The A64™ Package

VERSION 3.0

QUESTRONIX™

COPYRIGHT

DISCLAIMER

QUESTRONIX, MAKES NO WARRANTIES, EITHER EXPRESSED OR IMPLIED, WITH RESPECT TO THE PRODUCT DESCRIBED HEREIN, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. THIS PRODUCT IS SOLD "AS IS." THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAM PROVE DEFECTIVE FOLLOWING ITS PURCHASE, THE BUYER (AND NOT QUESTRONIX OR ITS DEALERS) ASSUMES THE ENTIRE COST OF ALL NECESSARY DAMAGES. IN NO EVENT WILL QUESTRONIX BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE PRODUCT EVEN IF IT HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITIES FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY.

CREDITS

Program Design: Cliff Dugan
Sound Emulation: Bent Nielsen
Documentation: Cliff Dugan

ACKNOWLEDGEMENTS

Thanks to all the registered owners who have supported The A64 Package.

Thanks to all the people who have sent us technical information, computer programs and suggestions for improving The A64 Package.

ThanX to SteveX for solving the WorkBench V2.04 problem.

Thanks to Doug Cotton of Creative Micro Designs, Inc. for his help in emulating the C64's REUs.

Special thanks to Bent Nielsen for his numerous contributions to A64.

TRADEMARKS

The A64 Package, A64, A64 BASIC, A64 Prefs, A64Mon, A64Tools and QuesTronix are trademarks of QuesTronix.

Commodore 64 and C64 are trademarks and COMMODORE is a registered trademark of Commodore Electronics, Ltd. Amiga, AmigaDOS and Workbench are registered trademarks of Commodore-Amiga, Inc. GEOS is a trademark of Berkely Softworks. Archon is a trademark of Electronic Arts, Inc.

Any other products or services mentioned are trademarks or service marks of their respective holders.

SYSTEM REQUIREMENTS

All programs in The A64 Package are compatible with all Amiga models using V1.3 - V3.0 of the operating system and with all microprocessors, from the 68000 to the 68040.

A64 requires a minimum of 1mb RAM. 2mb RAM is recommended.

Two floppy drives or a hard disk is recommended.

A separate adapter cable is required for the Amiga 1000.

UPDATES

QuesTronix will notify registered owners whenever updates or revisions are made to The A64 Package. To be eligible for updates you must fill out the enclosed registration card and mail it to QuesTronix.

REPLACEMENTS

QuesTronix will replace, free of charge, disks or hardware interfaces that have defects in materials or workmanship, providing they are returned to us within 90 days of purchase and they are accompanied by a dated sales receipt. Please contact QuesTronix before returning merchandise.

TECHNICAL SUPPORT

Technical support is available to registered owners only. To be eligible for technical support you must fill out the enclosed registration card and mail it to QuesTronix.

HOW TO CONTACT QUESTRONIX

You can contact QuesTronix by calling (203)666-8260, 9 am - 5 pm EST, or by writing to:

QuesTronix
P.O. Box 340265
Hartford, CT 06134-0265
U.S.A.

Table of Contents

Chapter 5
A64Mon

Chapter 6
Convert

Chapter 7
A64Tools

Chapter 8
CLI Utilities

Chapter 1
Introduction

Welcome to The A64 Package. This manual assumes that you are familiar with the basic operation of the Amiga and the terms used to describe it. If you are new to the Amiga, please refer to your Amiga user's guide when necessary. This manual also assumes that you are familiar with basic use of the Commodore 64. If you would like to learn more about using the Commodore 64, there are many good books available for all levels of expierence.

Throughout this manual some names are used that we would like to clarify. When you see the name "*The A64 Package,*" we are referring to the entire package, including all of its utilities. When you see the name "*A64*" by itself, we are referring only to the A64 program. The names "*A64's hardware interface*", "*hardware interface*" or just "*interface*" all refer to the same interface.

Some parts of this manual may appear technical to some users. We did our best to try and explain things in a way that an average user would understand them, but it is not always easy to explain technical features simply. If you do not understand a feature, don't hesitate to experiment with it.

We recommend that you fully read this manual before attempting to use any of the programs included with The A64 Package. The more familiar you are with The A64 Package, the easier it will be to use and you will be less likely to encounter problems that you can not solve.

What is The A64 Package

The A64 Package is a hardware and software product. It was designed to give Amiga owners the ability to access Commodore 64 peripherals, programs and data directly from their Amiga.

The hardware is an interface that plugs into the parallel port of your Amiga and allows The A64 Package to access Commodore 64 disk drives and printers.

The software is a collection of Amiga programs that help bring the world of the C64 to the Amiga. The main program included with The A64 Package is A64. A64 is a Commodore 64 emulator. The word emulate means "*to strive to equal or excel, especially through imitation.*" A64 emulates the Commodore 64. This means that using A64 is just like using a C64. There are literally thousands of programs available for the Commodore 64 and A64 will run a majority of them. The A64 Package also includes several utilities that are meant to compliment A64. These utilities serve a variety of valuable purposes: like transferring files between C64 and Amiga disks and speeding up C64 emulation.

The A64 Package is especially well suited for C64 owners who want to upgrade to the Amiga. Many Commodore 64 owners have spent hundreds or even thousands of dollars on their C64 peripherals and software. Many of them would like to upgrade to the Amiga, but they do not want to lose their investment. There is a solution! The A64 Package will allow them to continue to use their familiar Commodore 64 software and peripherals while they're learning how to use their powerful, new Amiga.

Now bridging the gap from the Commodore 64 to the Amiga is easier than ever before. The A64 Package helps to make the upgrade path a little easier to take.

Chapter 2
Getting Started

The A64 Package is distributed on two disks. Disk #1 is for A64 and disk #2 is for The A64 Package's utilities. The first thing you should do is make backups of these disks and store the originals in a safe place. The A64 Package is not copy protected in any way and the disks can be copied from the WorkBench.

Installing A64's Hardware Interface

The hardware interface is a small box with two connectors on it. The large connector is the end that plugs into your Amiga and the small connector is the end where you plug in your C64 disk drives and printers.

To install the hardware interface, use the following procedure:

1. Make sure your Amiga is turned off. You should NEVER plug anything into your Amiga while it's turned on. You can damage your Amiga and peripherals by doing so.
2. If you have any Amiga model, other than the A1000, you may now plug the interface directly into the parallel port. If you have the Amiga 1000, you must first plug the adapter cable into the parallel port and then connect the interface to the cable. *Please note that this adapter cable is not a simple gender changer and only a cable specifically made to connect A500/A2000 devices to an A1000 parallel port should be used. QuesTronix has these cables available. If you need one, please contact us directly. Using any other type of cable can damage your Amiga!*
3. You are now ready to connect your C64 disk drives and printers to the interface. Make sure that each C64 device that you wish to connect has its power turned off. Using the serial cable that came with your disk drive or printer, plug one end of the cable into the interface and the other into your C64 peripheral. The C64 peripherals may now be "daisy chained" in the same way as they would be on a C64; by plugging each new device into the back of each previous device. After all C64 devices are connected you can then turn them on. A64 will support up to four different C64 disk drives and two separate C64 printers, simultaneously.
4. You may now turn on your Amiga and boot it up as you normally would.

NOTE: Pin #3 is intentionally removed from the 25 pin connector on A64's hardware interface.

The hardware interface can stay plugged in at all times without interfering with any programs on your Amiga as long as they do not use the parallel port. If you want to use your parallel port with some other piece of hardware, like a printer or digitizer, you must either remove the hardware interface and install the other device or use an A-B switch box. A-B switch boxes are simple devices that allow you to switch between two or more devices that use the same port. If you will be using more than one device on your parallel port, we highly recommend that you use a switch box. Switch boxes can be purchased from most computer stores and mail order companies. The type of switch box you should get is "*Centronics*

Parallel with female DB-25 connectors."

Installing The A64 Package Software

Before running A64 you must first use the Install utility that is on disk #1. The Install utility will copy some files that are needed by A64 to either your WorkBench disk or your SYS: partition of your hard disk. If you do not have a hard disk and you do not want to add these files to your normal WorkBench disk, you should make a copy of your WorkBench and use the copy whenever you want to use A64. If you decide to use a copy of your WorkBench, make sure you reboot your Amiga using the copy before running the Install utility.

To install The A64 Package software, use the following procedure:

1. Insert the A64 disk (disk #1) into any drive.
2. Double-click the A64 disk icon to open the disk window.
3. Double-click the TAP drawer icon to open the drawer window.
4. Double-click the Install icon.
5. A window will open and you will be prompted if it is ok to continue. Type [Y] to continue. The Install utility will then add the necessary files to your WorkBench or SYS: partition of your hard disk.

After running the Install utility, you are now ready to run A64.

Installing The A64 Package on a Hard Disk

The optimum set up for The A64 Package is to install the entire package onto a hard disk. If you don't have a hard disk, you can skip this section.

To install The A64 Package on a hard disk:

1. Insert the A64 disk (disk #1) into any drive.
2. Double-click the A64 disk icon to open the disk window.
3. Drag the TAP drawer from the A64 disk to the hard disk.
4. Double-click the TAP drawer icon (on the hard disk) to open the drawer window.
5. Insert the A64Utils disk (disk #2) into any drive.
6. Double-click the A64Utils disk icon to open the disk window.
7. Extend select all of the icons on disk #2 and then drag them to the TAP window on the hard disk.
8. If you have not already used the Install utility, you should do so now.

There are several Config File icons on disk #1 that have their default tool set to "A64:TAP/A64." If you install The A64 Package on your hard disk and then try running A64 with these Config Files, the Amiga will try to load A64 from a floppy disk named "A64" instead of loading A64 from your hard disk. Before using these Config Files on your hard disk, you should update their default tool with the WorkBench Information menu option in the Icons menu. To update the Config File icons, follow this procedure for each icon:

1. Click the left mouse button once on the Config File icon.
2. Select the Information menu option in the Icons menu on the WorkBench Screen.
3. A window will open showing a picture of the icon and miscellaneous information about the Config File. Select the Default Tool gadget.
4. Delete the A64: portion of the name and replace it with the name of the hard disk partition that you installed The A64 Package to. For example, dh0: or Sys: or Work:
5. Select the Save gadget.

Running A64 for the First Time

To run A64:

1. Insert the A64 disk (disk #1) into any drive.
2. Double-click the A64 disk icon to open the disk window.
3. Double-click the TAP drawer icon to open the drawer window.
4. Double-click the A64 icon. It's the icon that looks like an Amiga 2000.
5. You will then see A64's title screen. There will be a brief pause while A64 loads its files. After it's done loading, you will see some text scrolling across the bottom of the screen. You should read all of this text before proceeding.
6. Press the left mouse button to enter A64.

You have now turned your Amiga into a Commodore 64. You can do almost everything in A64 that you would do on a C64. You can try all the Basic commands or POKE and PEEK around memory. It's all there.

Before moving on, we would like to introduce you to A64 Prefs. While running A64, if you press the two ALT keys on your keyboard at the same time, you will be put into A64 Prefs. A64 Prefs is an integrated part of A64. When you enter A64 Prefs, A64 will pause the C64 program that is currently running and a title bar will be displayed over the A64 screen. A64 Prefs serves many purposes. It allows you to quit or reset A64, to change A64's performance and appearance and to access the WorkBench and any other programs that you may want to use concurrently with A64. All of the features that are available while in A64 Prefs are explained further in this manual. To exit back to A64, select Exit in the A64 menu.

Installing the C64 ROMs

NOTE: This section contains very valuable information concerning A64's compatibility with the C64. Please read this section before attempting to run any C64 programs with A64.

When you run A64, you will notice that A64's Basic startup screen is different from a real C64's in that the startup message will say:

```
"*** QUESTRONIX A64 BASIC V2 ***"
```

instead of:

```
"**** COMMODORE 64 BASIC V2 ****"
```

This is because the Commodore 64 ROMs are not present in A64 (the C64 ROMs are two computer chips in the C64 that contain the C64's operating system). The C64 ROMs are not included with A64 because they are copyrighted by Commodore. A64 contains a complete C64 ROM emulation which allows A64 to run most C64 programs without the C64's ROMs. A64's ROM emulation is very compatible with the C64 ROMs, but there may be some C64 programs that will not run because of the lack of the real C64 ROMs in A64. We do provide a simple solution to this problem. By using The A64 Packages utilities, and by running a Basic program on a real C64, you can install the C64 ROMs on your A64 disk. This will improve A64's compatibility with the C64.

To install the C64 ROMs on your A64 disk you will need the following:

1. Access to either a real C64, C128 (running in C64 mode) or a SX-64
2. A C64 disk drive (i.e., 1541)
3. The A64 Package software and hardware
4. A 5 1/4" disk, formatted on a C64, with at least 20k free
   NOTE: You can format C64 disks on the Amiga using the 64Cmd utility that is included with The A64 Package.

The following procedure outlines the installation of the C64 ROMs. Part of the procedure has to be done on a real C64 and the rest must be done on the Amiga.

The first thing you need to do is copy the C64 program "saveroms" from the A64 disk to a C64 disk. This program is used to save the Commodore 64's ROMs to a C64 disk.

1. With A64's hardware interface already installed, run the program A64Tools (A64Tools can be found on disk #2).
2. Select the Source Device Type gadget to change it to Amiga.
3. For the Source Device Name, enter the following:

   ```
   A64:TAP/64Prgs [RETURN]
   ```

4. You will then be prompted to insert volume A64: into any drive. Insert the A64 disk (disk #1) into any drive.
5. You will then see the file named "saveroms" listed in the source file area. Click on the file name to highlight it.
6. Select the Destination Device Type gadget to change it to C64.
7. Insert the formatted 5 1/4" disk in your C64 drive.
8. Select the Start gadget to transfer the program "saveroms" to the C64 disk.
9. Select Quit in the A64Tools menu to quit A64Tools.

The next part of the procedure involves running the program "saveroms" on a real C64.

1. Either unplug the C64 disk drive from the interface, or use another C64 disk drive and connect it to a real C64.
2. Insert the disk containing the program "saveroms" in the C64 disk drive and then type the following:

   ```
   LOAD "SAVEROMS",8 [RETURN]
   ```

3. After the program is done loading, type:

   ```
   RUN [RETURN]
   ```

4. You will then be prompted to insert a disk into drive 8 and press any key when ready.
5. After pressing any key, the C64 ROMs will be saved to the C64 disk

Now that the C64's ROMs are saved to a C64 disk you need to transfer them to your A64 disk.

1. If you need to reconnect the C64 disk drive to your Amiga with A64's hardware interface, do so.
2. Insert the disk containing the C64 ROMs into the C64 drive.
3. Run A64Tools again.
4. Select Strip Load Address in the Conversions menu (this is very important!).
5. For the Destination Device Name, enter the following:

   ```
   A64:TAP/Data [RETURN]
   ```

6. Select the Start gadget to transfer the C64 ROMs to your A64 disk.
7. Select Quit in the A64Tools menu to quit A64Tools.

The C64 ROMs are now installed on your A64 disk and they will be used by A64 automatically. You can verify that the entire procedure went ok by looking at the Basic startup message in A64. It should now read:

```
"**** COMMODORE 64 BASIC V2 ****"
```

instead of:

```
"*** QUESTRONIX A64 BASIC V2 ***"
```

Installing the C64 ROMs does not automatically replace A64's ROM emulation. The C64 ROMs are used to improve A64's ROM emulation. Once the C64 ROMs are installed you can disable A64's ROM emulation with A64 Prefs (*See Chapter 4, A64 ROMs, in the Code Menu section*), but disabling A64's ROM emulation will not improve compatibility and it will slow down many programs. The only reason that you might want to disable A64's ROM emulation is if you are running low on memory.

Chapter 3
A64

A64 is a complete Commodore 64 emulator. Sound, graphics, Basic and machine language are all emulated. A64 also fully integrates the C64 with the Amiga by allowing you to use Amiga devices while running C64 software. A64 supports Amiga drives (including hard disks and RAM drives), Amiga printers and Amiga modems.

A64's Two Modes of Operation

A64 has two modes of operation: Pause Mode and C64 Mode. These modes are explained as follows:

Pause Mode

A64 is in Pause Mode whenever you are in A64 Prefs or in A64Mon. While in Pause Mode, you have complete access to the entire Amiga system and multitasking is completely functional. When in Pause Mode, the current C64 program that A64 is running will be paused.

C64 Mode

A64 is in C64 Mode whenever you exit to the C64 environment. That means whenever A64 is actually running a C64 program, A64 is in C64 Mode. While in C64 Mode, you do not have access to the Amiga system, including A64 Prefs. To get access to A64 Prefs and the WorkBench, you must put A64 into Pause Mode by pressing the two ALT keys simultaneously. While A64 is in C64 Mode, multitasking is not disabled and any programs that were running when C64 Mode was entered will still be running, however, A64 does control a substantial amount of the Amiga system while in C64 Mode and most programs will just sit idle in memory.

C64 Mode also has an extended mode called "SuperState Mode." While in SuperState Mode multitasking will be totally disabled and all programs that are running (except for A64) will be stopped. SuperState Mode allows A64 to run slightly faster, but disallows the use of Amiga drives and printers from within A64. You can turn SuperState Mode on and off in A64 Prefs.

A64 and Multitasking

A64 is fully multitasking compatible. A64 will work with any programs that follows the rules of multitasking programming on the Amiga. The only programs that A64 may not work with are programs that take over some part or all of your Amiga, like some games and some other emulators. Which brings us to our next point. A64 can not multitask itself. Even though A64 is capable of multitasking, it still takes over system resources that can not be shared with other programs. So one copy of A64 can not share these resources with another copy of A64.

A64 and Memory

A64 is a very memory hungry program and in its full capacity, it can easily use over 3mb of your Amiga's memory. And under normal operations, it will require close to 1mb of free memory. If you only have 1mb of RAM on your Amiga you will have to make sure that all unnecessary programs are removed from memory before running A64. If you experience problems getting A64 to run because of a lack of free memory, it is still possible to run A64. If you install the C64 ROMs on your A64 disk (*See Chapter 2, Installing The C64 ROMs*), A64 will be able to run without using its ROM emulation and this will free about 180k of memory, which should be enough to get A64 to work. A64 will automatically not use its ROM emulation when you are low on memory and if you have the C64 ROMs installed. A64, not being able to use its ROM emulation, has its costs. The speed of Basic programs will be cut in half. We do recommend at least 2mb of RAM for A64 and with this amount of RAM you will be able to take advantage of all of A64's features, except the larger RAM expansion sizes.

