Mysql

<?php

// If this file is not included from the MMHTTPDB possible hacking problem.

if (!function\_exists('create\_error')){

die();

}

define('MYSQL\_NOT\_EXISTS', create\_error("Your PHP server doesn't have the MySQL module loaded or you can't use the mysql\_(p)connect functions."));

define('CONN\_NOT\_OPEN\_GET\_TABLES', create\_error('The Connection could not be opened when trying to retrieve the tables.'));

define('CONN\_NOT\_OPEN\_GET\_DB\_LIST', create\_error('The Connection could not be opened when trying to retrieve the database list.'));

if (!function\_exists('mysql\_connect') || !function\_exists('mysql\_pconnect') || !extension\_loaded('mysql')){

echo MYSQL\_NOT\_EXISTS;

die();

}

// Now let's handle the crashes or any other PHP errors that we can catch

function KT\_ErrorHandler($errno, $errstr, $errfile, $errline) {

global $f, $already\_sent;

$errortype = array (

1 => "Error",

2 => "Warning",

4 => "Parsing Error",

8 => "Notice",

16 => "Core Error",

32 => "Core Warning",

64 => "Compile Error",

128 => "Compile Warning",

256 => "User Error",

512 => "User Warning",

1024=> "User Notice",

2048=> "E\_ALL",

2049=> "PHP5 E\_STRICT"

);

$str = sprintf("[%s]\n%s:\t%s\nFile:\t\t'%s'\nLine:\t\t%s\n\n", date('d-m-Y H:i:s'),(isset($errortype[@$errno])?$errortype[@$errno]:('Unknown '.$errno)),@$errstr,@$errfile,@$errline);

if (error\_reporting() != 0) {

@fwrite($f, $str);

if (@$errno == 2 && isset($already\_sent) && !$already\_sent==true){

$error = '<ERRORS>'."\n";

$error .= '<ERROR><DESCRIPTION>An Warning Type error appeared. The error is logged into the log file.</DESCRIPTION></ERROR>'."\n";

$error .= '</ERRORS>'."\n";

$already\_sent = true;

echo $error;

}

}

}

if ($debug\_to\_file){

$old\_error\_handler = set\_error\_handler("KT\_ErrorHandler");

}

class MySqlConnection

{

/\*

// The 'var' keyword is deprecated in PHP5 ... we will define these variables at runtime.

var $isOpen;

var $hostname;

var $database;

var $username;

var $password;

var $timeout;

var $connectionId;

var $error;

\*/

function MySqlConnection($ConnectionString, $Timeout, $Host, $DB, $UID, $Pwd)

{

$this->isOpen = false;

$this->timeout = $Timeout;

$this->error = '';

if( $Host ) {

$this->hostname = $Host;

}

elseif( preg\_match("/host=([^;]+);/", $ConnectionString, $ret) ) {

$this->hostname = $ret[1];

}

if( $DB ) {

$this->database = $DB;

}

elseif( preg\_match("/db=([^;]+);/", $ConnectionString, $ret) ) {

$this->database = $ret[1];

}

if( $UID ) {

$this->username = $UID;

}

elseif( preg\_match("/uid=([^;]+);/", $ConnectionString, $ret) ) {

$this->username = $ret[1];

}

if( $Pwd ) {

$this->password = $Pwd;

}

elseif( preg\_match("/pwd=([^;]+);/", $ConnectionString, $ret) ) {

$this->password = $ret[1];

}

}

function Open()

{

$this->connectionId = mysql\_connect($this->hostname, $this->username, $this->password);

if (isset($this->connectionId) && $this->connectionId && is\_resource($this->connectionId))

{

$this->isOpen = ($this->database == "") ? true : mysql\_select\_db($this->database, $this->connectionId);

}

else

{

$this->isOpen = false;

}

}

function TestOpen()

{

return ($this->isOpen) ? '<TEST status=true></TEST>' : $this->HandleException();

