VISUAL EXPERT

Getting Started

http://www.visual-expert.com

Contents

ABOUT THIS GUIDE

Welcome to Visual Expert.

This manual is intended to show you how to use the product; it will guide you through the main features.

You will work on a sample application installed with Visual Expert, which will run automatically when you launch the product for the first time.

Do not hesitate to contact us if you have any questions about Visual Expert.

Audience

This guide is intended for any PowerBuilder developer.
No specific technical skill level is required.

Installing Visual Expert

Minimum configuration:

- Windows 98, NT 4.0, 2000 or XP
- PC Pentium II or over.
- 64 Mb. RAM.
- 100 Mb. of free space on hard drive.
- Microsoft Word 6 (or later).
- An Html browser (IE 4 / Netscape 4 or later).

Installing the evaluation version:

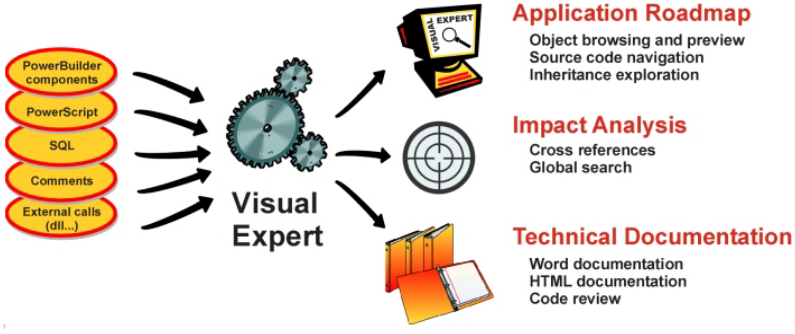
Open **index.htm** file from the CD-ROM and follow the instructions. We strongly recommend you to go through this Getting Started section.

Installing the full version:

Launch the **setup.exe** file from the directory ***\setup*** of your CD-ROM and follow the steps in the install procedure wizard.

INTRODUCTION TO VISUAL EXPERT

Visual Expert instantaneously analyzes your application code, data and comments.
All of the extracted information is viewed on screen or formatted to automatically generate documentation.



Candidate applications for VE analysis

All applications developed with PowerBuilder, versions 4 to 10.

Applications of any size (Visual Expert can analyze several hundred PowerBuilder libraries).

Any framework or object library developed with PowerBuilder.

Visual Expert includes powerful source code parsers. It will analyze 10 to 30 Mbytes of code per minute depending on your machine speed.

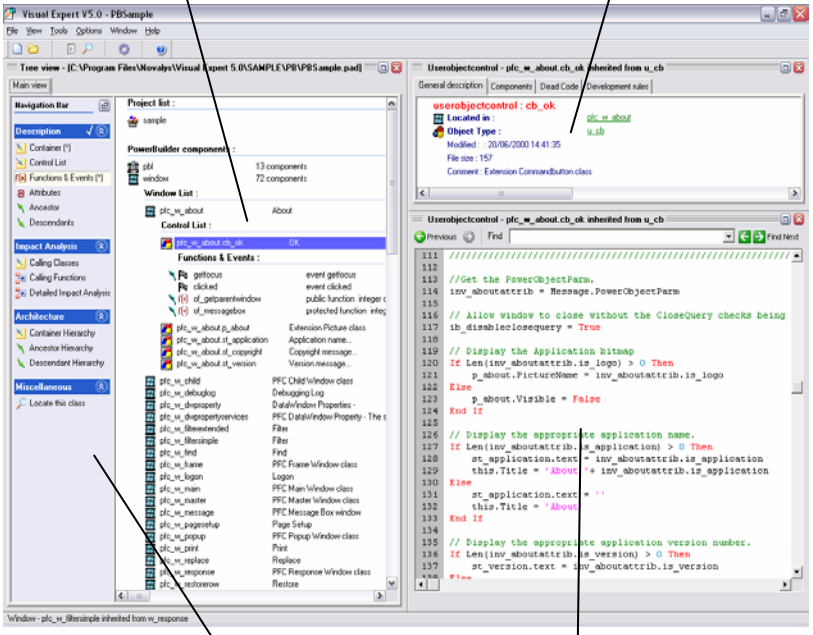
Comments inserted in the source code are not required by Visual Expert to analyze your application. However, if your application does contain comments, Visual Expert can include them in the information provided to the developers.

EXPLORING AN APPLICATION

Graphical interface overview

TREEVIEW
It is the core of Visual Expert.
It displays the application components and their relations.

DETAIL VIEW
gives **Information** about the component selected in the treeview.

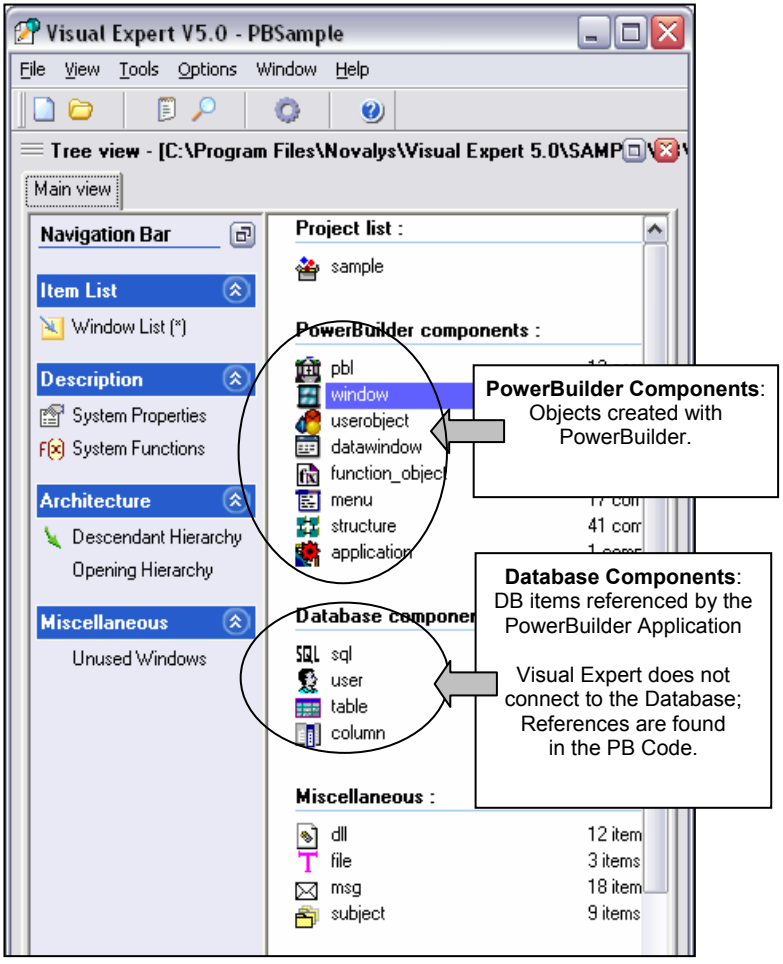


TREEVIEW NAVIGATION BAR
presents various **actions** according to the component selected in the treeview.

SOURCE CODE VIEW
displays the **Source code** of the selected component.
It works as an **HTML browser**.

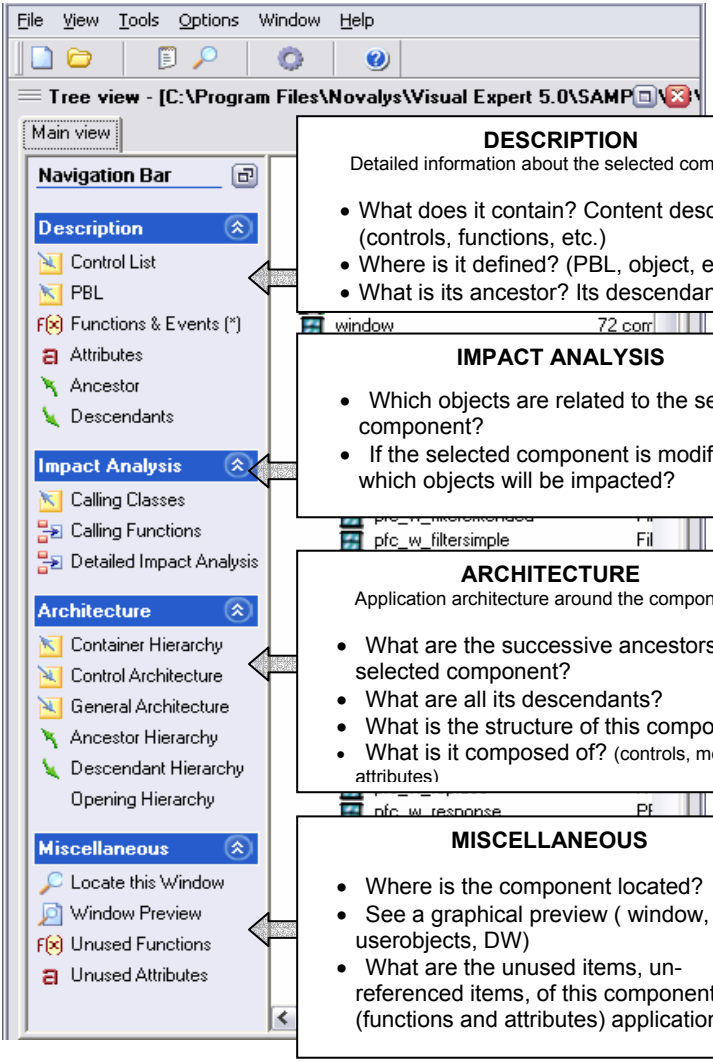
Starting the exploration – the entry points

After opening a project, the treeview displays a list of items sorted by type. Each item is a possible entry point to explore the application:



