

AGK – host your own high scores using PHP\MySQL

This document has been put together to aid the AGK developer when integrating PHP scripts and MySQL into their AGK projects, to be able to control high scores. This is not a step by step detailed guide, more an overview that once you've read will give you the needed information to add this to your own games\apps.

We are using a high scores table as an example, although the principal is the same for any MySQL table you want to interact with for any purpose of storing\returning data in MySQL.

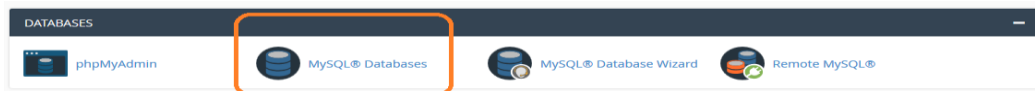
The key items you need are :

- A website
- A hosting package (which gives you MySQL and PHP)
- FTP access to your website folders
- ... not forgetting AGK itself :)

Assumption I have is that you will control the data being passed, i.e. validate etc, ensuring the data being sent conforms to the definition of the intended table field.

MySQL

I use GoDaddy for all my needs in terms of websites\PHP\MySQL etc. You'll need to set yourself up with a new MySQL. The examples I show are using the GoDaddy methods.



From here, create yourself a new database. You'll need to create a user and add that user to the database to grant access, used within the PHP scripts.

At this stage, your new MySQL database is set. However, you don't need to create a new MySQL database if you already have one, you can simply jump to the create table part.

Create Table

You need to define a table for your database. I use phpMyAdmin to do this. This is my table structure:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	PID	int(11)			No	None		AUTO_INCREMENT
2	NAME	varchar(30)	latin1_swedish_ci		No	None		
3	SCORE	int(11)			No	None		
4	MORE1	varchar(30)	latin1_swedish_ci		No	none		
5	MORE2	varchar(30)	latin1_swedish_ci		No	none		
6	MORE3	varchar(30)	latin1_swedish_ci		No	none		
7	MORE4	varchar(30)	latin1_swedish_ci		No	none		
8	MORE5	varchar(30)	latin1_swedish_ci		No	none		
9	GAMEID	varchar(30)	latin1_swedish_ci		No	None		

Let me explain why I chose this structure.

I have the typical NAME and SCORE, so the name of who got the score and the score itself. I also wanted to have some fields to hold additional data, so I have MORE1-5.

I have a GAMEID too. This allows me to store scores for different game projects without having to have a new table for each one.

Finally, I wanted an automatic generated unique number – I called it a PID (Player ID). This is useful for knowing which record to update with a new score etc, although this action I'll not be demonstrating in this document as I don't tend to use it in games\apps, but only for my own high score maintenance app whereby I can manage the table from AGK.

OK, we have a database and a table. We now need to interact with that, using PHP and AGK. At this stage you should know your:

Note – specify everything as created – treat everything as case sensitive.

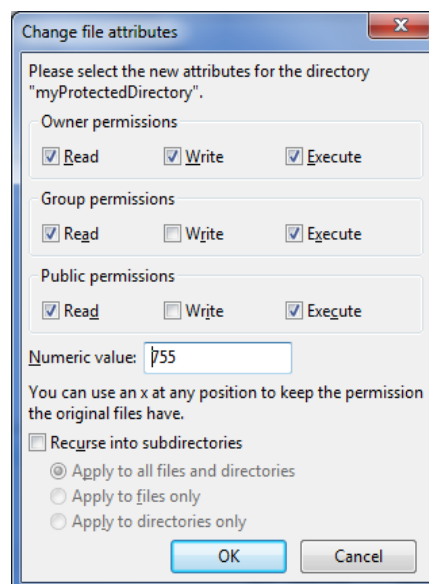
- 1) Database name
- 2) Table name
- 3) User name and password who has access to the database

Let's assume you have a game and the player has just finished and you want their score adding to the table.

Storing your PHP

I access my website folders using FileZilla. You will need to add a folder to hold your PHP scripts. You will need to add some security to the new folder ensuring only access via AGK can be done – as you'll be passing that information in via AGK.

You'll see from the folder permissions that I've used a folder named 'myProtectedDirectory'.



I use File Manager on GoDaddy to create a user and assign that to the folder with the permissions I need. I use FileZilla to assign the folder attributes as shown above.

At this stage you should know:

- 1) The FTP folder name you are holding your PHP files
- 2) Your username and password to this folder

NOTE – you don't have to have a secure folder... my preference. Just note without it, someone can type a URL with all the associated parameters and bypass your AGK coding.... unlikely they'll know the URL syntax, but it's a risk nevertheless.

