CrashMail II

The Next Generation!

...a stranger in a strange land...

============

Introduction

============

Welcome to CrashMail II! CrashMail II is basically a more portable version

of CrashMail, my tosser for Amiga computers. Users of the old Amiga

version will probably find some things familiar while some features are

gone such as the ARexx port (for obvious reasons!) and the GUI

configuration editor. The only feature that CrashMail II has and the old

CrashMail hasn't is support for JAM messagebases. (By the way, I have also

written a tick file processor called CrashTick for the Amiga. If someone

wants to port it, contect me and I'll give you the source.)

For suggestions, bug reports and questions, don't hesitate to contact me at:

billing@df.lth.se

=========

Copyright

=========

Copyright (C) 1998-2004, Johan Billing <billing@df.lth.se>

Copyright (C) 1999-2010, Peter Krefting <peter@softwolves.pp.se>

Copyright (C) 2009-2013, Robert James Clay <jame@rocasa.us>

Copyright (C) 2013, Lars Kellogg-Stedman <lars@oddbit.com>

JAMLIB is copyright (c) 1999 Björn Stenberg. JAMLIB is released under the

GNU Lesser General Public License, See src/jamlib/jamlib.doc for more

information.

Except where explicitly stated otherwise, all other parts of CrashMail are

copyright 1998-2004 Johan Billing. Permission to use, copy and distribute

CrashMail is granted provided that this copyright notice is included. Permission

to modify CrashMail is granted. Distributing modified versions of CrashMail is

allowed provided that the documentation clearly states that it is a modified

version. Parts of CrashMail may be freely used in other projects as long as

the documentation mentions the original copyright holder.

================

Acknowledgements

================

Many thanks to Björn Stenberg for creating the excellent subroutine library

JAMLIB which CrashMail uses for handling JAM messagebases.

Thanks for Peter Karlsson for porting CrashMail II to OS/2 and the man pages.

=============

Documentation

=============

The documentation is very brief and CrashMail probably isn't the ideal

choice for Fidonet beginners. All documentation of the available keywords

in the configuration file can be found in the example crashmail.prefs file.

Some other items about CrashMail that are worth mentioning can be found in

the section below.

Items that need to be discussed

===============================

Platforms

---------

This version of CrashMail can be compiled for Win32, Linux and OS/2. If you

are interested in running CrashMail on another platform, please contact me if

you are willing to do the work necessary to adapt CrashMail to your platform.

The amount of work required mostly depends on whether your C-compiler supports

some common POSIX-functions which CrashMail uses.

Some notes on different platforms:

Win32 & OS/2

If you want to use an old reader that only can handle 8+3 filenames,

you have to use %8 in the path of your DEFAULT area if you are using

the auto-add feature. This creates an 8 digit serial number to use as

the path for the area. Note that if CrashMail is run twice in a short

period of time (a few seconds), it might create duplicate paths. Avoid

%8 if it is at all possible.

Linux

Don't use the ~ character in paths. Such paths are expanded to point

to your home directory by the shell and not by the i/o functions in

the system. They will not work in CrashMail.

In \*.msg areas, make sure that all files are named \*.msg and not \*.MSG!

If they are not named in lowercase, CrashMail will not export them.

As an extra bonus, the Linux version of CrashMail can use the syslog instead

of using its own log file. Just use "syslog" as the name of your log file.

If the precompiled binaries in the CrashMail archive don't work on your

system, you will have to compile your own. See INSTALL for more

information about this.

Arguments

---------

Available arguments for CrashMail:

SCAN

Scan all areas for messages to export.

TOSS

Toss all .pkt files and bundles in inbound directory.

TOSSFILE <string>

Toss the specified file.

TOSSDIR <string>

Toss all files in the specified directory.

SCANAREA <string>

Scan the specified area.

SCANLIST <string>

Scan all areas listed in the specified file.

SCANDOTJAM <string>

Scan all areas listed in an echomail.jam/netmail.jam file. The main difference

between SCANDOTJAM and SCANLIST is that a \*.jam file contains the paths to the

messagebases instead of tagnames. Areas are only scanned once even if listed

multiple times.

RESCAN <string>

RESCANNODE <string>

RESCANMAX <string>

Rescans the specied area for the specied node. If RESCANMAX is specified,

it sets the maximum number of messages to rescan.

SETTINGS <string>

Use this configuration file instead of the default. You can use the

environment variable CMCONFIGFILE to set the default configuration file.

VERSION

Show version information about CrashMail.

LOCK

Locks CrashMail's configuration file and then exits. CrashMail has a simple

locking mechanism to ensure that two instances of CrashMail never use the

same configuration file at the same time. You can use this if you want to

temporarily want to stop CrashMail from running, e.g. when updating the

nodelist.

UNLOCK

Removes the lock on CrashMail's configuration file. Only use this when the

configuration file previously has been locked with LOCK, otherwise terrible

things might happen.

NOSECURITY

Process all packets without security checks. This is intended to be used

mainly with TOSSDIR/TOSSFILE and with packets created by CrashWrite.

