zmime

COLLABORATORS

	TITLE : zmime		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 23, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

zmime

1.1 $VER: zmime.guide 1.3 (23.6.96)

Zodiac's MIME 1.3 (23.6.96), alias zmime, is Copyright © Ralph ↩
                Seichter,
all rights reserved. You have permission to use this tool freely and make
as many copies as you wish. You must not charge other people for supplying
them with copies of this program. Adding any files to or removing files
from the distribution archive is prohibited.

If you have any questions or suggestions, feel free to contact the author,
but make sure to read this document first. If you are an artist and have an
idea for a suitable zmime icon (NewIcon, MagicWB or standard format), I'd
very much like to see your work. If you want to see a catalog file in your
favourite language, tell me so, and I will supply you with the descriptor
file you require to create a new zmime catalog.

```
  Ralph Seichter    Email: zodiac@darkness.gun.de
  Am Christenrain 12  Phone: +49-2667-969000
  56479 Stein
  Germany
```

zmime is a localized MIME user agent with a MUI interface. It was designed
to both parse and create MIME messages. If you are not familiar with MIME
(Multipurpose Internet Mail Extensions), please refer to RFC-1521 for
details.

Requirements

Disclaimer

Installation

Getting started

Configuration files

Parameter slots

Action configuration

Parsing messages

Composing messages

Sending messages

ARexx commands

History

About MUI

1.2 Requirements

To run zmime, you need Kickstart 2.1 or better and
MUI
Release 3.1 or
better. To make full use of all features, you also need a
text editor
and a
sendmail agent
. The builtin language is english. If you want to use
catalog files (currently there's only a german catalog), locale.library is
also required.

1.3 Disclaimer

DISCLAIMER

THIS MATERIAL IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER
EXPRESS OR IMPLIED, STATUTORY OR OTHERWISE, INCLUDING WITHOUT LIMITATION
ANY IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY AND FITNESS FOR
A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE USE, RESULTS, AND
PERFORMANCE OF THIS MATERIAL IS ASSUMED BY YOU AND IF THE PRODUCT SHOULD
PROVE TO BE DEFECTIVE, YOU ASSUME THE ENTIRE COST OF ALL NECESSARY
SERVICING, REPAIR, OR OTHER REMEDIATION.

1.4 Installation

The easiest way to install zmime is using the supplied Commodore Installer
script. If you insist on a manual installation, please make sure that
zmime and zmime.guide reside in the same directory. Otherwise the online
help will only be available if "ENV:AmigaGuide/Path" contains the directory
in which zmime.guide can be found.

1.5   Getting started

zmime can be run both from the Workbench and from a CLI. The ←
                following
optional parameters can be specified either as ToolTypes or as CLI
parameters.

FILE   Input file, will be parsed immediately.

CONFIG/K
                Configuration file.
                LANGUAGE/K Which catalog file to use ("english" or "deutsch"). If ←
                    you
    don't specify a language, your global preferences will be
    respected.

OPENWIN/K zmime always opens the main user interface window on
    startup. You can use OPENWIN=compose to open the message
    composition aswell, or OPENWIN=config for the configuration
    windown.

PUBSCREEN/K Name of the public screen to open the windows on.

ASKQUIT/S Use this option if you want zmime to pop up a requester
    before exiting.

ICONIFIED/S Open zmime in iconified state. This is useful if you only
    want to use the
                ARexx commands
                .

1.6   Configuration files

On startup, zmime reads a configuration file which contains a list ←
                of
type/subtype patterns and their associated
                actions
                . The default config
file is "zmime.config". Here's an example:

  ; $VER: zmime.config 1.3 (14.4.96)
  ; This file was generated 14-Apr-96 23:30:19

  ADDRESS   joe@foo.bar.edu (Joe Random)
  EDITOR   C:ed "%s" -sticky
  USECRLF   NO

  (text|message)/#?   SYS:Utilities/MultiView $FILE$
  image/#?   SYS:Utilities/viewtek $FILE$

Valid entries consist of an indentifier, at least one whitespace, followed
by the definition body, which may contain blanks. Blank lines and comments
introduced by a semicolon are ignored. If you need to specify a ';' in an

entry, escape it with a backslash. Content type patterns must contain a
slash '/' to allow zmime to distinguish between them and other parameters
in this list:

                    ADDRESS

                    EDITOR

                    REPLYTO

                    SENDMAIL

                    SIGNATURE

                    TEMPDIR

                    USECRLF
                  Please note that when you write back a configuration file, all ↩
                        comments
will be lost. If you want to keep them, you'll have to edit the config file
manually.