}

function Close()

{

if (is\_resource($this->connectionId) && $this->isOpen)

{

if (mysql\_close($this->connectionId))

{

$this->isOpen = false;

unset($this->connectionId);

}

}

}

function GetTables($table\_name = '')

{

$xmlOutput = "";

if ($this->isOpen && isset($this->connectionId) && is\_resource($this->connectionId)){

// 1. mysql\_list\_tables and mysql\_tablename are deprecated in PHP5

// 2. For backward compatibility GetTables don't have any parameters

if ($table\_name === ''){

$table\_name = @$\_POST['Database'];

}

//added backtick for handling reserved words and special characters

//http://dev.mysql.com/doc/refman/5.0/en/legal-names.html

$sql = ' SHOW TABLES FROM ' . $this->ensureTicks($table\_name) ;

$results = mysql\_query($sql, $this->connectionId) or $this->HandleException();

$xmlOutput = "<RESULTSET><FIELDS>";

// Columns are referenced by index, so Schema and

// Catalog must be specified even though they are not supported

$xmlOutput .= '<FIELD><NAME>TABLE\_CATALOG</NAME></FIELD>'; // column 0 (zero-based)

$xmlOutput .= '<FIELD><NAME>TABLE\_SCHEMA</NAME></FIELD>'; // column 1

$xmlOutput .= '<FIELD><NAME>TABLE\_NAME</NAME></FIELD>'; // column 2

$xmlOutput .= "</FIELDS><ROWS>";

if (is\_resource($results) && mysql\_num\_rows($results) > 0){

while ($row = mysql\_fetch\_array($results)){

$xmlOutput .= '<ROW><VALUE/><VALUE/><VALUE>' . $row[0]. '</VALUE></ROW>';

}

}

$xmlOutput .= "</ROWS></RESULTSET>";

}

return $xmlOutput;

}

function GetViews()

{

// not supported

return "<RESULTSET><FIELDS></FIELDS><ROWS></ROWS></RESULTSET>";

}

function GetProcedures()

{

// not supported

return "<RESULTSET><FIELDS></FIELDS><ROWS></ROWS></RESULTSET>";

}

function GetColumnsOfTable($TableName)

{

$xmlOutput = "";

//added backtick for handling reserved words and special characters

//http://dev.mysql.com/doc/refman/5.0/en/legal-names.html

$query = "DESCRIBE ".$this->ensureTicks($TableName);

$result = mysql\_query($query) or $this->HandleException();

if ($result)

{

$xmlOutput = "<RESULTSET><FIELDS>";

// Columns are referenced by index, so Schema and

// Catalog must be specified even though they are not supported

$xmlOutput .= "<FIELD><NAME>TABLE\_CATALOG</NAME></FIELD>"; // column 0 (zero-based)

$xmlOutput .= "<FIELD><NAME>TABLE\_SCHEMA</NAME></FIELD>"; // column 1

$xmlOutput .= "<FIELD><NAME>TABLE\_NAME</NAME></FIELD>"; // column 2

$xmlOutput .= "<FIELD><NAME>COLUMN\_NAME</NAME></FIELD>";

$xmlOutput .= "<FIELD><NAME>DATA\_TYPE</NAME></FIELD>";

$xmlOutput .= "<FIELD><NAME>IS\_NULLABLE</NAME></FIELD>";

$xmlOutput .= "<FIELD><NAME>COLUMN\_SIZE</NAME></FIELD>";

$xmlOutput .= "</FIELDS><ROWS>";

// The fields returned from DESCRIBE are: Field, Type, Null, Key, Default, Extra

while ($row = mysql\_fetch\_array($result, MYSQL\_ASSOC))

{

$xmlOutput .= "<ROW><VALUE/><VALUE/><VALUE/>";

// Separate type from size. Format is: type(size)

if (preg\_match("/(.\*)\((.\*)\)/", $row["Type"], $ret))

