DATEL Electronics LIMITED

Amiga ACTION REPLAY MK III

INSTRUCTION MANUAL

by Olaf Boehm and Joerg Zanger

documentation by Wayne.H.Beckett

CONTENTS

1 INTRODUCTION

CONGRATULATIONS on your purchase of The Amiga Action Replay which we think, and you will soon find, is the most powerful utility available for the Amiga. With a whole host of graphics, sound and programming features it is possible to get 100% more from your computer. Before you start we strongly recommend that you read the manual, even though the temptation is to plug in and go, as reading what one command does will help you understand others. Until you are familiar with some of the more complex coding instructions like CODE, don't use valuable disks. If you do have problems with a command, please do not dwell on it too long, come back to it later and it may become clearer. A few of the sections such as that on the monitor commands and the System information are rather complex for the beginner so don't expect to understand them straight away. Remember that the Amiga hardware can cover many manuals, so if you really want to learn buy, beg or borrow a few of these and soon with the help of your Action Replay you will become an expert on the Amiga. Good luck.

2 INSTALLATION

First and foremost you must *never* plug in or unplug your Action Replay from your machine while it is switched on! The installation of the two versions of Action Replay is slightly different so please read the relevant section following.

AMIGA 500

First switch your machine off. Next on the left side of the machine by the keyboard you will find a removable panel which hides the expansion port - it will come off but may be a little stiff. Now, with the button and switch facing up, insert the Action Replay firmly into this port; it should now be quite solid. Now switch your computer on. The green light on the Action Replay will be on and the red light off. If the red light is on then the slow-mo is active; throw the switch to disable it and the machine should boot as normal. If the machine does not boot properly switch the machine off and repeat the installation procedure. When the Kickstart screen appears, press the red freezer button and a blue screen will appear - this is the screen from which you use the following commands. Before you continue you should check the country code of your machine by pressing the F9 key. It will toggle between US and German - there are slight differences between the two (e.g. X and Y are reversed). Once the keymap has been set for your machine you need not set it again till the power is switched off. To restart press X (return).

AMIGA 1500/2000

For owners of the Amiga 1500 or 2000 the installation is slightly different. You must first switch the machine off and take the case off the machine. The slot which the Action Replay is inserted in is the one slightly to the left of the disk drives looking from the front (it is also slightly to the left of the 68000 chip). The connector which leads to the remote points to the rear of the machine. Slot the card in firmly and trail the remote through an appropriate gap in the case of the machine and keep it to hand. Put the case back on the machine and switch the power on. The red light on the remote should be off. If it is on then the slow-mo is enabled; throw the switch to disable it. When the Kickstart screen appears press the red button and a blue screen will appear. If this does not happen repeat the installation procedure. Before you proceed check that the country code is set correctly by pressing the F9 key a couple of times. If it is set incorrectly the X and Y keys will be reversed. Once the keymap is set it need not be changed until the machine is switched off.

A590

For owners of the A590 hard drive, when you see the blue screen you should press the F3 key and clear the addmem option and set the autoconfig option. Restart the Amiga and reboot the machine. The hard drive will boot.

3 GETTING STARTED

There are several simple features available on the Action Replay which you will find very useful.

(HELP)-	Probably the most used key of all, gives you a brief description of all commands.
(SHIFT)-	No scroll / Pause
(TAB) -	Insert spaces
(ESC) -	Will abort most commands
(F1) -	Clear screen and Home cursor
(F2) -	Home cursor without clearing screen
(F3) -	Preference screens
(F5) -	Print screen to printer if attached
(F6) -	Switch printer dump on/off
(F7) -	Switch between overwrite/insert mode
(F8) -	Show instructions for mempeeker
(F9) -	Toggle between US and German keyboards
(F10) -	Switch to second screen (switch back)

When the power light is dim in Action Replay mode the computer is waiting for an instruction. When it is flashing Action Replay is working on a command.

The slow-mo can be activated at any time by flicking the switch so the red light is on. Adjusting the potentiometer will vary the speed. It is obviously not a good idea to keep the slow-mo on during disk access and may cause loading problems.

There is one of the above function keys which requires a more detailed description

3 GETTING STARTED

and this is the (F3) key. This will display a preference screen which may be exited at any time by pressing the (ESC) key.

The options are selected by using the mouse and left mouse button.

MEMORY CONTROL

The display to the top left of the screen shows the memory available for use by the Amiga. This can be changed by clicking on the appropriate squares which will select/ disable memory as appropriate. By clicking on the values beneath External Memory, the area of memory can be altered.

MODULE INTERNA

This set of options is to do with the machine resets, and set up. The No res option removes the Amiga Action Replay reset screen. The test1 and test2 options are apparently different types of system reconfiguration when using the X command to restart the machine after freezing, in some cases the machine can lock up. If this happens you should try setting the test1 and/or the test2 options. The blanker option is a screen saving facility, when switched on, if no keys are pressed on the freeze screen for a while, then it is blanked. To get it back press any key (shift is probably best as no keys are typed).

The bottom left section is used to select the colours that the Action Replay screen is displayed in.

MEGASTICK

The megastick option is for the joystick translation codes which can be entered using the megastick command.

The two meters on the right of the screen are used to allocate an auto fire rate for the two joysticks. They can be set totally independently of one another so a player can be handicapped.

AUTOCONFIG

The autoconfig option allows Action Replay to detect what it thinks the computer is, for example when on it will detect and boot an A590 hard drive.

3 GETTING STARTED

There is now a second preference screen available by clicking on the next page box.

BOOTSELECTOR

The boot selector in the top left is used so you can get the machine to boot from any drive. You may select any drive available or variable so that the computer will search each drive for a bootable disk. With this option there are two things you should bear in mind. The first is that when you have selected this option and exited you must reboot the machine before it will work even if you are on the kickstart screen. The second thing is that the success of this option depends on the disk you are booting from, i.e. if it starts to boot from disk df1: and the program tells the computer to go and read from df0: the boot will fail, so please bear this in mind. The easiest way to try out the variable boot is to use an Action Replay disk that has a bootblock on it (using the install command).

DISKCODER

The disk coding section is now available from the second preference screen. Each option can either be enabled or disabled by clicking on the appropriate icon. For more information on these features please see the section on disk based instructions.

DRIVECONTROL

Drives can simply be turned off or on by clicking on the appropriate boxes.

VIRUS TEST

Using these options you can switch off the automatic virus detection. This is useful when you are using disks that have built in virus protection, e.g. Sentinel, which contain parts of the viruses so they can be identified. This code is then interpreted by Action Replay as a virus itself. Kill is useful for similar reasons. You would normally leave it active so a virus that is found is killed. Virus boot will check the bootblock of your disk as well as checking for a virus in memory. This option shouid be turned off if you are using a hard drive.

3 GETTING STARTED

BURST NIBBLER

The Burst Nibbler faststart option allows entry to the nibbler screen on a reset by pressing the left mouse button and holding it down.

SAVE + LOAD

There is now an option to save and load your Action Replay preferences to a floppy disk; simply click on the save/load option on the second screen and select an appropriate path/filename.

SETMAP D

The setmap D function is for German users and sets the keymap to their German standard (e.g. Z and Y are switched)

SAFE DISK

There are two options here. Resident cures the ROM bugs present in the Amiga which can cause disk failure and the No Click option prevents the floppy disk drive from clicking. For further info see the Safedisk command.

In the following text the actual commands available to Action Replay are described. There follows one or two pointers to aid your understanding of the commands.

Firstly, the number systems that can be used by Action Replay are decimal, hexadecimal and binary. It is important to note that the default base is *NOT* decimal but hexadecimal, so if you type the number 10 the Action Replay will think you mean 16 decimal. You can easily enter decimal by preceding the number with a (!) i.e. !10 would be 10 decimal. The prefix for binary is %. If you can remember this fact early on it will save you some headaches.

