

*PRICE US\$49.95*

# Creating & Manipulating Windows

Published in Australia by

*Objectives*

*Publishing and Distribution*

1 Bellambi Place WESTLAKE QLD 4074

© Lorraine Cobcroft  
Qld. Australia, 1998

**For Student Support,**  
Call 07 3279 5091  
Fax 07 3279 4494  
Email [student@objectivespd.com](mailto:student@objectivespd.com)

Layout mini-tutorials are available on a wide range of topics to assist you to learn to use Layout's most powerful features. Each tutorial contains information on specific features, plus practice exercises to demonstrate the use of features and options. Some tutorials contain sample flowcharts supplied on accompanying diskettes or by email.

Mini-tutorials are supplied without warranty as to the accuracy or currency of content in relation to specific Layout versions or as to the clarity of text. However, we welcome user comments and suggestions which may assist us to improve the quality of the product.

If you experience difficulty working through a mini-tutorial, study of an associated programming feature may be necessary. Alternately, you may request personalised assistance from our Student Support Services division. The normal technical support charge rates will apply.

*This work is protected by international copyright.*

*Reproduction of all or any part without the express written permission of the author is strictly prohibited*

## **This mini-tutorial is suitable for**

- Beginners learning programming basics
- Intermediate to advanced programmers wanting specific information on the creation, manipulation, usage, and characteristics of windows in Layout

## **Aims:**

- To explain the characteristics and uses of windows in program design
- To demonstrate the creation and management of windows
- To explain the options and selections and input parameters required when creating windows
- To explain the use of windows management blackboxes

## **Objectives**

After completing this tutorial, the user should be able to

- Understand the different types of windows and their features and usage.
- Understand the checkbox options and inputs required when creating and managing windows and dialog boxes
- Understand the meaning and use of result variables returned by Window management blackboxes and window functions
- Be able to create desired special display and information management effects using Windows and Dialog Boxes on screen



# Table of Contents

Typographic Conventions .....	6
Introduction .....	7
Options for Controlling the Window Display and Usage .....	8
Start of Program Window Options .....	8
Window Menus .....	10
Creating a Window .....	10
Open a Window Blackbox .....	10
Advanced Positioning Options .....	14
Important Notes: .....	15
Creating a Dialog Box .....	16
Open a Dialog Box .....	17
The Window Status .....	18
Some Common Mistakes With Windows & Dialog Boxes .....	20
Disabling Automatic Window Management .....	22
Window Management Blackboxes .....	23
Switch Window .....	23
Redraw a Window .....	24
Redraw all Windows .....	25
Get Window Information .....	25
Other Blackboxes used when Working with Windows .....	26
Program Control Folder - Get Object Information .....	26
Graphics Folder Blackboxes - Set Graphics Co-ordinates , Draw a Box .....	27
Screen Display Folder - Color/System Defaults .....	27
Text and Data Files Folder - View a Text file .....	28
Power Tools Folder - Advanced Screen Display -Remove Redraw Objects .....	29
Scroll Bar Blackboxes .....	29

## Sample Flowcharts

Creating an Auto-Sized "Generic" Window .....	30
Special Effects with Windows .....	32
Hiding a Window .....	32
Colouring a Window Background .....	32
Managing the Window Color Scheme .....	33
Using a Variable or Blank Window Title .....	34
Using a Window To Control When the Menu Appears .....	34
Using a Variable Window Name .....	35
Switching Windows .....	36

## USEFUL LAYOUT EXTRAS

For Layout for DOS .....	38
For Layout for Windows .....	38

# Typographic Conventions

The following typographic conventions are used throughout tutorials to assist in understanding:

Menu options - bold type with first character underlined

eg. **Print**

Checkbox options - bold text preceded by checkbox symbol

eg.  **Window is Movable**

Button text - small caps

eg. CONTINUE

Names of Blackboxes and Windows - bold italics

eg. ***Address Window***

***Open a Window***

Text to be typed by user - normal, in inverted commas except where show via screen grab or boxed to appear as screen grab

eg. "Enter the Customer Address"

Selected Value or Variable names are shown in italics

eg. *Card X*, *Selected Object*

In Flowchart instructions, a small bullet marks the start of each instruction.

In blackbox descriptions, a blackbox is preceded by a blackbox bullet symbol, while a folder symbol indicates a blackbox folder

# Introduction

Long before the advent of MS Windows, the idea of using windows to control screen display and active tasks was adopted by many programmers because of the convenience this device provided in managing the screen display. Windows allowed an activity to temporarily change all or part of the screen display, create a new and “clean” environment in which a program activity could take place, and return to the previous display on request. Where a number of different displays or activities were taking place at one time, the Window allowed the programmer to indicate clearly to the user in which area of the screen the current activity was taking place. Users could be given the opportunity to determine when they wished to switch to another activity and to tell the program their intention.

Both Layout for DOS and Layout for Windows make extensive use of Windows and provide a range of options for determining type, size and position of windows, and for manipulating the windows used in your programs.

In Layout, a Window is defined as “an area in which part or all of your program takes place”. Windows can be used to obscure portions of the screen, clear portions of the screen, and to confine program activity to a given area.

Windows have a white background. A little trickery can be employed to use alternate colored backgrounds, with some restrictions. (In Laywin, a Layout Extra is available to provide the capability to use a gray window background. In LayDOS, a Layout Extra allows transparent dialog boxes.) You can define the border and title bar colors of windows, or set up facilities to allow your user to define these.

Two basic types of Windows are used:

**The conventional window** has a title bar and may be sized, closed, or moved by the user. Multiple windows may be displayed on screen at one time. Users may switch between windows, specifying which are to be active at a given time. Windows may be created in such a way that the screen display behind the window is restored on closing - or so that the screen behind the window is cleared. In Layout for DOS, objects may be displayed outside the window, or you can specify that any object outside the window will not be visible to the user.

**The Dialog Box** has no title bar. It dominates the screen display while open. Any object placed outside the dialog box while the dialog box is open will be invisible to the user. While a dialog box is open, all activity is confined to the dialog box unless a *Switch Window* element transfers it elsewhere.