Posting a score PHP

Create a PHP file like this and save it in the new FTP folder. It's assumed that the script filename is 'scores.php' as this is what is being reference in the AGK code. Case sensitive remember!

<?PHP

```
$score = $_POST['score'];
$name = $_POST['name'];
$more1 = $_POST['more1'];
$more2 = $_POST['more2'];
$more3 = $_POST['more3'];
$more4 = $_POST['more4'];
$more5 = $_POST['more5'];
$gameid = $_POST['gameid'];
```

```
$dbip = 'localhost';
$dbu = 'database username';
$dbp = 'database password';
$dbn = 'database name';
```

```
$con=@mysqli_connect($dbip , $dbu , $dbp , $dbn);
```

```
//The @ symbol can be used in front of MySQL commands so that they don't output warnings
// or other error methods. While testing you'll want to use command without the @ symbol.
```

```
if(@mysqli_connect_errno())
{
    Die('0'); //output 0 or some other code so that your app knows data was NOT sent.
}
else
{
    $newScore_qry = "INSERT INTO tablename
(NAME,SCORE,MORE1,MORE2,MORE3,MORE4,MORE5, GAMEID) VALUES('".$name."',
".$score."', ".$more1."', ".$more2."', ".$more3."', ".$more4."', ".$more5."', ".$gameid."");
    $resultNewScore = @mysqli_query( $con , $newScore_qry);
```

```

}
@mysqli_close($con);
?>

```

The above PHP script takes the parameters passed from the AGK code (next section), connects to your database and inserts that data into the new table you added.

Adding a Score via AGK

To use the PHP you just created, you need coding like this (yellow highlight where the noted information is needed) to call it:

Typical function call:

```
SendScore(10356,"Ian","23","5","","","","CavemanCasinoSlots")
```

The above sends a score to my table with the score, name, level and gems, for my game CavemanCasinoSlots. You can see here how useful having the gameid on the table is and how then the same table can be used for different games.

Then level and gems are held within the more1 and more2 fields.

Function SendScore(iScore , sUserName\$, smore1\$, smore2\$, smore3\$, smore4\$, smore5\$, gameid\$)

```

    If smore1$ = "" Then smore1$ = " "
    If smore2$ = "" Then smore2$ = " "
    If smore3$ = "" Then smore3$ = " "
    If smore4$ = "" Then smore4$ = " "
    If smore5$ = "" Then smore5$ = " "

    con = CreateHTTPConnection()
    httpsOn = 0
    domain$ = "www.yourwebsitename.com"
    directoryUserName$ = "folderusername"
    directoryPassword$ = "folderpassword"
    secureDirectory = 1
    directoryName$ = "myProtectedDirectory/"
    scriptName$ = "scores.php"
    if secureDirectory = 1
        SetHTTPHost(con , domain$ , httpsOn , directoryUserName$ , directoryPassword$)
    else
        SetHTTPHost(con , domain$ , httpsOn)
    endif
    post$ = "name=" + sUserName$ + "&score=" + str(iScore) + "&more1=" + smore1$ +
"&more2=" + smore2$ + "&more3=" + smore3$ + "&more4=" + smore4$ + "&more5=" +
smore5$ + "&gameid=" + gameid$
    SendHTTPRequestASync(con , directoryName$ + scriptName$ , post$)
    time0 = GetMilliseconds()

```

```

repeat
    timeNow = GetMilliseconds()
    timeElapsed = (timeNow - time0)
    ready = GetHTTPResponseReady(con)
    if ready <> 0 or timeElapsed > 60000
        done = 1
    endif
    Sync()
until done = 1
CloseHTTPConnection(con)
DeleteHTTPConnection(con)

```

Endfunction

You'll notice that the above function will either exit once the action is complete or after 60 seconds, whichever comes first.

OK, so we have a MySQL database, FTP Folder, table and PHP file, with all associated permissions and we have AGK code to use the PHP file and post a score.

Now you'd want to read scores back in, so you can present a high score table to your game player.

For this, the returned results will be stored in a series of arrays. The process flow for managing the data in\out of the MySQL database will be:

- 1) Upon first loading of the game, load the high scores into arrays
- 2) Player plays the game and when finished...
- 3) Check the player score and see if they have made the top 10 (or how many high scores you want)

Simply check if their score is higher or equal to the array 10th position. If they have made it into the list, post the score. OK not exactly real-time as another player on another device could have posted a score, so your list is out of date.... but no fear, the player don't know and they'll see themselves going into the top 10, but upon next time they play the game, they've dropped out of it.... beaten by a better player.... it happens... get over it :)

- 4) If applicable, after making the top 10 and before showing the high scores to the player, you can either insert the new score into the correct position in the array.... or you can re-load the scores back into the arrays

