



A Beginner's Guide to Configuring FooBar2000

**by
Howard J. Rogers**

Contents

<i>Preface</i>	3
<i>1.0 Obtaining the Software</i>	4
<i>2.0 Enabling the Columns UI</i>	6
<i>3.0 Changing the Play Control buttons</i>	7
<i>4.0 Basic Playlist Columns Configuration</i>	9
<i>5.0 Ripping Music</i>	13
<i>6.0 The Album Database</i>	26
<i>7.0 Backing it up!</i>	31
<i>8.0 More Advanced Configuration Options</i>	32
<i>9.0 Restructuring the layout</i>	39
<i>10.0 Playlist Formatting</i>	44
<i>11.0 Final Refinements</i>	50
<i>12.0 Now go back and do it all over again!</i>	54
<i>13.0 Conclusion</i>	54
<i>14.0 Legal Matters</i>	55

Preface

Foobar2000 is the best media player available today, bar none. I say that as someone who has no intellectual investment in it whatsoever. I didn't write it, develop anything for it, or sell a product which interfaces with it. I just use it. Every day, I listen to my reasonably large collection of music files with it, and it is the only media player I have ever used where I've felt in control and comfortable with the functionality it offers. That's because Foobar2000's basic functionality (which is already more than adequate, incidentally) is hugely extensible and configurable once some of its many possible additional *components* are deployed.

There are many sources of information about which components are available, and how they can be configured to transform Foobar2000's capabilities, but it is scattered all over the place. In the official Foobar2000 forum, for example, there is a thread on how to set up the *Columns_UI* component¹, but it is (at the time of writing) 92 pages long, and contains 2,275 different replies. Not the sort of thing you want to go wading through when you're trying to find out simply how to set a font or a background colour on some part of the Foobar2000 interface!

Many people are probably going to give up at that point: they want to play music, not spend an age trying to find out how to tweak the simplest bit of the program interface. But that would mean missing out on the pleasure of using a media player that is there to do your bidding (rather than the other way around), and would be a great shame.

So, I wrote this article. It's for anyone who wants to know how to take an out-of-the-box Foobar2000 installation and turn it into the sort of graphical masterpiece the experts keep posting about in forum threads such as the Foobar Gallery one.² It seeks to consolidate all the configuration information that is currently scattered around the Internet or lost inside massive threads into one, compact and easy-to-read document. It then takes you gently through the easy early stages of personalisation, and gradually introduces you to some of the harder stuff.

By my reckoning, if you plod through this article page by page, doing what it tells you to do as it tells you to do it, you'll have a fully personalised Foobar2000 configuration in about 3 hours. *Mastering* the details might take a bit longer than that, but not much, I think.

I should stress that I am no artistic genius, and this is very much an *introduction* to the art of Foobar2000 configuration, not its final apotheosis. I claim no especial expertise, other than that I've tweaked my own setup often enough over the months. There are probably more efficient ways of doing some things than I describe, and I won't even begin to debate whether there are more attractive configurations than the one I'm using! But the steps I take you through here should nevertheless get you, at least, started in the business of making Foobar2000 your own. I hope you find it of use, anyway.

Regards

A handwritten signature in blue ink that reads "Howard Rogers". The signature is stylized with a long horizontal line underneath the name.

¹ <http://www.hydrogenaudio.org/forums/index.php?showtopic=28647>

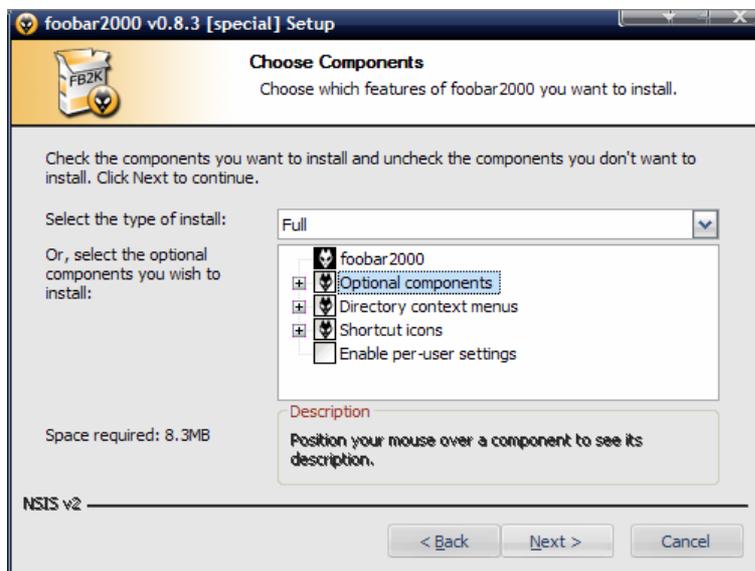
² <http://www.hydrogenaudio.org/forums/index.php?showtopic=30988>

1.0 Obtaining the Software

Foobar2000 can be downloaded from <http://www.foobar2000.org/>. You'll notice from that page that the current version is 0.8.3, and that it is downloadable in one of four different configurations ('Normal', 'Lite', 'SDK' and 'Special'). The basic difference between these versions is that the Lite version includes very few *Foobar2000 Components*, the Special version includes quite a few, and the Normal version is somewhere in between. Components are small DLL files which extend the functionality and usefulness of Foobar2000, and the more you have of them, pretty much the more you can do with the program.

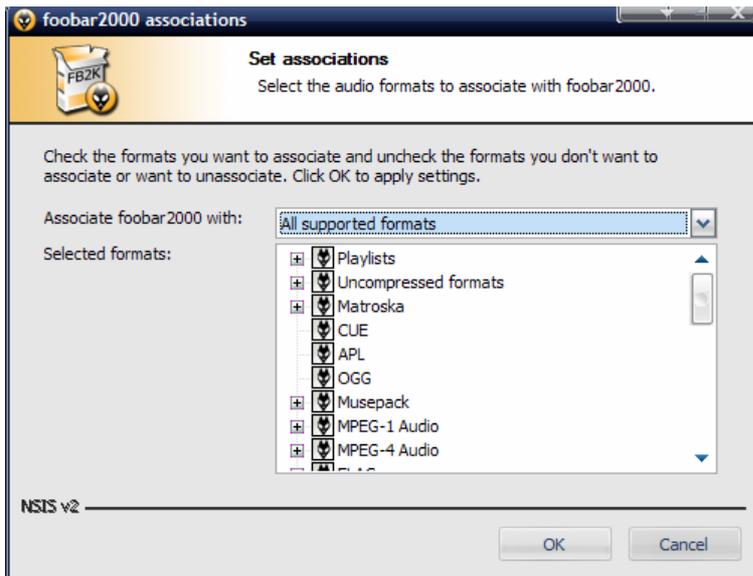
My advice, therefore, is to download the Special edition of the software.

Once downloaded, simply run the executable. Agree to the licence, and then make sure you select to install all the optional components:



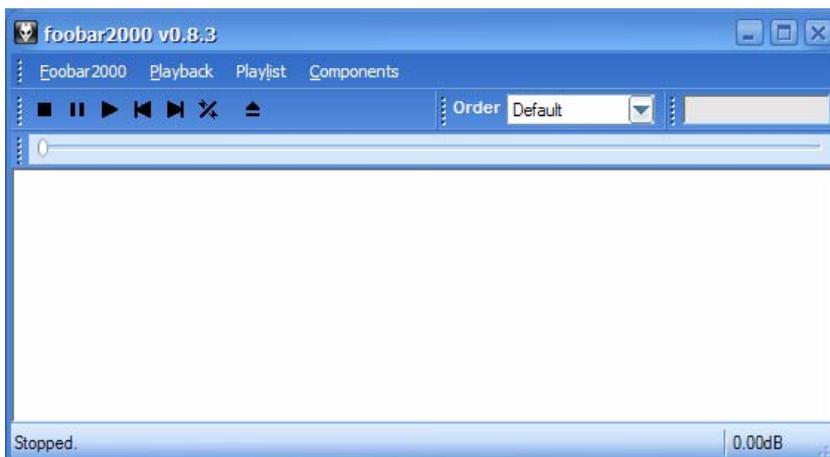
This will mean that your new Foobar2000 installation will start with as many extra bits of potential functionality as possible. After that, just click through the rest of the installation wizard, and either accept whatever it suggests, or supply alternatives that suit your environment (such as a place to house the installation).

At the end of the main installation process, you'll be asked which media formats should be opened in Foobar2000 by default:



As you can see, I usually just alter the combo-box so that it reads, 'All supported formats' and have done with it. But if you have other media programs associated with specific audio formats, feel free to exempt them from this 'hijack' by Foobar2000.

The last thing the installation wizard offers to do is to run Foobar2000 for you. If you accept that suggestion, you should see something like this appear:



This is Foobar2000, basic, plain and very much default. It is, however, perfectly functional: try clicking *Playlist -> Add directory...* and then navigating to one of the hard disk directories where you have stored music files (assuming you have some, of course. I'll discuss how to create some if not in just a moment). That will add a list of tracks to the main area of the program display, and you can simply double-click any one of them to begin playback (or, of course, highlight one of them and then use the button controls towards the top of the screen to start, stop, fast forward or fast reverse playback at will).

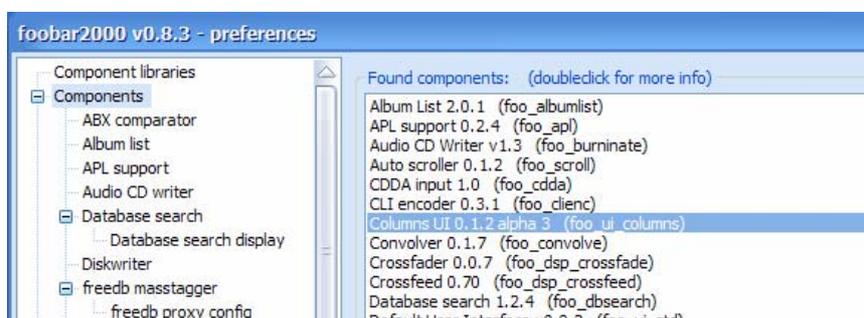
In this configuration, Foobar will perform perfectly adequate media playback functions, and will do so with less distracting 'frills' -and hence a rather smaller memory footprint than most other media players available today. You could leave it there, and go no further in this article, and you would still have yourself a fine media player. But if you'd like to make Foobar2000 look a bit sexier -and a bit more functional, too- then read on.

2.0 Enabling the Columns UI

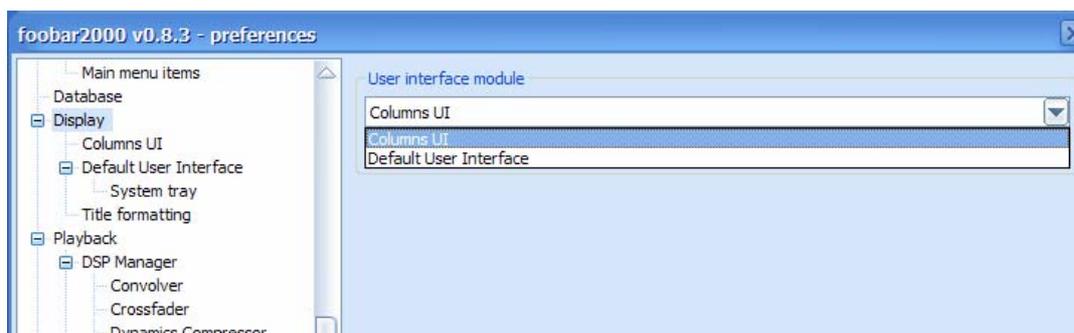
The default look and feel for Foobar2000 is, as you've already seen, a bit Spartan. If you want to jazz it up a little, the first thing you'll probably want to do is to enable the **Columns User Interface**. That ships with the Special download version of Foobar2000, so if you followed the installation instructions above, you'll already have the necessary software.

If you didn't, then you'll want to download the relevant software.³ At the time of writing, the latest version for this plug-in was 0.1.2, and it was an 86KB download. Once downloaded, extract the **foo_ui_columns.dll** file it contains into the main Foobar2000 Components directory (for me, that's `C:\Program Files\foobar2000\components`). If Foobar2000 is running when you do this, you'll have to shut it down and re-start it before it can make use of the new DLL.

However you obtained the DLL, the first thing you should do is to check that Foobar2000 has actually loaded it and can use it. Click *Foobar2000* -> *Preferences*, and search for the **Components** branch:

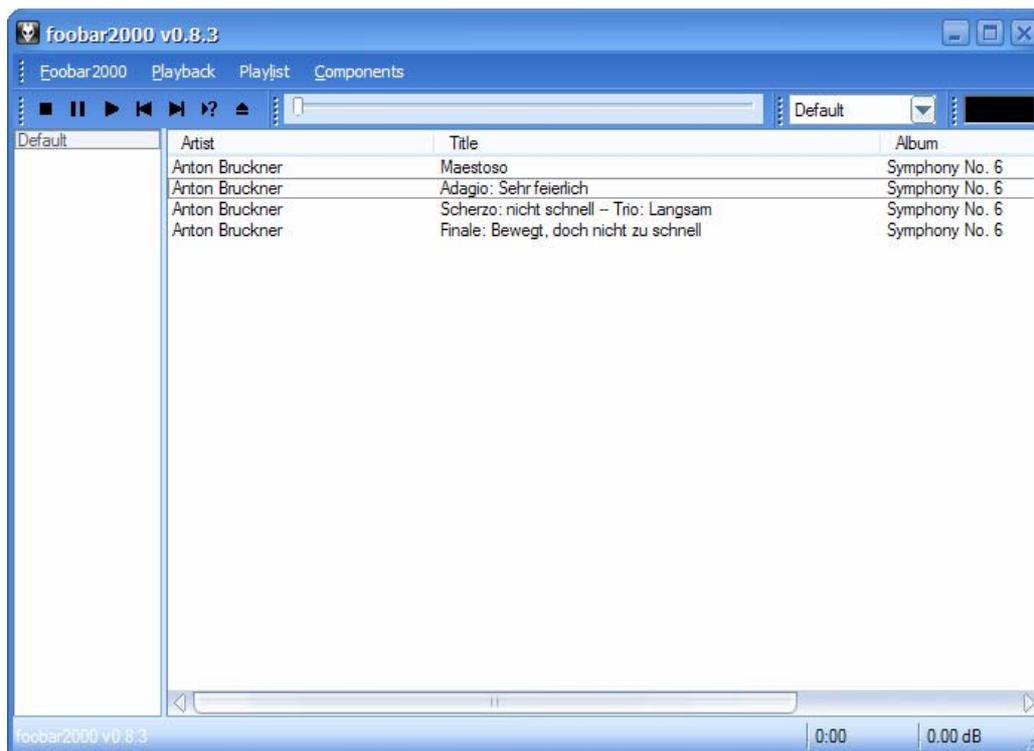


The panel on the right shows *all* the various 'extras' that Foobar2000 can make use of, and you should be able to see Columns UI in the list, just as I have highlighted it here. If you can see that, then scroll down the preferences panel on the left, and click on the **Display** branch:



As you can see, if you expand the combo box here, you'll see that you can choose between the 'Default User Interface' (the one you started with) and the Columns User Interface. Select *Columns UI*, therefore, and then click [Close]. You'll have to re-start Foobar2000 before any change takes effect, but when you do, Foobar2000 should come back looking like this:

³ From <http://music.morbo.org/columns.php>



Now this might not look very different from the first screenshot of Foobar2000 you saw earlier, but you will notice the new panel taking up the entire left-hand edge of the screen, whose purpose we'll investigate later. You'll also notice that there is now a "slide-bar" along the top of the display, next to the play control buttons, which indicates how much of a track has been played. And, finally, there's a play order combo box: currently it displays 'Default', but it can also be used to switch into random and repeat play modes.

