.DRW

PalDRAW File Format Specification

release 1.0

Document No : DRW_FF10

To contact us :

NAR Software
Apartado 15056
28080 MADRID
SPAIN

e-mail : aroch@geocities.com
www : http:\\ourworld.compuserve.com\homepages\ariel_rocholl\palmtop.htm

CONTENTS

1OVERVIEW

This document include all the required information to understand, read and write DRW PalDRAW native files.

1.1DRW FILE FORMAT REQUIREMENTS

Primary requirements :

1. To provide a file format to be used as native in the PalDRAW[1] application.
2. To be very compact in terms of memory/disk usage.
3. To be able to be easily edited by humans.
4. To be able to be easily edited/created by software processes.
5. To be easily transmitted by e-mail, forum messages and fax.
6. To be consistent with future improvements.
7. To have all the information needed for a PalDRAW draw, not needing extra file information.

Deducted requirements :

1. To be ASCII (from primary 3, 4, 5 reqs.)
2. To not include duplicated information (from primary 2, 7)

1.2FUTURE ENHANCEMENTS

The DRW file format will be improved in future PalDRAW releases, but always will be backward compatible (So, a DRW created now, will be readable by future PalDRAW software versions).

We will document the DRW file format with any new PalDRAW release.

2FILE HEADER

The DRW file have a header line with a fixed format, to allow identify the current DRW file version. In the present release (1.0), the header is

```
#PalDRAW file
```

Any other character after those, in this line, will be ignored by PalDRAW.

This must be the first line in the file.

3THE STATUS SECTION

The Status section is optional. It indicates the internal PalDRAW variables to know how to draw the entities.

[1] *PalDRAW* is a Draw/CAD application specifically designed to be used in HP100/200 LX palmtops. See our web site for more information. In particular, this file format has nothing to do with the DRW file format of other software companies.

While the Section itself is optional, the inner variables are not. If you specify this section, all parameters must be placed, an the order IS significant (to accelerate the reading process).

The section header is

```
[Status]
```

and the parameters are

Posi-tion	Type	Parameters
1	**G** (grid)	xGridStep yGridStep xGridVisible yGridVisible Snap GridVisible
2	**P** (position)	xPos yPos xLeft yBottom xRight yTop Width Height
3	**S** (switches)	QuickMode FillGlobal Cursor Coordinates
4	**E** (scale)	ScaleValue SelectorSize
5	**N** (new entity values)	TextSize CircleRadius BlockXScale BlockYScale BlockRotation StartAngle EndAngle
6	**M1** (memory 1)	xPos yPos
...	...	...
14	**M9** (memory 9)	xPos yPos

where

Type Name	Parameter Name	Format	Default Value	Meaning
G	xGridStep	real	1.0	The grid x step value to use with Snap
	yGridStep	real	1.0	The grid y step value to use with Snap
	xGridVisible	real	1.0	The grid x step value
	yGridVisible	real	1.0	The grid y step value
	Snap	integer	1	The snap status. (1) is on, (0) is off
	GridVisible	integer	1	The grid visibility status. (0 or 1)
P	xPos	real	10.0	The current x cursor position
	yPos	real	3.0	The current y cursor position
	xLeft	real	-0.1	The x value of the left-bottom corner actual screen window
	yBottom	real	-0.1	The y value of the left-bottom corner actual screen window
	xRight	real	(637/ScaleValue) +xRight	The x value of the right-top corner actual screen window
	yTop	real	178/ (0.826*ScaleValue)+yBottom	The y value of the right-top corner actual screen window
	Width	real	xRight-xLeft	The x screen window size
	Height	real	yTop-yBottom	The y screen window size
S	QuickMode	integer	0	The QuickMode status. (0 or 1)
	FillGlobal	integer	0	The FillGlobal status, to use when creating new entities. (0 or 1)
	Cursor	integer	1	The Cursor visibility status.(0 or 1)
	Coordinates	integer	1	The cursor Coordinates visibility (0 or 1)
E	SelectorSize ScaleValue	real real	10/ScaleValue 32	The selector cursor size The current visual scale value

Type Name	Parameter Name	Format	Default Value	Meaning
N	TextSize	real	0	The text height. When the value is 0, then PalDRAW uses P::Height/100
	CircleRadius	real	0	The default circle/arc radius to use in new entities. When the value is 0, PalDRAW uses P::Height/4
	BlockXScale	real	-	The global x scale value to use in new blocks. (Ignored in release 1.0)
	BlockYScale	real	-	The global y scale value to use in new blocks. (Ignored in release 1.0)
	BlockRotation	real	-	The global rotation angle to use in new blocks. (Ignored in release 1.0)
	StartAngle	real	0	The global start angle (degrees) to use in new arcs
	EndAngle	real	120	The global start angle (degrees) to use in new arcs
M1-M9	xPos	real	-	A memorized x cursor posicion
	yPos	real	-	A memorized y cursor posicion

4THE *BLOCKS* SECTION

The blocks section is optional. If exists, It contains all blocks definitions used in the file.

