MHonArc

v1.2.2

April 17, 1996

Table of Contents

Introduction

MHonArc is a Perl program for converting e-mail messages as specified in RFC 822 and RFC 1521 (*MIME*) to HTML. *MHonArc* can perform the following tasks:

- Convert `mh`(1) mail folders or UUCP/Unix style mailboxes into an HTML mail archive.
- Add or remove messages to an existing HTML mail archive generated by *MHonArc*.
- Convert a single message to HTML.

Along with these tasks, *MHonArc* provides the following:

- A main customizable index page for mail messages archived.
- A customizable thread index page listing messages by thread.
- Control over message formatting.
- The ability to hook in your own custom message filters.

Why Use MHonArc?

Here are some reasons for using *MHonArc*:

- You want to keep organized archives of mail messages and/or news articles for a World Wide Web (*WWW*) server; complete with live hypertext pointers to their authors and to any url's mentioned.
- You would like to control the layout of mail/news archives to keep a consistent style to your WWW pages.
- You have a WWW client, but no MIME mail reader. *MHonArc* will allow you to read MIME messages that includes images, audio, video, etc via your Web client.
- Muli-platform support: MS-DOS and Unix.
- You think the *MHonArc* logo is really cool, and it deserves to be used.
- You like Perl, and you want to see what it can do.
- Just cuz.

Supported Platforms

MHonArc (version 1.1, or later) will run under Unix or MS-DOS operating systems with Perl 4 or 5 installed.

Availability

The latest information on *MHonArc*, and its availability, may be obtained at <URL:http://www.oac.uci.edu/indiv/ehood/mhonarc.html>.

About the Documentation

The documentation is oriented towards Unix users. However, the section on Installation does conver MS-DOS installation instructions. Notes are made in the documentation when something may differ due to the operating system.

Installation

This section instructs you on how to install and get *MHonArc* running on your machine. The section covers *Unix* installation and *MS-DOS/Windows* installation.

NOTE

For brevity, anything that applies to MS-DOS also applies to Windows.

System Requirements

MHonArc is written in Perl 4. Therefore, you must have Perl 4 or 5 installed on your system. If you do not know if Perl is installed on your system, ask your system administrator.

If Perl is not installed on your system, you can retrieve Perl at <URL:http://www.cis.ufl.edu/perl/ftp.html>. I recommend version 4.0 patchlevel 34, or later. *MHonArc* has not been tested on earlier versions.

NOTE

MHonArc makes use of the Perl libraries `newgetopt.pl` and `timelocal.pl`. These libraries are part of the normal Perl distribution.

Extracting the Distribution File

Before extracting the distribution file, you may want copy the distribution file into scratch directory, and work in there during installation.

Tar/Gzip Distribution

You must have *gzip* and *tar* installed on your system. If *gzip* is not installed, you may obtain *gzip* at <URL:ftp://prep.ai.mit.edu/pub/gnu>. *Tar* comes with all Unix systems. However, MS-DOS users may have to obtain *tar*.

To extract the file, type the following command at your shell's prompt:

Unix

```
zcat MHonArc.tar.gz | tar xvof -
```

MS-DOS

```
gunzip -dv MHonArc.tar.gz
tar xvf MHonArc.tar
```

A directory called "`MHonArc`" should be created. The directory contains all the files need for installing *MHonArc*.

NOTE

The actual name of the distribution file may differ from the example given.

Zip Distribution

You must have *pkzip* or *unzip* installed on your system.

To extract the file, type the following command at your shell's prompt:

```
unzip mhonarc.zip
```

OR

```
pkunzip -d mhonarc.zip
```

IMPORTANT

The directory structure of the zip file must be preserved during extraction to insure proper installation.

A directory called "`MHonArc`" should be created. The directory contains all the files need for installing *MHonArc*.

NOTE

The actual name of the distribution file may differ from the example given.

Installing the Software

Once you have extracted the distribution file, change your current working directory into the `MHonArc` directory created during the extraction of the distribution file.

Example: Assuming you are in the directory you extracted the distribution file in, you can type the following on your command-line:

Unix

```
cd MHonArc
```

MS-DOS

```
cd MHONARC
```

`install.me`

Contained in the `MHonArc` directory is a Perl program called "`install.me`". This program will perform the tasks required to install *MHonArc* on you machine. The install program is capable of running interactively, or in batch.

Interactive Mode

To run `install.me` in interactive mode, type the following at your shell's prompt:

```
perl install.me
```

NOTE

Make sure you are in the same directory as the `install.me` program.

The program will then prompt you for the necessary information to install *MHonArc* on your system.

Here’s an example (Unix) session:

```
% perl install.me
MHonArc Installation
====================
The installation process will ask you a series of questions on where
the Perl executable is and where to put MHonArc files.  Just hit <CR>
to accept the default values listed in ()'s.

If directory path does not exist on your system, the installation
program will create the path for you.

-----------------------------------------------
Note:  Make sure all pathnames are absolute.
-----------------------------------------------

Hit <CR> to continue ...
Perl executable ("/usr/local/bin/perl")
-> /usr/bin/perl
Location to install programs ("/usr/local/bin")
-> /mnt/ehood/bin
Location to install libraries ("/usr/local/lib/MHonArc")
-> /mnt/ehood/lib/MHonArc
Install documentation ("y")?  y
Location to install docs ("/usr/local/lib/MHonArc/doc")
-> /mnt/ehood/lib/MHonArc/doc

You've specified the following:
        Perl location: /usr/bin/perl
        Program directory: /mnt/ehood/bin
        Library directory: /mnt/ehood/lib/MHonArc
        Doc directory: /mnt/ehood/lib/MHonArc/doc
Is this correct ("y")?  y
Installing the following into /mnt/ehood/bin
        mhonarc
Installing the following into /mnt/ehood/lib/MHonArc
        base64.pl
        mhexternal.pl
        mhtxthtml.pl
        mhtxtplain.pl
        mhtxtsetext.pl
        qprint.pl
        readmail.pl
Installing the following into /mnt/ehood/lib/MHonArc/doc
        mhonarc.txt
        ...
```

Batch Mode

To run `install.me` in batch mode, type the following at your shell’s prompt:

```
perl install.me install.cfg
```

NOTE

Make sure you are in the same directory as the `install.me` program.

The `install.cfg` contains the necessary information for intalling *MHonArc* on your system. You will need to edit `install.cfg` to reflect your installation requirements.

Here is an example `install.cfg`:

```
# Should executables be installed. 0 => NO, non-zero => YES.
#
$dobin = 1;

# Should libraries be installed. 0 => NO, non-zero => YES.
#
$dolib = 1;

# Should documentation be installed. 0 => NO, non-zero => YES.
#
$dodoc = 1;

# Location for executable.  If using ms-dos, use something like
# 'C:\\BIN'.
#
$bindir  = '/usr/local/bin';

# Location for libraries.  If using ms-dos, use something like
# 'C:\\LIB\\MHONARC'.
#
$libdir  = '/usr/local/lib/MHonArc';

# Location for documents.  If using ms-dos, use something like
# 'C:\\DOC\\MHONARC'.
#
$docdir  = '/usr/local/lib/MHonArc/doc';

# Location of perl executable.  If using ms-dos, use something like
# 'C:\\BIN\\PERL.EXE'.
#
$perlprg = '/usr/local/bin/perl';


1; # DO NOT DELETE THIS LINE
```

The file is Perl code, and therefore, must follow Perl syntax rules:

- Anything following a '`#`' character is ignored.
- Strings values need to be enclosed in quotes.
- If you need to use a backslash in a string value, it must be escaped with a backslash. Example: `'C:\\LIB\\MHONARC'`. The same applies to the '`$`' character.
- All statements must end with a semi-colon.
- The "`1;`" line must not be deleted.

NOTE

You can verify the syntax of the configuration file by invoking "`perl -c`" on the file.

After you have successfully executed `install.me`, *MHonArc* is ready to use.

MS-DOS Post `install.me` Note

If you would like the ability to run *MHonArc* like other programs, then create a batch file that contains something like the following:

```
@ECHO OFF
C:\BIN\PERL.EXE C:\BIN\MHONARC %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Of course, you will need to change the paths to Perl and *MHonArc* to suit your systems configuration.

Sample batch files are available in the *MHonArc* distribution.

Notes on `install.me`

- If you do not know the location of the Perl executable on your system, ask your system administrator.
- All pathnames must be absolute.
- If a path does not exist that you specify, the path will be automatically created if running in interactive mode. In batch mode, all paths specified must already exist.
- During the installation process, the main *MHonArc* source file is modified to be aware of the location of the Perl executable and *MHonArc*'s library files. If you ever need to install *MHonArc* in a different location, rerun the `install.me` program.

  NOTE: Location of the Perl exectuble is only relevant for Unix systems. MS-DOS systems do not make use of the "`#!`" line in scripts.
- *MHonArc* requires the use of `timelocal.pl` and `newgetopt.pl`. These libraries are part of the normal Perl distribution.

Tested Environments

This section covers software environments *MHonArc* has worked successfully. Feedback is welcome about other success, or failure, stories covering *MHonArc* usage in other environments.

Perl

MHonArc is known to work with the following version of Perl 4, or later:

```
$RCSfile: perl.c,v $$Revision: 4.0.1.7 $$Date: 92/06/08
14:50:39 $
Patch level: 34
```

MHonArc is known also to work with Perl 5.001m and Perl 5.002 beta2.

NOTE

The version numbers are based upon the Unix versions of Perl. DOS version numbers may differ.

Unix

Mail Software

- *MH*
- *Elm*, *Mail*, *mail*, and any other mail software that stores e-mail in UUCP style mailbox format. UUCP format is where mail messages are separated by a line beginning with "`From `" (I.e. the word "From" followed by a space). You may need to utilize the `MSGSEP` resource if the message separator is different from standard mailbox files (eg. MMDF format).

News Software

Different news software store messages differently. Messages are either stored in a format similiar to *MH* or similiar to a mailbox file:

- *MH* style is where the messages are stored in a directory with each post a separate file, and each file has a numeric filename.
- Mailbox style is where messages are stored in a single file. You may need to utilize the `MSGSEP` resource if the message separator is different from standard mailbox files.

MS-DOS

Mail/News Software

MHonArc has been tested under MS-DOS with message files created by the following mail and news programs:

- *Eudora*
- *WinVN*
- *Windows Trumpet*
- *NUPop*

It also works with individual RFC822 mail messages, but you must run *MHonArc* without a batch file if you need to use redirection. For example:

```
perl c:\bin\mhonarc <one.msg >one.htm
perl c:\bin\mhonarc -add <one.msg
```

Quick Start

This section will give you a "*quick start*" on using *MHonArc*. However, I recommend reading through the entire documentation to take full advantage of all the features of *MHonArc*.

Before continuing, make sure *MHonArc* has been installed. See Installation if *MHonArc* has not been installed on your machine.

Converting MH mail folders or Mailbox files

Since *MHonArc* supports *MH* mail folders and UUCP/Unix mailbox files, the term "*mail folder*" will represent the *MH* mail folder or mailbox file you want to process. To convert your mail folder to an HTML archive, use the following:

`% mhonarc` <*path*>`/inbox`

Where <*path*> represents the path to the directory that contains the mail folder *inbox*. If you are in the directory that contains *inbox*, then you can leave out the "<*path*>/".

MHonArc prints out messages showing its progress as your e-mail is processed. When *MHonArc* finishes, the following files will be created:

- `maillist.html`: The main index file containing links to all mail messages converted. Messages are listed with subjects and who the messages are from. All messages are listed in sorted order by date received/sent.
- `threads.html`: The file listing messages by thread.
- `msg*.html`: HTML versions of the mail messages, where `*` represents a message number from 0 to the number of message processed minus 1.
- `.mhonarc.db` (or `MHONARC.DB` under MS-DOS): This database file is needed inorder for *MHonArc* to perform additions of new mail messages to the archive. Information is stored to perform mail threading updates when new messages are added, as well as any resources set via Environment variables, Resource File, and/or command-line Options.
- *Other*: Depending on the content-types of the e-mail messages converted, other files may be created for images, videos, binaries, etc. See the section on MIME for more information.

The format of each converted mail message is as follows:

- A `<LINK REV="made" HREF="mailto:`*`from_address`*`">` is inserted in the `HEAD` element of HTML mail message file. This allows readers of the message to send comments to the author of the mail message within Web browsers that support such functionality (like *Lynx*).
- The title (i.e. `TITLE` element) contains the subject of the message.
- Hyperlinks to the previous and next messages and the index pages are located at the top of the message.

- Next, the subject appears in a `H1` element.
- Next, follows the mail header with fields listed in a `UL` element surrounded by `HR`'s.
- Next, the actual body of the mail message.
- Next, links to any follow-up, or referenced, messages. The messages are listed by subject and who they are from. These links allows you to easily follow mail threads.
- Last, are verbose links to the previous mail message, next mail message, and index pages.

The following is also done for each mail message processed:

- Links are created in the "`References`" and "`In-Reply-To`" header fields, and possibly the message body, if the destination message-ids are being processed.
- E-mail addresses are converted to "`mailto`" hyperlinks in "`To:`", "`From:`", "`Cc:`", and "`Sender:`" mail header fields. Currently, not all Web browsers support the `mailto` URL.
- Newsgroups listed in a "`Newsgroups:`" mail header field are converted to `news` hyperlinks.

MHonArc allows you to specify more than one mail folder to process on the command-line.

Example

