

dopus/layout.h

COLLABORATORS

	<i>TITLE :</i> dopus/layout.h		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 1, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	dopus/layout.h	1
1.1	dopus/layout.h	1
1.2	dopus/layout.h	3

Chapter 1

dopus/layout.h

1.1 dopus/layout.h

dopus/layout.h
Structures

_GL_Object _ObjectList gpResize

Typedefs

ConfigWindow GL_Object ListViewDraw MenuData
NewConfigWindow ObjectDef ObjectList WindowData
WindowID

#defines

AREA ()	AREAFLAG_ERASE
AREAFLAG_ICON	AREAFLAG_LINE
AREAFLAG_NOFILL	AREAFLAG_OPTIM
AREAFLAG_RAISED	AREAFLAG_RECESSED
AREAFLAG_THIN	AREAFLAG_TITLE
BUTTONFLAG_CANCEL_BUTTON	BUTTONFLAG_OKAY_BUTTON
BUTTONFLAG_THIN_BORDERS	BUTTONFLAG_TOGGLE_SELECT
CFGDATA ()	DATA ()
DFB_DefPath	DIA_FrontPen
DIA_Type	DIR_BUTTON_KIND
DIR_GLASS_KIND	DLV_Check
DLV_DoubleClick	DLV_DragNotify
DLV_DrawLine	DLV_Flags
DLV_GetLine	DLV_Highlight
DLV_Labels	DLV_Lines
DLV_MakeVisible	DLV_MultiSelect
DLV_NoScroller	DLV_Object
DLV_ReadOnly	DLV_RightJustify
DLV_ScrollDown	DLV_ScrollUp
DLV_ScrollWidth	DLV_Selected
DLV_SelectNext	DLV_SelectPrevious
DLV_ShowChecks	DLV_ShowFileNames
DLV_ShowSelected	DLV_TextAttr

