Barista Query Definition System

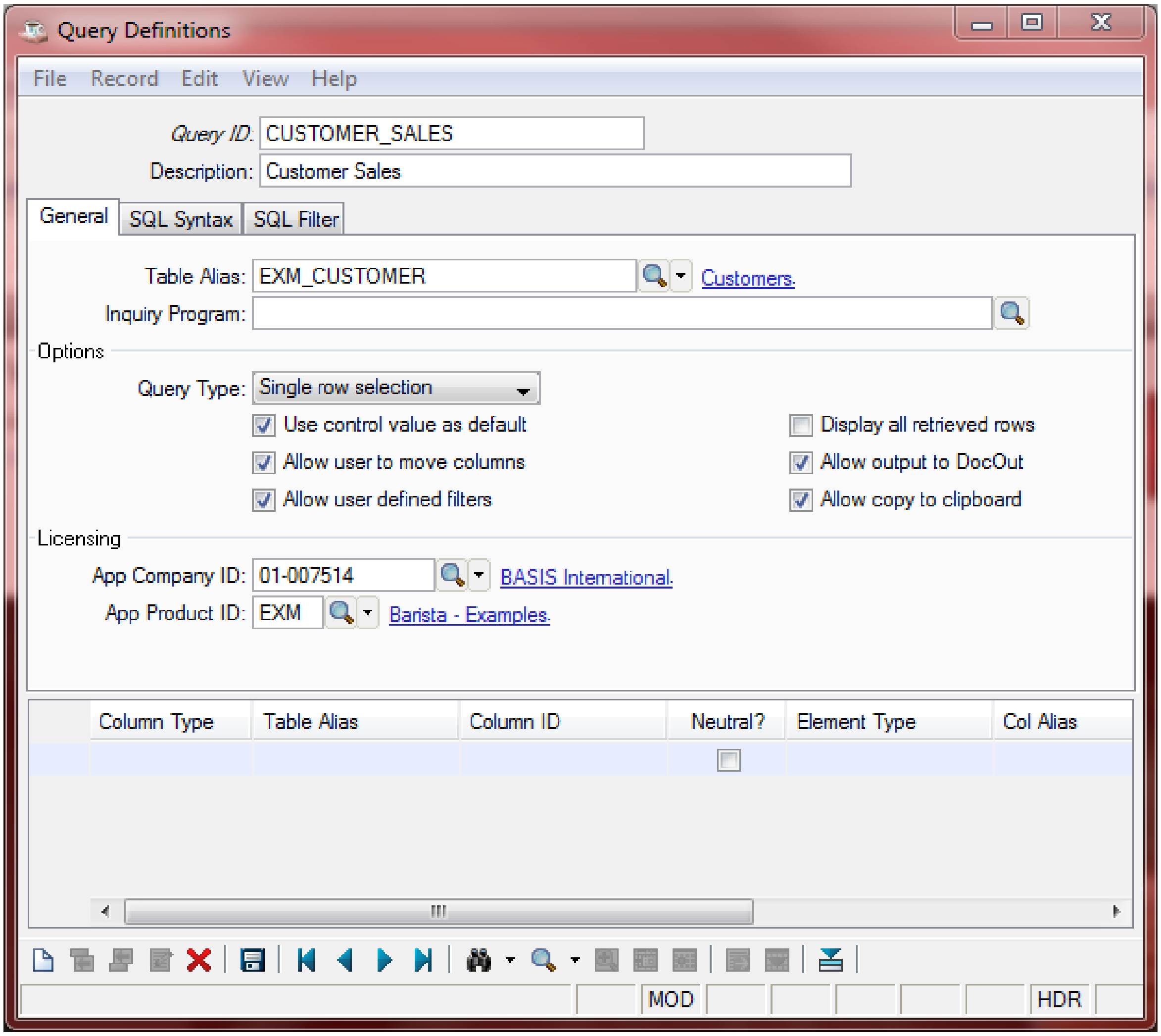
Overview

Barista version 11.10 expands the functionality of its built-in inquiry engine. New features include:

- Access to multiple tables including “virtual tables”
- Ability to use stored procedures to create those virtual tables and to join them to physical tables
- Ability to apply filters to the result set before being presented in the inquiry engine
- Create calculated or derived columns from existing columns in the tables
- Adding a column totaling function
- Expand sorting to include multiple columns
- Quickly output the results of the query to a report

Located on the Barista Development menu under Inquiries.

Query Definition Header



Query ID: CUSTOMER_SALES

Description: Customer Sales

Table Alias: EXM_CUSTOMER Customers.

Inquiry Program:

Options

Query Type: Single row selection

☑ Use control value as default

☑ Allow user to move columns

☑ Allow user defined filters

☐ Display all retrieved rows

☑ Allow output to DocOut

☑ Allow copy to clipboard

Licensing

App Company ID: 01-007514 BASIS International.

App Product ID: EXM Barista - Examples.

Column Type	Table Alias	Column ID	Neutral?	Element Type	Col Alias
			☐		

About the Fields

The Query Definition Header contains information about the general query, including primary tables, selection options and licensing. Additional tabs include the ability to modify the automatically generated SQL statement.

Query ID

The identifier for the specific query definition. Used to assign queries to Barista forms.

Description

Query description, which displays on the query form title bar during processing.

Table Alias

The primary selection table for the query.

Inquiry Program

Custom program called when a query menu option is selected (see Query Menus), or when one or more rows are selected in the inquiry grid. To enable the latter, add the [SELECT] token in front of the program name (i.e., [SELECT]<program_name>).

Query Type

Type of selection allowed in the displayed query grid:

- Single row selection - Allows selection and return of a single row of data. Default value when used on a Barista form.
- Multiple row selection - Allows selection and return of multiple rows of data. Used by custom application calling the query subsystem.
- View only - Restricts user to viewing data. Used in custom applications for displaying data.

Use control value as default

In Barista, uses the data from the calling form as the default search text in the query.

Allow user to move columns

Allows user to change the displayed sequence of query columns.

Allow user defined filters

Allows user to define and save filter criteria for the query. Filters are saved only for the current user.

Display all retrieved rows

Displays the entire data set at once instead of using the Barista standard of retrieving data as needed. Used for information displays where totals are requested. Note: selecting this option can have a negative impact on performance when querying large files.

Allow output to DocOut

Allows the user to send the returned data set directly to the Barista Document Output System, supplying additional options for pdf, txt, xml, etc. output.

Allow copy to clipboard
Allows user to copy contents from displayed query grid to system clipboard.