Dialog boxes are usually used to convey important information to users - such as error messages, program operation instructions, warnings about activities that are taking place, information about the status of program operations, etc. Icons are often displayed in the top left corner of the dialog box to re-inforce the message displayed. In Layout, checkbox options allow you to automatically specify a particular type of icon display - eg. you can choose an ALERT BOX, showing a stop sign. You can also display icons using DISPLAY A PICTURE or by placing a picture on a card and displaying, activating, or letting the user fill out the card.

## Options for Controlling the Window Display and Usage

### Start of Program Window Options

When creating a new program with Layout for Windows, you are required to specify the window options for the main program window. These options exist in Laywin only. Select ONE of the options for your program.

**Set Main Window Position**

You will be asked to stretch a rubber band box and position it to indicate the position of the first Window. When your compiled program is run, this is the window that will appear on screen. If the window is not full screen size, the user's desktop will be visible in the background. If you place future windows or dialog boxes outside the area of the main window, the appearance of the program when running may seem a little strange - since these windows will appear outside the window in which the program is running.

Using this option, it is possible to create a program which operates in a tiny section of the user's screen and leaves the remainder of the screen free for other screen displays.

**First Window is Main Window**

The first window you open in your program will be the main program window. If you do not use an *Open a Window* element, there will be no program window at all.

This option differs from others in that all other options automatically create a window of some kind for your program, without you having to specifically select the *Open a Window* blackbox and create a window.

Using this option, the menu appears in the window opened by the first *Open a Window* command. Using other options, the menu appears in the default window and the first *Open a Window* command creates a new window - usually within the default window.

**Note:** Use this option when creating BLACKBOX programs which do not display anything on screen.

**Use Default Window Position**

This option creates a window (containing your menu if you use one in your program) in the position your system regards as the default position. The program operation will be similar to when **Set Main Window Position** is used, except that you are not required, or even permitted, to size or position the window yourself. The sizing and positioning of the window is handled automatically by your system.

**Main Window is Maximised**

The main program window will be automatically created by the system, containing the menu if there is one, and will occupy the entire screen.

**Main Window is Minimised**

The program will commence without opening any window or displaying anything on screen. If using Windows 95, it will place an entry on the user's Windows 95 task bar to allow the user to restore or maximise the program when desired.

## Window Menus

In MS Windows, windows have their own drop-down menus. LayWIN follows the same standard, allowing you to **S**witch, **M**ove, **R**esize, **C**lose, **M**aximise, **M**inimise, and **R**estore Windows using menu options or mouse actions.

Windows in Layout for DOS also have drop down menus containing options to **S**witch, **M**ove, **R**esize, and **C**lose the window. The selections in **Windows Options** will determine which of these menu options appear greyed out and which can be used.

## Creating a Window



### *Open a Window Blackbox*

The *Open a Window* Blackbox inserts TWO elements into your flowchart - an Open Window command, and a matching *Close a Window*. All events that are to occur within the window must be placed between these two elements - by clicking on the *Close a Window* to insert blackboxes above it, or by selecting a group of elements and/or procedures, cutting, then selecting *Close a Window* and pressing **P**aste.

You cannot move or delete an *Open a Window* or a *Close a Window* independently. To move or delete a window, you must remove everything from inside the window (pasting it elsewhere if appropriate), and then highlight both the Open and Close elements.

You can alter the size, position, or other window specifications by clicking on either the Open or the Close element.

The Window parameters required by the *Open a Window* blackbox are:

### **Specify Window Name**

This is the name by which you refer to this window in your program. It need not be the same as the TITLE which will be displayed on the title bar. The Window name is a VALUE.

**Open a Window options:**

In this checkbox section, multiple options may be selected in any combination.

 **Window is a Child Window** (LayWin only)

A child window is special type of window which has a special relationship to another window in the program (its parent). A child window assumes a screen position relative to the parent. Depending on the position specified, part of the child window may be obscured beneath the parent. Even with **User May Switch Windows** checked on both parent and child windows, the child window may never display a highlighted title bar to indicate it is current. When you switch to the child window, the parent window is current.

When the child window is current, messages and instructions sent to the child may be responded to by the parent, and activities specified to occur in the child window may take place in the parent window. A Windows message which is not handled by the window to which the message is sent will normally be handled by MS Windows. If the window is a child window and the received message is not handled by the target window, the message will be sent to its parent window. If the message is not handled by the parent window, then the message is sent to MS Windows.

 **Window preserves screen** (DOS Only)

The display shown in the screen area covered by the window prior to opening the window will be restored automatically when the window is closed. Note that this option requires additional memory resources to store the displayed image while the window is active. Note also that all windows in LAYWIN preserve the screen as this is the convention in MS Windows.

 **Window is movable**

If this option is selected, the user will be permitted to move the window around on the screen by clicking on the window title bar or selecting **M**ove on the Window menu.

**Window is Resizable**

With this option checked, when the user places the mouse cursor on the side or bottom of the window, the cursor will change to a two-headed arrow and the user can press the mouse button and drag the mouse to shrink or expand the window. The **S**ize option on the Window menu will be active if this option is checked.

 **Window is Closeable**

With this option selected, the user can close the window by clicking on the Close button provided on the top right of the window, or by selecting **C**lose on the Window menu. If this option is not checked, the window should remain open until the *Close a Window* element relative to that window is encountered in the flowchart.

 **User can Switch to Another Window**

This option allows you to use multiple windows simultaneously in your program and either switch between them during program operation, or allow the user to switch between them using the Window Menu option or the mouse. Note that you must use the *Switch Window* blackbox to actually change the active window unless the user makes the switch by selecting the alternate window title bar or Window menu option. If you open two or more windows in succession, the last window will be the active window and the title bars of previously opened windows will be greyed out until the *Switch Window* blackbox specifies a different window as active, or until the user clicks the title bar of another window.