Secondly, the commands themselves are on an underlined line. Any part of the command that needs to be typed in exactly is outside brackets whereas all that inside brackets is not, e.g. the command to format looks like this.

3 GETTING STARTED

FORMAT (name)

means type FORMAT but follow it with a name of the disk like DISK1 or any valid disk name. When (path) is specified you can specify 0: for drive df0: and drive df1: etc. You can also specify a directory structure, for example the command delete is headed as follows

DELETE (path)(filename)

means type the letters FORMAT followed by a drive number 0: or 1: If no drive number is specified the current drive is accessed. A directory path can be specified if required (if you do not know what directories are you need not bother about this, for further details consult an Amiga DOS manual). This is then followed by the name of the file you wish to delete, e.g. a typical example of the above command would be as follows

DELETE 0:MAIN/SUB/WAYNE

where 0:MAIN/SUB is the path and WAYNE is the filename.
Finally all commands new to MKIII are marked with an Asterisk.

4 QUICKSTART

There can be no real equal to a good read of the manual but this guide is to point you in the right direction as to which commands are relevant to a particular procedure.

Before you start, check your installation and make sure the freeze option is working. Secondly press the F3 key and follow the “before you start” section and ensure that all options are set as you wish. If you are to be using the deep trainer or freezer features (SA etc.) you should switch out all possible spare memory, i.e. 512K chipmem only. If you wish you should save your options to a formatted floppy disk (see FORMAT). EXit back to the main kickstart screen and re-boot so that all your preferences are set to your liking. Insert a disk with a public domain or demo program and allow it to boot. If it is a playable game you may proceed to the trainer and deep trainer sections to start trying to find infinite lives. If it is public domain or your own software you may like to freeze out a new copy to a disk. First you must have a blank formatted disk to hand (if not you may like to format one using the FORMAT or FORMATV command). Then simply type the line

SA TEST

and a copy will be saved to your formatted disk. To reload this you could use the LA or LAR command right now, but if you wish to be able to reload the program at a later date without the need of Action Replay you should type the lines

SLOADER
INSTALL 1

and then reboot the machine. A DOS window will then appear ready for you to type a command. To start the frozen program simply type the line

ALOAD TEST

and your frozen program will reload to the exact spot you froze it at. For more information on these commands please see the freezer section.

5 DISK BASED INSTRUCTIONS

Note The following disk instructions act unless specified on the currently active drive. The format is in general very similar to that of the CLI in Amiga workbench, so a knowledge of this would be very useful. The concepts of directories and sub directories is somewhat complex so is not dealt with here in any detail. If you do not understand the fundamentals of directory structures then I refer you to numerous Amiga DOS publications that would explain these in more detail.

FORMAT (name)

This instruction will format a disk in the currently active drive to standard Amiga DOS format, with the disk titled (name). For example, type the line

FORMAT DEMO

The computer will respond with the line,

READY TO FORMAT DISK IN DRIVE DF0:

to which you should reply (if you are happy) **Y** return. The computer will then format a disk in drive DF0 and call it DEMO.

FORMATV (name)
Format and Verify Disk

This is essentially the same as FORMAT in that it will format a disk in the active drive and title it (name), the difference being that it will then step through the tracks of the disk again verifying that there are no errors on the disk.

FORMATQ (name)
Format a Disk Quickly

This will effectively reformat a previously formatted disk and hence all files will be erased yet tracks will not be reformatted so it does it *very* fast.

5 DISK BASED INSTRUCTIONS

DISKWIPE (drive)
Wipe Disk Clean of Data

Destroys data on the drive specified. The disk will then be of no use until it is completely reformatted using either FORMAT or FORMATV e.g.

DISKWIPE 0

will *destroy all data* on the disk in drive DF0

DISKCHECK (drive)
Check Disk in Drive

This instruction will scan all the tracks on a disk in the specified drive for errors; any none Amiga DOS tracks will be reported as errors.

DCOPY (source drive) (destination drive)
Diskcopy

This instruction will copy an Amiga DOS disk from the source drive to the destination drive. The source and the destination may be the same but the computer will warn you that any program in memory will be destroyed.
Due to the nature of Amiga drives we recommend that you use blank unformatted disks to copy onto. If you do not, no damage will occur but sometimes the copy may fail. e.g.

DCOPY 0 1

COPY (path)(source),(dest)
Copy a File

This will copy a file from the (source) to the (dest) leaving the original program intact. This is essentially the same as the CLI instruction.

CD (path)

This instruction on its own will display the current directory path or tree. If you specify

a path in the standard CLI format the directory will be changed to that specified by (path) e.g.

CD QWERTY/SUBDIR

will change the current directory to Sub-directory SUBDIR in directory QWERTY.

CD /

will return you to the previous directory.

DIR (path)

This instruction will give a list of all the files and sub-directories in the current directory if no path is specified. A path however may be specified to show the contents of a particular directory e.g.

DIR QWERTY/TWO

will list the contents of sub-directory two in directory qwerty regardless of the current directory.

DIRA (path)

This instruction is similar to DIR except it will list the contents of all sub-directories as well.

MAKEDIR (path)

This instruction will create a subdirectory at the point specified by (path). If no path is specified a new directory will be created in the current directory, e.g.

MAKEDIR SUB1

will create the sub directory SUB1 in the current directory

5 DISK BASED INSTRUCTIONS

MAKEDIR MAIN/SECOND/SUB1

will create the sub-directory SUB1 in sub-directory SECOND in the main directory MAIN.

INSTALL (bootblock number)

This is used to create an auto booting disk which enters a Dos shell. From this shell you may run programs which are on the current floppy disk especially the ALOAD program, which is used to load Action Replay freeze files independently of the cartridge. There are currently two forms of bootblock number which correspond to the values 0 and 1. The first is a standard bootblock, the second is an anti virus bootblock which can detect many forms of virus before they have chance to damage your valuable data. For example, to install an anti virus bootblock on the current disk in the active drive type the command

INSTALL 1

BOOTPROT (codenumber)

This instruction will protect the bootblock of the disk in the active drive with a unique 8 digit number and make it totally unbootable to normal users who do not possess the code. To boot a protected disk you must use the bootcode instruction. It is not recommended that you attempt to write a protection number to a disk more than once, e.g.

BOOTPROT 1234	{typed in }
SURE TO PROTECT DRIVE IN DF0:	{the reply }
Y	{typed in }

will code your disk with the unique number 1234 which should be recorded or memorized.

5 DISK BASED INSTRUCTIONS

BOOTCODE (codenumber)

This is the only way to boot a disk that has been protected using the BOOTPROT instruction as above. If you wish to boot a protected disk, first press the Action Replay freeze button, then type BOOTCODE followed by your code number, e.g. to boot the disk that was encoded in the above example use the following

BOOTCODE 1234

Then restart the machine by typing X to exit from the Action Replay, reboot the machine and the disk will start. Note that once a bootcode has been entered it will remain until the machine has been either switched off or a new bootcode has been entered. To show the current bootcode enter the line

BOOTCODE

with no number. To get rid of the bootcode use the value 0. The bootcode value can also be viewed from the F3 preference screen.

DELETE (path) (filename)

This is a simple delete instruction. The file specified will be removed from the current directory if no path is specified e.g.

DELETE WAYNE

will remove the file named wayne from the current directory. If, however, the path is specified the current directory is ignored e.g.

DELETE MAIN/SUB/WAYNE

will remove wayne from the sub-directory SUB which is in the main directory MAIN.

TYPE (path)(filename)

This will type the contents of a file in ASCII to the screen. An executable file may

appear as meaningless characters whereas a file from a word-processor may have extra characters which are control codes for text formatting. The path and filename are as the DELETE instruction.