1.7 Parameter slots

To pass parameters to external programs (or scripts), zmime uses "slots"
which are filled with the desired data on run-time. A slot is defined by
enclosing a parameter name with two dollar signs, e.g. $FILE$. If you need
a dollar sign in the resulting string, use $$ to define it. Please note
that slot names are not case-sensitive. Some examples:

  MultiView $file$ pubscreen=$PUBSCREEN$
  AmiTCP:bin/smtppost <$FILE$

There are some slot parameters which are always available, no matter what
command is currently executed:

BCC The content of the "Blind copy" gadget.
CC The content of the "Carbon copy" gadget.
$PUBSCREEN$ The string defined with the PUBSCREEN parameter.
$RECEIVER$ The content of the "Receiver" gadget.
$REPLYTO$ The content of the "Reply to" gadget.
$REXXPORT$ The ARexx port name of zmime.
$SENDER$ The content of the "From" gadget.
$SUBJECT$ The content of the "Subject" gadget.

You can refer to the work file by using the $FILE$ slot.

1.8 Action configuration

The configuration window of zmime allows you to define actions for ←
every
type of MIME message you encounter. You must define pairs consisting of a
type/subtype pattern (any dos.library pattern is accepted) and a command to
be executed if a matching entry is found. As an example, have a look at the
following:

```
   image/gif  DH1:gfxtools/ViewGIF $FILE$        $^1$)
   image/#? SYS:Utilities/MultiView $FILE$ pubscreen $PUBSCREEN$  $^2$)
```

If you double-click a message of the type "image/gif", zmime will find the
associated action "DH1:gfxtools/ViewGIF $FILE$". The desired body part of
the message will be saved (and decoded, if necessary) into a temporary
file. The name of this file is inserted in the $FILE$
parameter slot
, then
the command is executed. Please note that zmime will wait for the command
to return, the delete the temporary file.

If you chose a "image/jpeg" type message, line ^{2}) will match, and zmime
will run MultiView. In this situation it is quite clear why ordering does
matter in the action configuration. The first matching entry is taken, so
if you exchanges lines ^{1}) and ^{2}), "image/gif" would be treated in the same
way as "image/jpeg" or "image/{whatever}".

Please keep the following in mind:

   o Content type patterns must contain a slash '/'.

   o Always specify the complete command path.

You can also use two "pseudo commands" in the action definitions, like in
the following example entries:

```
   application/#? .SAVE
   multipart/#?   .FULLSAVE
```

.SAVE decodes and saves the body, while .FULLSAVE saves header and body
(note the preceding dot). If zmime encounters a type/subtype pair for which
no action is defined, it will perform .SAVE as the default action.

1.9 Parsing messages

To parse a MIME message, select it as the input file in the main ←
window. If
the file can be parsed, zmime will display its contents in a hierarchy in
the listview gadget. You are presented a type/subtype entry for each part
of the message, plus the content description or subject line if available.

You can select each part of the message with either the mouse or the cursor
keys. Pressing the return key, clicking the [Action] gadget or double
clicking an entry with the mouse is equivalent will execute the
action

matching the current entry's type/subtype.

If you want to parse multipart messages, select only the first part! zmime
will ask you to select the consecutive parts later on.

1.10 Composing messages

Composing messages is quite easy. You can enter a receiver and a ↩
 subject
line, then you define the message body by selecting as many attachment
files as you wish. When you open a file, zmime checks its contents to find
a type/subtype pair suitable to transmit this file, and will figure out if
encoding the data is necessary. You may change content type and encoding,
but these are power user features and you must make absolutely sure not to
use invalid settings. Leave them alone if you can!

For every file in the message body you can (and usually should) add a brief
descriptive text, which will make things easier for the receiving party.

When everything looks to your satisfaction, you can either
 send the message
 immediately or save it for later use.

