filter Operators

Filter operators allow users to set specific parameters per field within a filter. These parameters uniquely filter each field while maintaining the filter as a whole.

*Query Name:

Short Description:

Long Description:

Filter the data

ID	*Field	Operator	Value
1	<input type="text" value="sch.schoolID"/>	<input type="text"/>	<input type="text"/>
2	<input type="text" value="sch.districtID"/>	<input type="text"/>	<input type="text"/>
3	<input type="text" value="sch.number"/>	<input type="text"/>	<input type="text"/>

Logical Expression (Optional):

If logical expression is left blank, all filters are applied.
 Allowed symbols: AND OR NOT IN BETWEEN LIKE NOT LIKE SOUNDS LIKE CONTAINS STARTS WITH ENDS WITH
 Example Syntax: (1 AND (2 OR 3) NOT IN (4 OR 5 OR 6))

Save To: User Account Folder:

User Groups

Force Order ?

Users may apply multiple operators to the same field by clicking the **Add Filter** button and selecting a field. If a **Logical Expression** exists, all fields assigned an Operator must be included within the expression.

▶ [Click here to expand...](#)

The following table describes each available filter operators:

Operator	Results	Example
= (Equals)	Returns exact match of value.	student.grade=3 Only students in grade 3 are returned.

Operator	Results	Example
< > (Does not equal)	Returns results not equal to the value.	<p>student.gender < > M</p> <p>Students who have a Gender = F assigned on their Identities record or who do not have a value entered in the Grade field are returned.</p> <p>This operator allows NULL values.</p>
> (Greater than)	Returns results that are greater than the entered numeric value.	<p>student.age > 16</p> <p>All students older than 16 years of age are returned.</p>
> = (Greater than or equal to)	Returns results that are greater than or equal to the entered numeric value.	<p>student.age >= 16</p> <p>All students 16 years of age and older are returned.</p>
< (Less than)	Returns results that are less than the entered numeric value.	<p>student.age < 16</p> <p>All students under the age of 16 are returned.</p>
< = (Less than or equal to)	Returns results that are less than or equal to the entered numeric value.	<p>student.age <= 16</p> <p>All students 16 years of age and younger are returned.</p>
IN	Includes value.	<p>student.grade IN 9,10</p> <p>All students in 9th and 10 grade are returned.</p> <div style="background-color: #fff9c4; padding: 5px; border: 1px solid #ccc;"> <p>When using this format, do not put spaces after the comma.</p> </div>

Operator	Results	Example
NOT IN	Excludes value.	<p>student.grade NOT IN 11,12</p> <p>All students not in 11th or 12th grade are returned.</p> <p>This operator allows NULL values.</p> <div style="background-color: #fff9c4; padding: 5px; border: 1px solid #ccc;"> <p>When using this format, do not put spaces after the comma.</p> </div>
BETWEEN	<p>Filters data between two specified values. Works with numbers, dates and strings.</p> <p>If a date field is selected, the following options are available:</p> <ul style="list-style-type: none"> • DATE - Returns data based on the specified date range (where the starting date is sub-option 1 and the ending date is sub-option 2). • TODAY - Filters data based on dates that occur from a specific date through today or vice versa. • TOMORROW - Filters data based on dates that occur from a specific date through tomorrow or vice versa. • YESTERDAY - Filters data based on dates that occur from a specific date through yesterday or vice versa. • DAYS BEFORE - Filters data based on the number of days (sub-option 1) prior to sub-option 2 through sub-option 2. • MONTHS BEFORE - Filters data based on the number of months (sub-option 1) prior to sub-option 2 through sub-option 2. • DAYS AFTER - Filters data based on sub-option 1 through the number of days (sub-option 	<p>For BETWEEN: student.stateID BETWEEN 00001 THROUGH 100000.</p> <p>All students with a State ID between 00001 - 100000 are returned.</p> <p>For DATE: student.birthDate BETWEEN DATE 10151995 THROUGH DATE 10152010.</p> <p>All students with a birth date between 10/15/1995 - 10/15/2010 are returned.</p> <p>For TODAY: student.startDate BETWEEN TODAY THROUGH TODAY.</p> <p>All students who began an enrollment in the school today (current date) are returned.</p> <p>For YESTERDAY: student.startDate BETWEEN YESTERDAY THROUGH DATE 10152010.</p> <p>All students who began an enrollment in the school yesterday through 10/15/2010 are returned.</p> <p>For DAYS BEFORE: student.startDate BETWEEN DAYS BEFORE 4 THROUGH YESTERDAY.</p> <p>All students who began an enrollment in the school 4 days before yesterday through yesterday are returned.</p> <p>For MONTHS BEFORE: student.startDate</p>

Operator	Results	Example
	<p>2) after the sub-option 1 date.</p> <ul style="list-style-type: none"> • MONTHS AFTER - Filters data based on sub-option 1 through the number of months (sub-option 2) after the sub-option 1 date. 	<p>BETWEEN MONTHS BEFORE 5 THROUGH TODAY.</p> <p>All students who began an enrollment in the school 5 months prior to today through today are returned.</p> <p>For DAYS AFTER: student.startDate BETWEEN DATE 10152010 THROUGH DAYS AFTER 5.</p> <p>All students who began an enrollment in the school on 10/15/2010 through 10/20/2010 (5 days after) are returned.</p> <p>For MONTHS AFTER: student.startDate BETWEEN DATE 10152010 THROUGH MONTHS AFTER 5.</p> <p>All students who began enrolling in the school on 10/15/2010 through 3/15/2011 (five months after) are returned.</p>
IS CURRENT USER	Returns the current user's ID.	<p>Learner Plan Manager</p> <p>Setting learningPlan.planManagerPersonID IS CURRENT USER reports the current user's ID, along with data only applicable to that user.</p> <p>Current Teacher Sections</p> <p>For courseSection.personID IS CURRENT USER limits the results to students in the current teacher's section. This is useful for a report of student birthdays with a homeroom, or a Spirit Squad Advisor who needs to make locker signs and needs a list of participants and locker information.</p>
LIKE	Searches for test string in the field.	<p>course LIKE hist</p> <p>All courses like History 101 are returned.</p>
NOT LIKE	Searches for test string and filters data that is not like the user-defined value.	<p>course NOT LIKE hist</p> <p>All courses not like Hist are returned.</p> <p>This operator allows NULL values.</p>

Operator	Results	Example
SOUNDS LIKE	Uses a database function to return names with similar sound patterns.	student.lastName SOUNDS LIKE Ball Names such as "Ball," "Bell" and "Boll" are returned.
CONTAINS	Searches for strings that include the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry CONTAINS Cana All students with a Birth Country that contains "Cana" are returned.
STARTS WITH	Searches for strings that begin with the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry STARTS WITH Mexi All students with a Birth Country that begins with "Mexi" are returned.
ENDS WITH	Searches for strings that end with the same data entered by the user in the field. Any string that does not contain the user-defined value is filtered out. Any wildcard characters entered are treated as standard SQL wildcards.	student.birthCountry ENDS WITH many All students with a Birth Country that ends with "many" are returned.
IS NULL	Returns fields that are completely NULL (0 is considered a value).	student.stateID IS NULL All students who do not have a state ID are returned.
IS NOT NULL	Returns all fields that are not NULL (0 is considered a value).	student.ssn IS NOT NULL All students who do not have a stateID are returned.
IS TODAY	Returns result dates as the current date.	start.date IS TODAY Entries where the start.date is the current date are returned.
IS YESTERDAY	Returns result dates as of yesterday's date.	start.date IS YESTERDAY Results for one day previous to the current date are returned.

Operator	Results	Example
IS TOMORROW	Returns result dates as of tomorrow's date.	end.date IS TOMORROW Results for one day after the current date are returned.
IN THE MONTH	Returns all database field data for the month entered. This operator allows both numbered dates and spelled-out dates (e.g., 10 or October). It also allows for both upper and lower case letters. If spelling out a month, users must enter at least the first three characters (e.g., Oct for October).	employment.districtStartDate IN THE MONTH October All employees who have a district employment Start Date within the month of October are returned. This operator does not look at the Year or Calendar selected in the Campus toolbar. All historical and current district employment records with a Start Date in October are returned.
=TRUE	Returns checkbox values of "true" (checkbox is marked)	enrollment.stateExclude = TRUE All students with the State Exclude checkbox marked on their enrollment records are returned.
=FALSE	Returns checkbox values of "false" (checkbox is not marked)	enrollment.stateExclude = FALSE All students who do not have the State Exclude checkbox marked on their enrollment records are returned.

In addition to the options above, wildcard searching is also available. The following is a list of options:

Wildcard or Pattern	SQL Meaning	Standard Examples
%	0 or more characters	Entering the word <i>Man</i> returns the same results when entering <i>Man%</i> . <i>%son</i> finds names that end in <i>-son</i> : Johnson, Manson, Jason-Benson, etc.
_ (underscore)	One character	<i>Olson_Zierke</i> and <i>Olson Sierke</i> return the same results. <i>L__</i> (with two underscores) does not look only for 3-character names that start with <i>L</i> , but <i>_L_e_</i> finds names where <i>L</i> is the first and <i>e</i> the third character (e.g. <i>Lee</i> , <i>Luewenhook</i>). If the three underscores are entered at the end of a name, like <i>Dan___</i> , results list names with three additional letters (<i>Daniel</i>).

Wildcard or Pattern	SQL Meaning	Standard Examples
[token]	A range of possible characters	<i>L[ae]</i> finds names that start with <i>La</i> or <i>Le</i> .
,James	No SQL wildcard	Searches for first name equal to or beginning with James. This can only be used in the Quick Search fields.
Gonzales-Uribe	Compound name	Finds that last name. This returns compound names regardless of whether they are linked by a space or hyphen.
Gonzales Uribe or Gonzales_uribe or Gonzales%uribe	A compound name with a space.	Finds the name with or without a space or hyphen. Try wildcards if there is a space between the compound names.

Users can also use the following combinations when using the *Like* operator:

Wildcard or Pattern	SQL Meaning	Standard Examples
%	0 or more characters	<i>L%</i> finds names that start with L <i>L</i> finds names that contain an L <i>LAN</i> finds names containing LAN (Blanko, Landesburg, Blankenship, etc.)
_ (underscore)	One character	<i>L__</i> (two underscores) finds <i>Lee</i> and <i>Lor</i> , not <i>Luewenhook</i> .
[token]	A range of possible characters	<i>L[ae]%</i> finds names that start with <i>La</i> or <i>Le</i> .
^	Negation of token	<i>L[Query Wizard^ae]</i> finds names that do not start with <i>La</i> or <i>Le</i> .

Rules for Operators by Data Type

The following table describes all rules for allowing or disallowing operators by data type, where Y = Allowed, N = Not Allowed, and D = Depends on Field.

Option	Number	Float	String	Date	Text	Bit
>	Y	Y	Y	Y	Y	N

Option	Number	Float	String	Date	Text	Bit
>=	Y	Y	Y	Y	Y	N
<	Y	Y	Y	Y	Y	N
<=	Y	Y	Y	Y	Y	N
< >	Y	Y	Y	Y	Y	N
=	Y	Y	Y	Y	Y	N
IS NULL	D	D	D	D	D	N
IS NOT NULL	D	D	D	D	D	N
BETWEEN	Y	Y	Y	Y	Y	N
IS TODAY	N	N	N	Y	N	N
IS YESTERDAY	N	N	N	Y	N	N
IS TOMORROW	N	N	N	Y	N	N
IN	Y	Y	Y	Y	Y	N
NOT IN	Y	Y	Y	Y	Y	N
LIKE	N	N	Y	N	N	N
STARTS WITH	N	N	Y	N	N	N
ENDS WITH	N	N	Y	N	N	N
CONTAINS	N	N	Y	N	N	N
SOUNDS LIKE	N	N	Y	N	N	N
=TRUE	N	N	N	N	N	Y
=FALSE	N	N	N	N	N	Y

Use a Field as an Operator Value

Depending on the operator chosen for the field, a field may be used as an operator's value allowing a comparison between two fields. Logic only allows fields of the same data type to be used as the Operator's Value. For example, date fields are allowed to use other date fields as an operator value. When the appropriate operator is used, the Value column can act as a dropdown list while remaining static allowing the user to select a field or input a value. Deleting a field also removes it from the Value field, clearing out the operator for the field using it. Additionally, replacing a field with the [Element Replacement tool](#) replaces the field and the operator's value if the replaced field is being used as the value.

Ad Hoc Query Wizard - Filter Parameters

*Query Name: Fees charged during a student's enrollment

Short Description:

Long Description:

Filter the data

ID	*Field	Operator	Value
X 1	student.personID		<input type="text"/>
X 2	student.lastName		<input type="text"/>
X 3	student.firstName		<input type="text"/>
X 4	feeDetail.feeID		<input type="text"/>
X 5	feeDetail.feeType		<input type="text"/>
X 6	feeDetail.feeAmount		<input type="text"/>
X 7	activeEnrollment.startDate		<input type="text"/>
X 8	activeEnrollment.endDate		<input type="text"/>
X 9	feeDetail.dueDate	>=	activeEnrollment.startDate
X 10	feeDetail.dueDate	<=	activeEnrollment.endDate

Logical Expression (Optional):

If logical expression is left blank, all operators will be applied.
 Allowed symbols: AND OR NOT () IDs
 Example Syntax: (1 AND (2 OR 3) AND 4 AND (NOT 5 OR 6))

Save To: User Account
 Folder: /

User Groups

Force Order

In the above example, a query was set to report students with fees charged during enrollment. Using the fields activeEnrollment.startDate (7) and activeEnrollment.endDate (8) as operator values for feeDetail.dueDate (9, 10), the query reports students with fees due on or after the student's active enrollment start date AND on or before the student's active enrollment end date.

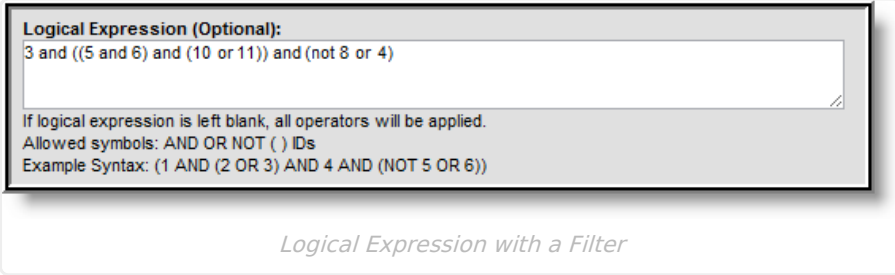
Operators Allowed to Use a Field as Values

Operator	Allowed
>	Y
>=	Y
<	Y
<=	Y
< >	Y
=	Y
BETWEEN	Y
IS CURRENT USER	N
IN THE MONTH OF	N
MONTHS BEFORE	N
DAYS BEFORE	N
IS NOT NULL	N
IS NULL	N
IS TODAY	N
IS YESTERDAY	N
IS TOMORROW	N
IN	N
NOT IN	N
LIKE	N
STARTS WITH	N
ENDS WITH	N
CONTAINS	N
SOUNDS LIKE	N
=TRUE	N
=FALSE	N

Logical Expressions

The Logical Expression field allows users to incorporate conditions between fields within a filter. This field effectively uses the OR, AND, and NOT conditions between fields and groups of fields.

- Only fields assigned an **Operator** are allowed to be included within logical expressions.
- Logical Expressions are created using the ID number associated with each field.



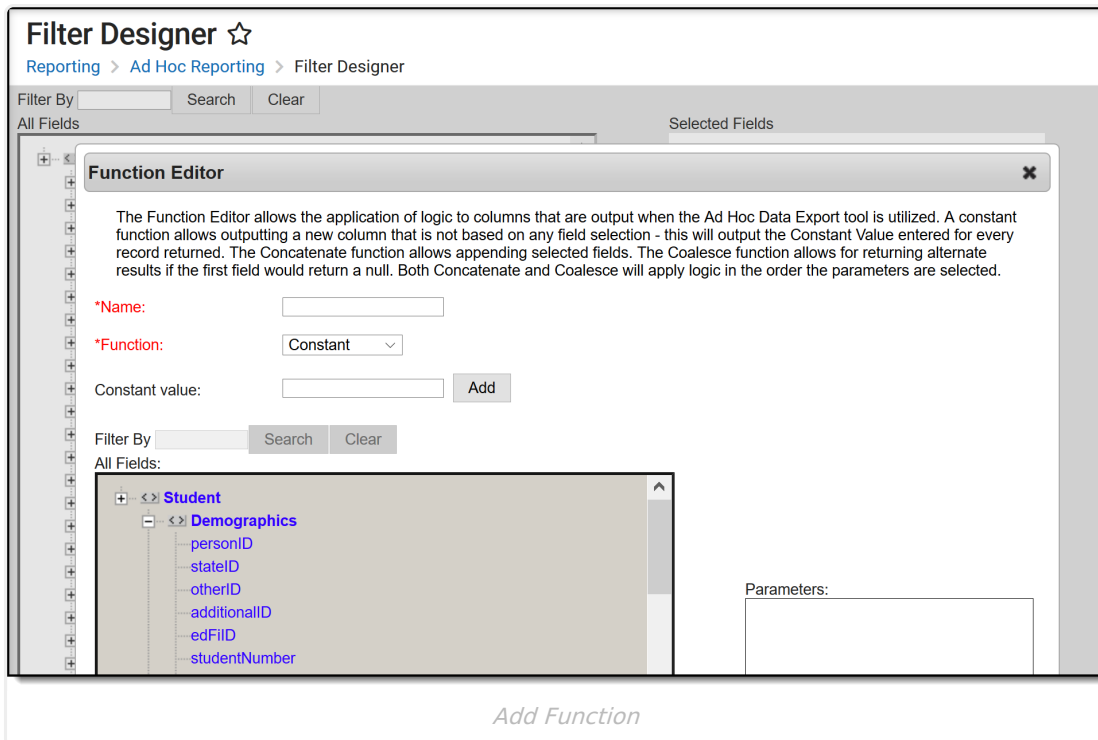
Logical Expression with a Filter

Logical expressions can be grouped using () symbols and the ID number to define the order in which the tool should include or exclude a person. In the example above, the () symbols indicate the tool should determine the student's End Date (5) and grade (6) and include these students depending on whether they are Asian (10) or White (11). This determination and group of students is then applied to the remaining parts of the logical expression.