680x0 Support

All of the programs in The A64 Package are compatible with all current microprocessors. They have been tested with the 68000, 68010, 68020, 68030 and 68040. No special handling is required to use any of these processors, except for the 68040. When using the 68040, Copy Back Mode must be disabled.

A64's Speed

The speed of A64 can vary greatly from program to program and from one Amiga to another. On 68000 based Amigas, A64 is not capable of running all programs at a usable speed. And on some accelerated Amigas, speeds of over 300% are easily reached with many programs. We highly recommend using A64 on an accelerated Amiga, especially if you are mainly interested in playing C64 games. This does not mean that you must have an accelerated Amiga to use A64. A64 is quite capable of running many C64 programs at usable speeds on 68000 based Amigas, but the difference in speed on an accelerated Amiga must be seen to be appreciated and you will be able to run more programs on an accelerated Amiga. Hopefully with the release of the A1200 and its relatively inexpensive accelerators, more people will be able to take advantage of using A64 on a "fast Amiga."

How to Speed Up A64

Every attempt has been made to make A64 as fast as possible. A64 is written entirely in assembly language for the fastest possible performance, but on 68000 based Amigas, A64 is still not capable of running all programs at usable speeds. In the sections of this manual on A64 Prefs, there are many references to A64 Prefs settings that have a direct effect on A64's speed. If you have a program that is running too slow, the first thing you should do is go over the A64 Prefs settings to see what settings might help speed up the program. Sometimes changing just one or two settings can mean the difference between a program that is unusable and one that runs very well. Also, there is a utility program included with The A64 Package, called "Convert," that is specifically designed to speed up C64 programs. The Convert utility can make most programs run much faster than they normally would.

How to Slow Down A64

When using A64 on an accelerated Amiga, you will encounter some programs that run much faster than the C64. In most cases this will not be a problem. Most games will not run over 100% speed no matter how fast your Amiga is (the reason for this is very technical). The programs that are more likely to run much faster than the C64 are Basic programs, utility programs and business programs and in most cases the extra speed is appreciated. If you do encounter a program that runs too fast, there are ways to slow down A64. In the sections of this manual on A64 Prefs, there are many references to A64 Prefs settings that have a direct effect on A64's speed. When describing these settings, emphasis is usually put on speeding up A64, but most of these settings can also be used to slow down A64. By changing the settings to their slowest settings you can usually slow down A64 enough where the program is usable. You can also run the program "NoFastMem," that is in the System drawer on your WorkBench before running A64. This will dramatically slow down A64 on Amigas with 32-bit Fast memory.

Incompatibilities

A great deal of time has been spent trying to make A64 as compatible as possible. Unfortunately, the only way to get 100% compatibility with a C64 is to use a C64. Listed below are the things that A64 does not currently handle or has problems with.

Fast Loaders

The biggest problem with compatibility is related to disk I/O. We all know how slow the 1541 disk drive is. Many software manufactures try to correct this problem by writing custom disk drive routines, called "Fast Loaders." Fast Loaders are designed to do what their name implies, load (or save) programs at a faster rate. These routines are extremely time critical and must be run at exactly 100% speed, with no interruptions, to function properly. This is a very difficult feat to accomplish in a computer emulation like A64. To complicate things even further, Fast Loaders vary from program to program. A Fast Loader that works with one program will not work with another. The only way to emulate Fast Loaders is to write a custom loader for each program. There are literally hundreds of Fast Loaders out there and it would be impossible to emulate all of them.

There is no way to tell if a program uses a Fast Loader until you try to load it. The most likely result, when trying to load a program that uses a Fast Loader, is the C64 program will stop and disk drive motor and/or light will stay on. It is also possible that the program may appear to finish loading, but it doesn't run correctly. If you encounter a program that uses a Fast Loader (most people have at least a couple of programs that use a Fast Loader) you must reset A64 and the disk drive to return everything back to normal.

A64 V3 does support some Fast Loaders using Patch Files (*See Chapter 4, Patch Files, in the Code Menu section*) and in the future we will make more Patch Files available so A64 can use more programs with Fast Loaders. If you have a C64 program that A64 does not run, send us a letter or Error Report and if possible, a copy of the program. We will try to support the most requested programs.

Cassette Port

The C64's cassette port is not supported in A64.

User Port

The C64's user port is not supported in A64, but A64 does support Amiga modems to emulate C64 modems. C64 modems are normally connected to the C64's user port (*See Chapter 4, Using Amiga Modems, in the System Menu section*).

Cartridge Port

The C64's cartridge port is not supported, but many cartridge based programs that have been converted to run from disk will work with A64.

Tips on Using A64

This section contains information that will make A64 easier to use and it is designed to help eliminate some of the most common problems.

- You should never attempt to enter A64 Prefs while I/O (Input/Output) is in progress. This means that whenever A64 is loading or saving from a disk drive, accessing a modem or using a printer, you should wait until the operation is completed before entering A64 Prefs. This warning should especially be noted when accessing C64 devices. Bringing up A64 Prefs during I/O will not crash A64, but may result in a C64 crash, forcing you to reset A64 and/or the C64 devices that you interrupted.

- In between running different C64 programs, you should always reset A64 with the Reset submenu in the A64 menu. This will ensure that everything within A64 is prepared for a new program. On the C64, you can type SYS 64738 to reset. You can also do this within A64, but the Reset submenu does more than the SYS 64738 and it is the preferred method for resetting A64.

- In between running different C64 programs, you should reset your C64 disk drive(s).

  You reset a disk drive by turning it off, waiting a couple of seconds and then turning it on again. Make sure that when you reset the drive that there is no disk in it. Resetting the drive with a disk in it can damage the disk. There are a couple of reasons why it is important to reset the disk drive. If you use a C64 program that uses copy protection or a fast loader, the C64 program can leave the drive in a state where it can't be used by any other C64 programs until it is reset. Another very common problem is with the familiar method of loading a C64 program with the command: LOAD"*",8,1. If you load a program with LOAD"*",8,1 and then switch disks and attempt to load a second program with LOAD"*",8,1 the second load will most likely fail. This is because the C64 drive interprets the name "*" as "*the last file accessed*" and if there is no last file, then it is interpreted as "*the first file on the disk.*" On the second use of LOAD"*",8,1 the C64 drive will attempt to load the last program you loaded, which is on a different disk, so of course the load will fail and the drive will respond with a "FILE NOT FOUND" error. You would experience these same problems on a real C64, if the C64 had the ability to reset any C64 program. Remember, on a real C64 you have to turn the C64 off to get out of many programs. When the C64 is turned off and then on again, any connected disk drives are also reset. In A64 you can reset any C64 program without turning your Amiga off and then on again, but this doesn't reset any disk drives that are connected. So you have to reset them manually by turning them off and on again. You can also experience problems with C64 printers if they are not reset after running some programs, but problems with C64 printers are not as common.

- If you ever experience problems loading from a C64 disk drive you should reset the drive (as outlined above). In many cases, this will eliminate the problem.

- When you reboot your Amiga, any connected C64 disk drives and printers will be reset. But, it is very impractical (and unnecessary) to reboot your Amiga in between running C64 programs.

Chapter 4

A64 Prefs

A64 Prefs is an integrated part of A64. You access A64 Prefs from within A64 by pressing the two ALT keys on your keyboard at the same time. When you enter A64 Prefs, A64 will pause the C64 program that is currently running and a title bar will be displayed over the A64 screen. A64 Prefs serves many purposes. It allows you to quit or reset A64, to change A64's performance and appearance and to access the WorkBench and any other programs that you may want to use concurrently with A64. A64 Prefs effects the operation of A64 much like the WorkBench Preferences programs effect the operation of the Amiga.

A64 Prefs Settings

A64 Prefs has many settings that you can change in order to customize A64 to run one or more C64 programs. With these settings, you can tell A64 what type of disk drives or printers to use, or how fast to draw the screen, or how many colors to use and much more.

All of A64 Prefs settings can be changed with menu options and requesters. These menu options and requesters follow some basic conventions:

- When a menu option is followed by the » symbol, it has a related submenu.
- When a menu option is followed by an ellipsis (...), it opens a requester or screen. Basically, the presence of an ellipsis means that more input will be required when that menu option is selected.
- When neither the » symbol or ellipsis is present, that menu option is a checkmark option. This type of option implies an on or off status. When the checkmark is visible, the setting is on or enabled and when the checkmark is missing, the setting is off or disabled.
- On all requesters that allow you to change A64 Prefs settings there will be an "Ok" or "Yes" gadget and a "Cancel" or "No" gadget. Selecting the "Ok" or "Yes" gadget will cause the operation to be carried out and if you made any changes to the settings they will be preserved. Selecting the "Cancel" or "No" gadget will abort the operation and any changes that you made to the settings will be ignored.
- Requesters that do not require a yes or no answer will only have one gadget. This gadget will be labelled "Continue" or "Ok" and selecting it will allow you to continue.

A64 Prefs Menus

When describing the A64 Prefs menu options you will see the name of each menu option as it appears on the A64 Prefs screen. In this manual, some of the menu options are either followed or proceeded by one or more words contained in parentheses. These words do not appear on the actual A64 Prefs screen, but are listed in the manual to help clarify the function of some menu options.

A64 Menu

Reset

Clear Map

Key Sequence: None

The Clear Map menu option is used to tell A64 whether or not to clear the C64's memory map when A64 is reset. This option does not perform the actual reset. It tells A64 how to reset when the Reset A64 menu option is selected.

When The Clear Map option is enabled, A64's reset will clear the C64's memory map. This is useful because some C64 programs will not run correctly if the memory map is not cleared. When this option is disabled, A64's reset will be non-destructive. A64 will not clear the C64's memory map and any programs that are in memory when A64 is reset will still be there after the reset.

NOTES

- Whether the Clear Map option is enabled or not, A64 does not clear out the memory in the RAM expansion emulation (*See Chapter 4, Ram Expansion, in the System Menu section*).

Reset (A64)

Key Sequence: [RIGHT AMIGA] [R]

The Reset A64 menu option is used to reset A64 back to the Basic startup screen (*See Chapter 3, Tips On Using A64, for important information on resetting A64*).

Config (Files)

All of A64 Prefs settings can be saved in a "Config File." When using any particular program with A64, you can set all of the A64 Prefs settings so that A64 will be able to run that program in the best way possible. You can then save the settings in a Config File so they can be re-loaded every time you use that program. This will keep you from having to figure out the best settings for a program every time you use it. You can set the settings once and then forget them. You will probably want to create individual Config Files for your most used C64 programs.

Whenever you run A64, it loads a Config File for its initial A64 Prefs settings. A64 normally loads the file named "A64.config" for the initial A64 Prefs settings, but you can make A64 load any Config File by double-clicking on a Config File icon. When you double-click on a Config File icon, A64 will automatically be started and the settings in the Config File will be used.

Config Files are a valuable feature in A64. Not only do they allow you to customize A64 to run a C64 program, but they can also be used to start A64 and automatically run a C64 program (*See Chapter 4, AutoLoading Files, in the Code Menu section*).

Load (Config)

Key Sequence: [RIGHT AMIGA] [L]

The Load Config menu option allows you to load a previously saved Config File.

When the Load Config menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Config File that you want to load.

Save (Config)

Key Sequence: [RIGHT AMIGA] [S]

The Save Config menu option allows you to save the current A64 Prefs settings in a Config File so they can be loaded and used by A64 at any time.

When the Save Config menu option is selected, you will be prompted with a file requester. With the file requester, you can select a displayed name or enter the name of the Config File that you want to save.

When A64 saves a Config File, it also creates an icon for the file. You can run A64 by double-clicking on this icon.

NOTES

- When saving a Config File, it is recommended that you end your file names with ".config" A64 uses this naming convention whenever referencing a Config File and it will help you to identify files if you use this convention as well.
- When using file requesters in A64 Prefs, it is better to use specific disk names instead of device names when saving files (ie. A64: instead of df0:). The names used in file requesters are just like all other A64 Prefs settings and they are saved whenever you save a Config File. Using generic device names instead of actual disk names can create problems with files not being found, especially when using A64's AutoLoad features (*See Chapter 4, AutoLoading Files in the Code Menu section*).

Reset (A64 Prefs)

Key Sequence: None

The Reset A64 Prefs menu option allows you to reset all of the A64 Prefs settings back to their default states.

About (A64)

Key Sequence: [RIGHT AMIGA] [A]

The About A64 menu option allows you to view some information about A64, like the current version number.

When the About A64 menu option is selected, you will be shown a screen with a cycling blue background and a requester displaying some information about A64.

NOTES

- When using extreme overscan on your Workbench screen it is possible for the Ok gadget not to function correctly. If this happens to you, there is a very simple solution:

1. Press [LEFT AMIGA] [N]. This will switch the Workbench screen to the front.
2. Press [LEFT AMIGA] [M]. This will switch the About screen back to the front, but now the cycling blue background will be gone and the requester colors will be incorrect.
3. You can now select the Ok gadget normally.

Quit (A64)

Key Sequence: [RIGHT AMIGA] [Q]

The Quit A64 menu option allows you to quit A64.

Exit (A64 Prefs)

Key Sequence: [RIGHT AMIGA] [X]

The Exit A64 Prefs menu option allows you to exit A64 Prefs back to A64.

When the Exit A64 Prefs menu option is selected you will return to A64 at the exact point in which you left it. If you select this option with the key sequence method, you will not exit A64 Prefs until all keys are released on the keyboard.

Once in A64 you can return to A64 Prefs by pressing the two ALT keys on your keyboard at the same time.

System Menu

Parallel Port

Key Sequence: None

The Parallel Port menu option allows you to select how A64 uses the Amiga's parallel port.

A64 can use the parallel port with either its hardware interface or with some other device like an Amiga printer, but it can only use one or the other at any one time.

When the Parallel Port menu option is selected, A64 will be able to use the parallel port for its hardware interface. When this option is unselected, A64 will be able to use the parallel port for other devices.

NOTES

- The Parallel Port setting is only in effect while A64 is in C64 Mode. While in Pause Mode, the parallel port is free to be used by any program.

VERY IMPORTANT WARNING!!!

There is a bug in version 1.3 of the Amiga's operating system that directly effects the Parallel Port setting. If you have the Parallel Port setting selected, and then attempt to use the parallel port for any purpose other than A64 using its hardware interface, your Amiga will crash! If you are using V1.3 of the OS and if you have the Parallel Port setting selected, you must be very careful not to make any program attempt to use the parallel port. This bug has been fixed in V2.0 and higher of the operating system, so users of these versions can ignore this warning. This bug may also be present in V1.2, but it has not been officially tested.

Serial Port

Key Sequence: None

The Serial Port menu option allows you to select how A64 uses the Amiga's serial port.

A64 can use the serial port for either emulating a C64 modem with an Amiga modem or for using some other device like an Amiga printer, but it can only do one or the other at any one time.

When the Serial Port menu option is selected, A64 will be able to use the serial port to emulate a C64 modem. When this option is unselected, A64 will be able to use the serial port for other purposes.

NOTES

- Unlike the handling of the parallel port, A64 will always control the serial port whenever the Serial Port setting is selected. As long as this setting is selected, you will not be able to use the serial port with any other programs, whether A64 is in Pause Mode or not.

Using Amiga Modems with A64

A64 supports Amiga modems for emulating C64 modems. To use an Amiga modem you must select the Serial Port setting and then run a C64 telecommunications program. The C64 program controls all telecommunications settings (ie. baud rate, parity and stop bits) and to change these settings, it must be done within the C64 program. A64 supports all baud rates from 110 to 19200, but you may find that baud rates over 2400 will significantly slow down A64 on an unaccelerated Amiga.

Amiga modems are very different from most C64 modems. Most Amiga modems are "Hayes Compatible" and use specific commands for things like auto-dial. Any C64 telecommunications program that you want to use with A64 should have the ability to change the type of modem you are using to Hayes Compatible or something similar. If the C64 program does not support Hayes Compatible modems, features like auto-dial may not work with A64. You will still be able to use the program, but you will have to dial the phone number manually or by typing modem commands (see your modems manual for more information on dialing commands).

Drives

Key Sequence: [RIGHT AMIGA] [D]

The Drives menu option allows you to change the type of disk drives you are using with A64.

Just like the C64, A64 allows you to utilize up to four separate disk drives at any one time. A64 supports all C64 compatible serial drives (ie. 1541, 1571, 1581 and clones) and Amiga drives, including floppy drives, hard disks and RAM drives.

When the Drives menu option is selected, you will be prompted with a requester. The Drives requester contains the following gadgets:

Drive Gadgets

The C64 uses device numbers to access its drives, numbered 8-11. These device numbers are listed on the left side of the Drives requester. To the right of each device number are two gadgets. These two gadgets allow you to set what drive you want A64 to use for that device number.

The first gadget (going left to right) is called the "Drive Type gadget" and it allows you to change the type of drive that A64 will use for that device number. You can change this gadget to either C64 or Amiga and each setting corresponds to the type of drive you want A64 to use.

The second gadget is called the "Drive Name gadget" and it is only used when you have the Drive Type gadget set to Amiga. Whenever the drive type is set to C64, the Drive Name gadget will be disabled. The Drive Name gadget allows you to enter the name of a drive or drawer that you want A64 to use for that device number. Some examples of drive and drawer names are: DF0:, RAM:, DH0:, A64: and A64:TAP/64Prgs. You can also add additional path information while in C64 Mode. For example, if you set drive 8 as an Amiga drive and the Drive name was set to A64:, you could type the following while in C64 mode:

```
LOAD"TAP/64PRGS/TEST",8 [RETURN]
```

This would load the file "TEST" from "A64:TAP/64PRGS" (note: this file does not exist and is only used as an example).

Work Dir Gadget

For certain functions involving Amiga drives, A64 needs to create temporary files. A64 creates these temporary files when loading the directory of an Amiga disk and when reading a drive error status. The Work Dir gadget is used to tell A64 where to put these files. To change the Work Dir setting, click on the gadget and then enter the name of the directory where you want A64 to put these files. We recommend using a RAM disk for these files.

Using Amiga Drives with A64

A64 supports all Amiga drives, including floppy drives, hard disks and RAM drives. When using Amiga drives, A64 expects the disks to be formatted as Amiga disks. A64 does not support reading or writing C64 formatted disks with Amiga drives (ie. 1581 disks on a 3 1/2" Amiga drive or 1541 disks on a 5 1/4" Amiga drive).

