pGraphicsD

COLLABORATORS			
	TITLE : pGraphicsD		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 29, 2025	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

pGraphicsD

1.1 pGraphicsD.doc

pGraphics.library

pOS_AddFont()
pOS_AddMonDevice()
pOS_AllocColorMapA()
pOS_AllocColorMapExA()
pOS_AllocGfxMapA()
pOS_AllocGfxMapExA()
pOS_AllocGfxObject()
pOS_AllocRastPortA()
pOS_AskSoftFontStyle()
pOS_BltGfxMap()
pOS_BltGfxMapClip()
pOS_BltGfxMapRastPort()
pOS_BltMask()
pOS_BltMaskRastPort()
pOS_CalcBorderBox()
pOS_CalcTextDim()
pOS_CloseFont()
pOS_CloseMonDevice()
pOS_CloseMonFile()
pOS_ConstructGfxFH()
pOS_ConstructGfxLk()
pOS_ConstructRastPort()
pOS_ConstructView()
pOS_ConstructViewPort()
pOS_CreateMonDevFromMount()
pOS_CreateMonMount()
pOS_DeleteMonMount()
pOS_DestructGfxFH()
pOS_DestructGfxLk()
pOS_DestructView()
pOS_DestructViewPort()
pOS_DisplayColorMap()
pOS_DisplayMonFile()
pOS_DisplayView()
pOS_DrawBorderBox()
pOS_DrawLine()
pOS_DrawRectFill()
pOS_DrawText()
pOS_DrawTextRect()
pOS_DrawTextWidth()
pOS_DupMonFile()
pOS_DupMonObjectLock()
pOS_EraseRect()
pOS_ExamineMonObject()
pOS_ExNextMonObject()
pOS_FindBestSoftFont()
pOS_FreeColorMap()
pOS_FreeColorMapExA()
pOS_FreeGfxMap()
pOS_FreeGfxMapExA()
pOS_FreeGfxObject()
pOS_FreeRastPort()
pOS_GetColor()
pOS_GetMonDevice()
pOS_GetMonDeviceName()
pOS_GetMonMountName()
pOS_GetNextMonDevice()
pOS_GetPixel()
pOS_GetPixelR()
pOS_GetRastPortA()
pOS_InitGfxLibrary()
pOS_InitMonDevice()
pOS_InitMonIOReq()
pOS_InitTCGfxMap()
pOS_LockMonObject()
pOS_MountMonDevice()
pOS_ObtainColorPen()
pOS_OpenFont()
pOS_OpenMonDevice()
pOS_OpenMonFile()
pOS_PolyFill()
pOS_ReleaseColorPen()

```
pOS_RemFont()                   pOS_RemMonDevice()
pOS_ScrollRaster()              pOS_SetABPenDrMd()
pOS_SetAPen()                   pOS_SetAPenR()
pOS_SetAreaFillPattern()        pOS_SetBPen()
pOS_SetBPenR()                  pOS_SetColor()
pOS_SetDrMd()                   pOS_SetFont()
pOS_SetGfxFHView()              pOS_SetLineFillPattern()
pOS_SetOPen()                   pOS_SetOPenR()
pOS_SetPixel()                  pOS_SetPixelR()
pOS_SetPosition()               pOS_SetRastPortA()
pOS_SetSoftFontStyle()          pOS_TextLength()
pOS_UnlockMonObject()           pOS_WaitBlit()
pOS_WritePixel()                pOS_WriteStdGfxMap()
pOS_WriteStdGfxMapRastPort()
```

1.2 pGraphics.library/pOS_InitGfxLibrary()

```
NAME
    pOS_InitGfxLibrary -- RTG-Library installieren

SYNOPSIS
     pOS_InitGfxLibrary(GfxLib*);
                        _R_A0

     VOID pOS_InitGfxLibrary(struct pOS_GfxLibrary*);

FUNCTION
     Jede RTG-Library muß sich beim Library-Init als RTG installieren.
     Alle nicht gesetzten Library-Funktionen werden durch System-Emulation
     ersetzt.

INPUTS
     GfxLib - neue RTG-Library

RESULT

SEE ALSO
```

1.3 pGraphics.library/pOS_CreateMonMount()

```
NAME
    pOS_CreateMonMount -- Monitorbeschreibung erzeugen

SYNOPSIS
     devMd = pOS_CreateMonMount(type);
      _R_D0                     _R_D0

     struct pOS_MonMountDevice*
                pOS_CreateMonMount(ULONG);
```

```
FUNCTION
     Struktur für die Monitorbeschreibung erzeugen.

INPUTS
     type - Arts des Monitors

RESULT
     Monitorbeschreibung oder NULL

SEE ALSO
```

1.4 pGraphics.library/pOS_DeleteMonMount()

```
NAME
    pOS_DeleteMonMount -- Monitorbeschreibung löschen

SYNOPSIS
     pOS_DeleteMonMount(devMd);
                        _R_A0

     VOID pOS_DeleteMonMount(struct pOS_MonMountDevice*);

FUNCTION
     Erzeugte Monitorbeschreibung von pOS_CreateMonMount()
     wieder löschen.

INPUTS
     devMd - Monitorbeschreibung
RESULT

SEE ALSO
```

1.5 pGraphics.library/pOS_AllocGfxObject()

```
NAME
    pOS_AllocGfxObject -- Gfx-Objekt erzeugen

SYNOPSIS
     obj = pOS_AllocGfxObject(type,reserved);
      _R_D0                   _R_D0  _R_D1

     __ARID__ APTR pOS_AllocGfxObject(ULONG,ULONG);

FUNCTION
     Gfx-Datenstruktur erzeugen.

INPUTS
     type     - (enum pOS_GfxObjects)
     reserved - immer 0
```

```
RESULT

SEE ALSO
```

1.6 pGraphics.library/pOS_FreeGfxObject()

```
NAME
    pOS_FreeGfxObject -- Gfx-Objekt löschen

SYNOPSIS
     pOS_FreeGfxObject(type,obj);
                       _R_D0  _R_A0

     VOID pOS_FreeGfxObject(ULONG,__ARID__ APTR);

FUNCTION
     Gfx-Datenstruktur von pOS_AllocGfxObject() löschen.

INPUTS
     type - (enum pOS_GfxObjects)
     obj  - zu löschendes Objekt

RESULT

SEE ALSO
```

1.7 pGraphics.library/pOS_WaitBlit()

```
NAME
    pOS_WaitBlit -- Synchronisation mit dem Blitter

SYNOPSIS
     pOS_WaitBlit(GfxLib);
                  _R_A0

     VOID pOS_WaitBlit(struct pOS_GfxLibrary*);

FUNCTION
     Setzt den aktuellen Task so lange in wait, bis der Blitter
     vom angegebenen RGB-Treiber 'GfxLib' seine Arbeitet vollendet hat.
     Läuft keine Blitter-Tätigkeit, kehr die Funktion sofort zurück.

INPUTS
     GfxLib - RTG-Treiber

RESULT

SEE ALSO
```

1.8 pGraphics.library/pOS_AllocGfxMapA()

```
NAME
    pOS_AllocGfxMapA -- GfxMap erzeugen

SYNOPSIS
     GfxMap = pOS_AllocGfxMapA(GfxLib,tagList);
      _R_D0                   _R_A0    _R_A1

     __ARID__ struct pOS_GfxMap*
                pOS_AllocGfxMapA(struct pOS_GfxLibrary*,
                                 const struct pOS_TagItem*);

FUNCTION

INPUTS
     GfxLib  - RTG-Treiber, NULL => Standard-Maps erzeugen
     tagList - (enum pOS_GfxTags)

TAG

     GFXTAG_Width (ULONG) - Pixel-Breite der Map

     GFXTAG_Height (ULONG) - Pixel-Höhe der Map

     GFXTAG_Colors (ULONG) - Anzahl der aufnehmbaren Farben

     GFXTAG_MasterRastPort (const pOS_RastPort*)
       Wird als Klone-Vorlage verwendet.

     GFXTAG_Error (ULONG*) (enum pOS_GfxErrors) aufgetretener Error

     GFXTAG_GfxFlags (enum pOS_GfxMapFlags)
        Flags:
        GFXMAPF_Draw    - in die GfxMap kann gezeichnet werden
        GFXMAPF_Display - die GfxMap kann angezeigt werden
        GFXMAPF_Clear   - GfxMap wird mit Nullen vorbelegt


     GFXTAG_GfxType (enum pOS_GfxMapType)
       RTG spezifische Typen:
        GFXMAPTYP_Planar  - Planar-Map
        GFXMAPTYP_BitMask - 1Bit - Map
        GFXMAPTYP_Chunky8 - 8Bit-Chunky

       Standard Typen: (Format liegt offen, kann direkt beschreiben werden)
        GFXMAPTYP_StdPlanar  - (struct pOS_StdPlanarGfxMap)
        GFXMAPTYP_StdChunky8 - (struct pOS_StdChunky8GfxMap)
        GFXMAPTYP_StdRGB8    - (struct pOS_StdRGB8GfxMap)
        GFXMAPTYP_StdMask    - (struct pOS_StdMaskGfxMap)
        GFXMAPTYP_StdRGBA8   - (struct pOS_StdRGBA8GfxMap)
        GFXMAPTYP_StdAlpha8  - (struct pOS_StdAlpha8GfxMap)


     GFXTAG_FriendGfxMap (const pOS_GfxMap*)
```

```
        Die neue GfxMap wird an die 'friend' angelehnt.

      GFXTAG_MonLock (pOS_MonLock*) - darstellende Gfx erzeugen

      GFXTAG_MonFile (pOS_MonFile*) - darstellende Gfx erzeugen

      GFXTAG_CrtGfxMapEx (BOOL) - (struct pOS_GfxMapEx*) erzeugen

      GFXTAG_Depth (ULONG) - nur bei Planar

      GFXTAG_CrtExtraMap (ULONG)
        Der RTG-Treiben erzeugt eine speicherarme Kopie einer Std-Gfx-Map
        als Friend-RTG-Map.
        Der Tag GFXTAG_StdGfxMap liefert die Vorlage.
        Der Tag GFXTAG_FriendGfxMap muß eine RTG-Friend-Map beinhalten.

        0 - nicht verwenden
        1 - speicherarme Kopie


RESULT
     neu erzeuge GfxMap

SEE ALSO
```