```
% mhonarc /home/ehood/mail/inbox1
/home/ehood/mail/inbox2 ...
```

All the files created will be put into the current working directory, by default. You can control the destination of the output location by using the `-outdir` option.

Example

```
% mhonarc -outdir /home/ehood/htmlarchive
/home/ehood/mail/inbox
```

Here is a sample session converting a mail folder:

```
% mhonarc ~/mail/inbox
Requiring MIME filter libraries ...
        mhexternal.pl
        mhtxthtml.pl
        mhtxtplain.pl
        mhtxtsetext.pl
Converting messages to ./maillist.html
Reading /mnt/ehood/mail/inbox ..........
Writing mail ...
Writing tmp/maillist.html ...
Writing tmp/threads.html ...
10 messages
```

Adding Messages to an Archive

If you have new messages you want to add to an existing archive, you must utilizing the `-add` command-line option. With the `-add`, you can do the following:

- Add a mail folder to an archive, or
- Add a single message to an archive.

Adding a mail folder to an archive in the current working directory can be done like the following:

`% mhonarc -add <`*path*`>/mailfolder`

If you are not in the same directory as the archive, then you can specify the location of the archive to add to with the `-outdir` option:

`% mhonarc -add -outdir <`*outdir_path*`> <`*path*`>/mailfolder`

NOTE

> *MHonArc* will skip any messages that already exist in the archive. Therefore, *MHonArc* can be used to rescan the same mail folder and only convert any new messages it finds.

If no mail folder arguments are specified, then *MHonArc* will attempt to add a *single* message read in from standard input.

Example

```
% mhonarc -add < single.msg
```

Or, from a pipe:

```
% cat single.msg | mhonarc -add
```

See the section on Adding Messages for more information and examples for adding messages to an archive.

Converting a single message

MHonArc has the ability to process a single mail message independent of creating, or modifying, and archive. To convert a single message to HTML use the `-single` command-line option. The message to process can be specified by a filename on the command-line, or by reading the message from standard input if no file is specified. The filtered message is sent to standard output. All formatting options apply to the single message as with messages being processed for an archive, with the exception of formatting related specificly to archive processing, like index links and mail thread links.

Examples

Input from standard input:

```
% mhonarc -single < messagefile > file.html
```

Filename on command-line:

```
% mhonarc -single messagefile > file.html
```

Overview

This section gives an overview of *MHonArc*'s command-line options and environment variables. The *MHonArc* resource file is covered in the section Resource File. The resource file allows you to specify most of the resources set by environment variables and command-line options, plus it give you the capability of completely customizing the HTML output generated by *MHonArc*.

Synopsis

Invoke *MHonArc* from your shell with the following syntax:

`% mhonarc` [*options*] *mhfolder*...
`% mhonarc` [*options*] *mailbox* ...
`% mhonarc -add` [*options*] `<` *message*
`% mhonarc -single` [*options*] `<` *message* `>` *message.html*
`% mhonarc -single` [*options*] *message* `>` *message.html*
`% mhonarc -rmm` [*options*] *msg#* ...

Options

The following options are available:

`-add`

Add new messages to an existing archive. If no mail folder arguments are given, *MHonArc* assumes that a **single** message is being added to the archive via *standard input*. Otherwise, *MHonArc* adds the messages contained in the mail folders specified.

`-dbfile` ***name***

Use *name* for the name of *MHonArc* database file. The default is "`.mhonarc.db`" (or "`MHONARC.DB`" under MS-DOS).

NOTE

You should not override the default name unless absolutely necessary, and you are confident about what you are doing.

`-docurl` ***url***

Use *url* as the URL to *MHonArc* documentation. The default is "`http://www.oac.uci.edu/indiv/ehood/mhonarc.html`".

-editidx

This option tells *MHonArc* to rewrite the index page and re-edit all mail messages in the archive. This option is useful if you need to change the layout of the index page and/or messages.

-footer *filename*

Insert contents of *filename* at the bottom of the index page. See Include Files in Index Page Customization for more information about the footer file.

-force

Override a lock on an archive if attempts to lock fail. I.e. After trying unsuccessfully to lock an archive, *MHonArc* will still perform the actions requested.

This option is useful to help dealing with locks that are no longer valid (i.e. *stale locks*). A stale lock can exist if the *MHonArc* process that created the lock abnormally terminated and could not perform the proper cleanup procedures.

-genidx

Output an index page to standard output based upon the contents of an archive, and utilizing any extra formatting options specified.

-header *filename*

Insert contents of *filename* at the beginning of the index page. See Include Files in Index Page Customization for more information about the header file.

-help

Print out a help message about *MHonArc*.

-idxfname *name*

Sets the name of the main index file to *name*. The default is "`maillist.html`".

-idxsize *#*

Set the maximum number of messages listed in the indexes.

-lockdelay *#*

The sleep time, in seconds, between attempts to lock the archive. The default value is 3.

-locktries *#*

Set the number of time *MHonArc* tries to lock a mail archive before processing new messages. The default value is `10`. *MHonArc* waits approximately 3 seconds before each try.

See Archive Integrity for more information on the `-locktries` options.

`-mailtourl` ***url***

Use *url* for e-mail address hyperlinks in mail message headers. The *url* can contain the following *variables* that get expanded during run-time:

`$FROM$`

Who the message is from.

`$MSGID$`

Message ID of the message.

`$SUBJECT$`

The subject of the message.

`$TO$`

Destination e-mail address of link.

The default URL is “`mailto:$TO$`“

The `-mailtourl` option has no effect if the `-nomailto` option is specified.

`-maxsize` ***#***

Set the maxinum number of messages allowed in the archive to #. If messages are added to the archive which would cause the total number of messages to exceed #, older messages (based on sort method) are removed automatically.

`-msgsep` ***expression***

Use the *expression* as the Perl regular expression that signifies the message separator in Unix mailbox files. The default expression is “`^From `“ (minus the quotes).

NOTE

There is a space character after the `From`.

`-nodoc`

Do not print link to documentation at end of index page.

`-nomailto`

Do not convert e-mail addresses in mail headers to `mailto` hyperlinks.

`-nonews`

Do not convert newsgroups in the `Newsgroups:` mail header field to `news` hyperlinks.

-noreverse

Do not perform a reverse listing of the mail messages in the index page.

-nosort

Do not sort messages by date. Messages will be in the order they appear in the mailboxes/folders. By default, *MHonArc* sorts messages by date sent/received.

`-nosort` takes precedence over the `-sort` option.

-nothread

Do not create a thread index page.

-notsubsort

Do not sort threads by subject on thread index page.

-notreverse

List threads in the thread index with oldest thread first.

-outdir *path*

Set destination/location of the HTML mail archive to *path*. By default, the current working directory is used.

-quiet

Suppress processing messages when *MHonArc* is running.

-rcfile *file*

Use *file* as the resource file for *MHonArc*. *MHonArc* does the following to determine the location of *file*:

1. If its an absolute pathname, use it.
2. If it is a relative pathname, check for it relative to the current working directory.
3. Otherwise, check for it relative to the location of the archive.

See Resource File for more information. There is no default resource file.

-reverse

List messages in reverse order of the sorting option specified. For example, if date sorting is specified, `-reverse` will cause messages to be listed in *reverse* chronological order.

-rmm

All non-option command-line arguments are treated as messages to remove from the archive. Messages to remove are denoted by their message numbers.

`-savemem`

Normally, all messages are stored in memory and then written in one shot. This option causes *MHonArc* to write the message data as it is processesd. This option will cause a slow down in execution time as more disk I/O required, but it may allow large amounts of data to be processed in a single process if memory is limited.

NOTE

The reason more disk I/O is required is that when the message data is first written, all the archive navigational link information is non-existant. The information required to correctly generate the navigational link information will not exist until all messages are processed. Therefore, each new message file must be reopened to add in the navigational link information after all messages are processed.

`-scan`

List contents of archive to standard output.

`-single`

Convert a single mail message to HTML. The message can be specified by a filename on the command-line, or read from standard input if no file is given. The filtered message is sent to standard output.

The `-single` option is useful tp convert individual messages to HTML not related to a specific mail archive. Any option related to how message formatting can be used with the `-single` option.

The `-single` takes precedence over the `-add` option.

`-sort`

Perform chronological date sorting. This is the default.

`-subsort`

Sort messages by subject. Subject sorting is case-insensitive, and begining "`Re:`", "`A`", "`An`", and "`The`" words are ignored.

`-tidxfname` ***name***

Sets the name of the thread index file to *name*. The default is "`threads.html`".

`-time`

Print out total CPU execution time taken when processing messages. Time information is written to standard error.

`-title` ***string***

Set the title of the main index page to *string*. The default is "*Mail Index*".

-thread

Create a thread index page. This is the default.

-tlevels

Set the maximum number of nested lists for the thread index page. The default is 3.

-treverse

List threads in the thread index with newest thread first.

-tsubsort

List threads in the thread index by subject.

-ttitle *string*

Set the title of the thread index page to *string*. The default is "*Mail Thread Index*".

-umask *umask*

Set the umask of the *MHonArc* process to *umask*. The value is treated as an octal number.

NOTE

The `-no*` options always take precedence over their counterparts. For example, if `-noreverse` and `-reverse` are both specified on the command-line, the `-noreverse` will be applied.

Environment

MHonArc supports the use of environment variables. The environment variables allow you to set default options everytime you invoke *MHonArc*. The following environment variables may be used:

M2H_DBFILE

Sets the name of *MHonArc* database file. The default is "`.mhonarc.db`" (or "`MHONARC.DB`" under MS-DOS).

NOTE

You should not override the default name unless absolutely necessary, and you are confident about what you are doing.

M2H_DOCURL

Set the URL used to point to *MHonArc* documentation. The default is, "`http://www.oac.uci.edu/indiv/ehood/mhonarc.html`".

M2H_FOOTER

Set the HTML footer file to insert at the bottom of the index page. No default footer file is defined. See Include Files in Index Page Customization for more information about the footer file.

M2H_HEADER

Set the HTML header file to insert at the top of the index page. No default header file is defined. See Include Files in Index Page Customization for more information about the header file.

M2H_IDXFNAME

Set the name of the index file. The default is, "`maillist.html`".

M2H_IDXSIZE

Sets the maximum number of messages listed in the indexes.

M2H_LOCKFILE

The sleep time, in seconds, between attempts to lock the archive. The default value is 3.

M2H_LOCKFILE

Set the name of the lock file. The default name use is "`.mhonarc.lck`" (or "`MHONARC.LCK`" under MS-DOS).

NOTE

You should not change the default unless absolutely necessary.

See Archive Integrity for more information about the lock file.

M2H_LOCKTRIES

Set the number of time *MHonArc* tries to lock a mail archive before processing new messages. The default value is `10`. *MHonArc* waits approximately 3 seconds before each try.

See Archive Integrity for more information on the `M2H_LOCKTRIES` environment variable.

M2H_MAILTOURL

Sets the URL for e-mail address hyperlinks in mail message headers. The URL can contain the following *variables* that get expanded during run-time:

`$FROM$`

Who the message is from.

`$MSGID$`

Message ID of the message.

`$SUBJECT$`

The subject of the message.

`$TO$`

Destination e-mail address of link.

The default URL is "`mailto:$TO$`"

M2H_MAXSIZE

Sets the maximum number of messages that an archive will contain. If messages are added to the archive which would cause the total number of messages to exceed `M2H_MAXSIZE`, older messages (based on sort method) are removed automatically.

M2H_OUTDIR

Sets the destination/location of the HTML mail archive. The default is the current working directory.

M2H_RCFILE

Specifies the Resource File for *MHonArc*. No default resource file is defined.

M2H_THREAD

Flag to determine if *MHonArc* generates a thread index. If set to zero, the thread index will not be created. The default behavior is to create a thread index.

M2H_TIDXFNAME

Sets the name of the thread index file. The default is "`threads.html`".

M2H_TITLE

Sets the default title of the index page. The default is "*Main Index*".

M2H_TLEVELS

Sets the maximum number of nested lists for the thread index page. The default is 3.

M2H_TTITLE

Sets the title of the thread index page. The default is "*Mail Thread Index*".

NOTE

Environment variables may be overriden by the Resource File or command-line Options.

Resource File

MHonArc supports the ability to read in a resource file to control the behavior of *MHonArc*. The resource file allows you to specify most of the resources set by environment variables and command-line options, and it allows you to specify other resources to control *MHonArc*'s behavior.

The resource file is specified by the `M2H_RCFILE` environment variable or the `-rcfile` command-line option. The command-line option overrides the environment variable if both are defined.

NOTE

MHonArc will store the information specified in the resource file in the database for the archive. Therefore, it is unnecessary to respecify the resource file duing archive updates unless changes are required from the current settings.

Resource Syntax

Resources are set in the file by using *elements* similiar in style to HTML/SGML markup. However, *MHonArc* uses simpler parsing rules for the resource file than standard SGML:

- Any line that is not a recognized element open tag, *and* the line is not contained within an element, is ignored. This implies that regular text can be put anywhere **outside** of recognized elements for commenting purposes.

  NOTE: You should use SGML comment declarations (`<!--` ... `-->`) when commenting a resource file. This will eliminate possible conflict with later versions of *MHonArc* if more stricter parsing rules are adopted.
- The opening tag of an element must occur by itself on a single line. Whitespace is allowed before the the open tag.
- No comments are allowed inside elements because the text will be treated as element content.
- Each element must be closed with a `</element_name>` tag on its own line unless *explicitly* stated otherwise in the Resource Elements section. Whitespace is allowed before the close tag.
- Some elements can take an optional attribute called "`Override`". This tells *MHonArc* that the contents of the element will completely override the default behavior of *MHonArc*, and previous instances of the element. **Example**: "`<EXCS Override>`". If "`Override`" is not specified, then the contents of the element augment the current setting.
- Element names are case-insensitive.
- Elements can occur in any order in the resource file.

Resource Variables

Some resource element contents may contain variables. Variables get expanded to strings at run-time.

NOTE

Variable expansion will only take place in resource elements that are intended to have variables as part of their content. If an element is not meant to have variables, the variable text will be taken literally as part of the element content.

The syntax of the *variables* to use in resource elements is as follows:

`$`*VARIABLE*[**:**[*N*][`U`]]`$`

The items in []'s are optional. Definition of each part:

`$`

The `$` character represents the beginning, and ending, of the variable.

VARIABLE

This is the the actual name of the variable. All variable names must be uppercase.

:[*N*][`U`] (*optional*)

This defines a maximum length of the replacement string for the variable. The option "`U`" denotes that the replacement string should be treated as part of a URL string. This can be useful when the variable may contain special characters, and the variable is used withing a URL.

No whitespace is allowed between the opening `$` and closing `$`. If an unrecognized variable is encountered, it gets replaced with an empty string. If a literal "`$`" is needed, use "`$$`".

SPECIAL NOTE

The `MAILTOURL` resource has different rules for variable expansion. If a variable does not exactly match the set of variables available for the `MAILTOURL`, the variable text will be taken literally as part of the element content. Therefore, a single "`$`" can be used to represent a "`$`" character.

Also, variables in the `MAILTOURL` should **NOT** have "**:***NU*" modifier. This will prevent the variables from be recognized. *MHonArc* will automatically treat the replacement value as a part of a URL string.

Here are some examples of legal variable usage:

- `$SUBJECT$`
- `$FROMNAME$`
- `$SUBJECT:50$`
- `$SUBJECTNA:60U$`
- `$FROMADDR:U$`

Each resource element will define what variables are defined for it.

Resource Elements

The following are complete listings of all the resource elements defined by *MHonArc*. Many element descriptions will reference to other sections of the documentation on the exact usage of the element.

Empty Elements

The following list of elements contain no textual content. Therefore, no end tag is required:

`NODOC`

Do not put link to documentation on main index page.

`NOMAILTO`

Do not convert e-mail addresses in mail headers to mailto hyperlinks.

`NONEWS`

Do not convert newsgroups in the `Newsgroups:` mail header field to `news` hyperlinks.

`NOREVERSE`

Do not perform a reverse listing of the mail messages in the main index page.

`NOSORT`

List messages in the index page in the order they are processed.

`NOTHREAD`

Do not create thread index.

`NOTREVERSE`

List threads in the thread index with oldest thread first.

`NOTSUBSORT`

Do not sort thread by subject in thread index page.

`REVERSE`

List messages in reverse listing order for the main index page.

`SORT`

List messages in the index page in chronological order.

SUBSORT

Sort messages by subject. Subject sorting is case-insensitive, and begining "`Re:`", "`A`", "`An`", and "`The`" words are ignored.

THREAD

Create thread index. This is the default.

TREVERSE

List threads in the thread index with newest thread first.

TSUBSORT

Sort thread by subject in thread index page.

Non-Empty Elements

The following list of elements contain textual content, therefore, each element must be **explicitly** closed with an element end tag (examples are given in Example Resource File):

BOTLINKS

Markup for defining the various hyperlinks at the bottom of converted messages. See Navigational Links of Message Customization for usage of this element.

DBFILE

Use *name* for the name of *MHonArc* database file. The default is "`.mhonarc.db`" (or "`MHONARC.DB`" under MS-DOS).

NOTE

You should not override the default name unless absolutely necessary, and you are confident about what you are doing.

DOCURL

URL to *MHonArc* documentation. The default is "`http://www.oac.uci.edu/indiv/ehood/mhonarc.html`".

EXCS

Set of message header fields to exclude from messages. See Excluding Fields of Message Customization for usage of this element.

FIELDORDER

The order the message header fields appear in messages. See Field Order of Message Customization for usage of this element.

FIELDSTYLES

The format specification for message header field values. See Field Formatting of Message Customization for usage of this element.

FOOTER

File to include at the end of the index page. See Include Files in Index Page Customization for more information about the footer file.

HEADER

File to include at the beginning of the index page. See Include Files in Index Page Customization for more information about the header file.

ICONS

The `ICONS` element is used to specify the icons that represent the different content-types of messages. See Icons in Index Page Customization for usage of this element.

IDXFNAME

The name of the index file. The default is "`maillist.html`".

LABELSTYLES

The format specification for message header field labels. See Field Formatting in Message Customization for usage of this element.

LISTBEGIN

Markup for beginning the main index list. See Listing Layout in Index Page Customization for usage of this element.

LITEMPLATE

Markup for an entry in the main index list. See Listing Layout in Index Page Customization for usage of this element.

LISTEND

Markup for terminating the main index list. See Listing Layout in Index Page Customization for usage of this element.

MAILTOURL

Url to use for e-mail hyperlinks. See E-mail Links in Message Customization for usage of this element.

MIMEARGS

Arguments to MIME filters. See Specifying Filters in MIME for usage of this element.

MIMEFILTERS

Routines for filtering messages. See Specifying Filters in MIME for usage of this element.

MSGFOOT

Footer text for converted messages. See Header and Footer in Message Customization for usage of this element.

MSGHEAD

Header text for converted messages. See Header and Footer in Message Customization for usage of this element.

MSGSEP

Perl regular expression that represents the message separator for mailbox files. The default expression is "`^From `".

OTHERINDEXES

List of resource files (one per line) defining other index pages to generate when creating, or updating, an archive.

CAUTION

It is very important that each resource file specified defines the `IDXFNAME` (or the `TIDXFNAME` **and** `THREAD` elements for a thread index) to prevent overwriting of the default index pages.

MHonArc will only store the name of the resource files listed in the database. Therefore, for any subsequent updates the archive, the extra index resource files must exist inorder to generate the extra index pages.

NOTE

Since *MHonArc* will look in the archive location for resource files specified with relative pathnames, you can keep the other index resource files in the same location as the archive, and just specify the filenames for the `OTHERINDEXES` element in the main resource file.

When create resource files for extra indexes, make sure to explicitly set all resources desired since some resource settings may no longer be set to the defaults due to database settings, or from a previously read resource file. Ie. *MHonArc* does not reset to the default settings when reading in the other resource files.

PERLINC

Each line represents a path to search when requiring MIME filters. See Specifying Filters in MIME for the use of this element.

TFOOT

Markup that appears after the thread index listing. See Listing Layout in Thread Index Customization for usage of this element.

THEAD

Markup that appears before the thread index listing. See Listing Layout in Thread Index Customization for usage of this element.

TIDXFNAME

The name of the thread index file. The default is "`threads.html`".

TIMEZONES

Each line of the `TIMEZONES` element defines a timezone acronym and its hour offset from UTC/GMT (*Universal Coordinate Time*). The format of each line is "*`timezone_acronym`*`:`*`hour_offset`*". Examples of timezone acronyms are: `UTC`, `PDT`, `EST`. The hour offset is should be positive for timezones West of UTC, and negative for time zones East of UTC.

MHonArc has a default list of timezone acronyms defined with hour offsets. Therefore, the list given in the resource file will augment the default list, unless the "`Override`" attribute is specified. If "`Override`" is specified, the default list, along with any other lists specified in previous `TIMEZONES` elements, are discarded, and only the timezone acronyms specified in the `TIMEZONES` element will be used.

The following is the default value for `TIMEZONES`:

