$VER:

COLLABORATORS

	TITLE : $VER:		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

$VER:

1.1 DebuggerTool DocFile

```
            DebuggerTool description file
      by TRZIL Bernhard  ©1993


            Read first

            Tool types

            Libfile description

            Disassemble window

            Memory window

            Register window

            Floitingpoint window

            Startup-Prefs

            Level6 Interrupt

            ARexx commands

            SCREEN

            General Keys

            History
```

1.2 read first

```
            READ FIRST!
```

FOR DEVPAC USER:

Don't assemble with DebugLine-Information, crashs will be the affect.
Deplode the DT file and rename it to MonAm. (Use Imploder or Powerpacker)

GENERAL USE:

Use

                STEP
                 and
                NEXT
                 to debug, because

                TRACE
                 is a stackoperation command. Use it only in ROM.
If a program goes to wait state, it could be possible that the program
wants a input at the std-input window. Some programs use the ReadArg
function of dos. (Like list)
So type in a argumentstring or just enter.

1.3 tooltypes

                TOOLTYPES:

You can set following tooltypes with the workbench:

PUBSCREEN=xxxx   This open the windows on the PublicScreen xxx
PREFS=xxxxx   Use special prefsfile normaly set to "ENVARC:DT/DT.Prefs"
       please remeber when you save the prefs!
LIBFILE=xxxx   Liboffset filename. Standard = "ENVARC:DT/DT.LibFile",
       Look at
                LIBFILE
                .
STARTUPPREFS=xxxx   Prefsfile for starting setups.
WORKBENCH   Use workbench for windows
COLOR   Make a colorflash at start.(Only for visual testing)
ICONIFY   Iconify the DT at start

If a keyword is wrong the screen will be flashing at start-> Please
check the info of the icon !

1.4 libfile

                LIBFILE DESCRIPTION:

The libfile for the automatic offsetname creation is normaly placed in
"ENV:DT/DT.LibFile". You can set the filename with a
                tooltype
                File format:

The first two numbers are the filelength in kbyte.
```
 |
 V
```
```
16*test.library  A star is the sign for a new Libraryname
FirstFunction    Begin with offset -30,-36,-42,...(FirstFunction=-30)
---NextFunction  - jumps over one function (if not implemented or
     private)
LastFunction     LastFuntion=-60
** ; Ende        If the first character of the libraryname is a * then is
     the end of the file reached.
                 Everything afer this two stars is not important
```

1.5 disassemble

```
                 DISASSEMBLE WINDOW:


This window is the quick disassemble of the current window
                 StartAdress
                 .
Shows labels if present and the current
                 PC
                  (>) and the Breakpoint (*).
Use Cursors UP/DOWN to scroll and LEFT/RIGHT to scroll wordwise.


Available keys:
A      Set
                 StartAdress
                 .
Z      Make
                 STEP
                 T
                 TRACE
                 S
                 BACK
                 R
                 RUN
                 U     Run
                 UNTIL
                 D     Run
                 UNTIL
                 J     Jump over one instruction without execute
M      Modify Register
L
                 Load
                  an executeable file. Please use fullpath and
       name if you simulate a workbench start!
F      Find a instruction
W      Find next in the last range, selected with find instruction
Y      Open foatingpoint window
P      Get
                 Startup
                  window
C      Calculate
                 Expression
```

```
                    the output will be in
        positiv HEX, negativ HEX, DECIMAL and BINARY
B      Set an Breakpoint
N      Jumps to
                 NEXT
                  intruction
E      Execute dos command
O      Set offset
  Displays disassembleadress sub. with the offset.
  Use offset zero to remove offset.
  If offset is activ, a point after the memoryadress will show you
  that the offset is active.
K      Kill old and open new tracetask. Use this to go back to run state if a
       program is in wait state.
G      Create a new memory window.
Q      Quit debugger
I      Iconify DT

Gadgets:
WA     Set
                 StartAdress
                 T
                 TRACE
                 S
                 BACK
                 R
                 RUN
                 L
                 Load
                 BS      Set SBreakpoint
BC     Clear Breakpoint
NW     New disassemble window
SP     Save windows size and positions
SM     Change screenmode
```

1.6 memory

```
                 MEMORY WINDOW:

Hexa- and ASCII outputwindow.

Keys available:
A      Set window
                 StartAdress
                 N       New memory window.
C      Calculate string
E      Go to editmode
       In the editmode the ESCAPE key is used to exit.
L      Set
                 Lock
                  expression
F      Find requester
  Examples:   'Hello',0.b,$34.l,???,"Shit"
        This searchs for the hex:
```

```
          $48656c6c6f0000000034xxxxxx53686974
          <   hello >0.b<$34.l>Joker <Shit>
```
  If the endadress string is empty, the startadress will be
  displayed.
W   Find next
I   Absolut binary load
  If the length is negative-> full file will be loaded.
S   Absolut save binary
  If length is negative length of the last loaded binary file will
  be used.
ESC   If not in editmode ESC close the memory window.
O   Set offset
  Displays memoryadress sub. with the offset.
  Use offset zero to remove offset.
  If offset is activ, a point after the memoryadress will show you
  that the offset is active. Otherwise a line is display.
V   Copy memory.

