PMDIB 1.0: A new D.I.B. Manager

PMDIB 1.0 is one of the first image processing software
for developers under OS/2 Presentation Manager
that supports DIB's: Device Independent Bitmaps
created under PM or Windows 3.

INTRODUCTION TO DIB's

The introduction of Device Independent Bitmaps (DIB) was one of the most significant enhancements to the *Microsoft Windows 3* Graphics Device Interface. Now they are also supported under *OS/2 Presentation Manager* by the use of different programming levels. One of them is by using PMDIB 1.0.

A DIB is defined with a color table that is not dependent to the color capabilities of the output device, normally the screen. In contrast, a device-specific bitmap, uses only the available colors on the display device. What PMDIB 1.0 offers here, is the opportunity to add bitmap graphics (monochrome or color) to your applications. You can easily display DIB file format bitmaps and forget about the color limitations of your output device. Of course, the same bitmap will be displayed more accurately under a 8514/A than on an VGA adapter, but that is work for PMDIB 1.0.

With PMDIB 1.0, you can display up to 4 DIB formats: Monochrome, 16-color, 256-color, or full 24-bit RGB bitmaps. The advantage here is that if you have a 256-color DIB that you want to display under an VGA adapter, PMDIB 1.0 will notify Presentation Manager upon this, making the bitmap be displayed using 16 colors with the most accurate color degradation.

PRODUCT DESCRIPTION:

PMDIB 1.0 allows you to display bitmaps in a PM window without the need to call a single PM API.. PMDIB 1.0 window is created with a single API call that allows you to create, refresh, destroy, reposition, as well as modify bitmap display options with subsequent API calls.

You can use PMDIB for your applications as if you controlled the graphic window dynamically, at the time you most prefer, and with the visual effects that you most desire.

PMDIB uses a verb code and additional data parameters to display a graphic window with a bitmap in it

The supported actions for PMDIB are :

Load and display a bitmap

Refresh window contents

Specify a new bitmap source to display

Contract / expand window dimensions, according to bitmap

Specify a new location for the window, with dimensions if desired:

Modify display options, without the need to use its built in popup menus.

PMDIB 1.0 can extract the bitmap information from two sources:

a)

b)
\\computer\resource\subdir\filename.ext

PMDIB 1.0 is a callable DLL routine that is compatible from any OS/2 high or low-level language, and it is supported under OS/2 1.x and OS/2 2.0.

PMDIB API Invocation - C Syntax:

USHORT PMDIBDLL (PSZ pszBitmapSource, USHORT usVerb, PRECTL prclRectangle) ;

PSZ pszBitmapSource - (input)
This parameter takes two forms, depending on the verb code. When you use *DIB_CREATE* or *DIB_REFRESH_NEW_BMP*, this parameter specifies the path for a DIB filename. It can be used under two syntax rules:

Or you can also use UNC (Universal Network Convention) path to locate a remote file. This uses the form:

When the verb is *DIB_CREATE_BYPTR* or *DIB_REFRESH_NEW_BMP_BYPTR*, this parameter points to a buffer where the hole bitmap is coded, including its header information (wether *OS/2* or *Windows 3* version). The maximum lenght for this buffer is 64K.

USHORT usVerb - (input)
A verb can be one of the following:

DIB_CREATE_BMP
Use this verb as the first call to *PMDIBDLL*(). A new window will be created, and the bitmap graphic file specified in *pszBitmapSource* will be drawn in the window. Its initial window placement will be in the lower-left corner of the Presentation Manager desktop and its dimensions adjusted to the bitmap file, unless you have passed a PRECTL as the third parameter, in which case the window will be adjusted to those coordinates and dimensions specified under PRECTL using universal coordinates.

DIB_REFRESH
This verb does not use any other parameter. It will redraw its window contents.

DIB_INVERSE
Inverts the Bitmap colors. This option is available on monochrome and color bitmaps.

DIB_ACTIVATE
Use this verb to activate PMDIB window, and refresh its graphical contents. Normally it is used when there are other windows behind PMDIB that break the graphic bitmap. This message implies that a graphic redraw be performed.

