

rcsgui

COLLABORATORS

	<i>TITLE :</i> rcsgui		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		September 19, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	rcsgui	1
1.1	RCSGUI © Tommy Svensson 1995	1
1.2	RCSGUI © Tommy Svensson 1995	2
1.3	RCSGUI © Tommy Svensson 1995	2
1.4	(;-) (-;) (-;) (-;) (-;)	3
1.5	RCSGUI © Tommy Svensson 1995	3
1.6	RCSGUI © Tommy Svensson 1995	4
1.7	RCSGUI © Tommy Svensson 1995	5
1.8	RCSGUI © Tommy Svensson 1995	7
1.9	RCSGUI © Tommy Svensson 1995	10
1.10	RCSGUI © Tommy Svensson 1995	11
1.11	RCSGUI © Tommy Svensson 1995	11
1.12	RCSGUI © Tommy Svensson 1995	12

1.2 RCGUI © Tommy Svensson 1995

INTRODUCTION

RCSGUI is as the name sounds like, a GUI for RCS. It of course does not include RCS. It

requires
HWGRCS.

I decided to start using RCS instead of copying old versions to another directory, or archiving it in a lha file. I downloaded it from Aminet, installed it, and played around with it a little. At first it seemed very nice (I still think it is), but since I have the SAS/C compiler, which have very nice workbench support, which I prefer using over using it from CLI, it became very irritating having to open a shell to check my files in and out of RCS. Since the icons for my source files were still left it was very hard to see if a file was checked out or not, and if it was locked.

I started dreaming, thinking that it would be nice if I could have files checked in and out by just dragging and dropping the icons. I started playing with this idea, and RCGUI was born.

1.3 RCGUI © Tommy Svensson 1995

COPYRIGHT/SHAREWARE

RCSGUI is Copyright © 1995 by Tommy Svensson.

This software is subject to the "Standard Amiga FD-Software Copyright Note" It is SHAREWARE as defined in paragraph 4s. For more information please read "AFD-COPYRIGHT" (Version 1 or higher).

I have decided to make this version shareware. I need some extra money to help finance an MMU ;-). If you like and use RCGUI, please send

me

150 SEK / 20 USD or an equal amount
in your local currency. The easiest is probably to send cash (well hidden). I'll also accept what my bank calls commercial checks. These are the kind of checks that you buy from the bank. They are not personalized. They should be in either SEK (Swedish crowns) or in the local currency, otherwise it will cost me more to cash them. I do not accept eurochecks as I have seen other shareware programmer do. My bank says that I will have problems cashing those. I will however accept US personal checks since I have a special way of cashing those.

Please include your name and address or email address with your payment.

I have not crippled RCGUI in any way. This benefits us both. You dont have to wait for a registered version or a key before you can use it. I dont have to create a personalized registered copy for each registration, and mail it out on disk, or e-mail which doesn't always work. Its up to your conscience to pay!

1.4 (;-) (;-) (;-) (;-) (;-)

This uses the new CASTSPELL feature of Amigaguide. Its A feature added for shareware programmers. A spell is cast to make the user feel obligated to pay the shareware fee or feel really bad.

The spell have been cast, can you feel it ?

1.5 RCGUI © Tommy Svensson 1995

FEATURES

RCGUI implements the basic features of RCS: checking in, checking out, locked and unlocked, rlog, and rcs -n to assign a symbolic name to a rev. It does not implement all options to all RCS commands.

Available in previous version (1.41):

Checkin in by drag and drop.

Checkout with double click or checkout button.

Lock/nolock selection for checkin and checkout.

Completely font sensitive GUI. The control panel buttons and the icons will be distributed according to window size.

All

configuration
can be done in the
tooltypes
of the icon.

Some of the parameters can also be saved with the snapshot button in the window. (Button was previously called SSP).

Shows all rcs operations in a console window.

Uses separate repositories for files and icons. (Icons are actually

not checked in under RCS, they are just kept in a directory).

New features:

Checkout by drag and drop.

Double click will also edit or view the file after checkout.

Separate lock/nolock selection for checkin and checkout. (usually you want nolock on checkin, but lock on checkout).

The state of the toggle gadgets are also saved on snapshot.

If the window isn't big enough to hold all icons, a scrollbar will be shown.

Can now hide or show the rcs operations in the console window with the Verbose/Quiet toggle button.