1.9 pGraphics.library/pOS_FreeGfxMap()

```
NAME
    pOS_FreeGfxMap -- GfxMap freigeben

SYNOPSIS
     pOS_FreeGfxMap(GfxMap);
                    _R_A0

     VOID pOS_FreeGfxMap(__ARID__ struct pOS_GfxMap*);

FUNCTION
     Gibt eine von pOS_AllocGfxMapA() erzeugte GfxMap wieder frei.

INPUTS
     GfxMap - freizugebende GfxMap
RESULT

SEE ALSO
```

1.10 pGraphics.library/pOS_AllocGfxMapExA()

```
NAME
    pOS_AllocGfxMapExA -- GfxMap-Extra erzeugen
```

```
SYNOPSIS
     GfxEx = pOS_AllocGfxMapExA(GfxMap,tagList);
     _R_D0                      _R_A0   _R_A1

     __ARID__ struct pOS_GfxMapEx*
                pOS_AllocGfxMapExA(const struct pOS_GfxMap*,
                                   const struct pOS_TagItem*);

FUNCTION

INPUTS
     GfxMap  - Friend-Map => ColorMap wird übertragen
     tagList - (enum pOS_GfxTags)
TAG

     GFXTAG_ColorMap (pOS_ColorMap*) - ColorMap für z.B. virt- TrueColor

RESULT

SEE ALSO
```

1.11 pGraphics.library/pOS_FreeGfxMapExA()

```
NAME
    pOS_FreeGfxMapExA -- GfxMap-Extra freigeben

SYNOPSIS
     pOS_FreeGfxMapExA(GfxEx);
                       _R_A0

     VOID pOS_FreeGfxMapExA(__ARID__ struct pOS_GfxMapEx*);

FUNCTION

INPUTS
     GfxMap - freizugebende Gfx-Extra Map
RESULT

SEE ALSO
```

1.12 pGraphics.library/pOS_InitTCGfxMap()

```
NAME
    pOS_InitTCGfxMap --  Internal

SYNOPSIS
     BOOL = pOS_InitTCGfxMap(GfxMap,ColorMap);
      _R_D0                  _R_A0     _R_A1

     BOOL pOS_InitTCGfxMap(struct pOS_GfxMap*,struct pOS_ColorMap*);
```

```
FUNCTION

INPUTS
     GfxMap   -
     ColorMap -

RESULT

SEE ALSO
```

1.13 pGraphics.library/pOS_WriteStdGfxMap()

```
NAME
    pOS_WriteStdGfxMap -- Standard-Map ausgeben

SYNOPSIS
     pOS_WriteStdGfxMap(src,   srcX, srcY, dest,destX,destY,
                        _R_A0 _R_D0 _R_D1 _R_A1 _R_D2 _R_D3
                        width,height,mode,patOffset);
                        _R_D4  _R_D5 _R_D6  _R_A3

     pOS_WriteStdGfxMap(const struct pOS_GfxMap*,SLONG,SLONG,
                        struct pOS_GfxMap*,SLONG,SLONG,ULONG,ULONG,
                        ULONG,const struct pOS_Point*);

FUNCTION
     StdMap wird ungeclipt nach 'dest' ausgegeben.

INPUTS
     src       - auszugebende StdMap
     srcX      - Lese-Offset in 'src'
     srcY      - Lese-Offset in 'src'
     dest      - GfxMap, in die gezeichnet wird
     destX     - Schreib-Offset in 'dest'
     destY     - Schreib-Offset in 'dest'
     width     - Pixel-Breite der zu schreibenden Grafik
     height    - Pixel-Height der zu schreibenden Grafik
     mode      - 0
     patOffset - Dither-Offset, NULL => (0,0)

RESULT

SEE ALSO
```

1.14 pGraphics.library/pOS_WriteStdGfxMapRastPort()

```
NAME
    pOS_WriteStdGfxMapRastPort -- Standard-Map ausgeben
```

```
SYNOPSIS
     pOS_WriteStdGfxMapRastPort(scr,   scrX, scrY, dest,destX,destY,
                                _R_A0 _R_D0 _R_D1 _R_A1 _R_D2 _R_D3
                                width,height,mode);
                                _R_D4  _R_D5 _R_D6

     pOS_WriteStdGfxMapRastPort(const struct pOS_GfxMap*,SLONG,SLONG,
                          struct pOS_RastPort*,SLONG,SLONG,ULONG,ULONG,
                          ULONG);

FUNCTION
     StdMap wird im RastPort geclipt nach 'dest' ausgegeben.

INPUTS
     src       - auszugebende StdMap
     srcX      - Lese-Offset in 'src'
     srcY      - Lese-Offset in 'src'
     dest      - RastPort, in die gezeichnet wird
     destX     - Schreib-Offset in 'dest'
     destY     - Schreib-Offset in 'dest'
     width     - Pixel-Breite der zu schreibenden Grafik
     height    - Pixel-Height der zu schreibenden Grafik
     mode      - 0

RESULT

SEE ALSO
```

1.15 pGraphics.library/pOS_AllocColorMapA()

```
NAME
    pOS_AllocColorMapA -- ColorMap erzeugen

SYNOPSIS
     ColorMap = pOS_AllocColorMapA(GfxLib,tagList);
     _R_D0                          _R_A1   _R_A1

     __ARID__ struct pOS_ColorMap*
                pOS_AllocColorMapA(struct pOS_GfxLibrary*,
                                   const struct pOS_TagItem*);

FUNCTION

INPUTS
     GfxLib  - RTG Treiber
     tagList - (enum pOS_GfxTags)

TAG

     GFXTAG_Colors (ULONG) - Anzahl der verwaltbaren Farben

     GFXTAG_Error (ULONG*) (enum pOS_GfxErrors) aufgetretener Fehler

     GFXTAG_MonLock (pOS_MonLock*) - darstellende Gfx erzeugen
```

```
     GFXTAG_MonFile (pOS_MonFile*) - darstellende Gfx erzeugen

     GFXTAG_ColorFlags (enum pOS_ColorMapFlags)
       COLMAPF_Display - die ColorMap kann dargestellt werden
       COLMAPF_Clear   - ColorMap wird mit Nullen vorbelegt

     GFXTAG_ColorType  (enum pOS_ColorMapType)
       RTG spezifische Typen:
        COLMAPTYP_RLut -

       Standard Typen: (Format liegt offen, kann direkt beschreiben werden)
        COLMAPTYP_StdRLut - (struct pOS_StdRLColorMap)

     GFXTAG_FriendColorMap (const pOS_ColorMap*)
       Die neue ColorMap wird an die 'friend' angelehnt.

     GFXTAG_CrtColorMapEx (BOOL) - struct pOS_ColorMapEx* erzeugen

     GFXTAG_RColorsArray (const struct pOS_RColor*) -
                 zu setzende Farben- Ist mit 0,0,0,0 zu beenden.

     GFXTAG_Depth (ULONG) - zur Berechnung der Farbanzahl


RESULT
     erzeugte ColorMap

SEE ALSO
```

1.16 pGraphics.library/pOS_FreeColorMap()

```
NAME
    pOS_FreeColorMap -- ColorMap freigeben

SYNOPSIS
     pOS_FreeColorMap(ColorMap);
                      _R_A1

     VOID pOS_FreeColorMap(__ARID__ struct pOS_ColorMap*);

FUNCTION

INPUTS
     ColorMap - freuzugebende ColorMap
RESULT

SEE ALSO
```

1.17 pGraphics.library/pOS_SetColor()

```
NAME
    pOS_SetColor -- Farbwert ändern

SYNOPSIS
     pOS_SetColor(ColorMap,entry,RColor,count);
                  _R_A0  _R_D0  _R_A2 _R_D1

     VOID pOS_SetColor(struct pOS_ColorMap*,ULONG,
                       const struct pOS_RColor*,ULONG);

FUNCTION

INPUTS
     ColorMap - zu ändernde Farbwerte-Tabelle
     entry    - zu ändernde Farb-Nummer (0,1,2,..)
     RColor   - Farbwerte
     count    - Anzahl der zu ändernden Farbwerte, RColor muß dann
                ein Array mit 'count' Elementen sein.

RESULT

SEE ALSO
```

1.18 pGraphics.library/pOS_GetColor()

```
NAME
    pOS_GetColor -- Farbwert ermitteln

SYNOPSIS
     pOS_GetColor(ColorMap,entry,RColor,count);
                  _R_A0  _R_D0  _R_A2 _R_D1

     VOID pOS_GetColor(const struct pOS_ColorMap*,ULONG,
                       struct pOS_RColor*,ULONG);

FUNCTION

INPUTS
     ColorMap - zu lesende Farbwerte-Tabelle
     entry    - zu lesende Farb-Nummer (0,1,2,..)
     RColor   - Farbwerte
     count    - Anzahl der zu lesenden Farbwerte, RColor muß dann
                ein Array mit 'count' Elementen sein.

RESULT

SEE ALSO
```