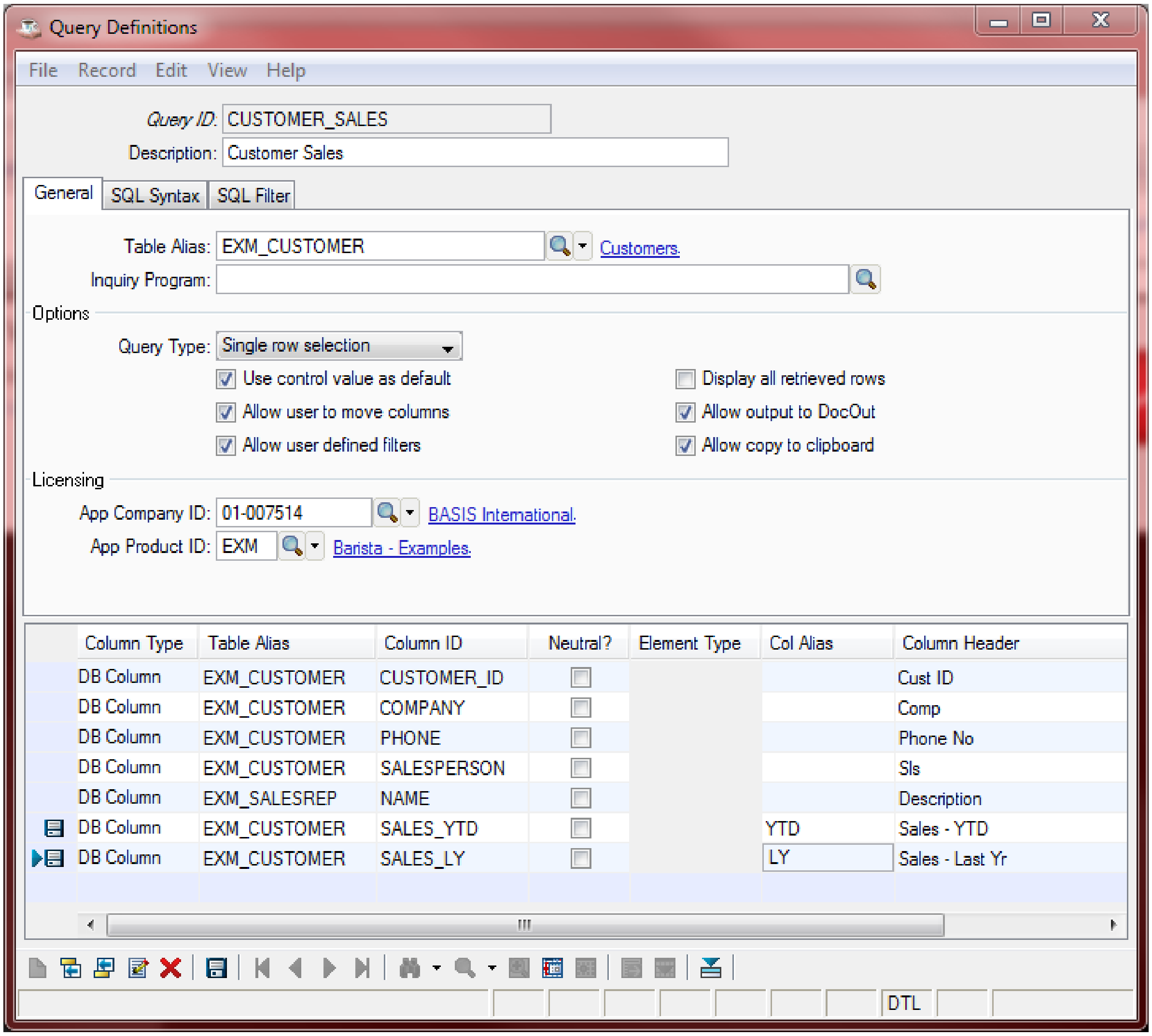
App Company ID
License Company ID

ASC Product ID
License product ID

SQL Syntax
Allows modification of the SQL statement generated by the “Generate SQL” additional option.

Select the Save icon at the bottom of the form.

Query Definition Detail



Query ID: CUSTOMER_SALES
Description: Customer Sales

Table Alias: EXM_CUSTOMER Customers
Inquiry Program:

Options

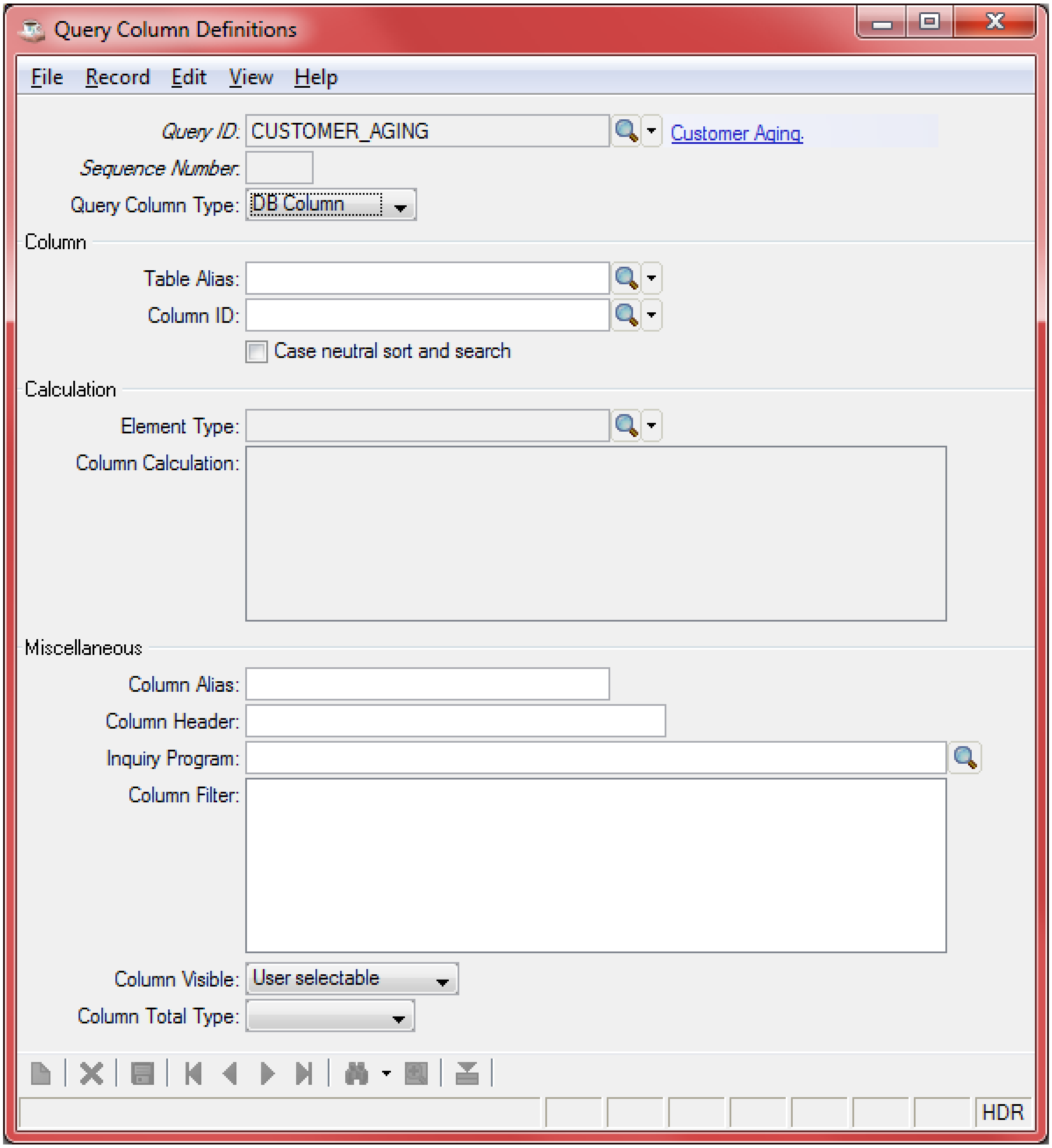
Query Type: Single row selection

☑ Use control value as default
☑ Allow user to move columns
☑ Allow user defined filters
☐ Display all retrieved rows
☑ Allow output to DocOut
☑ Allow copy to clipboard