Using () symbols is especially useful when using the OR condition, as users can include or exclude people based on whether or not they meet the criteria for the fields included within a group of fields. For example, students with a State ID less than 1000 (8) or an End Status populated (4) are not included in the remaining calculation for the logical expression.

Functions

Functions can be added to filters, which allow logic to be applied to field columns when the filter is generated via the Data Export tool. To add a function to a filter, select the **Add Function** button. The **Function Editor** appears in a new window.

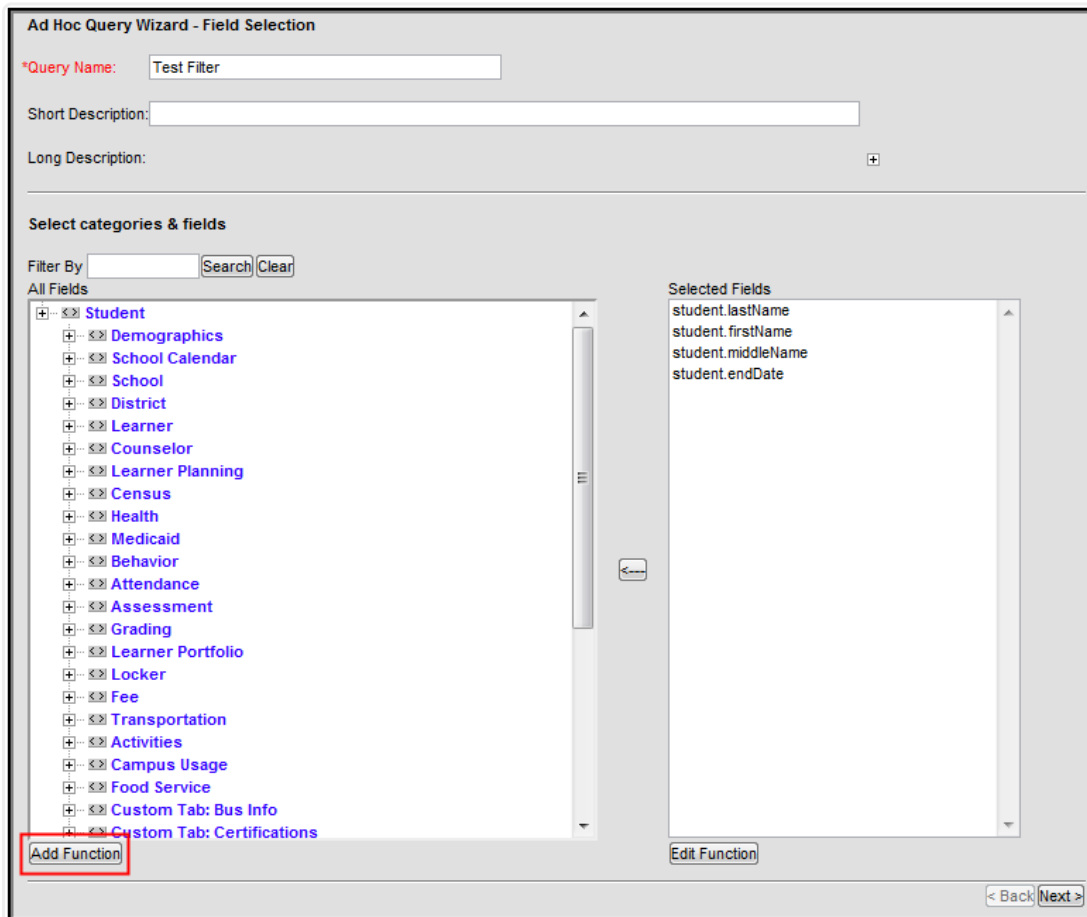


Add Function

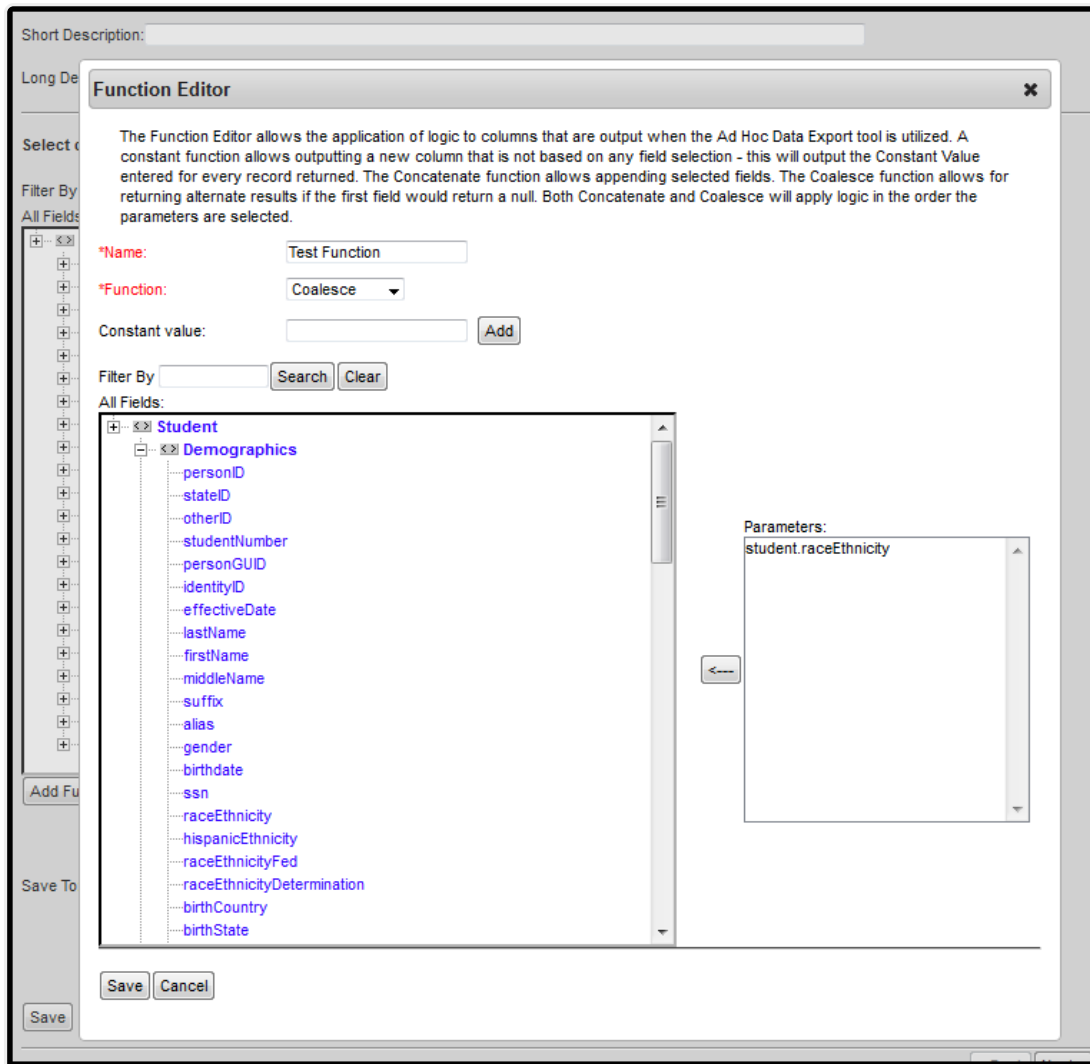
▶ [Click here to expand...](#)

Add Functions to Queries

1. Enter the **Name** of the function. This name differentiates the function from other functions within the **Selected Fields** window and on filters generated via the [Data Export](#) tool.
2. Select the desired **Function** from the dropdown list. The Function Descriptions section below provides descriptions and examples of each function. Once a function is selected, the **Filter By Search** field (see step 4) becomes active.
3. If the Constant **Function** is selected, enter the **Constant Value** and click the **Add** button. The value entered displays in the Parameters window and is reported on every record returned.
4. Use the **Filter By Search** field to search for desired fields. Entering a search value and clicking the **Search** button resets the list of fields to only return matching fields. Click the **Clear** button to remove entered search values and see the entire list of fields.
5. Select which fields to include within the function by clicking on each field within the **All Fields** window. Selected fields move into the **Parameters** window, indicating which fields have been added to the function.
6. Select the **Save** icon.

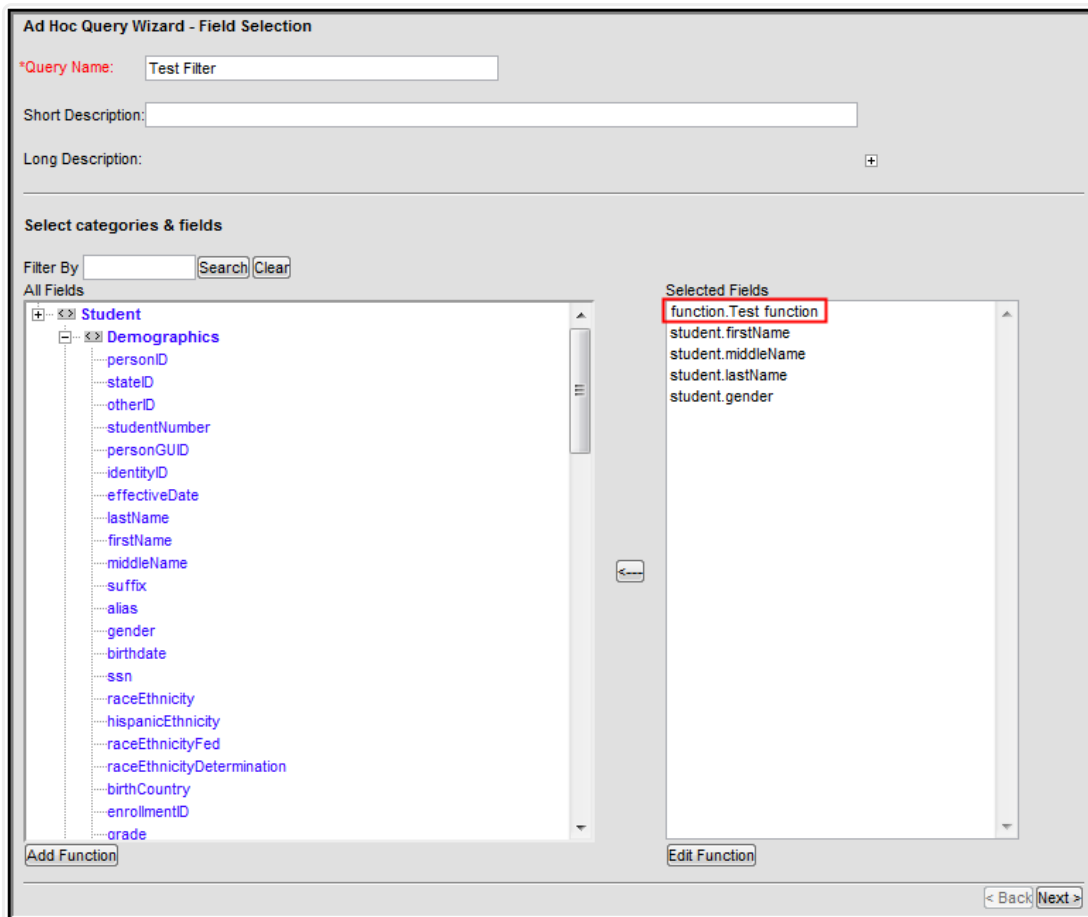


Adding a Function to a Filter



Enter Data into the Function Editor

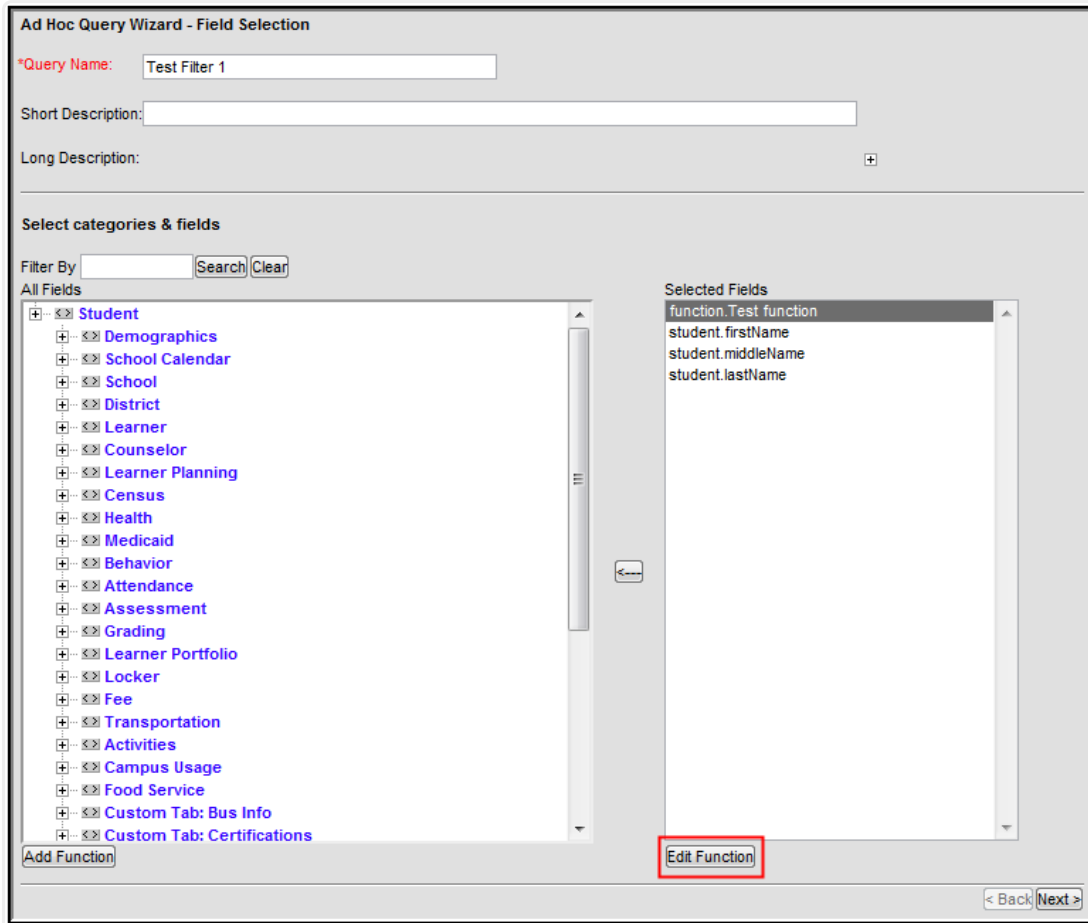
The **Field Selection** editor appears after saving the added function. Functions created and added to the filter are displayed in the Selected Fields window. The function's name always appears to the left of the period (*i.e.*, function.functionName).



View Functions Added to a Filter

Edit Functions

Existing functions can be edited by selecting the function within the **Selected Fields** window and clicking the **Edit Function** button.