 **Display objects outside Window** (Option not available in Layout for Windows)

This option determines whether or not objects displayed outside the currently active window are visible. In Layout for DOS, use of a window is not a requirement, although it is recommended. However, you may display objects outside the current window, or even outside all windows used in the program. However, with this box unchecked, such objects will not be visible to the user.

**Window is an MDI window** (Option not available in Layout for DOS)

MS Windows provides a Multiple Document Interface. This means that you can work on two or more documents or activities at one time, switching between the two activities at will by clicking in the window containing the activity you wish to switch to. Checking this option allows you to create an MDI environment for your user.

MDI Windows have a special relationship to a “parent” window which runs the main program. Each MDI window can be minimised and maximised independently, and a series of icons may be present on screen - each representing a different document. Maximising an MDI window icon causes parent program operation to resume, with the maximised or current window as the current document.

The use of MDI windows is extremely resource hungry and inefficient and should be avoided unless there is a very specific reason requiring it.

### **Specify Window Caption**

Here you should specify the text you wish to display on the title bar of your window. Type in the text exactly as you want it to appear, or select the **VARIABLE** button and choose a text variable.

### **Specify Window Location**

There are two ways to determine the window location. If **Ask for Positioning Method** is turned off in **Advanced Options**, the second method will not be permitted.

#### 1. Position using a VALUE

A rubber band box is presented on screen and you must stretch it to the desired size by clicking your mouse cursor on the small boxes displayed on the sides of the box. Set the desired position by placing the mouse cursor anywhere in the box, pressing the mouse button when the cursor turns to a four-headed arrow, and sliding the mouse until the box is positioned to your liking.

## 2. Position using a VARIABLE

You will be required to select four +/-32767 variables to specify the X and Y co-ordinates of the top right (start x, start y) and bottom left (end x, end y) of the window. Using this method, you can adjust the size and position of the window depending on its content, the user's screen size, other objects displayed on the user's screen, and various other conditions.

## Advanced Positioning Options

The following options may be checked in the **Advanced Options** section under the **Options** top menu. If checked, additional options will be presented when positioning a Window or Dialog box as well as in other *Display* blackboxes.

**Ask for Positioning Method**

see below

**Ask for Variable Positioning Parameters**

allows positioning using X and Y variable co-ordinates

Checking **Ask for positioning method** allows one of the following options to be chosen when positioning Windows and Dialog boxes on screen using a Value:

**Position is not affected**

Layout will position the object where you physically place it with your rubber band box. It will not adjust the position in relation to screen size or placement of other objects on screen.

**Position depends on size of screen**

Layout will attempt to position the window appropriately to the screen size being used, using the relationship of the position you specify to the screen size you are using as a guide. Remember that screen size is determined by pixel resolution, not by physical screen dimensions.

**Position depends on last opened object**

Layout will adjust the position of the window relative to the last object displayed on screen. This can have a dramatic effect on a program where, for example, a decision or match determines which of a group of differently positioned windows will open.

Imagine selecting one menu option opens a window in the bottom left corner of the screen, while a different menu option opens a window in the top right corner. Now a window opens which you drew in the bottom right corner. This window opens regardless of which menu option was selected. Checking **Position Depends on Last Opened Object** could result in a dramatically different positioning of the second window.

 **Position depends on screen and last opened object**

Checking this option will result in the position of the window being affected by both the screen size and the position of the last opened object.

In the above group of options, choose only one option

 **Size depends on size of screen**

Layout will attempt to size the window in proportion to the screen size.

It will use the relationship of the window you created to your screen size as a guide, and try to draw the window on the user's screen at approximately the same ratio of size to screen size.

 **Centre object on screen** (option not available in Layout for DOS)

Layout will attempt to position the window you have created in the exact centre of the screen.

Choose one, both or none of the above options.

## Important Notes:

1. Screen size is determined by the pixel resolution set on the user's system. A 1024 x 768 screen is larger than a 640 x 480 pixel screen, even though the physical screen dimensions (screen height and width) may be identical.

2. Layout presents a range of options at different stages of program design, depending on the options chosen in the **Advanced Options** menu.

The selections made in each option group are not exclusive. That is, the effect of any given selection will depend to a large extent on other selections made in the same element, or elsewhere in the program. In *Open a Window*, for example, selecting **Centre Object on Screen** may not result in a window which is located in the exact centre of the screen if **Set Main Window Position** was chosen in the *Start of Program* options. Rather, it will position the window in the centre of the main program window you specified.

The effect of a selection in the second group of Variable Positioning Options will also be altered by any selection made in the first group of options.

If results of selections are surprising at times, always remember that it is the overall combination of all options selected that you are seeing - not necessarily the result of a single option selection.

3. Sizing and positioning options may not always yield consistent results. The automatic screen positioning method used by Layout is excellent, but not perfect. There are vast numbers of variables affecting screen positioning - including video card types and drivers - and achieving accurate results can be tricky! It is recommended that you perform extensive program testing on a variety of display types.
4. The effect of some options may be altered by the operating system being used. Some operating systems use default or over-riding specifications which result in the options you choose during program creation being ignored.

## Creating a Dialog Box

A dialog box is created in the same way as a window - but with less parameters to complete. As with a Window, the *Open Dialog Box* and *Close Dialog Box* elements are joined, and must be moved or deleted together. Events and displays to take place in the dialog box must be inserted between the two elements.

## Open a Dialog Box

When using a Dialog Box, the name of the box must be typed in as a value. There is no provision for using a variable. The name of the box will not display on screen.

Specify the type of Dialog box to be used. This determines the type of icon that will be displayed in the top right corner.

