NOMADS QUERY AND REPORT WRITER ENHANCEMENTS 2020 – 2024

DireXions 2024

NOMADS QUERY ENHANCEMENTS 2020 – 2024

DireXions 2024

AGENDA

1. **Features**

2. **Query+ Options**

3. **Query-Based Utilities**

BRIDGING THE PAST AND THE FUTURE

FEATURES

Named Formulas

- Assign a name to a formula

- Easily identify a formula throughout the Query Subsystem

(Columns list, Filters, Formulas, Show/Hide/Reorder Columns,

Chart Definitions, Copy/Export, Profile columns displays, etc.)

- Now you can:
  - Use the evaluated result of the formula in another formula
  - Use the formula as a parameter in a display option
  - Reference the formula by name in a Report Writer definition
  - Specify a column name for use within the *QUERY* interface

Named Formulas

- Optional
- Names start with FML_ (default FML_QueryName_n)
 - E.g. FML_MyFormula

Drop Tree

- New query view similar to *Drop Query,* but uses a tree view to display data

- Displayed directly on the current panel adjacent to the multi-line

- Useful when records can be grouped

Drop Query Test: INV_TV_QRY

1000010001

- 100000 - McMullen Trucking
  - 0001 - Transport Van - Blue
  - 0002 - Bug - Silver
  - 0003 - Camper Van - Orange
  - 0004 - Pickup Panel Truck - Fuschia
  - 0005 - City Car SUV - Blue
  - 0006 - Luxury Limo - Burgundy
  - 0007 - Sport Convertible - Fuschia
  - 0008 - Luxury SUV - Green
- 100001 - Nolan Building Supplies
  - 0001 - Roadster Sport - Fuschia
- 100002 - Rusk Office Supplies
- 100003 - Runnels Theatre
- 100004 - Grayson Seafood and Steaks
- 100005 - Lampasas Lighting

Close

NOMADS QUERY FEATURES

Drop Tree

- Defined in Query Header Options tab - Query View

- Supports limited run-time popup menu:
  - Find, Refresh, Print, Filters, Profile

- Can also be used to load a smart tree view
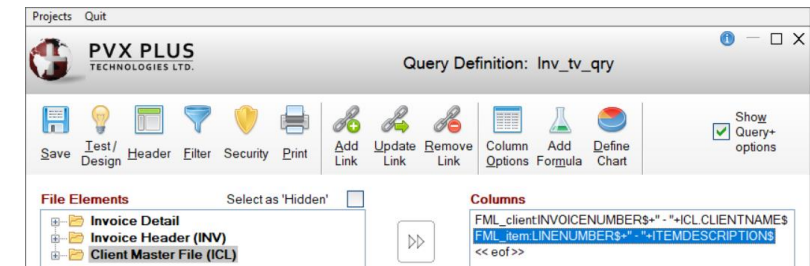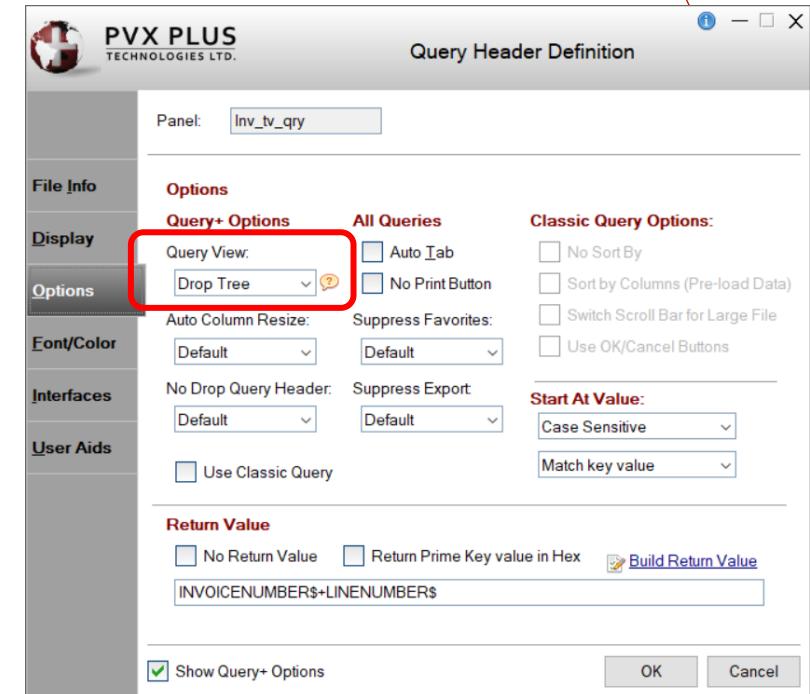
Access External Data

- External data can be passed to a query at run-time using global variables (not new)

- Can be used in filters and formulas

- New special %QRY_VARIOL$ and %QRY_VARDATA$ Query+ variables have been created to pass variable names and values that can be accessed by the query

Access External Data

- %QRY_VARIOL$ - A compiled IOLIST containing the names of the variables to hold the external data

- %QRY_VARDATA$ - The data for the variables in REC() format

LOCAL %QRY_VARIOL$=CPL("IOLIST ThisVal$,ThatVal")

LOCAL %QRY_VARDATA$=REC(IOL=%QRY_VARIOL$)

When the query runs, ThisVal$ and ThatVal will be populated and can be used in filters, formulas, etc.

Security

- Previously, security was based on Data Dictionary settings for files and fields, and this is still supported

- New security settings within the Query Definition based on the **Security Manager** with *Security Classes* and *Users*
  - Set Security Classifications to *Full Access* or *View Only*
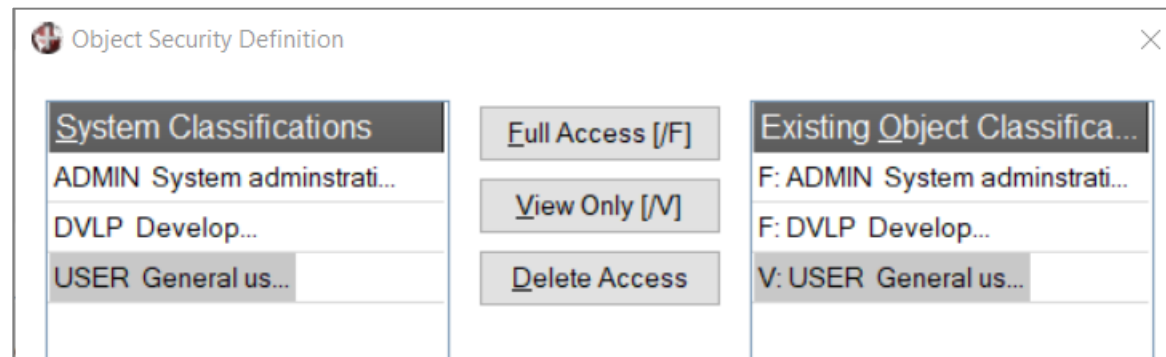
Security

- Three levels of security:
  - Top level (query/main file)
    - Determines if query can be displayed at all
  - Link file definitions
    - Determines if the fields in the linked files are displayed
  - Column definitions (fields/formulas)
    - Determines if specific fields are displayed
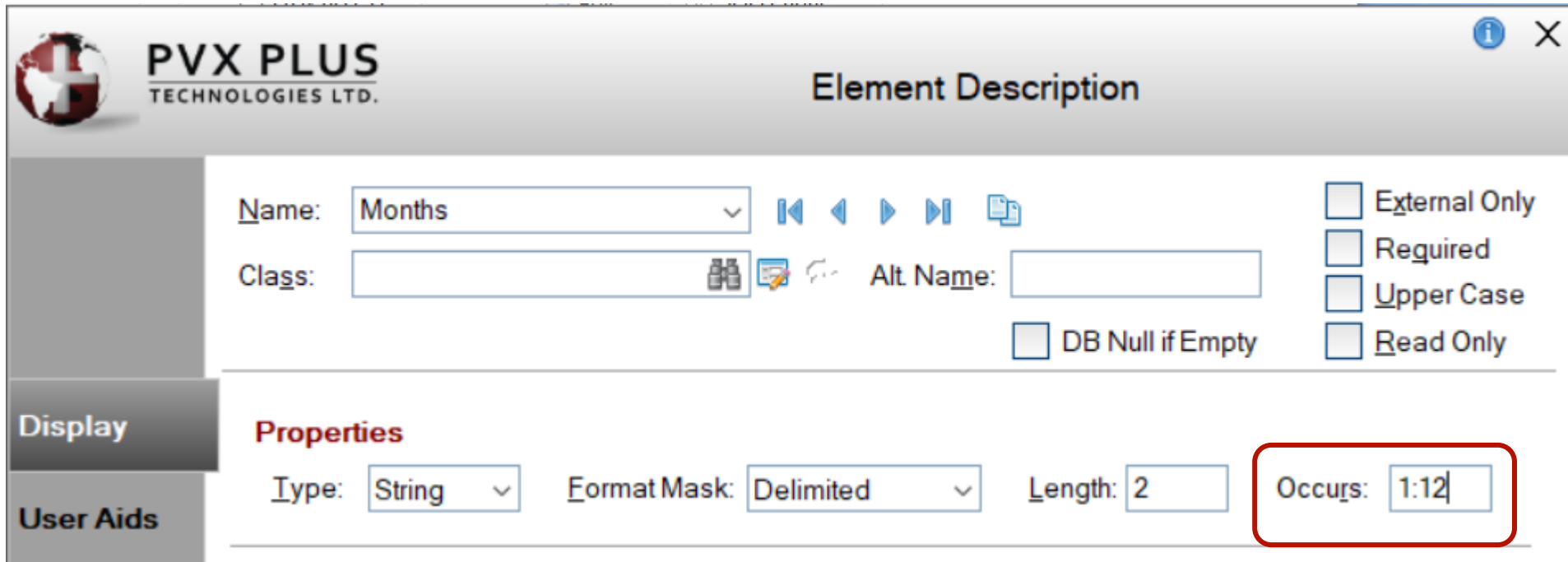
- Settings carry through to query-related features such as Smart Controls, Auto Charts, *QUERY*, etc.
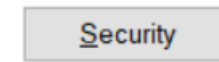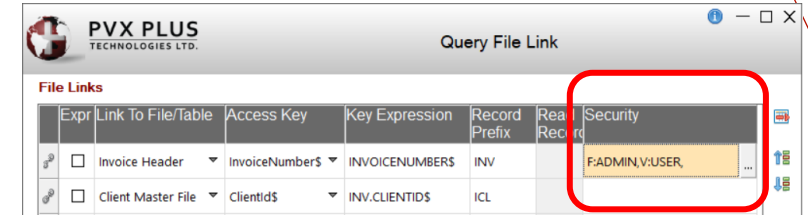
Array Support