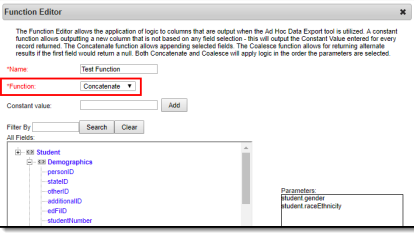
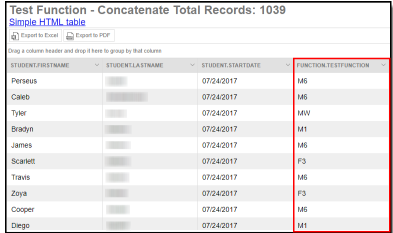


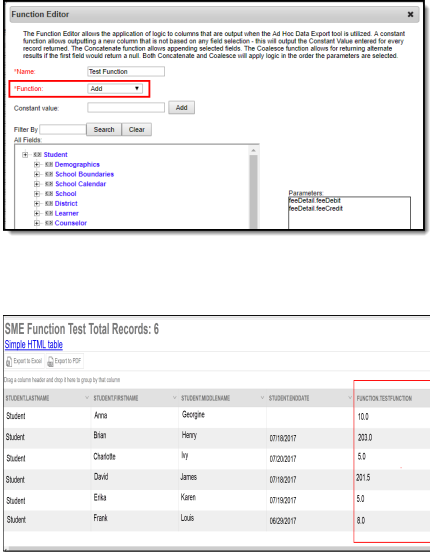
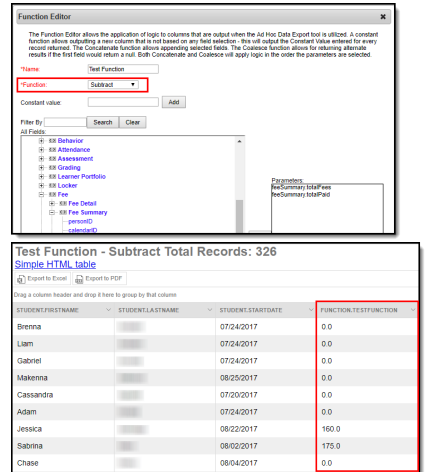
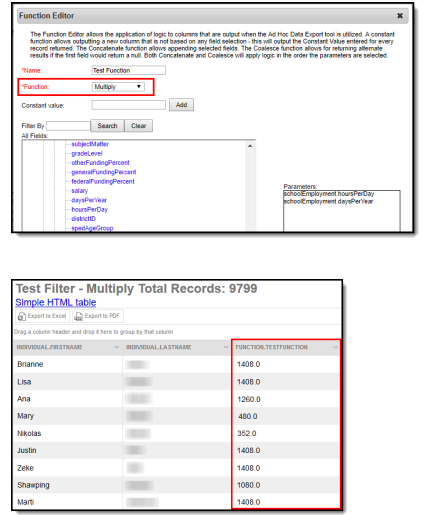
Edit Existing Functions

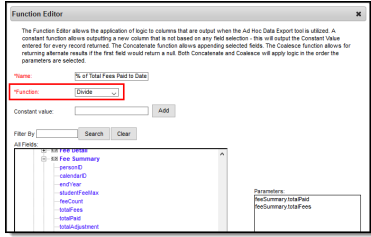
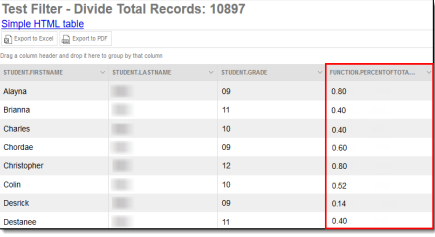
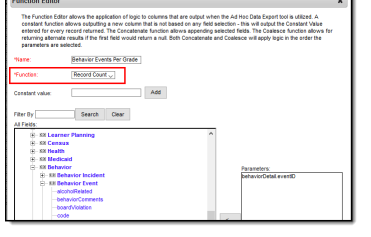
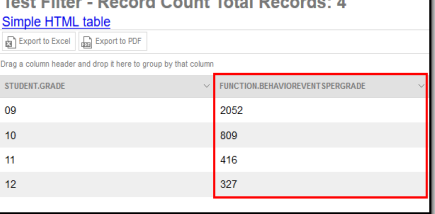
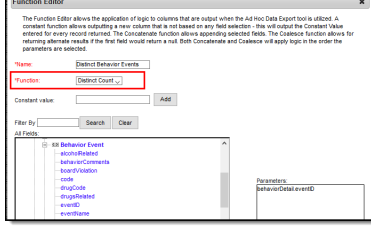
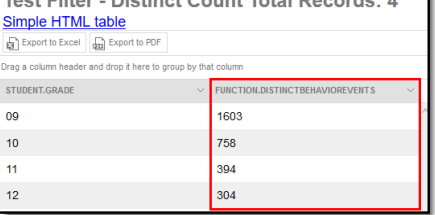
Function Descriptions

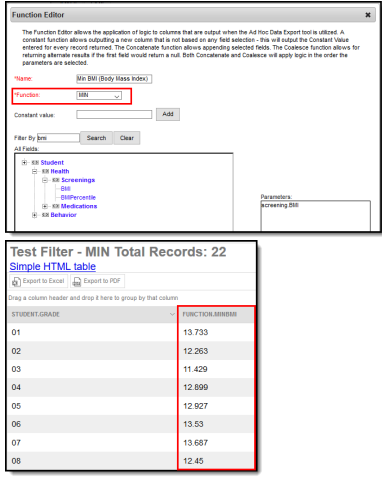
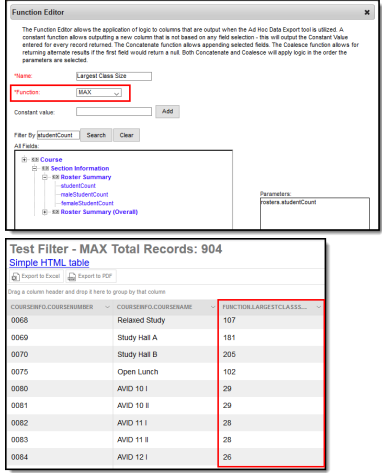
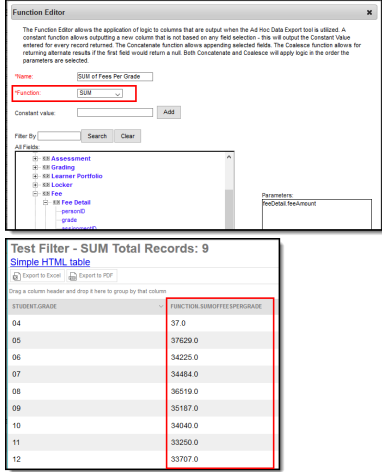
The following describes each available function.

Function	Description	Example
Constant	<p>The Constant function outputs the Constant Value entered on each record returned when the filter is exported.</p> <p>A Constant Value of 5 is entered and added to the filter in the examples to the right. When the filter is exported, a column is reported displaying the Constant Value entered.</p>	

Function	Description	Example																																												
<p>Coalesce</p>	<p>The Coalesce function allows users to define multiple fields where logic pulls the first field, and if NULL, the second field is pulled, and so on down the line of added fields until a value is found. Logic pulls field values in the order fields are selected in the Function Editor.</p> <p>A Coalesce function for Federal Race Ethnicity and Race Ethnicity Determination fields was added to the examples to the right. This means logic first pulls and reports student Federal Race Ethnicity field values and for any that are NULL, the student's Race Ethnicity Determination reports.</p> <p>When the filter is exported, the function reports field data within a specific column. Student Federal Race Ethnicity values are reported.</p>	 <p>The Function Editor shows the Coalesce function selected. Parameters include 'Student raceEthnicity' and 'Student raceEthnicityDetermination'.</p>  <table border="1"> <thead> <tr> <th>STUDENT.FIRSTNAME</th> <th>STUDENT.LASTNAME</th> <th>STUDENT.STARTDATE</th> <th>FUNCTION.TESTFUNCTION</th> </tr> </thead> <tbody> <tr><td>Perseus</td><td></td><td>07/24/2017</td><td>6</td></tr> <tr><td>Caleb</td><td></td><td>07/24/2017</td><td>6</td></tr> <tr><td>Tyler</td><td></td><td>07/24/2017</td><td>6</td></tr> <tr><td>Brady</td><td></td><td>07/24/2017</td><td>1</td></tr> <tr><td>James</td><td></td><td>07/24/2017</td><td>6</td></tr> <tr><td>Scarlett</td><td></td><td>07/24/2017</td><td>3</td></tr> <tr><td>Travis</td><td></td><td>07/24/2017</td><td>6</td></tr> <tr><td>Zoya</td><td></td><td>07/24/2017</td><td>3</td></tr> </tbody> </table>	STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION	Perseus		07/24/2017	6	Caleb		07/24/2017	6	Tyler		07/24/2017	6	Brady		07/24/2017	1	James		07/24/2017	6	Scarlett		07/24/2017	3	Travis		07/24/2017	6	Zoya		07/24/2017	3								
STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION																																											
Perseus		07/24/2017	6																																											
Caleb		07/24/2017	6																																											
Tyler		07/24/2017	6																																											
Brady		07/24/2017	1																																											
James		07/24/2017	6																																											
Scarlett		07/24/2017	3																																											
Travis		07/24/2017	6																																											
Zoya		07/24/2017	3																																											
<p>Concatenate</p>	<p>The Concatenate function allows field values to be appended when the filter is exported.</p> <p>In the example to the right, a Concatenate function for fields Gender and Race Ethnicity was added. When the filter is exported, field values are appended and reported. Student Gender values (M, F) are reported alongside student Race Ethnicity values.</p>	 <p>The Function Editor shows the Concatenate function selected. Parameters include 'Student gender' and 'Student raceEthnicity'.</p>  <table border="1"> <thead> <tr> <th>STUDENT.FIRSTNAME</th> <th>STUDENT.LASTNAME</th> <th>STUDENT.STARTDATE</th> <th>FUNCTION.TESTFUNCTION</th> </tr> </thead> <tbody> <tr><td>Perseus</td><td></td><td>07/24/2017</td><td>M6</td></tr> <tr><td>Caleb</td><td></td><td>07/24/2017</td><td>M6</td></tr> <tr><td>Tyler</td><td></td><td>07/24/2017</td><td>MW6</td></tr> <tr><td>Brady</td><td></td><td>07/24/2017</td><td>M1</td></tr> <tr><td>James</td><td></td><td>07/24/2017</td><td>M6</td></tr> <tr><td>Scarlett</td><td></td><td>07/24/2017</td><td>F3</td></tr> <tr><td>Travis</td><td></td><td>07/24/2017</td><td>M6</td></tr> <tr><td>Zoya</td><td></td><td>07/24/2017</td><td>F3</td></tr> <tr><td>Cooper</td><td></td><td>07/24/2017</td><td>M6</td></tr> <tr><td>Diego</td><td></td><td>07/24/2017</td><td>M1</td></tr> </tbody> </table>	STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION	Perseus		07/24/2017	M6	Caleb		07/24/2017	M6	Tyler		07/24/2017	MW6	Brady		07/24/2017	M1	James		07/24/2017	M6	Scarlett		07/24/2017	F3	Travis		07/24/2017	M6	Zoya		07/24/2017	F3	Cooper		07/24/2017	M6	Diego		07/24/2017	M1
STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION																																											
Perseus		07/24/2017	M6																																											
Caleb		07/24/2017	M6																																											
Tyler		07/24/2017	MW6																																											
Brady		07/24/2017	M1																																											
James		07/24/2017	M6																																											
Scarlett		07/24/2017	F3																																											
Travis		07/24/2017	M6																																											
Zoya		07/24/2017	F3																																											
Cooper		07/24/2017	M6																																											
Diego		07/24/2017	M1																																											

Function	Description	Example																																								
<p>Add</p> <p>The Add function allows field values to be added together to output a single result (i.e., field 1 + field 2).</p> <p>In the example to the right, Fee Debit is added to Fee Credit to generate a total balance. When the filter is exported, field values are added and reported as a single value.</p>		 <p>SME Function Test Total Records: 6</p> <table border="1"> <thead> <tr> <th>STUDENT.LASTNAME</th> <th>STUDENT.FIRSTNAME</th> <th>STUDENT.MIDDLENAME</th> <th>STUDENT.STARTDATE</th> <th>FUNCTION.TESTFUNCTION</th> </tr> </thead> <tbody> <tr> <td>Student</td> <td>Anne</td> <td>George</td> <td></td> <td>10.0</td> </tr> <tr> <td>Student</td> <td>Brian</td> <td>Henry</td> <td>07/18/2017</td> <td>203.0</td> </tr> <tr> <td>Student</td> <td>Chafote</td> <td>Ily</td> <td>07/20/2017</td> <td>5.0</td> </tr> <tr> <td>Student</td> <td>David</td> <td>James</td> <td>07/18/2017</td> <td>201.5</td> </tr> <tr> <td>Student</td> <td>Elka</td> <td>Karen</td> <td>07/18/2017</td> <td>5.0</td> </tr> <tr> <td>Student</td> <td>Frank</td> <td>Louis</td> <td>06/29/2017</td> <td>8.0</td> </tr> </tbody> </table>	STUDENT.LASTNAME	STUDENT.FIRSTNAME	STUDENT.MIDDLENAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION	Student	Anne	George		10.0	Student	Brian	Henry	07/18/2017	203.0	Student	Chafote	Ily	07/20/2017	5.0	Student	David	James	07/18/2017	201.5	Student	Elka	Karen	07/18/2017	5.0	Student	Frank	Louis	06/29/2017	8.0					
STUDENT.LASTNAME	STUDENT.FIRSTNAME	STUDENT.MIDDLENAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION																																						
Student	Anne	George		10.0																																						
Student	Brian	Henry	07/18/2017	203.0																																						
Student	Chafote	Ily	07/20/2017	5.0																																						
Student	David	James	07/18/2017	201.5																																						
Student	Elka	Karen	07/18/2017	5.0																																						
Student	Frank	Louis	06/29/2017	8.0																																						
<p>Subtract</p> <p>The Subtract function allows field values to be subtracted from each other to output a single result.</p> <p>In the example to the right, total Fees are subtracted from Total Paid to report a student's outstanding balance. When the filter is exported, field values are subtracted and reported as a single value.</p>		 <p>Test Function - Subtract Total Records: 326</p> <table border="1"> <thead> <tr> <th>STUDENT.FIRSTNAME</th> <th>STUDENT.LASTNAME</th> <th>STUDENT.STARTDATE</th> <th>FUNCTION.TESTFUNCTION</th> </tr> </thead> <tbody> <tr> <td>Breanna</td> <td></td> <td>07/24/2017</td> <td>0.0</td> </tr> <tr> <td>Liam</td> <td></td> <td>07/24/2017</td> <td>0.0</td> </tr> <tr> <td>Gabriel</td> <td></td> <td>07/24/2017</td> <td>0.0</td> </tr> <tr> <td>Makenna</td> <td></td> <td>08/29/2017</td> <td>0.0</td> </tr> <tr> <td>Cassandra</td> <td></td> <td>07/20/2017</td> <td>0.0</td> </tr> <tr> <td>Adam</td> <td></td> <td>07/24/2017</td> <td>0.0</td> </tr> <tr> <td>Jessica</td> <td></td> <td>08/22/2017</td> <td>160.0</td> </tr> <tr> <td>Sabrina</td> <td></td> <td>08/02/2017</td> <td>175.0</td> </tr> <tr> <td>Chase</td> <td></td> <td>08/04/2017</td> <td>0.0</td> </tr> </tbody> </table>	STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION	Breanna		07/24/2017	0.0	Liam		07/24/2017	0.0	Gabriel		07/24/2017	0.0	Makenna		08/29/2017	0.0	Cassandra		07/20/2017	0.0	Adam		07/24/2017	0.0	Jessica		08/22/2017	160.0	Sabrina		08/02/2017	175.0	Chase		08/04/2017	0.0
STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.STARTDATE	FUNCTION.TESTFUNCTION																																							
Breanna		07/24/2017	0.0																																							
Liam		07/24/2017	0.0																																							
Gabriel		07/24/2017	0.0																																							
Makenna		08/29/2017	0.0																																							
Cassandra		07/20/2017	0.0																																							
Adam		07/24/2017	0.0																																							
Jessica		08/22/2017	160.0																																							
Sabrina		08/02/2017	175.0																																							
Chase		08/04/2017	0.0																																							
<p>Multiply</p> <p>The Multiply function allows field values to be multiplied together to output a single result (i.e., field 1 x field 2).</p> <p>In the right-hand example, employee hours per day are multiplied by the number of days employed for the year. When the filter is exported, field values are multiplied and reported as a single value.</p>		 <p>Test Filter - Multiply Total Records: 9799</p> <table border="1"> <thead> <tr> <th>INDIVIDUAL.FIRSTNAME</th> <th>INDIVIDUAL.LASTNAME</th> <th>FUNCTION.TESTFUNCTION</th> </tr> </thead> <tbody> <tr> <td>Brianne</td> <td></td> <td>1408.0</td> </tr> <tr> <td>Lisa</td> <td></td> <td>1408.0</td> </tr> <tr> <td>Ana</td> <td></td> <td>1260.0</td> </tr> <tr> <td>Mary</td> <td></td> <td>480.0</td> </tr> <tr> <td>Nikolas</td> <td></td> <td>352.0</td> </tr> <tr> <td>Justin</td> <td></td> <td>1408.0</td> </tr> <tr> <td>Zeke</td> <td></td> <td>1408.0</td> </tr> <tr> <td>Shaoping</td> <td></td> <td>1080.0</td> </tr> <tr> <td>Marti</td> <td></td> <td>1408.0</td> </tr> </tbody> </table>	INDIVIDUAL.FIRSTNAME	INDIVIDUAL.LASTNAME	FUNCTION.TESTFUNCTION	Brianne		1408.0	Lisa		1408.0	Ana		1260.0	Mary		480.0	Nikolas		352.0	Justin		1408.0	Zeke		1408.0	Shaoping		1080.0	Marti		1408.0										
INDIVIDUAL.FIRSTNAME	INDIVIDUAL.LASTNAME	FUNCTION.TESTFUNCTION																																								
Brianne		1408.0																																								
Lisa		1408.0																																								
Ana		1260.0																																								
Mary		480.0																																								
Nikolas		352.0																																								
Justin		1408.0																																								
Zeke		1408.0																																								
Shaoping		1080.0																																								
Marti		1408.0																																								