### **Normal**

No icon is displayed

### **Alert**

In Layout for DOS, a red STOP SIGN is displayed

In Layout for Windows, a red circle containing an X is displayed

### **Information**

In Layout for Windows, a white exclamation bubble containing a blue lower-case letter i is displayed. (This option is not available in Layout for DOS)

### **Time**

In Layout for DOS, a small clock is displayed. (This option is not available in Layout for Windows)

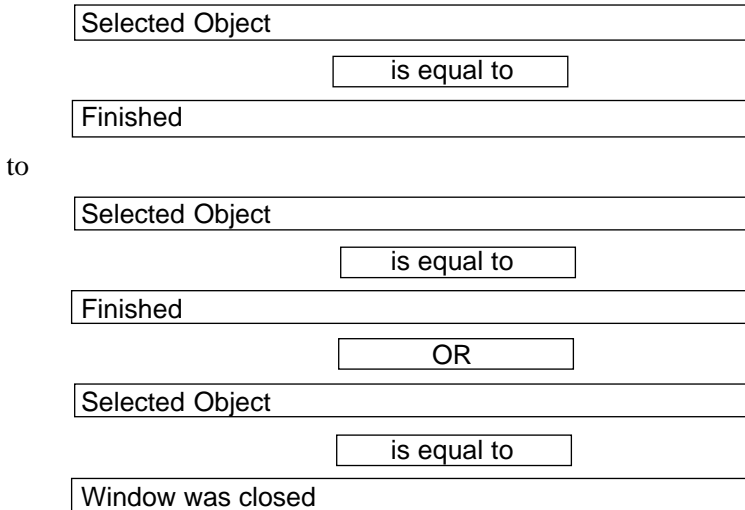
The sizing and positioning options are similar to those for sizing and positioning a window. You can use values or variables for positioning, and the advanced options operate in a similar fashion.

Dialog boxes do not have a title, nor do the child, move, size, switch, close, or MDI options apply.

Any items displayed on screen prior to opening the dialog box will remain visible to the extent that they are not covered by the dialog box. The display behind the dialog box will be restored fully when the dialog box is closed. However, dialog boxes are much more efficient in memory management than windows with **Preserve Screen** option checked.

# The Window Status

The Window Status is a checkbox variable. This is one of the default variables set up by Layout whenever a program is commenced - that is, it is a **Standard System Result**. You can use the Window Status in decisions, matches, and repeat loops. For example, if a window can be closed by the user, it may be appropriate to change a REPEAT UNTIL loop from

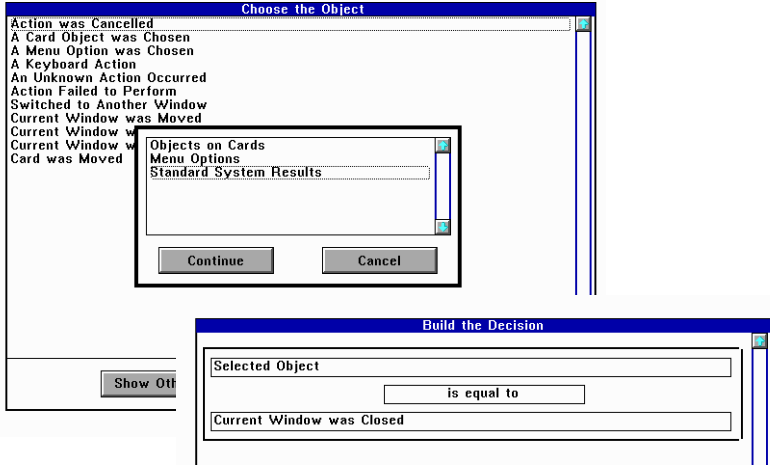


Without changes of this type, the user may never be able to actually select FINISHED and the program may be permanently locked in the repeat loop.

## Possible Window Status Results:

- Action was Cancelled
- A Card Object was Chosen
- A Menu Option was Chosen
- A Keyboard Action
- An Unknown Action Occurred
- Action Failed to Perform
- Switched to Another Window
- Current Window was Moved
- Current Window was Resized
- Current Window was Closed
- Card was Moved

To set up a condition using an action performed on a window in a decision, match, or repeat loop, select **ADD VALUE**. Press **SHOW OTHER TYPES** if necessary, and select **Standard System Results** as the value type. A list of possible conditions relating to windows will be shown.



## Some Common Mistakes With Windows & Dialog Boxes

1. Opening too many windows and/or dialog boxes inside one another

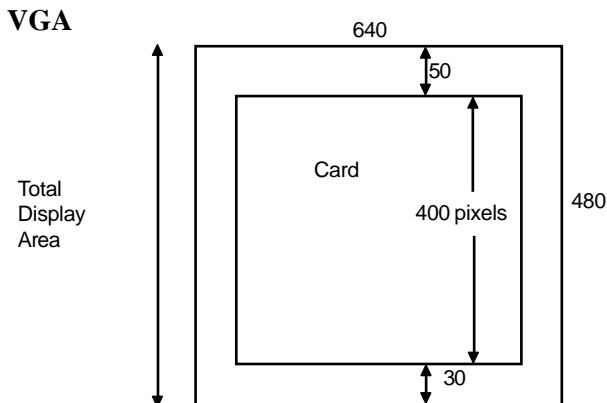
This can cause annoying flickering effects as multiple windows close rapidly one after another. It also consumes excess memory.

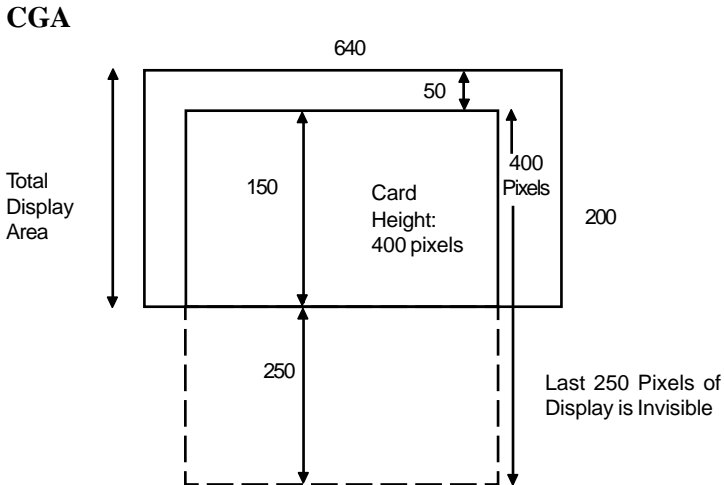
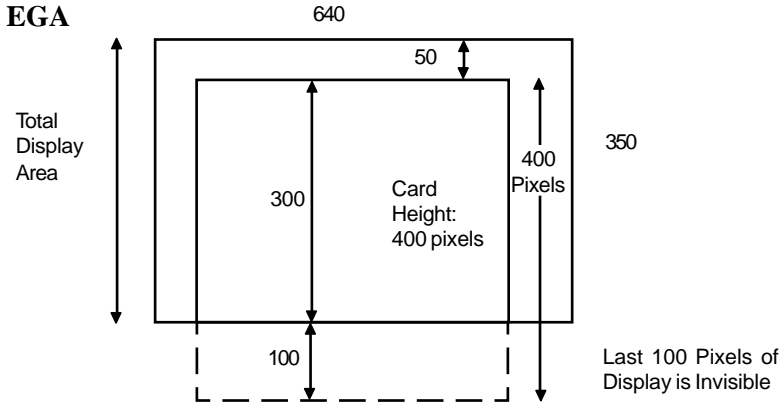
Unless there is a very good reason for needing multiple windows open on screen together, close one window before opening another.