Support programs

----------------

crashexport <crashmail.prefs> <output file> <format> [GROUP <groups>]

This command reads a CrashMail configuration file and creates an arealist.

If the GROUP keyword is used, only areas in the specified groups are

included. CrashExport can create lists in these formats:

AREASBBS

A standard areas.bbs file that can be read by many programs

FORWARD

A list of areas that can be used for forward-requests on other nodes.

The file is a pure ASCII file where each line contains the name of the

area and its description.

FORWARDNODESC

Same as FORWARD but without area descriptions.

GOLDED

Creates an area configuration file in GoldED format.

TIMED

Creates an area configuration file in timEd format.

crashstats <statsfile> [SORT <mode>] [LAST7] [NONODES] [NOAREAS]

This command displays the statistics file created by CrashMail. With the

SORT keyword you can specify these sort modes:

a Sort alphabetically

t Sort by total number of messages

m Sort by msgs/day

d Sort by first time messages were imported

l Sort by last time messages were importd

u Sort by number of dupes

With LAST7, you can see detailed information about the flow of messages in

area areas for the last seven days. With NONODES and NOAREAS you can decide

to hide node or area statistics.

crashlist [<dir>]

Builds an index for the nodelists in the specified directory (or in the

current directory if no directory is specified). To find out what

nodelists to read, CrashList uses a file called cmnodelist.prefs in the

nodelist directory. The format of this file is as follows:

<nodelist name> [<default zone>]

As the name of the nodelist, you can either specify the full name of the

nodelist or just the base name of the nodelist (without .xxx at the end).

If you just specify the base name, CrashList will use the latest nodelist

with that name (selected by date, not the extension). A default zone can

be used for regional nodelists without a Zone line. All lines beginning

with a semicolon are treated as comments. Pointlists should be in

BinkleyTerm format and should be specified after the real nodelists.

Example cmnodelist.prefs:

; Configuration for CrashList

;

; Format: <nodelist> [<default zone>]

NODELIST

BTPOINT

crashgetnode <node> [<nodelist dir>]

Looks up the specified node in the nodelist and prints the information

that was found. If no nodelist directory is specified, CrashGetNode uses

the path specified in the environment variable CMNODELISTDIR.

crashmaint [MAINT] [PACK] [VERBOSE] [SETTINGS <filename>] [PATTERN <pattern>]

Deletes old messages according to KEEPNUM and KEEPDAYS in crashmail.prefs. The

program can do two different operations on a messagebase, MAINT and PACK. The

meaning of these two modes are different for different messagebase formats.

\*.msg

MAINT deletes messages and PACK renumbers the area.

JAM

MAINT sets the Deleted flag for the messages. PACK removes all messages with

the Deleted flag from the messagebase.

Both MAINT and PACK can be specified at the same time. You can specify a

config file other than the default with the SETTINGS keyword (use the

environment variable CMCONFIGFILE to set the default configuration file).

Using the PATTERN keyword, you can perform the operations on only some of your

areas. VERBOSE gives you a lot of information which you don't really need.

Example: crashmaint MAINT PACK PATTERN R20\_AMIGA\*

crashwrite DIR <directory> ...

CrashWrite reads a text file and creates a .pkt file that can be processed

by CrashMail. This can be used to post announcements and other messages in

areas. The best way to use CrashWrite is to let it generate packets in a

separate directory and then toss them with TOSSDIR NOSECURITY.

There are many keywords for CrashWrite. All keywords are optional except for

DIRECTORY. If you do not enter a keyword, a default value will be used.

FROMNAME <string>

FROMADDR <node>

TONAME <string>

TOADDR <node>

SUBJECT <string>

Use these keywords to set the header of the message. You only need to enter

TONAME and TOADDR for netmails.

PKTFROMADDR <string>

PKTTOADDR <string>

Use these if you want to set the origin and destination address of the packet

to something other than the origin and destination address of the message

inside the packet. If you do not specify these keywords, FROMADDR and

TOADDR will be used for the packet as well.

PASSWORD <string>

You can use this keyword to set a password for the packet. The maximum

length of the password is eight characters.

AREA <area>

The area the message should be posted in. If you do not enter an area, the

message will be sent as a netmail.

ORIGIN <origin>

The origin line for the message. This keyword has no effect for netmail

messages.

DIR <dir>

The directory where the packet should be placed.

TEXT <filename>

The name of a text file that should be included as the message text.

NOMSGID

Prevents CrashWrite from adding a MSGID line.

FILEATTACH

Sets the file-attach flag for netmails. The filename should be put in the

subject line.

crashlistout <directory> [<zone>] [<pattern>] VERBOSE

This command lists the contents of a outbound directory. Use zone to specify

which zone the directory is for (the default is 2). It is possible to only

list files for nodes that match a specified pattern. If you use the VERBOSE

switch, crashlistout will also list the contents of any \*.req and flow files.

