functions

COLLABORATORS			
	TITLE : functions		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 22, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

functions

1.1 GUIEnvironment - Functions Reference

```
                              GUIEnvironment

                         Functions Reference Guide

 ==========================================================================

                         © 1994   Carsten Ziegeler
                                  Augustin-Wibbelt-Str.7
                                  D-33106 Paderborn
                                  Germany

 ==========================================================================
```

1.2 guienv.library/ChangeGUI

```
NAME
    ChangeGUIA -- Change the GUI appearance.
    ChangeGUI -- Varargs stub for ChangeGUIA.

SYNOPSIS
    error = ChangeGUIA( gui, tagList )
    D0                  A0   A1

    WORD ChangeGUIA( struct GUIInfo *, struct TagItem * );

    error = ChangeGUI( gui, Tag1, ... )

    WORD ChangeGUI( struct GUIInfo *, ULONG, ... );

FUNCTION
    This is the general method of changing the GUI appearance. It
    includes the displaying of the GUI as well as some important
    entries in the GUIInfo stucture.

INPUTS
    gui = pointer to the GUIInfo structure
    tagList = pointer to array of TagItems

RESULT
    If everything went fine, you will get GE_Done, otherwise some
    of the error codes defined in guienv.h.

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     The GUI tags
     The error codes
```

1.3 guienv.library/CloseGUIFont

```
NAME
    CloseGUIFont -- Close a font opened with OpenGUIFont

SYNOPSIS

    CloseGUIFont( font )
                  A0

    VOID CloseGUIFont( struct TextFont * );

FUNCTION
    This function closes a font, previously opened with OpenGUIFont.

INPUTS
    font = pointer to the font
```

```
SEE ALSO
     OpenGUIFont
```

1.4 guienv.library/CloseGUIScreen

```
NAME
    CloseGUIScreen -- Safe closing of a screen

SYNOPSIS
    CloseGUIScreen( screen )
                    A0

    VOID CloseGUIScreen( struct Screen * );

FUNCTION
    Before the screen is closed, all not yet closed windows on this
    screen using CloseGUIWindow are closed.

INPUTS
    screen = pointer to a screen open with OpenGUIScreen.

NOTES
    Use this function only for screens opened with OpenGUIScreen.
    It is not necessary for own screens, to close the windows by hand,
    as well as it is not necessary to free the GUIInfo structures
    by hand.

SEE ALSO
     OpenGUIScreen
     CloseGUIWindow
```

1.5 guienv.library/CloseGUIWindow

```
NAME
    CloseGUIWindow -- Safe closing of a window

SYNOPSIS
    CloseGUIWindow( window )
                    A0

    VOID CloseGUIWindow( struct Window * );

FUNCTION
    All outstanding messages will be replied before the window is
    closed and all GUIInfo structures which are in use for this
    window are automatically removed using FreeGUIInfo.

INPUTS
    window = pointer to a window opened with OpenGUIWindow.

NOTES
    Use this function only for windows opened with OpenGUIWindow.
    It is not necessary to free the GUIInfo structures by hand.

SEE ALSO
     FreeGUIInfo
     OpenGUIWindow
```

1.6 guienv.library/CreateGUIGadget

```
NAME
    CreateGUIGadgetA -- Add a gadget to the GUI
    CreateGUIGadget -- Varargs stub for CreateGUIGadget

SYNOPSIS
    CreateGUIGadgetA( gui, left, top, width, height, kind, tagList)
                      A0   D0    D1   D2     D3      D4    A1

    VOID CreateGUIGadgetA( struct GUIInfo *, WORD, WORD, WORD, WORD, LONG,
                           struct TagItem * );

    CreateGUIGadget( gui, left, top, width, height, kind, Tag1, ... )

    VOID CreateGUIGadget( struct GUIInfo *, WORD, WORD, WORD, WORD, LONG,
                          ULONG, ... );

FUNCTION
    This functions combines the GadTools CreateGadget and the Intuition
    NewObject functions together with an own gadget creation function.
    The gadget is linked into an internal list and will be displayed
    if DrawGUI is called.

INPUTS
    gui = pointer to a GUIInfo structure
    left = gadget left edge
    top = gadget top edge
    width = gadget width
    height = gadget height
    kind = gadget kind (GadTools or GUIEnv)
    tagList = pointer to array of TagItems

NOTES
    Even if the gadget can't be created, there will be no error code,
    because this function has no return value. An error will occure
    when the DrawGUI function is called.
    The gui pointer is not tested for NULL.
    You are limited to 256 gadgets for each GUIInfo !

SEE ALSO
     DrawGUI
     The gadget tags
```

