TransADF

COLLABORATORS

	TITLE : TransADF		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TransADF

1.1 TransADF v4.0.46

TransADF v4.0.46
TransADF-RT v4.0.46
TransADF-Lite v4.0.46

5th October 1998

Contact Info
TransADF is FreeWare as defined by the
GNU General Public License

Table of Contents

```
                 == You should read this, in case you missed something.

                 My HomePage
                 == Getting better :).


                 The Full Source
                 == More insight into the progam than you'll even want.
 Aminet readme file    == Guess what this is!
```

1.2 Contact Info

```
                  I can be contacted via the following e-mail address:

   kjots@lis.net.au

  Also, check out
                My HomePage
                .  It has links to all my software
  as well as some other stuff.


                 Back to Contents
```

1.3 The Legal Bit

```
                  Ok, before I go into the standard disclaimer bit, I just want to
  make two things very clear:

  1) TransADF is nothing more or less than a copy program, and whether
  it be a disk-to-disk or file-to-file copier, or as in this case,
  disk-to-file and back, the same rules apply.  IT IS ILLEGAL TO COPY
  AND DISTRIBUTE COPYRIGHTED MATERIAL.  Most companies will allow you
  to backup a disk for private use, but you can't then give that
  copy to someone else.  TransADF is no different.  Any ADF you
  create is under the same copyright protection as the disk it came
  from.

  2) Just because I wrote something that can be used with UAE or
  Fellow does not mean I'm going to give out Kickstart or Workbench
  images.  Don't mail me asking for them, I will won't even bother
  to reply.

  OK, now that that ugliness is over with, on to the next bit of
  ungliness.

Standard Disclaimer:
--------------------

  Copyright (C) 1998 Karl J. Ots
```

The author is trying to make this program as stable and usable as possible, so send me reports of any problems you have. I am open to suggestions on how to make this a better program. Send all reports and suggestions to my
email address
.

Back to Contents

1.4 Acknowlegments

I would like to acknowlage the following people:

The authors of the ZLib, Jean-loup Gailly and Mark Adler.
Visit http://www.cdrom.com/pub/infozip/zlib/ for more information about the ZLib.

The author of the DICE C compiler, Matt Dillon.
Visit http://www.obviously.com/ for more info about DICE.

Also, the follwing people whe gave me feedback:

"Treveur."	--	General feedback
Thomas Lorenz	--	Verify option
Mauro Sergio Lourenço	--	RAD: and FMS: support
Kovacs "ARCHON" Gabor	--	OS1.3 support
DoMiNATorR	--	Extended Track support
Chris Creevy	--	OS1.3 support
Henrik Nilsson	--	Sweedish translation of documents
"No Body"	--	Many ideas, including verify and compare
Popal	--	General feedback
Steven	--	OS1.3 support

Not all of these ideas have been implelented in this release, but most are on the cards for the next one.

Back to Contents

1.5 About TransADF

What is TransADF?

TransADF is a utility that can transfer Amiga Disk Files (ADFs) to and from Amiga disk drives, as well as any other disk-kile device (eg RAD:, FMS: etc)

An ADF is typically an unmodified 880kb file image of a disk drive. It can be used to transfer complete Amiga disks over a communication medium that doesn't take plastic (eg modems, the Internet). At the other end, a utility like TransADF can take the file and put it back on the disk. Simple.

ADFs are also used by the Amiga emulators UAE (Unix Amiga Emulator) and Fellow. The disk drives of a PC can't read Amiga disks (FACT!), so the emulators use the ADF as a virtual floppy disk drive.

Another virtual floppy drive, this time on the Amiga itself, is FMS. This is a device that allows a file to be mounted so that it looks like a floppy drive to the Amiga.

Enter TransADF

TransADF can create ADFs for all these purposes, as well as an added bonus of being able to handle ZLib, GZip and PKZip archives. You probably know what PKZip files are, they are are the standard archiving format on the PC systems. GZips are used mainly on UNIX systems and can only compress one file. ZLibs are defined in RFC-1950, and the format can be created by the ZLib without any interaction from the rest of the program, so it was simple to add it.

The extra extra bonus is the ability to add and remove a file from a PKZip archive by name. Any file within the archive can be removed, and any number of files can be added to an existing archive, both by TransADF and any other program that can Zip files.

TransADf comes in three 'flavours'. The standard version is simply TransADF, and was linked with the ZLib at compile time. It's quite large, about 46k. TransADF-RT uses the runtime "z.library", available on Aminet in util/libs/zlib.lha. It reduces the size of the executable to about 19k, but seems to degrade compression slightly. Ah well, nothing's perfect. The third version is TransADF-Lite. It includes no compression or decompression routines, nor GZip or PKZip support. The removal of these features reduce the size even furthur to 8k. All flavours have been compiled from the same sources .

The best way to learn how to use TrandADF is to read the examples and usage sections:

Command Line Options

Examples

Back to Contents

1.6 Command Line Arguments

Command Line Arguments

DRIVE The AmigaDOS drive on which to operate, either for input or output. This can be any drive which DOS recognises as a device eg DFx:, FMS:, RAD: etc. Note that drive must have a fixed geometry; devices like RAM: which grow and shrink cannot be used.
This argument is required.