So, even without dramatically changing the player's look and feel, switching on Columns UI has given us some new functionality. But we can go further than this. Much further!

3.0 Changing the Play Control buttons

Here's something fairly simple to alter: at the moment, you're using play control buttons which look like this:



...which is pretty basic, and not very inspiring. If you are any good with painting packages, create a set of buttons of your own (make each button 23x23 pixels, and then merge them together into a single image about 175 pixels long (since $7 \times 23 = 161$, that gives you 14 pixels with which to space apart the buttons). If your painting package is up to it, make the background transparent and save the image in PNG format. Otherwise, save it as a plain old BMP.

Here's my first go, in which I drew and spaced the buttons more or less freehand:



When I switched these on in Foobar2000, here's what I ended up with:



...which tells you that it certainly works, but you need to be precise on your button widths and spacing, otherwise parts of one button 'squeeze' into the space intended for the next. You'll probably end up having to make several attempts before you get it just perfect, but if you're artistically-inclined, the effort may be worth it!

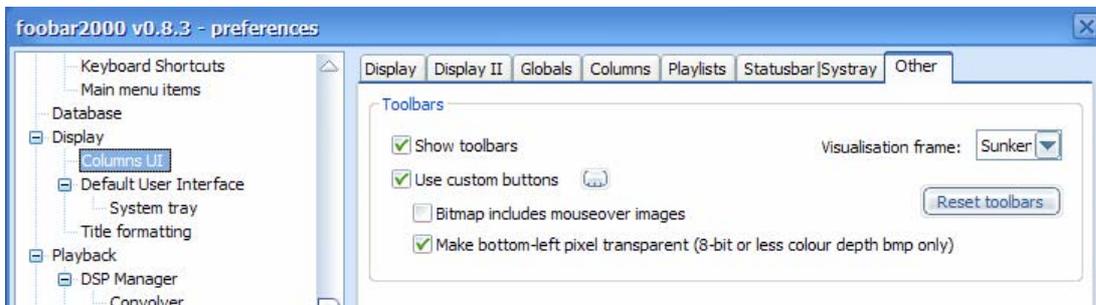
Now, as you can tell, I am absolutely hopeless at art, and besides I am not patient enough to keep refining my work until it is perfect. So I did what I always do in such desperate circumstances (and what I suggest most of my readers will end up doing!): steal someone else's buttons! If you go to the Hydrogen Audio forums⁴ () and search for (say) *custom buttons*, you'll find dozens of example buttons that people have already perfected and offered to the community at large.

For example, I easily found these⁵:



These buttons are twice as long as the original ones, because they incorporate a rollover effect. That is, there is a dull-blue stop button at the far left of the image, and a bright-blue stop button about half-way through: once Foobar2000 has been persuaded to use these buttons, the stop button will appear to glow if you hover your mouse over it -and obviously there are equivalent rollovers for each of the other buttons, too.

Save your file in the Foobar2000 icons directory (for me, that's C:\Program Files\foobar2000\icons). Then, click *Foobar2000* -> *Preferences*, and select **Display** -> **Columns UI** in the left-panel's tree of options:

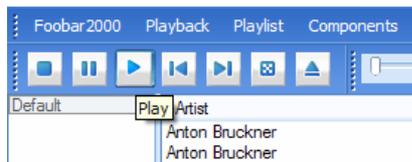


Select the **Other** tab in the right-hand panel, and switch on the *Use custom buttons* option. Use the (...) button to browse to where you saved your button bitmap, and highlight the relevant file, clicking OK to confirm your choice. If your button picture is of the twice-as-long variety, because it incorporates rollover effects, then select the *Bitmap includes mouseover images* option, too. If it doesn't, then do as I have here and leave that option un-selected.

Finally, click [Close], and shutdown and re-start Foobar2000. You should see something like this:

⁴ At <http://www.hydrogenaudio.org/forums/index.php?act=SF&s=&f=28>

⁵ At <http://www.hydrogenaudio.org/forums/index.php?act=Attach&type=post&id=1250>



...and, as you can see, the new buttons are now in place and fully functional. You can play around with these options as much as necessary until you get something you're happy with.

One issue you might encounter, especially if you borrow other people's buttons from the forums, is that the use of transparent PNG files might cause Foobar2000 to display this error when you try to use them:

```
ERROR (foo_ui_columns) : Failed to load libpng. Ensure you have
libpng12.dll and zlib1.dll present in an appropriate location.
INFO (CORE) : startup time: 281 ms
```

...at which point, the standard buttons we started with will be loaded instead of the ones you actually wanted. The problem here is that a couple of DLLs are required to interpret PNG files, and Foobar2000 doesn't ship with these by default. However, you can obtain them by visiting <http://tinyurl.com/84x64>. Using your browser to get to that URL should prompt you to save a single RAR file which contains the two necessary DLLs (it's a mere 113KB in size). Un-zip the downloaded file, and transfer the two DLLs it contains to the Foobar2000 main directory (on my machine, that's in C:\Program Files\foobar2000). You should also manually rename the file **libpng13.dll** to be **libpng12.dll**, otherwise the program won't realise it's there.

With those two files in place, you should be able to close Foobar2000, re-start it, and have no further problems with PNG button files.

4.0 Basic Playlist Columns Configuration

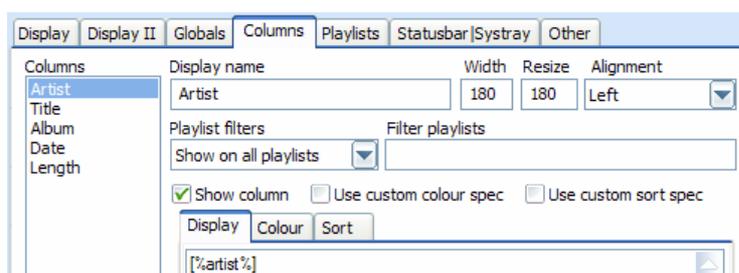
At the moment, your playlist columns probably look something like this:

Artist	Title	Album
Anton Bruckner	Maestoso	Symphony No
Anton Bruckner	Adagio: Sehr feierlich	Symphony No

The column positions, widths and titles might be acceptable for your purposes... but then again, they might not -in which case, of course, you'll want to alter them to suit you.

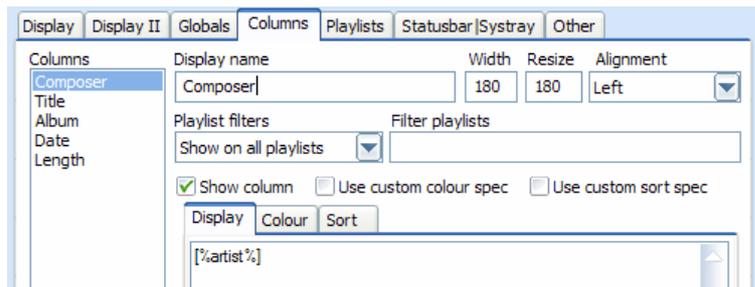
In my case, for example, I dislike composers of works being called 'Artists', and I would like to see who the real 'artists' actually performing the work are. So what I want is a column headed (say) COMPOSER, which displays 'Anton Bruckner' and a column headed 'Artists' which displays something like 'Georg Solti, Chicago Symphony Orchestra'.

Well, changes like these are made by clicking *Foobar2000* -> *Preferences* and selecting the *Display* -> *Columns UI* item in the left-hand tree. Click on the **Columns** tab, and you'll see something like this:



To the left of this panel, you'll see a list of the columns which the playlist window displays. These can be re-ordered by clicking the [Up] and [Down] buttons underneath the list (which I haven't shown in this screenshot).

Each column is made to display a particular piece of data extracted from the music file's ID3 tags by specifying the tag name as a variable in the Display section in the middle of the screen. In my screenshot, for example, the Artist column will display the contents of the %artist% tag. Now, it happens that I have tagged all my music so that the %artist% tag always contains the composer's name... so it might be more sensible to change the *name* of the column, without altering where it will get its information from, like so:

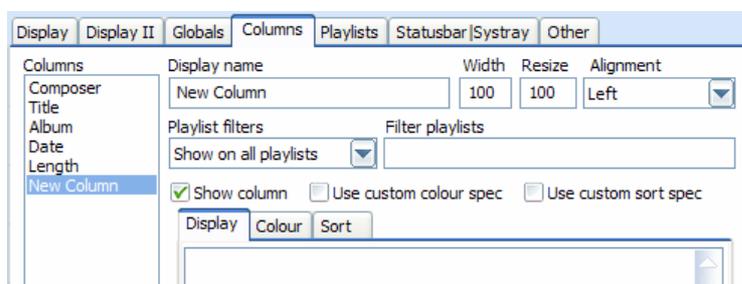


All I've done here is to type in a new 'Display name', but keeping everything else the same, including the contents of the 'Display' tab in the middle of the screen. When I click [Apply] and then [Close], this is the result:

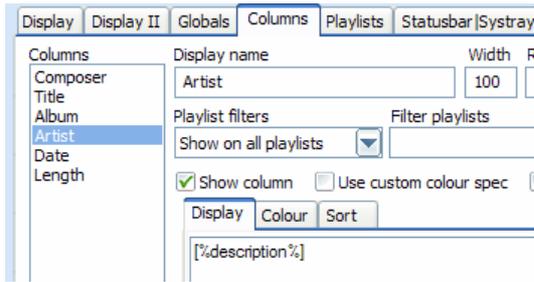
Composer	Title	Album
Anton Bruckner	Maestoso	Symphony N
Anton Bruckner	Adagio: Sehr feierlich	Symphony N
Anton Bruckner	Scherzo: nicht schnell – Trio: Langsam	Symphony N
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell	Symphony N

Note how my first column no longer reads 'Artist', which is exactly what I originally wanted. It's not a very profound change that I've made, but it makes the point that you decide what columns you want to appear in your playlists; and you decide what they'll be called, in what order they'll appear, and what information they'll each display. The entire thing, unlike any other media player you'll ever have used, is 100% user-configurable!

The other change I wanted was to display the performers of each track. So I return to the previous configuration screen, and this time click the [New] button. This adds a new column to the proposed display, like so:



Now I need to specify at least two pieces of information: what is the column to be displayed as, and what ID3 tag will it get its contents from? The first piece of information is purely cosmetic, and goes into the **Display name** field: I'll call mine 'Artist'. The second piece of information depends on how you tagged your music files when you ripped them, and I happen to know that all my files have their performers stored in a tag called DESCRIPTION. I can then use my [Up] button to move the new column higher in the list of all possible columns, so that it appears before the Date or Length ones. Here's how I would complete the screen to achieve all of that, therefore:

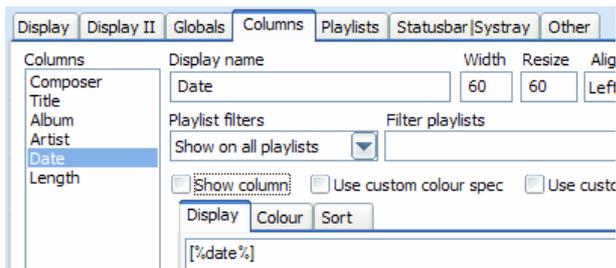


Once again, if I click the [Apply] and [Close] buttons, I see the results of my tweaking:

Composer	Title	Album	Artist	Date	Length
Anton Bruckner	Maestoso	Symphony No. 6	Georg Solti, Chicago Symphon...		17:42
Anton Bruckner	Adagio: Sehr feierlich	Symphony No. 6	Georg Solti, Chicago Symphon...		19:24
Anton Bruckner	Scherzo: nicht schnell -- Trio: Langsam	Symphony No. 6	Georg Solti, Chicago Symphon...		8:53
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell	Symphony No. 6	Georg Solti, Chicago Symphon...		15:13

I've had to manually re-size my columns (by hovering over the vertical line that divides one from another and then click-and-dragging left or right) to fit the new information in, but there it is, and I'm happy. Nearly... Because looking at this, I wonder what the purpose of the Date column is, and I decide that I don't really need it at all!

Well, back to the earlier dialog, and this time...



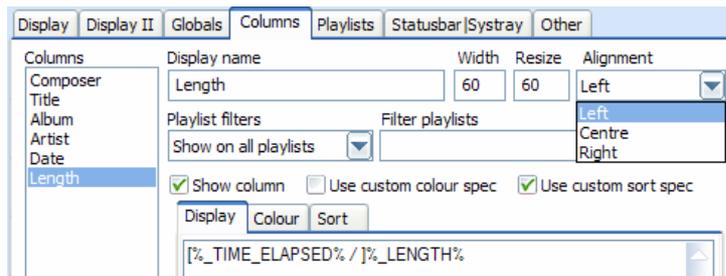
...I select the Date column in the panel on the left, and uncheck the 'Show Column' box in the main part of the display. This would mean that the Date field is still there, under the hood, and is thus capable of being searched for and sorted by... but it simply isn't displayed in the main playlist. Alternatively, I could click the [Delete] button underneath the list of columns, and delete the field entirely. That stops it being displayed, of course, but also means it's not there to be selected, searched or sorted by.

So one more click of [Apply] and [Close], and I now have this sort of result:

Composer	Title	Album	Artist	Length
Anton Bruckner	Maestoso	Symphony No. 6	Georg Solti, Chicago Symphon...	17:42
Anton Bruckner	Adagio: Sehr feierlich	Symphony No. 6	Georg Solti, Chicago Symphon...	19:24
Anton Bruckner	Scherzo: nicht schnell -- Trio: Langsam	Symphony No. 6	Georg Solti, Chicago Symphon...	8:53
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell	Symphony No. 6	Georg Solti, Chicago Symphon...	15:13

...and you'll notice that this time there's no date displayed.

You'll also notice another new thing: the lengths of each track are displayed correctly enough, but they are not displayed very nicely. Numeric quantities should be displayed right-justified, not left-justified as they are here. So, back to the previous dialog once more:

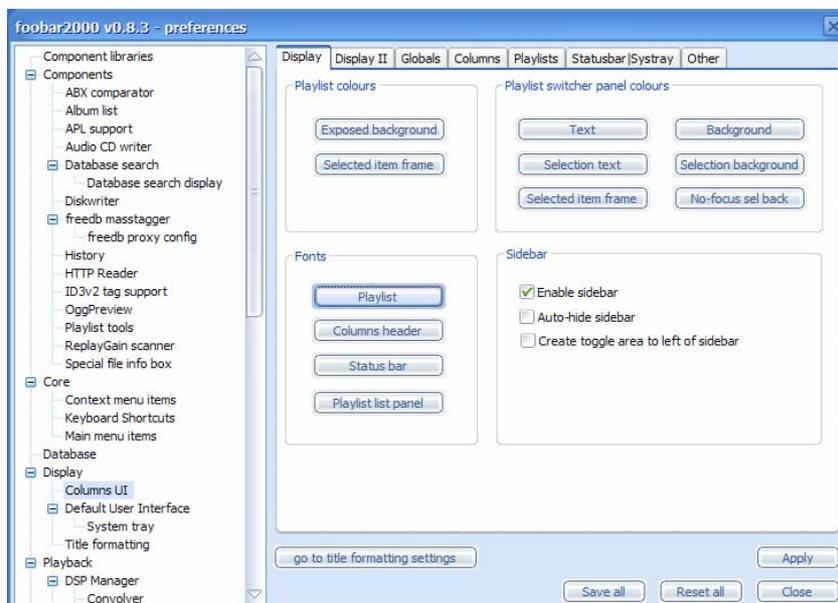


If you select a column in the left-hand panel, you can then select whether it should be left-, centre- or right-justified by selecting from the combo box shown here. I'd obviously want the **Right** option for the Length column. A final [Apply] and [Close], and the results are now like this:

Composer	Title	Album	Artist	Length
Anton Bruckner	Maestoso	Symphony No. 6	Georg Solti, Chicago Symphon...	17:42
Anton Bruckner	Adagio: Sehr feierlich	Symphony No. 6	Georg Solti, Chicago Symphon...	19:24
Anton Bruckner	Scherzo: nicht schnell -- Trio: Langsam	Symphony No. 6	Georg Solti, Chicago Symphon...	8:53
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell	Symphony No. 6	Georg Solti, Chicago Symphon...	15:13

Now, there's a lot more that we might consider doing here, but I hope this gives you a feel for what is possible. I've changed the column names, their ordering, and their justification. I've removed some columns from the display altogether, and added others of my own devising. I've also re-sized columns.

