By Filip
Published: 2007-08-31 13:31

Sending Email From PHP

This tutorial shows how you can easily send email messages from PHP-enabled web server. It can be utilized for processing forms data, sending alerts, etc. It also explains a bit about email headers formatting.

System Configuration

Obviously you need to have a web server installed together with PHP. You also need to make sure that your operating system has some email agent installed (for example Sendmail on Linux or SMTP service under IIS). And make sure that port 25 is not being blocked. This is required even if you relay messages via second server.

PHP has mail function enabled by default. However you can specify the additional options in 'php.ini' file in [mail function] section (line # around 700).

Syntax

Here is the full syntax of mail() command:

```
bool mail ( string $email_address_to, string $subject, string $message_contents [, string $additional_headers [, string $additional_parameters]]
);
```

The first 3 parameters do not require any explanation: recipient email address, subject of the message, contents of the message. ***$additional_headers*** parameter includes email headers which I will list later. ***$additional_parameters*** is used to pass options specific for the program used for sending email (such as sendmail) and is rarely used. You need to check the man pages of sendmail for the list of options.

The absolute minimum requires you to specify these:

```
mail ( $email_address_to , $subject, $message_contents );
```

However it is recomended to define additional headers for the compliance reasons. Email message must be compatible with ***RFC #822***.

Example below shows some email headers which can be used:

- ***From:*** The sender's email address.
- ***Reply-To:*** The email address where replies should be sent to.
- ***Return-Path:*** This is used in case message is not delivered and must be returned.
- ***Subject:*** Subject of the email.
- ***CC:*** Carbon Copy. A comma seperated list of more recipients that will be seen by all other recipients.
- ***BCC:*** Blind Carbon Copy. A comma seperated list of more recipients that will not be seen by any other recipients.
- ***Content-type:*** Defines the MIME type of the message.
- ***X-Mailer:*** Specifies email client used to send the message.

Important thing to remember is that header name are case-sensitive and that each header must be ended with ***return*** and ***newline*** characters.

Each message is routed via minimum 2 email servers (sender and recipient). However there might be more email servers in the route such as relay servers, antiSPAM servers, etc. Each one of them will add its own headers to your message. You also have to keep in mind that your web server might, depending on the settings, swap some header fields such as ***Return-Path***. Note that this field does not affect the reply-to address. ***Example***

Here is a simple example that will send an email message:

```
<?php
$email_address_to = "recipient@demo.com";
$subject = "Test email subject";
$message_contents = "Hi! This is the content of the test message.";
$header = "From: sender@demo.com\r\n";
$header .= "Reply-To: sender@demo.com\r\n";
$header .= "Return-Path: sender@demo.com\r\n";
mail($email_address_to,$subject,$message_contents,$header);
?>
```

Full Example

This example shows more organized structure with more options and more headers:

```
<?php
// --- CONFIG PARAMETERS --- //
//
   $email_recipient = "recipient@demo.com";
   $email_sender = "Sender Name";
   $email_return_to = "sender@demo.com";
   $email_content_type = "text/html; charset=us-ascii";
   $email_client = "PHP/" . phpversion();
//
// ------------------------- //

// --- DEFINE HEADERS --- //
//
   $email_header = "From: " . $email_sender . "\r\n";
   $email_header .= "Reply-To: " . $email_return_to . "\r\n";
   $email_header .= "Return-Path: " . $email_return_to . "\r\n";
   $email_header .= "Content-type: " . $email_content_type . "\r\n";
   $email_header .= "X-Mailer: " . $email_client . "\r\n";
//
// ---------------------- //

// --- SUBJECT AND CONTENTS --- //
//
   $email_subject = "Test email subject";
   $email_contents = "<html>";
   $email_contents .= "<h2>Test Email</h2>";
   $email_contents .= "<br><b>Sender: " . $email_sender;
   $email_contents .= "<br><b>Recipient: " . $email_recipient;
   $email_contents .= "</html>";
//
// ---------------------------- //

$email_result = mail($email_recipient, $email_subject, $email_contents, $email_header);
```

```
if ($email_result) echo "Email has been sent!";
else echo "Email has failed!";
?>
```

- ***Note:*** The Windows implementation of mail() differs in many ways from the Unix implementation. First, it doesn't use a local binary for composing messages but only operates on direct sockets which means a MTA is needed listening on a network socket (which can either on the localhost or a remote machine). Second, the custom headers like From:, Cc:, Bcc: and Date: are not interpreted by the MTA in the first place, but are parsed by PHP. As such, the to parameter should not be an address in the form of "Something <someone@example.com>". The mail command may not parse this properly while talking with the MTA.
- ***Note:*** It is worth mentioning that the mail() function is not suitable for larger volumes of email in a loop. This function opens and closes an SMTP socket for each email, which is not very efficient.