C64 drives are very different than Amiga drives. C64 drives are "intelligent devices." They have ROM and RAM (memory) giving them the ability to run programs, basically they are computers. This ability is used for many things like: reading drive errors, formatting disks, deleting, copying and renaming files, copy protection and fast loaders. Amiga drives are not intelligent devices and they must rely on the Amiga for most actions. Because of this, there are limitations when using Amiga drives with A64.

While using Amiga drives, A64 supports loading and saving all C64 file formats except relative files (REL), reading disk directories and reading the drives error status. Some of things that are not supported are: drive commands (ie. formatting disks, scratching, copying and renaming files), copy protection and fast loaders. While it may seem like there are many limitations to using Amiga drives you will find that a good percentage of non-copy protected software can be used successfully with Amiga drives (*It is possible to load many copy protected programs from Amiga drives by using A64's Save Map and Load Map features. See Chapter 4, Load Map and Save Map, in the Code Menu section*).

None of the limitations with using Amiga drives, excluding fast loaders (*See Chapter 3, Fast Loaders, in the Incompatibilities section*), exist while using C64 drives. If you have a program that will not load or work correctly with an Amiga drive, the program can still be used with a C64 drive.

NOTES

- You can not use Amiga drives while A64 is running in SuperState mode (*See Chapter 4, SuperState, in the System Menu section*).
- You should not use colons ':' or slashes '/' in C64 file names when using Amiga drives. These characters are used by the Amiga for separating disk, drawer and file names.
- When reading the directory from an Amiga disk, you can not use Amiga path names in front of the '$' symbol. For example:

  ```
  LOAD"A64:TAP/$",8 [RETURN]
  ```

  This would result in a file not found error. To load the directory, you have to set the Drive Name gadget so it contains the full path name. In this example, this would be "A64:TAP". You would then be able to read the directory be typing:

  ```
  LOAD"$",8 [RETURN]
  ```

  Note that no part of the Amiga path name is entered before the '$'.

Printers

Key Sequence: [RIGHT AMIGA] [P]

The Printers menu option allows you to change the type of printers you are using with A64.

Just like the C64, A64 allows you to utilize up to two separate printers at any one time. A64 supports all C64 compatible serial printers (ie. MPS series) and all Amiga printers.

When the Printers menu option is selected, you will be prompted with a requester. The Printers requester contains the following gadgets:

Printers Gadgets

The C64 uses device numbers to access its printers, numbered 4 & 5. These device numbers are listed on the left side of the Printers requester. To the right of each device number are two gadgets. These two gadgets allow you to set what printer you want A64 to use for that device number.

The first gadget (going left to right) is called the "Printer Type gadget" and it allows you to change the type of printer that A64 will use for that device number. You can change this gadget to either C64 or Amiga and each setting corresponds to the type of printer you want A64 to use.

The second gadget is called the "Printer Name gadget" and it is only used when you have the Printer Type gadget set to Amiga. Whenever the printer type is set to C64, the Printer Name gadget will be disabled. The Printer Name gadget allows you to enter the name of a printer that you want A64 to use for that device number. Some examples of printer names are: PAR:, SER: and PRT:.

Convert CRs Gadget

The C64 uses a different character than the Amiga to end a line of text that is sent to a printer. The C64 uses a Carriage Return (CR) and the Amiga uses a Line Feed (LF). When the Convert CRs gadget is selected, A64 will convert all Carriage Returns to Line Feeds that are being sent to an Amiga printer (this setting has no effect when using C64 printers).

Convert ASCIIs Gadget

The C64 uses a different way of representing characters than the Amiga. The C64 uses Pet ASCII to represent characters while the Amiga uses normal ASCII. When the Convert ASCIIs gadget is selected, A64 will convert all characters being sent to an Amiga printer from Pet ASCII to normal ASCII (this setting has no effect when using C64 printers).

Using Amiga Printers with A64

A64 supports all Amiga printers, but with most C64 programs, Amiga printers can only be used for printing text. To do graphics printing you must use a real C64 printer.

Using an Amiga printer with A64 can cause conflicts for the use of the parallel and/or serial ports. To avoid this, make sure when using Amiga printers that you set the Parallel Port and Serial Port settings appropriately (*See Chapter 4, Parallel Port and Serial Port, in the System Menu section*).

NOTES

- You can not use Amiga printers while A64 is running in SuperState mode (*See*

Chapter *4, SuperState, in the System Menu section*).

- Printer output can be sent to an Amiga file instead of a printer by setting the Printer Type to Amiga and the Printer Name to any valid Amiga file name. This feature can be useful for Listing C64 Basic programs to a file so they can be converted to Amiga Basic.

Game Ports (Joysticks)

Key Sequence: [RIGHT AMIGA] [G]

The Game Ports menu option allows you to change the input devices you are using with A64. When the Game Ports menu option is selected, you will be prompted with a requester. The Game Ports requester contains the following gadgets:

NOTE: In the following explanations, reference is made to each port in the following manner: port 1 means the port that your Amiga's mouse is normally plugged into and port 2 means the other port. By default, port 1 on the Amiga corresponds to port 1 on the C64 and port 2 on the Amiga corresponds to port 2 on the C64.

Swap Ports Gadget

The Swap Ports gadget allows you to swap the Amiga's game ports. When this option is selected, port 1 will act as port 2 and port 2 will act as port 1. For example, if you are using a C64 program that requires a joystick in port 1, instead of unplugging your Amiga's mouse from port 1, you can plug the joystick into port 2 and select the Swap Ports setting. A64 will now read the joystick as if it was plugged into port 1. With proper use, the swap ports option will allow you to use almost any C64 program and not have to unplug your Amiga's mouse.

Port 1 and Port 2 Gadgets

The Port 1 and Port 2 gadgets allow you to change the input device that A64 will use for that C64 port. These gadgets always correspond to the C64's game ports and not the Amiga's, regardless of the status of the Swap Ports setting. This means if a C64 program requires a joystick in port 1, you must set the Port 1 gadget to Joysticks, even if the Swap Ports setting is enabled and the joystick is actually plugged into the Amiga's port 2.

The Port Gadgets have two settings:

- Joystick: Selects a joystick for the corresponding C64 port.
- No Device: Acts as if there is no device connected regardless of what is actually plugged in.

NOTES

- While the Port 1 gadget is set to Joysticks and the Swap Ports setting is disabled, the Amiga's mouse will interfere with the keyboard and characters will appear on the screen when the mouse is moved. This is normal. The same thing would happen if you plugged an Amiga mouse into port 1 of the C64. This can be avoided by setting the Port 1 gadget to No Device whenever you do not need to use the port. Because of the way that the Swap Ports setting works, this situation can also be created with a device plugged into the Amiga's game port 2.

Keyboard

Key Sequence: None

The Keyboard menu option allows you to change A64's keyboard layout.

A64 supports two separate keyboard layouts, one for the United States and one for Germany.

Keyboard Differences

There are some minor differences between the C64 and the Amiga keyboards. The way A64 maps the keyboard is pretty much "what you see is what you get." In most cases the legends on the keys represent their equivalent function. The keys that have a different function than their legends are:

For the US keyboard:

AMIGA KEY	C64 EQUIVALENT
ESC	RUN/STOP
BACKSPACE (BACKARROW)	DEL/INST
DEL	CLR/HOME
HELP	RESTORE
RIGHT AMIGA	RESTORE
LEFT AMIGA	COMMODORE
`~	BACK ARROW
\\|	£
[{	[(when shifted gives PI)
]}	] (when shifted gives shift @)

For the German keyboard:

AMIGA KEY	C64 EQUIVALENT
ESC	RUN/STOP
BACKSPACE (BACKARROW)	DEL/INST
DEL	CLR/HOME
HELP	RESTORE
RIGHT AMIGA	RESTORE
LEFT AMIGA	COMMODORE
`~	BACK ARROW
\\|	£
SHIFT 3	#
_	/

SHIFT /	?
Ü	@
Ö	: (when shifted gives [)
Ä	; (when shifted gives])
[(KEYPAD)	(
] (KEYPAD)	)

NOTES

- The key sequence [RUN/STOP] [RESTORE] that is used to stop C64 BASIC programs, can be done by pressing [ESC] [HELP] or [ESC] [RIGHT AMIGA]. On some Amiga models the key sequence [ESC] [HELP] may not work and the second method of [ESC] [RIGHT AMIGA] should be used. This problem has been seen on A1200's and it may be due to a bug in the keyboard hardware.

A Problem with the Keyboard

Because of differences in a small number of keys on the keyboards of the C64 and the Amiga, a minor problem can occur with certain key sequences. The problem is related to pressing one of the shift keys and one of the numeric or punctuation keys simultaneously. What will happen is that A64 will think you pressed a key that you didn't and an unwanted character will appear on the screen. If this happens, use the backspace key to delete the unwanted character. This problem is more apparent when using the left shift key then when using the right shift key and this problem can be eliminated by using the numeric keypad for shifted numeric keys. This problem is not a bug in A64. It is caused by the way that A64 must remap certain keys so the desired character appears when a key is pressed.

The Numeric Keypad

The numeric keypad on the Amiga keyboard is fully supported. Shifting the keys on the keypad will give the C64 equivalent and not the Amiga's. For example, using the key sequence [SHIFT] [2]:

In A64:
Pressing [SHIFT] [2] (2 key at top left of keyboard) produces [@].
Pressing [SHIFT] [2] (2 key on the numeric keypad) produces ["].

On the C64:
Pressing [SHIFT] [2] produces ["], the same as the keypad in A64.

Special Keys

There are a few keys that perform a special function when pressed. These keys and their functions are:

<u>KEYS</u>	<u>FUNCTION</u>
ALT, ALT	When the two ALT keys are pressed simultaneously from C64 Mode,A64 Prefs will be entered.
TAB	Forces A64 to redraw the screen. This can be done in both C64 Mode and in A64 Prefs.
F9	Enter A64Mon. This can be done in both C64 Mode and in A64 Prefs.
F10	Toggle title bar on and off. This can only be done in A64 Prefs.

SuperState

Key Sequence: None

The SuperState menu option allows you to switch SuperState Mode on and off.

A64 supports a mode that allows C64 programs to run slightly faster. This mode is called SuperState Mode. While A64 is running in SuperState Mode, C64 programs will run slightly faster, but you will be unable to use Amiga drives and printers (*See Chapter 3, A64's Two Modes Of Operation*).

CIA Rate

Key Sequence: None

The CIA Rate menu option allows you to control the speed of C64 CIA interrupts.

The C64 has two computer chips called Complex Interface Adapters (CIAs). These chips are used for many useful things; like reading the keyboard and joysticks, disk I/O and interrupts. The CIA Rate option allows you to control the rate at which CIA interrupts are generated and changing its setting can have dramatic effects on the speed of some C64 programs (to see the effect this option can have , try the different settings and watch what happens to the cursor speed on the Basic screen).

The CIA Rate menu option has three settings:

- **Normal**: When selected, CIA interrupts will happen at 100% speed.
- **Fast**: When selected, CIA interrupts will happen at 200% speed.
- **Slow**: When selected, CIA interrupts will happen at 50% speed.

NOTES

- When running C64 telecommunications programs, the CIA Rate option should be set to Normal (100%).
- Changing the CIA Rate setting to anything but Normal, will throw off the C64's software clock. This will not effect most programs, but keep this in mind if you are timing anything with A64.

RAM Expansion

Key Sequence: None

The RAM Expansion menu option allows you turn A64's RAM expansion emulation on and off and to change the amount of memory used for RAM expansion.

A64 emulates the standard C64 RAM expansion units (REUs), including the 1700 (128k), 1764 (256k) and 1750 (512k). A64 also emulates two unstandard size REUs, 1mb and 2mb. To use A64's RAM expansion emulation, you must have enough free Amiga memory for the size you select and have a C64 program that supports RAM expansion. Most C64 programs do not support RAM expansion, so they will not take advantage of the extra memory. The programs, that do support RAM expansion may not support all of the sizes that A64 does. This will be especially true of the 1mb and 2mb settings. Very few C64 programs will recognize RAM expansion sizes above 512k.

The RAM Expansion menu option has six settings:

- **Enabled**: When selected A64 will emulate the C64 RAM expansion. This setting must be selected for A64 to actually emulate the REU, all of the other settings change the size of the RAM expansion and they do not enable or disable it.
- **128k, 256k, 512k, 1mb** and **2mb**: When one of the size settings are selected A64 will use that size for the RAM expansion.

NOTES

- When the RAM expansion is enabled, the number of bytes free on the Basic startup screen will not change. C64 Basic does not directly support RAM expansion.

Graphics Menu

Colors

Key Sequence : [RIGHT AMIGA] [C]

The Colors menu option allows you to change the colors that A64 uses to draw the C64 screen.

To fully describe the Colors option and its features, we must briefly discuss the C64's graphics. It is beyond the scope of this manual to fully explain the graphics capabilities of the C64. If you need more information, there are many good books available.

The Commodore 64 is capable of displaying a maximum of 16 colors at any one time and it uses numbers to signify its colors. These numbers range from 0 to 15. The numbers and their corresponding colors are:

0 = BLACK	4 = PURPLE	8 = ORANGE	12 = MED GRAY
1 = WHITE	5 = GREEN	9 = BROWN	13 = LGT GREEN
2 = RED	6 = BLUE	10 = LGT RED	14 = LGT BLUE
3 = CYAN	7 = YELLOW	11 = DARK GRAY	15 = LGT GRAY

In addition to being able to display these 16 colors for graphics data, the C64 also has a number of color registers (memory locations) that control the color of specific parts of the screen. By changing the value in one of these registers, you can change the color of that part of the screen. For example, the C64 has a register that controls the border color. If you change the register to one of the values above, the border color would change to the corresponding color. From Basic, this would be done with a POKE command:

```
POKE 53280,0          would change the border color to black
```

There are five separate color registers that are used to draw the screen. The C64 does not always use all five, it depends on what graphics mode the C64 is currently using. The five color registers are:

- Border Color (53280)
- Background Color 0 (53281)
- Background Color 1 (53282)
- Background Color 2 (53283)
- Background Color 3 (53284)

When the Colors menu option is selected, a screen will open at the bottom of the A64 Prefs screen. The Colors screen contains the following gadgets:

Color Mode Gadget

The Color Mode gadget is located at the top, right hand side of the Colors screen. It is shown by the word "Colors" followed by a gadget containing a number. This gadget allows you to pick how many colors A64 will use to draw the C64 screen. The possible settings are: 2, 16 and 20. Each setting corresponds to the number of colors that A64 will use.

The drawing of the C64's screen is the most time consuming part of A64. The general rule is that the more graphics a C64 program uses, the more work A64 has to do to display them and the slower A64 will run the C64 program. You can help compensate for this extra work, by lowering the Color Mode setting and decreasing the number of colors that A64 can use to draw the screen. When you decrease the number of colors used to draw the C64 screen, you will allow A64 to run faster, but you will also create some limitations on how well A64 can represent the C64 screen. A64's Color Modes are:

20 Color Mode

20 Color Mode allows A64 to draw the C64 screen without any limitations. When using this setting, A64 will be able to draw the screen as close as possible to the C64s, but it is also the slowest Color Mode.

16 Color Mode

16 Color Mode allows A64 to draw the screen with only one limitation: Color 15 (normally Light Grey on the C64) is replaced with Color 1 (normally White on the C64). This means that anything that is supposed to be drawn in color 15 will be drawn in color 1. This limitation is minor compared to the speed increase that can be gained by using 16 Color Mode (16 Color Mode is A64's default setting). When using 16 Color Mode, most C64 programs will appear as they should and this limitation will go unnoticed.

2 Color Mode

2 Color Mode is a monochrome mode and has very obvious limitations. When using 2 Color Mode, all of the C64's 16 colors will be drawn in one color and there is a second color for the background color. 2 Color Mode is the fastest way for A64 to draw the screen, but very few programs will look acceptable when using it. 2 Color Mode is well suited for programs that do not require many colors.

Color Gadgets

The Color Gadgets are the 20 small gadgets that start at the top, left hand side of the Colors screen. These gadgets represent A64's current color palette, with each gadget containing one of A64's colors. The Color Gadgets allow you to select one of A64's colors so it can be manipulated. The current selected gadget will be displayed with a red square around it.

Each of A64's Color Modes has it own color palette that is represented by the Color Gadgets. All 20 Color Gadgets are not always used. When you lower the Color Mode, some gadgets will become disabled. The number of enabled Color Gadgets will always be equal to the number of colors set by the Color Mode gadget, which is also the number of colors that A64 has available for drawing the C64's screen. The Color Gadgets are used for the following:

- The first Color Gadget is used to represent the C64's border color (53280) or background color (53281), depending on whether or not you have Borders enabled in A64 Prefs (*See Chapter 4, Borders, in the Graphics Menu section*). If Borders are enabled, this gadget will display the color that A64 is currently using to draw the C64's border. If Borders are disabled, this gadget will display the color that A64 is currently using to draw the C64's background color.
- The next 16 Color Gadgets represent the C64's 16 colors, with the first gadget representing the C64's color 0 and the last gadget representing the C64's color 15. These gadgets are referred to as "the main Color Gadgets."
- The last three color gadgets are only used with A64's 20 Color Mode and they are used to represent the C64's other background color registers (53282-53284).

When using 16 and 20 Color Modes, the Color Gadgets that represent the C64's color registers (ie. the first and last three gadgets) behave differently from the main 16 Color Gadgets. When you change the color in one of these gadgets, the change will only stay in effect while in Pause Mode. When you exit back to A64, the color will be changed back to its original value. Basically, changing the colors in these gadgets is only useful when saving screens as IFF pictures (*See Chapter 4, Save Screen, in the Tools Menu section*).

Slider Gadgets

The Slider Gadgets are the three long, multicolored gadgets located below the Color Gadgets. These gadgets are used to change the color contained in the selected Color Gadget. On the Amiga, colors are formed by adding a combination of Red, Green and Blue color parts. The Slider Gadgets represent these three color parts with; Red on top, Green in the middle and Blue on the bottom. Each of these gadgets contain 16 different colors. By selecting one of the displayed colors, you will change the selected Color Gadget to that color. You can also move the Slider Gadgets to display a new set of colors. You can pick any one of the Amiga's 4096 possible colors by changing the settings of these three gadgets.

Reset Gadget

The Reset Gadget allows you to reset all of the colors to either their original or last values. The Reset Gadget effects all Color Modes. When the Reset Gadget is selected you will be prompted with a requester. The requester contains the following gadgets:

- **DEF**, selecting this gadget will reset all of the colors to their default settings.
- **LAST**, selecting this gadget will reset all of the colors to their last saved settings. This will either be the settings set when A64 was first loaded or if you have changed some of the colors and preserved them, it will be the preserved settings.

