

# README

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> README	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		March 28, 2022
		<i>SIGNATURE</i>

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>README</b>	<b>1</b>
1.1	README.guide	1
1.2	README.guide/Copyrights	2
1.3	README.guide/Introduction	3
1.4	README.guide/Requirements	4
1.5	README.guide/Installation	4
1.6	README.guide/Updates	5
1.7	README.guide/Support	6
1.8	README.guide/Amiga Specifics	8
1.9	README.guide/RCS_link	8
1.10	README.guide/Filename Expansion	10
1.11	README.guide/Response Files	11
1.12	README.guide/Easy Project Setup	12
1.13	README.guide/Amiga rcsfreeze	12
1.14	README.guide/Amiga Comment Leaders	12
1.15	README.guide/Timezone Support	13
1.16	README.guide/RCSINIT	13
1.17	README.guide/Reference Directory	14
1.18	README.guide/New RCS Keywords	14
1.19	README.guide/Recompiling	18
1.20	README.guide/Archive Names	18
1.21	README.guide/Release Notes	19
1.22	README.guide/Patch Level 2	20
1.23	README.guide/Patch Level 3	20
1.24	README.guide/Patch Level 4	21
1.25	README.guide/Patch Level 5	21
1.26	README.guide/Patch Level 6	22
1.27	README.guide/Patch Level 7	24
1.28	README.guide/Patch Level 8	25

---

# Chapter 1

# README

## 1.1 README.guide

HWGRCS

\*\*\*\*\*

This document contains all the information about HWGRCS that is not contained in the original RCS man pages. It currently describes 'HWGRCS Patch Level 8'. Release notes for earlier releases can be found in

Release Notes

.

HWGRCS specific documentation:

Copyrights

Please take a look at this.

HWGRCS Introduction

Starting out with HWGRCS.

Requirements

The setup you need.

Installation

How to set up a working HWGRCS.

Updates

You sure want updates, don't you?

Support

You wouldn't mind support either, would you?

Bug Reports

Nobody is perfect.

Amiga Specific Features

The Amiga is some nice computer system.

---

Recompiling  
    For the hackers ...

Archive Names  
    What belongs to HWGRCS?

Release Notes  
    Don't forget to read these!

RCS 5.6.0.1 documentation:

RCS introduction            What is RCS anyway?  
RCS man pages               Docs for all the RCS commands.

DIFFUTILS and PATCH documentation:

DIFF docs

Please don't think that I wrote RCS. RCS is quite some piece of SW. But the work that I invested into HWGRCS is not exactly small either. This work is dedicated to Joan Thuesen.

Heinz Wrobel

NOTE: On pre V39 (< Workbench 3.0) AmigaGuide you will see some funny marks on the screen. They are V39 formatting commands which make the text look nicer under >=V39. They do not reduce the amount of information shown in this document.

## 1.2 README.guide/Copyrights

Copyrights

\*\*\*\*\*

HWGRCS is distributed under the GPL, with some additional things I want that do not restrict the GPL contents:

No changes to the HWGRCS distribution allowed unless you do all of this:

1. Change the name of the full distribution to something other than HWGRCS. I don't want 'bogus' HWGRCS versions floating around. It is mainly the prefix HWG that I am talking about here.
2. Mark clearly what and where your changes are and that I have nothing to do with them.
3. Leave my copyright notices and log comments in there, if there are any.

Again: If you decide to play with the sources, that's fine. But I don't want you to keep the name HWGRCS. Please rename it to whatever other name you like. (This name HWGRCS is mine, all mine)

---

Please note especially the 'No Warranty' sections of the GPL.

DISCLAIMER: I HOPE THIS STUFF HELPS YOU AS IT DOES ME. IF IT TRASHES YOUR SOURCES OR DOES ANY OTHER DAMAGE TO YOU OR YOUR COMPUTER I WON'T BE HELD RESPONSIBLE FOR IT IN ANY CASE!

For information about support of HWGRCS, read  
Support  
.

Thank you for reading this.

## 1.3 README.guide/Introduction

Introduction

\*\*\*\*\*

Welcome to the wonderful world of HWGRCS!

What is HWGRCS?

=====

HWGRCS is a port of GNU RCS 5.6.0.1 to the Amiga with several Amiga specific extensions. See

Amiga Specifics  
, for more info.

Here is a quote that describes it quite well, I think:

The Revision Control System (RCS) manages multiple revisions of files. RCS automates the storing, retrieval, logging, identification, and merging of revisions. RCS is useful for text that is revised frequently, for example programs, documentation, graphics, papers, and form letters.

This and the RCS Introduction in the main menu should help you decide if you can need it.

What is in the archive?

=====

HWGRCS contains a complete port of RCS 5.6.0.1, the GNU diffutils 2.6 and GNU patch 2.1 executables and my LP utility to support paged diff outputs. To get the docs and sources for 'diff' and 'patch', you will need to get the 'HWGdiffXX' and 'HWGpatchXX' archives ('XX' denotes a revision number). I will update these archives as long as their contents are used in HWGRCS. Both archives can be found (as HWGRCS) on any Aminet FTP site. See

Support  
, and  
Archive Names  
, for

more info.

Why is it called HWGRCS?

=====

Though the names of all the executables are unchanged, I called this distribution HWGRCS instead of just RCS. I hope to avoid confusion about what people refer to this way. So if you or anybody refers to this stuff, please call it HWGRCS. This way I get the blame and glory for what I did and maybe a chance to hear about bugs. Talking about glory, please read

Support

.