Function	Description	Example																																				
<p>Divide</p>	<p>The Divide function allows a field or more fields to be divided and output into a single result (i.e., field 1 / field 2).</p> <p>In the example to the right, the total number of fees is divided by the total amount of fees paid to get the percentage of total fees paid to date. If applicable, decimal places are included in the output.</p> <p>When the filter is exported, field values are divided and reported as a single value.</p>	 <p>The Function Editor shows the configuration for the 'Divide' function. The function name is '% of Total Fees Paid to Date'. The function is set to 'Divide'. The constant value is empty. The filter is set to 'All Fees'. The parameters are 'FeeSummaryInfoPaid' and 'FeeSummaryInfoFees'.</p>  <p>The Test Filter shows the results of the Divide function. The table has columns for Student First Name, Student Last Name, Student Grade, and Function Percent of Total. The results are as follows:</p> <table border="1"> <thead> <tr> <th>STUDENT.FIRSTNAME</th> <th>STUDENT.LASTNAME</th> <th>STUDENT.GRADE</th> <th>FUNCTION.PERCENTOFTOTAL</th> </tr> </thead> <tbody> <tr> <td>Alayna</td> <td></td> <td>09</td> <td>0.80</td> </tr> <tr> <td>Bhanna</td> <td></td> <td>11</td> <td>0.40</td> </tr> <tr> <td>Charles</td> <td></td> <td>10</td> <td>0.40</td> </tr> <tr> <td>Chordae</td> <td></td> <td>09</td> <td>0.60</td> </tr> <tr> <td>Christopher</td> <td></td> <td>12</td> <td>0.80</td> </tr> <tr> <td>Colin</td> <td></td> <td>10</td> <td>0.52</td> </tr> <tr> <td>Desrick</td> <td></td> <td>09</td> <td>0.14</td> </tr> <tr> <td>Destanee</td> <td></td> <td>11</td> <td>0.40</td> </tr> </tbody> </table>	STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.GRADE	FUNCTION.PERCENTOFTOTAL	Alayna		09	0.80	Bhanna		11	0.40	Charles		10	0.40	Chordae		09	0.60	Christopher		12	0.80	Colin		10	0.52	Desrick		09	0.14	Destanee		11	0.40
STUDENT.FIRSTNAME	STUDENT.LASTNAME	STUDENT.GRADE	FUNCTION.PERCENTOFTOTAL																																			
Alayna		09	0.80																																			
Bhanna		11	0.40																																			
Charles		10	0.40																																			
Chordae		09	0.60																																			
Christopher		12	0.80																																			
Colin		10	0.52																																			
Desrick		09	0.14																																			
Destanee		11	0.40																																			
<p>Record Count</p>	<p>The Record Count function allows users to report a record count for the field selected. In the example to the right, a record count of behavior events is used to report a count of behavior events per grade level.</p> <p>When the filter is exported, a record count of the field is calculated and reported.</p>	 <p>The Function Editor shows the configuration for the 'Record Count' function. The function name is 'Behavior Events Per Grade'. The function is set to 'Record Count'. The constant value is empty. The filter is set to 'All Events'. The parameters are 'BehaviorEventID'.</p>  <p>The Test Filter shows the results of the Record Count function. The table has columns for Student Grade and Function Behavior Event Sp/Grade. The results are as follows:</p> <table border="1"> <thead> <tr> <th>STUDENT.GRADE</th> <th>FUNCTION.BEHAVIOREVENT.SP/GRADE</th> </tr> </thead> <tbody> <tr> <td>09</td> <td>2052</td> </tr> <tr> <td>10</td> <td>809</td> </tr> <tr> <td>11</td> <td>416</td> </tr> <tr> <td>12</td> <td>327</td> </tr> </tbody> </table>	STUDENT.GRADE	FUNCTION.BEHAVIOREVENT.SP/GRADE	09	2052	10	809	11	416	12	327																										
STUDENT.GRADE	FUNCTION.BEHAVIOREVENT.SP/GRADE																																					
09	2052																																					
10	809																																					
11	416																																					
12	327																																					
<p>Distinct Count</p>	<p>The Distinct function allows users to report a distinct count for the field selected. In the example to the right, a distinct count of behavior events is used to report the distinct count of behavior events per grade level.</p> <p>When the filter is exported, a record count of the field is calculated and reported</p>	 <p>The Function Editor shows the configuration for the 'Distinct Count' function. The function name is 'Distinct Behavior Events'. The function is set to 'Distinct Count'. The constant value is empty. The filter is set to 'All Events'. The parameters are 'BehaviorEventID'.</p>  <p>The Test Filter shows the results of the Distinct Count function. The table has columns for Student Grade and Function Distinct Behavior Events. The results are as follows:</p> <table border="1"> <thead> <tr> <th>STUDENT.GRADE</th> <th>FUNCTION.DISTINCTBEHAVIOREVENTS</th> </tr> </thead> <tbody> <tr> <td>09</td> <td>1603</td> </tr> <tr> <td>10</td> <td>758</td> </tr> <tr> <td>11</td> <td>394</td> </tr> <tr> <td>12</td> <td>304</td> </tr> </tbody> </table>	STUDENT.GRADE	FUNCTION.DISTINCTBEHAVIOREVENTS	09	1603	10	758	11	394	12	304																										
STUDENT.GRADE	FUNCTION.DISTINCTBEHAVIOREVENTS																																					
09	1603																																					
10	758																																					
11	394																																					
12	304																																					

Function	Description	Example																														
MIN	<p>The MIN function allows users to report the minimum value for a field.</p> <p>In the example to the right, the MIN BMI (Body Mass Index) per grade level.</p> <p>When the filter is exported, the MIN of the field is calculated and reported.</p>	 <p>The Function Editor shows the function 'MIN' applied to the field 'BMI'. The Test Filter shows the results for grades 01 through 08, with the minimum BMI value for each grade highlighted in red.</p> <table border="1"> <thead> <tr> <th>STUDENT GRADE</th> <th>FUNCTION:MIN(BMI)</th> </tr> </thead> <tbody> <tr><td>01</td><td>13.733</td></tr> <tr><td>02</td><td>12.263</td></tr> <tr><td>03</td><td>11.429</td></tr> <tr><td>04</td><td>12.609</td></tr> <tr><td>05</td><td>12.927</td></tr> <tr><td>06</td><td>13.53</td></tr> <tr><td>07</td><td>13.687</td></tr> <tr><td>08</td><td>12.45</td></tr> </tbody> </table>	STUDENT GRADE	FUNCTION:MIN(BMI)	01	13.733	02	12.263	03	11.429	04	12.609	05	12.927	06	13.53	07	13.687	08	12.45												
STUDENT GRADE	FUNCTION:MIN(BMI)																															
01	13.733																															
02	12.263																															
03	11.429																															
04	12.609																															
05	12.927																															
06	13.53																															
07	13.687																															
08	12.45																															
MAX	<p>The MAX function allows users to report the maximum value for a field.</p> <p>In the example to the right, the MAX student count is used as the function to report the largest class size per course.</p> <p>When the filter is exported, the MAX of the field is calculated and reported.</p>	 <p>The Function Editor shows the function 'MAX' applied to the field 'studentCount'. The Test Filter shows the results for various courses, with the maximum student count for each course highlighted in red.</p> <table border="1"> <thead> <tr> <th>COURSEID/COURSENUMBER</th> <th>COURSE/COURSENAME</th> <th>FUNCTION:MAX(STUDENTCOUNT)</th> </tr> </thead> <tbody> <tr><td>0068</td><td>Released Study</td><td>107</td></tr> <tr><td>0069</td><td>Study Hall A</td><td>181</td></tr> <tr><td>0070</td><td>Study Hall B</td><td>265</td></tr> <tr><td>0075</td><td>Open Lunch</td><td>102</td></tr> <tr><td>0090</td><td>AVID 10 I</td><td>29</td></tr> <tr><td>0091</td><td>AVID 10 B</td><td>29</td></tr> <tr><td>0092</td><td>AVID 11 I</td><td>28</td></tr> <tr><td>0093</td><td>AVID 11 B</td><td>28</td></tr> <tr><td>0094</td><td>AVID 12 I</td><td>28</td></tr> </tbody> </table>	COURSEID/COURSENUMBER	COURSE/COURSENAME	FUNCTION:MAX(STUDENTCOUNT)	0068	Released Study	107	0069	Study Hall A	181	0070	Study Hall B	265	0075	Open Lunch	102	0090	AVID 10 I	29	0091	AVID 10 B	29	0092	AVID 11 I	28	0093	AVID 11 B	28	0094	AVID 12 I	28
COURSEID/COURSENUMBER	COURSE/COURSENAME	FUNCTION:MAX(STUDENTCOUNT)																														
0068	Released Study	107																														
0069	Study Hall A	181																														
0070	Study Hall B	265																														
0075	Open Lunch	102																														
0090	AVID 10 I	29																														
0091	AVID 10 B	29																														
0092	AVID 11 I	28																														
0093	AVID 11 B	28																														
0094	AVID 12 I	28																														
SUM	<p>The SUM function adds the value or field selected over all other aggregated fields.</p> <p>In the example to the right, the SUM of fee amounts is used to report the SUM of fees per grade.</p> <p>When the filter is exported, the SUM field is calculated and reported.</p>	 <p>The Function Editor shows the function 'SUM' applied to the field 'fee'. The Test Filter shows the results for grades 04 through 12, with the sum of fees for each grade highlighted in red.</p> <table border="1"> <thead> <tr> <th>STUDENT GRADE</th> <th>FUNCTION:SUM(FEE)</th> </tr> </thead> <tbody> <tr><td>04</td><td>37.0</td></tr> <tr><td>05</td><td>37629.0</td></tr> <tr><td>06</td><td>34225.0</td></tr> <tr><td>07</td><td>34484.0</td></tr> <tr><td>08</td><td>36519.0</td></tr> <tr><td>09</td><td>35187.0</td></tr> <tr><td>10</td><td>34040.0</td></tr> <tr><td>11</td><td>33256.0</td></tr> <tr><td>12</td><td>33707.0</td></tr> </tbody> </table>	STUDENT GRADE	FUNCTION:SUM(FEE)	04	37.0	05	37629.0	06	34225.0	07	34484.0	08	36519.0	09	35187.0	10	34040.0	11	33256.0	12	33707.0										
STUDENT GRADE	FUNCTION:SUM(FEE)																															
04	37.0																															
05	37629.0																															
06	34225.0																															
07	34484.0																															
08	36519.0																															
09	35187.0																															
10	34040.0																															
11	33256.0																															
12	33707.0																															

Function	Description	Example																						
AVG	<p>The AVG function allows users to report the average value for a field.</p> <p>In the example to the right, the AVG of roster student count is used to report the average class size per department.</p> <p>When the filter is exported, the AVG field is calculated and reported.</p>	<p>The Function Editor shows the configuration for the AVG function. The Name is 'AVG Class Size', the Function is 'AVG', and the Constant Value is empty. The Filter is 'StudentCount'. The Test Filter shows the results of the AVG function applied to the 'StudentCount' field across various departments.</p> <table border="1"> <thead> <tr> <th>COUSERIDDEPARTMENTNAME</th> <th>FUNCTION.AVGCLASSSIZE</th> </tr> </thead> <tbody> <tr><td>Business</td><td>17</td></tr> <tr><td>Counseling</td><td>11</td></tr> <tr><td>Electives</td><td>14</td></tr> <tr><td>ELL</td><td>14</td></tr> <tr><td>English</td><td>25</td></tr> <tr><td>Family Consumer Science</td><td>21</td></tr> <tr><td>Global Language</td><td>22</td></tr> <tr><td>Health</td><td>32</td></tr> <tr><td>Math</td><td>25</td></tr> <tr><td>Music</td><td>36</td></tr> </tbody> </table>	COUSERIDDEPARTMENTNAME	FUNCTION.AVGCLASSSIZE	Business	17	Counseling	11	Electives	14	ELL	14	English	25	Family Consumer Science	21	Global Language	22	Health	32	Math	25	Music	36
COUSERIDDEPARTMENTNAME	FUNCTION.AVGCLASSSIZE																							
Business	17																							
Counseling	11																							
Electives	14																							
ELL	14																							
English	25																							
Family Consumer Science	21																							
Global Language	22																							
Health	32																							
Math	25																							
Music	36																							

Output Formatting

The Output Formatting editor lets users control how each field is reported and displayed when exported.

*Query Name:

Short Description:

Long Description:

Format the output file/report

Output distinct records

Field	OutputSeq	Sort	Direction	Column Header	Alignment	Formatting	Length
student.raceEthnicityDetermination	<input checked="" type="checkbox"/>	1	1	Ascend			
student.birthCountry	<input checked="" type="checkbox"/>	2					
student.birthState	<input checked="" type="checkbox"/>	3					

Save To: User Account
Folder:

User Groups

Force Order [?](#)

▶ [Click here to expand...](#)

Output Formatting Descriptions

Field	Description
Output distinct records	<p>If marked, data is outputted in unduplicated records based on field values. The following is an example of a filter containing student first name, last name, grade, gender, and behavior event type:</p> <ul style="list-style-type: none"> • If a student has three behavior events for the same behavior event type and the Output distinct records checkbox is not marked, the student reports three records. • If the Output distinct records checkbox is marked, the same student now only reports one record.
Field	Fields selected from the All Fields window in the previous screen.
Output	This checkbox determines whether or not the field is included in outputted data. Deselecting this checkbox means data is still filtered and reported for this field and operators but not included in the output.
Seq	This field determines the sequence of outputted data.
Sort	This field determines the sort order of outputted field data.
Direction	This field determines if data is sorted ascending or descending. This field is only available if a value is entered in the Sort field.
Column Header	This field determines what header is displays for the field on exported files. Users are encouraged to enter a logical and easily identifiable column header for each field, as leaving the field blank results in the field name (i.e., student.stateID) being reported.
Alignment	The field determines how field data is aligned on files exported. Available options include: Left, Center and Right.

Field	Description
Formatting	<p>The field determines how values are reported for the field when used in reports and exported files. Formatting options are important for filters used with reports which require specific formatting in order for the file to be correctly submitted to an entity or system.</p> <p>The following formatting options are available:</p> <ul style="list-style-type: none"> • Zero Pad - numbers are padded with zeros to the left (<i>i.e.</i>, 444 zero padded becomes 000444) • Space Fill - values are filled with spaces in order to reach required field length • Upper Case - values are reported entirely in uppercase (<i>i.e.</i>, Course is reported COURSE). This option is only available for text, char and varchar fields. • Lower Case - values are reported entirely in lowercase (<i>i.e.</i>, Course is reported course). This option is only available for text, char and varchar fields. • MM/DD/YYYY • MM-DD-YYYY • MMDDYYYY • YYYY/MM/DD • YYYY-MM-DD • YYYYMMDD • YYYY • YYYY/MM • YYYY-MM • YYYYMM • MM/YYYY • MM-YYYY • MMYYYY • MM/DD/YYYY hh:mm AM • MM-DD-YYYY hh:mm AM • YYYYMMDDHHmm - This is similar to military time (<i>e.g.</i>, 1:00PM is 1300) because there is no AM/PM. • 1, 234.5; - 1,234.5 • 1,234.5; (1,234.5) • \$1,234.00; -\$1,234.00 • \$1,234.00; (\$1,234.00) • Y/N - Used with bit fields. If bit field is checked, Y is reported. If field is unchecked, N is reported. • YES/NO - Used with bit fields. If bit field is checked, YES is reported. If field is unchecked, NO is reported. • T/F - Used with bit fields. If bit field is checked, T is reported. If field is unchecked, F is reported. • TRUE/FALSE - Used with bit fields. If bit field is checked, TRUE is reported. If field is unchecked, FALSE is reported. • 1/0 - Used with bit fields. If bit field is checked, 1 is reported. If field is unchecked, 0 is reported.

Field	Description
Length	<p>This field determines the length of the column in the exported data file. This is the maximum amount of characters allowed to be reported in the column. Data which exceeds the defined length is truncated on the right side. Zero padding is added to the left of a value. Space filling is added to the right of a value.</p> <p>A length must be defined for each field when exporting the filter in Fixed Width format within the Data Export tool.</p>
Save To	<p>Indicates whether the filter saves to the current user, a user group(s) or specific folder.</p> <p>If a filter is saved to more than one user group, a separate copy is stored for each group. Each group can independently edit the filter without affecting another group's copy. If a filter with the same name already exists within a group, the filter name is appended with a number in parentheses indicating an incremented version number (<i>i.e.</i>, HonorStudents already exists for a group so saving a new filter with the same name appends the name to HonorStudents(2)). If the filter was saved across multiple groups, the filter name only displas as appended for groups where a filter with the same name already exists.</p>
Test	<p>This field allows users to test and preview a filter before saving it. Test results display in a separate window. To view the test filter, pop-up windows must be enabled on the web browser.</p>
Save	<p>Saves the filter.</p>

Grouping and Aggregation Descriptions

Grouping and aggregation places results into groups and calculations can be performed on the results. Aggregations display at the bottom of each data group when extracting the data. These options are not available for fixed-width output formats.

*Query Name:

Short Description:

Long Description:

Group the data into sections that can have aggregates/sub-totals

Grouping	Group by	Group Order
Tier 1	<input type="text" value="sch.schoolID"/>	<input type="text" value="Ascending"/>
Tier 2	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 3	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 4	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 5	<input type="text"/>	<input type="text" value="Ascending"/>

Aggregate/Sub Total by Aggregate Type

Aggregate/Sub Total by	Aggregate Type
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Save To: User Account
Folder:

User Groups

Force Order [?](#)

▶ [Click here to expand...](#)

The following describes the available options.

