Overview:

This isn't a proper component but simply a script that utilises 2 other components (**WSH panel mod** & **foo_customdb**) to do the following:

-Import your Last.fm playcount into foobar2000. It stores the data using the **foo_customdb** component which means your files remained untouched but you still have access to the data through normal foobar2000 title formatting (%LASTFM_PLAYCOUNT_DB% and %LASTFM_LOVED_DB% will be set to 1 if a track is loved).

-As well as doing a full library import, it also updates playcount data in the background as you keep on listening.

-It can love/unlove a track on Last.fm via a button.

Update 30/04/13: Script now loves/unloves tracks without the need for foo_softplaylists.

Limitations:

Unfortunately, I have no way to manipulate your foobar2000 library. Instead I have to fetch your whole Last.fm library and import that into **foo_customdb**. So the database may well contain tracks that don't exist in your foobar2000 library. Also, the spelling and punctuation of your artist and title tags have to be an exact match to what is fetched from Last.fm (case isn't important). This may be an issue for some because by default, Last.fm uses spelling correction on their website to "fix" any wrong tags you submit when scrobbling. When this happens you may well have blank entries in your playlists when displaying playcount data.

However, when doing background updates, it is possible fetch data for auto-corrected artist names and tie them to the correct track in your library.

If you don't like Last.fm auto-correcting your submitted data, you can turn this off by visiting this page…

http://www.last.fm/settings/website

If you do this, make sure you turn off the option found on this script's right click menu as well (**Auto-updates>Last.fm auto-correct**).

Setup:

1) For scrobbling, I recommend Florian's **foo_audioscrobbler** component found here:

   http://www.mp3tag.de/en/fb2k.html

   If you prefer the official Last.fm software, you need to access the **Tools>Options>Scrobbling** page and change it to at least 60%. The default of 50% conflicts with the way my script works. It's far easier for people to change this setting than it is for me to implement support for both versions.

2) This requires foobar2000 v1.1 or above and you need the following components:

   WSH panel mod (v1.5.0 or later) - http://code.google.com/p/foo-wsh-panel-mod/downloads/list
   foo_customdb - https://dl.dropboxusercontent.com/u/22801321/wsh/foo_customdb.zip

   A guide for how to install components can be found here.

3) Extract the **marc2003** folder from this zip into your foobar2000 profile directory. On a standard foobar2000 install this directory will be located at

   %appdata%\foobar2000\

   If using portable mode, just dump it inside the main foobar2000 install directory.

4) If you've never used **foo_customdb** before then you should also extract the **configuration** folder to the same folder location as above.

   If you already use **foo_customdb** for other purposes unrelated to any of my scripts, you'll need to update the **foo_customdb** preferences manually with several new entries. Don't overwrite your own **configuration** folder otherwise you'll lose your own settings. See **Appendix A** on the last page of this document.

5) With foobar2000 now running, go to **File>Preferences>Tools>WSH panel mod**. Make sure you uncheck the option labelled **Safe mode**. Restart foobar2000 when prompted.

6) Next you need to add a WSH panel to your layout. Check your UI documentation for how to do this. Now click the panel you've just added to open the editor dialog. Copy/paste the contents of **playcount sync.txt** into the editor. Now close the editor dialog.

7) Now right click the button and use the options to set your Last.fm username and optionally, your password. A password is only required to love/unlove tracks. It's not saved anywhere. It's just required temporarily to authenticate your account. The library import and background updates will work fine with just the username.

8) Lastly, you'll need to configure your playlist (or some other panel) to show the data.

```
//show playcount
[%LASTFM_PLAYCOUNT_DB%]

//show if a track is loved
$ifequal(%LASTFM_LOVED_DB%,1,♥,)
```

Usage:

Before starting, you should check the installation of **foo_customdb** was done correctly. Hold down the **shift** key and right click any entry in your playlist. Open the **Legacy Commands (unsorted)** sub-menu and check these 4 commands exist:

```
Customdb Love 1
Customdb Love 0
Customdb Delete Playcount
Customdb Refresh
```

If you can't see these, please go back to step 4 and try again. Do not proceed until these menu entries are present. Also, make sure foobar2000 is closed while you do this.

Full library import

The library import works in 2 stages. First of all it has to query last.fm for your entire library. It returns 100 tracks per page. So if you have 5000 tracks then that's 50 web requests the script needs to make. It takes roughly 1 second per page but can be faster/slower depending on network conditions and how busy the Last.fm servers are . You trigger this action by right clicking the panel and selecting **Library import>loved tracks and playcount**. You can view current progress in the foobar2000 console window – this will open for you automatically.

When downloading of all the data is complete, a blue command prompt will pop up instructing you to close foobar2000 manually. This is required so that the script has exclusive access to the **foo_customdb** database. It is very important that you do this otherwise it's very likely the update will fail. With foobar2000 now closed, simply press any key on the command prompt window. Even with huge collections, this shouldn't take much more than a few seconds to process. foobar2000 will then start up by itself.

There is another option under **Library import** labelled **from file**. This is a precautionary option that hopefully you won't have to use. It's only there in case the database update fails and it means you don't have to download the data all over again.

Update 29/04/10: The script now imports loved track status at the same time.

Automatic updating

Now that your library should be up to date, the script can fetch updated playback statistics as you listen meaning you shouldn’t have to do a full import ever again. You don’t do anything; it just runs in the background.

Loving/unloving tracks

Just click the heart icon to **love** a track. The local database is only updated on a successfully response from Last.fm. Check the foobar2000 console for confirmation or errors. If a track is already loved, clicking the button will simply **unlove** the track. Tooltips indicate what will happen before you click the button.

Finally....

If you like customising the context menu on your playlist (**File>Preferences>Display>Context Menu**), you can now hide the **Legacy Commands (unsorted)** sub-menu without affecting the functionality of this script.

Thanks

T.P Wang (WSH panel mod developer) for all his help on the hydrogenaudio forums.

APPENDIX A:

This is only for existing foo_customdb users who wish to preserve their existing database setups.

On the **Fields** tab, create the following 2 entries.

Display: LASTFM_PLAYCOUNT_DB
Name: LASTFM_PLAYCOUNT_DB
Key: Custom
$crc32($lower(%artist%%title%))

Display: LASTFM_LOVED_DB
Name: LASTFM_LOVED_DB
Key: Custom
$crc32($lower(%artist%%title%))

Now restart foobar. Next, go back to the preferences and switch to the **Action** tab. Create the following 4 actions.

Display: Customdb Love 1
Field: LASTFM_LOVED_DB
Update: Contextmenu
Set Value: 1

Display: Customdb Love 0
Field: LASTFM_LOVED_DB
Update: Contextmenu -> Erase

Display: Customdb Delete Playcount
Field: LASTFM_PLAYCOUNT_DB
Update: Contextmenu -> Erase

Display: Customdb Refresh
Field: LASTFM_PLAYCOUNT_DB
Update: Contextmenu
Set Value: [%LASTFM_PLAYCOUNT_DB%]

Restart foobar2000, job done.