The last thing you might want to consider doing just now is to alter the font the column information is displayed in: right now it's a rather nasty sans serif font, and I have nicer ones available on my PC. This, too, can be configured and to do so, you click *Foobar2000* -> *Preferences* once more, and again go to the *Display* -> *Columns UI* options in the left-hand options tree. This time, however, you want to select the **Display** tab in the main part of the right-hand panel:



Notice that this screen gives you the ability to change the colours and fonts for practically any part of the Foobar2000 interface, and you can certainly spend many hours exploring precisely what effects you can achieve using all its many options. For now, though, all we do is click the [Playlist] button in the *Fonts* part of the screen. A standard dialog pops up which you use to select a better font than the default MS Sans Serif. You can pick any font you like, of course, but

quite a lot of users seem (purely anecdotally) to like Franklin Gothic Medium or Frutiger Linotype at about 10 points in size.

Click [Apply] and [Close] to see the effects of changing the font, and you might see something like this:

Composer	Title
Anton Bruckner	Maestoso
Anton Bruckner	Adagio: Sehr feierlich
Anton Bruckner	Scherzo: nicht schnell – Trio: Langsam
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell

...which is, in my opinion, a bit of an improvement on the original (though you might not be able to tell the difference here! But look at the “g”):

Composer	Title
Anton Bruckner	Maestoso
Anton Bruckner	Adagio: Sehr feierlich
Anton Bruckner	Scherzo: nicht schnell -- Trio: Langsam
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell

The column titles are still the same, of course: the option I've just described only changes the font used for the *body* of the playlist display. If you want to alter the font used in your column headings, that would require going back to the earlier **Display** panel, and clicking the [Columns header] button (just underneath the [Playlist] one you've already used). Again, pick something appropriate (I use MS Trebuchet at 11 points), and you might end up with a finished display like this one:

Composer	Title	Album	Artist	Length
Anton Bruckner	Maestoso	Symphony No. 6	Georg Solti, Chicago Sym...	17:42
Anton Bruckner	Adagio: Sehr feierlich	Symphony No. 6	Georg Solti, Chicago Sym...	19:24
Anton Bruckner	Scherzo: nicht schnell – Trio: Langsam	Symphony No. 6	Georg Solti, Chicago Sym...	8:53
Anton Bruckner	Finale: Bewegt, doch nicht zu schnell	Symphony No. 6	Georg Solti, Chicago Sym...	15:13

Again, you might have greater artistic powers than me, and come up with a better, more pleasing combination of fonts, sizes and so on. But the point is not the specifics of your choices or mine, but knowing (a) that you *have* choices and (b) how to go about making them, whatever they may happen to be. Choice: that's the Foobar2000 paradigm in a nutshell.

5.0 Ripping Music

5.1 Automatic Tagging

So far, I've rather assumed that you already have ripped music tracks to work with, but that might not be true for you –and, in any case, you need to know that Foobar2000 is an excellent music *ripper*, not just a player.

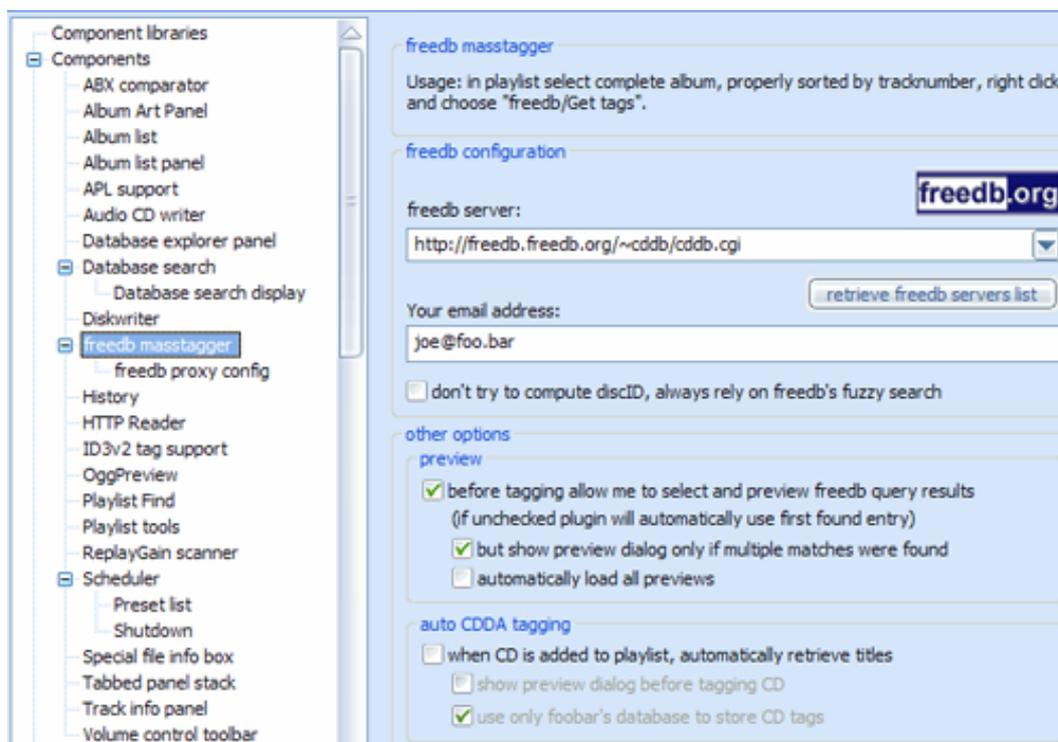
The physical act of ripping music is, of course, a piece of cake: insert CD, click buttons, sit back and wait. But the real *art* of ripping properly lies in tagging your music properly. Decently-tagged music files make for a well-organised and easy-to-use music database (about which I'll be writing more in the next section). Adding your own tags to every file you rip is a pain, but of course there has been a potential fix for this for years: a publicly-accessible database of nearly all CDs released over the years, with all the track tags already in place. All you have to do is get a program to read your CDs, compare the track layout with those already in the database, and if there's a match, you can 'pull' the database's tags down to your own system, and treat them as though you'd typed them in yourself.

For years, the database to use for this purpose was called CDDB ("The CD Database"), but that was purchased by a company called Gracenote, who then decided all the track details (freely

submitted by users over the years as a community gesture) were a commercial asset and required a license to access. At which point, users round the globe said a collective, "Stuff that!", and promptly set up their own, free CD database, called **freedb**. If you like, freedb is an open source equivalent to Gracenote's now-commercial CDDB.

Well, the job of accessing freedb and retrieving track tags is one for a Foobar2000 component called **foo_freedb.dll**. This should already be installed and ready to use if you downloaded the 'Special' version of Foobar2000, but if necessary you can obtain the component separately.⁶ As with all components, the downloaded DLL must be copied to the Foobar2000 components subdirectory, and Foobar2000 itself must then be re-started.

As usual, too, once the new component is installed, you should take a few moment to configure some of its many options. Do that by clicking *Foobar2000 -> Preferences* and then finding the *Components -> freedb masstagger* item in the configuration options tree at the left of the screen:



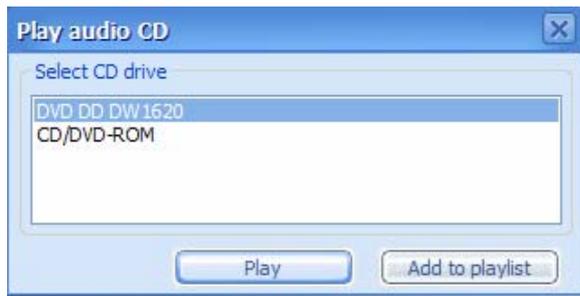
Usually, you won't have too much to alter here. The name of the freedb server is probably correct as it is, for example, but if not you can always click the [Retrieve freedb servers list] button to choose alternatives. You might like to alter the phoney email address here, too (but then again, with spam the way it is, you might not!)

About the only thing here I usually change is to select the *but show preview dialog only if multiple matches were found* item. There's little point in previewing a choice of one, I think!

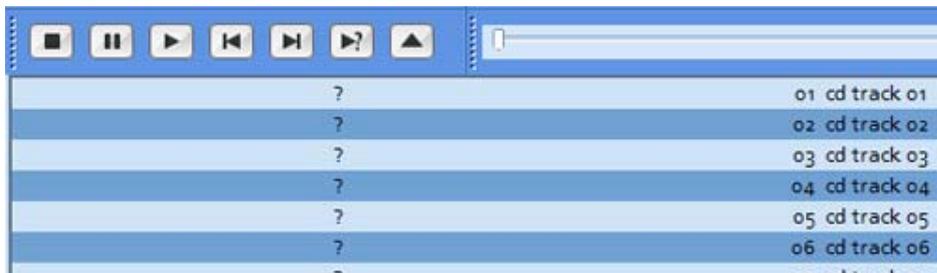
Of course, if you access the Internet via a proxy server, then you should select the *freedb proxy config* item in the left-hand tree structure and supply the details of that server's IP address.

With that done, you're ready to insert a new music CD into your computer's CD-ROM. You might also like to clear any playlist details that are already being displayed by selecting the *Playlist -> Clear* menu options. When you're ready, click the *Components -> Play audio CD* menu options. You'll probably see a dialog like this one appear:

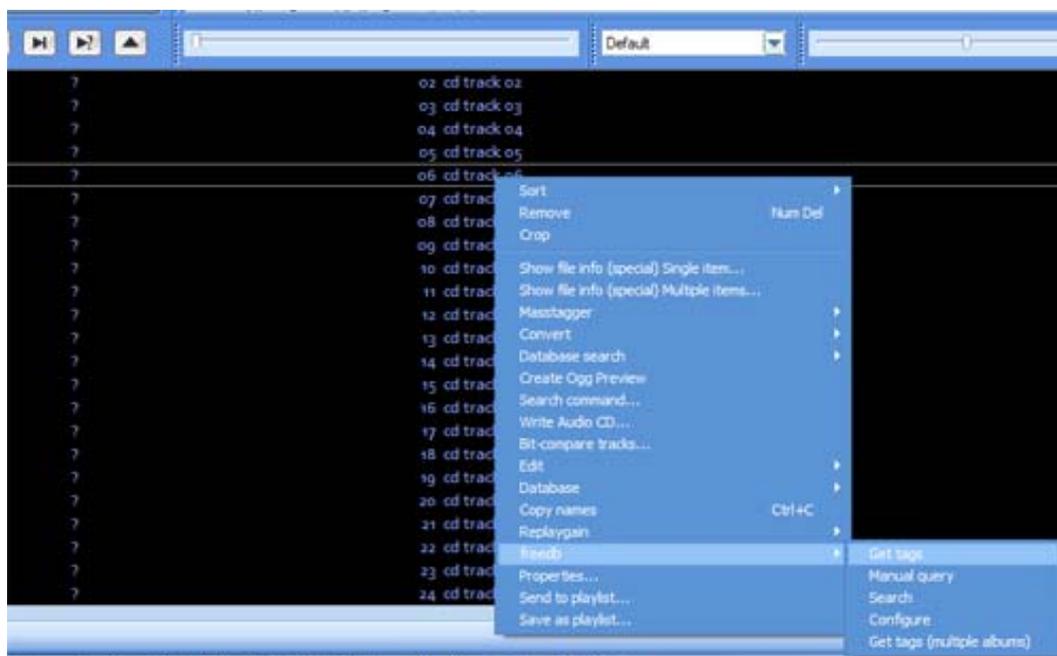
⁶ From <http://pelit.koillismaa.fi/plugins/redir.php?id=443>



If you've a choice of CD drives (or, as in my case, both a CD and a DVD drive), highlight the right one, and then click the [Add to playlist] button. At this point, your main playlist area will fill with track details, although none of the tracks will have much information to display, so mostly you'll see question marks and generic track names:

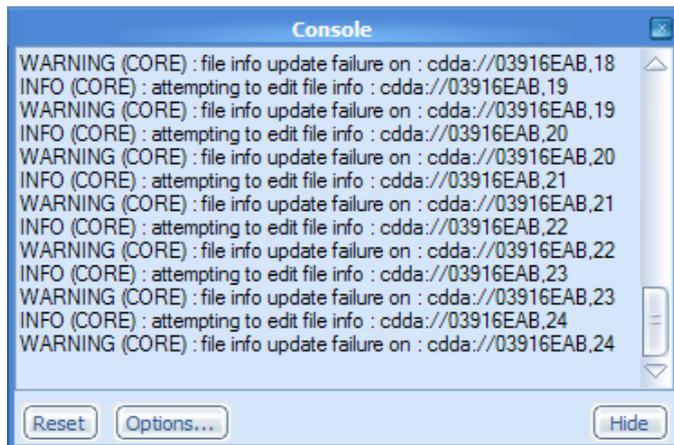


Once these sorts of details appear, you can close the earlier pop-up dialog. Now you can turn to the matter of retrieving track details from freedb. Select all the tracks in the main playlist area. Make sure you don't miss any of them out, because it's the pattern of tracks and their lengths on a CD that allows a unique match to items in the freedb database. The easiest way to select all tracks that I know of is to click track number 1, hold down the shift key, and then click the last track. That should cause all the tracks in between to be highlighted:



You now right-click the highlighted tracks to call up the large pop-up menu you see in this screenshot, and select the *freedb* -> *get tags* menu options. If you're lucky, and your CD matches ones already in the freedb database, you will either get a chance to preview and accept the tags found or, if there's exactly one matching entry in the freedb database and you selected the option in the *foo_freedb.dll*'s configuration options to *show preview dialog only if multiple*

matches were found that I mentioned earlier, the tag names will be automatically filled in for you without further prompting. You'll also get a pop-up warning box like this one:



This is the Foobar2000 console, and it's where all of Foobar2000's warnings and error messages are displayed. In this case, you can ignore the errors: they are telling you that Foobar2000 was unable to write the retrieved tags –which, given they are sitting on a CD-ROM, is not really very surprising! Nevertheless, the tracks will be displaying the retrieved values, and that's all we need at this stage.