Field	Description
Grouping	This is the order in which each group is reported. Users are allowed to report up to 5 tiers (or groups).
Group By	Determines which field is in the group and reports aggregate/sub-totals. Only fields included within the filter are available for selection.
Group Order	Determines how group aggregate/sub-totals are reported when exported via the Data Export tool.
Aggregate/Sub Total by	Determines which field within the filter is used for the 'Group by' fields. For example, a user creating a behavior Ad hoc filter who chooses to Group By behavior events and Aggregate By personID using an Aggregate Type of Distinct Count produces the number of students per Behavior Event Type.

Field	Description
Aggregate Type	<p>Determines which calculation is applied to the group when calculating and reporting aggregate/sub-totals. For example, a group containing student last names (student.lastName) with an Aggregate/Sub Total of State ID (student.stateID) and an Aggregate Type of Distinct Count reports individual groups based on student last names with a count of how many students within that group have distinct State IDs.</p> <p>Aggregate Types include:</p> <ul style="list-style-type: none"> • Record Count - Indicates the total number of records in the group. • Distinct Count - Indicates the total number of distinct records within a group based on the fields selected to be counted from the Aggregate By option. • MIN—Indicates the minimum value for the designated Aggregate/Sub Total field within a group (e.g., an Aggregate/Sub Total for State ID (student.stateID) with a MIN Aggregate Type reports the smallest State ID value with each group). • MAX—Indicates the maximum value for the designated Aggregate/Sub Total field within a group (e.g., an Aggregate/Sub Total for State ID (student.stateID) with a MAX Aggregate Type reports the largest State ID value within each group). • SUM - Indicates the sum of all values within a group for the Aggregate/Sub Total field selected (<i>i.e.</i>, an Aggregate/Sub Total for Present Minutes (attendanceDetail.presentMinutes) with a SUM Aggregate Type reports a sum of all Present Minutes with each group). • AVG - Indicates the average of all values within a group for the Aggregate/Sub Total field selected (<i>i.e.</i>, an Aggregate/Sub Total for Present Minutes (attendanceDetail.presentMinutes) with AVG Aggregate Type reports the average of Present Minutes for all students within each group) <p>See the Rules for Aggregate Calculations by Data Type table below for more information.</p>
Save To	<p>Indicates whether the filter saves to the current user, a user group(s) or specific folder.</p> <p>If a filter is saved to more than one User Group, a separate copy is stored for each group. Each group can independently edit the filter without affecting another group's copy. If a filter with the same name already exists within a group, the filter name is appended with a number in parentheses indicating an incremented version number (<i>i.e.</i>, HonorStudents already exists for a group so saving a new filter with the same name appends the name to HonorStudents(2)). If the filter was saved across multiple groups, the filter name only displays appended for groups where a filter with the same name already exists.</p>

Field	Description
Test	This field allows users to test and preview a filter before saving it. Test results display in a separate window. To view the test filter, pop-up windows must be enabled on the web browser.
Save	Saves the filter within Infinite Campus. The filter is now available for use in all Ad hoc Filter fields throughout Infinite Campus (if the user is part of the user group the filter was saved to).

Rules for Aggregate Calculations by Data Type

The following table describes all rules for allowing or disallowing aggregate calculations based on data type:

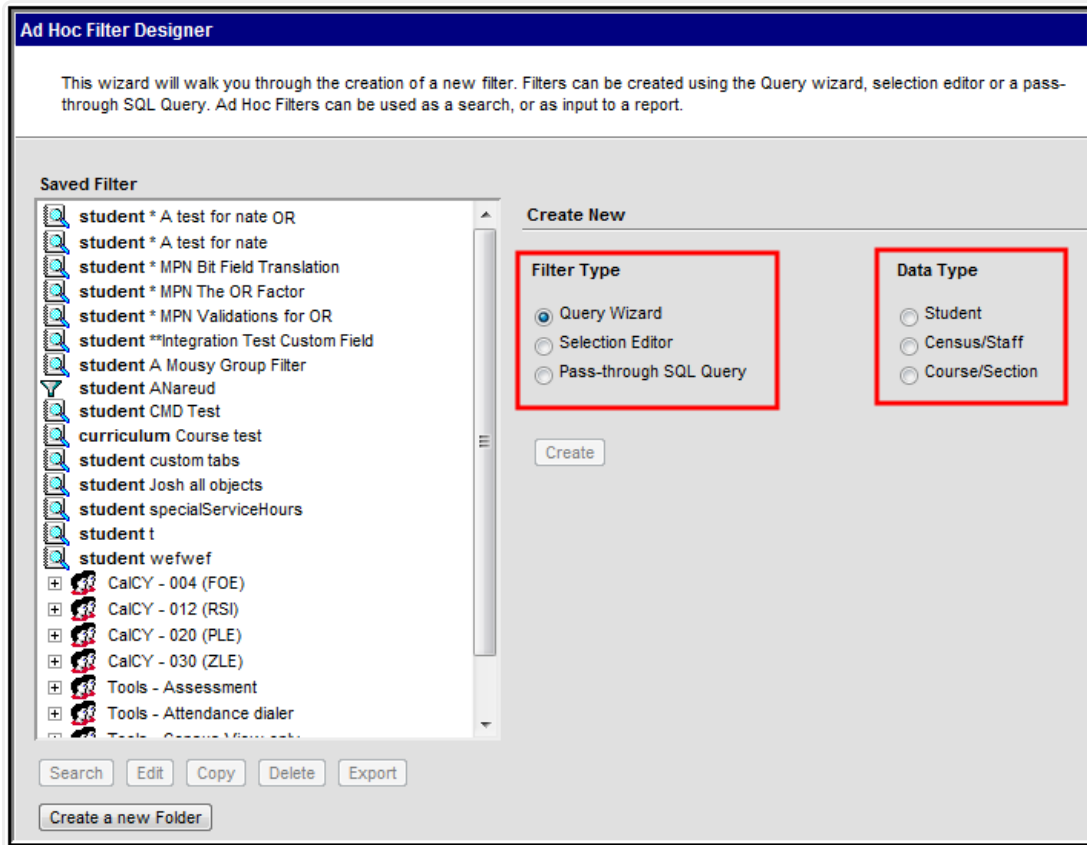
Data Type	Number	Float	String	Date	Text	Bit
MIN	Yes	Yes	Yes	Yes	Yes	Yes
MAX	Yes	Yes	Yes	Yes	Yes	Yes
AVG	Yes	Yes	No	No	No	No
SUM	Yes	Yes	No	No	No	No
Record Count	Yes	Yes	Yes	Yes	Yes	Yes
Distinct Count	Yes	Yes	Yes	Yes	Yes	Yes

Create a Filter

The following is a basic workflow for creating a filter. See the [Query Wizard Features](#) for additional formatting and modification that can be done for more advanced filters.

Step 1. Choose Filter and Data Type

1. Select the **Query Wizard** radio button.
2. Select a **Data Type**. This determines which fields are available for selection: **Student**, **Census/Staff**, or **Course/Section**.
3. Click the **Next** button. The screen displays a list of fields to select in order to create the filter.

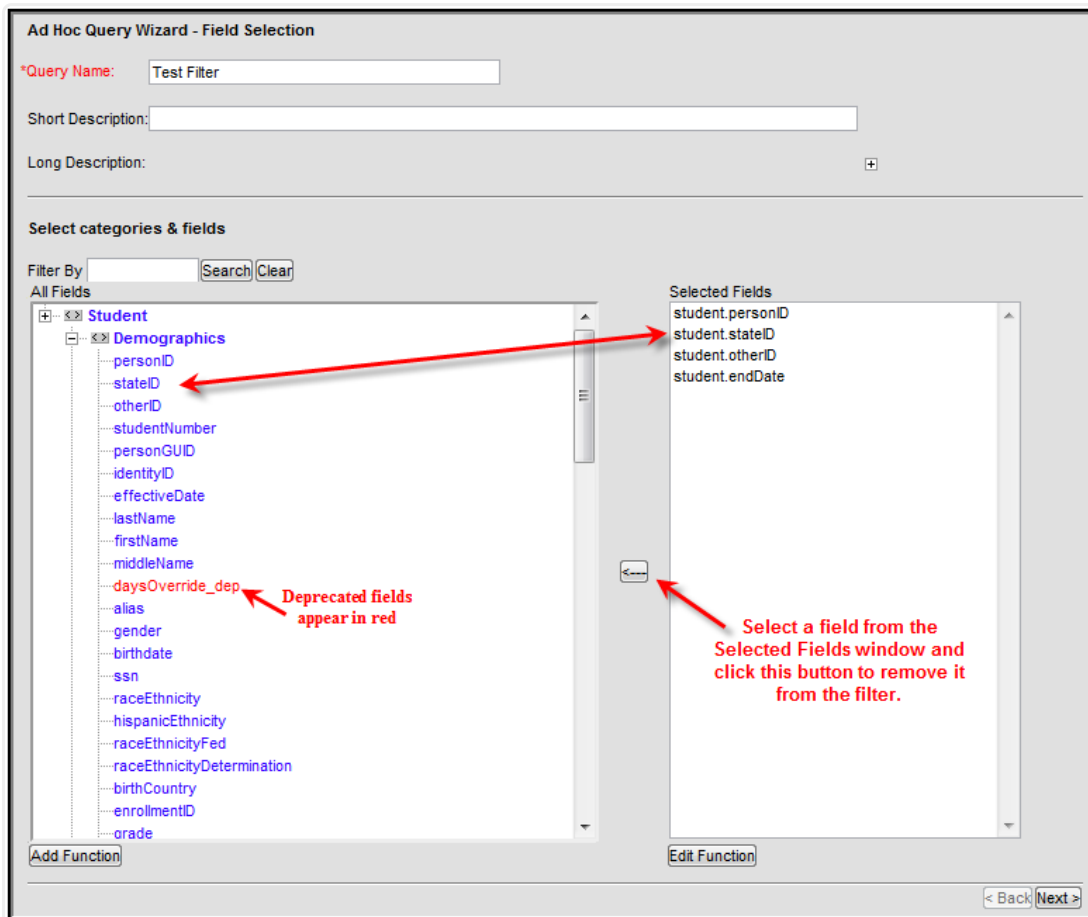


Filter Type, Data Type Selection

Step 2. Select Categories and Fields

Campus fields are organized into specific categories relating to the Filter Data Type selected on the previous screen. Categories are organized in a hierarchy format, where selecting the (+) open available fields and additional subcategories within the category. Users may include Campus and user-created custom fields when building filters.

1. Enter a **Query Name** for the filter.
2. Enter a **Short** and or **Long Description** about the filter (if applicable).
3. Select the data elements from the **All Fields** list by clicking on them. The fields move to the Selected Fields list. To remove a field from the Selected Fields list, click on it to highlight it and click the left-pointing arrow button.
4. Select the **Add Function** button to add a function to the filter.
5. To search for a particular field, enter part of its name in the Filter By section and click the Search button. Select the appropriate options for the query. All fields that contain that name display in the All Fields list. To clear the selection, click the Clear button, and all available fields display again.
6. To save the filter right now without testing it or modifying any results of the selected fields, choose **Save** or **Save and Test**.
7. To continue, click the **Next** button to continue creating the filter, narrow returned results and sort the filter into the desired order.



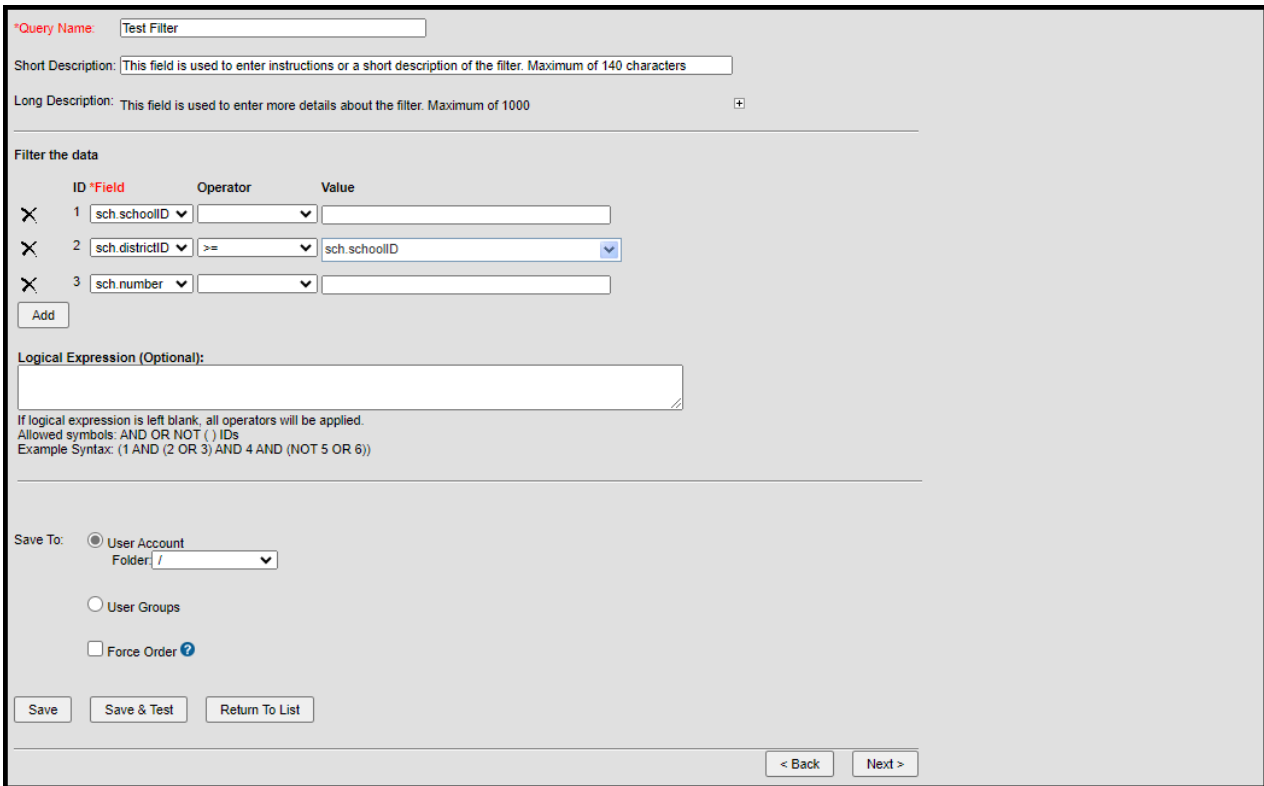
Adding/Removing Filter Fields

Step 3. Enter Filter Parameters

Filter parameters allow users to define specific constraints for how each field is filtered within the filter. This tool allows users to filter very specific data within reports and other exported files.

1. Enter the **Query Name** and a **Short/Long Description** (if applicable).
2. Select the **Operator** for each Field. The available fields are based on the data elements selected in the previous Field Selection screen.
3. Enter the **Value** for each Operator. This is the value being used in conjunction with the Operator selected (*i.e.*, student.age > 5, where 5 is the value entered and the output is all students older than 5 years of age).
4. If a BETWEEN Operator was selected, fill in all appropriate fields.
5. Click the Add Filter button to apply multiple operators to the same field(s). Selecting this button adds an additional field area where users can select an already existing filter field and apply additional operators.
6. Enter a **Logical Expression**, if necessary.
7. For complicated filters that report data from several calendars and/or have many fields from many different areas, mark the **Force Order** checkbox. When marked, the database fields in the query are executed in a particular order to increase the filter's performance. When a filter takes several minutes to generate, try generating it again with this checkbox marked. Marking this on every filter is not recommended.

8. To save the filter right now without testing it or modifying any results of the selected fields, choose **Save** or **Save and Test**.
9. Select the **Next** button if output formatting and/or group data needs to be defined for the filter.



*Query Name:

Short Description:

Long Description:

Filter the data

ID	*Field	Operator	Value
1	sch.schoolID		
2	sch.districtID	>=	sch.schoolID
3	sch.number		

Logical Expression (Optional):

If logical expression is left blank, all operators will be applied.
 Allowed symbols: AND OR NOT () IDs
 Example Syntax: (1 AND (2 OR 3) AND 4 AND (NOT 5 OR 6))

Save To: User Account
 Folder:

User Groups

Force Order [?](#)

Step 4. Enter Output Formatting Values

1. Enter the **Query Name** and a **Short/Long Description** (if applicable).
2. If data should output in unduplicated records based on field values, mark the **Output distinct records** checkbox.
3. If the field should appear in the filter output, verify the **Output** checkbox is marked. If it is not marked, the field does not display in the output but is used to filter data. For example, the field student.activeToday might be chosen to filter out inactive students (student.activeToday = 1), but the Output checkbox could be unselected so that field is not included in the output.
4. Enter the **Sequence**. This number places the field in that order on the output.
5. Enter a number in the **Sort** field. This determines the order in which fields are sorted.
6. If a number was entered in the Sort field, determine how the field should be sorted by selecting a **Direction**. Data can be sorted by ascending or descending direction. If the Sequence and Sort fields are left blank, the fields display in the order selected and sort how the elements appear on the screen.
7. Enter a **Column Header** for each field. This is the header that display in the column relating to the field. If no header is entered, the field name is used as the header for the column (*i.e.*, student.otherID displays a column name of student.otherID if no header is entered).
8. Determine the field's **Alignment** on files exported via the Data Export tool.
9. Select the **Formatting** of outputted field data. These options allow users to specify how data is reported in exported files.
10. Enter the field **Length**. This field determines the maximum amount of characters the field

reports data before truncation. If data is exported using the Fixed Width format, each field with the Output checkbox checked must have a length value entered.

