The README file M.T. Rose

Dover Beach Consulting, Inc.

February 22, 2000

Tcl MIME

Abstract

Tcl MIME generates and parses MIME body parts.

Table of Contents

1. SYNOPSIS . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.1 Requirements . . . . . . . . . . . . . . . . . . . . . . . . 3

1.2 Copyrights . . . . . . . . . . . . . . . . . . . . . . . . . 3

2. SYNTAX . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

3. SEMANTICS . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.1 mime::initialize . . . . . . . . . . . . . . . . . . . . . . 5

3.2 mime::finalize . . . . . . . . . . . . . . . . . . . . . . . 5

3.3 mime::getproperty . . . . . . . . . . . . . . . . . . . . . 5

3.4 mime::getheader . . . . . . . . . . . . . . . . . . . . . . 6

3.5 mime::setheader . . . . . . . . . . . . . . . . . . . . . . 6

3.6 mime::getbody . . . . . . . . . . . . . . . . . . . . . . . 6

3.7 mime::copymessage . . . . . . . . . . . . . . . . . . . . . 7

3.8 mime::buildmessage . . . . . . . . . . . . . . . . . . . . . 7

3.9 smtp::sendmessage . . . . . . . . . . . . . . . . . . . . . 7

3.10 mime::parseaddress . . . . . . . . . . . . . . . . . . . . . 8

3.11 mime::parsedatetime . . . . . . . . . . . . . . . . . . . . 9

3.12 mime::mapencoding . . . . . . . . . . . . . . . . . . . . . 9

3.13 mime::reversemapencoding . . . . . . . . . . . . . . . . . . 9

4. EXAMPLES . . . . . . . . . . . . . . . . . . . . . . . . . . 10

References . . . . . . . . . . . . . . . . . . . . . . . . . 12

Author's Address . . . . . . . . . . . . . . . . . . . . . . 12

A. TODO List . . . . . . . . . . . . . . . . . . . . . . . . . 13

B. Acknowledgements . . . . . . . . . . . . . . . . . . . . . . 14

Rose [Page 1]

README Tcl MIME February 2000

1. SYNOPSIS

package provide mime 1.2

package provide smtp 1.2

Tcl MIME is an implementation of a Tcl package that generates and

parses MIME[1] body parts.

Each MIME part consists of a header (zero or more key/value pairs),

an empty line, and a structured body. A MIME part is either a "leaf"

or has (zero or more) subordinates.

MIME defines four keys that may appear in the headers:

Content-Type: describes the data contained in the body ("the

content");

Content-Transfer-Encoding: describes how the content is encoded

for transmission in an ASCII stream;

Content-Description: a textual description of the content; and,

Content-ID: a globally-unique identifier for the content.

Consult [2] for a list of standard content types. Further, consult

[3] for a list of several other header keys (e.g., "To", "cc", etc.)

A simple example might be:

Date: Sun, 04 July 1999 10:38:25 -0600

From: Marshall Rose <mrose@dbc.mtview.ca.us>

To: Andreas Kupries <a.kupries@westend.com>

cc: dnew@messagemedia.com (Darren New)

MIME-Version: 1.0

Content-Type: text/plain; charset="us-ascii"

Content-Description: a simple example

Content-ID: <4294407315.931384918.1@dbc.mtview.ca.us>

Here is the body. In this case, simply plain text.

In addition to an implementation of the mime package, Tcl MIME

includes an implementation of the smtp package.

Rose [Page 2]

README Tcl MIME February 2000

1.1 Requirements

This package requires:

o Tcl/Tk version 8.0.3[4] or later

In addition, this package requires one of the following:

o Trf version 2.0p5[5] or later

o base64 version 2.0 or later (included with tcllib)

If it is available, Trf will be used to provide better performance;

if not, Tcl-only equivalent functions, based on the base64 package,

are used.

1.2 Copyrights

(c) 1999-2000 Marshall T. Rose

Hold harmless the author, and any lawful use is allowed.

Rose [Page 3]

README Tcl MIME February 2000

2. SYNTAX

mime::initialize (Section 3.1) returns a token. Parameters:

?-canonical type/subtype

?-param {key value}?...

?-encoding value?

?-header {key value}?... ?

(-file name | -string value | -parts {token1 ... tokenN})

mime::finalize (Section 3.2) returns an empty string. Parameters:

token ?-subordinates "all" | "dynamic" | "none"?

mime::getproperty (Section 3.3) returns a string or a list of

strings. Parameters:

token ?property | -names?

mime::getheader (Section 3.4) returns a list of strings. Parameters:

token ?key | -names?

mime::setheader (Section 3.5) returns a list of strings. Parameters:

token key value ?-mode "write" | "append" | "delete"?

mime::getbody (Section 3.6) returns a string. Parameters:

?-command callback ?-blocksize octets? ?

mime::copymessage (Section 3.7) returns an empty string. Parameters:

token channel

mime::buildmessage (Section 3.7) returns a string. Parameters:

token

smtp::sendmessage (Section 3.8) returns a list. Parameters:

token ?-servers list? ?-ports list?