The Treeview Navigation bar

The Navigation bar layout depends on the component selected in the treeview. Navigation bar options are grouped in 4 sections.
Each section provides a different type of information:



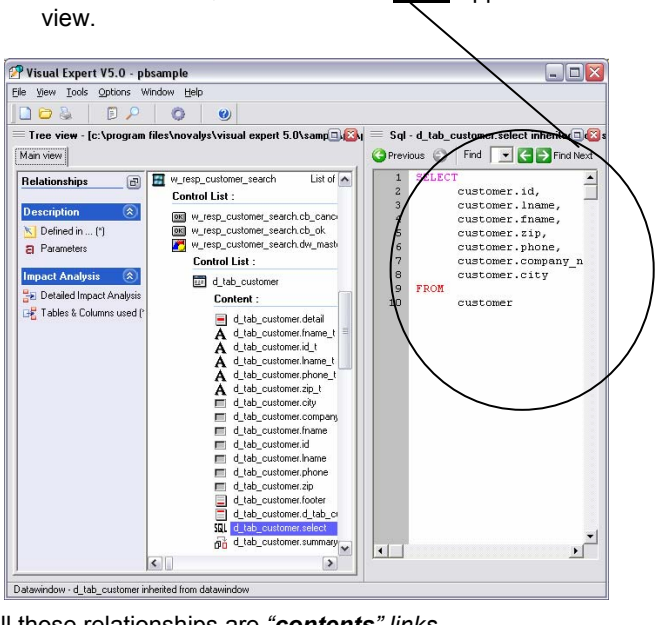
Exploring the application structure

In the Navigation bar, some options of the section ***Description*** show the CONTAINER/CONTENT relation between the components.

Example of the "contents" link

In the treeview, go to ***Project list*** and select the ***sample*** project.
In the Navigation bar, select ***PBL List*** to see the PBL related to this project.

- Select the PBL ***demopfc,*** click on **Component List** in the Navigation bar.
- Select ***w_resp_customer_search***, click on **Control List**.
- Select ***dw_master***, click on **Control List**.
- A datawindow appears, select it and ask for the **Content**
- Click on the ***SQL*** icon: see the **code** appear in the source code view.



All those relationships are *"**contents**" links*.
They allow you to deeply explore the application structure.

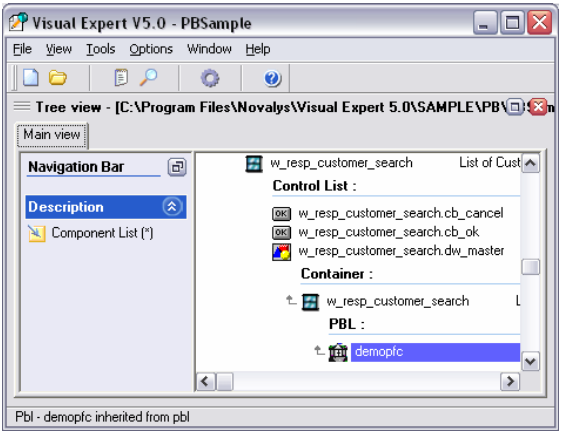
For each kind of relation, a reverse relation is available: in this case, the reverse of ***content is container***.

Example of the “container” link

The container of a window is a PBL; the container of a control is a window...

From our last example:

- Select ***dw_master***, click on **Container** in the Navigation bar.
- The window ***w_resp_customer_search*** appears again.
- Select the window, click on ***PBL***: ***demopfc*** is displayed again.



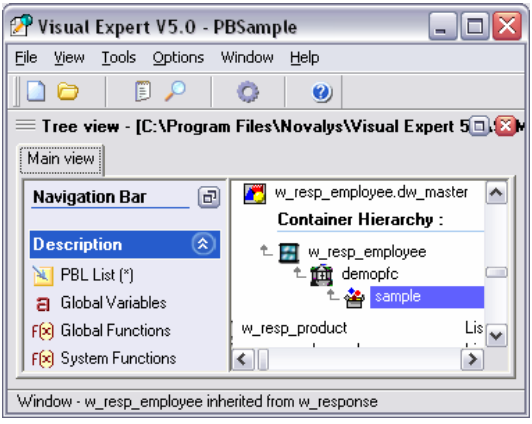
Notes:

- The “container” option will have a different label according to the selected object (***container*** for controls, ***PBL*** for windows, etc.).
- This label appears clearly in the treeview as a title (see “Control List”, “Container” and “PBL” in the screenshot above); the opened treeview is therefore very easy to read.

Advanced options

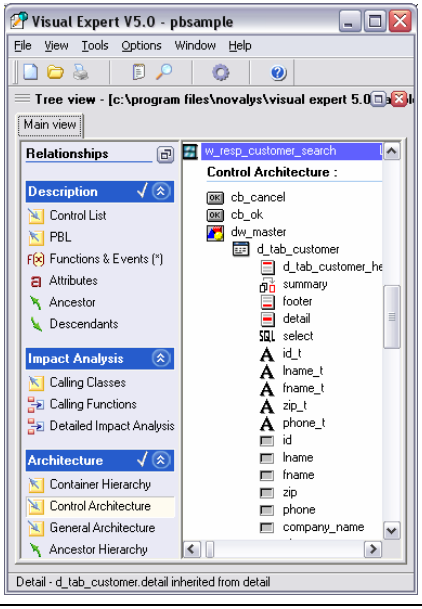
Display all successive containers of a component in one click

- Select ***dw_master*** again, click on **Container Hierarchy** in the Navigation bar.
- The **Window**, **PBL** and **Project** containing ***dw_master*** will all appear.



You can also move in the opposite direction, displaying all successive contents of a component.

- Select the window ***w_resp_customer_search*** again, click on **Control Architecture.**
- The complete hierarchy of controls appears.



If you want to see functions and events at the same time, try **General Architecture:** This option displays a combination of **Control List** and **Functions and Events** links, showing the complete interweaving of components.

With a single click you display the complete structure of the selected component.

Exploring inheritance relationships

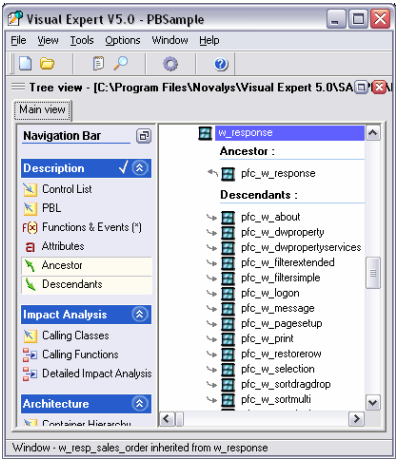
Visual Expert displays the ancestors or the descendants of each inherited object (Windows, UserObjects and Menus).

This improves your understanding of an existing application, as well as the one of the framework used.

Example of "Ancestor/Descendant" links

From our last example:

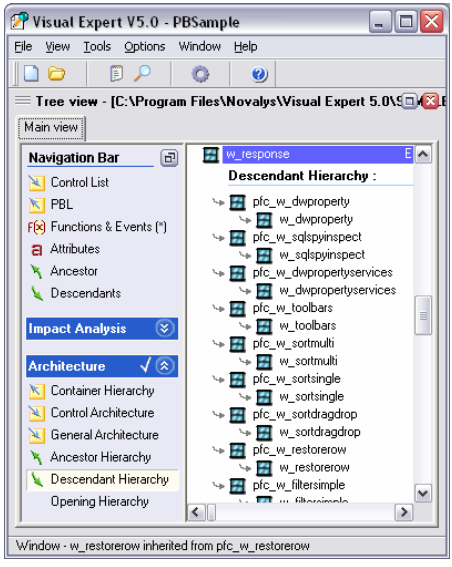
- Select ***w_resp_customer_search***, click on **Ancestor**: ***w_response*** appears.
- Select ***w_response***, click **Descendants**: several windows appear, Showing that ***w_response*** has been inherited several times.



Advanced options

You can display several levels of inheritance at once - For example:

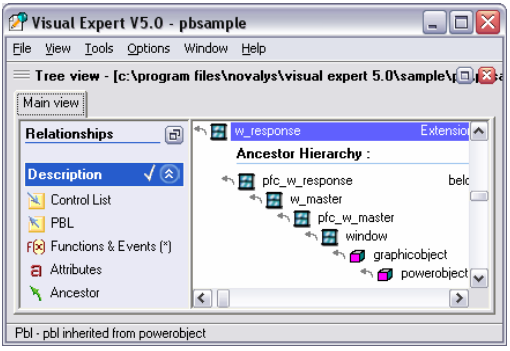
- select *w_response*, click on Descendant Hierarchy



Not only are the direct descendants of ***w_response*** displayed, but so are the successive descendants. Use this feature, for instance, to list the descendants impacted by a modification on the ancestor ***w_response***.

The opposite feature is also available:

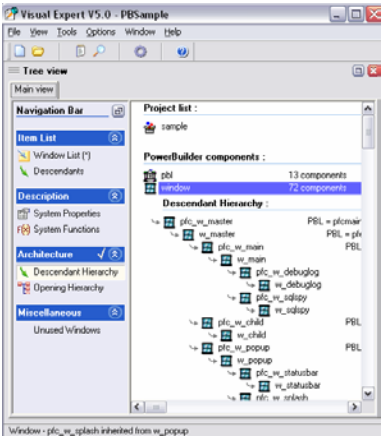
- Select ***w_response*** *again*, click on **Ancestor Hierarchy**: all the successive ancestors of this window appear.



