

REF

AD-A261 456

Form Approved  
OBM No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and for reviewing this burden, to Washington, DC 20543. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington, DC 20543. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington, DC 20543.

Use this form for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and for reviewing this burden, to Washington, DC 20543. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington, DC 20543.

1. Agency Use Only (Leave Blank)

December 1992

2. Report Type and Dates Covered.

Final - Proceedings

4. Title and Subtitle.

Implementation of Custom Colors in the DECwindows Environment

5. Funding Numbers.

Contract

Program Element No. 0604262

Project No.

Task No.

Accession No. DN257017

Work Unit No. 93512V

6. Author(s).

Stephanie Alfieris Myrick and Perry Bret Wischow

DTIC  
ELECTES MAR 1 1993 D  
C

7. Performing Organization Name(s) and Address(es).

Naval Research Laboratory  
Marine Geosciences Division  
Stennis Space Center, MS 39529-5004

8. Performing Organization

Report Number.

PR 92:085:351

9. Sponsoring/Monitoring Agency Name(s) and Address(es).

Naval Air Systems Command  
Department of the Navy (AIR-803)  
Washington, DC 20361-8030

10. Sponsoring/Monitoring Agency

Report Number.

PR 92:085:351

11. Supplementary Notes.

Published in DECUS.

12a. Distribution/Availability Statement.

Approved for public release; distribution is unlimited.

12b. Distribution Code.

13. Abstract (Maximum 200 words).

This paper describes the implementation of user-defined, or custom, colors in the DECwindows environment. Custom colors can be used to augment the standard color set that is associated with the hardware colormap. The custom color set that is included in this paper consists of 240 distinct colors, each of which is comprised of red, green, and blue intensities. Intensity levels range from zero (no intensity) to 255 (maximum intensity). The DECwindow and Xwindow statements, which are required to implement the custom color set, are identified and described. Some of the tasks performed by these statements include creating and installing a colormap and accessing individual colors. Using a VAX C programming language platform, examples of proper invocation and syntax of these statements are provided. Some familiarization with C, DECwindows and Xwindows is assumed.

14. Subject Terms.

Digital maps, optical storage, databases, data compression

15. Number of Pages.

9

16. Price Code.

17. Security Classification  
of Report.

Unclassified

18. Security Classification  
of This Page.

Unclassified

19. Security Classification  
of Abstract.

Unclassified

20. Limitation of Abstract.

SAR

GAP



SIG

Graphics Applications  
Special Interest Group



DECUS U.S. Chapter  
Fall Symposium  
December 5-11, 1992  
Las Vegas, NV

98

2

25

1992

93-04131



# GR012

## Implementaion of Custom Colors in the DECwindows Environment

N234-236

Monday, December 7, 1992

9:30 A.M.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Graphics

GAP  SIG

Implementation of Custom Colors  
in the DECwindows Environment

Stephanie Alfieris Myrick  
Perry Bret Wischow  
Naval Research Laboratory  
Mapping, Charting and Geodesy Division  
Mapping Sciences Branch

Abstract

This paper describes the implementation of user-defined, or custom, colors in the DECwindows environment. Custom colors can be used to augment the standard color set that is associated with the hardware colormap. The custom color set that is included in this paper consists of 240 distinct colors, each of which is comprised of red, green, and blue intensities. Intensity levels range from zero (no intensity) to 255 (maximum intensity). The DECwindow and Xwindow statements, which are required to implement the custom color set, are identified and described. Some of the tasks performed by these statements include creating and installing a colormap and accessing individual colors. Using a VAX C programming language platform, examples of proper invocation and syntax of these statements are provided. Some familiarization with C, DECwindows and Xwindows is assumed.

## Application Software

```
#define VMS      /* This definition must be declared elsewhere, */
                /* when appropriate. If defined, compilation */
                /* will build for a VMS DECwindows system.    */

#ifdef VMS
#include <decw$include/DwtAppl.h> /* DECwindows Toolkit */
#else
#include <X11/DwtAppl.h>         /* X11 DECwindows Toolkit */
#endif
```

## Variable Declarations

```
#define    MAX_COLORS 240 /* Maximum number of CAC colors.  */

/*****
/* Define colormaps, visuals, Graphics Contexts, etc. */
*****/

static Visual      *defVisual;  ! Points to the default visual
static XVisualInfo *visualList; ! Points to a visual structure
static XVisualInfo visualTemplate; ! Specifies attributes to be
                                   ! matched
static int         num_visuals; ! Number of visual structures
static Colormap    colormap;    ! The hardware colormap
static int         colorcount;  ! The number of CAC colors
static short       colormap_size; ! Size of the hardware
                                   ! colormap

static XColor      *colorcell_defs; ! XColor definition structure
static unsigned long *colors;       ! Points to XColor definition
                                   ! structure

static Display      *colormap_display; ! Display using CAC colors
static Window        colormap_window;  ! Window using CAC colors
```

