De anrop som funnits i PHP sedan version 1.99 för att ansluta till databasen mysql har till slut tagits bort i PHP 7. Det är säkert klokt då det finns nyare och bättre sätt att ansluta till databasen genom mysqli (funktioner eller objekt), PDO (flera databaser) med fler. Men om man nu har någon gammal rutin som fungerar baserad på den gamla koden... skall man portera databas-koden i denna, med de risker för buggar och fel det innebär?

Anropen till mysqli är mycket lika mysql, så en modul med enkla funktioner som översätter de skillnader som finns kan funka att slänga in för rutiner som använder en delmängd av de gamla mysql-anropen. Sagt och gjort, den simpla modulen mysql_compat.php skapad för de fall där man inte vill eller vågar portera koden - men någon ny kod skall man inte skriva mot denna modul. Använd genom raden: require_once("mysql_compat.php"); först i Din gamla modul.

Koden är såpass kort och simpel så behöver Du köra en gammal rutin som använder mysql_connect, mysql_query, mysql_fetch_array och några till under PHP7 så varsågod, använd koden nedan.

Naturligtvis finna andra alternativ, som att ha en äldre version av PHP installerad för att kunna köra gammal kod, men dels kan det vara besvärligt att skilja på moduler med ändelse .php som skall köras med PHP5 eller PHP7, så kan bibliotek krocka med mer - och man missar de hastighetsfördelar PHP7 ger om man kör under PHP5.

<&php // PHP7 mysql_ compatibility functions, simple wrappers by Erik Ltz @ eldata.se // // License: You have the right to use the code, please send improvements back :-) // erik at eldata.se // // FIRST: new code *SHOULD* use the supported database functions that has // been available in PHP for ages. Learn them, use them. // // This compatibility wrapper is *ONLY* to get old legacy code using the old // mysql_ functions in standard simple ways to work by calling the mysqli_ // equivalent. Not every case will work, just basic stuff like // mysql_connect(); // $res=mysql_query(); // $row=mysql_fetch_assoc($res); // // Again: If using special parameters and advanced things, this wrapper // module simply *will not cut it*. // // Global data ---------- $_mysql_glob=array(); // Connection ---------- function mysql_connect($host, $user, $pass, $new_link=false, $client_flags=0) { global $_mysql_glob; $_mysql_glob['host']=$host; $_mysql_glob['user']=$user; $_mysql_glob['pass']=$pass; $_mysql_glob['link']=mysqli_connect($_mysql_glob['host'], $_mysql_glob['user'], $_mysql_glob['pass']); } // Reestablish connection with other database function mysql_select_db($db) { global $_mysql_glob; $_mysql_glob['db']=$db;

$_mysql_glob['link']=mysqli_connect($_mysql_glob['host'], $_mysql_glob['user'], $_mysql_glob['pass'], $_mysql_glob['db']); } // Query and fetch data ---------- function mysql_query($query, $link=0) { global $_mysql_glob; return mysqli_query($link ? $link : $_mysql_glob['link'], $query); } function mysql_fetch_assoc($res) { return mysqli_fetch_assoc($res); } // Connection errors ---------- function mysql_connect_errno() { return mysqli_connect_errno(); } function mysql_connect_error() { return mysqli_connect_error(); } // Errors ---------- function mysql_errno($link=0) { global $_mysql_glob; return mysqli_errno($link ? $link : $_mysql_glob['link']); } function mysql_error($link=0) { global $_mysql_glob; return mysqli_error($link ? $link : $_mysql_glob['link']); } ?> Naturligtvis finns ett antal fler mysql-funktioner som skulle kunna översättas - dessa är de vanligaste och funkade i en gammal rutin vi ville köra. Hör gärna av Dig om Ni behöver hjälp att anpassa kod för PHP 7!