

Presentation Parameters and Combination Boxes for OS/2 Version 1.20 by Joe Hayes

Introduction

On January 8, 1990, Microsoft announced and shipped the OS/2 Presentation Manager ToolKit Version 1.20. This development environment, specifically designed for OS/2 and Presentation Manager applications, incorporated the new additions introduced in OS/2 Version 1.20. Two of these new features, Presentation Parameters (presparams) and Combination Boxes (combo boxes) are discussed here and are demonstrated in the sample application PRESPARM.EXE contained within PRESPARM.ARC.

Further information may be obtained from the "Microsoft Operating System/2 Programmer's Reference, Volume 4, "Microsoft Operating System/2 Presentation Manager Softset, Software Tools for Developing OS/2 Presentation Manager Programs", and from QuickHelp included with the OS/2 Programmer's ToolKit Version 1.20.

Presentation Parameters

Presparams are a feature of OS/2 Version 1.2 which control the appearance of a custom dialog box, window, or control when it is first created. Presparams can also be used to change the appearance of the dialog box, window, or control dynamically during program execution. This is a new feature, as by changing the presparam, a paint message is not sent to the dialog box, window, or control. This is handled internally by OS/2 as a final step in output pipeline.

Two examples of the presentation parameters demonstrated in the sample program are color and fonts. These parameters of the dialog box, window, or control can be set in the resource file through the use of the PRESPARAMS, CONTROL, or WINDOW statements. They can also be changed on the fly using the WinSetPresParam() function.

Combination Boxes

Combo boxes are a new feature of OS/2 Version 1.2. They can be thought of as a concatenation of two existing control types, a single line entry field and a list box. A combo box gives the user the flexibility of entering data in the entry field, as they have done in previous OS/2 applications. EM_ messages can also be sent to the combo box entry

field to provide a similar interface to the standard entry field.

The list box portion of the combo box gives the user a standard array of choices which can be selected into the entry field. This gives both the user and the developer some predefined choices which can be selected and optionally edited in the entry field before use in the application. LM_ messages can be sent to the list box portion of the combo box, again to add the standard interface now used for existing list boxes.

Other new CBM_ and CBN_ messages have been added to the ToolKit to give the application developer the ability to manipulate and be notified by the combo box. For example, an application using combo boxes which have drop down lists can display the list of a combo box by sending it a CBM_SHOWLIST message. The user would display the list by clicking on the drop-down icon just to the right of the entry field.

There are three types of combo boxes: simple combo boxes, drop-down combo boxes, and drop-down-list combo boxes. Simple combo boxes are a combination of an entry field and a list box which is always displayed. The user can choose list box selections and/or edit selections from the entry field. Drop-down combo boxes are identical to simple combo boxes, except the list box is not always displayed. The user accesses the list box portion by clicking the drop-down icon. Drop-down-list combo boxes are identical to drop-down combo boxes except that the user cannot edit the list box selections. Drop-down-list boxes are excellent for finite numbers of fixed choices where the selection itself is needed to be displayed after the selection is made.

Sample Program - PRESPARM.ARC

The sample program PRESPARM.EXE is a simple presentation manager window with a "Demo" menu presenting a "Dialog" menu item. By selecting the "Dialog" item, a dialog is displayed demonstrating presentation parameters and combo boxes. The dialog box contains two combo boxes, a list box, three entry fields, three push buttons, and several descriptive text fields.

The combo box toward the upper left of the dialog box is a drop-down combo box. It is used to control the editing of the text field immediately below it. The text field is intended to be the title of the list box. Thus, the user can select one of the pre-defined text items contained in the drop-down combo box list portion and can then edit and/or select that as the title of the list box below. The list box title can be changed by pressing the "Update Display" pushbutton.

Some code for this combo box can be found in the WM_INITDLG: portion of the GetPresParamsDlgProc() function. For example,

```
/* Initialize drop down combo box */

WinSetDlgItemText (hwnd, DID_EDITLIST, szTitle) ;

WinSendDlgItemMsg (hwnd, DID_EDITLIST, LM_INSERTITEM,
    MPFROMSHORT ((SHORT) LIT_END),
    MPFROMP ((PSZ) "Demo Strings:")) ;
WinSendDlgItemMsg (hwnd, DID_EDITLIST, LM_INSERTITEM,
    MPFROMSHORT ((SHORT) LIT_END),
    MPFROMP ((PSZ) "Test Strings:")) ;
WinSendDlgItemMsg (hwnd, DID_EDITLIST, LM_INSERTITEM,
    MPFROMSHORT ((SHORT) LIT_END),
    MPFROMP ((PSZ) "List Strings:")) ;
```

The three entry fields toward the upper right of the dialog box are the RGB values which correspond to three presentation parameters of the list box toward the lower left of the dialog box. By entering different RGB values and selecting "Update Display", the user can then change the color of the respective presentation parameter. The presparam can be selected using the drop-down-list combo box immediately under the RGB entry fields. A drop-down-list combo box was selected for this due to the fact that the choices for the presparams is finite cannot be edited by the user.

Code for the presentation parameters can be found in the WM_COMMAND: portion of the same function. For example,

```
/* Get new color value */

WinQueryDlgItemShort (hwnd, DID_RED, &usTempRed, FALSE) ;
WinQueryDlgItemShort (hwnd, DID_GREEN, &usTempGreen,
FALSE) ;
WinQueryDlgItemShort (hwnd, DID_BLUE, &usTempBlue, FALSE) ;

clr = RGB (usTempRed, usTempGreen, usTempBlue) ;

/* Query type of PresParam to change */

usTempListIndex = (USHORT) WinSendDlgItemMsg (
    hwnd,
    DID_LISTCOMBO,
    LM_QUERYSELECTION,
```

```
MPFROMSHORT ((SHORT) LIT_FIRST),  
0L) ;
```

```
/* Update PresParams */
```

```
WinSetPresParam (hwndListBox,  
idPresParam[usTempListIndex],  
(LONG) sizeof (LONG),  
(PVOID) &clr ) ;
```

To close the dialog box, select the "Save Changes" or "Cancel" push buttons. The "Save Changes" button saves the presparams for the duration of the applications execution.

The PRESPARAMS statement is demonstrated in the resource file "Presparm.rc" to set the font used by the "About" dialog box. To view this dialog box, select the "About" option under the "Demo" menu selection.