Documentation for the main tools in the Jaguar Developer's Kit is contained in separate chapters. This includes the following:

Madmac Macro Assembler
ALN Linker
DB Debugger

The documentation for some utilities may be provided in the same section as the documentation on the libraries or other tools they work with. If you don't see information on a particular utility here, please look in the appropriate sections of the **Libraries** chapter.

Some of the tools in the Jaguar Developer's Kit are used constantly, such as the Madmac assembler. Others are used much more rarely. For example, the XNOTES program that creates a NOTES.CNF file for the PARSE utility is not something you will need very often. The documentation for some of these tools are provided primarily in ASCII text files included with the program files. These files are found in the \JAGUAR\DOC directory of your Jaguar development system, or else in the subdirectory for that item (i.e. MUSIC.TXT inside \JAGUAR\MUSIC).

Note that the GASM macro assembler is no longer included as part of the distribution of tools in the Jaguar Developer Kit, and the section of documentation regarding GASM has been removed as well.

What's What: Tools & Related Files in \JAGUAR\BIN Directory

The table below describes in brief the tools available in the Jaguar Developer's Kit. Please note the following:

- Some tools named in the list may no longer be in general distribution, having been replaced by similar tools which serve the same purpose. For example, the BPEG image compression & decompression tools and library replaces the JAGPEG package. These tools are listed in a secondary table.
- Because Atari is constantly updating the Jaguar Developer's Kit, there will inevitably be new tools that either go along with or replace some of the ones listed below.
- It is assumed that your development system maintains the same directory hierarchy specified by the distribution archive files, and these files are located in a \JAGUAR\BIN directory on your system. Note that some filenames include path specifications for subdirectories of \JAGUAR\BIN.
- Atari distributes tools for several different development platforms. Note that some tools are not available on all platforms.
- The entries in the table are sorted according to filename, and are not grouped by platform, because some files, such as DB scripts, are not platform-specific.
- Some of the programs in \JAGUAR\BIN are meant to be called by other programs, and are not usually called directly by the user (although this may be possible, it's usually not desireable). This is noted in the file description where appropriate.

Currently Distributed Tools & Related Files

Filename	Platform	Description
32RTM.EXE	MSDOS	Used for 32-bit DOS Protected Mode support (DPMI) by certain tools. *(This is loaded automatically by those tools that require it.)*
3DS2JAG.EXE	MSDOS	3D-Studio to Jaguar 3D object conversion utility
3DS2JAG.TTP	Atari	3D-Studio to Jaguar 3D object conversion utility
AGPU\2.6\AS.EXE	MSDOS	Stub program used by GCC to call MADMAC assembler for GPU/DSP code. *(Normally called by GCC.EXE driver program, not directly by user.)*
AGPU\2.6\CC1.EXE	MSDOS	GCC GPU/DSP code generation module. *(Normally called by GCC.EXE driver program, not directly by user.)*
AGPU\2.6\CPP.EXE	MSDOS	GCC C Preprocessor for GPU/DSP. *(Normally called by GCC.EXE driver program, not directly by user.)*
AGPU\2.6\SPECS	MSDOS	GCC C compiler configuration file for GPU/DSP
ALN	Linux	ALN Linker
ALN.EXE	MSDOS	ALN Linker
ALN.TTP	Atari	ALN Linker
AR.EXE	MSDOS	Object Module Archive Librarian for BSD-format object modules
AR68.TTP	Atari	Object Module Archive Librarian for Alcyon-format object modules
CBPEG	Linux	BPEG Image Compression utility
CBPEG.EXE	MSDOS	BPEG Image Compression utility
CBPEG.TTP	Atari	BPEG Image Compression utility
CPP.EXE	MSDOS	C Language Preprocessor (used by GCC)