2. Placing windows where they will be only partly visible or not visible.

This results from using variable positioning parameters or positioning options carelessly. The use of variable positioning and positioning options is encouraged, as it usually yields a more professional program. However, these options must be used with extreme care and extensive testing should be practiced to make sure the results are as desired regardless of screen size, operating system used, and user selections.

eg. A screen display using 400 pixels on the Y axis with no automatic adjustment selected will appear as:-





3. Items disappear from the screen display

This can result from not selecting the **Preserve Screen** option (in DOS only) when opening a window, careless positioning of a windows (especially if using variable positioning), or from attempting to display something illegally (eg. trying to use *Display a Variable* to show something outside a dialog box, but placing the command element between the *Open a Dialog Box* and *Close Dialog Box*).

4. Attempting to display objects or open a new window after the main program window is closed. (LayWIN only)

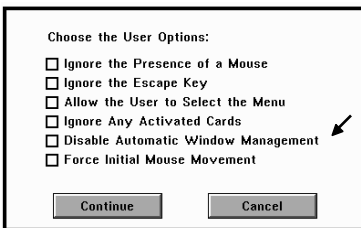
If the first window in your program is defined as the main window, it must remain open throughout the program. Closing it and opening another may result in an error message being displayed.

5. Opening multiple windows with the same name simultaneously.

Take care never to allow two or more windows with the same names to be open on screen simultaneously. Remember, the window name is the name by which you refer to it in your program. This name is not necessarily related to the title displayed in the Window Title Bar.

## Disabling Automatic Window Management

By default, *Automatic Window Management* is enabled while a user is filling out a card. This means that any highlighted menu options on the Window menu may be used by the user, and re-sizing, moving, closing and switching of windows by the user is permitted according to the *Window Options* you checked in the *Open a Window* blackbox.



If desired, automatic window management functions can be disabled while specific cards are being filled out.

Notice that there is an option - **Disable Automatic Window Management** on *Let User Fill Out A Card*. With this option checked, although moving, closing, switching

and re-sizing may be permitted for this window, it will not be allowed during the operation of filling out the current card. As the programmer, you may still cause a switch window operation or another window event by issuing a specific instruction to that effect, but the user will not be able to interact with the window using handles or the window menu until the operation of filling out the current card is completed.

Checking this option is generally recommended unless you want to take the trouble to set up sophisticated decisions to control user interaction with the window. For example, you may have made certain the `CONTINUE` button does not allow an exit from the card until certain information is filled correctly and the card is saved. However, if the user closes or switches the Window, the effect of all your precautions may be defeated.

On the other hand, if there is a likelihood that differing screen displays may prevent an entire card from showing in a window, it may be desirable to allow the user to move or resize the window, in which case you should not disable automatic window management, but you will need to include conditions to protect against the user closing or switching the window inappropriately.

If your user selects the **Switch to Another Window** option on the window menu, the switch will take place without you implementing a *Switch Window* blackbox in your program. If you have disabled automatic window management, you will need a *Switch Window* blackbox in order for the user's choice to switch windows to be implemented.

## Window Management Blackboxes

Layout contains a special folder of blackboxes to manage Window re-drawing and switching and to allow you to obtain information about a window status.

### *Switch Window*

*Switch Window* allows you, the programmer, to dictate that a different window should become current. In order for a user to work within a window - to select text or objects on a card inside the window - that window must be current. When multiple windows are displayed on the screen you can use *Switch a Window* to change which window is current without user interaction.

Note that the window you switch to must have been opened before this blackbox. Do not attempt to use a *Switch Window* blackbox prior to opening the window that you wish to make current. The blackbox parameters are:

**Window / Dialog Box (Window / Dialog box)**

Select the name of the window to switch to.

**Switch Result (Checkbox)**

Select or type the name of a Result Variable to hold the result of switching. eg. "Switch Window Result". This will be a checkbox variable with the possible values, **No checkbox is Selected, Window Was Successfully Switched, The New Window Was Not Open.**



***Redraw a Window***

*Redraw a Window* redraws a window on screen, replacing its title and repositioning it if desired. The window can be redrawn at a different location with a different title if you so desire. You might want to use this option to retitle an existing window or to move the window to a new location. You may also redraw the window to a new size. Redrawing windows uses less memory than creating new windows.

Note that in Layout for DOS, when a window is redrawn in a new location or moved by the user, any cards within the window, including all the objects on the card, all move with the window. However anything not on the card - that is, anything simply displayed on screen - will **NOT** move with the window but will be cleared from the screen and will have to be redrawn. In Layout for Windows all window contents move with the window.

Select the window to redraw. Type a new title or select the text variable holding the title. Stretch the rubber band box or enter positioning coordinates. The blackbox parameters are

**Window to Redraw (Window / Dialog Box)**


Select the name of the window to redraw

**Window Title (Text)**

Type the title or insert a text variable

**Window size and Position**

Specify with a rubber band box or variable co-ordinates

 ***Redraw all Windows***

*Redraw all Windows* redraws all the currently open windows on the screen, starting with the bottom most layer and working upwards. This blackbox would probably be used when the contents of a window has been updated, and there is another window which resides above it on the screen.