11. To save the filter right now without testing it or modifying any results of the selected fields, choose **Save** or **Save and Test**.
12. To continue, click the **Next** button to continue creating the filter, narrow returned results and sort the filter into the desired order.

***Query Name:**

Short Description:

Long Description: +

Format the output file/report

Output distinct records

Field	OutputSeq	Sort	Direction	Column Header	Alignment	Formatting	Length
sch.schoolID <input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>	▼	<input type="text"/>	▼	<input type="text"/>	▼
sch.districtID <input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>	▼	<input type="text"/>	▼	<input type="text"/>	▼
sch.number <input checked="" type="checkbox"/>	<input type="text"/>	<input type="text"/>	▼	<input type="text"/>	▼	<input type="text"/>	▼

Save To: User Account
Folder:

User Groups

Force Order ?

Step 5. Define Data Filter Grouping, Calculations and Subtotals

The Grouping and Aggregation editor allows users to group fields into sections and report specific aggregates/sub-totals for each section.

1. Enter the **Query Name** and a **Short/Long Description** (if applicable).
2. Select each field to **Group By** for each tier. This field determines which fields are grouped into sections, allowing the field to have separate aggregate/sub-totals reported.
3. Select each tier **Group Order**. This determines how aggregate/sub-total data is reported for the tier.
4. Select the field and determine the **Aggregate/Sub Total by Aggregate Type**. Data within each group aggregates based on the field and Aggregate Type selected. See the table below for information about each available aggregate type

*Query Name:

Short Description:

Long Description:

Group the data into sections that can have aggregates/sub-totals

Grouping	Group by	Group Order
Tier 1	<input type="text" value="sch.schoolID"/>	<input type="text" value="Ascending"/>
Tier 2	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 3	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 4	<input type="text"/>	<input type="text" value="Ascending"/>
Tier 5	<input type="text"/>	<input type="text" value="Ascending"/>

Aggregate/Sub Total by Aggregate Type

<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Save To: User Account
 Folder:

User Groups

Force Order [?](#)

Step 6. Save the Filter

To quickly save the filter, click the **Save** button. To quickly save and verify the filter's return data, click the Save and Test button. Both options save the filter and can be found in the Saved Filter list. The Save and Test option saves the filter and generates it in HTML format for a quick review of the selected fields and format. Users must have pop-ups enabled on the web browser in order to view Test results.

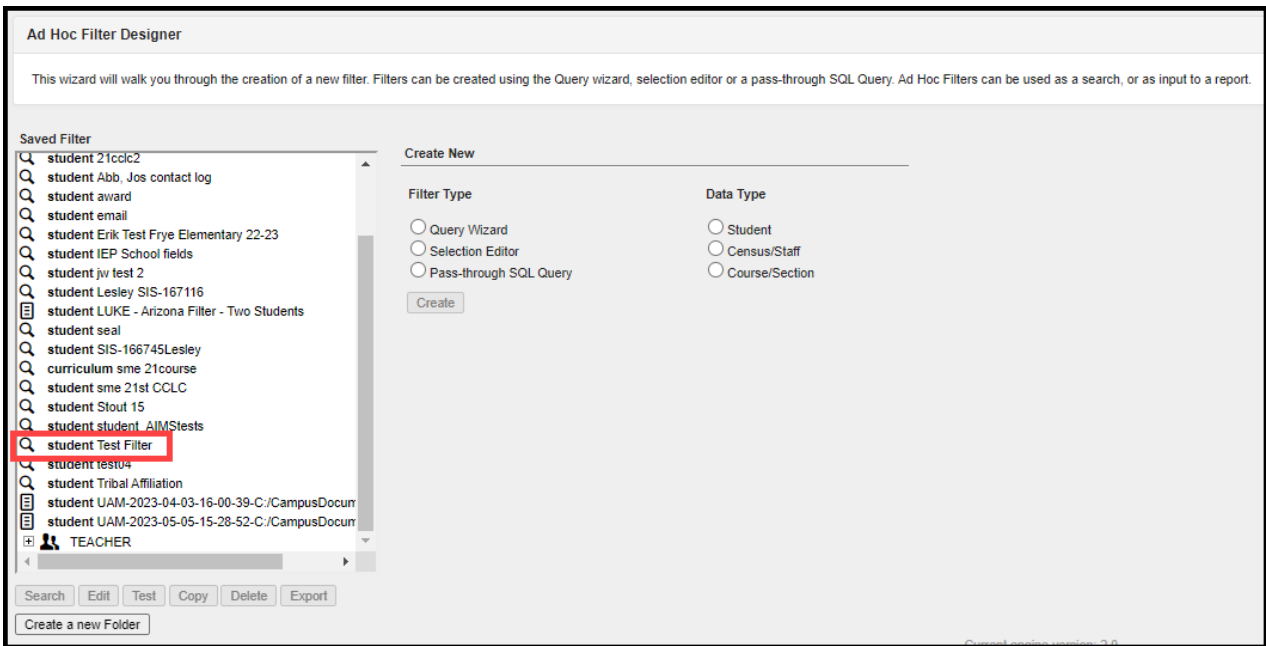
For more advanced save features, follow the procedures below.

1. Determine if the filter needs to be saved to a **User Account Folder**. If yes, choose that radio button and select the appropriate folder.
2. Determine if the filter needs to be available to particular User Groups. If yes, choose that radio button and select the appropriate user groups. If a filter is saved to more than one User Group, a separate copy is stored for each group. Each group can independently edit the filter without affecting another group's copy.

User Groups in which you are a member are the only groups that will be displayed. You cannot add a filter to a User Group if you are not already a member.

3. For complicated filters that report data from several calendars and/or have many fields from many different areas, mark the **Force Order** checkbox. When marked, the database fields in the query are executed in a particular order to increase the filter's performance. When a filter takes several minutes to generate, try generating it again with this checkbox marked. Marking this on every filter is not recommended.

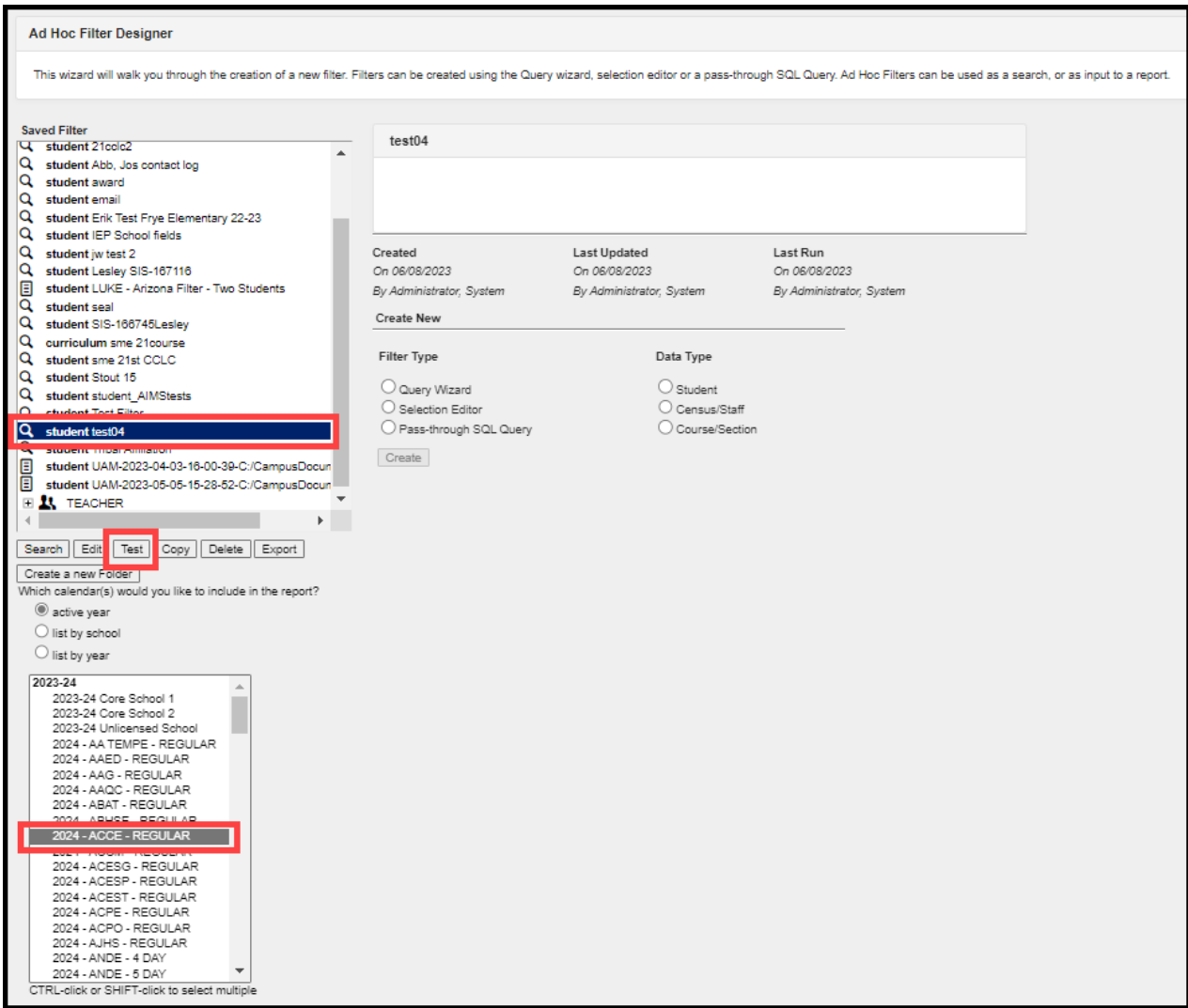
4. Select the **Save** icon. The filter is now saved and can be selected from the **Saved Filter** list on the main page of the Filter Designer.



To generate a saved filter:

1. Select the desired filter from the Saved Filter list.
2. Choose the appropriate Calendar.
3. Click the **Test** button. The filter will appear as a report in a separate window.

Calendars cannot be selected if the query is for Census/Staff Data Types. Only calendars to which the user is assigned calendar rights are available for selection.



Manage Filters

[Save Filters to Folders](#) |
 [Remove Fields from the Filter Parameters Editor](#) |
 [Create Folders for Filters](#) |
 [Add a Saved Query to a Folder](#) |
 [Move Filters between Folders](#) |
 [Copy Filters](#) |
 [Delete Filters](#) |
 [Modify a Query Created by Another User](#) |
 [Test Saved Filters](#) |
 [Last Updated, Last Run, and Last Run By Information](#)

Save Filters to Folders

Ad hoc filters can be saved to specific folders created within the Filter Designer tool, User Accounts, or User Groups.

For complicated filters that report data from several calendars and/or have many fields from many different areas, mark the **Force Order** checkbox. When marked, the database fields in the query are executed in a particular order to increase the filter's performance. When a filter takes several minutes to generate, try generating it again with this checkbox marked. Marking this on every filter is not recommended.

*Query Name:

Short Description:

Long Description:

Filter the data

ID	*Field	Operator	Value
1	<input type="text" value="sch.schoolID"/>	<input type="text"/>	<input type="text"/>
2	<input type="text" value="sch.districtID"/>	<input type="text" value=">="/>	<input type="text" value="sch.schoolID"/>
3	<input type="text" value="sch.number"/>	<input type="text"/>	<input type="text"/>

Logical Expression (Optional):

If logical expression is left blank, all operators will be applied.
 Allowed symbols: AND OR NOT () IDs
 Example Syntax: (1 AND (2 OR 3) AND 4 AND (NOT 5 OR 6))

Save To: User Account
 Folder:

User Groups

Force Order [?](#)

Remove Fields from the Filter Parameters Editor

Fields can be removed from the Filter Parameters editor without being removed from the filter as a whole. This allows users to reduce the Filter Parameters editor to only those fields in which operators are assigned or only those fields in which the user wants to see.

Fields removed from the Filter Parameters editor are not removed from the filter; they are only the user's view of the editor.

*Query Name:

Short Description:

Long Description:

Filter the data

ID	*Field	Operator	Value
1	<input type="text" value="sch.schoolID"/>	<input type="text" value=""/>	<input type="text" value=""/>
2	<input type="text" value="sch.districtID"/>	<input "="" type="text" value=">="/>	<input type="text" value="sch.schoolID"/>
3	<input type="text" value="sch.number"/>	<input type="text" value=""/>	<input type="text" value=""/>

Logical Expression (Optional):

If logical expression is left blank, all operators will be applied.
Allowed symbols: AND OR NOT () IDs
Example Syntax: (1 AND (2 OR 3) AND 4 AND (NOT 5 OR 6))

Save To: User Account
Folder:

User Groups

Force Order [?](#)

Select the X next to each field to remove fields from the Filter Parameters Editor.

Removing a field from the list does not remove it from the filter output.

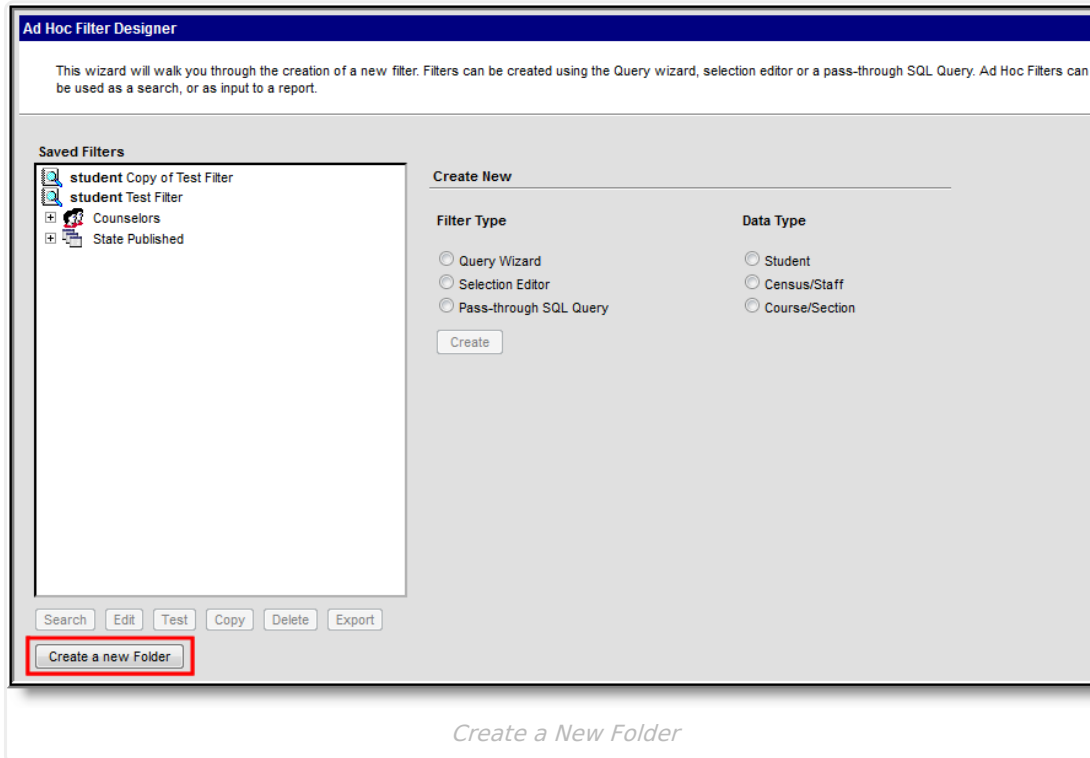
All fields not assigned an Operator were removed, and the field IDs were automatically renumbered. The Logical Expression automatically updates to match new field IDs.

The Filter Designer tool allows users to create folders for organizing and storing Ad hoc filters. Folders can be organized in a hierarchy format, where sub-folders exist within parent folders. By creating folders, users can better manage large volumes of existing Ad hoc filters and group them in a logical order.

If a field in the query has been deactivated (displays in red), use the [Element Replacement Tool](#) to update the filter. This removes the deactivated field and adds the equivalent field to the filter.

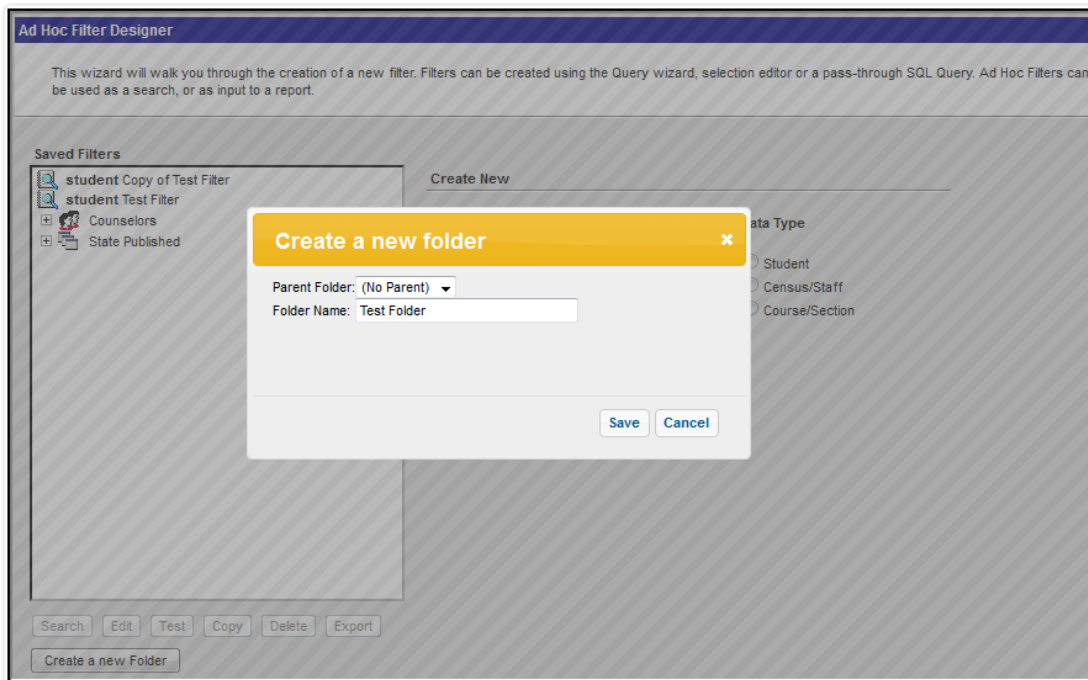
Create Folders for Filters

Folders allow users to better manage Ad hoc filters within the Filter Designer tool.