Currently Distributed Tools & Related Files		
Filename	**Platform**	**Description**
CW3211.DLL	MSDOS	Used for 32-bit DOS Protected Mode support (DPMI) by certain tools. (*This is loaded automatically by those tools that require it.*)
DEMO.DB	DB Script	Sample DB script file
DMP.EXE	MSDOS	Hex Dump Utility
DOINDEX.TTP	Atari	Index Creator (creates index files needed by ALN to go along with Alcyon-format archive libraries)
DPMI32VM.OVL	MSDOS	Used for 32-bit DOS Protected Mode support (DPMI) by certain tools. (*This is loaded automatically by those tools that require it.*)
DUMP.TTP	Atari	Hex Dump Utility. Displays a hexadecimal dump of specified file
FGREP.EXE	MSDOS	Fast General Regular Expression Parser utility.
FILEFIX.EXE	MSDOS	Filefix utility. Breaks down ABS or COF executable file into raw binary image files for each program segment.
FILEFIX.TTP	Atari	Filefix utility. Breaks down ABS or COF executable file into raw binary image files for each program segment.
FILL.DB	DB Script	Script file for fast memory fill with DB
FILLCODE.DAT	DB Data	Jaguar program code downloaded by FILL.DB
FLASH.COM	MSDOS	Flash ROM Writer Utility. Writes ROM image files to Flash ROM cartridges
FLMINFO.EXE	MSDOS	Jaguar Cinepak Film Information Browser
GCC.EXE	MSDOS	GCC C compiler driver program. This executes the various programs that make up the GCC C compiler.
GO32.EXE	MSDOS	Used for 32-bit DOS Protected Mode support (DPMI) by certain tools. (*This is loaded automatically by those tools that require it.*)
GPU.DB	DB Script	DB Script file for GPU/DSP Debugging Functions (*See also NGPU.DB for newer version.*)
GULAM.G	Atari	Starup Script for GULAM command line interpretter
GULAM.PRG	Atari	GULAM command line shell
LOADFIX.EXE	MSDOS	Utility to convert old Jaguar Sound Tool files to new format
LOADFIX.TTP	Atari	Utility to convert old Jaguar Sound Tool files to new format
LS.EXE	MSDOS	Unix-style Directory Listing Utility
EXE	MSDOS	LZSS For Jaguar compression utility
2.6\AS.EXE	MSDOS	Stub program used by GCC to call MADMAC assembler for Motorola 680x0 code. (*Normally called by GCC.EXE driver program, not directly by user.*)
2.6\CC1.EXE	MSDOS	GCC Motorola 680x0 code generation module (*Normally called by GCC.EXE driver program, not directly by user.*)
2.6\CPP.EXE	MSDOS	GCC C Preprocessor for Motorola 680x0. (*Normally called by GCC.EXE driver program, not directly by user.*)
2.6\SPECS	MSDOS	GCC C compiler configuration file for Motorola 680x0
	Linux	MADMAC Macro Assembler
E	MSDOS	MADMAC Macro Assembler
	Atari	MADMAC Macro Assembler
XE	MSDOS	Make / Program Builder Utility
TP	Atari	Make / Program Builder Utility
Y.BAT	MSDOS	Batch file to run TGA2CRY Utility (MSDOS command processor)
Y.G	Atari	Batch file to run TGA2CRY Utility (Gulam shell on Atari)
M.BAT	MSDOS	Batch file to use FILEFIX to create a ROM image file
EXE	MSDOS	Jaguar MIDI File Merge Utility
TTP	Atari	Jaguar MIDI File Merge Utility
XE	MSDOS	Converts raw sound sample files to AIFF format
TP	Atari	Converts raw sound sample files to AIFF format
XE	MSDOS	Converts stereo raw sound sample files to mono

LZJAG
M68K\2
M68K\2
M68K\2
M68K\2
MAC
MAC.EXE
MAC.TTP
MAKE.EX
MAKE.TT
MAKECR
MAKECR
MAKERO
MERGE.E
MERGE.T
MKAIF.EX
MKAIF.T
MONO.EX

Currently Distributed Tools & Related Files		
Filename	**Platform**	**Description**
MONO.TTP	Atari	Converts stereo raw sound sample files to mono
NGPU.DB	DB Script	DB Script for GPU/DSP debugging
NOTES.CNF	data file	Default configuration file for PARSE Utility
OD.DB	DB Script	DB Script for Object List display
PARSE.CNF	data file	Default configuration file for PARSE Utility
PARSE.EXE	MSDOS	Parses standard MIDI soundtrack files into format for Jaguar Music Driver.
PARSE.TTP	Atari	Parses standard MIDI soundtrack files into format for Jaguar Music Driver.
RANLIB.EXE	MSDOS	Utility for indexing & time/date-stamping archive files created with AR.EXE
RDB.RC	DB Script	Startup Script used by DB
RDBJAG	Linux	Command Line Interface version of DB Debugger
RDBJAG.EXE	MSDOS	Command Line Interface version of DB Debugger
RDBJAG.TOS	Atari	Command Line Interface version of DB Debugger
REGDUMP.BIN	DB Data	GPU/DSP Register dump code used by DB
ROMSPLIT.EXE	MSDOS	Splits a ROM image file into separate sections for each chip of a ca
SIZE.EXE	MSDOS	Displays code & data segment sizes of executable program, and optionally dumps the symbol list.
SIZE.TTP	Atari	Displays code & data segment sizes of executable program, and optionally dumps the symbol list.
SNDCMP.EXE	MSDOS	Compresses 16-bit raw sound sample files to 8-bit using square roo method (which are expanded back to 16-bit upon playback).
SNDCMP.TTP	Atari	Compresses 16-bit raw sound sample files to 8-bit using square roo method (which are expanded back to 16-bit upon playback).
STRIP68.TTP	Atari	Removes symbol table from executable program file
STRIPAIF.EXE	MSDOS	Strips the AIFF header information from a sound sample file to resu raw sample file.
STRIPAIF.TTP	Atari	Strips the AIFF header information from a sound sample file to resu raw sample file.
TGA2CRY.EXE	MSDOS	Converts Targa or GIF-format picture files into source code or raw data, in choice of RGB or CRY formats. Also has filtering, resizing, other image manipulation options.
TGA2CRY.TTP	Atari	Converts Targa or GIF-format picture files into source code or raw data, in choice of RGB or CRY formats. Also has filtering, resizing, other image manipulation options.
UNCMP.EXE	MSDOS	Decompresses sound files compressed by SNDCMP back to 16-bit
UNCMP.TTP	Atari	Decompresses sound files compressed by SNDCMP back to 16-bit
WAVEFM.EXE	MSDOS	Creates wave form files used by Jaguar Synth
WAVEFM.TTP	Atari	Creates wave form files used by Jaguar Synth
WDB	Linux	Window / Menu / Mouse Interface version of DB Debugger
WDB.EXE	MSDOS	Window / Menu / Mouse Interface version of DB Debugger
WINC.DB	data file	Sample WDB Script showing window manipulation
WINDPMI.386	MSDOS	Used for 32-bit DOS Protected Mode support (DPMI) required by ce tools. (*This is loaded automatically by those tools that require it.*)
WINS.DB	DB Script	Sample WDB Script showing window manipulation
WSAMPLE.DB	DB Script	Sample WDB Script showing window manipulation
XNOTES.EXE	MSDOS	Utility to create NOTES.CNF file for various sample rates.
XNOTES.TTP	Atari	Utility to create NOTES.CNF file for various sample rates.
ZAPJAG.DB	DB Script	DB Script for clearing entire Jaguar memory