1.11 Sending messages

As zmime can't send messages on its own, it relies on external ↩
 programs to
get messages on their way. As an example, you can use the definition

    SENDMAIL   SYS:Rexxc/rx SendMail.zprx $FILE$ "$RECEIVER$" "$SUBJECT$"

if you want to use "Zodiac's Point" to have your message sent. The quotes
in the example above make sure that you can pass blanks in the slots.

Starting with zmime 1.2, the user can configure wether zmime creates
 CR/LF
 (carriage return / line feed) pairs as end-of-line, or if single ↩
 LF codes
are used. In the latter case, the sendmail agent must take care of creating
the proper CR/LF pairs, which MIME messages require by definition.

1.12 Signature file

The file you define here is copied into the editor when you choose to edit
a new message. No signature will be appended if you edit existing files.

1.13   The editor command string

For some operations, zmime requires an editor. It is important that the
editor must not detach itself from the calling process. For example, the
Cygnus Ed requires a special switch to ensure this behaviour. The editor
string thus defaults to ed "$FILE$" -sticky.

1.14   Directory for temporary files

You can specify a directory where zmime's temporary files are put. The
default setting T: is usually the best choice.

1.15   Create CR/LF pairs at each end of line

The Amiga uses single line feed (LF) codes as end of line, while MIME
specifications require carriage return (CR) line feed pairs. Usually, the
sendmail agent will take care of converting the eol sequences if required,
but with this option you can make zmime create CR/LF pairs.

1.16   Your default Internet address

Your default Internet address is used by zmime while composing messages.
Make sure to enter a valid address here! My default address is currently
zodiac@darkness.gun.de (Ralph Seichter), which is a valid address as
defined in RFC 822.

1.17   Your default reply address

                 To redirect answers to your messages to a specific
                 Internet address
                 , define
it here.

1.18   ARexx commands

                 You can send several ARexx commands to zmime, which allow you to ↩
                     compose
and send messages. Here is the complete list:

Compose Add FILE/A,CT=CONTENTTYPE/K,ENCODING/K

   Add a file to the attachment list. Content type and encoding will be
   determined by zmime unless you specify these parameters.

ComposeBcc

   Specify the receiver(s) of a blind carbon copy.

ComposeCc

   Specify the receiver(s) of a carbon copy.

ComposeClear

   Clear the list of all attachments.

ComposeFrom ADDRESS/A

   Specify the originating address.

ComposeSave FILE/A

   Save the current message under the specified file name.

ComposeSend

                 Send the current message.
                 ComposeReceiver ADDRESS/A

   Specify the receiver(s).

ComposeReplyTo ADDRESS/A

   Specify the reply address.

ComposeSubject SUBJECT/A

   Specify the subject.

Quit

   Exit the program.

1.19 Program history

                 zmime 1.0 (24.1.96)

   Initial revision.

zmime 1.1 (4.3.96)

   The settings window and the configuration file format have been
improved.
                 TEMPDIR
                 ,
                 SENDMAIL

```
                    and
                    EDITOR
                     can now be specified in the config
file instead of the ToolType definitions of V1.0. Composing messages has
been improved significantly. Bubble help texts and links to the doc file
were added to many gadgets. If there is no action definition for a
type/subtype pair, zmime will now save the body as default action.
```

zmime 1.2 (29.4.96)

```
  The user can now define a
                    default reply address
                    . End-of-line can be
configured as being either LF or
                    CR/LF
                     pairs. Creating messages has been
generally improved.
```

zmime 1.3 (23.6.96)

```
  Users may now specify a
                    signature file
                     in the configuration. The
Content-Disposition header is now checked for filename specifications, too.
User defined message headers can be temporarily disabled.
```

1.20 About MUI

```
                    This application uses


                    MUI – MagicUserInterface

              (c) Copyright 1993/94 by Stefan Stuntz
```

```
MUI is a system to generate and maintain graphical user interfaces. With
the  aid  of  a  preferences program, the user of an application has the
ability to customize the outfit according to his personal taste.
```

```
MUI is distributed as shareware. To obtain a complete package containing
lots of examples and more information about registration please look for
a  file  called "muiXXusr.lha"  (XX means the latest version number) on
your local bulletin boards or on public domain disks.
```

```
        If you want to register directly, feel free to send


                    DM 30.-  or  US$ 20.-

                            to

                      Stefan Stuntz
                 Eduard-Spranger-Straße 7
```

80935 München
GERMANY

80935 München
GERMANY