Multiplayer Assembly Package
Master Server Installation

Thank you for purchasing MAP V.2.0. This piece of documentation will guide you through the setup of the PHP Master Server replacement provided with this version of MAP.

You will need:

- Torque3D MIT V3.5 or above
- A PHP 5.3 with MySQL Enabled Web Server
- Some Basic Coding Knowledge
- Time

I have broken this installation guide into two sections, please direct yourself to the relevant location for your portion.

Section 1 is for people who have already installed MAP and are looking to now install the Master Server.

Section 2 is for people who would like to only install the Master Server replacement without needing to install the rest of the pack.

SECTION 1

INSTALLATION OF MASTER SERVER WITH PRIOR MAP INSTALL

Part 1: PHP Installation / MySQL Installation

The PHP Master server replacement is an addon module (see the MAP Modules Tutorial) that functions in a near similar fashion as the old server query system. However, it replaces the core components of the server heartbeat system with one that PHP and MySQL can recognize to replace the need of a costly full active server in place of using a simple web server with PHP and MySQL.

First thing is first. You need to determine where on your website your master server will be located. You need to recall this location during part 2, so make sure you remember where it is (I'd recommend just using public_html so it's www.website.com/master/, you're free to change this as you wish). Inside the php folder of the new Master Server folder is a "master" folder. Upload this to your website. Inside the master folder is the myGame folder and the install.php file. We'll get to install.php here in a minute.

On your web server, rename the myGame folder to something relevant to your project (you'll need to remember the name of the folder). Since you've already done all of the MySQL work in the past (during the original MAP installation) you have an easy route here. Open a web browser and navigate to the install.php file. Follow the five step process, this will generate the necessary tables and settings files for you. Once this is done, delete the install.php file on your web server (hold on to it locally, you may need it for another project in the future). You're now done with PHP and MySQL, now let's do C++.

Part 2: C++ & Engine Installation

Since you already have MAP installed, the installation is quite simple. Open the new master server folder in the MAP folder and go into the Engine folder. You don't need to worry about re-installing curl, so open the source folder. Copy and replace the two files in the app/net folder to your Torque3D's app/net folder, replacing the two files named serverQuery.h and serverQuery.cpp. The modifications made to this file will allow the old system to be toggled off when the engine detects the new system running.

Now, inside the PGD/Solutions folder, you will see two new files named pgdServerQuery.h and pgdServerQuery.cpp. Copy and paste those files to your Torque3D's PGD/Solutions folder. Lastly we need to add these files to the module system so they are compiled and loaded.

Open PGD/Control/PGDMain.h:

If you're using an older version of MAP, you will need to add a line of code after the project definition settings (`#define _PGD_MYGAME 1`) If should look something like this when done:

```
//Project build settings
#define _PGD_MYGAME 1

#define _PGDSERVERQUERY_ACTIVE 1 //Define this if using the new master server query
system.
```

Next we need to tell your game where your master server is located; you should remember this from part 1. Inside the block related to the files loaded by your project we will add a define for the MasterServer variable, change the string definition below to match the path to index.php of the master server file you installed in part 1:

```
#ifdef _PGD_MYGAME
   //define MAP Module Loads here
   //#undef _PGD_OTHERGAME //example: call #undef on all of ther game defines here.
   //these are the three included "modules" with MAP, I recommend loading all three.
   #define MasterServer "www.website.com/path/to/master/index.php"

   #ifdef _LOADSTORE
      #include "PGD/Solutions/PGDStore.h"
   #endif
   #ifdef _LOADPGDPCON
      #include "PGD/Solutions/portControl.h"
   #endif
   #ifdef _LOADPGDTCPCON
      #include "PGD/Solutions/tcpCurl.h"
   #endif
#endif
```

Lastly, at the very bottom of the file add this block:

```
#ifdef _LOADPGDPMASTER
   #include "PGD/Solutions/pgdServerQuery.h"
#endif
```

We only need to make one more change, and that is to app/mainLoop.cpp to add the new module. Assuming you've used MAP the way I anticipated (:P), you might already know how to do this part, if not, that's all right, follow these instructions:

Open your Torque3D's app/mainLoop.cpp file and scroll down to the definitions you made during the MAP install. They look like this:

```
#define _LOADAUTH 1
#define _LOADSTORE 1
#define _LOAD_MYGAME 1
#define _LOADPGDPCON 1
#define _LOADPGDTCPCON 1
#include "PGD/Control/PGDMain.h"
```

You need to add another one right after _LOADUTH 1. It reads:

```
#define _LOADPGDPMASTER 1
```

