

Dialog Code

This appendix defines several procedures used by the Venn Diagrammer application to manage dialog boxes.

```

UNIT Dialog;           {routines to handle dialog boxes}
INTERFACE
    USES
        Global, Utilities, Preferences, VennProcs;

    PROCEDURE DoSetupUserItems (myKind: Integer; VAR myDialog: DialogPtr);
    PROCEDURE DoSetupCtrlValues (myDialog: DialogPtr);
    PROCEDURE DoUserItem (myDialog: DialogPtr; myItem: Integer);
    PROCEDURE DoModelessDialog (myKind: Integer; VAR myDialog: DialogPtr);
    FUNCTION DoHandleDialogEvent (myEvent: EventRecord): Boolean;

IMPLEMENTATION

{DoSetupUserItems: set up application-defined ("user") items in a dialog box}
PROCEDURE DoSetupUserItems (myKind: Integer; VAR myDialog: DialogPtr);
    VAR
        myType:      Integer;
        myHand:      Handle;
        myRect:      Rect;
        count:       Integer;
        origPort:    GrafPtr;
    BEGIN
        GetPort(origPort);
        SetPort(myDialog);

        CASE myKind OF
            rVennDPrefsDial:
                FOR count := 1 TO kVennPrefsItemCount DO
                    IF count IN [iExist1Icon..iExist4Icon,
                                iEmpty1Icon..iEmpty4Icon] THEN
                        BEGIN
                            GetDialogItem(myDialog, count, myType, myHand, myRect);
                            SetDialogItem(myDialog, count, myType, @DoUserItem,
                                            myRect);
                        END;
                OTHERWISE

```

Dialog Code

```

        ;
    END;

    SetPort(origPort);
END;

{DoSetupCtrlValues: install initial values in a dialog}
PROCEDURE DoSetupCtrlValues (myDialog: DialogPtr);
    VAR
        count:      Integer;
        myType:      Integer;
        myHand:      Handle;
        myRect:      Rect;
        origPort:    GrafPtr;
    BEGIN
        IF myDialog = NIL THEN
            exit(DoSetupCtrlValues);

        GetPort(origPort);           {save the current graphics port}
        SetPort(myDialog);           {always do this before drawing}
        ShowWindow(myDialog);

        IF myDialog = gPrefsDialog THEN
            BEGIN
                FOR count := 1 TO kVennPrefsItemCount DO
                    BEGIN
                        GetDialogItem(myDialog, count, myType, myHand,
                                      myRect);
                        IF myType = ctrlItem + radCtrl THEN
                            CASE count OF
                                iExist1Radio..iExist4Radio:
                                    SetCtlValue(ControlHandle(myHand),
                                                  ORD(gExistIndex = count - (iExist1Radio - 1)));
                                iEmpty1Radio..iEmpty4Radio:
                                    SetCtlValue(ControlHandle(myHand),
                                                  ORD(gEmptyIndex = count - (iEmpty1Radio - 1)));
                            OTHERWISE
                                ;
                            END;
                        IF myType = ctrlItem + chkCtrl THEN
                            CASE count OF
                                iGetNextRandomly:
                                    SetCtlValue(ControlHandle(myHand),

```

Dialog Code

```

                                ORD(gStepRandom = TRUE));
iShowSchoolNames:
    SetCtlValue(ControlHandle(myHand),
                ORD(gShowNames = TRUE));
iUseExistImport:
    SetCtlValue(ControlHandle(myHand),
                ORD(gGiveImport = TRUE));
iAutoAdjust:
    SetCtlValue(ControlHandle(myHand),
                ORD(gAutoAdjust = TRUE));
OTHERWISE
    ;
END;
END;
END;

SetPort(origPort);           {restore the previous graphics port}
END;

{DoUserItem: handle drawing of application-defined items in a dialog box}
PROCEDURE DoUserItem (myDialog: DialogPtr; myItem: Integer);
VAR
    myType:      Integer;
    myHand:      Handle;
    myRect:      Rect;
    origPort:    GrafPtr;
BEGIN
    GetPort(origPort);
    SetPort(myDialog);

    GetDialogItem(myDialog, myItem, myType, myHand, myRect);

    IF myDialog = gPrefsDialog THEN
        CASE myItem OF
            iExist1Icon..iExist4Icon:
                BEGIN
                    DoPlotIcon(myRect,
                                GetIcon(kExistID + myItem - iExist1Icon),
                                myDialog, srcCopy);
                END;
            iEmpty1Icon..iEmpty4Icon:
                BEGIN
                    DoPlotIcon(myRect,

```

Dialog Code

```

                                GetIcon(kEmptyID + myItem - iEmptyIcon),
                                myDialog, srcCopy);
                                FrameRect(myRect);
                                END;
                                OTHERWISE
                                ;
                                END; {CASE}

                                SetPort(origPort);           {restore original port}
                                END;

{DoModelessDialog: put up a modeless dialog box}
PROCEDURE DoModelessDialog (myKind: Integer; VAR myDialog: DialogPtr);
VAR
    myPointer: Ptr;
BEGIN
    IF myDialog = NIL THEN           {the dialog box doesn't exist yet}
        BEGIN
            myPointer := NewPtr(sizeof(DialogRecord));
            IF myPointer = NIL THEN
                exit(DoModelessDialog);

            myDialog := GetNewDialog(myKind, myPointer, WindowPtr(-1));
            IF myDialog <> NIL THEN
                BEGIN
                    DoSetupUserItems(myKind, myDialog);      {set up user items}
                    DoSetupCtrlValues(myDialog);              {set up initial values}
                END;
            END
        ELSE
            BEGIN
                ShowWindow(myDialog);
                SelectWindow(myDialog);
                SetPort(myDialog);
            END;
        END;
    END;

{DoHandleDialogEvent: handle events in modeless dialog boxes}
FUNCTION DoHandleDialogEvent (myEvent: EventRecord): Boolean;
VAR
    eventHandled: Boolean;           {did we handle the event?}
    myDialog: DialogPtr;
    myItem: Integer;

```

Dialog Code

```

BEGIN
  eventHandled := FALSE;
  IF FrontWindow <> NIL THEN
    IF IsDialogEvent(myEvent) THEN
      IF DialogSelect(myEvent, myDialog, myItem) THEN
        BEGIN
          eventHandled := TRUE;
          SetPort(myDialog);

          IF myDialog = gPrefsDialog THEN
            BEGIN
              CASE myItem OF
                iEmpty1Radio..iEmpty4Radio:
                  gEmptyIndex := myItem;
                iEmpty1Icon..iEmpty4Icon:
                  gEmptyIndex := myItem - 4;
                iExist1Radio..iExist4Radio:
                  gExistIndex := myItem - iEmpty4Icon;
                iExist1Icon..iExist4Icon:
                  gExistIndex := myItem - (iEmpty4Icon + 4);
                iGetNextRandomly:
                  gStepRandom := NOT gStepRandom;
                iAutoAdjust:
                  gAutoAdjust := NOT gAutoAdjust;
                iShowSchoolNames:
                  gShowNames := NOT gShowNames;
                iUseExistImport:
                  gGiveImport := NOT gGiveImport;
                iSaveVennPrefs:
                  DoSavePrefs;
                OTHERWISE
                  ;
              END;

              DoSetupCtrlValues(myDialog);           {update values}
            END;
          END;

          DoHandleDialogEvent := eventHandled;
        END;
      END.
    END.
  END.

```