DIB_REFRESH_NEW_BMP
Use this verb to specify a new bitmap file to show inside the window. The new bitmap will be replaced in the window, but its dimensions will be kept like before, unless you pass a PRECTL as the third parameter, in which case the window will be adjusted to those coordinates and dimensions specified under PRECTL using universal coordinates.

DIB_WND_BMP_STRETCH
The window dimensions will be stretched to those the bitmap specifies.

DIB_BMP_WND_STRETCH_ON
The bitmap will be stretched to the window dimensions.

DIB_BMP_WND_STRETCH_OFF
The bitmap will NOT be stretched to the window dimensions. It will be shown using its default dimensions.

DIB_MINIMIZE
The PMDIBDLL Window will be minimized, and its icon will be showed on PM icon list.

DIB_MAXIMIZE
PMDIBDLL window will get the hole PM screen, putting itself in maximized mode.

DIB_RESTORE
The window is restored to its remembered position coordinates when it is in minimized or maximized state.

DIB_EMPTY_WND
This will clear PMDIBDLL window, making it empty. The bitmap will be destroyed, so to redisplay the last bitmap, or whatever other you want, you must call verb *DIB_REFRESH_NEW_BMP* or *DIB_REFRESH_NEW_BMP_BYPTR*.

DIB_DESTROY_WND
PMDIBDLL Window will be destroyed and will disappear from PM window and task manager. To make it come out again, you must call verb *DIB_CREATE_BMP* or *DIB_CREATE_BMP_BYPTR*.

DIB_HIDE_WND
PMDIBDLL window will be made unvisible, but not destroyed. Use *DIB_SHOW_WND* to show it again.

DIB_SHOW_WND
Will show PMDIBDLL window when it has been hide using verb *DIB_HIDE_WND*.

DIB_POSITION_RECTL
Pass with this verb, the (RECTL *) structure to specify the new window placement, in absolute coordinates (those that PM Screen uses).

DIB_REPOSITION_INITIAL
Window will be placed in the PM screen lower-left corner, and its dimensions will be adjusted to the bitmap dimensions, unless you have passed a PRECTL as the third parameter to *DIB_CREATE_BMP*, in which case the window will be adjusted to those coordinates and dimensions specified under PRECTL using universal coordinates.

DIB_ENABLE_POPUPMENU
Mouse second button pops up image control menu.

DIB_DISABLE_POPUPMENU
Mouse second button does nothing. popup menu does not exist.

DIB_LOCK_UPDATE
This verb prevents a window from updating. While the window is locked, no drawing will take place on the screen. Use *DIB_UNLOCK_UPDATE* to repaint the window image contents.

DIB_UNLOCK_UPDATE
The window locking that was previously issued using *DIB_LOCK_UPDATE* verb, is now unlocked, thus allowing itself to repaint its invalidated window areas.

DIB_CREATE_BMP_BYPTR ***(pszBitmapSource points to DIB buffer)***
Use this verb as the first call to *PMDIBDLL*(). A new window will be created, and the bitmap graphic buffer pointed to by the paramter *pszBitmapSource* will be drawn in the window. Its initial window placement will be in the lower-left corner of the Presentation Manager desktop and its dimensions adjusted to the bitmap file, unless you have passed a PRECTL as the third

parameter, in which case the window will be adjusted to those coordinates and dimensions specified under PRECTL using universal coordinates.

DIB_REFRESH_NEW_BMP_BYPTR ***(pszBitmapSource points to DIB buffer)***
Use this verb to specify a new bitmap to show inside the window. The new bitmap will be replaced in the window, but its dimensions will be kept like before, unless you pass a PRECTL as the third parameter, in which case the window will be adjusted to those coordinates and dimensions specified under PRECTL using universal coordinates.

PRECTL prclRectangle - (input)
This is a pointer to a RECTL structure. You will use this parameter with the following verb calls only:

Any other verb ignores this parameter, you should supply a NULL pointer, explicitly.

The structure of a RECTL data type is defined as follows.

