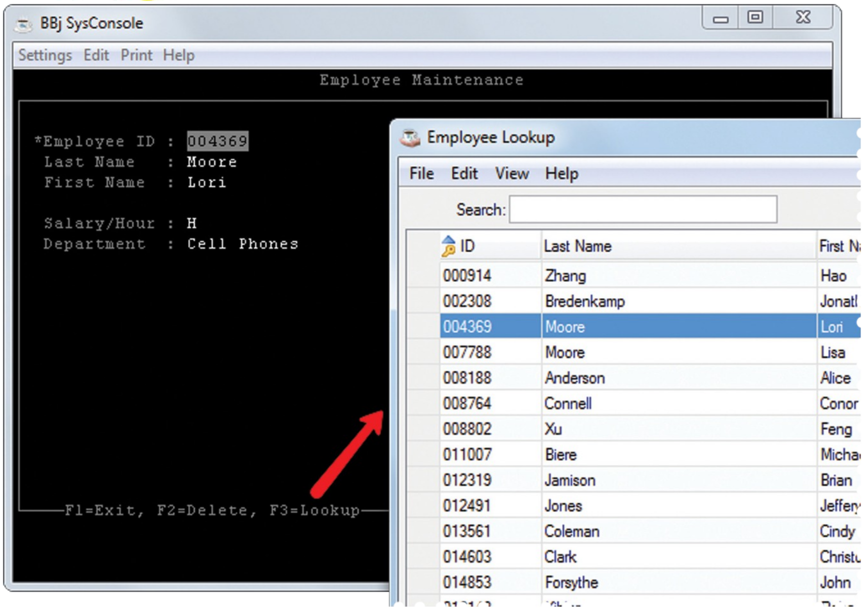
Barista Caffeinates a CUI App With GUI Sprinkles

By Ralph Lance
Software Engineer

Legacy applications can easily contain thousands of programs and data files created over many years. When making the decision to bring them all into the graphical world of today's applications, the task can be overwhelming if not taken one step at a time. This article gives an overview of how to begin integrating fully-functional graphical components with the least amount of effort using the Barista® Application Framework and instantly begin benefiting from the myriad of built-in features that Barista brings to your application.

To begin this overview, start with the ubiquitous grid inquiry that benefits most from a graphical user interface. For another example of integrating Barista functionality into a legacy application, read *DocOut Easily Modernizes BBx Report Output.*



```
                         Employee Maintenance

 *Employee ID : 004369
  Last Name   : Moore
  First Name  : Lori

  Salary/Hour : H
  Department  : Cell Phones

 ─────F1=Exit, F2=Delete, F3=Lookup─
```

ID	Last Name	First N
000914	Zhang	Hao
002308	Bredenkamp	Jonatl
004369	Moore	Lori
007788	Moore	Lisa
008188	Anderson	Alice
008764	Connell	Conor
008802	Xu	Feng
011007	Biere	Micha
012319	Jamison	Brian
012491	Jones	Jeffery
013561	Coleman	Cindy
014603	Clark	Christu
014853	Forsythe	John

Figure 1. The CD-Jazz Store CUI program with the GUI lookup

The "CD-Jazz Store," a simple BBx® character-based application containing 5 data files and 4 programs, is the application we will use for this illustration. The programs work together to provide minimal file maintenance capabilities for the employee data. After applying our Barista sprinkles to the CUI application, **Figure 1** shows the resulting GUI lookup launching from a keypress.

Although the user can easily navigate to the previous and next record in the CUI version using the [Page Up] and [Page Down] keys respectively, there is no provision for quickly finding one of the over 800 employees in the file if you don't know the employee's ID. This is exactly the deficit this article addresses.

Caffeinate

Accompany me as I take you through the general steps to the solution. If you would like more detail to these steps, check out the links at the end of this article and consider attending a Barista training class.