Licensing

App Company ID: 01-007514 BASIS International
App Product ID: EXM Barista - Examples

Column Type	Table Alias	Column ID	Neutral?	Element Type	Col Alias	Column Header
DB Column	EXM_CUSTOMER	CUSTOMER_ID	☐			Cust ID
DB Column	EXM_CUSTOMER	COMPANY	☐			Comp
DB Column	EXM_CUSTOMER	PHONE	☐			Phone No
DB Column	EXM_CUSTOMER	SALESPERSON	☐			Sls
DB Column	EXM_SALESREP	NAME	☐			Description
DB Column	EXM_CUSTOMER	SALES_YTD	☐		YTD	Sales - YTD
DB Column	EXM_CUSTOMER	SALES_LY	☐		LY	Sales - Last Yr



Query ID: CUSTOMER_AGING Customer Aging

Sequence Number:

Query Column Type: DB Column

Column

Table Alias:

Column ID:

□ Case neutral sort and search

Calculation

Element Type:

Column Calculation:

Miscellaneous

Column Alias:

Column Header:

Inquiry Program:

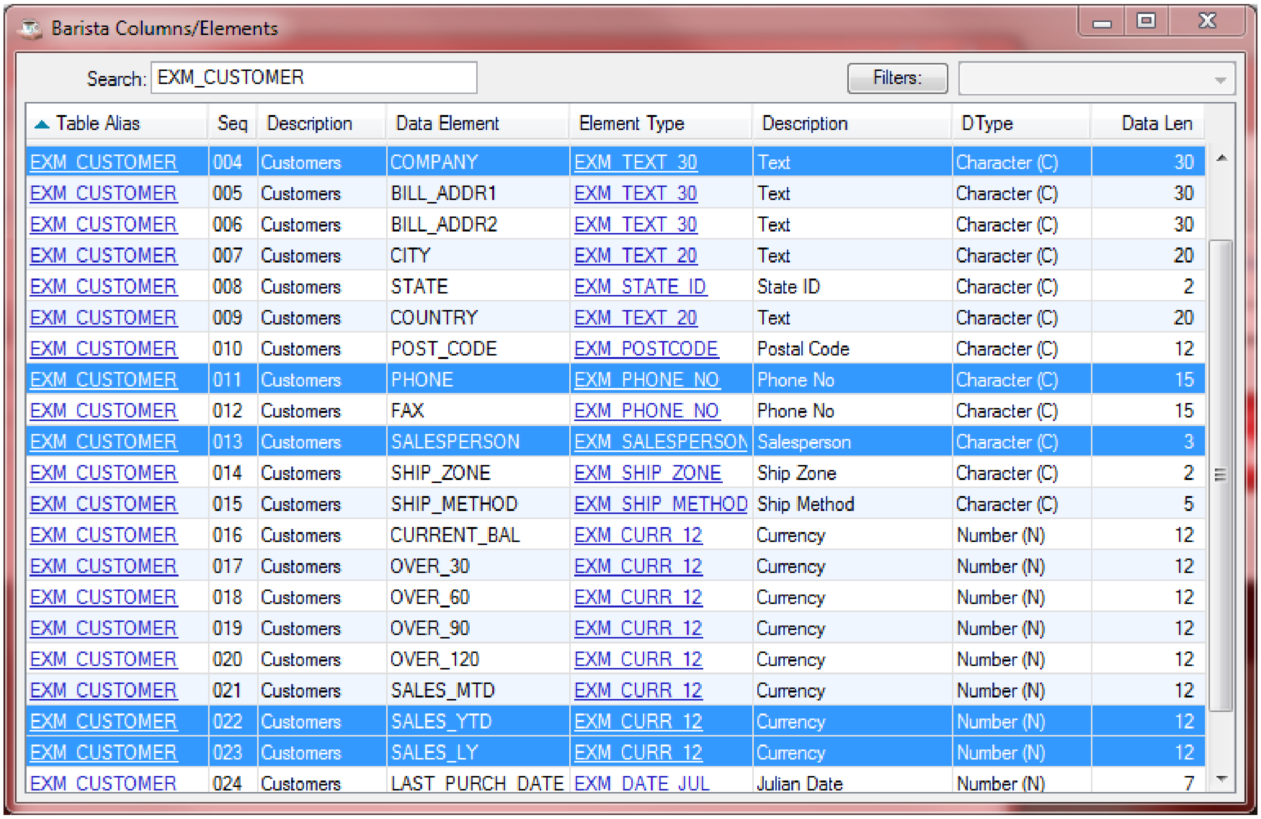
Column Filter:

Column Visible: User selectable

Column Total Type:

The next step is adding the query columns. This can be accomplished one of two ways. First, simply go to the detail portion of the form and enter the desired columns. The second method is to access the additional options menu with a right click, and select Generate Columns. This displays a list of all columns available on the primary SQL select table. Highlight the desired columns and press <enter>...all selected columns will be returned to the query definition. You must include all of the fields in the primary key, even if you are not using/displaying them all in the query output.

Note: Data columns validated to another table (ex: SALESPERSON validated to the EXM_SALESREP table) will also return the "joined" column specified in the Element Type definition.



