GadTools34

COLLABORATORS

	TITLE : GadTools34		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 1, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

GadTools34

1.1 Documentation of gadtools.library 34.1 and GTE.lib

GadTools Library 34.1 and GTE.lib 1.0

by Massimo Tantignone (27/01/93)

A MT Soft production

1.2 Introduction

1. INTRODUCTION

Gadtools.library 34.1 is a 1.3-compatible version of the library with

the same name normally present in the Kickstart 2.04 or higher.

To be precise, it's a shared library whose aim is to emulate the

gadtools.library V37 and that works even under operating systems previous

to 2.0, such as the OS 1.3.

GTE.lib is, instead, an object module linker library and performs the

same task.

These two public domain libraries allow to write and/or use programs

taking advantage even under 1.3 of the gadtools.library in the building and

handling of a graphical user interface (as long as these programs don't
require other 2.0-specific functions).
Both libraries are dedicated mainly to programmers, which will be able
to freely distribute gadtools.library 34.1 along with their programs, or
to directly link the code of GTE.lib to these programs (using the linker).
If nevertheless there were in circulation already made programs using
the gadtools.library AND NO OTHER 2.0 LIBRARY OR FUNCTION, these should
also work after the installation of gadtools.library 34.1 into your
Workbench 1.3's LIBS: directory (of course such programs can be found out
only by trying them).
For that, gadtools.library 34.1 can come out useful also to the final
user (but I doubt at the moment there be already such programs).
In the distribution directory of gadtools.library 34.1 and GTE.lib you
find two test programs: one of them (TestLinked) was linked with GTE.lib
and therefore contains the gadtools emulation code, the other (TestShared)
uses instead the external gadtools.library. The first can be executed
anytime, while to execute the other you need to install first the
gadtools.library 34.1 as explained in the next paragraph (or run it under
2.0 or higher, but in that case it would use the REAL gadtools.library).
Note: with the exception of the format (shared library or linker
library) the two libraries are ABSOLUTELY IDENTICAL in their working,
as they were obtained from the same source code.

1.3 Usage of gadtools.library 34.1

2. USAGE OF GADTOOLS.LIBRARY 34.1
To use gadtools.library 34.1 you must only copy it into your Workbench
1.3's LIBS: directory (floppy or hard disk). Every program attempting to
open the gadtools.library will work from that moment on even under 1.3 and
it will open this library instead of the "true" one of the 2.0 operating
system, without noticing any difference.
Obviously, under 1.3, the program must not make use of other libraries
reserved to 2.0 different from the gadtools.library.
This library must NOT be instead copied into the LIBS: directory of
a Workbench 2.0, as the 2.0 operating system already has its own
gadtools.library.
Note: the library is distributed with the name "gadtools_34.library"
for copyright reasons. After you have copied it into LIBS: you must
rename it exactly to "gadtools.library", or else it won't work.
If you don't want to manually copy the library you can run the automatic
installation script named "InstallGadTools34".

1.4 Usage of GTE.lib

3. USAGE OF GTE.LIB

GTE.lib, as it is a linker library, is useful only to programmers.

To take advantage of this library you must only write your source code

as you would write it if you did use the true gadtools.library, but adding

the prefix "NL_" to the name of each gadtools function you call.

Of course after the compilation you need to link with GTE.lib, and you

will obtain a program working under any operating system and having an user

interface that emulates the gadtools.library V37 under 1.x while directly

uses it under 2.0.

By adding the prefix "EF_" instead of "NL_", the program will be

slightly shorter but it will work in emulation mode EVEN under 2.0.

In both cases you need to #include into your source code the header

file "nl_gadtools.h" supplied with GTE.lib.

1.5 Why these two libraries exist

4. WHY THESE TWO LIBRARIES EXIST

The gadtools.library is certainly one of the most useful libraries

supplied starting with the 2.0 version of the Amiga operating system, as

it allows every program to implement graphical user interfaces that are

consistent and similar to each other in the look and feel.

Alas, however useful, that library isn't compatible with the 1.3 (or

lower) operating system, that, although obsolete, will be certainly used

on the oldest Amigas for some years yet.

Every new application making use of the gadtools.library for its own

user interface is thus useless under the 1.3 OS, even if it doesn't use

any other 2.0-specific function.

To remedy this problem I wrote a library that emulates version 37 of

the gadtools.library and can be integrated into one's own applications.

Actually I did two versions of it; a shared library (gadtools.library

34.1) and an object module linker library, that I named GTE.lib (GadTools

Emulation Library).

1.6 Technical details on GTE.lib

5. TECHNICAL DETAILS ON GTE.LIB

Some precisation about the usage of GTE.lib must be done.

At the moment it's only available the version for the SAS/C 6.x system.

I don't see this as a limitation, as sooner or later all "serious"

programmers should update to this version (and now they got one more reason

to do so).

With the SAS 6.x compiler you can use GTE.lib with

SC LINK LIB=GTE.lib <Your source code file names>

GTE.lib supports and emulates all gadtools.library V37 functions. These

functions are supplied in two versions: the functions with the prefix "EF_"

and those with the prefix "NL_".

The first are the true emulation functions and by calling them under

any operating system (1.x or 2.x) it'll be executed the code that emulates

the gadtools.library.

The others are named "interface functions": they, if called, will decide

on the basis of the operating system under which the program is running

whether to execute the emulation code (i. e. to call the EF_ functions) or

jump directly to the true gadtools.library functions.