Step 1: Create a Barista application

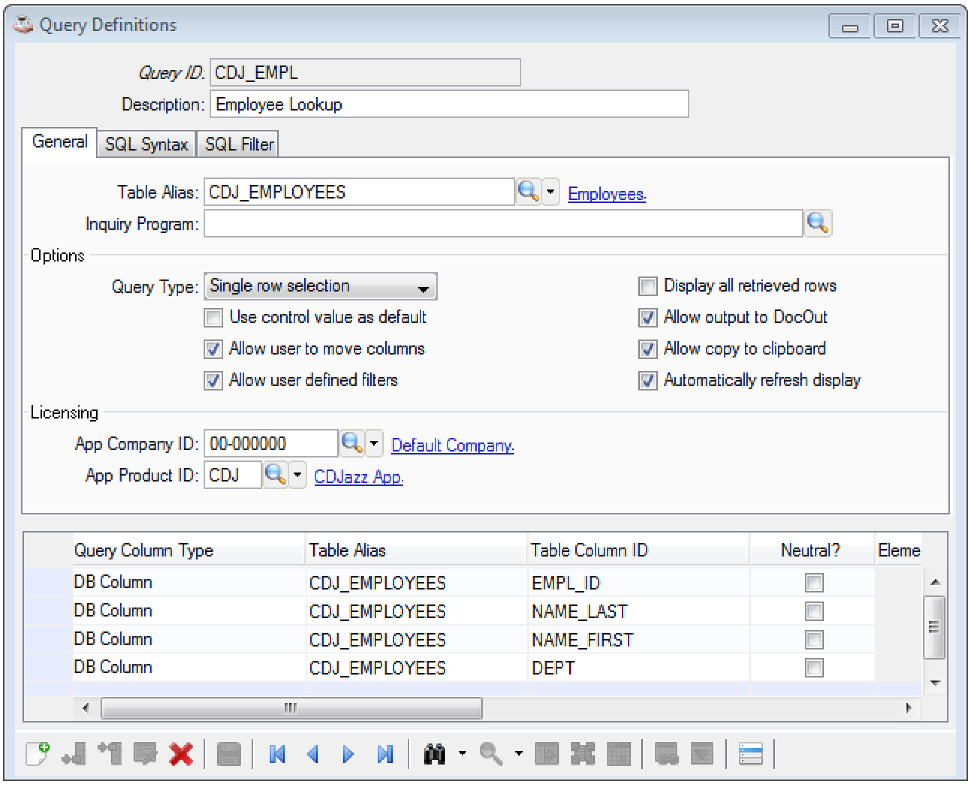
The first step in working with Barista is to create an application. For detailed "how to" steps, refer to Create Vertical and Customize Applications - Part 1.

Step 2: Import the application's data dictionary into Barista

Using the "Import to Barista Dictionary" function, import the BASIS data dictionary that describes the tables of our CD-Jazz Store application. If your application does not have a data dictionary, simply make one for the table(s) you will be importing – a process that can be as simple as cutting and pasting the string template for your table (file) from your program code into the BASIS Data Dictionary editor in Enterprise Manager. The import creates element types and table definitions that Barista uses for its maintenance forms, reporting, and inquiry systems. It also creates a beginning menu for our application. For our purposes, we will just import the one employee table used in the query.

Step 3: Create a custom query

Now that we have defined our application tables in Barista, we can create the query (**Figure 2**) that we want to bind into our existing character application by running "Query Definitions" in the "Barista Development" menu and describing the columns (fields) for our query.



Query ID: CDJ_EMPL

Description: Employee Lookup

Table Alias: CDJ_EMPLOYEES Employees.