CODE (drive) (code number)

This is another rather complex encryption tool that to get fully to grips with will take some experimentation before you will fully understand it. (drive) is the drive number 0-4 and (code number) is a value in the range 0-!65535. The easiest way to show you how to use this feature is by example.

CODE 0 3

will have the effect of en-coding all disk writes to drive 0 so that they can only be read in future by setting the drive code to 3. All reads from this drive will also be de-coded using this number so normal files will be read as corrupt data. For example the following

CODE 0	{remove en-cryption from drive 0}
SA TEST	{save a frozen file to disk}
CODE 0 3	{en-code disk 0 with number 3}
LA TEST	{re-load the frozen file)

will load a corrupt file as a different code has been used to load (3) as that used to save (none). The instruction

CODE

will display all the en-coded drives and an encryption number. Note that this number will not be the same as that entered, and whether the drive is protected or not e.g. the status after the previous command would be as follows.

ACTION REPLAY DISK CODER V1.1

DRIVE0 CODE:00000005 PROTECTED!
DRIVE1 CODE:00000000 NORMAL

5 DISK BASED INSTRUCTIONS

DRIVE2 CODE:00000000 NORMAL
DRIVE3 CODE:00000000 NORMAL
DRIVE4 CODE:00000000 NORMAL

The code of 5 will always correspond to the entered number 3! This is the same for all codes. PLEASE use this feature with care, we would hate you to save your most precious files and forget the code number! The Diskcode values can also be viewed from F3 preference screen.

CODECOPY (source) (destination)

This is used in conjunction with the CODE instruction. It will de-code all data from the source drive using the code number then en-code all the data going to the destination drive using its number. You may use this feature for encoding or de-coding entire disks.

RELABEL (name)

This command will change the name of the disk in the current drive to a new name.

RENAME (path)(oldname), (newname)

Will change the name of the file specified by (path) and (oldname) to (newname).

SAFEDISK (n/b/s/u/v/g/a) *

Trackdisk Functions
- Noclick (n)
- Bugfix (b)
- Read damaged tracks (s)
- Verify writes (v)
- Update Tracks (u)
- All above (a)
- Quit Trackdisk (q)

This command depends on the argument supplied. An argument of B will instruct the trackdisk feature to remove the Amiga Dos bugs which can cause files to be damaged and/or lost. The option N will stop the annoying clicking of drives that the Amiga does

every few seconds to test if there is a disk in the drive. The S option will, when there is a read error, attempt to load the track anyway and continue. The U option will, if there is no access to the disk for a short while, reset the trackbuffer and switch off the drive motor. An argument of V will verify all writes to the disk. An argument of A will switch all the above options on; Q will switch them all off.

BURST *

Activate Burst Nibbler

This is one of the major new introductions with MKIII Action Replay. It is an advanced fast disk copier similar to the DCOPY command but more powerful. You can enter the Nibbler section by either typing BURST from the freeze screen, or while doing a reset hold the left mouse button down (there is no returning from the nibbler, so make sure you have nothing important in memory). The advantage of the Nibbler is that you can copy a disk to multiple outputs, i.e. 0,1,2,3 and it can also be used for duplicating MSDOS and Atari disks.

The features of the Nibbler are not too difficult to understand. On the left hand side of the screen there is a mode box. This is to toggle between Amiga Dos disks and DEEP which is for other formats (i.e. Atari/MSDOS). The start and end values are the track numbers to be copied, for example if you are duplicating public domain disks with only the first 40 tracks used there is no point copying the higher tracks. The side selector is for the upper, lower or both sides of the disk - of most use if you are duplicating single sided Atari disks. The Sync option should only be set to either the default value for Amiga disks or index for MSDOS. Only use other values if you understand disc sync marks. The 4 images of disks each represent a drive number (only appropriate if they are present of course). Each disk is a colour representing what action is to be taken on that drive. Blue is not acted upon, green defines the source drive, brown defines a destination drive without verify and purple represents a destination drive with verify. The start button begins duplication (assuming both start and destination drives are selected). To exit you must reboot the Amiga. Quit will cause a system crash.

6 DISK MONITOR COMMANDS

RT (strack) (num) (dest)
Read tracks from active drive.

Will read tracks starting from track number STRACK and a total of NUM half tracks into the memory address pointed to by DEST. If DEST is not defined, Action Replay will attempt to assign an area of memory for the tracks. If you do not assign a memory area for the tracks you can use the monitor commands to reference the tracks - see the monitor section for more details. Note the number of tracks are half tracks, i.e.Track 0 side 0 counts as one half track and Track 0 side 1 counts as another! e.g. to read 20 half tracks (10 whole tracks)

RT 0 !20

WT (strack) (num) (source)
Write tracks to active drive.

Will write a total of NUM half tracks starting from the point in memory specified by SOURCE to the active drive starting from track STRACK. Again T can be used to define a track address e.g.

WT 0 7 T0

DMON
Display Disk Monitor buffer.

Will display the area that has been assigned to store disk tracks with the read track command (see above). This area can then be disassembled, dumped etc. using the standard monitor instructions.

CLRDMON
Clear Disk Monitor Buffer.

Will Clear and de-allocate the area of memory that has been defined as a buffer for disk track reading and writing. When examining a disk using the RT command you should use this command before loading another set of tracks to avoid any confusion between the two lots of data.

6 DISK MONITOR COMMANDS

BOOTCHK (sectoraddr)
Check Boot Block Checksum.

This command is used on a sector that has been read into memory using the RT command. It will make a checksum of a bootblock located in memory at address (sectoraddr). If the checksum is incorrect it will be changed. Note you can also use T0 etc. if you have created a track buffer using the dmon command.

DATACHK (sectoraddr)
Check Data Checksum.

This command is used on a sector that has been read into memory using the RT command. It will make a data checksum on the sector located in the computer's memory at address (sectoraddr). If there is a checksum error the checksum will be corrected. Note again this command can be used with the T option to define half tracks in the track buffer.

BAMCHK (sectoraddr)
Block Allocation Memory Checksum.

This command is used on a sector that has been read into memory using the RT command. It will make a checksum on the sector located in the computer's memory at address (sectoraddr). If there is a checksum error the checksum will be corrected. The T options can again be used.

7 FREEZER AND RIPPER COMMANDS

SA (path)(name),(crate)
Save All

This instruction will save a copy of the frozen program to disk in standard Amiga format. (path) is the standard path as mentioned in the disk section. (name) is what you wish to call your program. (crate) is the compression rate which is in the range 0-!65535; the higher the value you specify the shorter the final file will be, however the actual compression will take longer and longer. At the maximum value the compression can take very very long. As a simple rule a value of about !200 should be plenty.
The following is an example of workbench 1.3 with 0.5Meg Ram extension in place and not switched out.

SA TEST	{no compression}
SA TEST1 ,!50	{compression 50 decimal}
SA TEST2,!200	{compression 200 decimal}
SA TEST3,190	{compression 400 decimal}

Will give for example if we do a DIR.

184210 TEST
145862 TEST1
091226 TEST2
084690 TEST3

Notice how the saving becomes less even though the rate is doubled.

SR (path)(name),(crate)
Save and Restart

This instruction is identical to SA apart from the fact that as soon as the file is saved the frozen program will be restarted.

LA (path)(name)
Load All

This instruction will reload a frozen file from disk and place it in memory ready to

restart. Essentially the opposite of SA.

LR (path)(name)
Load and Restart

This instruction will load a frozen file from disk and restart it immeadiatly.

SLOADER
Save Loader

Saves a copy of the loading program to disk called Aload. This is used to load a frozen file independently from Action Replay. Its primary use is when a bootblock is installed onto a floppy disk drive using the install instruction. From the Dos shell which is displayed after the installed disk is booted you should type the following line

ALOAD (name)

where name is a valid frozen file saved using SA or SR commands. This command is also useful for loading frozen files when you are using a hard drive. You should transfer both your frozen files and the Aload program to your hard disk and then from CLI type the above command.