```
<TIMEZONES>
UTC:0
GMT:0
AST:4
ADT:3
EST:5
EDT:4
CST:6
CDT:5
MST:7
MDT:6
PST:8
PDT:7
</TIMEZONES>
```

Most of the time, the date used by *MHonArc* uses a hour offset instead of a timezone acronym. However, mail messages may contain timezone acronyms in received/sent dates and *MHonArc* must be told what the hour offset from UTC the timezone acronym represents in order to properly sort messages by date.

TITLE

Title for the main index page. The default is "*Mail Index*".

TLEVELS

The maximum number of nested lists for the thread index. The default is 3.

TLITXT

Markup for an entry in the thread index list. See Listing Layout in Thread Index Customization for usage of this element.

TOPLINKS

Markup for defining the various hyperlinks at the top of converted messages. See Navigational Links in Message Customization for usage of this element.

TTITLE

Title for the thread index page. The default is "*Mail Thread Index*".

UMASK

Sets the umask for the *MHonArc* process. The value is treated as an octal number. The resource is only applicable on Unix systems.

Example Resource File

```
<!--      MHonArc resource file
  -->
<SORT>
<TITLE>
MHonArc test
</TITLE>

<TTITLE>
MHonArc test
</TTITLE>

<!--=== Index Page Customizations =========================================-->

<!--      Have LISTBEGIN contain last updated information
  -->
<LISTBEGIN>
<address>
Last updated: $LOCALDATE$<br>
$NUMOFMSG$ messages in chronological order<br>
</address>
<ul>
<li><a href="$TIDXFNAME$">Thread Index</a></li>
</ul>
<p>
Listing format is the following:
<p>
<ul><li>
<strong>Subject</strong>
(# of follow-ups)
<em>From</em><br>
```

```
</ul>
<p>
<hr>
<ul>
</LISTBEGIN>


<!--      A compact listing template
  -->
<LITEMPLATE>
<li>
<strong>$SUBJECT:40$</strong>
($NUMFOLUP$) <em>$FROMNAME$</em><br>
</LITEMPLATE>

<LISTEND>
</ul>
<p>
<hr>
<strong>
<a href="http://foo.org/">Home</a>
</strong>
<p>
</LISTEND>

<!--=== Thread Index Page Customizations ==================================-->
<THEAD>
<address>
Thread index<br>
Last updated: $LOCALDATE$<br>
$NUMOFMSG$ messages<br>
</address>
<ul>
<li><a href="$IDXFNAME$">Main Index</a></li>
</ul>
<hr>
</THEAD>

<!--=== Message Customizations ============================================-->
<EXCS override>
apparently
errors-to
followup
forward
lines
message-id
mime-
nntp-
originator
path
precedence
received
replied
return-path
status
via
x-
```

```
</EXCS>

<LABELSTYLES>
-default-
subject:strong
from:strong
to:strong
</LABELSTYLES>

<FIELDSTYLES>
-default-
subject:strong
from:strong
to:strong
keywords:em
newsgroups:strong
</FIELDSTYLES>

<MSGHEAD>
<address>
MHonArc test archive
</address>
</MSGHEAD>

<MSGFOOT>
<strong>
<a href="http://foo.org/">Home</a> |
<a href="$IDXFNAME$">Main Index</a> |
<a href="$TIDXFNAME$">Thread Index</a>
</strong>
</MSGFOOT>

<!--=== Icons ==============================================================-->
<ICONS>
application/octet-stream:http://foo.org/icons/binary.xbm
application/postscript:http://foo.org/icons/postscript.xbm
audio/basic:http://foo.org/icons/sound.xbm
image/gif:http://foo.org/icons/image.xbm
image/jpeg:http://foo.org/icons/image.xbm
image/tiff:http://foo.org/icons/image.xbm
multipart/alternative:http://foo.org/icons/alternative.xbm
multipart/digest:http://foo.org/icons/text.xbm
multipart/mixed:http://foo.org/icons/mixed.xbm
multipart/parallel:http://foo.org/icons/mixed.xbm
text/richtext:http://foo.org/icons/mixed.xbm
text/html:http://foo.org/icons/mixed.xbm
text/plain:http://foo.org/icons/text.xbm
unknown:http://foo.org/icons/unknown.doc.xbm
video/mpeg:http://foo.org/icons/movie.xbm
video/quicktime:http://foo.org/icons/movie.xbm
</ICONS>
```

Notes on Resource File

- Elements can be duplicated. The following elements augment previous instances of themselves:
  - EXCS (can specify Override attribute)
  - FIELDSTYLES
  - ICONS (can specify Override attribute)
  - LABELSTYLES
  - MIMEFILTERS (can specify Override attribute)
  - PERLINC (can specify Override attribute)
  - TIMEZONES (can specify Override attribute)

  The Override attribute will discard previous settings of the element.
- If duplicate instances of other elements exist, the last instance takes precedence.
- If an element only accepts a single line of content, then the last line is used for the element's content.
- If elements have conflicting resource settings (eg. NOSORT and SORT), the last element defined takes precedence.
- Resource file settings override environment variables.
- Command-line options override any settings in the resource file.
- If you want to do an exact match of a field in the EXCS element, append a '$' after the field name.

Adding Messages

Adding messages to an archive is done via the `-add` option. If no mailbox/folder arguments are given, *MHonArc* assumes that a **single** message is being added to the archive via *standard input*. Otherwise, *MHonArc* adds the messages contained in the mail folders specified.

NOTE

MHonArc will skip any messages that already exist in an archive. If a message to be added has a message-ID that equals a message-ID of an archived message, the message is skipped.

Examples

Adding a mail folder

Here is example session adding an mail folder to an existing archive:

```
% mhonarc -add test/www
Requiring MIME filter libraries ...
        mhexternal.pl
        mhtxthtml.pl
        mhtxtplain.pl
        mhtxtsetext.pl
Adding messages to ./maillist.html
Reading test/www/ ......................................
Writing HTML ...
49 messages
```

.forward

MHonArc can be used to add new messages as they are received by using the "`.forward`" file in your home directory. Here is how I would set up my `.forward` file to invoke *MHonArc* on incoming mail:

```
\ehood, "|/mnt/ehood/bin/webnewmail #ehood"
```

NOTE on `.forward` entry:

The "`\ehood`" tells `sendmail` to still deposit the incoming message to my mail spool file. The "`#ehood`" Bourne shell comment is needed to insure the command is unique from another user. Otherwise, `sendmail` may not invoke the program for you or the other user.

"`webnewmail`" is a Perl program that calls *MHonArc* with the appropriate arguments. A wrapper program is used instead of calling *MHonArc* directly to keep the `.forward` file simple. Here is the code to the `webnewmail` program:

```
#!/usr/local/bin/perl

$cmd = "/mnt/ehood/bin/mhonarc -add -quiet " .
          "-outdir /mnt/ehood/public_html/newmail";
open(M2H, "|$cmd");
print M2H <STDIN>;
close(M2H);
```

The `webnewmail` can be modified to check the mail header before calling *MHonArc* to perform selective archiving of messages. For example, `webnewmail` can check the `To:` field and only archive messages that come from a specific mailing list.

Cron

This example uses `cron`(1) to update some mail archives from MH mail folders.

The following entry is in my crontab file:

```
0 0 * * * webmail
```

`webmail` is a script executed every night that calls *MHonArc* to perform the update:

```
#! /bin/csh -f

umask 022
setenv M2H_RCFILE $HOME/.mhonarc.rc
## WWW messages
mhonarc -add \
          -outdir $HOME/public_html/doc/wwwmail \
          $HOME/mail/www
folder +www >& /dev/null
refile first-last +www.ar >& /dev/null    # Archive original messages

## Tools messages
mhonarc -add \
          -outdir $HOME/public_html/doc/toolsmail \
          $HOME/mail/tools $HOME/mail/dtd
folder +tools >& /dev/null
refile first-last +tools.ar >& /dev/null  # Archive original messages
folder +dtd >& /dev/null
refile first-last +dtd.ar >& /dev/null    # Archive original messages

folder +inbox >& /dev/null                # Set current folder to inbox
```

To avoid mail everynight from `cron` due to output from *MHonArc*, the `-quiet` option can be used for each call to *MHonArc*, or use the following line in your crontab file:

```
0 0 * * * webmail > /dev/null
```

Standard error is not redirected to `/dev/null` so mail is still received if errors occured during *MHonArc* execution.

Removing Messages

Removing messages from an archive is done via the -rmm option. Messages to be deleted are designated by message numbers on the command-line.

Example

```
% mhonarc -rmm 24 28 39 48
Removing messages from ./maillist.html ...
        Removing message 24
        Removing message 28
        Removing message 39
        Removing message 48

Writing mail ...
Writing tmp/maillist.html ...
Writing tmp/threads.html ...
45 messages
```

Message Numbers

Normally, you will never have to worry about message numbers unless you want to remove messages from an archive. Therefore, you will need to know how *MHonArc* assigns message numbers when processing messages.

When a message is processed, the smallest available number is assigned to it, starting with 0. The number assigned to a message becomes part of the filename for the HTML version of the message (eg. `msg00042.html`).

To avoid message number conflicts, *MHonArc* determines the smallest available number by finding the largest assigned number and adding one to it.

Scanning an Archive

You will quickly find out that finding the message numbers for a messages you want to remove can be a cumbersome task if all you have to work with are the message filenames. To ease this task, *MHonArc* gives you the ability to scan an archives contents via the -scan command-line option.

Example

```
% mhonarc -scan
100 messages in .:

Msg #  YY/MM/DD  From             Subject
-----  --------  ---------------  ---------------------------------------------
  513  95/02/09  Rick Silterra    EDComment(sic)
  517  95/02/09  Earl Hood        Re: DTD2HTML
  512  95/02/09  Earl Hood        Re: edc2html
  516  95/02/09  John Barnum      Re: DTD2HTML
  515  95/02/09  Earl Hood        Re: DTD2HTML
```

```
  511  95/02/09  Rick Silterra    edc2html
  514  95/02/08  John Barnum      DTD2HTML
  510  95/02/06  jflores          mhonarc_diagnostics.doc.html
  509  95/02/06  web              Dr.Web: Status Review + Thank You
  508  95/02/05  Earl Hood        Re: sgml to html converters
  507  95/02/03  Aileen Barry     sgml to html converters
  506  95/01/28  Earl Hood        Re: MHonarc: Deleting Messages from an archiv
  505  95/01/28  Floyd Moore      MHonarc: Deleting Messages from an archive
  504  95/01/25  Earl Hood        Re: MHonArc
  503  95/01/25  Earl Hood        Re: MHonArc
...
```

The messages are listed in the same order as they are listed in the archive's index page. You will notice that the list order does not necessarily correspond with message number order. If you always want the messages listed in message number order when scanning, use the following:

```
% mhonarc -scan -nosort -noreverse
82 messages in .:

Msg #  YY/MM/DD  From              Subject
-----  --------  ---------------   ---------------------------------------------
    0  94/05/09  Michael O´Sulli  Re: Finger within an html
    1  94/04/31  John M. Troyer   Re: TROFF to HTML Converters
    2  94/05/04  John D. Kilburg  ANNOUNCE: Chimera 1.53
    3  94/05/17  Stephen Billing  Re: government www?
    4  94/05/21  C. Emory Tate    Re: government www?
    5  94/05/24  Daniel W. Conno  Re: Comments on HTML 2.0 document/DTD
    6  94/05/24  Dan Connolly     Re: Validating HTML documents:
    7  94/05/25  Henrik Frystyk   CERN Common World-Wide Web Library 2.16pre2 A
    8  94/06/04  Denesh Bhabuta   Re: Atari on www (revisited)
    9  94/06/07  Dale Newfield    ANNOUNCE: Come explore The Edge - SIGGRAPH 94
   10  94/06/11  Roy T. Fielding  Announcing libwww-perl 0.12
...
```

Index Page Customization

MHonArc creates an index page with links to all mail messages filtered (unless processing a single message with the `-single` option). *MHonArc* allows you to have complete customization over the appearance of the index page by setting various resource either through environment variables, command-line options, or the resource file.

Filename

By default, the filename of the index page is "`maillist.html`". However, a different name may be specified with the `M2H_IDXFNAME` environment variable, the `IDXFNAME` resource element, or the `-idxfname` command-line option.

Beginning Markup

MHonArc allows you to completely override the begining markup of the index page. I.e. You can control the opening <HTML> tag, the HEAD element contents, the opening <BODY> tag, etc. Therefore, if you are not satisfied with the default behavior of how the TITLE resource is used, or have other needs that require control on the beginning markup, you can set the `IDXPGBEGIN` resource file element.

IDXPGBEGIN

The best way to show how the `IDXPGBEGIN` works, the following represents the default setting *MHonArc* uses:

```
<IDXPGBEGIN>
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
<HEAD>
<TITLE>$IDXTITLE$</TITLE>
</HEAD>
<BODY>
<H1>$IDXTITLE$</H1>
</IDXPGBEGIN>
```

NOTE

Technically, setting the TITLE resource, via the `M2H_TITLE` environment variable, the `TITLE` resource element, or the `-title` command-line option, sets the `$IDXTITLE$` resource file variable.

The resource variables allowed in the `IDXPGBEGIN` element are the following:

- `$DOCURL$` -- URL to documentation
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.

- `$IDXSIZE$` -- Max number of messages that may be listed in the index.
- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.
- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.

See Resource Variables for more information on the usage of variables.

End Markup

Since *MHonArc* allows you to control the beginning markup, it makes sense for it to allow you to control the ending markup.

IDXPGEND

The `IDXPGEND` resource element may be used to define the ending markup of the index page. The default value is the following:

```
<IDXPGEND>
</BODY>
</HTML>
</IDXPGEND>
```

The resource variables allowed are the same as for `IDXPGBEGIN`.

Include Files

MHonArc allows you to include the contents of files into the index page via the header and footer resources.

NOTE

The use of include files is discouraged since the `LISTBEGIN` and `LISTEND` resources can be used to achieve the same results. Also, the support for the include resource may be removed in future releases.

The header file is specified via the `M2H_HEADER` environment variable, the `HEADER` resource element, or the `-header` command-line option. The contents of the header file are inserted above the message listing, and right after the `H1` title element.

NOTE

Filename should not contain the `<HTML>`, `<HEAD>`, and `<BODY>` tags; these tags are automatically provided by *MHonArc*, or defined by the `IDXPGBEGIN` resource file element.

The footer file is specified via the `M2H_FOOTER` environment variable, the `FOOTER` resource element, or the `-footer` command-line option. The contents of the footer file are inserted after the message listing.

NOTE

Filename should not contain the `</BODY>`, and `</HTML>` tags; these tags are automatically provided by *MHonArc*, or defined by the `IDXPGEND` resource file element.

The header and footer files allow you to incorporate search-forms, hyperlinks to other pages, or any other HTML markup you like.

It is only necessary to specify the header and/or footer files the first time you create an archive. The contents included from the header and/or footer files are preserved in any subsequent additions to the archive. Only respecify the header and/or footer files if you need to make changes to the header/footer contents.

Listing Layout

MHonArc lists messages in the order specified by the various sort options. However, you have complete control on how the message listing are formatted via the `LISTBEGIN`, `LITEMPLATE`, and `LISTEND` resource elements in the Resource File. These elements allow you to specify the HTML markup to use in the index page.

LISTBEGIN

The `LISTBEGIN` resource element specifies the text to begin the message list. The text can be any valid HTML markup. Plus, *MHonArc* defines the following variables you may use which get expanded at run-time:

- `$DOCURL$` -- URL to documentation
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.
- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.
- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.

- `$VERSION$` -- Program version.

MHonArc's `LISTBEGIN` default value is the following:

```
<LISTBEGIN>
<UL>
<LI><A HREF="$TIDXFNAME$">Thread Index</A></LI>
</UL>
<HR>
<UL>
</LISTBEGIN>
```

If the NOTHREAD resource is set, the following is the default value:

```
<LISTBEGIN>
<HR>
<UL>
</LISTBEGIN>
```

LITEMPLATE

The `LITEMPLATE` resoure element defines the HTML text to represent each message list item. You may use the following variables which are expanded at runtime:

- `$A_ATTR$` -- The `NAME` and `HREF` attributes to use in an anchor to link to the archived message. The `NAME` attribute links the messages to the index page.
- `$A_HREF$` -- The `HREF` attribute to use in an anchor to link to the archived message.
- `$A_NAME$` -- The `NAME` attributes to use in an anchor for messages to link to the index page.
- `$DATE$` -- The date of the message.
- `$DDMMYY$` -- Message date in dd/mm/yy format.
- `$ICON$` -- The context-type sensistive icon. See Icons for information.
- `$ICONURL$` -- The URL to the context-type sensistive icon. See Icons for information.
- `$MMDDYY$` -- Message date in mm/dd/yy format.
- `$NUMFOLUP$` -- Number of follow-ups for the given message.
- `$FROM$` -- The complete text in the `From:` field of the message.
- `$FROMADDR$` -- The e-mail address in the `From:` field of the message.
- `$FROMNAME$` -- The English name of the person in the `From:` field of the message. If no English name is found, the username specified in the e-mail address is used.
- `$MSGNUM$` -- The message numbers assigned to the message by *MHonArc*.
- `$ORDNUM$` -- The current listing number of the message.
- `$SUBJECT$` -- The subject text of the message wrapped in an anchor element that hyperlinks to the message.

- `$SUBJECTNA$` -- The subject text of the message without the anchor element.
- `$YYMMDD$` -- Message date in yy/mm/dd format.

NOTE

Do not specify `$A_ATTR$`, `$A_NAME`, and `$SUBJECT$` together in the `LITEMPLATE` element. Since all of these variables contain the `NAME` atrribute. Invalid HTML will be created since multiple anchors will have the same `NAME` identifier.

`LITEMPLATE`'s default value is the following:

```
<LITEMPLATE>
<LI><STRONG>$SUBJECT$</STRONG>
<UL><LI><EM>From</EM>: $FROM$</LI></UL>
</LI>
</LITEMPLATE>
```

`LISTEND`

The `LISTEND` resource element specifies the text to use to end the message list. The text can be any valid HTML markup. `LISTEND` may contain the same variables as `LISTBEGIN`.

`LISTEND`'s default value is the following:

```
<LISTEND>
</UL>
</LISTEND>
```

Icons

MHonArc supports the ability to insert icons in the index page for each message based on the message's content-type. For example: You can have text/plain messages use a different icon than text/html messages.

Defining Icons

To specify the icons for *MHonArc* to use, you use the `ICONS` resource element in the Resource File. The format of each line in the `ICONS` element is as follows:

<*content-type*>`:`<*URL for icon*>

<*content-type*> represents a MIME content-type. <*URL for icon*> is the URL to the icon. The special content-type called "`unknown`" may be defined to specify the icon to use for non-recognized content-types. If `unknown` is not defined, the text/plain icon is used for unknown content types.

Example