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

☑ Automatically refresh display

Licensing

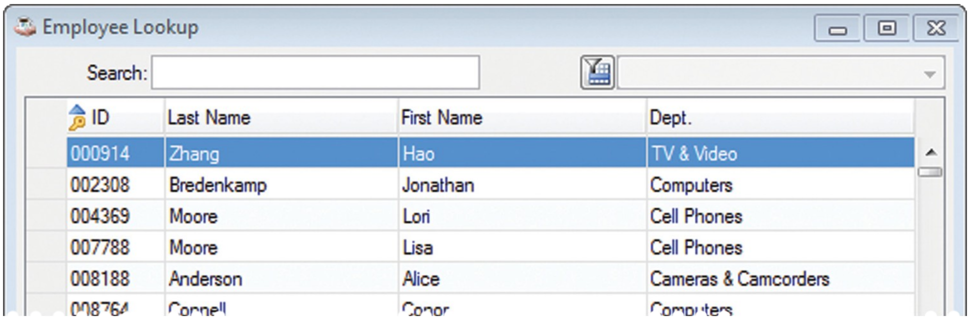
App Company ID: 00-000000 Default Company.

App Product ID: CDJ CDJazz App.

Query Column Type	Table Alias	Table Column ID	Neutral?	Eleme
DB Column	CDJ_EMPLOYEES	EMPL_ID	□	
DB Column	CDJ_EMPLOYEES	NAME_LAST	□	
DB Column	CDJ_EMPLOYEES	NAME_FIRST	□	
DB Column	CDJ_EMPLOYEES	DEPT	□	

Figure 2. Create an employee lookup query

After Barista generates the SQL statement, we can immediately test the query as shown in **Figure 3**.



Search:

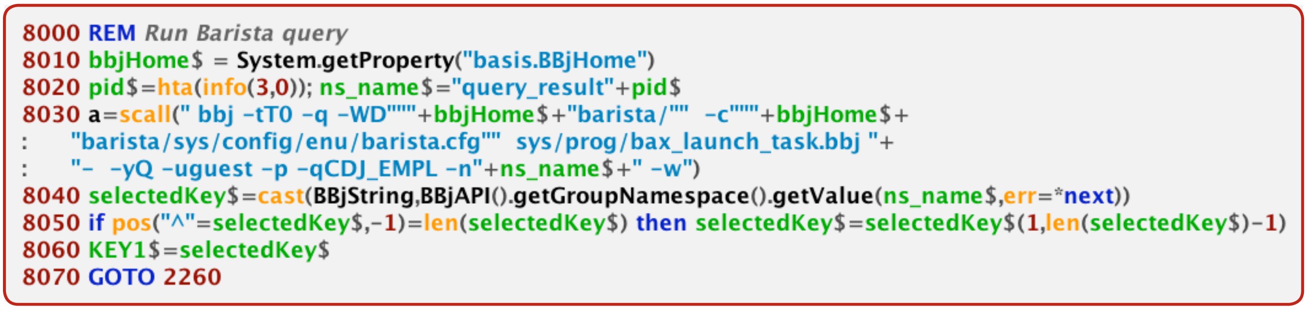
ID	Last Name	First Name	Dept.
000914	Zhang	Hao	TV & Video
002308	Bredenkamp	Jonathan	Computers
004369	Moore	Lori	Cell Phones
007788	Moore	Lisa	Cell Phones
008188	Anderson	Alice	Cameras & Camcorders
008764	Cornell	Conor	Computers

Figure 3. The result of the new employee query

Step 4: Incorporate the new lookup function into the character app

Next, we add a new [F3] lookup function to the employee maintenance screen and have it invoke our custom query via an SCALL to a Barista program and respond with the employee that the user selects. In order to retrieve the user selection from the query, we use a mechanism called a “namespace” with which two or more BBj programs can share data and event notification.