- Support for Data Dictionary elements defined via "Occurs"

Array Support

- Displayed in the Query Definition File Elements list with dimension in braces, e.g. Month${1:12}, ThreeD{1:2,1:3,1:4}
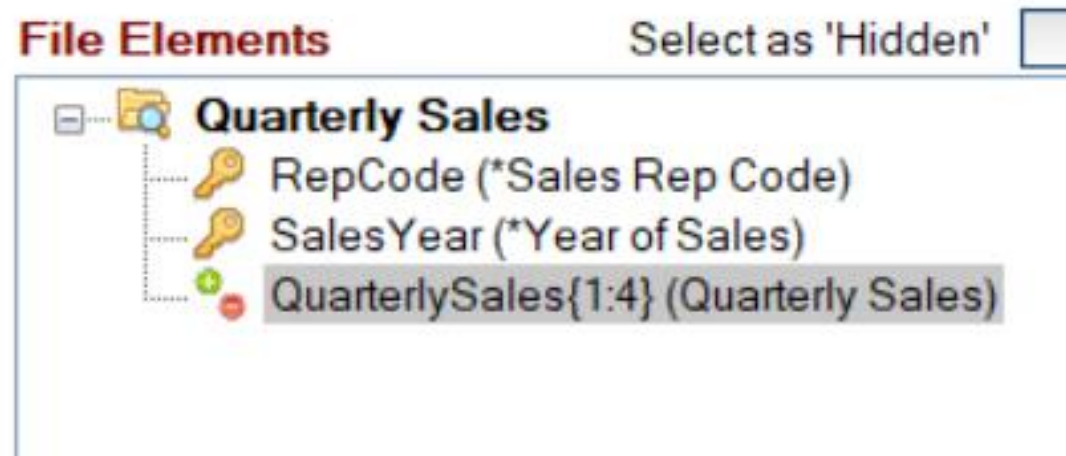
Array Support

- When you select an array entry from the *Elements* list you can specify *All Elements* or a *Single Element* by entering a subscript
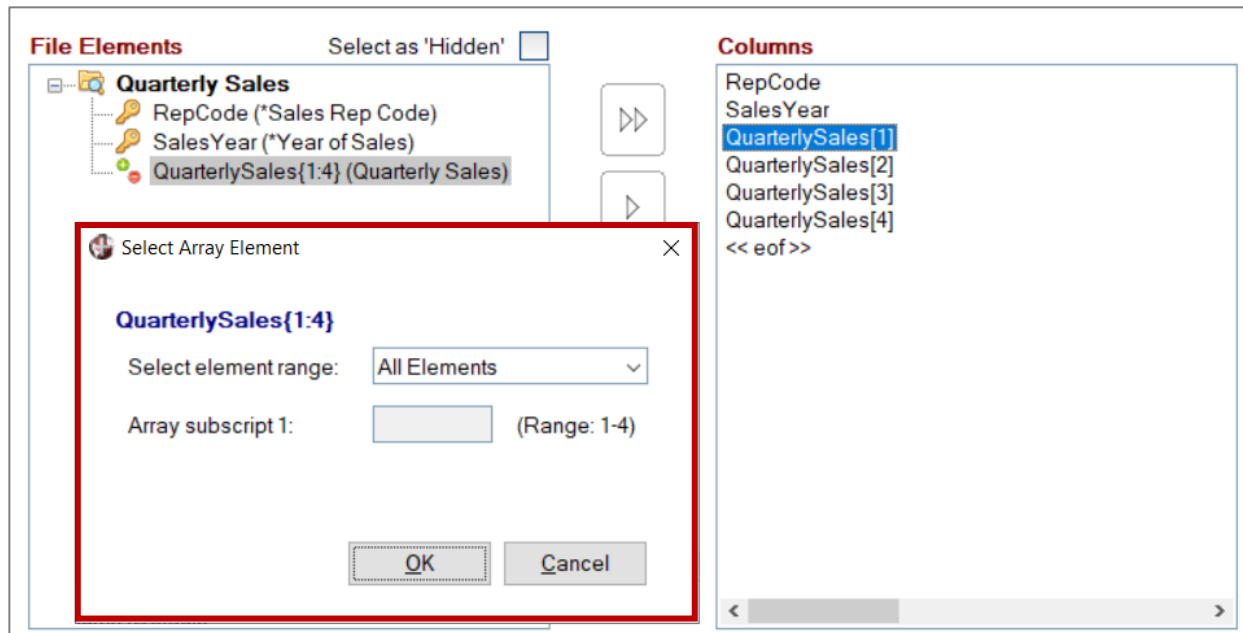
Array Support

- Array elements are identified using square brackets
  - E.g. Month$[1], ThreeD[2,1,1]

Formula Name	FML_Totals	
Column Title		
Fixed	Yearly Total	

Formula

QUARTERLYSALES[1]+QUARTERLYSALES[2]+QUARTERLYSALES[3]+QUARTERLYSALES[4]

Width	Alignment	Format
15	Default	⊙ Mask ◯ Program
		Fixed $###,##0.00

Composite Key Definitions for File Links

- Previously you had to know the segment details to concatenate the key string

    E.g. PAD(Company$,5,$00$)+PAD(Department$,10,00)+EmployeeID$

Composite Key Definitions for File Links

- Now you can generate a KEY=F1$:F2$ expression just knowing the fields
  - Check the *Generate a KEY=F1$:F2$ expression* option
  - Select the fields that make up the key and the *Key Expression Builder* utility builds the expression for you

Composite Key Definitions for File Links

- Generates a Key Expression using the F1$:F2$:f3$ syntax
    - **E.g.** KEY(__fileFN,KNO=0,KEY=Company$:Department$:EmployeeID$)

Composite Key Definitions for File Links

- Can handle changes in key segment length if file keys are updated
- Feature also added to other features such as *File Link Definitions*, Views *Data Source Maintenance*, etc.

Save Changes when Testing

- Often it is difficult to determine the widths of the columns or the size of the panels when creating a query definition
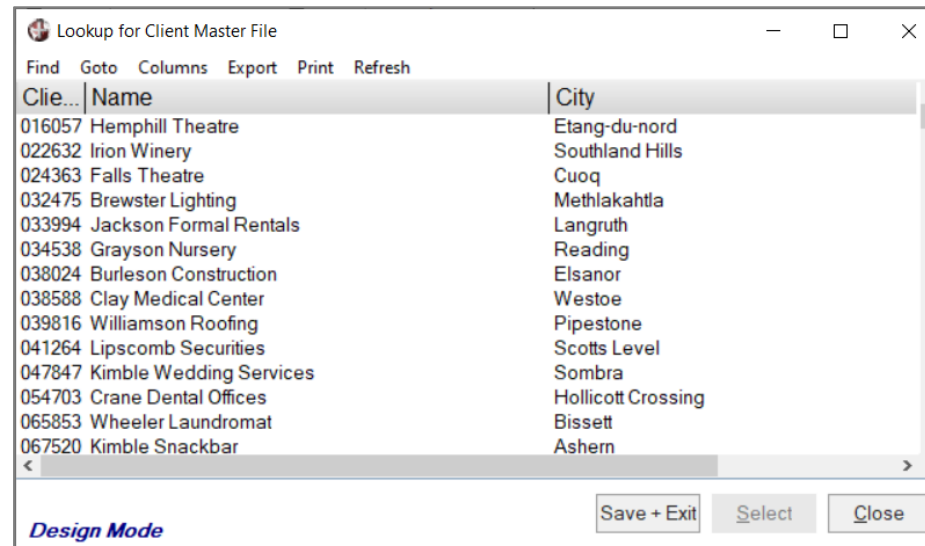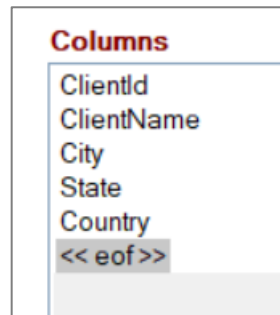
Save Changes when Testing

- When testing Query+ and Drop Query definitions certain changes can be saved when exiting the test panel
  - Column widths, *Show/Hide/Reorder Columns* may be altered
  - Query+ panel size can be adjusted
  - Save + Exit button added

