# MultiVer1.2

| COLLABORATORS | | | |
|---|---|---|---|
| | *TITLE* :<br>MultiVer1.2 | | |
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | January 31, 2023 | |

| REVISION HISTORY | | | |
|---|---|---|---|
| NUMBER | DATE | DESCRIPTION | NAME |
| | | | |

# Contents

# Chapter 1

# MultiVer1.2

## 1.1 main

```
                       ****************************************************
                      ®MultiVer V1.2 (9/4/95)
                      ------------------------
                       By $LiNg$HoT/DiMeNTiA
                       ---------------------
```

```
                       ****************************************************
                       Try my web page on URL:
                 http://www.york.ac.uk/~dmh11/slingshot
       ****************************************************
```

## 1.2 What the hell is MultiVer?

Have you ever just got hold of a new utilities disk or package and found that it has a rather large libs: directory with loads of libraries you already have? When this happens it is usually a good idea to check to see whether any of the libraries are newer versions than ones you may have on your harddrive. In this case, you will have to go through each one and check the version $ from the shell using the version command. This in it's self takes time but then you have to copy the new version across to your harddrive, which is another waste of time.

This is the very process that MultiVer takes care of. All you have to do to achieve the above, is to give the CLI program the source directory (e.g libs:) and the destination directory of the files you wish to compare (e.g df0:libs/). My program will then scan both directories and if any files with the same name are discovered the version $'s will be compared. At this point, if the COPY switch is set, you will be asked whether to copy the newer version to the source, not bother or opt to copy this AND ALL other newer versions without further prompting. If the files being compared do NOT have version strings, a check will be made on their file sizes. If the second file is bigger than the first, you will have the same options as if the second file had a newer version $. Where the COPY switch is NOT set, a message will be displayed. A message is also displayed if the source file is actually bigger than the second file being compared as some newer versions of programs/files may have a smaller file size (optimisations etc...). In this case, you will NOT be asked whether you wish to copy this file, as in most cases - smaller file size means older version!

MultiVer also has a LOG switch that will make a log (in RAM:) of the files compared and there version $'s. This is useful if the directories being compared are very big.

As of version 1.2a of GuideCompiler, a new
global preferences
system
is implemented that allows configuration of some of the ways in which the program functions. One of these allows ANSI output to be turned off for if you use a shell that cant handle them, or you just dont like the look.

Simple huh? It makes the tedious process of updating files to newer versions alot quicker and simpler. Give it a go, I dont think you'll be dissapointed! :)

## 1.3 So, what is my legal status MR Biffo?

MultiVer is Public Domain and thus is free to be used by ↩
anyone who
has a use for it.

You use MultiVer at you own risk and I cannot be held reponsible for any damage caused by use or misuse of the program. Not that I'm expecting any problems to arise, but it's better to be safe than sorry! :)

MultiVer has been written in AREXX but has been compiled to protect the code ( not that it's amazing or anything but.. ). This not only protects my routines but means that it is easier to run and the user does not have to mess about with the RX command or script bits!

I may at some time or another decide to release the source depending on how I feel about it :)

GuideCompiler is written by: James Allen A.K.A SlingShot/Dimentia U.K

Feel free to
contact
the author at any time, to discuss extra functionality
for MultiVer or just to talk about AREXX!

## 1.4 What ya need..

```
                MultiVer 1.2 needs:
-------------------

OS2.0+
RexxMast ( The Arexx interpreter running )
A Harddrive
The following AmigaDOS commands in C: ( sys:c/ )
- SORT
- LIST
- DELETE
- VERSION
```

Why use AmigaDOS commands?

## 1.5 Why have I used the DOS commands?

The following description contains technical terms which some ↩ people will
not understand.

When I first learnt AREXX, I didnt know there was a command in the AREXX support library that allowed directories to be scanned and the files returned to a variable. Therefore I used my own technique of redirecting the output of the list command to RAM: and sorting it.

I now know how to use the inbuilt support function ShowDir() but I have decided against using it in this instance. The problem is, due to AREXX being an interpreted language it aint the fastest thing around. Therefore, just like a C coder might use the AmigaDOS functions in dos.library for extra speed, I have decided to use the AmigaDOS commands in C:. This system works faster than it would if I used ShowDir() because using that would involve:

1. Reading the directory into a single simple symbol. ( A Variable )
2. Spliting it up into a set of compound symbols. ( An array )
3. Sorting it using a quick sort.

Even on my 030/40Mhz this all takes alot of time in AREXX and so by using AmigaDOS commands the running of the program is improved. I make LIST and SORT resident and so they donnot need to be repeatedly loaded off disk. This of course takes alittle bit of memory but not much.