1.19 pGraphics.library/pOS_ObtainColorPen()

```
NAME
    pOS_ObtainColorPen -- Zeichenstift reservieren

SYNOPSIS
     pen = pOS_ObtainColorPen(ColorMap,entry,RColor,flags);
      _R_D0                   _R_A0  _R_D0  _R_A2 _R_D1

     ULONG pOS_ObtainColorPen(struct pOS_ColorMap*,ULONG,
                              const struct pOS_RColor*,ULONG);

FUNCTION
     Ermitteln eines Zeichenstiftes für z.B. pOS_SetAPen().
     Wird der Stift mit OTCOLPF_Shared angefragt, darf der
     Pen NICHT mittels pOS_SetColor() verändert werden.

INPUTS
     ColorMap - Farbwert-Tabelle
     entry    - Nummer des zu reservierenden Farbwert-Eintrags,
                ~0 => beliebigen Eintrag
     RColor   - gewünschte Farbwerte
     flags    - (enum pOS_ObtainColorPenFlags)

       OTCOLPF_Exclusive - der Pen wird nur einmal vergeben
       OTCOLPF_Shared    - der Pen wird von mehreren Usern verwendet
                           (darf nicht über pOS_SetColor() verändert werden)
       OTCOLPF_Flow      - nur in Verbindung mit OTCOLPF_Shared
                           Der Farbwert wird bei Bedarf minimal verändert.

       OTCOLPF_Force     - Konnte auf herkömmliche Weise kein Pen gefunden
                           werden, dann wird der nacheliegenste Pen verwendet.
                         => Beachte: Es ist nicht sicher, daß immer ein Pen
                           gefunden wird (=>OTCOLPF_Exclusive)

       OTCOLPF_Display   -  Änderung sichtbar machen => Display

RESULT
     Zeichenstift oder ~0 für kein Stift gefunden

SEE ALSO
```

1.20 pGraphics.library/pOS_ReleaseColorPen()

```
NAME
    pOS_ReleaseColorPen -- Zeichenstift freigeben

SYNOPSIS
     pOS_ReleaseColorPen(ColorMap,pen);
                         _R_A0  _R_D0

     VOID pOS_ReleaseColorPen(struct pOS_ColorMap*,ULONG);

FUNCTION
```

```
     Ein von pOS_ObtainColorPen() ermittelter Zeichenstift MUSS
     wieder freigegeben werden.

INPUTS
     ColorMap - Farbwert-Tabelle, die 'pen' beinhaltet
     pen      - Zeichenstift, ~0 wird ignoriert

RESULT

SEE ALSO
```

1.21 pGraphics.library/pOS_AllocColorMapExA()

```
NAME
    pOS_AllocColorMapExA -- Color-Extra erzeugen

SYNOPSIS
     ColorEx = pOS_AllocColorMapExA(ColorMap,tagList);
      _R_D0                         _R_A0    _R_A1

     __ARID__ struct pOS_ColorMapEx*
                pOS_AllocColorMapExA(const struct pOS_ColorMap*,
                                     const struct pOS_TagItem*);

FUNCTION

INPUTS
     ColorMap - Friend-Map => GFXTAG_Colors werden übertragen
     tagList  - (enum pOS_GfxTags)

TAG
     GFXTAG_Colors (ULONG) - Anzahl der zu verwaltenden Farben

RESULT

SEE ALSO
```

1.22 pGraphics.library/pOS_FreeColorMapExA()

```
NAME
    pOS_FreeColorMapExA -- Color-Extra freigeben

SYNOPSIS
     pOS_FreeColorMapExA(ColorEx);
                         _R_A0

     VOID pOS_FreeColorMapExA(__ARID__ struct pOS_ColorMapEx*);

FUNCTION
```

```
INPUTS
     ColorEx - freizugebende Color-Extra Map

RESULT

SEE ALSO
```

1.23 pGraphics.library/pOS_DisplayColorMap()

```
NAME
    pOS_DisplayColorMap -- Farbdaten sichtbar machen

SYNOPSIS
     pOS_DisplayColorMap(ColorMap,flags);
                         _R_A0    _R_D0

     VOID pOS_DisplayColorMap(struct pOS_ColorMap*,ULONG);

FUNCTION

INPUTS
     ColorMap - Farbdaten
     flags    - (enum pOS_DisplayColorMapFlags)

       DISCOLF_Always - immer anzeigen
       DISCOLF_Update - Die Color werden nur bei einer Veränderung
                        neu berechnet und dargestellt.

RESULT

SEE ALSO
```

1.24 pGraphics.library/pOS_ConstructRastPort()

```
NAME
    pOS_ConstructRastPort -- RastPort initialisieren

SYNOPSIS
     pOS_ConstructRastPort(RastPort);
                           _R_A0

     VOID pOS_ConstructRastPort(struct pOS_RastPort*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     RastPort - zu initialisierende Struktur

RESULT
```

```
SEE ALSO
```

1.25 pGraphics.library/pOS_SetRastPortA()

```
NAME
    pOS_SetRastPortA -- Daten im RastPort setzen

SYNOPSIS
     res = pOS_SetRastPortA(RastPort,tagList);
     _R_D0                  _R_A0    _R_A1

     ULONG pOS_SetRastPortA(struct pOS_RastPort*,const struct pOS_TagItem*);

FUNCTION

INPUTS
     RastPort - zu ändernden RastPort
     tagList  - (enum pOS_GfxTags)

TAG
     GFXTAG_GfxMap (pOS_GfxMap*) - GfxMap setzen (rp_GfxMap)

     GFXTAG_APen (ULONG) - entspricht pOS_SetAPen()

     GFXTAG_BPen (ULONG) - entspricht pOS_SetBPen()

     GFXTAG_OPen (ULONG) - entspricht pOS_SetOPen()

RESULT
     Anzahl der bearbeiteten Tags.

SEE ALSO
```

1.26 pGraphics.library/pOS_GetRastPortA()

```
NAME
    pOS_GetRastPortA -- Daten vom RastPort lesen

SYNOPSIS
     res = pOS_GetRastPortA(RastPort,tagList);
     _R_D0                  _R_A0    _R_A1

     ULONG pOS_GetRastPortA(struct pOS_RastPort*,const struct pOS_TagItem*);

FUNCTION

INPUTS
     RastPort - zu lesenden RastPort
     tagList  - (enum pOS_GfxTags)
```

```
TAG
     GFXTAG_GfxMap (pOS_GfxMap**) - GfxMap ermitteln (rp_GfxMap)

     GFXTAG_APen (ULONG*)

     GFXTAG_BPen (ULONG*)

     GFXTAG_OPen (ULONG*)

RESULT
     Anzahl der bearbeiteten Tags.

SEE ALSO
```

1.27 pGraphics.library/pOS_SetPosition()

```
NAME
    pOS_SetPosition -- Zeichenposition setzen

SYNOPSIS
     pOS_SetPosition(RastPort, x,   y);
                     _R_A0  _R_D0 _R_D1

     VOID pOS_SetPosition(struct pOS_RastPort*,SLONG x,SLONG y);

FUNCTION

INPUTS
     RastPort - Struktur
     x,y      - Zeichenposition im RastPort für:
                pOS_DrawLine(), pOS_DrawText(),

RESULT

SEE ALSO
```

1.28 pGraphics.library/pOS_SetDrMd()

```
NAME
     pOS_SetDrMd -- Zeichenmodus setzen

SYNOPSIS
     old = pOS_SetDrMd(RastPort,mode);
     _R_D0             _R_A0  _R_D0

     ULONG pOS_SetDrMd(struct pOS_RastPort*,ULONG);

FUNCTION
```

```
INPUTS
     RastPort - Struktur
     mode     - (enum pOS_RastPortDrawMode)

      DRMD_Jam1       - nur mit APen zeichnen
      DRMD_Jam2       - mit A- und BPen zeichnen
                        (hat nur Auswirkung auf Draw-Funktionen, die
                         beide Pens verwenden, sonst ist DRMD_Jam2 gleich
                         wie DRMD_Jam1)
      DRMD_Complement - XOr zeichnen

RESULT

SEE ALSO
```

1.29 pGraphics.library/pOS_DrawLine()

```
NAME
     pOS_DrawLine -- Line zeichnen

SYNOPSIS
     pOS_DrawLine(RastPort, x,    y);
                  _R_A0  _R_D0  _R_D1

     VOID pOS_DrawLine(struct pOS_RastPort*,SLONG,SLONG);

FUNCTION
    Line von der aktuellen Position 'pOS_SetPosition()' bis zur
    angegebenen Position zeichnen. Die aktuelle Position wird autom.
    auf die Endposition gesetzt.
    Mit 'pOS_SetLineFillPattern()' kann die Linienart definiert werden.
    Mit 'pOS_SetAPen()' wird die Vordergrundfarbe definiert.
    Wird die Line unterbrochen gezeichnet, so werden die Lücken
    bei DRMD_Jam2 mit dem BPen gezeichnet.
    Bei DRMD_Jam1 werden die Lücken transparent zum Hintergrund
    dargestellt. Wird ein Zeichenmodus DRMD_Complement gewählt,
    sind A- und BPen unbenutzt.

INPUTS
     RastPort - RastPort für die Ausgabe der Linie
     x,y      - Endposition der Linie.

RESULT
     aktuelle Position (rp_CP) in RastPort wird auf (x,y) gesetzt

SEE ALSO
```