SQ
Save Quick

Will save the current program in memory to a Ramdisk if there is enough free memory to do so. For example, if you have a one meg chip ram machine, to save to a ram disk you should firstly switch off the second 512K on the preference screen and reboot the machine. Then load your 512K program into memory and press the freezer button. Now type the line

SQ

For information on how to restart see LQ, LQR, EXQ and EXQR.

7 FREEZER AND RIPPER COMMANDS

SQR
Save Quick and restart

Will save a program to Ramdisk as in SQ, only the frozen program will be restarted immediatly after completion. Ideal for saving a point in a game in case you die so you can restart from a set point quickly.

LQ
Load quick

This command is the opposite of save quick, it reloads a frozen file from the ramdisk ready to restart.

LQR
Load Quick and Restart

This command is the opposite of SQR/SQ. It will load a file that has previously been saved to a ramdisk using save quick or save quick and restart. After reloading it will start at the point it was originally frozen.

EXQ
Exchange Quick

Effectively a combination of SQ and LQ this command will swap the frozen program in the ramdisk with the frozen program in main memory.

EXQR
Exchange Quick and Restart

The same as EXQ except the program taken from the ramdisk is restarted automatically.

SQMEM
Save Quick in fastmem

This command allows you to define that fastmem is used as the save quick area, or

to release it if it is already defined, e.g.

SQMEM 200000

would define that fastmem area 200000 (the area used if you have memory in an A590) would be used for the save quick area. There is a unique command.

SQMEM 0

which tells Action Replay to use the normal area.

TRACKER

Search for music track.

Will search through frozen memory for certain music format sequences. This will work best on Public Domain software where the authors are more likely to have used these packages. Please do not expect this feature to work miracles as music has no standard format like screens and samples do - so many commercial programs use their own.

A counter will be displayed as memory is searched. Three passes are made for different music formats. If any music is recognised a display similar to the following is shown.

SONG LOCATED IN MEMORY AT:$011BB2,SONGTYPE=SOUNDTRACKER (32 SAMPLES)
SONGNAME:LOADING..... YOU CAN NOW:
F1 PLAY MODULE,F2=STOP MODULE,F3=SEE MORE DETAILS,F4 SAVE MODULE
F5 RENAME SONG,F6 SHOW SONG DATA,F7=CONTINUE SEARCHING
F8CHANGE TO ST-16,F9=CALCULATE PATTERN LENGTH,F10 EXIT

The first number in the display is the address in memory where the tune is situated. The songtype is the piece of software that has been used to write the track, in this case it was sound tracker with 32 sound samples available to the track. The songname is just that, the name that was given to the piece of track when it was written.

7 FREEZER AND RIPPER COMMANDS

The function keys act as follows

F1=PLAY MODULE	{needs no explanation}
F2=STOP MODULE	(just that}
F3=SEE MORE DETAILS	{gives a more detailed view of the tune i.e. sample names etc.}
F4=SAVE TUNE	{Simply enter a filename and the tune will be saved in the current format}
F5=RENAME SONG	{Will give the track a new name}
F6=SHOW SONGDATA	{useful for displaying song data while playing a track}
F7=CONTINUE SEARCHING	{will continue searching for more tunes from where the tracker left off.)
F8=CHANGE TO ST-16	{attempts to change the track to so undtracker with 16 samples}
F9=CALCULATE PATTERNLENGTH	{Recalculates data}
F10=EXIT	{Exits from tracker}

SCAN

Scan Memory For Sample

Will display a new menu as below and a graph. The graph displays the contents of chipmem as a sound sample.

F1-HEAR SOUND	{plays sample between selectors}
F2-CALCULATE NEW GRAPHICS	{redraws the screen to show the sample between selectors}
F3-RESET	{resets selectors to full memory size and redraws the graph}
F4-EXPAND RANGE	{expands the size of sample on screen and redraws the graph}
F5-SAVE SAMPLE	{saves the sample between selectors to disk in IFF format}
(space)	{toggles the active selector between start and end}

7 FREEZER AND RIPPER COMMANDS

(left-arrow)	{move the current selector down in memory}
(right-arrow)	{move the current selector up in memory}
(mouse)	{holding the left or right mouse button down and moving the mouse will move the start and end selectors}
(up-arrow)	{increase period of sample}
(down arrow)	{decrease peroid of sample}

To find an appropriate sample in memory first play the whole of memory (F1) and notice whereabouts the + symbol is when it plays your sample. Then move the start selector to about this point and the end selector to roughly the end. Now resize the screen using (F2) and play the sample again. You may keep doing this until you have found the correct limits for the sample then when you are happy save it (F5).

8 MEMPEEKER COMMANDS

P(picnr)

Where (picnr) is the picture number, usually 1 or left blank. The screen will then be displayed and a large range of commands made available for the manipulation of these screens which are shown below or shown when the (F8) key is pressed. The possible combinations and types of manipulation could take a book in themselves, due to the great complexity of Amiga graphics. A bit of practice will soon get you used to what effect they have even if you are not sure what they are doing.

a.....	autoplane
b.....	increase brightness
(shift) b.....	decrease brightness
c.....	increase colour register
d.....	dual playfield on
(shift d).....	dual playfield off
e increase.....	increase right border
(shift) e.....	decrease right border
f fast.....	fast plane up
(shift) f.....	fast plane down
g.....	Interlace mode on
(shift) g	Interlace mode off
h.....	hold and modify (HAM) on
(shift) h.....	hold and modify off
i.....	invert all colours
l.....	lores mode on
(shift) l.....	hires mode on
m.....	modulo 1+2 plus
n.....	modulo 1+2 minus
o.....	modulo 1 minus
(shift) o.....	modulo 2 minus
p.....	modulo 1 plus
(shift) p.....	modulo 2 plus
q.....	clear modulo 1+2
r.....	rotate plane pointer
s.....	decrease left border
(shift) s.....	increase left border

8 MEMPEEKER COMMANDS

w	white helpscreen
(shift) w	black helpscreen
x	decrease colour register
y	switch diw and ddf mode
0	unlock all planes
(shift) 0	lock all planes
1	lock plane 1
(shift) 1	unlock plane 1
2	lock plane 2
(shift) 2	unlock plane 2
3	lock plane 3
(shift) 3	unlock plane 3
4	lock plane 4
(shift) 4	unlock plane 4
5	lock plane 5
(shift) 5	unlock plane 5
6	lock plane 6
(shift) 6	unlock plane 6
7	lock plane 7
(shift) 7	unlock plane 7
8	lock plane 8
(shift) 8	unlock plane 8
9	lock plane 9
(shift) 9	unlock plane 9
(+)	1 bitplane plus
(-)	1 bitplane minus
(=)	set all colours to bitplane 1
(F1)	set to default colours
(F2)	random colours
(F10)	set chosen picture into current program
(left)	rotate picture left
(right)	rotate picture right
(up)	scroll picture up
(shift)(up)	scroll picture up fast
(down)	scroll picture down
(shift)(down)	scroll picture down fast
(delete)	hide helpscreen

8 MEMPEEKER COMMANDS

(left mouse button).....	increase picture height
(right mouse button).....	decrease picture height
.....	set helpscreen with mouse on position
(esc).....	quit mempeeker
(help).....	show helpscreen

SP (path)(name),(nr) (height)
Save Picture

(path) and (name) are the usual format. (nr) and (height) are the picture number and height of the screen. Say for example we wish to save the frozen kickstart 1.3 screen to save it as an IFF file we could use

SP KICKSTART,1 !232

We could then reload this file into something like Photon Paint and edit your own Kickstart screen.