?-queue boolean? ?-atleastone boolean?

?-originator string? ?-recipients string?

?-header {key value}?...

mime::parseaddress (Section 3.9) returns a list of serialized

arrays. Parameters:

string

mime::parsedatetime (Section 3.10) returns a string. Parameters:

[string | -now] property

mime::mapencoding (Section 3.10) returns a string. Parameters:

encoding\_name

mime::reversemapencoding (Section 3.10) returns a string. Parameters:

charset\_type

Rose [Page 4]

README Tcl MIME February 2000

3. SEMANTICS

3.1 mime::initialize

mime::initialize creates a MIME part:

o If the -canonical option is present, then the body is in

canonical (raw) form and is found by consulting either the -file,

-string, or -part option.

In addition, both the -param and -header options may occur zero

or more times to specify "Content-Type" parameters (e.g.,

"charset") and header keyword/values (e.g.,

"Content-Disposition"), respectively.

Also, -encoding, if present, specifies the

"Content-Transfer-Encoding" when copying the body.

o If the -canonical option is not present, then the MIME part

contained in either the -file or the -string option is parsed,

dynamically generating subordinates as appropriate.

3.2 mime::finalize

mime::finalize destroys a MIME part.

If the -subordinates option is present, it specifies which

subordinates should also be destroyed. The default value is

"dynamic".

3.3 mime::getproperty

mime::getproperty returns the properties of a MIME part.

The properties are:

property value

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

content the type/subtype describing the content

encoding the "Content-Transfer-Encoding"

params a list of "Content-Type" parameters

parts a list of tokens for the part's subordinates

size the approximate size of the content (unencoded)

The "parts" property is present only if the MIME part has

subordinates.

If mime::getproperty is invoked with the name of a specific

property, then the corresponding value is returned; instead, if

Rose [Page 5]

README Tcl MIME February 2000

-names is specified, a list of all properties is returned;

otherwise, a serialized array of properties and values is returned.

3.4 mime::getheader

mime::getheader returns the header of a MIME part.

A header consists of zero or more key/value pairs. Each value is a

list containing one or more strings.

If mime::getheader is invoked with the name of a specific key, then

a list containing the corresponding value(s) is returned; instead,

if -names is specified, a list of all keys is returned; otherwise, a

serialized array of keys and values is returned. Note that when a

key is specified (e.g., "Subject"), the list returned usually

contains exactly one string; however, some keys (e.g., "Received")

often occur more than once in the header, accordingly the list

returned usually contains more than one string.

3.5 mime::setheader

mime::setheader writes, appends to, or deletes the value associated

with a key in the header.

The value for -mode is one of:

write: the key/value is either created or overwritten (the

default);

append: a new value is appended for the key (creating it as

necessary); or,

delete: all values associated with the key are removed (the

"value" parameter is ignored).

Regardless, mime::setheader returns the previous value associated

with the key.

3.6 mime::getbody

mime::getbody returns the body of a leaf MIME part in canonical form.

If the -command option is present, then it is repeatedly invoked

with a fragment of the body as this:

uplevel #0 $callback [list "data" $fragment]

(The -blocksize option, if present, specifies the maximum size of

each fragment passed to the callback.)

Rose [Page 6]

README Tcl MIME February 2000

When the end of the body is reached, the callback is invoked as:

uplevel #0 $callback "end"

Alternatively, if an error occurs, the callback is invoked as:

uplevel #0 $callback [list "error" reason]

Regardless, the return value of the final invocation of the callback

is propagated upwards by mime::getbody.

If the -command option is absent, then the return value of

mime::getbody is a string containing the MIME part's entire body.

3.7 mime::copymessage

mime::copymessage copies the MIME part to the specified channel.

mime::copymessage operates synchronously, and uses fileevent to

allow asynchronous operations to proceed independently.

3.7 mime::buildmessage

mime::buildmessage returns the MIME part as a string. It is similar

to mime::copymessage, only it returns the data as a return string

instead of writing to a channel.

3.8 smtp::sendmessage

smtp::sendmessage sends a MIME part to an SMTP server. (Note that

this procedure is in the "smtp" package, not the "mime" package.)

The options are:

-servers: a list of SMTP servers (the default is "localhost");

-ports: a list of SMTP ports (the default is 25)

-queue: indicates that the SMTP server should be asked to queue

the message for later processing;

-atleastone: indicates that the SMTP server must find at least

one recipient acceptable for the message to be sent;

-originator: a string containing an 822-style address

specification (if present the header isn't examined for an

originator address);

-recipients: a string containing one or more 822-style address

specifications (if present the header isn't examined for

recipient addresses); and,

-header: a keyword/value pairing (may occur zero or more times).