1.30 pGraphics.library/pOS_DrawRectFill()

```
NAME
```

```
     pOS_DrawRectFill -- Gefüllte Fläche zeichnen

SYNOPSIS
     pOS_DrawRectFill(RastPort, x1,  y1,   x2,   y2);
                      _R_A0  _R_D0 _R_D1 _R_D2 _R_D3

     VOID pOS_DrawRectFill(struct pOS_RastPort*,
               SLONG,SLONG,SLONG,SLONG);

FUNCTION
     Zeichnet ein gefülltes Rechteck mit einer Startposition (x1,y1)
     und einer Endposition (x2,y2). Die Endposition MUSS größer als die
     Startposition sein.
     Mit pOS_SetAreaFillPattern() kann ein Muster definiert werden.
     Mit 'pOS_SetAPen()' wird die Vordergrundfarbe definiert.
     Wird das Rechteck mit Muster gezeichnet, so werden die Lücken
     bei DRMD_Jam2 mit dem BPen gezeichnet.
     Bei DRMD_Jam1 werden die Lücken transparent zum Hintergrund
     dargestellt. Wird ein Zeichenmodus DRMD_Complement gewählt,
     sind A- und BPen unbenutzt.

INPUTS
     RastPort - RastPort für die Ausgabe
     x1,y1    - linke obere Ecke
     x2,y2    - rechts untere Ecke

RESULT

SEE ALSO
```

1.31 pGraphics.library/pOS_SetAPen()

```
NAME
    pOS_SetAPen -- Vordergrund-Zeichenstift setzen

SYNOPSIS
     old = pOS_SetAPen(RastPort,pen);
     _R_D0             _R_A0  _R_D0

     ULONG pOS_SetAPen(struct pOS_RastPort*,ULONG);

FUNCTION
     Sämtliche Draw-Funktionen verwenden den APen als
     Vordergrund-Zeichenstift.

INPUTS
     RastPort - Struktur
     pen      - Vordergrund-Zeichenstift

RESULT
    zu letzt verwendeter Zeichenstift

SEE ALSO
```

1.32 pGraphics.library/pOS_SetBPen()

```
NAME
    pOS_SetBPen -- Hintergrund-Zeichenstift setzen

SYNOPSIS
     old = pOS_SetBPen(RastPort,pen);
     _R_D0             _R_A0  _R_D0

     ULONG pOS_SetBPen(struct pOS_RastPort*,ULONG);

FUNCTION

INPUTS
     RastPort - Struktur
     pen      - Hintergrund-Zeichenstift

RESULT
    zu letzt verwendeter Zeichenstift

SEE ALSO
```

1.33 pGraphics.library/pOS_SetOPen()

```
NAME
    pOS_SetOPen -- Outline-Zeichenstift setzen

SYNOPSIS
     old = pOS_SetOPen(RastPort,pen);
     _R_D0             _R_A0  _R_D0

     ULONG pOS_SetOPen(struct pOS_RastPort*,ULONG);

FUNCTION

INPUTS
     RastPort - Struktur
     pen      - Outline-Zeichenstift

RESULT
    zu letzt verwendeter Zeichenstift

SEE ALSO
```

1.34 pGraphics.library/pOS_SetAPenR()

```
NAME
    pOS_SetAPenR --  Vordergrund-Zeichenstift setzen
```

```
SYNOPSIS
     old = pOS_SetAPenR(RastPort,RColor);
     _R_D0              _R_A0     _R_A1

     ULONG pOS_SetAPenR(struct pOS_RastPort*,const struct pOS_RColor*);

FUNCTION
     Diese Funktion arbeitet nur auf virtuellen und echten TrueColor
     Bildschirmem.

INPUTS
     RastPort - Struktur
     pen      - geforderte Farbwerte

RESULT

SEE ALSO
```

1.35 pGraphics.library/pOS_SetBPenR()

```
NAME
    pOS_SetBPenR -- Hintergrund-Zeichenstift setzen

SYNOPSIS
     old = pOS_SetBPenR(RastPort,RColor);
     _R_D0              _R_A0     _R_A1

     ULONG pOS_SetBPenR(struct pOS_RastPort*,const struct pOS_RColor*);

FUNCTION
     Diese Funktion arbeitet nur auf virtuellen und echten TrueColor
     Bildschirmem.

INPUTS
     RastPort - Struktur
     pen      - geforderte Farbwerte

RESULT

SEE ALSO
```

1.36 pGraphics.library/pOS_SetOPenR()

```
NAME
    pOS_SetOPenR -- Outline-Zeichenstift setzen

SYNOPSIS
     old = pOS_SetOPenR(RastPort,RColor);
     _R_D0              _R_A0     _R_A1
```

```
     ULONG pOS_SetOPenR(struct pOS_RastPort*,const struct pOS_RColor*);

FUNCTION
     Diese Funktion arbeitet nur auf virtuellen und echten TrueColor
     Bildschirmem.

INPUTS
     RastPort - Struktur
     pen      - geforderte Farbwerte

RESULT

SEE ALSO
```

1.37 pGraphics.library/pOS_SetAreaFillPattern()

```
NAME
     pOS_SetAreaFillPattern -- Füllmuster setzen

SYNOPSIS
     BOOL pOS_SetAreaFillPattern(RastPort,pat,size);
     _R_D0                       _R_A0  _R_A1  _R_D0

     BOOL pOS_SetAreaFillPattern(struct pOS_RastPort*,const UWORD*,ULONG);

FUNCTION
     Beim Zeichen mit pOS_DrawRectFill() bzw. pOS_BltMaskRastPort()
     wird über den BPen das Muster gezeichnet.
     Das 'pat' Array muß so lange bestehen bleiben, bis der RastPort
     gelöscht wird oder ein anderes Muster gesetzt wird.

INPUTS
     RastPort - RastPort in dem das Muster gezeichnet werden soll
     pat      - Muster-Definitien, NULL => kein Muster
     size     - Musterhöhe in 2^size Pixeln,  size=log2(pixel)

RESULT
     TRUE - Muster wurde gesetzt.

SEE ALSO
```

1.38 pGraphics.library/pOS_SetLineFillPattern()

```
NAME
    pOS_SetLineFillPattern -- Linen-Muster setzen

SYNOPSIS
     BOOL pOS_SetLineFillPattern(RastPort, mask, shift);
     _R_D0                       _R_A0   _R_D0  _R_D1
```

```
     BOOL pOS_SetLineFillPattern(struct pOS_RastPort*,ULONG,ULONG);

FUNCTION

INPUTS
     RastPort - RastPort in dem das Muster gezeichnet werden soll
     mask     - Maske für die Line
     shift    - Verschiebungswert (15)

RESULT
     TRUE - Pattern wurde gesetzt

SEE ALSO
```

1.39 pGraphics.library/pOS_BltGfxMap()

```
NAME
    pOS_BltGfxMap -- RTG-Map blitten (kopieren)

SYNOPSIS
     pOS_BltGfxMap(src, srcX, srcY, dest,destX,destY,width,height,minterm);
                  _R_A0 _R_D0 _R_D1 _R_A1 _R_D2 _R_D3 _R_D4 _R_D5 _R_D6

     VOID pOS_BltGfxMap(const struct pOS_GfxMap*,SLONG,SLONG,
                struct pOS_GfxMap*,SLONG,SLONG,ULONG,ULONG,ULONG);

FUNCTION

INPUTS
     src     - zu kopierende Grafikdaten
     srcX    - horizontaler Lese-Offset in 'src'
     srcY    - vertikaler Lese-Offset in 'src'
     dest    - Map, in die gezeichnet wird
     destX   - horizontaler Schreib-Offset in 'dest'
     destY   - vertikaler Schreib-Offset in 'dest'
     width   - Pixel-Breite der zu schreibenden Grafik
     height  - Pixel-Height der zu schreibenden Grafik
     minterm - (enum pOS_BltStdMinterm)
       BLTMINT_Copy  - kopieren

RESULT

SEE ALSO
```

1.40 pGraphics.library/pOS_AllocRastPortA()

```
NAME
    pOS_AllocRastPortA -- RastPort erzeugen

SYNOPSIS
```

```
     RastPort = pOS_AllocRastPortA(GfxLib,tagList);
     _R_D0                         _R_A0   _R_A1

     __ARID__ struct pOS_RastPort*
               pOS_AllocRastPortA(struct pOS_GfxLibrary*,
                                  const struct pOS_TagItem*);

FUNCTION
     Erzeugen eines neuen RastPort.
     Alle neuen RastPorts bzw. Kopien vorhandener RastPorts müssen
     mit dieser Funktion erzeugt werden.

INPUTS
     GfxLib  - RTG Treiber
     tagList - (enum pOS_GfxTags)

TAG
     GFXTAG_MasterRastPort (const pOS_RastPort*) - als Klone-Vorlage
     GFXTAG_Error (ULONG*) (enum pOS_GfxErrors)

RESULT
     neu erzeugter RastPort

SEE ALSO
```

1.41 pGraphics.library/pOS_FreeRastPort()

```
NAME
    pOS_FreeRastPort -- RastPort freigeben

SYNOPSIS
     VOID pOS_FreeRastPort(RastPort);
                           _R_A0

     VOID pOS_FreeRastPort(__ARID__ struct pOS_RastPort*);

FUNCTION
     Freigeben eines mit pOS_AllocRastPort() erzeugten RastPorts.

INPUTS
     RastPort - freizugebenden rastport

RESULT

SEE ALSO
```

1.42 pGraphics.library/pOS_ScrollRaster()

```
NAME
    pOS_ScrollRaster -- Innerhalb einer Fläche scrollen
```

```
SYNOPSIS
     pOS_ScrollRaster(RastPort,dx,dy,x1,y1,x2,y2,mode);
                      _R_A0    _R_D0 - _R_D6

     VOID pOS_ScrollRaster(struct pOS_RastPort*,SLONG,SLONG,
                SLONG,SLONG,SLONG,SLONG,ULONG);

FUNCTION
     Verschiebt die durch x1,y2 x2,y2 definierte Fläche
     innerhalb eines RastPorts.
     Die durch die Verschiebung freiwerdene Fläche wird entsprechend
     dem Mode behandelt.

INPUTS
     RastPort - RastPort, dessen Inhalt verschoben werden soll
     dx,dy    - Scroll (Verschiebng)
     x1-y2    - Rectangle
     mode     - (enum pOS_ScrollRasterFlags)

       SCLRSTF_Std    - Standard => BPen ohne Damage
       SCLRSTF_Ignore - keine spezielle Aktion auslösen
       SCLRSTF_Damage - Alle freigewordenen Bereiche als Damage
                        kennzeichnen.
       SCLRSTF_BPen   - BPen verwenden
       SCLRSTF_Erase  - Die freigewordenen Bereiche werden mit dem
                        BackFill-Hook gefüllt. Schließt BPen-Draw aus.

RESULT

SEE ALSO
```