SPM (name)
Save Picture Mempeeker

Simply saves the screen we have been editing to disk and call it (name). Any modifications we have made to the mempeeker picture will also be saved out, for example changing the colours. This feature is ideal for saving out screens from games and using packages such as Photon Paint for editing and printing.

There now follows a very brief example of how to use the mempeeker. First switch on your Amiga and when the Kickstart screen appears press the freezer button and type in the following.

P (enter)	{to enter mempeeker}
(help)	{for the helpscreen}
(shift)-w	{to make helpscreen visible}
(left mouse button)	{to reduce size of screen}
9	{increase green}
(esc)	{exit mempeeker}
SPM KICKSTART	{new green kickstart}will save a new greener Kickstart.

9 TRAINER

The trainer commands are used to find various features such as infinite lives etc. which could ordinarily only be found by the most proficient hacker. There is a problem, however, in that if the original programmer did not want you to find his secrets he could easily protect his code. For example, if you want to find 3 lives the programmer could count this as 2 or even 128+3 so confusing the trainer. These techniques can now be beaten by the deep trainer, for more info see the following section.

The easiest way to understand how the trainer works is to explain how the programmer accesses the number of lives.

When the game starts the programmer sets up all his locations such as number of lives and energy and colours etc. The number of lives will be stored in a certain location in memory and each time a life is lost this value will be changed (usually decreased by one) until no more lives are present, in which case the game over sequence is displayed. What we need to find is the instruction that decreases the number of lives and remove it, then we will never get to 0 lives hence infinite lives.

TS (value)

Trainer Start

We use this instruction when starting to look for our feature (infinite lives, energy etc). This clears all locations found so far and searches for all occurences of the number (value).

T (value)

Trainer Continue

We use this command after we have used the TS command, exited and lost a life or energy etc. and re frozen. Usually the value will be one less than the TS command.

TF (address)

Trainer Find Decrement Instruction

We use this instruction after the TS and T commands and are left with one address. This will then find any possible Machine Code instructions that Decrement the location (address). This instruction is of more use to someone that has a bit of machine code knowledge. Others will find the TFD command more useful.

9 TRAINER

TFD (address)
Trainer Find Decrement Instruction And Remove

This command is the one that gives you your infinite lives. It will search through for decrements of (address) and remove them. (address) is the value found using TS and T. After issuing this instruction, try exiting and continuing the game. If it has worked you will have infite lives or energy.

TX
Exit Trainer

Exits trainer mode. Note if you have got infinite lives it will NOT go back to normal mode.

PC
Picture and energy count.

Displays the frozen screen so you can view how much energy or how many lives you had when the game was frozen.

USING THE TRAINER COMMANDS

The best way to show you how to use the trainer command is by example. In the following example we are using the remarkable, but difficult, Rick Dangerous game (not 2) by Firebird. If you have not got the game then follow the outline anyway, it is similar in many games.

First load up Rick and start your man running away from the ball and freeze the game before you lose a life. Then type the line

TS6

as we start Rick dangerous with 6 lives and the following should appear

FIRST TRAINPASS!
CHANGE THE COUNT VALUE NEXT TIME!
SEARCHED UP TO :0572A6
TRAINMODE ACTIVE!

9 TRAINER

Now reply with the X command to restart the game and lose a life (shouldn't be too hard), then as you start your next life freeze the game again and type the following

T5

As you have now got five lives the screen will show

POSSIBLE ADDRESSES:
044972
SEARCHED UP TO :080000
TRAINMODE ACTIVE:

We now only have one possible location for the lives counter so this should be it. If it displays more than one address you should loose another life and use T4 until you are left with one possibility. Now to give yourself infinite lives type the following.

TFD 44972

After a brief pause, all being well the computer should respond with the following lines

SUB FOUND AT :00045E3C
SUBS ELIMINATED!

This has now removed the SUB1 instruction which decreased your lives. Use the X instruction to restart the game and Hey presto millions of Ricks. You could have also used the Monitor command M to change the value at 44972 to say 0B (thats 11 decimal) for 11 lives.
There follows a few more examples of how to use this feature.

GOLDEN AXE

First you must start the game and after a brief start sequence when you are able to move your character around the screen you should press the freeze button. Even though you start the game with 3 lives enter the line

TS 2

9 TRAINER

Now get back to the game using the X command and lose a life. This is the standard technique to the trainer; you will always lose a life between the TS and every T command. When you are on your second life, press the freezer button again and enter the line.

T 1

The screen will show some possibilities for locations. You must continue till you only have one possibility. Restart the game again and lose a further life then freeze again. Now enter the line.

T0

You may notice that more than one possible location is being displayed and we have no further lives to lose, the solution is simple. Exit to the game using the X command and lose your last life. The game will end but all you need do is restart it. On your first life freeze the game and type the line

T2

Notice we do not use the TS even though we have restarted the game as we are still looking for the same lives location. Continue this losing a life and refreezing until all of memory is searched. There will be only one possible location. If there is none and the trainer has failed you should try the whole process again as you have made a mistake on one of the number of lives. The location displayed should be 005955. The TFD command will not work on this game (try it if you like) so we must resort to a different technique to increase the number of lives. Type the line

M 5955

The screen will display the current number of lives (only valid if you froze the game while it was being played) as follows

005955 02 03 0c

with a whole lot of other numbers. The only important one is the 02, i.e. the first one

after the location number. Use the cursor keys to move up to the 02 and change it to 7F (127 decimal) and press the return key when you have changed it. Press X to exit and you should have 127 lives which should be ample. If you come back to this game at a later date and wish to enter the cheat, you need not repeat the whole process again, simply start from the M command above.

RICK DANGEROUS 2

Start the game and as soon as Rick appears on the first screen and is able to move, press the freeze button and type the line

TS 6

Repeat the procedure X then lose another life and enter

T5

One further line should do it

T4

the screen will display only one possible address 0178AF. If you use the command TFD 0178AF nothing will be found but using a value one less will, i.e.

TFD 178AE

notice how leading zero's are not important but trailing ones are, as is the same with normal decimal numbers. This technique of using a value one less than that found using the T commands is very useful when the value is odd (i.e. ends in 1,3,5,7,9,B,D or F) and replacing this last digit with the even value one less i.e. (0,2,4,6,8,A,C,E respectively).

STRIDER 2

First start the game so your man is on the screen and able to move around, your lives counter will display 4, and freeze the game. In this case you should start the game with

a value one more than your life counter i.e.

TS 5

now exit the trainer using the X and lose a life, then refreeze and type the following

T4

the screen will display only one possibility. Repeating the process numerous times will prove that this is definetely the one you are after as the only location displayed will be 6AD5 - this is the correct one! TFD 6AD5 will not work so enter the following

TFD 6AD4

Action Replay will then remove the instruction involved in decreasing your lives. Now simply exit using X and you have infinite lives.

MIDNIGHT RESISTANCE

After reading through the previous examples you should be ready for one or two abbreviations. Between each of the following T instructions you should lose a life and then refreeze. If you lose so many lives that the game is over simply restart the game and carry on. Do not use the TS command though.

Start game
TS 2
T 1
T 0
Restart game
T2
T1
T0

You will notice that however many times you lose a life there are always two possible locations. This is probably due to the way the programmer handles two players. The only way to find out which is the correct one is to try them. The two values are 11692

and 11767. Typing the following

TFD 11692

and testing using X and playing the game will have the desired effect so there is no need to try the other value.

ESWAT

This is abbreviated in the same way as above. After each TS and T command you should lose a life and re-freeze.

Start game
TS 2
T 1
T 0
Restart game
T 2

again the TFD command will not work so you must change the number of lives manually. Type

M 1BC57

the screen will look something like the following

001BC57 01 00 00 etc.

and a whole line of numbers. Changing the 01 to 7F (it does not have to be 01 but it is the FIRST number on the line after the address) will give you 127 lives. Remember to press return after you have changed the line.