Auto-Select Key Field Columns

- When creating a new query, the Columns list box is now automatically populated with:
  - Elements that make up the chosen key
  - First non-key field >= 15 characters

  (Not applicable to queries based on Views or manually defined files)

- Then you can add/remove the fields you want
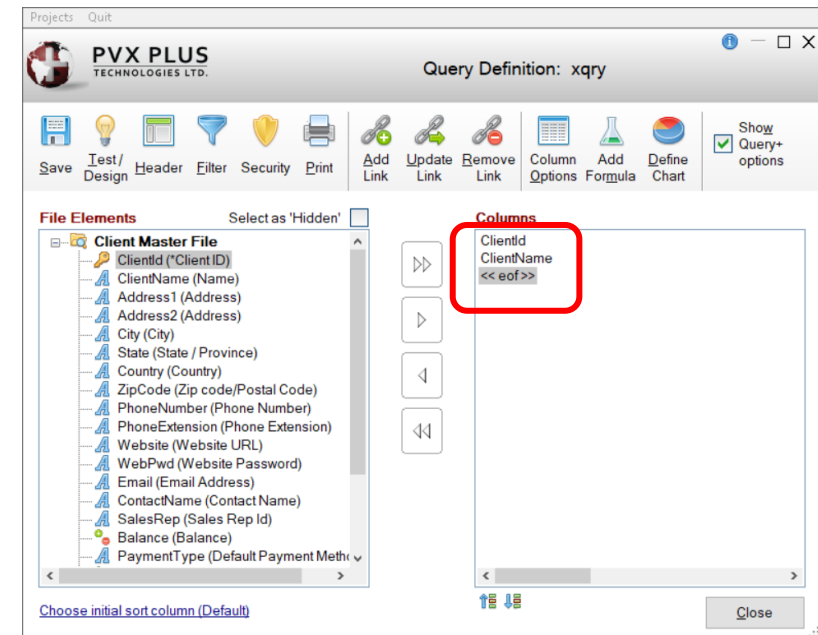
Query_Row$ Added

- Added special Query_Row$ variable containing the contents of the query row selected

- Can be used in *Selection Logic* or *Return Value* expressions

Selection Logic

| Test | ARG_1$="" OR POS(UCS(ARG_1$)=UCS(QUERY_ROW$))>0 |

Query_Row$ Added

- Allows testing the entire row to match a value

QUERY+ OPTIONS

Suppress Records with Duplicate Return Values

- Show/Process only the first record of a series
  - E.g. Displaying the invoices with VAN products where there could be several VAN type detail lines

Product Code	Item Description	Invoice Number	Name	Sales Rep Name
VAN-2120-BLU	Transport Van - Blue	100000	McMullen Trucking	Chuck Mitchell
VAN-1124-ORG	Camper Van - Orange	100000	McMullen Trucking	Chuck Mitchell
VAN-1125-VIO	Camper Van - Violet	100002	Rusk Office Supplies	Anastasia Nelson
VAN-4111-ROS	Panel Van - Pale Rose	100002	Rusk Office Supplies	Anastasia Nelson
VAN-2121-BUR	Transport Van - Burgundy	100002	Rusk Office Supplies	Anastasia Nelson
VAN-2124-SEA	Transport Van - Seagreen	100003	Runnels Theatre	Betty Turner
VAN-4124-ORG	Passenger Van - Orange	100007	Grimes Landscaping	Donna Kerr
VAN-4110-SLV	Panel Van - Silver	100008	Cameron Electrical Contracting	Anastasia Nelson
VAN-4122-PNK	Passenger Van - Fuschia	100009	Stephens Wine and Spirits	Anastasia Nelson

Suppress Records with Duplicate Return Values

- Query Selection Definition - *Suppress Duplicates*
  - *Suppress records based on duplicate return values* option

Suppress Duplicates
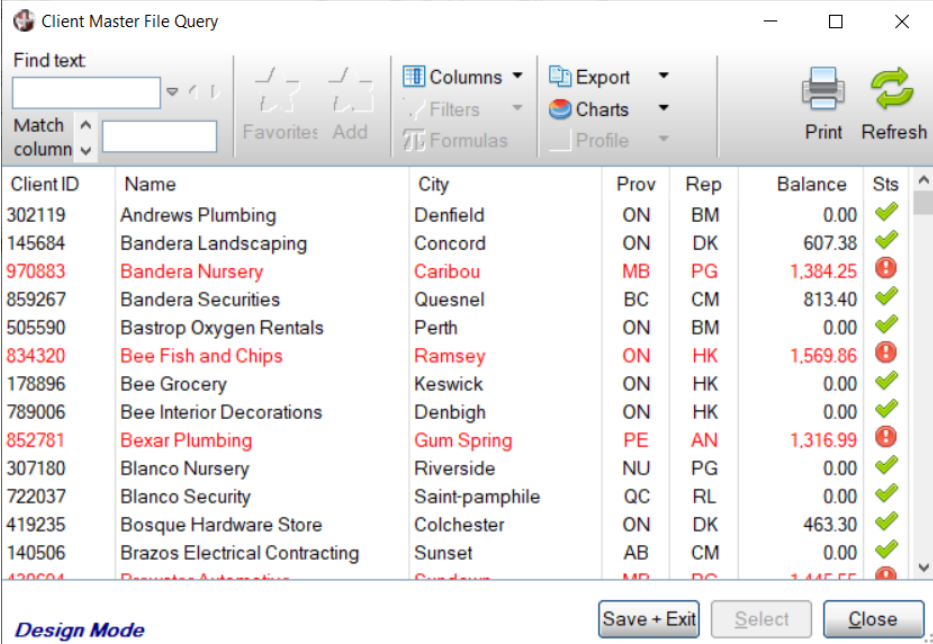
☑ Suppress records based on duplicate return values

  - Show/Process only the first record of a series

Product Code	Item Description	Invoice Number	Name	Sales Rep Name
VAN-2120-BLU	Transport Van - Blue	100000	McMullen Trucking	Chuck Mitchell
VAN-1125-VIO	Camper Van - Violet	100002	Rusk Office Supplies	Anastasia Nelson
VAN-2124-SEA	Transport Van - Seagreen	100003	Runnels Theatre	Betty Turner
VAN-4124-ORG	Passenger Van - Orange	100007	Grimes Landscaping	Donna Kerr
VAN-4110-SLV	Panel Van - Silver	100008	Cameron Electrical Contracting	Anastasia Nelson
VAN-4122-PNK	Passenger Van - Fuschia	100009	Stephens Wine and Spirits	Anastasia Nelson
VAN-4124-ORG	Passenger Van - Orange	100011	Cass Theatre	Hank Kane
VAN-1124-ORG	Camper Van - Orange	100014	Leon Bistro and Bar	Olivia Blackwood
VAN-4114-PRL	Panel Van - Pearl	100015	Dawson Farm Equipment	Gordon Taylor

Row Display Options

- Option to conditionally change color/highlighting on individual rows to highlight a condition:
  - *Bold Text* – Display text in bold font
  - *Colors* – set text & background colors
  - *HiLiteColor* – Set background color
  - *TextColor* – Set text color
  - *User display* – Specify your own display string

Row Display Options

- **Query Header *Display Tab***
  - *Row Highlight – Add row display options*

  Row Highlight

  📝 Edit row display (TextColor)

  - ***Row Display Option* interface**
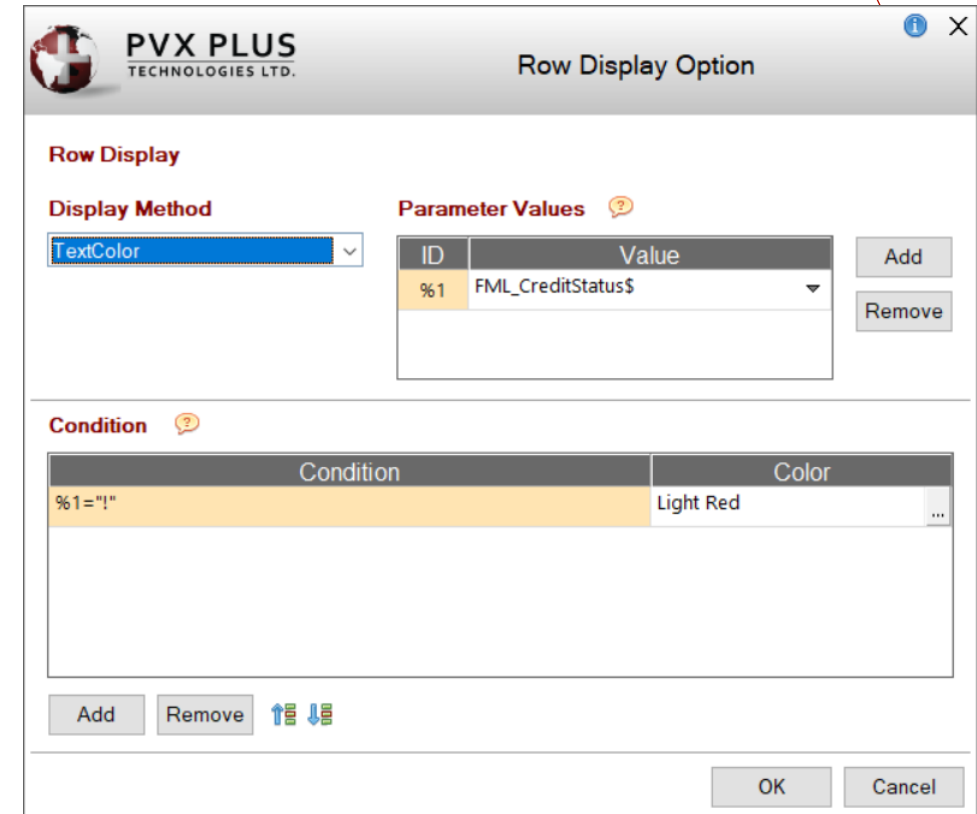
Change Font/Color