Your application's inheritance tree

With a single click, you can see the complete inheritance tree of your application - For example:

- Press the key **[F5]** to close the treeview and refresh the view.
- Go to the section ***PowerBuilder components*** of the treeview, select ***window***,
- Click on **Descendant Hierarchy** in the Navigation bar.
- Then Visual Expert displays the ancestor windows and all successive descendants.



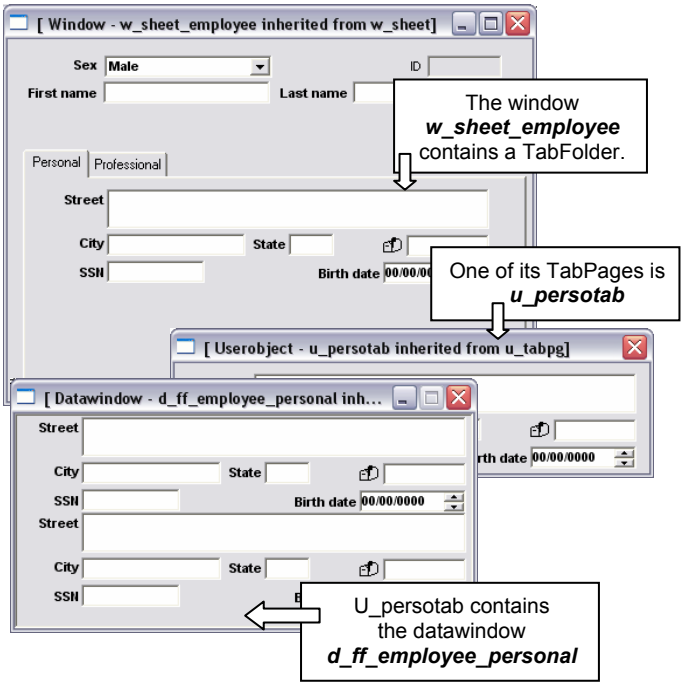
You can also get the inheritance tree for userobjects and menus.
If you wish, you can print this tree; select [File **- Print - Treeview**].

Previewing objects

To visualize or test graphical component, Visual Expert allows you to rapidly preview any of them (Windows, visual UserObjects and Datawindows).

For example:

- In the treeview, go to **Project list** and select the **sample project**.
- In the Navigation bar, click on **PBL List**, and select ***demopfc***
- Click on **Component List** in the Navigation bar.
- Select at the same time (with the [Ctrl] key of your keyboard): ***d_ff_employee_personal, u_persotab*** and ***w_sheet_employee***
- Click on Preview Object in the bottom of the Navigation bar.
- Visual Expert creates a preview of those 3 objects:



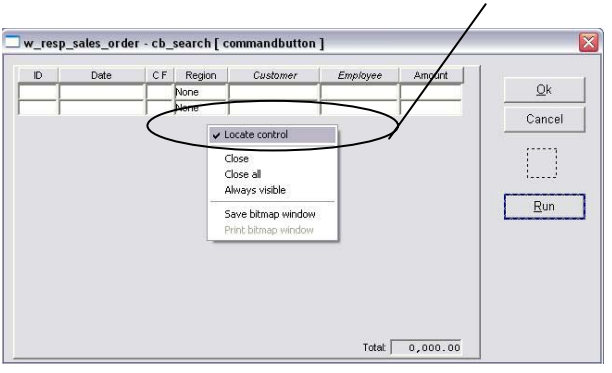
Note: these previews are included in the generated documentation.

Locating a control from a Preview

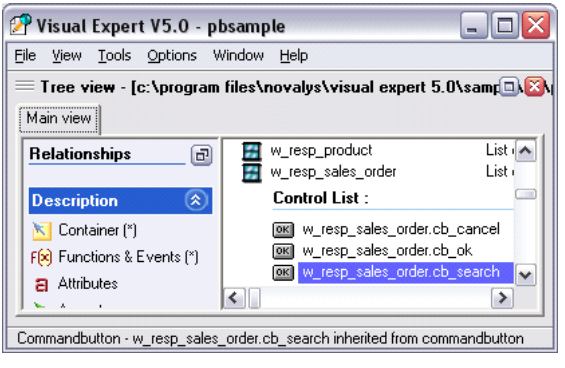
Sometimes, you know where a control is located on a window, but not in the source code! Visual Expert can locate it in the application.
Open a preview, select the control and the treeview will show its definition.

From our last example:

- Select the window ***w_resp_sales_order***.
- Click on ***Window Preview*** in the bottom of the Navigation bar
- Right-click on the preview. Select ***Locate Control***.



- Select the ***Run*** button.
- The treeview now shows the corresponding control (***cb_search***) and where it is located



Conclusion

Visual Expert makes it easy to navigate within the application no matter how large or complex it is.

Visual Expert helps you **gain time** in the development and maintenance of your application.

Visual Expert makes it easier **to integrate new developers**.

The following pages will show how Visual Expert helps you with IMPACT ANALYSIS.

PERFORMING AN IMPACT ANALYSIS

During application development or maintenance, developers repeatedly face the impact analysis problem.

When the code is modified, you need to be certain that these **modifications will not have a negative impact** on the rest of the application, because there is always a considerable risk of regression errors due to the modifications.

PowerBuilder does not provide any impact analysis feature. Global searches are possible, but they are tedious and often incomplete.

Visual Expert helps you to perform the impact analysis.

- With a single click, you know which components use the object you are working on
- You have advanced search features
- You can rapidly visualize the source code after a search

Impact Analysis is possible on all of the application's components.

This chapter shows how to perform an impact analysis on:

- A **column** from the database
- An **attribute**
- A **method** (function or event)
- A **global variable**
- A **global function**
- A **stored procedure** called by the PowerBuilder code.

The « ADVANCED FEATURES » chapter includes an example of a **DLL** called by a PowerBuilder application.

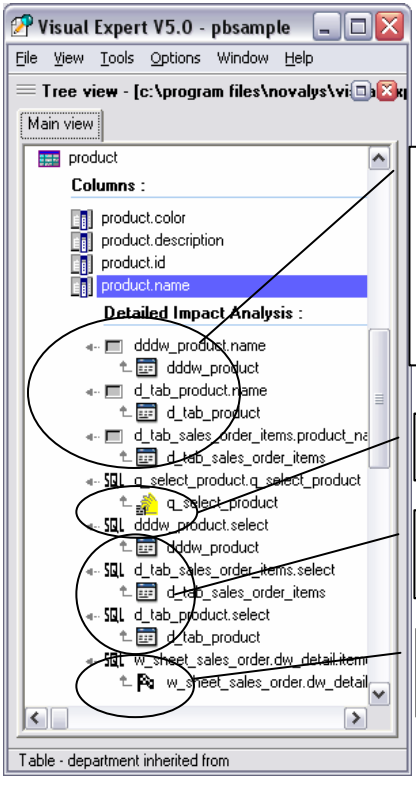
Impact analysis on a column

Visual Expert analyses the tables and columns manipulated by the application.

One of the tables used by the sample application is named ***product***. Let's assume that the column ***product.name*** needs to be modified; we need to evaluate the impact of such a modification on the PowerBuilder application.

To do so, press [**F5**] to refresh the view:

- Go to the section ***Database components*** of the treeview
- Select ***table***, click on **tables used** in the Navigation bar
- Select ***product***, click on **Columns** in the Navigation bar,
- Select ***product.name*** and click on **Detailed Impact Analysis**



These 3 datawindow columns are referencing the column ***name***.

The datawindows containing each column are displayed at the same time. For example: the first column belongs to ***dddw_product***.

This SQL is stored in a query object.

These 3 SQL statements come from 3 datawindows.

This SQL statement is embedded in the event ***itemchanged***.

The source code view:

- Click on the SQL item and see its source code in the source code view.
- Select the ***Itemchanged*** event source code and look for the embedded SQL in the source code view.

With PB 8, 9 or 10: To modify the element you are displaying, if PowerBuilder is already running on your seat. You can access it directly from Visual Expert: Select ***Open in PB IDE*** in the Navigation Bar.

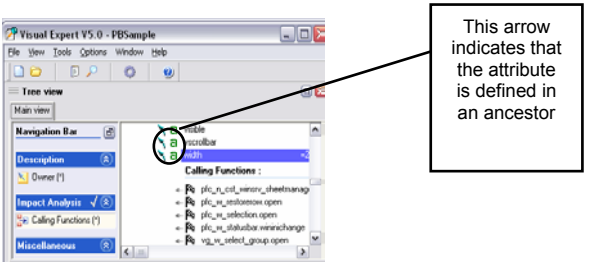
To Include the change in the Visual Expert analysis, you need to select ***refresh code analysis (Ctrl + A)*** in the Tools menu to relaunch the analysis.

Impact analysis on an attribute

Let's display the attributes of a window. We will then display the functions calling one of these attributes:

Press the key **[F5]** to refresh the treeview.

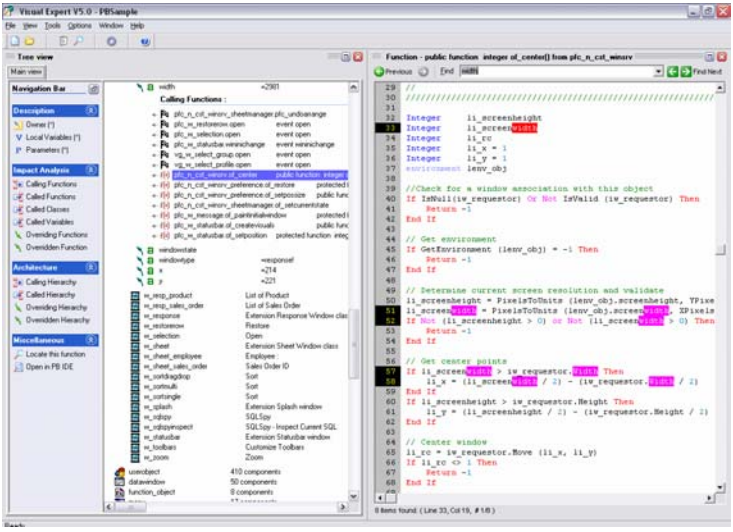
- Select the ***sample*** project in the ***Project list,*** click on **PBL List**
- Select ***demopfc***, click on **Component List**.
- Select ***w_resp_employee***, click on **Attributes**
- Among the displayed attributes, some - marked with an arrow to the left (see next page) - are defined in an ancestor
- Select the attribute ***"width"*** at the end of the list
- Click on **Calling Functions** in the Navigation bar
- Several Events and functions appear.