Figure 4 shows the code block that sets up and launches the Barista query after the user presses [F3].



```
8000 REM Run Barista query
8010 bbjHome$ = System.getProperty("basis.BBjHome")
8020 pid$=hta(info(3,0)); ns_name$="query_result"+pid$
8030 a=scall(" bbj -tT0 -q -WD"""+bbjHome$+"barista/"" -c"""+bbjHome$+
:     "barista/sys/config/enu/barista.cfg"" sys/prog/bax_launch_task.bbj "+
:     "- -yQ -uguest -p -qCDJ_EMPL -n"+ns_name$+" -w")
8040 selectedKey$=cast(BBjString,BBjAPI().getGroupNamespace().getValue(ns_name$,err=*next))
8050 if pos("^"=selectedKey$,-1)=len(selectedKey$) then selectedKey$=selectedKey$(1,len(selectedKey$)-1)
8060 KEY1$=selectedKey$
8070 GOTO 2260
```

Figure 4. Code to launch the Barista query

Let's take a look at this code block and dissect it by line number.

8010 determines the BBj installation directory

8020 sets up a unique name for the group namespace variable that we would like to use for the return value

8030 invokes the Barista query via an SCALL command. The `-w` argument in the SCALL command tells Barista that we will wait until the user dismisses the query, either by closing the query window or making a selection. Read more about it in the “Barista Launch Task” article noted at the end of this article for a detailed explanation of the launch program arguments.

8040 and **8050** query the group namespace variable to retrieve the selected employee stored in its value. If the user closes the query without choosing an employee, the namespace variable will not exist (err=) or be empty. The code that scans for the caret character (^) handles multiple selections, as Barista queries optionally allow for multiple selections separated by the caret.

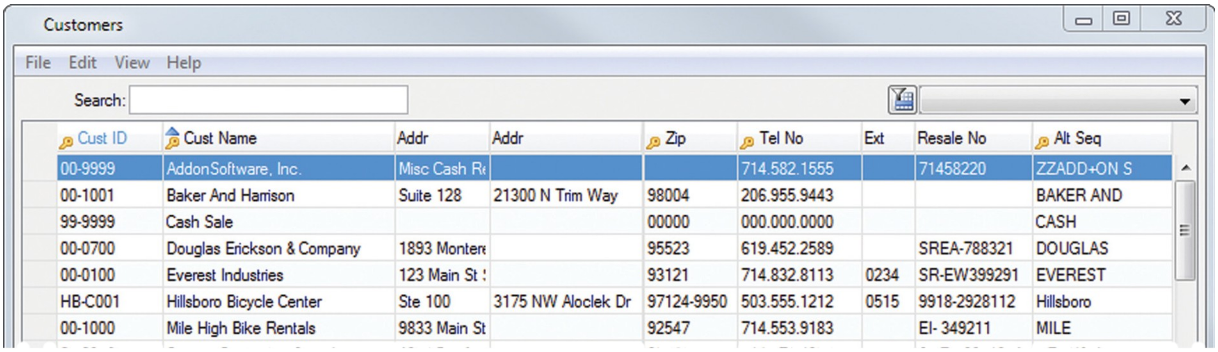
8060 and **8070** load the employee ID into the same input variable just as though the user had actually keyed in the ID in the CUI program.

The program continues normally, displaying the employee data in the CUI program previously shown in **Figure 1**.

Add Sprinkles

While the ease and efficiency of this GUI lookup is worthy on its own merit, this it really just the tip of the “whipped cream”-berg. Barista provides a whole new world of query and reporting capabilities to users who are now empowered in ways not previously possible. What might have required assistance in the past from an IT person, now can easily be completed without writing programs or purchasing a third party application. In addition, this new capability offers several output types for the grid data – output to a system printer or pdf, csv, xml, xls, and txt file formats.

Barista’s query for grids provide multiple methods to search, sort, filter, adjust column display, adjust column order, adjust column width, and to select all or only highlighted rows of the grid for output. The sample grid in **Figure 5** shows a typical layout with the Search bar in the upper left and the filter wizard button ▦ in the upper right. Special icons appear next to the column headers to give more information about that column; the blue triangle ▲ shows the “sorted by” column, an asterisk would indicate that a filter has been applied to the column, and the key icon 🔑 indicates that the field is indexed, which will sort faster than non-indexed fields.