▶ [Click here to expand...](#)

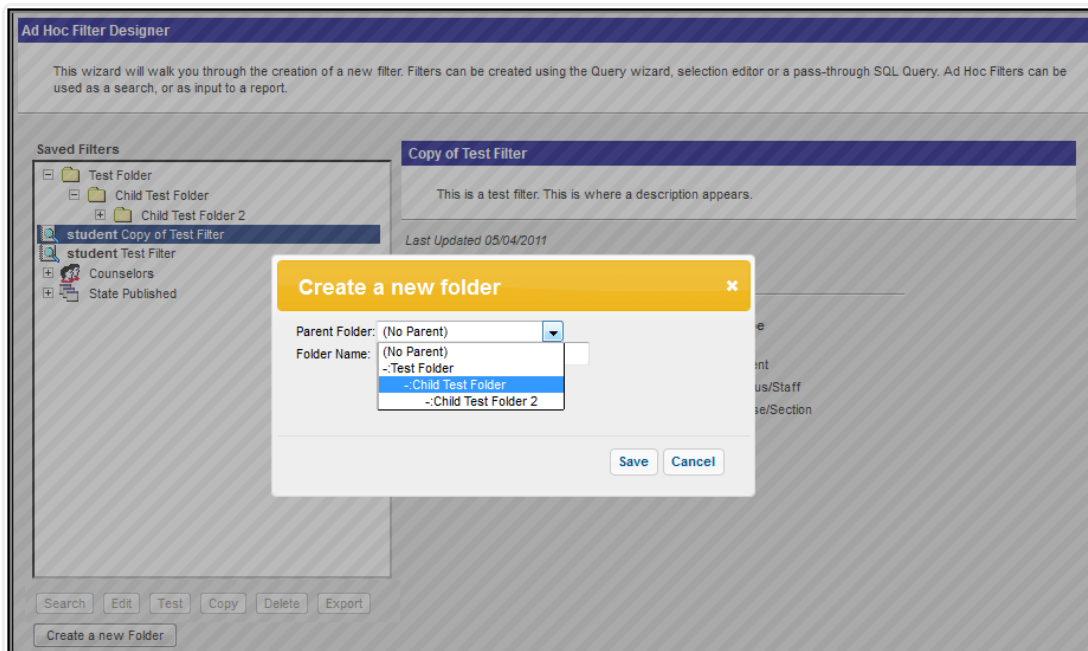
To create a new folder, select the **Create a new Folder** button. The **Create a new folder** editor displays.



Create a New Folder Editor

If the folder should not be tied to a parent folder, leave the **Parent Folder** field as (No Parent), enter a **Folder Name** and select the **Save** button. The folder displays in the **Saved Filters** field and is now available for storing Ad hoc filters.

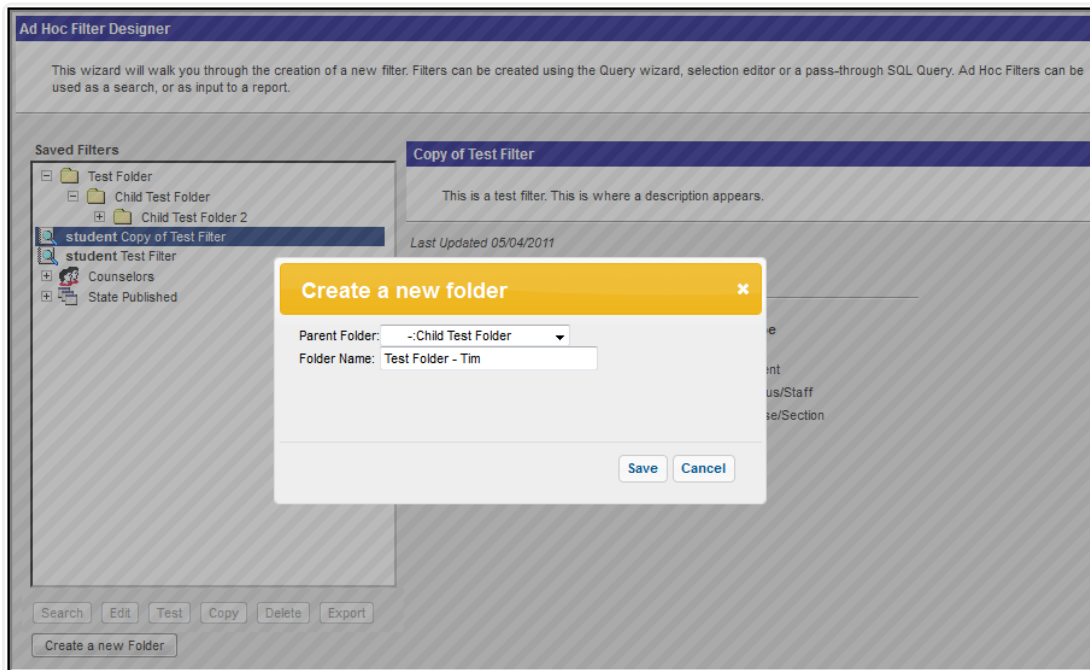
If the folder should be assigned to a parent folder, select the parent folder from the **Parent Folder** field.



Selecting a Parent Folder

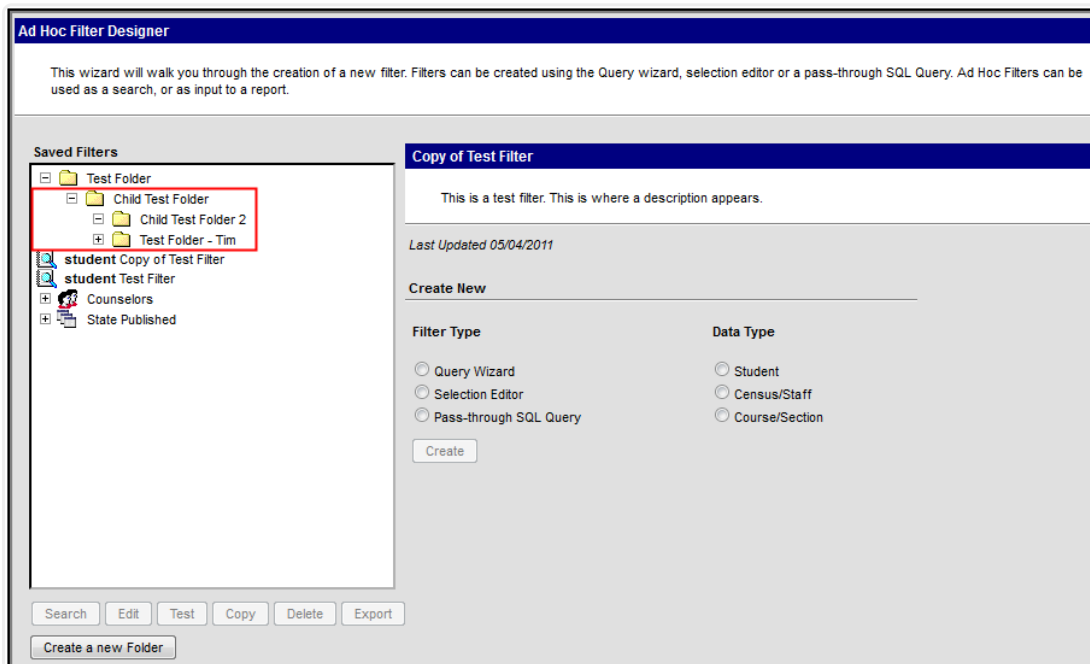
Locate the appropriate parent folder. The indentation next to each folder name indicates its

relationship to the previous folder (*i.e.*, the Grandchild 1 folder is indented two times because it exists within the Child 1 (Testing) folder which exists within the Parent Folder (Testing) folder). In the example above, the folder being created exists within the Child Test Folder parent folder.



Entering a Folder Name

Once the parent folder has been selected, it displays in the **Parent Folder** field. Enter the **Folder Name** of the folder being created and select the **Save** button. The folder displays in the Saved Filters field and is now available to store Ad hoc filters.

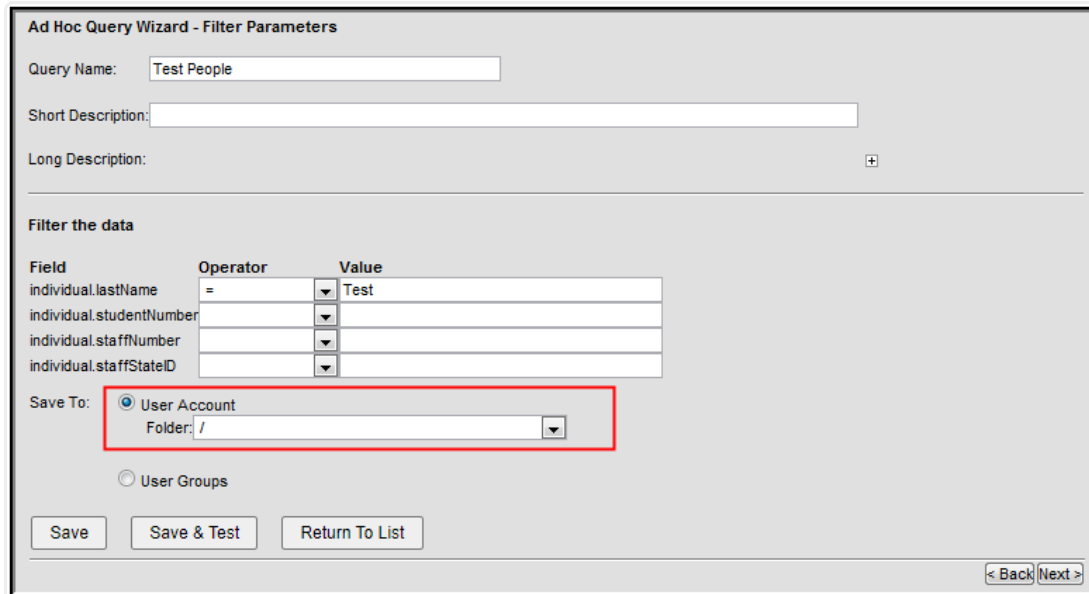


Created Folder

As the example above shows, the created folder Test Folder - Tim now exists within its parent folder Child Test Folder.

Add a Saved Query to a Folder

Once folders have been created, Ad hoc filters can now be assigned to those folders.



Ad Hoc Query Wizard - Filter Parameters

Query Name:

Short Description:

Long Description:

Filter the data

Field	Operator	Value
individual.lastName	=	Test
individual.studentNumber		
individual.staffNumber		
individual.staffStateID		

Save To: User Account
 User Groups

Folder:

Saving an Ad hoc Filter to a Folder

To assign an Ad hoc filter to a folder, click the **User Account** radio button and select the folder from the **Folder** field.

Ad Hoc Query Wizard - Filter Parameters

Query Name:

Short Description:

Long Description:

Filter the data

Field	Operator	Value
individual.lastName	=	Test
individual.studentNumber		
individual.staffNumber		
individual.staffStateID		

Save To: User Account

Folder:

- Mary
- Mary Testing
- Nate Test
- Nate Test 2
- Nate Test 3
- Parent Folder (Testing)
 - Child 1 (Testing)
 - Grandchild 1
 - Great Grandchild 1
 - Great Great Grandchild 1
 - Great Great Great Grandchild
 - Grandchild 2
 - Child 2 (Testing)
 - Child 3 (Testing)
- Testing 1 2 3
- Test Folder - Tim

Selecting the Saved Folder

In the example above, the Ad hoc filter is being assigned to the Test Folder - Tim folder.

Ad Hoc Query Wizard - Filter Parameters

Query Name:

Short Description:

Long Description:

Filter the data

Field	Operator	Value
individual.lastName	=	Test
individual.studentNumber		
individual.staffNumber		
individual.staffStateID		

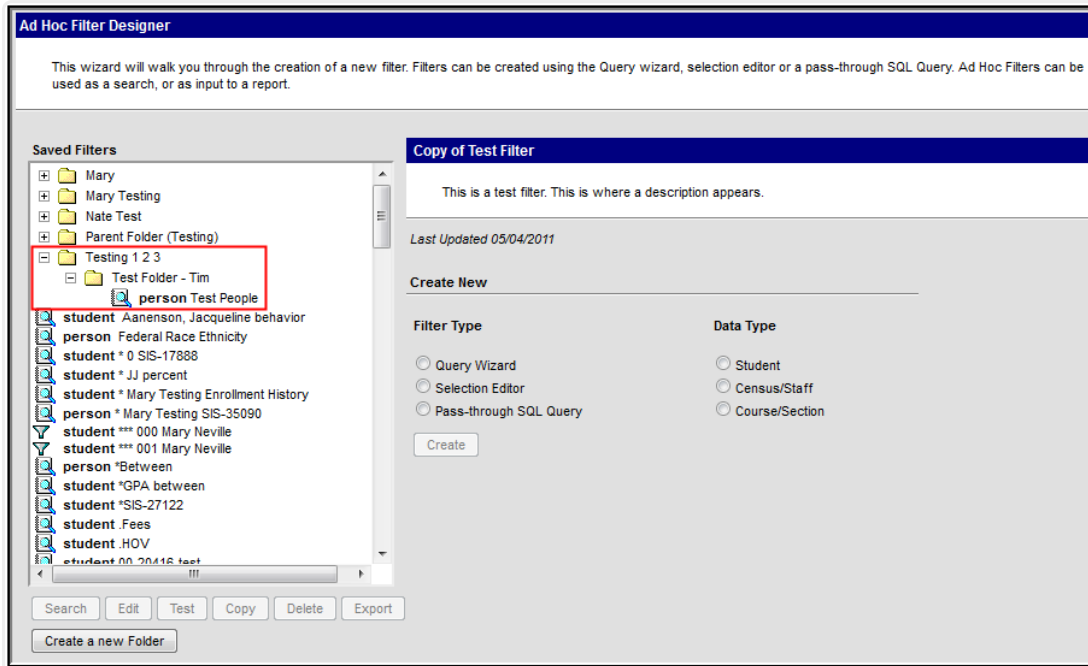
Save To: User Account

Folder:

User Groups

Saving the Ad hoc Filter to a Folder

Once the folder is selected, the **Folder** field displays the folder name. Select the **Save** button to save the filter to the folder.

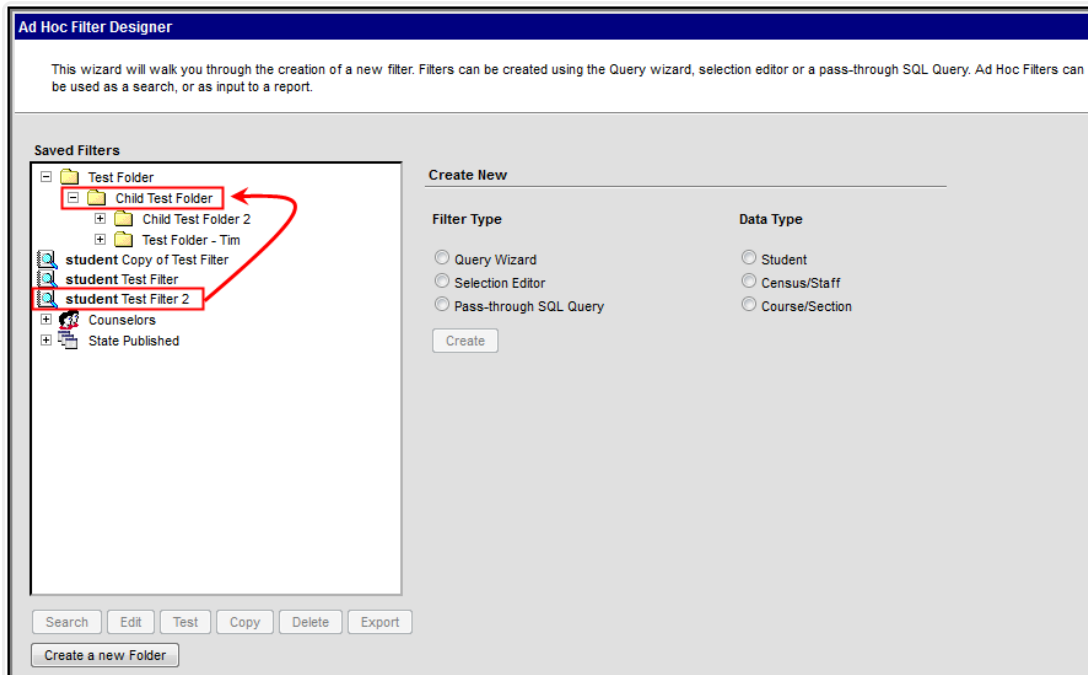


Viewing the Saved Filter in the Folder

The Ad hoc filter is now saved and accessible within the assigned folder.

