Contents

- 1 Introduction
  - 1.1 PHP Mailer
  - 1.2 PHP Mailer Windows
  - 1.3 Gmail TLS
  - 1.4 mailer_app.php
- 2 See Also

Introduction

PHP Mailer

Example setting in switch.conf.xml for sending email with PHP mailer. Make sure the path to PHP and mailer_app.php is correct.

```
<param name="mailer-app" value="/usr/bin/php /usr/local/www/mailer_app.php"/>
<param name="mailer-app-args" value=""/>
```

PHP Mailer Windows

Example setting in switch.conf.xml for sending email with PHP mailer. Make sure the path to PHP file is correct.

```
<param name="mailer-app" value="c:/php"/>
<param name="mailer-app-args" value="/fusionpbx/www/secure/mailer_app.php"/>
```

Gmail TLS

Example Gmail TLS setttings.

```
$tmp_smtphost   = "smtp.gmail.com:465";
$tmp_smtpsecure = "tls";  //options "", "tls", "ssl"
$tmp_smtpauth = "true";     // SMTP authentication: true or false
$tmp_smtpusername = "demo@gmail.com"; //Use your gmail email address here.
$tmp_smtppassword = "1234"; //Use your gmail email password here.
$tmp_smtpfrom = "demo@gmail.com"; //Use a valid email address here.
$tmp_smtpfromname = "voicemail"; //Use whatever name you desire here.
```