The first function (***pfc_n_cst_winsrv.of_center)*** centers the window on the screen. To do so, it uses the width of the window.

In the source code of the function ***of_center,*** you can look for the references to the attribute ***width***:

- In the ***find*** field of the source view, type « ***.width*** »
- Click on the **find next** button
- Each reference to the attribute ***width*** is highlighted.

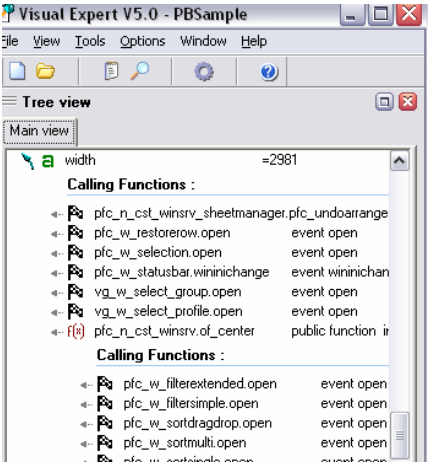


Impact analysis on a function

To find the methods that call a function, follow the same steps as that for an attribute.

Before modifying the function ***of_center***, perform an impact analysis to evaluate what items will be affected by the change.

- Select ***pfc_n_cst_winsrv.of_center,*** click on **calling functions**
- This function is used by several open events.



You can select the option ***Open in PB IDE*** in the Navigation Bar (Only available in PB 8 to 10) to modify your application in PowerBuilder.

To understand the impacts of a modification, it is useful to navigate through the results in order to check it in detail. Visual Expert facilitates this navigation by displaying the source code of the selected component.

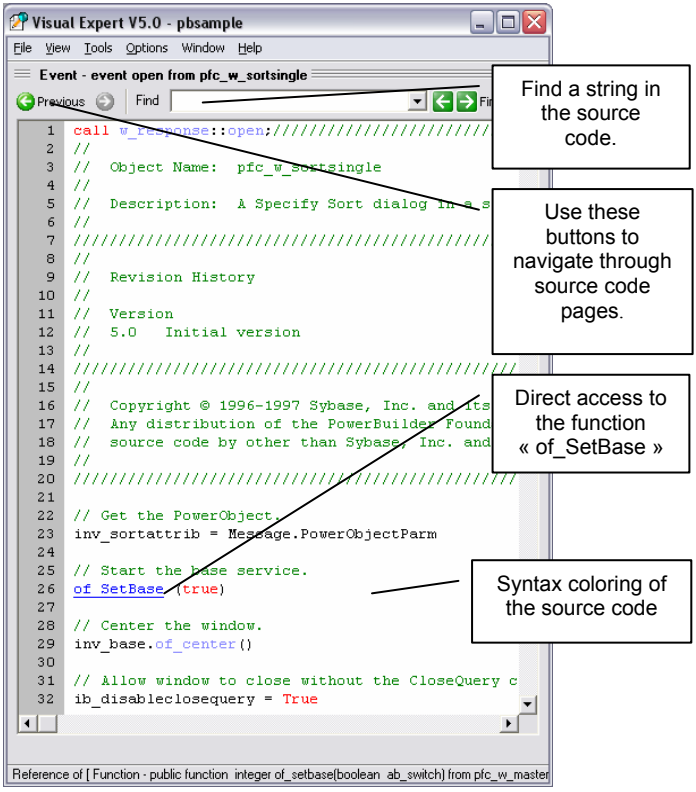
The following chapter explains how to use Visual Expert's source code view.

Navigating through the source code

The source code view works as an HTML browser:

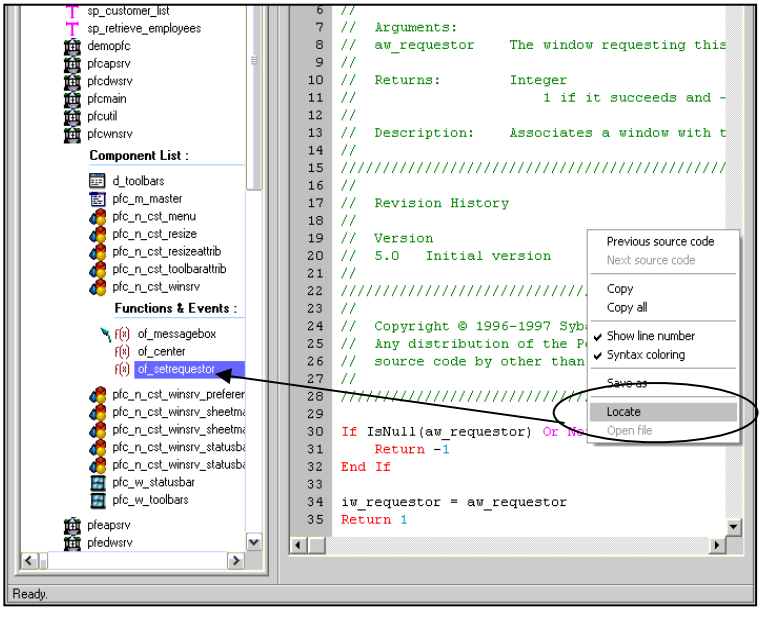
- Hyperlinks jump you directly to the referenced component
- The two buttons display the previous/next pages.
- The syntax coloring of the code makes reading easy.

- A search field helps you to find text in the code.



From our last example:

- Select ***pfc_w_sortingle.Open,*** from the events found in the previous impact analysis.
- The source code contains two hyperlinks to the ***of_SetBase*** and ***of_Center*** functions. Click on ***of_SetBase*** to display its code.
- This function contains a reference to ***of_SetRequestor,*** click on it. The ***of_SetRequestor*** source code is displayed.
- Once you have navigated to a function, it might be useful to know where this function is located within the application.
- To do so, the source code view provides a locate feature.
- Right click in the source code view, click on Locate
- The treeview opens the PBL and the object containing the function of_SetRequestor, which source code was displayed.

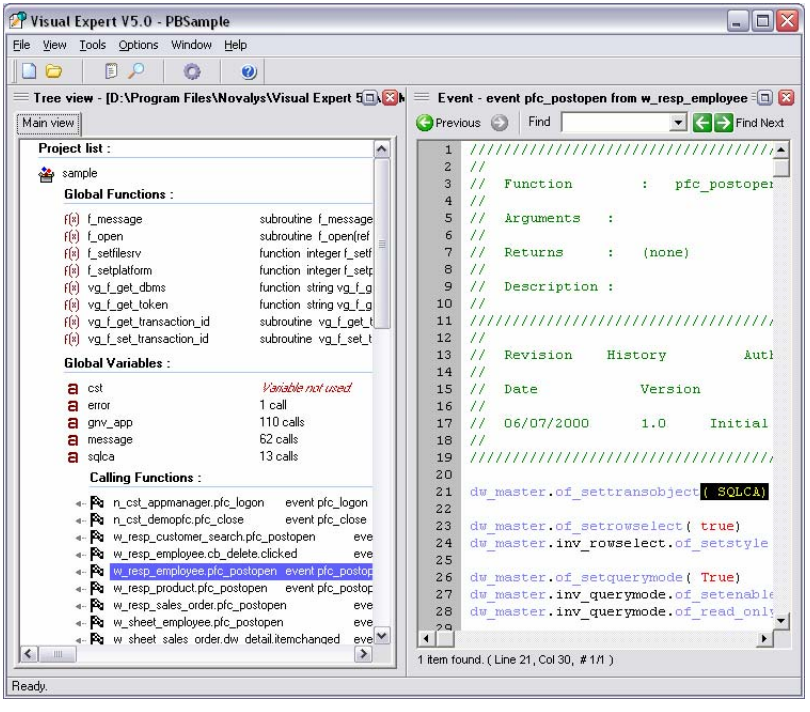


Impact analysis on a global variable

Global variables and functions are available when a project is selected in the treeview:

- Press the key [F5] to refresh the treeview.
- Select sample in the Project list, click on Global Variables
- Press the [Ctrl] key and click on Global functions: both Global variables and functions appear at the same time.
- Select the variable sqlca, click on Calling functions
- As a result, 13 methods calling sqlca appear in the treeview

The exact same feature is available for every global function.



Note: it is always possible to simultaneously select several options of the Navigation bar using the **[Ctrl]** key.

