

## **ObjectWindows 5.01**

### **TABLE OF CONTENTS**

1. New features or examples
2. API/Implementation changes and Fixes
3. Known problems

#### **1. New features or examples**

ObjectWindows v5.01 includes the following new features:

- Emulation of the Column Header common control when running 16-bit applications on Windows v3.1 or Windows NT.
- TListWindow has new methods for managing selected items (e.g. GetSelCount, GetSelIndices, GetSelStrings...)
- Programming of tree view controls has been simplified by the addition of new TTreeNode methods rendering TTreeItem obsolete. The existence of TTreeItem should now be considered internal implementation detail.

#### **OCF/OLE**

- OCF now supports the Automation VARIANT type via the TAutoVal class.
- The AUTOWORD and AUTOEXCL examples (\BC5\EXAMPLES\OCF\ ) illustrate MS Word and MS Excel controllers respectively.
- The COMSRVR illustrates a simple COM server.

#### **2. API/Implementation changes and Fixes**

This section describes changes that have been made to ObjectWindows v5.01

- The EvCommand method of TGadgetWindow now attempts to dispatch a menu style command before rerouting the command to the Command Target. This allows derived toolbars, statusbars and toolboxes to have first crack at commands.
- TGadgetWindow creates its tooltip window with the WS\_DISABLED flag enabled to prevent the tooltip from accidentally grabbing input focus.
- TOleWindow no longer exposes methods/routines related to hosting OCXes in 16-bit since the required low-level support is not implemented in the underlying helper DLL, BOCOLE.DLL.
- TFrameWindow's GetCommandTarget uses the following search logic:
  - a. Focus window if it's a child of our client
  - b. Our client window
  - c. Focus window if it's a child of ours
  - d. Last focus child of ours
  - e. The frame itself

NOTE: In prior versions items b. and c. were reversed; this often caused menus/buttongadgets to be disabled if a window not parent to the frame's client area (such as a control in a toolbar) has focus.

- Some TDibDC operations were failing under Win95 because DIB.DRV doesn't work there. TDibDC no longer uses DIB.DRV at all. It now requires the WinG DLL instead. (Like DIB.DRV, WinG is freely available from Microsoft.)
- TPrinter's SetPageSizes is now virtual.
- TLayoutWindow's Layout method is now virtual.
- TPaneSplitter's RemovePanels is now virtual.
- Nested classes within TDib are now properly exported from the ObjectWindows DLL.
- The size grip is removed from maximized frame.

NOTE: The size grip is not implemented in 16-bit as the current implementation of TUIPart does not emulate the functionality provided by the 'DrawFrameControl' API.
- The 'NGrowBy' member of TCellArray is initialized to 1 if an initial value was not specified.
- TTooltip only caches the hint text specified when OWL provides the underlying implementation

of tooltips.

- TPropertySheet's PageUnchanged method now passes the window's handle when sending the PSM\_UNCHANGED message.
- TPropertySheet provides a 'EnableSubclass' method to inform OWL that the property sheet should be subclassed and thunked. Use this method if you want to customize the sheet and/or add event handlers in a TPropertySheet-derived class. This method must be invoked when using a property sheet as a client of a frame window.
- The TWinsockDll class now provides static member functions which bind to the corresponding API in the Winsock DLL the first time they are invoked. The prior version exposed 'function' objects which required that the DLL implement all functions whether they were invoked or not. The same applies to the TWinGDll class.
- TWindow provides two new methods Attach and Detach for instances when the underlying HWND may be destroyed and recreated.
- TNoteTab tabs respond to arrow keys (by moving focus to the next tab).

#### BIDS/Class Library

- Static members of TRegKey such as CurrentUser and LocalMachine have been replaced with accessor functions such as CurrentUser() and LocalMachine(). This prevents an exception that debuggers noticed when TRegKey attempted to create instances of keys not supported on the run-time system. (See the comments in REGISTRY.H for more information.)

### **3. Known problems**

- The location is ignored when inserting a dockable window into the harbor.
- Various members of the TPrintDialog::TData object are not properly copied to the PRINTDLG or PAGESETUPDLG structures.
- Misc. painting problems with docking toolbars
- Enabling 'ShrinkToClient' does not work properly with dialogs as client with decorated frame windows.

#### OCF/OLE

- Embedded or linked objects revert to their original size after being resized and made persistent.
- OLE documents are considered dirty as soon as an object is activated
- TOcDocument creates a TOcPart for both 'regular' embedded objects and OLE control loaded from a storage/stream (NOTE: A TOcControl should be created for controls).

#### BIDS/Class Library

- TFileName's MergeParts fails to check for existing backslashes.
- TFileName's Move relies on the 'MoveFileEx' API which is not implemented on Windows 95.
- TButtonGadget fails to rebuild the bitmaps of the button upon a WM\_SYSCOLORCHANGE message.

## **ObjectWindows 5.0**

This file contains a list of the new features for Borland's ObjectWindows for C++ version 5.0 and information about the product that is not included in the documentation. The online Help contains updated information not found in the printed documentation. For the latest information about Borland C++ and for information about how to contact Borland, see the README.TXT file included with your distribution disks.