USHORT rc - (return)
PMDIBDLL Return code list:

DIBERR_BMP_NOTFOUND
DIBERR_BMP_ERROR_READING
DIBERR_BMP_BAD_SIGNATURE
DIBERR_BMP_CORRUPTED
DIBERR_BMP_ERROR_READ_COLORS
DIBERR_BMP_LARGER_64K
DIBERR_PMDIBDLL_EXISTS
DIBERR_RESOURCES_NOTFOUND
DIBERR_PMDIBDLL_NOEXISTS
DIBERR_THREAD_ERROR
DIBERR_RECTL_ERROR
DIBERR_RECTL_ERROR

DIBERR_VERB_UNKNOWN
DIBERR_OK

PMDIB API Invocation - COBOL Syntax:

As to be able to support the verb codes as constant strings in your program instead of USHORT values, you should include in your Cobol main program, the file *PMDIB.CPY* at the beginning of your source, like this:

```
COPY "PMDIB.CPY".
```

Parameter variables under Cobol should be defined as follows:

```
01 PARAMETERS.
   05 RECTANGLE-POINTER        PIC 9(9) COMP-5 VALUE 0.
   05 USVERB                  PIC 9(4) COMP-5.
   05 IMAGE-POINTER            USAGE IS POINTER.
```

Parameters to DLL under cobol are coded last to first, so the API call to PMDIB 1.0 should be coded like this:

```
CALL "PMDIBDLL" USING     BY VALUE RECTANGLE-POINTER,
                          BY VALUE USVERB,
                          BY REFERENCE IMAGE-POINTER.
```

In this example, the verb *DIB-CREATE-BMP-BYPTR* is used. Before this call, you should move the Bitmap information into this buffer, an then pass the address to the first item of this buffer to *PMDIBDLL*().

Note that the RC value will be placed in RETURN-CODE internal Cobol variable. You should test the return code using the formula below:

```
 IF RETURN-CODE NOT EQUAL DIBERR-OK
*      Process error here
 ELSE
*      API call has been successfull
 ENDIF.
```