NITRO

This is a good example of how the trainer can be used to find items that are not as predictable as lives and can go up as well as down. This cheat is for money that you

use for buying cars, fuel etc. First you must finish the first round and gain some money on your way. When you get to the parts shop, the screen will show you how much money you have got. Now freeze the game. If you have 6 units of money, say, start the trainer with

TS 6

If you had 8 units of money you would of course use TS 8 etc. Restart the game, buy a couple of items to change the remaining money. When I tried it it went down to 3. Again refreeze the game and type the line

T 3

or whatever. Repeating this procedure a couple of times should give you a single value, if not you should do another lap and get some more money and keep trying till you get the value 1FBC7. There is a difference to your normal infinite lives in that you do not really want to stop decrementing money (although it would help if you have no money to start with you cannot buy anything). The best way is to use the M command as follows

M 1FBC7

then change the first number after the 1FBC7 to a value of say 50 for 50 units of money. Remember to press enter after you have changed the line.

NIGHTBREED (arcade action)

To show you that once you have found the infinite lives location in a game you need not search for it again type the following

TFD 24A

and you will get infinite lives. The TS, T commands can be skipped out in all the previous examples if you like, the whole procedure is there for example only. This location was found using TS3, T2, T1, restart, T3, T2 and subtracting one from the number found.

9 TRAINER

ROBOCOP 2

Use the following in the normal way

Start game
TS 2
T 1
T 0
Restart game
T 2
TFD 8034 (one less than the value found)

JAMES POND

TS 3
T 2
T 1
TFD 1B0 (1B1 does not work)

BATMAN THE MOVIE

This was found using the standard technique, type the following

TFD 7C876

YOGI'S GREAT ESCAPE

This was found using a slightly different technique involving a little knowledge of machine code. I will only mention that the FA command was used, a very useful comand for the hacker. To get infinite lives use the monitor command M 7B5E6 and alter the first 6 numbers as follows

7B5E6 4E 71 4E 71 4E 71

then press return and you will have infinite lives.

9 TRAINER

**DEEP TRAINER *

There are several trainer type commands introduced with the MKIII Action Replay. These go under the generic title of deep trainer commands. These are totally separate to the above trainer commands and are treated as a different subject. To use the deep trainer you must have at least 1 MB of memory. This technique is of use when the previous commands fail and especially where energy bars are used although it can be *very* time consuming and at first may seem a little bewildering.

**TDS *
Deep trainer start

This command starts the deep trainer mode. Note how there are no variables in the line, it is quite simply TDS. You will use this command when you have just started a game, when you have full energy or lives. Each time you freeze the game and the value is the same (full energy 3 lives or whatever) you will use the TDS option.

**TDC *
Deep trainer change

This command is similar in a way to the T command. You enter this when the original value entered has changed. So if you started a game with, say, full energy you would start using the TDS command. Then you would lose some energy, re-freeze and enter the TDC command. Notice how no numbers are entered. From here on you keep re-freezing the game at different points. If you have full energy you will use the TDS command, if you have anything but full energy you will use the TDC command.

**TDD (start) (end) *
Deep Trainer Delete Addresses

This command is for the more experienced hacker. It will remove all search addresses within the range (start)-(end).

9 TRAINER

**TD *

Display deep trainer addresses

This command will display a list of all the addresses under investigation by the deep trainer. If you try it after just starting with the TDS command the list will be very long!

**TDI *

Display Probable Deep Trainer Addresses

This is the most important of the deep trainer commands. It will list all possible locations that your energy/lives etc. can be at. You should repeat the TDS and TDC commands over and over until typing the command TDI comes up with the same values over and over again. One of these values is the address you are looking for. If you find that there are no possible addresses it is more than likely that you have made a mistake and typed TDC instead of TDS or vice versa. You should TDX and then start again.

**TDX *

Exit Deep Trainer

This command is used to abort the previous deep trainer commands. All addresses will be reset. This is useful when you have made a mistake and need to restart.

USING THE DEEP TRAINER

There follows one or two examples on how to use the deep trainer. Even if you do not have the games mentioned you should try to follow what is happening as it will help you to understand the principle of the deep trainer.

ROBOCOP2

The following procedure will find the life counter on Robocop2. First switch the machine on and go to the F3 options screen. Switch off all extra memory, i.e. select 512K chip mem only and no extra fastmem. Load the game from disk and when loading is complete start the game. When the game is started press the freeze button. Now type the following lines

9 TRAINER

TDX
TDS

and return to the game. Lose one life and refreeze the game. Now type the line

TDC

Lose a further life and refreeze the game and type the line

TDC

again. If you lose another life the game will end. You should restart it and freeze again. You now have 3 lives which is the same as when you first used the TDS command so you should again type the line

TDS

and then repeat the procedure above i.e. lose life, TDC lose life and TDC again. After this procedure type the line

TDI

on my machine the list of possible locations was as follows

000B02 000C34 008035 00C689

you may find on your machine one or two extra, this depends at which points you had frozen the game. Only one of these values is the correct one so a technique of trial and error is called for.
If you try a TFD command (used in detail in the previous trainer section) on each of these values nothing will be found, however if you use the technique of using one less with the odd numbers on 8035 i.e. 8034 as follows

TFD 8034

Action Replay will find and eliminate anything that decreases your number of lives.

9 TRAINER

Those of you that have read the previous section will note that this address is the same as that found in the previous section only it took longer. The beauty of the deep trainer method is that no values are needed and therefore you can use it with energy as well as lives.

10 MISC COMMANDS

RAMTEST (start) (end)
Check Memory for Faults.

This instruction will destroy all data between (start) and (end). It writes 0 to all locations in the region (start)-(end) and reads them back, then does the same for FF (hex).

PACK (start) (end) (dest) (crrate)
Pack Memory

This command will pack the block of memory between start and end and place it at (dest). (dest) may be at (start) if required, and uses the compression rate (crrate) as explained in the SA command. When completed the length of the compressed data will be displayed, e.g.

PACK 40000 4FFFF 50000 !200

will pack all data between 40000 and 4FFFF and place the compressed data at 50000 using a compression rate of 200 decimal. Make sure you keep a note of the length!

UNPACK (dest) (end of packed)
Unpack Packed Data

This is the reverse of PACK. You must specify the destination of the unpacked area (NOT OVER ITSELF!) and the last byte of the compressed data (end of packed) which can be calculated by the start of the compressed data + length displayed after completion of the PACK command. For example assuming the length of the packed data in the previous command was 1234 then the reverse of the above example would be.

UNPACK 40000 51234

COLOR (back) (pen)
Set Screen Colours

Will set the screen colours using the Amiga palette (0-!4095) (back) being of course the background colour and (pen) the foreground colour. This is an alternative to the

preference screen (F3). When used on its own (COLOR) it will display the background and foreground settings.

RCOLOR
Reset Screen Colours

This command is for when you have made a mess using the COLOR command.

TMS (address)
Mark Address

This command is used as a notepad for reminding you of an address you are currently investigating. They are also saved using SA and reloaded using LA/LR so you can remember for example where an infinite lives location is. There are 10 possible memory markers.

TMD (address)
Marker Delete

This will delete one of the current memory markers

TM
Show marker

Will display the current memory markers.

SPR (nr|addr)
Edit Sprite

Will edit the sprite number (nr) or the sprite at address (addr). If, for example, we have frozen workbench we type the line

SPR 0

and press (return) about twenty times, the outline of the pointer would appear in various numbers, each number corresponds to a colour. We can if we like move the cursor keys

10 MISC COMMANDS

and edit the sprite making sure we press return after every line we have changed. Note you can only use the numbers 0-3 as you can only have 4 colours per sprite.

