c.generator

COLLABORATORS			
	TITLE : c.generator		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

c.generator

1.1 C.generator Help

c.generator v1.0

User's Manual

- FreeWare -

Introduction...

Requirements

Usage

Using c.generator...

Preferences

Source code

Other topics...

Copyrights

Disclaimer

History

Author

==========

¶ ¶ ¶\¶ ¶ /\ ¶\¶

|_| | | | \/ | |

1.2 Usage

Usage

=====

To generate source code you must have selected in GadToolsBox preferences window c.generator and then Project/Save Source As...

C.generator can also be run from CLI, GadToolsBox must be running and have loaded resource file:

FILE=<filename> - base file name to generate source

SEMAPHORE=<semaphore> - public GadToolsBox semaphore name

SETPREFS - only open preferences window

1.3 Requirements

Requirements

==========

C.Generator requires to run:

OS:

AmigaOS 3.0+

Processor:

MC680x0

Software:

GadToolsBox must be running!

1.4 Preferences

Preferences

==========

Gadgets:

Template

Tells generator to create <basename>_temp.c file with templates of called functions.

Screen

Source code will have SetupScreen() function and all other screen releated functions.

GUI Functions

Source code will have all gui creation routines like CreateGadgetEx().

Localize

Source code will be localized. <basename>.cd file will be created.

1.5 Source code

Source code

==========

Variables

Functions

1.6 Functions

Functions

=========

<basename>.c/ AdaptX

<basename>.c/ AdaptY

<basename>.c/ CalcAdaptivity

<basename>.c/ CalcLeftTop

<basename>.c/ CalcRightBottom

<basename>.c/ CloseDownScreen

<basename>.c/ CloseWindow_#?

<basename>.c/ CloseWindowEx

<basename>.c/ CreateGadgetExA

<basename>.c/DeleteGadgetExA

<basename>.c/ FixGadgetTags

<basename>.c/ FixMenus

<basename>.c/ FixWindowTags

<basename>.c/ FreeApplication

<basename>.c/ GetCatString

<basename>.c/ HandleIDCMP_#?

<basename>.c/ InitApplication

<basename>.c/ OpenWindow_#?

<basename>.c/ OpenWindowTagsEx

<basename>.c/ Render_#?

<basename>.c/ SetupScreen

1.7 Variables

Variables

=========

<basename>.c/ #?Gadgets

<basename>.c/ #?GTags

<basename>.c/ #?Menus

<basename>.c/ #?Msg

<basename>.c/ #?NewMenu

<basename>.c/ #?WApp

<basename>.c/ #?Wnd

<basename>.c/ #?WTags

<basename>.c/ DrawInfo

<basename>.c/ Scr

<basename>.c/ ScrDepth

<basename>.c/ ScrDisplayID

<basename>.c/ ScrHeight

<basename>.c/ ScrPubName

<basename>.c/ ScrTitle

<basename>.c/ ScrWidth

1.8 Copyrights

Copyrights

==========

C.generator

This program is FREEWARE. Generator can be freely distributed but only in original archive.

1.9 Disclaimer

Disclaimer

==========

C.GENERATOR IS A FREEWARE PRODUCT. USE IT AT YOUR OWN RISK. THERE IS NO WARRANTY FOR RELIABLE FUNCTIONING OF THIS PROGRAM.

1.10 History

History

=======

27.5.98 - v1.0

- initial release.

1.11 How to contact

Author

======

This program was developed and tested on my Amiga 4000/040/28MHz/16MB and SAS/C v6.58.

If you have new ideas, bug reports, catalog translations or something other - write to me:

smail:

Tomasz Muszynski

ul. Orzycka 4 m138

02-695 Warszawa

POLAND

mailto:thom@union.org.pl

http://thom.union.org.pl

1.12 AdaptX

NAME

AdaptX -- Calculate new horizontal dimension.

SYNOPSIS

x = AdaptX(value);

UWORD AdaptX(UWORD);

FUNCTION

Converts horizontal value in internal dimensions to new dimensions

based on current font size.

Before this function you must call CalcAdativity() .

INPUTS

value - value to convert

RESULT

x - new horizontal value

BUGS

This function has problems with proportional fonts.