DLV_Top	DLV_TopJustify
DPG_Pen	DPG_SelectNext
DPG_SelectPrevious	DRAWINFO()
FIELD_KIND	FILE_BUTTON_KIND
FILE_GLASS_KIND	FILEBUTFLAG_SAVE
FONT_BUTTON_KIND	FPOS_TEXT_OFFSET
FRAME_KIND	GAD_ID_ICONIFY
GADGET()	GADGET_NUMBER()
GADGET_SEL()	GADGET_SPECIAL()
GADGET_STRING()	GADNUMBER()
GADSEL()	GADSPECIAL()
GADSTRING()	GM_RESIZE
GTCustom_Bold	GTCustom_Borderless
GTCustom_Callback	GTCustom_ChangeSigBit
GTCustom_ChangeSigTask	GTCustom_Control
GTCustom_CopyTags	GTCustom_FontPenCount
GTCustom_FontPens	GTCustom_FontPenTable
GTCustom_FrameFlags	GTCustom_History
GTCustom_Image	GTCustom_Integer
GTCustom_Justify	GTCustom_LayoutPos
GTCustom_LayoutRel	GTCustom_LocaleKey
GTCustom_LocaleLabels	GTCustom_MinMax
GTCustom_NoGhost	GTCustom_NoSelectNext
GTCustom_PathFilter	GTCustom_Secure
GTCustom_Style	GTCustom_TextAttr
GTCustom_TextPlacement	GTCustom_ThinBorders
HOTKEY_KIND	IM_ARROW_DOWN
IM_ARROW_UP	IM_BBOX
IM_BORDER_BOX	IM_CHECK
IM_CROSS	IM_DRAWER
IM_ICONIFY	IM_LOCK
IS_GADTOOLS()	JUSTIFY_CENTER
JUSTIFY_LEFT	JUSTIFY_RIGHT
LAYOUTF_BOTTOM_ALIGN	LAYOUTF_LEFT_ALIGN
LAYOUTF_RIGHT_ALIGN	LAYOUTF_SAME_HEIGHT
LAYOUTF_SAME_WIDTH	LAYOUTF_TOP_ALIGN
LISTVIEWFLAG_CURSOR_KEYS	lve_Flags
lve_Pen	LVEF_SELECTED
LVEF_TEMP	LVEF_USE_PEN
MENUFLAG_AUTO_MUTEX	MENUFLAG_COMM_SEQ
MENUFLAG_GET_SEQ()	MENUFLAG_MAKE_SEQ()
MENUFLAG_TEXT_STRING	MENUFLAG_USE_SEQ
MENUID()	NM_BAR_LABEL
NM_NEXT	OBJECTF_HOTKEY
OBJECTF_INTEGER	OBJECTF_NO_SELECT_NEXT
OBJECTF_PATH_FILTER	OBJECTF_READ_ONLY
OBJECTF_SECURE	OBJECTFLAG_DRAWN
OBJLIST()	OD_AREA
OD_END	OD_GADGET
OD_IMAGE	OD_SKIP
OD_TEXT	OPEN_SHRUNK
OPEN_SHRUNK_HORIZ	OPEN_SHRUNK_VERT
OPEN_USED_DEFAULT	OPEN_USED_TOPAZ
OPUS_LISTVIEW_KIND	POS_CENTER
POS_MOUSE_CENTER	POS_MOUSE_REL
POS_PROPORTION	POS_REL_RIGHT
POS_RIGHT_JUSTIFY	POS_SQUARE

```

POSFLAG_ADJUST_POS_X      POSFLAG_ADJUST_POS_Y
POSFLAG_ALIGN_POS_X      POSFLAG_ALIGN_POS_Y
RECTHEIGHT()             RECTWIDTH()
SET_WINDOW_ID()          SIZE_MAX_LESS
SIZE_MAXIMUM              TEXTFLAG_ADJUST_TEXT
TEXTFLAG_CENTER          TEXTFLAG_NO_USCORE
TEXTFLAG_RIGHT_JUSTIFY   TEXTFLAG_TEXT_STRING
VISINFO()                WINDOW_AUTO_KEYS
WINDOW_AUTO_REFRESH      WINDOW_BACKDROP
WINDOW_BUTTONS           WINDOW_FUNCTION
WINDOW_GROUP             WINDOW_ICONIFY
WINDOW_LAYOUT_ADJUST     WINDOW_LISTER
WINDOW_LISTER_ICONS      WINDOW_MAGIC
WINDOW_NO_ACTIVATE       WINDOW_NO_BORDER
WINDOW_NO_CLOSE          WINDOW_OBJECT_PARENT
WINDOW_POPUP_MENU        WINDOW_REQ_FILL
WINDOW_SCREEN_PARENT     WINDOW_SIMPLE
WINDOW_SIZE_BOTTOM       WINDOW_SIZE_RIGHT
WINDOW_START             WINDOW_TEXT_VIEWER
WINDOW_UNDEFINED         WINDOW_UNKNOWN
WINDOW_USER              WINDOW_VISITOR
WINFLAG()               WINMEMORY()
WINREQUESTER()

```

1.2 dopus/layout.h

```

                #ifndef _DOPUS_LAYOUT
#define _DOPUS_LAYOUT

/*****

Layout routines

*****/

#ifndef LIBRARIES_GADTOOLS_H
#include <libraries/gadtools.h>
#endif

#define POS_CENTER          -1           // Center position
#define POS_RIGHT_JUSTIFY   -2           // Right-justified

#define POS_MOUSE_CENTER    -3           // Center over mouse
#define POS_MOUSE_REL       -4           // Relative to mouse

#define POS_PROPORTION      1024         // Proportion of space left
#define POS_SQUARE          1124
#define POS_REL_RIGHT       (1<<14)     // Relative to another

#define FPOS_TEXT_OFFSET    16384

#define SIZE_MAXIMUM        -1
#define SIZE_MAX_LESS      -101

