**Scores and statistics**

There are three ways to get kill information, and they each have their limitations:

**Using a trigger and a map script**

Within a map script, you can glue a trigger\_multiple that is sensitive to bullets (DAMAGE) to a player, and each time the player dies, you can try and infer from the last bullet that touched the trigger who killed the player; however, this can be inaccurate since two players could shoot at the player at the same time, or one of the player's own bullets that they are firing might leave their gun at the same time as their death - without looking at the source code to the game, it's not possible to determine the exact order in which events are processed within the game so it's hard to tell how accurate this method would actually be. If you wish to use this method, you should test it extensively on a server with many players to ensure that it's accurate and also to make sure it doesn't cause lag due to the triggers glued to all of the players.

The benefit of this method is that the data is available within the map script, if that's useful to you. The drawbacks are the potential for lag and inaccuracy, and the fact that you cannot obtain player names within a map script.

**Using an external tool that polls the server's query port**

This tool would access the game's query port in the same way as [GameSpy?](http://gronnevik.se/rjukan/index.php?n=Main.GameSpy?action=edit). Player scores can be obtained in this way, but not for round-based game types, such as round-based match, objective match, etc., since in those game types the server will return the team's score in place of the player's score, i.e. their frags will actually indicate the team's round wins, and their deaths will always be zero. Note that similar information is available via the rcon 'status' command, with the addition of the player's IP address.

Note that with this method you have to keep checking the scores table frequently enough that you're going to have gotten it at least once between the time when play ends on one map and the server starts loading the next map (how many seconds is that?). You'd of course have to keep the most recent scores table you downloaded in memory plus the one before that, and if the most recent one indicates a different map to the previous one, it means the previous one was the last one for the map and you should record the scores on that to your database. Another issue with this method is that you won't know anything about players who leave during the match, or even too soon after it finishes. To avoid the issue of people leaving too soon after it finishes, perhaps every time you download the scores table you could compare it to the previous one, and if the only difference is that a player left (i.e. no scores increased) you could ignore the new table and keep the old one.

<http://www.kquery.com/> has information about getting information from the query port of MOH servers, and other servers, using various programming languages.

**Reading the server log file**

If you enable logging, you should be able to obtain the messages such as "[UnnamedSoldier1?](http://gronnevik.se/rjukan/index.php?n=Main.UnnamedSoldier1?action=edit) was gunned down by [UnnamedSoldier2?](http://gronnevik.se/rjukan/index.php?n=Main.UnnamedSoldier2?action=edit)" that are normally displayed on all players' screens. You need to set the cvar 'logfile' to enable logging, which goes to main/qconsole.log (or mainta, maintt, etc.). 'logfile 2' causes each line that is sent to the server console to be written to the log file immediately, whereas 'logfile 1' causes them to be buffered a bit before being written. If you don't care about up-to-the-second statistics, you could probably use the buffered option to decrease file/disk access; if you use the value '2', your program could be written to continually read from the log file and read each line as soon as it's written to the file. What actually gets written to the log file depends on what cvars you set. I'm not sure what you have to set to see the "<player> was <weapon>ed by <player>" messages you see as a player, but I'm pretty certain it's possible. Setting developer mode might help. You can probably find information on what settings need to be set on the mohstats site ( <http://stats.clanfx.com/> ).

The benefit I see of reading the server log is that you can tell exactly when a map change occured. Also, if for some reason you weren't polling the [GameSpy?](http://gronnevik.se/rjukan/index.php?n=Main.GameSpy?action=edit) query port for a few moments (network outage, script crashed, etc.), the data is still there in the log file until the server gets shut down and restarted (every time the server starts the qconsole.log file is overwritten). Also you don't need any tricks to compare the scores tables like I suggested above. On the other hand, your code to parse the messages in the log file might get confused if somebody uses a name like "Foo was rifled" or "in the head" so you'd need to pay a lot of attention to that part of your code.

mohstats ( <http://stats.clanfx.com/>, <http://sf.net/projects/fxstats> ) is a tool which parses the log file. It is an open-source tool written in PHP (with a new version being made in Perl) so you can easily modify it to do what you want!

-- [dcoshea](http://gronnevik.se/rjukan/index.php?n=Profiles.Dcoshea)