### **TABLE OF CONTENTS**

1. New features
2. API, implementation and organization changes
3. ObjectWindows examples directory structure
4. Rebuilding ObjectWindows Library
5. Known problems

## **1. New features**

ObjectWindows 5.0 includes the following new features:

- Complete encapsulation of Windows Common Controls

### **Common Control**

Animation  
Column Header  
Progress Indicator  
Hotkey  
Track Bar  
List View  
Tree View  
Property Sheet & Property Pages  
Tab  
Tooltip  
UpDown  
RichEdit  
Drag Listbox

### **ObjectWindows Class(es)**

TAnimateCtrl  
TColumnHeader  
TGauge  
THotKey  
TSlider  
TListWindow  
TTreeWindow  
TPropertySheet/TPropertyPage  
TTabControl  
TTooltip  
TUpDown  
TRichEdit  
TDragList

- Additional ObjectWindows Controls such as glyph buttons, check listboxes and notetabs
- Improved toolbar and status bar implementation with docking support and new gadgets (See the DOCKING and GADGETS examples under EXAMPLES\OWL\CLASSES)
- OCX Container support: OWL's TOleWindow can now host OCX controls. The TOleDialog class provides a mechanism whereby OLE controls are dynamically instantiated when a dialog is created. The OCXDIALOG example illustrates. OWL and OCF also provides methods to catch events and invoke the properties and methods of the controls.

- New classes that manage UI drawing

### **UI Helper Class**

TUIHandle  
TUIBorder  
TUIPart  
TUIFace

### **Draws**

Manipulators for on-screen objects  
Border styles  
Frame and parts of controls  
Bitmap in various states (e.g., disabled)

- Support classes for OS extensions such as WinG, Winsock, and MAPI
- Improved routing of WM\_COMMAND messages
- Encapsulation of the Win95 Shell Namespace with easy methods to browse through the hierarchy of objects (files, storage devices, printers, networks, etc.) on the desktop.

## **2. API, implementation and organization changes**

This section describes changes that have been made to ObjectWindows

- Several methods have been updated to use the intrinsic 'bool' type supported by the compiler
- The virtual method 'Error' of TValidator now takes a 'TWindow\*' as parameter
- TBitmapGadget takes two additional, optional parameters: 'index' of image and 'flag' indicating whether the image is from the shared TCellArray instance
- Several methods of TDC which expected a TColor by value now take a 'const TColor&' instead
- Several methods of TEdit are now virtual so they may be overridden by the new TRichEdit class
- TGadget is now derived from TStreamableBase when compiled in an environment which does not support RTTI
- TGadgetWindow no longer relies on the WM\_SIZE message to layout its gadgets; the WM\_WINDOWPOSCHANGING message is used instead
- The 'margin' parameter of the TGauge constructor now defaults to 1
- TModule is no longer derived from TEventHandler
- The 'size' parameter of the TSeparatorGadget now defaults to 0, allowing the gadget to dynamically compute the proper value based on the environment settings.
- The constructor of TTextGadget now takes an additional, optional TFont\* as parameter

- TWindow's GetParent now returns a 'TWindow\*'. The 'GetParentH' returns the parent's window handle
- TControlGadget no longer defines the 'void Invalidate(bool = true)' method
- TTextGadgets now uses the 'tmMaxCharWidth' data member of the TEXTMETRICS structure to determine its size [instead of ((tmMaxCharWidth+tmAveCharWidth)/2)]
- TWindow's IdleAction now invokes the 'IdleAction' method of its children.

### **3. New examples directory structure**

The directories containing ObjectWindows examples have been restructured

#### **Directory structure**

#### **Contains**

+ BC5	
+ Examples	
+ OWL	
+ Apps	Samples using multiple OWL classes
+ Classes	Smaller samples illustrating one or two classes
+ Experts	Some standard AppExpert-generated applications
+ Games	Sample Games written with OWL
+ OCF	OLE Container/Server samples
+ Tasks	Tasks-oriented samples
+ Tools	Tools written with ObjectWindows
+ Tutorial	Examples accompanying OWL Tutorial manual
+ WinAPI	Examples exploring various Windows API

### **4. Rebuilding ObjectWindows Library**

The batch file BUILDOWL.BAT located in the SOURCE\OWL directory illustrates the MAKE options used to build the various versions of the ObjectWindows Library. The batch file only utilizes the basic options. You may run the batch file with additional parameters such as DIAGS=1 or CODEGUARD=2 to create the diagnostic or codeguarded versions of the library respectively. NOTE: The ObjectWindows library now contains two assembly modules (DIBBLT16.ASM & DIBBLT32.ASM) which require TASM when rebuilding the library. Prebuilt copies of the object modules are included in the respective subdirectories under the SOURCE\OWL\O directory if you do not have a copy of the Turbo Assembler.

### **5. Known problems**

Some methods of the TUIxxxx helper objects do not have proper emulation support when running under Windows 3.1. Subsequently, some classes, such as TSeparatorGadget may not behave correctly under Windows 3.1.