```

```

// Defines a window
typedef struct
    struct IBox cw_CharDims;
    struct IBox cw_FineDims;
} ConfigWindow;

// Opens a window
typedef struct
    void                                *nw_Parent;                // Parent to open on

    ConfigWindow
        *nw_Dims;                // Window dimensions

    char                                *nw_Title;                // Window title
    struct DOpusLocale                *nw_Locale;                // Locale to use
    struct MsgPort                    *nw_Port;                // Message port to use
    ULONG                                nw_Flags;                // Flags
    struct TextFont                    *nw_Font;                // Alternative font to use
} NewConfigWindow;

// Set by the user
#define WINDOW_SCREEN_PARENT        (1<<0)                // Parent is a screen
#define WINDOW_NO_CLOSE            (1<<1)                // No close gadget
#define WINDOW_NO_BORDER          (1<<2)                // No border
#define WINDOW_LAYOUT_ADJUST      (1<<3)                // Adjust window size to fit ↔
    objects
#define WINDOW_SIMPLE              (1<<4)                // Simple refresh
#define WINDOW_AUTO_REFRESH      (1<<5)                // Refresh window automatically
#define WINDOW_AUTO_KEYS          (1<<6)                // Handle keys automatically
#define WINDOW_OBJECT_PARENT      (1<<7)                // Parent is an existing object
#define WINDOW_REQ_FILL           (1<<8)                // Backfill as a requester
#define WINDOW_NO_ACTIVATE       (1<<9)                // Don't activate
#define WINDOW_VISITOR            (1<<10)               // Open as visitor window
#define WINDOW_SIZE_RIGHT         (1<<11)               // Size gadget, in right border
#define WINDOW_SIZE_BOTTOM       (1<<12)               // Size gadget, in bottom border
#define WINDOW_ICONIFY            (1<<13)               // Iconify gadget

// Set by the system
#define OPEN_USED_DEFAULT          (1<<16)               // To open had to use default font
#define OPEN_USED_TOPAZ           (1<<17)               // To open had to use topaz
#define OPEN_SHRUNK_VERT          (1<<18)               // Window is not full vertical ↔
    size requested
#define OPEN_SHRUNK_HORIZ        (1<<19)               // Window is not full horizontal ↔
    size requested
#define OPEN_SHRUNK              (
        OPEN_SHRUNK_VERT
        |
        OPEN_SHRUNK_HORIZ
    )

// ID of the iconify gadget
#define GAD_ID_ICONIFY            0xffa0

// Defines an object
typedef struct
    BYTE                                od_Type;
    UBYTE                               od_ObjectKind;

```

```

        struct IBox      od_CharDims;
        struct IBox      od_FineDims;
        ULONG            od_GadgetText;
        ULONG            od_Flags;
        USHORT           od_ID;
        struct TagItem *od_TagList;
} ObjectDef;

#define TEXTFLAG_TEXT_STRING      (1<<17)           // Text is a string, not a ↵
        Locale ID
#define TEXTFLAG_NO_USCORE        (1<<18)           // No underscore in text
#define BUTTONFLAG_OKAY_BUTTON    (1<<18)           // Button is an "ok" ↵
        button
#define BUTTONFLAG_CANCEL_BUTTON  (1<<19)           // Button is a "cancel" ↵
        button
#define BUTTONFLAG_TOGGLE_SELECT  (1<<20)           // Button is toggle-select
#define LISTVIEWFLAG_CURSOR_KEYS  (1<<21)           // Lister responds to ↵
        cursor
#define BUTTONFLAG_THIN_BORDERS   (1<<22)           // Button has thin borders
#define FILEBUTFLAG_SAVE          (1<<21)           // Save mode

#define TEXTFLAG_ADJUST_TEXT      (1<<23)           // Adjust for text
#define POSFLAG_ADJUST_POS_X      (1<<24)           // Position adjustor
#define POSFLAG_ADJUST_POS_Y      (1<<25)           // Position adjustor
#define POSFLAG_ALIGN_POS_X       (1<<26)           // Align
#define POSFLAG_ALIGN_POS_Y       (1<<27)           // Align

#define TEXTFLAG_RIGHT_JUSTIFY    (1<<1)           // Right-justify text
#define TEXTFLAG_CENTER           (1<<2)           // Center text

#define AREAFLAG_RAISED           (1<<8)           // Raised rectangle
#define AREAFLAG_RECESSED         (1<<9)           // Recessed rectangle
#define AREAFLAG_THIN             (1<<10)          // Thin borders
#define AREAFLAG_ICON             (1<<11)          // Icon drop box
#define AREAFLAG_ERASE            (1<<12)          // Erase interior
#define AREAFLAG_LINE             (1<<13)          // Line (separator)
#define AREAFLAG_OPTIM            (1<<15)          // Optimised refreshing
#define AREAFLAG_TITLE            (1<<16)          // Box with a title
#define AREAFLAG_NOFILL           (1<<18)          // No fill

#define OBJECTFLAG_DRAWN          (1<<31)          // Object has been drawn

#define OD_END                     0               // End of a list
#define OD_GADGET                   1               // A gadget
#define OD_TEXT                     2               // Some text
#define OD_AREA                     3               // A rectangular area
#define OD_IMAGE                    4               // An image
#define OD_SKIP                     -1              // Skip this entry

typedef struct _GL_Object
{
        struct _GL_Object      *next;           // Next object
        short                  type;           // Type of object
        char                    key;           // Key equivalent

        unsigned char          flags2;        // Additional flags