{

$type = $ret[1];

$size = $ret[2];

}

else

{

$type = $row["Type"];

$size = "";

}

// MySQL sets nullable to "YES" or "", so we need to set "NO"

$null = $row["Null"];

if ($null == "")

$null = "NO";

$xmlOutput .= "<VALUE>" . $row["Field"] . "</VALUE>";

$xmlOutput .= "<VALUE>" . $type . "</VALUE>";

$xmlOutput .= "<VALUE>" . $null . "</VALUE>";

$xmlOutput .= "<VALUE>" . $size . "</VALUE></ROW>";

}

mysql\_free\_result($result);

$xmlOutput .= "</ROWS></RESULTSET>";

}

return $xmlOutput;

}

function GetParametersOfProcedure($ProcedureName, $SchemaName, $CatalogName)

{

// not supported on MySQL

return '<RESULTSET><FIELDS></FIELDS><ROWS></ROWS></RESULTSET>';

}

function ExecuteSQL($aStatement, $MaxRows)

{

if ( get\_magic\_quotes\_gpc() )

{

$aStatement = stripslashes( $aStatement ) ;

}

$xmlOutput = "";

$result = mysql\_query($aStatement) or $this->HandleException();

if (isset($result) && is\_resource($result))

{

$xmlOutput = "<RESULTSET><FIELDS>";

$fieldCount = mysql\_num\_fields($result);

for ($i=0; $i < $fieldCount; $i++)

{

$meta = mysql\_fetch\_field($result);

if ($meta)

{

$xmlOutput .= '<FIELD';

$xmlOutput .= ' type="' . $meta->type;

$xmlOutput .= '" max\_length="' . $meta->max\_length;

$xmlOutput .= '" table="' . $meta->table;

$xmlOutput .= '" not\_null="' . $meta->not\_null;

$xmlOutput .= '" numeric="' . $meta->numeric;

$xmlOutput .= '" unsigned="' . $meta->unsigned;

$xmlOutput .= '" zerofill="' . $meta->zerofill;

$xmlOutput .= '" primary\_key="' . $meta->primary\_key;

$xmlOutput .= '" multiple\_key="'. $meta->multiple\_key;

$xmlOutput .= '" unique\_key="' . $meta->unique\_key;

$xmlOutput .= '"><NAME>' . $meta->name;

$xmlOutput .= '</NAME></FIELD>';

}

}

$xmlOutput .= "</FIELDS><ROWS>";

$row = mysql\_fetch\_assoc($result);

for ($i=0; $row && ($i < $MaxRows); $i++)

{

$xmlOutput .= "<ROW>";

foreach ($row as $key => $value)

{

$xmlOutput .= "<VALUE>";

$xmlOutput .= htmlspecialchars($value);

$xmlOutput .= "</VALUE>";

}

$xmlOutput .= "</ROW>";

$row = mysql\_fetch\_assoc($result);

}

mysql\_free\_result($result);

$xmlOutput .= "</ROWS></RESULTSET>";

}

return $xmlOutput;

}

function GetProviderTypes()

{

return '<RESULTSET><FIELDS></FIELDS><ROWS></ROWS></RESULTSET>';

}

function ExecuteSP($aProcStatement, $TimeOut, $Parameters)

{

return '<RESULTSET><FIELDS></FIELDS><ROWS></ROWS></RESULTSET>';

}

function ReturnsResultSet($ProcedureName)

{

return '<RETURNSRESULTSET status=false></RETURNSRESULTSET>';

}

function SupportsProcedure()

{

return '<SUPPORTSPROCEDURE status=false></SUPPORTSPROCEDURE>';

}

/\*

\* HandleException added by InterAKT for ease in database translation answer

\*/

function HandleException()

{

global $debug\_to\_file, $f;

$this->error = create\_error(' MySQL Error#: '. ((int)mysql\_errno()) . "\n\n".mysql\_error());

log\_messages($this->error);

die($this->error.'</HTML>');