Keys on numpad:

[Read disksectors. (512bytes = 1 Sector)
]   Write disksectors.
/   Calculate bootblockchecksume.

Gadgets:
Lock Set
              Lock
              WA   Set window
              StartAdress
              ED   Toggles EDIT-Mode (ESC to exit)
New   New memory window (Key N)

1.7 register

REGISTER WINDOW:

Show you the current Data and Adressregisters, PC and the
Status-Register.

Keys:
M   Change mregister
K   Reset registers

Gadgets:
C   Reset registers
R   Change register

1.8 floitingpoint

FLOITINGPOINT WINDOW:

This window show you the values of the FP-Registers. Only available on
FPU-Machines.

1.9 startup

STARTUP-PREFS:

The window allows you to set following parameters:

With the cycle-gadget you can choose the start emulation from workbench
or CLI.

```
Current-Dir      Actual directory for the debugfile
Std-Output       Output specification
Arguments        Cli arguments to set for the executeable file
    (only for CLI)
Start exec. This command line will be executed at start of the program.
    If the string starts with (the command will not be executed)
```

The LISTVIEW is for the workbench tooltypes simulation.
If you start a program from Devpac, then you will need a standard tool
icon in the ENVARC:Sys drawer called "def_tool.info".
Please use the full path in the load requester if you simulate a
workbench start!

1.10 startadress

STARTADRESS:

Key: A
Ask you for a
                Expression
                 and the display will be
refreshed. This function does'nt reset the
                Lock
                .

1.11 load

LOAD:

Key: L
Load an file and search for symboltable. Unload the last loaded file if
exists. Type full path and name if you simulate a workbench start.

1.12 run

RUN:

Key: R
Run the program started with the current PrgCounter.
Exceptions like ADRESS or ILLEGAL-Errors will be hunted and displayed
(the prg will be stoped).

1.13 next

NEXT:

Sets breakpoint over next instruction and run the program. Current Breakpoint
will be cleared.

1.14 step

                STEP:

Key: Z
Trace is a single step with the TraceMode of the 680xx. You can trace in
the ROM too. Be careful and use Breakpoints,
                NEXT
                or TRACE to debug.

1.15 trace

                TRACE:

Key: T
Normaly goes over the next instruction, not into subroutines. Warning:
This is a stack operating instruction. If the
                PC
                 is
pointing to a MOVEM xx,-(sp) or LINK, then this function will stop at the
equivalent MOVEM (sp)+,xx , UNLINK.

1.16 back

BACK:

Key: S

Goes one stackdepth back. Use it to go out from a subrountine.

1.17 until

```
UNTIL:

Key: U

Run until a memoryadress is changing or reached a constant.

Syntax examples:
 m$46.l           Stops if the longword of the address $46 is changing
 mDataBase+4*10   Stops if the word of DataBase+4*10 is changing
 sCount.b         You will be ask for the stop value. For example 46
                  If the byte on the address Count is 46, the program will
      be halted.
```

1.18 until register

```
UNTIL REGISTER:

Key: D

Runs until a register has an adress reached or has been modified.

Syntay examples:
 md0              Stops if the lower word of d0 has changed
 sa3.l            You will be ask for a stop value, for example $78ac30
                  The program will be halted if the Adressregister 3 has
        the value $78ac30

Praxis example:
 You want to know were the program open the dos.library.
 Search for the string 'dos.library' in the memory-window. The OpenLibrary
 function needs in a1 the pointer to the name of the library, so
 Run UNTIL Register
 dosstring reached. The program will stop after the instruction
 LEA $xxxxx,a1 or MOVE.L #$xxxx,a1,...
```

1.19 level6

```
LEVEL6 INTERRUPT:

Why not the Level7 ?
Simple reason: The A300,A4000,A1200 has no level7 interrupt.

Hardware:

You can stop the running program at any point exept in the ROM or when
the Task is in the waiting state. Maybe use ARTM-Tool to set to
RUNNING-State!

Hardware A500,A2000,A3000,A4000:
 Install a button between Pin22 and Ground (Pin1-4) on the expansion
```

 port.