Spread Gadget

The Spread Gadget allows you to create a spread of colors between two Color Gadgets. You can only do a spread between the main 16 Color Gadgets. To create a color spread between two colors:

1. Select the first Color Gadget that you want to spread from
2. Select the Spread Gadget. The mouse pointer will change to an arrow with the word "TO" under it.
3. Select the second Color Gadget that you wish to spread to. Selecting any gadget, other than a Color Gadget, will abort the operation.

Exch Gadget

The Exchange Gadget allows you to exchange the colors between two Color Gadgets. To exchange the colors between two Color Gadgets:

1. Select the first Color Gadget that you want to exchange
2. Select the Exch Gadget. The mouse pointer will change to an arrow with the word "TO" under it.
3. Select the second Color Gadget that you wish to exchange with. Selecting any gadget, other than a Color Gadget, will abort the operation.

Copy Gadget

The Copy Gadget allows you to copy the color from one Color Gadget to another Color Gadget. To copy a color from one Color Gadget to another:

1. Select the first Color Gadget that you want to copy from
2. Select the Copy Gadget. The mouse pointer will change to an arrow with the word "TO" under it.
3. Select the second Color Gadget that you wish to copy to. Selecting any gadget, other than a Color Gadget, will abort the operation.

Undo Gadget

The Undo Gadget allows you to undo any changes you made to the Color Gadgets. Selecting the Undo Gadget will restore the entire color palette to its last saved state.

NOTES

- The Cancel gadget will not cancel out changes to the current Color Mode setting nor will it restore changes made to another Color Modes' colors if you selected "Keep Changes" when changing Color Modes.
- In almost all cases, the lower the Color Mode, the faster A64 will run. But, with certain programs that use the C64 graphics modes, Multicolored Text or Extended Background Color Text, A64 will run faster in 20 Color Mode. You can determine if a C64 program is using one of these graphics modes by using the VICDump command in A64Mon.

Borders

Key Sequence: [RIGHT AMIGA] [B]

The Borders menu option allows you to turn the C64's borders on and off.

When the Borders menu option is selected, A64 will display a border around the C64 screen. Leaving the border on will make the screen look exactly like the C64's, but it will also interfere with a couple of features related to C64 sprites. When borders are enabled, sprite to foreground collisions are disabled and sprite priorities are forced to the front (*See Chapter 4, Sprites, in the Graphics Menu section*). When this option is disabled, the limitations with sprites do not exists and in most cases A64 will able to draw the screen faster. Having a border around the C64 screen creates more work for A64 and this can help to slow down A64. For many C64 programs, a border is unnecessary and we recommend that you turn the border off whenever possible.

Hardware Scroll

Key Sequence: None

The Hardware Scroll menu option allows you to enable or disable the C64's hardware screen scrolling capabilities.

A64 emulates the C64's hardware screen scrolling capabilities, but because A64 may not be able to emulate the C64 program exactly like the C64, hardware scrolling can make the screen appear very jittery. Disabling this option will eliminate this jitter in the screen.

Screen Refresh

Key Sequence: None

The Screen Refresh menu option allows you to change the way and speed in which A64 draws the C64 screen.

When the Screen Refresh menu option is selected, you will be prompted with a requester. The Screen Refresh requester contains the following gadgets:

Refresh Type Gadget

A64 is capable of drawing the screen in two distinct ways and the Refresh Type Gadget allows you to switch between these two methods. This gadget has two settings, Direct and Sample. Each setting corresponds to one of the ways that A64 can draw the C64 screen, called drawing modes. A64's screen drawing modes are:

Direct Mode

When A64 is drawing the screen in Direct Mode, it writes to the Amiga's screen directly. This means, when the C64 is performing an operation that changes some part of the C64's screen, A64 will make the same change immediately to the Amiga's screen. This way of drawing directly to the Amiga's screen has the advantage of showing changes to the C64's screen immediately, resulting in smoother screen updates, but has the disadvantage of having the tendency to slow down A64 when many changes are happening to the C64's screen seemingly at one time. For example, scrolling in games. In most cases, however, Direct Mode is the faster method of redrawing the screen.

Sample Mode

When A64 is drawing the screen in Sample Mode, it draws the changes in the C64 screen at a set Sample rate. This means that A64 will check the C64 graphics at a certain rate to see what has changed and then draw the changes. This way of drawing the screen has the advantage of being faster when many changes are happening to the C64's screen seemingly at one time, but has the disadvantage of making the screen updates appear jerky.

Out of the two ways of drawing the screen, Sample Mode is the most compatible with the C64, but also the slower of the two methods. Sample Mode has the ability to handle up to 25 raster IRQs per screen, spaced as little as 1 scan line apart. Direct Mode can handle up to 25 raster IRQs per screen, spaced as little as 8 scan lines apart. Although Direct Mode has this limitation, it will handle most C64 programs with no problems. If you do encounter a C64 program where the screen does not look like it should, try switching to Sample Mode to correct it.

Hardware Rate Gadget

The Hardware Rate Gadget allows you to control the speed in which A64 updates graphics information that is hardware related. This includes things like: border and background colors, 24/25 row mode, 38/40 column mode and hardware screen scrolling.

The Harwdare Rate Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to how many times per second A64 will update the graphics hardware. This setting can have a significant impact on the speed of some C64 programs and by decreasing this setting, it will help to speed up some programs. This setting can also be set to 0 which will turn this setting off and stop A64 from updating the graphics hardware.

Data Rate Gadget

The Data Rate Gadget allows you to control the speed in which A64 draws changes to the C64 screen.

The Data Rate Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to how many times per second A64 will draw changes to the C64 screen. This setting can have a significant impact on the speed of some C64 programs and by decreasing this setting, it will help to speed up some programs. This setting can also be set to 0 which will turn this setting off and stop A64 from updating the screen.

Limit Refreshes

The Limit Refreshes Gadget allows you to limit A64's screen drawing.

A64 can spend a great deal of time drawing the C64 screen. With some programs, the amount of time spent drawing the screen can be so great that the C64 program will slow down to the point where it is unusable. The Limit Refreshes option provides a way for A64 to ignore some things that would normally cause A64 to redraw the entire screen. This option has limited uses and it will not work with all programs and with some programs, it may work too well and A64 will not draw the screen when it should. Basically, the only way to tell if this option

will be of any use in a particular program is to enable it and see what happens. Keep in mind, you can press the TAB key at anytime to force A64 to redraw the screen. This can be useful when used with the Limit Refreshes option to see what the screen should look like at any given point.

Rasters

Key Sequence: None

The Rasters menu option allows you to change some settings that are related to the C64's scan lines.

When the Rasters menu option is selected, you will be prompted with a requester. The Rasters requester contains the following gadgets:

Scan Line Read Gadget

Before explaining this setting, we need to explain some basic principals of video circuitry and how the C64 uses them. Inside your computer's monitor (or TV) is an electron beam that sweeps across the inside of the picture tube, refreshing the picture as it goes. This electron beam sweeps across the screen and refreshes the screen one horizontal line at a time. This horizontal line is called a scan line. The standard NTSC screen is made up of 262 of these scan lines. This process of refreshing the screen, one scan line at time, happens very rapidly. It takes your monitor 1/60th of a second to refresh one entire screen. This means that in one second the screen will have been refreshed 60 times.

Inside the C64 is a pair of registers (memory locations) that contain the current scan line number that is being refreshed by the monitor. These registers are used by MANY C64 programs for a variety of purposes. Seeing how these registers change at a regular rate, many C64 programs use them for timing. Another popular use for these registers is for a program to wait for a specific scan line before continuing.

A64 emulates the scan line registers in the C64 with three separate methods. Each of the methods has its advantages and disadvantages. The three methods are:

Smart

When the Scan Line Read Gadget is set to Smart, A64 will try and figure out what scan line the C64 program is actually waiting for and do its best so that the C64 program waits the shortest amount of time possible. This method has the advantage of allowing some programs to run much faster than they normally would, but has the disadvantage of slowing down C64 programs if A64 can not figure out what scan line the C64 program is waiting for. If A64 can not determine the scan line that the C64 program is waiting for, it will use the Fake method for the scan line value.

Real

When the Scan Line Read Gadget is set to Real, A64 will use the Amiga's hardware to get the current scan line. This method has the advantage of being able to emulate the C64's scan line registers in the closest way possible, but has the disadvantage of not working with all programs. Because of internal timing in the Amiga, not all scan lines can be read and it is possible for a C64 program to get in an endless loop trying to wait for a scan line that can not be read.

Fake

When the Scan Line Read Gadget is set to Fake, A64 will simulate the function of the scan line registers. This method has the advantage of working reliably with all C64 programs, but has the disadvantage of sometimes being very slow.

Scan Line Read Counter Gadget

The Scan Line Read Counter Gadget is used in conjunction with the Smart and Fake settings in the Scan Line Read Gadget. This option allows you to specify how many times A64 should read the simulated scan line before incrementing the value.

The Scan Line Read Counter Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to how many times the scan line registers can be read before their value is incremented. Changing this setting can have a tremendous effect on the speed of some C64 programs and a lowest possible setting is not always the best possible setting. Normal ranges for this setting are between 1 and 6.

How to Set the Scan Line Read Gadgets

We realize that the functions of the Scan Line Read Gadget and Read Counter Gadget can be confusing. They are very difficult to explain simply, but their importance can not be underemphasized. Many C64 programs use the scan line registers and these settings can have a tremendous effect on these programs. Here is a simple procedure for determining the best settings for these gadgets:

1. The first thing that needs to be done is to determine if the C64 program is using the C64's scan line registers. This can be done by setting the Scan Line Read Gadget to Fake and the Read Counter Gadget to its maximum setting of 255 and then exiting to A64. If the C64 program is using the scan line registers, you will see a significant decrease in the speed of the program. If you do not see any change in the speed of the program, then the C64 program is not using the scan line registers and the Scan Line Read settings will not have any effect on the C64 program.
2. If you did see a change in speed in the C64 program, then it is using the scan line registers and you should now try and find the best possible settings. Change the Read Counter Gadget back to a low number (1 or 2) and then change the Scan Line Read Gadget to Smart, exit back to A64 and make a note of the speed. Is there any difference? Is it faster or slower? Repeat this by using the Real setting and the Fake setting. For most programs, one method should be noticeably better than the others.
3. If you have determined that the best Raster Read method is either Smart or Fake, you should then try changing the Read Counter Gadget to see if it also has any effect on the speed. You should try values between 1 and 6. If you do not see any change in speed, or if the program seems to run slower with higher values, then leave the Read Counter at 1 or 2.

Round Scan Lines Gadget

The C64 has the ability to change its graphics modes many times during one screen. This technique is called "Raster Interrupts." A64 fully supports Raster Interrupts, but due to a difference in the C64's and Amiga's hardware, a problem can occur when running a small number of C64 programs. The problem can occur when a C64 program does hardware vertical scrolling within a Raster Interrupt. The result will be a couple of scan lines on the screen will contain garbage data. The Round Scan Lines setting is for correcting this problem.

It is really difficult to explain the Round Scan Lines setting without getting overly technical. Simply, if you are running a C64 program and parts of the screen do not look correct, try enabling this option to see if it corrects the problem. This setting can also cause some program's screens to appear totally wrong and in this case, this option should not be used.

Sprites

Key Sequence: None

The Sprites menu option allows you to change how A64 handles C64 sprites.

This section describes sprites. Sprites are independent graphic images that are not part of the screen in the normal sense. If you are unfamiliar with sprites please refer to one the many books available on C64 graphics (the C64's User Guide contains a good introduction to sprites).

A64 emulates the C64's sprites with Amiga sprites. Unfortunately, The C64's sprites and the Amiga's sprites are about as compatible as fire and water. Trying to get Amiga sprites to behave like C64 sprites is not an easy feat. One of the biggest problems that you will notice almost immediately is that the sprites will flicker. Because of size differences between Amiga and C64 sprites, there are not enough Amiga sprites to emulate all of the C64's sprites at one time. Because of this, Amiga sprites must be re-used, causing the flicker. The more C64 sprites that are used, and if they are expanded horizontally, the worse the flickering gets. There are also other problems created because of incompatibilities between Amiga sprites and C64 sprites. Sprite collisions and sprite priorities are not 100% compatible. Most of the problems with emulating C64 sprites are caused by restrictions in the Amiga's hardware. It's hard to believe, but there are some things a C64 can do that an an Amiga can't.

When the Sprites menu option is selected, you will be prompted with a requester. The Sprites requester contains the following gadgets:

Refresh Type Gadget

A64 is capable of drawing sprites in two distinct ways and the Refresh Type Gadget allows you to switch between these two methods. This gadget has two settings, Direct and Sample. Each setting corresponds to one of the ways that A64 can draw the C64 sprites, called drawing modes. These drawing modes are explained as follows:

Direct Mode

When A64 is drawing the sprites in Direct Mode, it writes to the Amiga's sprites directly. This means, when the C64 is performing an operation that changes a sprite, A64 make the same change immediately to the Amiga's sprites. This way of drawing directly to the Amiga's sprites has the advantage of showing changes to the C64's sprites immediately, but it can slow down the sprite emulation if many changes are being made to the C64's sprites.

Sample Mode

When A64 is drawing the sprites in Sample Mode, it draws the changes in the C64 sprites at a set Sample rate. This means that A64 will check the C64 sprites at a certain rate to see what has changed and then draw the changes. This way of drawing the sprites has the advantage of being faster when many changes are happening to the C64's sprites and it is also the most compatible of the two drawing modes.

Hardware Rate Gadget

The Hardware Rate Gadget allows you to control the speed in which A64 updates sprite information that is hardware related. This includes things like: colors, positioning and re-use (what causes flickering sprites).

The Harwdare Rate Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to how many times per second A64 will update the sprite hardware. You should note that decreasing this setting can increase the flickering problem in sprites. You should only decrease this setting when you are using a C64 program that does not use enough sprites to create the flicker. This setting can also be set to 0 which will turn this setting off and stop A64 from updating the sprite hardware.

Data Rate Gadget

The Data Rate Gadget allows you to control the speed in which A64 draws changes to the actual C64 sprite images.

The Data Rate Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to how many times per second A64 will draw changes to the C64 sprite images. This setting can have a significant impact on the speed of some C64 programs and by decreasing this setting, it will help to speed up some programs. This setting can also be set to 0 which will turn this setting off and stop A64 from updating sprite images.

Monochrome Gadget

The C64 has two types of sprites: multicolored sprites that use 4 colors and standard sprites that use 2 colors. The emulation of multicolored sprites can decrease the speed in which A64 runs some programs. The Monochrome Gadget allows A64 to use standard sprites in place of multicolored sprites. This will allow A64 to run some programs faster.

Sprite Collisions Gadget

The C64 is capable of detecting when two sprites collide with each other. The Sprite Collisions Gadget allows you to disable sprite to sprite collisions. This can be very useful for getting endless lives in many games.

Foreground Collisions Gadget

The C64 is capable of detecting when a sprite collides with some part of the foreground (ie. graphics data). The Foreground Collisions Gadget allows you to disable sprite to foreground collisions. This can be very useful for getting endless lives in some games.

NOTES

- While Borders are enabled, the foreground Collisions gadget will be disabled and all sprite to foreground collisions will not be detected by A64 (*See Chapter 4, Borders, in the Graphics Menu section*).

Priorities Gadget

The C64 is capable of displaying sprites either in front of the rest of the screen or behind it. This capability is called sprite priorities. Because of incompatibilities between the Amiga hardware and the C64 hardware, these priorities can not be emulated 100% accurately. The Priorities Gadget allows you to select how A64 handles these priorities. This gadget has three settings:

- **Normal**: A64 will do its best to emulate the C64's sprite priorities
- **Front**: A64 will display all sprites in front of the rest of the screen

- **Back**: A64 will display all sprites behind the rest of the screen

NOTES

- While Borders are enabled, the Priorites gadget will be disabled and all sprites will be displayed in front of the rest of the screen.

All Graphics

Key Sequence: None

The most time consuming part of A64 is the emulation of C64 graphics. The All Graphics menu option allows you to turn all of A64's graphics handling on and off. Disabling this setting has very limited uses, but it can be useful to temporarily turn off all graphics allowing a C64 program to perform some operation at the fastest possible speed. For example: if the program you're running does extensive calculations or if you want to listen to a program's sound at the fastest possible speed.

While using the Direct Mode settings (*See Chapter 4, Screen Refresh and Sprites, in the Graphics Menu section*) you may still see some changes to the C64's graphics and disabling this setting may not result in any noticeable speed increase.

This setting is not saved in Config Files and by default, A64's graphics are enabled. You should be careful when disabling all of A64's graphics. When all of A64's graphics are disabled it is easy to think that A64 has crashed because you will not see anything changing. The screen will not even be redrawn if you reset A64.

Sound Menu

(Sound) Enabled

Key Sequence: None

The Sound Enabled menu option allows you to turn A64's sound emulation on and off.

A64 fully emulates the C64's sound capabilities. The only aspect of the C64's sound that is not emulated is filtering. Filtering requires a great deal of processing time and it is not practical to emulate it with software. Not emulating filtering does not pose any problems for C64 programs and in most cases A64's sound will be exactly like the C64s.

Emulating the C64's sound capabilities takes some processing time away from A64. Even when a program is not producing any sounds a small amount of time is lost to the sound emulation. The Sound Enabled menu option allows you to turn off the sound emulation so it won't take any time away from A64.

LED Filter

Key Sequence: None

The LED Filter menu option allows you turn the Amiga's built-in sound filter on and off. The Amiga has a built-in sound filter in its hardware that is designed to eliminate noise from the Amiga's sound. Turning on this filter may reduce unwanted noise in A64's sound, but the C64 normally has a noisy sound quality and turning on this filter may make A64 sound less like a real C64.

NOTES

- The LED Filter setting has no effect on A64's speed.
- Some early A500s do not have the sound filter built in to their hardware and changing the LED Filter setting will not have any effect on the sound quality on these machines.

(Sound) Quality

Key Sequence: None

The Sound Quality menu option allows you to change the quality of A64's sound emulation.

When the Sound Quality menu option is selected, you will be prompted with a requester. The Sound Quality requester contains the following gadgets:

Ring-Modulation Gadget

Ring-Modulation is a feature of the C64's sound capabilities that allows the C64 to produce complex sounds. Ring-modulated sounds have a bell-like quality.