Tools & Related Files No Longer In Main Distribution

Filename	Platform	Description	Replaced by
GASM.EXE	MSDOS	GASM Macro Assembler	*MADMAC*
GASM.TTP	Atari	GASM Macro Assembler	*MADMAC*
JCJPEG.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user	*BPEG*
JCJPEG.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam scriptfiles, not directly by user	*BPEG*
JMAKEQ.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JMAKEQ.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JMERGE.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JMERGE.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JMERGEH.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JMERGEH.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JMERGEQ.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user	*BPEG*
JMERGEQ.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JQUAD.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JQUAD.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JSPLIT.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JSPLIT.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JSPLITH.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JSPLITH.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JSPLITQ.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JSPLITQ.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
JSTRIP.EXE	MSDOS	Component of JAGPEG Compression Utilities normally called by TGAJAG driver program, not directly by user.	*BPEG*
JSTRIP.TTP	Atari	Component of JAGPEG Compression Utilities normally called by TGAJAG Gulam script files, not directly by user.	*BPEG*
LOCATE.EXE	SDOS	Utility to relocate JAGPEG decompression code table	*BPEG*
LOCATE.TTP	ari	Utility to relocate JAGPEG decompression code	*BPEG*
LTXCONV.EXE	SDOS	Utility to convert GASM Macro Assembler output to linkable format	*MADMAC*
LTXCONV.TTP	ari	Utility to convert GASM Macro Assembler output to linkable format	*MADMAC*
TGAJAG.EXE	SDOS	Driver program to convert Targa-format picture files into JAGPEG compressed image files using the JAGPEG utilities	*BPEG*

Tools & Related Files No Longer In Main Distribution			
Filename	**Platform**	**Description**	**Replaced by**
TGAJAG.G	Atari	Batch file that drives the JAGPEG utilities to convert Targa-format picture files into JAGPEG compressed image files (Gulam shell for Atari)	*BPEG*
TGAJAGH.G	Atari	Batch file that drives the JAGPEG utilities to convert Targa-format picture files into JAGPEG compressed image files (Gulam shell for Atari)	*BPEG*
TGAJAGQ.G	Atari	Batch file that drives the JAGPEG utilities to convert Targa-format picture files into JAGPEG compressed image files (Gulam shell for Atari)	*BPEG*

AR Archive Librarian

Note: The AR archive librarian for BSD-format archive libraries is available only for MSDOS systems. The AR68 archive librarian for Alcyon-format archive libraries is available only on the Atari/TOS platform. The documentation below is originally for AR68, but the basic functionality and operation of both programs is the same.

The AR archive librarian creates and maintains archive libraries of linkable object modules. It allows you to create these libraries and add, replace, delete, list, or extract object modules.

Usage

```
AR68 <options> ARCHIVE OBMOD1 [OBMOD2...] [>filespec]
```

be created as

ine options are

library in the

it uses as a

TMP is not

error messages

itional object

ies as you like,

components

. Redirects the

d output device,

the AR68

All command-line options must be specified first, followed by the name of the archive to be updated, followed by the a list of one or more filenames of object modules. Command-l not case-sensitive.

AR68 sequentially parses the command line once. AR68 acts upon object modules in the order they are specified on the command line.

When AR68 processes a command, it creates a temporary file called AR68.TMP. which scratch pad. After the operation is complete AR68 erases AR68.TMP. However, AR68.T always erased if an error occurs. If this occurs, erase AR68.TMP and refer to the list of output by AR68.

The *ARCHIVE* parameter is the filename of the archive library.

The *OBMOD1* parameter is the filename of the first object module being acted on. Add module filenames may optionally follow the first. You can specify as many object modu provided the command line does not exceed 127 bytes. The delimiter character between consists of one or more spaces.

The *>FILESPEC* parameter is the name of a file used for output with certain commands output to the file specification you specify, rather than sending the output to the standard which is usually the console device (CONSOLE). You can redirect the output for any of commands described below.