```



```

unsigned short      id;           // Object ID
unsigned short      control_id;   // Object that this controls
struct IBox         dims;        // Object dimensions
ULONG              flags;        // Object flags
char                *text;       // Text
USHORT             object_kind;   // Object kind

union
{
    struct _gl_gadget
    {
        struct Gadget *context; // Context data for the gadget
        struct Gadget *gadget;  // The gadget itself
        int components;         // Number of component gadgets
        LONG data;              // Some data for the gadget
        short choice_max;       // Number of choices
        short choice_min;       // Minimum choice
        struct Image *image;    // Gadget image
    } gl_gadget;

    struct _gl_text
    {
        struct IBox text_pos;    // Text position
        WORD base_pos;          // Baseline position
        short uscore_pos;       // Underscore position
    } gl_text;

    struct _gl_area
    {
        struct IBox text_pos;    // Text position within area
        struct IBox area_pos;    // Area position
        int frametype;          // Frame type
    } gl_area;

    struct _gl_image
    {
        struct IBox image_pos;   // Image position
        struct Image *image;     // Image
    } gl_image;
} gl_info;

APTR memory; // Any other memory

char *original_text; // Original text string
char fg,bg; // Current pen colours

ULONG data_ptr; // Pointer to other data

struct TagItem *tags; // Copy of tags

struct IBox char_dims; // Original dimensions
struct IBox fine_dims;

} GL_Object;

typedef struct _ObjectList {

    GL_Object
        *firstobject; // First object

```

```

        struct TextAttr      attr;           // Font used
        struct Window        *window;       // Window used
        struct _ObjectList   *next_list;    // Next list
} ObjectList;

#define OBJECTF_NO_SELECT_NEXT (1<<0)      // Don't select next field
#define OBJECTF_PATH_FILTER    (1<<1)      // Filter path characters
#define OBJECTF_SECURE         (1<<2)      // Hide string
#define OBJECTF_INTEGER        (1<<3)      // Integer gadget
#define OBJECTF_READ_ONLY      (1<<4)      // Read-only
#define OBJECTF_HOTKEY         (1<<5)      // Hotkey string

typedef struct
    UBYTE   md_Type;           // Menu type
    ULONG   md_ID;            // Menu ID
    ULONG   md_Name;          // Menu name
    ULONG   md_Flags;         // Menu flags
} MenuData;

#define MENUFLAG_TEXT_STRING    (1<<16)    // Menu name is a real string
#define MENUFLAG_COMM_SEQ      (1<<17)    // Give menu a command sequence
#define MENUFLAG_AUTO_MUTEX    (1<<18)    // Automatic mutual exclusion
#define MENUFLAG_USE_SEQ       (1<<19)    // Use command sequence supplied

#define MENUFLAG_MAKE_SEQ(c)    ((ULONG)(c)<<24)
#define MENUFLAG_GET_SEQ(fl)   ((char)((fl)>>24))

#define NM_NEXT                  10
#define NM_BAR_LABEL             (ULONG)NM_BARLABEL

#define IS_GADTOOLS(obj)        (BOOL)(obj->gl_info.gl_gadget.context)

typedef struct
    ULONG   magic;           // Magic ID
    struct Window *window;   // Pointer back to window
    ULONG   window_id;      // User window ID
    struct MsgPort *app_port; // "Window's" application port
} WindowID;

#define WINDOW_MAGIC             0x83224948
#define WINDOW_UNKNOWN          (ULONG)-1
#define WINDOW_UNDEFINED        0

#define SET_WINDOW_ID(w, id)    (((
    WindowID
    *) ((w)->UserData))->window_id=(id))

// Window types
#define WINDOW_BACKDROP          0x4000001
#define WINDOW_LISTER           0x4000002
#define WINDOW_BUTTONS          0x4000004
#define WINDOW_GROUP            0x4000008
#define WINDOW_LISTER_ICONS     0x4000010
#define WINDOW_FUNCTION          0x4000020 // not really a window
#define WINDOW_START            0x4000040

#define WINDOW_POPUP_MENU       0x0001200

