S&P Technology

Resources - PHP 5.4 Replacement HTML Functions

In PHP 5.4 the default character set encoding for the **htmlspecialchars()**, **htmlentities()**, **html_entity_decode()** and **get_html_translation_table()** functions changed from ISO-8859-1 to UTF-8.

If the encoding is specified when the functions are used then they will continue to work as before when upgrading to PHP 5.4. However a lot of PHP code doesn't specify the encoding, and consequently causes problems on PHP 5.4. This will generally show as the functions returning empty strings because the input string contains character that are valid ISO-8859-1 but not valid UTF-8.

One option would be to go through all your PHP code and manually edit every use of the functions like this:

```php
<?php
// Replace this
$text = htmlspecialchars ($text);
// With this
$text = htmlspecialchars ($text, ENT_COMPAT | ENT_HTML401, 'ISO-8859-1');

// Replace this
$text = htmlentities ($text);
// With this
$text = htmlentities ($text, ENT_COMPAT | ENT_HTML401, 'ISO-8859-1');

// Replace this
$text = html_entity_decode ($text);
// With this
$text = html_entity_decode ($text, ENT_COMPAT | ENT_HTML401, 'ISO-8859-1');

// Replace this
$table = get_html_translation_table ($table);
// With this
$table = get_html_translation_table ($table, ENT_COMPAT | ENT_HTML401, 'ISO-8859-1');
?>
```

With large projects that would be a nightmare!

The work-around here still requires editing of all your code, but it can be done with a global search-and-replace which is a lot quicker.

Create a new PHP file named **htmlXfunctions.php** with the following content:

```php
<?php
// Alternative functions to resolve default character set issues in PHP 5.4
// From http://www.sp-tech.co.uk/resources-php54.php

function htmlXspecialchars ($text, $flags = NULL, $enc = 'ISO-8859-1', $double = true) {
    if ($flags == NULL) $flags = (ENT_COMPAT | ENT_HTML401);
    return htmlspecialchars ($text, $flags, $enc, $double);
}

function htmlXentities ($text, $flags = NULL, $enc = 'ISO-8859-1', $double = true) {
    if ($flags == NULL) $flags = (ENT_COMPAT | ENT_HTML401);
    return htmlentities ($text, $flags, $enc, $double);
}

function htmlX_entity_decode ($text, $flags = NULL, $enc = 'ISO-8859-1') {
    if ($flags == NULL) $flags = (ENT_COMPAT | ENT_HTML401);
    return html_entity_decode ($text, $flags, $enc);
}

function get_htmlX_translation_table ($table, $flags = NULL, $enc = 'ISO-8859-1') {
    if ($flags == NULL) $flags = (ENT_COMPAT | ENT_HTML401);
    return get_html_translation_table ($table, $flags, $enc);
}
?>
```

In a file that will be run once only no matter what page is displayed (the configuration file could be a good place) add a require() call to the new file:

```php
<?php require ('../routines/htmlXfunctions.php'); ?>
```

If the path could vary you could make it relative to the calling PHP file with something like this:

```php
<?php require (dirname(__FILE__).'/../routines/htmlXfunctions.php'); ?>
```

You now need to do a global search-and-replace on every PHP file (apart from **htmlXfunctions.php**) to replace the original functions with our new ones. Most programmer's editors such as Notepad++ will do that, as will Dreamweaver.

```
<?php
// Search for
htmlspecialchars (
// Replace with
htmlXspecialchars (

// Search for
htmlentities (
// Replace with
htmlXentities (

// Search for
html_entity_decode (
// Replace with
htmlX_entity_decode (

// Search for
get_html_translation_table (
// Replace with
get_htmlX_translation_table (

// Use the "ignore whitespace differences" in the global search and replace
// to allow for spaces or no spaces between the function and the open bracket
?>
```

This PHP script is offered on an 'as-is' basis with no guarantee that it will work correctly etc. Please carry out your own tests before using it. No liability is accepted for any errors or damage however caused. We would appreciate a link to this website if you find it useful.

Back to Resources section

Copyright © 2014 S&P Technology