Michael Zuskin
https://www.linkedin.com/in/zuskin/

DataWindow Spy

It is a utility that enhances the productivity of PowerBuilder programmers providing them with a magic ability to quickly and easily view hidden DataWindow information at runtime.

Any displayed info can be copied to the Windows clipboard. For instance, you can copy the DataWindow's SQL SELECT statement and execute it in a DB tool to debug a retrieval issue.

This utility functions exclusively when the application is launched from within PowerBuilder IDE. It is automatically ignored in the executable.

The Spy does not require PFC or any other framework. It is easily integrated into any PB project (version 7 or later) by inserting a few lines of code.

This is not a commercial product. I initially developed it for personal use long time ago, and it has accompanied me through numerous workplaces for decades. If you find this small utility helpful and wish to use it, please feel free to do so; it is not copyrighted.

STEPS TO ADD THE SPY TO YOUR APPLICATION

1. Go to https://github.com/Ursego/DWSpy/blob/main/spy.pbl

2. Press Ctrl+Shift+S (or the ... icon in the upper-right corner > Download).

3. Save the spy.pbl file on your hard disk among other PBLs of your app.

4. Add the spy.pbl file to the end of your app's library list.

5. Add the following code fragment to the Clicked event of your base DW (like u_dw in PFC):

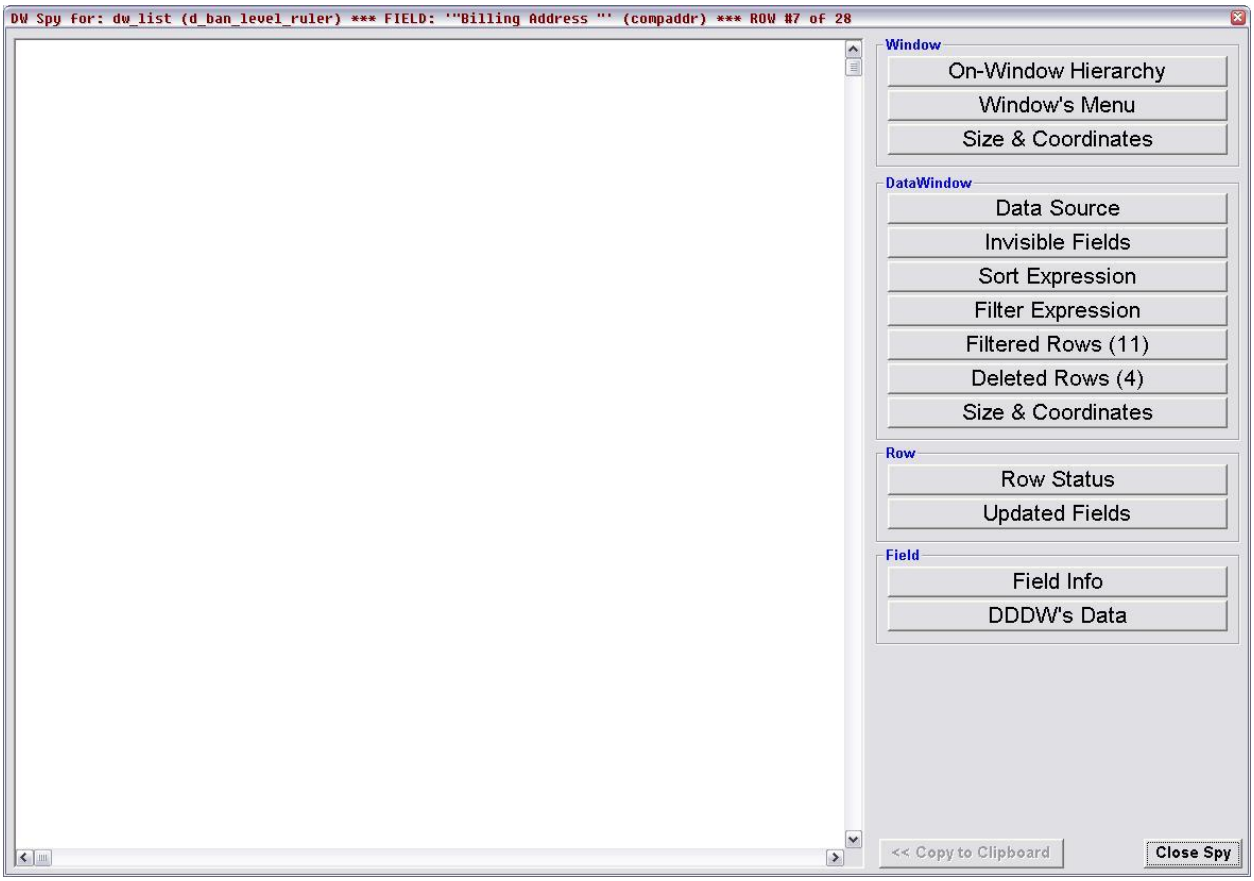
```
//------------------------ Open DW Spy ------------------------
n_parm lnv_parm

if KeyDown(KeyShift!) then
   if Handle(GetApplication()) = 0 /* running from PB, NOT as .exe */ then
      if IsNull(row) or IsNull(dwo) then
         MessageBox("DW Spy", "To open DW Spy, click the DW itself (not an
object on top of it).")
      else
         lnv_parm.uf_set(w_spy.PARM_NAME__DW, this)
         lnv_parm.uf_set(w_spy.PARM_NAME__ROW, row)
         lnv_parm.uf_set(w_spy.PARM_NAME__COL, dwo.name)
         OpenWithParm(w_spy, lnv_parm)
      end if
   end if
end if
//------------------------ Open DW Spy ------------------------
```

You can change KeyShift! to another key or keys combinations if you wish, or if Shift is already used for another functionality.