```

```

#define WINDOW_TEXT_VIEWER                0x0001300

#define WINDOW_USER                        0x2000000

// This structure is pointed to by Window->UserData
typedef struct

    WindowID                                id;                // Window ID information

    ObjectList
    *list;                                  // Window object list
    struct FileRequester *request;          // Window's file requester
    APTR visinfo;                           // Visual info
    struct DrawInfo *drawinfo;             // Draw info
    struct DOpusLocale *locale;           // Locale info
    struct MsgPort *window_port;          // Window message port (if ←
        supplied)
    struct NewMenu *new_menu;              // NewMenu structure allocated
    struct Menu *menu_strip;               // Menu strip allocated
    struct Requester *busy_req;           // Window busy requester
    ULONG data;                             // Window-specific data
    ULONG flags;                             // Flags
    APTR memory;                             // User memory pool, freed when ←
        window closes

    APTR hook_magic;                        // Magic for backfill hooks
    struct FontRequester *font_request;    // Window's font requester

    ULONG userdata;
    struct TagItem *user_tags;

    struct List boopsi_list;               // BOOPSI list
} WindowData;

#define FILE_GLASS_KIND 1000
#define DIR_GLASS_KIND 1001

#define GM_RESIZE (20)

struct gpResize
{
    ULONG MethodID;
    struct GadgetInfo *gpr_GInfo;
    struct RastPort *gpr_RPort;
    struct IBox gpr_Size;
    long gpr_Redraw;
    struct Window *gpr_Window;
    struct Requester *gpr_Requester;
};

// Custom tags
#define GTCustom_LocaleLabels TAG_USER + 0 // Points to list of Locale IDs
#define GTCustom_Image TAG_USER + 1 // Image for gadget
#define GTCustom_CallBack TAG_USER + 2 // Tag ID and data filled in by ←
    callback

```

```

#define GTCustom_LayoutRel      TAG_USER + 3    // Layout relative to this object ←
    ID
#define GTCustom_Control       TAG_USER + 4    // Controls another gadget
#define GTCustom_TextAttr     TAG_USER + 6    // TextAttr to use
#define GTCustom_MinMax       TAG_USER + 24   // Minimum and maximum bounds
#define GTCustom_ThinBorders  TAG_USER + 27   // Gadget has thin borders
#define GTCustom_LocaleKey    TAG_USER + 29   // Key from locale string
#define GTCustom_NoSelectNext TAG_USER + 31   // Don't select next field
#define GTCustom_PathFilter   TAG_USER + 32   // Filter path characters
#define GTCustom_History      TAG_USER + 33   // History
#define GTCustom_CopyTags     TAG_USER + 34   // Copy tags
#define GTCustom_FontPens     TAG_USER + 35   // Place to store pens and style
#define GTCustom_FontPenCount TAG_USER + 36   // Number of pens for font ←
    requester
#define GTCustom_FontPenTable TAG_USER + 37   // Table of pens for font ←
    requester
#define GTCustom_Bold         TAG_USER + 38   // Bold pen
#define GTCustom_Secure       TAG_USER + 39   // Secure string field
#define GTCustom_Integer      TAG_USER + 40   // Integer gadget
#define GTCustom_TextPlacement TAG_USER + 41  // Position of text
#define GTCustom_NoGhost      TAG_USER + 42   // Disable without ghosting
#define GTCustom_Style        TAG_USER + 44   // Pen styles
#define GTCustom_FrameFlags   TAG_USER + 45   // Frame flags
#define GTCustom_ChangeSigTask TAG_USER + 46  // Task to signal on change
#define GTCustom_ChangeSigBit TAG_USER + 47   // Signal bit to use
#define GTCustom_LayoutPos    TAG_USER + 49   // Use with the POSFLAGS
#define GTCustom_Borderless   TAG_USER + 50   // Borderless
#define GTCustom_Justify      TAG_USER + 51   // Justification

#define LAYOUTF_SAME_HEIGHT    (1<<0)
#define LAYOUTF_SAME_WIDTH    (1<<1)
#define LAYOUTF_TOP_ALIGN     (1<<2)
#define LAYOUTF_BOTTOM_ALIGN  (1<<3)
#define LAYOUTF_LEFT_ALIGN    (1<<4)
#define LAYOUTF_RIGHT_ALIGN   (1<<5)

#define JUSTIFY_LEFT          0
#define JUSTIFY_RIGHT         1
#define JUSTIFY_CENTER        2

#define DIA_Type              TAG_USER + 5    // Image type
#define DIA_FrontPen          TAG_USER + 7    // Image front pen

#define IM_ARROW_UP           0
#define IM_ARROW_DOWN         1
#define IM_CHECK               2
#define IM_DRAWER              3
#define IM_BORDER_BOX         4
#define IM_BBOX                 5
#define IM_ICONIFY             6
#define IM_CROSS               7
#define IM_LOCK                 8

#define OPUS_LISTVIEW_KIND    127            // Custom listview gadget
#define FILE_BUTTON_KIND      126            // File button gadget
#define DIR_BUTTON_KIND       125            // Directory button gadget
#define FONT_BUTTON_KIND      124            // Font button gadget