Briefly, by adding the NL_ prefix to all gadtools function calls in

your source code and by linking then with GTE.lib, you will get a program

using the true gadtools.library under the 2.0 (or higher) operating system

while executing the emulation code under 1.3 (or lower).

This of course implies the increase of the size of your executable by

about 30-40K.

Alternatively you can compile two versions of the same source code, the

one using only true calls to gadtools, the other calling only the emulation

functions, without passing through the interface ones (whose object code

won't be thus linked to the program) and make two versions of the same

application, a smaller one working only under 2.0, and a larger one working

under both operating systems but always in emulation mode.

In this case the choice about which one to use will be of the user.

Of course all this is valid only assuming that the program doesn't make

use of any other new 2.0 function or library except the gadtools.library

functions.

Note: if the increase in size of the executable is unacceptable the

only alternative is to use the shared version of the emulation library

(gadtools.library 34.1).

1.7 GTE.lib usage example

6. GTE.LIB USAGE EXAMPLE

Let's see now an example to make things clear:

A classic program using gadtools could for instance contain this fragment:

```
if (gad = CreateContext(&glist))
{
gad = CreateGadget(MX_KIND,gad,&ng,GTMX_Labels,labels,TAG_END);
}
```

and it would work only under 2.0 or higher.

By using GTE.lib you can modify it into

```
if (gad = NL_CreateContext(&glist))
{
gad = NL_CreateGadget(MX_KIND,gad,&ng,GTMX_Labels,labels,TAG_END);
}
```

making it working under any operating system, in emulation mode under 1.x and calling the true gadtools.library under 2.x.

Alternatively you can modify it so:

```
if (gad = EF_CreateContext(&glist))
{
gad = EF_CreateGadget(MX_KIND,gad,&ng,GTMX_Labels,labels,TAG_END);
}
```

This version also works with any operating system, but it operates always in emulation mode even under 2.x, and that allows to reduce slightly the executable's size, but obviously it isn't the best in terms of efficiency and prevents the program from taking advantage of possible future enhancements of the true gadtools.library.

Which method to use is each programmer's choice; that which is really important, if you use GTE.lib, is to always respect the following five rules:

· ALWAYS #include into your source code the file "nl_gadtools.h", supplied along with GTE.lib;

· DON'T ever use in your program other 2.x-only functions or libraries except for the gadtools.library;

· DON'T use more than one method in the same program: the functions you use must be all true gadtools functions, or all EF_, or all NL_;

· Always OPEN the gadtools.library if you are under 2.x and you use the NL_ functions (or directly the true ones, but this is obvious);

· Always OPEN the intuition.library and graphics.library.

1.8 Important note about GTE.lib

7. IMPORTANT NOTE ABOUT GTE.LIB

If you examine the file GTE.lib with "OML GTE.lib l" you'll notice that

it also contains many others functions other than those of the emulation

of gadtools.

They are used by the emulation functions and for the moment you must NOT

attempt to directly call them.

In the future I'll distribute these functions (and many other too) as

a stand-alone linked library with full documentation and perhaps even as a

shared library.

The only functions (that are present both in NL_ and EF_ version) that

you already can use are the following:

NL_FindTagItem(), NL_GetTagData(), NL_NextTagItem();

EF_FindTagItem(), EF_GetTagData(), EF_NextTagItem();

emulating three functions of the utility.library (another new 2.0 library).

For the usage of these you must follow the same rules I listed for the

gadtools emulation functions; besides, you must #include into your source

code the file "nl_utility.h", supplied with GTE.lib.

In the future these three functions will be distributes in a library

partially or fully emulating the utility.library.

1.9 Important note about gadtools.library 34.1 and GTE.lib

8. IMPORTANT NOTE ABOUT GADTOOLS.LIBRARY 34.1 AND GTE.LIB

The GetVisualInfoA() function of gadtools.library 34.1, just like its

corresponding EF_ and NL_ functions of GTE.lib, and unlike the omonymous

function of the true gadtools.library, accepts even a NULL as the first

argument, that is as the pointer to the screen; in this case it assumes

you mean the Workbench screen.

The purpose of this is to avoid to have to find the Workbench screen:

in fact, the intuition LockPubScreen() function, that under 2.x is used

just in order to do this, doesn't exist under 1.3 (and thus it must not

be called!!!!).

You'll need then to use some conditional code here, as the true

gadtools.library V37's GetVisualInfoA() function DOESN'T accept a NULL as

an argument.

Example:

if (SysBase->lib_Version >= 36) /* If we are under 2.0 or higher */

```
{
screen = LockPubScreen(NULL);
if (!screen) Error("No Workbench");
}
else /* If we are under 1.3 or lower */
{
screen = NULL;
}
visualinfo = GetVisualInfoA(screen,NULL);
... /* Rest of the program here */
FreeVisualInfo(visualinfo);
if (screen) UnLockPubScreen(NULL,screen);
```

1.10 Differences

9. DIFFERENCES
There are of course some slight differences between the working of GTE.lib (or gadtools.library 34.1) and that of the true gadtools.library, and they are all listed in the file "Differences.doc". It is however a matter of minimal and secondary details (at least I think so).

1.11 Conclusion

10. CONCLUSION
This is the first version of gadtools.library 34.x and of GTE.lib I released, therefore there is a high chance of finding bugs in its working. If this should happen, please report the problem to me and I will fix it (if possible) in a next release.

Write to:

Massimo Tantignone
via Campagnoli, 4
28100 Novara (NO)
ITALY

To conclude, I wish to mention that I'm working on the emulation of other 2.x libraries and I'll distribute them also when they are finished.

Massimo Tantignone, 27 january 1993