Will create icons in the icon repository for files that doesn't have any icons there. Useful when you use RCSGUI on an old RCS repository.

Uses HWGRCSs RCS_link feature. (Now actually requires HWGRCS).

Also implements an RCSGUI_link for the icon repository.

Will let you specify repository dirs with a file requester if no _link files are found in current directory. The _link files will then be created.

Support for suffix (-x option to the rcs commands).

You can set the background color to use for both the icon part of the window and the control panel.

It works nicely with the SAS/C WB compiler environment, but have no connection to the SAS/C compiler. It can be used with any compiler that lets you edit & compile from the workbench. (Well, it can be used with any CLI based compiler too, but it wouldn't make sense).

It can actually be used with anything that you use RCS for. It doesn't have to be programming. You could checkin documents, pictures, ray tracings, music, etc. Anything you want to keep several revisions of.

1.6 RCSGUI © Tommy Svensson 1995

REQUIREMENTS

RCSGUI was written on an Amiga 4000/030 with AmigaDos 3.0 (version 39), but should work on any Amiga with AmigaDos 2.04

(version 37) or above. It might even work on 2.0 (version 36), but it uses some functions that are described as buggy under version 36.

HWGRCS is now required! RCGUI is based on HWGRCS patch level 11, but might work with earlier patch levels. HWGRCS is available on AmiNet.

1.7 RCGUI © Tommy Svensson 1995

INSTALLATION / CONFIGURATION

Copy the rcsgui executable anywhere you want. The icon is a project icon and should have the default tool point to wherever you copy rcsgui. The icon should be put in each directory where you have stuff that you want to checkin to RCS. If you are using SAS/C I suggest that you put the icon under SC:starter_project. Then whenever you run SCSetup, it will be copied along with the other icons.

The following tooltypes are available:

RCSHOME have been removed since HWGRCS requires RCS: to be assigned to the bin directory of the HWGRCS installation. RCGUI will also use RCS:.

REPOSHOME=directory

This is just a suggested path that will be put in the file requester when RCGUI is started in a new directory without any RCS_link file. If this is commented out an RCS directory in the current directory will be suggested. If you have all your repositories in one place, point REPOSHOME there.

SUFFIX=suffix

This is the suffix to use on the files in the RCS repository. -x<suffix> will be passed to ci, co, and rcs.

EDITOR=editor

This is the editor to use for editing log messages. Default is ED.

VIEWER=viewer

This is the viewer to use for RLog output. Default is SYS:Utilities/More.

CILOCK=LOCK or NOLOCK

This is the status that the CILock/CINoLock toggle button will be started with. This tooltype is modified by the SnapShot button in the window.

COLOCK=LOCK or NOLOCK

This is the status that the COlock/CONoLock toggle button will be started with. This tooltip is modified by the SnapShot button in the window.

QUIET=TRUE or FALSE

If TRUE the Quiet/Verbose toggle button will be set to Quiet, and if FALSE to Verbose. This tooltip is modified by the SnapShot button in the window.

SPACEICONSEQUAL

If this tooltip is specified all icons will be aligned with the space between them set by the largest icon. This looks much nicer, but takes more space.

BGCOLOR=color

This is the color to use for the background of the icon part of the window. Default is color 0.

ICONTEXTCOLOR=color

This is the color to use for the icon name label text.

CONTROLCOLOR=color

This is the color to use as background in the control panel. Note that this does not affect the color of the control panel buttons.

WINDOWWIDTH=width

The window will be opened with this width if specified. Default 350. This tooltip is modified by the SnapShot button in the window.

WINDOWHEIGHT=height

The window will be opened with this height if specified. Default 200. This tooltip is modified by the SnapShot button in the window.

WINDOWTOPEDGE=topedge

The window will be opened with its top edge at this position. Default is to let intuition place it. This tooltip is modified by the SnapShot button in the window.

WINDOWLEFTEDGE=leftedge

The window will be opened with its left edge at this position. Default is to let intuition place it. This tooltip is modified by the SnapShot button in the window.

For WINDOWTOPEDGE and WINDOWLEFTEDGE both have to be set for it to work. Otherwise Intuition will place the window.