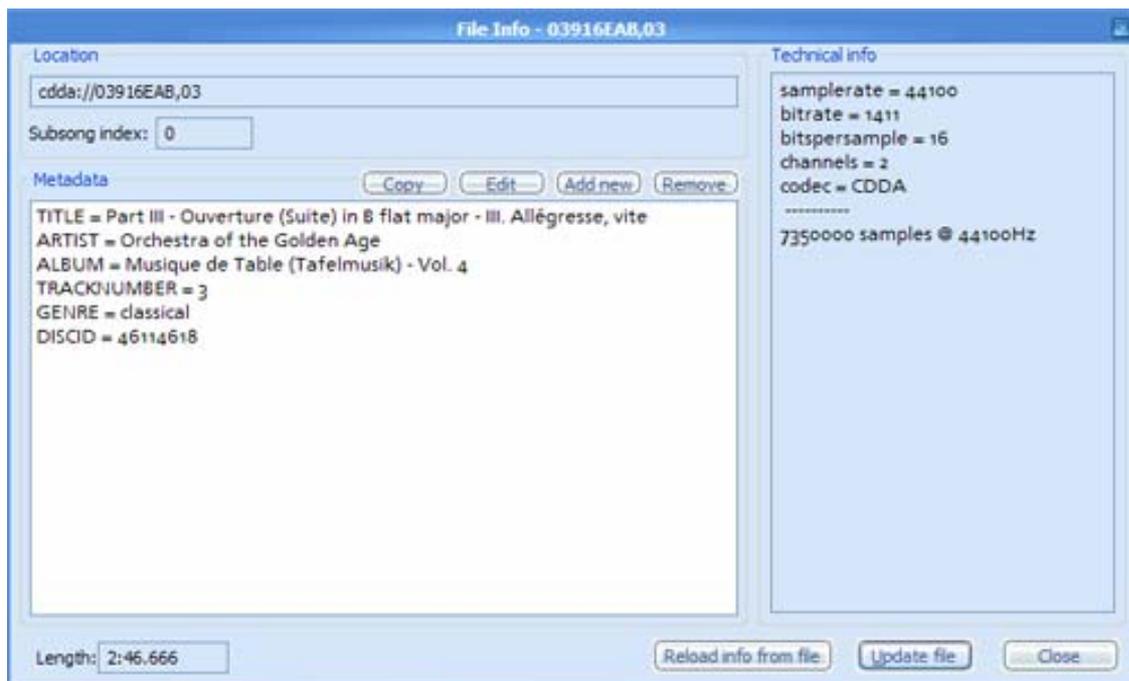
5.2 Manual Tagging

You should always check the results of the automatic retrieval of tags from freedb, because what's been retrieved is someone else's idea of how to tag music files –and those might not be *your* idea of good tagging! For example, I am forever having to deal with people who think that "Bach" or "Bach, J.S." was a composer, when in fact his name was 'Johann Sebastian Bach'. And freedb always seems to think 'Georg Solti, Vienna Phil' counts as an 'Artist', where I very much prefer to see that sort of information going to the DESCRIPTION tag.

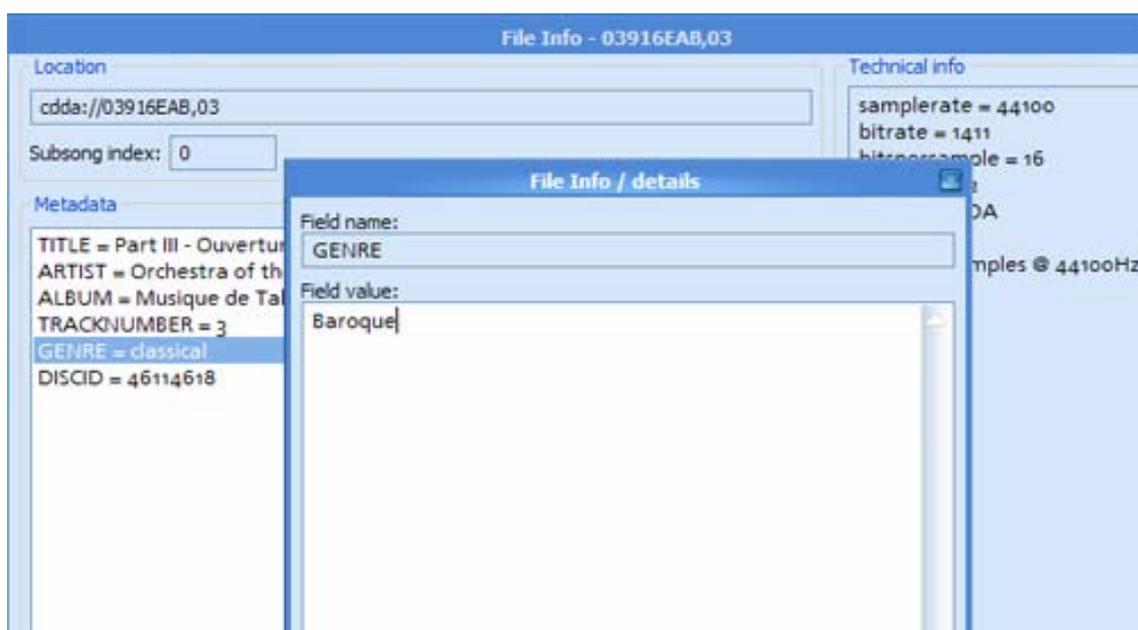
Additionally, if you are cursed with an esoteric taste in music such that you're about the only person on the planet to own a particular CD, you will likely draw a blank when trying to find a match in the freedb database, and will therefore have a bunch of anonymous tracks which need tagging by hand in their entirety.

So there are two good reasons why you will need to get the art of manual tagging under your belt!

The basic principle behind manual tagging is: highlight a track, right-click and select *Properties* from the menu that then appears. You'll see something that looks like this:

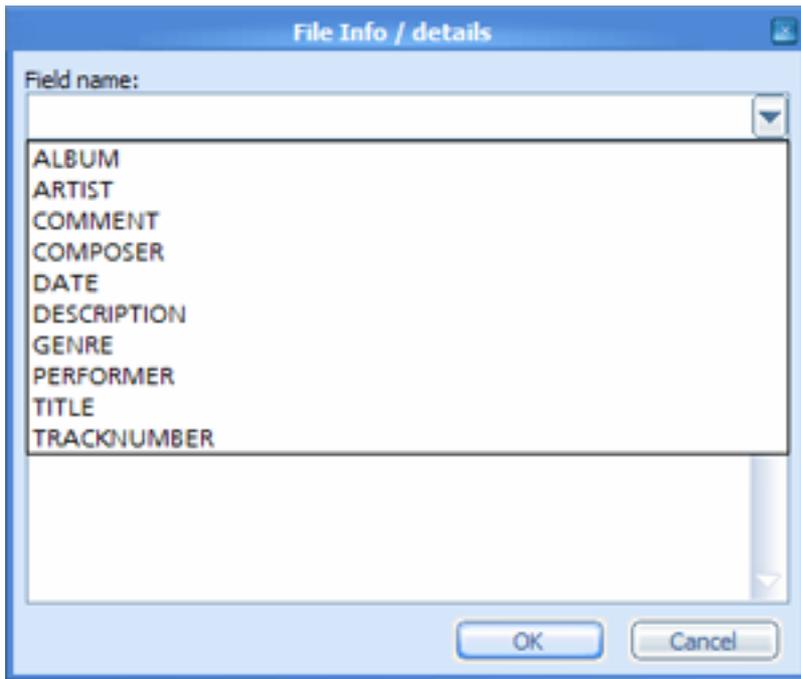


These happen to be tags retrieved from freedb. If I want to change any of them, I can just double-click one of them, and type in a new value:



In this example, you can see the original value for the GENRE tag in the background, and I'm replacing it with something more appropriate in the foreground. If there are tags completely missing, then you can use the [Add new] button on the main screen to create them. I, for example, always like to add a DISCNUMBER tag to my tracks, so that I can cope with multi-CD boxed sets.

When you click [Add new] you'll see a completely empty version of the pop-up box shown in the previous screenshot. You'll first have to specify which tag it is that you're creating. There's a combo box for that, and will display all the 'standard' choices:



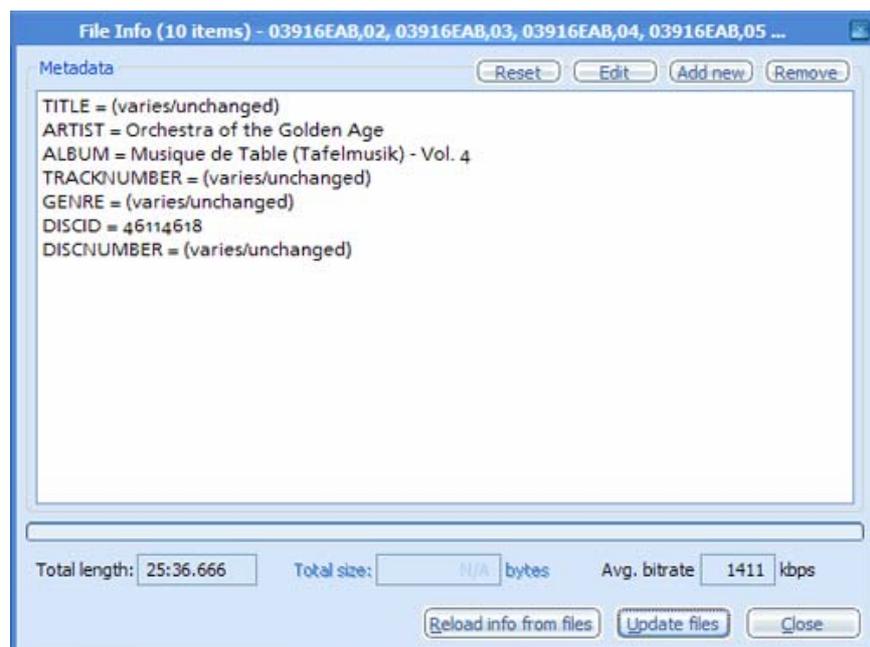
If none of those are suitable for your needs, you can type in any name you like in the empty field at the top of the box. You then supply a value in the lower part of the dialog:



Click [OK] to save the changes, and then you're done. Make any other alterations and additions to the tags as required for this track, and when you're done, click the [Update file] button. This will again cause Foobar2000's console window to pop up with another spurious error ...if you get sick of them, click that window's [Options] button and uncheck the *Popup on warnings* option.

5.3 Bulk Manual Tagging

Highlighting and right-clicking individual files as I've shown you so far is obviously vital if you need to create or modify a track-specific tag (such as its title). But it's a bit of a daft way to go about things if you're trying to specify something which applies to an entire album (such as the album's name or genre). In those cases, you should instead highlight several tracks at once (click the first, hold the shift key, then click the last) and then right-click on one of them (it won't matter which one). Select the *Properties* menu option as before, and you'll see something like this:

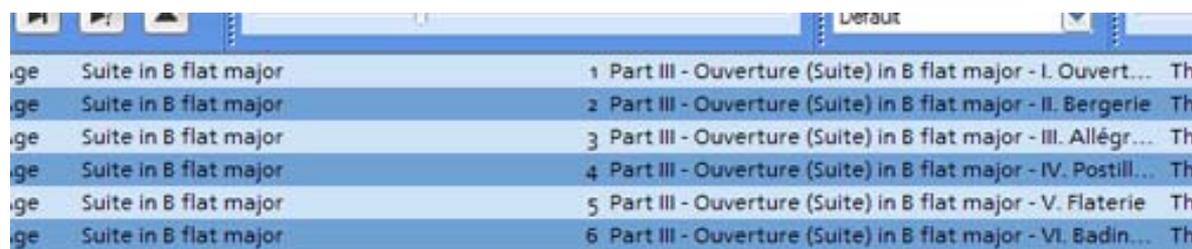


You'll notice here that most of the tags report 'varies or unchanged' values. That simply means that multiple tracks have, for example, different TITLES, so how can a one-line item in a dialog window display all of them at the same time? Because it can't, it doesn't try to.

However, if you double-click one of these tags, and edit its value as before, then the new value will apply to *all* the tracks that were selected before the *Properties* menu was invoked. In this way, you can quickly set album-specific tag values without having to do it all one track at a time.

5.4 Masstagger

Take a look at the results of my manual correction to the tags I downloaded from freedb:



I happen to have adopted an informal standard for my music ripping: if a single CD contains many 'pieces' or 'works', then each work is treated, and gets tagged, as a separate album. That might not be how you want your tags to appear, of course, but that's the beauty of choice. The point here, though, is that if the ALBUM tag tells me 'this is the Suite in B flat major', why do I

need that same information to be displayed in the TITLE tag for each track? And the answer, of course, is that I don't. Which would presumably mean I would have to right-click and edit the TITLE tags for each and every one of the tracks on this CD (and there are 24 of the them!): I'm likely to be finished sometime after Christmas!!

If this was any other media player that I know of, that's exactly what I'd have to do, and it's a tedious pain to have to do it –so much so that editing errors inevitably creep in. Fooobar2000, however, has a wonderful feature up it's sleeve called the **Masstagger**, and it makes stripping out unnecessary information from tags (and any other sort of bulk, automated editing you might need) a piece of cake.

It's not plain sailing though: Masstagger uses a scripting language that relies heavily on the sort of functions and syntax that we encountered when trying to specify a sort/display order for the album list. That means you need a coldly logical mind to use it successfully! In my case, for example: can you see from the previous screenshot that if I could somehow ask to 'strip out all characters up to and including the first full stop (period) in the TITLE tag', I would get rid of all that unnecessary stuff about 'Part III – Overture...', and just leave behind the suite's movement names: 'Overture', 'Bergerie' and so on.

If you now consult the **titleformat_help.html** document in your Fooobar2000 main directory, you'll discover that we have a function that can identify the position within a long string where a particular character is found: **\$strstr(x,y)**, which "Finds first occurrence of string y in string x" and (as the document doesn't tell you) returns that information as a number.

Given the TITLE tag value *Part III - Overture (Suite) in B flat major - I. Overture*, therefore, the function `$strstr(%title%, '.')` should return the value 49 (and in fact does... I checked!).

Well: that now gives us the *point* where we would like to cut the TITLE tag, but do we have a function that can do the actual cutting? Again, the **titleformat_help.html** document can be browsed to find that out, and you you will discover that there is a function **\$substr(a,b,c)** which "Returns substring of string a, starting from b-th character and ending at c-th character." Put like that, you'd be hard pressed to recognise anything much useful in that! But in plain English, it means that the function, when fed a long string A will skip through it until position B is reached, and then return anything from that point until the end of the string. In effect, anything before position B is forgotten about –or stripped away from the string, and that's exactly what we want.