HOW TO USE THE SPY?

As you saw in the code above, the Spy is invoked by pressing Shift and clicking on the DataWindow. To obtain information about a specific row or field, click directly on that row or field. The following illustrates the Spy's appearance immediately after invocation:



Then, to see hidden info, click one of the buttons. The subsequent paragraphs detail the functionality of each button.

On-Window Hierarchy

The complete hierarchy of visual objects, from window down to DataObject, the inheritance chain of each object derived not from a built-in PB class, and

PBLs of all these objects (including the ancestors at any level). This is particularly valuable for project newcomers, eliminating the need for time-consuming exploration of the application within the IDE and extensive scrolling in the Object Browser.

An example for a DW located on a tabpage and using a stored procedure:

```
###### WINDOW:

w_policy (C:\proj\policy.pbl)
    w_modal (C:\proj\lib.pbl)
        w_response (C:\proj\lib.pbl)
            w_base (C:\proj\lib.pbl)

###### TAB:

tab_policy

###### USEROBJECT:

tabpage_driver
    u_tab_driver (C:\proj\driver.pbl)
        u_tab (C:\proj\lib.pbl)
            u_custom_visual (C:\proj\lib.pbl)

###### DATAWINDOW:

dw_driver_claims
    u_dw_list_updatable (C:\proj\lib.pbl)
        u_dw_list (C:\proj\lib.pbl)
            u_dw (C:\proj\lib.pbl)

###### DATAOBJECT:

d_driver_claims (C:\proj\driver.pbl)

###### STORED PROCEDURE (DW'S DATA SOURCE):

s_driver_claims
```

Window's Menu

The menu attached to the window and its inheritance chain (including PBLs).

Size & Coordinates

Size & Coordinates of the window. Useful to troubleshoot a window whose size and coordinates are set dynamically in a script.

Data Source

DW's Data Source (SQL Select or Stored Procedure).

If retrieval arguments exist in the DW then there are the following changes:

1. The button's text becomes "Data Source **and Arguments**".
2. The displayed message contains the arguments' names, datatypes and the values used for the last retrieval.

Invisible Fields

For each invisible field, the following information is shown:

* Name in the DataWindow.
* Name in the DB (if different from the name in the DataWindow).
* Value (if it’s changed then both - the Original and the Current).
* Data type.
* For computed fields: the expression and the value evaluated for the clicked row.
* Updatable or Not Updatable.
* The DWItemStatus of the field.

If the DW has no rows then only row-independent info is shown (such as DB name and data type).

Sort Expression

If the DW sort is applied, displays the sort expression.

Otherwise, the button is disabled and titled "No Sort Applied".

Filter Expression

If a filter is applied, displays the filter expression.

Otherwise, the button is disabled and titled "No Filter Applied".

Filtered Rows

Displays the data contained in the Filter! buffer, using the DataObject of the investigated DW. If the Filter! buffer is empty, then the button is disabled and titled "No Filtered Rows".

The DataWindow, displaying the filtered records, can be sorted by a single column. To sort in ascending order, double-click the field (the column's font turns blue). To change the sorting to descending, double-click the same column again (the font turns red). To restore the original sorting, close and reopen the Spy.

Deleted Rows

Displays the contents of the Delete! buffer, using the DataObject of the inspected DataWindow. If the Delete! buffer is empty, the button is disabled and titled "No Deleted Rows".

Size & Coordinates

Displays the size and coordinates of the DataWindow. This is useful for troubleshooting DataWindows with sizes and coordinates that are dynamically set in a script.

Row Status

The DWItemStatus of the clicked row.

Updated Fields

Displays visible and invisible fields of the clicked row that have been updated (by the user or programmatically), along with their old and new values.

Field Info

A regular field: name in the DataWindow (and the DB name if different), the data type, whether it is updatable or not, the DWItemStatus, the expressions for enabled and update properties (if they exist), the list of the field's attributes and their current (and original, if changed) values.

A computed field: the expression and the data type.

A field with DropDownListBox: the code table ('display value' - 'data value').

A field with DropDownDataWindow: the values of the display column - data column, the DWC's data object and data source, retrieval arguments, etc.

DDDW's Data

Displays the data contained within the DropDownDataWindow of the clicked field, using the DataObject of that DropDownDataWindow.

The Spy displays the contents of the DataWindowChild's Primary! buffer only. If the DropDownDataWindow contains filtered or deleted rows, a message box appears, suggesting opening an additional instance of the Spy by clicking the DataWindow on the current instance, and then clicking the "Filtered Rows" or "Deleted Rows" button on that new instance. A similar message is displayed if there are invisible fields in the DDDW, advising clicking the "Invisible Fields" button on the new Spy's instance.