

**003c34e0-0**

**COLLABORATORS**

	<i>TITLE :</i> 003c34e0-0		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 18, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>003c34e0-0</b>	<b>1</b>
1.1	MSGRead-Guide . . . . .	1
1.2	Contents . . . . .	1
1.3	Disclaimer . . . . .	2
1.4	Introduction . . . . .	2
1.5	Technical Information . . . . .	3
1.6	Bugs . . . . .	3
1.7	Other Copyrights . . . . .	4
1.8	MSGIdx Usage . . . . .	4
1.9	Last Words... . . . .	4
1.10	Sysop's Task . . . . .	5
1.11	MSGReads CLI-Template . . . . .	5
1.12	Some things changed... . . . .	6
1.13	Things still to do... . . . .	6
1.14	MSGReads Setup and general Usage . . . . .	6
1.15	MSGRead in duty . . . . .	7

---

# Chapter 1

## 003c34e0-0

### 1.1 MSGRead-Guide

MSGRead

```
-----  
A StdInOut-MSG-Reader  
for use with BBS or  
as a VERY SIMPLE  
Standalone-Browser
```

```
(C) 1995 by Marc J. Führer, Germany  
      marc_fuehrer@scooter.ping.de  
MSGRead is FreeWare - all rights are reserved to the author
```

### 1.2 Contents

About MSGRead

Disclaimer

Using MSGIdx

Using MSGRead

MSGReads CL-Template

MSGRead in work

- what the user/you can do

MSGRead-after call

- what the sysop has to do afterwards

---

History  
some changes have been done

Bugs

Things to do

Last Words

Copyrights

Technical Information  
(MSG-Format)

## 1.3 Disclaimer

Disclaimer  
-----

MSGRead is FreeWare. That means it may be copied without making profit - and THIS MEANS: no charge at all may be taken for copying it. This does NOT include the price of disks, CDs or whatever carrier is used as long as they are not saled to gain profit with.

All rights of the program remain by the author (me...) Marc Führer. I am reachable via "marc\_fuehrer@scooter.ping.de".

No guarantee is given for anything.

AmiNet and Fred Fish are (of course) allowed to include MSGRead in their collections under the usual terms.

## 1.4 Introduction

Introduction  
-----

Yes... MSGRead is a really simple program. I only wrote it to include my Foozle-Messagebase into WWBBS (see copyrights). But it can be used with just any BBS or via CLI - as long as StdIn and StdOut are forced into the desired direction.

MSGRead gives the user a small choice of actions  
:

---

- walk along the message-headers - showing FROM, TO, SUBJECT and DATE of each message.
- read the actual message
- reply to a message, quoting directly from the original one
- mail a new message - privately or into the actual echo

There are still

enhancements

to be made - this is the first version to be released. I hope to get some hints or wishes!

## 1.5 Technical Information

Excerpt from FOOZLE.TECHREF:

-----

Foozle's MSG-files follow this format:

```

Offset
  0 36B  From User
 36 36B  To User
 72 72B  Subject
144 20B  Date / time
164 WORD Times read (ignored) (No, Peer... not really :- )
166 WORD Dest node
168 WORD Orig node
170 WORD Cost (ignored)
172 WORD Orig net
174 WORD Dest net
176 LONG  0 (ignored)
180 LONG  MS-DOS-TimeStamp of arrival time
184 WORD  Dest zone (matrix only)
186 WORD  Attributes (see FTS-1 for a description)
188 WORD  Dest point (matrix only)
190 STRCT Previous reply (MsgId-structure)
202 STRCT Next reply (MsgId-structure)
214 TEXT  Message text

```

Each MsgId-entry looks like:

```

 0 WORD Zone (bit 15:data-indicator)  \
 2 WORD Net                          \ of the originating
 4 WORD Node                          / system
 6 WORD Point                          /
 8 LONG Data

```

All strings are NULL-terminated.

## 1.6 Bugs

Bugs

----

...no serious ones found so far. I don't consider the missing wordwrapping a bug...

I hope there are no problems with the `_very_` simple break-handling.

## 1.7 Other Copyrights

Copyrights

-----

Foosle - is copyright © 1994 Peer Hasselmeyer

WWBBS - World Wide BBS is Copyright © 1995 Arthur Choung

## 1.8 MSGIdx Usage

MSGIdx's usage is quite simple:

Give it a specific directory name with  
DIR=... (e.g. DIR=mail:echo/amiga.ger)

or: give it a config-file with  
CONFIG=... (e.g. MSGIdx.config)

or: start it with HAVING a configfile named "msgidx.cfg"

The configfile simply consists of directorynames with full path! e.g.:

```
-----cut-----  
mail:echo/amiga.ger  
mail:echo/ami_bla.ger  
mail:echo/blabla.fin  
mail:echo/chat.int  
-----cut-----
```

After MSGIdx has done it's work you can call up MSGRead.

## 1.9 Last Words...

Last Words

-----

I wrote this program in the evenings hours with my little girl jumping onto me, my wife chatting to me, my computer gurning due to an overloaded powersupply, dates cutting down the time...

---

Still I hope to give away some software of use. I will improve it, I will kill the bugs and I will go to bed early tonight.

Please mail me if you found something to change, if you like the program, if the weather is fine or if you simply do not know what is meant by AMIGA.