```
<ICONS>
audio/basic:http://foo.org/gifs/gsound.gif
image/gif:http://foo.org/gifs/gimage.gif
image/jpeg:http://foo.org/gifs/gimage.gif
image/tiff:http://foo.org/gifs/ggraphic.gif
multipart/alternative:http://foo.org/gifs/gmulti.gif
multipart/digest:http://foo.org/gifs/gtext.gif
multipart/mixed:http://foo.org/gifs/gdoc2.gif
multipart/parallel:http://foo.org/gifs/gdoc.gif
text/richtext:http://foo.org/gifs/gdoc.gif
text/html:http://foo.org/gifs/gdoc.gif
text/plain:http://foo.org/gifs/gletter.gif
unknown:http://foo.org/gifs/gunknown.gif
video/mpeg:http://foo.org/gifs/gmovie.gif
</ICONS>
```

Using Icons

In order to incorporate icons into the index page, insert the `$ICON$` variable into the `LITEMPLATE` resource element.

Example

```
<litemplate>
$ICONURL$<strong>$SUBJECT:40$</strong>
($NUMFOLUP$) <em>$FROMNAME$</em><br>
</litemplate>
```

The `$ICON$` variable expands to the `IMG` HTML element with the appropriate URL in the `SRC` attribute to the icon. The `ALT` attribute of the `IMG` element contains the content-type of the message, surrounded by `[]`'s, for use with text based browsers.

`$ICONURL$` may also be used if you want redefine the format of the `IMG` element.

Example

```
<litemplate>
<img src=”$ICONURL$” alt=”* “><strong>$SUBJECT:40$</strong>
($NUMFOLUP$) <em>$FROMNAME$</em><br>
</litemplate>
```

This example overrides what is normally used in the `ALT` attribute.

Examples

Example 1

It may be easier to see how the `LISTBEGIN`, `LITEMPLATE`, `LISTEND` resource elements work when declared together:

```
<!-- This represents the default values used by MHonArc -->
<LISTBEGIN>
<UL>
<LI><A HREF="$TIDXFNAME$">Thread Index</A></LI>
</UL>
<HR>
<UL>
</LISTBEGIN>

<LITEMPLATE>
<LI><STRONG>$SUBJECT$</STRONG>
<UL><LI><EM>From</EM>: $FROM$</LI></UL>
</LI>
</LITEMPLATE>

<LISTEND>
</UL>
</LISTEND>
```

Example 2

Here's another example that changes the layout into a more compact listing, adds Icons usage, and adds a time stamp information on when the index page was last updated:

```
<listbegin>
<address>
Last update: $CURDATE$<br>
$NUMOFMSG$ messages<br>
</address>
<p>
<UL>
<LI><A HREF="$TIDXFNAME$">Thread Index</A></LI>
</UL>
<p>
Messages listed in chronological order.  Listing format is the following:
<blockquote>
<img src="http://foo.org/gifs/gletter.gif" alt="* ">
<strong>Subject</strong>
(# of follow-ups)
<em>From</em>.
</blockquote>
<p>
<hr>
</listbegin>

<litemplate>
<img src="$ICONURL$" alt="* "><strong>$SUBJECT:40$</strong>
($NUMFOLUP$) <em>$FROMNAME$</em><br>
</litemplate>

<listend>
</listend>
```

Thread Index Customization

The thread index compliments the main index. If a thread index is created, each message in the archive will contain an extra link to the thread index.

Customizing the thread index is very similiar to customizing the main index.

Filename

By default, the filename of the index page is "`threads.html`". However, a different name may be specified with the `M2H_TIDXFNAME` environment variable, the `TIDXFNAME` resource element, or the `-tidxfname` command-line option.

Beginning Markup

MHonArc allows you to completely override the begining markup of the thread index page. I.e. You can control the opening <HTML> tag, the HEAD element contents, the opening <BODY> tag, etc. Therefore, if you are not satisfied with the default behavior of how the TTITLE resource is used, or have other needs that require control on the beginning markup, you can set the `TIDXPGBEGIN` resource file element.

TIDXPGBEGIN

The best way to show how the `TIDXPGBEGIN` works, the following represents the default setting *MHonArc* uses:

```
<TIDXPGBEGIN>
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
<HEAD>
<TITLE>$TIDXTITLE$</TITLE>
</HEAD>
<BODY>
<H1>$TIDXTITLE$</H1>
</TIDXPGBEGIN>
```

NOTE

Technically, setting the TTITLE resource, via the `M2H_TTITLE` environment variable, the `TTITLE` resource file element, or the `-ttitle` command-line option, sets the `$TIDXTITLE$` resource file variable.

The resource variables allowed in the `TIDXPGBEGIN` element are the following:

- `$DOCURL$` -- URL to documentation
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.

- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.
- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.

See Resource Variables for more information on the usage of variables.

End Markup

Along with controlling the beginning markup, you can control the ending markup.

TIDXPGEND

The `TIDXPGEND` resource element may be used to define the ending markup of the index page. The default value is the following:

```
<TIDXPGEND>
</BODY>
</HTML>
</TIDXPGEND>
```

The resource variables allowed are the same as for `TIDXPGBEGIN`.

Listing Layout

By default, *MHonArc* lists the threads in chronological order. The thread with the oldest inititial messsage is listed first, and the thread with the newest inititial message is listed last. The TREVERSE resource may be set to have the thread listing from newest thread to oldest thread. If the TSUBSORT resource is set, then threads are sorted by subject.

HTML markup can be added to the thread index page by defining the `THEAD` and `TFOOT` resource file elements. Usage of `THEAD` and `TFOOT` is discussed later in this section.

MHonArc uses HTML unorder lists for formatting the thread index list. Nested list markup is used to show the threads. Since threads can become very large, the TLEVELS resource can be used to limit the number of nested lists allowed. The default nesting level is 3.

Customization over the thread index list format is more restrictive than the main index. All thread index entries are contained in the HTML `LI` element, which is contained in an `UL` element. The entry text itself can be customized via the TLITXT resource file element. Usage of TLITXT is discussed later in this section.

THEAD

The `THEAD` resource element specifies text to be inserted before the thread index listing. The text can be any valid HTML markup. Plus, *MHonArc* defines the following variables you may use which get expanded at run-time:

- `$DOCURL$` -- URL to documentation
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.
- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.
- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.

MHonArc's `THEAD` default value is the following:

```
<THEAD>
<UL>
<LI><A HREF="$IDXFNAME$">Main Index</A></LI>
</UL>
<HR>
</THEAD>
```

TLITXT

The `TLITXT` specifies the markup to use for each entry in the thread index. Only markup that is legal within the HTML `LI` element should be used. When you define the contents for the `TLITXT` resource element, you may use the following *variables* defined by *MHonArc* (see Resource Variables on the usage of variables):

- `$A_ATTR$` -- The `NAME` and `HREF` attributes to use in an anchor to link to the archived message. The `NAME` attribute links the messages to the index page.
- `$A_HREF$` -- The `HREF` attribute to use in an anchor to link to the archived message.
- `$A_NAME$` -- The `NAME` attributes to use in an anchor for messages to link to the index page.

- `$DATE$` -- The date of the message.
- `$DDMMYY$` -- Message date in dd/mm/yy format.
- `$ICON$` -- The context-type sensistive icon. See Icons for information.
- `$ICONURL$` -- The URL to the context-type sensistive icon. See Icons for information.
- `$MMDDYY$` -- Message date in mm/dd/yy format.
- `$NUMFOLUP$` -- Number of follow-ups for the given message.
- `$FROM$` -- The complete text in the `From:` field of the message.
- `$FROMADDR$` -- The e-mail address in the `From:` field of the message.
- `$FROMNAME$` -- The English name of the person in the `From:` field of the message. If no English name is found, the username specified in the e-mail address is used.
- `$MSGNUM$` -- The message numbers assigned to the message by *MHonArc*.
- `$ORDNUM$` -- The current listing number of the message.
- `$SUBJECT$` -- The subject text of the message wrapped in an anchor element that hyperlinks to the message.
- `$SUBJECTNA$` -- The subject text of the message without the anchor element.
- `$YYMMDD$` -- Message date in yy/mm/dd format.

NOTE

Do not specify `$A_ATTR$`, `$A_NAME`, and `$SUBJECT$` together in the `LITEMPLATE` element. Since all of these variables contain the `NAME` atrribute. Invalid HTML will be created since multiple anchors will have the same `NAME` identifier.

`TLITXT`'s default value is the following:

```
<TLITXT>
($NUMFOLUP$) <STRONG>$SUBJECT:40$</STRONG>, <EM>$FROMNAME$</EM>
</TLITXT>
```

TFOOT

The `TFOOT` resource element specifies the text to use after the end the thread index list. The text can be any valid HTML markup. `TFOOT` may contain the same variables as `THEAD`.

`TFOOT`'s default value is nil.

Message Customization

This sections shows how to customize the appearance of messages when converted to HTML.

Beginning Markup

MHonArc allows you to completely override the begining markup of the message pages. I.e. You can control the opening `<HTML>` tag, the `HEAD` element contents, the opening `<BODY>` tag, etc. Therefore, if you are not satisfied with the default markup used, or have other needs that require control on the beginning markup, you can set the `MSGPGBEGIN` resource file element.

MSGPGBEGIN

The `MSGPGBEGIN` resource file element has the default value:

```
<MSGPGBEGIN>
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML>
<HEAD>
<TITLE>$SUBJECTNA:72$</TITLE>
<LINK REV=”made” HREF=”mailto:$FROMADDR$”>
</HEAD>
<BODY>
</MSGPGBEGIN>
```

The following variables may be used in the `MSGPGBEGIN` element:

- `$DATE$` -- Message date.
- `$DDMMYY$` -- Message date in dd/mm/yy format.
- `$DOCURL$` -- URL to documentation.
- `$FROM$` -- Contents of From field of message.
- `$FROMADDR$` -- E-mail address contained in From field of message.
- `$FROMNAME$` -- “English” name contained in From field of message.
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.
- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$MMDDYY$` -- Message date in mm/dd/yy format.
- `$MSGID$` -- Message ID of message.
- `$MSGNUM$` -- Number assigned to message by *MHonArc*.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.

- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$SUBJECTNA$` -- Message subject text.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.
- `$YYMMDD$` -- Message date in yy/mm/dd format.

End Markup

The ending markup of messages can be controlled by the MSGPGEND resource file element.

MSGPGEND

The MSGPGEND resource element may be used to define the ending markup of the message pages. The default value is the following:

```
<MSGPGEND>
</BODY>
</HTML>
</MSGPGEND>
```

The resource variables allowed are the same as for MSGPGBEGIN.

Header and Footer

The MSGHEAD resource represents HTML text that should be inserted at the very beginning of each converted message. The MSGFOOT resource represents HTML text that should be appended to the end of each converted message. The default value for both resources is empty. The following variables may be used in the MSGHEAD and MSGFOOT content:

- `$DATE$` -- Message date.
- `$DDMMYY$` -- Message date in dd/mm/yy format.
- `$DOCURL$` -- URL to documentation.
- `$FROM$` -- Contents of From field of message.
- `$FROMADDR$` -- E-mail address contained in From field of message.
- `$FROMNAME$` -- “English” name contained in From field of message.
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.
- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.

- `$MMDDYY$` -- Message date in mm/dd/yy format.
- `$MSGID$` -- Message ID of message.
- `$MSGNUM$` -- Number assigned to message by *MHonArc*.
- `$OUTDIR$` -- Pathname of archive.
- `$PROG$` -- Program name.
- `$SUBJECTNA$` -- Message subject text.
- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.
- `$YYMMDD$` -- Message date in yy/mm/dd format.

Navigational Links

MHonArc gives you the ability to control the layout of the navigational links for each message page. Navigational links include links to previous and next messages, link to main index, link to thread index, etc. The layout of the navigational links are controlled by two resource file elements: `TOPLINKS` and BOTLINKS.

TOPLINKS

The `TOPLINKS` resource element defines the layout of the navigational links at the top of each message page. The markup defined, will appear after the `MSGHEAD` data and before the filtered message data.

The default value for `TOPLINKS` is the following:

