

Users Guide

PureVision

© 2024 Reel Media Productions

Table of Contents

Foreword	1
Part I PureVision	3
1 MenuBar	4
File	4
New Project.....	4
Open Project.....	5
Save Project.....	5
Save Project As.....	5
Archive Project.....	5
Save Current Form.....	5
Merge Form with Project.....	6
Export Project.....	7
Quit.....	9
Forms	9
Default Interface Positions.....	9
Tool Windows on Top.....	9
Display GroupBin.....	10
Using the GroupBin.....	10
Display Custom Gadgets.....	11
Enable Gadget Highlighting.....	11
Edit with Skins.....	12
Development Platform.....	13
Use Language.....	13
Settings.....	13
Confirm on Quit.....	13
Auto Save.....	14
Disable AutoUpdate Checking.....	14
Enable Left-Click Move/Size.....	14
Replace with BubbleTips.....	14
Include Images in EXE.....	15
Include PureSkin in EXE.....	15
Snap to Grid.....	15
Icon2PNG Size.....	16
Declare Forms on Export.....	16
Override Gadget Settings.....	16
Tools	17
PureTrans Language Manager.....	17
Print Used Variable Names.....	20
PureSkin Maker.....	21
PurePoint Maker.....	23
Font Manager.....	23
Plugins	25
Help	26
Help.....	26
PureVision Website.....	26
Check for Online Updates.....	26
Enter Registration Key.....	27
About.....	27

- 2 Extended MenuBar 28**
 - MenuBar Editor 29
 - ToolBar Editor 30
- 3 Gadget Creator 31**
 - PureBasic Gadgets 32
 - API PVGadgets 36
 - Custom Gadgets 37
- 4 Gadget Tree 40**
 - Copy 41
 - Paste 41
 - Duplicate 42
 - Duplicate Offset..... 43
 - Grouping 43
 - Hiding 44
 - Lock 45
 - Delete 45
 - Refresh Display 45
- 5 Properties 46**
 - Dynamic Resizing 47
 - Parent/Child Forms 49
 - Changing Fonts 49
 - Gadget Coloring 50
 - StatusBars 51
 - PanelGadgets 52
 - Code Inject 53

Part II PVGadgets User Library 59

- 1 PVGadgets Functions 59**
 - PVDynamic 59**
 - PVDynamic Example Code..... 60
 - PVDynamic_ColorGadget..... 60
 - PVDynamic_AddWindow..... 61
 - PVDynamic_AddGadget..... 61
 - PVDynamic_AddColorGadget..... 62
 - PVDynamic_FreeColorGadget..... 62
 - PVDynamic_AddStatusBar..... 62
 - PVGadgets 63**
 - AnimGadget..... 64
 - PVGadgets_InitAnimGadget..... 64
 - PVGadgets_AnimGadget..... 64
 - PVGadgets_AnimGadgetPlay..... 65
 - PVGadgets_AnimGadgetStop..... 66
 - PVGadgets_FreeAnimGadget..... 66
 - PVGadgets_FreeAnimGadgetImages..... 67
 - BubbleTipGadget..... 67
 - PVGadgets_BubbleTip..... 67
 - PVGadgets_BubbleTipText..... 67
 - PVGadgets_BubbleTipTitle..... 67
 - PVGadgets_BubbleTipColor..... 68
 - PVGadgets_BubbleTipDelete..... 68
 - CanvasGadget..... 68
 - PVGadgets_Canvas..... 68

PVGadgets_CanvasFree.....	69
PVGadgets_CanvasButtonAdd.....	69
PVGadgets_CanvasButtonHover.....	70
PVGadgets_CanvasButtonPressed.....	71
PVGadgets_CanvasButtonDelete.....	72
PVGadgets_CanvasEnableClick.....	72
PVGadgets_CanvasDisableClick.....	73
ProgressBarGadget.....	73
PVGadgets_ProgressBar.....	73
PVGadgets_FreeProgressBar.....	74
PieChartGadget.....	74
PVGadgets_PieChart.....	74
PVGadgets_FreePieChart.....	75
Refresh.....	75
PVGadgets_Refresh.....	75
PVGadgets_RefreshWindow.....	76
Misc Functions.....	76
Window Functions.....	76
PVGadgets_WindowFreeze.....	76
PVGadgets_WindowUnfreeze.....	77
PVGadgets_ToolWindow.....	77
PVGadgets_WindowTransparent.....	77
PVGadgets_WindowTop.....	77
PVGadgets_WindowReset.....	77
Gadget Functions.....	77
PVGadgets_Splitter.....	77
Path Functions.....	78
PVGadgets_CurDir.....	78
PVGadgets_TempDir.....	78
PVGadgets_ExeDir.....	78
ListIcon Functions.....	78
PVGadgets_CountListIconColumns.....	78
PVGadgets_GetListIconColumnWidth.....	79
PVGadgets_SetListIconColumnWidth.....	80
PVGadgets_JustifyListIconColumn.....	80
PVGadgets_LastListIconRow.....	81
PVGadgets_SelectListIconRow.....	82
PVGadgets_CreateListIconImageList.....	83
PVGadgets_AddListIconImageList.....	83
PVGadgets_FreeListIconImageList.....	84
PVGadgets_ChangeListIconImage.....	85
PVGadgets_ListIconTitle.....	86
Format Functions.....	87
PVGadgets_FormatNumber.....	87
Mutex Functions.....	88
PVGadgets_StartOnce.....	88
Error Functions.....	88
PVGadgets_APIError.....	88
PureSkin.....	88
PureSkin.....	88
PureSkinMem.....	89
PureSkinHide.....	90
PurePoint.....	91
PurePoint.....	92

PurePointMem.....	92
UsePurePoint.....	93
SysPurePoint.....	94
FreePurePoint.....	94
ResetPurePoint.....	95
PurePointChild.....	96
PurePointX.....	96
PurePointY.....	97
PurePointXX.....	98
PurePointYY.....	99
2 PVGadgets QuickStart	100
Part III PVPlugins User Library	103
1 PV_PluginsInfo	103
2 PV_PluginsWindowName	104
3 PV_PluginsWindowTitle	105
4 PV_PluginsWindowX	106
5 PV_PluginsWindowY	107
6 PV_PluginsWindowW	108
7 PV_PluginsWindowH	109
8 PV_PluginsWindowBGColor	110
9 PV_PluginsWindowTab	111
10 PV_PluginsGadgetType	112
11 PV_PluginsGadgetName	113
12 PV_PluginsGadgetText	114
13 PV_PluginsGadgetX	115
14 PV_PluginsGadgetY	116
15 PV_PluginsGadgetW	117
16 PV_PluginsGadgetH	118
17 PV_PluginsGadgetTabs	119
18 PV_PluginsMenuItemName	120
19 PV_PluginsMenuItemText	121
20 PV_PluginsMenuItemPos	122
21 PV_Plugins	123
Part IV PureBasic Integration	126
Part V SpiderBasic Export Plugin	133
Part VI History	138
Index	149

Foreword

PVGadgets is a collection of addon functions designed specifically to work with PureVision and the PureBasic Compiler.

Thank you for supporting PureVision and we hope that this Users Guide will help you to make the most of our Product.

Part



1 PureVision

The Professional Form Designer for PureBasic

PureVision provides you with a quick and easy way to create and manage forms while designing professional commercial applications.

Not only can you manage entire projects, PureVision allows you to export individual Forms from large projects and merge them with other projects. By reusing existing forms, this saves you both time and money.

PureVision comes complete with tools for giving your projects that "competitive edge". Add icons to MenuBars, create custom Skins for your Forms, design custom Pointers, manage Language Files for multilingual support and much more !!!

Have even more power at your fingertips by accessing PVGadgets, a custom library designed specifically for those wanting more out of their application.

PureVision is the choice of professionals and is used to design and create thousands of dollars worth of commercial software on a daily basis.

When you have PureVision, you have Pure Power !!

Features:

- Easy To Use Interface
- Design Complex Professional Forms Quickly
- Graphically Enhance Forms with PureSkins
- Group Gadgets for Fast Moving & Sizing
- Copy & Paste Gadgets from Clipboard
- Easily Adjustable Tabbing Order
- "Hot Keys" for Quick Gadget Manipulation
- Load/Save Full Work Projects
- Edit Multiple Forms in a Single Project
- Save Individual Forms to New Project
- Merge Individual Forms or Complete Projects
- Auto Include Images in Compiled EXE
- Create Dynamically Resizing Forms
- Export as "Easy To Use" PureBasic Source Code
- Select Windows or Linux Compliant Code
- Exports Full Project or Selected Forms Only
- Exports Multi Language Support File
- Export UTF-8 Unicode Support
- Includes PureTrans - Multi Language File Manager
- Includes PVGadgets - Custom Gadgets Library
- Supports PureBasic v6.xx Gadgets
- Supports PureBasic v6.xx Gadget Flags
- Allows for Additional (non PB) Flags
- Features Additional Windows API Gadgets
- No Limit to Number of Forms in a Project
- Number of Gadgets per Form is Unlimited
- Work Form Looks & Responds as it was in Your Final Application (WYSIWYG Interface)

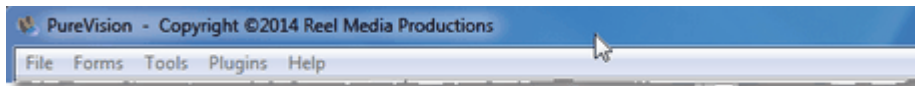
- DPI Aware / 64bit Application

Support:

- Email: support@reelmedia.org
- Web: <https://reelmedia.org/purevision>

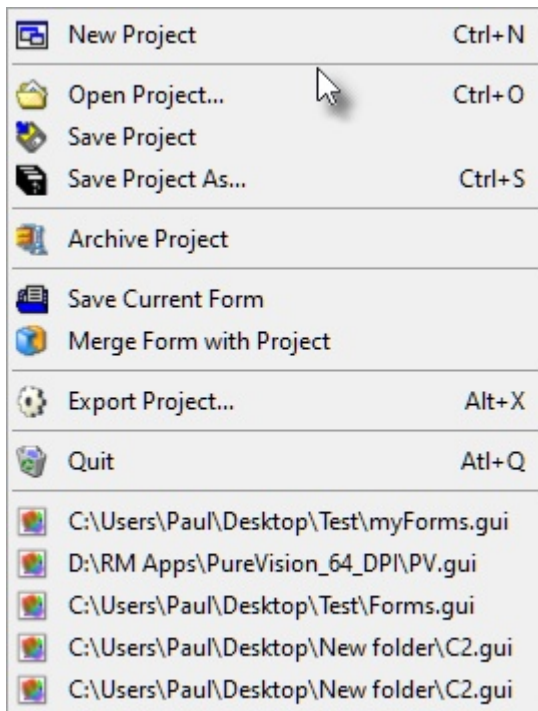
1.1 MenuBar

The PureVision MenuBar is used for various file operations, program settings, and accessing extra tools.



To use the MenuBar, simply select the appropriate heading and then select the required action.

1.1.1 File



The **File** menu allows you to perform various file related operations in PureVision.

1.1.1.1 New Project

Selecting **New Project** will delete any currently loaded project and start a new project, beginning with a new Form.

1.1.1.2 Open Project

Selecting **Open Project** will allow you to load a PureVision (GUI) project into PureVision.

You will be asked to select the appropriate GUI file using a file selector.

You can also open a PureVision GUI file by double clicking on a GUI file in Windows Explorer if PureVision has not yet been started.

1.1.1.3 Save Project

Selecting **Save Project** will save the current PureVision project in the GUI format.

The current filename will be used when saving. If a filename does not yet exist, you will be prompted to enter one.

1.1.1.4 Save Project As

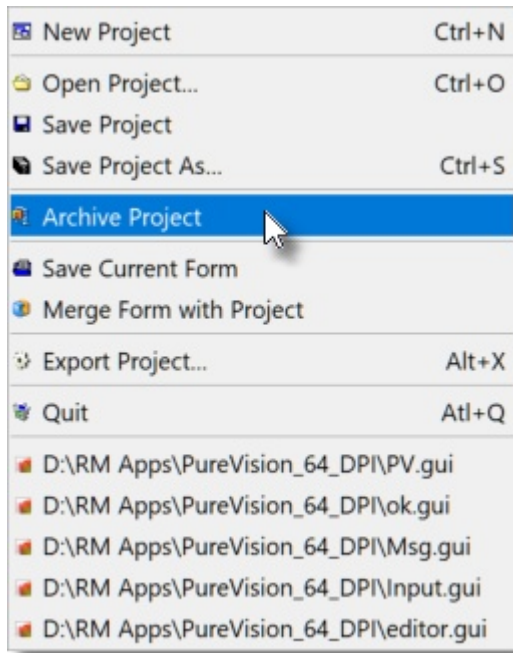
Selecting **Save Project As** will save the current PureVision project in the GUI format by asking for a filename and location, regardless of what filename has currently been set.

1.1.1.5 Archive Project

Selecting Archive Project will compress your project file along with any resources used by your project into a single ZIP file.

You will first be asked to save the current PureVision project in the GUI format by selecting filename and location.

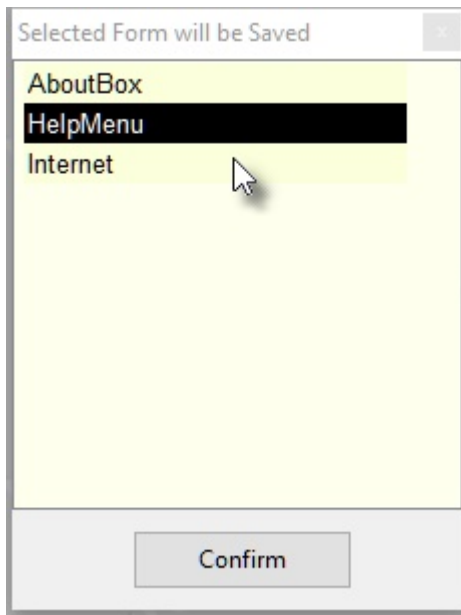
The resulting ZIP file will be located in that same folder



1.1.1.6 Save Current Form

When selecting **Save Current Form**, instead of saving the entire PureVision project, only the Form that is currently being displayed will be saved in the PureVision GUI project format.

You can Save multiple individual Forms by holding CTRL key while selecting other Forms from the list.

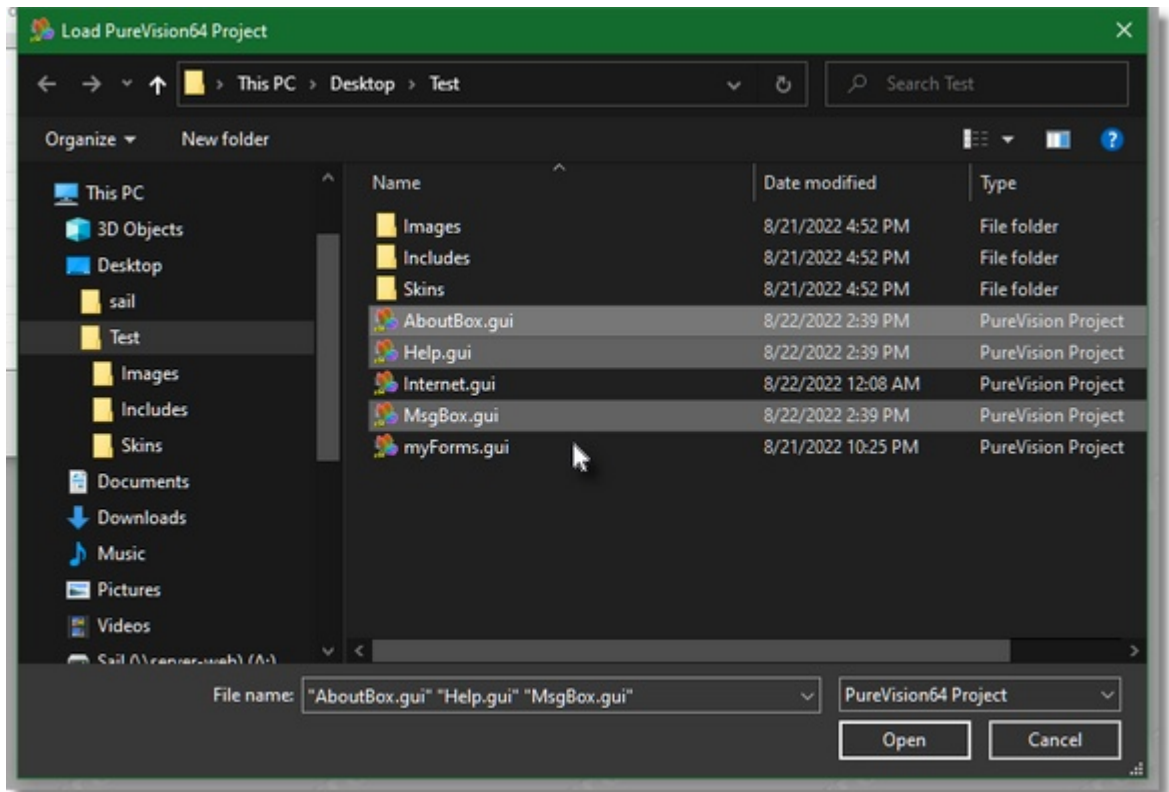


1.1.1.7 Merge Form with Project

Selecting **Merge Form with Project** allows you to combine Forms from other projects into the project you are currently working on.

For example, if you have created an "About Box" in a previous project and wish to reuse it without having to rebuild the entire Form. Simply export the "About Box" from the old project and use **Merge Form with Project** to merge it into your new project.

You can Merge multiple Forms into your Project by holding the CTRL key while selecting other Forms in the File Selector.

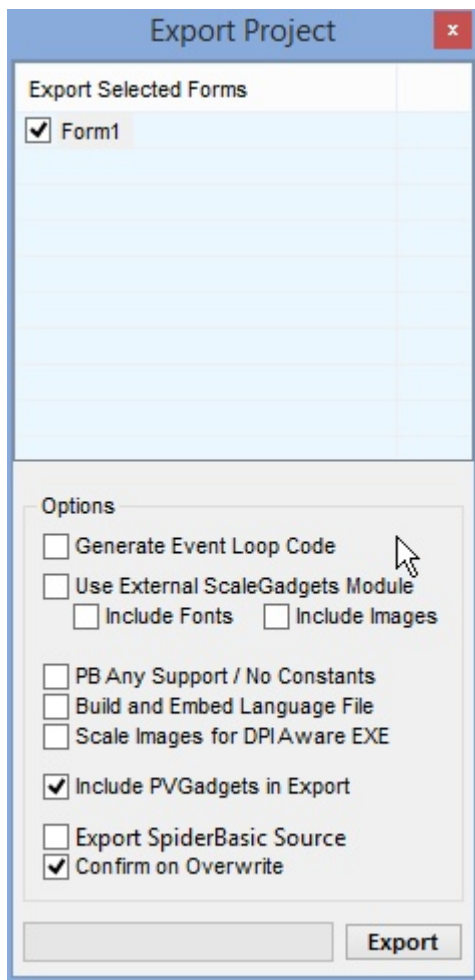


1.1.1.8 Export Project

To have PureVision convert your Project into Source Code which will compile using the PureBasic Compiler, select **Export Project** or press **ALT+X**.

PureVision will generate Source Code for all Forms in the Export List which contain a check. If your Project contains Forms you do not wish to have exported, simply remove the check from any Form you do not want exported.

By default, the first Form in the Export List will be selected. The first Form is always considered the Main Form and an Event Loop will be generated for this Form if **Generate Event Loop Code** has been checked. To have PureVision generate an Event Loop for a different Form, simply select another Form in the Export List.



By default PureVision exports code using Constants. If you prefer using Global variables assigned with #PB_Any, you can enable this export option by checking "PB Any Support/No Constants".

If you would like to use "Gadget Scaling" instead of "Dynamic Resizing" you select "Use External ScaleGadgets Module" to use the ScaleGadgets Module that has been made available by mk-soft.

If you would like to export your Project with support for multiple Languages, check "Build and Embed Language File". This will embed the current Project Language as well as export a Language File (.Lang) which you can easily edit using the [PureTrans Language Manager](#).

If you plan on Compiling your Project as a DPI Aware EXE, make sure you Export your Project with the "Scale Images for DPI Aware EXE" so all images in your Project are scaled accordingly.

If you have set up your Forms to use Dynamic Resizing or you would like to use any of the features available in the PVGadgets Module, make sure "Include PVGadgets in Export" is checked so the PVGadgets Module is included in your compiled project.

If the SpiderBasic Export Module is available, you have the option to export your code to use with SpiderBasic.

If an Exported Project or an Event Loop for an Exported Project already exists, you will be prompted to "Confirm Overwrite". This is a safety feature giving you a last chance before overwriting any existing data. If you DO NOT wish to be prompted every time PureVision is about to overwrite existing data, you can turn this warning feature off by removing the check in the "Confirm on Overwrite" check box.

By default PureVision will add a header to Exported Code containing the Version of PureVision used, the Date Exported and the Registered User of the software.

```
;/ Created with PureVision64 v5.73 x64  
;/ Mon, 19 Apr 2021 08:37:28  
;/ by Reel Media Productions
```

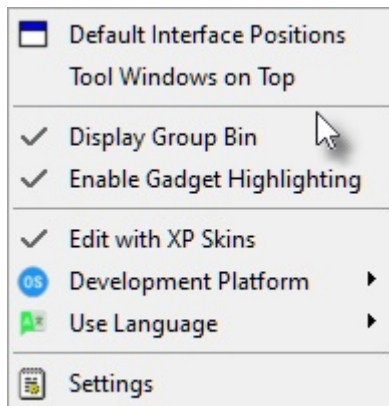
You can remove the Registered User portion of the Exported Code by selecting "Remove Reg Info from Header" so only the Version and Date are added.

```
;/ Created with PureVision64 v5.73 x64  
;/ Mon, 19 Apr 2021 08:37:28
```

1.1.1.9 Quit

Selecting **Quit** will close the PureVision application.

1.1.2 Forms



The **Forms** menu allows you to customize various Form settings and PureVision options.

1.1.2.1 Default Interface Positions

Default Interface Positions places all Forms and Tool Windows back to their original default locations.

Shortcut key is **HOME**

1.1.2.2 Tool Windows on Top

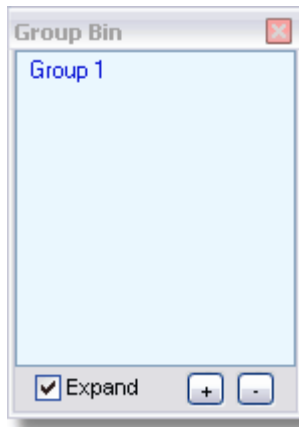
By default only the **Gadget Creator** tool window is on top of all other windows. Selecting the **Tool Windows on Top** option forces the other tool windows on top as well.

Shortcut key is **CTRL-PageUP**

1.1.2.3 Display GroupBin

The **GroupBin** allows you to easily arrange and manage groups of Gadgets. By grouping Gadgets together you can quickly select, move, size, hide, copy, paste and duplicate them.

Select **Display GroupBin** to display the GroupBin tool window.

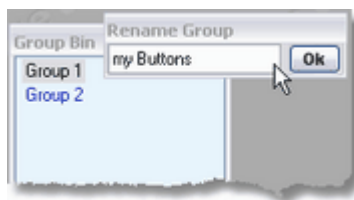


1.1.2.3.1 Using the GroupBin

To create new groups in the GroupBin, press the **+** button.



To give meaningful names to the various Groups, right mouse click on the Group you wish to name and select **Rename Group**. Enter the new name of the Group and press **Ok**.



To add Gadgets to a Group, first select the Group you wish to add to. Then put a check in front of the Gadgets you wish to add using the [Gadget Tree](#).



Finally, **right mouse click** on the GroupBin and select **Add Group to GroupBin**. The selected Gadgets will be added to the Group and can now quickly be re-selected by simply clicking on the name of the Group they belong to.



To remove a Gadget from a Group, select the Gadget you want to remove and press the - button.

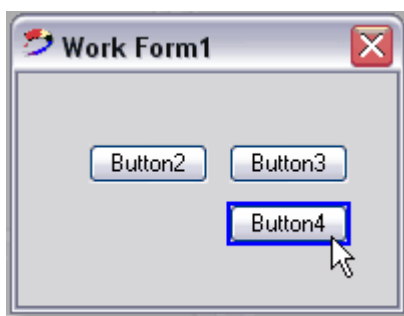
1.1.2.4 Display Custom Gadgets

This option will Display or Hide the [Custom Gadgets](#) ToolBar.

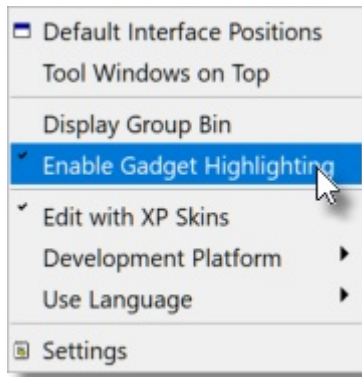


1.1.2.5 Enable Gadget Highlighting

When Gadget Highlighting is enabled, the Gadget you are currently editing will have a colored box around it. A green box indicates you are in resize mode and a blue box indicates you are in move mode.



Select **Enable Gadget Highlighting** to toggle this option on or off.

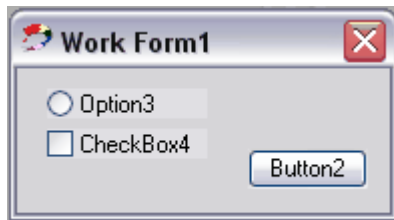


1.1.2.6 Edit with Skins

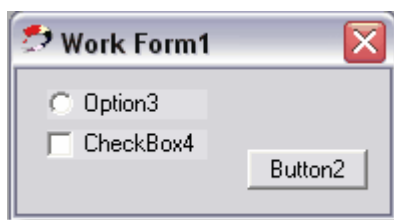
If you are running PureVision on Windows XP and higher, the Gadgets you create will inherit any Windows XP skinning features you have turned on. To see what the Gadgets would look like on an operating system that does not support skins (Window 2000/98/95) select **Edit with Skins** to disable skins.

This option requires you to restart PureVision.

Form with **Edit with Skins** enabled:



Form with **Edit with XP Skins** disabled:



If you are running PureVision on an operating system other than Windows XP and higher, your forms will look similar to "disabled" even if **Edit with Skins** is enabled.

1.1.2.7 Development Platform

The Development Platform option affects the source code that is exported by PureVision. If PureBasic Linux is selected, all Windows specific code will be removed including PVGadgets functions/features.

1.1.2.8 Use Language

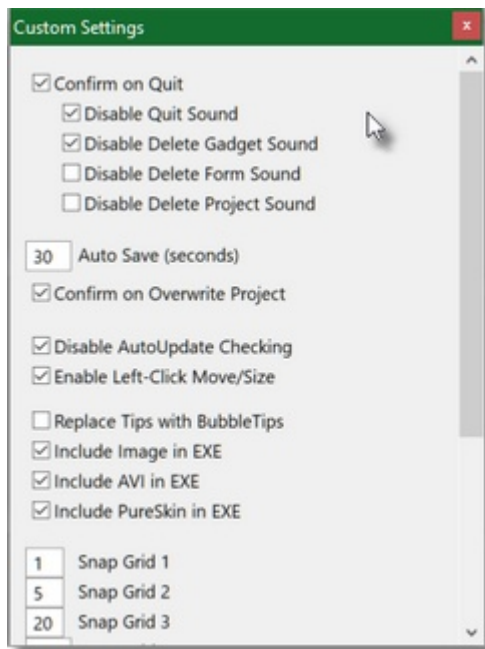
By default all PureVision text is in English.