Now, scroll down to the area where you used ::create(); on all of the PGD Modules, and add this after the xxz568 load:

```
   #ifdef _LOADPGDPMASTER
         pgdServerQuery::create();
   #endif
```

If you did it correctly, you should have something like:

```
   #ifdef _PGDAUTH
   Con::printf("PGD: xxz568 package activating...");
   xxz568::create();
#endif
#ifdef _LOADPGDPMASTER
      pgdServerQuery::create();
#endif
.
. More stuff down here
.
```

Now, rebuild your engine. If you did everything correctly you should see pgdServerQuery.cpp compile and you should have no errors. You only need to do one last thing and you're done.

Part 3: Torque Changes

You only need to do one thing to your actual game. Inside the TorqueScript folder is a replacement for the joinServerDlg.gui file. Simply replace your game's one with the one in the MAP folder (It's just changing the function calls to use the newly defined ones) and you're all done! Fire up your game and create a dedicated server. Click Query Master and enjoy seeing your server appear with your new PHP/MySQL solution behind it!

SECTION 2

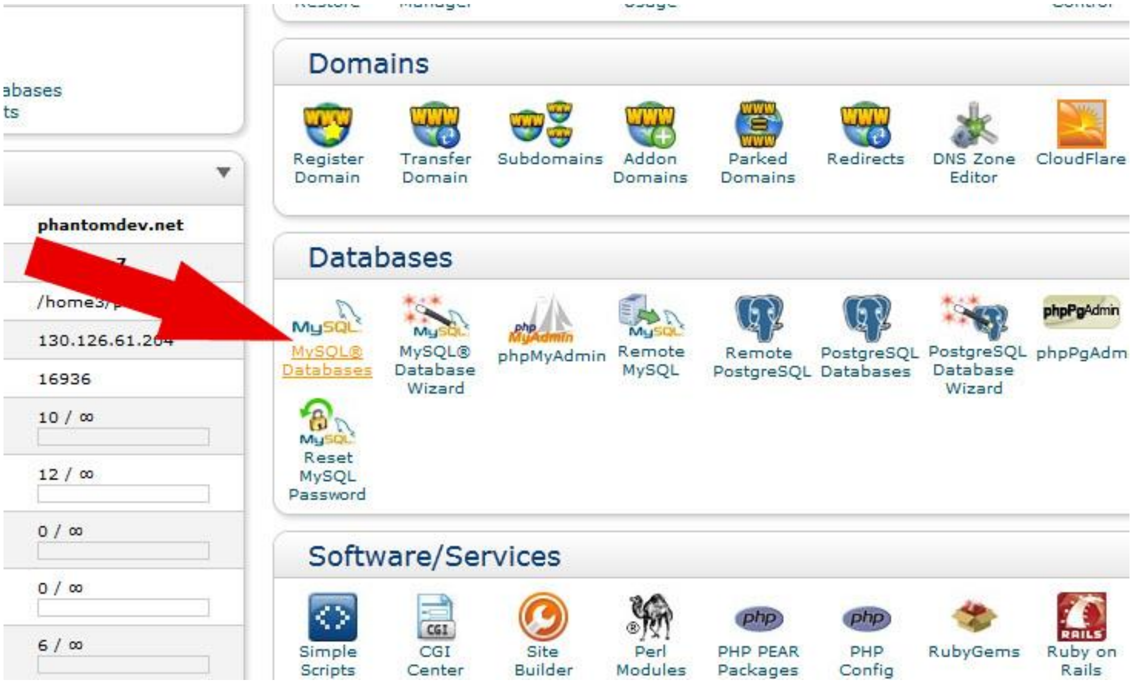
INSTALLATION OF MASTER SERVER WITH NO PRIOR MAP INSTALL

Part 1: PHP Installation / MySQL Installation

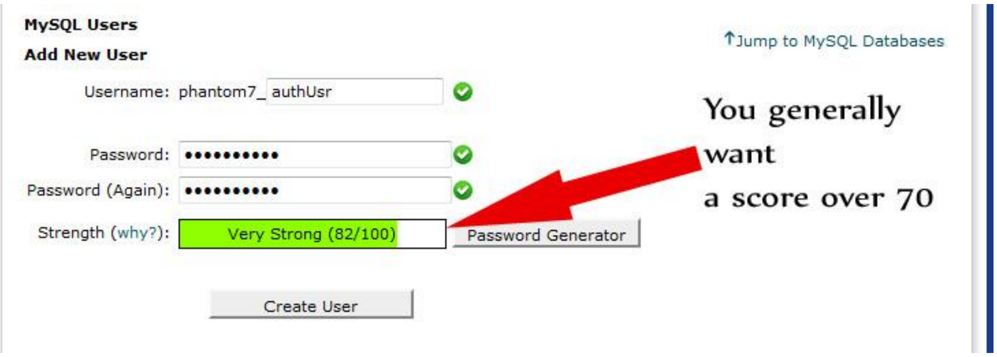
This addon allows you to replace the existing master server listing system with a system controlled by PHP and MySQL. Since you do not have an existing install of MAP you will need to do a little more work to your engine and on your web server.