This section starts with the section identifier

```
[Blocks]
```

and contains block definitions.

The order in this section is not significant.

Each block definition starts with a block definition header, with the next format

```
*BlockName EntityNumbers
```

where

Type Name	Parameter Name	Format	Default Value	Meaning
-	BlockName	string	-	The block identifier
-	EntityNumbers	string	-	The total used entities inside the block definition.

After the header, each block definition have all entities it defines. The entity format is full compatible with that described below in the Entities section.

5THE *ENTITIES* SECTION

The entity section starts with the header

```
[Entities]
```

and contains all the entities of the draw.

The order in this section is not significant.

Each entity has a definition as follows

```
EntityName EntityParam1 ... EntityParamN
```

where `EntityName` can be:

Posi-tion	Type	Parameters
-	A (Arc)	x y Radius StartAng EndAng
-	C (Circle)	x y Radius Filled
-	I (Insertion Block)	x y BlockName RotationAngle XScale YScale
-	L (Line)	xStart yStart xEnd yEnd
-	N (ICN Icon)	x y IconName InsertionScale
-	T (Text)	x y TextHeight TextStyle Content

where:

Type Name	Parameter Name	Format	Default Value	Meaning
A	x	real	-	x center coordinate
	y	real	-	y center coordinate
	Radius	real	-	Arc Radius
	StartAngle	real	-	Initial Angle
	EndAngle	real	-	End Angle
C	x	real	-	x center coordinate
	y	real	-	y center coordinate
	Radius	real	-	Circle Radius
	Filled	integer	-	Fill status. (0 or 1)
I	x	real	-	x Insertion point
	y	real	-	y Insertion point
	BlockName	string	-	The block name identifier
	RotationAngle	real	-	The rotation block angle in degrees. (ignored in release 1.0)
	XScale	real	-	The insertion block x scale value
	YScale	real	-	The insertion block y scale value
L	xStart	real	-	The x initial point value
	yStart	real	-	The y initial point value
	xEnd	real	-	The x end point value
	yEnd	real	-	The y end point value

Type Name	Parameter Name	Format	Default Value	Meaning
N	x	real	-	x Insertion point
	y	real	-	y Insertion point
	IconName	string	-	The icon name identifier
	InsertionScale	real	-	The scale value at the icon must be represented on bitmap format
T	x	real	-	x Insertion point
	y	real	-	y Insertion point
	TextHeight	real	-	The text height value
	TextStyle	string	-	A single word (string without spaces between chars) with the text style value. (in release 1.0, only "S" is valid, to indicate "Small" font)
	Content	string	-	A string with chars and spaces, containing the text string to be represented

6FILE EXAMPLE

This is a file generated with PalDRAW 1.00 :

```
#PalDRAW file Dec 31 1995
[Status]
G 1 0.5 1 0.5 1 0
P 6 4.5 3.632422 1.583565 13.585547 4.950696 9.953125 3.367131
S 0 0 1 0
E 64 0.15625
N 0.015 2 1 1 0 0 120
M0 0 0
M1 0 5
M2 5 0
M3 5 5
M4 10 0
M5 10 5
M6 15 0
M7 15 5
M8 10 3
[Blocks]
* COMPFLOW 14
L -1 0.5 -1 -0.4
L -1 -0.4 -0.1 -0.4
L -0.1 -0.4 -0.1 0.5
L -0.1 0.5 -1 0.5
L -0.9 -0.4 -0.9 -0.5
L -0.9 -0.5 0 -0.5
L 0 -0.5 0 -0.1
L 0 0.4 -0.1 0.4
L 0 -0.2 -0.1 0
L -0.1 0 0 0.2
L 1 -0.1 0 -0.1
L 0 0.1 1 0.1
L 1 0.1 1 -0.1
L 0 0.1 0 0.4
* HARROW2 7
L 0 0 0.1 0.2
L 0.1 0.2 0.1 0.1
L 0.1 0.1 1 0.1
L 1 0.1 1 -0.1
L 1 -0.1 0.1 -0.1
L 0.1 -0.1 0.1 -0.2
L 0.1 -0.2 0 0
* DOB-FLOW 8
L 0.1 0 1 0
L 0.9 1 0 1
L 0 0.1 0.9 0.1
L 0 1 0 0.1
```

```
L 0.1 0.1 0.1 0
L 1 0 1 0.9
L 0.9 0.1 0.9 1
L 1 0.9 0.9 0.9
[Entities]
I 10 4.3 COMPFLOW 0 1 1
I 10 2 COMPFLOW 0 1 1
N 9.1 4.1 ANTENNA 64
N 9.1 1.75 LOUDSPEA 64
N 12 2.8 GENTELME 64
N 9.1 2.8 MAIL 64
I 10 3.1 COMPFLOW 0 1 1
L 11.1 2 11.1 4.3
I 11.2 3.1 HARROW2 0 0.6 1
T 6 4.5 0.05 S Human
T 6 4 0.05 S Communication
T 6 3 0.03 S An example DRW file
T 6 2.70505 0.03 S using ICN icons
T 12.4 2.5 0.03 S Human
T 10.6 2.3 0.015 S SOUND/VOICE
T 10.6 3.4 0.015 S MAIL/TEXT
T 10.7 4.6 0.015 S E-MAIL/TEL/FAX
I 11.9 2.6 DOB-FLOW 0 1 1
```