Put those two functions together, therefore, and you come up with something like this in pseudocode:

```
$substr(%title%,$strstr(%title%, '.'),999)
```

I've thrown a '999' in at the end there, because I don't know how long the string (that is, the TITLE tag's contents) actually is for any track, but it's a fair bet it can't be further on than 999 characters! If I wanted to be a little more elegant about it, I could use the **\$len(x)** function, which returns the exact length of a given string, thus yielding this:

```
$substr(%title%,$strstr(%title%, '.'),$len(%title%))
```

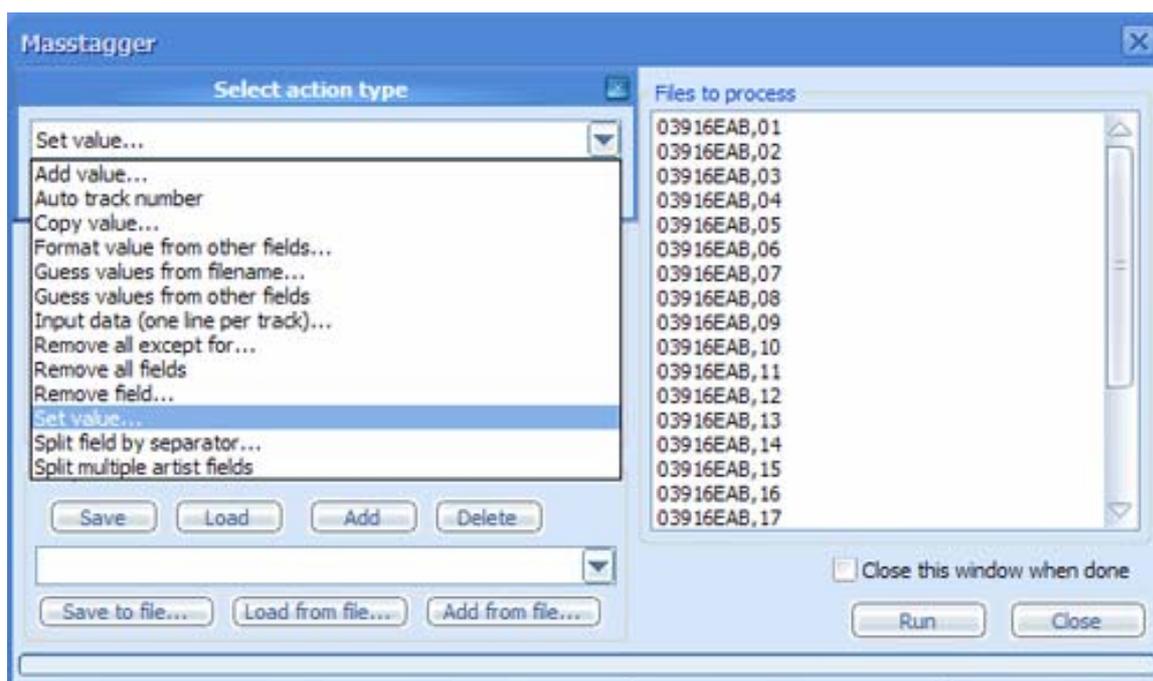
There's really just one final problem: the **\$strstr** function works out the position the full stop character is found within the TITLE tag... but the **\$substr** function returns the part of the tag which *starts* at that position –meaning that it will include the full stop we're actually trying to strip out. So, in fact, we want to add 1 to the position found by **\$strstr**. And, if you consult the screenshot again, you'll see that we actually, in this specific case, need to add 2 to it, using the **\$add** function, because it would be silly to start a TITLE with the space that appears after the full stop. Meaning that my final function should be:

```
$substr(%title%,$add($strstr(%title%, '.'),2),$len(%title%))
```

So that's the function I need (in this particular case, remember). The question is now merely one of how I actually employ it. And the answer to that is simple: highlight all the tracks to be edited, right-click and select *Masstagger* -> *Edit tags* from the resulting menus:



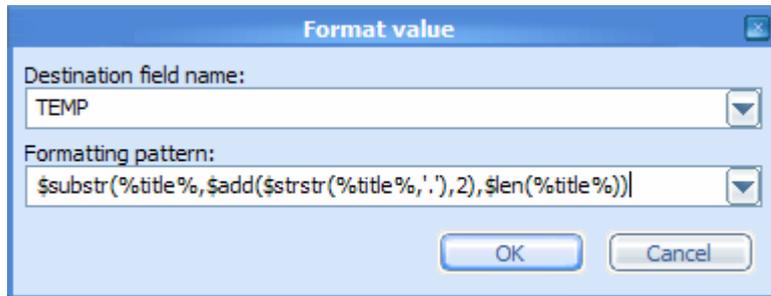
That will bring up a new screen where you can click the [Add] button, and then select one of the mass tagging functions that are available to you from the combo-box:



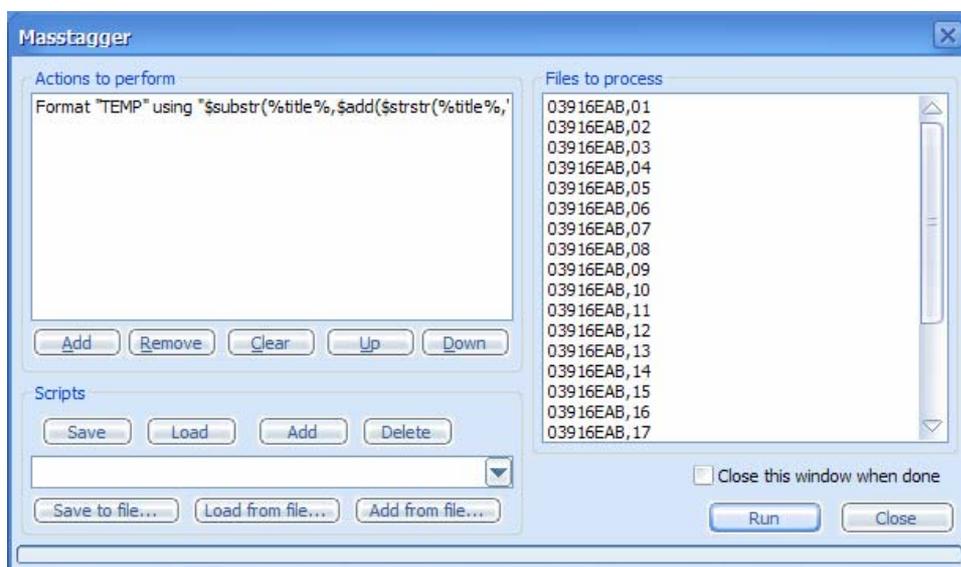
You'll notice that Masstagger can do things like remove a tag from multiple files in one pass, or derive values for unknown tags from known file names: it's a very powerful tool. In this case, I want to select the *Format value from other fields...* option, because I'm going to create ('format') a new TEMP tag from what the formula I worked out earlier makes of the TITLE tag.

You might think that's a bit long-winded: why not just use the formula to adjust the TITLE tag itself, but there are two good reasons for not doing it directly. First, the formula derives its value from the very tag it would be editing, which means you have a recursive edit on your hands... and, practically, it just won't work. But secondly, it is always risky to make bulk changes to data, so it's better to take it one step at a time so you can check your work before going further. That way you get accurate and desirable results, not a slapdash mess!

So, in my case, I'd select the 'Format value...' option, and then click [OK]. In the next dialog, I then specify TEMP as the destination field at the top, and the formatting formula I worked out earlier as the formatting pattern, like so:

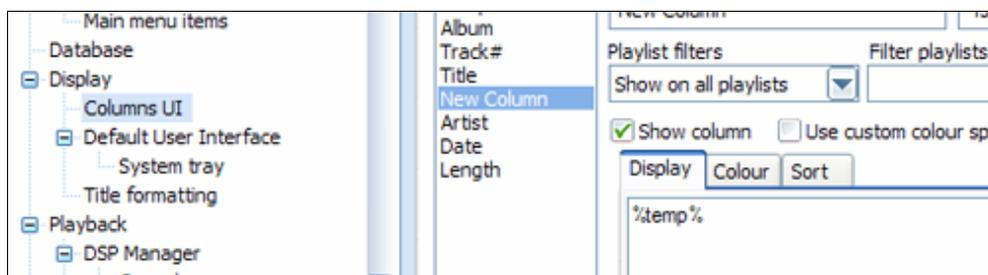


Once that's done, click [OK] once more. The results of your editing would look like this:



You can see the files which will be processed by this bulk tag re-format on the right (identified by their track numbers). All you have to do is to click the [Run] button, and the function will be applied to each track in turn. It takes just a second or so to complete its work.

Of course, you won't be able to see the results of its work, because all it's done is set the value of a new tag called TEMP to something, and that's not visible in the playlist. But you already know how to *make* it visible in the playlist: just navigate to the Columns UI preferences dialog, and create a new column whose display attribute is %temp%:

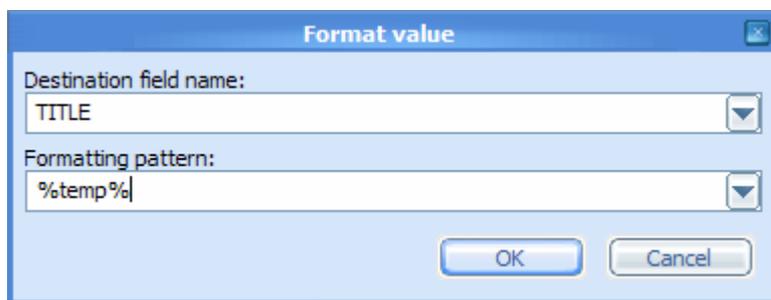


Once you click [Apply], you'll be able to see whether the formatting formula has worked correctly:

Title	New Column
Part III - Ouverture (Suite) in B flat major - I. Ouverture	Ouverture
Part III - Ouverture (Suite) in B flat major - II. Bergerie	Bergerie
Part III - Ouverture (Suite) in B flat major - III. Allégresse, vite	Allégresse, vite
Part III - Ouverture (Suite) in B flat major - IV. Postillons	Postillons
Part III - Ouverture (Suite) in B flat major - V. Flaterie	Flaterie
Part III - Ouverture (Suite) in B flat major - VI. Badinage	Badinage
Part III - Ouverture (Suite) in B flat major - VII. Menuet	Menuet

And if you compare the original title with the value for the new column that's displaying the value of the new TEMP tag, I think you'll agree that the formula has worked quite nicely!

Well, all that remains to do now is to set the TITLE tag to be whatever the TEMP tag is. So once again highlight all the relevant tracks, right-click, and launch the Masstagger to edit tags. Again, click [Add] and select the *Format value from other fields...* option. But this time, make the destination tag TITLE, and set the formatting pattern to %temp%:



Click [OK] and then [Run] and this time the real TITLE column should immediately show appropriate results. All that then remains to do is to delete the TEMP tag, as we no longer need it. So click [Clear] to wipe the 'Actions to perform' part of the Masstagger screen (and confirm that when asked), and then click [Add] once more. This time, select the *Remove field...* option, and when asked to select the field to remove, type in TEMP or select it from the combo box. Click [OK] and [Run] to make the change, then [Close] to return to the main Foobar2000 screen. You'll probably want to return to the Columns UI preferences dialog now to get rid of the extra column you added earlier to display the contents of the TEMP tag.

Now, of course, that's just a brief overview of what Masstagger is capable of –but even so, you've just seen a case of a quite complex formatting string being used to trim and polish tags retrieved from freedb, involving the creation of entirely new tags, the setting of one tag to the value of another, and the removal of tags deemed no longer necessary. It might all seem very complex when you first see it in action, but when you have hundreds of tracks and several tags per track to tidy up, this is the only tool capable of getting the job done. It all becomes just a matter of practice and experimentation (preferably not on real tracks with real tags you care about!).

What I've shown you here is a real example that is specific to my requirements, of course. But I hope you see the logic and principles that underlie this example, because those will be as applicable to you as to me!

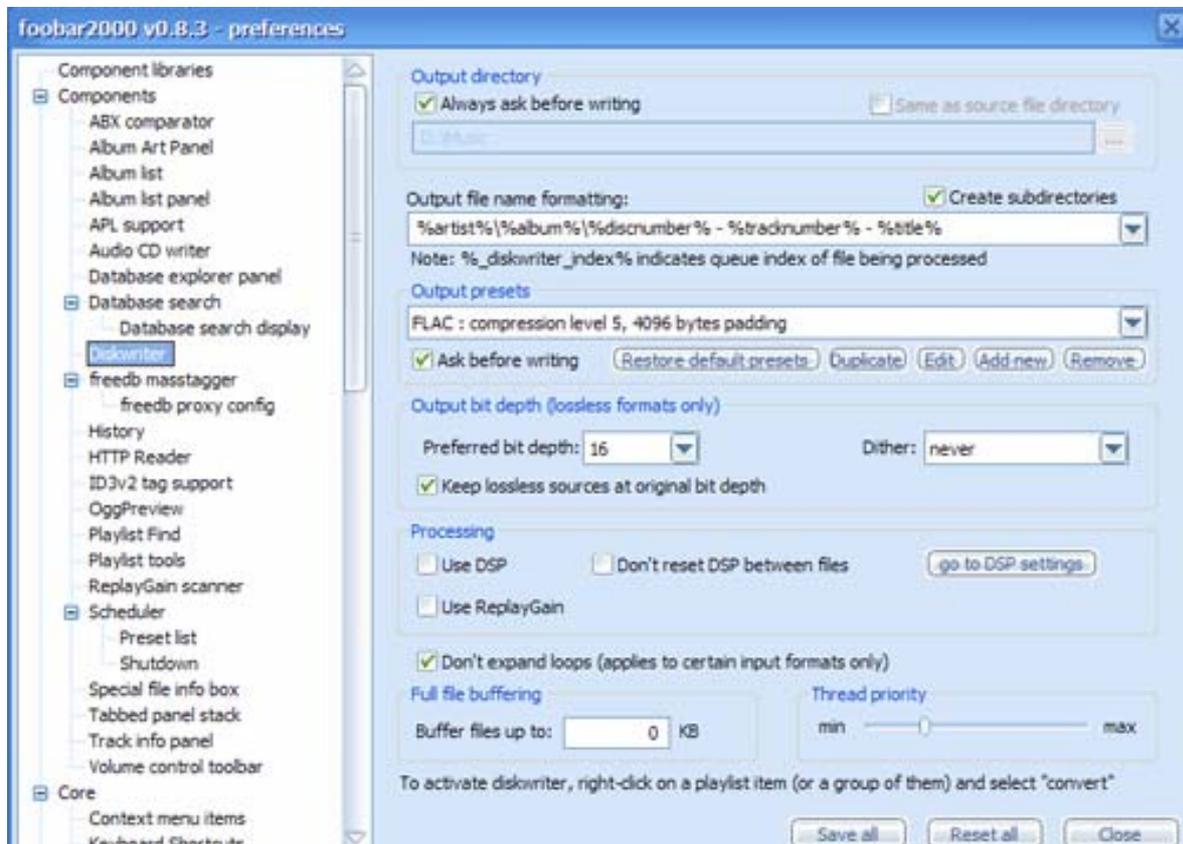
5.5 Doing the Rip

You might just have forgotten why we've spent so long getting the tags formatted so beautifully: it was actually so that we could rip tracks of music off CD and onto disk! Well, it's taken a while, for sure, but we are now ready to actually do that. Almost!

Before you do, you need to decide one or two things and configure the program accordingly. First, you need to know what audio format you are going to rip to: MP3, OGG, FLAC or something else. Secondly, if necessary, you need to know (and tell Foobar2000 about!) what quality settings you want the files ripped to: 64kbps, 192kbps or something else, for example.

Personally, with 300GB hard disks for sale at about \$1 per 1GB (this is Australia, incidentally, just in case you think those prices sound high), I've decided to rip to FLAC (the free lossless audio codec). The ripped files are enormous when compared to, say, an MP3 or an OGG, but they are lossless, meaning that they contain every last piece of data that was present on the original CD, and are thus, effectively, a perfect backup of the original CD. So in an age of dirt-cheap disk storage, the size of FLAC files is irrelevant, but their being perfect reproductions of the original CD is a nice plus for safety and permanence of my recordings.