- Enhanced the Query+ Font/Color definition

  - Add run-time support for Library Default and Panel settings for font and color
  - Allow Font Specification and Attributes for individual queries
  - Allow setting Background color for individual queries

Choose Initial Sort Column

- Allows ascending/descending sort on non-key columns
  - Can even sort on Formula values
  - Occurs after list is loaded when query first displayed
    - Doesn't use Load-on-demand or Background Loading
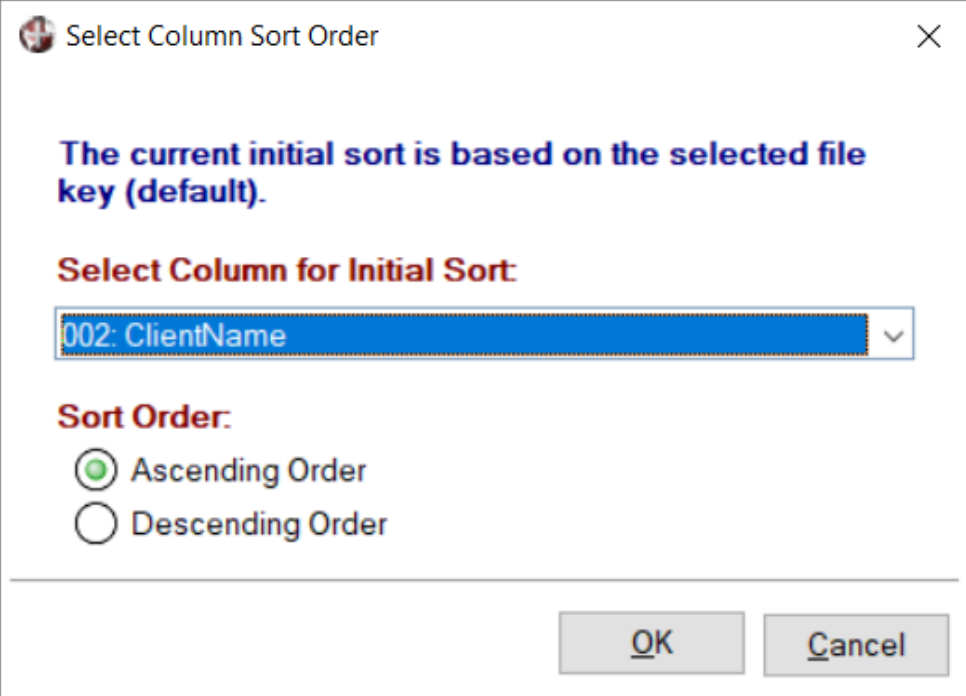
Choose Initial Sort Column

- *Choose initial sort column* at bottom of Query Definition screen

  Choose initial sort column (Default)

  - Invokes *Select Column Sort Order*

Start Value Option

- Sometimes the value in the query multi-line does not match the sort key used in the query, so the lookup fails
  - E.g. the multi-line has a client code but the query is sorted by client name

No match

Start Value Option

- New *Start At Value* option in Header *Options*
  - *Start At Value* options:
    - *Match Options*
      - *Match Key Value*
      - *Match Return Value*

Continue While

- Selection criteria option for when you want to stop a query before the logical end
  - E.g.
    - First 10 records
    - When a total is reached
    - Etc.

Continue While

- Query Selection Definition – Continue While…
  - Processing continues while the processing condition is true
  - Based on the initial order of the records read
  - Can be a *Test* expression or *Call* a program that returns an EXIT code to stop
  - Added a special query variable QRY_REC_COUNT for instances where the limit is the number of records
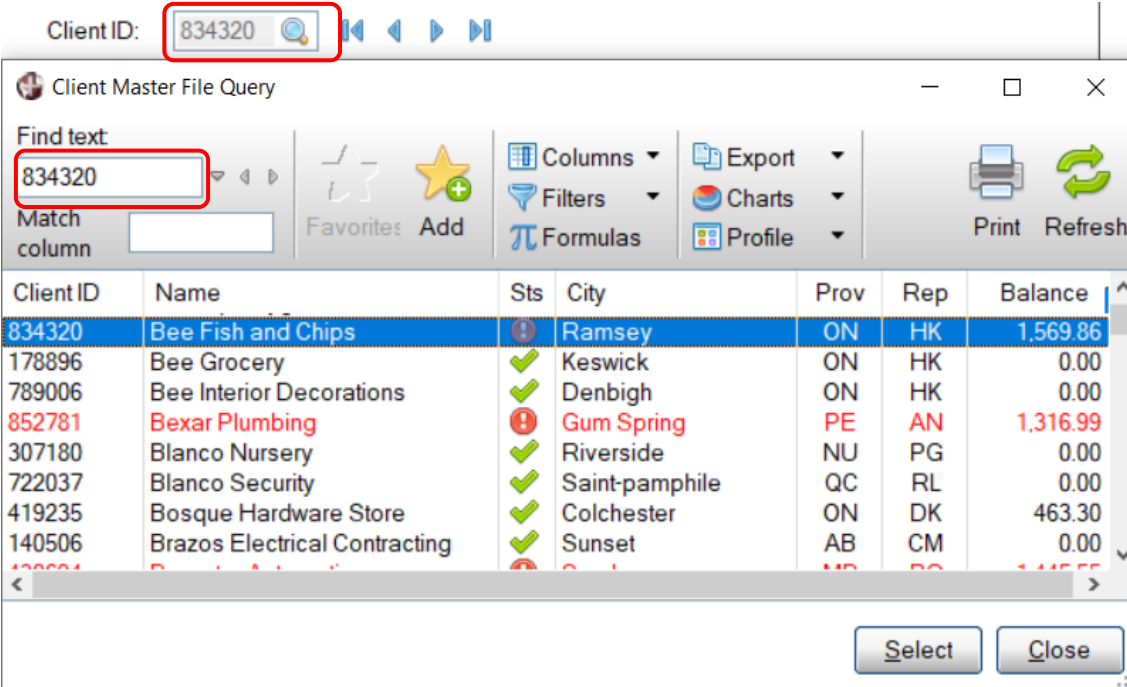
Find Text

- Display the query multi-line value passed to the query in the *Find Text* box of the run-time query

QUERY-BASED UTILITIES

QUERY

- Access the contents of a query like a standard read-only file
  - New KNO=* option to automatically use the key specified in the query definition
  - Enhanced to better handle duplicate, descending and composite keys
    - SEQPFX=Y – Prefixes a 10-digit sequence number to the key value to ensure unique values and correct record sequence
    - KEYPAD=Y – Replaces 00 padding in compound keys with spaces

*tools/qryprint

- Prints a formatted PDF report based on a query definition

QUERY-BASED UTILITIES
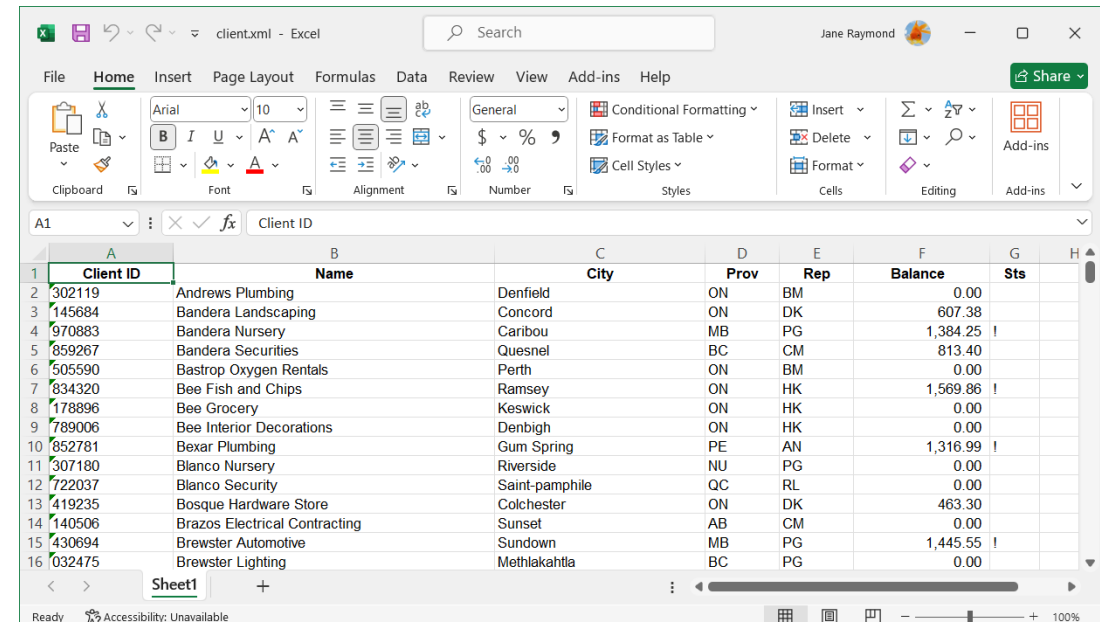
*tools/qryprint

CALL "*tools/qryprint", *queryName$, queryLib$, pdffile$ [, queryopts$, show_flg$, subheader$]*