When additional (.Lang) language files are available you can download and place them in the main PureVision folder. If they are valid Language Files they will automatically appear under Use Language. This will allow you to select the available language of your choice to better understand the various text within PureVision.

You can also use [PureTrans](#) (included with PureVision) to create your own language file if one does not currently exist.

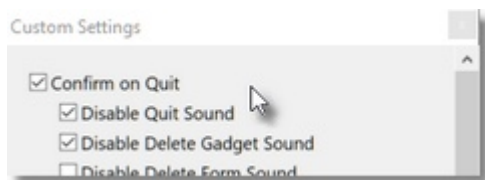
1.1.2.9 Settings

The Settings Menu allows you to customize various settings within PureVision to suit your personal preferences.



1.1.2.9.1 Confirm on Quit

By enabling **Confirm on Quit**, PureVision will ask "are sure you wish to quit" before actually closing the application. This option is enabled by default and is used as a security measure.

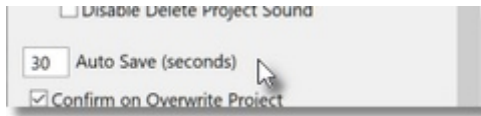


If **Confirm on Quit** is disabled, PureVision will close instantly when you tell it to exit, regardless if any project data exists.

1.1.2.9.2 Auto Save

When Auto Save/Crash Protection is enabled, PureVision will take a snapshot of the current Project at a set interval.

If PureVision does not shut down properly you will have the option to restore this snapshot when PureVision is restarted.



Setting this option to 0 (zero) will disable Auto Save.

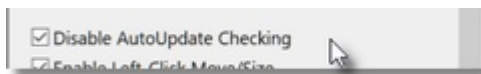
Setting this number to 30 will tell PureVision to save every 30 seconds and a save will only occur if a change has been made.

NOTE: Enabling this option may impact performance due to disk writes at set intervals.

1.1.2.9.3 Disable AutoUpdate Checking

Each time you run the software, PureVision will automatically connect to the Internet and check to make sure you are running the latest version.

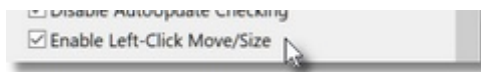
If you would like to disable this feature, simply put a check mark beside "Disable AutoUpdate Checking".



You can enable this feature at any time or you can manually check for updates by selecting "[Check for Online Updates](#)".

1.1.2.9.4 Enable Left-Click Move/Size

Normally the Moving and Sizing of Gadgets is handled by clicking the Right Mouse Button. If you would prefer instead to click the Left Mouse Button, you can enable this feature by selecting the "Enable Left-Click Move/Size" option.



Note: By enabling this option you may lose the ability to interact with some of the Gadgets placed on the Form.

1.1.2.9.5 Replace with BubbleTips

When this option is disabled, standard ToolTips are displayed when the mouse pointer hovers over a Gadget that contains a Tip.

Enabling this option forces the tips to be displayed in a fancy Bubble instead of the standard box.

Example 1: BubbleTip enabled



Example 2: standard ToolTip



1.1.2.9.6 Include Images in EXE

When this option is enabled, all images/icons used in your project will be included in your EXE when the exported source code is compiled.

PureVision has an Image Optimizing routine which prevents duplicate images from being included many times and bloating your application.

If this option is disabled, all images will be loaded from your applications sub folder called "Images".

1.1.2.9.7 Include PureSkin in EXE

When this option is enabled, any PureSkin that has been applied to a Form will be included in the EXE when the exported source code is compiled.

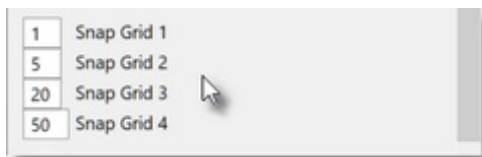
PureSkins that are not included in the EXE are stored in a sub folder of your application called "\Skins"

1.1.2.9.8 Snap to Grid

By default the Grid spacing on a Form is set to 1, 5, 20 & 50 and can be adjusted using the slider at the bottom of the Gadget Creator Panel.



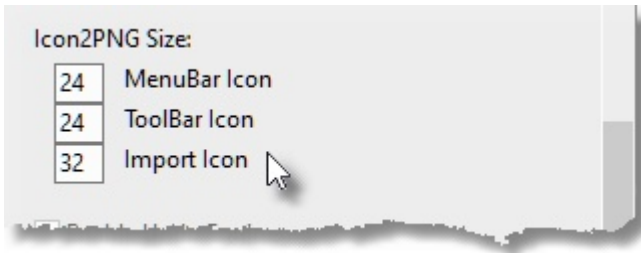
You can change these defaults values in the Settings Menu to better suit your preferences.



1.1.2.9.9 Icon2PNG Size

When an icon (.ico) file is selected, PureVision will give you the option to convert it to a PNG. This will allow for more flexibility when working on Projects to be used with SpiderBasic since SpiderBasic cannot use icon (.ico) files. It will also allow for PureBasic MenuBar, ToolBar and various Gadgets containing images to have proper image resizing on 4K (high DPI) monitors.

The default conversion size can be specified for individually for MenuBar, ToolBar and Gadgets.



1.1.2.9.10 Declare Forms on Export

When this option is selected, a Declare statement is created for each Window that is Exported.

Selected:

```

43
44 Declare.i Window_Form1(X=0,Y=0,W=400,H=300,Fla
45
46 Procedure.i Window_Form1(X=0,Y=0,W=400,H=300
47   If OpenWindow(#Window_Form1,X,Y,W,H,"Wor
48     HideWindow(#Window_Form1,#False)
49   ProcedureReturn WindowID(#w
50   EndIf

```

Not Selected:

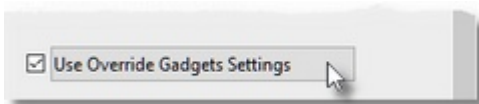
```

42
43 Procedure.i Window_Form1()
44   If OpenWindow(#Window_Form1,0,0,400,300,"
45     HideWindow(#Window_Form1,#False)
46   ProcedureReturn WindowID(#w
47   EndIf
48 EndProcedure

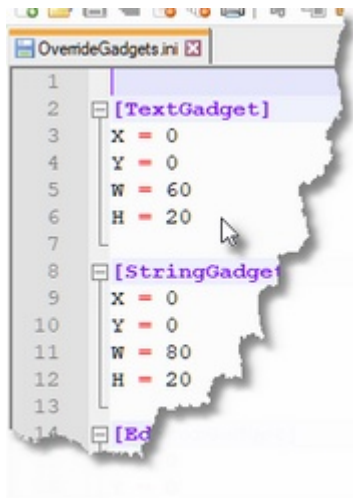
```

1.1.2.9.11 Override Gadget Settings

When Override Gadget Settings is enabled, you will have the ability to modify the Default X,Y,W,H of each Gadget and override the Default settings. By disabling this option the Default Gadget Settings will be used.



When the **"Use Override Gadget Settings"** button is pressed, a special "OverrideGadgets.ini" file will be opened in your default Text Editor software.

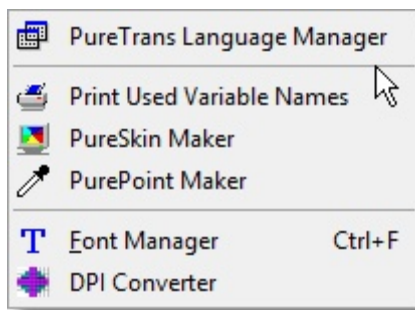


Example:

When a TextGadget is placed on a Form the default x,y position is 0,0 and the width/height is 60x15. If you prefer the height to be 20, you can edit the TextGadget default height and change H=15 to H=20. Now every time you place a TextGadget on a Form the width/height will be 60x20.

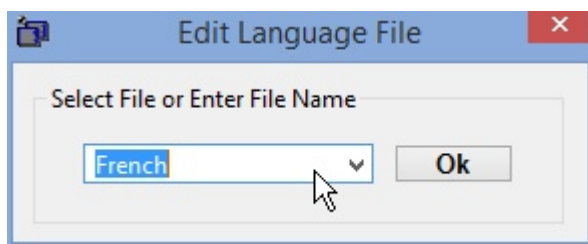
1.1.3 Tools

The **Tools** menu provides access to addons which help create and enhance Forms within your project.



1.1.3.1 PureTrans Language Manager

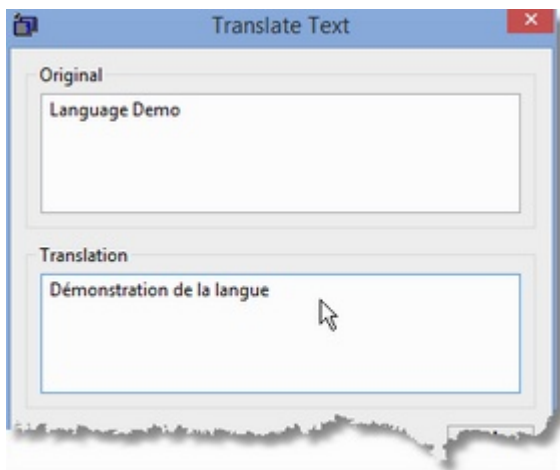
When starting the **PureTrans Language Manager** you will be asked to locate the Folder containing the language file to be translated. Next you will be asked to either select an existing language file you have already started to convert or enter the name of a language file you wish to start working on.



After selecting or entering a language file name, a form will appear with the text for all Gadgets. The original language will be located on the left and the translated language will appear on the right.

Language ID	Original	Translated
Window_Main	Language Demo	
Gadget_Main_Frame3...	Welcome	
Gadget_Main_Info	PureVision allows you to create a default interfa...	
Gadget_Main_TryAgain	Try Again	
Window_Setup	Setup	
Gadget_Setup_Frame...	Select Language	
Gadget_Setup_Ian1	English	
Gadget_Setup_Ian2	French	
Gadget_Setup_Ian3	German	
Gadget_Setup_Ok	Ok	
Gadget_Setup_Ok_To...	Select Language then click here	

Double left click on the text you wish to translate. You will be prompted to enter the new text. Enter the new translated text and press the **Ok** button.



Once your list of text has been translated, press the **Save** button to save all changes.

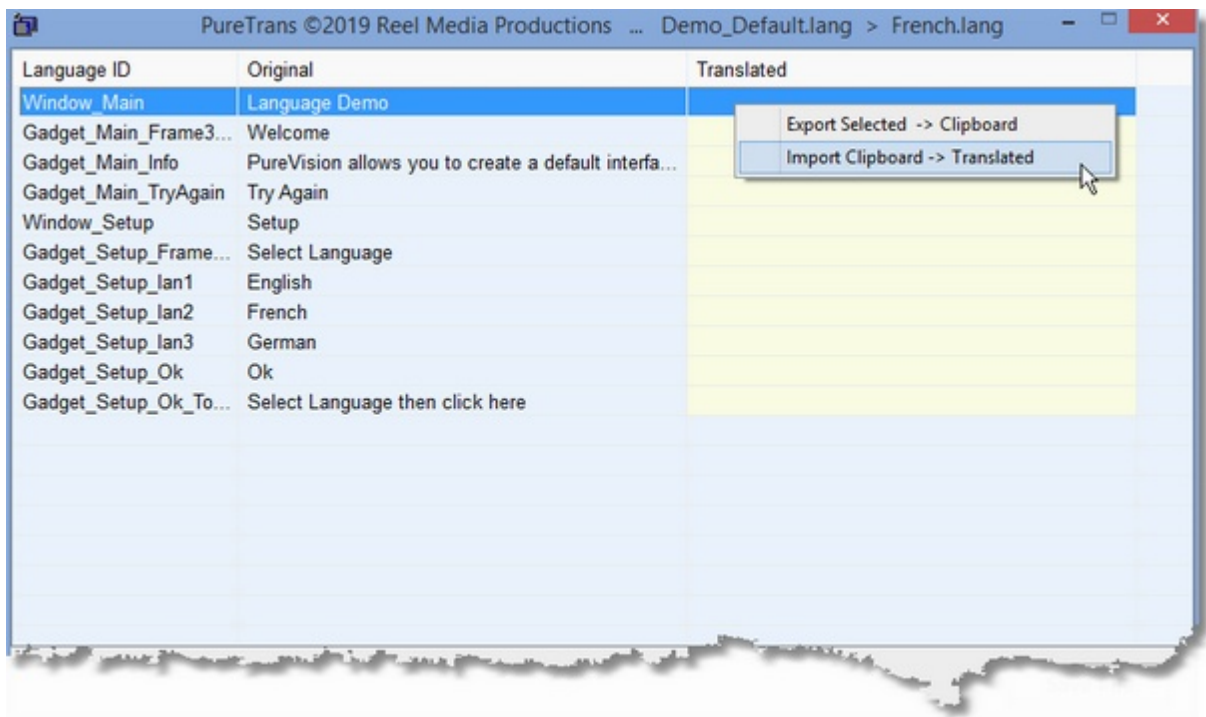
Language ID	Original	Translated
Window_Main	Language Demo	Démonstration de la langue
Gadget_Main_Frame3...	Welcome	
Gadget_Main_Info	PureVision allows you to create a default interfa...	
Gadget_Main_TryAgain	Try Again	
Window_Setup	Setup	
Gadget_Setup_Frame...	Select Language	
Gadget_Setup_lan1	English	
Gadget_Setup_lan2	French	
Gadget_Setup_lan3	German	
Gadget_Setup_Ok	Ok	
Gadget_Setup_Ok_To...	Select Language then click here	

You can also select all the items you wish to translate and then right click to export them to the Clipboard. This will allow you to paste into an online translation service like [DeepL Translator](#) and then copy the translated text back to your clipboard.

Language ID	Original	Translated
Window_Main	Language Demo	
Gadget_Main_Frame3...	Welcome	
Gadget_Main_Info	PureVision allows you to create a default interfa...	
Gadget_Main_TryAgain	Try Again	
Window_Setup	Setup	
Gadget_Setup_Frame...	Select Language	
Gadget_Setup_lan1	English	
Gadget_Setup_lan2	French	
Gadget_Setup_lan3	German	
Gadget_Setup_Ok	Ok	
Gadget_Setup_Ok_To...	Select Language then click here	

Export Selected -> Clipboard
 Import Clipboard -> Translated

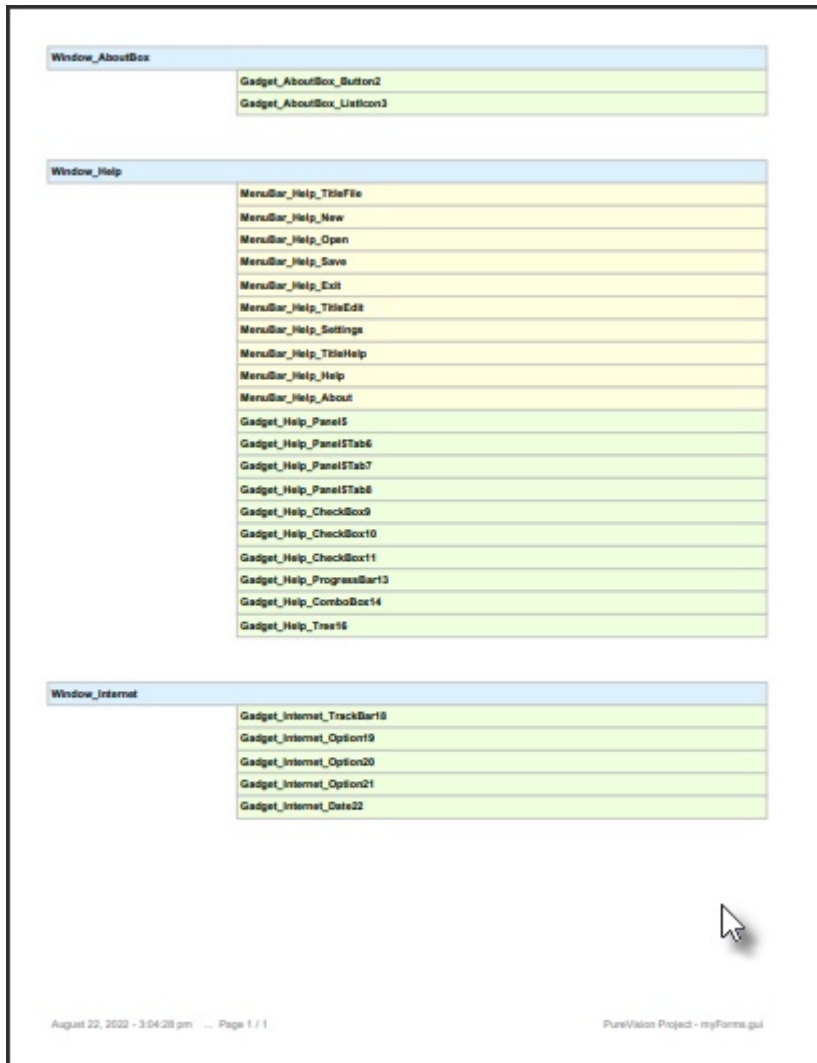
After the translated text is back in your clipboard, you can select where you want to start inserting the text back into PureTrans and then right click to Import the Clipboard.



Note: *The **PureTrans Language Manager** comes as a separate module than can be run without having PureVision. This allows you to give a copy of the **PureTrans Language Manager** (along with your language file) to the person or group who is doing the translating for you and they simply give back the translated file that has been saved.*

1.1.3.2 Print Used Variable Names

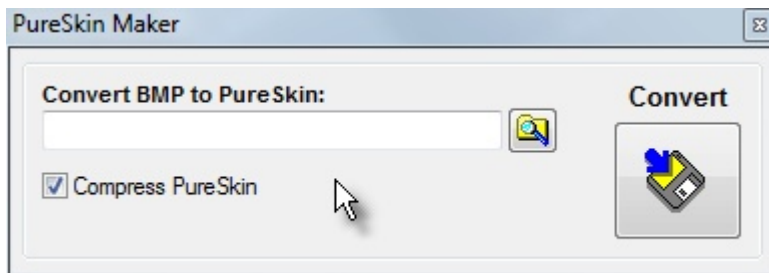
Print Used Variable Names gives you a quick overview of all the Constant names you have used in your Project. By printing this information, you will have the ability to quickly and easily look up what names you have used when you are manually writing code.



1.1.3.3 PureSkin Maker

PureSkin Maker allows you to quickly and easily convert a standard bitmap (BMP) into a PureSkin (PVS). A PureSkin is a very small and fast loading file compared to a standard bitmap and is applied to a PureVision Form through a [PureSkin Gadget](#).

To create a **PureSkin** simply select PureSkin Maker from the Tools menu, locate the bitmap file you wish to convert to a PureSkin, then press the Convert button.



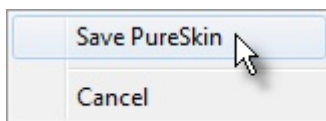
PureSkin Maker will display the selected bitmap. Use the color picker icon to select the color of the bitmap you wish to be transparent. (This example image is a 357KB bitmap)



PureSkin Maker will display the bitmap with selected color removed.



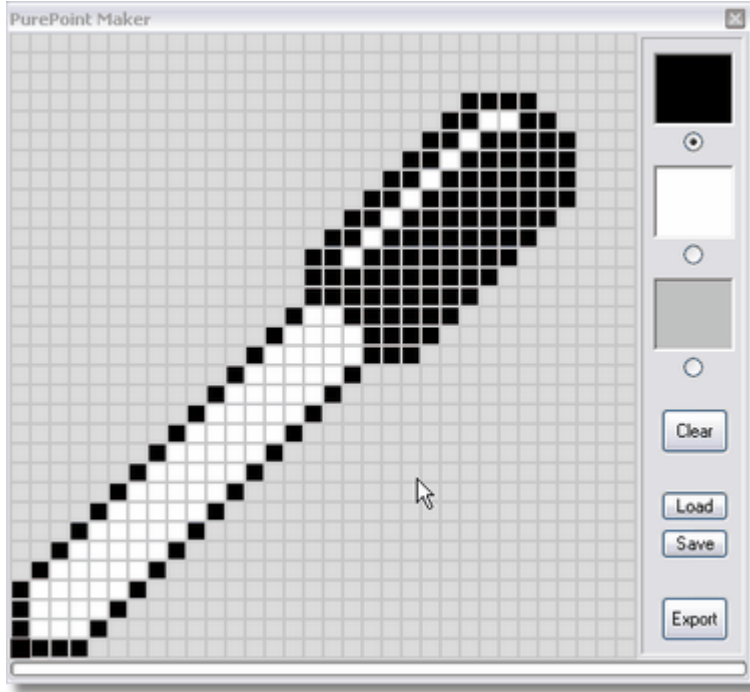
If you are pleased with the results, right click on the skin and select "Save PureSkin". The skin will be saved to the PureSkin (PVS) format. (This example skin saves to a 33KB PureSkin file using file compression)



NOTE: Unless specified, the converted PureSkin (PVS) file has a DPI scaling of 100% and Forms

compiled with PureSkins must have "DPI aware executable" flag turned off.
The "DPI aware executable" flag can be set if a PureSkin has been created for the DPI scaling being used.

1.1.3.4 PurePoint Maker



PurePoint Maker is used to create PurePoint (.pvp) files for use with the PurePoint User Library, allowing you to easily change your Mouse Pointer.

Use the mouse to select Black/White/Transparent and then draw on the PurePoint Grid by pressing the Left Mouse Button.

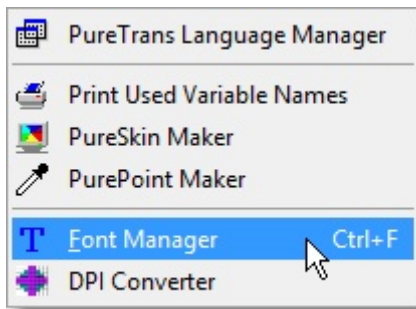
Press the Right Mouse Button to set a click point. This will determine the active part of the PurePoint and will be outlined in red.

- Clear** Clears the entire Grid with selected color
- Load** Loads a Raw PureSkin File (.raw)
- Save** Saves a Raw PureSkin File (.raw)
- Export** Exports a PureSkin to be used with PureSkin User Library (.pvp)

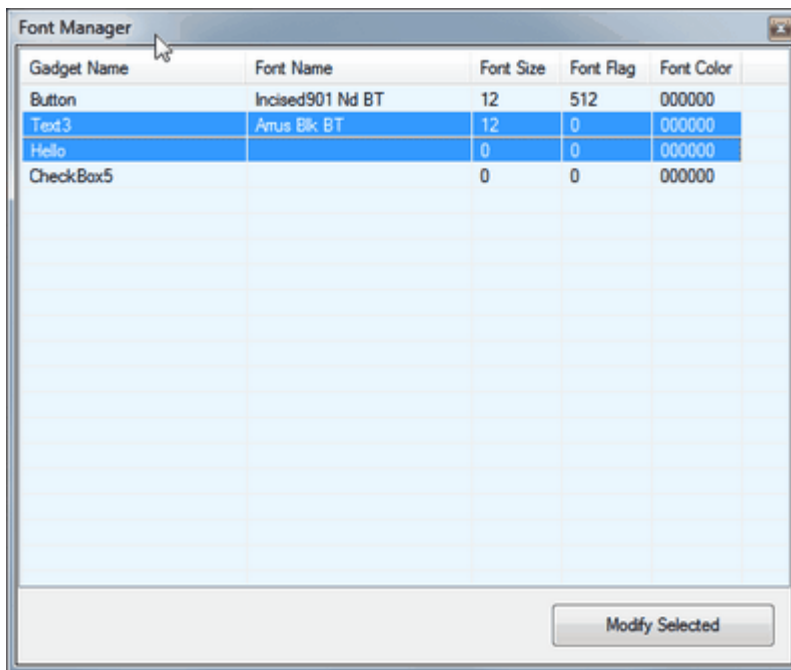
PurePoints are added to your application using special [PurePoint](#) functions.

1.1.3.5 Font Manager

Font Manager allows you so quickly and easily modify Font attributes of multiple Gadgets.



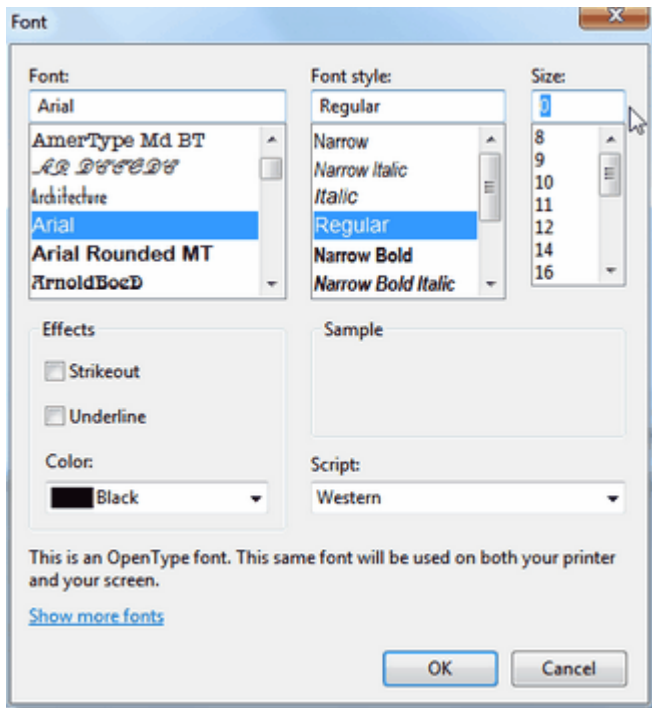
The Font Manager will display a list of all Gadgets which allow for Font Attributes to be Modified. Select the Gadgets you wish to modify by using the Left Mouse Button. Multiple Gadgets may be selected by combining the Left Mouse Button with the CTRL or SHIFT keys.



Press the **Modify Selected** button to bring up the Font Requester. Select the Attributes you wish to modify and press OK to apply the changes.

New attributes will be applied to ALL selected Gadgets.

If you would like the selected Gadgets to revert to **Windows Default Settings**, set the **Font Size** to 0 (zero).

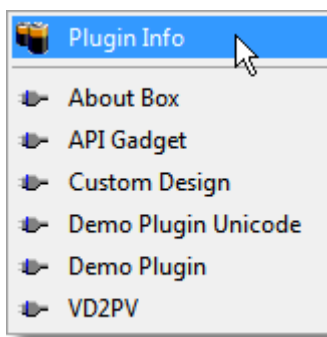


1.1.4 Plugins

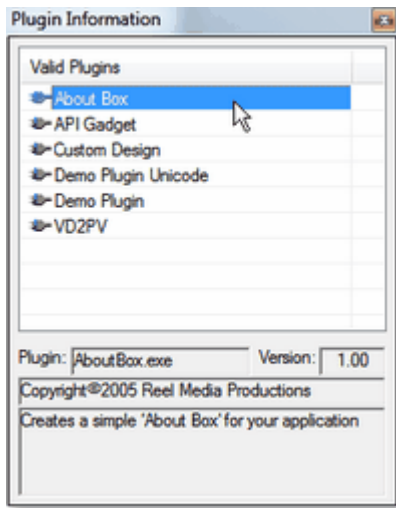
PureVision provides a simple way to design your own Plugins through a special PureBasic User Library called PV_Plugins.

When a Plug in is designed and placed in the **PureVision/Plugins** folder, it can be accessed from the Plugins option in the MenuBar.