Using my current technique I redirect the output of list to ram creating 2 files holding the files and the directories. I then read these into compound symbols. This is alot faster than using the above method, but I suppose reading them into an array takes alittle bit of time. Still, I am happy with the speed my program runs at, and dont think it could be speeded up much more. If it doesnt run very fast on your system,
mail me
your system setup as I am interested on how it runs on slower machines.

I have written a number of procedures which use the above techniques and these allow me to write other AREXX programs that need to scan directories quickly and easily.

In this program I use the AmigaDOS command Version to grab the version $. I use this as it makes sure that my program gets hold of the most complete value possible and as such means that MultiVer is pretty reliable.

## 1.6 So.....How do I use it then?

Before running MultiVer, the AREXX interpreter RexxMast ↩
MUST be
running. If this is not run from your Startup-sequence or User-Startup, go to the CLI and type:

```
sys:system/rexxmast >nil:
```

IF you want it to be run every time, place the above line in your startup-sequence OR user-startup. You could also merely drag the RexxMast file from sys:system into your sys:WBStartUp drawer and this would have the same effect.

Usage is pretty straight forward. Here is the standard AmigaDOS template:

```
MultiVer DIRECTORY1/A,DIRECTORY2/A,COPY/S,LOG/S
```

```
DIRECTORY1 = This is the Source directory of the files to compare.
DIRECTORY2 = This is the Destination directory of the files to compare to
             directory1.
COPY       = This switch turns on the copy mechanism.
LOG        = Makes a log in RAM: called - "log"
OS3.1      = (OS3.1 ONLY) Will make the version command ALWAYS read from file.
```

Note: The OS3.1 option can only be used if you have the OS3.1 version command installed. It makes the version command read from file even if the file is already in memory. This makes MultiVer return the correct version string if the file being compared is loaded in memory (Libraries etc..). There is a bug with the version command supplied with OS3.0 and OS2. See the
bugs
section.

Note: Files are copied from directory2 - directory 1. Make sure you specify the correct source directory so that the files are copied to the correct place.

Args can be placed in ANY order as long as Dir1 is before Dir2! E.G:

```
1> MultiVer log dh1:temp copy dh2:temp2
```

In the above, Dir1 = dh1:temp and Dir2 = dh2:temp2.

When using the copy option, you will be prompted when a newer version of a file has been located in Directory2. The prompt will look something like this:

```
version.library           34.1    40.42
Do you wish to Copy to Dir1 (Y/N/A)?
```

Selecting Y will copy the file to the source directory, overwriting the old version.
Selecting N will merely continue the scan.
Selecting A (ALL) will result in the newer version being copied to the source directory and will mean that any newer versions found will be AUTOMATICALLY copied to the source, overwriting the original.

If NO $VER's are found a size check is made. For bigger files in the second directory the same applies to the case when a bigger $VER is found. Therefore, you will be asked if you want to copy the file IF the copy switch is set. Otherwise a message is displayed on screen and the optional log file.

IF the source file size is BIGGER then a message is displayed on the screen and log. Even if the COPY option is set, you will NOT be asked whether you wish to copy the file. In this case, it's best to check the file out by hand and make an educated guess as to how new it is compared to the second.

Here is an example that I use to compare files in libs: with those on disc:

```
1> MultiVer libs: df0:libs/ copy log
```

## 1.7 There arent many but it does the job I guess!

V1.2 Features:

- Scans 2 directories and compares version $'s of files.
- Allows user to copy newer version directly into the source.
- Makes a log in RAM: if LOG switch is specified.
- Supports SlingShot
  global preferences
  for customising operation.
- Checks for filesize difference if NO $VER found.
- Displays message if Source is bigger than destination. NEW files could be smaller!
- Alleviates the boredom of updating files.. :)
- Gives an option for OS3.1 users to allow files to be read from disk NOT

```
  from memory!
```

## 1.8 Arhhhhh no not them....!

```
KNOWN BUGS in V1.2
------------------
- Some programs contain incorrect version $'s

     This  is  a  mistake  that some coders make.  They specify the version
string  in  the  incorrect  format and this means that the AmigaDOS version
command returns an incorrect value to MultiVer.  Therefore, the program may
think  an  older  version of a program is newer or whatever, and this could
mean that you copy a older version over a newer one!

Possible Solution:

     Errr,  not much really.  Just make sure that you look carefully at the
string  listed  by  MultiVer  and  do NOT copy the file over if the $ looks
strange.

-  The version command which comes with OS3.0/OS3.1/OS2 reads the $VER
from memory.

     This  is  a  highly  annoying problem with the older version commands.
The  problem  is that these versions of the command will read the $VER from
memory  if  the  file  being  compared  is currently in memory.  Therefore,
multiver  will  return  identical  $VER's  for  some  files - libraries for
example, if you are running OS2/OS3.0/3.1

 Possible Solution:

     In  the  case  of OS3.1, you can use the OS3.1 switch to make multiver
call  the  version  command  with  the FILE switch.  This makes the version
command actually read the file specified and returns the correct $VER.  For
some  strange  reason,  the  file switch for the version command in OS2/3.0
does  not  work  and  just returns the name of the file specified!!!!  This
means  that  those  people  running the older version command, will get the
incorrect  version  string  in  some  cases.  I cant think of anyway around
this,  as thats just the stupid way that the older version commands work :(
It wont happen much with normal programs, but libraries which are currently
being used will make multiver return the version of the loaded file and not
the one on disk...The only thing you can do really, is load the file in
Directory2 into a file editor and manually search for the $VER - Sorry.
```