Command Line Options

Option	Description

D	The D command deletes from the library one or more object modules. Can be used with the **V** option (see description below). For example:
	
	`ar68 dv myrah.lib orc.o`
	`c red.o`
	`c blue.o`
	`d orc.o`
	`c white.o`
	
	The ORC.O object module is being deleted from the archive library MYRAH.LIB, and the RED.O, BLUE.O, and WHITE.O modules are left untouched.

R	The R command creates a library when the one specified in the command line does not exist, or replaces or adds object modules to an existing library. You must specify one or more object modules.
	
	You can replace more than one object module in the library by specifying the module names in the command line. However, when the library contains two or more modules with the same name, AR68 replaces only the first module it finds that matches the one specified in the command line. AR68 replaces modules already in the library only if you specify their names prior to the names of new modules to be added to the library. For example, if you specify the name of a module you want replaced after the name of a module you are adding to the library. AR68 adds both modules to the end of the library.
	
	By default, the **R** command adds new modules to the end of the library. The **R** command adds an object module to a library, instead of replacing one, if:
	
	• The object module does not already exist in the library.
	• You specify the **A** option in the command line.
	• The name of the module follows the name of a module that does not already exist in the library.
	
	For example:
	
	`ar68 rv junk.lib nail.o wrench.o`
	`c saw.o`
	`c ham.o`
	`r nail.o`
	`c screw,o`
	`a wrench.o`
	
	The R command replaces the object module NAIL.O and adds the module WRENCH.O to the library JUNK.LIB. The V option lists object modules in the library and indicates which modules are being replaced or added. Each object module that is replaced is preceded with the lowercase letter r and each onc that is added is preceded with the lowercase letter a.

T

The **T** command requests that AR68 print a table of contents or a list of specified modules in the library. The **T** command prints a table of contents of all modules in the library only when you do not specify names of object modules in the command line. It supports the **V** option. For example:

```
ar68 tv wine.lib
rw-rw-rw-      0/0      6818  rose.o
rw-rw-rw-      0/0      2348  white.o
rw-rw-rw-      0/0      396   red.o
```

The **T** command prints a table of contents in the library WINE.LIB. In addition to listing the modules in the library. the **V** option requests the size of each module. The character string "***rw-rw-rw- 0/0***" that precedes the module size is meaningless for GEMDOS. However. if the file is transferred to a UNIX... system. the character string denotes the file protection and file owner. decimal number that precedes the object module name indicates the

e

py of an object module in the library to the standard output, which

nless the output is redirected by using the *>filespec* parameter on

mand allows you to extract a copy of a module from a library and

rite it to another disk, as shown below. For this command to be

output using the *>filespec* parameter.

```
b:\root\newd\file.o
```

ect module NOW.O from the library GO.LIB to the file FILE.O in

y.

opy of one or more object modules from a library and writes them

ry. If no object modules are specified in the command line, the **X**

each module in the library. The **X** command supports the **V**

```
o ham.o screw.o
```

s a modifier for the **R** option. It specifies that one or more object

the library. The specified files will be added to the library following

by the *opmod* parameter, which is expected to be the name of an

library. The *opmod* parameter always comes after all the

name of the archive. For example:

```
ah.lib work.o mail.o
```

it should add the object modules WORK.O and MAIL.O after the

ry MYRAH.LIB. The **V** option tells AR68 to list all the modules in

. New modules are preceded by the lowercase letter "**a**" and

ded by the lowercase letter "**c**".

The size specified by the

number of bytes in the modul

W

The **W** command writes a cop

will normally be the screen u

the command line. This comm

rename the copy when you w

useful, you must redirect the

```
ar68 w go.lib now.o >
```

This writes a copy of the obje

the B:\ROOT\NEWD director

X

The **X** command extracts a c

to the current default director

command extracts a copy of

option. For example:

```
ar68 xv junk.lib saw.
x     saw.o
x     ham.o
x     screw.o
```

A[V] ***opmod***

The **A** option is used only as

modules are to be added to

the object module specified

object module already in the

specified options, before the

```
AR68 rav sdav.o rnyra
c much.o
c sdav.o
a work.o
a mail.o
c less.o
```

The **RAV** options tell AR68

module SDAV.O in the libra

the library after this is done

existing modules are

V	The **V** option lists the modules in the library and indicates the result of the operation performed on the library. The **V** option can only be used with one of the other option. In the resulting listing, each object module name will have a letter code in front indicating what action was taken: c No action taken, object module not updated, deleted, or added. a Object module added to archive library.

dule deleted from archive library. d Object mod

dule replaced in archive library

F filename

Specifies the path to the directory in which the temporary file created by AR68 resides. If no path name is specified. the current default directory is used. AR68 creates a temporary file called AR68.TMP that is used as a scratch pad area.

rrors

8 incurs an error during an operation, the operation is not completed. The original library is d if the operation would have modified the library. Thus, no modules in the library are laced, added, or extracted.

pecify the *>filespec* parameter in the command line to redirect the output, and one or more , the error messages are sent to the output file. Thus, you cannot detect the errors without or printing the file to which the output was sent. If the contents of the output file is an object W command), you must use the DUMP utility to read any error messages.

ns two types of fatal error messages: diagnostic and logic. Both types of fatal error messages console as they occur.

gnostic Error Messages

ot in archive file
module indicated by the specified filename is not in the library. Check the filename before the command line.