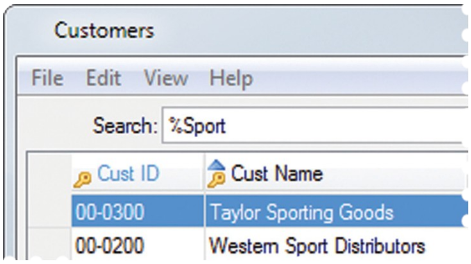
Search:

Cust ID	Cust Name	Addr	Addr	Zip	Tel No	Ext	Resale No	Alt Seq
00-9999	AddonSoftware, Inc.	Misc Cash Re			714.582.1555		71458220	ZZADD+ON S
00-1001	Baker And Harrison	Suite 128	21300 N Trim Way	98004	206.955.9443			BAKER AND
99-9999	Cash Sale			00000	000.000.0000			CASH
00-0700	Douglas Erickson & Company	1893 Montere		95523	619.452.2589		SREA-788321	DOUGLAS
00-0100	Everest Industries	123 Main St :		93121	714.832.8113	0234	SR-EW399291	EVEREST
HB-C001	Hillsboro Bicycle Center	Ste 100	3175 NW Aloclek Dr	97124-9950	503.555.1212	0515	9918-2928112	Hillsboro
00-1000	Mile High Bike Rentals	9833 Main St		92547	714.553.9183		EI- 349211	MILE

Figure 5. Sample customer master grid

Search/Sort

The Search feature creates a quick case-sensitive or case-insensitive search filter of the query grid. As soon as the user selects the column header and then begins typing the search characters in the Search bar, the filtering begins. The % is a wildcard character so entering '%Sport' will return all strings in the selected column that contain the word 'Sport' (see **Figure 6**). To restore the grid back to displaying the full result set, the user would simply clear the contents of the Search field.



Search: %Sport

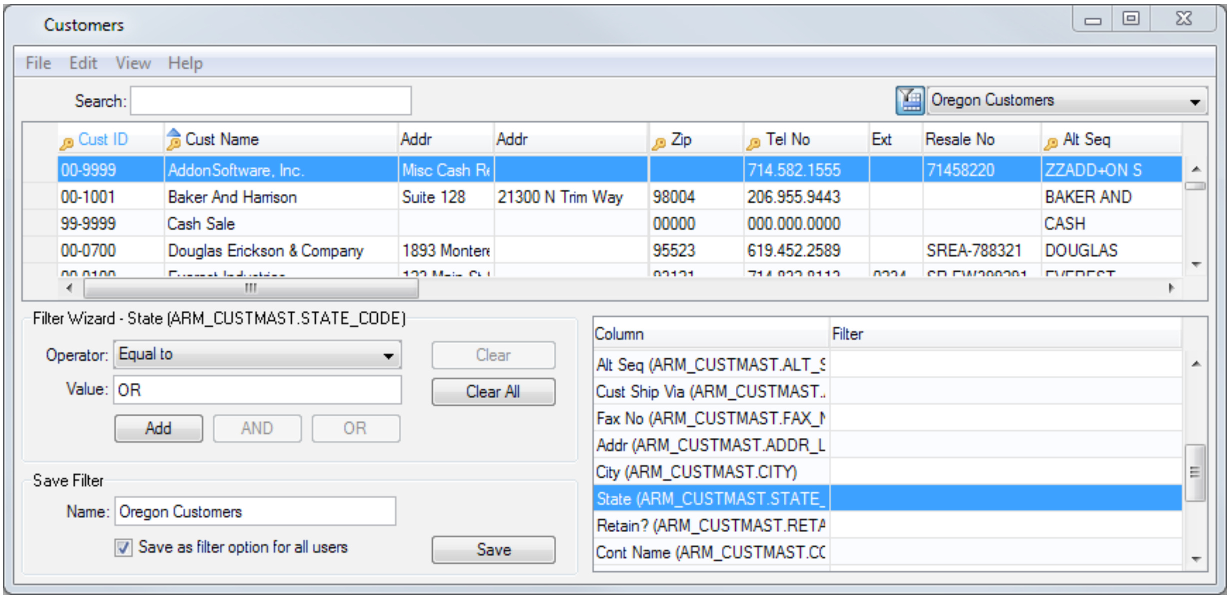
Cust ID	Cust Name
00-0300	Taylor Sporting Goods
00-0200	Western Sport Distributors