SEE ALSO

AdaptY() , CalcAdaptivity()

1.13 AdaptY

NAME

AdaptY -- Calculate new vertical dimension.

SYNOPSIS

x = AdaptY(value);

UWORD AdaptY(UWORD);

FUNCTION

Converts vertical value in internal dimensions to new dimensions

based on current font size.

Before this function you must call CalcAdativity() .

INPUTS

value - value to convert

RESULT

x - new vertical value

BUGS

This function has problems with proportional fonts.

SEE ALSO

AdaptX() , CalcAdaptivity()

1.14 CalcAdaptivity

NAME

CalcAdaptivity -- Calculate new dimensions and select proper font.

SYNOPSIS

CalcAdaptivity(width,height);

void CalcAdaptivity(UWORD,UWORD);

FUNCTION

This function checks passed width and height to fit on screen. If
new dimensions will be to big, topaz.font/8 will be used instead of
screen font.

INPUTS

width - max width (usually width of window)

height - max height (usually height of window)

RESULT

Global variable Font will be set to new font.

SEE ALSO

AdaptX() , AdaptY()

1.15 CalcLeftTop

NAME

CalcLeftTop -- Calculate left and top window border.

SYNOPSIS

CalcLeftTop(offx, offy, tags);

void ClacLeftTop(UWORD *,UWORD *,struct TagItem *);

FUNCTION

Calculates left and top border of closed window basing on tags.

INPUTS

offx - pointer to store new value in

offy - pointer to store new value in

tags - window tags

RESULT

offx - left border of window

offy - top border of window

SEE ALSO

CalcRightBottom()

1.16 CalcRightBottom

NAME

CalcRightBottom -- Calculate right and bottom window border.

SYNOPSIS

CalcRightBottom(offx, offy, tags);

void CalcRightBottom(UWORD *,UWORD *,struct TagItem *);

FUNCTION

Calculates right and bottom border of closed window basing on tags.

INPUTS

offx - pointer to store new value in

offy - pointer to store new value in

tags - window tags

RESULT

offx - right border of window

offy - bottom border of window

SEE ALSO

CalcLeftTop()

1.17 CloseDownScreen

NAME

CloseDownScreen -- close opened screen.

SYNOPSIS

CloseDownScreen();

void CloseDownScreen(void);

FUNCTION

Closes screen opened by SetupScreen().

SEE ALSO

SetupScreen()

1.18 CloseWindow_#?

NAME

CloseWindow_#? -- close opened window.

SYNOPSIS

CloseWindow_#?();

void CloseWindow_#?(void);

FUNCTION

Close window opened by OpenWindow_#?() .

SEE ALSO

OpenWindow_#?()

1.19 CloseWindowEx

NAME

CloseWindowEx -- close opened window.

SYNOPSIS

CloseWindowEx(window);

void CloseWindowEx(struct Window *);

FUNCTION

Closes window opened by OpenWindowTagsEx() .

INPUTS

window - pointer to opened window

SEE ALSO

OpenWindowTagsEx()

1.20 CreateGadgetExA

NAME

CreateGadgetExA -- Create gadtools or BOOPSI gadget.

SYNOPSIS

gad = CreateGadgetExA(offx,offy,offr,offb,prev,tags);

struct Gadget *CreateGadgetExA(UWORD,UWORD,UWORD,UWORD,

struct Gadget *, struct TagItem *);

FUNCTION

Create gadget, calculate it's position using offx, offy,

offr and offb.

INPUTS

offx - left border returned by CalcLeftTop()

offy - top border returned by CalcLeftTop()

offr - right border returned by CalcRightBottom()

offb - bottom border returned by CalcRightBottom()

prev - pointer to previous gadget

tags - pointer to tagitems. First tag is KIND of gadget and first

data is gadtools placement.

RESULT

gad - pointer to created gadget

SEE ALSO

DeleteGadgetEx()

1.21 DeleteGadgetEx

NAME

DeleteGadgetEx -- delete gadget

SYNOPSIS

DeleteGadgetEx(gad);

void DeleteGadgetEx(struct Gadget *);

FUNCTION

Delete gadget created by CreateGadgetExA() .