Man, this guy writes some strange text!

=====

Please remember that english is not my native language. So if my choice of words in this document may occasionally puzzle you a bit, send me mail with a suggestion on how to improve the wording of the text. See

Support

, for my address.

## 1.4 README.guide/Requirements

Requirements for using HWGRCS

\*\*\*\*\*

You need an Amiga with a harddrive. An OS version  $\geq$  V37 (i.e. Kickstart 2.04) is required. You won't get far if you don't have 1MB free RAM space.

## 1.5 README.guide/Installation

Installation of HWGRCS

\*\*\*\*\*

Picking a location

=====

First you need to find a place to put HWGRCS. Basically you can put it anywhere on your HD you like.

Don't forget to use the '-a' option if you unpack the archive with 'lha'. The file attributes should be preserved when unpacking the archive.

Please don't change the directory structure and

---

don't put other tools into the HWGRCS directory tree. If you do it anyway you will probably have a problem updating HWGRCS easily. Putting a 'path' command into your 'S:User-Startup' is much less of a problem.

Adding HWGRCS to your command set

=====

To make HWGRCS commands work, you need to do a few things:

1. Add an 'Assign RCS: <hwgrcsplace>/bin' and a 'Path RCS: add' to your 'S:User-Startup'. '<hwgrcsplace>' denotes the place where you put HWGRCS and there the directory where this very document is located. In other words: 'RCS:' must point to the 'bin' directory in the distribution.

You might want to add the 'Path' statement before e.g. your compiler path if you use e.g. SAS/C. The GNU 'diff' used in HWGRCS is very powerful and you might prefer it to the SAS/C version.

2. Set the timezone correctly. See  
    Timezone Support  
    , for more info.
3. Set 'ENV:USER' to your user name. This name will appear in the log. An alias for this is 'ENV:LOGNAME'. Note that the user name should be like a login name or email name, e.g. 'heinz' instead of 'Heinz Wrobel' on my computer.
4. Read at least some of the RCS docs. As you are reading the docs now it should be no problem for you to go up to the main menu and browse through the RCS documentation.
5. Reboot

Notes on using HWGRCS

=====

When you decide to use HWGRCS, please send me some email. I'll put you into a mailing list for info on bugs, workarounds, and misc. news concerning HWGRCS. See

    Support  
    , for more info.

## 1.6 README.guide/Updates

Updates

\*\*\*\*\*

You can pick up updates for HWGRCS from any Aminet FTP site. Updates are announced to all users on my mailing list and on comp.sys.amiga.announce. I might do extra announcements in other



places, but that depends on the circumstances.

If you don't have FTP access, I am willing to send you the archives per snail for \$10 or DM 20,- up front. Why this price for two disks? I am not really set up for this: I need to buy stamps and envelopes, and go to the post office. Very annoying. ;^)

See

Support  
, for information about the mailing list and my address.

## 1.7 README.guide/Support

Support and Bug Reports

\*\*\*\*\*

Bugs most likely have to do with me porting RCS to the Amiga, so the bug report address mentioned in the original docs won't be of much use to you. Please report all HWGRCS bugs only to me. I will forward 'true' RCS bug reports.

There is no free lunch  
=====

You are free to use HWGRCS and it is free for you. I don't charge you for my work that has gone into HWGRCS.

But I'd like you to think of my support of HWGRCS more like 'Giftware'. If you want my support, please send me some 'Motivation' for supporting HWGRCS. My time ain't free and supporting HWGRCS should continue to be fun for me. Books, HD installable games, A9000's, HD's, networking boards, word processors, money, anything nice accepted. This is directed especially but not at all exclusively at the commercial users of HWGRCS.

I don't want to say:

I will answer to your questions if you register. To register send \$x to my address.

I'd rather have your imagination do some work. If you think that HWGRCS is a rather nice piece of supported work, please don't forget me.

No Piracy!  
=====

I will not help anyone who uses pirated software including pirated kickstart or workbench files. Some people seem to see C= OS pirating as an exception. I can't understand that.

Mailing List for HWGRCS  
=====

---

I have set up a huge mail alias for HWGRCS users to notify them on updates, bugs, etc. Anyone who sends me email on HWGRCS for some reason will be put on the list. Don't forget that this mailing list is mainly on my phone bill. I pay for all the mails going out, you pay for one coming in. Please don't be offended when I ask you to reread the 'Giftware' paragraphs above. Thank you!

Information you have to provide

=====

For any question or bug report you send me, I need some information about your setup. Please include always the output of the 'VERSION' command for the HWGRCS commands, your computer configuration, and the OS version. Really helpful would be bug reports created with the C= 'Report' tool. This tool can be found on Fish Disk 920. New versions might be available when you read this. Don't forget to tell me if you are using the C= Shell or some replacement and if the problem occurs with the C= Shell.

The easiest way for you is probably to put that information together right now and save it into a file. You can insert this file then into any email request. Don't forget to update it when your setup changes.

You help me lot in helping you by providing me with this information.

Don't forget to read the docs!

=====

This may sound really stupid, but did you check the docs when you encountered a problem? No? Hmm. Please read them now. Maybe you don't even have to send a bug report. See

Top

, for a pointer to the RCS man

pages.

What's the address?

=====

Please send all gifts, questions, and bug reports to:

Heinz Wrobel  
Karlstr. 16  
82131 Gauting  
Germany  
FAX +49 89 850 5125  
<heinz@hwg.muc.de>

Thank you!