Hardware A1200:
 Connect 138 (EINT) with 139(GND).
  (Not tested)

If you press and the screen turns red-> The task is in the WAITING-Sate
If the screen will be green-> Task runs in the rom.

1.20 lock

                  LOCK:

Key: L

Locks the windowrefresh after
                  TRACE

                  STEP

                  BACK

                  RUN

                  NEXT
                   to the lockexpression. To clear the lock
type in a empty string. The lock can be any
                  Expression
                  .

1.21 pc

PC:

Program counter is displayed in the RegisterWindow and points
to the next instruction that will be executed.

1.22 expression

                  EXPRESSION:

An expression can be a complex calculationstring.You can use brackets
to discribe a order.
                  Labels
                   are strings starting with a
ALPHA char. Use '$' for hexadezimal and '%' for binary strings. You don't
have to kill the expresion. With {}.x you can read the adress.
(For pointer operations!) (x for size)

```
Examples:
{4}.x+29   : Show you the ExecBase+297. (Processorinfos)
       .x statds for the size (w=word, l=long, u=unsigned word)

You typed in $c56a67. Now you want to set to the label Start
Add this   : Start$c56a67
OK. Simple : If the place were the next mathematik operand should be, a
       Char or '$','%',';' or ',' is, will the calcstring be cuted.
Think about... Its easy and useful.

If you click into a disasseble or memory window wihle the expression window
is opened then the line- or value-adress of the memory will be replaced
with the current editstring if the cursor is in the first charactar, else
the value will be added at the cursor.
```

1.23 labels

```
              LABELS:

Reserved:
a0-a7, d0-d7, fpiar (FPU only), PC (current
              PrgCounter
              )
seg0,seg1,...  is used for the segment startadresses. Useful for save and
search operations.

The debuger get the labels from the filesymboltable or directly from
DEVPAC. This DT is special used for DEVPAC. You can replace the MONAM with
this DT. Just rename the original and replace it with DT and rename it
to "MonAm".

Show at
              Read first
```

1.24 rexx

```
REXX commands:

The rexxport is called "DT.x". (X is the number of the DT-Task.)

General error codes (RC of arexx):
  1 = Wrong bracket
  2 = Bracket missing
  3 = " missing
  4 = Wrong size with {}.x
  5 = Division error

  6 = RexxSyntax error

(1-5 = Error in a expression string)
```

```
  7,8 = Special cases (read command instructions)
   9 = Cancel selected
  10 = Diskop. failed

 27 = Argument missing or to much arguments
 20 = Hard error (No such command)
```

Following commands are supported:

```
QUIT              Quit the Debugger

MEM adr [x y w h] New mem window

DSM adr [x y w h] New dsm window. Width will be override.

LOAD Name         Load a program
     RC =7 if program is running.

UNLOAD            Free all mem for loaded program.
     RC =7 if program dont quit

GETOFFSETNAME     Returns in result the last offset- and the libraryname
     of a jsr.
                  Example:   "exec.library/OpenLibrary"
                  Look at arexx demo "GetDTOffSet.rexx" it search the
     function description in the include: path and displayed
     it with the Amigaguide or CED.

DTTOFRONT   DT-Screen to front.

DTTOBACK    DT-Screen to the background.

CLEARWINDOW ID NR
     Close a window
     ID : ID_number of window
     NR : Witch window to close.
          If NR is zero all windows with given ID will be closed.

     ID are:
       0  = Main Debugwindow (Forbidden!)
                 1  = Memory window
       2  = Regwindow (Be carefully!)
       3  = Expressionstring window (Be carefully!)
       4  = Debugwindow
       5  = FPWindow
                 6  = Infowindow (from calc expression or INFO-ARexx)
       7  = Prefswindow
       8  = Sure requester
       9  = Labellist window
SETADDRESS ID NR EX
     Set windowstartaddress.
     The window with number NR and the specified ID will be set to
     the expression EX.

LOCKMEM NR [EX]   Lock a memory window. If no EX is given then unlock.
     If NR is zero all memorywindows will be effect.
```

GETADDRESS ID NR Get a startadress. If NR is zero the first window with
        the right ID will be return.
        The adress will be returned as hex. Please use x2d(EX)
        to convert it to a calcable decimal number.

CALC EX   Returns the result of the expression.
        RESULT is hexadezimal, use x2d(EX) to convert.

REFRESH   Refreshs all windows.

DISASSEMBLE   Disassemble one line. Use DISADR to set disassembleadress.

SETDISADR EX   Set disassemble adress for the next arexx disassemble
        command.

