Zend_Mail_Transport_File - Alexander Steshenko & Oleg Lobach

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

<ac:macro ac:name="unmigrated-inline-wiki-markup"><ac:plain-text-body><![CDATA[

Zend Framework: \Zend\Mail\Transport\File Component Proposal

Proposed Component Name	\Zend\Mail\Transport\File
Developer Notes	http://framework.zend.com/wiki/display/ZFDEV/\Zend\Mail\Transport\File
Proposers	Alexander Steshenko
Zend Liaison	TBD
Revision	(wiki revision: 12)

Table of Contents

1. Overview
2. References
3. Component Requirements, Constraints, and Acceptance Criteria
4. Dependencies on Other Framework Components
5. Theory of Operation
6. Milestones / Tasks
7. Class Index
8. Use Cases
9. Class Skeletons

1. Overview

\Zend\Mail\Transport\File being used in place of a regular mail transport (like Smpt or Sendmail) instead of sending the mail just saves all email messages as they are in the file system.

2. References

Source Code (Working version with unit-tests)

- git repository
- testcase

3. Component Requirements, Constraints, and Acceptance Criteria

WILL save outgoing email in the file system
MUST provide some standard way to generate file names

MUST let users to use their own algorithms for generating file names
MUST allow configuration through Zend_Application_Resource_Mail options

WILL NOT really send an e-mail message

4. Dependencies on Other Framework Components

- \Zend\Mail\Transport\Abstract
- \Zend\Mail\Transport\Exception

5. Theory of Operation

When transport's internal *sendMail() is called Zend\Mail\Transport\File takes prepared headers and body and dumps them in a file in the specified directory. Name for the file is generated based on the callback you provide. Standard callback uses 'ZendMail'* . time() . '_' . mt_rand() . '.tmp' format.

Both path to the directory you want to save the files to and callback function may be passed to the transport's constructor in an array, which makes it also possible to be configured via the Zend_Application Mail resource like this:

```
[development]
;    This options for the dev env:
resources.mail.transport.type = file
resources.mail.transport.path = "/tmp/send"
```

So that it's convinient to use in development environments like default mail transport for various e-mail messagin testing without changing actual code at all.

If 'path' value is not set manually, default is defined by calling sys_get_temp_dir php function.

The callback used for generating filenames gets the transport instance as the only parameter, so that it can use email message details for composing the filename and also it can use default callback (let's say to prepend something to the filename it generates).

Adapters

About different storage options / adapters etc. I suggest keeping this transport as easy as it is now. It would allow great but simple posibility for different email messaging testings especially for systems where bulk emails and mass messaging take place.

For different storage formats (if it is required and desirable to be implemented as a Zend_Mail transport) I suggest implementing another transport, like \Zend\Mail\Transport\Storage which would utilize already implemented storages from Zend/Mail/Storage/*. I see Zend_Mail_Storage_Writable_Maildir currently implements Zend_Mail_Storage_Writable_Interface, the others may be tweaked same way probably, not went into much details about that yet.

Any comments on this one?

6. Milestones / Tasks

- DONE Milestone 1: Complete theory of operation
- Milestone 2: Define if some additional features are required (using feedback).
- Milestone 3: Documentation

7. Class Index

- \Zend\Mail\Transport\File

8. Use Cases

UC-01: Configuring the transport using Zend_Application_Resource_Mail

This is how the configuration might be done in application.ini configuration file. This would make File mail transport the default transport for the development environment, while keeping regular smtp for the production:

```
[production]
;    Config we use for production:
resources.mail.transport.type = smtp
resources.mail.transport.host = "smtp.example.com"
resources.mail.transport.auth = login
resources.mail.transport.username = myUsername
resources.mail.transport.password = myPassword

[development]
;    This options for the dev env:
resources.mail.transport.type = file
resources.mail.transport.path = "/tmp/send"
```

UC-02: Using a custom callback to generate file names

To use a custom callback just pass a function that accepts the transport's instance as first parameter as 'callback' option to the \Zend\Mail\Transport\File constructor (or to setOptions method). In the use case below, callback utilizes default callback, but prepends the file name with the recipient email:

```
use Zend\Mail;

// callback utilizes default callback and prepends recipient email
$callback = function($transport) {
    $defaultCallback = $transport->getDefaultCallback();
    return $transport->recipients . '_' . $defaultCallback($transport);
};

$transport = new Mail\Transport\File(array('callback' => $callback));

// sending a mail using the transport
$mail = new Mail\Mail();
$mail->setBodyText('This is the text of the mail.')
     ->setFrom('alexander@example.com', 'Alexander Steshenko')
     ->addTo('oleg@example.com', 'Oleg Lobach')
     ->setSubject('TestSubject')
     ->send($transport);
```