You will go through a similar decision-making process, of course, and you might well decide things differently from me. That's cool: choice is a wonderful thing. Once you've made your choice, you need to tell Foobar2000 about it, which you do by using the *Foobar2000* -> *Preferences* menu options, and then locating the **Diskwriter** item in the *Components* branch of the options tree at the left of the screen:



There are large numbers of very esoteric options that can be set here. A lot of them involve tinkering around with the results of a rip (for example, by clicking the *Use DSP* option, the ripped file can be made to be louder or softer than the original, or have its stereo channels reversed, and so on). I don't bother with them: FLACs are supposed to be backups of CD tracks, not a chance to play recording engineer. But you might have a need for them.

The options you will definitely want to look at are towards the top of the screen: the *Output file name formatting* option lets you specify where to save the ripped tracks. You might be able to make out my entry here, but if not it reads:

```
[%artist%]\[%album%]\[%discnumber% - ]%tracknumber% - %title%
```

This means that my tracks will be stored in an album-specific sub-directory, underneath an artist-specific one, though the use of square brackets means that if any of the tags from which such information is derived is missing, that piece of the format will be missed out. But I'll ordinarily have a directory structure that reads something like

```
William Walton
Wolfgang Amadeus Mozart
Zoltan Kodaly
```

And within each of those directories will be new directories for each of the works or albums those 'artists' created:

```
D:\Music\William Walton>dir
Volume in drive D is Data
Volume Serial Number is 6431-60C2
```

```
Directory of D:\Music\William Walton
```

```
18/09/2005  10:12 AM    <DIR>          .
18/09/2005  10:12 AM    <DIR>          ..
18/09/2005  10:12 AM    <DIR>          Belshazzar's Feast
18/09/2005  10:12 AM    <DIR>          Cello Concerto
18/09/2005  10:12 AM    <DIR>          Coronation Marches
18/09/2005  10:12 AM    <DIR>          English Sacred Choral Music
18/09/2005  10:12 AM    <DIR>          Façade
18/09/2005  10:11 AM    <DIR>          Henry V
18/09/2005  10:11 AM    <DIR>          Music from Henry V
18/09/2005  10:11 AM    <DIR>          Three Overtures
18/09/2005  10:11 AM    <DIR>          Violin Concerto
```

And so on. Within each of these album-specific directories, files will exist that have names derived from their DISCNUMBER, TRACNUMBER and TITLE ID3 tags. So I end up with files named like this:

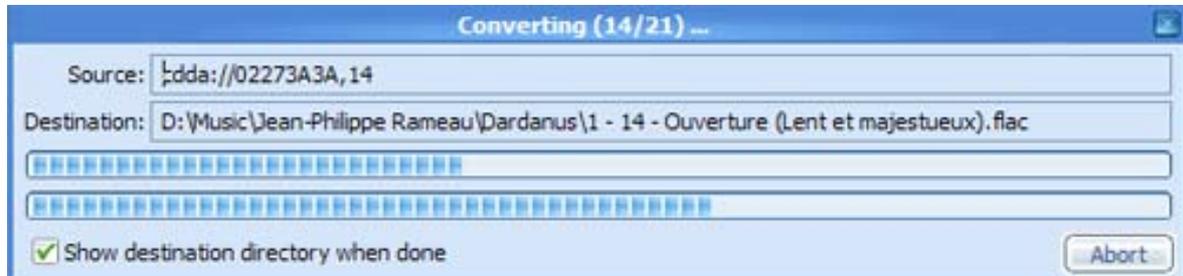
```
1 - 01 - Thus spake Isaiah.flac
1 - 02 - If I forget thee, O Jerusalem.flac
1 - 03 - Babylon was a great city.flac
1 - 04 - Praise ye.flac
1 - 05 - Thus in Babylon.flac
1 - 06 - And in that same hour.flac
1 - 07 - Then sing aloud to God.flac
1 - 08 - The trumpeters and pipers are silent.flac
1 - 09 - Then sing aloud to God.flac
```

Now, again, this might not suit everybody, and your naming ideas might be different from mine. But you have to think about these issues, and here is where you tell Foobar2000 about your deliberations.

The other thing you need to configure in the Diskwriter preferences is just beneath this file-naming one: it's the business of specifying which audio encoding you want to use for the ripped tracks. In the previous screenshot, you'd have seen me selecting one of the in-built FLAC options, but if you'd prefer to rip to OGG or MP3, this is where you make that choice. And if you feel the need to customise the various presets with which Foobar2000 ships –for example, to alter the default compression settings- that's what the small [Edit] button is for in the middle of the screen.

But apart from setting up the ripped files' naming conventions and the audio encoding to be used in this way, you're ready to actually rip your first CD using Foobar2000. Close the preferences screen, and return to the main playlist window. Highlight all the tracks that you

wish to rip, and then right-click. From the menu that now pops up, select the *Convert* option. After that, you'll be prompted to confirm your choice of audio ripping format (and can change it on-the-fly to one of the other in-built encoding schemes, if you prefer) and the location where the files should be ripped to. It's *within* this location that the artist\album sub-directories (or whatever other directory and file naming conventions you specified earlier) will get created. At that point, ripping will start:

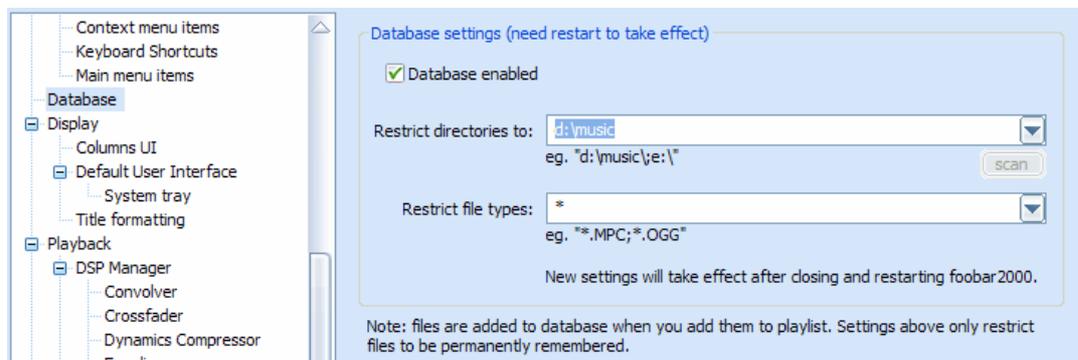


Once it's all finished, you can clear the playlist display, insert a new CD... and do it all over again, several hundred times if your CD collection is anything like mine!

6.0 The Album Database

Right now, the only way to get Foobar2000 to play music is to click *Playlist -> Add directory (or Add files, I suppose)*. Which is fine, but limited. Although it works well enough, it does mean that each new directory selected obliterates the previous one: the program never *remembers* what has been played.

Well, not by default it doesn't... but, like practically everything else, that's configurable. In this case, what you'll want to turn on is Foobar2000's **database** functionality. You do that by clicking *Foobar2000 -> Preferences* and then locating the **Database** item in the left-hand options tree:



As you can see, I've checked the *Database enabled* option, and then typed in the path to the root of where I've stored all my ripped music tracks in the *Restrict directories to* field. In my case, it happens that they are stored in D:\Music, which is exactly the location the helpful example note underneath that field mentions, but it can of course be anywhere on whatever hard disk you have available.

Note that I've left the *Restrict file types* field set to its default entry of *. In fact, it doesn't really matter in my case, since all my music is in FLAC format. But if you had a mix of formats in the one directory structure, here's how you could get Foobar2000 to notice, say, the MP3s without trying to remember the WMAs in there, too.

Having made these entries, you'd expect to then proceed with a scan for media by clicking the [Scan] button... except that you can't, because that button is disabled. Instead, you'll just have to

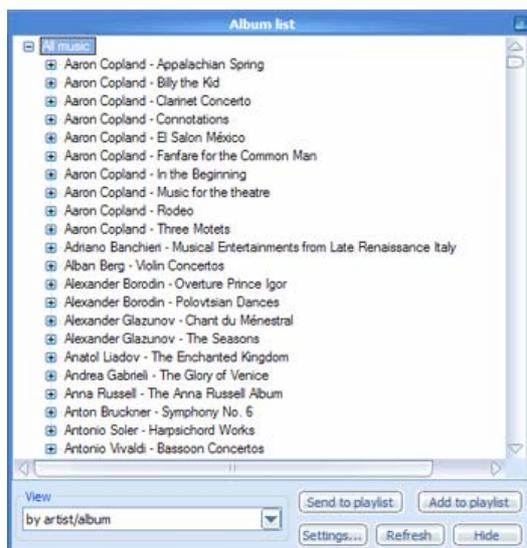
click the [Close] button, shut Foobar2000 down entirely, and then start it back up. Returning to this same screen, you'll now find the button operative, and can click it to begin the process of populating the database with details of all your music:



If you have many different music tracks, the scan at this point will take quite a few minutes. My 12,000+ tracks take about 10 minutes to scan fully, using a PC with an Athlon64 3500. Eventually, you'll be returned to the main options panel, where you can again click [Close].

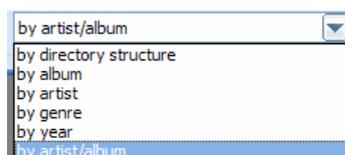
To display the contents of your new database, you need to get the Album List displayed. Do that by clicking Components -> Album List. Note that this option will work straight out of the box if you downloaded and installed the 'Normal' or 'Special' Foobar2000 versions, but if you instead downloaded the 'Lite' version, then it won't be available to you until you install it separately. To do that, obtain the Album List DLL⁷, and then copy the DLL to the Foobar2000 Components sub-directory (for me, that's found at C:\Program Files\foobar2000\components). Re-start Foobar2000, and you should find the new option is now available.

The Album List first appears like this:



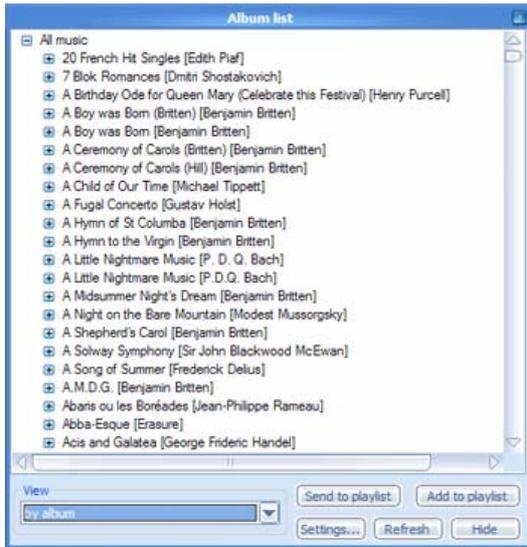
Remembering that the composer in my tracks actually occupies the ARTIST field, it might therefore be apparent that what we have here is a list of tracks grouped and sorted by ARTIST - ALBUM... which, in fact, requires no real detective work on your part after all, because that is precisely what it displays in the **View** combo box at the bottom of the screen!

Other sorting options can be selected from that same combo box, of course:

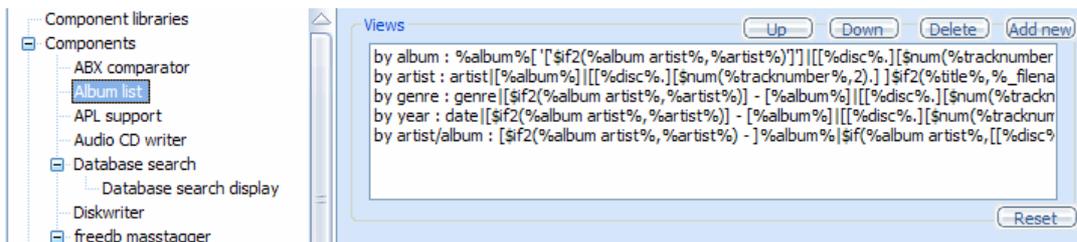


The "by album" option, for example, changes my playlist display to this:

⁷ From http://www.foobar2000.org/components.html#foo_albumlist



Foobar2000's ability to let you see your music data in so many different ways like this is a really powerful feature -but it's one that quite a lot of media players possess. Windows Media Player, for example, can let you view tracks by their Artist, Album or Genre. But where Foobar2000 really beats the competition is that if the various sort/group by functions are not sufficient for your purposes, you can create new ones! Click the [Settings] button, for example, and you'll be taken to this screen:



What you see in the right-hand panel here is the technical details of how the previous screen sorts music tracks: anything surrounded by the '%' character is a name of an ID3 tag, and anything that starts with a '\$' is the name of an in-built Foobar2000 string formatting function.

If you double-click the 'by album' item in the **Views** pane, for example, you'll see this:

```
%album% [ '$if2(%album
artist%, %artist%)' ] [ [%disc%.] [$num(%tracknumber%, 2) .]
] $if2(%title%, %_filename_ext%)
```

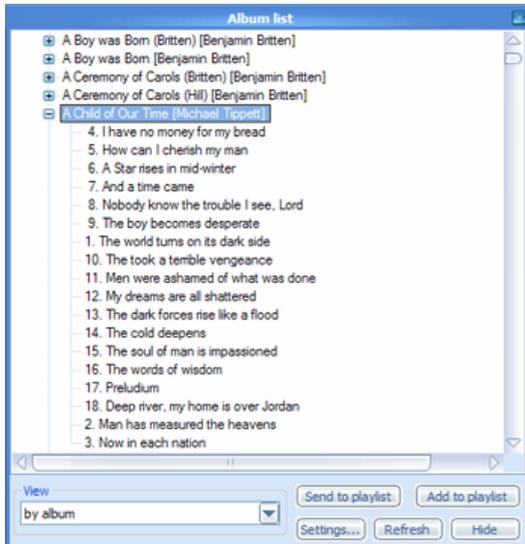
...which, being translated, means (roughly):

'Sort by whatever is in the ALBUM tag of each music track. When you display it, stick the artist name in square brackets after the name of the album, retrieving that information from either the ALBUM ARTIST or the ARTIST ID3 tag stored in the music file itself. If the user clicks on the name of an album, then display (and sort by) the disc number as found in the DISC tag, if it's present, and the track number stored in the TRACKNUMBER tag, fixed as two digits (so track 1 will be displayed as 01, for example), if present -and, of course, don't forget to display the actual track name as stored in the TITLE tag, though if that's not present, display the track's file name instead.'

Now, if any of that is not true for you, then feel free to alter it! The string is, after all, entirely editable. For example, I don't like track numbers being displayed as two digits, so I might alter that bit of the string to read:

```
[%tracknumber%.]
```

I will just display whatever is in the TRACKNUMBER tag, in other words, rather than use the \$num function to convert it to a number. Sounds good, but here's the result:



By removing the \$num function, the tag contents are displayed and sorted as text strings, which means that the numeric sort of tracks that logic and commonsense would demand flies right out of the window! 1 sorts together with 10 and 2 with 20, and the net result is that you wouldn't want to play this CD with this track order at all!