Heinz Wrobel

---

## 1.8 README.guide/Amiga Specifics

Amiga specific Features of HWGRCS

\*\*\*\*\*

HWGRCS contains half a ton of Amiga specific features. Here is a list of the more involved stuff:

RCS\_link

Filename Expansion

Response Files

Easy Project Setup

Amiga rcsfreeze

Amiga Comment Leaders

Timezone Support

RCSINIT

Reference Directory

New RCS Keywords

In addition to that you get some nice, small features:

- All the executables should be residentable.
- Starting with Patch Level 8, local environment variables are checked in addition to 'ENV:'. Note that I use the 'ENV:' prefix in this manual to denote an environment variable. This can be local or global whichever is available. So e.g. a variable described as 'ENV:foo' does not actually have to be in 'ENV:'. It could be a process local environment variable.

## 1.9 README.guide/RCS\_link

The 'RCS\_link' feature

\*\*\*\*\*

'RCS\_link' as repository link

=====

I decided to reimplement the 'RCS\_link' feature found in the RCS 4.x port on the old Fish disks 281/282, and 451.

Usually, HWGRCS puts all the repository files into an 'rcs' subdirectory if available. You might want to have a separate HD partition or network volume for the repositories, though. Unfortunately AmigaOS has still problems with softlinks which are needed to do this through the OS. With 'RCS\_link', you can still have the repository files in a completely different location even though no softlinks are available.

Basically HWGRCS looks for the correct place in this order:

1. Use the first line of the file 'RCS\_link' in the working directory if it exists and is non empty. A slash is added if necessary.
2. Check 'ENV:RCS/DIRECTORY' for the name. A slash is added if necessary.
3. Use 'rcs/' as name.

The first one found will be used.

Check the release notes for HWGRCS Patch level 3 and upwards. See

Patch Level 3  
, for more info.

NOTE: I might remove the check for 'ENV:RCS/DIRECTORY' in a future release or rename the environment variable. I don't think of this as a 'good' feature anymore.

'RCS\_link' as configuration file

=====

Starting with

Patch Level 8  
, 'RCS\_link' has become much more than a simple repository directory link. It is now more a special configuration file where many options pertaining to the current directory can be set.

Note that this makes 'RCS\_link' sort of a two-headed janus file. As configuration file, it is read from the current directory and only on startup of any HWGRCS command. As repository link, only the first line of 'RCS\_link' will be used as described above, which might very well be an 'RCS\_link' file in a completely different directory.

Currently, 'RCS\_link' lets you specify the place where repository files go in its first line, a place where the latest checked in version (the reference version) of the working files go in the second line, and miscellaneous options in the lines after that. What does this mean exactly? Well, the first line and its use has been explained above. In the second line you set the name of the

Reference Directory

. You may

leave it empty as the first line if you don't want to specify a reference directory.

---

The lines after that are in the the format '`<option>: <stuff>`'. The possibly options are:

`'ignore'`

You specify one filename or AmigaDOS filename pattern (no `'*'`!). All arguments specified on the command line that are not starting with a `'-'` and where the filename (ignoring its path) matches the specified pattern will be removed from the command line before processing. This lets you do a `'ci -u #?'` while having specific files automatically ignored. You do not need to specify `'RCS_link'` here. `'RCS_link'` will always be overlooked by all commands capable of handling `'RCS_link'`. Multiple `'ignore'` lines may be used.

`'<cmd>'`

`'<cmd>'` can be one of `'ci'`, `'co'`, `'ident'`, `'merge'`, `'rcs'`, `'rcsclean'`, `'rcsdiff'`, `'rcsfreeze'`, `'rcsmerge'`, and `'rlog'`. The remainder of the line will be used as described in

RCSINIT

.

Note that only one line per command may be used. Also note that any specification will fully override any options described in

RCSINIT

. If you specify any options here,

RCSINIT

, won't be even

looked at.

Exceptions

=====

NOTE: `'rcsfreeze'` does not currently respect any configuration options and flags used in `'RCS_link'` except for the repository link line.

## 1.10 README.guide/Filename Expansion

Automatic filename expansion

\*\*\*\*\*

HWGRCS has been linked with `'hwunix.lib'`. See

Recompiling

, for more

info. It has a modified startup in it. This new startup code does filename expansion via the V37 pattern matching functions for any argument that meets these criteria:

- doesn't start with `'-'`
- and contains at least one of `'?'`, `'*'`, `'#'`, `'~'`, `'('`, `'|'`, or `'\'`

As you see, it won't expand unix style options. I needed this as the shell doesn't support command lines of arbitrary length and the standard SAS/C startup up to and including SAS/C 6.50 doesn't support more than 32 arguments. My `'_main.c'` should be able to handle any number of arguments up to the available memory limit.

You can do a `'ci (#!.c|#!.h)'` just by typing it.

I say this is a hack for several reasons:

1. Anything that meets the criteria mentioned above is considered a filename. No matter what it really is.
2. This filename expansion is done on startup. Things like `'RCS_link'` won't be checked.

Unix path specifications `'..'` and `'./'` are converted to AmigaDOS paths by my modified startup. See

`Patch Level 6`  
, for more info.

Starting with

`Patch Level 8`  
, a `'*'` is converted to `'#?'` before

processing.

Another nice feature is `'response file expansion'`. See

`Response Files`  
, for more info.

