

Table of content

[Overall](#)
[Dependencies](#)
[Functionality](#)
[Other](#)

Overall

This project is written in the course Practical Programming with Perl.

The goal is to create a simple tool that will extract images from email and will convert those images to a specified format. The format and directory are configured through a configuration file. It has two interfaces - web and console.

Dependencies

ImageExtractor::Config relies on the following external packages:

- List::MoreUtils
- Config
- File::Spec

ImageExtractor::Server relies on the following external packages:

- HTTP::Server::Simple::CGI

ImageExtractor::EmailParser relies on the following external packages:

- Email::MIME

ImageExtractor::Extractor relies on the following external packages:

- Image::Magick

Functionality

ImageExtractor::Config

This package is used for configuring the image extractor, regardless on the type of usage - console or web interface.

sub load - load configuration from a file. Depending on the operating system, the files will be searched in "/usr/local/etc/imgstore.con"(unix) or current directory(win) with name config.ini. If no configuration file exists, program will exit.
During the parsing of the configuration file, invalid configurations will be reported in the standart output.
If the specified directory fo images does not exists, it will be created.
The configuration file must have the following format:
key1 = value1
key2 = value2
...
where permitted keys are {dest_dir, dest_format} and for dest_format permitted values are {jpeg, jpg, tiff, tif, png, bmp, gif}.

sub get_save_file_format - returns the file format for images that will be saved after converting.

ImageExtractor::EmailParser

This package is used for processing email (raw) content.

sub get_information - returns the email of the sender, the unique message id and a hash of binary images found in the email.
The hash keys are the filenames of the images and the values are the binary images themselves.

ImageExtractor::Server

This package is used for the web interface of the image extractor.
The web server will answer on the following paths:
"/home" - consists of the email content upload form.
"/upload" - for processing uploaded email content file.
By default, the server will render the upload form.

sub resp_home - response for the upload form. Will render a simple form with and upload button and submit button.
The form's action is "/upload"

sub resp_upload - response for the uploaded email file.

If no file has been submitted, response will render a warning message and will stop before attempting to extract images.

Else, it will load the configuration, read email content as a whole string and uses the Extractor to process the images.

Last, will render a successful message.

ImageExtractor::Extractor

sub extract - extracts images from the given string content of the email and saves the images.

First parameter is the email's as a string.

Second parameter is the format that will be used for the filenames of the images. This format should be generated by the Config class.

When the email is passed, this method will get message params(sender email, message id, images) and will convert the images using Image::Magick

Other

server.pl - starts the server on port 8080, on background mode. Displays process id needed to kill the process.

main.pl - main script for console interface.

Requires at least one filename as an argument. The file must be in the same directory as this script. If no file is specified, it will display an error message and end the script. Else, it will check each file for existence and then do the same as the resp_upload of the Server - read email content as a whole string and use the Extractor to process the images.

For help, use main.pl -h.