The Ring-Modulation Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to the quality of the Ring-Modulation. The higher the number, the better the sound quality will be. These settings will produce the following:

- 0 Ring-Modulation is off
- 32 Medium Quality, sounds ok, but not as good as the C64
- 48 Very Good Quality, sounds very close to the C64 (this is the default setting)
- 63 Superb Quality, sounds exactly like the C64

Emulating Ring-Modulation can be a very time consuming process and the higher the quality, the more time it takes to produce the sounds. Lowering this setting can help to speed up A64 and if a program is not using Ring-Modulation, this setting should be disabled (set to 0). You can determine if a program is using Ring-Modulation by changing this setting to 0 to see if there is any change in the sound quality or by using the SIDDump command in A64Mon.

Synchronization Gadget

Synchronization is a feature of the C64's sound capabilities that allows the C64 to produce a certain type of sound. Synchronized sounds are much like a scratching sound.

The Synchronization Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to the quality of the Synchronization. The higher the number, the better the sound quality will be. These settings will produce the following:

- 0 Synchronization is off
- 32 Medium Quality, sounds ok, but not as good as the C64
- 48 Very Good Quality, sounds very close to the C64 (this is the default setting)
- 63 Superb Quality, sounds exactly like the C64

Emulating Synchronization can be a very time consuming process and the higher the quality, the more time it takes to produce the sounds. Lowering this setting can help to speed up A64 and if a program is not using Synchronization, this setting should be disabled (set to 0). You can determine if a program is using Synchronization by changing this setting to 0 to see if there is any change in the sound quality or by using the SIDDump command in A64Mon.

ADSR Gadget

ADSR is short for Attack, Decay, Substain and Release, and it is used to describe and define a sound waveform. All sounds on the C64 have an ADSR. By changing the ADSR of a sound, you can change its waveform to produce a totally different sound.

The ADSR Gadget is represented by a slider gadget. To the right of the gadget is a number that shows the current setting. This setting corresponds to the speed of the ADSR. The higher the number, the faster the ADSR, and the better the sound quality will be. These settings will produce the following:

- 0 Very slow ADSR (ADSR can not be turned off. If it was turned off there would be no sound produced)
- 32 Half the speed of the C64
- 63 Same speed as the C64

Unlike the other sound quality settings, ADSR can not be turned off and changing its setting will not change the speed of A64. The ADSR setting is provided to make fine adjustments to the sound quality.

Code Menu

Patch (Files)

Patch Files allow A64 to run specific C64 programs that normally, for one reason or another, do not run properly. Basically, a Patch File is used to "patch" a C64 program so that A64 can run it correctly. The main purpose for Patch Files is for supporting Fast Loaders, but they can be used for many things.

We developed Patch Files so that we can continue to add Fast Loader support without having to make changes to A64. We can develop Patch Files for different C64 programs and then make them available to our registered owners, without having to update A64.

A64 includes several Patch Files for running certain programs. These files can be found in various drawers on disk #1. You can identify Patch Files by their names. A Patch File will always end in ".Patch" and there is usually a document file included with the Patch Files that will contain information on using the program that the Patch File was designed for.

To use a Patch File, you simply load it with the Load Patch menu option before running the C64 program that the Patch File was designed for. The Patch File will then automatically handle what it was designed to do. When you're done running a program, that uses a Patch File you must select the Free Patch menu option to remove the Patch File from memory.

If you have a C64 program that A64 does not run, send us a letter or Error Report and if possible, a copy of the program. We will try to support the most requested programs.

AutoLoad (Patch)

Key Sequence: None

The AutoLoad Patch menu option allows you to enable or disable AutoLoading of a Patch File. (*See Chapter 4, AutoLoading Files in the Code Menu section for a complete description of AutoLoading*).

Load (Patch)

Key Sequence: None

The Load Patch menu option allows you to load and use a Patch File.

When the Load Patch menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Patch File that you want to load.

Free (Patch)

Key Sequence: None

The Free Patch menu option allows you to remove a loaded Patch File from memory.

When you are done using a C64 program that uses a Patch File, you need to remove the Patch File from memory. This is done by selecting the Free Patch menu option. If you do not remove the Patch File when you are done using it, it can interfere with any other programs that you try to run and cause unpredictable results.

Code (Files)

Code Files are files that are created by the Convert utility. *(See Chapter 6, Convert, for more information on Code Files).*

AutoLoad (Code)

Key Sequence: None

The AutoLoad Code menu option allows you to enable or disable AutoLoading of a Code File. *(See Chapter 4, AutoLoading Files in the Code Menu section for a complete description of AutoLoading).*

Load (Code)

Key Sequence: None

The Load Code menu option allows you to load and use a Code File.

When the Load Code menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Code File that you want to load.

Whenever you load a Code File, A64 will automatically load the Map File that was used to create the Code File as well.

Free (Code)

Key Sequence: None

The Free Code menu option allows you to remove a loaded Code File from memory.

When you are done using a C64 program that uses a Code File you need to remove the Code File from memory. This is done by either selecting the Free Code menu option or by resetting A64 (whenever you reset A64 the Code file is automatically freed).

Map (Files)

A64 allows you to freeze any C64 program that you are running and save the program to an Amiga disk in a "Map File." Once you create a Map File for a program, the Map File can be loaded at anytime, eliminating the need of having to load the C64 program from a C64 disk drive. This can greatly speed up the loading time of programs. Map Files also provide a way to load copy protected C64 software from Amiga disks. You can load a copy protected C64 program from a C64 drive and then, once the program is running, you can save a Map File of the program. A64's Map File feature works much like a C64 freeze cartridge.

AutoLoad (Map)

Key Sequence: None

The AutoLoad Map menu option allows you to enable or disable AutoLoading of a Map File. *(See Chapter 4, AutoLoading Files in the Code Menu section for a complete description of AutoLoading).*

Load (Map)

Key Sequence: None

The Load Map menu option allows you to load and use a previously saved Map File.

When the Load Map menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Map File that you want to load.

When a Map File is loaded, A64 will be reset and the C64 program contained in the Map File will be loaded. Then when you exit A64 Prefs, you will be left in the C64 program at the exact point that it was saved.

Save (Map)

Key Sequence: None

The Save Map menu option allows you to save a Map File of the C64 program that A64 is currently running.

When the Save Map menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Map File that you want to save.

NOTES

- When saving a Map File, it is recommended that you end your file names with ".map" A64 uses this naming convention whenever referencing a Map File and it will help you to identify files if you use this convention as well.

Don't Reset (A64)

Key Sequence: None

The Don't Reset A64 menu option allows you to enable or disable the resetting of A64 when a Map File is loaded.
Normally, A64 resets whenever a Map File is loaded. When the Don't Reset A64 menu option is enabled, A64 will not reset when loading a Map File. For almost all programs this option can be left disabled. There are some programs that use fast loaders where this option must be enabled for the programs to work correctly. For these programs, there will always be a Patch File and a document file stating that the Don't Reset option should be enabled.

Map File Example

The following example describes Saving and Loading a Map File:

Before you save a Map File, you need to run a C64 program. For this example we will use a simple one-line Basic program. While in C64 mode, type in the following:

```
100 PRINT"THIS IS A TEST":GOTO100 [RETURN]

RUN [RETURN]
```

You will then see "THIS IS A TEST" printing endlessly on the screen. Now that you have a program running, you are ready to save a Map File:

1. Enter A64 Prefs by pressing the two ALT keys at the same time.
2. Select the Save Map menu option in the Code menu.
3. You will be prompted with a file requester. For the name of the file, enter "Test.map" (note: if you don't want to save the Map File to your A64 disk, you can enter "RAM:" for the drawer name).
4. Select the Ok gadget and A64 will save the Map File.

You now have a Map File containing the Basic program. This Map File can be loaded at anytime. You could exit back to A64 and load and run as many programs as you like or even quit A64. You will still be able to load this Map File anytime you use A64.

To load the Map File:

1. Select the Load Map menu option in the Code menu.
2. You will be prompted with a file requester Normally you would have to enter the name of the Map File yourself, but for this example, the Map File name will be entered for you. It will be the same name you used when you saved the Map File.
3. Select the Ok gadget and A64 will load the Map File.
4. Exit back to A64 by selecting the Exit menu option in the A64 Menu and the Basic program will continue running at the exact point where it was saved.

That's all there is to it. You could have used any C64 program instead of the Basic program you typed in. The procedures for saving and loading Map Files are the same, regardless of what program A64 may be running.

AutoLoading Files

A64 supports a feature called "AutoLoading." AutoLoading allows a Patch File, Code File and/or Map File to be automatically loaded when a Config File is loaded. If you save a Config File with one or more of the AutoLoad menu options selected, whenever you load that Config File, A64 will automatically load the selected files. You can completely customize A64 to run a particular program, by creating a Config File, Map File and Code File for the program and then have all the files load in one simple operation. A64 can be started by double-clicking on a Config File icon. If you run A64 with this method, and if you have any of the AutoLoad settings enabled in the Config File, A64 can be made to automatically run any C64 program.

AutoLoad Example

This example uses the Map File that was saved in the Map File Example above. The following example shows how to AutoLoad a Map File. The procedures for AutoLoading Patch Files and Code Files are basically the same:

1. Select the Load Map menu option in the Code menu.
2. You will be prompted with a file requester. For the name of the file enter "Test.map"
3. Select the Ok gadget and A64 will load the Map File.
4. Select the AutoLoad Map menu option in the Code menu. This enables AutoLoading of the Map File "Test.map."
5. Select the Save Config menu option in the A64 menu
6. You will be prompted with a file requester. For the name of the file, enter "Test.config" (note: if you don't want to save the Config File to your A64 disk, you can enter "RAM:" as the drawer name).
7. Select the Ok gadget and A64 will save the Config File.

You now have a Config File that will AutoLoad the Map File "Test.map." Anytime you load this Config File, the Map File will also be loaded. You should now quit A64 by selecting the Quit menu option in the A64 menu. Once you are back to the WorkBench you will see an icon for the Config File "Test.config" (you may need to close and re-open the TAP drawer window before the icon will appear or if you used RAM: as the drawer name when you saved the Config File, you will have to open the RAM: window). If you double-click on this icon, A64 will be started and the Map File will be AutoLoaded. When A64 is done loading, you will be left in C64 mode and the Basic program will be running. This procedure could be done for any C64 program and you don't need to Quit A64 to take advantage of AutoLoading. You can load a Config File with the Load Config menu option and AutoLoading will work in the same way. We quit A64 because we wanted to demonstrate how A64 can be made to automatically start running a C64 program.

A64 ROMs

Key Sequence: None

The A64 ROMS menu option allows you to enable and disable A64's ROM emulation. This feature is only available if you have installed the C64 ROMs (*See Chapter 2, Installing The C64 ROMs*).

A64's ROM emulation requires about 180k of memory. If you are running low on memory, you may want to disable the ROM emulation. Disabling the ROM emulation will not improve A64's compatibility and it will greatly reduce the speed of some programs.

Tools Menu

Save Screen

Key Sequence: F1

The Save Screen menu option allows you to save the C64 screen as an IFF file. After saving the screen, you can load it with any Amiga graphics program that supports IFF pictures.

When the Save Screen menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the file that you want to save.

IFF Format

Key Sequence: None

A64 can save screens in two different IFF formats: **ILBM** and **ACBM**. The IFF Format menu option allows you to switch between these two formats.

Load (Font)

Key Sequence: None

The Load Font menu option allows you to load a previously saved C64 font.

When the Load Font menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the font that you want to load.

The A64 Package comes with several C64 fonts that are located on disk 1 in the 64Fonts directory.

Save (Font)

Key Sequence: None

The Save Font menu option allows you to save the font that the C64 program is using.

When the Save Font menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the font that you want to save.

View (Font)

Key Sequence: None

The View Font menu option allows you to view the current font being used by a C64 program.

The C64 is capable of using more than one font at a time and the current font being used may not necessarily be the one you think it is. While running the same C64 program, you may enter A64 Prefs several times and find that the font has changed every time. The View Font menu option will show you exactly what will be saved if you select the Save Font menu option.

When the View Font menu option is selected, you will be shown a screen containing the current C64 font. After your are done viewing the font, press the left mouse button to return to A64 Prefs.

Chapter 5
A64Mon

Key Sequence: [F9] (can be done from C64 Mode and A64 Prefs)

A64Mon is provided for more experienced C64 users. This section of the manual is technical and assumes that your are familiar with the C64's architecture and 6510 machine language.

WARNING: *A64Mon should be used with caution! It is very easy to cause a "C64 crash" if some commands are used incorrectly.*

A64Mon is a powerful utility program that allows you to examine, at the machine level, any C64 program that A64 is running. A64Mon is very much like many machine language monitors available for the C64. However, A64Mon does not use any of the C64's memory so it is able to run totally transparent to all C64 programs. With A64Mon, virtually any C64 program can be investigated.

When the A64Mon menu option is selected the A64Mon screen will be opened. The A64Mon screen can be broken down into three areas:

6510 Status Area

At the top left of the A64Mon screen is a small area used to display the 6510 microprocessor's current status. The Program Counter (PC), Accumulator (A), X Register (X), Y Register (Y), Stack Pointer (SP) and Status Register (SR) are displayed (set SR flags are high-lighted in yellow).

Command Area

Below the 6510 Status Area is a large area that covers most of the left side of the A64Mon screen. This is A64Mon's Command Area. This is where you enter commands and receive their output.

The Command Area acts like a full screen editor and you can move the cursor around with the keyboard or mouse (note: you can only move to a line that starts with a period).

The following keyboard sequences can be used in the Command Area:

[LEFT AMIGA] [C]	Clear the Command Area and home the cursor
[LEFT AMIGA] [L]	Clear the current line, from the cursor position to the end of the line
[CRSR UP]	Move up 1 line, if the current command is repeatable do command backward for 1 line
[SHIFT] [CRSR UP]	Move up 1 page, if the current command is repeatable do command backward for 1 page
[CRSR DOWN]	Move down 1 line, if the current command is repeatable do command forward for 1 line

[SHIFT] [CRSR DOWN]	Move down 1 page, if the current command is repeatable do command forward for 1 page
[ESC]	Stop a command
[SPACE BAR]	Pause and unpause a command

NOTES

- If ESC fails to stop a command or the SPACE BAR fails to pause a command, make sure the A64Mon screen is active by clicking the left mouse button anywhere on the screen.
- Throughout this section of the manual reference is made to a "page." A page is the entire Command Area (18 lines).

Command Status Area

Below the Command Area is a small area used to display status messages.

A64Mon Menu

A64Prefs (P)

Function: Exit to A64Prefs
Usage: A64Prefs

When exiting to A64Prefs the A64Mon screen is not closed, but you will be unable to access A64Mon's commands. To regain access to A64Mon, press the [F9] key or select the A64Mon menu option in the Tools menu.

Exit (X)

Function: Exit to C64 Mode
Usage: Exit

If the Exit command is entered with the keyboard method, you will enter C64 Mode until all keys are released on the keyboard.

Quit (Q)

Function: Quit A64Mon
Usage: Quit

Settings Menu

Number Base

Hex (None)

Function: Change default number base to hexadecimal
Usage: Hex

Dec (None)

Function: Change default number base to decimal
Usage: Dec

UndefinedOps (UO)

Function: Enable and disable 6510 undefined opcodes
Usage: UndefinedOps

By default A64Mon will not assemble or disassemble the 6510's undefined opcodes. The UndefinedOps command allows A64Mon to recognize these opcodes.

NOTE: There is a document file included with The A64 Package that describes how A64 and A64Mon interpret the 6510's undefined opcodes.

Commands Menu

Assemble (A)

Function: Assemble 6510 mnemonics into machine language
Usage: Assemble <addr> <opcode>
<addr> address
<opcode> 6510 opcode

Disassemble (D)

Function: Disassemble machine language into 6510 mnemonics
Usage: Disassemble [start addr] [end addr]
[start addr] start address
[end addr] end address

The Disassemble command is repeatable.

The Disassemble command is capable of disassembling backwards through memory (ie. higher address to lower address), but it is very difficult to get accurate disassemblies in this way. You should only disassemble in reverse to get the current address in a desired range and then disassemble one page forward by pressing [SHIFT] [CRSR DOWN], to get an accurate disassembly.

MemDump (M)

Function: Do numeric dump of memory
Usage: MemDump [start addr] [end addr]
[start addr] start address
[end addr] end address

The MemDump command is repeatable.

You can modify memory by editing the MemDump command's output.

ASCIIDump (AD)

Function: Do numeric and PetASCII dump of memory
Usage: ASCIIDump [start addr] [end addr]
[start addr] start address
[end addr] end address

The ASCIIDump command is repeatable.

The Command Area uses true ASCII and not PetASCII. PetASCII dumps of memory will only show alpha-numeric characters. All other characters will be represented by an underscore.

You can modify memory by editing the ASCIIDump command's numeric output.

CIADump (CD)

Function: Dump the contents of the CIA chips
Usage: CIADump <chip number>
<chip number> 1 = CIA 1, 2 = CIA 2

You can modify the CIA registers by editing the CIADump command's output.

SIDDump (SD)

Function: Dump the contents of the SID chip
Usage: SIDDump

You can modify the SID registers by editing the SIDDump command's output.

VICDump (VD)

Function: Dump the contents of the VIC chip and raster interrupts
Usage: VICDump

The VICDump command is repeatable if the current C64 program is using raster IRQs with more than one screen division.

You can modify the VIC registers by editing the VICDump command's output.

IntVecs (IV)

Function: Display current interrupt vectors
Usage: IntVecs

If the C64 program has the Kernal ROM banked in, the IntVecs command will display the Kernal address of the interrupt handler followed by the RAM address that is jumped to by the handler.

MemCntrl (MC)

Function: Display the C64's current memory configuration
Usage: MemCntrl

The C64's map configuration can be changed by editing the first line of the MemCntrl command's output. You are responsible for restoring the original map configuration whenever it is changed in A64Mon.

Poke (PO)

Function: Poke (write) a value to a memory location
Usage: Poke <addr>,<value>
<addr> address
<value> value to write

Peek (PE)

Function: Peek (read) a value from a memory location
Usage: Peek <addr>
<addr> address

Transfer (T)

Function: Transfer (copy) a block of memory
Usage: Transfer <start addr> <end addr> <dest>
<start addr> start address
<end addr> end address (not included in transfer)
<dest addr> destination address

Comapre (C)

Function: Compare two blocks of memory
Usage: Compare <start addr> <end addr> <dest>
<start addr> start address
<end addr> end address (not included in compare)
<dest addr> destination address

Fill (F)

Function: Fill a block of memory with a value
Usage: Fill <start addr> <end addr> <value>
<start addr> start address
<end addr> end address (not included in fill)
<value> fill value

Search (SE)

Function: Search memory for values
Usage: Search <start addr> <end addr> <value(s)>
<start addr> start address
<end addr> end address (not included in search)
<value(s)> value(s) to search for

The Search command can search for up to eight byte values that are stored consecutively in memory. The values parameter can either be an ASCII string or numeric byte values. To search for an ASCII string, enclose the values parameter in quotes. To search for numeric values, type each value separated by one or more spaces.