The values in Cobol for the PMDIB action verb codes are listed below. This is *PMDIB.CPY* file, actually, the one you include with a *COPY* Cobol statement in your source.

```
*
* PMDIBDLL COPY FILE FOR COBOL
*
* This include file contains the necessary verb codes
* and error codes for calling PMDIBDLL,
*
* (c) Midori S.A. 1991
* by Skarbat@informat
```

```
*
*
* ------------------------------------------------
*
*   Below follows the PMDIBDLL verbs definition
*
 01 DEFINE.
*
   05 DIB-CREATE-BMP           PIC 9(9) COMP-5 VALUE 5000.
   05 DIB-REFRESH              PIC 9(9) COMP-5 VALUE 5010.
   05 DIB-INVERSE              PIC 9(9) COMP-5 VALUE 5020.
   05 DIB-ACTIVATE             PIC 9(9) COMP-5 VALUE 5030.
   05 DIB-REFRESH-NEW-BMP      PIC 9(9) COMP-5 VALUE 5040.
   05 DIB-WND-BMP-STRETCH      PIC 9(9) COMP-5 VALUE 5050.
   05 DIB-BMP-WND-STRETCH-ON   PIC 9(9) COMP-5 VALUE 5060.
   05 DIB-BMP-WND-STRETCH-OFF  PIC 9(9) COMP-5 VALUE 5070.
   05 DIB-MINIMIZE             PIC 9(9) COMP-5 VALUE 5080.
   05 DIB-MAXIMIZE             PIC 9(9) COMP-5 VALUE 5090.
   05 DIB-RESTORE              PIC 9(9) COMP-5 VALUE 5100.
   05 DIB-EMPTY-WND            PIC 9(9) COMP-5 VALUE 5110.
   05 DIB-DESTROY-WND          PIC 9(9) COMP-5 VALUE 5120.
   05 DIB-HIDE-WND             PIC 9(9) COMP-5 VALUE 5130.
   05 DIB-SHOW-WND             PIC 9(9) COMP-5 VALUE 5140.
   05 DIB-POSITION-RECTL       PIC 9(9) COMP-5 VALUE 5150.
   05 DIB-REPOSITION-INITIAL   PIC 9(9) COMP-5 VALUE 5160.
   05 DIB-ENABLE-POPUPMENU     PIC 9(9) COMP-5 VALUE 5170.
   05 DIB-DISABLE-POPUPMENU    PIC 9(9) COMP-5 VALUE 5180.
   05 DIB-LOCK-UPDATE          PIC 9(9) COMP-5 VALUE 5190.
   05 DIB-UNLOCK-UPDATE        PIC 9(9) COMP-5 VALUE 5200.
   05 DIB-CREATE-BMP-BYPTR     PIC 9(9) COMP-5 VALUE 5210.
   05 DIB-REFRESH-NEW-BMP-BYPTR PIC 9(9) COMP-5 VALUE 5220.
*
*   Below follows the PMDIBDLL return code error constants
*
 01 DEFINE.
*
   05 DIBERR-BMP-NOTFOUND         PIC 9(9) COMP-5 VALUE  6010.
   05 DIBERR-BMP-ERROR-READING    PIC 9(9) COMP-5 VALUE  6020.
   05 DIBERR-BMP-BAD-SIGNATURE    PIC 9(9) COMP-5 VALUE  6030.
   05 DIBERR-BMP-CORRUPTED        PIC 9(9) COMP-5 VALUE  6040.
   05 DIBERR-BMP-ERROR-READ-COLORS PIC 9(9) COMP-5 VALUE  6050.
   05 DIBERR-BMP-LARGER-64K       PIC 9(9) COMP-5 VALUE  6060.
   05 DIBERR-PMDIBDLL-EXISTS      PIC 9(9) COMP-5 VALUE  6070.
   05 DIBERR-RESOURCES-NOTFOUND   PIC 9(9) COMP-5 VALUE  6080.
   05 DIBERR-PMDIBDLL-NOEXISTS    PIC 9(9) COMP-5 VALUE  6090.
   05 DIBERR-THREAD-ERROR         PIC 9(9) COMP-5 VALUE  6100.
   05 DIBERR-RECTL-ERROR          PIC 9(9) COMP-5 VALUE  6110.
```

```
    05 DIBERR-VERB-UNKNOWN          PIC 9(9) COMP-5 VALUE  6120.
    05 DIBERR-OK                    PIC 9(9) COMP-5 VALUE  8000.
*
*   End of copy file.
*
```

Statements of Development
In our development politics, we facilitate to our users, minimal cost upgrades (based on delivery rates) as well as information on new actualizations as soon as they come out.

We offer technical service for any doubt or problem that implies the use or implantation of our products by telephone contact or through the cyberspace, by using email, reaching then, all the users of our applications.

In preparation for a new version of PMDIB, the following items are being used now:

it available to develop distributed applications.

List Prices

Our list prices for PMDIB 1.0 are the following:

		Final Distributors
A copy of PMDIB 1.0, for single use	$90, 49£	$110, 65£
A copy of PMDIB 1.0, unlimited development license	$330, 195£	$250, 152£

Final Notes
MIDORI S.A. is a registered trademark of MIDORI S.A.
PMDIB 1.0 is a registered trademark of MIDORI S.A.
Microsoft C 6.0 is a registered trademark of Microsoft Corp.
Windows 3 is a registered trademark of Microsoft Corp.
OS/2 is a registered trademark of IBM Corp. & Microsoft Corp.

This is an evaluation copy of a final version. It will allow to be operative in your workstation for certain random time, normally not more than half an hour. After it will invisibly disappear from the screen. You can start it again, although.

If you like this product, and are interested to contact us in some way, do not hesitate to drop us a line, fax or email.

MIDORI S.A.
Calabria, 241 - Entlo. 5ª
Tel: 93-419 12 37
Fax: 93-430 74 08
08029 Barcelona

Internet: 100021.2114@compuserve.com
CompuServe ID: 100021,2114