1.8 RCGUI © Tommy Svensson 1995

OPERATION

When RCGUI starts up, it makes the directory where the RCGUI icon you clicked on to start it resides the current directory. The RCS_link file is then checked if it exists. If it exists the first line is read to get the repository directory. If it doesn't exist an RCS directory in the current directory will be checked if it exists. If so it will be used. Otherwise a file requester will be shown asking for RCS repository directory. If the directory specified doesn't exist it will be created. An RCS_link file will also be created then. Then the same procedure is done for the icon repository, but replacing RCS_link with RCGUI_link, and the RCS directory in the current directory with RCGUIIcons.

If no icon is found for a file in the icon repository, or if a file without an icon is checked in, an icon will be created. ENV:sys/def_<extension>.info will be used as icon if it exist, otherwise the default project icons will be used.

The title of the window contains "RCGUI <version> Dir: <dirname>" where <version> is the version of RCGUI, and <dirname> is the name of the directory where the RCGUI icon you clicked on to start it resides. You can only check in files from this directory, and checkout to this directory.

On top of the window you have the "control panel". It is a set of 10 buttons, and one string gadget.

The following "control panel" gadgets are available:

[All] [None] [CILock/CINoLock] [COLock/CONoLock] [Verbose/Quiet]
[Checkout] [RLog] [SymName] [string gadget] [SnapShot] [About]

Click here for a demo. The icons for the files in the current

directory will be displayed, and no operations will be done.

All

Selects all icons in the window.

None

Unselects all icons in the window. Clicking anywhere in the window where there is no icon or gadget will also unselect all icons, just like in a WB window.

CILock/CINoLock

This is a toggle button. The state of this button affects checkin. If CILock then `ci -l <...>` will be done on checkin, otherwise `ci -u <...>` will be done.

COLock/CONoLock

This is a toggle button. The state of this button affects checkout. If COLock `co -l <...>` will be done on checkout, otherwise `co <...>` will be done.

Verbose/Quiet

This is a toggle button. If set to Quiet `-q` will be added to each rcs command. If set to Verbose the output of each rcs command will be displayed in a console window. Note that if set to Quiet any errors will still be displayed in the console window. No output is redirected to NIL: by R CSGUI. This button only sets the `-q` flag.

Checkout

This button will checkout all selected icons. See checkout below for more info on checkout.

RLog

This will run `rlog` on the selected files/icons. The result will be displayed with the viewer you specified in the

VIEWER

tooltype

.

SymName

This will assign a symbolic name to all the selected files. The name should be specified in the string gadget. `rcs -n <string gadget>` will be done. No ":" will be automatically added to the name. Therefore you can also use this to remove a symbolic name. See RCS docs for more info.

string gad

The text in the string gadget is used by checkin, checkout, and symname.

Snapshot

This button will save the window size and position, and the state of the 3 toggle buttons in the icons tooltypes.

About

This will bring up an about requester. It contains the RCGUI version, my name and address, both the RCS and icon repository paths, and the number of files checked in.

Checkin

There is no button for checkin. Checkin is done by dragging and dropping the file you want to checkin on the window. The editor will then be brought up to edit the log message. Checkin looks both at the state of the CILock/CINoLock button, and the string gadget. If the string gadget is empty "ci <file>" will be done, and rcs will bump the revision from the old revision. If a rev is specified in the string gadget ci -r<string gadget> will be done. If the CILock/CINoLock button is set to CILock, -l will also be added to the command. This will cause RCS to immediately checkout the file with a lock again after checking it in. RCGUI will then only copy the icon to the RCGUIIcons dir, and not remove it from the current dir.

PLEASE NOTE that no verification of the path is done on a dropped file, you can checkin a file from any directory! Checkouts however can only be done to the current directory (the directory RCGUI was started from).

Checkout

There are 3 ways to checkout a file:

If you want to checkout several files at the same time, you have to select the files, and then press the checkout button.

Double clicking on an icon will check the file out, and also bring it up in the editor if COLock otherwise the file will be brought up in the viewer.

Dragging an icon and dropping it somewhere outside of RCGUIs window will checkout that file. Since the workbench doesn't support drag and drop from an application, only to it, there is no way for RCGUI to know if it was dropped on a workbench window, and if so what the path to that window is. So dropping anywhere outside of RCGUIs window is considered an OK drop. You have to drag an icon at least 5 pixels in any direction before it is registered as a drag. This decrease the chance of accidentally starting a drag when trying to double click.

Files will always be checked out to the directory that RCGSUI was started from (which was set to the current directory on startup).

The checked out file(s) will be locked or not locked depending on the state of the COLock/CONoLock button. Icons will only be checked out if the file is locked. This makes only locked files visible on the workbench (if you dont do show all).

