

# Mail

*a simple mail API for Drupal*

Gerhard Killesreiter

`gerhard@killesreiter.de`

# Why?

---

# Why? Drupal sends mail.

---

# Why? Drupal sends mail.

---

- Drupal core

# Why? Drupal sends mail.

---

- Drupal core
  - passwords

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions



# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters
  - event reminders

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters
  - event reminders
  - forum subscriptions

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters
  - event reminders
  - forum subscriptions
  - order confirmations

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters
  - event reminders
  - forum subscriptions
  - order confirmations
  - massmailing

# Why? Drupal sends mail.

---

- Drupal core
  - passwords
  - new passwords
  - contact mail
- contributions
  - newsletters
  - event reminders
  - forum subscriptions
  - order confirmations
  - massmailing
  - ...

# Current situation

---

# Current situation

---

- `user_mail( )` is a wrapper for PHP's `mail( )`



# Current situation

---

- `user_mail()` is a wrapper for PHP's `mail()`
- PHP's `mail()` is a wrapper for the MTA

# Current situation

---

- `user_mail()` is a wrapper for PHP's `mail()`
- PHP's `mail()` is a wrapper for the MTA
- `mail()` forks a child process: not very fast, limits efficiency

# Current situation

---

- `user_mail()` is a wrapper for PHP's `mail()`
- PHP's `mail()` is a wrapper for the MTA
- `mail()` forks a child process: not very fast, limits efficiency
- `mail()` can be bypassed by defining a SMTP library.

# Current situation

---

- `user_mail()` is a wrapper for PHP's `mail()`
- PHP's `mail()` is a wrapper for the MTA
- `mail()` forks a child process: not very fast, limits efficiency
- `mail()` can be bypassed by defining a SMTP library.
- bypassing `mail()` does not help, if you cannot change the way, modules pass the mail.

# Requirements

---

# Requirements

---

- small set of features – no bloat in core

# Requirements

---

- small set of features – no bloat in core
- core module usable for everybody – depends on PHP's mail()

# Requirements

---

- small set of features – no bloat in core
- core module usable for everybody – depends on PHP's mail()
- customizable – use your own backend, by using another module



# Features

---

# Features

---

- three modes of mail sending:

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)
- simple SQL based mail queue

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)
- simple SQL based mail queue
- html to text conversion

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)
- simple SQL based mail queue
- html to text conversion



# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)
- simple SQL based mail queue
- html to text conversion

# Features

---

- three modes of mail sending:
  - 'single' – prepare and send one mail to one recipient
  - 'multiple' – do the same for several recipients
  - 'send as is' – send preformatted email(s)
- simple SQL based mail queue
- html to text conversion

# Implementation I

---

```
function mail_mailengine($op, $message = array()) {  
    switch ($op) {  
        case 'name':  
            return 'Drupal';  
            break;  
        case 'description':  
            return t("Lightweight mailing engine  
                with a simple queue.  
                Uses PHP's mail().");  
            break;  
    }  
}
```

# Implementation II

---

```
case 'settings':  
    $unsent_mails = db_result(db_query('SELECT CO  
    $form = form_item(t('Mail engine'), 'Drupal'  
    $form .= form_item(t('Queue status'), format  
    $form .= form_item(t('Last sent'), variable_g  
    $throttle = array(0, 50, 100, 200, 500, 1000  
    $throttle[0] = t('none');  
    $throttle['1000000000'] = t('No limit'); // V  
    $form .= form_select(t('E-mail queue throttle  
    return $form;  
    break;
```

# Implementation III

---

```
case 'single': // send a single mail with some
    $message['headers'] = mail_prepare_headers($message);
    $message['address'] = mime_header_encode($message['address']);
    $message['subject'] = mime_header_encode($message['subject']);
    $message['body'] = str_replace("\r", '\r', $message['body']);
case 'send as is': // no formatting whatsoever
    $status = mail(
        $message['address'],
        $message['subject'],
        $message['body'],
        $message['headers'],
        $message['parameters']
    );
```

# Implementation IV

---

```
        case 'multiple': // send multiple mails
            mail_queue($message['address'], $message['subject']);
            break;
        default: // method not implemented.
            return FALSE;
            break;
    }
}
```