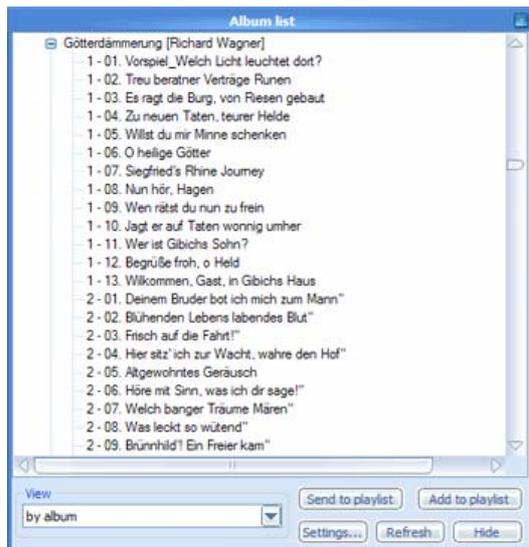
A more meaningful (and more successful!) change I could make to this string is the fact that when I ripped my CDs, I used the DISCNUMBER ID3 tag to store the CD's number, rather than just DISC. So the bit of format string you saw earlier that reads:

```
[[%disc%.]][$num(%tracknumber%,2).]
```

...is wrong for me, because DISC will never be present in my tracks, and hence (because the tag name is contained within square brackets) nothing at all will be displayed. So naturally, I want to edit that to be this:

```
[[%discnumber% - ]][$num(%tracknumber%,2).]
```

I've also taken the opportunity to separate the disc number from the track number by a hyphen rather than just a dot. If I now [Hide] the original album list and then use *Components -> Album list* to display it afresh, the results of the change are obvious:



So now tracks from different CDs within the same boxed set will display (and order) correctly, and track 1 from CD1 won't get confused with track 1 from CD2. But although this works for me, it might not work for you, because you might have ripped your music with different ID3 tags. So a more 'user-independent' version of what I've just done would better be written as:

```
%album% [ '['$if2(%album artist%,%artist%)']' ] | [ ['$if2(%discnumber%,%disc%) - ] [$num(%tracknumber%,2). ] $if2(%title%,%_filename_ext%)
```

Now I have "if there is a DISCNUMBER tag, display that, otherwise use DISC, and if neither is present, don't bother displaying anything".

The point of this example is to demonstrate that the configuration language that Foobar2000 and its components uses is immensely concise, powerful and configurable... but that innocent-looking changes can result in dramatic changes to ultimate functionality, so be warned! To learn all the subtleties of the configuration language is beyond the scope of this paper, but I hope the examples I've shown you nevertheless give you a feel for the basic principles:

- Stored ID3 Tags have names enclosed by the '%' character: e.g., `%album%`
- Derived values (that is, operating system or Foobar2000 internal attributes which are not actually stored as tags within the music track itself) have names enclosed by the '%' character, but with a preceding underscore: e.g., `_%filename%`.⁸
- String manipulation or logic functions have names that begin with a '\$'
- Tag names in square brackets mean 'display the tag if it's there, but omit it if not'
- Square brackets that are actually meant to appear as square brackets (and are not, in other words, merely enclosing tags for syntactical reasons, need to be quoted: '['
- Things which appear after the | character are only displayed when the stuff before it is clicked by the user, thereby 'expanding' it
- You can use 'if...then' logic to display either one tag or another. Hence

```
if2(%album artist%,%artist%)
```

...means 'display the album artist tag, but if that's empty, then display the artist tag instead'.

I've been using these syntactical rules to help modify and write Album List display strings, but the same rules will pretty much apply *anywhere* you start to modify how Foobar2000 should

⁸ Sometimes, preceding double underscores are used. A list of all the possible underscore and double-underscore attributes is contained within the `titleformat_help.html` document, which is found in the main Foobar2000 program directory. There's a list of functions there, too.

Fcs files are also a good way to share settings that work well or look especially good, and on the Foobar2000 forums, you will often see posters asking, 'Do you have that as an .fcs', or making their fcs files available for download. By importing them, their Foobar2000 configuration becomes yours.

In fairness, I should mention that with this technique you are only exporting configuration changes you make on the Columns UI tabs themselves. You aren't, for example, backing up any Album List sort/display order strings you might have invented, as I did in the last section for example. If you truly want to back up everything in sight, the only way of doing that which I know of is, literally, to shut Foobar2000 down, and then copy the `C:\Program Files\Foobar2000` directory in its entirety.

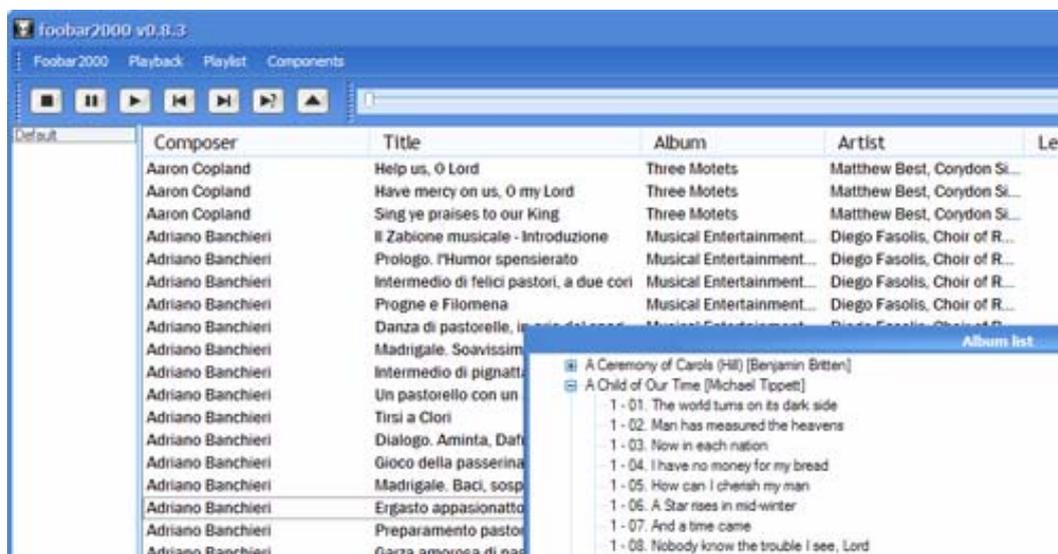
Restoring from this sort of backup is a bit tricky. The basic principle is, obviously, to delete everything in the program directory, plus all the sub-directories, and then copy their equivalents back from backup. The trouble is that Windows will report that it can't just delete the main program directory, because some of it is still in use (well, it does on my copy of Windows XP, anyway), even when you know for certain that Foobar2000 is not actually running.

Instead, you end up having to delete as much of the original installation as you can, by hand. I ended up being able to delete everything bar `foobar2000.exe` itself and some of the contents of the `icon` folder. Once that's done, you can copy everything bar those files back from the backup.

It's a bit messy, and it's not the kind of thing you'd want to have to do on a daily basis, but if you have a heavy investment in non-Columns UI modifications, it's about the only way to protect it.

8.0 More Advanced Configuration Options

At the moment, my Foobar2000 looks like this:



There are some quite nice play control buttons, the fonts being used in the play list are slightly better than the defaults, and the Album List window is displaying results in a customised sort and display order.

All of which is not bad, but I can't honestly say that my Foobar2000 looks very different from the default look and feel: it still looks pretty Spartan. If I had free rein, I'd like to see the Album List window appear permanently, somehow 'embedded' into the program as a display panel. I'd

also like an on-screen volume control. And it would be nice if I could display the CD covers ("album art") whenever I am playing a piece of music.

Well, all of these things (and more!) are possible, and I'll tackle them in turn.

8.1 Graphical Volume Control

Foobar2000 ships out of the box with a perfectly-functional volume control called *the numeric keypad*: just press your + and – buttons there to increase or decrease the volume! It works a treat, and requires no configuration. But if you want a graphical volume control displayed in the main program window, you can do that, too: simply download and install a new Foobar2000 component, called **foobar_uie_volume**⁹. Once downloaded, extract the file from the ZIP archive, and copy it to the Foobar2000 Components directory.

To activate the volume control, just make sure Foobar2000 is shut down and re-started, and then right-click on the main menu bar. You should see the following menu appear:

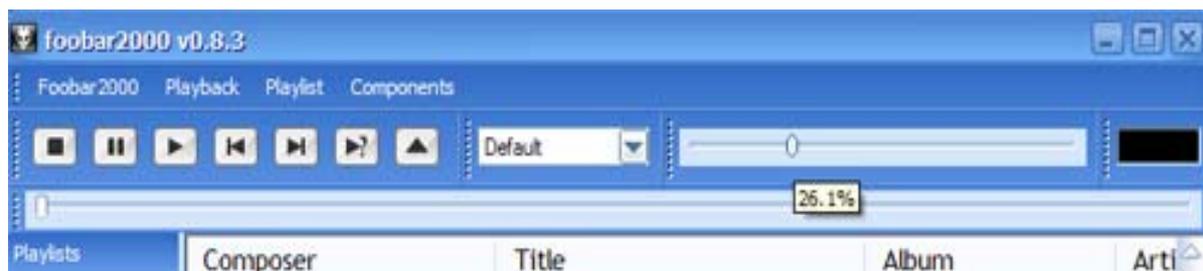


Under *Toolbars*, you'll find the *Volume control* option, which you should click –and a small volume control will appear somewhere on the screen as you do so:



Mine got squeezed right at the end of the row of existing controls. To make it bigger, or the other components smaller, just click on the dotted vertical bars at the edge of each one and drag left or right. You can also click on that same vertical bar and drag *down or upwards*, and the relevant component will be moved onto an entirely separate line, leaving more room for the others.

After a bit of fiddling, I ended up with this layout:

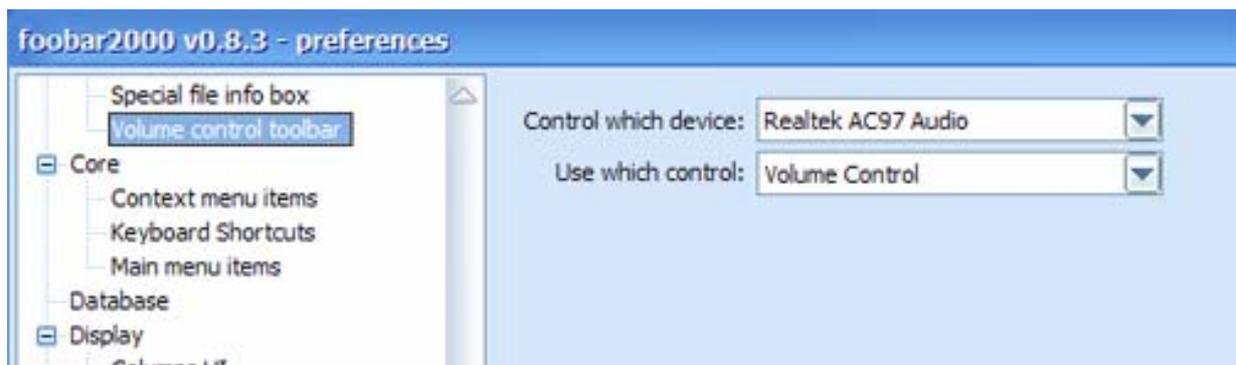


⁹ From <http://static.morbo.org/kode54/>

The volume control is on the same line as the playback buttons (and, incidentally, is showing that my volume is currently set to 26.1% of its maximum value). To make room for it, I dragged the similar-looking playback slide-bar (which indicates how far through a track you've played) onto its own toolbar line, underneath all the others.

The only other thing to do as far as the volume control is concerned is to configure what, precisely, is adjusted when you slide the control. Are you adjusting, in other words, only what Foobar2000 is outputting to your PC's sound card? Or are you actually changing the volume level of the sound card itself (in which case, what you change in Foobar2000 will, inevitably, affect every other program you run, too)?

Click *Foobar2000* -> *Preferences* and select the *Components* -> *Volume control toolbar* items in the options panel. You should see something like this:



Here, I've chosen to control the entire sound card –and I therefore also have to specify which bit of the sound card I'm controlling. Basically, the second combo box here contains all the options you would see on Windows' own volume control panel, and you can therefore select the master volume control, or the Wave one, and so on.

This is quite a neat example of how Foobar2000's functionality can be extended or enhanced by deploying additional components. Installing a new component is almost always a question of copying a DLL to the program's `components` sub-directory, re-starting Foobar2000 and then trawling around the preferences dialog to configure the new component's precise functionality. We'll be doing that a number of times in the future!

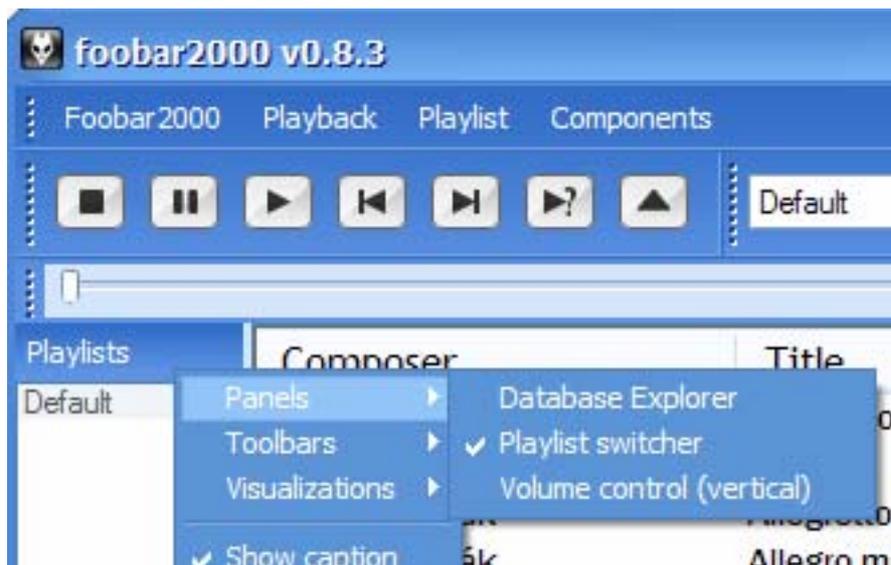
8.2 Embedding the Album Playlist

The Album List itself cannot be embedded within the main Foobar2000 player, but something very similar to it can be. You'll first need to download and install another Foobar2000 component, called `foo_uie_dbexplorer`.¹⁰

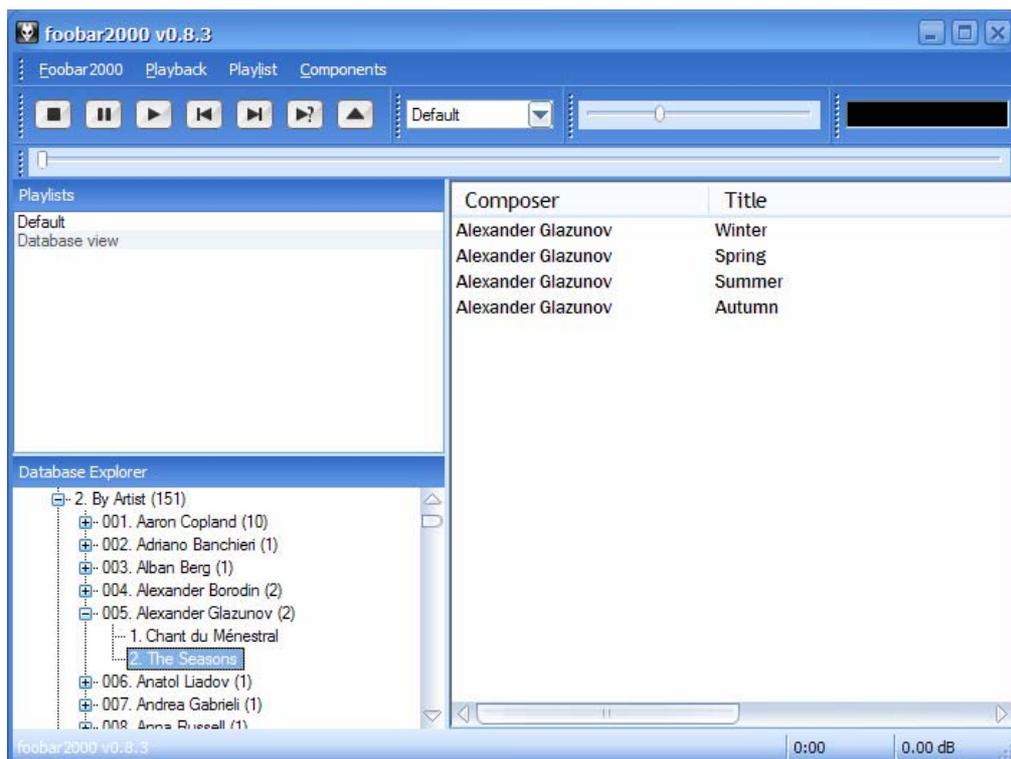
The download is actually a zipped file, and can be opened in WinRar or Winzip (it's odd extension comes from the fact that it was encoded with a free zipping utility called 7-Zip). As usual, extract the `foo_uie_dbexplorer.dll` file from the download, copy it to the Foobar2000 `components` sub-directory and then re-start Foobar2000. To begin with, nothing will have altered, and it won't be obvious that any new functionality is available to you.