```

```

#define FIELD_KIND          123          // Text field (no editing)
#define FRAME_KIND         122          // Frame
#define HOTKEY_KIND        121          // Hotkey field

// Listview tags
#define DLV_TextAttr        TAG_USER + 6  // TextAttr to use
#define DLV_ScrollUp        TAG_USER + 7  // Scroll list up
#define DLV_ScrollDown      TAG_USER + 8  // Scroll list down
#define DLV_SelectPrevious  TAG_USER + 11 // Select previous item
#define DLV_SelectNext      TAG_USER + 12 // Select next item
#define DLV_Labels          GTLV_Labels   // Labels
#define DLV_Top             GTLV_Top      // Top item
#define DLV_MakeVisible     GTLV_MakeVisible // Make visible
#define DLV_Selected        GTLV_Selected // Selected
#define DLV_ScrollWidth     GTLV_ScrollWidth // Scroller width
#define DLV_ShowSelected    GTLV_ShowSelected // Show selected
#define DLV_Check           TAG_USER + 10 // Check selection
#define DLV_Highlight       TAG_USER + 14 // Highlight selection
#define DLV_MultiSelect     TAG_USER + 9  // Multi-selection
#define DLV_ReadOnly        GTLV_ReadOnly // Read only
#define DLV_Lines           TAG_USER + 13 // Visible lines (get only ←
)
#define DLV_ShowChecks      TAG_USER + 15 // Show checkmarks
#define DLV_Flags           TAG_USER + 16 // Layout flags
#define DLV_NoScroller      TAG_USER + 17 // No scroller necessary
#define DLV_TopJustify      TAG_USER + 18 // Top-justify items
#define DLV_RightJustify    TAG_USER + 19 // Right-justify items
#define DLV_DragNotify      TAG_USER + 20 // Notify of drags
#define DLV_GetLine         TAG_USER + 25 // Get line from ←
coordinate
#define DLV_DrawLine        TAG_USER + 26 // Draw a line from the ←
listview
#define DLV_Object          TAG_USER + 27 // Pointer to object
#define DLV_DoubleClick     TAG_USER + 28 // Indicates double-click
#define DLV_ShowFileNames   TAG_USER + 48 // Show filenames only

typedef struct
{
    struct RastPort      *rp;
    struct DrawInfo     *drawinfo;
    struct Node         *node;
    unsigned short      line;
    struct IBox         box;
} ListViewDraw;

// Listview node data
#define lve_Flags          ln_Type        // Listview entry flags
#define lve_Pen            ln_Pri        // Listview entry pen
#define LVEF_SELECTED      (1<<0)      // Entry is selected
#define LVEF_USE_PEN       (1<<1)      // Use pen to render entry
#define LVEF_TEMP          (1<<2)      // Temporary flag for ←
something

// File button tags
#define DFB_DefPath        TAG_USER + 19 // Default path

// Palette tags
#define DPG_Pen            TAG_USER + 21 // Ordinal selected pen