(The first 10 Plugins located in the **PureVision\Plugins** folder will appear in the Menu List)



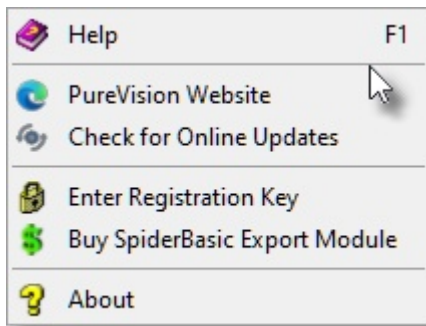
By selecting Plugin Info from the drop down menu, you can view information specific to that Plugin.



To run a specific Plugin, select it from the drop down menu by pressing the left mouse button.

For information on how to use the PV_Plugins User Library, see the [PVPlugins User Library](#) section.

1.1.5 Help



The **Help** menu gives you access to various help and support resources as well as application information.

1.1.5.1 Help

Selecting **Help** opens the windows help file containing all the information you are reading here. For quick reference, you can open the help file at any time while using PureVision.

1.1.5.2 PureVision Website

Selecting **PureVision Website** will open your default web browser and take you to the official PureVision website.

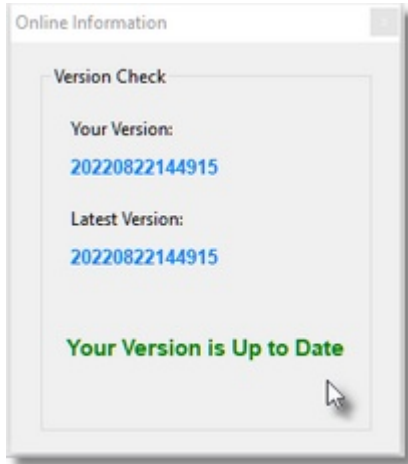
The PureVision website contains the latest software updates as well as a support forum for extra help.

<https://www.reelmedia.org/purevision>

1.1.5.3 Check for Online Updates

Check for Online Updates will connect to the PureVision Version Server and compares your version of PureVision to the newest version available for download. This gives you a quick and easy way to

ensure you are always using the latest and best software.



If you are using an old version of PureVision, you will be prompted to go to the PureVision website to download the latest version.

1.1.5.4 Enter Registration Key

The Demo Version of PureVision has certain features disabled until a full version is purchased. When a key is purchased, a personal registration key will be emailed to you. Select **Enter Registration Key** and enter your personal key code into the appropriate fields.

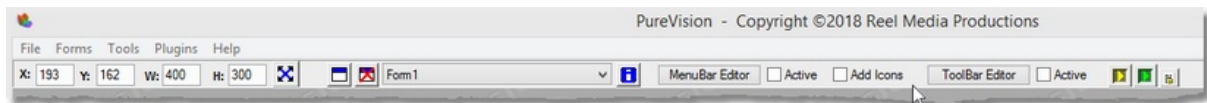


1.1.5.5 About

Selecting **About** will display the current PureVision version you are using along with personal license information.





1.2 Extended MenuBar



The **Extended MenuBar** is located right below the Main MenuBar.

The X,Y properties of the currently selected Form/Gadget are displayed along with Width and Height information.



The Move/Size toggle button allows you to change between Gadget Moving and Gadget Sizing on your Form. When you are in Moving mode, the button will display  and when you are in Sizing mode it will display .

For quick access when working on Forms, you can also press the **middle mouse button** to toggle between Move and Size. Once you are in Move or Size mode, you can use the **right mouse button** to Move or Size the selected Gadget. You can also use the arrow keys to Move or Size the selected Gadgets.



This button will add a new Form to your current project. If you have more than one Form in your project, use the Form Selector Drop Box to select the name of the Form you wish to view/edit.



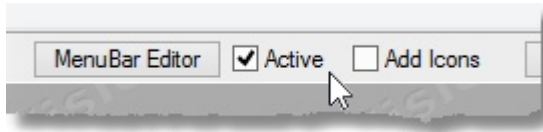
This button will delete the current selected Form from your project. Once a Form is deleted, all Gadgets contained on that Form will also be deleted.



While working on Gadgets you may wish to view or change the Form properties. Pressing this button will change to Form Properties mode. You can also press the **Home** key.

All Forms can contain a MenuBar. To build a MenuBar, press the [MenuBar Editor](#) button to start the

MenuBar Editor. Once a MenuBar has been created, check the **Active** box to display the MenuBar. If you create a MenuBar but later decide not to use it, simply uncheck the **Active** box and the MenuBar will not be exported with the source code.



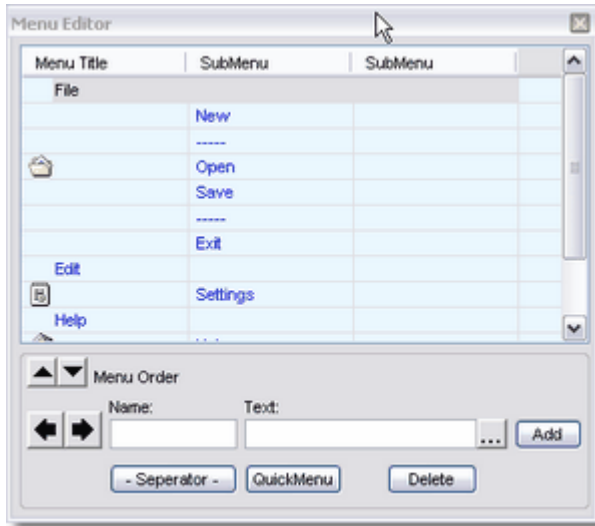
If you add icons to your MenuBar, you can enable them by checking the **Add Icons** button.

All Forms can also contain a Toolbar. To build a Toolbar, press the [ToolBar Editor](#) button to start the Toolbar Editor. After creating a Toolbar, check the Active box to activate and display the Toolbar. If you create a Toolbar but later decide not to use it, simply uncheck the **Active** box and the Toolbar will not be exported with the source code.



1.2.1 MenuBar Editor

The **MenuBar Editor** allows you to easily create MenuBars for your Forms. You can quickly create a standard MenuBar by pressing the **QuickMenu** button.



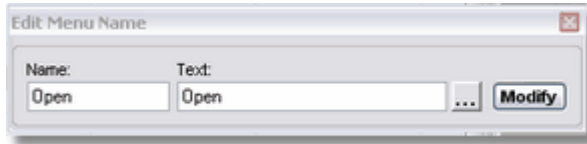
Enter the name of the MenuBar item you wish to create along with the text you wish to be displayed. If you would also like an icon displayed, press the ... button and select an icon. Press **Add** when you have entered all the appropriate information and you will see your item added to the Menu Title column.

If a current MenuBar item is highlighted when the **Add** button is pressed, the new item will be added after the highlighted item.

Use the **Up** and **Down** buttons to position the MenuBar item in the proper location.

Use the **Left** and **Right** buttons to make your MenuBar item a SubMenu item.

You can edit any MenuBar items by double left clicking on the appropriate row.

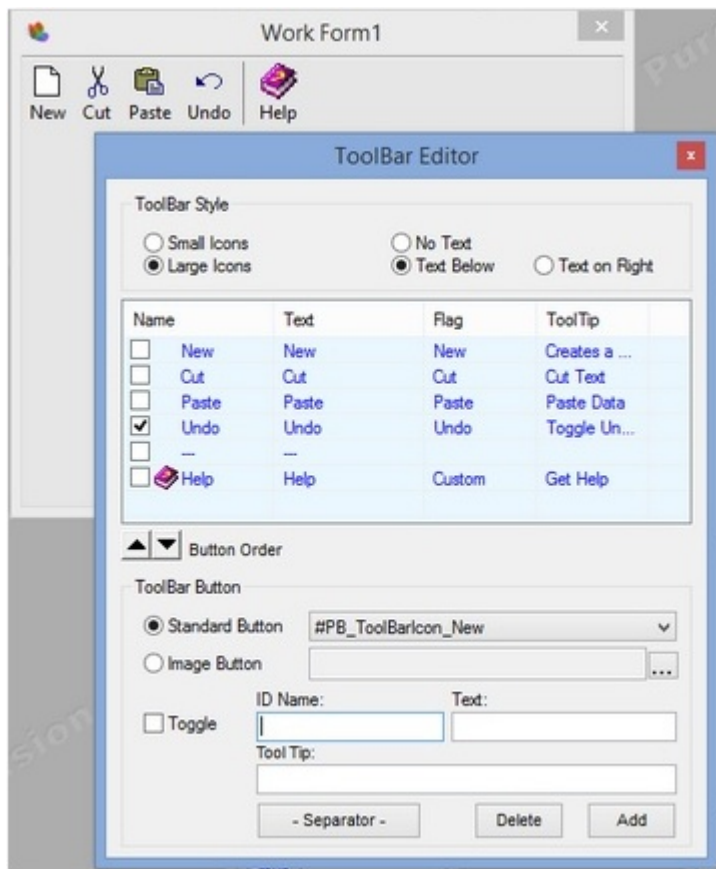


Make any required changes and press the **Modify** button to save the changes.

You can remove MenuBar items by selecting the appropriate row and pressing the **Delete** button.

1.2.2 ToolBar Editor

The **ToolBar Editor** allows you to easily create ToolBars for your Forms.



Button Order

The order of the ToolBar Buttons can be changed by selecting the Button and using either the Up and Down arrow buttons in the ToolBar Editor Interface or using the PGUP and PGDN keys on the keyboard.

Delete Button

To Delete a ToolBar Button, select the Button you wish to delete and either press the Delete Button in the ToolBar Editor Interface or press the DEL key on the keyboard.

Add Button

To Add a ToolBar Button, select the type of button you wish to add. This can be a Standard Button (pre-made) or an Image Button (custom image). If you wish to have the button toggle, check the Toggle option. After entering an ID Name and any other additional information, press the Add button to add the ToolBar Button to the list.

If you select an item in the ToolBar list, your new ToolBar Button will be added at that location and all items after your selected location will be moved down in the list.

Edit Button

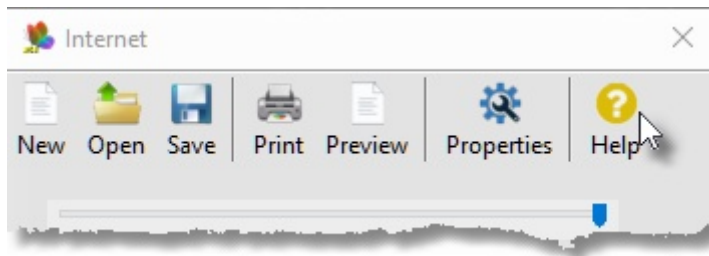
Editing an existing ToolBar Button can be performed by double-left clicking the appropriate Button in the List.

The Button will be removed from the List and the Button data will be moved down into the ToolBar Editor. You then have the option of making changes and putting the Button back into the List by pressing the Add Button or you can put the Button back into the List at a different location by selecting a location in the List and then pressing the Add Button.

(The Button will be added at the selected location and all other Buttons will be moved down the list from the selected location)

When you have finished adding your ToolBar Buttons, you can view them in your Form by closing the ToolBar Editor and activating the ToolBar from the PureVision main interface.

NOTE: Starting PureBasic v6.00 ToolBarStandardButton has been removed but PureVision's PVGadgets Module provides it's own custom images allowing you to continue using it.

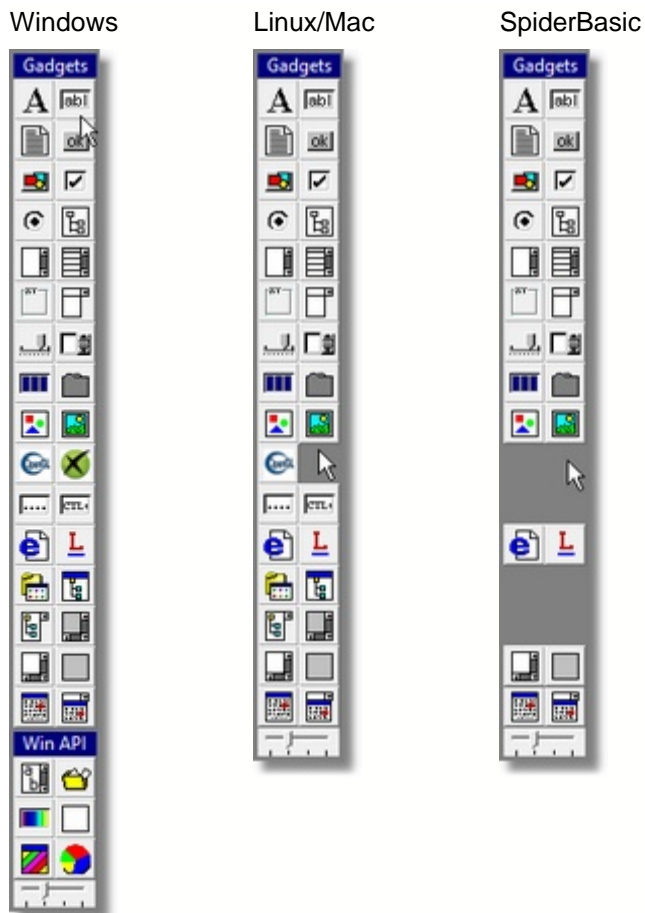


1.3 Gadget Creator

The Gadget Creator allows you to add Gadgets to your PureVision Forms. Simply left click on the Gadget you wish to add.

Native PureBasic Gadgets are located in the top half of the Gadget Creator window and API/PVGadgets are located in the bottom half.

The Gadget Creator window will look different depending on what Development Platform you have selected.



The grid size selector at the bottom of the Gadget Creator window allows you to select the grid size on your PureVision Form that Gadgets will "Auto Snap" to. The increments are 1, 5, 20, and 50 pixels.

1.3.1 PureBasic Gadgets



Creates a TextGadget.
Used for adding text labels to a Form.



Creates a StringGadget.
Used for allowing user to input data.



Creates an EditorGadget.
Used for inputting or displaying multiple lines of data.



Creates a ButtonGadget.
The button can contain text and triggers an event when pressed.



Creates a ButtonImageGadget.
This button can contain a graphic image and triggers an event when pressed.



Creates a CheckBoxGadget.
This gadget displays a check mark when enabled and an empty box when disabled.



Creates an OptionGadget.
These gadgets are placed in groups and are used when a single choice is needed from many options. Selecting one option in a group deselect's all other options.



Creates a TreeGadget.



Creates a ListViewGadget.



Creates a ListIconGadget.
The active part of a ListIconGadget is the title bar section. Right click this area to Move or Size this Gadget.



Creates a Frame3DGadget.



Creates a ComboBoxGadget.



Creates a TrackBarGadget.



Creates a SpinGadget.



Creates a ProgressBarGadget.



Creates a PanelGadget.

A PanelGadget contains Tabs which more Gadgets can be placed. Select the Tab in the GadgetTree to place Gadgets on that Tab.

Container type Gadgets (PanelGadget, ScrollAreaGadget, ContainerGadget) cannot be placed inside a PanelGadget.



Creates a CanvasGadget.



Creates an ImageGadget.



Places an OpenGL canvas on the current Form.



Places a DirectX screen on the current Form. A single project can only contain one instance of a DirectX screen.

Exported code will be `OpenWindowedScreen()`



Creates an IPAddressGadget.



Creates a ShortcutGadget.



Creates a WebGadget.



Creates a WebView Gadget



Creates a HyperLinkGadget.



Creates an ExplorerListGadget.



Creates an ExplorerTreeGadget.



Creates an ExplorerComboGadget.



Creates a ScrollGadget.



Creates a ScrollAreaGadget.

A ScrollAreaGadget contains an Area which more Gadgets can be placed. Select the Area in the GadgetTree to place Gadgets on that Area.

Container type Gadgets (PanelGadget, ScrollAreaGadget, ContainerGadget) cannot be placed inside a ScrollAreaGadget.



Creates an ContainerGadget.

A ContainerGadget contains an Area which more Gadgets can be placed. Select the Area in the GadgetTree to place Gadgets on that Area.

Container type Gadgets (PanelGadget, ScrollAreaGadget, ContainerGadget) cannot be placed inside a ContainerGadget.



Creates a CalendarGadget.



Creates a DateGadget.

1.3.2 API PVGadgets



Creates an InputGadget.

This gadget is very similar to the standard StringGadget but has a scroll bar and allows multiple lines of text.



Creates an AnimGadget.

An AnimGadget can display a number of Windows animations like the animated search icon, files being copied animation, files being thrown into trash animation, etc.

NOTE: This gadget will NOT display built in Windows animations on version of Windows higher than Windows XP. If you need this Gadget to work on ALL versions of Windows you will need to supply your own custom animation.



Creates a PureSkinGadget.

Once a PureSkin Gadget is applied to a Form, a PureSkin is then assigned to the Gadget and the Form is instantly skinned. PureSkins are created using the built in [PureSkin Maker](#).



Creates a CanvasGadget.

This Gadget is a simple colored box that allows you to draw on or insert images into. The [PVGadgets_Canvas](#) command allows you to easily convert a Canvas Gadget into a special image button that changes when the mouse passes over it.



Creates a ProgressBarGadget.

This progress bar allows you to set starting and ending colors, creating very nice gradients. Text can also be enabled showing percentage left. It is easily controlled using the [PVGadgets_ProgressBar](#) command.



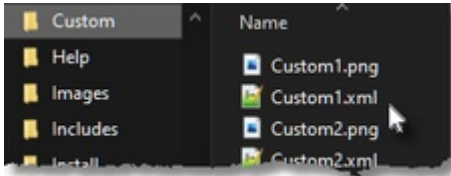
Creates a PieChartGadget.

The PieChart Gadget allows you to create a Pie Chart progress indicator. Text can also be enabled showing percentage left. It is easily controlled using the [PVGadgets PieChart](#) command.

1.3.3 Custom Gadgets

PureVision allows up to 6 Custom Gadgets to be included and used in the Main Interface.

Each Custom Gadget has an image file (Custom1.png - Custom6.png) and a code file (Custom1.xml - Custom6.xml) which tells PureVision how to export the appropriate code.



An include file containing the Custom Gadget Code/Module can also be specified and [injected](#) into the exported file.

PureVision provides a number of keywords (Tokens) which are replaced with PureVision data from the [Properties](#) Form when the project is Exported:

#CURRENTWINDOW#	Replaced with current Window Name
#CURRENTGADGET#	Replaced with current Gadget ID Name
#X#	Replaced with Gadget X position
#Y#	Replaced with Gadget Y position
#W#	Replaced with Gadget Width
#H#	Replaced with Gadget Height
#TEXT# / @TEXT@	Replaced with Display Text and adds Quotes / Replaced with Display Text no added Quotes
#FLAG#	Replaced with any selected Flags
#LANG#	Replaced with Display Text which can be modified with Language Translations
#BGCOLOR#	Replaced with selected Back Color
#FGCOLOR#	Replaced with selected Front Color
#LIMIT#	Replaced with specified Limit value
#MINLIMIT#	Replaced with specified Min value
#MAXLIMIT#	Replaced with specified Max value
#FONTNAME#	Replaced with font Name
#FONTFLAG#	Replaced with font Flags
#FONTSIZE#	Replaced with font Size

The default Custom.xml template file looks like this:

```

<gadgetdata>
  <type>Custom1Gadget</type>
  <customname>GadgetName</customname>
  <customcode>#X#, #Y#, #W#, #H#, #TEXT#, #FLAG#</customcode>
  <customcodelang>#X#, #Y#, #W#, #H#, #LANG#, #FLAG#</customcodelang>
  <customcodetext></customcodetext>
  <customcodefont></customcodefont>
  <customcodetooltip></customcodetooltip>
  <customcodebg></customcodebg>
  <customcodefg></customcodefg>
  <customcodelimit></customcodelimit>
  <customcodemin></customcodemin>
  <customcodemax></customcodemax>
  <general>100000000001</general>
  <inject>000000011</inject>
  <x>10</x>
  <y>10</y>
  <w>120</w>
  <h>20</h>
  <text></text>
  <bgcolor></bgcolor>
  <fgcolor></fgcolor>
  <limit></limit>
  <minlimit></minlimit>
  <maxlimit></maxlimit>
  <flags>
    <element>
      <flag></flag>
      <pbflag></pbflag>
      <flagnum></flagnum>
    </element>
  </flags>
</gadgetdata>

```

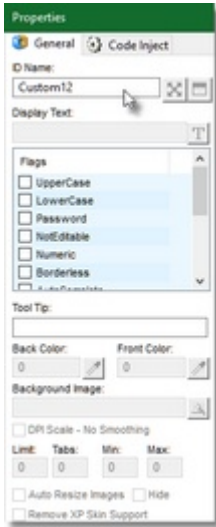
Properties Window (Enable/Disable Options)

General Flags / 12 Slots (0=Disable / 1=Enable)

ID Name	AutoSize	Text	Font	Flags	ToolTip	BGColor	FGColor	Limit	Min	Max	Hide
0	0	0	0	0	0	0	0	0	0	0	0

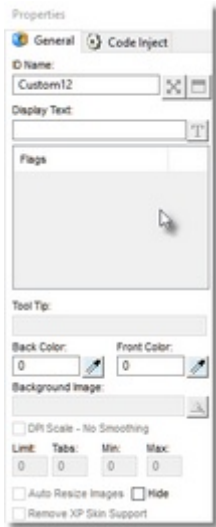
<general>100011000000</general>

This line would enable changing the ID Name, modifying Flags and entering a ToolTip.



`<general>101000110001</general>`

This line would enable changing the ID Name, enable adding Text to the Gadget, modifying the Background Color, modifying the Foreground Color and allowing the Gadget to be Hidden.



Code Inject Flags / 9 Slots (0=Disable / 1=Enable)

Left Click	Left Double Click	Right Click	Right Double Click	Change	Focus	Lost Focus	Outer Loop	Inner Loop
0	0	0	0	0	0	0	0	0

`<inject>100000011</inject>`

This line would enable LeftClick, OuterLoop and InnerLoop in the [Properties/Code Inject](#) options and allow code to be injected into them. It would also Export Code for a LeftClick Gadget Event.

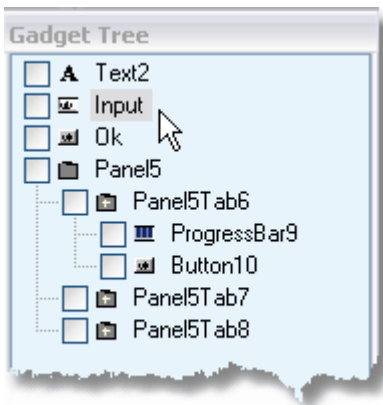
```
<inject>000011111</inject>
```

This would enable Change, Focus, LostFocus, OuterLoop & InnerLoop code inject options and export Gadget Events for Change, Focus and LostFocus.



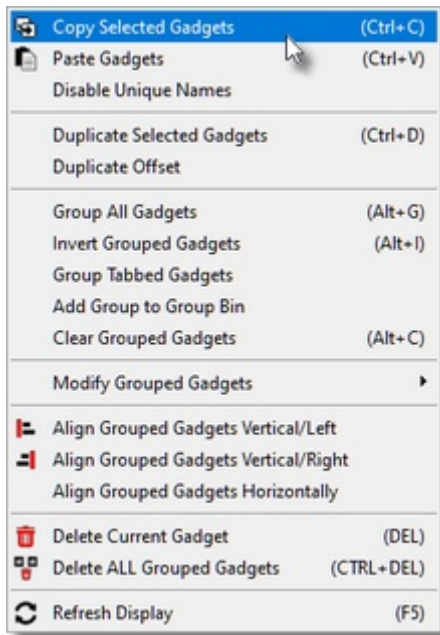
1.4 Gadget Tree

The **Gadget Tree** displays a list of all Gadgets on your current Form in the order they are to be drawn and tabbed through. To change the tab order of a Gadget, select the Gadget you wish to change and use the **PageUp** or **PageDown** keys to move the Gadget up or down the list.



When you select a Gadget in the Gadget Tree, the Gadget is also highlighted on the Form if [Enable Gadget Highlighting](#) is enabled. You can also select a Gadget on a Form using the left or right mouse button and the Gadget selected will also be highlighted in the Gadget Tree.

Additional Gadget Tree options are displayed by pressing the **right mouse button** over the Gadget Tree window.



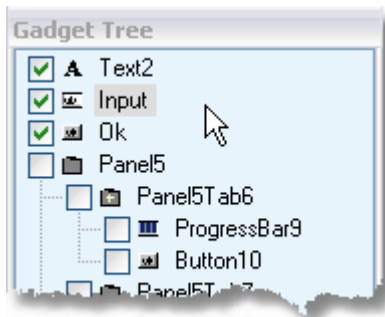
1.4.1 Copy

To **Copy** a Gadget, select the Gadget you wish to copy and press CTRL+C. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Copy Selected Gadgets**.

A copy of the selected Gadget will be copied to the PureVision clipboard.

Once a Gadget is copied to the PureVision clipboard, you can use the [Paste](#) function to paste it onto a Form.

If you wish to copy more than one Gadget at a time, use the [Gadget Tree](#) and place a check in front of all Gadgets to would like to copy. Then press CTRL+C to copy them to the PureVision clipboard.

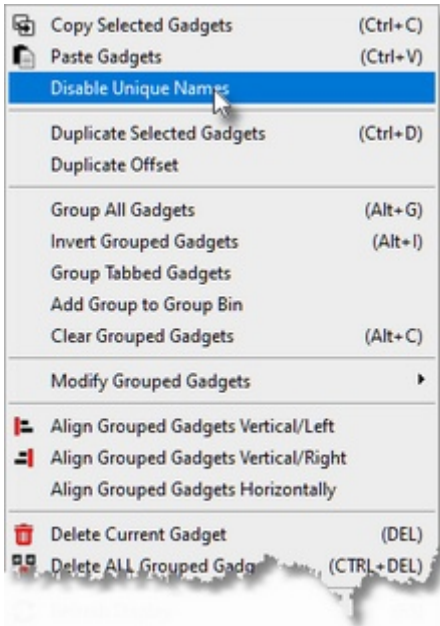


1.4.2 Paste

To **Paste** a Gadget from the PureVision clipboard to the current Form, press CTRL+V. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Paste Gadgets**.

When a Gadget is pasted from the PureVision clipboard, it is automatically given a unique name since duplicate names will give errors when the [exported project](#) is compiled. There may be times when you are pasting Gadgets from one form to another and you would like to keep the original names because the original Gadgets will be deleted.

To do this, right mouse click on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Disable Unique Names** (a check will appear when this option is selected). Next paste the Gadgets from the PureVision clipboard using CTRL+V. The Gadgets will be placed on the Form and will have their original names.

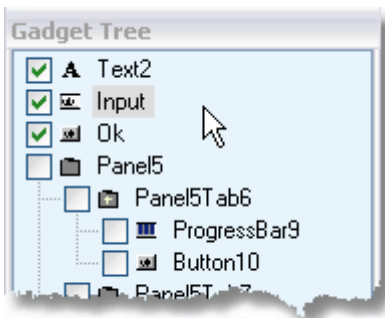


1.4.3 Duplicate

To quickly **Duplicate** a Gadget, select the Gadget you wish to duplicate and press CTRL+D. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Duplicate Selected Gadgets**.