1.43 pGraphics.library/pOS_BltGfxMapRastPort()

```
NAME
    pOS_BltGfxMapRastPort -- RTG-Map blittern

SYNOPSIS
     pOS_BltGfxMapRastPort(src, srcX ,srcY,dest,destX, destY,
                          _R_A0 _R_D0 _R_D1 _R_A1 _R_D2 _R_D3
                           width,height,minterm);
                          _R_D4   _R_D5   _R_D6

     VOID pOS_BltGfxMapRastPort(const struct pOS_GfxMap*,SLONG,ULONG,
                struct pOS_RastPort*,SLONG,SLONG,ULONG,ULONG,ULONG);

FUNCTION
     Kopiert eine GfxMap bzw. einen Teil einer GfxMap in den
     angegebenden RastPort.

INPUTS
     src       - lese Map
     srcX      - horizontaler Lese-Offset in 'src'
     srcY      - vertikaler Lese-Offset in 'src'
```

```
     dest      - RastPort, in den gezeichnet wird
     destX     - horizontaler Schreib-Offset in 'dest'
     destY     - vertikaler Schreib-Offset in 'dest'
     width     - Pixel-Breite der zu schreibenden Grafik
     height    - Pixel-Height der zu schreibenden Grafik
     minterm   - (enum pOS_BltStdMinterm)

RESULT

SEE ALSO
```

1.44 pGraphics.library/pOS_BltGfxMapClip()

```
NAME
    pOS_BltGfxMapClip -- RTG-Map blittern

SYNOPSIS
     pOS_BltGfxMapClip(src, srcX ,srcY,dest,destX, destY,
                       _R_A0 _R_D0 _R_D1 _R_A1 _R_D2 _R_D3
                       width,height,minterm);
                       _R_D4   _R_D5   _R_D6

     VOID pOS_BltGfxMapClip(const struct pOS_RastPort*,SLONG,ULONG,
                 struct pOS_RastPort*,SLONG,SLONG,ULONG,ULONG,ULONG);

FUNCTION
     Kopiert einen Bereich aus einen RastPort in einen RastPort.

INPUTS
     src       - lese RastPort
     srcX      - horizontaler Lese-Offset in 'src'
     srcY      - vertikaler Lese-Offset in 'src'
     dest      - RastPort, in den gezeichnet wird
     destX     - horizontaler Schreib-Offset in 'dest'
     destY     - vertikaler Schreib-Offset in 'dest'
     width     - Pixel-Breite der zu schreibenden Grafik
     height    - Pixel-Height der zu schreibenden Grafik
     minterm   - (enum pOS_BltStdMinterm)

RESULT

SEE ALSO
```

1.45 pGraphics.library/pOS_SetABPenDrMd()

```
NAME
    pOS_SetABPenDrMd -- Zeichen-Daten setzen

SYNOPSIS
     pOS_SetABPenDrMd(RastPort,aPen,bPen,drMode);
```

```
                        _R_A0   _R_D0 _R_D1 _R_D2

     VOID pOS_SetABPenDrMd(struct pOS_RastPort*,ULONG,ULONG,ULONG);

FUNCTION

INPUTS
     RastPort - Struktur
     aPen     - Vordergrund-Zeichenstift 'pOS_SetAPen()'
     aPen     - Hintergrund-Zeichenstift 'pOS_SetAPen()'
     drMode   - Zeichenmodus (enum pOS_RastPortDrawMode)
                                        'pOS_SetDrMd()'

RESULT

SEE ALSO
```

1.46 pGraphics.library/pOS_PolyFill()

```
NAME
    pOS_PolyFill -- Gefülltes Vieleck zeichnen

SYNOPSIS
     pOS_PolyFill(RastPort,WPoint,cnt,mode);
                  _R_A0   _R_A1 _R_D0 _R_D1

     VOID pOS_PolyFill(struct pOS_RastPort*,const struct pOS_WPoint*,
                ULONG,ULONG);

FUNCTION
     Mit pOS_SetAreaFillPattern() kann ein Muster definiert werden.
     Mit 'pOS_SetAPen()' wird die Vordergrundfarbe definiert.
     Wird das Vieleck mit Muster gezeichnet, so werden die Lücken
     bei DRMD_Jam2 mit dem BPen gezeichnet.
     Bei DRMD_Jam1 werden die Lücken transparent zum Hintergrund
     dargestellt.

INPUTS
     RastPort - Ausgabe RastPort
     WPoint   - Array mit 'cnt' Elementen
     cnt      - Anzahl der Eckpunkte des Vielecks
     mode     - 0

RESULT

SEE ALSO
```

1.47 pGraphics.library/pOS_SetPixelR()

```
NAME
```

```
    pOS_SetPixelR -- INTERNAL

SYNOPSIS
     pOS_SetPixelR(RastPort,GfxMap, x,  y,   RColor,count);
                   _R_A0   _R_A1 _R_D0 _R_D1 _R_A2   _R_D2

     VOID pOS_SetPixelR(struct pOS_RastPort*,struct pOS_GfxMap*,
                SLONG,SLONG,const struct pOS_RColor*,ULONG);

FUNCTION

INPUTS

RESULT

SEE ALSO
```

1.48 pGraphics.library/pOS_SetPixel()

```
NAME
    pOS_SetPixel -- Pixel setzen

SYNOPSIS
     pOS_SetPixel(RastPort,GfxMap, x,   y,   pix);
                   _R_A0   _R_A1 _R_D0 _R_D1 _R_D2

     VOID pOS_SetPixel(struct pOS_RastPort*,struct pOS_GfxMap*,
                SLONG,SLONG,ULONG);

FUNCTION

INPUTS
     RastPort - Ausgabe RastPort oder NULL
     GfxMap   - Ausgabe GfxMap oder NULL
     x,y,     - Position
     pix      - Pixel-Wert

RESULT

SEE ALSO
```

1.49 pGraphics.library/pOS_GetPixelR()

```
NAME
    pOS_GetPixelR -- INTERNAL

SYNOPSIS
     pOS_GetPixelR(RastPort,GfxMap, x,   y,   RColor,cnt);
                   _R_A0   _R_A1 _R_D0 _R_D1 _R_A2  _R_D3
```

```
     pOS_GetPixelR(const struct pOS_RastPort*,const struct pOS_GfxMap*,
                SLONG,SLONG,struct pOS_RColor*,ULONG);

FUNCTION

INPUTS

RESULT

SEE ALSO
```

1.50 pGraphics.library/pOS_GetPixel()

```
NAME
    pOS_GetPixel -- Pixel lesen

SYNOPSIS
     pix = pOS_GetPixel(RastPort,GfxMap, x,   y);
     _R_D0              _R_A0   _R_A1 _R_D0 _R_D1

     ULONG pOS_GetPixel(const struct pOS_RastPort*,
                const struct pOS_GfxMap*,SLONG,SLONG);

FUNCTION

INPUTS
     RastPort - lese RastPort oder NULL
     GfxMap   - lese GfxMap oder NULL
     x,y      - Position

RESULT
     gelesener Pixel-Wert

SEE ALSO
```

1.51 pGraphics.library/pOS_WritePixel()

```
NAME
    pOS_WritePixel -- Pixel geclippt ausgeben.

SYNOPSIS
     pOS_WritePixel(RastPort,x,  y);
                    _R_A0  _R_D0 _R_D1

     VOID pOS_WritePixel(struct pOS_RastPort*,SLONG,SLONG);

FUNCTION
     Es wird ein Pixel mit dem Vordergrund-Zeichenstift (APen)
     gezeichnet.
```

```
INPUTS
     RastPort - Ausgabe RastPort
     x,y      - Position

RESULT

SEE ALSO
```

1.52 pGraphics.library/pOS_EraseRect()

```
NAME
    pOS_EraseRect -- Rechteck mit dem Hintergrund füllen

SYNOPSIS
     pOS_EraseRect(RastPort,x1,y1,x2,y2);
                   _R_A0  _R_D0 - _R_D3

     VOID pOS_EraseRect(struct pOS_RastPort*,SLONG,SLONG,SLONG,SLONG);

FUNCTION
     Das angegebene Rechteck wird mit der gelöscht. Ist kein Backfill
     Callback installiert, so wird mit dem BPen gelöscht. Bei
     installiertem Backfill wird dieser zum Löschen des Bereiches
     verwendet.

INPUTS
     RastPort - Ausgabe RastPort
     x1-y2    - Rechteck, das gezeichnet werden soll

RESULT

SEE ALSO
```

1.53 pGraphics.library/pOS_BltMask()

```
NAME
    pOS_BltMask -- Durch Maske zeichnen

SYNOPSIS
     pOS_BltMask(BltData,src,RastPort,dest,mask,minterm);
                 _R_A0  _R_A1 _R_A2  _R_A3 _R_A4 _R_D0

     VOID pOS_BltMask(const struct pOS_BltData*,const struct pOS_GfxMap*,
                struct pOS_RastPort*,struct pOS_GfxMap*,
                const struct pOS_GfxMap*,ULONG);

FUNCTION

INPUTS
     BltData  - Pixel-Daten
```

```
       btd_SrcX, btd_SrcY    - src Startoffset
       btd_DstX, btd_DstY    - dest Startoffset
       btd_MskX, btd_MskY    - mask Startoffset
       btd_Width,btd_Height  - Größe

     src      - Source-Map oder NULL, ist src==NULL, dann muß RastPort
                angegeben werden.
     RastPort - wird nur angegeben, fall 'src' null ist.
     dest     - Ziel-Map
     mask     - Blit-Maske oder NULL
     minterm  - (enum pOS_BltStdMinterm)

RESULT

SEE ALSO
```

