TransADF

COLLABORATORS

| | *TITLE* :

TransADF | |
|---|---|---|
| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
| WRITTEN BY | | June 25, 2022 | |

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

TransADF

1.1 TransADF v3.100.27

```
                 TransADF v3.100.27
  TransADF-RT v3.100.27
  TransADF-Lite v3.100.27
  Copyright © Karl J. Ots
  13th July 1998


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

Table of Contents


          Disclaimer
          == It's important that you read this.

          Acknowlegments
          == The people who made it possible (other than me)


          About TransADF
          == What is it and how do I use it?

          Command Line Options
          == Making TransADF do what you want it to do.

          Examples
          == Some examples (No kidding?).


          History of TransADF
          == The past and the future.

          What TODO next...
          == Some things that might happen.  Maybe.

          Miscellaneous Notes
```

                    == 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

You can read the GNU General Public License by following the
preceding link, or by viewing the file COPYING included in this
distribution.

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
DoMiNAToR              -- 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.

  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 Options

                Command Line Options

 DRIVE The disk drive to operate on, either reading from it
    or writing to it. Only standard TrackDisk drives are
    supported at this time (see the
                TODO
                 section for future
    plans in this area), so DRIVE must be of the format
    DFx:, where x is a digit between '0' and '3'. Case in
    unimportant, and the colon (:) must be included.
    This argument is required.

 FILE The file to operate on, be it a standard Amiga Disk File,
    or a ZLib, GZip or PKZip file. The file is created (or
    updated, see ADD) depending on the compression switches.
    If you are writing to the drive (with WRITE), then
    TransADF autodetects the type of file by the first few
    bytes (eg, Amiga disks start with the characters DOS).
    This argument is required.

 START These indicate the tracks to start and end the operation.
 END The default values are 0 for START and 79 for END, and any
    value between and including these defaults may be used, as
    long as START is not larger than END. They may be the
    same value, in which case a single track is copied.
    The upper limit of END track has been removed, so you can
    write extended track ADFs to floppys now. You will get
    a warning if END is greater than 79.

 WRITE Causes TransADF to write the FILE to the DRIVE. The
    default operation is to read the DRIVE and create a new
    file (except for PKZIP with ADD), but with this option
    set the contents of DRIVE are replaced with those on the
    file. This will destroy the current data on the disk, get
    it? It's just like formatting. In fact, it is formatting.

TransADF uses the TD_FORMAT command to write new data to
the disk. This means that unformatted disks (eg PC disks)
can be used directly without prior formatting. This was
something that
              TransDisk
               was incapable of.

  VERIFY With this option specified, TransADF will read from the disk
    after each track is written to make sure the data was written
    correctly. It only works in conjunction with the WRITE option.
    Verifying will slow down the transfer by about 40%.

The following options are not available in the Lite version.
Except for NAME, none of these have any effect if WRITE is selected.

  ZLIB Create a ZLib file, as defined by RFC-1950. This
    option overides the GZIP and PKZIP options.

  GZIP Create GZip file, as defined by RFC-1952, that is
    compatible with GZip 1.2.4. This option overides
    the PKZIP option.

  PKZIP Create a PKZip file that is compatible with PKZip
    and Info-Zip. Default behaivour is to create a new
    file but this can be changed wih ADD. This option
    should be used in conjunction with NAME, otherwise
    the file is stored with the name "disk.adf", even if
    you are ADDing.

  NAME The name to store within GZip and PKZip archives when
    reading a disk, or the name of the file to read from
    within a PKZip archive when writing to a disk. The
    latter variety supports standard Amiga wildcards, and the
    first match is used. No matches result in an error.

  LEVEL The compression level to use, a number between 1 and 9,
    with 1 being the fastest and 9 resulting in the best
    compression. The default is 6.

  ADD Causes the FILE to be added to a PKZIP archive with NAME.
    This option is ignored if the PKZIP option is not selected.
    The Zip file is not backed up by TransADF during the
    operation, so if something goes wrong, the file will be
    lost. You should back up a zip file every time you want
    to add to it.
    Attempting to ADD to a non-existent file will result in
    an error. This problem will be addressed in a future
    version of TransADF (see
              TODO
).

              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



                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.
```

```
The Current Version
-------------------
```

```
   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.
```

```
                 Back to Contents
```

1.9 Things TODO

```
                 The future of TransADF
   ----------------------
```

```
  Well, I've got a few ideas.  I would like to make TransADF
  create ADFs for non-TrackDisk disks, such as PC or DiskSpare
  devices.  This will create a problem, where the user will have
  to supply a whole lot of details such as the number of sectors
  per track and the highest and lowest tracks of the disk, which
  would make the progam less usable by people who don't know those
  sort of things.
```

```
  One solution would be the creation of an 'Extended' ADF format.
  Such a format would include embeded information about the disk
  geometry, as well as where the operation started and stopped.  I
  suppose it could also include an option to compress the data before
  adding it, either as a whole block or one track at a time in the
  style of DMS.  This would eliminate the need for external
  compression formats like GZip and PKZip.
```

```
  Version 4 is going to be a major change.  I plan to completely
  alter the way it works internally.  Among the changes will be
       support for OS1.3, and maybe multiple device support, if I can
  work out the details.
```

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

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.
```

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).
```

1.12   The Full Source

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

  This was written with DICE C.

     DMakeFile
```

```
    main.c
    args.c
    mem_chunks.c
    td.c
    read_disk.c
    write_disk.c
    defl_disk.c
    infl_disk.c
    util.c
    gzip.c
    pkzip.c
    errors.c
    version.c
    dos_fprintf.c


    main.h
    args.h
    mem_chunks.h
    td.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