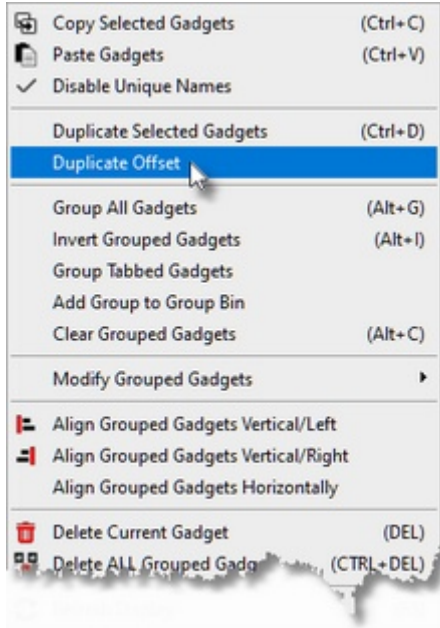
The selected Gadget will be instantly copied and pasted as the offset specified in the [Duplicate Offset](#) settings.

If you wish to duplicate more than one Gadget at a time, use the [Gadget Tree](#) and place a check in front of all Gadgets to would like to duplicate. Then press CTRL+D to duplicate them on your Form.

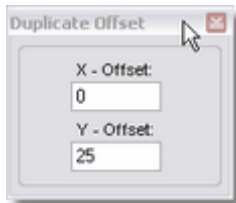


1.4.3.1 Duplicate Offset

To adjust the Duplicate Offset, **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Duplicate Offset**.



A window will appear allowing you to adjust the X and Y offset in pixels.



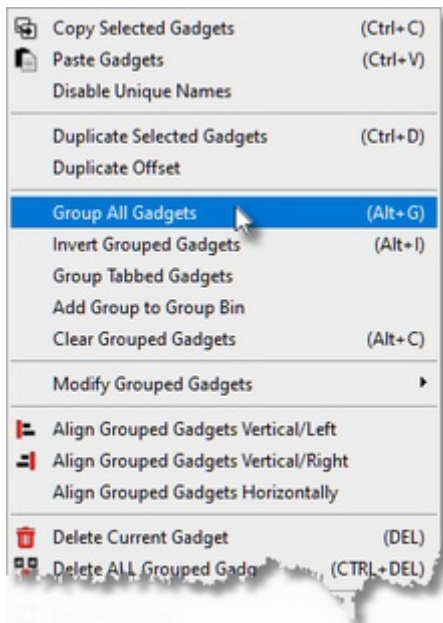
If you enter a value of 25 as the Y-Offset, the Gadget will be placed on the same X axis as the original Gadget but will be placed 25 pixels to the right.

1.4.4 Grouping

Whenever you use the [Gadget Tree](#) and place a check in front of a Gadget, you are grouping Gadgets. When Gadgets are grouped, you are able to Move, Size, Copy, Duplicate many Gadgets at one time.

To Move or Size a Group of Gadgets, select the Gadgets you wish to be in the Group (or select the predefined Group from the Group Bin) and then use the arrow keys to Move or Size the Gadgets.

Right mouse click on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and use will see a number of options for Grouping Gadgets.



Group All Gadgets (ALT+G)

This automatically places a check in front of all Gadgets in the [Gadget Tree](#).

Invert Grouped Gadgets (ALT-I)

This inverts all Gadgets that have been grouped. All Gadgets that have been checked become unchecked and all gadgets that were unchecked become checked.

Group Tabbed Gadgets

Select a Tab in the [Gadget Tree](#) and then select this option. All Gadgets under the selected Tab will become checked.

Add Group to GroupBin

All Gadgets currently grouped in the [Gadget Tree](#) (containing a check) will be added to the selected [GroupBin](#) Group.


Clear Grouped Gadgets (ALT-C)

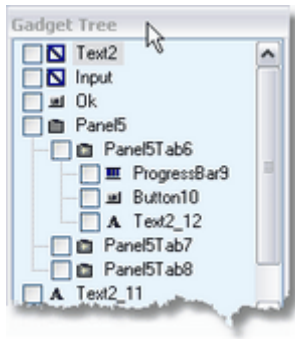
This clears ALL grouped gadgets in the [Gadget Tree](#). (All checks are removed)

1.4.5 Hiding

To have PureVision export code to create a Gadget but not display it on your Form, you can choose to Hide the Gadgets. You can then use the HideGadget command to hide or display the Gadgets when you wish.

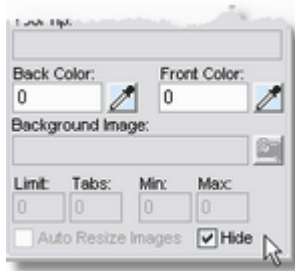
To Hide a Gadget or Group of Gadgets, [Group](#) the Gadgets you wish to hide and press ALT-H. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Hide Grouped Gadgets**. The Gadgets will still be part of your Project but will not be displayed on the Form.

Gadgets that have been hidden will have their icon changed to  in the [Gadget Tree](#).



To Unhide a Gadget or Group of Gadgets, [Group](#) the Gadgets you wish to unhide and press ALT-U. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Unhide Grouped Gadgets**.

You can also Hide and Unhide individual Gadgets using the [Properties](#) Window. Select the appropriate Gadget and check or uncheck the **Hide** box.



1.4.6 Lock

If you do not want to accidentally Move or Size a Gadget, you can Lock the Gadget into position by checking **Lock**.

The only way to make changes after you have locked a Gadget is to Unlock it by unchecking **Lock**.

1.4.7 Delete

To Delete a Gadget from your Form, select the Gadget you wish to delete and press the **DEL** key. You can also **right mouse click** on the [Gadget Tree](#) to display the [Gadget Tree Menu](#) and select **Delete Current Gadget**.

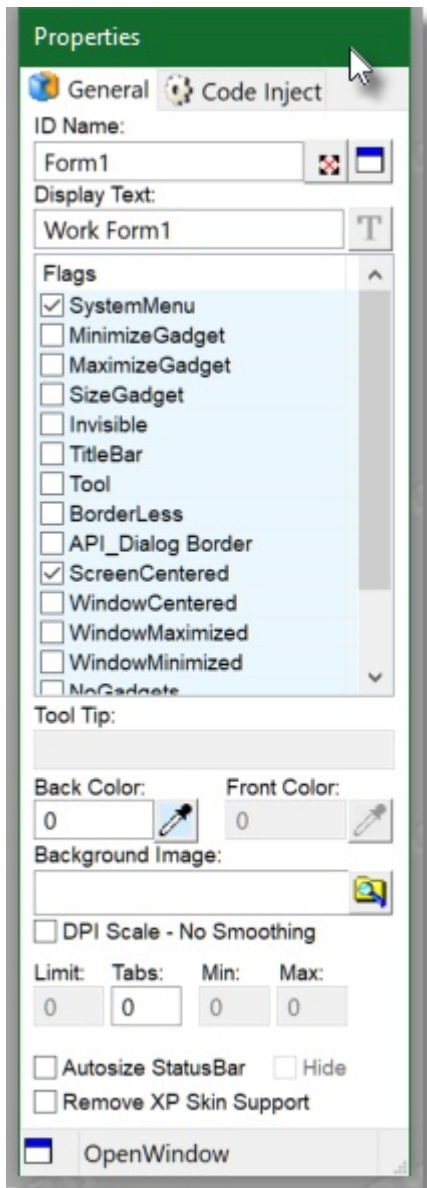
As a safety measure, you will be asked to confirm this action.

1.4.8 Refresh Display

If you ever need to refresh (redraw) the current Form, simply press the function key **F5**. Another way is to **right mouse click** on the [Gadget Tree](#) window to display the [Gadget Tree Menu](#) and select **Refresh Display**.

1.5 Properties

The **Properties** Window allows you to modify the various properties of Forms and Gadgets.



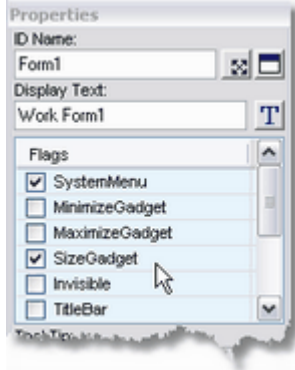
The properties you are allowed to modify will depend on the Form or Gadget you currently have selected.

NOTE: When "DPI Scale - No Smoothing" is selected, the image will be resized with the [#PB_Image_Raw](#) flag otherwise it will use the [#PB_Image_Smooth](#) flag. This is only effective when the Project is Exported with the "[Scale Images for DPI Aware EXE](#)"

1.5.1 Dynamic Resizing

PureVision's powerful **Dynamic Resizing** gives you the ability to easily create Gadgets that resize instantly with your Forms.

To tell PureVision you would like a Gadget to Dynamically Resize, first add the SizeGadget flag or the MaximizeGadget flag to your Form. If either of these flags are not set, your Form will not be exported with Dynamic Resizing.

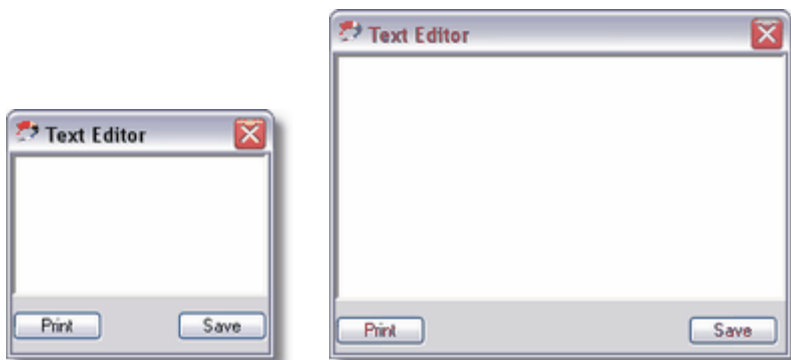




Next place a Gadget on your Form and press the  button in the [Properties](#) Window next to the ID Name of your Gadget. Finally set the Resizing Properties to determine how you would like the Gadget to Move and Size.




Example:

Lets say you would like this Form to resize to this one...



First display the Form [Properties](#) by pressing the  button in the [Extended MenuBar](#) and set the SizeGadget for that Form. Then press the  button and tell PureVision you would like the Form to Size both Horizontally and Vertically.

Starting with the EditorGadget, select this Gadget on your Form and press the  button. We want the EditorGadget to get bigger in size both Horizontally and Vertically when the Form is resized so we need to check both these options.



The ButtonGadget we have called **Print** does not need to change in size but it does need to move Vertically when the Form is resized. Check the Move box to allow Vertical movement only.





If you allowed both Horizontal and Vertical Movement, the result would be this...



Notice how the **Print** ButtonGadget moves both Horizontally and Vertically when the Form is resized. By NOT allowing Horizontal Movement, the Gadget would be locked in place on the Horizontal axis (which is the result we are looking for in this example).

The ButtonGadget we have called **Save** does not need to change in size but it does need to move both Horizontally and Vertically when the Form is resized.




If any Dynamic Resizing Properties have been set for a Form or Gadget, the Dynamic resize button will change from  to .

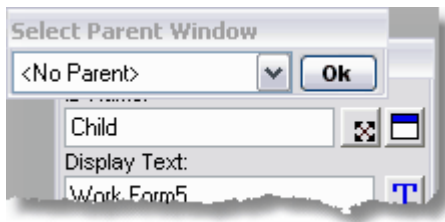
1.5.2 Parent/Child Forms


PureVision makes it easy to define Child Forms and attach them to Parent Forms.

When a Form has been designated as a Child Form, the Form will open on top of the Parent Form and stay on top, even when the Parent beneath has been give the focus.


Once you have created some Forms, select the Form you wish to be the Child Form by using the Form Selector drop box in the [Extended MenuBar](#). Next press the  button in the Properties Window.

The **Select Parent Window** will appear and you can select the Parent Form from the drop box.



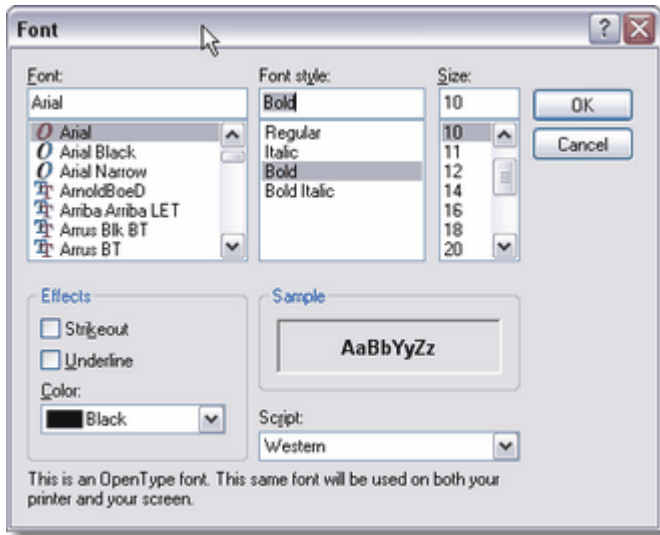
After selecting the Parent Form, press **Ok** and the current Form will become a Child of the selected Parent Form. When a Form has been made a Child Form, the Parent/Child button will change to .

1.5.3 Changing Fonts

Certain Gadgets allow you to modify their Font properties. To change the Font properties on a selected Gadget, press the  button.

The Font Selector will appear and you can select new fonts, styles and sizes.


When the default Font properties have been changed, the Font properties button will change to .

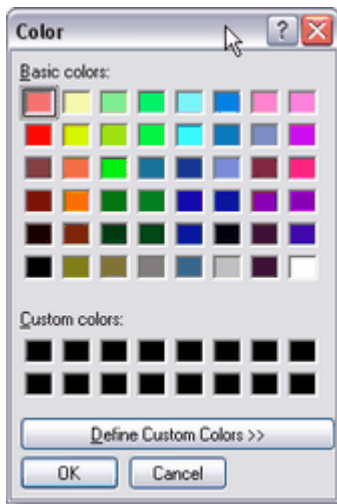


To reset the fonts back to their default settings, set the font size to 0 and press **OK**.

Note: *The color selector will only affect certain Gadgets that allow their color to be changed.*

1.5.4 Gadget Coloring

Depending on the Gadget selected, you may have the option to modify its Back and Front colors. Press the  button to open the Color Selector window.



Setting the Back color to 0 will tell PureVision to use the default Form color as the Back color. In some situations, this will give an undesired result.

In this image, the Back color is set to use the default Form color. The TextGadget overlaps the ImageGadget and by using the default Form color, we see a solid box around the text.



To use a transparent Back color, press the = key on the number pad. The Back color will show as -1 and tells PureVision to create a transparent Back color.



Note: A transparent Back color is not refreshed when the text is changed. If you change your text using SetGadgetText, you must remember to refresh your entire Form after changing or the new text will appear to be drawn over the old text.

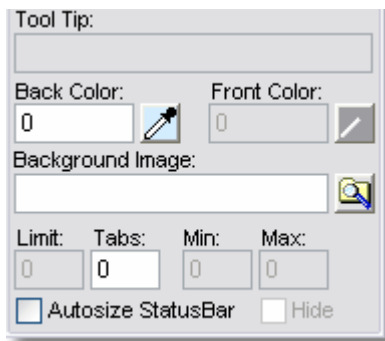
Windows XP overrides many of the Gadget Coloring when [XP Skins](#) is enabled.

1.5.5 StatusBars

The **StatusBar** is part of the Form Properties.

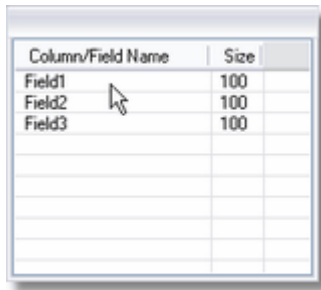
By default the Tab/Fields in the Properties tool window is set to 0, meaning there is no StatusBar added to your form.

Change this number to higher than 0 to enable the StatusBar with the number of Tabs/Fields you wish displayed.



If you are creating a resizable Form, you can have the StatusBar automatically resize with the Form by checking the **Autosize StatusBar** option.

Once Tabs/Fields have been added to a StatusBar, the Tabs/Fields tool window will display the name of each field along with the field size. To edit this information, double click on the appropriate field with the left mouse button.



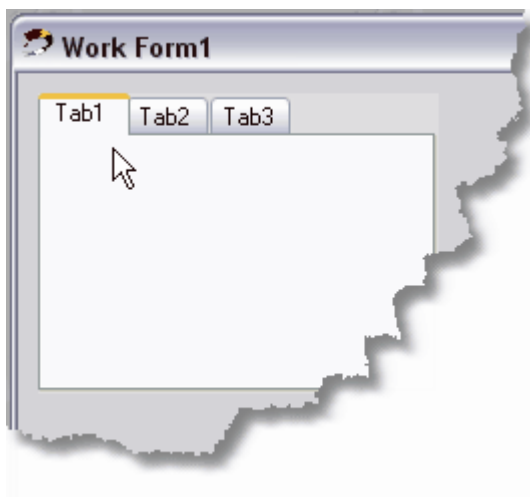
Column/Field Name	Size
Field1	100
Field2	100
Field3	100

To Insert or Delete StatusBar Tabs/Fields, use the Insert or Delete keys.

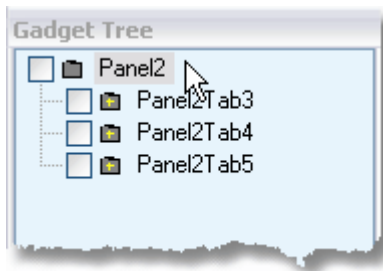
Note: If you would like to assign the #PB_Ignore variable to the Statusbar Width, set the field size to zero.

1.5.6 PanelGadgets

The **PanelGadget** is a special container consisting of multiple tabs, each tab holding any number of other Gadgets.



When a PanelGadget is added to a Form, 3 Tabs are created by default. To add a new Tab to a Panel, select the Panel in the Gadget Tree and press the INSERT key on the keyboard.



To delete a Tab from a Panel, select the Tab in the Gadget Tree that you wish to delete and press the

DELETE key on the keyboard.

To change the order of a Tab within a PanelGadget, select the Tab you want to move and press the PGUP button to move the Tab up in the Gadget Tree.

NOTES:

(1) When a Tab is deleted, all Gadgets on the Tab are also deleted. When a PanelGadget is deleted, all Tabs and their Gadgets are also deleted.

(2) Container type Gadgets cannot currently be placed in other container type Gadget. For example, you cannot place a PanelGadget inside a ContainerGadget or a ScrollArea Gadget inside a Panel Gadget.

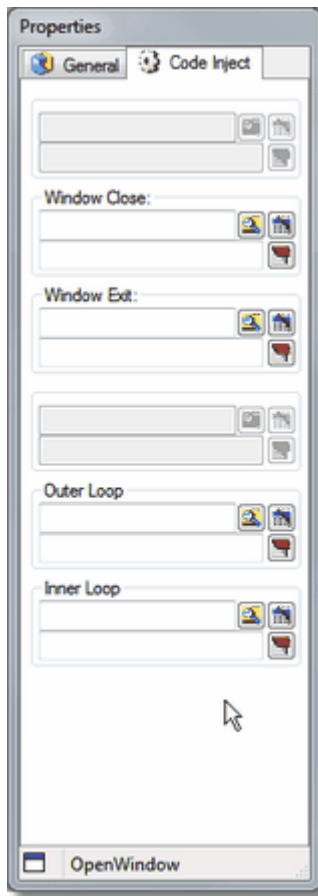
1.5.7 Code Inject

Code Inject allows you to compose various code snippets and have PureVision Inject them into the exported code.

Code snippets are injected into various locations of the exported code depending on whether Windows (Forms) or Gadgets are being used.

Windows

When working with a Form, Code Inject allows code to be injected on Window Close, Windows Exit, Outer Loop and Inner Loop.



The following image shows where code is injected when using each of the Form Injection options.

```

5
6
7
8 XIncludeFile "PV_debug_Constants.pb"
9 XIncludeFile "PV_debug_Windows.pb"
10
11
12 MessageRequester("Test", "Test 3")
13
14
15
16
17
18 :- Main Loop
19 If Window_Form1()
20     MessageRequester("Test", "Test 4")
21
22
23     Define quitForm1=0
24     Repeat
25         EventID =WaitWindowEvent()
26         MenuID  =EventMenu()
27         GadgetID =EventGadget()
28         WindowID =EventWindow()
29
30         Select EventID
31             Case #PB_Event_CloseWindow
32                 Select WindowID
33                     Case #Window_Form1
34                         quitForm1=1
35                     EndSelect
36
37             Case #PB_Event_Gadget
38                 Select GadgetID
39                     EndSelect
40
41             EndSelect
42         Until quitForm1
43         MessageRequester("Test", "Test 1")
44
45         CloseWindow(#Window_Form1)
46         MessageRequester("Test", "Test 2")
47
48     EndIf
49 End
50
51

```

The Properties window on the right shows the 'Code Inject' tab with the following settings:

- Custom Includes: Includes\CustomInclude
- Window Close: Includes\test1.pb
- Window Exit: Includes\test2.pb
- Outer Loop: Includes\test3.pb
- Inner Loop: Includes\test4.pb

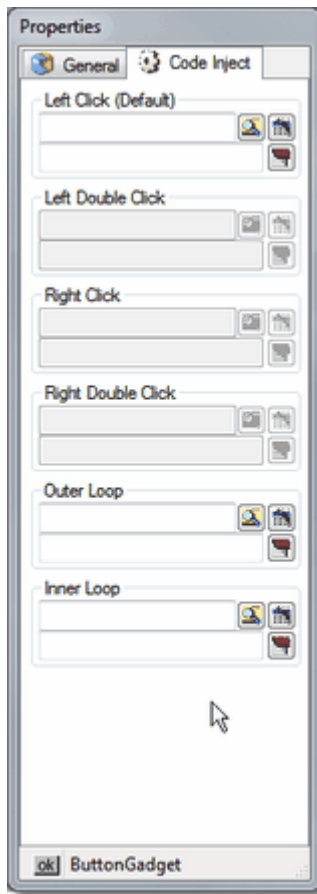
Arrows indicate the injection points: a red arrow points from the 'Custom Includes' field to the `XIncludeFile` lines; a blue arrow points from the 'Window Close' field to the `CloseWindow` call; a red arrow points from the 'Window Exit' field to the `MessageRequester` call for 'Test 1'; a blue arrow points from the 'Outer Loop' field to the `Repeat` loop; and a blue arrow points from the 'Inner Loop' field to the `Until` loop.

The options for Window Close, Window Exit, Outer Loop and Inner Loop all have Code Injected in the Main Event Loop.

If "Custom Includes" is used, the Code is injected in the "_Constants.pb" file when Exported. This is extremely useful when using [Custom Gadgets](#) since you can include custom Modules and other code files which will be used by the Custom Gadgets.

Gadgets

When working with a Gadget like a ButtonGadget, Code Inject allows code to be injected on Left Click, Outer Loop and Inner Loop.



The following image shows where code is injected when using each of the Gadget Injection options.

The image shows a screenshot of the PureVision development environment. On the left is a code editor with a light blue background, displaying a PureBasic script. On the right is the 'Properties' window, which has two tabs: 'General' and 'Code Inject'. The 'Code Inject' tab is active, showing a list of event handlers for a 'ButtonGadget'. The list includes 'Left Click (Default)', 'Left Double Click', 'Right Click', 'Right Double Click', 'Outer Loop', and 'Inner Loop'. Each event handler has a corresponding 'Includes' field. The 'Outer Loop' field contains 'Includes\test2.pb' and the 'Inner Loop' field contains 'Includes\test3.pb'. Two red arrows point from the 'Code Inject' tab to the code editor: one from the 'Left Click (Default)' field to the 'MessageRequester("Test", "Test 2")' line (line 12), and another from the 'Outer Loop' field to the 'MessageRequester("Test", "Test 3")' line (line 20). A blue arrow points from the 'Inner Loop' field to the 'MessageRequester("Test", "Test 1")' line (line 41). The code in the editor is as follows:

```
7
8 XIncludeFile "PV_debug_Constants.pb"
9 XIncludeFile "PV_debug_Windows.pb"
10
11
12 MessageRequester("Test", "Test 2")
13
14
15
16
17
18 :- Main Loop
19 If Window_Form1()
20   MessageRequester("Test", "Test 3")
21
22
23   Define quitForm1=0
24   Repeat
25     EventID =WaitWindowEvent()
26     MenuID =EventMenu()
27     GadgetID =EventGadget()
28     WindowID =EventWindow()
29
30     Select EventID
31       Case #PB_Event_CloseWindow
32         Select WindowID
33           Case #Window_Form1
34             quitForm1=1
35         EndSelect
36
37       Case #PB_Event_Gadget
38         Select GadgetID
39           Case #Gadget_Form1_Button2
40             MessageRequester("Test", "Test 1")
41
42         EndSelect
43
44     EndSelect
45   Until quitForm1
46   CloseWindow(#Window_Form1)
47 EndIf
48 End
49
50
```

Visit the PureVision website to view a video on using the Code Inject function.
<https://reelmedia.org/purevision/index.php/support>

Part



2 PVGadgets User Library

2.1 PVGadgets Functions

PVGadgets are a collection of Gadgets and Functions that extend and enhance what is currently available in the PureBasic Compiler. Many of these Functions are automatically created and implemented through the PureVision form designer.

This reference is for those wanting to understand why PureVision creates and exports the syntax it does and how to implement these functions manually when customizing and fine tuning your final source code.

To use the PVGadgets Module there must be an include reference to the Module in your source code. PureVision will automatically include this reference for you.

```
XIncludeFile "C:/PureVision/Modules/Module_PVGadgets.pbi" ;<-- Modify as needed for path of  
Modules Folder
```

[PVDynamic](#)
[PVGadgets](#)

[PureSkin](#)
[PurePoint](#)

Note: PureVision will reference the Module Functions with **PVX::** when generating Source Code but for backwards compatibility this is currently not needed.

Copyright © 2002 - 2022, Reel Media Productions™
"PureVision" and "PVGadgets" are trademarks of Reel Media Productions. All rights are reserved.

2.1.1 PVDynamic

PVDynamic functions give PureVision the ability to Dynamically Resize/Move/Color/Lock Forms and Gadgets, turning an ordinary Form into a highly configurable dynamic one.

[PVDynamic Example Code](#)

PVX::[PVDynamic_AddWindow](#)
PVX::[PVDynamic_ColorGadget](#)

PVX::[PVDynamic_AddGadget](#)
PVX::[PVDynamic_AddColorGadget](#)
PVX::[PVDynamic_FreeColorGadget](#)
PVX::[PVDynamic_AddStatusBar](#)

Copyright © 2002 - 2022, Reel Media Productions™
"PureVision" and "PVDynamic" are trademarks of Reel Media Productions. All rights are reserved.

2.1.1.1 PVDynamic Example Code

PureVision automatically adds this to the start of your Event Loop when a project is exported.

A full Initialization & Callback loop in your main source code might look like this:

```
;-Init Includes
XIncludeFile "Example_Constants.pb"
XIncludeFile "Example_Windows.pb"

Procedure WindowCallback(WindowID,Message,wParam,lParam)
  ReturnValue=#PB_ProcessPureBasicEvents
  If Message=#WM_CTLCOLORSTATIC Or Message=#WM_CTLCOLOREDIT Or
  Message=#WM_CTLCOLORLISTBOX
    ReturnValue=PVX::PVDynamic_ColorGadget(lParam,wParam)
  EndIf
  ProcedureReturn ReturnValue
EndProcedure
```

If you have made a number of changes to your project regarding Dynamic Resizing and your Event Loop already exists from an earlier Export, make sure the SizeWindowHandler() Procedure is at the beginning of your Main Event Loop.

