Overview:

This isn’t a proper component but simply a script that utilises 3 other components (**WSH panel mod**, **foo_softplaylists** & **foo_customdb**) to do the following:

-Import your Last.fm playcount into foobar. 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 foobar title formatting.

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

-It can love/unlove a track on Last.fm via a button. To do this it uses the **foo_softplaylists** component. Although you can use that component on its own without this script, the difference is that this can store the“loved track” status in **foo_customdb** so it’s available through title formatting. It also prevents you from “loving” a track that is already “loved”.

Limitations:

Unfortunately, I have no way to manipulate your foobar 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 foobar 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 context menu as well (**Auto-updates>Use spelling correction**).

Setup:

1) First of all, you’ll need a Last.fm api key. You can get that here. Make a note of this as you’ll need it later.

2) This requires foobar v1 or above. I’ve bundled the required components in to this zip so all you have to do is extract the contents into your main foobar directory – with foobar not running, obviously.

3) If you’ve never used **foo_customdb** before just copy the file **foo_customdb.dll.cfg** into your **configuration** directory.

   On a standard foobar install this directory will be located at

   %appdata%\foobar2000\configuration

   You can copy/paste the above into **Start>Search/Run**). If you’re running portable foobar, this folder will be inside your foobar install directory.)

If you’ve used earlier versions of this script (without import support) then I strongly recommend backing up your old **customdb_sqlite.db** file and then delete it from your profile folder so this script can start with a clean slate. There are significant and enforced changes I’ve had to make. Although it’s technically possible to use the old database, it would be clogged with all your old records and they would be inaccessible. Make sure you overwrite your old **foo_customdb.dll.cfg** with the new file included in this zip also.

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

4) With foobar now running, go to **File>Preferences>Tools**. From here you’ll need to configure **Soft Playlists** with your Last.fm username and password. This component is required to “love” and “unlove” tracks.

5) Also under **Tools**, you’ll find the options for **WSH panel mod**. Make sure you uncheck the option labelled **Safe mode**. Restart foobar when prompted.

6) Next you need to add a WSH panel to your layout. Hopefully that needs no explanation. Now click the panel you’ve just added to open the editor dialog. Now click on **Tools>Import** and browse to **foobar install directory\playcount sync readme\playcount sync.txt**. Close the editor dialog.

7) Now right click the panel and use the 2 options to set your Last.fm username and API KEY. You can also use this context menu to set the background colour for the panel.

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

```
//this is what you use to display the play count.
[%LASTFM_PLAYCOUNT_DB%]

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

Usage:

Note: Do not hide **Last.fm** or **Legacy commands (unsorted)** in your context menu options. The script needs these enabled to work. Before starting you should right click any track in your playlist and check the **Legacy commands (unsorted)** context menu contains these following 4 commands

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

If it doesn’t, this means you haven’t followed step 3 correctly. You must go back and correct this before continuing. Make sure foobar 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 50 tracks per page. So if you have 5000 tracks then that's 100 web requests the script needs to make. I haven't done any proper timing as such but I'm seeing around 50 pages processed a minute. So in this example, it would take roughly 2 minutes. You trigger this action by right clicking the panel and selecting **Library import>Create and import SQL file**. You can check it's progress via the foobar **Console** (found on the **View** menu).

When downloading of all the data is complete, a command prompt will pop up instructing you to close foobar manually. This is required so that the script has exclusive access to the database. This is very important that you do this otherwise it's very likely the update will fail. With foobar 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. foobar will then startup by itself.

There is a 2nd context menu item under **Library import** labelled **Import SQL 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 script should fetch the updated status from Last.fm and store it in the database. If a track is already loved, then of course clicking the button will "Unlove" the track and again the local database will be updated to reflect that after querying Last.fm.

Credits:
T.P Wang: WSH panel mod
JohanDeBock : foo_softplaylists
http://blog.99ravens.net/index.php?e=283 foo_customdb

Original button/tooltip code: T.P Wang & Tedgo

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 foobar, job done.