Please remember that french is not my mother's motherlanguage (or was it italian I wrote this in?)

## 1.10 Sysop's Task

Sysop's task

After MSGRead is ended you, as the sysop, have to do whatever is needed to rescan, toss, export the new messages. If you use Foozle this would be:

```
TellFoozle FRescan Chg (use FRescan instead of Rescan: reb. .map-file!)
- and maybe, if you export every time "tellfoozle export"
```

AND: of course you sometimes have to update your MSG.IDX-files that are written by

MSGIdx

. They are not updated automatically for a simple reason: if you, as the sysop, want to "control" the new messages before exporting them (maybe to delete those written by an unwanted user) you would have to kill the corresponding lines from the indeces.

Don't worry that MSGRead could ever write a new message twice... it does not since it takes the message-number from the ".map" file in the area's directory, keeping Foozle (or whatever) away from overwriting these messages itself:

That means: if you use some tosser other than foozle (which should work if the

\*MSG-format

is the same) you can easily generate some .map-file for each area-directory. It has to be at least 8 bytes long, since the number of the next message to be written starts at offset 4.

## 1.11 MSGReads CLI-Template

Template:

```
MSGRead DIR=directory-to-scan LINES=number_of_screen_lines USER=well_guess!
```

DIR

I think you can imagine what the keyword DIR is for. MSGRead is designed to work with a SINGLE directory on every call - so all "level-controlling", all "groupings" and what-the-heck can be done via your BBS. You simply call MSGRead if you want to.



## LINES

Since there is no standard screen-height in BBS (how could?) you can specify the number of lines your actual user has chosen. Refer to your BBS's manual for how to get that value! 24 is taken as standard. Please remember that LINES means all lines on the screen including MSGRead's own header - and there has to be a "minimum" of lines to really work with - it doesn't make sense to use MSGRead with a single line... so in short: minimum LINES is 10.

## USER

All characters till the end of the line are taken as part of the user's name. Again, refer to your BBS's manual for how to get the user's name - mostly you can somehow put it into an ENV-variable.

## 1.12 Some things changed...

### History

-----

- V1.0 - first official release though a bit buggy
- V1.1 - fixed bug that caused MSGRead to convert ALL texts from config to lower case
  - fixed bug that cut off the Origin
  - FRescan should be used with Foozle
- V1.2 - internal release, fixed some memory bugs
- V1.3 - improved BreakHandling

## 1.13 Things still to do...

### Things to do

-----

I do not dare to count them... :-)

In fact: there are some things I would like to do:

- rethink format of overviewpage (that is: should I cut the FROM and TO field to give it an unique image?)
- built-in word wrapping
- better break-handling (yet only via CheckSignal)
- help-pages? may be...
- perhaps NodeList support

## 1.14 MSGReads Setup and general Usage

### Usage

-----

Err... Well... I'd say... ok.

Let me explain how MSGRead works. You should find a program called

---

MSGIdx

in the same directory you unpacked MSGRead to. It generates an Indexfile for ←

each area-directory you want MSGRead to use in. This Indexfile contains the writer, receiver and subject of every message in the directory, sorted by their `_originating_` date. Meaning: the first listed message in the Index is the oldest, the last the latest - depending NOT on the date of tossing but the date of writing! (...and that's the reason why MSGIdx takes some time to generate the Index... it has to open each MSG-file)

Use it to generate the Indexfiles for MSGRead!

MSGRead itself looks for a configfile called MSGRead.cfg. This config must contain some keywords as there are:

```
NODE=<your node number> (Format zone:net/node.point)
ORIGIN=<your desired origin-line> (no " * Origin:", no (x:xxx/xxx) needed)
TEARLINE=<your tearline> (may be a single space)
EDITOR=<commandline>
MATRIX=<full path to your NetMail-Folder>
```

Node is, as I suppose, selfexplanatory. Such is Origin and Tearline... Editor must hold the call for your BBS-online-editor (e.g. emacs), where the filename to edit is added by MSGRead.

Matrix is the path to your NetMail-folder, e.g. MATRIX=mail:echo/matrix It is used for private replies or mails.

As soon as you have set up MSGRead.cfg you can start MSGRead with its

```
template
.
```

## 1.15 MSGRead in duty

Working with MSGRead

You (or the user) get a short help line at the screentop, showing all keys to use while MSGRead is working. These are:

```
8: scroll mark one message up (or, when viewing a message: scroll one page
up)
2: scroll mark one message down (or, when viewing a message: scroll one
page down)
Return: read actually marked message
r: reply to actually marked message, quoting it's text
m: mail a new message into the echo or privately into MATRIX
x: leave MSGRead (or, when viewing a message: go back to overview)
```

After you have written or edited a message you are asked some questions:

- Do you want to address your mail to ...  
Here the author of the message you replied to is put in - or ALL.
  - Is your mail private (No=EchoMail!) ?  
If you answer with "y" (Yes) your mail is sent via MATRIX, not into the echo. You have \_then\_ to answer:
    - Is this address correct: x:x/x.x  
It should be correct if you REPLY to a message, if you originate one you should answer "n" (NO) and type in the correct address.
  - Do you want to change the subject (...)  
If you don't, answer "n" (NO) - elseway type in a new subjectline.
-