FILE The name of the file to operate on. This file is created (or updated, see ADD) if you are reading from the drive. If you are writing to the drive (with WRITE below), then TransADF will auto-detect the file type.
This argument is required.

START These indicate the cylinder numbers to start and end the
END operation. The default values are 0 for START, and the highest cylinder number for the drive for END (79 for DFx:). Any value between and including these defaults may be used, with the exception that START may not be larger than END (but they can be equal, in which case only one cylinder is transfered).
WARNING: It is legal for END to be larger then the highest cylinder number for the drive. This should be avoided unless you know what you are doing.

WRITE By default, TransADF copies data from the drive into the file. This switch reverses that operation, causing it to write from the file onto the disk. This will cause all the data currently on the disk, within the ranges indicated by START and END, to be overwritten.
Starting with version 4, TransADF will not, by default, format the disk as it writed the data. This was changed to be more compatible with non-TrackDisk devices. See FORMAT

below.

VEFIRY With this option specified, TransADF will read data from the drive after each track has been written to make sure it has been written correctly.
This option will only work if WRITE has also been specified and will slow down the transfer by about 40%.

FORMAT As stated above, TransADF no longer formats while it writes. With this option, you can turn that feature back on, which means you can still transfer data to disks that are either unformatted, or formatted to a different geometry (eg PC disks). This is something that
TransDisk
is incapable of.
WARNING: This option should only be used with drives that understand formatting (eg DFx:), or those that don't need to be formatted (eg FMS:, RAD:). It may fail with other 'real' disk drives like CrossDOS or DiskSpare.
This option will only work if WRITE has also been specified and will slow down the transfer by about 20%.

The following option are not acvitve in the Lite version, and will be ignored. With the excpetion of NAME, none of these have any effect if WRITE is specified.

ZLIB Create a ZLib stream as defined by RFC-1950. This option overrides the GZIP and PKZIP option.

GZIP Create a GZip file as defined by RFC-1952, and is compatible with GZip version 1.2.4. This option overrides the PKZIP option.

PKZIP Create a PKZip file that is capatible with PKZip, Info-Zip and WinZip etc. The default action is to create a new archive, but you can add to an existing archive by specifing ADD.

NAME The name to store in PKZip and GZip files when reading from the drive, or the name of the file within the PKZip file to transfer when writing to the drive. The latter case uses Amiga wildcards to search for filenames. The first match is used, and no matches produces and error.

LEVEL Compression level, a number between 1 and 9, with 1 being the fastest and 9 resulting in the best compression and a smaller file. 0 can be used but is not recomented and it will create a file that is larger than a 'vanilla' ADF.

ADD Causes the file to be added to a PKZip archive, rather than creating a new file. This option only works with PKZIP specified. Attempting to add to a non-existent zip file will result in an error.
WARNING: The PKZip file is not backed up in any way; TransADF attempts to work in low-memory, low-capacity environments. So if the computer crashes during the operation, you will most likely lose the orginal PKZip file. So make sure you back up the file before you ADD to it.

Back to Contents

1.7 Some Examples

```
                Some Examples
-------------

  The following are some examples of how to use TransADF


  1> TransADF DRIVE DF0: FILE T:wb.adf

  Creates a new file "T:wb.adf" and transfers the entire
  contents of the disk in DF0: to it.


  1> TransADF DRIVE DF0: FILE T:wb.adf START 40 END 79

  As above, but only transfers the last 40 tracks to the
  file.


  1> TransADF DRIVE DF1: FILE T:wb.adf WRITE

  Writes the contents of "T:wb.adf" to the disk in write DF1:


  1> TransADF DRIVE DF0: FILE T:wb.gz GZIP NAME wb.adf

  Creates a new file called "T:wb.gz" and compressed the disk
  in DF0: into a GZip format file.  The NAME "wb.adf" is stored
  in the GZip file.  GZIP can be replaced by PKZIP to create
  PKZip files, or ZLIB to create ZLib format files.


  1> TransADF DRIVE DF0: FILE T:wb.zip PKZIP ADD NAME ex.adf

  Adds the disk in drive DF0: to the already existing PKZip file
  T:wb.zip, with the name "ex.adf".

  1> TransADF DRIVE DF1: FILE T:wb.zip WRITE NAME ex.adf

  Extract the file within the PKZip archive "T:wb.zip" named "ex.zip"
  to the disk in drive DF0:

  Transfer an ADF to disk with verification.

  1> TransADF DRIVE DF1: FILE T:wb.adf WRITE VERIFY

  Transfer an ADF to the RAD: disk.

  1> TransADF DRIVE RAD: FILE T:wd.adf WRITE
```

Back to Contents

1.8 History

The History of TransADF

Background

TransADF was inspired (well, I suppose that's the word you could use...) by TransDisk that comes with UAE (The UNIX Amiga Emulator). TransDisk is usably, but not very nice. For one thing, it writes the ADF data to the StdOut (ie the shell), expecting you to redirect it into a file. The only problem was that it wrote progress info to the StdErr, and the shell I use (ZShell) can't distinguish between the two. So I end up with a file that has "Reading track blablabla" between eack track. Another thing that TransDisk did, or rather didn't do, was Inhibit access to the drive that you are operating on. This means that you could attempt to write to a disk that contained the file you were reading. What a mess! Anyway, I decided write a better one, one that did everything right.