GETDISADR   Get the actual disassemble adress from the arexx dis-
        assemble command. After disassemble, you can get the
        adress of the next instruction.
        RESULT is hex.

INFO EX   Open info window with the EX in hex,dec and binary.

CALC STR   Ask after a string with the windowtitle STR.
        Full multiwindow operations are allowed.
        Warning: The first char in the STR is the stringpuffer
        number.
        Examples:
        GETEX '0Which Adress';
        GETEX '!Test';
        Allowed buffers are: 0-9,: ; < = > ? @
        Please prefer ?, @ or 0 for arexx.
        0 is used for calcstring and all startadresses.

FILEREQ STR   Open a filerequester with the windowtitle STR.
        RESULT is the full path.
        RC = 9 if Cancel was selected.

MESSAGE STR   Show the STR as screentitle (For a short time)

CREATELABEL STR EX
        Create a label with the name STR and the number EX.
        Note: You must not free the labels at programend !

DELETELABEL STR   Delete the label STR. If this label did'nt exists, the
        RC will be set to 7 if the label not exists.

NEXT   Go to next line.
        RC=7 Task is still running.RC=8 The programend is reached.

STACKBACK   Go out of the current subprogram or jump to UNLINK.
        RC=7 Task is still running.RC=8 The programend is reached.

DTTRACE   Trace one step.
        RC=7 Task is still running.RC=8 The programend is reached.

STEP   STEP with stack.

```
        RC=7 Task is still running.RC=8 The programend is reached.

BREAKPOINT EX     Set breakpoint to EX. (Old one will be deleted)


UNTILREG  mREG[.b,.w,.l]
        Run and stop program when the register is modified.
        RC = 6 for syntax error
UNTILREG  sREG[.b,.w,.l] EX
        Run the program until the register REG reached the until
        value.
        RC = 6 for syntax error
UNTIL mADR[.b,.w,.l]
        Run until the adress with the given size is modified.
        RC = 6 for syntax error
UNTIL sADR[.b,.w,.l] EX
        Run until the adress has the specified value.
        RC = 6 for syntax error


LASTNAME    Get name of last loaded file.
        RC = 8 for no program loaded.
RUN     Start the program.
READSECTORS EX START LENGTH UNIT DEVICE
        Read sectors from a device.
WRITESECTORS EX START LENGTH UNIT DEVICE
        Read sectors from a device.
FINDDSM START END EX
        Find an instruction between START and END.
        RC = 7 : found expression
        RC = 9 : User break
FINDMEM START END EX
        Find an instruction between START and END.
        RC = 7 : found expression
        RC = 9 : User break
        RC = 3 : Error in size statement
        RC = 4 : Error in calcexpr
Other notes:

Following arexx exaples are in the s:rexx drawer:

GetDTOffset.rexx      Show you with the amigaguide library and the
          autodoc guide files the description of the last
          function of an system call.
          Please look at the arexx-program for more infos
          about some amigaguide versions if this program
          does'nt work.

GetDTOffsetCED.rexx   The same function as above, but show the desc.
          with CygnosED. (Start CED first)
          For CED V3.5. (Preversions has incompatible search
          function)

DTDis.rexx            Little Disassembler by Stefan Guttmann.
```

1.25   screen

```
SCREEN:
```

The DT is a public screen called "DTSCREEN.x". X is the number of the
DT-Task.

Example to use:

NEWSHELL CON:0/0/600/200/DT-Shell/CLOSE/SCREENDTSCREEN.1

To open a shell on the DT Screen.

1.26 general keys

```
GENERAL KEYS:
```

TAB-Key is General used to switch to next window. Please use this option
it is very useful.

1.27 history

```
V1.07              28.08.93
  Nearly first bugfree version.
V1.10              02.09.93
  Binary save: If filelength -1, length of last loaded file will be
  used.
V1.12              04.09.93
  After use at the prefswindow, refresh of all windows
  Refresh after image load.
  Error with timebar display with little adresses corrected.
  Create segment names with devpacstart too.
  Only escape breaks the find requester.
  Cancel with error pubscreen close ends the debugger.
V1.14              06.09.93
  J-Key is jump over instruction without execute.
  W-Key is find next in debug window.
  Adressregister check for illegal adresses (Gurus were possible)
  Lock in memory window now with key L too.
  Prefswindow-> Use gadget quits the prefswindow.
  Memwindow: C-Key is calcstring not close (Esc-key not in editmode
  is close)
  68000: Guru with odd adresses in registerwindow.
V1.16              18.09.93
  If startchar is '(' then jump over this tooltype.
  Dynamic labellist. Now program is 42k shorter!
  Key e in debugwindow is execute dos command.
  Modify register with running task forbidden.
  {xx} in calcstring for pointer operations! (Also recursiv useable)
  Labellist window with HELP-Key in all expression windows.
  (You can fix the list window with SavePrefs!)
  Bug fixed with unfreed memory. (4k for gadgets)
  Bug fixed with odd adresses in memory window in edit mode.
```

Save prefs with open expression window failed!