Searching for text in the application

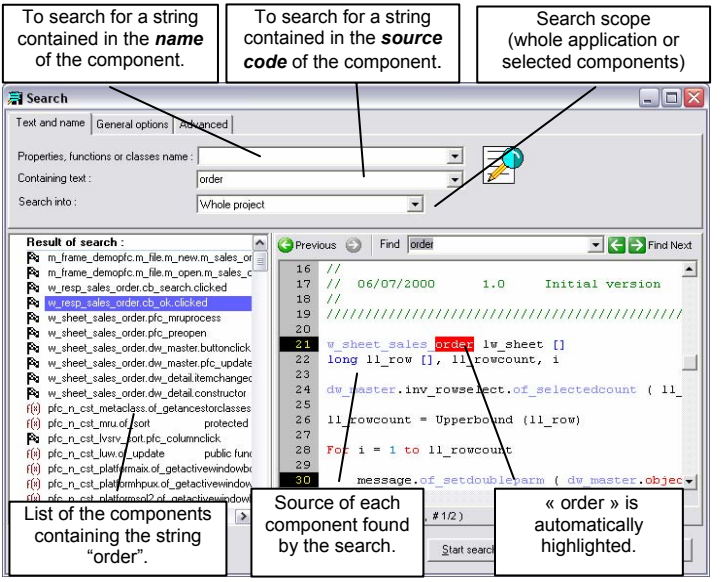
As in PowerBuilder, it is possible to search for a string in the whole application. However, the results obtained in Visual Expert are more valuable than the results of a PowerBuilder search.

You can look for:

- Components whose name contains the string your are looking for
- Components whose source code contains this string
- A combination of both.
- You may complete the search with several parameters.

Let us look for all the components containing the string ***order*** in their source code:

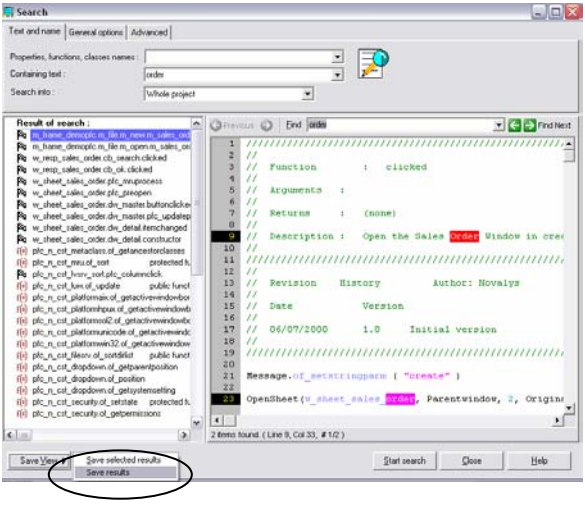
- Press the key [Ctrl-R] to open the global search window,
- Type out order in the containing text field, Click on Search
- A list of functions and events appears
- The source code of the selected method is displayed on the right pane of the window.



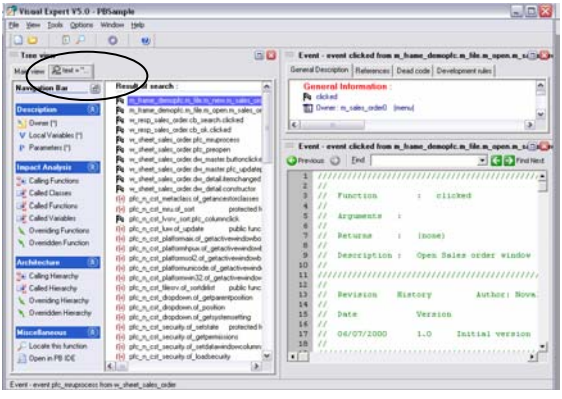
Saving results in a view

In our example:

- In the search view, click on the button Save view
- Select the option Save result



- Close the search window
- A new tab page named order - the searched text - appears in the treeview. Click on the tab to see its contents



The result of the search has been transferred to the treeview.
It allows you to:

- navigate through the result,
- save the result for later use,
- compare several search results,
- Use the Navigation Bar
- Etc…

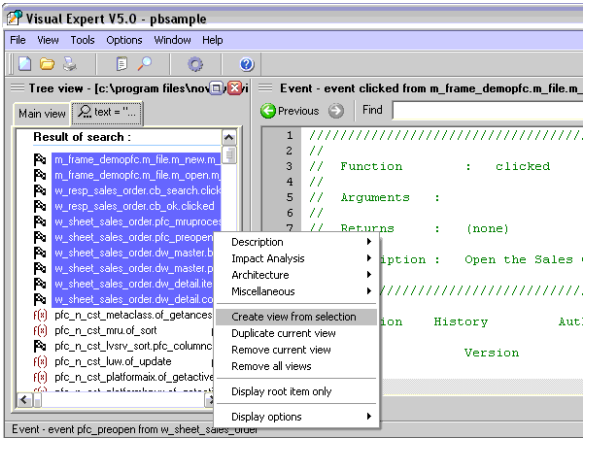
The treeview tabs are also called "***views***". You can create as many views as needed. There are two reasons to create a view:

- To store a search result.
- To bring out a part of a view.

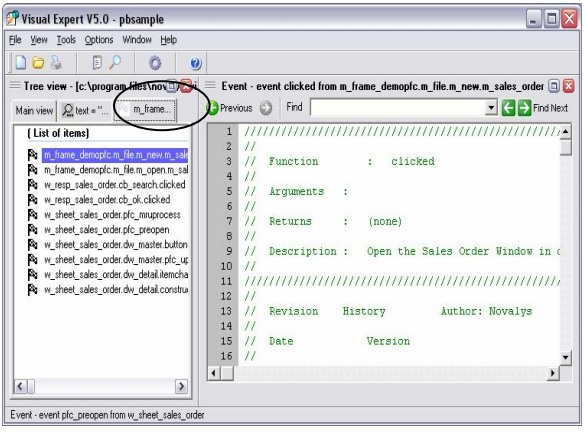
Creating a view from another view

To create a view from another view:

- Select components in the **main view**
- Click on the **first** item of the view
- Press the **[Shift]** key (and keep it pressed)
- Click for instance on the **10th** item of the selection



- Right-click on the selection, select ***Create view from selection***
- A new view is created in the treeview with the selection



Impact analysis of a stored procedure

If the application contains calls to stored procedures, it is useful to analyze the relationships between the PowerBuilder code and these procedures.
Since these stored procedure calls are coded in PowerBuilder, Visual Expert fully analyses them.

Visual Expert

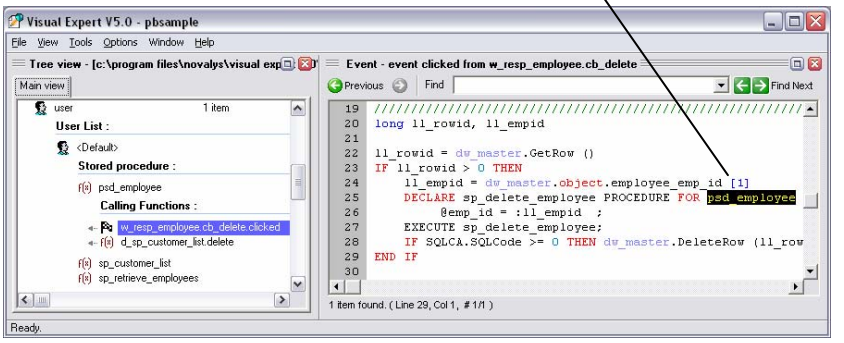
- Lists the called procedures
- Finds all the references to a stored procedure within the
- PowerBuilder application (impact analysis).

However, Visual Expert does not connect to databases and cannot automatically read their source code. To view the stored procedure source code by Visual Expert, see the « ***external files*** » chapter.

To do an impact analysis on a stored procedure:

- Press the key [F5] to refresh the treeview.
- Select User in the database component section
- Click on User List in the Navigation bar.
- Select the user <default>, click on Stored procedure.
- By default, stored procedures are linked to the <default> user unless another user is declared in the code.
- Select the procedure psd_employee, click on calling functions.

- Check in the event source code the call to psd_employee in the clicked event.



Viewing a stored procedure in the source code

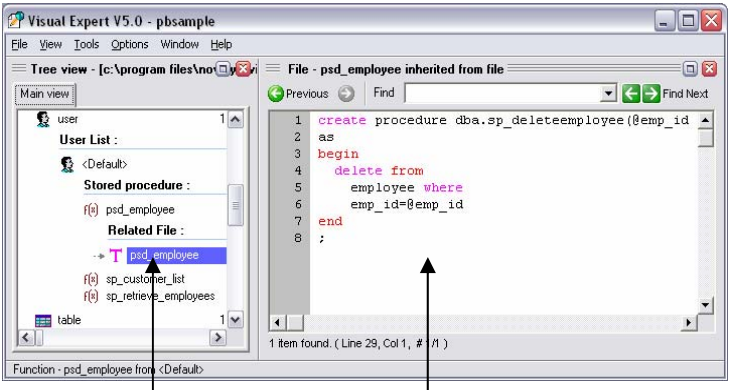
In addition to the application PBL, other files may be added to the Visual Expert project. Those could be text files, like .ini files or ***stored procedure exports***. Visual Expert helps in 2 ways.

1. Visual Expert takes into account stored procedures (or other text files) when you ***search for a string*** in the whole application. This feature is always available as long as you include the related files in the Visual Expert project.

2. You can view stored procedures source code. The following conditions should be fulfilled:
   - Each stored procedure should be exported as a separate file.
   - This export file should have the same name as the store Procedure referenced in the PB code.
   - Export files should be included in the Visual Expert project.

In our example:

- Select again ***psd_employee***, click on ***Related File***.
- Select the file "***psd_employee"*** and look at its source code.



This file is related to the stored procedure ***psd_employee***

It contains the source code of ***psd_employee***

Conclusion

You have now used the main features provided by Visual Expert to explore the source code and perform impact analysis.

These features are very useful for developers day after day.

Sometimes you need to share technical information about the application on printouts or through the web. To do so, we will go through the documentation features provided with Visual Expert.

GENERATING TECHNICAL DOCUMENTATION

The documentation templates

Until now, we have only used the information analyzed by Visual Expert on screen.

This information can also be "exported" to generate technical documentation.