Obtaining the scores PHP

Create a PHP file like this and save it in the new FTP folder. It's assumed that the script filename is 'getscores.php' as this is what is being reference in the AGK code. Case sensitive remember!

```
<?PHP
```

```

$gameid = $_POST['gameid'];
$ilimit = $_POST['ilimit'];

```

```

$dbip = 'localhost';
$dbu = 'database username';
$dbp = 'database password';
$dbn = 'database name';
$con=@mysqli_connect($dbip , $dbu , $dbp , $dbn);

if(@mysqli_connect_errno())
{
    Die('0'); //output 0 or some other code so that your app knows data was NOT sent.
}
else
{
    //Now that we're connected to the database we can search for the top nn scores.

    $qry = "SELECT * FROM tablename WHERE gameid = ".$gameid." ORDER BY SCORE DESC
LIMIT ".$limit."";

    $result = @mysqli_query( $con , $qry);
    if (!$result)
    {
        Die('0');
    }

    //We'll store every row in an array element so we can convert it to a string later for
output.

    $out_array = array();
    while($row = @mysqli_fetch_assoc($result))
    {
        //Convert this row's array into a comma separated string
        $thisRow = implode(', ' , $row);

        //insert this string into the final output array
        array_push($out_array , $thisRow);
    }

    //Now we can convert out array into a string where each row is separated by a new line
character Chr(10)
    $out = implode(', ' , $out_array);

    echo $out;
}
@mysqli_close($con);
?>

```

You may notice that for this script we are only passing the gameid and a limit. This is so we can request through the script which set of high scores to return. As you can imagine a high score table could get rather large, so you really only want to return as much as you are intending to show the

player on a high scores screen. This is where the second value, `ilimit` comes in. If for example you only intend to show 10 scores on a screen, you'd pass a value of 10 from AGK. You'll see this in play in the next code example. It also means the arrays can be set to 10 too.

The above script also sorts the results by score, highest first, so the order when returned to AGK is as you need.

Getting scores via AGK

To use the PHP you just created, you need coding like this (yellow highlight where the noted information is needed) to call it:

Typical function call:

```
GetHighScores("CavemanCasinoSlots",10)
```

The above requests 10 top scores to be returned from the high score table for CavemanCasinoSlots.

Function GetHighScores(GameID\$, ilimit)

```
For L = 1 to ilimit
```

```
    Score[L] = 0  
    Name[L] = ""  
    More1[L] = ""  
    More2[L] = ""  
    More3[L] = ""  
    More4[L] = ""  
    More5[L] = ""
```

```
Next L
```

```
con = CreateHTTPConnection()  
httpsOn = 0  
domain$ = "www.yourwebsitename.com"  
directoryUserName$ = "folderusername"  
directoryPassword$ = "folderpassword"  
secureDirectory = 1  
directoryName$ = "myProtectedDirectory/"  
scriptName$ = "getscores.php"  
if secureDirectory = 1  
    SetHTTPHost(con , domain$ , httpsOn , directoryUserName$ , directoryPassword$)  
else  
    SetHTTPHost(con , domain$ , httpsOn)  
endif  
post$ = "gameid=" + GameID$ + "&ilimit=" + str(ilimit)  
SendHTTPRequestASync(con , directoryName$ + scriptName$ , post$)  
time0 = GetMilliseconds()
```

```

repeat
    timeNow = GetMilliseconds()
    timeElapsed = (timeNow - time0)
    ready = GetHTTPResponseReady(con)
    if ready <> 0 or timeElapsed > 60000 //set a timeout limit
        done = 1
    endif
    Sync()
until done = 1

response$ = GetHTTPResponse(con)
Tc = 1
For L = 1 to ilimit
    Name${L} = GetStringToken( Response$, ",", Tc+1 )
    Score[L] = Val(GetStringToken( Response$, ",", Tc+2 ))
    More1${L} = GetStringToken( Response$, ",", Tc+3 )
    More2${L} = GetStringToken( Response$, ",", Tc+4 )
    More3${L} = GetStringToken( Response$, ",", Tc+5 )
    More4${L} = GetStringToken( Response$, ",", Tc+6 )
    More5${L} = GetStringToken( Response$, ",", Tc+7 )
    Tc = Tc + 9

    If More1${L} = " " Then More1${L} = ""
    If More2${L} = " " Then More2${L} = ""
    If More3${L} = " " Then More3${L} = ""
    If More4${L} = " " Then More4${L} = ""
    If More5${L} = " " Then More5${L} = ""

Next L
CloseHTTPConnection(con)
DeleteHTTPConnection(con)

```

Endfunction

You'll notice that the above function will either exit once the action is complete or after 60 seconds, whichever comes first.

That's it. You have everything to go set this up and have a fully hosted high score database\table with PHP scripts.