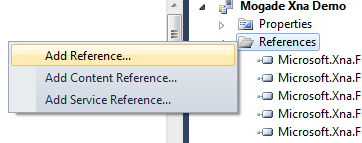
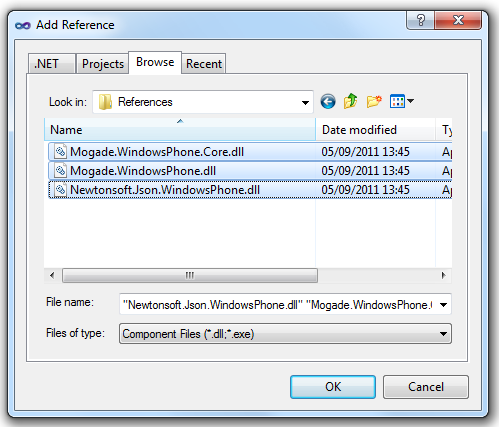
**Mogade Xna Demo**

Thanks for downloading this demo, in it we will go through the steps required to interface mogade into a windows phone 7 xna game. I will point out that there may be better ways to do this but this way worked for me. If you are using Silverlight then you should download the Silverlight demo although much of the stuff in here will still apply.

**Setting up Mogade**

Ok let’s get started, the first thing you need to do is download the windows phone drivers, there are links on mogade.com or they are in the demo project. Next we need to add them to our project so with visual studio open and your project loaded right click on references and click add reference. Next click the browse tab and select the three drivers you downloaded, finally press ok.





Right next we need to add the mogade helper class to our project, this will contain your game key and game secret as well as your leaderboard key, but we can just use the details for the test scoreboard for now. Borrow this class from the example and add it to your project. Remember to change the namespace to that of your own. Finally in this section add the following using statements

* using Mogade.WindowsPhone;
* using Mogade;

add the following global variable which we will use to access the mogade features

* public IMogadeClient Mogade { get; private set; }

and last but not least in your games constructor add the following

* Mogade = MogadeHelper.CreateInstance();
* Mogade.LogApplicationStart();

These will create an instance of mogade and log when your game starts so you can see how many users are loading your game.

**Submitting a Score**

To submit a score first of all we make a variable that contains the users level, score and name, then we call Mogade.SaveScore.



Whether or not the score is submitted successfully the method ScoreResponseHandler is called and allows you to handle any errors or collect any results. It looks a bit like this



This method also allows you to return the users ranks in each of the different leaderboards and you can see how it’s done under else, scoreboardRanks is just a string variable that I made.

**Retrieving LeaderBoards**

So submitting the scores was easy enough don’t you think and I’m sure you’ll be glad to know retrieving the leaderboards isn’t much harder. Firstly we call Mogade.GetLeaderBoard and pass it the scope and page that we want, the scope is the leaderboard that we want and can either be Overall, Weekly or Daily. Page is the page that we want on that leaderboard and starts at one, there will be a maximum of ten scores per page.

This then calls the LeaderBoardReceived response handler method where we can handle any errors and parse the results that we get.



I created a class called ScoreboardEntry which has level, score and name variables and I use it to store the data that is returned which is then added to a list, check the demo project to see how this is done. Note that you don’t have to do it this way.

**Retrieving Rivals**

Rivals are the 3 people that are above you on which ever of the scoreboards you choose, retrieve them by calling Mogade.GetRivals. Pass in a leaderboard scope and the player’s username.



The RivalResponseHandler method is then called which is used to handle any errors and retrieve the results.



I use the same method as when we retrieve the leaderboards to parse the results.

**Getting the Total Number of Scores**

Getting the total number of scores is pretty easy, all you have to do is call Mogade.GetLeaderBoardCount and pass it a leaderboard scope.



The TotalResponseHandler method is then called which allows you to handle any errors and retrieve the results. In this case totalscores is just an integer variable.

**Logging Errors**

Errors can be logged using Mogade.LogError, you simply have to supply a subject and description and these will show up on your mogade control panel.

**Finally**

That should cover most of the mogade features, there’s also an option to add assets on the website that can be then downloaded by the users of your app but I haven’t gotten around to playing with that yet. Also note that all the stuff shown here is demonstrated in the sample and in many cases with more detail so if you don’t understand something here check in there. As a last note remember to put in your own game key, secret and leaderboard key into the mogade helper class when you are done testing. If you need any more help just post on the mogade forums and someone will most likely help you out.

Finally I’d like to thank the mogade team for providing us with a brilliant service at absolutely no cost and with no branding or advertisements. Good job to you all.

I hope you enjoyed reading and that this guide helped you in some way.