AR68 E

When AR68 not modifie deleted, rep

When you s errors occur displaying o file (see the

AR68 retur show at the

Fatal Diag

filename no
The object r you reenter

cannot create filename
The path name for the file indicated by the specified filename is invalid or the disk to which AR68 is writing is full. Check the path name. If it is valid, the disk is full. Erase unnecessary files, if any, or insert a new floppy disk before you reenter the command line.

cannot open filename
The file indicated by the specified filename cannot be opened because the filename or the path name is incorrect. Check the path name and the filename before you reenter the command line.

invalid option flag: x
The symbol, letter, or number in the command line indicated by the variable x is an invalid option. Refer to the Command Line Options section for an explanation of the AR68 command line options. Specify a valid option and reenter the command line.

not archive format: filename
The file indicated by the specified filename is not a library. Ensure that you are using the correct filename before you reenter the command line.

not object file: filename
The file indicated by the specified filename is not an object file, and cannot be added to the library. Any file added to the library must be an Alcyon-format object file. Assemble or compile the file before you reenter the AR68 command line.

one and only one of DRTWX flags required
The AR68 command line requires one of the D, R, T, W, or X commands, but not more than one. Reenter the command line with the correct command.

filename not in library
The object module indicated by the specified filename is not in the library. Ensure that you are requesting the filename of an existing object module before you reenter the command line.

Read error on filename
The file indicated by the specified filename cannot be read. This message means one of three things: the file specified is corrupted; a hardware error has occurred; or when the file was created. it was not correctly written by AR68 due to an error in the internal logic of AR68.

Cold start the system and retry the operation. If you receive this error message again. you must erase and recreate the file. Use your backup file, if you maintained one.

ull. Erase unnecessary files, if any, or insert a new floppy disk before you reenter

]TWX[F D:] [OPMOD] ARCHIVE OBMOD1 [OBMOD2...] [>filespec]
: a syntax error in the command line. The correct format for the command line is
e options in brackets.

me
58 is writing the file indicated by the specified filename is full. Erase unnecessary
new floppy disk before you reenter the command line.

gic Error Messages

ssages that indicate fatal errors in the internal logic of AR68:

filename
ibrary
empname
eate--library is in filename

temp file write error
The temporary file is fu
the command line.

usage: AR68 DR[AV
This message indicates
given. with the possibl

Write error on filena
The disk to which AR6
files, if any. or insert a

AR68 Internal Log

The following are mes

cannot reopen
seek error on li
Seek error on t
Unable to recr

For the last error, **Unable to recreate--library is in filename**, you should rename the temporary file indicated by the variable filename. AR68 used the library to create the temporary tile, then deleted the library in order to replace it with the updated temporary file. This error occurred because AR68 cannot write the temporary file back to the original location. The entire library is in the temporary file.

Dump Utility

The DUMP utility is a very simple hex-dump program that takes a filename and optionally a starting file position as its input parameters:

```
dmp <filename> [fileposition]
```

The *fileposition* parameter indicates the offset from the start of the file where the hex dump will begin.

Size Utility

SIZE is a utility that examines an executable program file or linkable object module file and prints out information about the TEXT, DATA, and BSS segments of the file (size, starting address, etc.)

Please note that some information is not appropriate for some files. For example, segments within a linkable object module do not have a start address until they are linked together into a program file.

```
size [-s] [-sd] [-v] <file>
```

Option	Description
-s	Show symbols in file. The symbols will be sorted alphabetically. The information shown is the symbol value, symbol name, and symbol type. Symbols with the same name will be skipped (usually these are local labels which are used in different routines, equates included into several different source code files, or else special source-level information used by the debugger).
-sd	Same as the **-s** flag, except that duplicate symbol names will not be skipped.
-v	When showing symbols, sort by value, not name.

The parameter *file* is the filename of the file to be analyzed. SIZE will first look for the filename and extension exactly as specified. If no extension is found, it will then try extensions of .COF and .ABS (in that order). SIZE understands the following file formats:

Alcyon/DRI format executables. (These normally use a file extension of *.ABS)

COFF encapsulated format executables. (These normally use a file extension of *.COF)

Alcyon/DRI or BSD format object module files. (These normally use a file extension of *.O, *.OJ, or *.OT. SIZE will not automatically look for these extensions; you must specify the extension on the commandline.)

Archive libraries created by AR or AR68 are not recognized by this version of SIZE.

Filefix Utility

The FILEFIX utility converts a Alcyon/DRI-format (*.ABS) or COFF-format (*.COF) absolute position executable program file output by the ALN linker into separate files containing the raw data for the TEXT and DATA sections of the program, and a symbol table containing the symbol information for the program, and an RDBJAG script file for loading it all into the ALPINE board of a Jaguar Development System. Optionally, FILEFIX can instead create ROM image files that contain a raw binary image of what a ROM cartridge of the program would look like.

```
filefix [options] filename
```

filename An Alcyon/DRI or BSD/COFF format absolute-position executable file. A filename extension of .COF or .ABS is assumed if none is given. (i.e. "FILEFIX testprog\" will look for <testprog>, then <testprog.cof>, then <testprog.abs>, before giving up.