Several documentation types are supported by Visual Expert. These are called "templates". Visual Expert is delivered with about thirty different templates. These templates cover most of the needs a developer may encounter.

The results can be RTF or HTML documents.

NOTE:

For an existing application, the graphical interface is more efficient than hard copy technical documentation because you can make use of the hyperlinks.

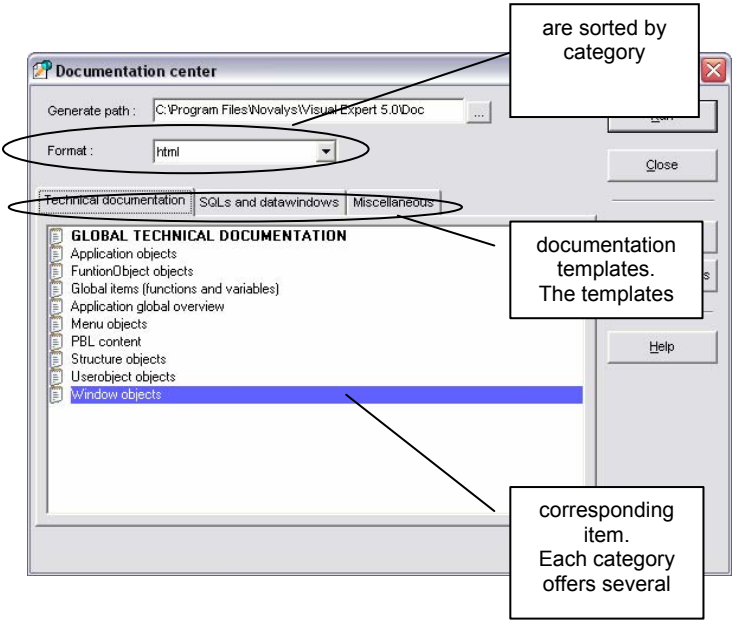
The information displayed on screen is more complete and flexible than printed technical documentation.

How to generate documentation

To generate documentation with Visual Expert, press [Ctrl+W] or click on the button. 

To create the document, select the appropriate template and click on the ***Run*** button.

Most of the templates include a wizard to help you define the contents of the documentation.



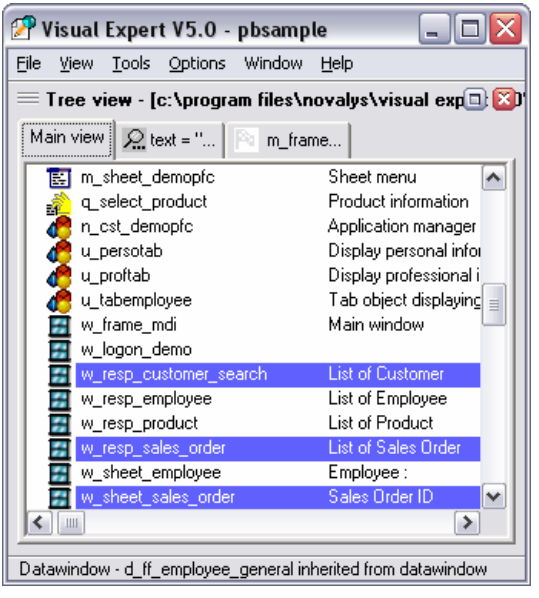
Let us assume that a new developer needs to modify the *sales order* part of the application.

He/she needs to generate documentation concerning the impacted windows:

- w_resp_sales_order selects a purchase order
- w_sheet_sales_order modifies a purchase order
- w_resp_customer_search selects the customer associated to the purchase order

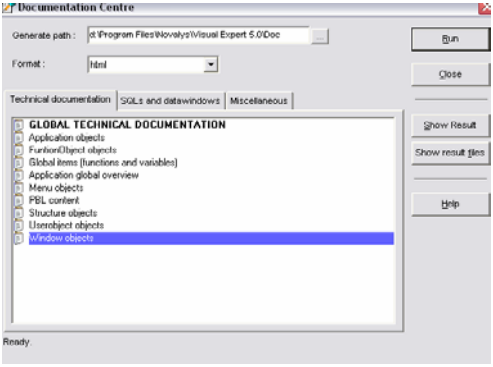
To generate the documentation for these three windows:

- Press the key [F5] to refresh the treeview.
- Select the sample project in the Project list, click on PBL List
- Select demopfc, click on Component List.
- Press the [Ctrl] key (and keep it pressed)
- Select w_resp_sales_order, w_sheet_sales_order and w_resp_customer_search



Open the **documentation centre** (button ).

- Select ***HTML*** format,
- Click on **Technical documentation tab**
- Select the ***Window objects*** template.
- Click on the button ***Run***



The ***Window objects*** template opens a wizard. In order to view all the possibilities visual expert offers, we will request detailed information

- On the first wizard page, check the <u>3 first boxes</u>:

- On the 2nd wizard page, go to the next step without modifying the default options.

- On the 3rd wizard page, check the option ***selected components***

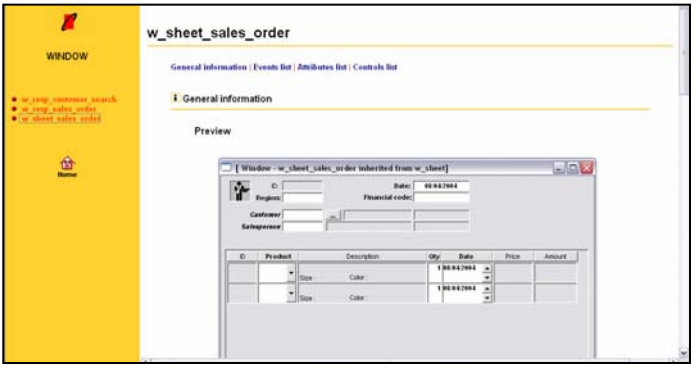
As we selected 3 specific items in the treeview.

- Click on the Generate button to create the documentation.
- A message box will ask you if you want to see the result.

Click yes and you will display a web page since we selected an HTML format.

The left frame lists the documented components.

- Choose the window ***w_sheet_sales_order***.
- The information about this window is displayed in the right frame.
- Shortcuts jump you to the main chapters of this page.



- Click on the shortcut ***events list***.

- Click on ***pfc_postopen*** to see this event in detail.

Conclusion

The documentation templates delivered with Visual Expert cover most of the developer's needs for documentation.

They provide very detailed information on your applications and adapt their contents depending on your needs.

Let us discover how Visual Expert helps in improving the quality of the source code.

IMPROVING THE SOURCE CODE

Development rules and standards

Why use Visual Expert?

Projects need to follow development rules in order to:

- **normalize** the code
- **simplify** its reading
- **anticipate** maintenance

Although there is usually a desire to apply these rules, time is often too short and developers do not always remember to apply all the rules!
On the other hand, such rules must be applied at the beginning of the project because it is more difficult to do it later …

To answer these demands, Visual Expert offers a solution which:

- is easy to use by developers
- is implemented in few minutes at the beginning of the project
- take little time during development
- saves a lot of time during the maintenance cycle

Visual Expert will help you during 3 phases:

1. **Before** the development, Visual Expert helps to define the development rules (you can use the ones provided by default)

2. **During** the development, Visual Expert helps to apply the rules and correcting errors

3. **After** the development, Visual Expert can search the application for components that do not comply with the rules or components which are unused.

The goal is to get a standard, documented and easy to maintain application that complies with the development rules.

Defining your own standards

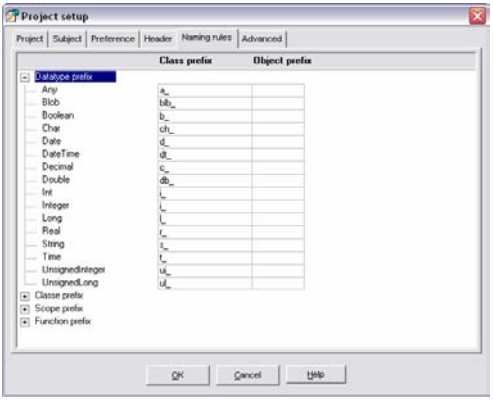
Some rules have to be defined (they depend on arbitrary choices).
This applies to the naming rules and source code comments' definitions.

Naming standards

A tab of the ***project setup*** window is dedicated to these standards.

To define them:

- click on the **Project setup** icon 
- Open the **Naming rules** tab.



Naming standards can be defined according to:

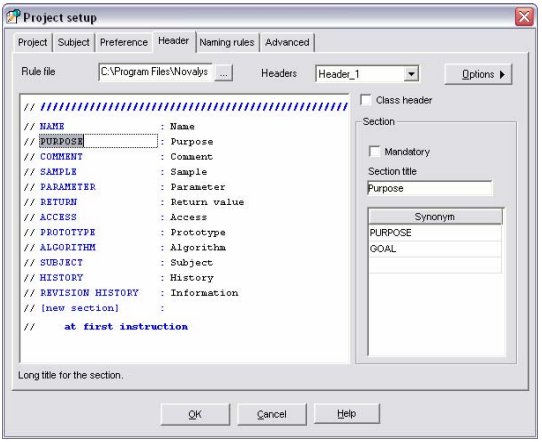
- The object type (windows, userobject, …)
- The function type (menu function, window function, …)
- The attribute type (long, integer, …)
- Its scope (global, local, …)

Source code comment rules

To have meaningful information in the source code, it is strongly recommended that you comment the scripts in a structured way.

For this reason, a script header is to be defined at the beginning of each script. It will include different kinds of information (function name, purpose, prototype …).

The ***Header*** tab of the ***project setup*** window is dedicated to implementing such standards.