Search: EXM_CUSTOMER

Table Alias	Seq	Description	Data Element	Element Type	Description	DType	Data Len
EXM_CUSTOMER	004	Customers	COMPANY	EXM_TEXT_30	Text	Character (C)	30
EXM_CUSTOMER	005	Customers	BILL_ADDR1	EXM_TEXT_30	Text	Character (C)	30
EXM_CUSTOMER	006	Customers	BILL_ADDR2	EXM_TEXT_30	Text	Character (C)	30
EXM_CUSTOMER	007	Customers	CITY	EXM_TEXT_20	Text	Character (C)	20
EXM_CUSTOMER	008	Customers	STATE	EXM_STATE_ID	State ID	Character (C)	2
EXM_CUSTOMER	009	Customers	COUNTRY	EXM_TEXT_20	Text	Character (C)	20
EXM_CUSTOMER	010	Customers	POST_CODE	EXM_POSTCODE	Postal Code	Character (C)	12
EXM_CUSTOMER	011	Customers	PHONE	EXM_PHONE_NO	Phone No	Character (C)	15
EXM_CUSTOMER	012	Customers	FAX	EXM_PHONE_NO	Phone No	Character (C)	15
EXM_CUSTOMER	013	Customers	SALESPERSON	EXM_SALESPERSON	Salesperson	Character (C)	3
EXM_CUSTOMER	014	Customers	SHIP_ZONE	EXM_SHIP_ZONE	Ship Zone	Character (C)	2
EXM_CUSTOMER	015	Customers	SHIP_METHOD	EXM_SHIP_METHOD	Ship Method	Character (C)	5
EXM_CUSTOMER	016	Customers	CURRENT_BAL	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	017	Customers	OVER_30	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	018	Customers	OVER_60	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	019	Customers	OVER_90	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	020	Customers	OVER_120	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	021	Customers	SALES_MTD	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	022	Customers	SALES_YTD	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	023	Customers	SALES_LY	EXM_CURR_12	Currency	Number (N)	12
EXM_CUSTOMER	024	Customers	LAST_PURCH_DATE	EXM_DATE_JUL	Julian Date	Number (N)	7

About the Fields

The Query Definition Detail contains information about the query columns including tables, columns and calculations. The most relevant information is displayed on the detail grid. Each grid row may be expanded using Ctrl+E to display the full query detail record.

Column Type

- DB Column - A previously defined column existing on a Barista table.
- Calculation - A displayed/calculated column not existing in a Barista database.

Table Alias

The table where the specified column resides

Column ID

The column name on the specified Table Alias

Neutral

Instructs Barista to use case neutral searching and sorting when using this column

Element Type

A predefined Barista Element containing display attributes needed for a “Calculation” column.

Col Alias

An alias to be used by the query system when referring to data in this column. Normally used to shorten the text on column names (CUST instead of EXM_CUSTOMER.CUSTOMER_ID).

Column Header
The text displayed in the header portion of the column.

Visible
Indicates the “visible” status for the column

- Always visible - Column is always visible, and cannot be hidden by users
- User selectable - Column may be either displayed ot hidden by users.
- Always hidden - Column is never visible to users.

Column Calculation (Expanded view only)
Calculation used to create column result. See “Defining Calculated Columns” (below).

Inquiry Program (Expanded view only)
Allows users to specify a program to be called after each data row is retrieved. This custom program can be used for additional filtering or row modification prior to display, and is called with the following arguments:

inq_mode$	Not currently used
rec_tpl$	Current inquiry row database columns
calc_tpl$	Current inquiry row calculation columns
table_chans$[]	Barista opened table/channel array
status$	Returned status: ● ‘’ = Display current row ● ‘SKIP’ = Skip current row

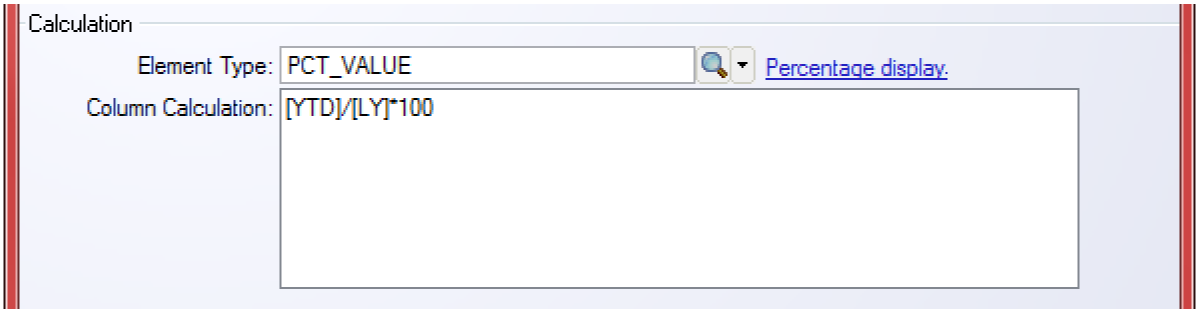
Note: if the inquiry program needs to inherit values/settings that aren't in the enter list, they should be set/retrieved using a namespace rather than an STBL.

Column Filter (Expanded view only)
Will be used to allow additional filtering when retrieving the data set.

Column Total Type (Expanded view only)
The type of total to display for the selected column:

- Sum - display the sum of the values in the column, even if the column contains a calculated value.
- Calculation - Display the value using the defined calcuation for the column.

Defining Calculated Columns



Calculation

Element Type: PCT_VALUE

Percentage display.

Column Calculation: [YTD]/[LY]*100

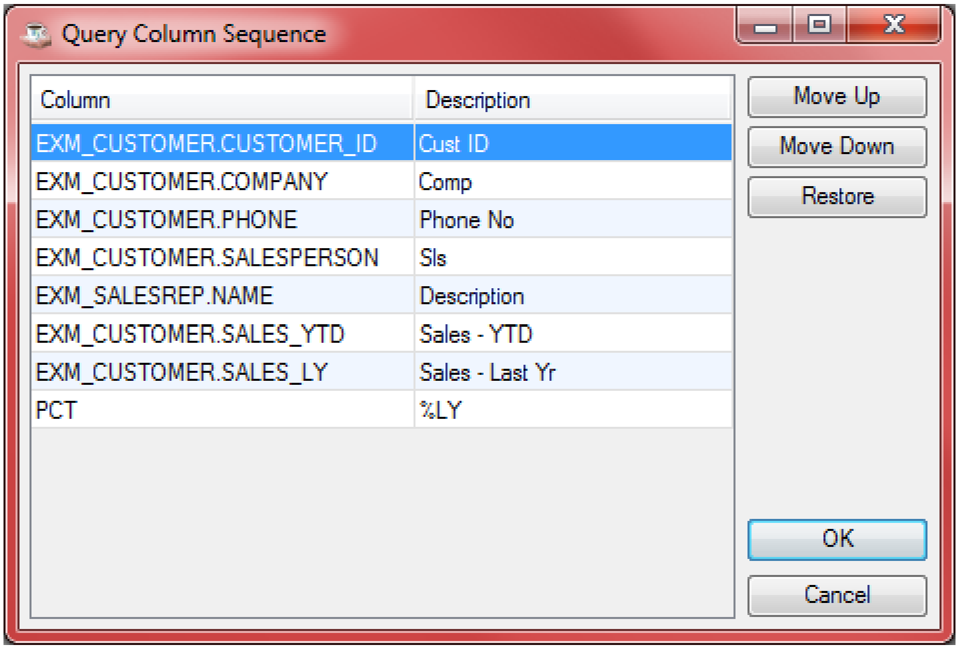
One of the most powerful additions to the query system is the ability to define calculated columns in a query. Select the Column Type and using the drop down tool, select Calculation. Then enter the element type that contains the display attributes for the type of column we are adding to the query.