There are no blackbox parameters.

 ***Get Window Information***

*Get Window Information* reports the current position and size of a window, the currently selected window controls, the window options, and whether or not the window preserves the screen when closed. This can be particularly useful for determining the new position of a window after a user has moved it.

To use this blackbox, select the window by name from the displayed list. Select or create a variable to hold the Window Controls. This will be a checkbox variable with multiple checkbox selections allowed. Enter or select variables to hold Window Co-Ordinates. eg. Window Start X, Window Start Y, Window End X, Window End Y. The blackbox parameters are:

**Window (Window / Dialog Box)**

Select the window name

**Checkbox Variable (Checkbox)**

Create a variable to hold the Window options

**Window position co-ordinates (+/- 32,767)**

Specify with a rubber band box or variable co-ordinates

# Other Blackboxes used when Working with Windows

## Program Control Folder

### *Get Object Information*

*Get Object Information* enables you to obtain information about a Window additional to that returned by *Get Window Information*. Input the name of the "object to get information about", which will, in this case, be the name of a Window. (Note that it is the NAME of the Window, not the title, that must be specified here). In the Window parameter, the blackbox will return the name of the Window referred to if a window has been switched, resized, or moved.

The Object Type will return the type of action that took place. Possible values include: Action was Cancelled, A Card Object was Chosen, A Keyboard Action, An Unknown Action Occurred, Action Failed to Perform, Card was Moved, or, if the object is a Window, Current Window was Moved, Switched to Another Window, Current Window was Resized, or Current Window was Closed.

Next, variables Object Info 1 through Object Info 4 return various information of a type dependent on the Object being investigated. In this case, we are seeking information about a Window. Therefore, Object Info 1 returns the new start X position, and Object Info 2 returns the new start Y position for the window. Object Info 3 returns the width of the window, and Object Info 4 returns the height of the window.

Using this blackbox enables you to determine what the user did with the Window, and how to redraw the screen appropriately, re-position the cursor, highlight a different object (eg. in another window), etc.

 **Graphics Folder Blackboxes** ***Set Graphics Co-ordinates***

*Set Graphics Co-ordinates* creates a frame of reference on screen, defined by variable co-ordinates (start x, start y, end x, end y). Everything drawn or placed within this frame of reference will assume its position relative to its defining co-ordinates. This means that when the rectangle defined by *Set Graphics Co-ordinates* moves, everything positioned using the x and y co-ordinates defined will move in relativity. You should generally use *Set Graphics Co-ordinates* to position windows and other objects on screen, rather than positioning with simple values. It requires just one additional step - but enables the display to adjust automatically to a large extent to different screen resolutions and different program conditions.

 ***Draw a Box***

*Draw a Box* may be used to fill or partially fill a window with color, or to clear the window or a portion of a window of any current display. Set the color for the box before drawing, and select Solid Black or Solid White Figure, or Solid Figure in the Current Color.

Selecting one of the inverted color options will have the effect of creating a contrasting highlight box over the area contained in the box you draw. This is effective to draw attention to a section of the window display. For example, if the window contains a list of objects, a thin rectangular shaped box drawn over a line of the list can serve as a highlight to show the currently selected list item. This is demonstrated in the tutorial called *Working with Lists - Advanced*.

 **Screen Display Folder** ***Color/System Defaults***

*Color/System Defaults* allows you to set the System Default Colors for Windows appearing in your application. You can vary the System Defaults at strategic points in the program. You can also use the *Color/System*

*Default* blackbox to obtain the current default colors for windows. This is useful if the user has chosen colors using the *ColorBox*, or if another program may have changed the colors of the window. It is good programming practice to Get the *Color/System Defaults* at the start of any program which may change colors so that you can restore the colors before closing the application and exiting back to the user's desktop.

To use *Color/System Defaults* to define window colors, create the variables "Background Color", "Foreground Color" and "Border Color". Use the *Colorbox*, or three *Put Value Into Variable* blackboxes to define the colors to use, followed by a single *Color/system Default* Blackbox specifying **Window** as the checkbox in System Object. Check SET to tell the system to set the new colors.

The Background Color is the color used for the background of the title bar. The Foreground Color is the color used for the text written on the title bar. The Border Color is the color used for the side and bottom border of the window.

## **Text and Data Files Folder**

### *View a Text file*

*View a Text file* is a blackbox which has little significance in a study of Windows and Dialog Boxes - however it is useful to note that this blackbox will create its own Window on screen automatically with appropriate scroll bars to allow the user to view the text file.

Note that when using *View a Text file*, you must use *Set Graphics Coordinates* to setup the display area and use variables to position the window in which the file will be viewed. Specify the filename using a value or a variable.

The system will set up a window which can be closed, moved or sized. The system will apply scroll bars if appropriate to allow vertical and/or horizontal scrolling.



## Power Tools Folder - Advanced Screen Display



### ***Remove Redraw Objects (Layout for Windows Only)***

**The *Remove Redraw Objects*** blackbox (Layout for Windows only) should be used after scrolling a screen area, using a ***Draw Box*** blackbox to clear a window, or performing other actions which alter the display within a window. This blackbox removes any graphics objects that are stored in the redraw list maintained by the current window. Whenever a graphics object, such as a line, box, ellipse, picture, display text, or display variable is drawn in a window, the objects are stored in a redraw list. When a window is repainted, the window uses the redraw list to restore the objects. A window is usually repainted if it is moved or covered by another window. ***Remove Redraw Objects*** clears the list of objects and prevents Windows from performing an automatic Window Redraw which may be undesirable.