1.7 guienv.library/CreateGUIInfo

```
NAME
    CreateGUIInfoA -- Create the import GUIInfo structure
    CreateGUIInfo -- Varargs stub for CreateGUIInfoA

SYNOPSIS
    gui = CreateGUIInfoA( window, tagList )
    D0                    A0      A1

    struct GUIInfo *CreateGUIInfoA( struct Window *, struct TagItem * );

    gui = CreateGUIInfo( window, Tag1, ... )

    struct GUIInfo *CreateGUIInfo( struct Window *, ULONG, ... );

FUNCTION
    This function allocates memory for the important GUIInfo structure.
    This structure contains some very important and usefull information
    about the GUI.
    The GUIInfo structure is bound to an previously opened window.

INPUTS
    window = pointer to an opened window
    tagList = pointer to array of TagItems

RESULT
    A pointer to a full initialized GUIInfo structure.

SEE ALSO
     FreeGUIInfo
     The GUI Tags
```

1.8 guienv.library/CreateGUIMenuEntry

```
NAME
    CreateGUIMenuEntryA -- Add a new menu item to the menu
    CreateGUIMenuEntry -- Varargs stub for CreateGUIMenuEntry

SYNOPSIS
    CreateGUIMenuEntryA( gui, type, text, tagList )
                         A0   D0    A1    A2

    VOID CreateGUIMenuEntryA( struct GUIInfo *, BYTE, STRPTR,
                              struct TagItem * );

    CreateGUIMenuEntry( gui, type, text, Tag1, ... )

    VOID CreateGUIMenuEntry( struct GUIInfo *, BYTE, STRPTR, ULONG, ... );

FUNCTION
    This function creates a new menu item, which is linked to the last
    created one.

INPUTS
    gui = pointer to a GUIInfo structure
    type = the menu item type (menu title, menu item or menu sub item)
    text = the menu item text
    tagList = pointer to array of TagItems

NOTES
    Even if the menu item can’t be created, there will be no error code,
    because this function has no return value. An error will occure
    when the DrawGUI function is called.
    The gui pointer is not tested for NULL.
    You are limited to 256 menu items for each GUIInfo !

SEE ALSO
     The menu tags
```

1.9 guienv.library/DrawGUI

```
NAME
    DrawGUIA -- Draw all gadgets and set the menu
    DrawGUI -- Varargs stub for DrawGUIA

SYNOPSIS
    error = DrawGUIA( gui, tagList )
    D0                A0   A1

    WORD DrawGUIA( struct GUIInfo *, struct TagItem * );

    error = DrawGUI( gui, Tag1, ... )

    WORD DrawGUI( struct GUIInfo *, ULONG, ... );

FUNCTION
    Draw all gadgets and set the menu. It is possible to change some
    attributes of the GUI and then let it draw using this function.

INPUTS
    gui = pointer to a GUIInfo structure
    tagList = pointer to array of TagItems.

RESULT
    GE_Done if everything went fine, or any error code.

NOTES
    The gui pointer is not tested for NULL.
    The tagList parameter is currently not used, set it to NULL !
    This functions does not first clear the window contents.

SEE ALSO
     The GUITags
     The error codes
```

1.10 guienv.library/FreeGUIInfo

```
NAME
    FreeGUIInfo -- Free all structures for the GUI

SYNOPSIS
    FreeGUIInfo( gui )
                 A0

    VOID FreeGUIInfo( struct GUIInfo * );

FUNCTION
    This function frees all resources allocated with CreateGUIInfo.

INPUTS
    gui = pointer to a GUIInfo structure

SEE ALSO
     CreateGUIInfo
```

1.11 guienv.library/GetCatStr

```
NAME
    GetCatStr -- Get the catalog string

SYNOPSIS
    string = GetCatStr( gui, stringNbr, default )
    D0                  A0   D0         A1

    STRPTR GetCatStr( struct GUIInfo *, LONG, STRPTR );

FUNCTION
    This function tries to get a localized string out of the catalog
    defined in the catalogInfo entry of the GUIInfo structure.

INPUTS
    gui = pointer to a GUIInfo structure
    stringNbr = the number of the string in the catalog
    default   = the default string

RESULT
    If the number and the catalog are available a pointer to the
    localized string is returned, otherwise the default string is
    returned.

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     Localization
```

1.12 guienv.library/GetGUIGadget

```
NAME
    GetGUIGadget -- Get gadget attribute

SYNOPSIS
    data = GetGUIGadget( gui, number, attribute )
    D0                   A0   D0      D1

    LONG GetGUIGadget( struct GUIInfo *, WORD, Tag );

FUNCTION
    Try to get a gadget attribute value.

INPUTS
    gui = pointer to a GUIInfo structure
    number = gadget number / gadget ID
    attribute = the gadget attribute

RESULT
    The data of the attribute or -1 if the attribute wasn't available.

NOTES
    The gui pointer is not tested for NULL.
    If the gadtools.library version is lower than 39, you can't get
    attributes of gadtools gadgets which are managed by gadtools !

BUGS
    If you are running under pre 39 versions of gadtools, GetGUIGadget
    calls nevertheless the 39 function to get gadtools gadget attributes,
    so probably your task will crash !

SEE ALSO
     The gadget tags
```