## 1.11 README.guide/Response Files

Response File Expansion

\*\*\*\*\*

HWGRCS has been linked with `'hwgunix.lib'`. See

`Recompiling`  
, for more

info. It has a modified startup in it. This new startup code does response file expansion.

If you specify a filename `'@file'`, this file will be opened and the contents read in as arguments for the command. You can use this e.g. to read in a file list if standard pattern matching doesn't do the job for you. See

`Filename Expansion`  
, for more info. Any number of response

files can be specified and recursion is possible to a depth of at least three levels.

Pattern matching is not done on purpose for response file names.

## 1.12 README.guide/Easy Project Setup

Easy Project Setup

\*\*\*\*\*

For each new project you need to set up the repository directory if you don't want to have your working directory cluttered with stuff that does not belong there. Just run the tiny script 'setuprcs' and the directory will be set up for you. If you run it by accident on an existing setup, nothing should be disturbed.

I suggest to use 'setuprcs'. I might enhance it in the future and if you are used to it, life will be even easier for you then.

## 1.13 README.guide/Amiga rcsfreeze

The Amiga 'rcsfreeze'

\*\*\*\*\*

While there was the original rcsfreeze script, you'll find an AmigaOS implementation of rcsfreeze, too. There is one change though which I find useful. If you have a default branch set for a file, this branch number is used to freeze the revision instead of the head value.

Check the release notes for Patch 4 and Patch 7. See  
Patch Level 4

and

'  
Patch Level 7  
, for more info.

## 1.14 README.guide/Amiga Comment Leaders

Comment Leaders for the Amiga

\*\*\*\*\*

I changed the standard setup for the comment leaders a little to support Amiga specific file types a little better. HWGRCS supports correct comment leaders for '.a', '.asm', '.i', and '.fd' files. The comment leader for these files is '\* '. The comment leader for '.guide' and '.ag' files is '@REMARK ', for '.ps' files '% '.

---

## 1.15 README.guide/Timezone Support

Timezone support on the Amiga

\*\*\*\*\*

HWGRCS won't work if your timezone is not set. So you definitely NEED TO SET THE TZ ENVIRONMENT VARIABLE CORRECTLY! If you use at least Workbench 2.1 (>= V38), the choice you made in the 'Locale' Editor will give HWGRCS a nice approximation.

For more detailed timezone support you need to set the environment variable 'TZ' in the SAS/C timezone format.

The SAS/C 'TZ' format is 'aaabbbccc':

'aaa'

the name of your timezone like CST or MET. HWGRCS supports a huge amount of names. You shouldn't have a problem finding one that fits your needs. Just set 'TZ', check in some dummy file and look at the date with the rlog command to see if it works.

'bbb'

the offset in hours to UTC.

'ccc'

If this is present the internal SAS/C variable '\_\_daylight' will be set. As HWGRCS accepts timezone names referring to daylight savings time, I suggest you don't use this field. It's not tested.

On my machine I currently use 'UTC-1' for 'ENV:TZ'.

'TZ' is not stored in 'ENV:RCS' as it should be of 'global' use to you. It is not just used by HWGRCS.

NOTE: Don't forget to copy the timezone to 'ENVARC: '!

## 1.16 README.guide/RCSINIT

The 'RCSINIT' environment variable

\*\*\*\*\*

There is an environment variable RCSINIT mentioned in the RCS man pages. With HWGRCS this is actually 'RCS/INIT' in 'ENV:' to keep everything pertaining to HWGRCS in one place.

NOTE: I am not so sure anymore that it was a good idea to rename this environment variable. I might return to the original 'RCSINIT' in a future release.

With

Patch Level 8

, an additional feature has been added which allows command specific initialisation values. Before 'RCS/INIT' is checked,



the commands look for 'RCS/INIT\_<cmd>'. This variable is used instead if it exists. In other words, it overrides 'RCS/INIT'. '<cmd>' stands for the name the program has been called with (i.e. 'argv[0]' for C). For e.g. 'co', you would set an environment variable named 'RCS/INIT\_co'. Note that all these setting can be overridden by a setting in

```
RCS_link
.
```

## 1.17 README.guide/Reference Directory

The Reference Directory.

\*\*\*\*\*

Say you want to have a directory where you always keep read only copies of the latest working files as reference for you and maybe other developers in a network situation. How would do this with standard RCS? You would use e.g. 'ci -u file' and copy the working file afterwards to your reference directory. Pretty annoying after a while, don't you think?

Starting with

```
Patch Level 8
```

```
, HWGRCS supports the notion of a
```

'Reference Directory'. You can specify a reference directory name in the second line of the file

```
RCS_link
```

```
. On each check in operation, a
```

read only copy of the working file will be automatically put into your reference directory. It is just like you used 'ci -u file' and copied the working file to your reference directory yourself.

## 1.18 README.guide/New RCS Keywords

Amiga specific RCS Keywords

\*\*\*\*\*

NOTE: The keywords described here are non standard. They are only handled by HWGRCS and by no other RCS implementation. They also have some very special behaviour compared to the usual RCS keywords. The implementation (if you want to call it that) is rather hairy and much of a hack. So please read this very carefully.

A table below describes the keywords in detail. As this very document must be RCS compatible, too, I can't use a '\$' character in the keyword examples, as you would have to do in real life. I use '\$' instead in this text.

Starting with

Patch Level 8

, HWGRCS supports four new and very special keywords for the Amiga. The keywords are '\$VERS\$', '\$AmigaREVS\$', '\$AmigaDATES\$', and '\$AmigaProtection\$'.