### ***Scroll Bar Blackboxes***

***Draw Scroll Bar, Activate Scroll Bar, Deactivate Scroll Bar, and Check Scroll Bar*** provide you with the facilities to draw and manipulate your own scroll bars on windows, list boxes, etc. The drawing and operation of scroll bars is usually handled automatically by Layout - but using these blackboxes can give you powerful advanced screen display management capabilities. The use of these blackboxes is beyond the scope of this tutorial and is covered in other Advanced Tutorials including *Working with Lists - Advanced*.

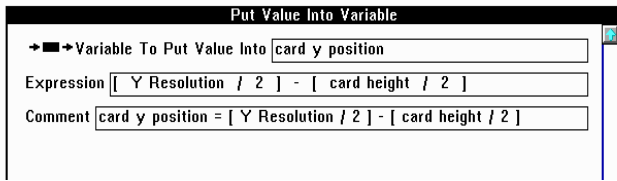
# Creating an Auto-Sized “Generic” Window

## GENERIC.FLO

This routine is handy for automatically creating a window in which to display your cards instead of having to draw every window individually. It is also useful if you are accommodating multiple different types of screen display and you have selected the option to make the card adjust to the screen display. This routine will automatically adjust the window to the card size.

- Start a **New Flowchart**. Select the **CARD DESIGN TOOL**, **IMPORT CARD DESIGN**, *Address Card (USA Address)*.
- **Open a Cardfile** selecting  **Open an existing file with an unknown card**. Ignore the Open Cardfile Options. In Cardfile to open, select **NEW VARIABLE** and type "Cardfile to Open". In DOS Filename type "\\LAYOUT\SAMPLES\FLOWCHART\ADDRESS.CRD" (adjust file location if required).
- Use **Get Card Dimensions** in the *Advanced Cards* folder of the *Cards* blackbox folder to find the width and the height of the card.

- Make a New Variable - a text variable and name it **Window Title**. Use **Put Value Into Variable** to insert the value "Address" into the window title.
- Create 3 +/- 32,767 number variables to hold "Card X Position", "Card Y Position", "End Card X" and "End Card Y". A series of **Put Value Into Variable** elements now calculates the size of your window. The *Card Y Position* is the row on which the window will commence.



Of course this formula places the card in the centre of the screen. If you wish to place the display elsewhere you will need to amend the calculation accordingly.

- ***Put Value Into Variable*** - *Card Y Position = Card Y Position - 4*. (This subtracts 4 rows from the starting row and moves the starting row up 4 pixels in order to allow space for the window title bar to be displayed).
- ***Put Value Into Variable*** - *Card X Position = (X Resolution / 2) - (Card Width / 2)*. (Again this places the card in the centre of the screen).
- ***Put Value Into Variable*** - *End Card X = Card X Position + Card Width*.
- ***Put Value Into Variable*** - *End Card Y = Card Y Position + 4 + Card Height*
- ***Open a Window*** and elect to define the screen location using variables. Start X is *Card X Position*, Start Y is *Card Y Position*, the End X is *End Card X* and the End Y is *End Card Y*.
- The window caption, of course, is the variable *Window Title*.
- ***Let User Fill Out Card***. Card to fill out is the *Card to Use* and file is *Cardfile to Use*. Position the card on the screen using variable coordinates - *Card X Position*, *Card Y Position*, *End Card X* and *End Card Y*.

**Note:** If you are confused by the use unknown cards, order the mini-tutorial **Using Unknown Cards** for detailed information on this powerful programming feature.

# Special Effects with Windows

## Hiding a Window

HIDEWIN.FLO

Use the *Color/System Defaults* blackbox to set the Window Colors. Precede this blackbox with *Put Value Into Variable* blackboxes which set the *Background Color*, *Foreground Color*, and *Border Color* to the same color as the desktop background or the background of the card. If using the same color as the background of the card, adjust the window size to the exact card size or very fractionally smaller. The result will be a Window or Dialog box which behaves like any other, but is not visible. The card will appear to be drawn directly onto the desktop.

NOTE: In Windows95 and Windows 3.x a black outline appears around the window, and in Windows95 when the window is inactive, it's title bar turns grey. This procedure works well in DOS.

NOTE: Windows do not have borders if **Move** or **Resize** options are checked.

## Colouring a Window Background

BACKCOL.FLO

To achieve a colored window background, create a card slightly larger than the white area of the window. Using the Background Paint tool, draw a filled box in the desired color covering the entire background of the card. When saving the card, ensure **Display Background Text Transparently** and **Display Background Picture Transparently** are checked on the Save a Card Screen.

When the card is displayed, the white window will not show.

Alternately, immediately after the *Open a Window* element, use *Set Color* to specify the desired color, and *Draw Box* to draw a filled box over the white area of the window. Using this technique requires extreme care if the window size or position can be changed by the user or by program events. Rather than using values to position the box, it is recommended you use variable co-ordinates calculated by either reading the Window co-ordinates (using Get Window Information) or by using the blackbox *Set Graphic Co-ordinates* and stretching a rubber band box over the window.

Note that the latter process works better in Windows than in DOS, as MS Windows controls the movement of graphics drawn in a window more effectively.

## Managing the Window Color Scheme

### COLORWIN.FLO

This flowchart demonstrates the controlling of window color schemes by setting the values for Foreground, Background and Border colors then using the Get Color/System defaults blackbox to set the default color scheme for Windows and Dialog boxes used in your program.

The flowchart also demonstrates the use of the *ColorBox* to allow users to select their preferred color choices.

Notice that border colors do not display for windows unless the **M**ove and/or **R**esize options are checked, thus creating a side and bottom border which the user can select to move or resize the window.

Take special note of the *Color/System Defaults* blackboxes at the start of the program to **Get Colors** and another at the end to return the original colors. It is a courtesy to your user to always follow this procedure to ensure you do not alter their color choices on the desktop or in other programs they use after running the program you create. You would normally need additional instances of the *Color/System Defaults* blackboxes to read and restore the colors for the desktop background, text, menus, etc.

## Using a Variable or Blank Window Title

NAME.FLO

Sometimes a colored window bar with no title is preferred. To achieve this, simply type a single space, or nothing at all, when asked to enter the Window Title.

