## Examples using PHPMailer

### 1. Advanced Example

This demonstrates sending multiple email messages with binary attachments from a MySQL database using multipart/alternative messages.

require 'PHPMailerAutoload.php';  
  
$mail = new PHPMailer();  
  
$mail->setFrom('list@example.com', 'List manager');  
$mail->Host = 'smtp1.example.com;smtp2.example.com';  
$mail->Mailer = 'smtp';  
  
@mysqli\_connect('localhost','root','password');  
@mysqli\_select\_db("my\_company");  
$query = "SELECT full\_name, email, photo FROM employee";  
$result = @mysqli\_query($query);  
  
while ($row = mysqli\_fetch\_assoc($result))  
{  
 // HTML body  
 $body = "Hello <font size=\"4\">" . $row['full\_name'] . "</font>, <p>";  
 $body .= "<i>Your</i> personal photograph to this message.<p>";  
 $body .= "Sincerely, <br>";  
 $body .= "phpmailer List manager";  
  
 // Plain text body (for mail clients that cannot read HTML)  
 $text\_body = 'Hello ' . $row['full\_name'] . ", \n\n";  
 $text\_body .= "Your personal photograph to this message.\n\n";  
 $text\_body .= "Sincerely, \n";  
 $text\_body .= 'phpmailer List manager';  
  
 $mail->Body = $body;  
 $mail->AltBody = $text\_body;  
 $mail->addAddress($row['email'], $row['full\_name']);  
 $mail->addStringAttachment($row['photo'], 'YourPhoto.jpg');  
  
 if(!$mail->send())  
 echo "There has been a mail error sending to " . $row['email'] . "<br>";  
  
 // Clear all addresses and attachments for next loop  
 $mail->clearAddresses();  
 $mail->clearAttachments();  
}

### 2. Extending PHPMailer

Extending classes with inheritance is one of the most powerful features of object-oriented programming. It allows you to make changes to the original class for your own personal use without hacking the original classes, and it's very easy to do:

Here's a class that extends the phpmailer class and sets the defaults for the particular site:

PHP include file: my\_phpmailer.php

require 'PHPMailerAutoload.php';  
  
class my\_phpmailer extends PHPMailer {  
 // Set default variables for all new objects  
 public $From = 'from@example.com';  
 public $FromName = 'Mailer';  
 public $Host = 'smtp1.example.com;smtp2.example.com';  
 public $Mailer = 'smtp'; // Alternative to isSMTP()  
 public $WordWrap = 75;  
  
 // Replace the default debug output function  
 protected function edebug($msg) {  
 print('My Site Error');  
 print('Description:');  
 printf('%s', $msg);  
 exit;  
 }  
  
 //Extend the send function  
 public function send() {  
 $this->Subject = '[Yay for me!] '.$this->Subject;  
 return parent::send()  
 }  
  
 // Create an additional function  
 public function do\_something($something) {  
 // Place your new code here  
 }  
}

Now here's a normal PHP page in the site, which will have all the defaults set above:

require 'my\_phpmailer.php';  
  
// Instantiate your new class  
$mail = new my\_phpmailer;  
  
// Now you only need to add the necessary stuff  
$mail->addAddress('josh@example.com', 'Josh Adams');  
$mail->Subject = 'Here is the subject';  
$mail->Body = 'This is the message body';  
$mail->addAttachment('c:/temp/11-10-00.zip', 'new\_name.zip'); // optional name  
  
if(!$mail->send())  
{  
 echo 'There was an error sending the message';  
 exit;  
}  
  
echo 'Message was sent successfully';