

PM Controls: Merlin Visuals

- ◆ All controls have new visuals (except value set/)
- ◆ Design objectives:
 - improve visual appearance of existing PM controls
 - no new styles or APIs required to use new look
 - existing applications using controls inherit new appearance
 - no regressions in existing applications
- ◆ Other new visual enhancements:
 - new notebook style - the tabbed dialog
 - new color enhancements
 - new default font
 - close button on title bars

PM Controls: New Notebook

- ◆ New styles:
 - **BKS_TABBEDDIALOG**
 - **BKS_BUTTONAREA**
- ◆ New messages:
 - **BKM_SETTABCOLOR**
 - **BKM_SETNOTEBOOKBUTTONS**
- ◆ New push button style:
 - **BS_NOTEBOOKBUTTON**
- ◆ New push button message:
 - **BM_AUTOSIZE**

PM Controls: New Color APIs

- ◆ Controls obtain colors from following sources (descending order of preference):
 - presentation parameter
 - application control color (*new*)
 - global control color (*new*)
 - default system color
- ◆ Control colors are set and queried via new APIs (user interface available).
- ◆ All PM controls respond to new control colors, will not interfere with existing color schemes.
- ◆ New presentation parameters created for some controls.

PM Controls: New Color APIs

- ◆ Each control has its own *control color type*:
 - » CCT_NOTEBOOK
 - » CCT_SCROLLBAR
 - » CCT_STATICTEXT
 - » CCT_GROUPBOX
- ◆ Each color used by a control type has a *control color index*:
 - » CCI_BACKGROUND
 - » CCI_FOREGROUND
 - » CCI_BORDERLIGHT
 - » CCI_BORDERDARK
- ◆ Not all color indexes are used by all color types
 - e.g. CCT_STATICTEXT does not use CCI_BORDERLIGHT

PM Controls: New Color APIs

- ◆ New color APIs:
 - **WinQueryControlColors**
 - **WinSetControlColors**
- ◆ New color messages:
 - **WM_QUERYCTLTYPE**
 - **WM_CTLCOLORCHANGE**

PM Controls: Other New Features

◆ Close button:

- added automatically to windows which already have a min/max menu on title bar.
- if window does not have min/max menu, can use **FCF_CLOSEBUTTON** to add it.
- removed from titlebar if window does not have enabled “Close” item in the system menu.
- system menu notifies frame whenever the state of the “Close” item is changed.

◆ New default font: **9.WarpSans**

- used in default scheme for menu and titlebar.
- not a feature of new controls - controls use whichever font is used by application or Workplace Shell.

PM Controls: Debugging Visuals

- ◆ New painting code replaced old - no major interface redesign - mostly rewritten functions
- ◆ Most bitmaps replaced by drawing code - allow colors to be changed
- ◆ MAGNIFY is a useful tool to allow you to view visuals close up
- ◆ If you fix a visual defect, make sure that all views of the control are working:
 - e.g. for listbox, combinations of the following:
 - » normal
 - » owner-draw
 - » variable height owner-draw
 - » right-to-left (bidi)
 - » borders/no borders
 - » horizontal scroll on/off

PM Controls: Debugging Notebook

- ◆ Old notebook style should work exactly as in W 3.0 - no differences
- ◆ New notebook style piggy-backed on top of old
 - no minor tabs - only code for major tabs has been added
 - most of the new code follows either:
 - » `if (pBookInstance->ulNotebookStyles & BKS_TABBEDDIALOG)`
 - » `if (pBookInstance->ulNotebookStyles & BKS_BUTTONAREA)`
 - new tab painting functions in **bookdtab.c**
- ◆ Most bugs probably deal with painting tabs correctly
 - important functions which set up data for painting:
 - » **ResetVisPagePtrs**
 - » **ResetTabData**
 - best place to start in most cases:
 - » **GotoPage**

PM Controls: Color Internals

- ◆ All presentation parameters and system colors s work as before where applicable
 - **Note:** PP_BORDERCOLOR is not used in as many places
- ◆ New internal function for querying color in a co
 - **CtlQueryColor** replaces **QueryPresColors**
 - **CtlQueryColor** called by control when it is created, and w color change has occurred.
 - **CtlQueryColor** uses tables set up in **initterm.c** to keep trac which color should be used where.

PM Controls: Color Internals

- ◆ Each color used in controls has unique identifier
 - a constant in **controlp.h**, for example:
 - » **CCV_LISTBOXBGND** : list box background color
 - » **CCV_LISTBOXFGND** : list box foreground (text) color
 - » **CCV_LISTBOXBDRLIGHT** : light outer list box border
 - » **CCV_LISTBOXBDRDARK** : dark outer list box border
 - » **CCV_COMBOBOXBGND** : combo box background color
 - » **CCV_COMBOBOXFGND** : combo box foreground color
 - » **etc.**
 - each constant is used in the controls to identify which color need to query, and maps to an entry in the master table:

```
plbox->clrHiliteFore = CtlQueryColor(hwnd, hps, CCV_LISTBOXHILITEFGND, 0);  
plbox->clrHiliteBack = CtlQueryColor(hwnd, hps, CCV_LISTBOXHILITEBGND, 0);  
plbox->clrTopOuterBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDRDARK, 0);  
plbox->clrTopInnerBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDR2DARK, 0);  
plbox->clrBottomOuterBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDRLIGHT, 0);  
plbox->clrBottomInnerBorder = CtlQueryColor(hwnd, hps, CCV_LISTBOXBDR2LIGHT, 0);
```

