

```

/*****

FnMisc.cp

*****/

/*
    Here are some miscellaneous functions which tend to be used a lot.

    Functions Include:

        FnMisc_ReadPrefs      Reads string from res, converts to int
        FnMisc_SavePrefs      Stores int as a string in res fork
        FnMisc_ColorAvailability Checks for Color QuickDraw
        FnMisc_GetPixelDepth   Returns current monitor setting
        FnMisc_FrameButton     Frames default dialog button
        FnMisc_TitleBarHeight  Returns titlebar height of window
        FnMisc_LeftBorderWidth Returns left border width of window
        FnMisc_RightBorderWidth Returns right (and bottom) border width
*/

// Prototypes

long   FnMisc_ReadPrefs      ( int prefStrID );
void   FnMisc_SavePrefs      ( int prefStrID, long value );
Boolean FnMisc_ColorAvailability ( void );
int     FnMisc_GetPixelDepth  ( GDHandle theDevice );
void     FnMisc_FrameButton   ( DialogPtr theDialog,
                                short buttonID );
int      FnMisc_TitleBarHeight ( WindowPtr w );
int      FnMisc_LeftBorderWidth ( WindowPtr w );
int      FnMisc_RightBorderWidth ( WindowPtr w );

/***** ReadPrefs */

long FnMisc_ReadPrefs( int prefStrID )
/*
    Reads in a string from resource fork specified by prefStrID,
    converts it to an integer, and returns result.
*/

    StringHandle prefStrH;
    Str255        prefStr;
    unsigned char *tempStr;
    int           strLength, defaultResult, i;
    long          result;

    defaultResult = 0;
    if( (prefStrH = GetString( prefStrID )) == NULL )

        result = defaultResult;

    else

        HLock( (Handle)prefStrH );
        strLength = (int)(**prefStrH);
        tempStr = *prefStrH;
        for( i=0; i<=strLength; i++ )

            prefStr[i] = tempStr[i];

        StringToNum( prefStr, &result );

```

```

        HUnlock( (Handle)prefStrH );

return result;

/***** SavePrefs */

void FnMisc_SavePrefs( int prefStrID, long value )
/*
    Takes a value, converts it into a string, and saves it into an
    existing resource 'STR' identified by prefStrID.  The length of
    the string must be less than or equal to the existing string
    length.  If smaller, string is padded with leading spaces so
    string length in resource is left unchanged.
*/

    StringHandle  prefStrH;
    Str255        prefStr, valueStr;
    unsigned char *tempStr;
    int           strLength, numLength, i;

    if( (prefStrH = GetString( prefStrID )) == NULL )

        // do nothing, string doesn't exist (add error routine?)

    else

        HLock( (Handle)prefStrH );
        strLength = (int)(**prefStrH);
        tempStr = *prefStrH;
        NumToString( value, valueStr );
        numLength = (int)(*valueStr);
        if( numLength <= strLength )

            for( i=1; i<=strLength; i++ )
                tempStr[i] = ' ';
            for( i=(strLength - numLength + 1); i<=strLength; i++ )
                tempStr[i] = valueStr[i-(strLength-numLength)];

        ChangedResource( (Handle)prefStrH );
        WriteResource( (Handle)prefStrH );
        HUnlock( (Handle)prefStrH );

/***** ColorAvailability */

Boolean FnMisc_ColorAvailability( void )
/*
    Checks to see if the current machine supports Color QuickDraw.  Use
    this routine once at the beginning of your program.

    This routine is obsolete, use gSysConfig.hasColorQD variable from
    Argus Starter : Main.cp.  Kept here for versions 2.x of Argus
    Libraries for compatability.
*/

    SysEnvRec mySystem;

    SysEnvirons( 2, &mySystem );

```

```
return( mySystem.hasColorQD );
```

```
/****** GetPixelDepth */
```

```
int FnMisc_GetPixelDepth( GDHandle theDevice )
```

```
/*
```

Returns the current pixel depth setting of the machine. Since the user can change the setting of the pixel depth on-the-fly (using the 'Monitor' control panel), this routine should be called each time you do any drawing.