If you are updating an existing Project and are unable to Export your original GUI File, you can manually edit your "_Window.pb" file.
Any line containing PVDynamic_AddLockWindow, change to the following...

```
PVDynamic_AddLockWindow(#Window_Form1,1) -->
WindowBounds(#Window_Form1,WindowWidth(#Window_Form1),WindowHeight(#Window_Form1),WindowWidth(#Window_Form1),#PB_Ignore)

PVDynamic_AddLockWindow(#Window_Form1,2) -->
WindowBounds(#Window_Form1,WindowWidth(#Window_Form1),WindowHeight(#Window_Form1),#PB_Ignore,WindowHeight(#Window_Form1))

PVDynamic_AddLockWindow(#Window_Form1,3) -->
WindowBounds(#Window_Form1,WindowWidth(#Window_Form1),WindowHeight(#Window_Form1),#PB_Ignore,#PB_Ignore)
```

2.1.1.2 PVDynamic_ColorGadget

Use this function in a Callback to Dynamically Color Gadgets.

This is automatically created by PureVision when an Event Loop is exported.

Example:

```

Procedure.i GadgetCallback(GadgetID,Message,wParam,lParam)
  Protected ReturnValue

  If Message=#WM_CTLCOLORSTATIC Or Message=#WM_CTLCOLOREDIT Or
  Message=#WM_CTLCOLORLISTBOX
    ReturnValue=PVX::PVDynamic_ColorGadget(lParam,wParam)
    If ReturnValue=#PB_ProcessPureBasicEvents
      ReturnValue=0
    EndIf
  EndIf

  If ReturnValue
    ProcedureReturn ReturnValue
  Else
    ProcedureReturn
  EndIf
SubClass::CallGadgetProc(GadgetID,Message,wParam,lParam)
  EndIf
EndProcedure

```

```

Procedure.i WindowCallback(WindowID,Message,wParam,lParam)
  Protected ReturnValue=#PB_ProcessPureBasicEvents

  If Message=#WM_CTLCOLORSTATIC Or Message=#WM_CTLCOLOREDIT Or
  Message=#WM_CTLCOLORLISTBOX
    ReturnValue=PVX::PVDynamic_ColorGadget(lParam,wParam)
  EndIf

  ProcedureReturn ReturnValue
EndProcedure

```

2.1.1.3 PVDynamic_AddWindow

Adds a Form to the Dynamic Resize Library to be Dynamically Resized.

Use:

PVX::PVDynamic_AddWindow (#Window)

This is automatically created by PureVision when a project is exported.
Any Form you wish to contain Dynamic Resizing Gadgets must first be added with PVDynamic_AddWindow

2.1.1.4 PVDynamic_AddGadget

Add a Gadget to be Dynamically Resize.

Use:

PVX::PVDynamic_AddGadget(#Window,#Gadget,4)

This is automatically created by PureVision when a project is exported.

MoveSize

- 1 - Vertical Sizing
- 2 - Horizontal Sizing
- 4 - Vertical Movement
- 8 - Horizontal Moving

To allow a Gadget to Dynamically size horizontally and move vertically:
PVX::PVDynamic_AddGadget(#Window,#Gadget,2|4)

For Gadgets to Dynamically Resize on a Form, the Form must first be added using [PVDynamic_AddWindow](#)

2.1.1.5 PVDynamic_AddColorGadget

Add a Gadget to be Dynamically Colored when it cannot be colored by normal means.

Use:

```
PVX::PVDynamic_AddColorGadget(#Gadget,$FF0000,$00FFFF)
```

or

```
PVX::PVDynamic_AddColorGadget(#Gadget,RGB(0,0,255),RGB(255,255,0))
```

If a background color of -1 is used, the background will be transparent.

This is automatically created by PureVision when a project is exported.

This applies to the following Gadgets:

- TextGadget
- OptionGadget
- CheckBoxGadget
- ComboBoxGadget
- TrackBarGadget

2.1.1.6 PVDynamic_FreeColorGadget

Free a Dynamic Color Gadget.

Use:

```
PVX::PVDynamic_FreeColorGadget (#Gadget)
```

2.1.1.7 PVDynamic_AddStatusBar

Adds Dynamic AutoSizing to a Statusbar.

Use:

```
PVX::PVDynamic_AddStatusBar (#Window,#Statusbar,hStatusBar,2)
```

This is automatically created by PureVision when a project is exported.

Example:

```
hStatusBar=CreateStatusBar( #StatusBar_Main,WindowID( #Window_Main) )
PVX::PVDynamic_AddStatusBar( #Window_Main,#StatusBar_Main,hStatusBar,2)
    AddStatusBarField(100)
    AddStatusBarField(100)
```

2.1.2 PVGadgets

PVGadgets are a collection of custom Gadgets and Functions that add to what is currently available in the PureBasic Compiler.

[PVX::PVGadgets_InitAnimGadget](#)
[PVX::PVGadgets_AnimGadget](#)
[PVX::PVGadgets_FreeAnimGadget](#)

[PVX::PVGadgets_BubbleTip](#)

[PVX::PVGadgets_Canvas](#)
[PVX::PVGadgets_CanvasFree](#)
[PVX::PVGadgets_CanvasButtonAdd](#)
[PVX::PVGadgets_CanvasButtonHover](#)
[PVX::PVGadgets_CanvasButtonPressed](#)
[PVX::PVGadgets_CanvasButtonDelete](#)
[PVX::PVGadgets_CanvasEnableClick](#)
[PVX::PVGadgets_CanvasDisableClick](#)

[PVX::PVGadgets_ProgressBar](#)
[PVX::PVGadgets_FreeProgressBar](#)

[PVX::PVGadgets_Refresh](#)
[PVX::PVGadgets_RefreshWindow](#)

[PVX::PVGadgets_WindowFreeze](#)
[PVX::PVGadgets_WindowUnfreeze](#)
[PVX::PVGadgets_ToolWindow](#)
[PVX::PVGadgets_WindowTransparent](#)
[PVX::PVGadgets_WindowTop](#)
[PVX::PVGadgets_WindowReset](#)
[PVX::PVGadgets_CurDir](#)
[PVX::PVGadgets_TempDir](#)
[PVX::PVGadgets_ExeDir](#)

Copyright © 2002 - 2022, Reel Media Productions™
"PureVision" and "PVGadgets" are trademarks of Reel Media Productions. All rights are reserved.

2.1.2.1 AnimGadget

2.1.2.1.1 PVGadgets_InitAnimGadget

Initializes the PVGadgets_AnimGadget.

Use:

```
hAnimGadget=PVX::PVGadgets_InitAnimadget()
```

Returns special AnimGadget Handle used for creating the [PVX::PVGadgets_AnimGadget](#).

This command must be called at the start of any program using [PVX::PVGadgets_AnimGadget](#).

Example:

```
hAnimGadget=PVX::PVGadgets_InitAnimGadget()  
#Window_Main=1  
  
If  
OpenWindow(#Window_Main,175,0,282,68,#PB_Window_SystemMenu|#PB_Window_Scre  
enCentered,"Demo")  
  
    Gadget_Main_Anim=PVX::PVGadgets_AnimGadget(hAnimGadget,#Window_Main,0,0  
    ,280,65,160)  
  
    Repeat  
    Until WaitWindowEvent()=#PB_Event_CloseWindow  
    PVX::PVGadgets_FreeAnimGadget(hAnimGadget)  
  
EndIf
```

2.1.2.1.2 PVGadgets_AnimGadget

Creates a PVGadgets_AnimGadget using the specified animation.

Use:

```
hGadget=PVX::PVGadgets_InitAnimadget(hAnimGadget)
```

Returns Handle of created Gadget.

Flag

- 150 - Search
- 151 - Search Document
- 152 - Search Computer
- 160 - Copy Multi Files
- 161 - Copy File
- 162 - Delete Files
- 163 - Empty Recycle Bin
- 164 - Kill File

Example:

```

hAnimGadget=PVX::PVGadgets_InitAnimGadget( )
#Window_Main=1

If
OpenWindow( #Window_Main,175,0,282,68,#PB_Window_SystemMenu|#PB_Window_Scre
enCentered,"Demo")

    Gadget_Main_Anim=PVGadgets_AnimGadget(hAnimGadget,#Window_Main,0,0,280,
65,160)

    Repeat
    Until WaitWindowEvent( )=#PB_Event_CloseWindow
    PVX::PVGadgets_FreeAnimGadget(hAnimGadget)

EndIf

```

2.1.2.1.3 PVGadgets_AnimGadgetPlay

Plays an AnimGadget animation sequence forever unless number of loops is specified. Start and end range can also be specified.

Use:

```
PVGadgets_AnimGadgetPlay(Gadget,4,6,12)
```

Loop

Number of times to loop animation sequence (default= forever)

First

Starting frame of animation sequence (default=first frame)

Last

Ending frame of animation sequence (default=last frame)

Example:

```

hAnimGadget=PVX::PVGadgets_InitAnimGadget( )
#Window_Main=1

If
OpenWindow( #Window_Main,175,0,282,68,#PB_Window_SystemMenu|#PB_Window_Scre
enCentered,"Demo")

    Gadget_Main_Anim=PVX::PVGadgets_AnimGadget(hAnimGadget,#Window_Main,0,0
,280,65,160)

    PVX::PVGadget_AnimGadgetPlay(Gadget_Main_Anim,10,6,12)

    Repeat
    Until WaitWindowEvent( )=#PB_Event_CloseWindow
    PVX::PVGadgets_FreeAnimGadget(hAnimGadget)

EndIf

```

2.1.2.1.4 PVGadgets_AnimGadgetStop

Stops AnimGadget animation sequence.

Use:

PVGadgets_AnimGadgetStop(Gadget)

Example:

```
hAnimGadget=P VX::PVGadgets_InitAnimGadget()  
#Window_Main=1  
  
If  
OpenWindow(#Window_Main,175,0,282,68,#PB_Window_SystemMenu|#PB_Window_Scre  
enCentered,"Demo")  
  
    Gadget_Main_Anim=P VX::PVGadgets_AnimGadget(hAnimGadget,#Window_Main,0,0  
    ,280,65,160)  
  
    P VX::PVGadget_AnimGadgetStop(Gadget_Main_Anim)  
  
    Repeat  
    Until WaitWindowEvent()=#PB_Event_CloseWindow  
    P VX::PVGadgets_FreeAnimGadget(hAnimGadget)  
  
EndIf
```

2.1.2.1.5 PVGadgets_FreeAnimGadget

Frees all PVGadgets_AnimGadget resources.

Use:

PVGadgets_InitAnimadget(hAnimGadget)

Use this command to free all resources when your application is closed.

Example:

```
hAnimGadget=P VX::PVGadgets_InitAnimGadget()  
#Window_Main=1  
  
If  
OpenWindow(#Window_Main,175,0,282,68,#PB_Window_SystemMenu|#PB_Window_Scre  
enCentered,"Demo")  
  
    Gadget_Main_Anim=P VX::PVGadgets_AnimGadget(hAnimGadget,#Window_Main,0,0  
    ,280,65,160)  
  
    Repeat  
    Until WaitWindowEvent()=#PB_Event_CloseWindow  
    P VX::PVGadgets_FreeAnimGadget(hAnimGadget)  
  
EndIf
```

2.1.2.1.6 PVGadgets_FreeAnimGadgetImages

Free AnimImageGadget Images (AVI Files) by deleting them from the specified path. All AVI's found in the path will be deleted.

Use this command only if you want your AVI's deleted.
By default, PureVision places AVI images stored in the EXE in the Windows TEMP folder.

Use:
PVX::PVGadgets_FreeAnimGadgetImages("Temp")

2.1.2.2 BubbleTipGadget

2.1.2.2.1 PVGadgets_BubbleTip

Creates a Bubble Style Tool Tip and assigns it to a Gadget.

Use:
PVX::PVGadgets_BubbleTip(Window,Gadget,"This is a BubbleTip",PVX::#BubbleTip,\$DFFFFFF)

Both Flag and BColor are optional.

Flag

PVX::#BubbleTip - BubbleTip Style
PVX::#ToolTip - Regular ToolTip Style
PVX::#DisableTip - Disable BubbleTip

Using the Flag allows you to change the style of your bubble tip or disable it completely during the execution of your program.

BColor

\$0 - \$FFFFFF - Set Background color of BubbleTip

2.1.2.2.2 PVGadgets_BubbleTipText

Changes the Text of a BubbleTip.

Use:
PVX::PVGadgets_BubbleTipText(Gadget,"Hello World")

2.1.2.2.3 PVGadgets_BubbleTipTitle

Changes the Title of a BubbleTip.

Use:
PVX::PVGadgets_BubbleTipTitle(Gadget,"My Title",#IDI_INFORMATION)

Both Icon and Size are optional.

Icon

#IDI_APPLICATION ;32512
#IDI_ERROR ;32513


```
#IDI_QUESTION ;32514
#IDI_WARNING ;32515
#IDI_INFORMATION ;32516
#IDI_WINLOGO ;32517
#IDI_SHIELD ;32518
```

Specifying an Icon will put an icon to the left of the Title

Size

Adjusts the size of the icon. Default is 16

2.1.2.2.4 PVGadgets_BubbleTipColor

Changes the Color of a BubbleTip.

Use:

```
PVX::PVGadgets_BubbleTipColor(Gadget,#PB_Gadget_BackColor,$FF0000)
```

Gadget can be #PB_Any to change ALL BubbleTips at once

Attribute

```
#PB_Gadget_BackColor
#PB_Gadget_FrontColor
```

Value

\$0 - \$FFFFFF - Set Color of BubbleTip or BubbleTip Text

2.1.2.2.5 PVGadgets_BubbleTipDelete

Deletes a BubbleTip assigned to a Gadget and frees resources.

Use:

```
PVX::PVGadgets_BubbleTipDelete(Gadget)
```

2.1.2.3 CanvasGadget

2.1.2.3.1 PVGadgets_Canvas

Creates a Canvas Gadget for you to Draw with.

Use:

```
PVGadgets_Canvas(#Gadget,#Image,10,10,200,200,$EEEEEE)
```

Returns 0 if Failed

BColor

\$0 - \$FFFFFF (RGB Background Color Value)

Example:

```
#Window_Main=1

#Gadget_Main_Canvas=2
#Image_Main_Canvas=2

If
OpenWindow( #Window_Main,175,0,184,172,#PB_Window_SystemMenu|#PB_Window_Scr
eenCentered,"Demo" )

    PVGadgets_Canvas( #Gadget_Main_Canvas,#Image_Main_Canvas,15,25,150,100,$
EEEEEE )

    UseImage( #Image_Main_Canvas )
    StartDrawing( ImageOutput() )
        Circle(50,50,50,$FF0000)
        Box(60,60,70,30,$0000FF)
    StopDrawing()
    PVGadgets_Refresh( #Gadget_Main_Canvas )

    Repeat
    Until WaitWindowEvent()=#PB_Event_CloseWindow

EndIf
```

2.1.2.3.2 PVGadgets_CanvasFree

Frees Resources of Specified Canvas/Image.

Use:

PVGadgets_CanvasFree (#Gadget,#Image)

All resources are freed and Canvas can no longer be used.

2.1.2.3.3 PVGadgets_CanvasButtonAdd

Creates a CanvasButton on specific Window by specifying the CanvasID, the Default Image and the Replacement Image.

Use:

PVGadgets_CanvasButtonAdd(#Window,#Gadget,#Image1,#Image2)

Example:

```
XIncludeFile "Buttons_Constants.pb"
XIncludeFile "Buttons_Windows.pb"

;-Main Loop
If Window_Main()
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Settings,#Image_Main_
  Settings,#Image_Icons_Settings)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Web,#Image_Main_Web,#
  Image_Icons_Web)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Info,#Image_Main_Info
  ,#Image_Icons_Info)

  quitMain=0
  Repeat
    PVGadgets_CanvasButtonHover(#Window_Main)
    EventID=WindowEvent()
    If EventID
      Select EventID
        Case #PB_Event_CloseWindow
          If EventWindowID()=#Window_Main
            quitMain=1
          EndIf

        Case #WM_LBUTTONDOWN
          Select PVGadgets_CanvasButtonPressed()
            Case #Gadget_Main_Settings
              SetGadgetText(#Gadget_Main_Press,"S e t t i n g s . . .")
            Case #Gadget_Main_Web
              SetGadgetText(#Gadget_Main_Press,"W e b s i t e . . .")
            Case #Gadget_Main_Info
              SetGadgetText(#Gadget_Main_Press,"I n f o r m a t i o n . . .")
          EndSelect
        EndSelect

      Else
        Delay(1)
      EndIf
    Until quitMain
    CloseWindow(#Window_Main)
  EndIf
```

2.1.2.3.4 PVGadgets_CanvasButtonHover

Checks if Mouse Pointer is over CanvasButton on specified Window and replaces with Sub Image.

Use:

Result=PVGadgets_CanvasButtonHover()

Returns CanvasID of CanvasButton that mouse hovers over.

Example:

```

XIncludeFile "Buttons_Constants.pb"
XIncludeFile "Buttons_Windows.pb"

;-Main Loop
If Window_Main()
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Settings,#Image_Main_
  Settings,#Image_Icons_Settings)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Web,#Image_Main_Web,#
  Image_Icons_Web)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Info,#Image_Main_Info
  ,#Image_Icons_Info)

quitMain=0
Repeat
  PVGadgets_CanvasButtonHover(#Window_Main)
  EventID=WindowEvent()
  If EventID
    Select EventID
      Case #PB_Event_CloseWindow
        If EventWindowID()=#Window_Main
          quitMain=1
        EndIf

      Case #WM_LBUTTONDOWN
        Select PVGadgets_CanvasButtonPressed()
          Case #Gadget_Main_Settings
            SetGadgetText(#Gadget_Main_Press,"S e t t i n g s . . .")
          Case #Gadget_Main_Web
            SetGadgetText(#Gadget_Main_Press,"W e b s i t e . . .")
          Case #Gadget_Main_Info
            SetGadgetText(#Gadget_Main_Press,"I n f o r m a t i o n . . .")
        EndSelect
      EndSelect

    Else
      Delay(1)
    EndIf
  Until quitMain
  CloseWindow(#Window_Main)
EndIf

```

2.1.2.3.5 PVGadgets_CanvasButtonPressed

Returns CanvasID of CanvasButton that Mouse Pointer is over when pressed.

Use:

Result=PVGadgets_CanvasButtonPressed(#Window_Main)

Example:

```
XIncludeFile "Buttons_Constants.pb"
XIncludeFile "Buttons_Windows.pb"

;-Main Loop
If Window_Main()
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Settings,#Image_Main_
  Settings,#Image_Icons_Settings)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Web,#Image_Main_Web,#
  Image_Icons_Web)
  PVGadgets_CanvasButtonAdd(#Window_Main,#Gadget_Main_Info,#Image_Main_Info
  ,#Image_Icons_Info)

  quitMain=0
  Repeat
    PVGadgets_CanvasButtonHover(#Window_Main)
    EventID=WindowEvent()
    If EventID
      Select EventID
        Case #PB_Event_CloseWindow
          If EventWindowID()=#Window_Main
            quitMain=1
          EndIf

        Case #WM_LBUTTONDOWN
          Select PVGadgets_CanvasButtonPressed()
            Case #Gadget_Main_Settings
              SetGadgetText(#Gadget_Main_Press,"S e t t i n g s . . .")
            Case #Gadget_Main_Web
              SetGadgetText(#Gadget_Main_Press,"W e b s i t e . . .")
            Case #Gadget_Main_Info
              SetGadgetText(#Gadget_Main_Press,"I n f o r m a t i o n . . .")
          EndSelect
        EndSelect

      Else
        Delay(1)
      EndIf
    Until quitMain
    CloseWindow(#Window_Main)
  EndIf
```

2.1.2.3.6 PVGadgets_CanvasButtonDelete

Deletes a CanvasButton making it no longer detected by [PVGadgets_CanvasButtonHover](#) and [PVGadgets_CanvasButtonPressed](#)

Use:

Result=PVGadgets_CanvasButtonDelete(#Gadget)

Returns 0 if Failed

2.1.2.3.7 PVGadgets_CanvasEnableClick

Enables a mouse click event to be returned from an Image. (PureBaisc's default behavior)

Use:
Result=PVGadgets_CanvasEnableClick(#Gadget)

2.1.2.3.8 PVGadgets_CanvasDisableClick

Disables an Image from returning a mouse click event.

Use:
Result=PVGadgets_CanvasDisableClick(#Gadget)

2.1.2.4 ProgressBarGadget

2.1.2.4.1 PVGadgets_ProgressBar

Create/Update a PVGadgets ProgressBar.

Use:
PVGadgets_ProgressBar(#Gadget,#Image,10,10,100,20,0,1,\$000000,\$0000FF,\$00FFFF,#PB_Image_Border)

Progress

0 - 100 (representing %)

Shadow

#NoShadow - no shadow on number

#Shadow - shadow on number

#NoNumber - no number

BColor

\$0 - \$FFFFFF (RGB values of background color)

#SystemColor - Use Background Color of Form

FColor1

\$0 - \$FFFFFF (RGB values of foreground color)

FColor2

\$0 - \$FFFFFF (RGB values of foreground color)

Flag (optional)

#NoBorder - no border

#FlatBorder - flat border

#SunkenBorder - sunken border

- Once this Gadget has been created, you can set x,y,w,h to 0 as they are not needed.
- Setting FColor1 different from FColor2 will cause the color of the bar to fade from one color to the next.

Example:

```
#Gadget=1
#Image=1

If
OpenWindow(0,0,0,200,100,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"
Demo")

    PVGadgets_ProgressBar(#Gadget,#Image,10,40,170,20,0,1,$000000,$0000FF,$
00FFFF,#SunkenBorder)

For tmp=0 To 100
    PVGadgets_ProgressBar(#Gadget,#Image,0,0,0,0,tmp,1,$000000,$0000FF,$00F
FFF)
    Delay(30)
Next
EndIf
```

2.1.2.4.2 PVGadgets_FreeProgressBar

Frees Resources of Specified ProgressBar/Image.

Use:

PVGadgets_FreeProgressBar (#Gadget,#Image)

All resources are freed and ProgressBar can no longer be used.

2.1.2.5 PieChartGadget

2.1.2.5.1 PVGadgets_PieChart

Create/Update a PVGadgets PieChart Gadget.

Use:

PVGadgets_PieChart(#Gadget,#Image,10,10,100,20,-1,\$0000FF,\$00FFFF,35)

Progress

0 - 100 (representing %)

BackColor

\$0 - \$FFFFFF (RGB values of background color)

#SystemColor - Set Background Color of Form to System Form color

PieBack

\$0 - \$FFFFFF (RGB values of background pie color)

PieFront

\$0 - \$FFFFFF (RGB values of foreground pie color)

FontID (optional)

ID of loaded font to use

#SystemFont - Use Default System Font

FontColor (optional)

\$0 - \$FFFFFF (RGB values of font color)

Default is Black

Example:

```
#ChartGad=1
#ChartImg=1

If
OpenWindow(0,0,0,220,220,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"
Pie Chart")

    myfont=LoadFont(1,"Arial",20,#PB_Font_Bold)

    Repeat
        bar+1
        If bar>100:bar=0:EndIf
        PVGadgets_PieChart(#ChartGad,#ChartImg,10,10,200,200,-
            1,RGB(160,160,160),RGB(255,255,0),bar,myfont,RGB(0,0,150))

        Delay(10)
    Until WindowEvent()=#PB_Event_CloseWindow
    PVGadgets_FreePieChart(#ChartGad,#ChartImg)

EndIf
```

2.1.2.5.2 PVGadgets_FreePieChart

Frees Resources of Specified PieChart/Image.

Use:
PVGadgets_FreePieChart (#Gadget,#Image)

All resources are freed and PieChart can no longer be used.

2.1.2.6 Refresh

2.1.2.6.1 PVGadgets_Refresh

Forces the Specified Gadget to be Redrawn.

Use:
PVGadgets_Refresh(#Gadget)

Example:


```

#Window_Main=0
#Gadget_Main_Ok=1

#Gadget_Main_Canvas=2
#Image_Main_Canvas=2

If
OpenWindow( #Window_Main,175,0,184,172,#PB_Window_SystemMenu|#PB_Window_Scr
eenCentered,"Demo")

    PVGadgets_Canvas(#Gadget_Main_Canvas,#Image_Main_Canvas,15,25,150,100,1
2632256)
    ButtonGadget(#Gadget_Main_Ok,105,140,60,20,"Press Me")

    UseImage(#Image_Main_Canvas)
    StartDrawing(ImageOutput())
        DrawingMode(4)
        Box(10,10,130,80,$00FFFF)
        Box(20,20,110,60,$00FFFF)
    StopDrawing()
    PVGadgets_Refresh(#Gadget_Main_Canvas) ;<-- After drawing to the
Canvas, Refresh it so we can see results

Repeat
    EventID=WaitWindowEvent()
    If EventID=#PB_Event_Gadget
        If EventGadgetID()=#Gadget_Main_Ok
            UseImage(#Image_Main_Canvas)
            StartDrawing(ImageOutput())
                Circle(50,50,50,$FF0000)
                Box(60,60,70,30,$0000FF)
            StopDrawing()
            PVGadgets_Refresh(#Gadget_Main_Canvas) ;<-- After drawing to the
Canvas, Refresh it so we can see results
        EndIf
    EndIf
Until EventID=#PB_Event_CloseWindow

EndIf

```

2.1.2.6.2 PVGadgets_RefreshWindow

Forces the Specified Window to be Refreshed.

Use:

PVGadgets_RefreshWindow(#Window)

2.1.2.7 Misc Functions

2.1.2.7.1 Window Functions

2.1.2.7.1.1 PVGadgets_WindowFreeze

Freezes all Windows except the specified Window. Total number of Windows in the project must also be specified (this can be found in the Constant #WindowIndex)

Use:

PVGadgets_WindowFreeze(#Window_Main,#WindowIndex)

2.1.2.7.1.2 PVGadgets_WindowUnfreeze

Unfreezes all Windows and activates the specified Window. Total number of Windows in the project must also be specified (this can be found in the Constant #WindowIndex)

Use:

```
PVGadgets_WindowUnfreeze(#Window_Main,#WindowIndex)
```

2.1.2.7.1.3 PVGadgets_ToolWindow

Convert a standard Window to a ToolWindow.

Use:

```
PVGadgets_ToolWindow(#Window_Main)
```

2.1.2.7.1.4 PVGadgets_WindowTransparent

Set the Transparency on a Window (works with Win 2000/XP).

Use:

```
PVGadgets_WindowTransparent(#Window_Main,100)
```

Transparency

0 - 255 (0=invisible / 255=full visible)

2.1.2.7.1.5 PVGadgets_WindowTop

Force Window to always on top.

Use:

```
PVGadgets_WindowTop(#Window_Main)
```

2.1.2.7.1.6 PVGadgets_WindowReset

Reset Window to not always on top.

Use:

```
PVGadgets_WindowReset(#Window_Main)
```

2.1.2.7.2 Gadget Functions

2.1.2.7.2.1 PVGadgets_Splitter

Creates a SplitterGadget between two specified Gadgets. Result is a SplitterGadget container created with #PB_Any

Use:

```
NewGadget=PVX::PVGadgets_Splitter(#Gadget1,#Gadget2)
```

Flags

#PB_Splitter_Vertical

2.1.2.7.3 Path Functions

2.1.2.7.3.1 PVGadgets_CurDir

Returns Path of Current Directory.

Use:
CurDir\$=PVGadgets_CurDir()

Note:
Returned path always contains trailing backslash.

2.1.2.7.3.2 PVGadgets_TempDir

Returns Path of Windows Temporary Directory.

Use:
TempDir\$=PVGadgets_TempDir()

Note:
Returned path always contains trailing backslash.

2.1.2.7.3.3 PVGadgets_ExeDir

Returns Directory of Current Executable.

Use:
ExeDir\$=PVGadgets_ExeDir()

Note:
Returned path always contains trailing backslash.