There are some extremely special and somewhat unusual things about keyword expansion for these new keywords. I put them up front here, even though the keywords are not described yet, because it is very important that you note these points well. When I talk about version and revision numbers below, I don't mean the RCS numbers (unless explicitly stated), but the special ones for the Amiga keywords '\$VERS\$' and '\$AmigaREVS\$'.

1. E.g. the command 'ci' does not take keyword values into account when comparing revisions. So if you e.g. just change the date in a '\$VERS\$' string and do a 'ci -u', this will fail! 'ci' will report an unchanged file, your change will not be checked in, even though it stays in your working file. If you do only changes to any of the keyword values and nothing else that would make 'ci' check in the code, you need to use the '-f' option to force a checkin.
2. An increment of the version/revision or a date update in '\$AmigaDATES\$' will only be done on check in and only if you use a statement like 'ci -l -f'. That is you need to keep a locked working copy of the file you want to check in, and you should force the current working file into the repository.
3. The file is checked in as is, and after that the working copy is updated with the new keyword values. Read below on what this means for version bumping.
4. Again: If you don't use '-l', you won't get any Amiga keyword updates in your working copy.
5. If you don't specify '-f', you won't get any updates either.
6. If you specify '-k', you won't get any updates either.

When I talk about 'check in' below, the above criteria must be fulfilled to make anything happen and I will assume that they are, unless stated otherwise.

For you as user this might mean a change to your work habits. Say you have some file that you want to change where the '\$AmigaREVS\$' revision number is currently 11. You want to create revision 12. These are the steps you have to perform:

1. Check out a copy of the file to work on: 'co -l file'.
  2. Do the needed changes in the working copy for revision 12.
  3. Increase the working copy's '\$AmigaREVS\$' revision number with 'ci -l -f file'. This checks in the changes into the repository and bumps the revision to 12 in the working file.
-

4. Your working file has now the number 12. Note, that the revision number 12 is not yet in the repository, only the changes to the source.
5. Do the check in of the finished revision 12: `'ci -u -f file'`. Now revision 12 is in the repository. You can specify a log message of e.g. `'Amiga Revision bumped'`. A simple Shell alias with the `'-m'` option might be helpful.

As you see there is one 'dummy' check in operation to increase the revision number. The cost of it is about 300 bytes in the repository and one lost RCS revision. For a simple bump of the number without source changes, you loose two RCS revisions because a final `'ci -u -f'` is needed to get the new revision into the repository. Not much to get the Amiga keywords, is it? I strongly suggest to use an alias with the `'-m'` option as mentioned above. This gives you a clear indication in the log where the different Amiga revisions start.

In the smakefile for HWGRCS, I have a special target `'bumprev'`:

```
bumprev:
    @co -q -l conf.h.amiga
    @ci -q -f -l -mAmiga_BUMPREV_Start conf.h.amiga
    @ci -q -f -u -mAmiga_BUMPREV_End conf.h.amiga
    @echo "Bumped patch revision"
```

But let me show you the detailed description of the keywords now. Here it finally comes:

#### `'$VERS'`

This keyword helps you to have a C= style version strings in your files. You have to put a line like this into your files:

```
'$VER: progname 39.0 (28.1.94) $'
```

On each check in the revision number (0 in this example) will be incremented and the date will be set to the current date. Do not try any tricky formatting stunts that violate the C= version string format. The behaviour is undefined in this case.

You can find more information on the format and use of C= style version strings in the 'Amiga User Interface Style Guide', published by Addison Wesley (You should have read it already, anyway). There is further interesting information in the C= AmigaMail issue March/April 1991.

If you still want to handle your own `'$VERS'` without having HWGRCS reformat everything in way that you don't want, use the C string concatenation feature like `''$" "VER: foo''`. This way, HWGRCS will not recognize the keyword. Note that HWGRCS is rather careful though in preserving your layout. There shouldn't be much need for the above mentioned trick unless you want to do extra fiddling with e.g. date macros.

#### `'$AmigaREVS'`

If you need a single revision number or counter that is incremented with each check in, you can use this keyword. You can use it multiple times for multiple counters in one file like this:

```
`#define MYREV /* $AmigaREV: */ 1 /* $ */'
```

Tricky, isn't it? But it works pretty well. The non digit characters are overlooked and preserved. This makes enclosing the special RCS characters with comment tokens possible.

``$AmigaDATES'`

This expands to a date specification of the current date as used in the C= style version strings excluding parentheses. As with ``$AmigaREVS'`, HWGRCS tries to preserve surrounding characters. So you could add parentheses yourself. You can use it like this:

```
`#define MYDATE /* $AmigaDATE: */ "13.02.94" /* $ */'
```

``$AmigaProtection$'`

There are two protection bits on the Amiga that can be quite useful: 'S' for scripts, and 'P' for pure, residentable data. As long as you are using HWGRCS only on Amiga computers, these bits will be preserved. But if you have your repository files via a network on a computer that doesn't understand them, those bits are lost on the way. Just think of a Unix box. It only has read, write, and execute bits. ``$AmigaProtection$'` helps here. If you know that a file needs to have e.g. both 'S' and 'P' set, add a line like this to your file:

```
`$AmigaProtection: SP $'
```

When a file is checked out that contains a keyword line like this, HWGRCS will attempt to set the protection bits accordingly. Note, that case or order is not important when specifying the letters for the bits. Currently only 'S' and 'P' are supported.