INPUTS

gad - pointer to gadget

SEE ALSO

CreateGadgetExA()

1.22 FixGadgetTags

NAME

FixGadgetTags -- fixes all gadgets to use locale.

SYNOPSIS

FixGadgetTags(tags);

void FixGadgetTags(struct TagItem *);

FUNCTION

Finds all strings in tagitems and localizes them.

INPUTS

tags - pointer to tags to fix

SEE ALSO

1.23 FixMenus

NAME

FixMenus -- fixes menu to use locale.

SYNOPSIS

FixMenus(menu);

void FixMenus(struct NewMenu *);

FUNCTION

Finds all strings in menu and localizes them.

INPUTS

menu - pointer to menu to fix

SEE ALSO

1.24 FixWindowTags

NAME

FixWindowTags -- fixes window to use locale.

SYNOPSIS

FixWindowTags(tags);

void FixWindowTags(struct TagItem *);

FUNCTION

Finds all strings in tagitems and localizes them.

INPUTS

tags - pointer to tags to fix

SEE ALSO

1.25 FreeApplication

NAME

FreeApplication -- frees all opened libraries.

SYNOPSIS

FreeApplication();

void FreeApplication(void);

FUNCTION

Closes all libraries opened by InitApplication() .

SEE ALSO

InitApplication()

1.26 GetCatString

NAME

GetCatString -- gets localized string from catalog.

SYNOPSIS

str = GetCatString(id);

STRPTR GetCatString(APTR);

FUNCTION

Gets localized string from catalog table. This function also works

if source code is not localized.

INPUTS

id - id of string to get

RESULT

str - pointer to localized string

SEE ALSO

1.27 HandleIDCMP_#?

NAME

HandleIDCMP_#? -- handles idcmp messages of given window.

SYNOPSIS

retval = HandleIDCMP_#?();

int HandleIDCMP_#?(void);

FUNCTION

Handles all IDCMP messages of given window. When IDCMP is handled it calls a callback function and returns value returned by callback. When 0 was returned program wants to quit and close all windows at the end of program. When 1 was returned program countinues normally. When higher values are returned it means that IDCMP window was closed and HandleIDCMP_#?() shouldn't get next messages from list.

RESULT

retval - value returned by callback function

SEE ALSO

1.28 InitApplication

NAME

InitApplication -- initializes new application.

SYNOPSIS

locale version

error = InitApplication(language);

UWORD InitApplication(STRPTR);

non-locale version

error = InitApplication();

UWORD InitApplication(void);

FUNCTION

Opens all libraries required by GUI.

INPUTS

language - language to use

RESULT

error - error. See <basename>.h/GTBERR_#? defines to see what goes wrong.

SEE ALSO

FreeApplication()

1.29 OpenWindow_#?

NAME
OpenWindow_#? -- open defined window.
SYNOPSIS
error = OpenWindow_#?();
UWORD OpenWindow_#?(void);
FUNCTION
Creates all gadgets, menus and opens window.
RESULT
error - error. See <basename>.h/GTBERR_#? defines to see what
goes wrong.
SEE ALSO
CloseWindow_#?()

1.30 OpenWindowTagsEx

NAME
OpenWindowTagsEx -- open window.
SYNOPSIS
window = OpenWindowTagsEx(tags, ...);
UWORD OpenWindowTagsEx(ULONG, ...);
FUNCTION
Open window with given tags. This window has expanded features
to old intuition.library/OpenWindowTags(). You can create appwindow,
make new dragbars, open it at special dimensions (eg. centered), etc.
INPUTS
tags - a list of tags
RESULT
window - pointer to opened window
SEE ALSO
CloseWindowEx()

1.31 Render_#?

NAME
Render_#? -- make all window renderings
SYNOPSIS
Render_#?();
void Render_#?(void);
FUNCTION
Do all renderings on window.
SEE ALSO

1.32 SetupScreen

NAME

SetupScreen -- open and setup screen

SYNOPSIS

error = SetupScreen(num);

UWORD SetupScreen(UBYTE);

FUNCTION

Opens screen and initializes GUI (eg. creates image objects, menus, etc).

INPUTS

num - number of screen to open. This number will be used for all ports

that this program uses (eg. Arexx port, public screen name). Also

screen title can contain that number.

