This is built for PHP Mailer 1.72 and was not tested with any previous version. It was developed under PHP 4.3.11 (E\_ALL). It works under PHP 5 and 5.1 with E\_ALL, but not in Strict mode due to var deprecation (but then neither does PHP Mailer either!). It follows the RFC 1939 standard explicitly and is fully commented.

With that noted, here is how to implement it:

I didn't want to modify the PHP Mailer classes at all, so you will have to include/require this class along with the base one. It can sit quite happily in the phpmailer directory.

When you need it, create your POP3 object

Right before I invoke PHP Mailer I activate the POP3 authorisation. POP3 before SMTP is a process whereby you login to your web hosts POP3 mail server BEFORE sending out any emails via SMTP. The POP3 logon 'verifies' your ability to send email by SMTP, which typically otherwise blocks you. On my web host (Pair Networks) a single POP3 logon is enough to 'verify' you for 90 minutes. Here is some sample PHP code that activates the POP3 logon and then sends an email via PHP Mailer:

<?php

$pop->authorise('pop3.example.com', 110, 30, 'mailer', 'password', 1);

$mail = new PHPMailer(); $mail->SMTPDebug = 2; $mail->isSMTP();

$mail->isHTML(false); $mail->Host = 'relay.example.com';

$mail->From = 'mailer@example.com';

$mail->FromName = 'Example Mailer';

$mail->Subject = 'My subject';

$mail->Body = 'Hello world';

$mail->addAddress('rich@corephp.co.uk', 'Richard Davey');

if (!$mail->send()) {

echo $mail->ErrorInfo;

}

?>

The PHP Mailer parts of this code should be obvious to anyone who has used PHP Mailer before. One thing to note - you almost certainly will not need to use SMTP Authentication \*and\* POP3 before SMTP together. The Authorisation method is a proxy method to all of the others within that class. There are connect, Logon and disconnect methods available, but I wrapped them in the single Authorisation one to make things easier.

The Parameters

The authorise parameters are as follows:

$pop->authorise('pop3.example.com', 110, 30, 'mailer', 'password', 1);

1. pop3.example.com - The POP3 Mail Server Name (hostname or IP address)

2. 110 - The POP3 Port on which to connect (default is usually 110, but check with your host)

3. 30 - A connection time-out value (in seconds)

4. mailer - The POP3 Username required to logon

5. password - The POP3 Password required to logon

6. 1 - The class debug level (0 = off, 1+ = debug output is echoed to the browser)

Final Comments + the Download

1) This class does not support APOP connections. This is only because I did not have an APOP server to test with, but if you'd like to see that added just contact me.

2) Opening and closing lots of POP3 connections can be quite a resource/network drain. If you need to send a whole batch of emails then just perform the authentication once at the start, and then loop through your mail sending script. Providing this process doesn't take longer than the verification period lasts on your POP3 server, you should be fine. With my host that period is 90 minutes, i.e. plenty of time.

3) If you have heavy requirements for this script (i.e. send a LOT of email on a frequent basis) then I would advise seeking out an alternative sending method (direct SMTP ideally). If this isn't possible then you could modify this class so the 'last authorised' date is recorded somewhere (MySQL, Flat file, etc) meaning you only open a new connection if the old one has expired, saving you precious overhead.

4) There are lots of other POP3 classes for PHP available. However most of them implement the full POP3 command set, where-as this one is purely for authentication, and much lighter as a result. However using any of the other POP3 classes to just logon to your server would have the same net result. At the end of the day, use whatever method you feel most comfortable with.

Download

My thanks to Chris Ryan for the inspiration (even if indirectly, via his SMTP class)