For example, enter PCT_VALUE and enter the column alias of PCT and for the column heading, label it % LY. This will add a column that calculates the percent of this year's sales versus last year. Select the Expand grid record icon at the bottom of the window to enter the calculation code. The calculation code uses the same syntax found in the rest of Barista. We simply enclose the calculation field column alias in brackets. In the example, the calculation is **[YTD] /[LY]*100**.

Query Column Sequence

The next step is to set the default column sequence. This is done by accessing the additional options menu with a right click and selecting **Set Column Sequence**.

Move the columns to their desired location by either selecting a column and pressing Move Up/Down, or double-clicking the column grid row, and drag/drop it in the new location. Press “OK” to save the changes. The new column sequence is now displayed in the Query Definitions form.



Column	Description
EXM_CUSTOMER.CUSTOMER_ID	Cust ID
EXM_CUSTOMER.COMPANY	Comp
EXM_CUSTOMER.PHONE	Phone No
EXM_CUSTOMER.SALESPERSON	Sls
EXM_SALESREP.NAME	Description
EXM_CUSTOMER.SALES_YTD	Sales - YTD
EXM_CUSTOMER.SALES_LY	Sales - Last Yr
PCT	%LY

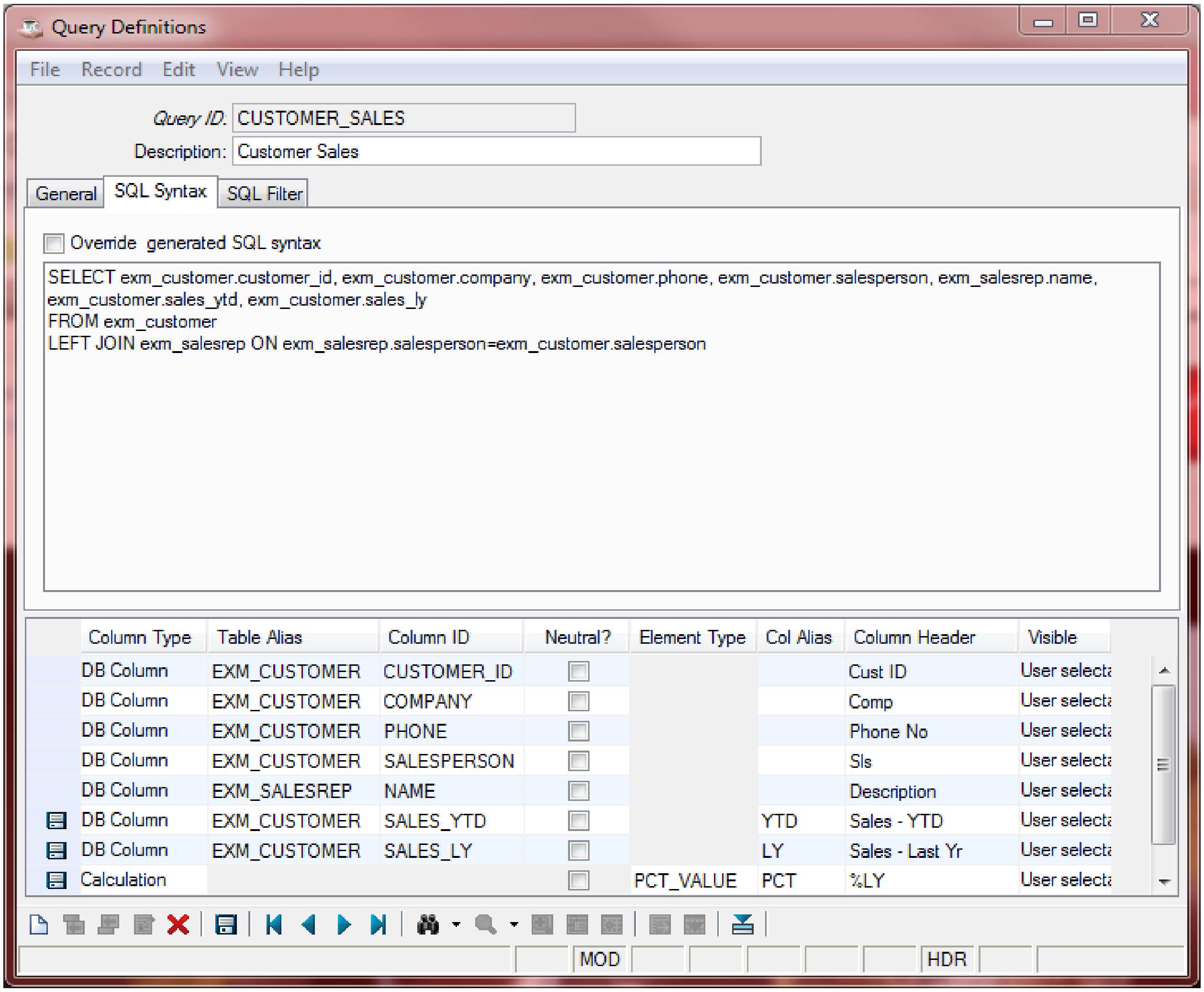
Defining the SQL Select Statement

Access the additional options menu with a right click and select **Generate SQL.** Barista automatically defines the SQL based on the data entered in the column detail, including an unlimited number of SQL JOINS.

Modifications to the SQL can be entered by simply checking the Override generated SQL syntax check box. In certain instances, Barista may not be able to determine the primary key columns for joined tables. In this case, the column names will be enclosed in angled brackets (<column>) indicating modifications are required for the query to process correctly.

Stored procedures and/or sub-queries can be added at this step. Note, the modified SQL should NOT contain ‘WHERE’ and ‘ORDER BY’ clauses. Barista generates them as selection criteria is entered. If WHERE filtering is required when calling the query from a custom program, use the filter_defs$[] array described in the “***Accessing Defined Queries in Applications”*** section below.

Select the Save icon.