1.54 pGraphics.library/pOS_BltMaskRastPort()

```
NAME
    pOS_BltMaskRastPort -- Durch Maske in RastPort zeichnen

SYNOPSIS
     pOS_BltMaskRastPort(BltData,src,RastPort,mask,minterm);
                         _R_A0  _R_A1 _R_A2  _R_A3  _R_D0

     VOID pOS_BltMaskRastPort(const struct pOS_BltData*,
                const struct pOS_GfxMap*,struct pOS_RastPort*,
                const struct pOS_GfxMap*,ULONG);

FUNCTION

INPUTS
     BltData  - Pixel-Daten
       btd_SrcX, btd_SrcY    - src Startoffset
       btd_DstX, btd_DstY    - RastPort Startoffset
       btd_MskX, btd_MskY    - mask Startoffset
       btd_Width,btd_Height  - Größe

     src      - Source-Map oder NULL
     RastPort - Ziel-RastPort
     mask     - Bilt-Maske oder NULL
     minterm  - (enum pOS_BltStdMinterm)

RESULT

SEE ALSO
```

1.55 pGraphics.library/pOS_ConstructView()

```
NAME
```

```
    pOS_ConstructView -- View initialisieren

SYNOPSIS
     pOS_ConstructView(View);
                       _R_A0

     VOID pOS_ConstructView(struct pOS_View*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     View - zu initialisierende Struktur

RESULT

SEE ALSO
```

1.56 pGraphics.library/pOS_DestructView()

```
NAME
    pOS_DestructView -- View de-installieren

SYNOPSIS
     pOS_DestructView(View);
                      _R_A0

     VOID pOS_DestructView(struct pOS_View*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     View - zu de-installierende Struktur

RESULT

SEE ALSO
```

1.57 pGraphics.library/pOS_ConstructViewPort()

```
NAME
    pOS_ConstructViewPort -- ViewPort initialisieren

SYNOPSIS
     pOS_ConstructViewPort(ViewPort);
                           _R_A0

     VOID pOS_ConstructViewPort(struct pOS_ViewPort*);
```

```
FUNCTION
     Nur für RTG-Treiber.

INPUTS
     ViewPort - zu initialisierende Struktur

RESULT

SEE ALSO
```

1.58 pGraphics.library/pOS_DestructViewPort()

```
NAME
    pOS_DestructViewPort -- ViewPort de-installieren

SYNOPSIS
     pOS_DestructViewPort(ViewPort);
                          _R_A0

     VOID pOS_DestructViewPort(struct pOS_ViewPort*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     ViewPort - zu de-installierende Struktur

RESULT

SEE ALSO
```

1.59 pGraphics.library/pOS_DisplayView()

```
NAME
    pOS_DisplayView -- Display neu aufbauen

SYNOPSIS
     pOS_DisplayView(View,ViewPort);
                     _R_A0   _R_A1

     VOID pOS_DisplayView(struct pOS_View*,struct pOS_ViewPort*);

FUNCTION

INPUTS
     View     - View für die Darstellung
     ViewPort - nur diesen ViewPort erzeuern, NULL => alle ViewPorts
                von 'View' erneuern.

RESULT
```

```
SEE ALSO
```

1.60 pGraphics.library/pOS_AddFont()

```
NAME
    pOS_AddFont -- Font in System-Liste einhängen

SYNOPSIS
     pOS_AddFont(TextFont);
                 _R_A0

     VOID pOS_AddFont(struct pOS_TextFont*);

FUNCTION

INPUTS
     TextFont - einzuhängenden verwendbaren Font

RESULT

SEE ALSO
```

1.61 pGraphics.library/pOS_RemFont()

```
NAME
    pOS_RemFont -- Font aus der System-Liste entfernen

SYNOPSIS
     pOS_RemFont(TextFont);
                 _R_A0

     VOID pOS_RemFont(struct pOS_TextFont*);

FUNCTION

INPUTS
     TextFont - zu entfernenden Font

RESULT

SEE ALSO
```

1.62 pGraphics.library/pOS_OpenFont()

```
NAME
    pOS_OpenFont -- Font über Attribute öffnen
```

```
SYNOPSIS
     TextFont = pOS_OpenFont(TextAttr);
      _R_D0                  _R_A0

     __ARID__ struct pOS_TextFont *pOS_OpenFont(const struct pOS_TextAttr*);

FUNCTION
     Anhand der Attribute wird der Font in der System-Liste gesucht.
     Wird kein passender Font gefunden, wird unter Umständen aus einem
     'Plain'-Font ein Bold, Italic, Unterline,... - Font berechnet.
     Konnte überhaupt kein Font entdeckt werden und das FNTFLGF_DiskFont-Flag
     ist nicht gesetzt, so wird der Font von 'FONTS:' nachgeladen.

INPUTS
     TextAttr - Font-Beschreibung
        ta_Name  - FontName z.B. "topaz.font"
        ta_YSize - Font-Höhe z.B. 11
        ta_Style - (enum pOS_FontStyles)
               FNTSTY_Normal     - normalen Text
               FNTSTYF_Underline - unterstrichen
               FNTSTYF_Bold      - fette Glyphen
               FNTSTYF_Italic    - nach rechts geneigt

        ta_Flags - (enum pOS_FontFlags)
               FNTFLGF_DiskFont     - NICHT setzen, sonst kann Font nicht
                                      nachgeladen werden.
               FNTFLGF_Proportional - Font darf proportional sein

        ta_Tags  - extended attributes (NULL)

RESULT
     geöffneten Font oder NULL

SEE ALSO
```

1.63 pGraphics.library/pOS_CloseFont()

```
NAME
    pOS_CloseFont -- Font nach der Bearbeitung schließen

SYNOPSIS
     pOS_CloseFont(TextFont);
                   _R_A0

     VOID pOS_CloseFont(__ARID__ struct pOS_TextFont*);

FUNCTION

INPUTS
     TextFont - von pOS_OpenFont() geöffnenten Font wieder schließen

RESULT
```

```
SEE ALSO
```

1.64 pGraphics.library/pOS_DrawText()

```
NAME
    pOS_DrawText -- Text ausgeben

SYNOPSIS
     pOS_DrawText(RastPort,text,length);
                  _R_A0   _R_A1   _R_D0

     VOID pOS_DrawText(struct pOS_RastPort*,const CHAR*,ULONG);

FUNCTION
     Bei DRMD_Jam1:
       Der Text wird im RastPort mit dem Zeichenstift (APen) ausgegeben.
       An den freien Textbereichen wird der Hintergrund nicht verändert.
     Bei DRMD_Jam2:
       Der Text wird im RastPort mit dem Zeichenstift (APen) ausgegeben.
       An den freien Textbereichen wird der Hintergrund mit
       (BPen) überschrieben..

INPUTS
     RastPort - Ausgabe RastPort
     text     - Text-String
     length   - Anzahl der Text-String Zeichen

RESULT
     Die aktuelle Position wird auf das Textende gesetzt (rp_CP).

SEE ALSO
```

1.65 pGraphics.library/pOS_SetFont()

```
NAME
    pOS_SetFont -- Font zur Ausgebe setzen

SYNOPSIS
     pOS_SetFont(RastPort,TextFont);
                 _R_A0   _R_A1

     VOID pOS_SetFont(struct pOS_RastPort*,struct pOS_TextFont*);

FUNCTION
     Jede Textausgabe und Textberechnung bezieht sich auf den RastPort-Font.
     Die Funktion setzt den Font im RastPort und berechnet einige Werte.
        rp_TxHeight   - Texthöhe (tf_YSize)
        rp_TxWidth    - Textbreite (tf_XSize)
        rp_TxBaseline - Abstand der Basisline von oben (tf_Baseline)
```

```
INPUTS
     RastPort - Struktur
     TextFont - zu setzenden Font, NULL setzt den System-Default-Font

RESULT

SEE ALSO
```

1.66 pGraphics.library/pOS_AskSoftFontStyle()

```
NAME
    pOS_AskSoftFontStyle -- Font-Styles ermitteln

SYNOPSIS
     flags = pOS_AskSoftFontStyle(RastPort);
      _R_D0                       _R_A0

     ULONG pOS_AskSoftFontStyle(struct pOS_RastPort*);

FUNCTION

INPUTS
     RastPort - Struktur

RESULT
     mögliche Font-Styles

SEE ALSO
```

1.67 pGraphics.library/pOS_SetSoftFontStyle()

```
NAME
    pOS_SetSoftFontStyle -- Font-Style setzen

SYNOPSIS
     style = pOS_SetSoftFontStyle(RastPort,style,mask);
     _R_D0                       _R_A0     _R_D0  _R_D1

     ULONG pOS_SetSoftFontStyle(struct pOS_RastPort*,ULONG,ULONG);

FUNCTION

INPUTS
     RastPort - Struktur
     style    - zu bearbeitende Styles
     mask     - Bit == set   => Style setzen
                Bit == clear => Style löschen
RESULT

SEE ALSO
```