If the -originator option is not present, the originator address is

taken from "From" (or "Resent-From"); similarly, if the -recipients

option is not present, recipient addresses are taken from "To",

Rose [Page 7]

README Tcl MIME February 2000

"cc", and "Bcc" (or "Resent-To", and so on). Note that the header

key/values supplied by the "-header" option (not those present in

the MIME part) are consulted. Regardless, header key/values are

added to the outgoing message as necessary to ensure that a valid

822-style message is sent.

smtp::sendmessage returns a list indicating which recipients were

unacceptable to the SMTP server. Each element of the list is another

list, containing the address, an SMTP error code, and a textual

diagnostic. Depending on the -atleastone option and the intended

recipients,, a non-empty list may still indicate that the message

was accepted by the server.

3.9 mime::parseaddress

mime::parseaddr takes a string containing one or more 822-style

address specifications and returns a list of serialized arrays, one

element for each address specified in the argument.

Each serialized array contains these properties:

property value

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

address local@domain

comment 822-style comment

domain the domain part (rhs)

error non-empty on a parse error

group this address begins a group

friendly user-friendly rendering

local the local part (lhs)

memberP this address belongs to a group

phrase the phrase part

proper 822-style address specification

route 822-style route specification (obsolete)

Note that one or more of these properties may be empty.

Rose [Page 8]

README Tcl MIME February 2000

3.10 mime::parsedatetime

mime::parsedatetime takes a string containing an 822-style date-time

specification and returns the specified property.

The list of properties and their ranges are:

property range

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

hour 0 .. 23

lmonth January, February, ..., December

lweekday Sunday, Monday, ... Saturday

mday 1 .. 31

min 0 .. 59

mon 1 .. 12

month Jan, Feb, ..., Dec

proper 822-style date-time specification

rclock elapsed seconds between then and now

sec 0 .. 59

wday 0 .. 6 (Sun .. Mon)

weekday Sun, Mon, ..., Sat

yday 1 .. 366

year 1900 ...

zone -720 .. 720 (minutes east of GMT)

3.10 mime::mapencoding

mime::mapencodings maps tcl encodings onto the proper names for their

MIME charset type. This is only done for encodings whose charset types

were known. The remaining encodings return "" for now.

3.10 mime::reversemapencoding

mime::reversemapencoding maps MIME charset types onto tcl encoding names.

Those that are unknown return "".

Rose [Page 9]

README Tcl MIME February 2000

4. EXAMPLES

package require mime 1.0

package require smtp 1.0

# create an image

set imageT [mime::initialize -canonical image/gif \

-file logo.gif]

# parse a message

set messageT [mime::initialize -file example.msg]

# recursively traverse a message looking for primary recipients

proc traverse {token} {

set result ""

# depth-first search

if {![catch { mime::getproperty $token parts } parts]} {

foreach part $parts {

set result [concat $result [traverse $part]]

}

}

# one value for each line occuring in the header

foreach value [mime::getheader $token To] {

foreach addr [mime::parseaddress $value] {

catch { unset aprops }

array set aprops $addr

lappend result $aprops(address)

}

}

return $result

}

# create a multipart containing both, and a timestamp

set multiT [mime::initialize -canonical multipart/mixed

-parts [list $imageT $messageT]]

Rose [Page 10]

README Tcl MIME February 2000

# send it to some friends

smtp::sendmessage $multiT \

-header [list From "Marshall Rose <mrose@dbc.mtview.ca.us>"] \

-header [list To "Andreas Kupries <a.kupries@westend.com>"] \

-header [list cc "dnew@messagemedia.com (Darren New)"] \

-header [list Subject "test message..."]

# clean everything up

mime::finalize $multiT -subordinates all

Rose [Page 11]

README Tcl MIME February 2000

References

[1] Freed, N. and N.S. Borenstein, "Multipurpose Internet Mail

Extensions (MIME) Part One: Format of Internet Message Bodies",

RFC 2045, November 1996.

[2] Freed, N. and N.S. Borenstein, "Multipurpose Internet Mail

Extensions (MIME) Part Two: Media Types", RFC 2046, November

1995.

[3] Crocker, D., "Standard for the format of ARPA Internet Text

Messages", RFC 822, STD 11, August 1982.

[4] http://www.scriptics.com/software/8.1.html

[5] http://www.oche.de/~akupries/soft/trf/

[6] mailto:dnew@messagemedia.com

[7] mailto:a.kupries@westend.com

Author's Address

Marshall T. Rose

Dover Beach Consulting, Inc.

POB 255268

Sacramento, CA 95865-5268

US

Phone: +1 916 483 8878

Fax: +1 916 483 8848

EMail: mrose@dbc.mtview.ca.us

Rose [Page 12]

README Tcl MIME February 2000

Appendix A. TODO List

mime::initialize

\* well-defined errorCode values

\* catch nested errors when processing a multipart

Rose [Page 13]

README Tcl MIME February 2000

Appendix B. Acknowledgements

This package is influenced by the safe-tcl package (Borenstein and

Rose, circa 1993), and also by Darren New[6]'s unpublished package

of 1999.

This package makes use of Andreas Kupries[7]'s excellent Trf package.

Rose [Page 14]