UPGRADING PHP SCRIPTS AND MYSQL DATA BASES AND TABLES TO WORK WITH INTERNATIONAL CHARS ON PHP7 UP FROM 5

By Sergio Reyes
www.sreyes.org – Dic. 12, 2016

1. Just when people who need to work with any other language but English had figured it, the PHP “community” decided to improve things and change defaults to use character code set UTF8. This has been a source of great frustration, which can be verified by doing a simple Internet search for this issue.

The PHP group itself didn’t have the foresight of isolating this change and explaining it in simple terms for people to adjust their old scripts to work without getting the infamous, ominous replacement character <?> to appear every time an international character, such as a vowel with accent, should be displayed.

Most web hosting companies, including 1and1.com, which I use, are also less than helpful in this area. 1and1.com, for instance, had a great “business”solution to this technical issue: *to maintain older php versions, “provided that customers pay them $7 per month or $94 a year” for the life of the contract.* In my judgement, this is an utterly unfair solution that benefits them moneywise but hurts their customers.

Here is what I did to solve my legacy, international characters databases and scrips to make it work, no thanks to my web host 1and1.

1. Make a backup copy of your full database and all your scripts.

2. Change your php version to 7 in the control panel provided by your web host.

3. Transform all tables and fields within it to UTF8.

Most web hosts provide access to phpmyadmin. There you can issue mySQL commands.

Use the `ALTER DATABASE` and `ALTER TABLE` commands.

```
ALTER DATABASE databasename CHARACTER SET utf8 COLLATE utf8_unicode_ci;

ALTER TABLE tablename CONVERT TO CHARACTER SET utf8 COLLATE utf8_unicode_ci;
```

4. It is most important to edit your database connection script to include simple php sintax to force UTF8 flow between your browser and the database:

```
<?php
    $host_name  = "your-host";
    $database   = "your-db";
    $user_name  = "your-usr-name";
    $password   = "your-password";

    $connect = mysqli_connect($host_name, $user_name, $password, $database);

    if(mysqli_connect_errno())
    {
    echo '<p>Could not connect to MYSQL server.</p>';
    }

# DEAL WITH CHARACTER SET TO UTF8

/* change character set to utf8 */
if (!mysqli_set_charset($connect, "utf8")) {
    printf("Error loading character set utf8: %s\n", mysqli_error($connect));
    exit();
}
# edited to work on PHP7
?>
```

5. Edit some key commands on your reading, writing and updating scripts. Modify commands that no longer work on php 7 such as:

```
mysql_query($sql); to mysqli_query($connect, $sql);
mysql_fecth_array($mysql_result); to mysqli_fetch_array($mysql_result);
mysql_num_rows($mysql_result) to mysqli_num_rows($mysql_result);
mysql_close(); to mysqli_close();
```

The variables here should all refer to the variable names you used, of course.

6. Note that if you include hard-coded international characters in your script itself such as “señor”, this will not display correctly unless you **save your script itself in UTF8** (if you are using version 7 of Windows Notepad, for instance, the default is ANSI. You have to select UTF8 when you save.)

7. Test, test, and test.