## 1.9 What has been changed and added..

```
                MultiVer Version History
------------------------


V1.0 - 16/9/94 (Released: 6/03/95)
```

```
 - A program to scan 2 directories and compare the version of each file.
 - If the COPY switch is set, will prompt the user to copy newer files
 - from the destination directory to the source.
 - The LOG switch will make a log of the versions and whether the file
 - was copied in Ram:Log.

V1.1 - 17/3/95 (Released: 20/3/95)

 - Fixed bug that meant library $VER's would be read from memory.
   (Spotted by: Venom/Nerve Axis)
 - Extended filename output to 30 spaces instead of 25.
   (Suggested by: Frederic Steinfels)
 - Dramatically speeded up scanning operation (more than 2 times!)
 - Implemented
                global preferences
                 idea to customise operation.
 - Global preference option - NOANSI implemented.
 - Size check made if NO $VER found. (Both suggested by: Frederic Steinfels)
 - Checks whether destination is bigger OR whether source is bigger. Only prompts
   for copying when the destination is bigger. Some NEWER progs are smaller!
 - Improved CLI parsing routine. Now supports AmigaDOS "?" anywhere on line.
 - Args can be placed in any order as long as Dir1 is before Dir2 :)

V1.2 - 9/4/95 (Released: 9/4/95)

 - Found a bloody annoying bug in 1.1. The program didnt work with OS2 &
   OS3.0!!! No one reported the fact (I dont know!!!). The problem lied with
   the FILE switch I started using with the 'version' command. Only the
   OS3.1 version of the command works correctly with it! Basically it makes
   the command read from file and not from memory if the file is currently
   loaded. Anyway, the other versions dont work properly and so alot of
   garbage was returned when running MultiVer on anything lower than 3.1!
 - Added OS3.1 switch to make the program use the FILE switch with version.
```

## 1.10 A new system designed to make life easier..

As of version 1.1 of multiver I have implemented an expandable system of global preferences. Basically this means that all of my future programs and updates will conform to a number of preferences set by the user.

These preferences will tend to be environmental ones, and affect the way in which my programs output text etc...

At the moment, only the following option is supported in MultiVer:

o NOANSI - Turns OFF ansi output.

This was requested by a user and so I decided that the best way to do this was to implement it globally. Therefore, once set it would affect all of my programs that support the global preferences idea and allows the user greater control over my programs.

The global preference file is stored in ENV: and is called slingshot. It consists of a number of keywords placed on each line. If you don't know anything about the environmental variable system used on the Amiga (and

unix ;> ) then
click here
for some help.

Therefore to turn off ANSI output in all of my programs which support global preferences, place NOANSI on a line in env:slingshot and envarc:slingshot (if you want it to be permanent).

If anyone has any environmental options they would like to see implemented in my global prefs system, please
mail them to me
, as I can’t
really think of any new options that would be useful at the moment.

## 1.11 Help! Help! I’m being repressed....

Right then...If you dont know much, even anything about ↩
Environmental
variables dont worry there not very hard to understand.

When you boot up your computer, the startup-sequence does a number of different tasks. One of which is to set up an area on the Ram Disk called ENV. It then copies a number of files across from the harddisk which contain information used by different programs. This information could be anything from the size of a window a program uses, to some messages you have entered into a notebook program.

Anyway, this is where our variable containing the SlingShot global preferences is stored.

To set an environmental variable you go to the CLI and type something like this:

```
1> setenv slingshot NOANSI
```

This file is now stored in RAM:ENV/ as slingshot. It contains on the first line the parameter NOANSI.

The above system is OK for single options (setenv destroys any currently defined variables!), but you must be aware that if you want to add more options (as they become implemented ;> ) then you will need to edit the file using a text editor. See the info about using the AmigaDOS editor ED below.

Another thing you must also realise is that once the computer has been reset this variable will be wiped from memory. Therefore, if you want to keep the global preferences you will need to have them stored actually on the harddrive. All the permanent ENV files are stored on your harddrive in:

- sys:prefs/env-archive

(sys: refers to the partition from which you booted)

This path is assigned to the logical device called - ENVARC:
The easiest way to create a permanent env varaible for GuideCompiler is to echo the viewer name straight down ENVARC:

Using the above example we could create a permanent GuideCompiler variable using the following CLI command.

```
1> echo NOANSI >envarc:slingshot
```

This will write the variable to the harddrive and each time the computer is reset it will be copied to RAM:ENV/ along with the other permanent variables.

