

**zmime**

**COLLABORATORS**

	<i>TITLE :</i> zmime		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		February 12, 2023	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>zmime</b>	<b>1</b>
1.1	\$VER: zmime.guide 1.0 (24.1.96)	1
1.2	Requirements	2
1.3	Disclaimer	2
1.4	Installation	2
1.5	Getting started	2
1.6	Configuration files	3
1.7	Parameter slots	4
1.8	Parsing messages	4
1.9	Action definitions	4
1.10	Composing messages	5
1.11	Sending messages	5
1.12	The editor command string	6
1.13	ARexx commands	6
1.14	About MUI	7

# Chapter 1

## zmime

### 1.1 \$VER: zmime.guide 1.0 (24.1.96)

"Zodiac's MIME" (alias zmime) is Copyright (c) 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!

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

Parsing messages

Action definitions

Composing messages

Sending messages

ARexx commands

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.

EDITOR/K

Editor command string.

LANGUAGE/K Which catalog file to use ("english" or "deutsch"). If you

don't specify a language, your global preferences will be respected.

SENDMAIL/K Command string which will be used when the user wants to

send a MIME message

.

TEMPDIR/K Directory for temporary files (default: "T:")

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 window.

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

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:

```
; zmime configuration file, generated 24-Jan-96 17:12:35
```

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

Valid entries consist of a pattern, at least one whitespace, followed by the action definition, 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.

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=$PUBSCREEN$
tcp:bin/sendmail -s "$Subject$" $Receiver$ < $FILE$
```

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

```
$PUBSCREEN$ The string defined with the PUBSCREEN parameter.
$RECEIVER$ The content of the "Receiver" gadget.
$REXXPORT$ The ARexx port name of zmime.
$SUBJECT$ The content of the "Subject" gadget.
```

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

## 1.8 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.

## 1.9 Action definitions

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, then 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 exchange lines \$^1\$) and \$^2\$), "image/gif" would be treated in the same way as "image/jpeg" or "image/{whatever}").

It is important that you always specify the complete command path!

## 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 encode the data if necessary. For every file in the message body you can (and 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.

One very important thing you need to know about the MIME messages zmime



creates is the fact that each line of output ends with a single linefeed code only! RFC specifications require that lines end with carriage-return/linefeed pairs, so you must make sure that your sendmail agent takes care of the end-of-line conversion.

You might ask yourself why zmime doesn't take care of this, and the answer is quite simple. The AmigaOS uses single linefeeds as end-of-line, and as long as the files are not actually transmitted, it is quite usefully to respect the local (AmigaOS) conventions. In the example above, "Zodiac's Point" takes care of the EOL's anyway.

## 1.12 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.

## 1.13 ARexx commands

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

`ComposeAdd 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.

`ComposeClear`

Clear the list of all attachments.

`ComposeSave FILE/A`

Save the current message under the specified file name.

`ComposeSend`

Send the current message.

`ComposeReceiver RECEIVER/A`

Specify the receiver.

`ComposeSubject SUBJECT/A`

Specify the subject.

`Quit`

Exit the program.

---

## 1.14 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