```
<TOPLINKS>
<HR>
$PREVBUTTON$$NEXTBUTTON$<A HREF="$IDXFNAME$#$MSGNUM$">[Index]</A><A
HREF="$TIDXFNAME$#$MSGNUM$">[Thread]</A>
</TOPLINKS>
```

If no thread index is specified, then the thread link markup is removed. The following variables are available:

- `$DATE$` -- Message date.
- `$DDMMYY$` -- Message date in dd/mm/yy format.
- `$DOCURL$` -- URL to documentation.
- `$FROM$` -- Contents of From field of message.
- `$FROMADDR$` -- E-mail address contained in From field of message.
- `$FROMNAME$` -- “English” name contained in From field of message.
- `$GMTDATE$` -- Current GMT date.
- `$IDXFNAME$` -- Filename of main index page.
- `$IDXSIZE$` -- Max number of messages that may be listed in the index.

- `$IDXTITLE$` -- The title of the index page.
- `$LOCALDATE$` -- Current local date.
- `$MMDDYY$` -- Message date in mm/dd/yy format.
- `$MSGID$` -- Message ID of message.
- `$MSGNUM$` -- Number assigned to message by *MHonArc*.
- `$NEXTBUTTON$` -- Next button markup. See Conditional Links for more information.
- `$NEXTFROM$` -- Contenst of From field of the next message according to the list order of the main index.
- `$NEXTFROMADDR$` -- E-mail address contained in From field of the next message according to the list order of the main index.
- `$NEXTFROMNAME$` -- English" name contained in From field of the next message according to the list order of the main index.
- `$NEXTLINK$` -- Next link markup. See Conditional Links for more information.
- `$NEXTMSG$` -- Filename of next message according to the list order of the main index.
- `$NEXTMSGNUM$` -- Number assigned to next message according to the list order of the main index.
- `$NEXTSUBJECT$` -- Subject of next message according to the list order of the main index.
- `$NUMOFIDXMSG$` -- Number of message listed.
- `$NUMOFMSG$` -- Number of messages in the archive.
- `$PREVBUTTON$` -- Previous button markup. See Conditional Links for more information.
- `$PREVFROM$` -- Contenst of From field of the previous message according to the list order of the main index.
- `$PREVFROMADDR$` -- E-mail address contained in From field of the previous message according to the list order of the main index.
- `$PREVFROMNAME$` -- English" name contained in From field of the previous message according to the list order of the main index.
- `$PREVLINK$` -- Previous link markup. See Conditional Links for more information.
- `$PREVMSG$` -- Filename of previous message according to the list order of the main index.
- `$PREVMSGNUM$` -- Number assigned to previous message according to the list order of the main index.
- `$PREVSUBJECT$` -- Subject of previous message according to the list order of the main index.
- `$PROG$` -- Program name.
- `$SUBJECTNA$` -- Message subject text.

- `$TIDXFNAME$` -- Filename of thread index page.
- `$TIDXTITLE$` -- Title of thread index page.
- `$VERSION$` -- Program version.
- `$YYMMDD$` -- Message date in yy/mm/dd format.

BOTLINKS

The BOTLINKS resource element defines the layout of the navigational links at the bottom of each message page. The markup defined, will appear after the filtered message data and any thread links, and before the MSGFOOT data.

The default value for BOTLINKS is the following:

```
<BOTLINKS>
<HR>
<UL>
$PREVLINK$
$NEXTLINK$
<LI>Index(es):
<UL>
<LI><A HREF="$IDXFNAME$#$MSGNUM$"><STRONG>Main</STRONG></A></LI>
<LI><A HREF="$TIDXFNAME$#$MSGNUM$"><STRONG>Thread</STRONG></A></LI>
</BOTLINKS>
```

If no thread index is specified, then the thread link markup is removed. The variables available for BOTLINKS are the same as for TOPLINKS.

Conditional Links

Since the state of some navigational links can change due the position of the message in the archive (eg. first and last messages), special resources exist that allows you to control the markup of some of the links based upon if the link is valid or not for a given message.

The resource elements for defining the conditional links are the following: PREVBUTTON, NEXTBUTTON, PREVLINK, and NEXTLINK, and their inactive counterparts, PREVBUTTONIA, NEXTBUTTONIA, PREVLINKIA, and NEXTLINKIA. The appropriate value of these elements (ie. if it is active, or inactive) are represented by the $PREVBUTTON$, $NEXTBUTTON$, $PREVLINK$, and $NEXTLINK$ resource file variables, respectively, which may be used in other resource elements' contents (TOPLINKS and BOTLINKS in particular).

The defaults values for each conditional link resource is as follows:

PREVBUTTON

```
<PREVBUTTON>
<A HREF="$PREVMSG$">[Prev]</A>
</PREVBUTTON>
```

NEXTBUTTON

```
<NEXTBUTTON>
<A HREF="$NEXTMSG$">[Next]</A>
</NEXTBUTTON>
```

PREVLINK

```
<PREVLINK>
<LI>Prev: <STRONG><A HREF="$PREVMSG$">$PREVSUBJECT$</A></STRONG></LI>
</PREVLINK>
```

NEXTLINK

```
<NEXTLINK>
<LI>Next: <STRONG><A HREF="$NEXTMSG$">$NEXTSUBJECT$</A></STRONG></LI>
</NEXTLINK>
```

All the "`IA`" elements default to empty content.

NOTE

The last newline for the `PREVBUTTON`, `NEXTBUTTON`, `PREVBUTTONIA`, and `NEXTBUTTONIA` elements is ignored by *MHonArc*. This allows a "tight" grouping of button links; ie. no space between buttons. If you desire to have a newline in the content, just insert a trailing blank line at the end of the element's content.

You should note that there is a correlation between the value of the conditional links elements and the contents of the `TOPLINKS` and `BOTLINKS` elements.

The following variables may be used within the conditional link elements.

- `$NEXTFROM$` -- Contenst of From field of the next message according to the list order of the main index.
- `$NEXTFROMADDR$` -- E-mail address contained in From field of the next message according to the list order of the main index.
- `$NEXTFROMNAME$` -- English" name contained in From field of the next message according to the list order of the main index.
- `$NEXTMSG$` -- Filename of next message according to the list order of the main index.
- `$NEXTMSGNUM$` -- Number assigned to next message according to the list order of the main index.
- `$NEXTSUBJECT$` -- Subject of next message according to the list order of the main index.
- `$PREVFROM$` -- Contenst of From field of the previous message according to the list order of the main index.
- `$PREVFROMADDR$` -- E-mail address contained in From field of the previous message according to the list order of the main index.
- `$PREVFROMNAME$` -- English" name contained in From field of the previous message according to the list order of the main index.

- `$PREVMSG$` -- Filename of previous message according to the list order of the main index.
- `$PREVMSGNUM$` -- Number assigned to previous message according to the list order of the main index.
- `$PREVSUBJECT$` -- Subject of previous message according to the list order of the main index.

WARNING

Never include conditional link variables (`$PREVBUTTON$`, `$NEXTBUTTON$`, `$PREVLINK$`, and `$NEXTLINK$`) in conditional link element content. This will cause an infinite loop during execution and will eventually lead to a crash due to a lack of memory.

Message Layout

Defining the format for the actual mail message data is divided into two parts: the message head and the message body. Customizing the message header markup is described in this section, but due to the nature of how messages are processed, the message body format is controlled by the various MIME filters directly (see the section on MIME for further details).

Excluding Fields

The `EXCS` resource allows you to specify what fields should be excluded in the HTML output.

EXCS

Each line of the `EXCS` element specifies a mail header field to exclude in the converted HTML output. Each line is treated as a Perl regular expression (**NOTE**: *the regular expression is already anchored to the begining of the line*).

The default value for `EXCS` is the following:

```
<EXCS>
content-
errors-to
forward
lines
message-id
mime-
nntp-
originator
path
precedence
received
replied
return-path
```

```
status
via
x-
</EXCS>
```

Any fields you specify for the EXCS resource will augment the default list, unless the "Override" attribute is specified. If "Override" is specified, the default list is discarded along with any other lists specified from previous EXCS elements; and only header fields specified in the EXCS element are excluded.

Field Order

The FIELDORDER resource allows you to control the order the message header fields appear in the HTML output.

FIELDORDER

Each line of the FIELDORDER element is the exact case-insensitive name of a message header field. The order the fields are listed is the order they will appear in the filtered message. The special field value "-extra-" represents all fields not explicitly specified in the FIELDORDER element and not excluded by the EXCS element. Extra fields are listed in sorted order.

The following represents the default value of the FIELDORDER resource:

```
<FIELDORDER>
to
subject
from
date
-extra-
</FIELDORDER>
```

Field Formatting

The FIELDSTYLES and LABELSTYLES resources allow to control how each message header field is formatted.

FIELDSTYLES

Each line in the FIELDSTYLES element defines HTML elements to wrap around the field text in mail headers (e.g. "To: *field text*", "From: *field text*"). The format of each line is "*field_name*:*html_element*". This specifies to wrap *html_element* around the text associated with *field_name*. If *html_element* is empty, then the field text is not wrapped in any element.

MHonArc defines a special *field_name* called "-default-". This is default HTML element to wrap field text in if no explicit specific element is defined for the label.

field_name must be the exact name of a header field name, but character case is ignored.

The default value of `FIELDSTYLES` is the following:

```
<FIELDSTYLES>
-default-
</FIELDSTYLES>
```

LABELSTYLES

Each line in the `LABELSTYLES` element defines HTML elements to wrap around labels in mail headers (e.g. "`To:`", "`From:`"). The format of each line is "*`field_name`*`:`*`html_element`*". This specifies to wrap *`html_element`* around *`field_name`*. If *`html_element`* is empty, then the label is not wrapped in any element.

MHonArc defines a special *`field_name`* called "`-default-`". This is default HTML element to wrap a label in if no explicit specific element is defined for the label.

`field_name` must be the exact name of a header field name, but character case is ignored.

The default value of `LABELSTYLES` is the following:

```
<LABELSTYLES>
-default-:em
</LABELSTYLES>
```

Other Resources

E-mail Links

MAILTOURL

URL to use for e-mail address hyperlinks in e-mail message header fields. The following variables are defined for the `MAILTOURL` resource:

- `$FROM$` -- Who the message is from.
- `$MSGID$` -- Message ID of the message.
- `$SUBJECT$` -- The subject of the message.
- `$TO$` -- Destination e-mail address of link.

MHonArc will use the following URL by default: "`mailto:$TO$`".

NOTE

The `MAILTOURL` resource has different rules for variable expansion. If a variable does not exactly match the set of variables available for the `MAILTOURL`, the variable text will be taken literally as part of the element content. Therefore, a single "`$`" can be used to represent a "`$`" character.

Also, variables in the `MAILTOURL` should **NOT** have " **:** *NU*" modifier. This will prevent the variables from be recognized. *MHonArc* will automatically treat the replacement value as a part of a URL string.

MIME

MHonArc has support for e-mail messages with *Multipurpose Internet Mail Extensions* (MIME) as defined in RFC 1521. *MHonArc* handles the filtering of the various content-types used in MIME in a modular fashion. Since new content-types are occasionally defined for MIME, this modularity allows users to add new filters to accomodate new content-types. Also, filters can be hooked in to override *MHonArc's* default filters, or provide *MHonArc* with the ability to process existing content-types that it cannot handle currently.

Default Filters

The default filters provided by *MHonArc* supports the following MIME content-types, which may be overriden by user-defined filters:

- application/*
- audio/*
- image/*
- message/news
- message/partial
- message/rfc822
- multipart/alternative
- multipart/digest
- multipart/mixed
- multipart/parallel
- text/html
- text/plain
- text/setext
- video/*

For more information on how to write your filters, or replace existing filters, see Writing Filters.

The next sections describes how *MHonArc* processes the content-types listed above.

application/*

MHonArc extracts the data into a separate file and puts a hyperlink to the file into the HTMLized message.

By default, *MHonArc* ignores any filename specification (the "*name*" attribute as defined in the `Content-Type` header field) given in the message when writing the data to disk. *MHonArc* generates a unique filename with an extenstion based upon sub-type.

If you want *MHonArc* to use the filename, then you can use the MIMEARGS resource and specify an argument string of "`usename`".

Example:

```
<MIMEARGS>
application/postscript:usename
</MIMEARGS>
```

If you want *MHonArc* to use specified filename for all application types, then use the following:

```
<MIMEARGS>
application/*:usename
</MIMEARGS>
```

CAUTION

The use of "`usename`" is discouraged since it can lead to filename conflicts and security problems.

Here is the current list of *application* sub-types (with their filename extensions) supported by *MHonArc*:

- mac-binhex40 (hqx)
- octet-stream (bin)
- oda (oda)
- pdf (pdf)
- postscript (ps)
- rtf (rtf)
- x-bcpio (bcpio)
- x-cpio (cpio)
- x-csh (csh)
- x-dvi (dvi)
- x-gtar (gtar)
- x-hdf (hdf)
- x-latex (latex)
- x-mif (mif)
- x-netcdf (cdf)
- x-patch (*no extension*; processed by the text/plain filter)
- x-sh (sh)
- x-shar (shar)
- x-sv4cpio (sv4cpio)

- x-sv4crc (sv4crc)
- x-tar (tar)
- x-tcl (tcl)
- x-tex (tex)
- x-texinfo (texinfo)
- x-troff (roff)
- x-troff-man (man)
- x-troff-me (me)
- x-troff-ms (ms)
- x-ustar (ustar)
- x-wais-source (src)
- zip (zip)

audio/*

MHonArc extracts the data into a separate file and puts a hyperlink to the file into the HTMLized message. The name of the file created follows the same guidelines mentioned under application/*.

Here is the current list of *audio* sub-types (with their filename extensions) supported by *MHonArc*:

- basic (snd)
- x-aiff (aif)
- x-wav (wav)

image/*

MHonArc will extract the data into a separate file and puts a hyperlink to the file into the HTMLized message. The name of the file created follows the same guidelines mentioned under application/*. In addition to the filename specification mentioned under application/*, an "`inline`" argument may be declared to instruct *MHonArc* to inline the image in the generated HTML.

Example:

```
<MIMEARGS>
image/gif:inline
</MIMEARGS>
```

The following examples says to inline XBM images and use the name attribute as the filename if defined:

```
<MIMEARGS>
image/x-xbm:inline usename
</MIMEARGS>
```

The following represents the default argument settings used by *MHonArc*:

```
<MIMEARGS>
image/gif:inline
image/x-xbitmap:inline
image/x-xbm:inline
</MIMEARGS>
```

Here is the current list of *image* sub-types (with their filename extensions) supported by *MHonArc*:

- gif (gif)
- ief (ief)
- jpeg (jpg)
- tiff (tif)
- x-bmp (bmp)
- x-cmu-raster (ras)
- x-pcx (pcx)
- x-pict (pict)
- x-portable-anymap (pnm)
- x-pnm (pnm)
- x-portable-bitmap (pbm)
- x-pbm (pbm)
- x-portable-graymap (pgm)
- x-pgm (pgm)
- x-portable-pixmap (ppm)
- x-ppm (ppm)
- x-rgb (rgb)
- x-xbitmap (xbm)
- x-xbm (xbm)
- x-xpixmap (xpm)
- x-xpm (xpm)
- x-xwindowdump (xwd)
- x-xwd (xwd)

If the image is a GIF or XBM (X bitmap), the HTML `IMG` element will be used to in-line the image into the HTMLized message.

message/news

message/news signifies an included (MIME) USENET news message. The data associated with a *message/news* part is processed by *MHonArc* in the same manner as a regular mail message.

message/partial

message/partial signifies that the content is a single part of a message split into multiple mail messages. *message/partial* is treated in the same manner as text/plain.

message/rfc822

message/rfc822 signifies an included (MIME) mail message. The data associated with a *message/rfc822* part is processed by *MHonArc* in the same manner as a regular mail message.

multipart/alternative

multipart/alternative signifies multiple content-types with the same (or similiar) information. *MHonArc* processes only the latest part that has a content-type filter.

multipart/digest

mulltipart/digest signifies a series of included mail messages. Each part is processed in the same manner as message/rfc822 unless an explicit content-type is specifed for each part.

multipart/mixed

multipart/mixed signifies data with multiple content-types. *MHonArc* extracts each part and calls the appropriate content-type filter for each part, if defined.

multipart/parallel

multipart/parallel is processed in the same manner as multipart/mixed.

text/html

text/html signifies that the data is HTML markup. The data as left "as is" with the exception of some processing to legally include the HTML into the HTMLized mail message. I.e. *MHonArc* removes the `HEAD` an `BODY` tags, the `TITLE` element will be replaced with an `ADDRESS` element surrounded by `HR`'s, and the `BASE` element URL will be propogated to relative URLs.

text/plain

text/plain signifies ASCII character data. In the HTMLized message, the data is wrapped in a `PRE` element with special characters (`< > &`) converted to entity references. *MHonArc* will also make any URLs into hyperlinks. The following URL types are recognized:

- `http://...`
- `ftp://...`
- `afs://...`
- `wais://...`
- `telnet://...`

- `gopher://...`
- `news:...`
- `nntp:...`
- `mid:...`
- `cid:...`
- `mailto:...`
- `prospero:...`

text/setext

text/setext signifies "*structure enhanced text*". The data is converted into HTML containing hyperlinks as defined by the *setext* data. For more information on *setext*, see <URL:http://www.bsdi.com/setext/>.

video/*

MHonArc extracts the data into a separate file and puts a hyperlink to the file into the HTMLized message. The name of the file created follows the same guidelines mentioned under application/*.

Here is the current list of *video* sub-types (with their filename extensions) supported by *MHonArc*:

- mpeg (mpg)
- quicktime (mov)
- x-msvideo (avi)
- x-sgi-movie (movie)

Non-MIME Messages

Messages that do not contain a MIME `Content-Type` header field are processed as text/plain messages.

Writing Filters

If you want to write your own filter for use in *MHonArc*, you need to know the Perl programming language. The following information assumes you know Perl. To learn how to hook in your filters into *MHonArc*, see Specifying Filters.

Function Interface of Filter

MHonArc interfaces with MIME filters by calling a routine with a specific set of arguments. The prototype of the interface routine is as follows:

```
sub filter {
    local($head, *fields, $data, $decoded, $argstring) = @_;

    # Filter code here

    # The last statement should be the return value, unless an
    # explicit return is done. See the following for the format of the
    # return value.
}
```

Argument Descriptions

`$head`

This is the header text of the message (or body part if called in a mulitpart message).

`*fields`

This is a pointer to an associative array that has broken down `$head` into field label/field value components. The keys are the lower-case representations of the field values. **Example**: If you would like to retrieve the value of the Content-Type field, then use the following: `$fields{'content-type'}`.

If a field occurs more than once in a header, *MHonArc* separates the field values in the associative array by a '`\034`' character. To make your filter less likely to break due to changes in *MHonArc*, you may use the `$'X` variable instead of '`\034`'.

`$data`

This is a copy of the message (or body part if called in a mulitpart message) body.

`$decoded`

This flag is set to 1 if *MHonArc* decoded the message and $data represents the orginal data before encoded by the sender. If set to 0, $data has not been decoded. The failure to decode occurs if *MHonArc* does not recognizeed the encoding specified in the Content-Transfer-Encoding field.

MHonArc has decoded the data for you if it was encoded in 7-Bit, 8-Bit, Binary, Quoted-Printable, Base64, or X-Uuencode.

`$argstring`

This is an optional argument string that may be used to modify the behavior of the filter. The format of this string is determined by the filter itself.

The value of the string is set by the MIMEARGS resource.

Return Value

The return value is treated as an array. The first item in the array is a string representing the HTML markup to insert in the HTMLized message. An empty string may be returned to tell *MHonArc* that the routine was unable to filter data.

Any other array items are treated as names of any files that were generated by the filter. *MHonArc* needs to keep track if any extra files that a filter may generate in order for *MHonArc* to delete those files if the message gets removed from the archive.

Filter Writing Tips

The following recommendations/tips are given to help you write filters:

- Qualify your filter in its own package. This eliminates possible variable/routine conflicts with *MHonArc*.
- If the filter creates derived files (like the image filters), you may use the variable `$'OUTDIR` to determine the location of the mail archive.

  NOTE: *Do not include* `$'OUTDIR` *as part as the filename that is returned to MHonArc. If the filter does create files, just return the base name.*
- Look at the default filters contained in the distribution of *MHonArc*. You can use these as templates for writing your own.
- Make sure your Perl source file ends with a true statement (like "`1;`"). *MHonArc* just performs a `require` on the file, and if the file does not return true, Perl will abort execution.

Using C

If a MIME filter requires the utilization of a C program, or other non-Perl executable, a Perl wrapper must be written for the program in-order to interface with *MHonArc*. The wrapper must follow the rules as specified in Function Interface of Filter.

Specifying Filters

Adding new filters, or overriding existing ones, are done via the Resource File. The two resources for specifying and controlling MIME filters are MIMEFILTERS and MIMEARGS.

MIMEFILTERS

The resource element `MIMEFILTERS` in the Resource File is used to hook in user specifed filters into *MHonArc*. The syntax for each line of the the `MIMEFILTERS` element is as follows:

<*content-type*>**:**<*routine-name*>**:**<*file-of-routine*>

The definition of each colon-separated value is as follows:

<*content-type*>

The MIME content-type the filter processes.

<*routine-name*>

The actual routine name of the filter. The name should be fully qualified by the package it is definedi (e.g. "`mypackage'filter`").

<*file-of-routine*>

The name of the file that defines <*routine-name*>. If the file is not a full pathname, *MHonArc* finds the file by looking in the standard include paths of Perl, and the paths specified by the `PERLINC` resource element.

Any whitespace is stripped out before processing.

Example

The following represents the default value of `MIMEFILTERS`:

```
<MIMEFILTERS>
application/mac-binhex40:m2h_external'filter:mhexternal.pl
application/octet-stream:m2h_external'filter:mhexternal.pl
application/oda:m2h_external'filter:mhexternal.pl
application/pdf:m2h_external'filter:mhexternal.pl
application/postscript:m2h_external'filter:mhexternal.pl
application/rtf:m2h_external'filter:mhexternal.pl
application/x-bcpio:m2h_external'filter:mhexternal.pl
application/x-cpio:m2h_external'filter:mhexternal.pl
application/x-csh:m2h_external'filter:mhexternal.pl
application/x-dvi:m2h_external'filter:mhexternal.pl
application/x-gtar:m2h_external'filter:mhexternal.pl
application/x-hdf:m2h_external'filter:mhexternal.pl
application/x-latex:m2h_external'filter:mhexternal.pl
application/x-mif:m2h_external'filter:mhexternal.pl
application/x-netcdf:m2h_external'filter:mhexternal.pl
application/x-patch:m2h_text_plain'filter:mhtxtplain.pl
application/x-sh:m2h_external'filter:mhexternal.pl
application/x-shar:m2h_external'filter:mhexternal.pl
application/x-sv4cpio:m2h_external'filter:mhexternal.pl
application/x-sv4crc:m2h_external'filter:mhexternal.pl
application/x-tar:m2h_external'filter:mhexternal.pl
application/x-tcl:m2h_external'filter:mhexternal.pl
application/x-tex:m2h_external'filter:mhexternal.pl
application/x-texinfo:m2h_external'filter:mhexternal.pl
application/x-troff-man:m2h_external'filter:mhexternal.pl
application/x-troff-me:m2h_external'filter:mhexternal.pl
application/x-troff-ms:m2h_external'filter:mhexternal.pl
application/x-troff:m2h_external'filter:mhexternal.pl
application/x-ustar:m2h_external'filter:mhexternal.pl
application/x-wais-source:m2h_external'filter:mhexternal.pl
application/zip:m2h_external'filter:mhexternal.pl
audio/basic:m2h_external'filter:mhexternal.pl
audio/x-aiff:m2h_external'filter:mhexternal.pl
audio/x-wav:m2h_external'filter:mhexternal.pl
image/gif:m2h_external'filter:mhexternal.pl
image/ief:m2h_external'filter:mhexternal.pl
```

```
image/jpeg:m2h_external'filter:mhexternal.pl
image/tiff:m2h_external'filter:mhexternal.pl
image/x-bmp:m2h_external'filter:mhexternal.pl
image/x-cmu-raster:m2h_external'filter:mhexternal.pl
image/x-pbm:m2h_external'filter:mhexternal.pl
image/x-pcx:m2h_external'filter:mhexternal.pl
image/x-pgm:m2h_external'filter:mhexternal.pl
image/x-pict:m2h_external'filter:mhexternal.pl
image/x-pnm:m2h_external'filter:mhexternal.pl
image/x-portable-anymap:m2h_external'filter:mhexternal.pl
image/x-portable-bitmap:m2h_external'filter:mhexternal.pl
image/x-portable-graymap:m2h_external'filter:mhexternal.pl
image/x-portable-pixmap:m2h_external'filter:mhexternal.pl
image/x-ppm:m2h_external'filter:mhexternal.pl
image/x-rgb:m2h_external'filter:mhexternal.pl
image/x-xbitmap:m2h_external'filter:mhexternal.pl
image/x-xbm:m2h_external'filter:mhexternal.pl
image/x-xpixmap:m2h_external'filter:mhexternal.pl
image/x-xpm:m2h_external'filter:mhexternal.pl
image/x-xwd:m2h_external'filter:mhexternal.pl
image/x-xwindowdump:m2h_external'filter:mhexternal.pl
message/partial:m2h_text_plain'filter:mhtxtplain.pl
text/html:m2h_text_html'filter:mhtxthtml.pl
text/plain:m2h_text_plain'filter:mhtxtplain.pl
text/richtext:m2h_text_plain'filter:mhtxtplain.pl
text/setext:m2h_text_setext'filter:mhtxtsetext.pl
text/tab-separated-values:m2h_text_plain'filter:mhtxtplain.pl
text/x-html:m2h_text_html'filter:mhtxthtml.pl
text/x-setext:m2h_text_setext'filter:mhtxtsetext.pl
video/mpeg:m2h_external'filter:mhexternal.pl
video/quicktime:m2h_external'filter:mhexternal.pl
video/x-msvideo:m2h_external'filter:mhexternal.pl
video/x-sgi-movie:m2h_external'filter:mhexternal.pl
</MIMEFILTERS>
```

MIMEARGS

The `MIMEARGS` resource may be used to pass optional arguments to filters to control their behavior. Arguments may be defined on a per content-type basis, or for a specific filter itself. The syntax for each line of the the `MIMEARGS` element is as follows:

<*content-type*>**:**<*argument-string*>

Or,

<*filter-name*>**:**<*argument-string*>

The format of argument strings is dependent on the filter that processes <*content-type*> or by the specified filter, <*filter-name*>.

If an argument string is defined for a filter explicitly and for a content-type that the filter processes, the content-type string will override the filter string.

Examples

The following example represents the default settings used by *MHonArc*:

```
<MIMEARGS>
image/gif:inline
image/x-xbitmap:inline
image/x-xbm:inline
</MIMEARGS>
```

The following example tells the filter that deals handles content-types that cannot be converted directly into HTML to use the "*name*" attribute as defined in the `Content-Type` header field as the name of the file generated:

```
<MIMARGS>
m2h_external'filter:usename
</MIMEARGS>
```

The following examples says to inline XBM images and use the name attribute as the filename if defined:

```
<MIMEARGS>
image/x-xbm:inline usename
</MIMEARGS>
```

Gory Details

This sections explain in detail how *MHonArc* functions. Knowing the material covered in this section may help you when trouble shooting.

OS Detection

MHonArc will automatically detect which operating system it is running under. If the following list of conditions are true, *MHonArc* assumes it is running under MS-DOS:

- The `COMSPEC` environment variable is defined.
- The value of the `COMSPEC` environment variable is a legal MS-DOS pathname.
- The value of the `COMSPEC` environment variable is an executable file.

If any of the above conditions is false, *MHonArc* assumes it is running under Unix.

NOTE

The previous conditions are used since the conditions will exist if Perl has been installed on an MS-DOS machine. None of the above conditions exist when Perl is installed on a Unix system.

Processing Steps

This section describes the steps *MHonArc* performs when creating/editting an archive. Anytime messages are added or deleted or the index page layout is changed, *MHonArc* will perform the following steps.

- Creates a lock file. This insures only one *MHonArc* process is updating the archive at any given moment. See Archive Integrity for more information.
- Reads the database file. The name, and location, of the database file can be explicitly specified via the `M2H_DBFILE` and `M2H_OUTDIR` environment variables or the command-line options `-dbfile` and `-outdir`. Otherwise, the current working directory is used.

  NOTE: *The database file must be in the same location as the archive since the* `M2H_OUTDIR` *variable and* `-outdir` *option also specify the location of the archive.*

  The database file contains data to update any mail threads and the resource settings when *MHonArc* was last invoked. This allows new messages to contain the same formatting/resource specifications as existing messages in the archive without having to re-specify the resources each time new messages are added. Resources defined in the database file override the environment variables.

  NOTE: *If no database file is found, MHonArc will create a new archive.*

- Read the *MHonArc* resource file, *if specified*. The resource file will override any settings contained in the database file.
- Read the settings specified on the command-line. Command-line options override any settings in the database and/or resource file.
- Update archive.
- Rewrites the index pages to reflect the update.
- Writes a new database file containing the new state of the archive and all (new) resource settings.

Normally, knowing all the previous steps is unnecessary. However, it may be useful to be aware of them if unexpected behavior, or errors, occur.

Archive Integrity

MHonArc applies safeguards to try to insure that a mail archive does not get corrupted due to exceptional circumstances. *MHonArc* does the following to insure a mail archive does not get corrupted:

- *MHonArc* creates a lock file, "`.mhonarc.lck`", when creating/updating an archive. The lock file insures that only one *MHonArc* process is modifying an archive at any given moment. The `-locktries` command-line option, or the `M2H_LOCKTRIES` environment variable, allows you to control how long a given *MHonArc* process will wait if an archive is currently locked. If *MHonArc* can not lock the archive after the specified number of tries, *MHonArc* will exit, unless the `-force` option is specified.
- *MHonArc* will ignore the following signals once messages are actually being written to disk: `SIGABRT`, `SIGHUP`, `SIGINT`, `SIGQUIT`, `SIGPIPE`, `SIGTERM`. Possible archive corruption can still occur if a `SIGKILL` signal is received since `SIGKILL`s are uncatchable. A `SIGKILL` will also prevent *MHonArc* from deleting the lock file.

File Formats

Database File

The *MHonArc* database file is actual Perl code. *MHonArc* requires it like any other Perl library to load in the contents of the database.

CAUTION

You should never modify the database file by hand. Changing the file by hand could cause future incorrect/unpredictable behavior when processing the archive.

Index and Message Files

The indexes and message files are legal HTML documents. However, manual editting of the documents is discouraged. The documents contain special comment declarations. The comment declarations act as markers which allow *MHonArc* to correctly edit the documents when needed.

The comment declarations look like the following:

```
<!--X-Body-Begin-->
<!--X-User-Header-->
<!--X-User-Header-End-->
<!--X-TopPNI-->
...
```

Derived Files

Derived files are files that are generated by the MIME filters. These files are created when the data being processed in messages cannot be converted to HTML (eg. images, postscript, video, binaries). The format of these files depend on the content-type of the data.

Notes

- Here is the explicit order of decreasing precedence when setting resources/options:
  - command-line options (*highest precedence*)
  - resource file
  - database file
  - environment variables (*lowest precedence*)
- Mail thread detection is dependent upon the mail messages containing the message id(s) of referenced messages. Most mailers *reply function* will automatically include the message id of the message being replied to.
- All mail message being converted into HTML are stored in memory before they are written to disk. This can eat up much memory if many mail messages are being converted. If you are processing multiple mailboxes/folders and worried about memory, you can try the following:
  - Invoke *MHonArc* on each one separately using the `-add` option.
  - Or, invoke *MHonArc* with the `-savemem` option.
- The database file, and the index pages, are completely rewritten evertime new messages are added. This may cause slight slow-downs when archives become very large.
- When reading *MH* mail folders, mail message are assumed to have numeric filenames.

- When sorting by date, *MHonArc* tries to use the date listed in the first *Received* field of the message. If no *Received* field exists, than the *Date* field is used.
- No distinction is made, in the output, on which messages came from which mail folder if multiple mail folders are processed.
- *MHonArc* can probably be modified to handle other types of mailers (which has been done since the original version only supported *MH* mail folders). The `MSGSEP` resource gives flexibility in processing Unix style mailbox files.

Diagnostics

Three types of messages exist in *MHonArc*: *Informative* messages, *Warnings*, and *Errors*. *Informative* messages give you the current status of *MHonArc*'s execution. *Warnings* signify undesired conditions, but are *not* critical in *MHonArc*'s exection. *Errors* signify critical conditions that inhibit *MHonArc* from finishing its task.

Another set of messages exists that are generated from the Perl interpreter itself. *MHonArc* tries its best to catch any conditions that may cause Perl to abnormally abort, but conditions may arise where this is not possible.

This section describes the various diagnostics *MHonArc* may produce and messages Perl may produce.

Informative messages

Informative messages may be suppressed via the `-quiet` command-line option. Only the more important Informative messages are listed here.

Could not process message with given Content-Type: ...

MHonArc will output this statement *in* filtered mail messages for content-types it is unable to process. See Default Filters in MIME for content-types that *MHonArc* supports by default. See Writing Filters for adding new filters into *MHonArc*.

This is the only Informative message that does not go to standard output, but into the actual filtered mail message.

No new messages

No mail messages exist when performing an add operation to an archive. This can occur if an empty *MH* mail folder, or empty mailbox file, is passed to *MHonArc*.

Requiring MIME filter libraries ...

Indicates *MHonArc* is loading external libraries for filtering mail messages. *MHonArc* will output each library it loads. See MIME for more information of filter libraries.

Trying to lock mail archive ...

The statement means that a lock file is in place for the archive you are trying to update. Normally, an existing lock file implies that another *MHonArc* process is currently using the archive, and other *MHonArc* processes will wait awhile to see if the archive will be unlocked.

However, there are times when a lock file exists, but no *MHonArc* process is modifying the archive. This can occur if *MHonArc* is abnormally terminated. If you know that no other *MHonArc* process is editting the archive you are try to modify, then manually remove the lock file or use the `-force` option.

See Archive Integrity for more information.

Warnings

Warning messages denote some undesired event occurred, but the event is not severe enough to cause program termination.

Warning: Could not find date for message

MHonArc was unable to find a received/sent date for a mail message. With respect to other mail messages, a message with no received/sent date is first in chronological order.

Warning: Database (<dbversion>) != program (<prgversion>) version

Indicates that the version of *MHonArc* updating an archive is different from the version of *MHonArc* that created the database file. Problems can arise if the database file changes in format from different version of *MHonArc*. See the release notes of the *MHonArc* distribution if changes in the databse format has effects on older archives.

Warning: Unable to create <outdir>/<dbfile>

Indicates *MHonArc* was unable to create the database file <*dbfile*> for the mail archive created/modified in <*outdir*>. This message can occur if <*outdir*> permissions changed during *MHonArc* execution, the existing <*dbfile*> is read-only, or the file system is full.

This message can be severe because no future add operations can be performed to the archive.

Warning: Unable to open footer: <footer>

MHonArc was unable to open the footer file, <*footer*>, for inclusion into the index page. Make sure <*footer*> exists, and is readable by you.

Warning: Unable to open header: <header>

MHonArc was unable to open the header file, <*header*>, for inclusion into the index page. Make sure <*header*> exists, and is readable by you.

Warning: Unable to open <folder>

MHonArc was unable to open the specified mail <*folder*> for reading. Make sure <*folder*> exists and is readable (and executable if a directory) by you.

Warning: Unable to open message: <folder>/<message>

MHonArc was unable to open the specified *MH* mail message <*folder*>/<*message*> for reading. Make sure <*folder*>/<*message*> exists and is readable by you.

Warning: Unable to open resource file: <file>

MHonArc was unable to open the resource file, <*file*>, for reading. Make sure <*file*> exists, and is readable by you.

Warning: Undefined time zone: "<timezone>"

MHonArc has found an unrecognized timezone acronym, <*timezone*>, in a mail message. You can tell *MHonArc* about other timezone acronyms, and their hour offset to UTC, by using the `TIMEZONES` resource element of the Resource File.

The timezone UTC (or GMT) is used for an undefined timezone acronym

Errors

Errors denote conditions that cause *MHonArc* to abort execution.

Some error conditions may cause the *MHonArc* archive to become corrupted. If the error occurs when *MHonArc* is writing files, you may have to recreate the archive from the original messages.

ERROR: Database read error of <dbfile>

An error occured when trying to read an archive's database. The error can occur if the database file is not readable or the file got corrupted.

ERROR: Unable to create <file>

MHonArc was unable to create <*file*>. This message can occur if the directory being written to is not writable, a read-only file with the same name exists, or the file system is full.

ERROR: Unable to create <lockfile> after <#> tries

The statement means that a lock file is in place for the archive you are trying to update.

Sometimes a lock file exists, but no *MHonArc* process is modifying the archive. This can occur if *MHonArc* is abnormally terminated. If you know that no other *MHonArc* process is editting the archive you are try to modify, then manually remove the lock file or use the `-force` option.

ERROR: Unable to open <file>

MHonArc was unable to open <*file*> for reading. Make sure <*file*> exists, and is readable by you.

ERROR: Unable to require newgetopt.pl

The `newgetopt.pl` library is needed for *MHonArc* to parse the command-line. `newgetopt.pl` is part of the standard Perl distribution. Make sure Perl has been correctly installed at your site.

ERROR: Unable to require timelocal.pl

The `timelocal.pl` library is needed for *MHonArc* to process dates in messages. `timelocal.pl` is part of the standard Perl distribution. Make sure Perl has been correctly installed at your site.

ERROR: Unable to require <file>

This message signifies *MHonArc* was unable to require the library <*file*>. Make sure you properly installed *MHonArc* via the installation program. If <*file*> is your own custom filter, make sure you properly registered it in the Resource File. See also Specifying Filters and the `PERLINC` resource element.

Perl Messages

Generally, if execution is aborted and the following error messages appear, then you will have to manually delete the lock file since *MHonArc* will not have the chance to delete the file.

Can't locate <file> in @INC at <file> line <number>.

A library that *MHonArc* tried to load was not found in the Perl include search paths. This error usually implies that *MHonArc* was not installed correctly. Make sure that *MHonArc* was installed via the `install.me` program that is provided in the *MHonArc* distribution.

<file> did not return a true value at <file> line <number>.

If you are using your own MIME filters with *MHonArc*, make sure the library files return 1.

Glossary

HTML

Hypertext Markup Language. HTML is the main document markup language for the World Wide Web.

MIME

Multipurpose Internet Mail Extensions. MIME allows the transmission of non-ASCII data, and mixed content data, in electronic mail messages.

MH

Message Handler. MH is a free message handling system initially developed by the *RAND Corporation*, with subsequent development done at the University of Califonia: Irvine.

Perl

Practical Extraction and Report Language. Perl is an interpreted programming language suited for processing text and generating reports.

SGML

Standard Generalized Markup Language. SGML is a language for document representation.

Contacts

Mailing List

A mailing list, mhonarc@rosat.mpe-garching.mpg.de, is available to provide a discussion forum on the usage and development of *MHonArc*. Appropriate topics for the list include: usage questions, bug reports, behavioral enhancements, documentation bugs, and general help.

To subscribe to the mailing list, send mail to mhonarc-request@rosat.mpe-garching.mpg.de with the command,

```
subscribe
```

as the message body.

If you send mail mhonarc@rosat.mpe-garching.mpg.de, your message will be distributed to all subscribers on the list.

The mailing list is archived by Majordomo. You can also use the WWW to access the archive (with full text search using *glimpse*) at <URL:http://www.rosat.mpe-garching.mpg.de/mailing-lists/mhonarc/>

People

Earl Hood

ehood@convex.com

Main developer of *MHonArc*. Contact for bug reports, behavioral enhancements, documentation bugs, and Unix usage issues.

Steve Pacenka

sp17@cornell.edu

Contributing developer. Worked on isolating code that would conflict with MS-DOS. Contact for MS-DOS installation problems or MS-DOS usage issues.

Achim Bohnet

ach@rosat.mpe-garching.mpg.de

Contributing developer. Administrator, and maintainer, of the *MHonArc* mailing list.