Value (V)

Function: Display a number in hexadecimal, decimal and binary formats
Usage: Value <number>
<number> number to display

Trace Commands Menu

A64Mon supports a number of commands called "Trace Commands." Trace Commands are used in conjunction with tracing C64 programs. When a Trace Command is entered, it becomes enabled and will stay in effect until it is disabled. Due to the nature of these commands, they will dramatically decrease the execution speed of A64, especially when more than one command is used at a time. We highly recommend an accelerated Amiga when using these commands and that they only be left enabled when necessary.

TraceCmds (TC)

Function: Enable and disable A64Mon's Trace Commands.
Usage: TraceCmds

Resetting A64 will automatically disable A64Mon's Trace Commands.

Registers (R)

Function: Display and edit the 6510's registers
Usage: Registers

You can change the 6510's registers by editing the Registers commands output.

SaveRegs (SR)

Function: Save the status of the 6510's registers
Usage: SaveRegs

ResetRegs (RR)

Function:	Reset (restore) the status of the 6510's registers
Usage:	ResetRegs

Break (B)

Function:	Set 6510 PC break points
Usage:	Break <opt> [n] (opt = addr, list, clr)
	<opt> addr = set or clear a break point at address
	list = list all current break points
	clr = clr all current break points
	[n] allow break point to be hit [n] times before breaking

You can set up to four individual break points at a time.

BreakA (BA)

Function:	Set break based on Accumulator value
Usage:	BreakA <opt> (opt = value, list)
	<opt> value = value to break on
	list = show current break value

BreakX (BX)

Function:	Set break based on X register value
Usage:	BreakX <opt> (opt = value, list)
	<opt> value = value to break on
	list = show current break value

BreakY (BY)

Function:	Set break based on Y register value
Usage:	BreakY <opt> (opt = value, list)
	<opt> value = value to break on
	list = show current break value

BreakSP (BS)

Function:	Set break based on SP register value
Usage:	BreakSP <opt> (opt = value, list)
	<opt> value = value to break on
	list = show current break value

BreakSR (BP)

Function:	Set break based on SR register value
Usage:	BreakSR <opt> (opt = flags, list)
	<opt> flags = flags to break on
	list = show current break flags

The BreakSR command will cause a break when one or more of the entered flags change.

Walk (W)

Function:	Enable and disable Walk mode
Usage:	Walk

When Walk mode is enabled, anytime you exit to C64 Mode, the current program will be single stepped.

Go (G)

Function:	Exit to C64 Mode using address for PC
Usage:	Go [addr]
	[addr] address to go to

The Go command is very similar to the Exit command, except that you can change the PC address that A64 is entered with.

Jsr (J)

Function:	Execute a 6510 subroutine
Usage:	Jsr [addr]

The Jsr command is the only A64Mon command that uses the C64's memory to store values. The command pushes a RTS address on the C64's stack and returns to A64Mon when this address is pulled by A64. The subroutine being called can call other subroutines. The Jsr command will return to A64Mon when the last RTS is executed.

You must be careful when using the Jsr command. If the subroutine being called does any extra pulling of values from the stack the RTS address will be lost and the Jsr command may never return to A64Mon (you can still return to A64Mon by pressing F9). Also, if you do not allow the Jsr command to finish execution (by entering A64Mon before the RTS address is pulled by A64), the RTS address will be left on the stack and could cause a C64 crash when you exit back to C64 Mode. To help eliminate any problems with the Jsr command, you should save the 6510's registers before using the Jsr command and then restore the registers when you are done (*See the SaveRegs and ResetRegs commands*).

MemWatch (MW)

Function: Break on changes of memory
Usage: MemWatch <opt> (opt = addr, list, clr)
<opt> addr = set or clear MemWatch point at address
list = list all MemWatch points
clr = clr all MemWatch points

You can set up to four individual MemWatch addresses at a time.

BankWatch (BW)

Function: Watch the execution of a C64 program
Usage: BankWatch [opt] (opt = list, clr)
<opt> list = list how many opcodes have been executed in each 4k bank
clr = clr all values

The BankWatch command can be used to determine where in memory a C64 program has code and where it is spending the most time. This can be useful when building Address Files for the Convert utility.

TrapBadOp (TO)

Function: Cause break on execution of undefined opcode
Usage: TrapBadOp

Files Menu

OpenFile (OF)

Function: Send A64Mon's output to a file
Usage: OpenFile <"filename">
<"filename"> name of file to send output to

If the OpenFile command is entered with the menu option or without a filename, you will be prompted with a file requester. If you type in the filename, it must be enclosed in quotes.

Comments can be sent to an output file by typing a period followed by your comments. Output can be sent to a printer by using the name of the device for the filename.

> *NOTE: If you are sending output to a parallel printer, please read Parallel Port in the System Menu section for information on a bug in V1.3 of the Amiga's operating system.*

CloseFile (CF)

Function: Close the output file opened with the OpenFile command
Usage: CloseFile

Quitting A64Mon will automatically close the output file.

Load (L)

Function: Load file to memory
Usage: Load <"filename"> [addr] [-s]
<"filename"> name of file to load
[addr] load address
[-s] treat file like a sequential file

If the Load command is entered without a filename, you will be prompted with a file requester.

By default, A64Mon treats files like C64 PRG files and it is assumed that the first two bytes of the file are the load address. By using the -s parameter and addr parameter, you can force A64Mon to treat the file like a C64 SEQ file and it will be assumed that the file does not contain a load address.

Save (S)

Function: Save memory to file
Usage: Save <"filename"> [start addr] [end addr] [-s]
<"filename"> name of file to save
[start addr] start address
[end addr] end address (not included in save)
[-s] treat file like a sequential file

Unless the -s option is used, A64Mon will treat the file like a C64 PRG file and the load address will be saved as the first two bytes of the file.

Saves are slow, they are done one byte at a time.

Command Notes

- All of A64Mon's commands can be entered either by selecting a menu or by typing the command in the Command Area. When a command is entered with a menu option, the command will be printed to the Command Area as if you had typed in the command manually. If the command does not require parameters, it will be executed immediately. Otherwise, the cursor will be printed to the right of the command so you can enter the parameters.
- Most commands have an abbreviated form that can be used instead of typing in the entire command. The abbreviated form is one or two characters long as is listed in the menus.
- To get a command's usage, enter the command name followed by a '?'. When a command's parameters are listed, parameters enclosed in < > are required and parameters enclosed in [ ] are optional
- Commands and parameters are not case sensitive.
- Commands and parameters should be separated by one or more spaces.
- A64Mon uses a default number base, either hexadecimal or decimal, for all of its input and output. Commands that use numbers for parameters can accept any combination of hexadecimal and decimal numbers. Hexadecimal numbers are denoted by a "$" and decimal numbers are denoted by a "#". If no symbol is used then the default number base will be used to determine a numbers base (*See the Hex and Dec commands*).
- A64Mon uses a "current address" for its commands that require addresses. The current address starts at the address held in the 6510's PC when A64Mon is entered and changes whenever a command is used that uses addresses.
- Some of A64Mon's commands are repeatable. Meaning, if you enter a repeatable command, its output can be continued using the cursor keys or mouse. For example, if you enter: Disassemble C000, A64Mon will disassemble one page starting at address $C000. You can then continue disassembling by either using the cursor keys or the mouse. To use the mouse, hold down the left mouse bottom and move the mouse at either the top or bottom of the Command Area. Repeatable commands can either move forward or backward through memory.
- All of A64Mon's commands that deal with the C64 memory map use the map's current configuration. If you wish to read or write to a portion of the memory map that is not currently banked in, you must switch in the memory yourself. You should restore the memory map configuration before exiting A64Mon (*See the MemCntrl command*).

- For commands that use a start address and end address as optional parameters (ie. Disassemble, MemDump and ASCIIDump): If no parameters are given, A64Mon will perform the command for one page, starting at the current address. If no end address is given, A64Mon will perform the command for one page, starting at the start address. You can use a period as an end address to force A64Mon to perform the command for just one line.

Convert

Convert is a utility program that is specifically designed to increase the speed of C64 programs (*See Chapter 3, A64's Speed*). The speed increases that can be gained will vary from program to program, but the average result will be A64 running two times faster than normal. The Convert utility works with Map Files created with A64 (*See Chapter 4, Map Files, in the Code Menu section*). You should familiarize yourself with this feature of A64 before attempting to use the Convert utility.

How A64 Runs C64 Programs

Before explaining the Convert utility, we would like to briefly explain machine language and how A64 runs C64 programs. You do not need to know machine language or any other programming language to use the Convert utility. This information is provided so users can gain a better understanding of how Convert works.

Machine language programs are made up of opcodes. Opcodes are machine language instructions that tell the computer to perform some tiny operation, like move a value from one place to another or perform some kind of operation to a value. It takes many, many opcodes to perform even the simplest operations on your computer.

The opcodes of a machine language program are directly executed by a computer's microprocessor. The Amiga and the C64 use different microprocessors so their machine languages are different. This is one of the reasons why you usually can't take a program written for one computer and run it on a different computer.

Because the Amiga uses a different microprocessor than the C64, the Amiga can not directly run a C64's machine language. To run a C64 machine language program on the Amiga, the program must be interpreted. Meaning, an Amiga program must fetch each C64 machine language opcode and then execute a group of instructions that simulate the operations of the C64's opcode. This is how A64 runs a C64 program. It continuously fetches and simulates C64 opcodes. As you might imagine, this process of interpreting the C64's opcodes, is very time consuming. A64 must be able to interpret thousands of C64 opcodes per second and still handle sound, graphics, I/O and many other miscellaneous functions.

How Convert Works

The Convert utility converts C64 machine language into Amiga machine language. This conversion process eliminates a great deal of work that A64 would normally have to do "on the fly." A64 will no longer have to interpret the C64 machine language because it will already be in a form that A64 understands, pure Amiga machine language.

The process of converting a program can be broken down into three simple steps:

1. Using A64 to save a Map File of a C64 program
2. Loading the Map File with Convert
3. Converting the program

When the Convert utility is done converting a C64 program, the converted program is automatically saved in a "Code File." The Code File will be named using the name of the Map File followed by ".code" You then load this file with A64's Load Code feature (*See Chapter 4, Load Code, in the Code Menu section*). Once you have a Code File for a program, you can load and use it with A64 at anytime. If you are familiar with loading Map Files with A64, using Code Files will be very easy. Whenever you want to use a particular C64 program, you will load the Code File instead of the Map File.

Code files do not actually replace Map Files. The Map File is still needed by A64 to run the C64 program correctly. When you load a Code File, A64 will automatically load the Map File too.

NOTE: Code Files can be very large. They average between 100-400k in size. If you have under 2mb of RAM, you may not have enough memory to use Code Files with A64.

The Convert utility may appear complex at first, but it is actually very easy to use. We realize that technical words like "machine language" and "opcodes" are intimidating to many users. Rest assured, you do not need to be a programmer to use the Convert utility. Once you see how easy it is to convert a C64 program and the speed increases that can be gained, you'll probably want to convert as many programs as possible.

Convert Menus

Convert has many menu options, but to convert C64 programs you will mainly be interested in the menu options in the Convert menu. The rest of the menu options are for more specialized conversions and you do not need to be concerned with them to convert most C64 programs.

Convert Menu

Load Map

Key Sequence: [RIGHT AMIGA] [L]

The Load Map menu option allows you to load a Map File so that it can be converted.

Before you can convert a C64 program, you must create a Map File with A64's Save Map feature. You then load this Map File with the Load Map menu option.

When the Load Map menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Map File that you want to load.

Optimize

Key Sequence: [RIGHT AMIGA] [O]

The Optimize menu option allows Convert to optimize the C64 program before it is converted.

Believe it or not, many C64 programs spend a great deal of time sitting in loops doing absolutely nothing, but wasting time. This is done to keep C64 programs from performing some operations too fast. For example, if the C64 were to move a sprite across the screen as fast as it was able, the sprite would become a blur. So programmers put delay loops in their programs to slow things down. The Optimize menu option will allow you to remove these delay loops from C64 programs.

When the Optimize menu option is selected, Convert will search the loaded Map File for delay loops. When a delay loop is found, it will be listed on the screen and you will be asked if Convert should remove the delay. Press the [Y] key to remove the delay or any other key to ignore it.

The reason you are asked if the delay should be removed is that there may be times when you don't want to remove all of the delays in a C64 program. Removing some delays can cause some C64 programs to run too fast. If you encounter this problem, you'll have to experiment with removing some delays and not others until you find the one(s) that are making A64 run too fast.

After Convert is done searching for delays, it will print how many optimizations were done. If there were any optimizations, you will be prompted with a file requester. The file requester will display the name of the loaded Map File with the suffix ".optmap." This is an abbreviation for Optimizied Map. You should then select the Ok gadget and Convert will save a new Map File that contains the optimizations. Creating a new Map File allows you to preserve the original Map File so that it can be converted again if necessary.

Convert

Key Sequence: [RIGHT AMIGA] [C]

The Convert menu option performs the actual conversion process.

When the Convert menu option is selected, the Convert utility will convert the C64 machine language into Amiga machine language and automatically save the resulting converted code in a Code File. This Code File can then be loaded with A64's Load Code feature.

About

Key Sequence: [RIGHT AMIGA] [A]

The About menu option allows you to view some information about Convert.

Quit

Key Sequence: [RIGHT AMIGA] [Q]

The Quit menu option allows you to quit the Convert utility.

> The following sections pertain to more advanced uses of the Convert utility. These sections are technical and assume that you have a basic understanding of the C64's architecture and 6510 assembly language. You do not need to use the following features of the Convert utility to convert C64 programs. They are provided for more experienced users.
>
> Unless you are familiar with the C64's architecture and 6510 assembly language you should skip ahead to the section: Converting Your First Program.
>
> Before attemping to use the following features ,you should read the sections: How Convert Really Works and Address Files.

Map Config Menu

Map Info

Key Sequence: None

The Map Info menu option allows you to view the current memory configuration being used by the loaded Map File.

Map Config

Key Sequence: None

The Map Config menu option allows you to tell Convert whether or not the C64 program is using bank switching or not.

Many C64 programs do not change the C64's memory configuration once the program is loaded and initialized. If a program does not use bank switching, Convert can make many optimizations to the converted code.

The Map Config menu option has two submenus:

- **Any**: When this submenu is selected, Convert will assume that the program is using bank switching and Convert will be unable to make any optimizations to the code that rely on a fixed memory map.
- **Fixed**: When this submenu is selected, Convert will assume that the current memory configuration (as listed with the Map Info menu option) will remain constant throughout the C64 program. This will allow Convert to perform many optimizations to the converted code.

VIC Bank

Key Sequence: None

The VIC Bank menu option allows you to tell Convert whether or not the C64 program is always using the same VIC bank or not.

Many C64 programs do not change the VIC chip's graphics bank once the program is loaded and initialized. If a program does not change the VIC bank, Convert can make many optimizations to the converted code.

The VIC Bank menu option has two submenus:

- **Any**: When this submenu is selected, Convert will assume that the program is using multiple VIC banks and Convert will be unable to make any optimizations to the code that rely on a fixed VIC Bank.
- **Fixed**: When this submenu is selected, Convert will assume that the current VIC Bank (as listed with the Map Info menu option) will remain constant throughout the C64 program. This will allow Convert to perform many optimizations to the converted code.

Addresses Menu

Clear

Key Sequence: None

The Clear menu option allows you to clear all Include and Exclude addresses currently being stored by Convert.

Includes

Key Sequence: None

The Includes menu option allows you to view and select what 4k blocks will be included in the conversion.

The Include menu option has 16 submenus. Each submenu represents a 4k block of RAM in the C64's memory map. By selecting and unselecting these submenus you can force Convert to Include or Exclude blocks.

Analyze Map

Key Sequence: None

The Analyze Map menu option will cause Convert to analyze the Map File to determine what addresses should be Included and Exclude.

The process of analyzing the Map File is usually the first step in converting a program (selecting the Convert menu option automatically analyzes the Map File if it has not been done already).

Load Address File

Key Sequence: None

The Load Address File menu option will allow you to load a previously saved Address File.

When the Load Address File menu option is selected, you will be prompted with a file requester. With the file requester, you can either select a displayed name or enter the name of the Address File that you want to load.

When an address file is loaded, Convert will parse the addresses stored in the Address File and use them for the conversion process.

Options Menu

List Files

Key Sequence: None

The List Files menu option allows you to enable and disable Convert's List Files.

Convert has the ability to create List Files that contain 6510 disassemblies of the program being converted. These List Files can be used to see exactly what Convert has Included and Excluded.

When the List Files menu option is selected, you will be prompted with a requester asking you to enter the directory name of where Convert should save the List Files. By default, Convert will save the List Files to RAM:, but you can save them to any valid Amiga directory.

During the conversion, Convert will create a separate List File for each 4k Included block. The List Files will be automatically named using the name of the Map File followed by ".xxxx-yyyy.list" (xxxx = start address, yyyy = end address).

List Files are fairly large. They average about 50k in size. An average program, that has 6-8 Included blocks, will require between 300-400k of free disk space for List Files.

Save Address File

Key Sequence: None

The Save Address File will allow you to save an Address File for the program being converted (*See Address Files below*).

When the Save Address File menu option is selected, Convert will save an Address File for the program that was converted. The Address File is saved when the conversion is done and it will automatically be named using the name of the Map File followed by ".addr"

How Convert Really Works

The C64 memory map contains 64k of RAM. Whenever the C64 loads a program, it stores the program somewhere in this 64k of RAM. The initial step in converting a C64 program is determining what parts of the C64's memory map contain actual machine language instructions and what parts contain data or are unused. The Convert utility handles this automatically by analyzing the Map File (*See Chapter 6, Analyze Map, in the Addresses Menu section*).

When Convert analyzes the Map File, it breaks the memory map down into 4k blocks. There are 16 4k blocks in the memory map. Convert then analyzes each 4k block. If a block contains 20% or more of code, it will be "Included" in the conversion. Anytime a block is Included, it must also be searched to see if it contains data. When data is found in a block the addresses are marked as "Excluded."