Command Line Options

Switch	Description
-q	Quiet mode, don't print information about executable file.
-r *romfile*	Create ROM image file named *romfile* from executable The DATA segment must not overlap or come before the TEXT segment. If the DATA segment is not contiguous with the TEXT segment, then zero bytes will be written to the file between the end of the TEXT segment and the start of the DATA segment.
-rs *romfile*	Same as **-r**, except also create DB script to load and run file.
-p	Pad ROM file with zero bytes to next 2mb boundary. This must be used along with the -r or **-rs** switch.
-p4	Same as **-p**, except pads to a 4mb boundary. This must be used along with the **-r** or **-rs** switch.

Unless you have specified the **-r** or **-rs** command line switches, the output files created will be *filename*.TXT (the program's TEXT segment), *filename*.DTA (the program's DATA segment), *filename*.SYM (the program's symbol table, if the source is not a COFF-format executable), and *filename*.DB (a DB script file to load everything), where *filename* is the root portion of the input filename. If you use the **-r** or **-rs** command line switches, the output filename must be specified.

Note: If the input filename supplied to FILEFIX has a filename extension, then FILEFIX will look specifically only for that file. However, if you leave off the extension, it will look for *filename*.COF and then *filename*.ABS.

Note: The symbol table file is not output for COFF-format executables. The DB script file output by FILEFIX will not reference it. Instead, it references the original executable file, which has the symbol information inside. Also, for either DRI or COFF-format files, if the program's TEXT and/or DATA segments are empty, then no output file will be created, and the script file will not reference the output files.

ity

y is a Fast General Regular Expression Parser. That's UNIX-speak. In English, it's a
ches text files for a specified string expression. The FGREP utility supplied in the
s Kit is a pretty standard version of GREP, so if you're familiar with another version,
works mostly the same way. Strictly speaking, FGREP is not limited to searching text
iour can be somewhat unpredictable when searching binary files.

```
ns...] [pattern] [filelist]
```

ne Options

ds a number of different switches that alters its mode of operation. None are normally

escription
ith each output line, print the block number in which the line started.
int the number of matches, rather than the lines of text themselves.
he following argument is the search pattern. (Useful when the pattern itself starts with the '-' haracter.)
he next argument is the filename of a text file containing a list of different patterns, eparated by newlines. In this instance, no pattern is specified on the commandline.
hen more than one source file is specified, output lines normally include the filename. This ption supresses this.
int the name of each file that contains matches for the pattern, rather than the lines emselves. This is useful in creating lists of files for a batch operation.
hen a line is printed, also print the line number within the file.
upress all output, just return exit status.
int a line only if the pattern is not found in the line (the opposite of the normal operation).
owercase letters in the pattern match either lowercase or uppercase characters in the ource file. However, any characters following the "/" escape character must match exactly.

pattern is a string expression with optional wildcards that FGREP searches for in the
ce files. Note that depending on the options used, it may sometimes be necessary to
ose your patterns in double quotation marks. Wild cards can include:

Description
tch the beginning of a line, unless it appears immediately after '['
tch the end of a line
tch zero or more repetitions of the preceeding character. Basically, match anything.
tch any single character except newline
tch any one of the enclosed characters. Ranges of letters or digits may be indicated by ng '-' (i.e. [1-9] matches any character in "123456789").
tch any character that is not one of the enclosed characters. Ranges of letters or digits y be indicated by using '-'.)
regard special meaning of the character '*c*'. (i.e. "*" would mean match the asterisk racter rather than using it as a wildcard.)

Fgrep Utili

The FGREP utilit
program that searc
Jaguar Developer'
this one probably
files, but it's behav

```
fgrep [optior
```

Command Li

FGREP understanc
required.

Options	D
-b	W
-c	Pr
-e	Th ch
-f	Th se
-h	W op
-l	Pr th
-n	W
-s	Su
-v	Pr
-y	Lo sc

pattern The
sour
encl

Wildcard	
^	Ma
$	Ma
*	Ma
. (period)	Ma
[chars]	Ma usi
[^chars]	Ma ma
\c	Dis cha

Wildcard	Description
|	Match the preceding pattern or the following pattern. For example,. **red|blue** would match either "**red**" or "**blue**". A newline within the pattern has the same meaning as '|'.
+	Match one or more occurances of the previous pattern element. Similar to the '*' wildcard, except at least one occurance is required instead of zero or more.
?	Match zero or one occurances of the previous pattern element.
(...)	Parenthesis are used to group patterns. For example **(abc)+** matches a sequence of one or more occurances of any of the three letters 'a', 'b', or 'c'.

filelist — A list of one or more filenames to be searched. If no file is specified, FGREP takes characters from the standard input device.

Examples:

```
fgrep A1_BASE *.s
```

This would search all files in the current directory that have filename extensions of .S, and print the filename of any lines that included "A1_BASE" in them.

```
fgrep -n dc\.[bwl] *.s
```

This would search all files in the current directory that have filename extensions of .S, and print the filename and line number of any lines that included "dc.b" or "dc.w" or "dc.l" in them.