VERSION
Show ROM Version

Will display the date and version number of the current ROM's

MEGASTICK (1) *
Joystick Handler

This command invokes a new screen - the joystick handler screen. It is used so you can set the joystick up to act as certain keys, e.g. type megastick, press the fire button on player 1 joystick, pull to the left and press space. From now on when you restart, moving the player 1 joystick to the left and pressing fire will produce a space. All directions can be set up and combinations of fire-pressed etc. Press escape to accept a setup. Use the option 1 if you only wish to select player 1.

NOSTICK *
Remove joystick handler

Disable the mega stick feature.

CLRSTICK *
Clear megastick values

This command clears any codes set by the mega stick command back to zero.

SSTICK *
Save megastick values

Saves the values that have been set up using the megastick command to disk for re-use at a later date.

10 MISC COMMANDS

LSTICK ***

Load megastick values

Loads the values for the megastick command that have been saved using the Sstick command.

RESET ***

Reset Amiga

Resets the Amiga as though ctrl-A-A has been pressed.

PAL ***

Switch to PAL mode

Switches the Amiga into the Eurpoean PAL standard.

NTSC ***

Switch to NTSC mode

Switches the Amiga into the American/Japanese NTSC mode.

SETMAP ***

Set keymap

Will display a screen which will (if possible) allow the editing of keys on the Amiga keyboard. For example the function key 1 can be changed to send the words “Function key pressed”. To use this command simply type setmap. Use the mouse to select which key you wish to change and click on it. A new window is displayed with the current string and allows you to change it. When you have entered a new value press (enter) and click on (OK) to accept. When you are happy with the new keymap you can save it by clicking on the appropriate option for reloading at a later date. When you wish to install the new keymap click on (install) and the new map will be written to memory so when you exit Action Replay the new map will be used.

10 MISC COMMANDS

ASCII ***
Display ASCII map

This deceptively useful command will show the ASCII (American Standard Code for Information Interchange) character map and the corresponding codes. It is ideal for programmers who do not have a reference manual to hand.

ALERT (guru-number) *
Display Guru List

This command is useful for programmers - when you get the guru numbers it is often difficult to remember what the function is. For example the guru exception 8400000C does not mean much to me but if you type the line

ALERT 8400000C

will reveal all (well maybe if you are a programmer). Typing Alert on its own will list all the gurus and descriptions of such.

SMALLCHAR *
Small Printer characters

If you have an Epson printer attached this command will reduce the size of the text sent to your printer.

NORMALCHAR *
Normal Printer characters

Reverses the action of smallchar.

PRT *
Print String

This instruction sends a sequence of characters to the printer. Either text in quotation marks may be sent or a number may be sent in which case the ASCII value is sent to the printer. The following example will print “hello there” in expanded print to an Epson printer.

10 MISC COMMANDS

PRT 1B 57 31 “hello there”

VIRUS

Search memory for virus

Will search through memory for any known virus. This is not usually necessary if you have left the virus test ON in the (F3) preference screen as you will be warned of any virus. Please note if you have any virus killers in memory this may confuse Action Replay as the program will have bits of the viruses' DNA in memory.

KILLVIRUS

Search and Remove Virus

Search for and remove any viruses resident in memory.

11 MONITOR

The following set of commands are of most use to machine code programmers. Again I must stress that to beginners they may seem completely bewildering. However, if you are interested in programming machine code there are numerous Amiga books that could start you off and as you read the Action Replay monitor commands, they will begin to make sense. For those of you that are already conversant in machine code you will find that these commands will become essential and you will wonder how you ever did without them. One or two significant additions to the MK II have been made to make life easier, such as the Im command and the trace commands.

SETEXCEPT

Sets The Exception handler

Gets rid of those annoying guru messages.

COMP (start) (end) (dest)

Compare Memory Blocks

Will compare the memory between (start) and (end) with that located at (dest). All diferences in the destination block will be displayed.

LM (path)(name),(dest) *

Load file to memory

This allows machine code or data to be loaded straight into memory at any point and be investigated at leisure. The memory remains when the program is restarted so is ideal for changing blocks of data/graphics etc. e.g.

LM DATA,70000

will load the contents of file data into memory starting at location 70000 in hexadecimal.

SM (path) (name),(start) (end)

Save Memory To Disk

This command is used to save a block of memory to disk, in the standard (path)(name) format,which is located between (start) and (end). Ideal for saving blocks of machine

code or other forms of data, e.g.

SM scroll,3021F 312EA

SMDC (path)(name),(start) (end)
Save Memory To Disk In DC.B format

This feature is ideal for the assembly language programer where a block of memory can be saved out as ASCII in the style DC.B followed by the byte values separated by commas which can be loaded into most assembly packages such as DEVPAC, ARGASEM etc.

SMDATA (path)(name),(start) (end)
Save Memory To Disk In DATA format

This feature is similar to SMDC only the format is DATA instead of DC.B

A (address)
Start M68000 Assembler

Begin entering assembly language instructions from the specified address, all standard 68000 instructions are supported and as long as you enter valid instructions you can keep on entering. To finish simply press (ESC). I am *NOT* going to explain the 68000 instruction set, there are plenty of good books on that.

BS (address)
Set Breakpoint

This command is ideal for the hacker or programmer who requires information at a certain point in a program. You can set a point in memory specified by (address) to break out of the main program and enter Action Replay. You may use this instruction after using the TF command and finding a life decrement instruction, say at location 41259. So you could investigate in more detail the screen and registers, use the commands

11 MONITOR

BS 49152
X

and as soon as the location 49152 is reached you will get the Action Replay screen back.

B
Show Breakpoints

Will show all breakpoints set using the BS command

BD (address)
Breakpoint Delete

Delete the breakpoint previously set at memory location (address).

BDA (address)
Delete all breakpoints

Remove all breakpoints that have been set using the BS command.

X
Restart current program

This will quite simply start the program from where you pressed the freeze button. Any changes you made to the registers or memory will be kept and so if you have made major changes the computer may produce unexpected results or crash.

C (1|2|address)
Copper Assembler/Disassembler

The line that should be entered is C then 1 or 2 or an address. The Amiga is equiped with a copper or co-processor which can be programmed in a similar, but more limited fashion to the 68000. After you enter the C and one of the possible options the copper instructions of the locations will be displayed. Pressing (return) will enter the instruction on the line and move to the next instruction in memory. For information on the copper

chips please consult an Amiga Technical reference manual.

D (address)
Start Disassembly

D followed by the (return) key will begin Disassembly in 68000 m/c from the point at which the program was frozen. If an (address) is entered then dissassembly is started from that point. Please note you can specify numbers in different ways e.g. T0 for track 0 if the diskmonitor is being used or \a0 etc. for the machine code registers, for example

R

the computer may respond with

D0=FFFF1234 00000000 00000028 000059d2 etc.
A0=00000676 000FFE1E 000FFF75 00004984 etc.

to view the machine code starting at the address in A1

d \A1

will start dissassembly from address FFE1E, pressing (return) will dissassemble the next line and so on until (ESC) is pressed. The cursor keys can be used to control the direction of assembly.

E (offset)
Show/Edit Chip Registers

Will show the contents of the chip registers in binary. These values can be changed using the cursor keys and pressing (return) when a line has been modified.

F (string),(start)(end)
Find String In Memory

Will search through memory between (start) and (end) for all occurrences of (string), which should be surrounded by quotes. The search is case sensitive. This command

is also useful for finding occurrences of bytes in memory or even instructions. FS will search for a string but is not case sensitive e.g.

F "COMMODORE",0 7FFFF

will search for COMMODORE (note in upper case) between the limits 0 and 7FFFF.