1.13 guienv.library/GetGUIMsg

```
NAME
    GetGUIMsg -- Look at the message port for a new message

SYNOPSIS
    success = GetGUIMsg( gui )
    D0                   A0

    BOOL GetGUIMsg( struct GUIInfo * );

FUNCTION
    Tries to get a message and then this message is handled by GUIEnv.

INPUTS
    gui = pointer to a GUIInfo structure

RESULTS
    TRUE if there was any message or FALSE if not.

NOTES
```

```
    The gui pointer is not tested for NULL.

SEE ALSO
     Message handling
     WaitGUIMsg
```

1.14 guienv.library/GetLocStr

```
NAME
    GetLocStr -- Get a string of the locale environment

SYNOPSIS
    string = GetLocStr( gui, stringNbr, default )
    D0                  A0   D0         A1

    STRPTR GetLocStr( struct GUIInfo *, LONG, STRPTR );

FUNCTION
    This function tries to get a localized string out of the locale
    environment defined in the localeInfo entry of the GUIInfo structure.

INPUTS
    gui = pointer to a GUIInfo structure
    stringNbr = the number of the string in the locale environment
    default   = the default string

RESULT
    If the number and the locale environment are available a pointer
    to the localized string is returned, otherwise the default string
    is returned.

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     Localization
```

1.15 guienv.library/GUIGadgetAction

```
NAME
    GUIGadgetActionA -- Do a gadget action
    GUIGadgetAction -- Varargs stub for GUIGadgetActionA

SYNOPSIS
    GUIGadgetActionA( gui, tagList )
                      A0   A1

    VOID GUIGadgetActionA( struct GUIInfo *, struct TagItem * );

    GUIGadgetAction( gui, Tag1, ... )

    VOID GUIGadgetAction( struct GUIInfo *, ULONG, ... );
```

```
FUNCTION
    Perform some actions on some gadgets, e.g disable/enable, getting
    and setting...

INPUTS
    gui = pointer to a GUIInfo structure
    tagList = pointer to array of TagItems

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     The Gadget Action
```

1.16 guienv.library/GUIRequest

```
NAME
    GUIRequestA -- Show a requester
    GUIRequest -- Varargs stub for GUIRequestA

SYNOPSIS
    success = GUIRequestA( gui, text, kind, tagList )
    D0                     A0   A1    D0    A2

    LONG GUIRequestA( struct GUIInfo *, STRPTR, LONG, struct TagItem * );

    success = GUIRequest( gui, text, kind, Tag1, ... )

    LONG GUIRequest( struct GUIInfo *, STRPTR, LONG, ULONG, ... );

FUNCTION
    This function provides an easy interface to the intuition requesters,
    the asl file requesters and some of the ReqTools requesters.

INPUTS
    gui = pointer to a GUIInfo structure
    text = pointer to the requester text
    kind = the requester kind
    tagList = pointer to array of TagItems

RESULT
    See RKRMs for EasyReq and Asl functions resp the ReqTools docs.

NOTES
    It is possible to pass NULL as gui pointer for warning the user.
    When doing so, all extra functions (like localization etc) are
    not available. This should only be used for error messages
    when it is not possible to open the GUI !

SEE ALSO
     The Requester guide
```

1.17 guienv.library/OpenGUIFont

```
NAME
    OpenGUIFont -- Open a font

SYNOPSIS
    font = OpenGUIFont( name, size, fontAttr )
    D0                  A0    D0    A1

    struct TextFont *OpenGUIFont( STRPTR, WORD, struct TextAttr * );

FUNCTION
    Open the font and fill the TextAttr structure with the data of
    this font.
    This function makes no difference if the font is a rom or a disk
    font.

INPUTS
    name = the font name (e.g. "topaz.font")
    size = font height
    fontAttr = pointer to a TextAttr structure or NULL. If you specify
               this, the structure will be filled with the data about
               the opened font.

RESULT
    Pointer to the font or NULL if it failed.

SEE ALSO
     CloseGUIFont
```

1.18 guienv.library/OpenGUIScreen

```
NAME
    OpenGUIScreenA -- Open a new screen
    OpenGUIScreen -- Varargs stub for OpenGUIScreenA

SYNOPSIS
    screen = OpenGUIScreenA( id, depth, name, tagList )
    D0                       D0  D1     A0    A1

    struct Screen *OpenGUIScreenA( ULONG, WORD, STRPTR, struct TagItem * );

    screen = OpenGUIScreen( id, depth, name, Tag1, ... )

    struct Screen *OpenGUIScreen( ULONG, WORD, STRPTR, ULONG, ... );

FUNCTION
    Open the screen.

INPUTS
    id = the screen ID
    depth = number of bitplanes for this screen
    name = pointer to the screen's title text
    tagList = pointer to array of TagItems (See intuition's screen
              tags)

RESULT
    Pointer to the opened screen or NULL if it failed.

SEE ALSO
     CloseGUIScreen
```