You could achieve the same result by editing the file using a text editor such as ed. This is the way to set the prefs if you want to set MORE than 1 parameter.

E.G:

```
1> ed envarc:slingshot
```

Using ED, you would type in each option and use RETURN to go to the next line. When you have finished you could go to the menu and select save, or press ESCAPE and type sa <RETURN>.

NOTE: If you have any trouble using the global prefs system with MultiVer, feel free to

mail me

for help!

## 1.12 What next then?

- Copy files that arent in the source?

I was wondering whether to include an option that will copy all files that arent in the source. This would be especially useful in the case of libs: comparisons, as it's always useful to have as many libraries on your harddrive as possible.

So...If anyone would like to see this option implemented..

Mail me!

(Havent had any for Version 1.0, so if you want it - MAIL ME!!!)

- Even better non-$VER checks on files

Although MultiVer does quite a few checks on files it finds, there is room for even more. I've just got to work them out.! :)

## 1.13 Hi mum!

Greets go out to the following people:

Laurie Lee ( The C God ) - Hey, what are you doing on your Amiga? ;>
Dave Hollway - Money and map NOW!!! :)
Mark & Scott/Visage Computers - What are YOU doing with one of my programs?
Catlord - When ya gonna code a demo then.... *;)
Draxx/Desire - Built any CrayAmigas lately?
Darren Bloor - Psst....Got any......dirty videos... :)
Creator/Dimentia - Get that Babylon 5 beater finished! ;)
KenD/LSD - See yers on mono m8.
CounterPoint/Westower - Hey, keep those cool mods coming..
Chalky - Was it 2 disks or 3 you gave me? ;>
WhiteWolf - Lets get the Amiga Society into action..!
WolfLord - If your reading this.it must be on my home page..Get yours done!:>
The_Don - SO...When are we gonna get our OWN section then? ;>
Jason Compton - Keep producing your great mag!
Deztech - Right then. Heres mine..Now get some of your own done m8.. :)

----
IRC:

```
FUNdament - Hey, get that paint package done..I want to do the AREXX :)
Secka     - Wonder if we'll get banned again... ;>
Wangi     - Your mail address is dead...!!
Xanim     - Hey Terje..so when are you gonna optimise xanim then.. ;>
```

Oh and greets to all on #amiga..

AND..

All the programmers on the world wide Amiga scene, who make our machine such a joy to use.

While I'm at it, greets to ALL Amiga owners everywhere. Keep the dream alive!

## 1.14 So you want to talk to me huh? :)

Well, I hope you find MultiVer useful and if you have ANY suggestions or comments please MAIL ME..I didnt recieve any for DeArchive.. :(
Oh and thanks to all the un-lazy people who mailed me useful comments and suggestions.

Right then, here are my current E-Mail addresses. They remain valid until:

Either:

A). July/September 95' when I finish year 2 of my degree.
B). The operators de-user me for.....Errmmmm well, I dont think we need to go into that right now *8-}}

Mail me - James Allen - at either:

- cc022jwa01@ntu.ac.uk OR

- sys2jal@doc.ntu.ac.uk

You can also contact me on the "MonoChrome" Internet BBS run from City University London, England.

```
Telnet : electron.mono.ac.uk
PAD    : uk.ac.city.mono
         ( When prompted, enter the above TELNET address. )
```

My account name is : SlingShot ( Also my nick on IRC! )

I look forward to hearing from you.....

Oh and check out my WWW home page some time (Thanks Threapwood!). It has links to my programs on aminet, upcoming projects and Amiga links etc..

http://www.york.ac.uk/~dmh11/slingshot

----------------------------------------------------------------------------

Look out for my other releases ( I finally got my AREXX compiler! ) such as:

```
GuideCompiler - Scans recursively and builds an ®AmigaGuide of text files.
                (Aminet: text/hyper/GuideCompil12a.lha) Includes:
NameStrip     - "        "      " and strips file names of spaces etc..
DeliList      - "        "      " and builds up a ®DeliTracker2 playlist.
                (Aminet: mus/misc/DeliList11.lha)
DeArchive     - Unpacks all LHA/LZH/ZIP files into a directory automatically
                (Aminet: util/arc/DeArchive15 - Most recent)
```

Oh and my one released music mod is on aminet in mods/pro and is called:

WhoIsElvis94.lha

It’s a remix of the techno track, "Who is Elvis" By Interaction.

I have also done a REKO Klondike AGA cardset. The theme is space and it is a 256 colour set. It’s on Aminet as:

game/think/SpaceREKO.lha