Implementing these keywords, even as rudimentary as they are, was non trivial for me. If you look into the source you might get a wrong impression of the work behind it. Please read

Support

.

I don't feel that this is the last word on these keywords yet. Pun intended. The implementation and their behaviour especially with respect to expansion might change. I am e.g. thinking of changing ``$AmigaDATES'` to always expand to the current date on any RCS operation.

If you have any ideas for improvement on all of this, tell me about it! It is still a very open field and unfortunately somewhat limited by the inner workings of RCS. If you hate these keyword because they e.g. break your handling of ``$VERS'` strings, tell me about it, too. If you don't tell me, nothing will get better! For my address, check

Support

.

## 1.19 README.guide/Recompiling

Recompiling HWGRCS

\*\*\*\*\*

To recompile HWGRCS yourself, you need SAS/C 6.51. Earlier releases won't do the job because of dependencies hidden inside my 'hwgunix.lib'.

Some SAS/C compiler headers are missing 'const' keywords in prototypes. You'll see that when you compile HWGRCS and you should correct your headers according to the SAS/C documentation if this problem occurs.

Other than that a simple run of 'SMAKE' should do the job.

All the modifications to the original RCS code are enclosed in '#ifdef \_AMIGA' statements. A short note on why the change is in the code follows each statement.

The source to 'hwgunix.lib' is not included with HWGRCS as 'hwgunix.lib' is not really part of HWGRCS. It is a helper lib that I created for various projects, and contains some private code. It also contains bug fixes to SAS/C and is therefore dependent on the compiler revision. There is nothing HWGRCS specific in there, but if you don't want to use SAS/C, you might have to write your own compiler/environment dependent helper functions.

If you think you have found bugs in 'hwgunix.lib' or have the absolute need to get more information about it, contact me under the address mentioned in

Support

.

Please don't just fix a bug and recompile. Report any bugs to me as described in

Support

. Thanks!

## 1.20 README.guide/Archive Names

HWGRCS Archive Files

\*\*\*\*\*

Starting with

Patch Level 7

, the full HWGRCS distribution comes in

three archives. This is the list of the archive names where '?' will be the current patch level number.

---

**`HWGRCSp?f.lha'**

The main archive contains everything needed to run HWGRCS including AmigaGuide® docs. You have unpacked this archive successfully, because you are reading the docs now.

**`HWGRCSsrcp?f.lha'**

In this archive you will find everything needed to recompile HWGRCS. See  
    Recompiling  
    , for more info.

**`HWGRCSmanp?f.lha'**

This archive contains all the Unix style man pages for RCS. I took them out of the main archive because the AmigaGuide® docs are much nicer.

The archive **`HWGRCSmanp7f.lha'**, done first for

    Patch Level 7  
    , will not

be updated if no changes to the man pages occur in the original RCS distribution. So even if I do a bug fix for HWGRCS, I will not rerelease this archive if it is not changed. Be prepared for the **'7'** in the archive name to stay there for a while.

## 1.21 README.guide/Release Notes

    Release Notes

\*\*\*\*\*

For first time users of HWGRCS, I suggest working your way through all the release notes.

The current release is **`HWGRCS Patch Level 8'**.

    Patch Level 8

    Patch Level 7

    Patch Level 6

    Patch Level 5

    Patch Level 4

    Patch Level 3

    Patch Level 2

---

## 1.22 README.guide/Patch Level 2

Release Notes for HWGRCS Patch Level 2

\*\*\*\*\*

- `'diff'` had problems with the AmigaOS filepath format
- `'-M'` did not work with the appropriate HWGRCS commands because of a library bug in SAS/C.

## 1.23 README.guide/Patch Level 3

Release Notes for HWGRCS Patch Level 3

\*\*\*\*\*

- The `'-j'` option of `'co'` didn't work because of a file that was open to long.
- There were still problems with the networking group/user protection bits. I hope to really finally have this fixed now. I rewrote `'stat()'`, `'fstat()'` and `'chmod()'` for this.
- A positive side effect is, that HWGRCS never leaves the archive bit set now. Previously it restored it after any modification that cleared it. Now incremental backups should be safe.
- `'rcs.guide'` should be V37 AmigaGuide® compatible now. Thanks to Joerg Plate for the patches.
- The old and dreadful `'RCS_link'` feature I took over from Fish RCS 4.x should work more like you would expect it to, now. E.g. if you do a `'co dir/file'` this will (hopefully) happen:

1. Check for `'dir/RCS_link'`, if ok, get the first line out of it as link path.

If it is an absolute path (path with colon) replace `'dir'` with it. Otherwise tack it onto `'dir'`.

Goto 4.

2. Check for `'RCS_link'`, if ok, get the first line out of it as link path.

If it is an absolute path (path with colon) replace `'dir'` with it. Otherwise tack it onto `'dir'`.

Goto 4.

3. Add `'RCS/'` to `'dir'`

4. Check if `'dir/file'` is there, if ok, fine, the end.

5. Use the original `'dir'` and try `'dir/file'`.
- With help of a patient beta tester (Thanks Bernd!), I got around a problem with network filesystems not supporting `'ACTION_EXAMINE_FH'`. Another rewrite of the `'fstat()'` internals ...
  - The `'guide'` extension should be recognized now with a comment leader of `'@REMARK '`.
  - Still no C= style version string. I'm open for suggestions to fit that into the RCS keyword grammar. I won't consider suggestions though that don't include a solution to the problem of keeping the arbitrary text after the date (displayed with `'VERSION FULL'`) without an unwanted `'$'` sign in between.