Pre-release versions

The following versions of TransADF were never relased. Each was completed, but when it came time to start writing the docs (something I despise!!!), I had a better idea how to do something. So these got left behind.

TransADF v1.42 (September 1997)

This was my first attempt at TransADF, and was, as stated above, completed. It didn't have any compression or decompression abilities. Apart from some re-arranging to incorperate the newer functions in future versions, the TrackDisk reading and writing routines havn't changed. This version is basically the same as TransADF-Lite.

TransADF v2.10 (November 1997)

This was my first attempt to incorperate decompression, and I did it by adding the module "inflate.c" from the GZip 1.2.4 distribution, unmodified, along with a "taylor.h" and "gzip.h" modified to enable inflate() to compile. A "gzip.c" module that allowed it to operate was added, and guess what? It worked! So I was really pissed off when I found out about the ZLib and all that hard work was wasted (This is the best argument for a permanant home connectoin to the Internet!). Still, I learned a lot, so I suppose it wasn't a complete loss.

Released versions

TransADF v3.72 (8th December 1997)

Well, I think I can finally release this version without needing to start all over again :) That's not to say it's finished, there's still tonnes of things I could add (see the TODO section), but it's as stable and usable as it needs to be.
Anyway, as a summery of new features, this version uses the Zlib to do the decompression, and also gives it the ability to compress files, which 2.10 couldn't. It comes in three flavoures, standard, RT (RunTime) and Lite. More info can be found elsewhere in this document.

TransADF v3.100.27 (13th July 1998)

This is primarily a cleanup version, in prerparation for version 4 which should be out later this year. It includes verify and extended track support, which were easy enough to insert without a major re-write.

The Current Version

TransADF v4.0.46 (5th October 1998)

Over a year since I started, TransADF is now more feature packed then ever. The big change with this version is the inclusion device-independent I/O routines, makeing TransADF usable on RAD:, FMS: and other disk-like devices. Apart from a bit of cleaning of the code, nothing mush else has changed.

Back to Contents

1.9 Things TODO

The future of TransADF

I'm planning OS3.1 support for v4.1, but it probably won't be ready this year. I'm also still planning an 'Extended' ADF format that stores rlaeded info about the disk along with the actual data.

```
  Of course, future version will also include fixes for any bugs.
  Let me know if you find one.
```

Back to Contents

1.10 Notes

```
                Notes
-----
  The current time and date is stored in both the GZip and PKZip
  archives.  The time recorded is the time just before beginning
  of compression, as recommended in the GZip documentation.

  TransADF-RT offers slightly less compression than TransADF with
  the same options.  I assume this is because of some compile-time
  options of the z.library.

  All versions of TransADF are ’pure’ and can be made resident.
```

Back to Contents

1.11 My Homepage

```
                My World Wide Web page
----------------------

  My new web-page is at:

    http://www.lis.net.au/~kjots

  And by the time you read this, it should be up and running (fingers
  crossed).
```

Back to Contents

1.12 The Full Source

```
                  The Full Source
  ---------------

  This was written with DICE C.
```

```
     DMakeFile

     main.c
     args.c
     mem_chunks.c
     tdevice.c
     read_disk.c
     write_disk.c
     defl_disk.c
     infl_disk.c
     util.c
    &{" util-asm.a    " LINK Source/util-asm.a/MAIN}
     gzip.c
     pkzip.c
     errors.c
     version.c
     dos_fprintf.c


     main.h
     args.h
     mem_chunks.h
     device.h
     read_disk.h
     write_disk.h
     defl_disk.h
     infl_disk.h
     util.h
     gzip.h
     pkzip.h
     errors.h
     version.h


  Missing Files
  -------------

  If you want to recompile the sources, you'll need the following:

  z.lib, zlib.h, zconf.h
    z.lib was compiled from the sources, available at the
    ZLib homepage http://www.cdrom.com/pub/infozip/zlib/
    There is a DMakeFile in the directory ZLib/ of this
    distribution that you can use to make the required
    model of the library (zsr.lib).
    The two header files are included with the ZLib
    source.

  z.library, z_pragmas.h
    These two are necessary if you want to compile the
    RT version of TransADF.  They are available on Aminet
    in util/libs/zlib.lha.  Note that the two header files,
    zlib.h and zconf.h, are included with these with two files,
    and are identical to the ones with the ZLib source.

  The three header files should be placed into the directory Source/.
```

```
You require no extra files if you want to compile the Lite version
of TransADF.
```

Back to Contents