mailer_app.php

Save the following PHP code as mailer_app.php. Requires PHPMailer class files class.phpmailer.php and class.smtp.php inside the same directory. Make sure to update the path to PHP.

```
<?php
/* $Id$ */
/*
      mailer_app.php
      Copyright (C) 2008 Mark J Crane
      All rights reserved.

      Redistribution and use in source and binary forms, with or without
      modification, are permitted provided that the following conditions are met:

      1. Redistributions of source code must retain the above copyright notice,
         this list of conditions and the following disclaimer.

      2. Redistributions in binary form must reproduce the above copyright
         notice, this list of conditions and the following disclaimer in the
         documentation and/or other materials provided with the distribution.

      THIS SOFTWARE IS PROVIDED ``AS IS      AND ANY EXPRESS OR IMPLIED WARRANTIES,
      INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY
      AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE
      AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY,
      OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
      SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
      INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
      CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
      ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
      POSSIBILITY OF SUCH DAMAGE.
*/

$tmp_smtphost   = "";
$tmp_smtpsecure = "";  //options "", "tls", "ssl"
$tmp_smtpauth = "";     // SMTP authentication: true or false
$tmp_smtpusername = "";
$tmp_smtppassword = "";
$tmp_smtpfrom = "";
$tmp_smtpfromname = "";


ini_set(max_execution_time,900); //15 minutes
ini_set('memory_limit', '96M');
$fd = fopen("php://stdin", "r");

$email = file_get_contents ("php://stdin");
fclose($fd);
$fp = fopen("/tmp/voicemailtoemail.txt", "w");

ob_end_clean();
ob_start();


//get main header and body
    $tmparray = split("\n\n", $email);
    $mainheader = $tmparray[0];
    $maincontent = substr($email, strlen($mainheader), strlen($email));

//get the boundary
```

```
    $tmparray = split("\n", $mainheader);
    $contenttmp = $tmparray[1]; //Content-Type: multipart/mixed; boundary="XXXX_boundary_XXXX"
    $tmparray = split('; ', $contenttmp); //boundary="XXXX_boundary_XXXX"
    $contenttmp = $tmparray[1];
    $tmparray = split('=', $contenttmp); //"XXXX_boundary_XXXX"
    $boundary = $tmparray[1];
    $boundary = trim($boundary,'"');
    //echo "boundary: $boundary\n";

//put the main headers into an array
    $mainheaderarray = split("\n", $mainheader);
    //print_r($mainheaderarray);
    foreach ($mainheaderarray as $val) {
        $tmparray = split(': ', $val);
        //print_r($tmparray);
        $var[$tmparray[0]] = trim($tmparray[1]);
    }

    $var['To'] = str_replace("<", "", $var['To']);
    $var['To'] = str_replace(">", "", $var['To']);

    echo "To: ".$var['To']."\n";
    echo "From: ".$var['From']."\n";
    echo "Subject: ".$var['Subject']."\n";
    //print_r($var);
    echo "\n\n";


// split mime type multi-part into each part
    $maincontent = str_replace($boundary."--", $boundary, $maincontent);
    $tmparray = split("--".$boundary, $maincontent);

// loop through each mime part
    $i=0;
    foreach ($tmparray as $mimepart) {

      $mimearray = split("\n\n", $mimepart);
      $subheader = $mimearray[0];
      $headermimearray = split("\n", trim($subheader));

      $x=0;
      foreach ($headermimearray as $val) {
          if(stristr($val, ':') === FALSE) {
              $tmparray = split('=', $val); //':' not found
              if (trim($tmparray[0]) == "boundary") {
                  $subboundary = $tmparray[1];
                  $subboundary = trim($subboundary,'"');
                  //echo "subboundary: ".$subboundary."\n";
              }
          }
          else {
              $tmparray = split(':', $val); //':' found
          }

          //print_r($tmparray);
          $var[trim($tmparray[0])] = trim($tmparray[1]);
      }
      //print_r($var);


      $contenttypearray = split(' ', $headermimearray[0]);
```

```
        if ($contenttypearray[0] == "Content-Type:") {
            $contenttype = trim($contenttypearray[1]);

            switch ($contenttype) {
            case "multipart/alternative;":

                //echo "type: ".$contenttype."\n";
                $content = trim(substr($mimepart, strlen($subheader), strlen($mimepart)));

                $content = str_replace($subboundary."--", $subboundary, $content);
                $tmpsubarray = split("--".$subboundary, $content);
                foreach ($tmpsubarray as $mimesubsubpart) {

                  $mimesubsubarray = split("\n\n", $mimesubsubpart);
                  $subsubheader = $mimesubsubarray[0];

                  $headersubsubmimeearray = split("\n", trim($subsubheader));
                  $subsubcontenttypearray = split(' ', $headersubsubmimeearray[0]);
                  //echo "subsubcontenttypearray[0] ".$subsubcontenttypearray[0]."\n";

                  if ($subsubcontenttypearray[0] == "Content-Type:") {
                      $subsubcontenttype = trim($subsubcontenttypearray[1]);
                      switch ($subsubcontenttype) {
                      case "text/plain;":
                        $textplain = trim(substr($mimesubsubpart, strlen($subsubheader), strlen($
                        //echo "text/plain: $textplain\n";
                        break;
                      case "text/html;":
                        $texthtml = trim(substr($mimesubsubpart, strlen($subsubheader), strlen($m
                        //echo "text/html: $texthtml\n";
                        break;
                      }
                  } //end if


                } //end foreach

                break;
            case "audio/wav;":
                //echo "type: ".$contenttype."\n";
                $strwav = trim(substr($mimepart, strlen($subheader), strlen($mimepart)));
                //echo "\n*** begin wav ***\n".$strwav."\n*** end wav ***\n";
                break;

            }//end switch
        } //end if

        $i++;

      } //end foreach


  //send the email

       include "class.phpmailer.php";
       include "class.smtp.php"; // optional, gets called from within class.phpmailer.php if not

       $mail = new PHPMailer();

       $mail->IsSMTP(); // set mailer to use SMTP
       if ($tmp_smtpauth == "true") {
        $mail->SMTPAuth = $tmp_smtpauth; //turn on/off SMTP authentication
```

```
    }
    $mail->Host   = $tmp_smtphost;
    if (strlen($tmp_smtpsecure)>0) {
        $mail->SMTPSecure = $tmp_smtpsecure;
    }
    if ($tmp_smtpusername) {
         $mail->Username = $tmp_smtpusername;
         $mail->Password = $tmp_smtppassword;
    }
    $mail->SMTPDebug  = 2;

    $mail->From       = $tmp_smtpfrom;
    $mail->FromName   = $tmp_smtpfromname;
    $mail->Subject    = $var['Subject'];
    $mail->AltBody    = $textplain;   // optional, comment out and test
    $mail->MsgHTML($texthtml);

    $tmp_to = $var['To'];
    $tmp_to = str_replace(";", ",", $tmp_to);
    $tmp_to_array = split(",", $tmp_to);
    foreach($tmp_to_array as $tmp_to_row) {
            if (strlen($tmp_to_row) > 0) {
                    $mail->AddAddress($tmp_to_row);
            }
    }
    if (strlen($strwav) > 0) {
            $filename='voicemail.wav'; $encoding = "base64"; $type = "audio/wav";
            $mail->AddStringAttachment(base64_decode($strwav),$filename,$encoding,$type);
    }
    unset($strwav);

    if(!$mail->Send()) {
       echo "Mailer Error: " . $mail->ErrorInfo;
    }
    else {
       echo "Message sent!";
    }

$content = ob_get_contents(); //get the output from the buffer
ob_end_clean(); //clean the buffer

fwrite($fp, $content);
fclose($fp);

?>
```

See Also

- mod_voicemail