Query ID: CUSTOMER_SALES

Description: Customer Sales

☐ Override generated SQL syntax

```
SELECT exm_customer.customer_id, exm_customer.company, exm_customer.phone, exm_customer.salesperson, exm_salesrep.name,
exm_customer.sales_ytd, exm_customer.sales_ly
FROM exm_customer
LEFT JOIN exm_salesrep ON exm_salesrep.salesperson=exm_customer.salesperson
```

	Column Type	Table Alias	Column ID	Neutral?	Element Type	Col Alias	Column Header	Visible
	DB Column	EXM_CUSTOMER	CUSTOMER_ID	☐			Cust ID	User selecta
	DB Column	EXM_CUSTOMER	COMPANY	☐			Comp	User selecta
	DB Column	EXM_CUSTOMER	PHONE	☐			Phone No	User selecta
	DB Column	EXM_CUSTOMER	SALESPERSON	☐			Sls	User selecta
	DB Column	EXM_SALESREP	NAME	☐			Description	User selecta
▤	DB Column	EXM_CUSTOMER	SALES_YTD	☐		YTD	Sales - YTD	User selecta
▤	DB Column	EXM_CUSTOMER	SALES_LY	☐		LY	Sales - Last Yr	User selecta
▤	Calculation			☐	PCT_VALUE	PCT	%LY	User selecta

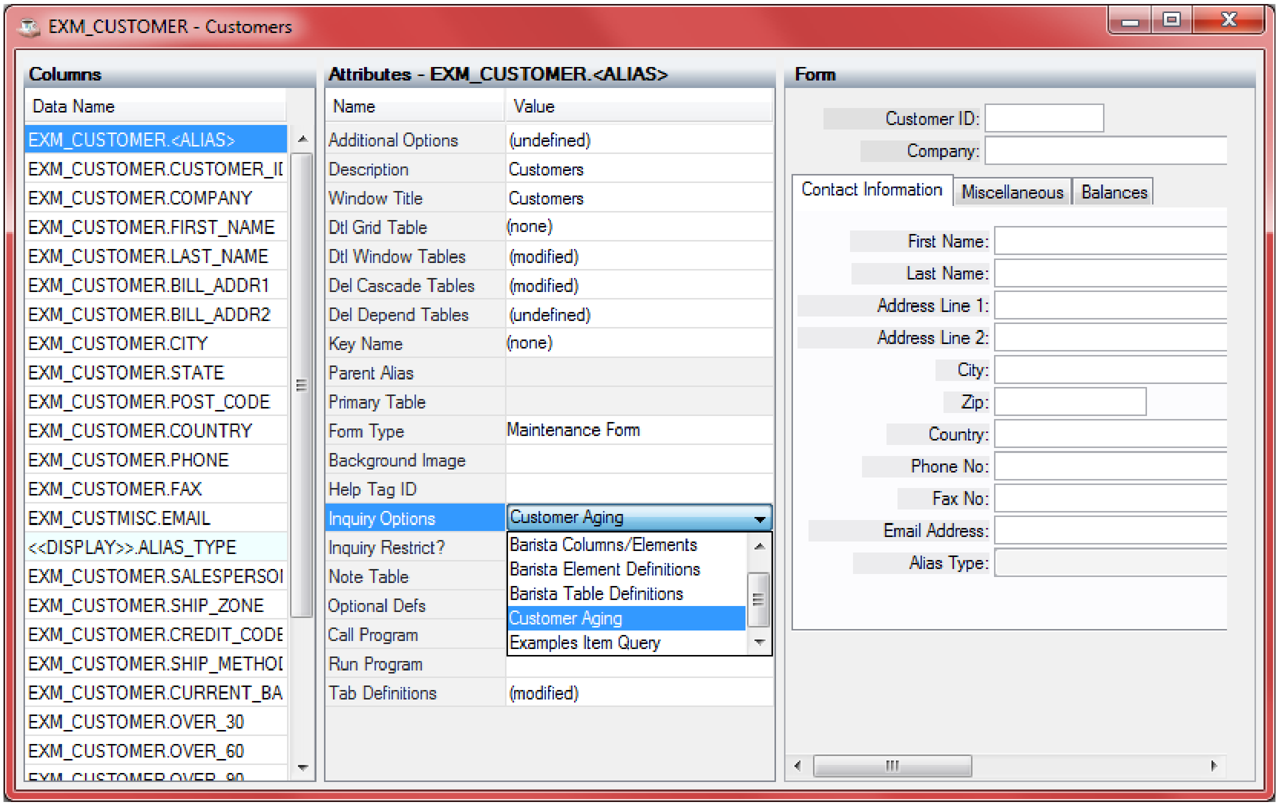
Test the Query Definition

Access the additional options menu with a right click and select **Test Query**. Barista compiles the definition and displays the query including column definitions, any JOINS, column calculations and total columns.

Adding the Defined Query to a Form

Launch the **Barista Form Manager** using the button at the top of the MDI frame. Double-click on the form (EXM_CUSTOMER in our example) to launch the **Barista Form Designer**.

By default the selection will launch the standard Barista inquiry system automatically generated for each form. Select **Inquiry Options** then select the new value from the list (in our example Customer Sales) to replace that inquiry with the new defined query.



EXM_CUSTOMER - Customers

Columns

Data Name
EXM_CUSTOMER.<ALIAS>
EXM_CUSTOMER.CUSTOMER_II
EXM_CUSTOMER.COMPANY
EXM_CUSTOMER.FIRST_NAME
EXM_CUSTOMER.LAST_NAME
EXM_CUSTOMER.BILL_ADDR1
EXM_CUSTOMER.BILL_ADDR2
EXM_CUSTOMER.CITY
EXM_CUSTOMER.STATE
EXM_CUSTOMER.POST_CODE
EXM_CUSTOMER.COUNTRY
EXM_CUSTOMER.PHONE
EXM_CUSTOMER.FAX
EXM_CUSTMISC.EMAIL
<<DISPLAY>>.ALIAS_TYPE
EXM_CUSTOMER.SALESPERSOI
EXM_CUSTOMER.SHIP_ZONE
EXM_CUSTOMER.CREDIT_CODE
EXM_CUSTOMER.SHIP_METHOI
EXM_CUSTOMER.CURRENT_BA
EXM_CUSTOMER.OVER_30
EXM_CUSTOMER.OVER_60
EXM_CUSTOMER.OVER_90

Attributes - EXM_CUSTOMER.<ALIAS>

Name	Value
Additional Options	(undefined)
Description	Customers
Window Title	Customers
Dtl Grid Table	(none)
Dtl Window Tables	(modified)
Del Cascade Tables	(modified)
Del Depend Tables	(undefined)
Key Name	(none)
Parent Alias	
Primary Table	
Form Type	Maintenance Form
Background Image	
Help Tag ID	
Inquiry Options	Customer Aging
Inquiry Restrict?	
Note Table	
Optional Defs	
Call Program	
Run Program	
Tab Definitions	(modified)

Form

Customer ID:
Company:

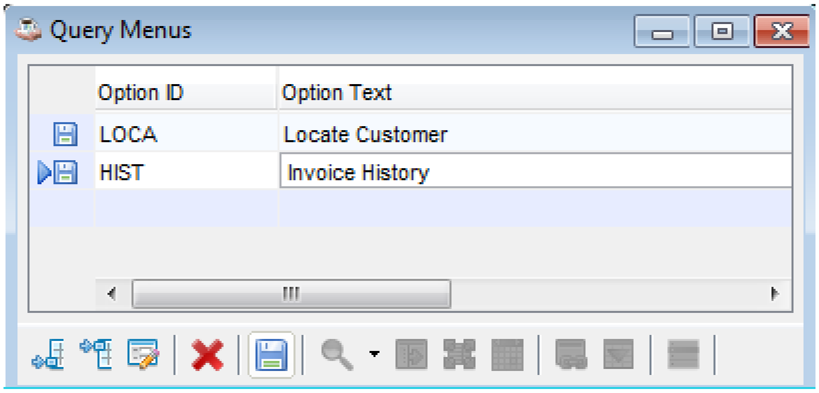
First Name:
Last Name:
Address Line 1:
Address Line 2:
City:
Zip:
Country:
Phone No:
Fax No:
Email Address:
Alias Type:

Select the **Run** process arrow button located at the top of the MDI frame to launch the form. Continuing with the example, the Customers form displays. Selecting the Record query (binocular) button displays the newly defined query.