Example Usage:

```
GDHandle gCurrentDevice;  
int      gPixelDepth;
```

```
gCurrentDevice = GetDeviceList();  
gPixelDepth = GetPixelDepth( gCurrentDevice );
```

```
*/
```

```
PixMapHandle screenPMapH;  
int          pixelDepth;
```

```
screenPMapH = (**theDevice).gdPMap;  
pixelDepth = (**screenPMapH).pixelSize;  
return( pixelDepth );
```

```
/****** FrameButton */
```

```
/*
```

Frames a button (usually the OK button) in a dialog.

```
*/
```

```
void FnMisc_FrameButton( DialogPtr theDialog, short buttonID )
```

```
const int kButtonFrameInset = -4;  
const int kButtonFrameSize = 3;  
const int kFilletSize = 16;
```

```
short itemType;  
Rect itemRect;  
Handle itemHandle;  
PenState thePnState;  
GrafPtr oldPort;
```

```
GetPort( &oldPort );  
SetPort( theDialog );  
GetDItem( theDialog, buttonID, &itemType, &itemHandle, &itemRect );  
GetPenState( &thePnState );  
PenNormal();  
PenSize( kButtonFrameSize, kButtonFrameSize );  
InsetRect( &itemRect, kButtonFrameInset, kButtonFrameInset );  
FrameRoundRect( &itemRect, kFilletSize, kFilletSize );  
SetPenState( &thePnState );  
SetPort( oldPort );
```

```
/****** TitleBarHeight */
```

```
/*
```

Returns height of window titlebar in pixels. Reference IM

'Macintosh Toolbox Essentials', listing 4-12, page 4-55.

```
*/
int FnMisc_TitleBarHeight( WindowPtr w )

int      titleBarHeight;
Rect      wRect;
WindowPeek wPeek;
GrafPtr   oldPort;
Point     pt;

GetPort( &oldPort );
SetPort( w );
wRect = w->portRect;
pt.h = wRect.left;
pt.v = wRect.top;
LocalToGlobal( &pt );
wRect.left = pt.h;
wRect.top = pt.v;
wPeek = (WindowPeek)w;
// following line different from IM in that '1' is not subtracted
// from equation.  Count the bits yourself, it works.
titleBarHeight = wRect.top - (*(wPeek->strucRgn))->rgnBBox.top;
SetPort( oldPort );
return titleBarHeight;
```

```
/****** LeftBorderWidth */
/*
Returns width of left window border in pixels.
*/
int FnMisc_LeftBorderWidth( WindowPtr w )
```

```
int      borderWidth;
Rect      wRect;
WindowPeek wPeek;
GrafPtr   oldPort;
Point     pt;

GetPort( &oldPort );
SetPort( w );
wRect = w->portRect;
pt.h = wRect.left;
pt.v = wRect.top;
LocalToGlobal( &pt );
wRect.left = pt.h;
wRect.top = pt.v;
wPeek = (WindowPeek)w;
borderWidth = wRect.left - (*(wPeek->strucRgn))->rgnBBox.left;
SetPort( oldPort );
return borderWidth;
```

```
/****** RightBorderWidth */
/*
Returns width of right window border in pixels.
*/
int FnMisc_RightBorderWidth( WindowPtr w )
```

```

int      borderWidth;
Rect      wRect;
WindowPeek wPeek;
GrafPtr   oldPort;
Point     pt;

GetPort( &oldPort );
SetPort( w );
wRect = w->portRect;
pt.h = wRect.right;
pt.v = wRect.bottom;
LocalToGlobal( &pt );
wRect.right = pt.h;
wRect.bottom = pt.v;
wPeek = (WindowPeek)w;
borderWidth = (*(wPeek->strucRgn))->rgnBBox.right - wRect.right;
SetPort( oldPort );
return borderWidth;

// End of File

```