Such parameters allow for the analysis of the comments of any existing application, as well as for the generation of consistent technical documentation.

In this case, it is sometimes necessary to define several headers to analyze different types of comments in the existing application.

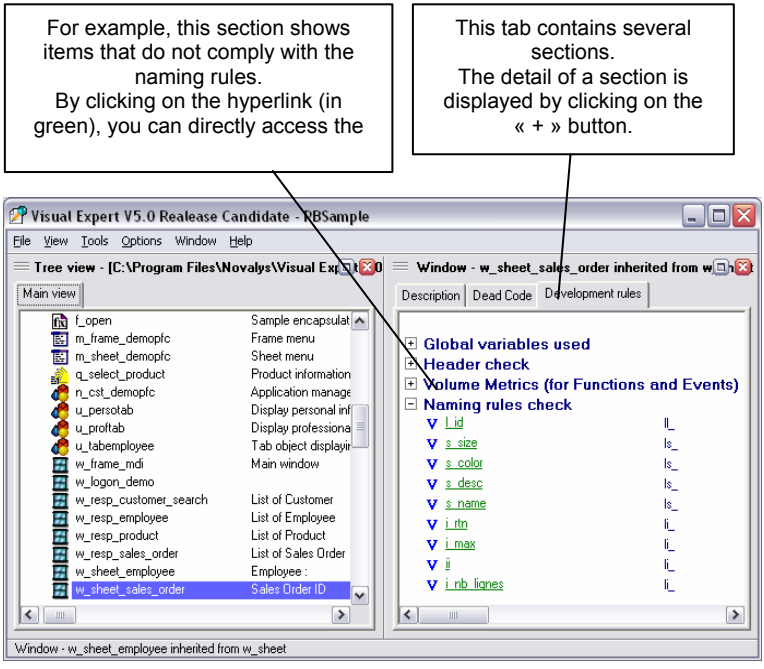
Complying with rules during the development

After creating an object, the developer may check with Visual Expert that the object complies with the development rules.

For example:

- Press the key **[F5]** to refresh the treeview.
- Select the ***sample*** project in the **Project list,** click on **PBL List**
- Select ***demopfc***, click on **Component List**.
- Select ***w_sheet_sales_order***,
- Open the **development rules** tab in the detail view.

Verifying rules after development



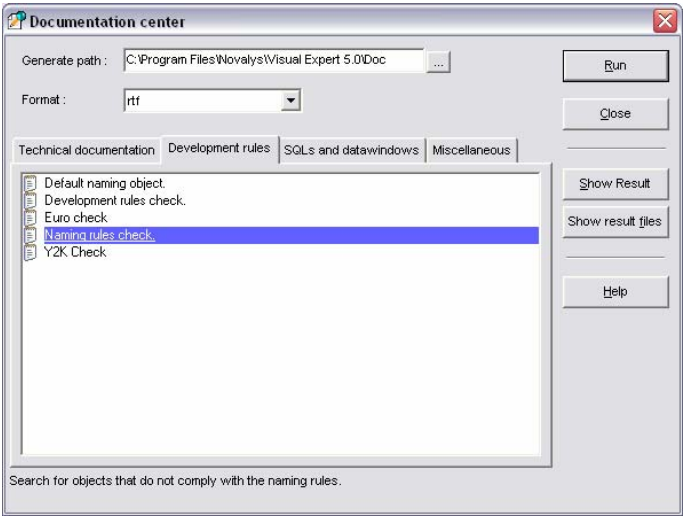
At any moment, Visual Expert helps analyze the whole application to find the components that:

- Do not comply with the development rules
- Are not commented
- Contain too many lines of code
- Have too many levels of inheritance
- Are unused
- Etc…

To see this in action:

- Open the ***documentation center*** (button ).
- Select ***RTF*** format,
- Click on the **Development rules** tab

- Choose the template according to your needs.
- Click on the button Run and follow the wizard instructions.



Unused components

Visual Expert helps to find **unused components** in the application.

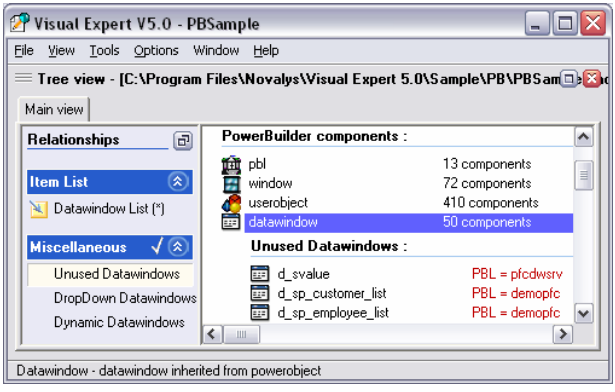
IMPORTANT: By unused components, we mean "***components for which Visual Expert can't find static references to in the source code***". It does not mean a component is not used in the application.

- At least, every item not mentioned as "unused" is definitely referenced somewhere in the application.
- For each "unused" item, you may use Visual Expert Global Search to look for dynamic references to this item somewhere in the code.
- Some dynamic references are not defined in the PB code (for instance, the name of the component can be loaded from a database). In such a case, Visual Expert will not find the name of this component at all.

Unused objects

For PowerBuilder major components (windows, UserObjects, Datawindow, etc.) a list of unused items can be displayed in the treeview.

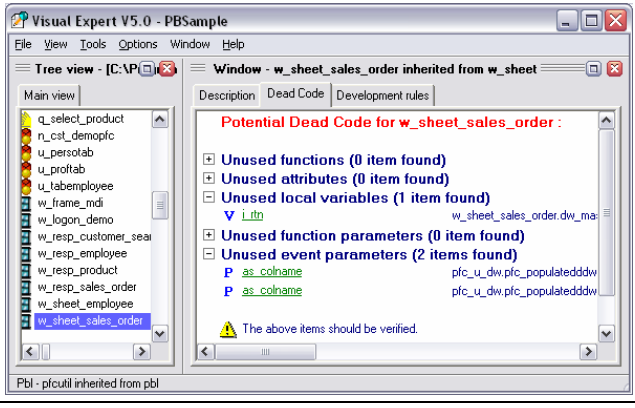
- Select ***datawindow*** in **PowerBuilder components**
- Click on **Unused Datawindows**



Unused functions, variables and parameters

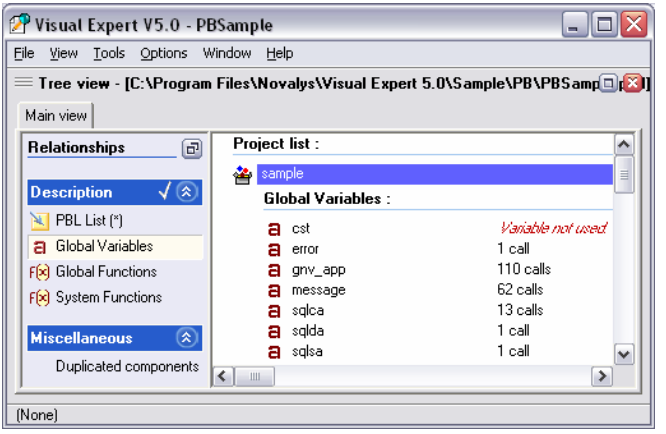
For a given object, you can display unused items in the detail view.

- Select again ***w_sheet_sales_order*** (in the PBL ***demopfc***)
- Go to the **detail view**, open the tab **Dead code**
- Unused functions, attributes, functions parameter, etc. appear:



Unused global variables & functions can be displayed in the treeview:

- Press the key **[F5]** to refresh the treeview.
- Select **sample** in the **Project list**, click on **Global Variables**
- The usage of each variable appears next to its name :



Conclusion

Until now, we have discussed the main features provided by Visual Expert.

To improve your knowledge of Visual Expert, we suggest that you go through the practical exercises presented in the following chapter.

The **« Advanced Features »** chapter allows you to go deeper in using Visual Expert with some supplementary services.

PRACTICAL EXCERCISES

Exploring an application

Using the object types

To visualize the whole **inheritance tree** of the application windows:

- Select the entry point ***Window*** in the treeview
- Click on ***Descendant hierarchy*** in the navigation bar.

Do the same for the category ***userobject*** to visualize the complete object inheritance tree of the PowerBuilder Foundation Classes.

Navigating in the application

In this example, we will use Visual Expert to understand how the window ***w_sheet_employee*** works.

- Select ***w_sheet_employee*** in the **window** list.
- Ask for a preview of this window by clicking on ***preview window***
- Determine where it is located in the application (***locate***).
- Determine the parent window (***Ancestor***).
- Visualize the objects & code contained (***Architecture***).
- Determine the objects calling this window ***(Calling functions)***

Getting details about an object and navigating in its source code

The "detail" view is used to get more information on a given component:

- Select ***w_sheet_employee*** in the windows list.
- Click on ***Functions & Events*** to list all its methods.
- Select the method ***of_accepttext***.
- In the ***Description*** tab of the "detail" view, parameter descriptions and return code values can be viewed**.**

- In the "source code" view, search for the ***of_isupdateable*** function and click on its' hyperlink to display its source code.
- Right click on the source code view, select Locate in the pop-up menu. The object containing this function appears.

Searching for a string

Using the Global Search window, we will find where the date format "dd/mm/yy" is applied.

- Open the Global Search window and search for all the objects containing the text "dd/mm/yy".
- For each object found, check that the mask used is "dd/mm/yyyy".
- Open the « ***General options*** » tab of the Search window and check the option « ***Search into script only*** ».
- Click the ***search*** button: the same search is performed in the scripts of the application instead of the objects of the application.