## 1.24 README.guide/Patch Level 4

Release Notes for HWGRCS Patch Level 4

\*\*\*\*\*

- The `'a'` Ada file extension and its comment leader are now finally replaced by an Amiga assembler compatible `'a'` with a leader of `'*'`.

Therefore the complete list of Amiga specific extensions and comment leaders is now:

```
'a'      '* '
'.fd'    '* '
'.guide' '@REMARK '
'.i'     '* '
'.ps'    '% '
```

No other changes to the extensions have been made.

- `'rcsfreeze'` is now a SAS/C 6.3 source. This was done because the script froze the branch number instead of the highest revision on that branch. This shouldn't happen anymore, but be careful!

## 1.25 README.guide/Patch Level 5

Release Notes for HWGRCS Patch Level 5

\*\*\*\*\*

- I set `'__stack'` to 32768 to make SAS/C automatically swap stacks if there is not enough. This should fix stack problems for those who forgot to use the stack command.

NOTE: I had to compile everything with `'NOSTKCHK'` now because of a really nasty bug in the SAS



startup which I won't fix myself for certain reasons. So there is currently NO AUTOMATIC STACK CHECKING ANYMORE!

- There was a bug in 'ci' with '-u' on an unchanged working file. It used to append the last log entry again and trashed the working file this way.
- 'rcsdiff' now supports '-L' and ked hopefully won't be needed anymore. Forgot that ... :-((
- There is a bug in the internal SAS/C 6.3 '\_\_datecv' function. It uses the timezone to convert dates which messed up all dates with HWGRCS. I rewrote it and put it into 'amiga.c'.

NOTE: You might want to wait for 24 hours after applying this patch before using HWGRCS if you use options like '-d' or '-M'. This should get you out of any 'time warp'.

## 1.26 README.guide/Patch Level 6

Release Notes for HWGRCS Patch Level 6

\*\*\*\*\*

- The structure of the source tree has changed. So if you are updating from a previous HWGRCS you might want to consider doing a separate installation first. If it works, you have basically only the 'RCS:' assign to change. If it doesn't, you can mail me bug reports and keep on using e.g. HWGRCS Patch 5 until I can fix your problems! :-)
- Unfortunately the 'fix' for '-d' and '-M' that I did for patch 5 only obscured the problem further. It should be fixed now. Inside the RCS files you should see only UTC dates now.

NOTE: You might want to wait for 24 hours after applying this patch before using HWGRCS if you use options like '-d' or '-M'. This should get you out of any 'time warp'.

- HWGRCS now uses the POSIX compliant default value for the '-x' option which is '-x/,v'. This means that newly created RCS files will have no ',v' extension, and files with ',v' extension will be used after files with no extension. This also means for certain reasons that you should use a sub directory 'rcs' to put RCS files into. See below. Working setups should not be affected as 'RCS\_link' is obeyed as usual and ',v' files are found and used, too.
  - To support the POSIX compliant behaviour and some standard code management tools, setuprcs has changed. It now sets up a standard 'rcs' subdirectory instead of 'RCS.RCSFiles'. It should not disturb existing 'RCS\_link' setups.
-

- I killed `'_main.c'` and most of `'amiga.c'` in HWGRCS. All of the unix quirks and workarounds are now contained in `'hwgunix.lib'` in the `addlib` directory. It is a SAS/C 6.3 specific library which contains patches and addenda for unix source compatibility. If you need to know more about the contents or think you have found bugs, contact me.
- Note that with SAS/C 6.3 `'getcwd()'` seems to be incorrectly prototyped with a `'const char *'` as first argument. This was in `'conf.h.amiga'`, too. As this clashed with some other source here, I fixed the prototype to `'char *'` in `'stdio.h'` and `'dos.h'`. This might affect your recompile. I sure hope you added the missing `'const'` already in the SAS/C headers for some other functions. ;^)
- Rework of `'rcsfnms.c'`. Relative paths should work a little better now.
- Still no `'$VER$'` keyword for C= style version strings. Any input on a suitable grammar?

Issues:

- What version.revision should be used if it's a branch file like 2.3.4.1?
- Where does the filename come from if it hasn't been set already?
- How to avoid the final `'$'? '\0'` works only for C strings, not for e.g. `AmigaGuide@` or `AREXX`.
- What about special text after the version info (for `VERSION FULL`)?

Any input welcome.

- LP has changed, too. It checks CTRL-C a lot better and should finally handle/filter ANSI codes well, if asked to. It got smaller, too. ;^)
- Cleaned up the patches to the GNU source. They should all have a small description now telling about the `'Why?'`.
- Timezone handling has changed. If `'ENV:TZ'` is not set, `locale.library` is tried. If `'locale.library'` is available, an `'ENV:TZ'` is set up automatically if it doesn't exist. It won't be saved to `'ENVARC:'`, though. Once the timezone is set in a running command it won't be reevaluated. This saves quite a few unnecessary `'ENV:TZ'` checks.
- NEW UNIX FILENAME SUPPORT ON STARTUP.

All the HWGRCS commands (not `'LP'`) are compiled with a new startup that understands how to rework `'.'` and `'..'` path specs into AmigaOS paths automatically. Examples:

```
'.'          [An empty argv value]
```

```

    './bla'      'bla'
    '../test'    '/test'

```

In essence: Leading single dot specs '<nothing else>' or './' and any '..' that qualify as a complete path part are removed.