```

```
#define DPG_SelectPrevious      TAG_USER + 22          // Select previous pen
#define DPG_SelectNext        TAG_USER + 23          // Select next pen

// Some useful macros
#define GADGET(obj) (obj->gl_info.gl_gadget.gadget)
#define AREA(obj) obj->gl_info.gl_area
#define DATA(win) ((
    WindowData
    *)win->UserData)
#define WINFLAG(win) (
    DATA
    (win)->flags)
#define WINMEMORY(win) (
    DATA
    (win)->memory)
#define WINREQUESTER(win) (
    DATA
    (win)->request)
#define OBJLIST(win) (
    DATA
    (win)->list)
#define DRAWINFO(win) (
    DATA
    (win)->drawinfo)
#define VISINFO(win) (
    DATA
    (win)->visinfo)
#define GADSPECIAL(list,id) (
    GADGET
    (L_GetObject(list,id)->SpecialInfo)
#define GADSTRING(list,id) ((struct StringInfo *)
    GADSPECIAL
    (list,id)->Buffer)
#define GADNUMBER(list,id) ((struct StringInfo *)
    GADSPECIAL
    (list,id)->LongInt)
#define GADSEL(list,id) (
    GADGET
    (L_GetObject(list,id)->Flags&GFLG_SELECTED)
#define GADGET_SPECIAL(list,id) (
    GADGET
    (GetObject(list,id)->SpecialInfo)
#define GADGET_STRING(list,id) ((struct StringInfo *)
    GADGET_SPECIAL
    (list,id)->Buffer)
#define GADGET_NUMBER(list,id) ((struct StringInfo *)
    GADGET_SPECIAL
    (list,id)->LongInt)
#define GADGET_SEL(list,id) (
    GADGET
    (GetObject(list,id)->Flags&GFLG_SELECTED)
#define CFGDATA(win) (((
    WindowData
    *)win->UserData)->data)

#define MENUID(menu) ((ULONG)GTMENUITEM_USERDATA(menu))
```

```
// Layout functions
struct Window *OpenConfigWindow(NewConfigWindow *);
void CloseConfigWindow(struct Window *);

struct IntuiMessage *GetWindowMsg(struct MsgPort *);
void ReplyWindowMsg(struct IntuiMessage *);

        ObjectList
        *AddObjectList(struct Window *,ObjectDef *);
void FreeObjectList (
        ObjectList
        *);

        GL_Object
        *GetObject (
        ObjectList
        *,ULONG);
void SetGadgetValue (
        ObjectList
        *,USHORT,ULONG);
long GetGadgetValue (
        ObjectList
        *,USHORT);
void SetGadgetChoices (
        ObjectList
        *list,ULONG id,APTR choices);
BOOL CheckObjectArea (
        GL_Object
        *,long,long);
void DisableObject (
        ObjectList
        *,ULONG,BOOL);

void DisplayObject(struct Window *,GL_Object *,long,long,char *);
void SetWindowBusy(struct Window *);
void ClearWindowBusy(struct Window *);
long BoundsCheckGadget (
        ObjectList
        *,ULONG,long,long);
BOOL GetObjectRect (
        ObjectList
        *,ULONG,struct Rectangle *);
void SetConfigWindowLimits(struct Window *,ConfigWindow *,ConfigWindow *);
void LayoutResize(struct Window *);

void AddWindowMenus(struct Window *,MenuData *);
void FreeWindowMenus(struct Window *);

void StartRefreshConfigWindow(struct Window *,long);
void EndRefreshConfigWindow(struct Window *);

struct Menu *BuildMenuStrip(MenuData *,struct DopusLocale *);
struct MenuItem *FindMenuItem(struct Menu *,USHORT);
void SetWindowID(struct Window *,WindowID *,ULONG,struct MsgPort *);
ULONG GetWindowID(struct Window *);
struct MsgPort *GetWindowAppPort(struct Window *);
```

```
#define RECTWIDTH(rect)      (1+(rect)->MaxX-(rect)->MinX)
#define RECTHEIGHT(rect)    (1+(rect)->MaxY-(rect)->MinY)

#endif
```