## Visual Class

```
/* Initialize a count of the maximum number of custom colors. */  
colorcount = MAX_COLORS;
```

```
/* Identify the colormap display and window areas */  
colormap_display = display;  
colormap_window = window;
```

```
/* Get the default visual */  
defVisual = DefaultVisual(colormap_display,0);
```

```
/* Use the default visualid for our template */  
visualTemplate.visualid = defVisual->visualid;
```

```
/* Get the supported visuals.3 */  
visualList = XGetVisualInfo  
    (colormap_display, ! Pointer to Display structure  
     VisualIDMask,    ! Visual mask, elements to match  
     &visualTemplate, ! Attributes to use in matching  
     &num_visuals)    ! Returns number of matching  
                     ! visual structures
```

```
/* This application must use either PseudoColor or DirectColor. */  
if (visualList -> class != PseudoColor &&  
    visualList -> class != DirectColor)  
{  
    printf("Unable to use this Visual Class...\n");  
    exit (0);  
}
```

```
/* Determine that an adequate number of planes */  
/* is available to allow for 256 colors.      */  
if (visualList->colormap_size < colorcount)  
{  
    printf("This visual must support at least %d colors...  
          \n",colorcount);  
}
```

## Loading The Hardware Colormap with Custom Colors

```
/* Associate the default colormap with the colormap display. */
colormap = XDefaultColormap
           (colormap_display,
            DefaultScreen(colormap_display));

/* Associate the default colormap with the specified window.3 */
XSetWindowColormap
  (colormap_display, ! Pointer to Display structure
   colormap_window, ! Window id, for which to set the colormap
   colormap);      ! Specifies the colormap

/* Allocate enough memory to accommodate CAC custom colors. */
colorcell_defs = (XColor *)
                  XtMalloc(sizeof(*colorcell_defs)*colorcount);

/* Allocate space for the XAllocateColorCells call */
colors = (unsigned long *)
          XtMalloc(sizeof(*colors)*colorcount);

/* Allocate the CAC custom colors into colorcells.3 */
if (!XAllocColorCells
    (colormap_display, ! Pointer to Display structure
     colormap,        ! Colormap id
     0,               ! T/F if planes are/are not contiguous
     &dummy,          ! Returns an array of plane masks (unused)
     0,               ! Number of planes in the plan mask array
     colors,          ! Returns an array of HW colormap indices
     colorcount))    ! Number of colorcells allocated
{
  /* Ascertain that all colors were indeed allocated. */
  printf ("Could not allocate enough colors \n");
  exit (0);
}

/* Fill the color definition structure with CAC colors. */
for (i=0; i<colorcount ; i++)
{
  colorcell_defs[i].pixel = colors[i]; ! load a color offset
  colorcell_defs[i].flags = DoRed | DoGreen | DoBlue;
  colorcell_defs[i].red   = rbuf[i]<<8; ! shift red,
  colorcell_defs[i].green = bbuf[i]<<8; ! green, and
  colorcell_defs[i].blue  = gbuf[i]<<8; ! blue over 8 bits
}
```

Loading The Hardware Colormap with Custom Colors (cont)

```
/* Store the newly-filled colorcells in the */
/* hardware color map.          */
XStoreColors
  (colormap_display, ! Pointer to Display structure
   colormap,         ! the color map to use
   colorcell_defs,  ! Color definition structures
   colorcount);     ! Number of XColor structures
```

The hardware colormap offset, for the beginning of custom colors is found within colors[0]. Therefore, if the following instruction is used to contain the offset:

```
colormap_offset = colors[0];
```

then the following instructions can be used to adjust CAC data values to the appropriate hardware colormap address:

```
for (i=0; i<=numpoints; i++)
    plot_buffer[i] += colormap_offset;
```

References

1. M.C. Lohrenz and J.E. Ryan, The Navy Standard Compressed Aeronautical Chart Database, Naval Research Laboratory Report 8 (July 1990).
2. A. Nye, Xlib Programming Manual for version 11, O'Reilly and Associates, Inc., Sebastopol, CA, 1988
3. A. Nye, Xlib Reference Manual for version 11, O'Reilly and Associates, Inc., Sebastopol, CA, 1988

## Figures

Chart Scale	Source Chart
1:2,000,000	Joint Navigation Chart
1:1,000,000	Operational Navigation Chart
1:500,000	Tactical Pilotage Chart
1:250,000	Joint Operational Graphic
1:100,000	Joint Operational Graphic
1:50,000	Topographic Line Map