Paths

-----

You should always use absolute paths in crashmail.prefs, otherwise CrashMail

will fail to unpack incoming bundles. If you use relative paths, CrashMail

will also use relative paths in flow files which might confuse your mailer.

Outbound

--------

CrashMail uses a 5D BinkleyTerm outbound. If there is a demand for FrontDoor

style outbounds (\*.msg based), it might be implemented in a future version.

Messagebase formats

-------------------

CrashMail currently can use \*.msg messagebase and JAM messagebases.

Some notes on the different messagebase formats:

\*.msg

\*.msg is the most basic format for Fidonet messages. It is specified in

FTS-0001 and most Fidonet programs can handle this. There are however some

variations. There are Zone and Point fields in the message header, but

since some programs use them for other purposes, CrashMail doesn't read

them. This means that CrashMail won't work if your reader doesn't create

INTL, TOPT and FMPT kludge lines. Most readers do so this probably won't

be a problem.

JAM

JAM is a newer messageformat which while not perfect at least is much

better than \*.msg. It provides reply-linking, but unfortunately not

between areas. JAM has a few odd features which CrashMail does not

support. CrashMail will not create TRACE fields from Via kludges, it

does not support messages with multiple recipients (carbon copies) and

it does not support file-attaches with wildcards, indirect file-attaches

or file-attaches with aliases. CrashMail also handles only one attached

file/file request per message.

Highwater marks

---------------

CrashMail can use highwater marks to speed up the exporting of messages. The

highwater mark is only the number of the highest exported message in an area.

If you decide to use highwater marks, CrashMail will only export messages

with a message number that is higher that the old highwater mark. If you want

to export messages with a lower number than the highwater mark, you have to

force CrashMail to scan the whole area by deleting the file where the

highwater mark is stored. In \*.msg areas the highwater mark is stored in the

first message of the area (1.msg) and in JAM areas it is stored in the

<basename>.cmhw file. (Also note that this is why the first message in a

\*.msg area never is exported.)

Nodelists

---------

CrashMail can use two nodelist formats:

1) Its own nodelist format ("CMNL"). The format consists of a rather simple

index which is created by the program CrashList. See the descriptions of

CrashList and CrashGetNode for more information.

2) A nodelist in the Version7+ format ("V7+") used by BinkleyTerm and other

programs.

Patterns

--------

String patterns

String patterns are rather primitive in CrashMail. There are two available

wildcards, ? and \*. ? matches any character and \* matches the rest of the

string. ab\*, ab\*de and ab\*de\* are therefore equivalent and all match all

strings beginning with ab. String patterns are used for robot names, remap

names etc.

Node patterns

CrashMail has very powerful pattern matching for nodes. "\*" and "?" can

be used as wildcards and there a special keywords that matches all nodes

that belongs to a zone, region, net, hub or a node.

2:200/207.\*

This would match 2:200/207.1, 2:200/207.2, 2:200/207.42 etc

2:200/2\*.\*

This would match 2:200/213.99, 2:200/224.48, 2:200/207.0 etc.

This would NOT match 2:200/103.42.

2:200/2?.\*

This would match 2:200/24.42, 2:200/25.52 but not 2:200/200.0.

2:\*/100.0

This would match 2:200/100.0, 2:200/100.0, 2:300/100.0 etc.

ZONE 2

This matches everything in zone 2. This has the same effect as 2:\*/\*.\*.

REGION 2:20

This matches everything in region 2:20. You can only use the REGION

keyword if you use a nodelist.

NET 2:200

Matches everything in net 2:200. This is the same as 2:200/\*.\*.

HUB 2:205/300

Matches all node that belongs to the hub 2:205/300. You can only use

the HUB keyword if you use a nodelist.

NODE 2:200/108

Matches the node 2:200/108 and all its points. This does exactly the

same as 2:200/108.\*.

\*:\*/\*.\*

This would match everything.

Destination node patterns

These are a bit more complicated since the destination node of the

operation is also involved. This is best explained with netmail routing

as an example. In CrashMail, destination node patterns are also used

in the remap function, but it works very similarly there.

\*:\*/\*.0, netmail for 2:200/108.7

This netmail would be routed to 2:200/108.0

\*:\*/0.0, netmail for 2:200/108.7

This netmail would be routed to 2:200/0.0

ZONE, netmail for 2:201/274

This netmail is routed to the Zone Coordinator, in this case 2:2/0.

REGION, netmail for 2:200/207.5

This netmail is routed to the Region Coordinator, in this case 2:20/0.

You can only use this keyword if you use a nodelist.

NET, netmail for 2:200/108.7

This netmail is routed to the host of the net, in this case 2:200/0.

This is the same as \*:\*/0.0

HUB, netmail for 2:200/108.7

This netmail is routed to the hub of the node, in this case 2:200/100.

You can only use this keyword if you use a nodelist.

NODE, netmail for 2:200/108.7

This netmail is routed to the boss of the point, in this case 2:200/108.0.

This is equivalent to \*:\*/\*.0.

\*:\*/\*.\*, mail for 2:203/699.0

This would be routed to 2:203/699.0