RESULT

error - error. See <basename>.h/GTBERR_#? defines to see what

goes wrong.

SEE ALSO

CloseDownScreen()

1.33 #?Gadgets

NAME

#?Gadgets -- table of created gadgets

SYNOPSIS

struct Gadget *#?Gadgets[];

DESCRIPTION

This is table of initialized gadgets. It's only valid beetween

OpenWindow_#?() and CloseWindow_#?() .

Use GD_#? defines to access to this table.

SEE ALSO

1.34 #?GTags

NAME

#?GTags -- tags of gadgets

SYNOPSIS

ULONG #?GTags[];

DESCRIPTION

This is table of tags for gadgets.

SEE ALSO

1.35 #?Menus

NAME

#?Menus -- created menu

SYNOPSIS

struct Menu *#?Menus;

DESCRIPTION

This is initialized menu. It's only valid beetween

OpenWindow_#?() and CloseWindow_#?() .

SEE ALSO

1.36 #?Msg

NAME

#?Msg -- messages structure

SYNOPSIS

struct IntuiMessage #?Msg;

DESCRIPTION

This is IntuiMessage structure returned by exec.library/GetMsg().

This is copy of original message, so verify messages will not work.

It's only valid after HandleIDCMP_#?() .

SEE ALSO

1.37 #?NewMenu

NAME

#?NewMenu -- table of menu defines

SYNOPSIS

struct NewMenu #?NewMenu[];

DESCRIPTION

Table of menu defines to create menu.

SEE ALSO

1.38 #?WApp

NAME

#?WApp -- appwindow data

SYNOPSIS

APTR #?WApp;

DESCRIPTION

Data returned by workbench.library/AddAppWindow().

SEE ALSO

1.39 #?Wnd

NAME

#?Wnd -- pointer to opened window

SYNOPSIS

struct Window *#?Wnd;

DESCRIPTION

This is window structure. It's only valid beetween

OpenWindow_#?() and CloseWindow_#?() .

SEE ALSO

1.40 #?WTags

NAME

#?WTags -- tags of window

SYNOPSIS

ULONG #?WTags[];

DESCRIPTION

This is table of tags for window.

SEE ALSO

1.41 DrawInfo

NAME

DrawInfo -- draw information

SYNOPSIS

struct DrawInfo *DrawInfo;

DESCRIPTION

This is drawinfo structure. It's only valid beetween

SetupScreen() and CloseDownScreen() .

SEE ALSO

1.42 Scr

NAME

Scr -- screen structure

SYNOPSIS

struct Screen *Scr;

DESCRIPTION

This is screen structure. It's only valid beetween

SetupScreen() and CloseDownScreen() .

SEE ALSO

1.43 ScrDepth

NAME

ScrDepth -- screen depth to open

SYNOPSIS

UBYTE ScrDepth;

DESCRIPTION

Depth of screen to open.

SEE ALSO

1.44 ScrDisplayID

NAME

ScrDisplayID -- screen displayid to open

SYNOPSIS

UBYTE ScrDisplayID;

DESCRIPTION

DisplayID of screen to open.

SEE ALSO

1.45 ScrHeight

NAME

ScrHeight -- screen height to open

SYNOPSIS

UWORD ScrHeight;

DESCRIPTION

Height of screen to open.

SEE ALSO

1.46 ScrPubName

NAME

ScrPubName -- screen public name to open

SYNOPSIS

UBYTE ScrPubName[30];

DESCRIPTION

Name of public screen to create. If %d exists in that string it will be

filled with screen number (name of public screen should look eg: EXAMLE.1).

All public ports should be also created with that string.

SEE ALSO

1.47 ScrTitle

NAME

ScrTitle -- screen title to open

SYNOPSIS

UBYTE ScrTitle[256];

DESCRIPTION

Name of public screen to create. If %d exists in that string it will be

filled with screen number (name of public screen should look eg: EXAMLE.1).

All public ports should be also created with that string.

SEE ALSO

1.48 ScrWidth

NAME

ScrWidth -- screen width to open

SYNOPSIS

UWORD ScrWidth;

DESCRIPTION

Width of screen to open.

SEE ALSO