Using a Variable as a Window Title permits the use of a single window for many different screen displays, with the title adjusting according to the window content. To achieve this, when asked for the Window Title, select the `USE A VARIABLE` button. Type "Window Title" or select the variable *Window Title* from the variables list if you have already created it.

Next, precede the *Open a Window* blackbox with a *Put Value Into Variable* element. Select the variable *Window Title* and type the title you wish to use.

You may wish to vary the Window Title according to the window content, or other program conditions. Use Matches or Decisions to achieve this.

You can also change take the Window Title from the Card Name, a File Name, or a variable on the Card. Using Get Card Objects in the Professional Edition of Layout, you can obtain the name of the card being displayed and use this as the Window Title.

## Using a Window To Control When the Menu Appears

HIDEMEN.FLO

This technique can be used to control when the top menu appears. Normally, the menu appears immediately the program is run. However, by opening a window which covers the entire screen (and checking the option **Window Preserves Screen** if working in LayDOS), you can prevent the top menu appearing until this window is closed.

Note that if you position the Window using a rubber band box, it is impossible to hide the top menu bar - but the options will be hidden. Ensure **Ask for**

**Variable Positioning Parameters** is checked in **Advanced Options** and choose to position the window with variables. Enter "Start X", "Start Y", "End X", and "End Y" as the positioning variables. Now, place *Put Value Into Variable* elements before the *Open a Window* element and set *Start X* and *Start Y* to "0". Set *End X* and *End Y* to the correct values for the screen display type being used (639 and 479 for VGA).

If you do not wish to see a title bar on the window, set the default window foreground and background colors to white.

The window will fill the entire screen and hide the top menu. When the window is closed, the top menu will appear.

## Using a Variable Window Name

VARWIN.FLO

Although it isn't immediately obvious, you can also use a variable Window Name, allowing you to reopen the same window later in your program and saving the memory space a new window would occupy. A quirky design fault in Layout means that many users overlook this feature.

To use a variable Window Name:

- *Open a Window*, entering "Window 1 Name" in the Window/Dialog Box name and selecting *Use A Variable* to enter "Window 1 Caption" in the Window Caption.
- Repeat to open a second window called "Window 2", using *Window 1 Caption* again for the title.
- Click on the *Open A Window* element for *Window 2* and click on the Window Name.
- Select Window 1 from the list.
- Clean up the Flowchart to remove Window 1

NOTE: Take care that multiple windows of the same name created this way are not opened together on screen. Close one window before opening another of the same name.

## Switching Windows

### SWITCHWN.FLO

Using the ADDRESS.CRD file in the Layout samples and the Tape Player card in the Card Library, we have constructed a screen with two windows - an Address Window to contain the Address Card and a Control Window to contain the Tape Player card.

In drawing the screen, we must SWITCH to the correct window before displaying each card. Note that a card displays in the last window opened, unless you switch to a different open window before displaying the card. Any action, such as filling out a card, takes place in the last window opened, unless a *Switch Window* precedes the action or the user has physically switched windows.

In the *Action On Buttons* procedure, the STOP button switches to the Address Window to allow the user to edit the card and press the SAVE or CANCEL button. On exiting this procedure, regardless of which button was pressed or which window is current, we ensure the Tape Player can be used again by switching to the control window before returning to the start of the Repeat Loop.

Notice that you must press F1 to exit this flowchart, as it isn't possible to select **Exit** on the menu to end the repeat. This is because the user cannot switch from the control window to access the menu.

Load and run SWITCHW2.FLO. The only change is that the Control Window has **User Can Switch to Another Window** checked. The user can now select the menu to Exit the program. However, it is now possible for the user to switch to the address window and become locked in it, unable to return to the Control Window or the Menu. To remedy this, the Address Window must also be altered to allow the user to switch windows. These flowcharts demonstrate the importance of care in controlling user interaction with windows.

In both these flowcharts, note the temporary flicker on the Address Window Title Bar as that window becomes current each time a new card is retrieved and displayed.

Load and run SWITCHW0.FLO to see the effect of not switching windows at this point. The new address card is displayed in the Control Window!

# USEFUL LAYOUT EXTRAS

## For Layout for DOS

*PowerPack Pro* provides tools to

- Create a Transparent Dialog Box (behaves as a Layout dialog box but no box displays - only the box content is visible)
- Locate centre of screen and Locate centre of object - allow centering of windows)
- Automatically or uniformly resize a displayed object
- Re-define colors - enables you to temporarily change all white objects to a specified alternate color so that your window backgrounds are coloured (Use with caution - it redefines the color completely so that all white objects are changed until a further Redefine Color blackbox is used to restore the original color)

## For Layout for Windows

*Custom Controls* provides tools to

- **Grey Window Background**
- **Reset Window**

Used before *Close a Window* to remove the grey background (necessary for effective and error-free memory management)

- **Enable & Disable 3D effects**

Adds 3D effects to such objects as the *Windows Filebox*, *Fontbox*, *Colorbox*, and *Message Boxes* (see below)

- **Add Controls to Window**

Adds Minimise and Maximise controls to windows

**■ Change Window Status**

Maximises, Minimises, or Restores a window

**■ Message Box**

Automatically creates a Windows standard Message box, which is similar to a dialog box. Adds the message text, a selection of buttons, and a choice of Stop, Exclamation, Question Mark, or Information icons in a single step.

**■ Enhanced Message Box**

Automatically creates a Windows Standard Message box with enhanced appearance.

**■ Create Message Bar**

Creates a message bar at the bottom of the current window to hold help text, status information, etc. Only one message bar is permitted per program, so it is usually placed on the *Main Program Window*.

**■ Add text to Message Bar**

Places specified text in the current Message Bar.

**■ Create Toolbar**

Creates a Toolbar menu. Only one Toolbar is permitted per program, so it is usually attached to the Main Program Window. Tools are also provided to add buttons, grey buttons, reset buttons, delete buttons, and insert spaces between buttons on the toolbar.

Custom Controls also provides the ability to create floating Tear-Off menus, although this option is not directly related to Windows and Dialog boxes.