Figure 1. Chart Scales and Source Charts

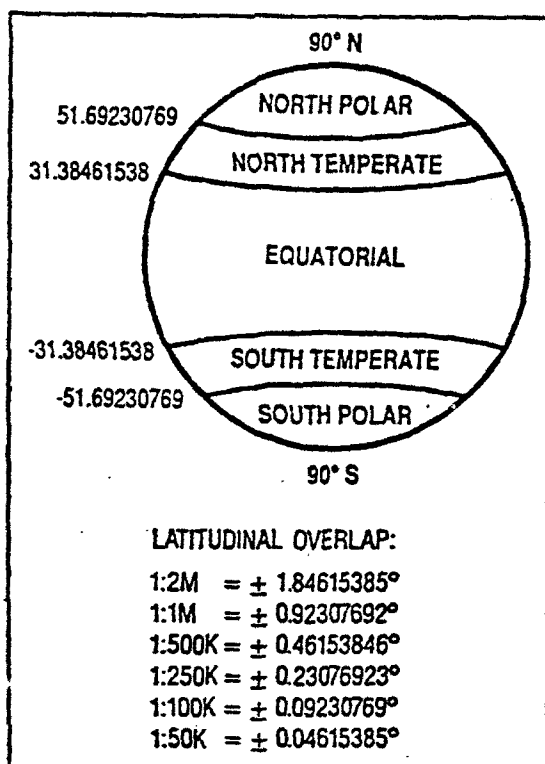


Figure 2 Geographic Zones with Latitudinal Boundaries. <sup>1</sup>

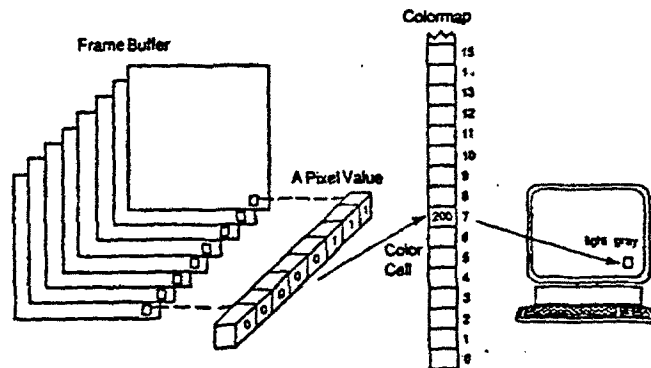


Figure 3 Pixel value to RGB mapping: gray scale and monochrome displays <sup>2</sup>

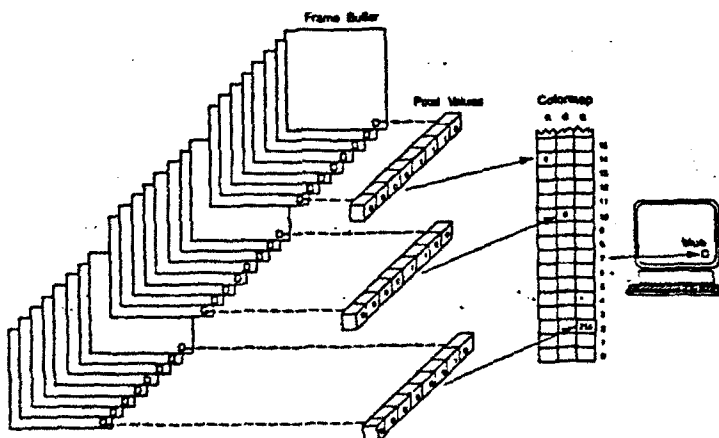


Figure 4 Pixel value to RGB mapping: high-performance color displays <sup>2</sup>

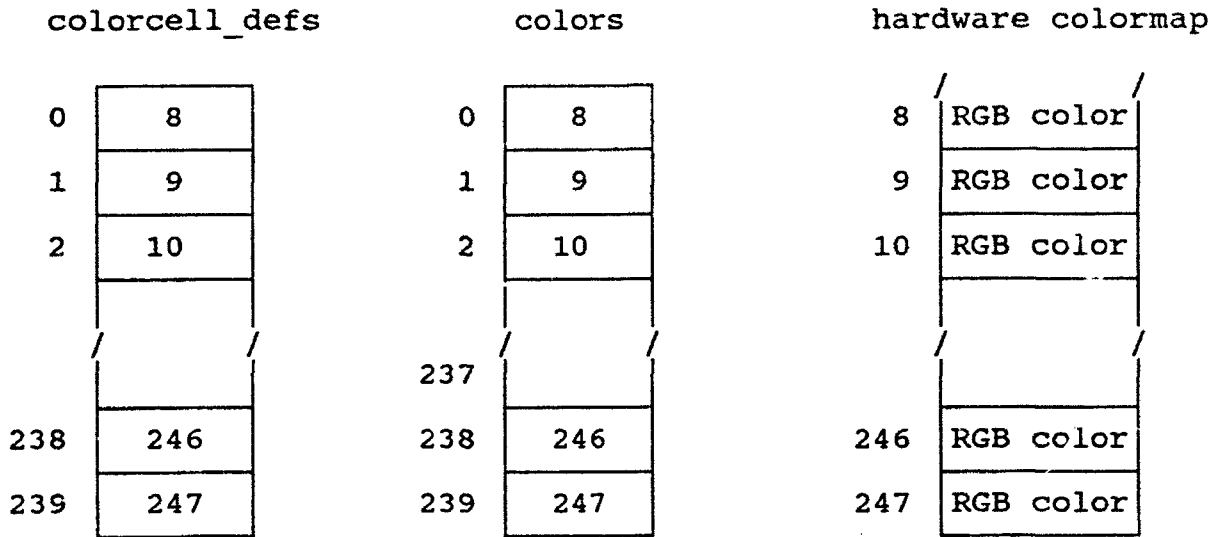


Figure 5. Hardware Colormap Indices