The process of analyzing the memory map is very accurate and it will handle the majority of C64 programs, but there may be times when you want more control over what addresses are Included and Excluded in the conversion:

- It is possible for a C64 program to store routines that are called very frequently in a block that contains less than 20% code. Convert would automatically Exclude this block, but if you Included this block, with the Includes menu option, you would get significantly better results from the converted code.

- Convert assumes that the code being converted will remain the same. If the C64 program loads new code into an area that has been Included, the C64 program will fail to function correctly. Whenever you are converting C64 programs that do additional loading (after the point in which the Map File was saved), you must be careful that the memory being loaded to is Excluded. Usually when a program does multiple loads, one or more sections of the memory map will remain constant and the C64 program will load code, as needed, into the other sections of the memory map. GEOS for example, stores the GEOS Kernal routines at $a000-$ffff. The code in this area stays constant whenever a GEOS application is being used and can be converted without any problems. But the rest of the memory map, will change whenever a new GEOS program is used and it should not be Included in the conversion.

Address Files

Convert has the ability to save the addresses that were Included and Excluded in a conversion. The addresses are saved in an "Address File" (*See Chapter 6, Save Address File, in the Options Menu section*). Address Files are simple ASCII text files and they can be read (and edited) with any text editor or word processor that supports pure ASCII files. Whenever you make changes to the addresses in Convert, you can create an Address File. This will allow you to experiment with different addresses without having to make manual changes every time you want to convert a program. When an Address File is loaded, Convert will skip analyzing the Map File and use the addresses listed in the Address File instead.

The addresses listed in an Address File must appear in a very specific format. This is a partial listing of an actual Address File:

```
; Archon.addr         ; this line is a comment

#$1000-$1FFF          ; these are the addresses that are Included in
#$3000-$3FFF          ; the conversion
#$6000-$6FFF
#$7000-$7FFF
...                   ; continuation of Included addresses

!$1000-$105A          ; these are the addresses that are Excluded
!$106A-$106D          ; from the conversion
!$1082-$108B
!$10A3-$10A3
...                   ; continuation of Excluded addresses
```

Lines that begin with the '#' symbol represent a 4k block that is to be Included in the conversion. Each line will have the '#' symbol followed by two addresses. The first address will be the start address of the 4k block and the second address will be the end address of the 4k block. Included blocks always begin on an even 4k boundary and are 4k in length. They are always listed first in the Address File and they must be listed in address order (lower addresses before higher addresses).

Lines that begin with the '!' symbol represent a group of addresses that are to be Excluded from the conversion. Each line will start with the '!' symbol followed by two addresses. The first address will be the start address of the group and the second address will be the end address of the group. Excluded addresses must always appear within an Included block. For example the first Exclude line: !$1000-$105A, these addresses are within the Included block $1000-$1FFF. Excluded addresses are always listed last in the Address File and they must be listed in address order (lower addresses before higher addresses). There are three instances when addresses would be Excluded:

1. The C64 program has data imbedded in its code
2. The C64 program is using undefined opcodes
3. The C64 program is using self-modifying code

Addresses in an Address File can be listed in either hexadecimal or decimal. Addresses are assumed to be decimal unless they begin with the '$' symbol.

Lines that begin with any character other than the '#' or '!' symbols are considered to be comments and they can appear anywhere in the Address File.

You can edit Address Files manually, but under normal circumstances Convert will create them for you and editing them will be unnecessary.

A good way to get a better understanding of Address Files is to create Address Files and List Files for a couple of programs and then look at the files to see what addresses were used.

What Can and Can't Be Converted

Virtually any C64 machine language program can be converted, but there are two things to be aware of. Programs that use multiple loads should be converted with caution. If a program loads different code over an area that you have previously converted, we can guarantee that you will have problems running the C64 program. Also, Convert can not catch all instances of self-modifying code. If the self-modifying is done using one of the 6510's indirect addressing modes, Convert can not trap it. This problem is very hard to find and correct. If you find a C64 program that runs fine under A64, but when converted, it fails to run correctly there is a good chance that you have included some code that is being self-modified. To correct the problem, you'll either have to try to track down the self-modifying code by examining List Files or experiment with not including blocks until you eliminate the problem.

Basic programs can not be converted.

Converting Your First Program

The following procedure outlines converting a C64 program. Before following this example, you must create a Map File for a C64 program (*See Chapter 4, Map File Example, in the Code Menu section*). This procedure can be used for almost any C64 program. But for this example you should select a program that works well with A64 and one that does not do any loading after the point in which the Map File was saved.

1. Run Convert by double-clicking on the Convert icon (Convert can be found on disk #2).
2. Once Convert is running, select Load Map in the Convert Menu. You will be prompted with a file requester. Enter the name of the Map File that you are using for this example.
3. Select Optimize in the Convert Menu. Convert will then search the Map File to see if there are any delays that can be removed. If Convert finds any delays, you will be prompted for each one. Type the [Y] key each time you are prompted.
4. If Convert was able to remove any delays from the C64 program, you will then be prompted with a file requester. Select the OK gadget to save the new Map File.
5. Select Convert in the Convert Menu. Convert will convert the C64 program and when it is done, it will save a Code File in the same drawer that contains the Map File.
6. Quit Convert by selecting Quit in the Convert Menu.

That's all there is to it. You can now load the converted program with A64's Load Code feature.

Chapter 7

A64Tools

A64Tools is a very flexible file transfer and file conversion utility. A64Tools allows you to easily copy, convert and print files, using any combination of Amiga and C64 disk drives and printers.

A64Tools Screen

The A64Tools screen can be separated in half. The left side of the screen is called the "Source Device Area" and it is used to select the device you are copying "from" and the files that are to be copied. The right side of the screen is called the "Destination Device Area" and it is used to select the device you are copying "to." Both sides of the screen look almost identical and they both operate in a very similar manner.

In the following sections, we describe the functions of the Source Device Area and the Destination Device Area together. Whenever there is a difference between the two areas, it will be noted.

Device Type Gadget

At the top left corner of each device area is the Device Type gadget. This gadget allows you to select either a C64 device or an Amiga device.

Device Name Gadget

To the right of the Device Type gadget is the Device Name gadget. This gadget allows you to enter the specific name of the device.

If you are using a C64 device, the Device Name gadget is used to set the C64 device number (4 or 5 for C64 printers and 8-11 for C64 disk drives).

If you are using an Amiga device, the Device Name gadget is used to specify the device and/or directory name.

Whenever you select one of the Device Name gadgets and press the RETURN key, A64Tools will rebuild the list of names shown with the Device List gadgets.

Device List Gadgets

Below the Device Type and Device Name gadgets are the Device List gadgets. These gadgets are used to display a scrolling list of names. The lists are used to show what devices, directories and files are available. You can scroll through the list by selecting the scroll bar that appears to the right of the names or by selecting the arrow gadgets that appear below the scroll bar. If you are familiar with file requesters, these gadgets work in the same way.

When you select a device or directory name from the displayed list, you will move to that device or directory.

For the Source Device Area only: When you select a file name from the displayed list, that file will be included in the copy. When you unselect a file name, that file will be excluded from the copy.

Devices Gadget

Below the Device List gadgets is the Devices gadget. This gadget is used to display a list of the current devices that can be used.

Assigns Gadget

To the right of the Devices gadget is the Assigns gadget. This gadget is to display a list of the current Assigns.

If the device type is set to C64, the Assigns Gadget will be disabled.

Parent Gadget

To the right of the Assigns gadget is the Parent gadget. This gadget is to move to the parent directory of the current directory.

If the device type is set to C64, the Parent Gadget will be disabled.

ST Gadget

Below the Devices gadget is the ST gadget. This gadget is used to read the current status of the device.

NOTES

- If both the source Device Type and destination Device Type are set to Amiga, both ST gadgets will give the status of the last device accessed.

Main Status Area

Covering most of the bottom of the screen is the Main Status area. This area is used to show A64Tool's progress in copying, converting and printing files.

Start/Stop Gadget

At the bottom right corner of the screen is the Start/Stop gadget. This gadget is used to start and stop the copying or printing of files.

NOTES

- When A64Tools is accessing a C64 device you will be unable to move the mouse pointer and the mouse pointer will flash different colors. This is normal.
- The Stop gadget is only checked in between operations. If you are copying a large file it may take A64Tools some time before it actually stops the operation in progress. If you see the Stop gadget highlight when you select it, A64Tools will recognize your request to stop the operation.

A64Tools Menus

A64Tools Menu

Modes

A64Tools has two modes of operation, copying files and printing files. The Modes menu option allows you to select one of these modes. The Modes menu option has two submenus:

Copy (Mode)

Key Sequence: [Right Amiga] [C]

The Copy Mode menu option allows you to put A64Tools in copy mode. All selected files will be "copied" from the source device to the destination device.

Print (Mode)

Key Sequence: [Right Amiga] [P]

The Print Mode menu option allows you to put A64Tools in print mode. All selected files will be "printed" from the source device to the destination device.

About (A64Tools)

Key Sequence: [Right Amiga] [A]

The About menu option allows you to view some information about A64Tools.

Quit (A64Tools)

Key Sequence: [Right Amiga] [Q]

The Quit menu option allows you to quit A64Tools.

Conversions Menu

A64Tools supports a number of file conversions that can be done to files as they are being copied or printed. The menu options in the Conversions Menu allow you to select one of these file conversions.

No Conversion

Key Sequence: None

When the No Conversion menu option is selected, A64Tools will not do any conversions to the files.

PetASCII > ASCII

Key Sequence: None

When the PetASCII > ASCII menu option is selected, A64Tools will convert the files from Pet ASCII to standard ASCII.

ASCII > PetASCII

Key Sequence: None

When the ASCII > PetASCII menu option is selected, A64Tools will convert the files from standard ASCII to Pet ASCII.

Screen Codes > ASCII

Key Sequence: None

When the Screen Codes > ASCII menu option is selected, A64Tools will convert the files from C64 screen codes to standard ASCII.

Strip Load Address

Key Sequence: None

When the Strip Load Address menu option is selected, A64Tools will strip the first two bytes from the files.

NOTE

- The Strip Load Address option must be used when installing the C64 ROMs on your A64 disk (*See Chapter 2, Installing the C64 ROMs*), but this option does not need to be used when copying C64 files to Amiga disks so they can be used with A64.

Filters Menu

Include All (Files)

Key Sequence: [Right Amiga] [I]

The Include All Files menu option allows you to select all the files on the source device.

Exclude All (Files)

Key Sequence: [Right Amiga] [E]

The Exclude All Files menu option allows you to unselect all the files on the source device.

C64 Filters

The C64 Filters menu option allows A64Tools to filter out certain files from C64 devices. The C64 Filters menu option has four submenus:

No REL

Key Sequence: None

The No REL menu option allows you to filter out C64 Relative files from the source device.

NOTE

- The current version of A64Tools does not support transferring C64 REL files. This menu option is provided for future compatibility.

No USR

Key Sequence: None

The No USR menu option allows you to filter out C64 User files from the source device.

No SEQ

Key Sequence: None

The No SEQ menu option allows you to filter out C64 Sequential files from the source device.

No PRG

Key Sequence: None

The No PRG menu option allows you to filter out C64 Program files from the source device.

Amiga Filters

The Amiga Filters menu option allows A64Tools to filter out certain files from Amiga devices. The Amiga Filters menu option has one submenu:

No .info

Key Sequence: None

The No .info menu option allows you to filter out .info files from the source device.

Options Menu

Print FF Between Files

Key Sequence: None

The Print FF Between Files menu option allows you to print a Form Feed after each file that is printed. This option only works when A64Tools is in Print Mode.

Overwrite Existing Files

Key Sequence: None

The Overwrite Existing Files menu option allows A64Tools to overwrite existing files without prompting you that the file already exists.

When the Overwrite Existing Files menu option is selected, A64Tools will overwrite any file that has the same name as a file being copied. If this option is unselected, A64Tools will warn you if it encounters a file that has the same name as the file being copied.

Prompt Illegal Filenames

Key Sequence: None

The Prompt Illegal Filenames menu option will allow A64Tools to warn you when it encounters illegal characters in an Amiga filename.

The ':' and '/' characters are illegal characters in Amiga filenames. When A64Tools encounters one of the characters in a filename, the character will automatically be changed to a '-'.

When the Prompt Illegal Filenames menu option is selected, A64Tools will warn you when it encounters illegal characters in an Amiga filename.

Chapter 8
CLI Utilities

The A64 Package includes a number of utility programs that are designed to be used from the CLI or Shell. These utilities serve a variety of purposes, like formatting C64 disks, copy files from C64 drives to Amiga drives and file conversions. Most of the functions of these utilities can also be done with A64Tools (*See Chapter 7, A64Tools*). These utilities are provided for users who are familiar with the CLI.

Utility Summary

The following list gives a brief summary of the utilities included with The A64 Package. All of these utilities can be found on disk #2 in the drawer named "Utils." The heading HDW is short for HarDWare and is used to signify if the utility uses A64's hardware interface or not.

UTILITY NAME	HDW	FUNCTION
64Cmd	YES	Send a command to a C64 drive
64Dir	YES	Read the directory of a C64 disk
64Print	YES	Print an Amiga file to a C64 printer
64Status	YES	Read the status of a C64 drive
64ToAmiga	YES	Copy a file from a C64 drive to an Amiga drive
AmigaTo64	YES	Copy a file from an Amiga drive to a C64 drive
Pet2ASCII	NO	Convert a file from Pet ASCII to ASCII
ASCII2Pet	NO	Convert a file from ASCII to Pet ASCII
ScreenCode2ASCII	NO	Convert a file from C64 screen codes to ASCII
StripLoadAddr	NO	Strip the load address from a file

Utility Conventions

To get a utilities usage, enter the utility name followed by a '?'. When a utilities parameters are listed, parameters enclosed in < > are required and parameters enclosed in [ ] are optional. If you do not enter a value for optional parameters, the utility will use a default value.

64Cmd

Function: *Send a command to a C64 disk drive*
Usage: *64Cmd <command> [device]*
<command> drive command string
[device] drive device number (8-11, default = 8)

Format a disk: <command> = N0:disk name,2 digit id
Example: 64Cmd N0:testdisk,64
Result: Will format the disk in drive 8 with the name "testdisk" and an id of "64"

Rename a file: <command> = R0:new name=old name
Example: 64Cmd R0:testfile1=testfile2 9
Result: Will rename the file "testfile2" to "testfile1" on the disk in drive 9

Delete a file: <command> = S0:filename
Example: 64Cmd S0:testfile 8
Result: Will delete the file "testfile" on the disk in drive 8

64Cmd supports all of the standard Commodore DOS commands. For more information on C64 DOS commands, see your C64 disk drive user's manual.

64Dir

Function: Read the directory of a C64 disk
Usage: 64Dir <$pattern> [device]
<pattern> standard C64 DOS pattern ($ = show all files)
[device] drive device number (8-11, default = 8)

Example: 64Dir $
Result: List the entire directory of the disk in drive 8

Example: 64Dir $:a* 9
Result: List only the files that begin with "a" on the disk in drive 9

For more information on C64 DOS patterns see your C64 disk drive user's manual. The directory can be sent to a file by using standard AmigaDOS redirection.

64Print

Function: Print an Amiga text file to a C64 printer
Usage: 64Print <file> [device] [-B]
<file> name of the Amiga file you want to print
<device> printer device number (4 or 5, default = 4)
[-B] treat file as binary file. If this option is not specified, the file being printed will automatically be converted from ASCII to Pet ASCII

Example: 64Print df0:testfile
Result: Print the file "df0:testfile" to printer 4. The file will be converted from ASCII to Pet ASCII.

Example: 64Print RAM:testfile2 5 -B
Result: Print the file "RAM:testfile2" to printer 5. The file will not be converted.

64Status

Function: Read the status of a C64 disk drive
Usage: 64Status [device]
[device] drive device number (8-11, default = 8)

Example: 64Status
Result: Read the status of drive 8

Example: 64Status 9
Result: Read the status of drive 9

64ToAmiga

Function: Copy a file from a C64 drive to an Amiga drive
Usage: 64ToAmiga <source> <destination> [device] [-filetype]
<source> C64 file name
<destination> Amiga file name
[device] drive device number (8-11, default = 8)
[-filetype] filetype can be -P for PRG file, -S for SEQ file or -U for USR file (default is -P)

Example: 64ToAmiga testfile df0:testfile
Result: Copy the PRG file "testfile" from drive 8 to "df0:testfile"

Example: 64ToAmiga testfile2 dh0:testfile 9 -S
Result Copy the SEQ file "testfile2" form drive 9 to "dh0:testfile"

AmigaTo64

Function: Copy a file from an Amiga drive to a C64 drive
Usage AmigaTo64 <source> <destination> [device] [-filetype]
<source> Amiga file name
<destination> C64 file name
[device] drive device number (8-11, default = 8)
[-filetype] filetype can be -P for PRG file, -S for SEQ file or -U for USR file (default is -P)

Example: AmigaTo64 df0:testfile testfile
Result: Copy the file "df0:testfile" to drive 8 as a PRG file named "testfile"

Example: AmigaTo64 dh0:testfile2 testfile 9 -S
Result: Copy the file "dh0:testfile2" to drive 9 as a SEQ file named "testfile"

Pet2ASCII

Function: *Convert a file from Pet ASCII to standard ASCII*
Usage: *Pet2ASCII <source> [destination]*
<source> file to convert
[destination] destination file, if not specified, <source> file will be overwritten with converted file

Example: Pet2ASCII df0:testfile
Result: Convert the file "df0:testfile," from Pet ASCII to standard ASCII and overwrite the original file.

Example: Pet2ASCII df0:testfile df0:testfile2
Result: Convert the file "df0:testfile," from Pet ASCII to standard ASCII and save the result in "df0:testfile2" (the original file will be unchanged).

ASCII2Pet

Function: *Convert a file from standard ASCII to Pet ASCII*
Usage: *ASCII2Pet <source> [destination]*
<source> file to convert
[destination] destination file, if not specified, <source> file will be overwritten with converted file

Example: ASCII2Pet df0:testfile
Result: Convert the file "df0:testfile," from standard ASCII to Pet ASCII and overwrite the original file.

Example: ASCII2Pet df0:testfile df0:testfile2
Result: Convert the file "df0:testfile," from standard ASCII to Pet ASCII and save the result in "df0:testfile2" (the original file will be unchanged).