Figure 6. Wildcard search result

Users can select a single column header for an ascending or descending sort or can select multiple columns by pressing the [Shift] key while selecting the columns to sort on multiple columns.

Filter

A Filter Wizard allows users to create and save custom filters quickly and easily. Select the [**Filter Wizard**] button in the top right corner of the query grid frame to launch the wizard. The Wizard (shown in **Figure 7**) requires user input for the Operator, Value, and Column.



Search:

Oregon Customers

Cust ID	Cust Name	Addr	Addr	Zip	Tel No	Ext	Resale No	Alt Seq
00-9999	AddonSoftware, Inc.	Misc Cash Re			714.582.1555		71458220	ZZADD+ON S
00-1001	Baker And Harrison	Suite 128	21300 N Trim Way	98004	206.955.9443			BAKER AND
99-9999	Cash Sale			00000	000.000.0000			CASH
00-0700	Douglas Erickson & Company	1893 Montere		95523	619.452.2589		SREA-788321	DOUGLAS

Filter Wizard - State (ARM_CUSTMAST.STATE_CODE)

Operator: Equal to

Value: OR

Save Filter

Name: Oregon Customers

☑ Save as filter option for all users

Column	Filter
Alt Seq (ARM_CUSTMAST.ALT_S	
Cust Ship Via (ARM_CUSTMAST..	
Fax No (ARM_CUSTMAST.FAX_N	
Addr (ARM_CUSTMAST.ADDR_L	
City (ARM_CUSTMAST.CITY)	
State (ARM_CUSTMAST.STATE_	
Retain? (ARM_CUSTMAST.RETA	
Cont Name (ARM_CUSTMAST.CC	

Figure 7. Filter Wizard

The Operator options appear in a dropdown list containing Greater than, Less than, Greater than/Equal to, Equal to, Not equal to, Begins with, Ends with, Contains, Does not contain, Is contained in, and Is not contained in.

To reuse a defined filter, the user may save the filter by giving it a name, and further mark whether to share the custom filter to all users in the organization. The shared filter will appear as a dropdown option next to the Filter Wizard button in the top right corner of the frame. See the filter "Oregon Customers" displayed in **Figure 7**.

Query Selection Options

Barista provides additional options when right clicking on a record in the grid (**Figure 8**). While all of these options are valuable, the Inquiry Columns function is one worth looking at more closely.

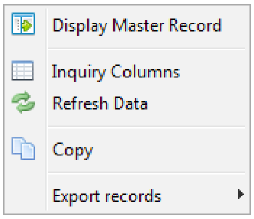
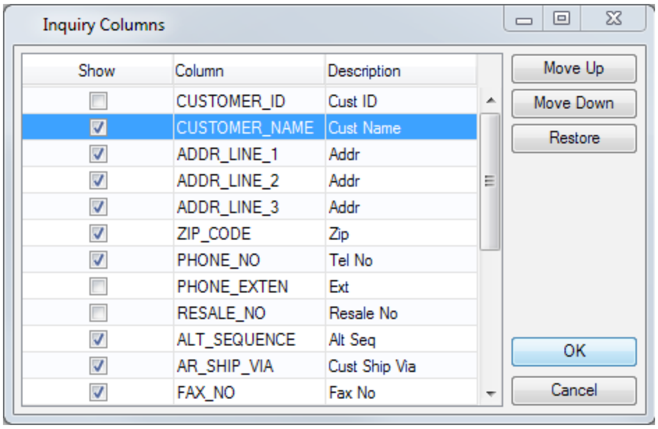