This conversion is not done for args quoted with ''. If there is really some conflict with existing SW, please tell me about it as soon as possible.

Unix makefiles should be easier to use now with HWGRCS.

No root slash conversion is done for two reasons:

1. A leading slash is ok with AmigaOS.
  2. Even if it wasn't which root should be used?
- In a separate package 'HWGdiff2\_6', you'll find the sources to the now included GNU 'diff' 2.6. I removed the old 1.15 'diff' distribution from HWGRCS. If you need (AmigaGuide®) docs for the 'diff', you'll need to get the 'HWGdiff2\_6' package.
  - Same thing goes for 'patch'. I included the executable from the 'HWGpatch2\_1' package.

## 1.27 README.guide/Patch Level 7

Release Notes for HWGRCS Patch Level 7

\*\*\*\*\*

- HWGRCS 'as is' will only compile under >= SAS/C 6.51 now. There are certain dependencies on SAS/C 6.51 inside the 'hwgunix.lib'.
- rcs clean without a file spec should work as documented now. It has never worked in HWGRCS, I think. :-)
- RCS bug fixed that caused workfile names to be messed up on e.g. a 'co rcs/#?' with a suffix default of ',v'.
- diff3 uses correct arguments for diff now.
- rcs freeze works through all the files in the rcs directory, not just the ones ending in ',v'. Files that are not a repository should be ignored with a message that 'rlog' failed.
- The '\$Source\$' keyword no longer gives an absolute filename with the colon after the drive volume name replaced by a slash.
- Brand new startup for SAS/C 6.51. Now response files are supported. When you specify '@file' as argument, the contents of the file will be read in as arguments at the position of the file

specifier. You could set up a file list this way if e.g. `#?` or `#?.c` doesn't do what you need. See  
     Response Files  
     , for more info.

- HWGRCS should be compatible to AS225 now. Don't know about the Oxxi Novell client yet. I worked my way around missing support for V37 packets.
- I deleted the translated man pages from the distribution. The same text is available as AmigaGuide@ document. The ?roff sources are still there for those who want to generate ASCII man pages.
- I converted the readme.first and release.notes contents to AmigaGuide@ format for easier reading. Actually, you are reading them now.
- I split HWGRCS into three parts. The binary archive 'HWGRCSp7f.lha' (with AmigaGuide@ docs), the source archive 'HWGRCSsrcp7f.lha', and the man page archive 'HWGRCSmanp7f.lha'. The man page archive contains all the unix man pages. I took them out of the binary archive because most people will use the hypertext docs, I think.
- NOTE: I am not so sure anymore that it was a good idea to rename the 'RCSINIT' environment variable to 'RCS/INIT'. I might return to the original 'RCSINIT' in a future release.
- NOTE: I might remove the check for 'ENV:RCS/DIRECTORY' in a future release or rename the environment variable. I don't think of this as a 'good' feature anymore.

## 1.28 README.guide/Patch Level 8

### Release Notes for HWGRCS Patch Level 8

\*\*\*\*\*

Patch 8 can be considered a 'major change release'. So please be careful! Make a backup of your sources, and keep a copy of HWGRCS Patch Level 7!

- 'ag' is now recognized as an additional AmigaGuide extension.  
     See  
         Amiga Comment Leaders  
         , for a complete list of file extensions  
     and their comment leaders.
- HWGRCS supports  
     RCS\_link  
     . This filename is now ignored as  
     filename in these HWGRCS commands: 'rcs', 'rcsclean', 'rcsdiff',  
     'rcsmerge', 'ci', 'co', and 'rlog'. The filename is simply removed  
     from the argument list before the command gets around to use it.

This should make e.g. a `co #?` much less of a problem.

- A `\*` as wildcard is now reworked into `#?` before processing. This should take care of unix filenames. Take a look at

Filename Expansion  
, for more information.

- Before `ENV:RCS/INIT` is checked, the commands look for `ENV:RCS/INIT\_<cmd>`. This variable is used instead if it exists. `<cmd>` stands for the name the program has been called with (i.e. `argv[0]` for C). See  
RCSINIT  
, for more info.

- Four new RCS keywords are now available for the Amiga with HWGRCS. `VER` for C= style version strings compatible to the `VERSION` command, `AmigaREV` to implement a smart revision counter, `AmigaDATE` to get a date suitable for a version string, and `AmigaProtection` to preserve the protection bits `S` and `P` even when the repository is e.g. on a Unix server. Note, that these keywords behave in a very special way due to how RCS works. I feel that they are still somewhat experimental. See

New RCS Keywords

, for more info. NOTE: YOU SHOULD READ ABOUT THIS FIRST if you have sources using the Commodore `VER` keyword in HWGRCS already. I HAVE WARNED YOU!

- The decision on what arguments to expand in the startup has been changed a little. Now everything that does not start with a `-` and has at least one of `?`, `\*`, `#`, `~`, `(`, `|`, and `)` will be expanded. See

Filename Expansion  
, for more info.

- RCS\_link  
, previously just a hack, is now a full fledged configuration file. It supports the notion of a

Reference Directory  
, and an extension to  
RCSINIT  
, which allows

directory specific options for all commands. There is also directory specific exclusion of filenames on the command line possible now. Please note that the `Reference Directory` feature is rather experimental.

- Local environment variables are checked now and as usual they have higher priority than variables in `ENV:`.
- Due to popular demand, I added the `diff` docs to the main archive. Information about `patch` is included, too. The archive size has gone up again. Ouch, my phone bill!