1.68 pGraphics.library/pOS_FindBestSoftFont()

```
NAME
    pOS_FindBestSoftFont -- Ermitteln vom besten Font

SYNOPSIS
     TextFont = pOS_FindBestSoftFont(TextAttr);
     _R_D0                           _R_A0

     struct pOS_TextFont *pOS_FindBestSoftFont(const struct pOS_TextAttr*);

FUNCTION

INPUTS
     TextAttr - zu ermittelnder Font

RESULT
     gefundener Font

SEE ALSO
```

1.69 pGraphics.library/pOS_TextLength()

```
NAME
    pOS_TextLength -- Berechnet die Pixelbreite einer Text-Zeile

SYNOPSIS
     size = pOS_TextLength(RastPort,text,length);
     _R_D0                 _R_A0  _R_A1  _R_D0

     ULONG pOS_TextLength(const struct pOS_RastPort*,const CHAR*,ULONG);

FUNCTION

INPUTS
     RastPort - Ausgabe RastPort
     text     - Text-String
     length   - Anzahl der zu berechnenden Zeichen in 'text'

RESULT
     Pixel-Breite bei einer eventuellen Textausgabe über 'length'
     Zeichen.

SEE ALSO
```

1.70 pGraphics.library/pOS_DrawTextWidth()

```
NAME
    pOS_DrawTextWidth -- Textausgaben mit einer Breitenangabe
```

```
SYNOPSIS
     pOS_DrawTextWidth(RastPort,text,length,width);
                       _R_A0  _R_A1  _R_D0   _R_D1

     VOID pOS_DrawTextWidth(struct pOS_RastPort*,const CHAR*,ULONG,ULONG);

FUNCTION

INPUTS
     RastPort - Ausgabe RastPort
     text     - auszugebender Text-String
     length   - Anzahl der zu zeichnenden Zeichen
     width    - maximale Pixel-Breite. Die Textausgabe wird auf 'width'
                geclippt. 'width' = ~0 entspricht pOS_DrawText()

RESULT
     Die aktuelle Position wird auf das Textende gesetzt (rp_CP).

SEE ALSO
```

1.71 pGraphics.library/pOS_DrawTextRect()

```
NAME
    pOS_DrawTextRect -- Text geclipt ausgeben

SYNOPSIS
     pOS_DrawTextRect(RastPort,text,length,rectangle);
                       _R_A0  _R_A1  _R_D0   _R_A2

     VOID pOS_DrawTextRect(struct pOS_RastPort*,const CHAR*,
                ULONG,const struct pOS_Rectangle*);

FUNCTION

INPUTS
     RastPort  - Ausgabe RastPort
     text      - auszugebender Text-String
     length    - Anzahl der zu zeichnenden Zeichen
     rectangle - Der Text wird nur innerhalb dieser Box
                 ausgegeben.

RESULT
     Die aktuelle Position wird auf das Textende gesetzt (rp_CP).

SEE ALSO
```

1.72 pGraphics.library/pOS_CalcTextDim()

```
NAME
```

```
    pOS_CalcTextDim -- Text-Dimension berechnen

SYNOPSIS
     pOS_CalcTextDim(RastPort,text,length,TextDim,mode);
                     _R_A0  _R_A1  _R_D0  _R_A2  _R_D2

     VOID pOS_CalcTextDim(const struct pOS_RastPort*,const CHAR*,
               ULONG,struct pOS_TextDim*,ULONG);

FUNCTION

INPUTS
     RastPort - Struktur
     text     - zu berechnenden Text-String
     length   - Anzahl der zu untersuchenden Zeichen in 'text'

     TextDim  - Initialisierung und Ergebnis
       Muß vor dem Funktionsaufruf gesetzt werden:
         td_MaxWidth     - maximale Pixel-Width

       Wird von der Funktion gesetzt:
         td_UsedNumChars - Anzahl der vollständig sichtbaren Zeichen
         td_UsedWidth    -  Pixel-Width (von td_UsedNumChars Zeichen)

     mode     - 0

RESULT

SEE ALSO
```

1.73 pGraphics.library/pOS_InitMonIOReq()

```
NAME
    pOS_InitMonIOReq -- Monitor-IO vorbereiten

SYNOPSIS
     proc = pOS_InitMonIOReq(MonDevice,IOReq);
     _R_D0                   _R_A0       _R_A1

     struct pOS_Process *pOS_InitMonIOReq(const struct pOS_MonDevice*,
                                          struct pOS_MonIOReq*);

FUNCTION

INPUTS
     IOReq - zu initialisierende IO-Struktur

RESULT
     aktueller Process
SEE ALSO
```

1.74 pGraphics.library/pOS_InitMonDevice()

```
NAME
    pOS_InitMonDevice -- Monitor-Struktur initialisieren

SYNOPSIS
     pOS_InitMonDevice(MonDevice);
                       _R_A0

     VOID pOS_InitMonDevice(struct pOS_MonDevice*);

FUNCTION

INPUTS
     MonDevice - zu initialisierende Struktur

RESULT

SEE ALSO
```

1.75 pGraphics.library/pOS_AddMonDevice()

```
NAME
    pOS_AddMonDevice -- Monitor in die System-Liste einhängen

SYNOPSIS
     pOS_AddMonDevice(MonDevice);
                      _R_A0

     VOID pOS_AddMonDevice(struct pOS_MonDevice*);

FUNCTION

INPUTS
     MonDevice - einzuhämgender Monitor

RESULT

SEE ALSO
```

1.76 pGraphics.library/pOS_RemMonDevice()

```
NAME
    pOS_RemMonDevice -- versucht Monitor aus der Sys-Liste zu entfernen

SYNOPSIS
     pOS_RemMonDevice(MonDevice);
                      _R_A0
```

```
     VOID pOS_RemMonDevice(struct pOS_MonDevice*);

FUNCTION

INPUTS
     MonDevice - zu entfernenden Monitor

RESULT

SEE ALSO
```

1.77 pGraphics.library/pOS_OpenMonDevice()

```
NAME
    pOS_OpenMonDevice -- Monitor über Pfad öffnen

SYNOPSIS
     BOOL = pOS_OpenMonDevice(PathInfo);
     _R_D0                    _R_A0

     BOOL pOS_OpenMonDevice(struct pOS_MonDevPathInfo*);

FUNCTION

INPUTS
     PathInfo - Pfad

RESULT
     TRUE - Monitor konnte geöffnet werden

SEE ALSO
```

1.78 pGraphics.library/pOS_CloseMonDevice()

```
NAME
    pOS_CloseMonDevice -- Monitor über Path schließen

SYNOPSIS
     pOS_CloseMonDevice(PathInfo);
                        _R_A0

     VOID pOS_CloseMonDevice(struct pOS_MonDevPathInfo*);

FUNCTION

INPUTS
     PathInfo - Pfad

RESULT
```

```
SEE ALSO
```

1.79 pGraphics.library/pOS_GetNextMonDevice()

```
NAME
    pOS_GetNextMonDevice -- nächsten passenden Monitor ermitteln

SYNOPSIS
     BOOL = pOS_GetNextMonDevice(PathInfo);
     _R_D0                       _R_A0

     BOOL pOS_GetNextMonDevice(struct pOS_MonDevPathInfo*);

FUNCTION

INPUTS
     PathInfo - Pfad

RESULT
     nächster geöffneter Monitor oder NULL

SEE ALSO
```

1.80 pGraphics.library/pOS_GetMonDevice()

```
NAME
    pOS_GetMonDevice -- Monitor über Namen ermitteln

SYNOPSIS
     MonDevice = pOS_GetMonDevice(MonLock,path);
     _R_D0                        _R_A0    _R_A1

     struct pOS_MonDevice* pOS_GetMonDevice(
                const struct pOS_MonLock*,const dosname_t*);

FUNCTION

INPUTS
     MonLock - relativer Monitor-Pfad oder NULL
     path    - Pfad

RESULT
     gefundenen Monitor

SEE ALSO
```

1.81 pGraphics.library/pOS_GetMonDeviceName()

```
NAME
    pOS_GetMonDeviceName -- bestimmten Monitor über Namen ermitteln

SYNOPSIS
     MonDevice = pOS_GetMonDeviceName(name,type);
     _R_D0                          _R_A0  _R_D0

     struct pOS_MonDevice* pOS_GetMonDeviceName(const CHAR*,ULONG);

FUNCTION

INPUTS
     name - Monitor-name
     type - zu suchender Type (enum pOS_MonMountDevType)

RESULT
     gefundenen Monitor

SEE ALSO
```

1.82 pGraphics.library/pOS_GetMonMountName()

```
NAME
    pOS_GetMonMountName -- Monitor-Mount über Namen ermitteln

SYNOPSIS
     MonMount = pOS_GetMonMountName(name);
     _R_D0                          _R_A0

     struct pOS_MonMountDevice* pOS_GetMonMountName(const CHAR*);

FUNCTION

INPUTS
     name - Monitor-Namen

RESULT
     gefundene Monitor-Mount-Daten oder NULL

SEE ALSO
```

1.83 pGraphics.library/pOS_CreateMonDevFromMount()

```
NAME
    pOS_CreateMonDevFromMount -- Monitor über die Mount-Daten erzeugen

SYNOPSIS
     MonDevice = pOS_CreateMonDevFromMount(MonMount);
```

```
     _R_D0                                    _R_A0

     struct pOS_MonDevice* pOS_CreateMonDevFromMount(
                const struct pOS_MonMountDevice*);

FUNCTION

INPUTS
     MonMount - Beschreibung des zu erzeugenden Monitors

RESULT
     neu erzeugten Monitor oder NULL

SEE ALSO
```