In the target folder a file with name like "oleg@example.com_ZendMail_1276322086_1871539247.tmp" appears and the content is

```
From: Alexander Steshenko <alexander@example.com>
To: Oleg Lobach <oleg@example.com>
Subject: TestSubject
Date: Sat, 12 Jun 2010 09:54:46 +0400
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: quoted-printable
Content-Disposition: inline
MIME-Version: 1.0

This is the text of the mail.
```

9. Class Skeletons

```php
<?php
/**
 * Zend Framework
 *
 * LICENSE
 *
 * This source file is subject to the new BSD license that is bundled
 * with this package in the file LICENSE.txt.
 * It is also available through the world-wide-web at this URL:
 * http://framework.zend.com/license/new-bsd
 * If you did not receive a copy of the license and are unable to
 * obtain it through the world-wide-web, please send an email
 * to license@zend.com so we can send you a copy immediately.
 *
 * @category   Zend
 * @package    Zend_Mail
 * @subpackage Transport
 * @copyright  Copyright (c) 2005-2010 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 * @version    $Id$
 */

/**
 * @namespace
 */
namespace Zend\Mail\Transport;
use Zend\Config\Config;

/**
 * File transport
 *
 * Class for saving outgoing emails in filesystem
 *
 * @uses       \Zend\Mail\Transport\AbstractTransport
 * @uses       \Zend\Mail\Transport\Exception
 * @category   Zend
 * @package    Zend_Mail
 * @subpackage Transport
 * @copyright  Copyright (c) 2005-2010 Zend Technologies USA Inc. (http://www.zend.com)
 * @license    http://framework.zend.com/license/new-bsd     New BSD License
 */
class File extends AbstractTransport
{
    /**
     * Target directory for saving sent email messages
     *
     * @var string
     */
    protected $_path;

    /**
     * A callback function generating a file name
     *
     * @var string|array|Closure
     */
    protected $_callback;

    /**
     * Constructor
     *
     * @param  array|\Zend\Config\Config $options OPTIONAL (Default: null)
     * @return void
     */
    public function __construct($options = null)
    {
```

```
        if ($options instanceof Config) {
            $options = $options->toArray();
        } elseif (!is_array($options)) {
            $options = array();
        }

        // Make sure we have some defaults to work with
        if (!isset($options['path'])) {
            $options['path'] = sys_get_temp_dir();
        }
        if (!isset($options['callback'])) {
            $options['callback'] = $this->getDefaultCallback();
        }

        $this->setOptions($options);
    }

    /**
     * Sets options
     *
     * @param  array $options
     * @return void
     * @throws \Zend\Mail\Transport\Exception on wrong type of options
     */
    public function setOptions(array $options)
    {
        if (isset($options['path'])) {
            $this->_path = $options['path'];
        }
        if (isset($options['callback'])) {
            $this->_callback = $options['callback'];
        }
    }

    /**
     * Saves e-mail message to a file
     *
     * @return void
     * @throws \Zend\Mail\Transport\Exception on not writable target directory
     * @throws \Zend\Mail\Transport\Exception on file_put_contents() failure
     */
    protected function _sendMail()
    {
        $file = $this->_path . DIRECTORY_SEPARATOR . call_user_func($this->_callback, $this);

        if (!is_writable(dirname($file))) {
            throw new Exception('Target directory ' . dirname($file)
                . ' does not exist or not writable ');
        }

        $email = $this->header . $this->EOL . $this->body;

        if (!file_put_contents($file, $email)) {
            throw new Exception('Unable to send mail');
        }
    }

    /**
     * Returns the default callback for generating file names
     *
     * @return Closure
     */
    public function getDefaultCallback()
    {
        return function($transport) {
            return 'ZendMail_' . time() . '_' . mt_rand() . '.tmp';
```

```
            };
        }
    }
```

]]></ac:plain-text-body></ac:macro>
]]></ac:plain-text-body></ac:macro>