F 4E 71,70000 80000

will search through memory from hex 70000 to hex 80000 for occurrences of bytes 4E followed by 71 which also corresponds to the assembler instruction NOP.

FA (address) (start) (end)
Search for Adr Addressing opcode

Will search through memory for Adr Addressing opcodes. This is a very useful debugging/hacking tool for the more advanced user. Its action is to search through all locations between (start) and (end) for machine code instructions that access location (address). FAQ will fastsearch through memory. The use of these instructions is to find out where a subroutine is called from, e.g.

FA 71000 60000 80000

on my machine responded with the following lines

070002 BSR 00071000
07000C BRA 00071000
Searched up to adr: 080000

FR (string),(start) (end)
Search relative for String

Searches through memory for a relative string and displays the offset. For example if a piece of text is hidden by adding one character it will be shown as occurring with an offset of one so a search for "HELLO" on finding "IFMMP" will display an offset of one

11 MONITOR

G (address)
Goto Address

G on its own will act the same as X. With an address specified G will set all the registers to what they were when the program was frozen (unless they have since been modified) and jump to (address). Ideal if you want to redirect the program or test out an Assembly program you have just entered.

TRANS (start) (end) (dest)
Transfer Memory Block

Will transfer a block of memory between (start) and (end) to Destination (dest). The transfer is intelligent so you may specify a (dest) which lies between start and end.

WS (string),(address)
Write String To Memory.

Will write a STRING (bounded by quotes) to the memory starting at (address) you can also use numerical values (not in quotes), e.g.

WS "HI There",41259

M (address)
SHOW/EDIT Memory As Bytes

This will display a line of data in byte format in Hexadecimal starting from location (address). The bytes can be changed to any valid Hexadecimal number using the cursor keys and approriate numbers. When a line has been altered pressing (return) will enter those alterations to memory and display the next set of bytes from memory. Press escape to exit this feature.

N (address)
Show/Edit Memory As Bytes

Similar to the M Command but memory is shown as ASCII.

11 MONITOR

NO (offset)
Show/Set ASCII Offset

When a value is entered as (offset) any further use of the N command will add the (offset) value to the characters in memory that are displayed, e.g. if we did an ASCII dump using N and got the following in a block of text

IFMMP

and we then entered the line

NO 1

The same piece of text viewed using another N command would be

HELLO

So the command can be used for showing hidden text. NO on its own shows the current (offset).

NQ (address)
Display Memory Quick

Will display memory in ASCII in quick format

MEMCODE (start) (end) (code)
Encode Memory

Will go through a block of memory encrypting each byte using the 68000 EOR instruction with (code). This has the effect of rendering the block unreadable. To decypher this code you may either use the 68000 instruction EOR.B or execute the instruction again in an identical manner e.g.

MEMCODE 312FE 345A1 EA

will encode a block of memory using the key EA (in hex). To get it back to normal simply

type

MEMCODE 312FE 345A1 EA

O (string),(start) (end)
Fill Memory With String

This command will fill a block of memory with repeated sequences of (string), which should be bounded by quotes.

R (register) (value)
Display/Modify Registers

R (return) will display the contents of all the 68000 registers at the time of freezing. If you wish to modify these registers you should specify both a (register) number and the (value) to be inserted in it e.g.

R D0 !1000

will insert 1000 decimal into the 32 bit data register D0. Any modifications to these registers will become valid when you either exit using the X or G commands.

W (register)
Display CIA Contents

Displays the contents of both the CIA (8520) chips. The value in (register) used to define the offset from the start of the CIA's. Both chips are displayed on the same line, the first and second numbers being the first and second CIA's correspondingly, for example

W 4

will display the contents of the fourth register on both the first and second CIA's.

Y (address)
Display memory in binary form

11 MONITOR

This instruction is very similar to the M command although memory is displayed and edited in binary form. The amount of bits displayed on one line is specified by YS (value) where value is 1-8 giving the amount of bytes displayed per line.e.g.

YS 8

will tell the Y command to display 64 bits per line. The address will be updated accordingly.

MS (address) *
Set memwatch point

This is a wonderful instruction for all you programmers out there - how many times have you found memory locations becoming corrupted and wondering what on earth was responsible for doing this? All you need to do is use the MS command with the byte address you wish to keep a check on and start your program (either change register PC or the G or X commands) and as soon as any command changes your location, Action Replay is activated and the program counter points to the next instruction after the one which has done the damage, e.g.

MS 70000

would, as soon as any new value is entered into location 70000, freeze the program and display a message. This I find is of more use than the breakpoint instructions.

MW *
Show watchpoints

This will display a list of all the memory locations set using the MS command.

MD (address) *
Delete Mem watchpoint

Will delete an address from the memwatch list.

11 MONITOR

MDA ***

Delete All Watchpoints

Will remove all addresses set using the MS command.

TR (steps) *

Trace without subroutines

This is another useful command, it will trace a machine code program (steps) number of steps from the current frozen position (defined in the registers). All memory and system data is updated including screens etc. This makes this instruction far more powerful than equivalent disk based monitor trace commands as all system information can be viewed, e.g. the screen using PC. Note that all subroutines are executed as a single instruction.

ST (steps) *

Trace with subroutines

This is similar to the TR command apart from the one difference in that subroutines are stepped down as continuous order and are not treated as a single instruction as is TR.

? EQUATION *

Calculate Value

This instruction is ideal for altering between number bases. The result of the (equation) will be displayed in binary Hex and Decimal e.g.

?8+7+!15*2

will give the result !60. Notice this is not the same as previous versions of Action Replay, the standard priority orders of multiplication and division first and addition and subtraction second are now used. Valid symbols are (+,-,/,*)

12 COMMANDS FOR SYSTEM INFORMATION

Most of the commands in this section are devoted to complex System information. Some, however, such as Avail are useful to everyone.

AVAIL
Available Memory

Quite simply displays the memory available on the machine both chip ram and Fast ram. Ideal if you have fitted one of DATEL's Rammaster II's for example to check that it is functioning properly.

INFO
Display System Info

Will display a list of important system information for example the colour palette values.

LIBRARIES
Show Execbase Librarylist

This command will display a list of the Execbase Library list and their corresponding addresses.

INTERRUPTS
Show Execbase Interrupt vectors

Lists all currently active Execbase interrupts.

EXCEPTIONS
Show Exception and Interupt vectors

Shows the list of 68000 exceptions and also shows where in memory they are currently pointing.

EXECBASE

Show Exception and Interupt vectors.

12 COMMANDS FOR SYSTEM INFORMATION

RESOURCES

Show execbase Resource List

CHIPREGS

Show Name And Offset Of Chipregs

DEVICES

Show Execbase Devicelist

TASKS

Show Execbase Tasklists

PORTS

Show Execbase Portlist

INDEX

INDEX

NOTES

ADDENDUM

When using their Amiga Action Replay there are one or two important points that some people seem to have trouble with.

1)The F3 Options Screen - most of the features on this screen do not take effect until the machine has been Warm Booted, i.e. select F3 for the Options Screen then change to the appropriate Option, click on "use" then "x" to get back to the normal Amiga screen.

Finally do a Warm Reset (press the Ctrl-Amiga-Amiga keys all together) and the selected features such as Removing Memory, Variable Boot will now take effect. This feature is especially important when using the Deep Trainer command. If you do not follow this procedure you will get the message "Nowork mem"!

2) Do not use the install 1 command (install a Virus Checking Boot Block). If you wish the Autobooting Disk to work under Kick-start 2 you must use install 0.

NOTES

Amiga ACTION REPLAY MK III

GOVAN ROAD, FENTON INDUSTRIAL ESTATE,
FENTON, STOKE-ON-TRENT, ST4 2HS, ENGLAND.

Amiga Hardware World

Everything about Amiga hardware...

~

http://amiga.resource.cx