

Quick guide to PHP session management, 03/Mar/2006

This document gives a quick overview to PHP session management. Note that use given as below will work without cookies being enabled in the browser.

Entry page: xxx_login.php

```
// Identify the session if any.
session_start();

if ( !isset($_SESSION['USERID']) || !isset($_SESSION['UPASSW']) )
{
//    Interrogate to get login tokens, say USERID and UPASSW. Then ...
    $_SESSION['USERID'] = $USERID;
    $_SESSION['UPASSW'] = $UPASSW;
    $_SESSION['ANYOTHERSESSIONVARIABLEYOUWANT'] = ""; // Declare
//    You can store arrays and all sorts of things if you want.
}

Header("Location: xxx_start.php");
```

Next page: xxx_start.php (a typical working page)

```
// identify the session.

session_start();

// $_SESSION['USERID'] contains the userid etc.

$_SESSION['ANYOTHERSESSIONVARIABLEYOUWANT'] = 42;

Header("Location: xxx_end.php");
```

Exit page: xxx_end.php (a typical ending page)

```
// identify the session.

session_start();

echo "<HTML><BODY><P>The answer is .."
    .     $_SESSION['ANYOTHERSESSIONVARIABLEYOUWANT']
    .     "</P></BODY></HTML>\n";

// Destroy the session.

session_destroy();
```

Notes:

You don't have to do anything else at all. The session id is managed entirely internally through a code used to generate the session file, usually in APACHE_USER/tmp on the server. The function session_destroy() deletes the file. If a session terminates without going through session_destroy(), it times out and self-destructs after some specified interval. If you want to prevent hacks and script kiddies, load some codes into \$_SESSION and check them throughout the site. PHP tracks the session id without cookies. If you use REQUIRE files, make sure they have

the .php suffix so people can't see them. It works like a charm.