LS Utility

The LS utility is a UNIX-style LiSt files utility. It has several advantages over the standard MS-DOS 'DIR' command, including the ability to search directories recursively.

```
ls [-?alrstxzAR1] [path1...] [path2...]
```

Option	Description
-?	HELP... print USAGE
-a	List all files, including hidden and system files, ".", and ".."
-l	Long listing form (extra information)
-r	Reverse order of sorting
-s	Display size of each file in kilobytes, and total for each directory
-t	Sort by time/date (latest first)
-x	Sort by extension
-z	Sort by size
-A	List all files except "." and ".."
-R	List subdirectories recursively
-1	Display 1 entry per line of short form

If you use multiple options together, you can use just one "-" character at the beginning. For example:

```
ls -l -t
```

and

```
ls -lt
```

would produce the same results and provide a long listing of files sorted by their time/date stamp.

Make Utility

The MAKE program is program-building utility that originated in the UNIX world, but which has since spread to just about every kind of computer system there is. In a nutshell, MAKE checks the time/date stamp of your source code files and the cooresponding object code files, and recompile and/or reassembles any source code files that have changed since they were last compiled. Then it also links the new program file as necessary.

A special script file, known as a MAKEFILE (and usually named MAKEFILE as well), tells the MAKE utility the names of your source code files, your target program name, and what commands a necessary to turn your source code into object code and link everything into a program.

The version of MAKE supplied with the developer's kit is a pretty standard version of MAKI is one thing to watch for, however. When using the "\" character, MAKE always interprets t line-continuation character, even when it occurs other than at the end of a line. If you need t path specifications in your makefile, you may need to work around this. With many of the to supplied with the developer's kit, you can use a "/" character in place of the "\" character with

re

E.[1] There
his as a
o include
ools
hout any

Usage:

```
3DS2JAG [options] filename
```

filename The complete filename for an AutoDesk 3D-Studio object file (*.3DS) to be converted

Command Line Options

Option	Description
-f	Combines faces of the model to convert adjacent triangle shaped faces to rectangular faces yet. Note: This does not yet work reliably as of the current version when this was written.
-l label	Specifies the label for the object the label is an identifier string. An optional number tag can be added using the "-n" option below. Default: <label>
-n	No Normals Option. Supresses the output of the normals in the face list.
-v	Consolidate vertices option. Consolidates duplicate vertices in output file.
-z	Zdyble Option. This is a slightly different output format of the face list. The first word in the face list is the texture index. If it's $FFFF the face is not texture mapped and the second word is the color information. Otherwise the second word is an index into the texture points array. The third word is the number of vertecies.

Source Code

The source code for the 3DS2JAG utility is available to developers upon request, with the restriction that you must supply Atari with any modifications (source code and executable) that you create.

Parse Utility

The PARSE utility is used to convert standard MIDI files into a format that can be used with the Jaguar Music Driver and Synthesizer. The output of the parser is a MADMAC assembler source file (ASCII) containing the sound data for the synthesizer in assembly language format. This file has to be assembled and linked in with your program, playing the music.

Usage:

```
    parse [options] [inputname]
```

Command Line Options

Option	Description
-q	Quiet mode, suppress MIDI notes on/off messages.
-o	Specify output filename, must be followed by a valid filename specification. If the "-o" option is not used, the filename of the output file will be TEST.OUT.
-n *x*	Set the number of voices to be used to *x*.
-x *n*	Add offset *n* to the used voices. Voices lower than *n* will not be used.
-z *n*	Set down scaling factor for the MIDI volume command to *n*. This is useful to avoid an overflow of the volume. The default is 256.

The *inputname* parameter is the filename of the MIDI file. If no filename is provided, PARSE looks for a file called `TEST.MID'.

The created output file will have a list of assembly **`dc.l'** statements containg the music data for the synthesizer. The global pointer *scoretab* points to the beginning of the music data.

Configuration Files

The files PARSE.CNF and NOTES.CNF allow you to configure the parser by changing their contents. The PARSE.CNF file gives you the ability to have a certain pitch range change to a specific patch. Each line in PARSE.CNF contains one pitch range. The format of the line is:

Meaning:	channel:	pitch_range_start - pitch_range_end	patch	pitch_offset
Example:	0:	2 - 20	24	64

The *channel* parameter specifies the MIDI channel (minus 1) that the rest of the line affects. In the example this would be channel 0. The *pitch_range_start* and *pitch_range_end* parameters specify the range of notes affected by the *pitch_offset* parameter. In this example this would be 2 through 20. The *patch* parameter indicates which synthesizer patch will be used for notes on this particular MIDI channel. In the example this would be patch #24. The *pitch_offset* parameter is a note offset which will be added to pitch. Negative offsets are possible. In the example this would be 64. If you don't want an offset added, enter `0' into that field. All parameters must be provided.

Also, you can specify the maximum number of voices to be used. To do so, just enter the line

```
n = x
```

(where x is the number of channels) into your PARSE.CNF file. For example:

```
n = 5
```

will specify that no more than 5 voices will be used by the synthesizer when the score is played.

The `NOTES.CNF' file contains the frequencies cooresponding to each note. The format is very simple. To change the frequencies (which is probably not necessary in most cases), just change the file with a text editor.

