

# The Perl Email Project

# The Basics

- <http://pep.kwiki.org>
- Simple as Possible
  - No Simpler

# Advantages

- CPAN Modules are Small
- Simple Code
  - Less Horrible Bugs
- Modular Needs

# Disadvantages

- Complexity is Offset
- Large Number of Modules
- More Work Required of You

# Starting Simple

## • Email::Simple

```
my $msg      = do { local $/; <> }  
my $simple = Email::Simple->new($msg);  
print $simple->header('Subject'), "\n";  
print $simple->body;
```

```
$simple->header_set('X-PM', 'Pittsburgh');  
$simple->body_set(`fortune`);
```

# What About MIME Mail

## • Email::MIME

```
my $msg = do { local $/; <> }  
my $mime = Email::MIME->new($msg);  
print $mime->header('Subject'), "\n";  
print $mime->body;
```

```
my $i = 1;  
printf "%d: %s", $i++, $_->filename(1)  
    for $mime->parts;
```

# Simply Create Email

- Email::Simple::Creator

```
my $snew = Email::Simple->create(  
  header => [  
    From => 'casey@geeknest.com',  
    To    => 'pgh-pm@mail.pm.org',  
    Subject => 'Welcome to the Meeting',  
  ],  
  body => $message,  
);
```

# MI MI MIME

- Email::MIME::Creator
- Email::MIME::Modifer

```
my $single = Email::MIME->create(  
    body => $message,  
);
```



# MI MI MIME

- Email::MIME::Creator
- Email::MIME::Modifer

```
my $single_enc = Email::MIME->create(  
  attributes => {  
    encoding      => 'base64',  
    content_type => 'application/pdf',  
  },  
  body => $pdf,  
);
```

# MI MI MIME

- Email::MIME::Creator
- Email::MIME::Modifer

```
my $multi = Email::MIME->create(  
  header => [  
    From => 'casey@geeknest.com',  
    To    => 'pgh-pm@mail.pm.org',  
    Subject => 'Welcome to the Meeting',  
  ],  
  parts => [ $single, $single_enc ],  
);
```

# MI MI MIME

- Email::MIME::Creator
- Email::MIME::Modifer

```
my ($single, $enc) = $multi->parts;  
$enc->encoding_set('quoted-printable');  
$enc->disposition_set('attachment');  
  
$single->content_type_set('text/html');  
$single->encoding_set('base64');  
$single->body_set($html);  
  
$multi->parts_set([ $single, $enc ]);
```

# Deliver My Email

- Email::FolderTypes
- Email::Folder
- Email::LocalDelivery

```
perl -MEmail::LocalDelivery  
-le'Email::LocalDelivery->deliver(  
    join("",<>), "~/INBOX")'
```

```
my $folder  
    = Email::Folder->new("~/INBOX");  
print $_->header('Subject')  
    for $folder->messages;
```

# Protect My Email

- Email::Filter
- Email::Filter::Rules

```
sub box ($) { "~Mail/$_[0]" }  
my $email = Email::Filter->new;  
$email->accept(box 'INBOX');
```

```
$email->accept(box 'pgh-pm')  
  if $email->to =~ /pgh-pm@mail\.pm\.org/  
  || $email->cc =~ /pgh-pm@mail\.pm\.org/;
```

# Send my Email

- Email::Send

- Email::Reply

```
send IO => $message, 'geeknest.com';
```

```
send SMTP => $message, 'geeknest.com',  
  username => $user, password => $pass;
```

```
my $reply = reply to => $message,  
  from => 'casey@perl.org',  
  body => $response;
```

# Store My Email

## • Email::Store

```
use Email::Store;
```

```
use Acme::Rant;
```

```
@#$$-ing die RANT {
```

```
    Have you seen Acme::Rant?
```

```
    In other words, I haven't played with  
    this yet.
```

```
}
```

# Oddities

- Email::Address
- Email::Abstract
- Email::MessageID
- Email::Valid

```
my $message = Email::Abstract->cast(  
    $object_or_string => "Email::Simple");  
my $subj = Email::Abstract->get_header(  
    'Subject');
```



Do We Have Time for  
Cool Tricks?