with a graphic representation as follow :

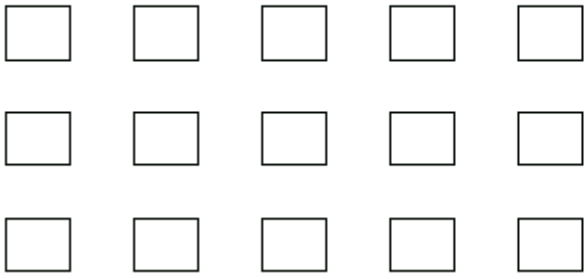
EXAMPLE C APPLICATION

As an example, you can see an application written in standard C which generates an 5x3 array of boxes of size 1.0x1.0 in a DRW file format.

```
/*

This example app, written for a standard C compiler, generates an array
of 5x3 boxes of size 1.0x1.0 each.
The information is written in ARRAY.DRW file to be loaded in the PalDRAW app.
The DRW file format is defined in the document F150797M available in our
www site.

  (C) Ariel Rocholl 1996,1997
  e-mail: aroch@geocities.com
  http:\\ourworld.compuserve.com\homepages\ariel_rocholl\palmtop.htm

This app was compiled using Microsoft Visual C++ 4.2.

*/

/* ------------------------------------------------------------------------ */

#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
```

```
#include <io.h>
#include <stdio.h>
#include <string.h>

/* ------------------------------------------------------------------------ */
/*Function which writes a string in the opened file as a text line*/

void WriteStringLine(int nFileHandle, const char* pString)
{
    _write(nFileHandle,pString,strlen(pString));
    _write(nFileHandle,"\n",1);
}

/* ------------------------------------------------------------------------ */
/* C main process*/

void main (void)
{
    /*Creates the output file in the current directory*/
    int nFileHandle=_open("ARRAY.DRW",_O_WRONLY|_O_CREAT|_O_TEXT,_S_IREAD|_S_IWRITE);
    int nHorizPoint, nVertPoint;

    /*Writes the header file*/
    WriteStringLine(nFileHandle, "#PalDRAW file - ARRAY example written in C");
    /*Don't write Status section as it is optional and not required in this example*/
    /*Don't write Blocks section as it is optional and not required in this example*/
    /*Writes the entities section name*/
    WriteStringLine(nFileHandle,"[Entities]");
    /*Write each box increasing horizontal and vertical origin*/
    for (nHorizPoint=0; nHorizPoint<10; nHorizPoint+=2)
    {
        for (nVertPoint=0; nVertPoint<6; nVertPoint+=2)
        {
            char pLine[100];
            /*Write 4 lines as a box*/
            sprintf(pLine,"L%d %d %d %d",nHorizPoint+0,nVertPoint+0,nHorizPoint+1,nVertPoint+0);
            WriteStringLine(nFileHandle,pLine);
            sprintf(pLine,"L %d %d %d %d",nHorizPoint+1,nVertPoint+0,nHorizPoint+1,nVertPoint+1);
            WriteStringLine(nFileHandle,pLine);
            sprintf(pLine,"L %d %d %d %d",nHorizPoint+1,nVertPoint+1,nHorizPoint+0,nVertPoint+1);
            WriteStringLine(nFileHandle,pLine);
            sprintf(pLine,"L %d %d %d %d",nHorizPoint+0,nVertPoint+1,nHorizPoint+0,nVertPoint+0);
            WriteStringLine(nFileHandle,pLine);
        }
    }

    /*Close the file*/
    _close(nFileHandle);
}

/* ------------------------------------------------------------------------ */
/* End of file                                                              */
/* ------------------------------------------------------------------------ */
```

Which represents a image as follows:



The Array.Exe application can be found in the same ZIP which contains this document.

```
***End of file.
```