First thing is first. You need to determine where on your website your master server will be located. You need to recall this location during part 2, so make sure you remember where it is (I'd recommend just using public_html so it's www.website.com/master/, you're free to change this as you wish). Inside the php folder of the new Master Server folder is a "master" folder. Upload this to your website. Inside the master folder is the myGame folder and the install.php file. We'll get to install.php here in a minute.

On your web server, rename the myGame folder to something relevant to your project (you'll need to remember the name of the folder). Now we need to do some MySQL work to create the necessary database. First find the location on your website where you can create MySQL Databases, a CPanel example follows:



Now you need to create a new database. You may name it what you want, but you will need to remember its name for the install.php process. Once you create the new database, you will need to create a user that has access to the database.



MySQL Users

Add New User

Username: phantom7_ authUsr

Password: •••••••••••

Password (Again): •••••••••••

Strength (why?): Very Strong (82/100)

You generally want a score over 70

And then add your new user to your newly created database, giving your new user all permissions. Now with the pre-install work done, we can move on.

Open a web browser and navigate to the install.php file. Follow the five step process; this will generate the necessary tables and settings files for you. Once this is done, delete the install.php file on your web server (hold on to it locally, you may need it for another project in the future). You’re now done with PHP and MySQL, now let’s do C++.

Part 2: Library Installation and Pre-C++ work

This new system uses the cURL library to bypass needing Torque3D’s flawed TCPObject. In order to install the new library however, you need to make some edits to some existing T3D engine files to accomplish this. As of this version of MAP, the replacement files are that of T3D MIT 2.0, but no changes to these files have been made since then. If a case does come up in which these files are modified, a new version of MAP will be released that contains these adjustments.

To install the necessary files, copy and paste everything from the Engine folder to your own, replacing all existing files.

Lastly copy and paste the files from the tools folder to your own to add the module to the engine and re-generate your C++ project to include the new files.

Part 3: C++ Code

First we need to tell the new system where the Master Server is located. Open PGD/Control/PGDMain.h and look for the line that defines the MasterServer variable. Point it to your website where the index.php file inside the folder you renamed in part 1 is.

Now we need to make a little adjustment to app/mainLoop.cpp. Since this is your first time (since you have not installed MAP), I will explain the steps to this in the same way as if you were installing MAP. Start by scrolling down to the end of the #include statements, the last one reads:

```
// For the TickMs define... fix this for T2D...
#include "T3D/gameBase/processList.h"
```

Below that, add this code block:

```
#define _LOADPGDPMASTER 1
#include "PGD/Control/PGDMain.h"
```

Now scroll down to the definition of `void StandardMainLoop::init()` and after the block with Con::init(); add this block of code:

```
   #ifdef _LOADPGDPMASTER
         pgdServerQuery::create();
   #endif
```

It should look something like this when done:

```
   // Set up the resource manager and get some basic file types in it.
   Con::init();
   Platform::initConsole();
   NetStringTable::create();

   #ifdef _LOADPGDPMASTER
         pgdServerQuery::create();
   #endif
```

Once done, compile everything. If you get any linker errors relating to LIBCMT, simply add it to the list of ignored libraries and the engine should compile fine. Once it compiles, you've completed the hard stuff!

Part 4: Last Step, Adding TorqueScript Changes

You only need to do one thing to your actual game. Inside the TorqueScript folder is a replacement for the joinServerDlg.gui file. Simply replace your game's one with the one in the MAP folder (It's just changing the function calls to use the newly defined ones) and you're all done! Fire up your game and create a dedicated server. Click Query Master and enjoy seeing your server appear with your new PHP/MySQL solution behind it!

Final Notes

Thank you for using the Multiplayer Assembly Package, if you have any questions, comments, concerns, or issues building your engine with this pack, please ask on our forums. I’d be glad to help you get it working. Thanks again for supporting PGD and I hope you enjoy our work in the future!

~ Robert Fritzen