# Alternative implementation I

---

```
function simplemail_mailengine($op, $message = array()) {  
    require_once('activeMailLib.php');  
    switch ($op) {  
        case 'name':  
            return 'Simplemail';  
            break;  
        case 'description':  
            return t("Mailing engine  
                using activeMailLib.");  
            break;  
        case 'settings':  
            $form = form_item(t('Mail engine'), 'Simplemail',  
                t('This mailing engine does not have  
                any special settings.'));  
            return $form;  
            break;  
    }  
}
```

# Alternative implementation II

```
case 'single': // send a single mail with some
    $message['headers'] = mail_prepare_headers($message['headers']);
    $message['address'] = mime_header_encode($message['address']);
    $message['subject'] = mime_header_encode($message['subject']);
    $message['body'] = str_replace("\r", '\r', $message['body']);
case 'send as is': // no formatting whatsoever
    $email = new activeMailLib('plain');
    $email->From($headers['From']); $email->To($headers['To']);
    $email->Subject($message['subject']);
    $email->Message($message['body'], $headers['headers']);
    $email->Send();
    $status = $email->isSent($message['address']);
    if (!$status) {
```



# Alternative implementation III

---

```
case 'multiple': // send multiple mails
    foreach ($message['address'] as $address){
        $mail = $message;
        $mail['address'] = $address;
        $status = simplemail_mailengine('single', $mail);
    }
    break;
default: // method not implemented.
    return FALSE;
    break;
}
}
```

# Module settings

---

```
function mail_settings() {
  $form = '';
  $form .= form_textfield(t('E-mail address'), 'sitename');
  $choices = mail_get_engines();
  $form .= form_select(t('E-mail engine'), 'mail_engine',
    variable_get('mail_engine', 'mail'),
    $choices, t('Choose an e-mail engine for this site'));
  if (variable_get('mail_engine', 0)) {
    $form .= form_group(t('Engine specific settings'),
      $form);
  }
  else {
    drupal_set_message(t('Please choose a mail engine for this site.'));
  }
}
```

# Module selector

---

```
function mail_get_engines() {  
    $engines = array();  
    foreach(module_implements('mailengine') as $module) {  
        $function = $module . '_mailengine';  
        if (function_exists($function)) {  
            $engines[$module] = $function('name') . ' - ' . $module;  
        }  
    }  
    return $engines;  
}
```

# Implementation: Mail queue I

---

```
function mail_queue($addresses, $subject, $body, $additional_headers) {
    require_once './includes/mail.inc';
    $mail_id = db_next_id('{mail_queue}_mid');
    $headers = mail_prepare_headers($additional_headers);
    $subject = mime_header_encode($subject);
    db_query("INSERT INTO {mail_queue} (mid, timestamp) VALUES ($mail_id, NOW())");
    foreach ($addresses as $address) {
        db_query("INSERT INTO {mail_queue_addresses} (mail_id, address) VALUES ($mail_id, '$address')");
    }
    watchdog('mail', t('Mail with ID %mail_id queued'), WATCHDOG_NOTICE);
}
```

# Implementation: Mail queue II

```
function mail_queue_send() {  
    $max_can_send = variable_get('mail_queue_throttle');  
    $last_sent = variable_get('mail_queue_last_sent');  
    $can_send = 0;  
    if ($max_can_send > 0) {  
        $period = time() - $last_sent;  
        $can_send = floor($period / 3600 * $max_can_send);  
        $can_send = ($can_send < 1) ? 1 : $can_send;  
    }  
    if ($can_send) {  
        $result = db_query('SELECT q.*, a.* FROM {mail_queue} q JOIN {mail_recipients} a ON q.recipient_id = a.id WHERE q.scheduled_time <= %d', time());  
        $mails_sent = array();  
        while ($row = $result->fetch_assoc()) {  
            $sent = 0;  
            foreach ($row['recipients'] as $recipient) {  
                $sent++;  
                $recipient =>> send_mail($recipient, $row['subject'], $row['body']);  
            }  
            $row['sent_count'] += $sent;  
            $mails_sent[] = $row;  
            $last_sent = time();  
        }  
        variable_set('mail_queue_last_sent', $last_sent);  
    }  
}
```