Generating technical documentation

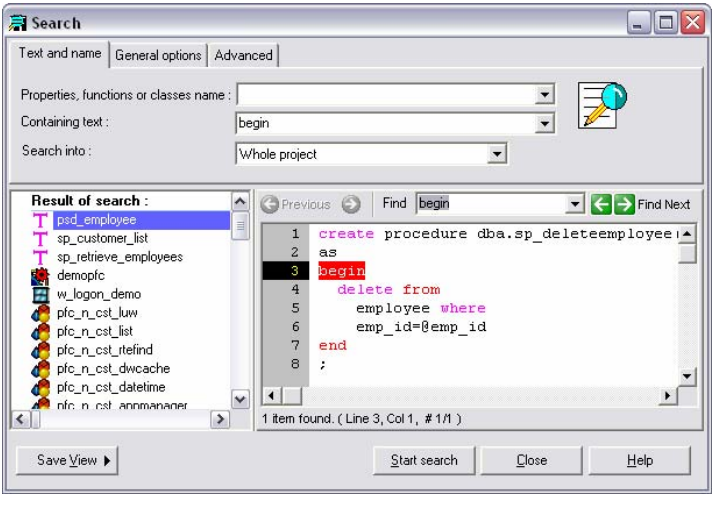
- Open the **documentation centre window**
- Select **RTF** Format
- Search for all the unused variables by using the template **Development rules** check in the **Development rules tab**.
- Generate a RTF document containing the application class inheritance tree by using the template **Ancestor hierarchy** in the **Miscellaneous tab.**
- Generate the same document in **HTML** format.

Global search with stored procedures and text files

As mentioned before, a global search will take into account the text files included in your Visual Expert project.

For example:

- Type [**Ctrl + R**] to open the Search window
- Type ***begin*** in the field ***containing text***
- Open the tab ***General options***
- Uncheck search ***into script only*** (otherwise, only PB events and functions are browsed)
- Click on **Start Search**
- The result contains 3 files (stored procedure exports) and several PB components:



ADVANCED FEATURES

We have already seen how Visual Expert uses the information analyzed in the application.

This information comes from:

- An analysis of the PowerBuilder source code files (PBL).
- A syntactical understanding of PowerBuilder and SQL.

To go further, we will now expand the analysis to the following:

- Localizing Overriding functions
- Identifying the dropdown datawindows and the dynamic windows
- Analysis of calls from PowerBuilder to external components (stored procedure, DLL …).

This helps to evaluate the impact of these components on the PowerBuilder application.
Non-PowerBuilder source files can be added to the analysis (exported stored procedures, .INI files …). Visual Expert will include them in its Global Searches (search for a text in the whole application) and these files will be displayed in the source code view. However, the syntax of these files will not be analyzed.

Overriding functions

You may have the need of finding the overriding and overridden functions of your application.

Visual Expert helps in:

- Listing the overridden function and the functions they override
- Finding all the windows in which the same function is overridden in the PB application

In our example: The sample application is based on a framework, where the ancestor window is defined; if a window has an unusual behaviour, some of its function may have been overridden.

- Press the key [F5] to refresh the treeview.
- Go to ***PBL components*** section in the treeview
- Select ***window***, click on ***Window list in*** the Navigation bar.
- Select ***pfc_w_dwproperty***
- Look at its preview - ***Window preview – Miscellaneous*** in the Navigation Bar.
- Click on ***all overriding functions*** in the Navigation Bar.

The function which replaces one of the ancestor function is displayed.

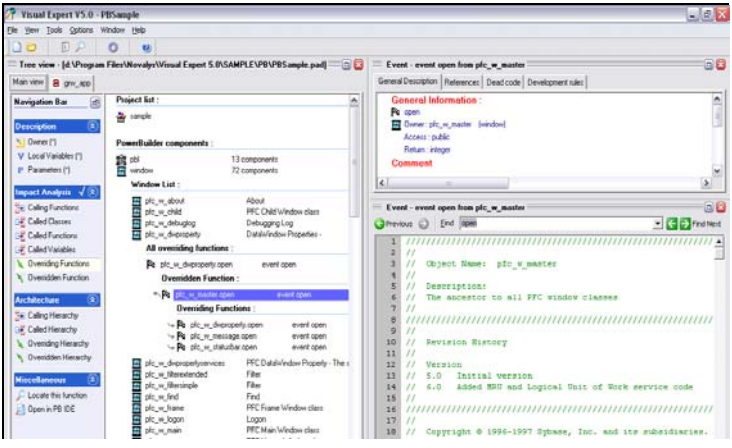
Let's find out which one it replaces:

- Select ***pfc_w_dwproperty.open***
- Click on ***Overridden functions*** in the Navigation Bar.

The function replaced is ***pfc_w_master.open***

Has it been replaced in other parts of the application?

- Select ***pfc_w_master.open.***
- Click on ***Overriding functions*** in the Navigation Bar.
- A list of three windows is displayed, in which the function ***pfc_w_master.open*** is overridden; including our initial window ***pfc_w_dwproperty***



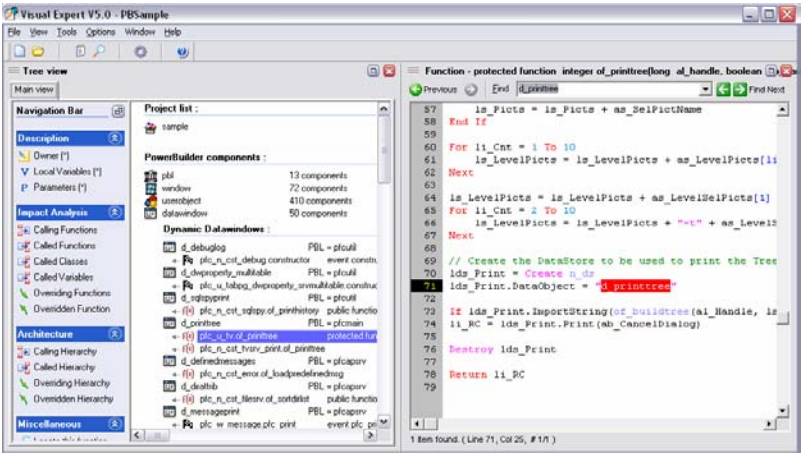
DropDown datawindows and Dynamic datawindows

Some datawindows may be used as Drop Down datawindows in the application. Visual Expert can display them all. This helps the user select items from the database.

- Select ***datawindow*** in the ***datawindow components*** in the treeview
- Select ***DropDown datawindows*** in the Navigation bar
- You may preview **d_tab_departement** by selecting it and clicking on ***Window preview*** in the Navigation Bar
- It is used as a DropDown datawindow by ***d_ff_employee_professional***
- You may preview ***d_ff_employee_professional*** too by selecting it and clicking on ***Window preview*** in the Navigation Bar
- You can display the dropdown datawindow from its *department field.*

Visual Expert can display all dynamic datawindows, hence the datawindows referenced in the source code of a function.

- Select ***datawindow*** in the ***datawindow components*** in the treeview
- Select ***Dynamic datawindows*** in the Navigation bar.
- The datawindow ***d_printtree*** is referenced in the source code of 2 functions. Let's look for it in the source code of the first one*:* Select ***pfc-u-tv.of_printtree***
- Copy and Paste **"**d_printtree" in the find field of the source code view and press on the key ***enter***
- Line 71 is automatically highlighted



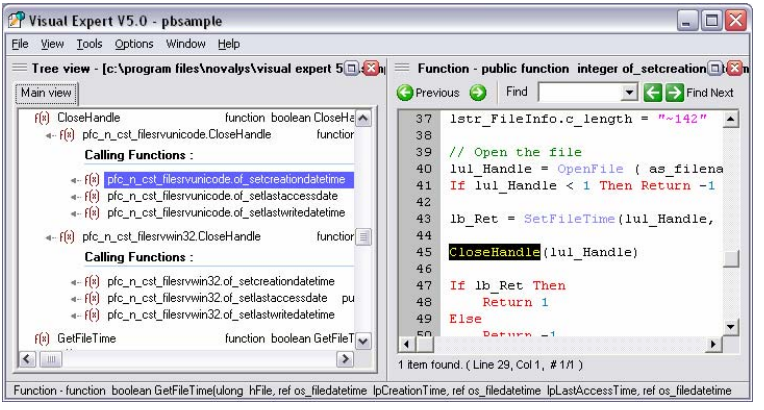
DLL impact analysis

As for stored procedures, the application can call some DLLs for certain processes (calls to Windows basic functions,).
Visual Expert helps in:

- Listing **DLLs** called by PB,
- Finding all references to a DLL function in the PB application

In our example:

- Press the key [F5] to refresh the treeview.
- Select ***dll*** in ***Miscellaneous*** section in the treeview, click on ***Used DLLs*** in the Navigation bar.
- Select ***kernel32.dll***, click on **declared functions**,
- Find the dll function ***CloseHandle***. It has been declared twice in 2 different objects; ***pfc_n_cst_filesrvunicode*** & ***pfc_n_cst_filesrvwin32***
- Select the first declaration, click on ***calling functions***.
- 3 PowerBuilder functions call this function Select the first one (***of_setcreationdatetime***) & check in its source code the call to ***closehandle***



NOTE: only the DLL and DLL functions called by the PowerBuilder application are displayed in Visual Expert. In particular, a DLL may contain more functions than the ones found by Visual Expert in the PowerBuilder application.

This documentation is about the tutorial of the following software:

Visual Expert

IDDN.FR.001.400002.02.S.P.1995.000.10600

NOVALYS S.A
41/43 rue Paul Bert
92100 Boulogne Billancourt
France
Phone: (+33) 1.41.31.82.82
Fax: (+33) 1.41.31.82.90

Visit our web site: http://www.visual-expert.com