


```

        GA_Previous,      gradslid,
        GA_ID,            7L,
        TAG_END);

if (gradslid && colwheel)
{
    if (Mywindow = OpenWindowTags(NULL, WA_Left,      10,
        WA_Top,        20,
        WA_Height,     200,
        WA_Width,      400,
        WA_Title,      "WheelGrad Window",
        WA_CustomScreen, Myscreen,
        WA_IDCMP,       IDCMP_CLOSEWINDOW | IDCMP_MOUSEMOVE,
        WA_SizeGadget, TRUE,
        WA_DragBar,     TRUE,
        WA_CloseGadget, TRUE,
        WA_Gadgets,     gradslid,
        TAG_DONE))
    {
        mywinsig = 1 << Mywindow->UserPort->mp_SigBit;
        do
        {
            Wait(mywinsig);

            while (msg = (struct IntuiMessage *)GetMsg(Mywindow->UserPort))
            {
                switch (msg->Class)
                {
                    case IDCMP_CLOSEWINDOW:
                        Closeflag = TRUE;
                        break;
                    case IDCMP_MOUSEMOVE:

                        /*
                         * Change gradient slider color each time
                         * colorwheel knob is moved. This is one
                         * method you can use.
                         */

                        /* Query the colorwheel */
                        GetAttr(WHEEL_HSB,colwheel,(ULONG *)&hsb);

                        i = 0;

                        while (i < numPens)
                        {
                            hsb.cw_Brightness =
                                0xffffffff - ((0xffffffff / numPens) * i);
                            ConvertHSBToRGB(&hsb,&rgb);

                            color_list[i].l32_red = rgb.cw_Red;
                            color_list[i].l32_grn = rgb.cw_Green;
                            color_list[i].l32_blu = rgb.cw_Blue;
                            i++;
                        }
                        LoadRGB32(&Myscreen->ViewPort,(ULONG *)color_list);
                        break;
                    }
                ReplyMsg((struct Message *)msg);
            }
        } while (Closeflag == FALSE);
    }
    CloseWindow(Mywindow);
}

/* Get rid of the gadgets */
DisposeObject(colwheel);
DisposeObject(gradslid);

/* Always release the pens */
while (numPens > 0)
{
    numPens--;

```

```

        }
        ReleasePen(Myscreen->ViewPort.ColorMap.penns[numPens]);
    }

    CloseScreen(Myscreen);
    exitvalue = RETURN_OK;
}
else
    printf("Failed to open screen\n");
}

if (gdidres == 0)
    printf("Screen mode dimension information not available\n");

if (displayhandle == NULL)
    printf("Failed to find HIRES_KEY in display database\n");

if (GradientSliderBase)
    CloseLibrary(GradientSliderBase);
else
    printf("Failed to open gadgets/gradientslider.gadget\n");

if (ColorWheelBase)
    CloseLibrary(ColorWheelBase);
else
    printf("Failed to open gadgets/colorwheel.gadget\n");

if (GfxBase)
    CloseLibrary(GfxBase);
else
    printf("Failed to open graphics.library\n");

if (IntuitionBase)
    CloseLibrary(IntuitionBase);
else
    printf("Failed to open intuition.library\n");

exit(exitvalue);
}

```