- Includes content options (queryopts$) to show hidden columns in the query definition (H) and display numeric calculation options (Total/Average/Minimum/Maximum) in subheader (C)

- Includes interactive options (show_flg$) to show error messages (E), display a progress bar (P) and display the PDF file when done (D)

- Can include text for a subheader on the first page of the report

*tools/qryexport

- Exports data to a spreadsheet-supported file based on a query definition
  - Supported file types:
    - .csv – Comma separated
    - .txt – Tab delimited
    - .slk – Symbolic link
    - .xml – Microsoft XML

*tools/qryexport

CALL "*tools/qryexport", *queryName$, queryLib$, exportfile$ [, queryopts$, show_flg$]*

- Type of export file determined by file extension
- Includes content options (queryopts$) to show hidden columns in the query definition (H)

- Includes interactive options (show_flg$) to show error messages (E), display a progress bar (P) and display the file in the default spreadsheet app when done (D)

REPORT WRITER ENHANCEMENTS 2020 – 2024

DireXions 2024

AGENDA

1. **New Input Source**

2. **Parameter Specifications**

3. **Report Designer Enhancements**

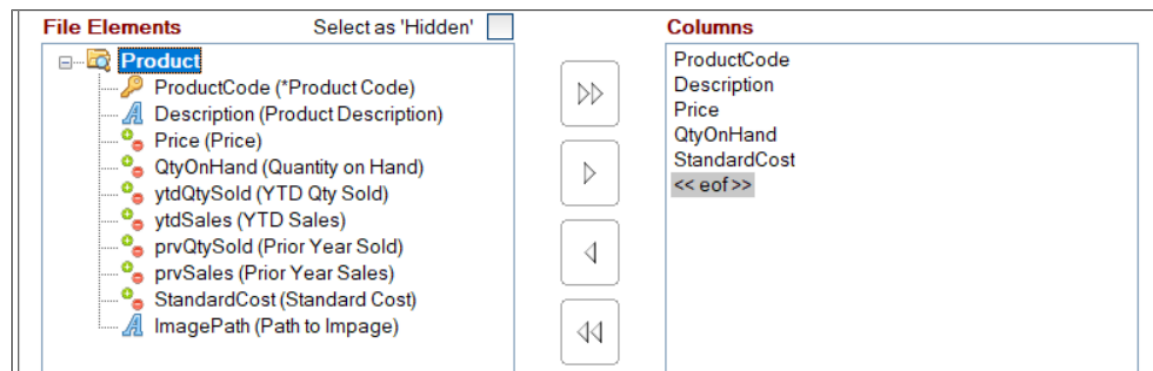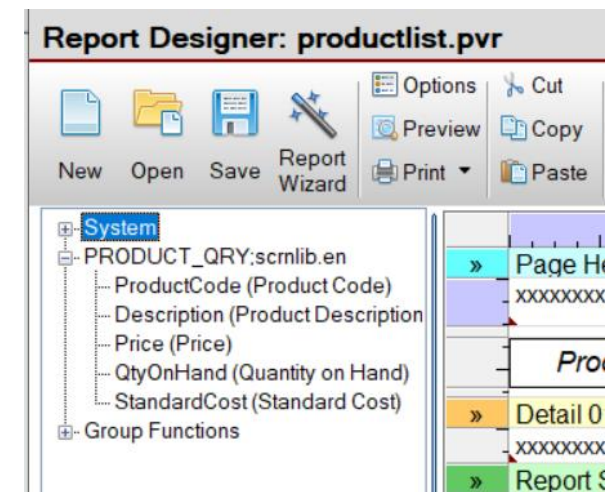4. **New Report Options**

5. **Dynamic Run-Time Filters**

NEW INPUT SOURCE

New Input Source

- Query Definition as an Input Source
  - Supports Query Definitions based on PxPlus files with embedded dictionaries, View definitions or external database
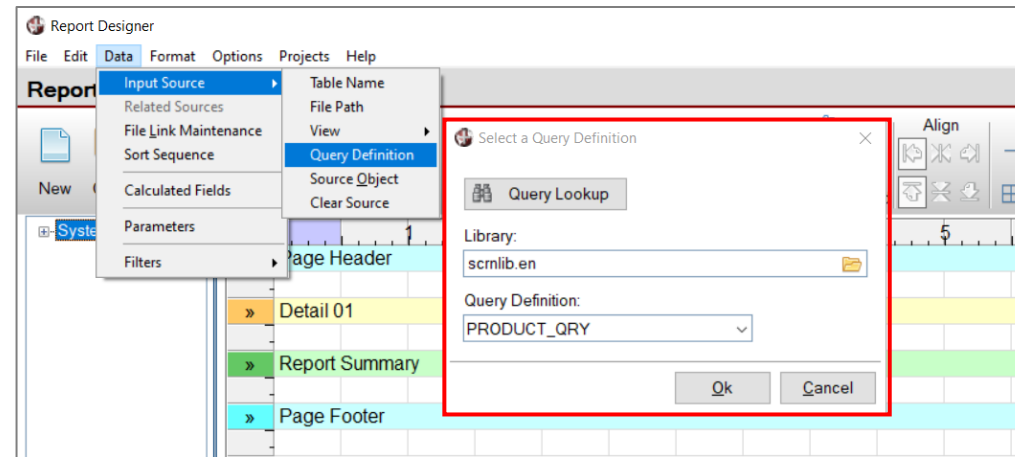 - Adds another way to access a dataset comprised of multiple related files

New Input Source

- <u>To assign a query as an input source</u>
  - Data > Input source > Query Definition

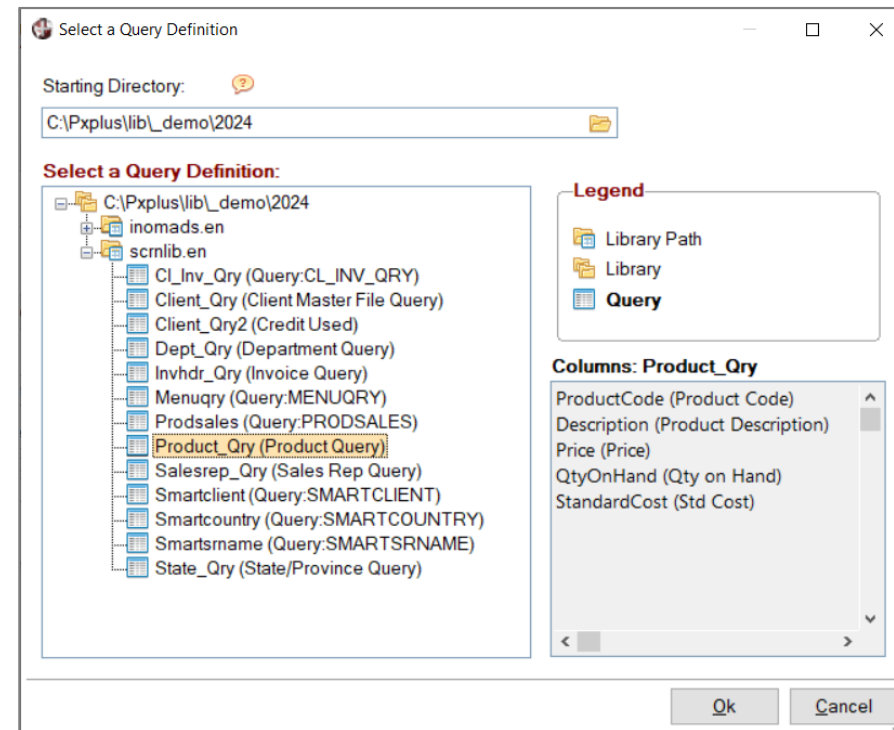

Select the query by identifying the library and query name or by clicking the *Query Lookup* button.

New Input Source

- *Query Lookup* – Select a Query Definition

  - Lists queries that exist in the current directory level and lower

New Input Source

When selected, the query columns are listed in the Report Designer data pane for inclusion in the report layout

PARAMETER SPECIFICATIONS

PARAMETER SPECIFICATIONS

Parameters

- Values that are supplied by the user or application program when a report is generated.

  E.g.

  - Generate data for a particular department to be specified by the user requesting the report.

  - Display transactions between a certain start and end date to be specified at run time.

- Can be displayed in the report, used in formulas, or used to set up selection criteria.