}

function ensureTicks($inputSQL)

{

$outSQL = $inputSQL;

//added backtick for handling reserved words and special characters

//http://dev.mysql.com/doc/refman/5.0/en/legal-names.html

//only add ticks if not already there

$oLength = strlen($outSQL);

$bHasTick = false;

if (($oLength > 0) && (($outSQL[0] == "`") && ($outSQL[$oLength-1] == "`")))

{

$bHasTick = true;

}

if ($bHasTick == false)

{

$outSQL = "`".$outSQL."`";

}

return $outSQL;

}

function GetDatabaseList()

{

$xmlOutput = '<RESULTSET><FIELDS><FIELD><NAME>NAME</NAME></FIELD></FIELDS><ROWS>';

if (isset($this->connectionId) && is\_resource($this->connectionId)){

$dbList = mysql\_list\_dbs($this->connectionId);

while ($row = mysql\_fetch\_object($dbList))

{

$xmlOutput .= '<ROW><VALUE>' . $row->Database . '</VALUE></ROW>';

}

}else{

$this->error = CONN\_NOT\_OPEN\_GET\_DB\_LIST;

return $this->error;

}

$xmlOutput .= '</ROWS></RESULTSET>';

return $xmlOutput;

}

function GetPrimaryKeysOfTable($TableName)

{

$xmlOutput = '';

//added backtick for handling reserved words and special characters

//http://dev.mysql.com/doc/refman/5.0/en/legal-names.html

$query = "DESCRIBE ".$this->ensureTicks($TableName);

$result = mysql\_query($query) or $this->HandleException();

if ($result)

{

$xmlOutput = '<RESULTSET><FIELDS>';

// Columns are referenced by index, so Schema and

// Catalog must be specified even though they are not supported

$xmlOutput .= '<FIELD><NAME>TABLE\_CATALOG</NAME></FIELD>'; // column 0 (zero-based)

$xmlOutput .= '<FIELD><NAME>TABLE\_SCHEMA</NAME></FIELD>'; // column 1

$xmlOutput .= '<FIELD><NAME>TABLE\_NAME</NAME></FIELD>'; // column 2

$xmlOutput .= '<FIELD><NAME>COLUMN\_NAME</NAME></FIELD>';

$xmlOutput .= '<FIELD><NAME>DATA\_TYPE</NAME></FIELD>';

$xmlOutput .= '<FIELD><NAME>IS\_NULLABLE</NAME></FIELD>';

$xmlOutput .= '<FIELD><NAME>COLUMN\_SIZE</NAME></FIELD>';

$xmlOutput .= '</FIELDS><ROWS>';

// The fields returned from DESCRIBE are: Field, Type, Null, Key, Default, Extra

while ($row = mysql\_fetch\_array($result, MYSQL\_ASSOC))

{

if (strtoupper($row['Key']) == 'PRI'){

$xmlOutput .= '<ROW><VALUE/><VALUE/><VALUE/>';

// Separate type from size. Format is: type(size)

if (preg\_match("/(.\*)\((.\*)\)/", $row['Type'], $ret))

{

$type = $ret[1];

$size = $ret[2];

}

else

{

$type = $row['Type'];

$size = '';

}

// MySQL sets nullable to "YES" or "", so we need to set "NO"

$null = $row['Null'];

if ($null == '')

$null = 'NO';

$xmlOutput .= '<VALUE>' . $row['Field'] . '</VALUE>';

$xmlOutput .= '<VALUE>' . $type . '</VALUE>';

$xmlOutput .= '<VALUE>' . $null . '</VALUE>';

$xmlOutput .= '<VALUE>' . $size . '</VALUE></ROW>';

}

}

mysql\_free\_result($result);

$xmlOutput .= '</ROWS></RESULTSET>';

}

return $xmlOutput;

}

} // class MySqlConnection

?>