2.1.2.7.4 ListIcon Functions

2.1.2.7.4.1 PVGadgets_CountListIconColumns

Counts the number of columns in a ListIconGadget.

Use:
Result=PVGadgets_CountListIconColumns(#Gadget)

Example:

```

#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",80)
    AddGadgetColumn(#List,2,"Column 3",50)

    totalcolumns=PVGadgets_CountListIconColumns(#List)
    MessageRequester("Result","Total Columns: "+Str(totalcolumns))

    oldcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    PVGadgets_SetListIconColumnWidth(#List,2,100)
    newcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    MessageRequester("Result","Old Width: "+Str(oldcolumnwidth)+Chr(10)+"New
    Width: "+Str(newcolumnwidth))

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf

```

2.1.2.7.4.2 PVGadgets_GetListIconColumnWidth

Returns the width of the selected column in a ListIconGadget.

Use:

Result=PVGadgets_GetListIconColumnWidth(#Gadget,Column)

Example:

```

#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",80)
    AddGadgetColumn(#List,2,"Column 3",50)

    totalcolumns=PVGadgets_CountListIconColumns(#List)
    MessageRequester("Result","Total Columns: "+Str(totalcolumns))

    oldcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    PVGadgets_SetListIconColumnWidth(#List,2,100)
    newcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    MessageRequester("Result","Old Width: "+Str(oldcolumnwidth)+Chr(10)+"New
    Width: "+Str(newcolumnwidth))

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf

```

2.1.2.7.4.3 PVGadgets_SetListIconColumnWidth

Sets the width of the selected column in a ListIconGadget.

Use:

Result=PVGadgets_SetListIconColumnWidth(#Gadget,Column,Width)

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",80)
    AddGadgetColumn(#List,2,"Column 3",50)

    totalcolumns=PVGadgets_CountListIconColumns(#List)
    MessageRequester("Result","Total Columns: "+Str(totalcolumns))

    oldcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    PVGadgets_SetListIconColumnWidth(#List,2,100)
    newcolumnwidth=PVGadgets_GetListIconColumnWidth(#List,2)
    MessageRequester("Result","Old Width: "+Str(oldcolumnwidth)+Chr(10)+"New
    Width: "+Str(newcolumnwidth))

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.4 PVGadgets_JustifyListIconColumn

Modify the justification of the selected column in a ListIconGadget.

Use:

PVGadgets_JustifyListIconColumn(#Gadget,Column,Flag)

Flag:

#Left - Left (Default)
 #Right - Right
 #Center - Center

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    PVGadgets_JustifyListIconColumn(#List,0,#Center) ;justify column 0 to
center
    PVGadgets_JustifyListIconColumn(#List,1,#Left) ;justify column 1 to
left
    PVGadgets_JustifyListIconColumn(#List,2,#Right) ;justify column 2 to
right

    For tmp=0 To 20
        AddGadgetItem(#List,-
1,"TestA_"+Str(tmp)+Chr(10)+"TestB_"+Str(tmp)+Chr(10)+"TestC_"+Str(tmp)
)
    Next

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.5 PVGadgets_LastListIconRow

Forces the last row of a ListIconGadget into view.

Use:

PVGadgets_LastListIconRow(#Gadget)

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"TestA_"+Str(tmp)+Chr(10)+"TestB_"+Str(tmp)+Chr(10)+"TestC_"+Str(tmp)
        )
    Next

    PVGadgets_LastListIconRow(#List) ;last row is forced into view

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.6 PVGadgets_SelectListIconRow

Forces the selected row of a ListIconGadget into view.

Use:

PVGadgets_SelectListIconRow(#Gadget,Row)

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"TestA_"+Str(tmp)+Chr(10)+"TestB_"+Str(tmp)+Chr(10)+"TestC_"+Str(tmp)
        )
    Next

    PVGadgets_SelectListIconRow(#List,18) ;row 18 is selected and forced
into view

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.7 PVGadgets_CreateListIconImageList

Creates an ImageList to which icons will be added for using in a ListIconGadget.

Use:

hImageList=PVGadgets_CreateListIconImageList(#Gadget)

Returns handle to the created ImageList to be used with [PVGadgets_AddListIconImageList](#)

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"A_"+Str(tmp)+Chr(10)+"B_"+Str(tmp)+Chr(10)+"C_"+Str(tmp))
    Next

    hImageList=PVGadgets_CreateListIconImageList(#List)
    HelpIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(0,"help.ico")
    )
    InfoIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(1,"info.ico")
    )

    PVGadgets_ChangeListIconImage(#List,3,1,HelpIcon)
    PVGadgets_ChangeListIconImage(#List,5,2,InfoIcon)

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.8 PVGadgets_AddListIconImageList

Add an icon to an ImageList.

Use:

ImageID=PVGadgets_AddListIconImageList(hImageList,hImage)

Returns an ImageID to be used with [PVGadgets_ChangeListIconImage](#)

Example:


```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"A_"+Str(tmp)+Chr(10)+"B_"+Str(tmp)+Chr(10)+"C_"+Str(tmp))
    Next

    hImageList=PVGadgets_CreateListIconImageList(#List)
    HelpIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(0,"help.ico"
    ))
    InfoIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(1,"info.ico"
    ))

    PVGadgets_ChangeListIconImage(#List,3,1,HelpIcon)
    PVGadgets_ChangeListIconImage(#List,5,2,InfoIcon)

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.9 PVGadgets_FreeListIconImageList

Free a previously created ImageList.

Use:
PVGadgets_FreeListIconImageList(hImageList)

Example:

```

#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"A_"+Str(tmp)+Chr(10)+"B_"+Str(tmp)+Chr(10)+"C_"+Str(tmp))
    Next

    hImageList=PVGadgets_CreateListIconImageList(#List)
    HelpIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(0,"help.ico")
    )
    InfoIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(1,"info.ico")
    )

    PVGadgets_ChangeListIconImage(#List,3,1,HelpIcon)
    PVGadgets_ChangeListIconImage(#List,5,2,InfoIcon)

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
    PVGadgets_FreeListIconImageList(hImageList)
EndIf

```

2.1.2.7.4.10 PVGadgets_ChangeListIconImage

Change/assign an Icon to a specified Row/Column of an ListIconGadget.

Use:

PVGadgets_ChangeListIconImage(#List,5,2,InfoIcon)

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
    1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"A_"+Str(tmp)+Chr(10)+"B_"+Str(tmp)+Chr(10)+"C_"+Str(tmp))
    Next

    hImageList=PVGadgets_CreateListIconImageList(#List)
    HelpIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(0,"help.ico"
    ))
    InfoIcon=PVGadgets_AddListIconImageList(hImageList,LoadImage(1,"info.ico"
    ))

    PVGadgets_ChangeListIconImage(#List,3,1,HelpIcon)    ;changes row 3 column
    1 to HelpIcon
    PVGadgets_ChangeListIconImage(#List,5,2,InfoIcon)    ;changes row 5
    column 2 to InfoIcon

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf
```

2.1.2.7.4.11 PVGadgets_ListIconTitle

Change the Title Text of specified Column (column index starts at 0).

Use:

PVGadgets_ListIconTitle(#Gadget,Column,"New Title")

Example:

```

#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"")

    ListIconGadget(#List,0,0,400,300,"Column
1",150,#PB_ListIcon_GridLines|#PB_ListIcon_FullRowSelect)
    AddGadgetColumn(#List,1,"Column 2",100)
    AddGadgetColumn(#List,2,"Column 3",100)

    For tmp=0 To 20
        AddGadgetItem(#List,-
            1,"TestA_"+Str(tmp)+Chr(10)+"TestB_"+Str(tmp)+Chr(10)+"TestC_"+Str(tmp)
        )
    Next

    PVGadgets_ListIconTitle(#List,1,"New Title") ;Change text of second
column

    Repeat:Until WaitWindowEvent()=#PB_Event_CloseWindow
EndIf

```

2.1.2.7.5 Format Functions

2.1.2.7.5.1 PVGadgets_FormatNumber

Formats a number string.

Use:

Result\$=PVGadgets_FormatNumber(Number.s,Group.l,DecDig.l,DecSep.s,GrpSep.s,Neg.l)

Number.s - Number to format (in string format)

Group.l - group in bunches of

DecDig.l - number of decimal places

DecSep.s - Decimal separator character

GrpSep.s - Group separator Character

Neg.l - Format negative values

0 = (000)

1 = -000

2 = - 000

3 = 000-

4 = 000 -

Example:

```
Debug PVGadgets_FormatNumber("3343200",3,2,".",",",0)
```

returns:

3,343,200.00

2.1.2.7.6 Mutex Functions

2.1.2.7.6.1 PVGadgets_StartOnce

Allow only one instance of your application to start

```
Result=PVGadgets_StartOnce("MyApp")
```

2.1.2.7.7 Error Functions

2.1.2.7.7.1 PVGadgets_APIError

Return Last Error Message from API Call

```
Result$=PVGadgets_APIError()
```

2.1.3 PureSkin

PureSkin functions allow you to easily apply skins to your Windows forms.

[PureSkin](#)
[PureSkinMem](#)
[PureSkinHide](#)

Skins are created from standard BMP files using the PureSkin Maker in PureVision.

Copyright © 2002 - 2022, Reel Media Productions™
"PureVision" and "PureSkin" are trademarks of Reel Media Productions. All rights are reserved.

2.1.3.1 PureSkin

Skin the Selected Window from File.

Use:
`PureSkin(#Window,#Gadget,#Image,"MySkin.pvs",#True)`

Flags: #True = Center Window

Returns 1 if success.

Example:

```
#Main=0
#Gadget_Skin=1
#Image_Skin=2
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo")

    PureSkin(#Main,#Gadget_Skin,#Image_Skin,"MySkin.pvs",#True)    ;<-- Loads
    PureSkin from File
    PureSkinHide(#Main,#False)

quitMain=0
Repeat
    Select WaitWindowEvent()
        Case #PB_Event_CloseWindow
            quitMain=1
    EndSelect
Until quitMain

EndIf
End
```

2.1.3.2 PureSkinMem

Skin the Selected Window from Memory.

Use:

```
PureSkinMem(#Window,#Gadget,#Image,?_MySkin,1)
```

Flags: True = Center Window

Returns 1 if success.

Example:

```
#Main=0
#Gadget_Skin=1
#Image_Skin=2
If
OpenWindow( #Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo" )

    PureSkinMem(#Main,#Gadget_Skin,#Image_Skin,?_MySkin,#True)    ;<-- Loads
    PureSkin from Memory
    PureSkinHide(#Main,#False)

quitMain=0
Repeat
    Select WaitWindowEvent()
        Case #PB_Event_CloseWindow
            quitMain=1
    EndSelect
Until quitMain

EndIf
End

DataSection
    _MySkin:IncludeBinary "MySkin.pvs"
EndDataSection
```

2.1.3.3 PureSkinHide

Hides or Displays a PureSkin.

Use:

PureSkinHide(#Window,#False)

Flags: #True = Hide Window

Example:

```
#Main=0
#Skin=1
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo")

    PureSkin(#Main,#Skin,"MySkin.pvs",#True) ;<-- Loads PureSkin from File
    PureSkinHide(#Main,#False)

quitMain=0
Repeat
    Select WaitWindowEvent()
        Case #PB_Event_CloseWindow
            quitMain=1
    EndSelect
Until quitMain

EndIf
End
```

2.1.4 PurePoint

PurePoint functions allow you to create your own mouse pointers and use them in your applications.

[PurePoint](#)
[PurePointMem](#)
[UsePurePoint](#)
[SysPurePoint](#)
[FreePurePoint](#)
[ResetPurePoint](#)
[PurePointChild](#)
[PurePointX](#)
[PurePointY](#)
[PurePointXX](#)
[PurePointYY](#)

PurePoints are created using our special [PurePoint Maker](#) software.

Copyright © 2002 - 2022, Reel Media Productions™
"PureVision" and "PurePoint" are trademarks of Reel Media Productions. All rights are reserved.

2.1.4.1 PurePoint

Loads a PurePoint from File.

Use:
hPoint=PurePoint("Cross.pvp")

Returns Handle to PurePoint.

Example:

```
#Main=0
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
    hPoint=PurePoint("Cross.pvp") ;<-- Loads PurePoint from File

    quitMain=0
    Repeat
        UsePurePoint(hPoint)

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End
```

2.1.4.2 PurePointMem

Loads a PurePoint from Memory.

Use:
hPoint=PurePointMem(MemoryAddress)

Returns Handle to PurePoint.

Example:

```

#Main=0
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
  hPoint=PurePointMem(?_Cross) ;<-- Loads PurePoint from Memory (Included Binary)

  quitMain=0
  Repeat
    UsePurePoint(hPoint)

    Select WaitWindowEvent()
      Case #PB_Event_CloseWindow
        quitMain=1
    EndSelect
  Until quitMain

  FreePurePoint(hPoint)
EndIf
End

DataSection
  _Cross:IncludeBinary "Cross.pvp"
EndDataSection

```

2.1.4.3 UsePurePoint

Uses/Displays a PurePoint that has previously been loaded.

Use:
UsePurePoint(hPoint)

Example:

```

#Main=0
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
  hPoint=PurePoint("Cross.pvp")

  quitMain=0
  Repeat
    UsePurePoint(hPoint) ;<-- Use and Display PurePoint

    Select WaitWindowEvent()
      Case #PB_Event_CloseWindow
        quitMain=1
    EndSelect
  Until quitMain

  FreePurePoint(hPoint)
EndIf
End

```

2.1.4.4 SysPurePoint

Sets System Cursor to specified PurePoint.

Use:
SysPurePoint(hPoint)

Returns 0 if Fail.

Example:

```
#Main=0
If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
  hPoint=PurePoint("Cross.pvp")
  SysPurePoint(hPoint) ;<-- Sets System Cursor to PurePoint

  quitMain=0
  Repeat
    Select WaitWindowEvent()
      Case #PB_Event_CloseWindow
        quitMain=1
    EndSelect
  Until quitMain

  FreePurePoint(hPoint)
  ResetPurePoint()
EndIf
End
```

2.1.4.5 FreePurePoint

Free PurePoint Resources of specified Handle.

Use:
FreePurePoint(hPoint)

Returns 0 if success.

Example:

```

#Main=0

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
  hPoint=PurePoint("Cross.pvp")

  quitMain=0
  Repeat
    UsePurePoint(hPoint)

    Select WaitWindowEvent()
      Case #PB_Event_CloseWindow
        quitMain=1
    EndSelect
  Until quitMain

  FreePurePoint(hPoint) ;<-- Free PurePoint Resources
EndIf
End

```

2.1.4.6 ResetPurePoint

Resets a System Cursor to the Default Arrow.

Use:
ResetPurePoint()

Example:

```

#Main=0

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)
  hPoint=PurePoint("Cross.pvp")
  SysPurePoint(hPoint)

  quitMain=0
  Repeat
    Select WaitWindowEvent()
      Case #PB_Event_CloseWindow
        quitMain=1
    EndSelect
  Until quitMain

  FreePurePoint(hPoint)
  ResetPurePoint() ;<-- Resets System Cursor
EndIf
End

```

2.1.4.7 PurePointChild

Used to Identify what Gadget a PurePoint is Over.

Use:

```
hGadget=PurePointChild(Window)
```

Returns Handle of Gadget.

Example:

```
#Main=0
#List=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)

    ListIconGadget(#List,10,10,300,200,"Demo",100)

    hPoint=PurePoint("Cross.pvp")

    quitMain=0
    Repeat
        If PurePointChild(#Main)=GadgetID(#List) ;<-- Checks is PurePoint is over ListIcon
            UsePurePoint(hPoint) ;<-- Displays PurePoint only over ListIcon
        EndIf

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End
```

2.1.4.8 PurePointX

Returns Horizontal Position of PurePoint in specified Window.

Use:

```
X=PurePointX(Window)
```

Example:

```
#Main=0
#Text=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)

    TextGadget(#Text,10,10,200,20,"")

    hPoint=PurePoint("PP.pvp")

    quitMain=0
    Repeat
        UsePurePoint(hPoint)

        x=PurePointX(#Main) ;<-- Return Horizontal Position of PurePoint
        y=PurePointY(#Main)
        SetGadgetText(#Text,"X="+Str(x)+" Y="+Str(y))

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End
```

2.1.4.9 PurePointY

Returns Vertical Position of PurePoint in specified Window.

Use:
Y=PurePointY(Window)

Example:

```
#Main=0
#Text=1

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)

    TextGadget(#Text,10,10,200,20,"")

    hPoint=PurePoint("PP.pvp")

    quitMain=0
    Repeat
        UsePurePoint(hPoint)

        x=PurePointX(#Main)
        y=PurePointY(#Main) ;<-- Return Vertical Position of PurePoint
        SetGadgetText(#Text,"X="+Str(x)+" Y="+Str(y))

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End
```

2.1.4.10 PurePointXX

Returns Horizontal Position of PurePoint in specified Gadget.

Use:

X=PurePointXX(Gadget)

Example:

```

#Main=0
#Text=1
#Button=3

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)

    TextGadget(#Text,10,10,200,20,"")
    ButtonGadget(#Button,10,50,80,20,"Demo")

    hPoint=PurePoint("PP.pvp")

    quitMain=0
    Repeat
        UsePurePoint(hPoint)

        x=PurePointXX(#Button)    ;<-- Return Horizontal Position of PurePoint on Button
        y=PurePointYY(#Button)

        If x<0 Or y<0
            SetGadgetText(#Text,"Null")
        Else
            SetGadgetText(#Text,"X="+Str(x)+" Y="+Str(y))
        EndIf

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End

```

2.1.4.11 PurePointYY

Returns Vertical Position of PurePoint in specified Gadget.

Use:
Y=PurePointYY(Gadget)

Example:


```
#Main=0
#Text=1
#Button=3

If
OpenWindow(#Main,0,0,400,300,#PB_Window_SystemMenu|#PB_Window_ScreenCentered,"Demo"
)

    TextGadget(#Text,10,10,200,20,"")
    ButtonGadget(#Button,10,50,80,20,"Demo")

    hPoint=PurePoint("PP.pvp")

    quitMain=0
    Repeat
        UsePurePoint(hPoint)

        x=PurePointXX(#Button)
        y=PurePointYY(#Button) ;<-- Return Vertical Position of PurePoint on Button

        If x<0 Or y<0
            SetGadgetText(#Text,"Null")
        Else
            SetGadgetText(#Text,"X="+Str(x)+" Y="+Str(y))
        EndIf

        Select WaitWindowEvent()
            Case #PB_Event_CloseWindow
                quitMain=1
        EndSelect
    Until quitMain

    FreePurePoint(hPoint)
EndIf
End
```

2.2 PVGadgets QuickStart

If you are having problems with any functions from the PVGadgets Module, make sure your Main Events Loop contains the proper functions and code to initialize the PVGadgets and make them work.

If the Gadget or Form you have created needs any of the following, simply copy and paste it into your project.

```
;This must be called at the start of your program if special Gadget  
Coloring is used
```

```
Procedure WindowCallback(WindowID,Message,wParam,lParam)  
  ReturnValue=#PB_ProcessPureBasicEvents  
  If Message=#WM_CTLCOLORSTATIC Or Message=#WM_CTLCOLOREDIT Or  
  Message=#WM_CTLCOLORLISTBOX  
    ReturnValue=PVDynamic_ColorGadget(lParam,wParam)  
  EndIf  
  ProcedureReturn ReturnValue  
EndProcedure
```

```
;This must be called after your Form has been created  
SetWindowCallback(@WindowCallback())
```

This is inserted into `_Constants.pb` by PureVision. If you want to use any of the Functions in the PVGadgets Module without using PureVision, you must place this at the beginning of your Project Source Code...

```
XIncludeFile "C:/PureVision/Modules/Module_PVGadgets.pbi" ;<-- Modify as needed for path of  
Modules Folder
```

Part



3 PVPlugins User Library

PureVision provides a simple API accessible through the PVPlugins Module. This allows for PureVision [Plugins](#) to be created to perform various tasks and these Plugins will appear in the PureVision Main MenuBar.

Examples are provided in the "PureVision64\Plugins" folder.

3.1 PV_PluginsInfo

Allows PureVision to identify the Plugin as a valid and allows you to specify internal information.

Name

Name of your Plugin

HiVersion

High version number

LoVersion

Low version number

Info

Short description of your plugin

Copyright

Copyright information for plugin

Returns 0 if failure.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.2 PV_PluginsWindowName

Sets the ID Name for the Plugin Window.

This function MUST be called before calling any of the other PluginWindow functions.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.3 PV_PluginsWindowTitle

Sets the Window Title text for the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.4 PV_PluginsWindowX

Sets the X position of the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.5 PV_PluginsWindowY

Sets the Y position of the Plugin Window.

Example:


```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.6 PV_PluginsWindowW

Sets the width of the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.7 PV_PluginsWindowH

Sets the height of the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.8 PV_PluginsWindowBGColor

Sets the background color of the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.9 PV_PluginsWindowTab

Sets the number of Statusbar Tabs for the Plugin Window.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.10 PV_PluginsGadgetType

Creates and adds the specified Gadget Type to the Plugin Window.

This function MUST be called before calling any of the other PluginGadget functions.

Type

TextGadget
ButtonGadget
StringGadget

...

any valid Gadget available in PureVision (excluding Container type Gadgets)

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.11 PV_PluginsGadgetName

Sets the ID Name of the Plugin Gadget.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.12 PV_PluginsGadgetText

Sets the Gadget Text of the Plugin Gadget.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.13 PV_PluginsGadgetX

Sets the X position of the Plugin Gadget.

Example:


```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.14 PV_PluginsGadgetY

Sets the Y position of the Plugin Gadget.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.15 PV_PluginsGadgetW

Sets the width of the Plugin Gadget.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(0,0,150)

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.16 PV_PluginsGadgetH

Sets the height of the Plugin Gadget.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.17 PV_PluginsGadgetTabs

Sets the number of tabs for the Plugin Gadget.

This is used for ListIconGadget

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

PV_Plugins()
```

3.18 PV_PluginsMenuItemName

Sets the ID Name of the MenuItem.

If name is <BREAK> then a Separator Bar is added

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Menuitem Test","2011 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)

  PV_PluginsMenuitemName("File")
  PV_PluginsMenuitemText("File")
  PV_PluginsMenuitemPos(0)

  PV_PluginsMenuitemName("New")
  PV_PluginsMenuitemText("New")
  PV_PluginsMenuitemPos(1)
  PV_PluginsMenuitemName("<BREAK>")
  PV_PluginsMenuitemPos(1)
  PV_PluginsMenuitemName("Exit")
  PV_PluginsMenuitemText("Exit")
  PV_PluginsMenuitemPos(1)
EndIf

PV_Plugins()
```

3.19 PV_PluginsMenuitemText

Sets the Displayed Text of the Menuitem.

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Menuitem Test","2011 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)

  PV_PluginsMenuitemName("File")
  PV_PluginsMenuitemText("File")
  PV_PluginsMenuitemPos(0)

  PV_PluginsMenuitemName("New")
  PV_PluginsMenuitemText("New")
  PV_PluginsMenuitemPos(1)
  PV_PluginsMenuitemName("<BREAK>")
  PV_PluginsMenuitemPos(1)
  PV_PluginsMenuitemName("Exit")
  PV_PluginsMenuitemText("Exit")
  PV_PluginsMenuitemPos(1)
EndIf

PV_Plugins()
```

3.20 PV_PluginsMenuitemPos

Sets the Position of the Menuitem.

Position:
0 = Main Section
1 = Sub Section

Example:

```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Menuitem Test","2011 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)

  PV_PluginsMenuItemName("File")
  PV_PluginsMenuItemText("File")
  PV_PluginsMenuItemPos(0)

  PV_PluginsMenuItemName("New")
  PV_PluginsMenuItemText("New")
  PV_PluginsMenuItemPos(1)
  PV_PluginsMenuItemName("<BREAK>")
  PV_PluginsMenuItemPos(1)
  PV_PluginsMenuItemName("Exit")
  PV_PluginsMenuItemText("Exit")
  PV_PluginsMenuItemPos(1)
EndIf

PV_Plugins()
```

3.21 PV_Plugins

This function is called at the end of your Plugin, enabling communication between the Plugin and PureVision.

Example:


```
XIncludeFile "../Modules/Module_PVPlugins.pbi" ;<-- Modify as needed for path of Modules Folder

Result=PV_PluginsInfo("Test",1,2,"Simple Test Plugin","2005 Reel Media Productions")

If Result
  PV_PluginsWindowName("TestForm")
  PV_PluginsWindowTitle("Created with Plugin")
  PV_PluginsWindowX(80)
  PV_PluginsWindowY(80)
  PV_PluginsWindowW(300)
  PV_PluginsWindowH(200)
  PV_PluginsWindowBGColor(RGB(0,0,150))

  PV_PluginsGadgetType("ButtonGadget")
  PV_PluginsGadgetName("TestButton")
  PV_PluginsGadgetText("Click Here")
  PV_PluginsGadgetX(10)
  PV_PluginsGadgetY(10)
  PV_PluginsGadgetW(80)
  PV_PluginsGadgetH(20)
EndIf