Parameter Expression Support

- Enhanced to allow expressions for Maximum Length, Minimum and Maximum Values, Default Value and Format
  - Expression values preceded by '=', e.g. =DTE(0:"%Y%Mz%Dz")
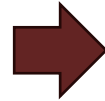
Parameters Using Dynamic Data Classes

• Data class definition have descriptions, input lengths, default values, formats and validation rules assigned to them, as well as drop box definitions, checkbox input and queries

• By assigning a *Data Class* to a parameter definition, these settings can be used to format input in the default parameter interface at run-time
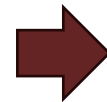
PARAMETER SPECIFICATIONS

Parameters Using Dynamic Data Classes

Class selection ➡️

Class info ➡️

PARAMETER SPECIFICATIONS

Parameters Using Dynamic Data Classes

Parameters – Ignore Class Validation

- Some Data Classes have validation rules assigned
  - Validation rules require an exact match
  - Sometimes you don't want an exact match, such as when setting up ranges

- *Ignore class validation* option
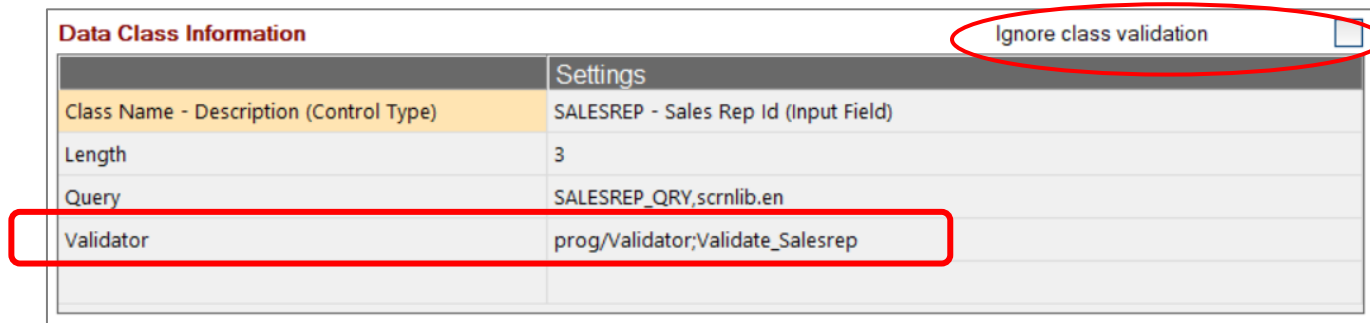 - Allows you to use the class for input formatting but not for validation

Data Class Information		Ignore class validation
	Settings	
Class Name - Description (Control Type)	SALESREP - Sales Rep Id (Input Field)	
Length	3	
Query	SALESREP_QRY,scrnlib.en	
Validator	prog/Validator;Validate_Salesrep	

REPORT DESIGNER ENHANCEMENTS

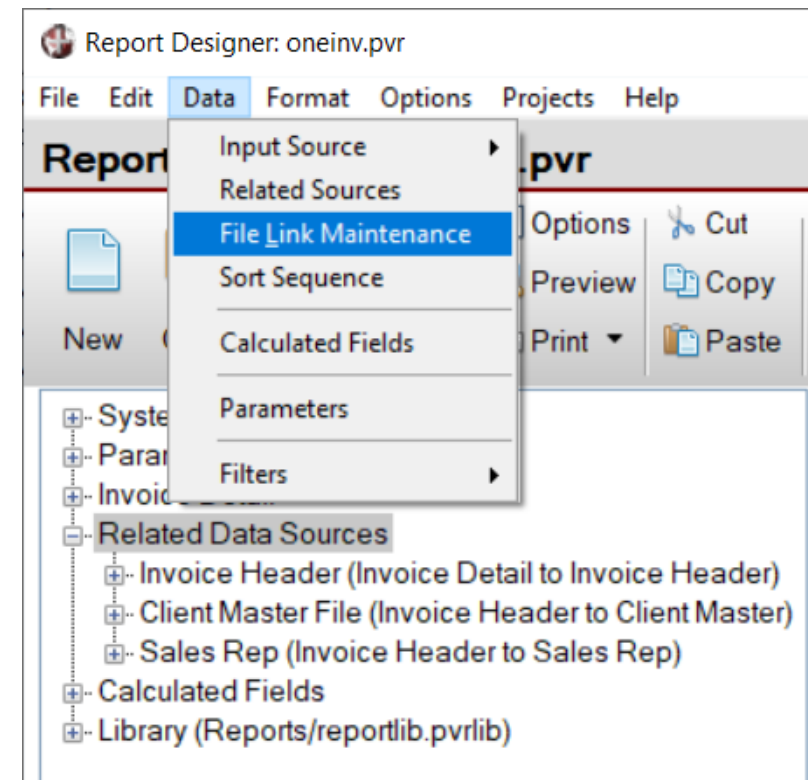- <u>Text Alignment Made Easier</u>

Before

Now

- More choices
- Shows alignment of current cell

Links to File Link Maintenance

- File Link Maintenance utility lets you define and maintain cross-reference linkages that exist between data files

- Link to *File Link Maintenance* added to *Data* menu
- Don't have to exit the Report Designer to set up file links

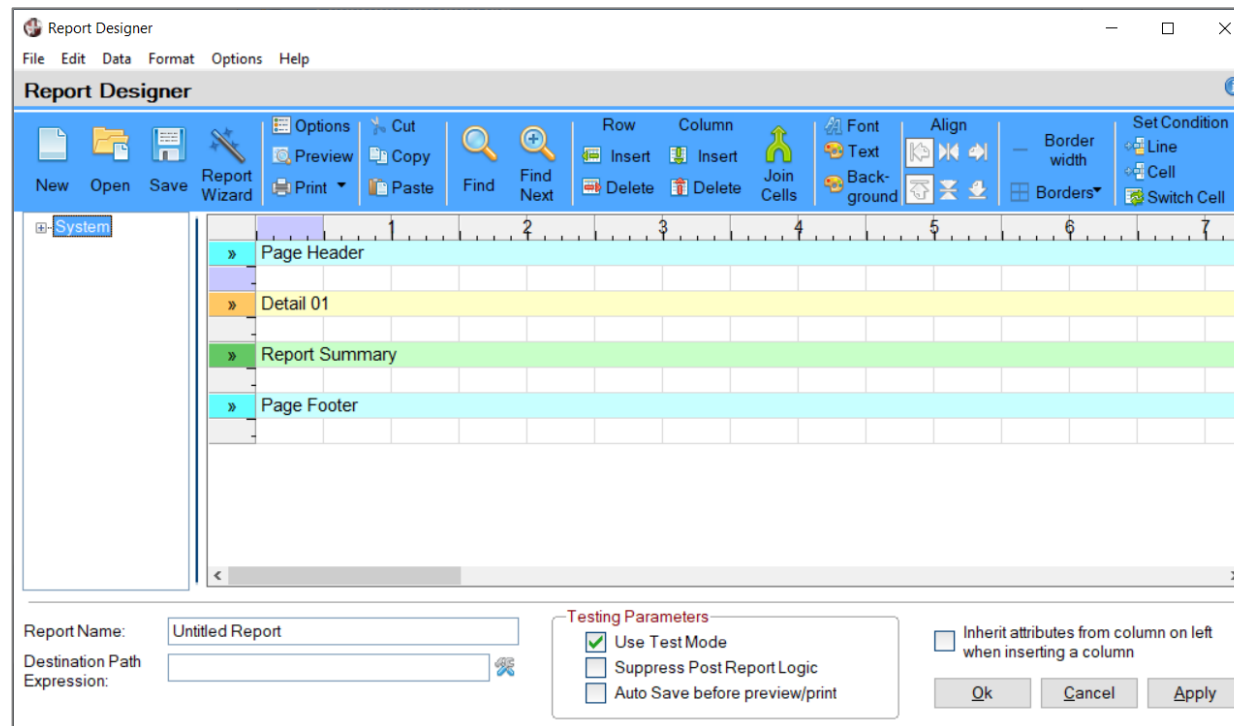## Inherit Cell Attributes from Left when Inserting Column

- Option to inherit attributes (font, alignment, word-wrap, colors and borders) from the current cell when inserting a column (like Excel)
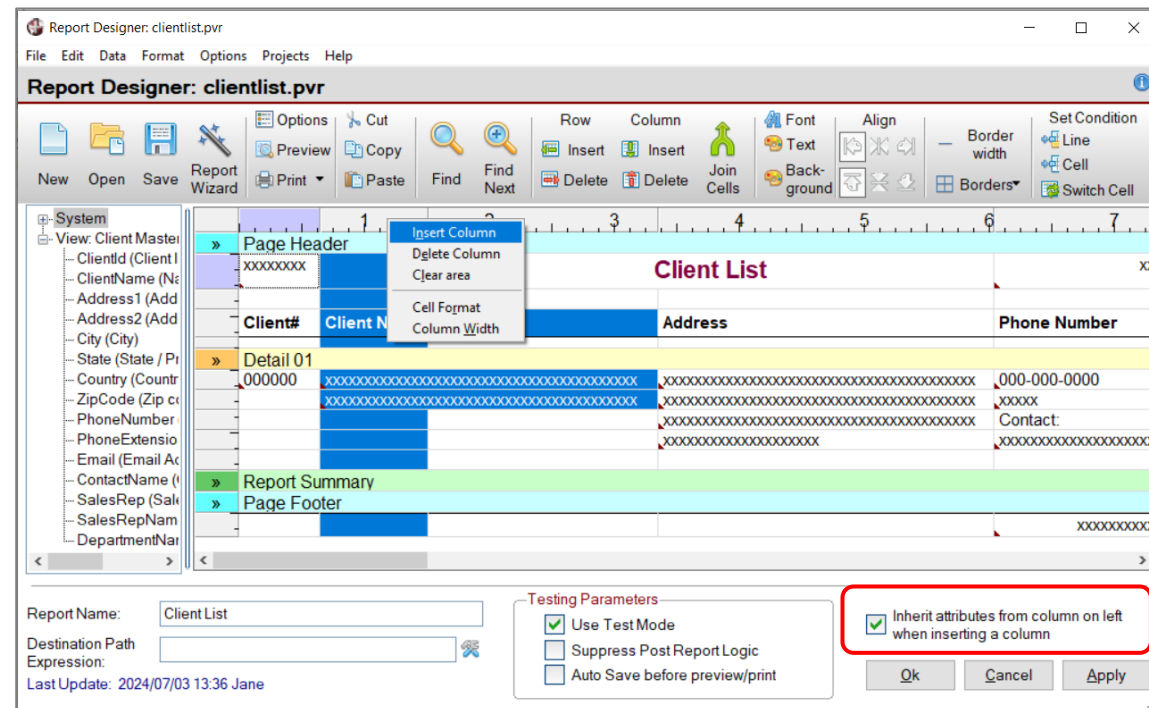
Theme Colors

- The Report Designer can be a user-facing feature, so will use the application theme colors, if set.