V1.23   05.11.93
  Size check in expression strings for relative adresses ({}.w, {}.l).
  (Standard is long)
  Calcerror: Better failure description.
  Replace STD-Include in prefs window with start execute command.
  (To start rexx or other programs at prgstart), (for overjump exec.
  O Key in memory- and dsmwindow for offset displaying.
  W is find next in memorywindow.
  Size .u with relative adresses for unsingned wordread.
  CLI-Argument is the filename to load at start.
  Sort labels in labellistwindow.
  Labellistwindow refresh speeded.
  Clear labels when a prg is loaded and an other prg want to load.
  Clear reg in regwindow clears FP-Registers too.
  Case insesitive labels (for arexx reqiered!)
  Rexx commands added().
  Error with lock and offset removed.
  Corrected error with relative calc with complex expressions.
  Now use utility librarys mul and div for the math operations.
  Added mathoperator : for mod function.
  Added mathoperator ^.
  If a new Stringgad is empty, the last string contents will be shown.
  Disassembler error with ROR,ROL,ROXL,ROXR corrected.
  Added diskoperations sectorread,write, bootblockchks.(Keys:[,] , /)
  Error with closeing timebar window.
  Copymemory added. (Key: V)
  Corrected calculateroutine for sign regarding operations.(programmed
        by Steve Guttmann)
  Image load bug fixed with refresh. (hang up sometimes)
  Expression window: When cursor is not on first position, the values
  -that clicked from other windows will be inserted at cursor.
  Locale added. (german, english)
  68030 adressingmodes added.
  Disassembler speeded.
  Error with longadresses with Bcc.w .
  Automatic close screen if last pupwindow closed.
  SetDisAdr-Rexx command does'nt cut the result.
  Ask screenmode if open of the screen failed or theres no prefsfile.
  Sometimes guru when no DT-prefs file in the s-drawer.(Screenmode)
  Prefsfile now iff-standard.
  Prefs added: Case insesitive for labels, Sort labels at start->
  Add Function key definingto prefs.
  Added labels ENDx for end of segmends.
  Main prefs now iff.
  Size for all asl reqesters come from mainprefsfile.
  Sizeable debugwindow, changing all gadgets.
  Stack will not be deleted to direction down. (To look for last JSR
  -backadress!)
  Std-windows changed.

V1.24   08.12.93
  Workbench start simulation works now.
  Wrong adresses of endX if started from devpac.
  Offset displaying works now with second (all) debugwindows too.
  Minimal size of second debugwindow corrected.

Checked rexx message also if rexx not present.

V1.25
  Problems with open prefswindow->Quit->Restart (with devpac)
  Find Next Intruction works now correct. (Searched with odd addresses)
  Removed size gadget on are-you-sure-window.
  Rewritten the ParseFormula routine for bracket calculations.(Works
  not correct in special cases.
  Prefsfiles default now in ENVARC:DT drawer.
V1.27   21.01.94
  Sizechange of labellistwindow speeded. (No more window close&open)
  Waiting pointer for timebar. (Search functions)
  Expressionwindow ESC-Key is close.
  Cursor for history puffer.
  ESC key in expression window to cancel.
  MaxGadgets corrected for prefswindow recoverable alerts.
  Added integer gadget in prefswindow for history size.
  With relative optimizing 1k shorter,faster.
  Iconify with key I and tooltype ICONIFY.
  Drag program on appicon possible.
  In the editmode of the memory window the key q was quit.
  ARexx FindDsm and FindMem added.
  UnLoad with prg at prgend failure.
  Optimized errorcode handling.
  Close all expr windows if the window of the expr is closed.
  Sure requester are now autorequesters.
  No more unload question if program is finished.
  Click on labellist now imports the labelname not the labelvalue.
  DT starts sometimes with std. windows. (Error with labelname copylength)
  Debug key K for kill and add new debug trace task.
  After SET in prefswindow, only drawers was possible to load.
  Key k hangs up if the debugtask was suspended from the system.
  If one hunk has no symbols the next symbols are added to the wrong hunk.
  ARexx struct reader added.
  Unlock missing with workbench emulation.
  Key G to open a memory window (if all closes the only way).
  Correct display of memorydump with odd addresses.
  Special version for work with enforcer developertools.

  First official release.