The `convertMSmail` Script

Michael L. Hall

September 3, 2002

Abstract

The **`convertMSmail`** script is a perl script that is used to convert a mail file in Microsoft Mail format into one that is readable by Eudora. This file may also be readable by Unix mail.

Table of Contents

1 Author, Date and Version Information

Filename: convertMSmail
Author: Michael L. Hall
Los Alamos National Laboratory
P.O. Box 1663, MS-D409
Los Alamos, NM 87545
Email: `hall@lanl.gov`

CVS Info: `Id: convertMSmail,v 3.2 2002/09/04 01:10:09 hall Exp`

2 Usage

To use the `convertMSmail` script to convert an input file in MS mail format to an output file in standard mail format:

```
% convertMSmail [<] input_file [> output_file]
```

Here are some sample usages:

```
galt% convertMSmail < MSmailfile > mailfile

galt% convertMSmail MSmailfile > mailfile
```

3 Coding Details

```
# Settings.

$, = '';                    # Set output field separator.
$\ = "\n";                  # Set output record separator.

# Define subroutine to print out a letter.

sub print_letter {

    # Try to find a "From:" field in the header.

    $user = "???";
    $machine = "???";
    $body = 0;    # Set to true when the body of the letter begins.
    $header = 0;  # Set to true when the header of the letter begins.
    foreach (@letter) {
        /^\w*:/ && ($header = true);
$header && /^$/ && ($body = true);

        # Set user and machine from "From:" field in the header.

if (/^From:/ && !$body) {
    ($user,$machine) = /(\w*)@([\w\.]*)/;
    ($user =~ /^$/) && (($user) = /^From: .* (\w*)$/);
            ($user =~ /^$/) && ($user = "???");
            ($machine =~ /^$/) && ($machine = "local");
}
    }

    $from = $user."@".$machine;
```

```
    # Search for the "Date:" line in the header and output
    # the modified version. Only print a line for the first
    # one found.

    $founddate = 0;
    foreach (@letter) {
if (/^Date:/ && !$founddate) {
    $founddate = true;
    ($date,$weekday,$month,$day,$year,$time) = split;

            # Remove commas.
    $weekday =~ s/,//;
    $day =~ s/,//;

            # Add spaces and zeroes for alignment.
    ($day =~ /^[1-9]$/) && ($day = " ".$day);
    ($time =~ /^[0-9]:/) && ($time = "0".$time);
    $time = $time.":00";

    # Print out the starting line for the message.
    print "From ", $from, " ", $weekday, " ", $month,
          " ", $day, " ", $time, " ", $year;

    # Fix the "Date:" field.
    $_ = "Date: ".$weekday.", ".$day." ".$month." ".$year." ".$time;
}
    }

    # Print out error message and try to handle the error.

    if (!$founddate) {
print STDERR "Could not find date field, printing fake one.";
print "From", $from, "Thu Jan  1 00:00:00 1997";
    }

    # Don't print lines until a line with a word and then
    # a colon is found, then print all lines.

    $startprint = 0;
    foreach (@letter) {
/^\w*:/ && ($startprint = true);
$startprint && (print);
    }

    # Reset letter array.

    @letter = 0;
}

# End of subroutine print_letter.


# Start with linecount set to zero.
```

```
$linecount = 0;

while (<>) {
    chop;   # Strip record separator.

    # Test for start of new letter.

    if (/^=============================================================$/) {
&print_letter;
$linecount = 0;
    }

    # Increment linecount and store line.

    $linecount++;
    $letter[$linecount] = $_;

}

# Print out the last letter.

&print_letter;
```