NEW REPORT OPTIONS

Line Advancement

- Control how line advancement occurs

  Normal Line Advancement

  Controlled Line Advancement

REPORT WRITER OPTIONS

Line Advancement Options

- Normal – Regular line advancement
- Overlay Next Line – Write line, do *not* advance vertical positioning, write next line
- Overlay Previous Line – Write line, advance vertical positioning; subtract vertical positioning and write next line
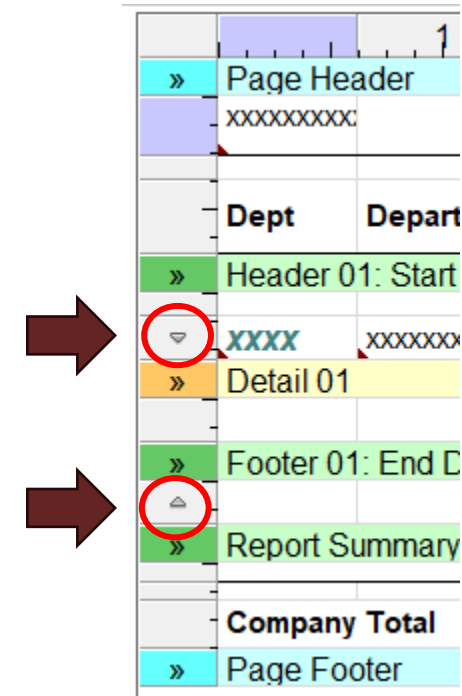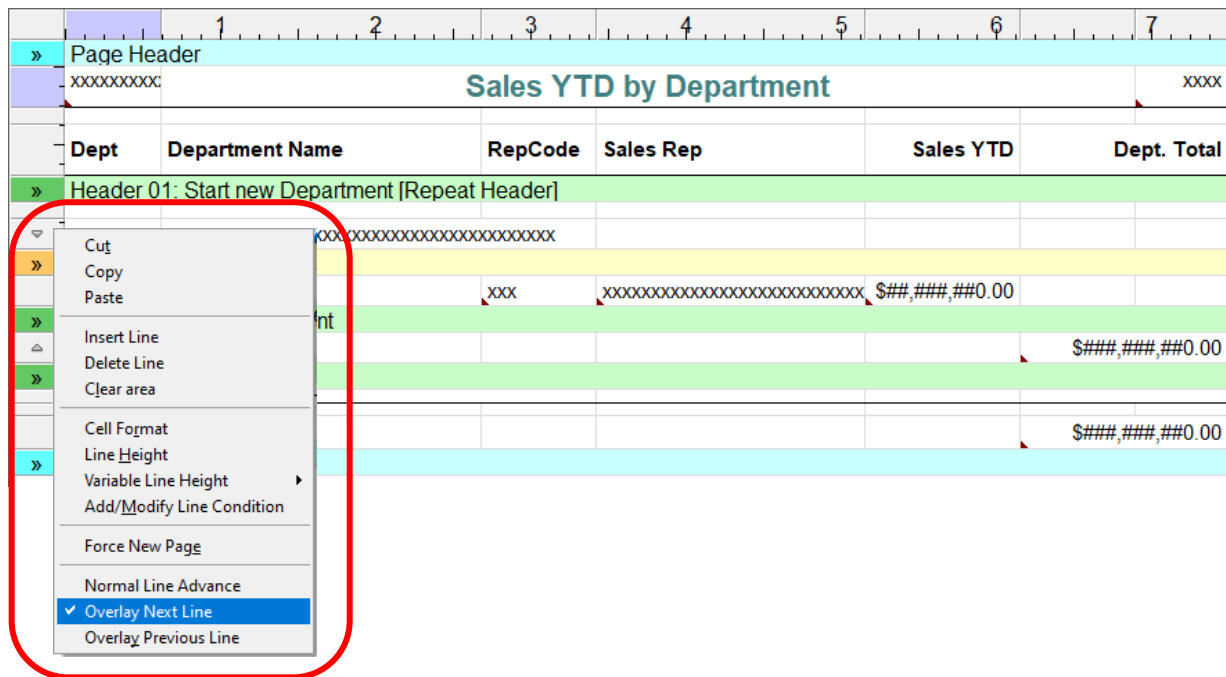
Setting Line Advancement Options

- Set from popup menu by right-clicking on row header column

- Ticks indicate Next/Previous selections

Force New Page

- Insert form feeds at specific points in the report layout
    - Could always specify new pages for report and group headers and footers
    - Now you can force a new page within a Detail line group
- Set from row header column popup
    - Marked by line indicating new page
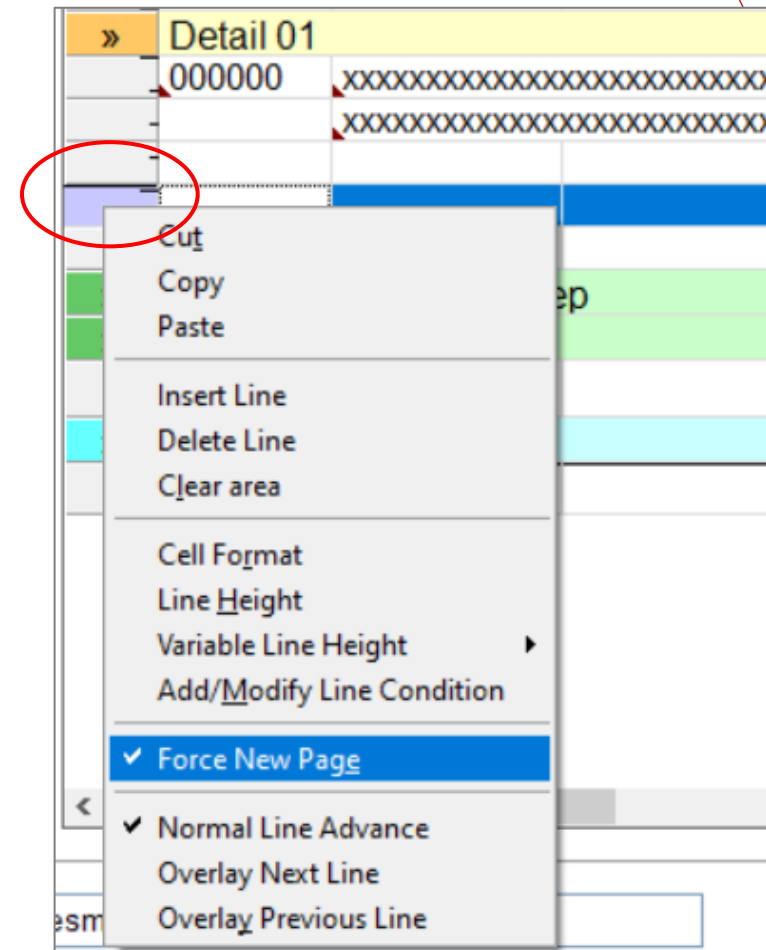
Post-Report Logic

- Specify logic to execute when a report is complete
  - E.g. mail PDF files to a client's email account

- Options:
  - Call – Program to call and optional arguments
  - Execute – PxPlus statement to execute
  - Method – Method in user-defined Logic Object Interface + arguments
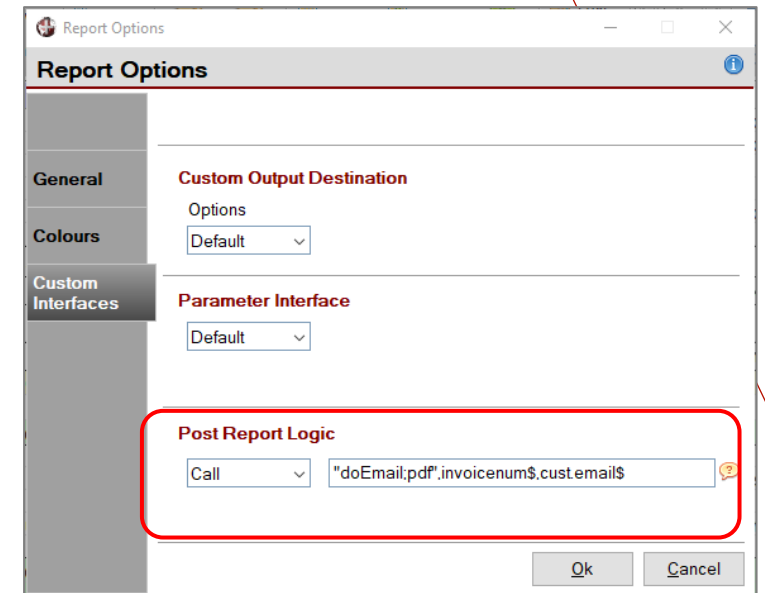
Suppress Post Report Logic

- Testing Parameter on the Report Designer
 - Suppresses the execution of *Post-Report Logic* while testing

Multiple Report Generation

- E.g. Printing invoices, customer statements, etc.