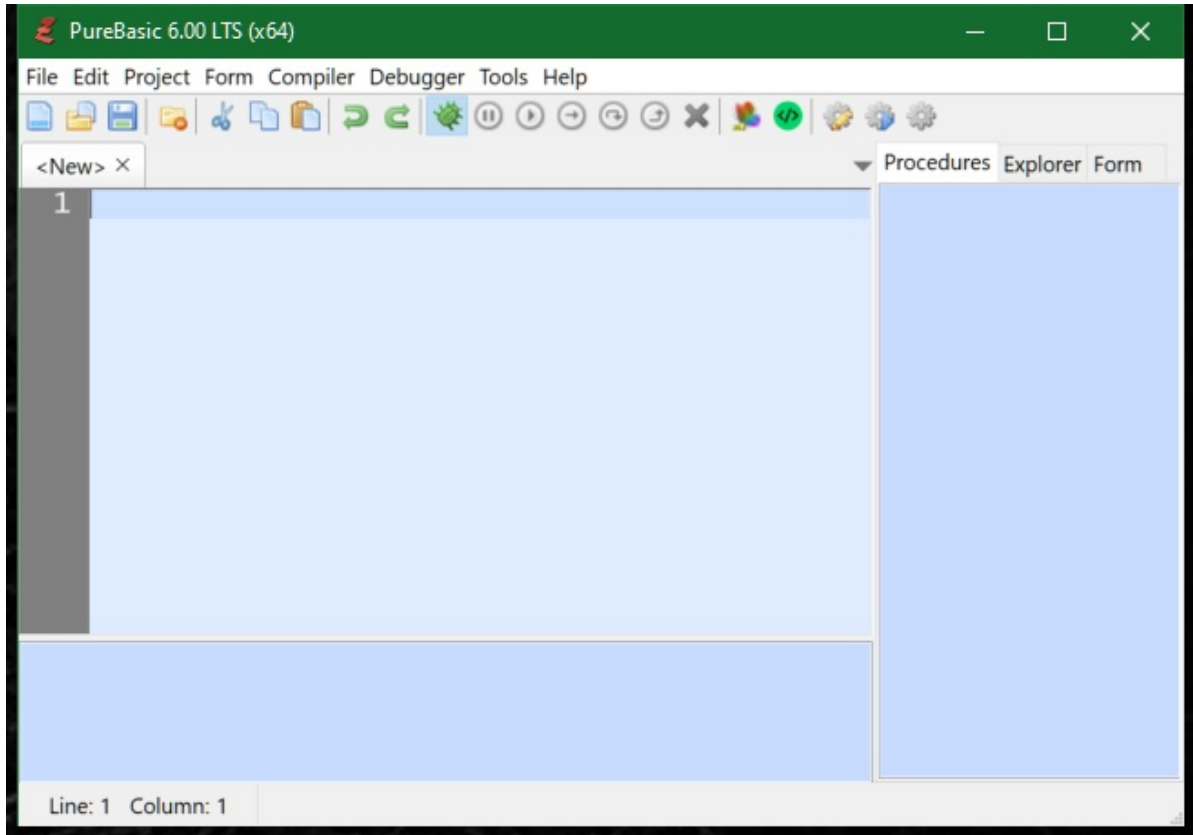
PV_Plugins()
```

Part

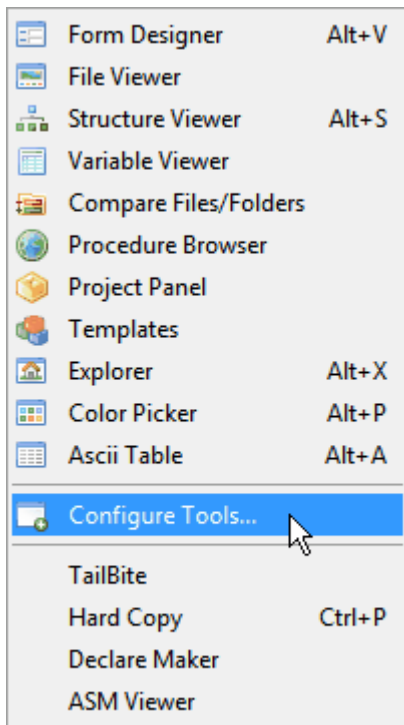
IV

4 PureBasic Integration

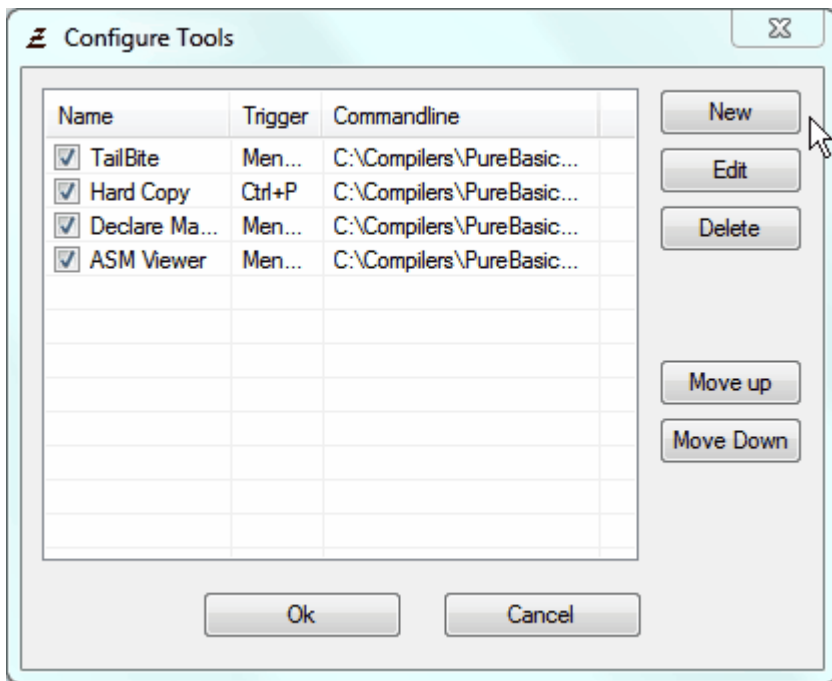
To integrate PureVision into the PureBasic 6.0+ IDE MenuBar, select Tools from the Main MenuBar.



Select Configure Tools from the Drop Down List.

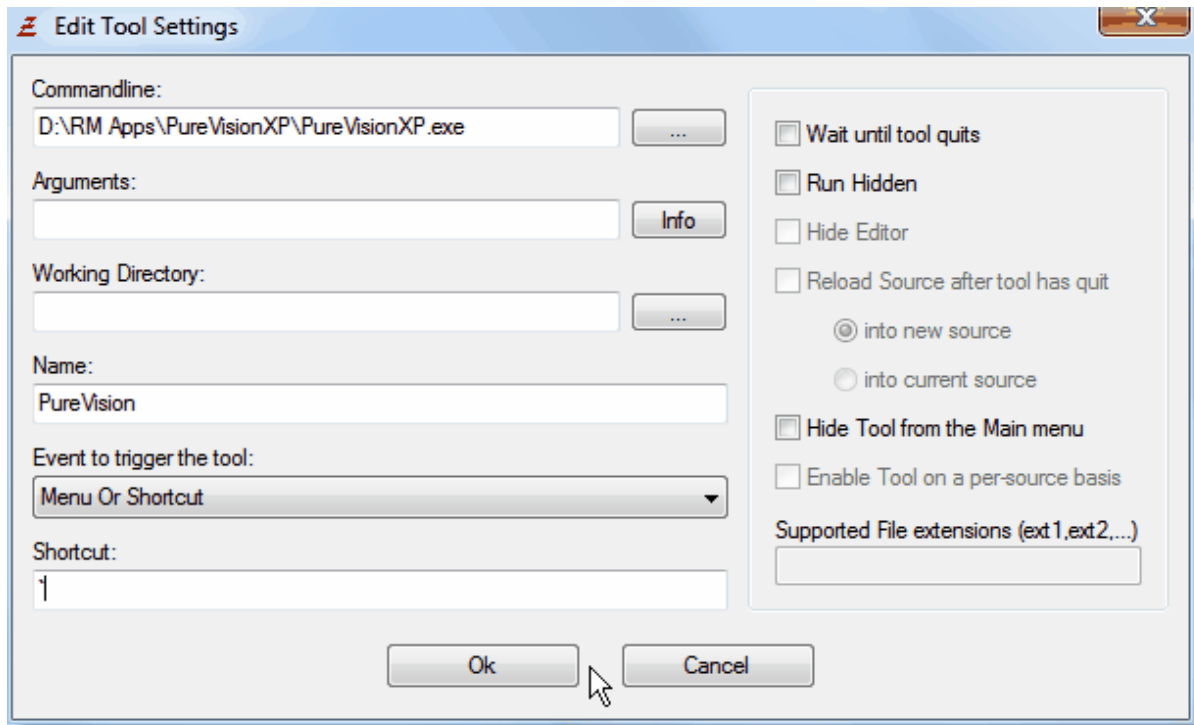


Select New to add a New Tool.

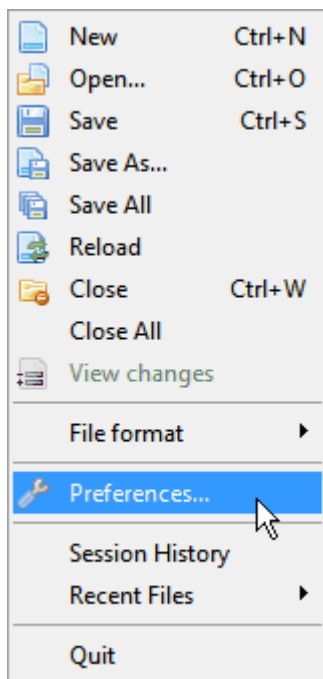


Locate the folder containing PureVision and select the main PureVision executable.
Enter the name of the Form Designer under Name and select a Shortcut key if you wish to use a shortcut to launch PureVision.

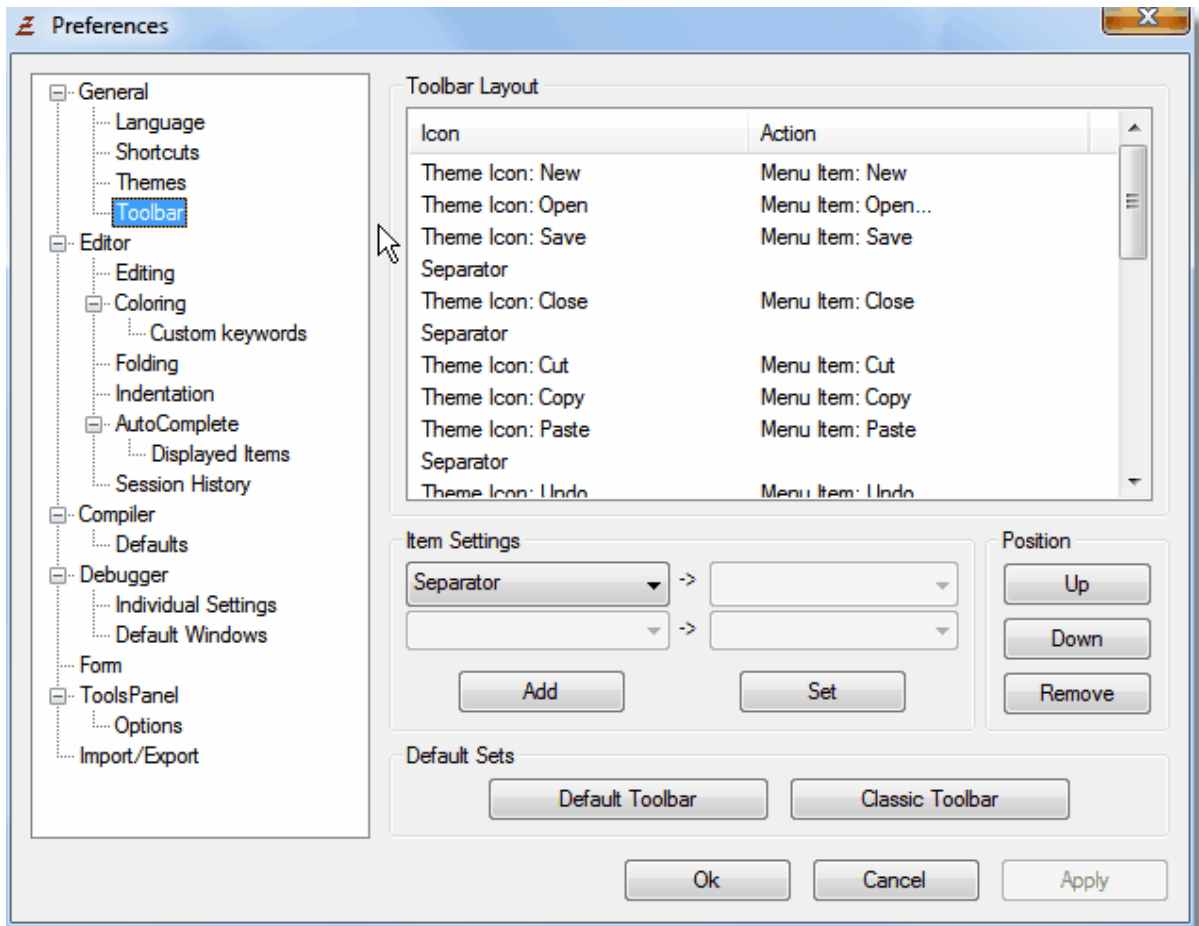
Select OK when you are done to go back to the main screen.



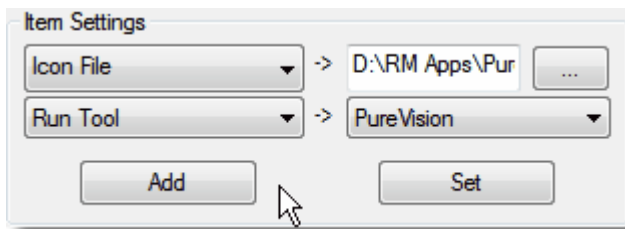
From the Main MenuBar under File select Preferences.



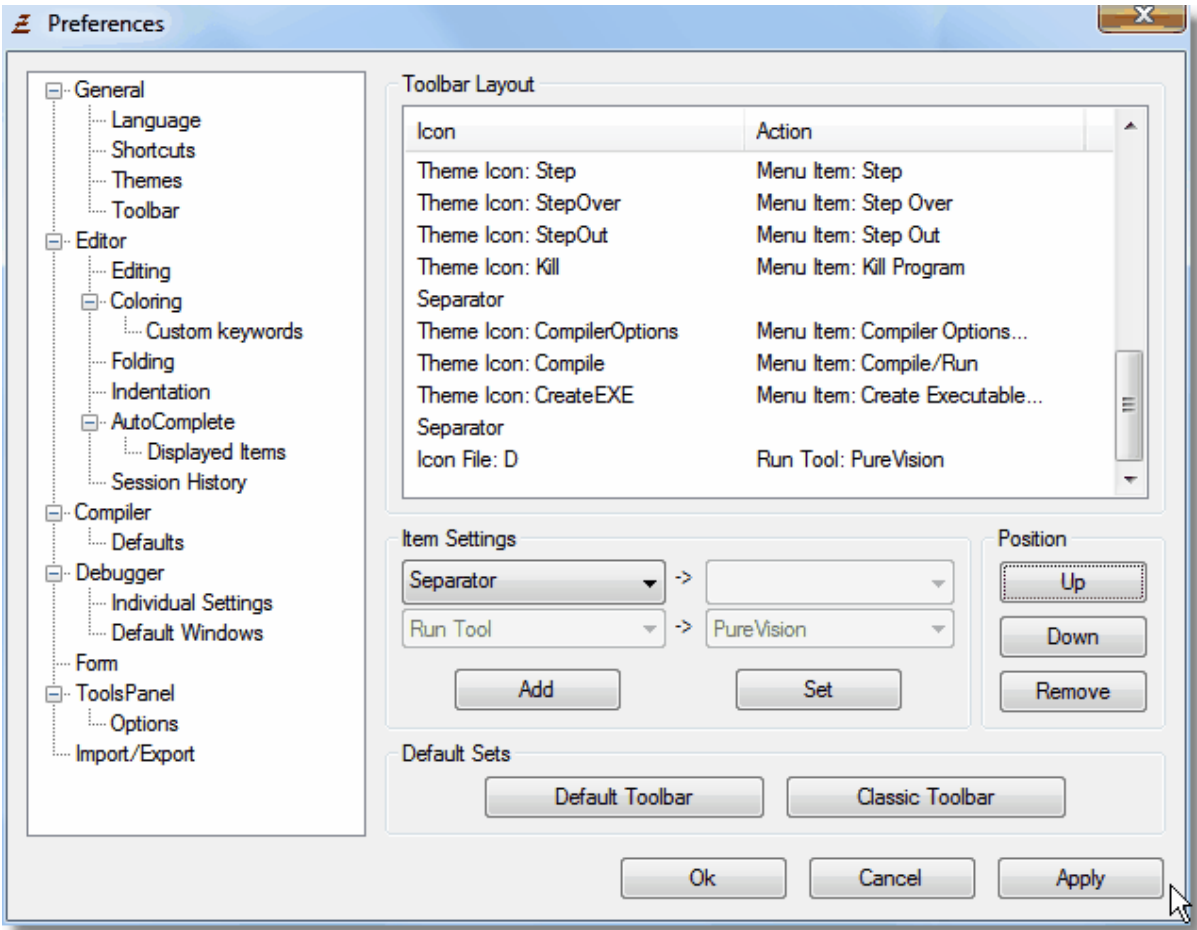
Select Toolbar in the Preferences Window.



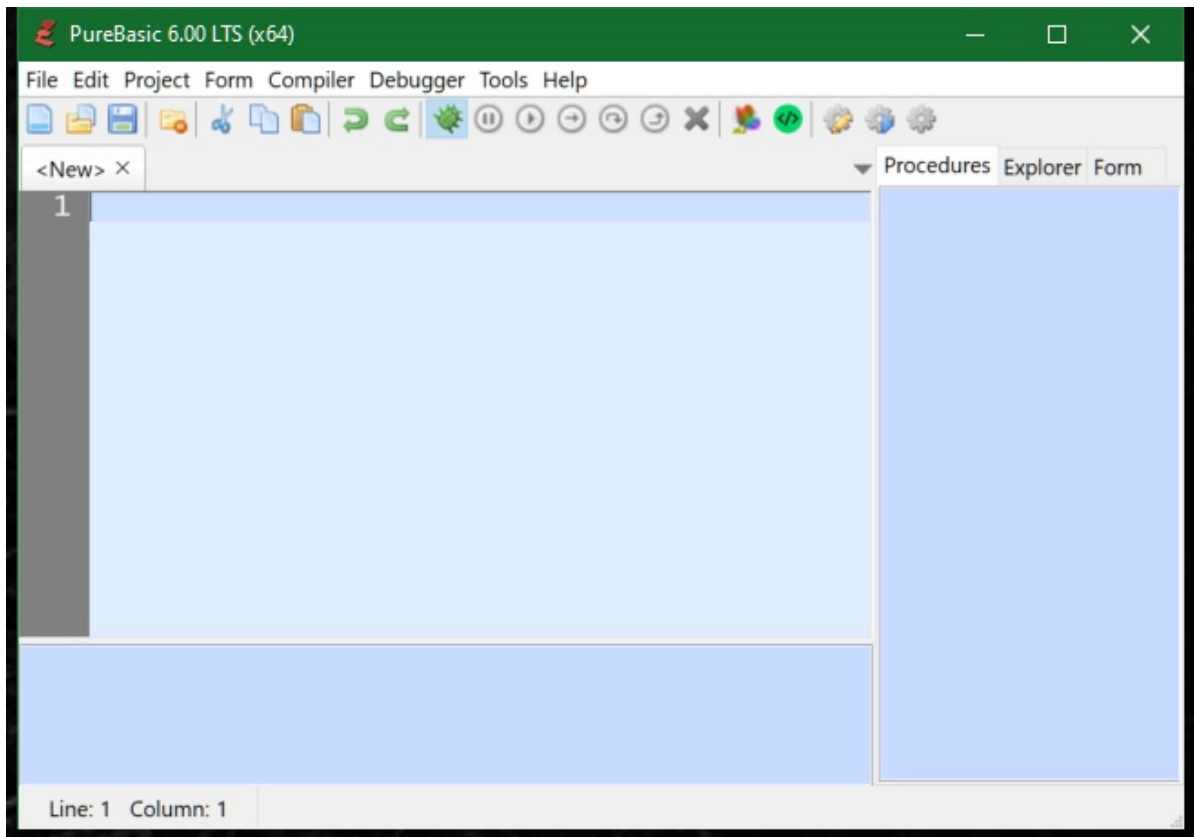
Under Item Settings, select Icon File and choose the icon you wish to represent PureVision in the Main MenuBar.
 Select Run Tool and then select PureVision.
 Press Add when you are complete.



Press the Apply button to save your new settings.



The Main MenuBar will now contain a new Icon which will launch PureVision when clicked or when you press the appropriate Shortcut Key.



Part

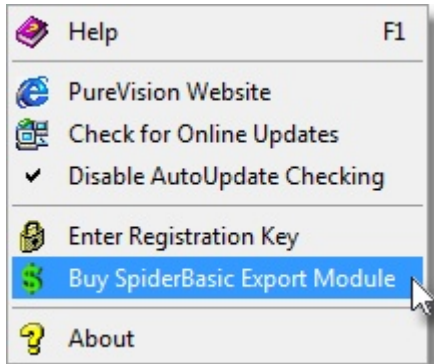


5 SpiderBasic Export Plugin

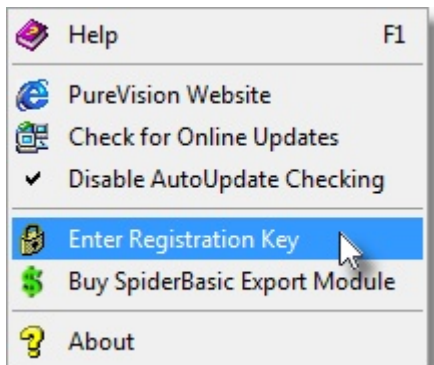
PureVision has the ability to export source code compatible with [Spider Basic](#).

If PureVision was purchased bundled with the SpiderBasic Export Module, exporting to SpiderBasic will be activated when your Registration Key has been entered.

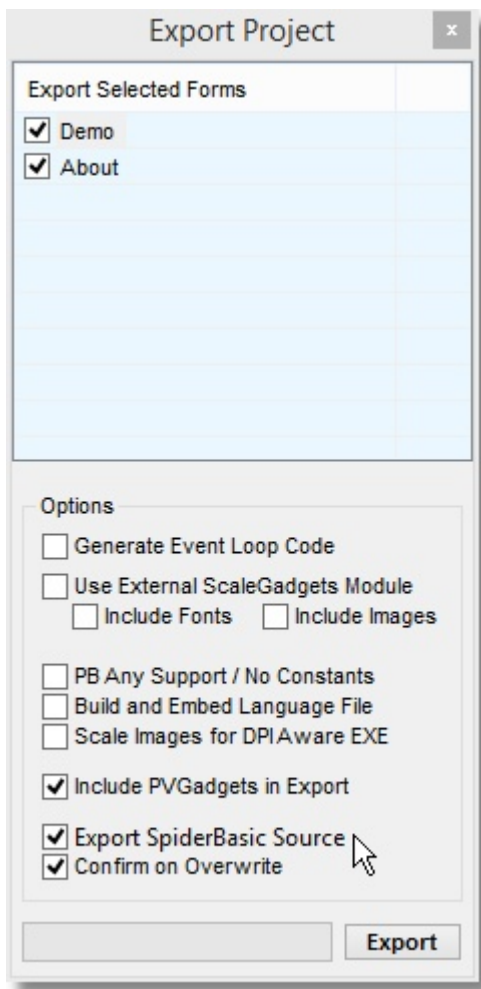
If you are already a Registered User of PureVision, you can purchase the SpiderBasic Export Module separately for only **\$9.95** by selecting Help from the Main MenuBar and selecting Buy SpiderBasic Export Module.



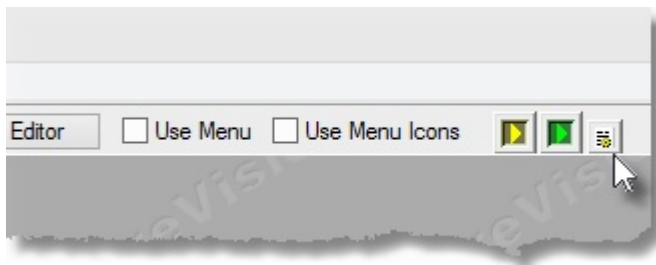
If the SpiderBasic Export Module has been purchased separately, you will need to enter your new Registration Key by selecting Help from the MenuBar and selecting Enter Registration Key.



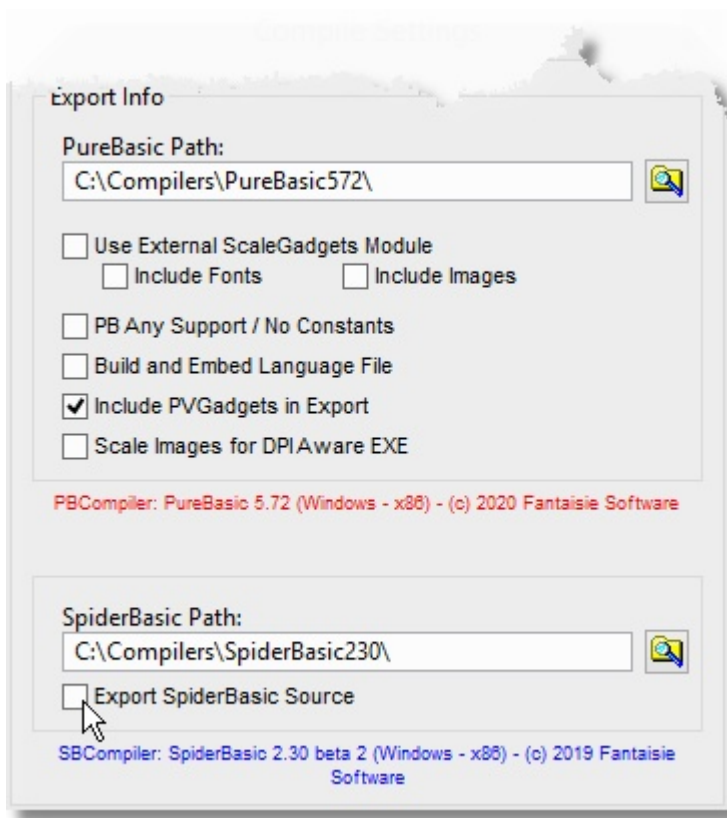
When exporting your project, you will have the option to Export SpiderBasic Code. If this option is checked your code will be exported and saved in the SpiderBasic .sb format.



When working in PureVision make sure you set up the Compiler Settings by opening the Compiler Settings Window.



Enter the location of where SpiderBasic has been installed on your Hard Drive.
If the correct folder has been entered, the current version of SpiderBasic will be displayed.

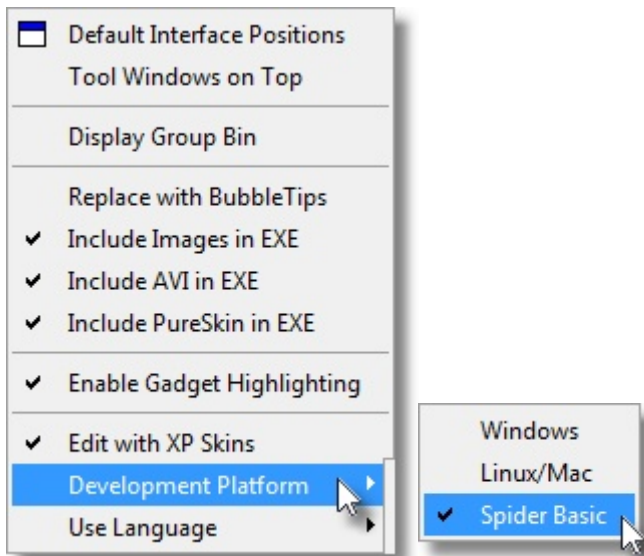


To preview exported code and see how it will compile in SpiderBasic, make sure **Export SpiderBasic Source** is checked.

If this is not checked, source code will be exported and previewed in PureBasic.

When designing Forms for use with SpiderBasic, it is best tell PureVision that your Development Platform is SpiderBasic.

This is done by selecting Forms from the PureVision MenuBar, selecting Development Platform and choosing Spider Basic.



This will remove all Gadgets from the [Gadget Creator](#) ToolBar that are not compatible with SpiderBasic.

NOTE: (SpiderBasic does not support these flags)

- Do not use any API flags when creating Gadgets
- Do not use MinimizeGadget or MaximizeGadget flag when creating Windows
- SpiderBasic only supports Dynamic Resizing from the PVGadgets Module

Part



6 History

Version 6.10

- Compiled with latest PureBasic 6.10 x64
- Added WebView Gadget
- Added more options when Managing Fonts
- Added more flexibility to BubbleTips
- Bug Fixes

Version 6.04

- Compiled with latest PureBasic 6.04 x64
- Added Lock/Unlock Gadget to prevent accidentally moving or resizing a Gadget

Version 6.03

- Compiled with latest PureBasic 6.03 x64
- Added Code Inject option for MenuBar and ToolBar
- Added Lock Tools option to prevent movement of Gadget Create, Gadget Properties & Gadget Tree Window
- Added DEFAULT commandline flag to start PureVision with default Window positions (same as pressing keyboard HOME button in PureVision)
- Added Named Enumerations to each Constant Group in the "Constants.pb" file
- Fixed WindowsCallback not being included in Main Loop when needed

Version 6.01

- Added ability to include and use up to 6 [Custom Gadgets](#) in PureVision
- Added [Override Gadget Settings](#) (X,Y,W,H) when adding Gadgets to a Form
- Added additional Gadget EventTypes & Code Inject fields for Exported Projects
- Fixed [PureSkin Maker](#) color picker showing incorrect coordinates
- Fixed [PureSkin Maker](#) saving corrupt PVS files when using standard DPI
- Fixed Font attributes missing when using Copy/Paste/Duplicate
- Fixed Font Requester not appearing when using the [Font Manager](#) Tool
- Fixed invalid code when exporting a [ShortcutGadget](#) with no flags selected
- Removed unneeded data from Saved Projects

Version 6.00f

- Fixed small Main MenuBar bug

Version 6.00e

- Added ability to Save multiple individual Forms at one time under "File/[Save Current Form](#)"
- Added ability to Merge multiple individual Forms at one time under "[Merge Form with Project](#)"
- Added Simple PureBasic Form/Gadget Code Importer [Plugin](#)
- Updated [PureTrans](#) software
- Fixed Main PureVision Display from Freezing if [Plugin](#) takes to long to respond

Version 6.00d

- Fixed Copy/Paste bug not copying Code Inject properties

- Fixed Export bug with DirectX Gadget and Linux/MacOS

Version 6.00c

- Fixed bug when creating/exporting [MenuBar](#)

Version 6.00b

- Added more checking to SpiderBasic Export to prevent invalid Flags from being Exported
- Fixed Image Loading bug in SpiderBasic Export

Version 6.00

- Compiled with latest PureBasic 6.00 x64
- Added Max Width and Height resize option to exported Form
- Added settings option to Declare Forms in Constant file
- Added custom ToolBarStandardButton to PVGadgets Module for PureBasic 6.00+
- Added Gadget Flags shown based on Development Platform selected
- Added CTRL-LeftClick to select multiple Gadgets on Form
- Added View all Source Code in Single File when Debugging a Project
- Fixed ICO file from being imported & overwritten without asking first
- Fixed bug with Gadgets messing up when selecting between Form & Gadget
- Fixed Gadgets not highlighting inside ScrollAreaGadget
- Fixed bug where Crash Protection would corrupt Gadget pointers causing wrong Gadget to be selected
- Changed 'Print Used Variable Names' now prints to PDF instead of HTML Web Page
- Changed saved Projects are now in SQLite format instead of Binary

Version 5.73f

- Added ability to remove User Registration Info from header of Exported Code
- Fixed problem when Duplicating multiple gadgets
- Fixed interface randomly becoming unresponsive after MessageBox displayed

Version 5.73e

- Added font selector for Spin Gadget
- Added ability to use negative numbers with TrackBar Gadget and Spin Gadget
- Fixed problem with trying to resize MenuBar and ToolBar images in Exported Code

Version 5.73d

- Fixed sub windows changing position with high DPI monitors (PureBasic's ResizeWindow command is still bugged, had to change to WinAPI)
- Changed exported code for Gadgets that use #PB_EventType_LeftClick flag

Version 5.73c

- Fixed "Column/Field" Input Form getting stuck behind Main PV Interface
- Fixed Icon glitch in Properties Form
- Minor Tweaks

Version 5.73b

- Added [Settings Menu](#) to expose more internal settings for customizing and cleaned up MenuBar options
- Fixed X/Y position not displaying properly when selecting Gadget in a Container type Gadget (PanelGadget, ScrollAreaGadget, ContainerGadget)
- Fixed Gadget/Form corruption when moving/sizing with Left-Click

Version 5.73

- Compiled PureVision as a 64bit Application (PureVision will no longer run on 32bit OS)
- Compiled with latest PureBasic 5.73 x64
- Added DPI awareness so PureVision looks good on 4k Displays with Windows Scaling
- Updated functions in PVGadgets Module with DPI awareness
- Menubar, Toolbar & Gadgets that use Icons (ICO) now give option to convert to PNG for better DPI and Platform compatibility
- Added custom Message Boxes for better visual information
- Added option for each image to be scaled with [Raw or Smoothing](#) flags
- Added [PVGadgets Splitter](#) command for easier creation of Splitter Gadgets
- Added [Enable Left-Click to Move/Size](#) Gadgets (for those who don't like using Right-Click)
- Fixed HyperLink Gadget missing background color

Version 5.72b

- Removed the need to install PVGadgets Library since it is now a Module and can be found in the "/Modules" folder.
- Fixed problem with [AutoUpdate Checking](#)

Version 5.72

- Added [Shortcut Gadget](#)
- Changed PVGadgets & other Modules are now stored in "DRIVE:/PureVision_Path/Modules" Folder
- Changed PVGadgets from PureBasic Library to PureBasic Module for use with SpideBasic and User Customization
- Changed PVPlugins from PureBasic Library to PureBasic Module for better User Customization

The Dynamic Resizing that is available in the PVGadgets Module is also available to use in SpiderBasic Projects. This Module is made available to Registered PureVision End Users to use Royalty Free in Private or Commercial Compiled Projects.

You are free to modify this Modules Source Code and/or add to it as you wish.

The Source Code in this Module is only for Registered PureVision End Users and is not to be distributed publicly.

Version 5.71

- Added support for mk-soft's ScaleGadgets Module (v0.31)
- Fixed sorting in Recent File List
- Fixed bug when reordering Gadgets in a Container type Gadgets (PanelGadget, ScrollAreaGadget, ContainerGadget)
- Fixed some Bugs when Exporting to SpideBasic
- Changed PureTrans and Embedded Language Files to use Maps for easier access and functionality

Version 5.70

- Fixed problem with Font Flags when exporting to MacOS & Linux

Version 5.63f

- Added support for open/save/edit .sb & .sbi files when using Code Inject

Version 5.63e

- Fixed incorrect Constant for MenuBar ToolTip when exporting with #PB_Any
- Fixed Edit Columns windows getting stuck behind another window when "Tool Windows on Top" is set

Version 5.63d

- Fixed ToolBar export to make it compatible with PB5.46 LTS
- Added extra checks when Opening/Importing a file to prevent wrong Form Name displaying in Form List Selection Box

Version 5.63c

- Added a little more information under [Code Inject](#) in the manual

Version 5.63b

- Added checks when switching between different [Development Platforms](#) for better code export and preventing crashes

Version 5.63

- Added [ToolBar Editor](#)
- Fixed Right Clicks being lost in Main Interface
- Fixed proper "/" and "\" being exported when target compiler is Linux

Version 5.62c

- Added GIF support for ImageGadget, ButtonImageGadget and Window Background
- Fixed Interface not restoring properly if quitting PureVision while minimized
- Fixed incorrect filename appearing in "File name" field when Exporting Project

Version 5.62b

- Fixed Gadgets marked as Hidden not exporting properly
- Fixed SpiderBasic Export not adding ParentID when a Child Form was assigned to a Parent Form
- Added SpiderBasic specific Gadget Flags for OpenWindow, TextGadget, StringGadget
- Updated PureBasic Gadget Flags

Version 5.62

- Recompiled with PureBasic 5.62
- Added DPI scaling to PureVision Interface to prevent text and gadgets from being cut off when Windows DPI is scaled above 100%
- Added #PB_Ignore option to StatusBar by setting Field size to zero
- Fixed many spelling mistakes in Help File/Manual (thanks to FangBeast)

Version 5.61

- Recompiled with PureBasic 5.61
- Updated Help File to better explain the PVDynamic changes in the PVGadgets Library

Version 5.60e

- Fixed Gadgets highlighting incorrectly in PanelGadget is large font is used in the Tabs
- Fixed Gadget Selection in ScrollAreaGadget so you can use mouse to Move/Size Gadgets
- Changed the way Dynamic Resizing works. [BindEvent using SizeWindow](#) now replaces 2 Windows Callbacks
- [Windows Callback](#) is only required if a Gadget Color not supported by PureBasic's native Foreground/Background Color commands is used
- Due to these changes, install new PVGadgets Library, re-[export PureVision Projects](#) and verify your Main Event Loop contains proper [Dynamic Sizing code](#).

Version 5.60d

- Added CTRL-A to [Export Project](#) Form allowing quick Select/Deselect of Forms to Export

Version 5.60c

- Fixed MenuBar errors when loading older non Unicode Projects

Version 5.60

- Recompiled with PureBasic 5.60

Version 5.51

- Recompiled with PureBasic 5.51
- Added [CrashProtection](#) feature. Ability to restore session if PureVision does not exit properly
- Added ability for PureVision to automatically detect and install the appropriate 32/64bit version of PVGadgets
- Fixed missing Bind for CloseWindow in SpiderBasic Export Module

Version 5.50

- Recompiled with PureBasic 5.50

Version 5.42

- Recompiled with PureBasic 5.42
- Fixed export bug when exporting using #PB_Any support
- Fixed failed install of PVGadgets when PureBasic has been installed into "Program Files" or "Program Files (x86)" on Windows 8.1 and Windows 10

Version 5.41

- Recompiled with PureBasic 5.41
- Changed the way PureVision exports Menu Items. Since PureBasic does not allow #PB_Any for Menultem, these values are exported as Enumerated Constants instead of Globals

Version 5.40

- Recompiled with PureBasic 5.40

Version 5.31c

- Added support for SpiderBasic. Forms can now be exported using optional

SpiderBasic Export PlugIn

Version 5.31b

- Fixed bug preventing Language files from loading

Version 5.31

- Recompiled with PureBasic 5.31
- INI file is now stored in User/AppData/Roaming

Version 5.30c

- Fixed bug where icons might not display correctly in MenuBar Editor
- Added ability for ColorRequester to store custom colors
- Added modify background color of grouped gadgets
- Added modify foreground color of grouped gadgets
- Added enable/disable XP Skin support of grouped gadgets

Version 5.30b

- Reworked Gadget Repositioning in Gadget Tree
- Added ability to Archive Project

Version 5.30

- Recompiled with PureBasic 5.30
- Added OpenGL Gadget
- Added PBAny Support/No Constants, Build and Embed Language File & Export Unicode Export Flags are now saved with GUI file
- Fixed Include Files not being copied into new Include Folder if Save Path was changed
- Fixed display problem with API Gadgets when adjusting position in ScrollArea or Container Gadget

Version 5.22d

- Exported code is now compatible with PureBasic "EnableExplicit" command
- Fixed bug in exported code using PB_Any support

Version 5.22c

- Fixed bug with coloring gadgets using transparent (setting color to -1) and displaying color properly

Version 5.22b

- Added Unicode support to PureVision
- Added Unicode support to PureTrans