Figure 8. Query selection options

Selecting Inquiry Columns (**Figure 9**) allows users to change which columns display and in what order by marking the Show checkbox and then moving the item into the desired position using either the [Move Up]/[Move Down] buttons or dragging and dropping the highlighted column. All edits are preserved at the user level when returning to the grid. To undo selection and return to the default column view, simply click the [Restore] button.



Show	Column	Description
□	CUSTOMER_ID	Cust ID
☑	CUSTOMER_NAME	Cust Name
☑	ADDR_LINE_1	Addr
☑	ADDR_LINE_2	Addr
☑	ADDR_LINE_3	Addr
☑	ZIP_CODE	Zip
☑	PHONE_NO	Tel No
□	PHONE_EXTEN	Ext
□	RESALE_NO	Resale No
☑	ALT_SEQUENCE	Alt Seq
☑	AR_SHIP_VIA	Cust Ship Via
☑	FAX_NO	Fax No

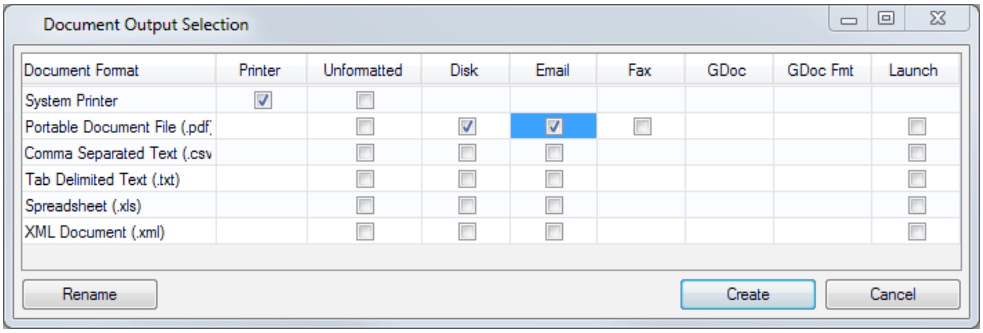
Figure 9. Inquiry Columns window

Other Query selection options include –

- Copy - copies the columns to the clipboard, allowing the user to choose whether to include the database table/column names and text column headings, and to select the column delimiter and text identifier characters.
- Export Records - quickly exports the contents of the grid or selected columns while offering several export options; Document Output Viewer, Document Output Selection Form, as well as the type of output.

Output

After sorting and filtering and massaging the data to the user's delight, the user can easily output the results in a variety of ways by just making one or more selections in the the Document Output Selection window (**Figure 10**).



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)		□	□	□				□

Figure 10. Document Output Selection window

Gaining all of this functionality handily meets most all of any user's simple report and output needs or wishes.

Summary

BASIS developers can use this technique to ease their users into graphical user interfaces with minimal effort and disruption to existing code by leveraging the out-of-the-box functionality Barista provides. With the addition of as little as ten lines of code to a CUI or GUI application and a few minutes spent configuring a Barista query, you can delight your users by delivering untold productivity gains to their daily tasks. In fact, with the query's sorting and filtering capabilities as well as flexible output of the grid contents or selections in various formats and mechanisms such as PDF, XLS, fax or email, users get a big bang for their buck, all with very little effort on the part of the developer. You'll be their Barista of choice to caffeinate their applications! ■

- Review *DocOut Easily Modernizes BBx Report Output*
- For more information, refer to these Barista resources:
  - Create Vertical and Customize Applications - Part 1, Part 2, Part 3
  - Create and Synchronize Applications
  - BASIS Data Dictionary Import
  - Query Definition System
  - Barista Launch Task