1.19 guienv.library/OpenGUIWindow

```
NAME
    OpenGUIWindowA -- Open a new window
    OpenGUIWindow -- Varargs stub for OpenGUIWindowA

SYNOPSIS
    window = OpenGUIWindowA( left, top, width, height, name, idcmpFlags,
    D0                       D0    D1   D2     D3      A0    D4

                             windowFlags, screen, tagList )
                             D5           A1      A2

    struct Window *OpenGUIWindowA( WORD, WORD, WORD, WORD, STRPTR, ULONG,
                                   ULONG, struct Screen *, struct TagItem * );

    window = OpenGUIWindow( left, top, width, height, name, idcmpFlags,
                            windowFlags, screen, Tag1, ... )

    struct Window *OpenGUIWindow( WORD, WORD, WORD, WORD, STRPTR, ULONG,
                                  ULONG, struct Screen *, ULONG, ... );

FUNCTION
    This function openes a new window.

INPUTS
    left, top, width, height = the dimensions of the window, where the
                               width and height define the inner size
                               of the window.
                               Use the GEW_OuterSize tag to define
                               the "usual" size of a window (with any
                               value) !
    name = pointer to window's title text
    idcmpFlags = IDCMP flags for this window
    windowFlags = the flags for this window
    screen = The screen the window is opened on or NULL for the current
             public screen
    tagList = pointer to array of TagItems (See also intuition's
              window tags)

RESULT
    Pointer to the opened window or NULL if it failed.

NOTES
    Don't use GimmeZeroZero windows within GUIEnvironment !

SEE ALSO
```

```
     CloseGUIWindow
```

1.20 guienv.library/SetGUIGadget

```
NAME
    SetGUIGadgetA -- Set gadget attributes
    SetGUIGadget -- Varargs stub for SetGUIGadgetA

SYNOPSIS
    SetGUIGadgetA( gui, number, tagList )
                   A0   D0      A1

    VOID SetGUIGadgetA( struct GUIInfo *, WORD, struct TagItem * );

    SetGUIGadget( gui, number, Tag1, ... )

    VOID SetGUIGadget( struct GUIInfo *, WORD, ULONG, ... );

FUNCTION
    Set gadget attributes regardless if it is a gadtools, a BOOPSI or
    any other gadget.

INPUTS
    gui = pointer to a GUIInfo structure
    number = the gadget number / ID
    taglist = pointer to a taglist of attributes

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     Getting and setting gadget attributes
```

1.21 guienv.library/ShowGuideNode

```
NAME
    ShowGuideNodeA -- Show a AmigaGuide text node
    ShowGuideNode -- Varargs stub for ShowGuideNodeA

SYNOPSIS
    error = ShowGuideNodeA( gui, guide, node, tagList )
    D0                      A0   A1     A2    A3

    WORD ShowGuideNodeA( struct GUIInfo *, STRPTR, STRPTR, struct TagItem * );

    error = ShowGuideNode( gui, guide, node, Tag1, ... )

    WORD ShowGuideNode( struct GUIInfo *, STRPTR, STRPTR, ULONG, ... );

FUNCTION
    Show a AmigaGuide node in synchron mode.
```

```
INPUTS
    gui = pointer to a GUIInfo structure
    guide = pointer to the AmigaGuide filename
    node = pointer to the node name
    tagList = pointer to array of TagItems

RESULT
    If the node could be displayed, GE_Done is returned, otherwiese
    GE_GuideErr. But notice, that this error could have many different
    reasons: no AmigaGuide, no Guide, wrong node name etc.

NOTES
    The gui pointer is not tested for NULL.
```

1.22 guienv.library/WaitGUIMsg

```
NAME
    WaitGUIMsg -- Wait for a message and handle it

SYNOPSIS
    WaitGUIMsg( gui )
                A0

    VOID WaitGUIMsg( struct GUIInfo * );

FUNCTION
    Wait for a message and then handle it.

INPUTS
    gui = pointer to a GUIInfo structure

NOTES
    The gui pointer is not tested for NULL.

SEE ALSO
     Message handling
     GetGUIMsg
```

1.23 rcs

```
$RCSfile: Functions.guide $

$Revision: 1.7 $
    $Date: 1994/12/16 21:21:29 $

  GUIEnvironment Function Reference

  Copyright © 1994, Carsten Ziegeler
                    Augustin-Wibbelt-Str.7, 33106 Paderborn, Germany
```