1.84 pGraphics.library/pOS_LockMonObject()

```
NAME
    pOS_LockMonObject -- Monitor-Pfad locken

SYNOPSIS
     MonLock = pOS_LockMonObject(curr,path,mode);
     _R_D0                      _R_A0 _R_A1 _R_D0

     __ARID__ struct pOS_MonLock* pOS_LockMonObject(
                const struct pOS_MonLock*,const dosname_t*,ULONG);

FUNCTION

INPUTS
     curr - bereits bestehender Pfad für einen relativen Bezug mit 'path'
            NULL => wird ignoriert
     path - Monitor-Pfad
     mode - (enum pOS_MonLockAccess)

RESULT

SEE ALSO
```

1.85 pGraphics.library/pOS_UnlockMonObject()

```
NAME
    pOS_UnlockMonObject -- MonLock freigeben

SYNOPSIS
     pOS_UnlockMonObject(MonLock);
                         _R_A0

     VOID pOS_UnlockMonObject(__ARID__ struct pOS_MonLock*);
```

```
FUNCTION

INPUTS
     MonLock - freizugebender Lock

RESULT

SEE ALSO
```

1.86 pGraphics.library/pOS_DupMonObjectLock()

```
NAME
    pOS_DupMonObjectLock -- Lock duplizieren

SYNOPSIS
     MonLock = pOS_DupMonObjectLock(monLock);
     _R_D0                          _R_A0

     __ARID__ struct pOS_MonLock* pOS_DupMonObjectLock(
                const struct pOS_MonLock*);

FUNCTION

INPUTS
     monLock - zu duplizierender Lock

RESULT
    neuer Lock, der mitteln pOS_UnlockMonObject() freigegeben
    werden muß

SEE ALSO
```

1.87 pGraphics.library/pOS_ExamineMonObject()

```
NAME
    pOS_ExamineMonObject -- Daten über einen Lock ermitteln

SYNOPSIS
     BOOL = pOS_ExamineMonObject(MonLock,MonFile,MonInfoBlock);
     _R_D0                       _R_A0    _R_A1     _R_A2

     BOOL pOS_ExamineMonObject(const struct pOS_MonLock*,
                const struct pOS_MonFile*,struct pOS_MonInfoBlock*);

FUNCTION

INPUTS
     MonLock      - Daten zu diesem Lock ermitteln oder NULL
     MonFile      - Daten zu diesem File ermitteln oder NULL
     MonInfoBlock - Ergebnis
```

```
RESULT
     TRUE - Daten wurden ermittelt

SEE ALSO
```

1.88 pGraphics.library/pOS_ExNextMonObject()

```
NAME
    pOS_ExNextMonObject -- Iterative Ermittlung von  Daten

SYNOPSIS
     BOOL = pOS_ExNextMonObject(MonLock,MonInfoBlock);
     _R_D0                      _R_A0    _R_A1

     BOOL pOS_ExNextMonObject(const struct pOS_MonLock*,
                              struct pOS_MonInfoBlock*);

FUNCTION

INPUTS
     MonLock -  Struktur
     MonInfoBlock - Ergebnis

RESULT
     TRUE - Daten wurden ermittelt
     FALSE - keine weiteren Daten verfügbar

SEE ALSO
```

1.89 pGraphics.library/pOS_ConstructGfxLk()

```
NAME
    pOS_ConstructGfxLk -- Lock initialisieren

SYNOPSIS
     pOS_ConstructGfxLk(MonLock,MonDevice,mode);
                        _R_A0    _R_A1     _R_D0

     VOID pOS_ConstructGfxLk(struct pOS_MonLock*,
                struct pOS_MonDevice*,ULONG);

FUNCTION
     Für RTG Treiber

INPUTS
     MonLock   - zu initialisierender Lock
     MonDevice - 'Eigentümer'
     mode      - (enum pOS_MonLockAccess)
```

```
RESULT

SEE ALSO
```

1.90 pGraphics.library/pOS_DestructGfxLk()

```
NAME
    pOS_DestructGfxLk -- Lock de-installieren

SYNOPSIS
     pOS_DestructGfxLk(MonLock);
                       _R_A0

     VOID pOS_DestructGfxLk(struct pOS_MonLock*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     MonLock - zu de-installierende Struktur

RESULT

SEE ALSO
```

1.91 pGraphics.library/pOS_OpenMonFile()

```
NAME
    pOS_OpenMonFile -- Monitor-File öffnen

SYNOPSIS
     MonFile = pOS_OpenMonFile(curr,path,mode);
     _R_D0                     _R_A0  _R_A1 _R_D0

     __ARID__ struct pOS_MonFile* pOS_OpenMonFile(
                const struct pOS_MonLock*,const dosname_t*,ULONG);

FUNCTION

INPUTS
     curr - relativer Bezug für 'path' oder NULL
     path - zu öffnenden Monitor-Pfad
     mode - (enum pOS_MonFileMode)

RESULT
     geöffnetes Monitor-File

SEE ALSO
```

1.92 pGraphics.library/pOS_CloseMonFile()

```
NAME
    pOS_CloseMonFile -- Monitor-File schließen

SYNOPSIS
     pOS_CloseMonFile(MonFile);
                      _R_A0

     VOID pOS_CloseMonFile(__ARID__ struct pOS_MonFile*);

FUNCTION

INPUTS
     MonFile - zu schließendes Monitor-File

RESULT

SEE ALSO
```

1.93 pGraphics.library/pOS_ConstructGfxFH()

```
NAME
    pOS_ConstructGfxFH -- Monitor-File initialisieren

SYNOPSIS
     pOS_ConstructGfxFH(MonFile,MonDevice,mode);
                        _R_A0    _R_A1     _R_D0

     VOID pOS_ConstructGfxFH(struct pOS_MonFile*,
                struct pOS_MonDevice*,ULONG);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     MonFile   - zu initialisierende Struktur
     MonDevice - 'Eigentümer'
     mode      - (enum pOS_MonFileMode)

RESULT

SEE ALSO
```

1.94 pGraphics.library/pOS_DestructGfxFH()

```
NAME
    pOS_DestructGfxFH -- Monitor-File de-installieren
```

```
SYNOPSIS
     pOS_DestructGfxFH(MonFile);
                       _R_A0

     VOID pOS_DestructGfxFH(struct pOS_MonFile*);

FUNCTION
     Nur für RTG-Treiber.

INPUTS
     MonFile - zu de-installierende Struktur

RESULT

SEE ALSO
```

1.95 pGraphics.library/pOS_DupMonFile()

```
NAME
    pOS_DupMonFile -- Monitor-File duplizieren

SYNOPSIS
     MonFile = pOS_DupMonFile(monFile);
     _R_D0                    _R_A0

     struct pOS_MonFile* pOS_DupMonFile(const struct pOS_MonFile*);

FUNCTION

INPUTS
     monFile - zu duplizierendes Monitor-File

RESULT
    dupliziertes Monitor-File, muß mit pOS_CloseMonFile() geschlossen
    werden

SEE ALSO
```

1.96 pGraphics.library/pOS_SetGfxFHView()

```
NAME
    pOS_SetGfxFHView -- Verbindung von View-Strukturen herstellen

SYNOPSIS
     pOS_SetGfxFHView(MonFile,View,ViewPort);
                      _R_A0    _R_A1  _R_A2

     VOID pOS_SetGfxFHView(struct pOS_MonFile*,struct pOS_View*,
                struct pOS_ViewPort*);
```

```
FUNCTION

INPUTS
     MonFile  - fertiges Monitor-File
     View     - fertiger View
     ViewPort - fertiger ViewPort

RESULT

SEE ALSO
```

1.97 pGraphics.library/pOS_DisplayMonFile()

```
NAME
    pOS_DisplayMonFile -- Monitor-File anzeigen, update

SYNOPSIS
     BOOL = pOS_DisplayMonFile(File);
     _R_D0                     _R_A0

     BOOL pOS_DisplayMonFile(struct pOS_MonFile*);

FUNCTION

INPUTS
     File - anzuzeigendes File (Darstellung)

RESULT
     TRUE - wurde korrekt bearbeitet

SEE ALSO
```

1.98 pGraphics.library/pOS_DrawBorderBox()

```
NAME
    pOS_DrawBorderBox -- Standard-Border zeichnen

SYNOPSIS
     pOS_DrawBorderBox(RastPort,DrawInfo,type,rectangle);
                       _R_A0    _R_A1   _R_D0  _R_A2

     VOID pOS_DrawBorderBox(struct pOS_RastPort*,const struct pOS_DrawInfo*,
               ULONG,const struct pOS_Rectangle*);

FUNCTION

INPUTS
     RastPort  - Ausgabe RastPort
     DrawInfo  - Daten zur Ausgabe (wie Farbwerte, ...)
     type      - (enum pOS_GadgetBorderTypes) Siehe 'gadget.class'
```

```
     rectangle - Größe der Box

RESULT

SEE ALSO
```

1.99 pGraphics.library/pOS_CalcBorderBox()

```
NAME
    pOS_CalcBorderBox -- Berechnet die Randgröße eines Borders

SYNOPSIS
     pOS_CalcBorderBox(type,rectangle);
                       _R_D0  _R_A0

     VOID pOS_CalcBorderBox(ULONG,struct pOS_Rectangle*);

FUNCTION

INPUTS
     type      - (enum pOS_GadgetBorderTypes)
     rectangle - Ergebnis der Berechnung
RESULT

SEE ALSO
```

1.100 pGraphics.library/pOS_MountMonDevice()

```
NAME
    pOS_MountMonDevice -- Monitor-Mount erzeugen

SYNOPSIS
     err = pOS_MountMonDevice(devName,fileName,args);
     _R_D0                    _R_A0   _R_A1   _R_D0

     UWORD pOS_MountMonDevice(const CHAR*,const dosname_t*,const CHAR*);

FUNCTION

INPUTS
     devName  - zu mountender Monitor (Name)
     fileName - Mount-File
     args     - Argumente

RESULT
     Fehler-Code

SEE ALSO
```