PM Controls: Color Internals

- ◆ Master table of color usage is **vCCmap** in **init**
- ◆ Contains entries for all colors used in all controls
 - Four columns:

1. CCI_ index	control color index
2. PP_ index	presentation parameter
3. QPF_ flags	e.g. QPF_INHERIT
4. SYSCLR_ index	system color
or	
RGB_ value	RGB color if system color available.
- ◆ **vCCmap** used to build global and per-process color tables:
 - **vCCglobal** global default colors
 - **vCCapp** per process default colors

PM Controls: Color Internals

- ◆ **vCCapp** entries initialized to -1 which means no process default set - use entry in **vCCglobal**
- ◆ Examples:
 - **Push button background** (**CCV_PUSHBUTTONBGND**)
 - » No presentation parameter
 - » No entry in **vCCapp** (-1)
 - » Entry in **vCCglobal** set to **SYSCLR_BUTTONMIDDLE**Push button will paint with **SYSCLR_BUTTONMIDDLE**
 - **Scroll bar inner border** (**CCV_SCROLLBARBDR2LIGHT**)
 - » Owner has **PP_BORDER2LIGHTCOLOR** set to **RGB_RED**
 - » **vCCapp** contains **RGB_GREEN**
 - » **vCCglobal** contains **RGB_YELLOW**Scroll bar inner border will paint with **RGB_RED** color.

PM Controls: Debugging Colors

◆ Potential problems:

- incorrect colors are used
 - » an entry in **vCCmap** is wrong
- in 256 color mode, palette manager ruins colors in color
 - » customer has set colors which are not in the default 256 set

◆ Where to start:

- set breakpoint on **CtlQueryColor** and trace through to find where the color value is coming from.

PM Controls: Potential APARs

◆ Potential Merlin APARs with new visuals:

– **clipping:**

» most controls have wider borders (2 pels vs. 1 pel) reducing space available inside control. Affects controls with text

◆ **push button**

◆ **list box / combo box**

◆ **entry field**

◆ **spin button**

– **emulated controls:**

» some apps. create controls which look like system controls (e.g. Lotus 123/G menus). User has no way of knowing if it is an application or OS/2 bug

– **visual inconsistency:**

» possible that some apps will not look as good with new visuals

PM Controls: Potential APARs

◆ Other potential APARs:

– close button:

- » Does not work well with Lotus 123/G and Freelance (certain versions). Patch available on system (LOTUSFIX.CMD).

– new default font (9.WarpSans):

- » *not* a system font - will not appear in application dialog boxes, set by application or user.
- » Can be lost when playing around with schemes and presentation notebooks.
- » Old device drivers replace DSPRES.DLL which will display new WarpSans font from system.
- » Controls in new settings notebooks may be clipped in certain resolutions (800x600, 1280x1024, etc). WPS layout problem.

PM Controls: Color Internals

Control Color Index -----	CCT_STATICTEXT -----	CCT_GROUPBOX -----	CCT_EDITTEXT -----
CCI_FOREGROUND	Y	Y	Y
CCI_FOREGROUNDREADONLY			
CCI_BACKGROUND	Y	Y	Y
CCI_BACKGROUNDDDIALOG	Y	Y	
CCI_BORDER			
CCI_BORDERLIGHT		Y	
CCI_BORDERDARK		Y	
CCI_BORDER2			
CCI_BORDER2LIGHT			
CCI_BORDER2DARK			
CCI_BORDERDEFAULT			
.			
.			
.			

PM Controls: Color Internals

Control Color Index

CCT_STATICTEXT

CCT_PUSHB

CCI_FOREGROUND

CCV_STATICTEXTFGND

CCV_PUSHB

CCI_FOREGROUNDREADONLY

CCI_BACKGROUND

CCV_STATICTEXTBGND

CCV_PUSHB

CCI_BACKGROUNDDIALOG

CCV_STATICTEXTBGNDDLG

CCV_PUSHB

CCI_BORDER

CCI_BORDERLIGHT

CCV_PUSHB

CCI_BORDERDARK

CCV_PUSHB

CCI_BORDER2

CCI_BORDER2LIGHT

CCI_BORDER2DARK

CCI_BORDERDEFAULT

CCV_PUSHB

.

.

.

PM Controls: Color Internals

CCV_ Constants -----	Control Color Index -----	Presentation Parameter -----	Presp Flags -----
/* CCV_PUSHBUTTONBGND	*/ CCI_BACKGROUND,	PP_BACKGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONFGND	*/ CCI_FOREGROUND,	PP_FOREGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDISABLEDBGND	*/ CCI_DISABLED_BACKGROUND,	PP_DISABLED_BACKGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDISABLEDFGND	*/ CCI_DISABLED_FOREGROUND,	PP_DISABLED_FOREGROUND_COLOR,	QPF_NOINHERIT
/* CCV_PUSHBUTTONDEFAULTBDR	*/ CCI_BORDERDEFAULT,	PP_BORDERDEFAULT_COLOR,	0,
/* CCV_PUSHBUTTONBDRLIGHT	*/ CCI_BORDERLIGHT,	PP_BORDERLIGHT_COLOR,	0,
/* CCV_PUSHBUTTONBDRDARK	*/ CCI_BORDERDARK,	PP_BORDERDARK_COLOR,	0,
// CCV_PUSHBUTTONFIRST			
// CCV_PUSHBUTTONLAST			

PM Controls Enhancements

Mike Walker - Visuals

Tom Bellwood - DAX

Merle Sterling - Container