Location of Configuration Files

The PARSE program looks in the following locations for the PARSE.CNF file, in this order.

1) Current directory
2) Directories specified by PATH environment variable.

Older versions of PARSE viewed the PARSE.CNF file as optional, but the current version requires that it be present. A default PARSE.CNF is provided in the JAGUAR\BIN directory, which should be included in your PATH if your system is set up correctly. You will normally create project-specific versions of PARSE.CNF in your project directories.

The PARSE program looks only in the current directory for the NOTES.CNF file. This file is optional.

MERGE Utility

The MERGE utility is designed to take music data files created with PARSE and merge them together into a single file that will contain all the music data interleaved together appropriately.

```
merge outputfile input1 input2 [input3...]
```

outputfile Filename for the desired output file. The combined contents of the input files will be output to this file.

input1, etc... Filenames for files to be merged. You can have up to 32 separate input files merged together at once (possibly less depending on your system configuration).

SNDCOMP Utility

The SNDCOMP utility is used to compress digital sound samples. It is designed to take a 16-bit digitized sound file and compress it to 50% of its original size. The compression it does is a "lossy" compression, but the quality is quite good. The compressed sound files it creates can then be used with the Jaguar Synthesizer.

```
SNDCOMP inputfile
```

inputfile Filename of the source file containing the original 16-bit digitized sound data.

The output file created has the same filename as the input file, except with a .CMP extension.

CBPEG Utility

The CBPEG tool takes a Targa-format[2] or GIF-format picture file and converts it into the Jaguar BPEG format, a variation of the JPEG[3] lossy compression standard for graphics images. Pictures compressed into the BPEG format show little or no visible reduction in image quality, but typically take between 1/10th and 1/50th as much space as the original.

Usage

```
CBPEG [options] inputfile
```

The command line options may be used in any order, but only one input file may be specified, or else an error is generated.

Option	Description
-maxmemory *n*	Sets the maximum amount of memory to use, where *n* is the amount in kilobytes (i.e. **-maxmemory 512** would specify that CBPEG can use up to 512k of memory)
-qtables *file*	Specify that CBPEG should use the quantization tables specified in *file* for the image compression. This option should only be used by those people who consider themselves experts regarding JPEG.
-quality *qual*	Sets the JPEG compression quality/compression ratio percentage. The *qual* value must be between 2-100. For most purposes, a value between 60 and 80 will provide the best balance between compression ratio and image quality. Higher numbers produce output with better image quality but don't compress the image as much. Lower numbers provide better compression ratios at the cost of image quality, but if the number is too low you will get a visable degradation in visual quality (this will appear as fuzzyness and/or blockiness). The goal is to find a number that gives you acceptable compression and a picture that is visually close to indistinguishable from the original image. This "ideal" setting is different for different pictures, so it's a matter of trial and error. The default setting is 75, which is usually a good starting point. If you go much above 75 you lose more and more compression without a significant gain in quality.
smooth *n*	Sets the smoothness for dithering the input file, where *n* is the amount from 1 to 100.
-targa	Specifies that the input file is a Targa-format picture file. This is usually not required, as CBPEG can usually detect this automatically. Use this option if the file is not properly recognized.
-verbose or **-debug**	Specify that verbose output/debugging information should be displayed throughout the conversion process.

[2] Targa is a popular image file format for 16-bit and 24-bit RGB true color graphics. If your graphics programs do not support the Targa file format, then you should investigate one of the various file format conversion utilities. HiJack Pro for Windows is available at computer stores everywhere, and the shareware program Paint Shop Pro (for MS-Windows) is available online.

[3] JPEG stands for "Joint Photographic Experts Group". This is a "lossy" image compression scheme that is capable of extremely good compression ratios with little visible loss of image quality. Additionally, the image quality/compression ratio tradeoff is user-selectable so you can fine tune the compression for different images.

Example:

```
cbpeg -quality 60 cat.tga    (convert CAT.TGA to CAT.BPG with quality of 60)
```

The *inputfile* parameter is the filename of a Targa or GIF format picture file. The CBPEG tool will always create an output file similar to the input filename, except with an extension of ".BPG".

Picture Depth Considerations

The BPEG image format is 24-bits per pixel. When you compress a picture that uses less than 24-bits per pixel with CBPEG, it is expanded to 24-bit prior to compression. The BPEG decompression routines that run on the Jaguar GPU decompresses into either 16-bit or 24-bit per pixel bitmaps, depending on your BPEG decompression options.

Most Targa picture files are either 16-bits or 24-bits per pixel, and are ideal candidates for BPEG compression. However, note that GIF format pictures can only use up to 8-bits per pixel (256 colors), and some use only 4 bits per pixel (16 colors). These images are converted to 24-bit before being compressed, but when the images are later decompressed on the Jaguar, you still get bitmaps with either 16 or 24 bits per pixel. Despite the ROM storage space savings realized by using BPEG, you end up using two or three times as much RAM for the bitmap at runtime (assuming an 8-bit picture). If you need to compress an 8-bits per pixel (or less) picture and end up with the same format when it is decompressed, the BPEG format is not your best choice. You may wish to investigate the LZSS compression library instead. See the **Libraries** chapter for more information.

[MF1]