The same holds true for column queries. As new queries are defined, select a column attribute by selecting **Inquiry Option** and then selecting the desired query from the defined query list.

Query Menus

Items can be added to the pop-up menu that appears when the user right-clicks in the query grid or selects the Additional Options menu. Specify a four-character code and description for each menu item.



Query Menus

Option ID	Option Text
LOCA	Locate Customer
HIST	Invoice History

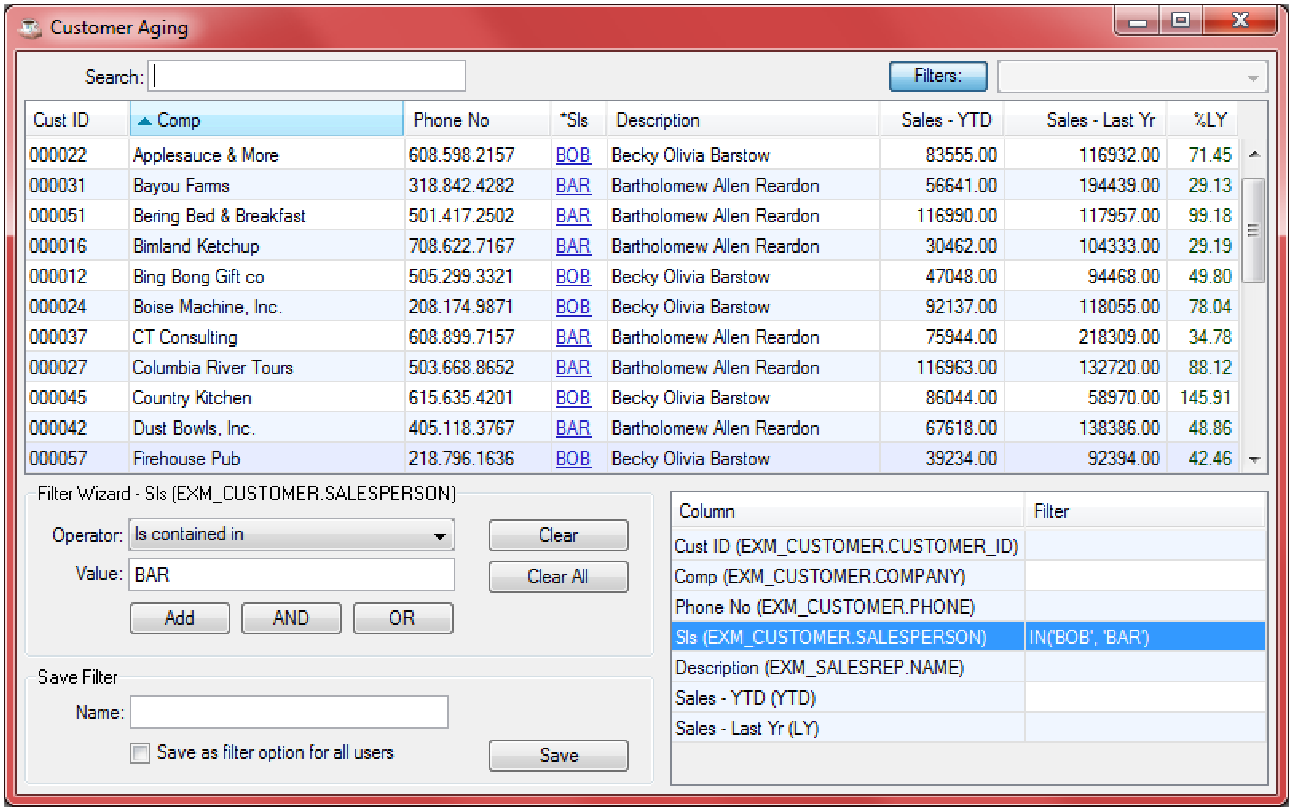
When the query is run and a menu item is selected, Barista calls the Inquiry Program specified in the Query Definition Header with the arguments shown below. Use a switch/case or other parsing code inside the program to determine which option the user selected (e.g., OPT-LOCA, OPT-HIST based on the screenshot above) and perform the desired function: launching a different Barista form, an external application such as a web page, etc.

rd_gui_dev	GUI device channel
rdInqWin!	Calling inquiry form object
rd_inq_event$	Query triggering call (OPT-xxxx from Query Menu, or SELECT if row(s) have been selected in the query grid and the [SELECT] token is used.)
rd_alias_id$	Primary inquiry table
rd_sel_key$	Selected inquiry key
rd_table_chans$[]	Opened table/channels array
rd_status$	Return status for query display (UPDATE = update selected row(s), REFRESH = refresh entire grid)

The skeleton code below shows an example structure for the inquiry program:

```
  setesc std_error
  seterr std_error

  enter rd_gui_dev, rdInqWin!, rd_inq_event$, rd_alias_id$, rd_sel_key$, rd_table_chans$[all],
rd_status$

rem --- Route inquiry event action
  rd_sel_key$=rd_sel_key$(1,pos("^"=rd_sel_key$+"^")-1)

  rdInqEvent!=rd_inq_event$
  switch (1)
    case rdInqEvent!.equals("SELECT")
      rem - user has selected grid row(s)
      rem - requires [SELECT] token in Query Definition Header Inquiry Program field
      <<your code here>>
    break
    case rdInqEvent!.equals("OPT-<OPT1>")
      rem - user selected <OPT1> menu option
      <<your code here>>
    break
    case rdInqEvent!.equals("OPT-<OPT2>")
      rem - user selected <OPT2> menu option
      <<your code here>>
    break
    case default
    break
  swend
  endif

  exit
```

Review of New Features of Query Definition System



Search:

Cust ID	▲ Comp	Phone No	*Sls	Description	Sales - YTD	Sales - Last Yr	%LY
000022	Applesauce & More	608.598.2157	BOB	Becky Olivia Barstow	83555.00	116932.00	71.45
000031	Bayou Farms	318.842.4282	BAR	Bartholomew Allen Reardon	56641.00	194439.00	29.13
000051	Bering Bed & Breakfast	501.417.2502	BAR	Bartholomew Allen Reardon	116990.00	117957.00	99.18
000016	Bimland Ketchup	708.622.7167	BAR	Bartholomew Allen Reardon	30462.00	104333.00	29.19
000012	Bing Bong Gift co	505.299.3321	BOB	Becky Olivia Barstow	47048.00	94468.00	49.80
000024	Boise Machine, Inc.	208.174.9871	BOB	Becky Olivia Barstow	92137.00	118055.00	78.04
000037	CT Consulting	608.899.7157	BAR	Bartholomew Allen Reardon	75944.00	218309.00	34.78
000027	Columbia River Tours	503.668.8652	BAR	Bartholomew Allen Reardon	116963.00	132720.00	88.12
000045	Country Kitchen	615.635.4201	BOB	Becky Olivia Barstow	86044.00	58970.00	145.91
000042	Dust Bowls, Inc.	405.118.3767	BAR	Bartholomew Allen Reardon	67618.00	138386.00	48.86
000057	Firehouse Pub	218.796.1636	BOB	Becky Olivia Barstow	39234.00	92394.00	42.46

Filter Wizard - Sls (EXM_CUSTOMER.SALESPERSON)

Operator: Is contained in

Value: BAR

Save Filter

Name:

□ Save as filter option for all users

Column	Filter
Cust ID (EXM_CUSTOMER.CUSTOMER_ID)	
Comp (EXM_CUSTOMER.COMPANY)	
Phone No (EXM_CUSTOMER.PHONE)	
Sls (EXM_CUSTOMER.SALESPERSON)	IN('BOB', 'BAR')
Description (EXM_SALESREP.NAME)	
Sales - YTD (YTD)	
Sales - Last Yr (LY)	

Selection and sorting

- Hyperlinks in tables are now underlined. Simply click and the target information is displayed.
- Multiple column sorting:
  - Select a primary sort column. Pressing Shift and selecting additional columns will add them to the secondary sorting. Hover over any of the column headers to view the current sort columns.
  - To reverse sorting, hold Shift and select the column
  - To break multiple sorting, select a single column without holding the Shift key
  - Sorting can be done on any defined column except for calculations
  - Can also sort on the secondary description pulled from joined tables

Filtering

Filtering is now in-line...data is redisplayed as filter criteria is added.

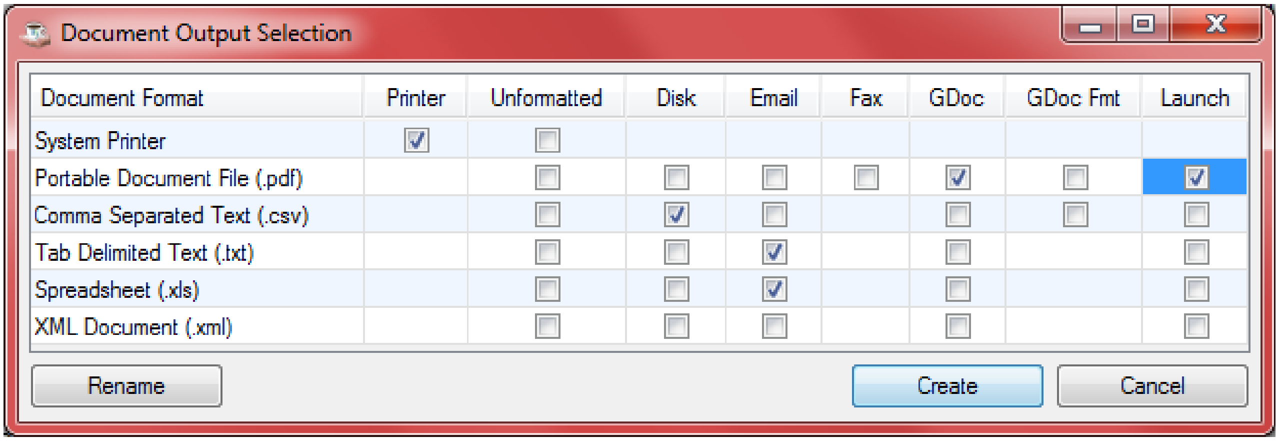
- Select the Filters button to display the Filter panel. It contains three sections:
  - Filter Wizard to the left allows filter definitions to be built step-by-step
  - Save Filter allows filters to be named and saved with the option to share the filter to all users. Saved filters are added to the Filters drop down list.
  - Manual filter grid to the right to manually enter filter information

- The “Is Contained In” filter in the Wizard drop down allows for the selection of a series of values, and is the equivalent of multiple “equal to” clauses.

Document Output (DocOut) Options

The new query system also takes advantage of the powerful Barista DocOut system.

- Right click on the grid and select Print to send the query in its current state to the DocOut system where multiple output types are available:
  - Create pdf, csv, text, xls, and xml files in formatted or unformatted on disc
  - Send emails, faxes, upload to Google Docs
  - Launch the target document once the processing is completed



Document Format	Printer	Unformatted	Disk	Email	Fax	GDoc	GDoc Fmt	Launch
System Printer	☑	□						
Portable Document File (.pdf)		□	□	□	□	☑	□	☑
Comma Separated Text (.csv)		□	☑	□		□	□	□
Tab Delimited Text (.txt)		□	□	☑		□		□
Spreadsheet (.xls)		□	□	☑		□		□
XML Document (.xml)		□	□	□		□		□

- Quickly send to Google Docs by selecting the drop down control next to the Save button at the top of the query preview window

Direct Output from Grid

Direct output linking bypasses the normal DocOut steps to quickly create a document type from a grid. Right click on the grid and select **Export Records** to open a list of output types. Make a selection and Barista automatically launches the application associated with the selected output type.

For example, selecting the .xls output type launches Microsoft Excel. Saved documents can be accessed later using the Barista Document Inquiry System and searching by document number or saved name.

Accessing Defined Queries in Applications

Defined queries can be accessed two different ways from custom application code:

“Standalone” mode: To launch a query in “standalone” mode, please see the Barista Launch Task documentation.

Called: To call a defined query, use the following program and arguments:

call STBL("+DIR_SYP")+"bax_query.bbj", gui_dev, form!, query_id$, query_mode$, table_chans$[all], sel_key$

gui_dev	Previously opened GUI device channel (Required)
form!	Form object of calling task. Null() if directly launched. (Required)
query_id$	Query definition ID (Required)
query_mode$	Inquiry mode (Optional): DEFAULT - rd_search_defs$[all] contains default value from calling control TEST - Test mode from Query maintenance form. Exits on SQL error
table_chans$[]	Barista opened table/channel array
sel_key$	Selected key (if any) returned to calling program
filter_defs$[]	Two dimensional array containing preset filter definitions. n = filter number. (Optional) [n,0] - Filter <alias>.<column> [n,1] - SQL filter syntax for column. Examples: ● ='100' to find items equal to '100' ● LIKE '503%' to find items starting with '503' ● LIKE '%503%' to find items containing '503' [n,2] - "LOCK" to disallow changes to this filter (future)
search_defs$[]	One dimensional array containing preset search definitions. (Optional) [0] - Search <alias>.<column> (blank = default primary column) [1] - Partial search text (case sensitive) [2] - Sort order: <A>scending (default) or <D>escending