If the string gadget is empty `co <...>` will be done otherwise `co -r<string gadget> <...>` will be done.

RCSSGUI will open a console window for the output of the `ci`, `co`, and `rco` commands. This console window is of type AUTO/WAIT. That means that it wont open until there is output in it, and wont dissappear until it is closed with the close button. This console window is acutally a built in feature of the SAS/C compiler. If you set the Verbose/Quiet button to Quiet no output will be generated by the `rco` commands, and the console window will never open, unless there is an error.

If you get more icons than will fit in the window, a scrollbar will be shown. This scrollbar does not scroll rows! It scrolls icons. For example, say you have 9 files checked in. The window contains 6 icons. That means that 3 icons are not visible. The scrollbar can be moved 8 (total-1) positions. Each position will increase the offset of the first icon. So if the scrollbar is on top the first 6 icons are shown. If the scrollbar is at the bottom, only the last icon will be shown. The point of this is that as long as the window is big enough to hold the largest icon, all icons can be reached with the scrollbar. At first I made it scroll only the number of icons not visible, so that when the scrollbar was at the bottom the last x icons was shown, where x is the number of visible icons in the window. This was no good. If there was a very large icon, that needed to be in the first position in the window to fit in the window, you could never reach it if it came last. Making the number of scroll positions be the number of icons - 1 solves this problem. NOTE that if you have the SPACEICONSEQUAL tooltype set, the number of columns in the window might differ as you scroll. The number of columns is dependent on the sizes of the icons currently displayed in the window.

You might notice that the buttons, and string gadget have smaller edges than gadgets you see in other programs. This is because all the control panel gadgets are BOOPSI gadgets. I'm not using gadtools because gadtools would have been to complicated here, and I dont really like gadtools. I think that Commodore should have implemented all the gadtools gadgets as BOOPSI gadgets instead.

1.9 RCGSUI © Tommy Svensson 1995

COMMENTS

If you make the window so small that no icon will fit in there

no icon will be displayed. I didn't want to waste any memory on a gimezerozero window with 2 rastports. This will cause gadgets to overwrite the window border instead of being clipped. Therefore the icons have to fit completely in the window for them to be shown at all.

If you have icons of different sizes in the window, you might get less icons in the window than will fit in there, even though you know that there are more icons. R CSGUI will think that it have filled up the window when an icon wont fit vertically in the window. For example, if you have a window wide enough to fit 5 icons in one row, but the third icon is higher than the other icons, and dont completely fit vertically, it wont be displayed, and no more icons after it either. I haven't figured out how to check if the window really is full in a smarter way yet. For now, just make the window a little bit higher, so that all the icons in a row will fit in the window. The scrollbar will also help here, but you cannot get any icon that doesn't fit vertically in the window.

If you are wondering about the version number (1.41 -> 2.41), I dont update the version number only when I make a new release, but whenever I think a change should increase the version. I can go through several versions between each release. I will however update the major number for each release. That the minor number happen to be the same as in the previous release is just a coincident.

1.10 R CSGUI © Tommy Svensson 1995

BUGS

I got a bug report on the previous version (1.41) that it caused an enforcer hit on resize. I have the cheap 68E030, whitout a real MMU in my machine (A4000/030). Therefore I can't use enforcer, and am not that familiar with it. I have looked over the resize routine to see what could possible have caused an enforcer hit. I couldn't find anything. The resize routine have changed quite a lot in this version, so maybee the problem have been acidentally fixed (I can always hope ;-). Anyway, I have not been able to make R CSGUI crash in any way.

1.11 R CSGUI © Tommy Svensson 1995

FUTURE

Depending on the response I get on this (how many people pay), I will continue and improve R CSGUI. If you have paid, you will of course get any future versions free. You only have to pay once.

I'm considering:

- Adding a general options window for all options, so that the icon tooltypes wouldn't have to be edited directly.
- Adding an options window for each rcs command where you can specify all parameters for the commands. Similar to SCOptions of SAS/C.
- Adding a line under the text label of an icon with the version of the file which should also reflect any symbolic name specified in the string gadget.

1.12 RCSGUI © Tommy Svensson 1995

AUTHOR

Tommy Svensson
Fregattvägen 36
11768 Stockholm
Sweden

E-mail: bms@algonet.se

My E-mail address might change in the near future. I'll update this package if and when it does.