Version 5.22

- Recompiled with PureBasic 5.22 LTS
- Fixed bug causing Form to lose focus and not update properly / become unresponsive

Version 5.21b

- Reworked some of the draw routines when resizing Forms in PureVision. This should fix some

problems when running under Windows 8

- Fixed a selector bug that appeared when PureBasic changed syntax of Frame3DGadget to FrameGadget
- Added warning message when trying to install PVGadgets with no Write Access to folder
- Updated PVGadgets x86 and x64 Library for better compatibility under Windows 8
- Removed XP from PureVisionXP name since PureVision is mainly on Windows 7 and now Windows 8 desktop's. This means PureVision will install in C:\PureVision by default and settings will be in PureVision.ini
- PureVision is no longer forced to Maximize, screens are so big these days it makes no sense taking up all that space
- Added CTRL+PageUp to toggle Tool Windows on Top, same as "Menubar/Forms/Tool Windows on Top"

Version 5.21

- Includes PVPlugins for 64bit Windows

Version 5.20

- Recompiled with PureBasic 5.20 LTS
- Fixed Code Export for compatibility with PureBasic 5.20
- Updated PVGadgets Library for compatibility with PureBasic 5.20

Version 5.11

- Recompiled with PureBasic 5.11 to fix bug with "Save Current Form"

Version 5.10

- Updated PureSkin to use new Compression Library, making PureSkin compatible with x64
- Form will now automatically default to Borderless when PureSkin is on a Form
- Added some missing Flags to Gadgets
- Added API_ to beginning of Windows specific Flags. Don't use these Flags if building Cross Platform Apps.

Version 5.01

- Added Font Manager to quickly and easily modify Font attributes of multiple Gadgets
- Fixed bug with the way Area type Gadgets are exported

Version 5.00f

- Removed MenuBar Icons API callback, now uses native PureBasic MenuBar Icons code

Version 5.00d

- Added DirectX Gadget (OpenWindowedScreen) - Great for creating template for Windowed Game.

Version 5.00c

- Added ability to remove XP Skin Support from individual Gadgets and Forms
- Added PVGadgets_NoSkin command to PVGadgets User Library
- Added AutoUpdate Checking so you always know when update is available
- Fixed problem with images not changing properly in some of the ButtonImage Gadgets due to PureBasic changing from SetGadgetState() to SetGadgetAttribute()

Version 5.00b

- Both PVGadgets 32bit & 64bit can be installed from same PureVision executable

Version 5.00

- Compiled for PureBasic 5.0
- Added new flags for Forms (OpenWindow)

Version 4.61e

- Fixed bug in ListView Gadget where exported code missed extra flags.

Version 4.61d

- Added Font support for ExplorerTreeGadget, ExplorerListGadget & ExplorerComboGadget

Version 4.61b

- Added Font flag for PanelGadget allowing font used for Panel Tabs to be modified

Version 4.61

- Added currently selected Gadget type to bottom of Properties Window
- Fixed bug with Vertical or Horizontal aligning of Grouped Gadgets

Version 4.60

- Added new PureBasic CanvasGadget
- Added #ThreeState flag for TreeGadget and ListIconGadget
- Recompiled PVGadgets User Library to support PureBasic 4.60 - 32bit & 64bit

Version 4.51g

- MenuBar Editor now allows inserting items where highlighted
- Made Properties Window re sizable
- Added some MenuItem commands to PV_Plugins Library

Version 4.51f

- Updated flags for many Gadgets
- Added native coloring for many Gadgets

Version 4.51e

- Fixed problem that could cause MenuBar Items to get scrambled when creating a MenuBar

Version 4.51c

- Fixed StatusBar resize bug which could cause crash in Unicode or Threadsafe mode

Version 4.51b

- Added ability to move menu/toolbar when full screen is disabled

Version 4.51

- Added tool tips for IP Gadget
- Fixed bug with Exit Window hiding behind About screen

Version 4.50

- Recompiled for PureBasic 4.50

Version 4.41b

- Optimized exported code when including AVI files for AnimGadget
- Added commands PVGadgets_AnimGadgetPlay, PVGadgets_AnimGadgetStop and PVGadgets_AnimGadgetHide for more control over the API AnimGadget

Version 4.41

- Added Font & Color support for Editor Gadget
- Added #CURRENTWINDOW# and #CURRENTGADGET# macro to Code Injector

Version 4.40

- Recompiled for PureBasic 4.40
- PVGadgets ThreadSafe, Unicode & ThreadSafeUnicode are all in one Library now, no need for separate subsystem settings.

Version 4.30

- Recompiled for PureBasic 4.30
- Removed export code for versions of PureBasic earlier than 4.30 (use PVXP 4.20 if you want to use earlier versions of PureBasic)

Version 4.20

- Recompiled for PureBasic 4.20

Version 4.02

- PVGadgets is now available in ThreadSafe and Unicode Mode. If you are compiling to Unicode in PureBasic check "Create Unicode Executable" under Compiler Options and enter Unicode as Library Subsystem. Change "Unicode" to "ThreadSafe" if you are compiling to ThreadSafe.
- Fixed problem with menubar icons not being copied over with other graphics/icons if new folder is selected
- Fixed EditorGadget and ReadOnly flag

Version 4.00 beta 14

- Exported code for PureBasic 4.x uses more native color commands
- Updated BubbleTip commands to fix problems
- Updated PieChartGadget to fix problems

Version 4.00 beta

- Modified to compile using PureBasic 4.00
- Fixed problem with form size when renaming Bin Groups
- Fixed bug to prevent reading invalid Language Files

- Fixed bug with Code Injection using Form and OuterLoop

Version 2.15

- Changed the way PVXP reads Language files to support more Languages

Version 2.14

- Added Code Injection feature to Form and Gadget Properties
- Added Compile/Run and View Source from PureVisionXP Interface
- Enhanced the Plug-In support and made simple VD to PV form converter to demonstrate
- Fixed conflicting Constant problem with StatusBar
- Updated PVGadgets Library

Version 2.12

- Optimized and recompiled PVGadgets library
- Fixed problem with exported code not using "auto resize image" flag properly for ImageGadget
- Fixed bug when exporting ImageGadget using PB_Any
- Fixed small bug with Bubble Tips preference not saving when exiting PVXP
- Removed #PB_Image_BorderRaised flag from API ProgressBar. Something has changed in PB that no longer allows this to work properly.

Version 2.11

- Added new PieChart Gadget
- Added new Functions to PVGadgets Library
- Fixed problem with required ImageDecoders not being included in Exported code due to case sensitive. "png" included decoder, "PNG" did not.

Version 2.10

- Added option to bypass "Confirm Overwrite" when Exporting projects
- Fixed problem with menubar settings not being read from INI file when PVXP is launched from double clicking GUI Project Files
- Fixed problem with images not being copied over when project is saved in a new folder

Version 2.09

- Added option to load other Language Files for PVXP interface
- Added option to select custom AVI for AnimImageGadget
- Made Properties Window re sizable
- Fixed problem with tool windows not repositioning on startup
- Fixed problem with DisableFullScreen flag

Version 2.08

- Fixed problem with Online Translator

Version 2.07

- Recomplied both PVXP and PVGadgets for PureBasic 3.93
- Fixed problem with Images folder not being created
- Fixed problem with CTRL-S and ALT-X not working

Version 2.06

- API CanvasGadget and API ProgressBarGadget can now be put into Area/Container Gadgets
- Fixed Export bug when using CanvasGadget or ProgressBarGadget and the #PB_Any option

Version 2.05

- Fixed problem when Merging Forms

Version 2.04

- Added coloring for CalendarGadget
- Added help for PVGadgets_FormatNumber() command under Extra Functions
- Added a number of ListIcon Functions under Extra Functions
- Fixed problem with DatePickerGadget format being reset

Version 2.03

- Added DisableFullScreen flag to PureVision.ini file
- Fixed bug that allowed Form to have Font option (causing PVXP shutdown)

Version 2.02

- Added PanelGadget information to Help File
- Recompiled with new PureBasic Gadget Library to fix problem with CalendarGadget

Version 2.01

- Fixed problem with Properties Form losing focus when entering ID Name
- Auto Resize Code is now included when MaximizeGadget is added to Form
- Added snap grid sizes in PureVision.ini file
- Updated Help File

Nov. 08 , 2004

- PureVisionXP 2.0 officially released on November 8, 2004

Index

- A -

About 27
 Add CanvasButton 69
 Add Color Gadgets 62
 Add Gadget 31
 Add Icon 83
 Add Image 83
 Adding Forms 61
 Adding Gadgets 61
 Adding Groups 10
 Adding Windows 61
 AnimGadget 64
 API 31, 36
 API Gadgets 31, 36
 App Directory 78
 App Folder 78
 App Path 78
 Ask to Quit 13
 Auto Resize 47
 Auto Sizing 59
 Auto Update 14
 AutoSave 14
 Autosizing Gadgets 61

- B -

Bin 10
 Bubble 14, 67
 BubbleTip 67
 BubbleTips 14
 Bug Fixes 138
 Button Pressed 71

- C -

CallBack 60
 callbacks 100
 Canvas 68
 Canvas Gadget 68
 CanvasButton 71
 CanvasButtonHover 70
 CanvasButtonPressed 71

Change Icon 85
 Change ListIcon Icon 85
 Child 49
 Child Form 49
 Close 9
 Code Generate 7
 Color 49, 50, 60
 Color CallBack 60
 Color Gadgets 62
 Coloring 50
 Column Width 79
 Columns 78
 Combining Forms 6
 Commands 59
 Compile Images 15
 Confirm 7, 13
 Confirm Overwrite 7
 Confirm Quit 13
 Constant 7
 Convert PureSkin 21
 Copy 41
 Copy Gadget 41
 copy paste 100
 Copyright 27
 Count Columns 78
 Crash Protection 14
 Create ImageList 83
 Create PureSkin 21
 CreateCanvasButton 69
 Creating Code 7
 Current Directory 78
 Current Folder 78
 Current Path 78
 Cursors 23
 Custom 9, 37
 Custom Cursor 23
 Custom Gadgets 36, 37
 Custom Pointer 23
 Customize 9

- D -

Data 20
 Data Sheet 20
 Default 9
 Default Cursor 95
 Default Pointer 95
 Default Positions 9

Delete 45
Delete CanvasButton 72
Delete Form 28
Delete Gadget 45
Development 13
Development Platform 13
Directory 78
Display 93
Duplicate 42, 43
Duplicate Offset 43
Dynamic 47, 59
Dynamic Color 60, 62
Dynamic Gadgets 61
Dynamic Resize 47
Dynamic Resizing 59
Dynamic Window 61

- E -

Edit MenuBar 28, 29
Edit ToolBar 28, 30
Edit with Skins 12
Editor 29
Embed 7
End 9
English 13
Event 7
Event Loop 7, 100
Example Code 60
Exe Directory 78
Exe Folder 78
Exe Path 78
Export 7, 133
export plugin 133
Export Project 7
Exporting 7
Extended Gadgets 63
Extended Menubar 28

- F -

Field 51
File 4
File Menu 4
Folder 78
Font Color 49
Font Size 49

Font Style 49
Fonts 49
Form 28
Form Properties 46
Format 87
Format Number 87
Forms 9
Free 66, 69, 94
Free Anim Gadget 66
Free Canvas 69
Free CanvasButton 72
Free Gadgets 66
Free ImageList 84
Free PieChart 75
Free ProgressBar 74
Free Resources 69, 74, 94
Freeze 76
Freeze Window 76
French 13

- G -

Gadget Color 50, 60
Gadget Creator 31
Gadget Highlighting 11
Gadget Move 28
Gadget Over 96
Gadget Position 98, 99
Gadget Properties 46
Gadget Selector 40
Gadget Size 28
Gadget Toolwindow 31
Gadget Tree 40
Gadgets 31, 32, 63
Gadgets to Group 10
General Flags 37
Generate 7
German 13
Get Column Width 79
Grid 31
Grid Size 31
Group 10, 43
Group Gadgets 43
GroupBin 10
Grouping 10, 43
Grouping Gadgets 10

- H -

Help 26
Help File 26
Hide 44, 90
Hide Gadget 44
Hide Skin 90
Hiding 44
Highlight 11
Highlighting 11
History 138
Horizontal 96, 98
Hover 70

- I -

Icons 83
Identify Gadget 96
Image to PureSkin 21
ImageList 83
ImageList Icon 83
Images 15
Images in EXE 15
Import 6
Importing Forms 6
Include 15
Include Images 15
Include PureSkin 15
Index 59
Info 27
Information 26, 27
Init 60, 64
InitAnimGadgets 64
Initialize Anim 64
Initializing 60
Initializing PVDynamic 60
Inject Flags 37
Intro 59

- J -

Justify 80
Justify Columns 80

- K -

Key 27
Key words 59

- L -

Language 7, 13, 17
Language Manager 17
Last Row 81
Library 59
License 27
Linux 13
Load 5, 92
Load Project 5
Load PurePoint 92
Localize 13
Lock Form 61
Lock Window 61

- M -

Main 59
Main Interface 9
Make Tool Window 77
Manual 26
Maximize 47
Memory 92
Menu 59
MenuBar 4, 28
Menubar Editor 29
Menubar Icons 29
Merge 6
Merging Forms 6
Middle Mouse 28
More Gadgets 63
Mouse Move 28
Mouse Size 28
Move 28
Move Gadget 28
Multi Language 17

- N -

Naming Project 5
Native Gadgets 32

New 4
New Form 28
New Project 4
Number Formatting 87

- O -

Offset 43
On Top 9
Online 26
Online Updates 26
Open 5
Open Project 5
Optimize 15
Options 9
Override 16
Override Gadget Settings 16
Override Settings 16

- P -

Panel 52
Panel Order 52
Panel Tab 52
PanelGadget 52
Parent 49
Parent Form 49
Paste 41
Paste Gadget 41
Path 78
PB Any 7
PB Gadgets 31, 32
Pie 74
PieChart 74
Platform 13
Pointer Over 96
Pointers 23
Position 96, 97, 98, 99
Positions 9
Print 20
Print Data Sheet 20
Print Info 20
Print Project 20
ProgressBar 73
Project Data 20
Properties 46
PureBasic Gadgets 32

PurePoint 23, 91, 92
PurePoint Maker 23
PureSkin 15, 21, 88
PureSkin in EXE 15
PureSkin Maker 21
PureSkinHide 90
PureSkinMem 89
PureTrans 17
PureVision Website 26
PVDynamic 59
PVDynamic_AddColorGadget 62
PVDynamic_AddGadget 61
PVDynamic_AddWindow 61
PVDynamic_ColorGadget 60
PVGadgets 36, 47, 59, 63
PVGadgets_AnimGadget 64
PVGadgets_BubbleTip 67
PVGadgets_Canvas 68
PVGadgets_CanvasButtonAdd 69
PVGadgets_CanvasButtonDelete 72
PVGadgets_CanvasButtonHover 70
PVGadgets_CanvasButtonPressed 71
PVGadgets_CanvasFree 69
PVGadgets_FreeAnimGadget 66
PVGadgets_FreeCanvas 69
PVGadgets_FreePieChart 75
PVGadgets_FreeProgressBar 74
PVGadgets_InitAnimGadgets 64
PVGadgets_PieChart 74
PVGadgets_PieChartFree 75
PVGadgets_ProgressBar 73
PVGadgets_ProgressBarFree 74
PVGadgets_Refresh 75

- Q -

quick start 100
Quit 9

- R -

Redraw 45
Refresh 45, 75, 76
Refresh Form 45
Refresh Gadgets 75
Refresh Window 76
Register 27

Register CanvasButton 69
 Registration Key 27
 Reset 95
 Reset Pointer 95
 Reset Window 77
 Resize 28, 47
 Resize Gadget 28
 Resizing 47
 Revisions 138
 Right Mouse 28
 Row 81, 82

- S -

Save 5
 Save Current Form 5
 Save Form 5
 Save Project 5
 Select 40
 Select Gadgets 40
 Select Row 82
 Set Column 80
 Set Column Width 80
 Set Width 80
 Settings 4
 Single Form 5
 Size 28, 49
 Skin 21
 Skin from Memory 89
 Skinning 21
 Skinning a Form 88, 89
 Skinning Form 21
 Skins 12, 88, 89
 Snap 31
 Snap to Grid 31
 spider 133
 spider basic export 133
 spiderbasic 133
 Splitter 77
 SplitterGadget 77
 Statusbar 51
 Style 49
 Support 26
 Syntax 59
 System Cursor 94
 System Pointer 94

- T -

Tab 40, 52
 Tab Order 40, 52
 Tabs 52
 Temp Directory 78
 Temp Folder 78
 Temp Path 78
 Themes 12
 Tips 14, 67
 Toggle 28
 Tokens 37
 Tool Menu 17
 Tool Window 77
 Tool Windows 9
 ToolBar 28
 ToolBar Editor 30
 Tools 4, 17
 Tools Menu 17
 ToolTip 67
 Top 9
 Top Most Window 77
 Top Window 77
 Transparent 50, 77
 Transparent Gadget 50
 Transparent Window 77

- U -

Unfreeze 77
 Unfreeze Window 77
 Update 14
 Update Checking 14
 Updates 26, 138
 Use Language 13
 Use PurePoint 93
 User Library 59

- V -

Version 26, 27
 Version Checking 26
 Vertical 97, 99

- W -

Website 26
Width 79
Window on Top 77
Window Position 96, 97
Window Reset 77
Window Transparent 77
Windows 13

- X -

X Position 96, 98
XML 37
XP Style 12
XP Theme 12

- Y -

Y Position 97, 99