ScreenCode2ASCII

Function: *Convert a file from C64 screen codes to standard ASCII*
Usage: *ScreenCode2ASCII <source> [destination]*

<source> file to convert
[destination] destination file, if not specified, <source> file will be overwritten with converted file

Example: ScreenCode2ASCII df0:testfile
Result: Convert the file "df0:testfile," from C64 screen codes to standard ASCII and overwrite the original file.

Example: ScreenCode2ASCII df0:testfile df0:testfile2
Result: Convert the file "df0:testfile," from C64 screen codes to standard ASCII and save the result in "df0:testfile2" (the original file will be unchanged).

StripLoadAddr

Function: *Strip the load address from a file*
Usage: *StripLoadAddr <source> [destination]*
<source> file to strip load address from
[destination] destination file, if not specified, <source> file will be overwritten with converted file

Example: StripLoadAddr df0:testfile
Result: Strip the first two bytes from the file "df0:testfile" and overwrite the original file.

Example: StripLoadAddr df0:testfile df0:testfile2
Result: Strip the first two bytes from the file "df0:testfile" and save the result in "df0:testfile2" (the original file will be unchanged).

Chapter 9
Error Handling

All programs included with The A64 Package have extensive error handling and they all handle errors in a very similar way. Whenever an error occurs, the program will display an "Error Requester." Error Requesters will display a unique error number followed by several lines of text that will describe the problem and if possible, suggest a remedy. Error Requesters contain two gadgets. One gadget is labeled "Continue." Selecting this gadget will remove the Error Requester and if possible, allow the program to continue. The other gadget is labeled "Debug." Selecting this gadget will display an alert that shows the internal status of the program when the error occurred. This information will probably not mean anything to you, but it can be invaluable to us for tracking down errors. After viewing the alert, you can press either mouse button to allow the program to continue. There are two types of Error Requesters:

Fatal Errors

When a program encounters a situation that forbids the program from continuing normally, you will be prompted with a "Fatal Error Requester." When a Fatal Error occurs, after selecting one of the gadgets on the requester, the program will automatically quit.

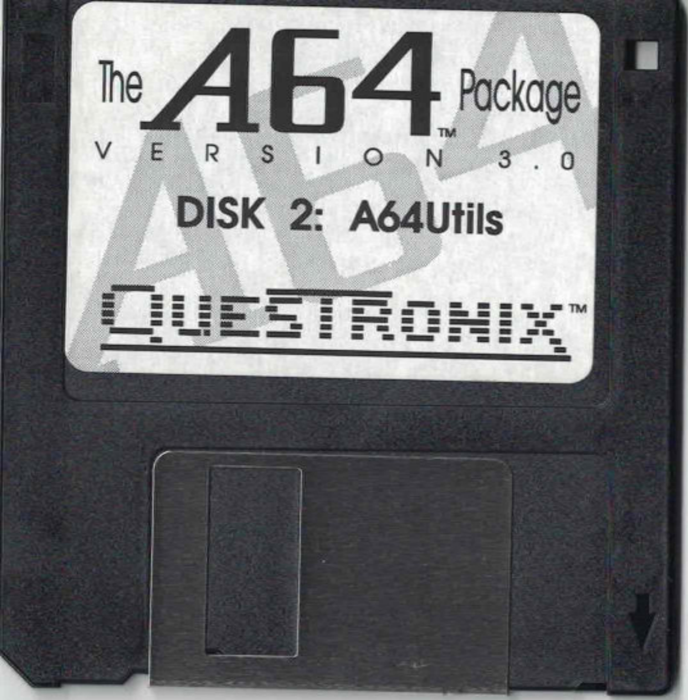
The most common cause of Fatal Errors is a lack of free memory, but there are many other situations that can cause them as well.

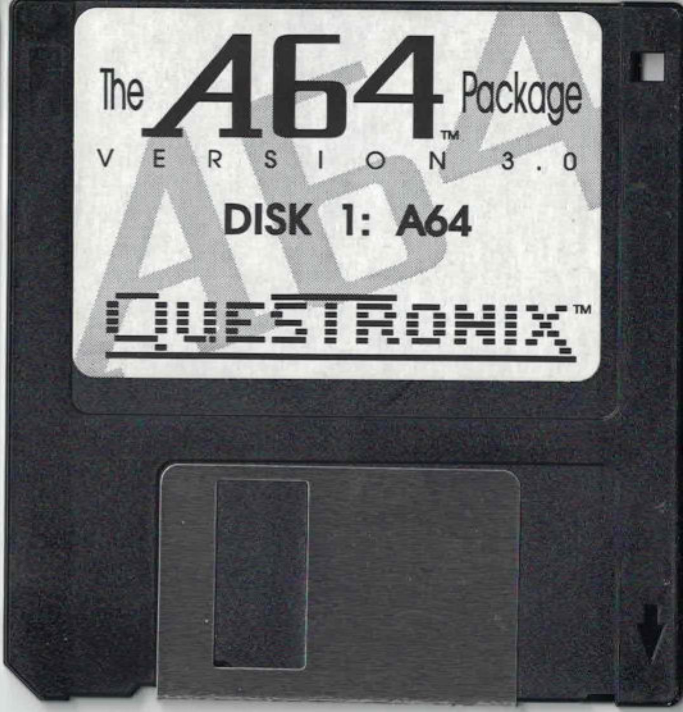
Non-Fatal Errors

When a program encounters some situation that needs attention, but is not serious enough for the program to quit, you will be prompted with a "Non-Fatal Error Requester." Non-Fatal Errors are a little more forgiving than Fatal Errors. You will usually be able to correct the situation that created the error, or at the very least, perform some action that will remove the error.

Error Reports

On disk #2 is a file named "ErrorReport." If you are experiencing problems with running a particular C64 program, or if you encounter an error that you can not correct, you can send us a copy of the Error Report. In the past, Error Reports have been a great source of information and have helped us to find and correct many problems. With a program as large and complex as A64, it is very difficult to test everything under all possible circumstances and it is impossible to test every single C64 program. You can help make A64 a better emulator by telling us about any problems that you encounter. We can't fix problems unless we know that they exist.


The A64™ Package
VERSION 3.0
DISK 2: A64Utils
QUESTRONIX™


The A64™ Package
VERSION 3.0
DISK 1: A64
QUESTRONIX™

The A64 Package V3 Registration

You must fill out this card and return it to QuesTronix to be eligible for technical support and updates.

First Name ____________________ Last Name ____________________

Address ____

City ____________________ State/Providence ____________________

Zip/Postal Code ____________________ Country (if not USA) ____________________

Day Phone (   ) ____________________

Place of Purchase ____________________ Phone (   ) ____________________

Purchase Date ____________________ Price ____________________

Where did you first here about A64? ____________________

What Amiga model do you own? ____________________

What other computers do you own? ____________________

What computer peripherals do you own? ____________________

____

What Computer magazines do you read? ____________________

____

What types of Amiga products would you like to see more of? ____________

____

Comments ____

____

____

____

____

Place
Stamp
Here

Product Registrations
P.O. Box 340265
Hartford, CT. 06134-0265
U.S.A.

**** COMMODORE 64 BASIC V2 ****
64K RAM SYSTEM 38911 BASIC BYTES FREE
READY.

Run C64 Software On Your Amiga!

NEW VERSION

The A64™ Package

VERSION 3.0

- Fully emulates the C64's sound, graphics, CIAs, Basic and machine language.
- Supports all C64 compatible disk drives and printers and includes a hardware adapter to connect them to your Amiga's parallel port.
- Fully integrates the C64 with the Amiga by supporting the use of Amiga devices with C64 programs, including disk drives, hard disks, printers and modems.
- Includes a complete file transfer utility for moving programs and files between C64 and Amiga disk drives.
- Compatible with all Amiga models, including the A1200 and A4000.
- Supports all current microprocessors, including the 68000, 68010, 68020, 68030 and 68040.

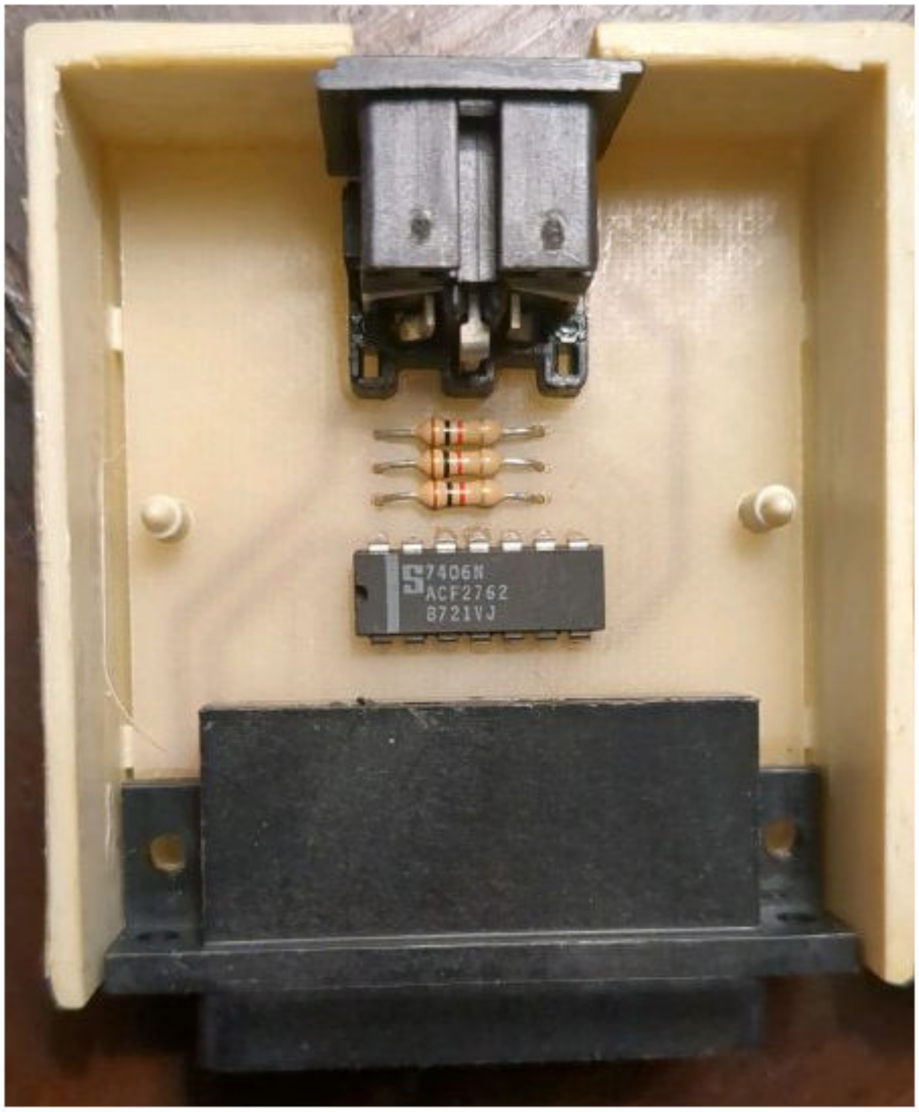
QUESTRONIX™

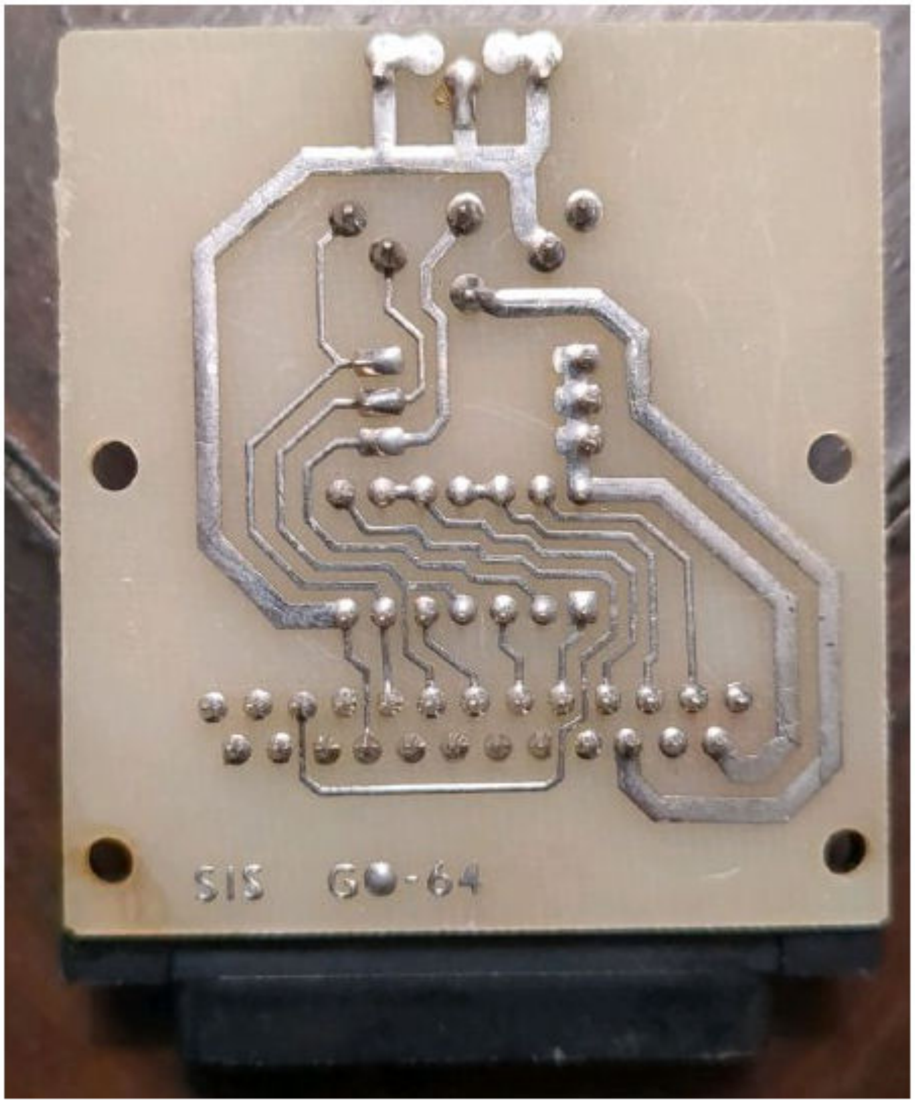
P.O. Box 340265
Hartford, CT 06134-0265
U.S.A.

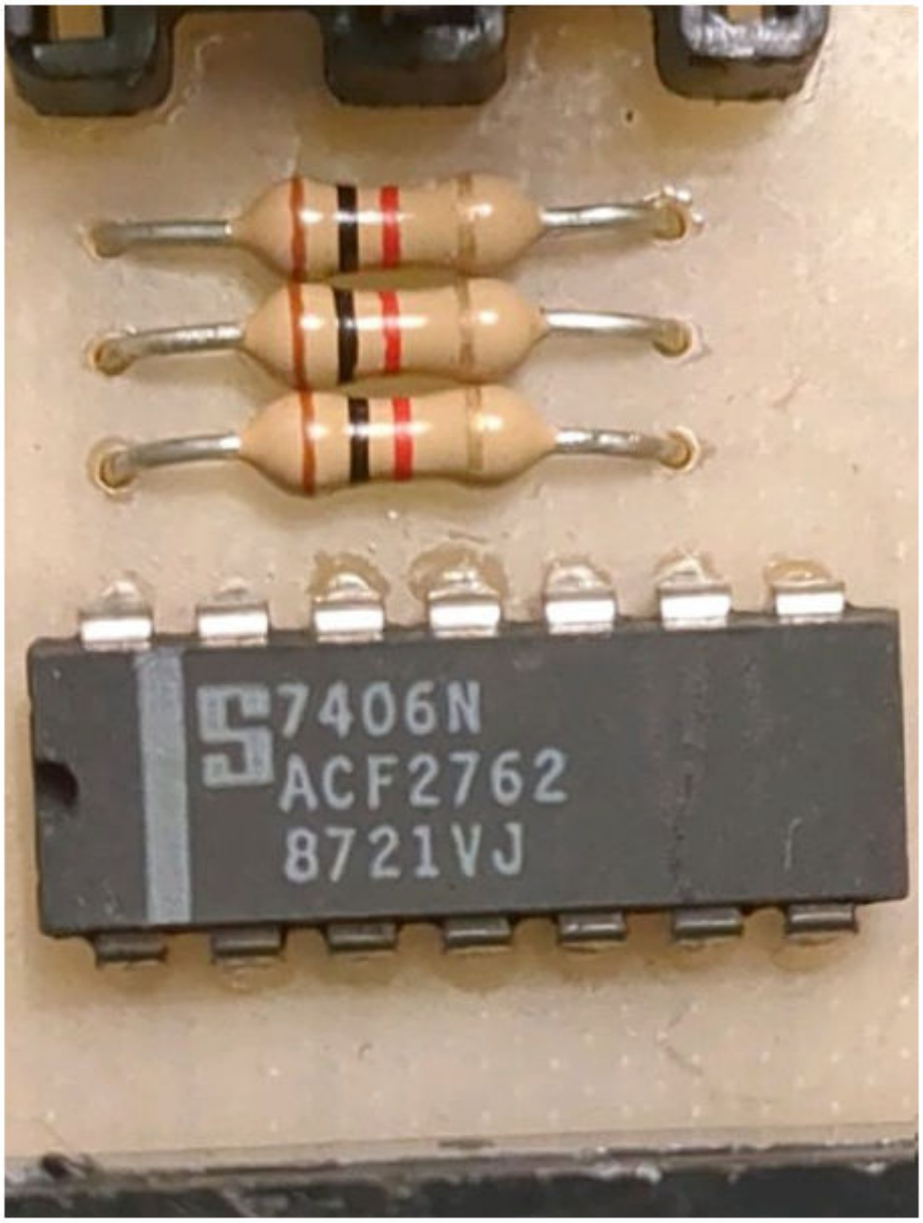
Requirements: Workbench V1.3 or later. 1mb RAM. 2mb recommended. C64 compatible disk drive and printer (optional). A1000 requires an additional adapter cable.

Amiga and Workbench are trademarks of Commodore-Amiga, Inc. C64 and Commodore 64 are trademarks of Commodore Electronics, Ltd. The A64 Package and A64 are trademarks of QuesTronix.

The A64 Package
VERSION 3.0
Run C64 Software On Your Amiga!
QUESTRONIX
USA
The A64 Package
VERSION 3.0
QUESTRONIX
The A64 Package
DISK 1: A64
QUESTRONIX
The A64 Package
DISK 2: A64Utils
QUESTRONIX
A64
QUESTRONIX


7406N
ACF2762
8721VJ


SIS GO-64


7406N
ACF2762
8721VJ