Move Filters between Folders

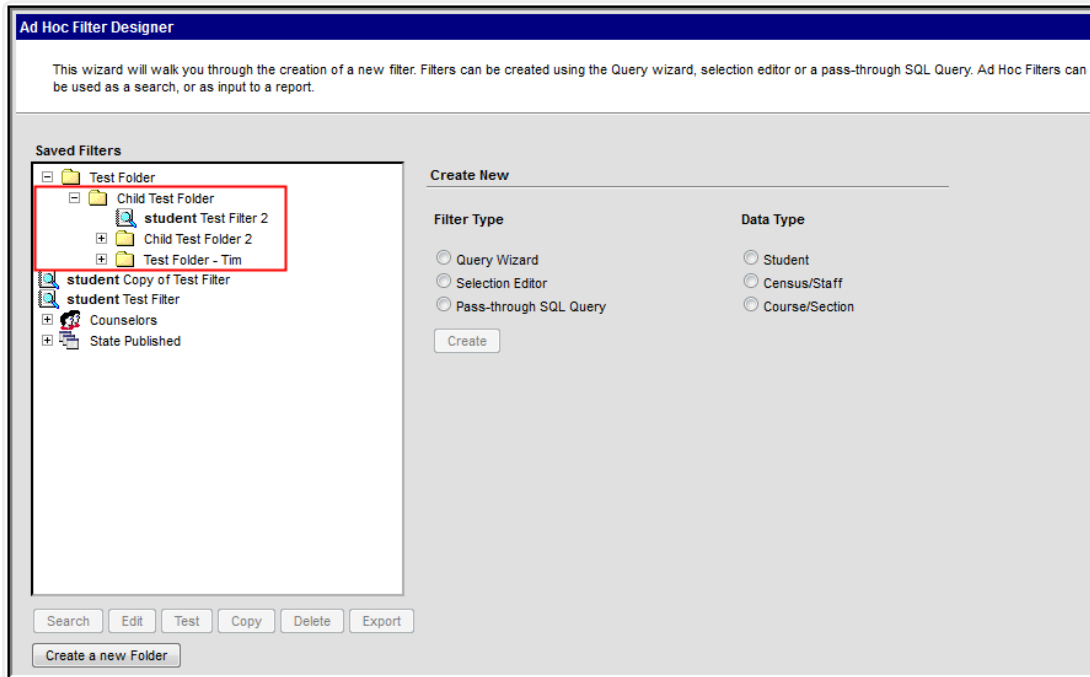
Ad hoc filters can be easily moved and organized between folders.



Moving an Ad hoc Filter to a Folder

To move an Ad hoc filter into an existing folder, left-click, hold, and drag the filter into the

designated folder. A pop-up message displays, asking the user to confirm the action. Select the **OK** button to move the Ad hoc filter.



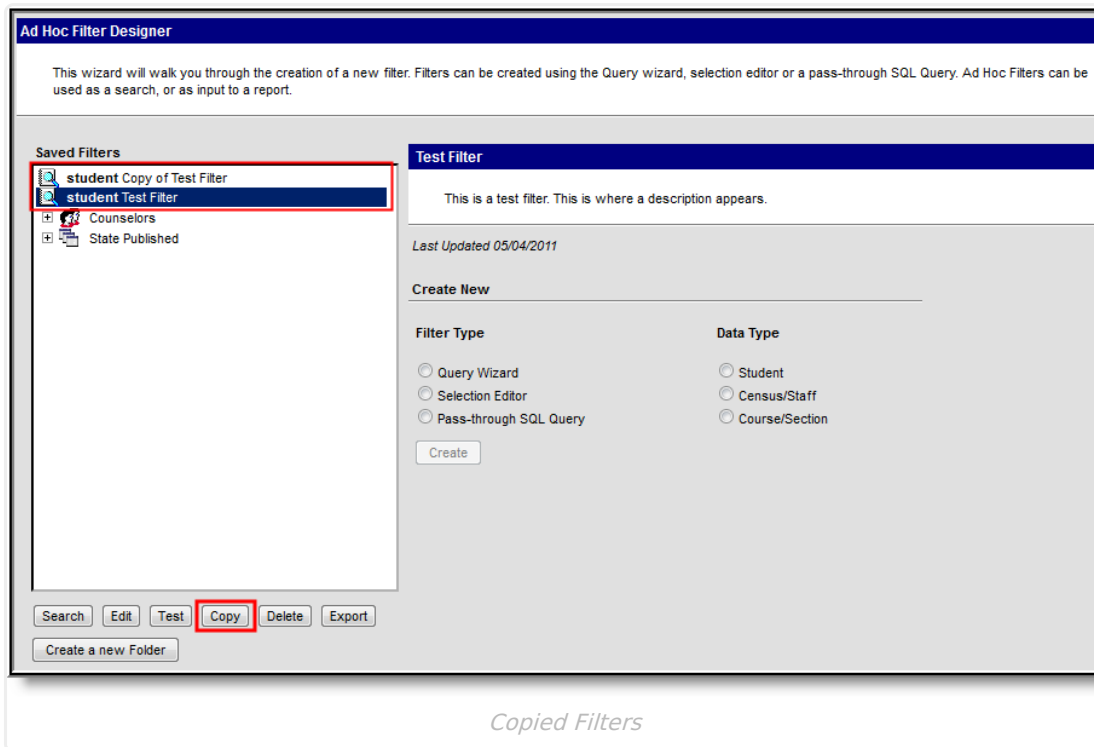
Viewing a Moved Ad hoc Filter

The moved filter now displays under the appropriate folder. This functionality moves filters in, out, and to another folder.

Existing filters can be easily copied if desired. This maintains the original version of the filter and lets users change a filter to add new fields and functions.

Copy Filters

Filters can be copied for additional editing. Select a saved filter and click the **Copy** button. A pop-up message displays indicating the filter has been copied. Copied filters are named Copy of [Original Filter Name].

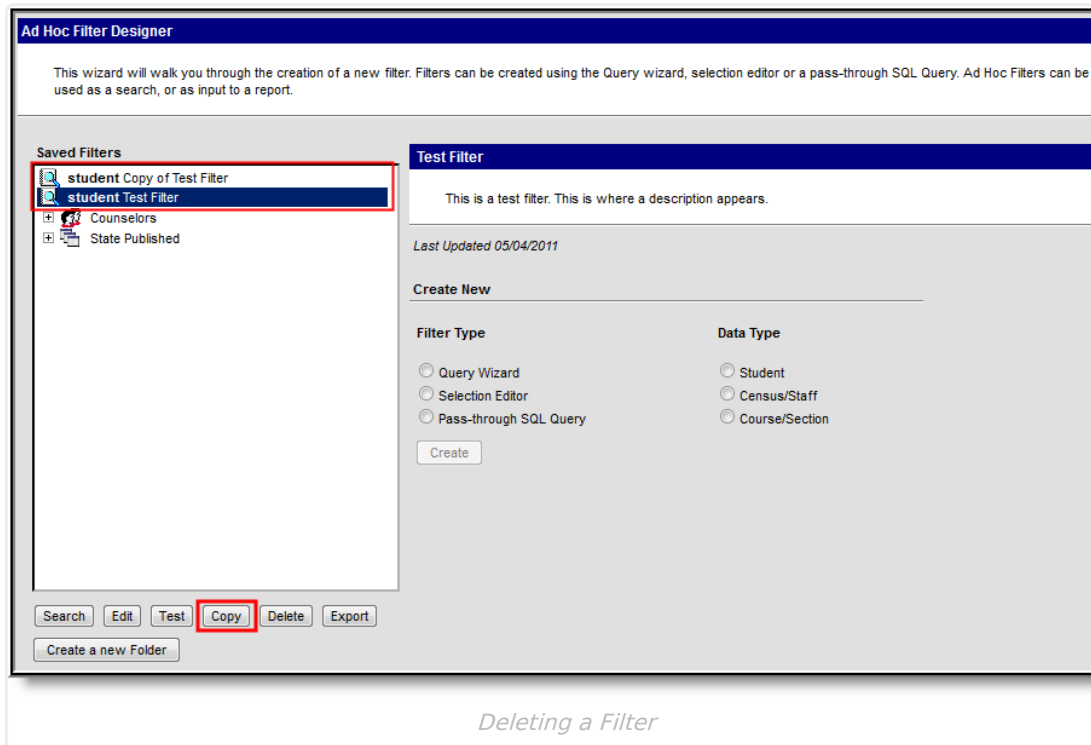


Copied Filters

Delete Filters

A saved filter created by a user can also be deleted by that user. However, because filters can be shared with other users, only the person who created the filter can delete it.

District users cannot delete State-Published filters.

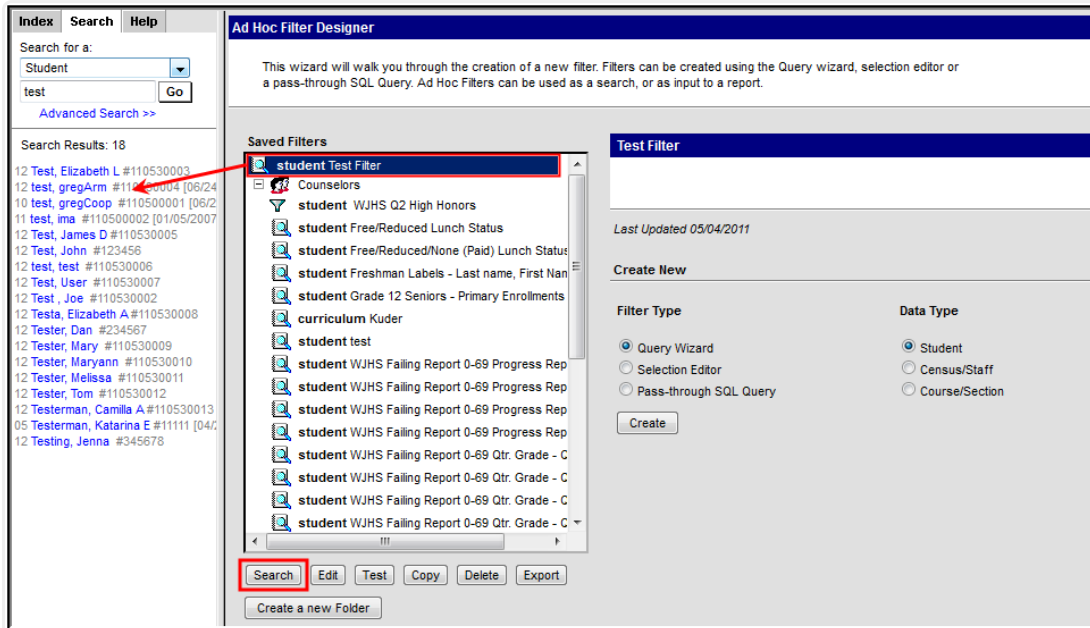


Select a filter from the Saved Filters window and click the Delete button to delete it. A pop-up message confirms the deletion. You can also delete multiple filters by holding the **Ctrl** key, selecting each filter, and clicking the **Delete** button.

Modify a Query Created by Another User

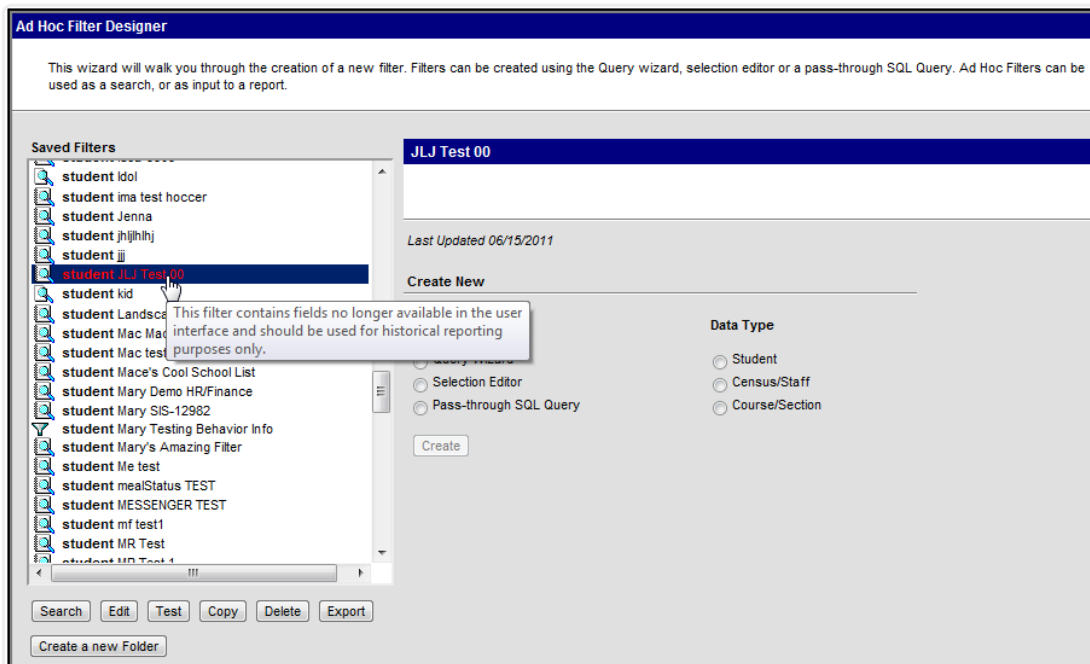
Saved filters can be edited anytime by selecting the filter and clicking the **Edit** button. This displays the filter so users can modify the selected fields and verify the operations and export options.

Search results on the Search tab can be populated with saved filters. When a saved filter is selected, click the **Search** button. Results returned in the filter displays in the Search tab.



Searching with a Filter

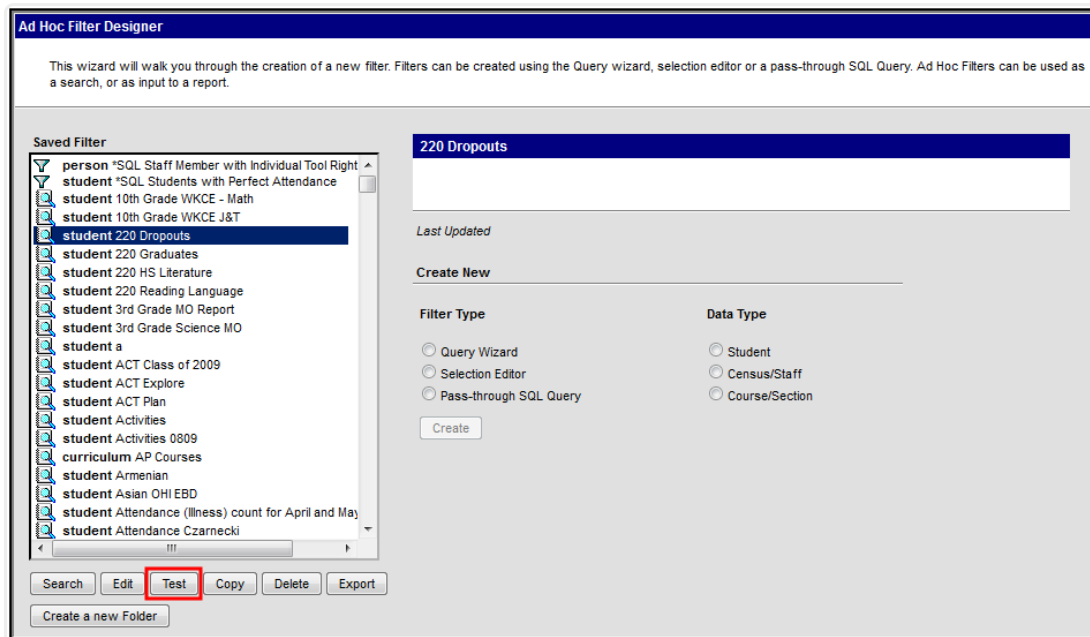
If a saved filter contains deprecated fields, the filter is highlighted in red within the Saved Filters window.



Filter Containing Deprecated Fields

Test Saved Filters

Select the filter from the Saved Filter window to test an existing filter and click the **Test** button. A separate window displays, displaying filter results in HTML format.



Testing an Existing Filter

Last Updated, Last Run, and Last Run By Information

Users can view the last time an existing filter was updated, the last time a test of the filter was run, and who ran the last test of the filter.

If the timestamp or user is unknown, a value of Unknown is reported.

The screenshot displays the 'Saved Filter' interface. On the left, a list of filters is shown, with 'person Number of Staff Members' selected. The main panel on the right, titled 'Number of Staff Members', contains tracking information: 'Last Updated 09/12/2018', 'Last Run 09/12/2018', and 'Last Run By Tester, Charlie'. A red box highlights this information, with a red arrow pointing to it. Below the tracking information is a 'Create New' section with radio buttons for 'Filter Type' (Query Wizard, Selection Editor, Pass-through SQL Query) and 'Data Type' (Student, Census/Staff, Course/Section). At the bottom of the interface are buttons for 'Search', 'Edit', 'Test', 'Copy', 'Delete', 'Export', and 'Create a new Folder'.

Example of a Filter Tracking Information