- Previously…

  - Generate multiple reports to a printer

  - Generate multiple concatenated reports to a single file

  - Write program code to execute single reports multiple times

    - Only this last option lends itself to doing *Post-Report Logic*
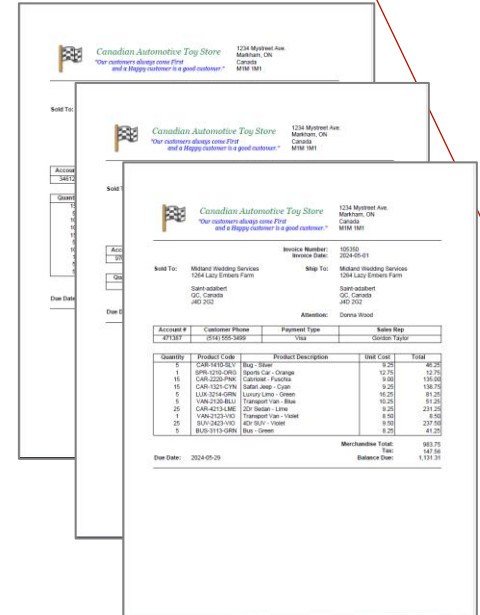
Multiple Report Generation

- Now…
  - One call generates multiple reports to separate physical files
    - Post-Report logic executed for each report
  - You need only:
    - Supply a *Destination Path Expression* to evaluate for a unique name for each file, e.g. _DateYYYYMMDD$+ClientID$[+".pdf"]

Destination Path Expression: InvoiceNumber$

    - Set a *Group Header* option to generate multiple individual reports

Multiple Report Generation

- Dependent on report grouping
  - E.g. When the Invoice number or Client code changes
  - [Group Header Option](#) - When group changes:

    - *Continue with in-line placement*
    - *Start new page* – Starts a new page for each new group
    - *Start new report* – Creates a separate individual report for the group, including report summary, header and footer sections

DYNAMIC RUN-TIME FILTERS

DYNAMIC RUN-TIME FILTERS

Static Filters

- Previously…
  - Set specific filters and parameters within the report definition
  - Filter values entered as parameters at run-time
  - E.g. Client list for a specific salesman
  - One-use report
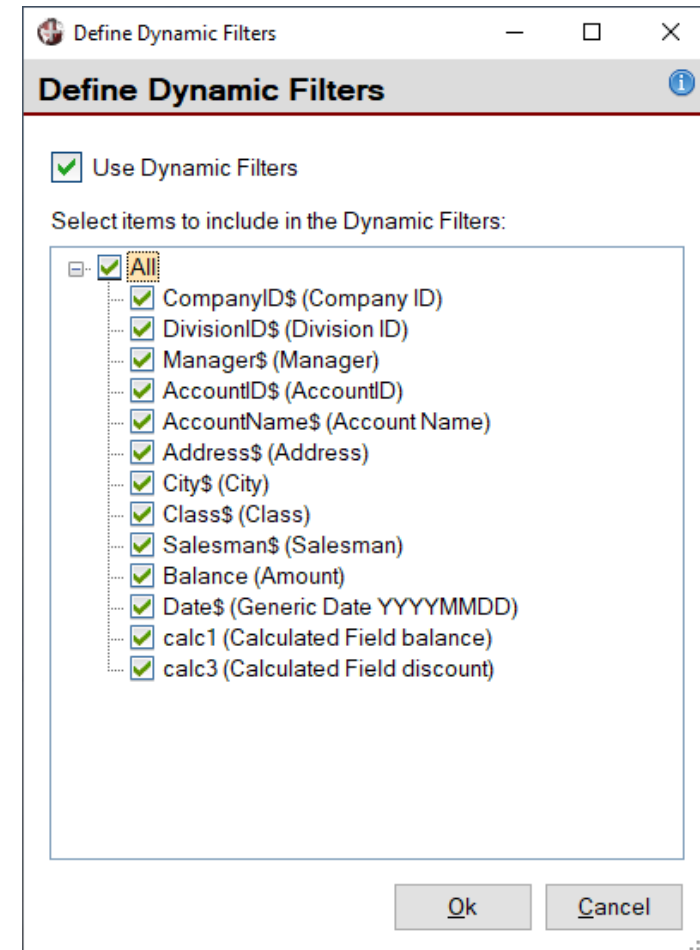
Dynamic Filters Defined by User at Run-Time

- Now…
  - The user can define their own filters at run-time
  - Results in *flexible multi-use* reports
    - E.g. Client list for a salesman, contact person, region, etc.
    - Can include/exclude records based on multiple conditions
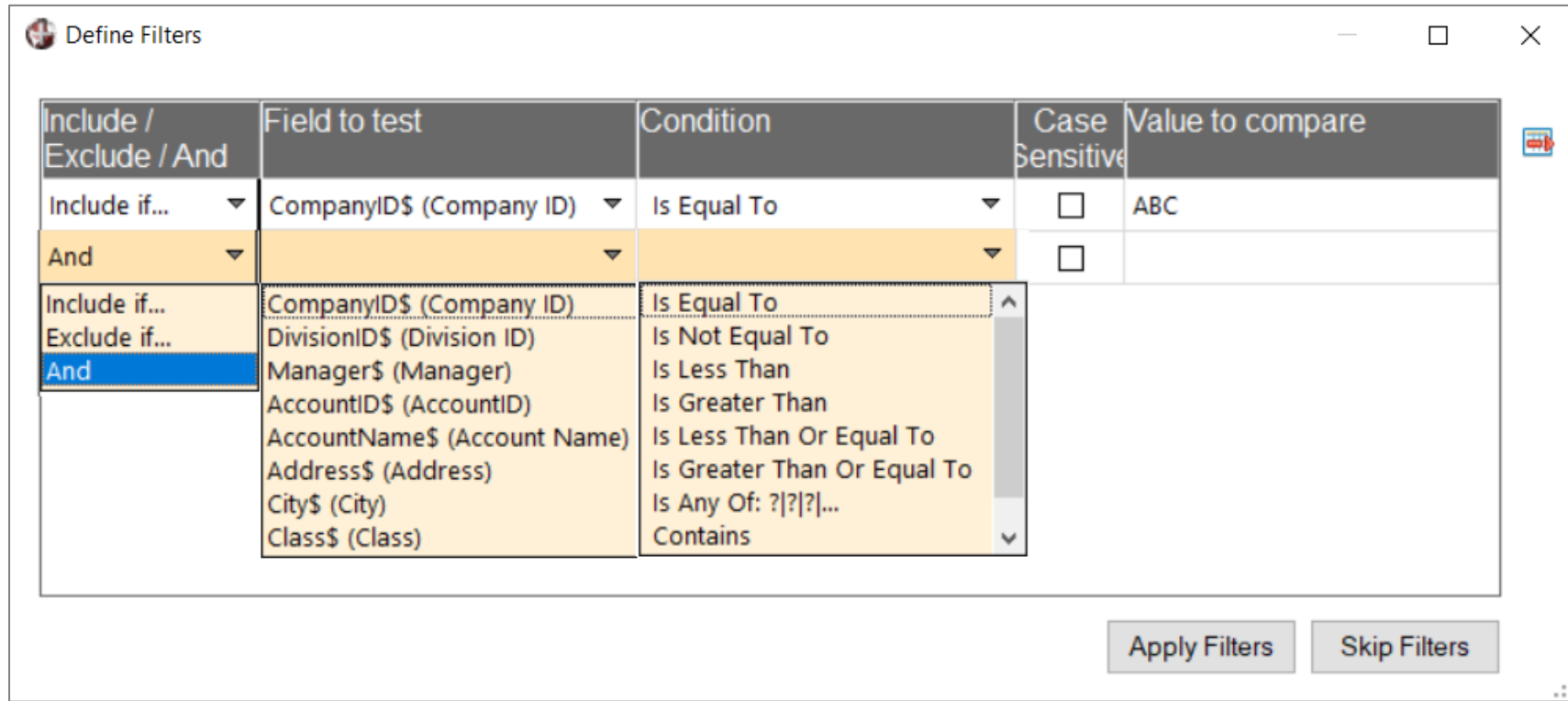
Setting up Dynamic Filters for a Report

- *Data > Filters > Dynamic Filters*
  - *Define Dynamic Filters*
    - *Use Dynamic Filters* option
    - Select the items to include

DYNAMIC RUN-TIME FILTERS

Filters Defined by User at Run-Time

- At run-time…

DYNAMIC RUN-TIME FILTERS

Filters Defined by User at Run-Time

System *Dynamic Filter* variable available to include in your report

OBJECT-ORIENTED PROGRAMMING

All of the features and options described in this presentation are supported in the Report Writer Object-Oriented interface.

```
! Change the output destination to HTML file
!
    rpt=new("*rpt/pvxreport");
    if rpt=0 \
        then end
!
    serial "myrpt.htm ",err=*next
    open purge (hfn,err=WrapUp)"myrpt.htm";
    prt=lfo
!
    if rpt'open("MyRptDef.pvr")=0 \
        then goto WrapUp ! open and init the report definition
    rpt'OutputHTML(prt) ! set the output channel
    if rpt'ParamCount()>0 \
        then rpt'AcceptParameters() ! get parameter values from user
    rpt'RunReport() ! generate the report
    rpt'close() ! close the report object
    close (prt)
!
    system_help "myrpt.htm"
!
WrapUp: \
    drop object rpt
    end
```