But, in fact, it is. Just right-click on the panel which is displayed to the right of the main playlist window:

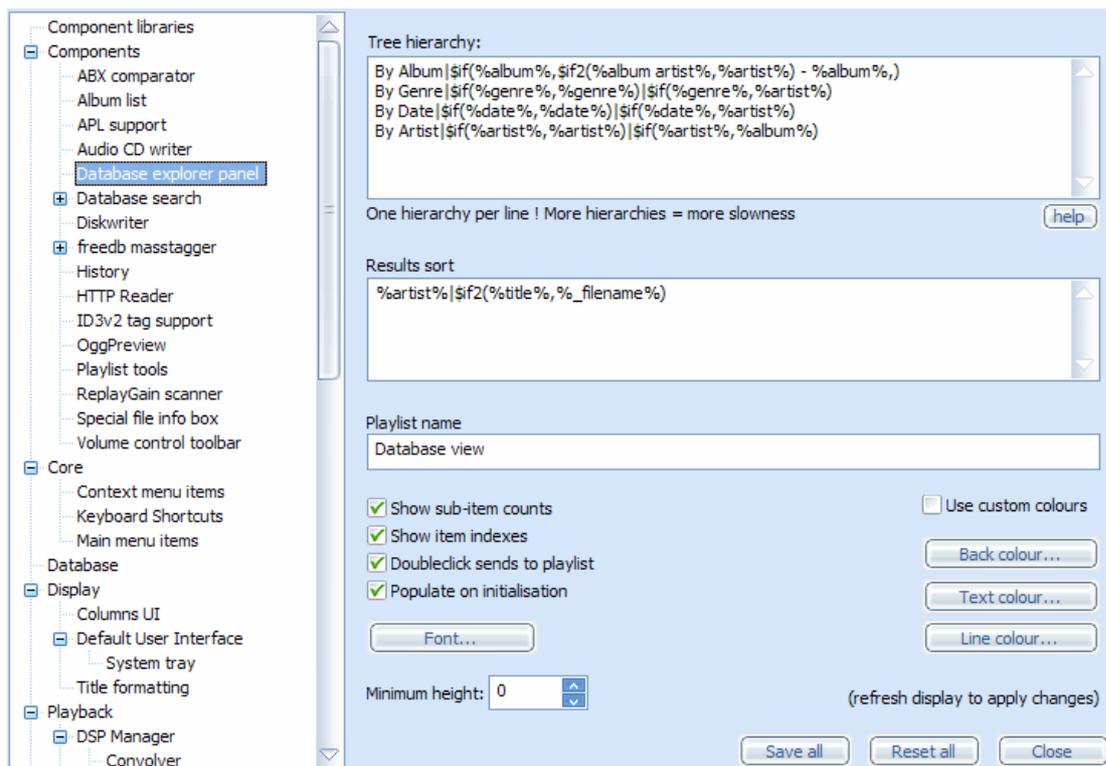
¹⁰ From <http://tinyurl.com/dajok>



Click the *Database Explorer* panel option, and your Foobar2000 main display will suddenly acquire a new area in which something very similar to the Album List is displayed. You will probably want to re-size your program window and its various component panels to make the new one fit properly, but eventually you might end up with something that looks a bit like this:



By clicking the '+' next to an item, its child components are displayed. Clicking on one of those causes music tracks to be added to the main playlist. As with the volume control, there are many options to configure for the new component—all inevitably found by clicking *Foobar2000* -> *Preferences* and then locating the Database explorer panel item in the options list:



Just looking at the buttons available to you here, it is clear you can change the font the new panel uses, as well as its colour. You can also see that the kind of sort/display options we configured earlier for the real Album List panel could come in useful here as new or modified tree hierarchies: they use much the same sort of syntax, after all. There's also a useful note to warn you that having lots of possible hierarchies slows things down: so I tend to delete most of them, and paste in the Artist/Album display string from the Album List instead. But, as ever, the real point here is that the thing is as configurable as any other part of Foobar2000, and it all comes down to experimentation and how imaginative you want to get.

8.3 Album Art

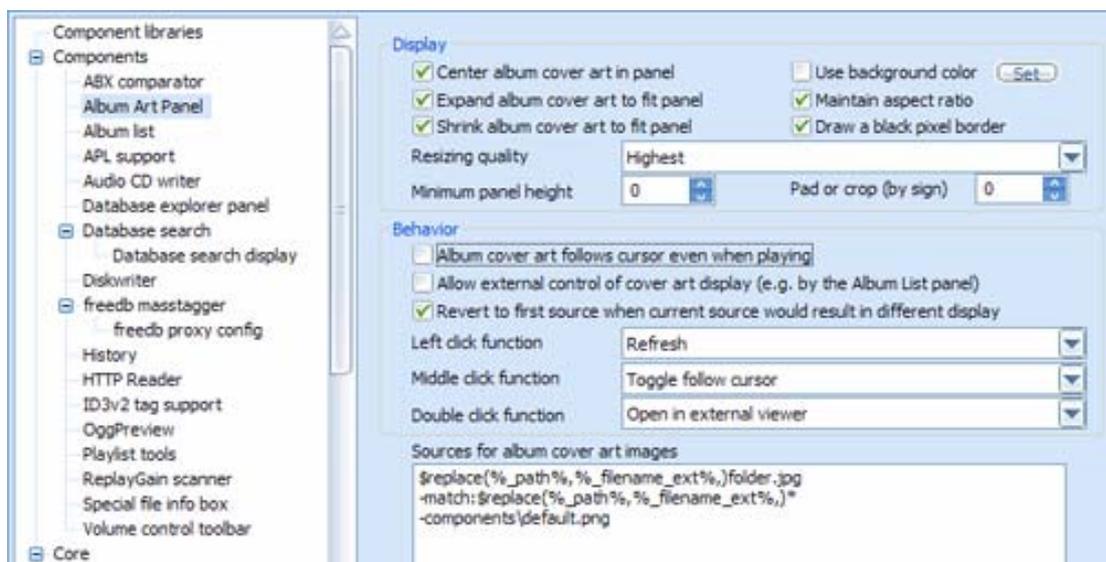
The final part of my 'ideal configuration' for Foobar2000, if you can remember that far back, was that it should be possible to display some album art when a track is playing. Well, that too is do-able with the help of yet another additional Foobar2000 component. This time, you'll want to download and install **foo_uie_albumart.dll**¹¹

As ever, extract all three files from the zip download, and copy them to the Foobar2000 Components sub-directory, and then re-start Foobar2000. You'll need to add the Album Art panel to your program window by right-clicking in the panel area to the left of the main playlist. Select the *Panels* menu option and then click on the *Album Art* option. Don't switch on the *Album list* panel just yet, though. Again, you may need to re-size some existing panels to make the display look OK. After that, once you start to play a music track, you should see its album art appear:

¹¹ From <http://pelit.koillismaa.fi/plugins/interface.php#128>.



The component by default assumes all album art is found in the same directory as the music track itself, and is called **folder.jpg** (which is what, for example, Windows Media Player would have assumed too, so the naming convention is in fact something of a standard). However, and somewhat inevitably, you can configure the way the new panel works by selecting *Foobar2000* > *Preferences* and then the **Album Art Panel** item in the options on the left:



Perhaps the most useful options to set here are those which expand or shrink the album art so that it always fits in the panel, but at the bottom of this screenshot you'll see how you might go about telling the component that all your album art is called, say, *cover.jpg* rather than *folder.jpg*: just edit the 'Sources' string. The syntax it uses should be broadly familiar to you from the sort of work we did with display and sort strings earlier, but specifics on how it works can be found in the Forum discussion for the component.¹²

The *Album cover art follows cursor even when playing* option is interesting: left unchecked, as it is by default, it means that the only album art displayed is that for the currently playing song.

¹² At <http://www.hydrogenaudio.org/forums/index.php?showtopic=29383>.

Click on other albums in the playlist window, and the album art won't change. Switch it on by selecting this option, though, and the album cover for the currently playing tracks will disappear, to be replaced by the album cover for whatever tracks you happen to have clicked on.

Awkwardly, I sometimes like the component to behave in both ways at different times. If I'm browsing for the next album to play, for example, I would like to see the cover art for whatever albums I happen to click on. Once I've had enough of 'browse mode' though, I would prefer to revert to displaying the cover art for whatever track happens to be playing. Well, by configuring things as you see in the screenshot, one click of my mouse scroll wheel (which counts as the mouse's middle button!) yields precisely that result.

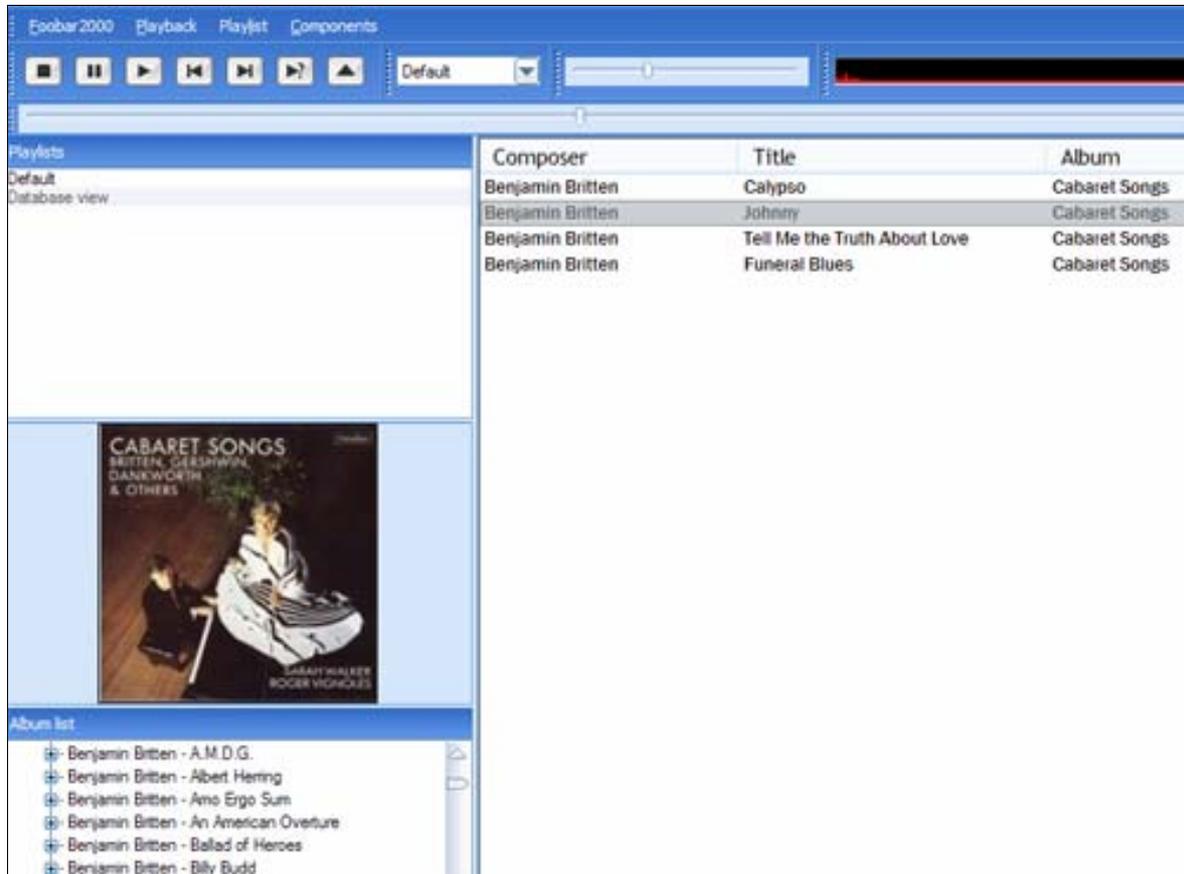
You may recall that the Album Art download also shipped with an *Album List* panel component. You might like to switch that on now by, once again right-clicking anywhere in a title area of one of the existing panels in the area to the left of the main playlist display, and selecting the *Panels -> Album list* menu options. You'll see a panel which looks very similar to the *Database Explorer* one, and this might give you an idea that different people develop different components to achieve the same sort of outcomes. The net result is a sometimes a bit of overlapping or duplicated functionality –but, of course, you can resolve that easily enough by switching off panel(s) you no longer require.

I tend to get rid of the *Database Explorer* panel at this point and keep the *Album list* one because I find that with the *Album list*, I can browse purely within the panel itself, and yet the album art will correctly reflect what I'm clicking on. With the *Database Explorer*, however, I have to select items in the panel, and then additionally have to click tracks in the main playlist area before the cover art changes. The extra clicking is a pain, and I prefer not to have to do it. But that's just me: it all comes down to a matter of personal choice, and you may want to experiment with both to see which best suits you, your music collection and your track selection habits.

Finally: don't forget that the sort/display strings we worked out earlier for the separate album list pop-up window can be used verbatim as sort/display strings for the embedded album list panel. Just go into the Foobar2000 preferences and cut-and-paste the strings from one into the other.

9.0 Restructuring the layout

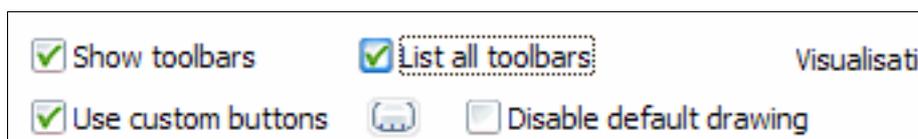
Currently, if you've been following the story so far, you'll have a Foobar2000 layout that looks a bit like this:



I've got here the various panels we've just been discussing on the left; album art is being successfully displayed for the tracks selected from the Album list panel; there's a volume control next to the spectrum analyser display, and the play slider control has been re-located to beneath the playback controls. So far so good... but it's looking a bit cramped down the left-hand edge. If we start switching on any more panels, we're going to be in serious space difficulty.

Well, there are a couple of things we can do about that. For starters, we can move the panels so that they appear horizontally *above* the main playlist area. Secondly, we might consider *stacking* some of them, and accessing them via tabbed dialogs.

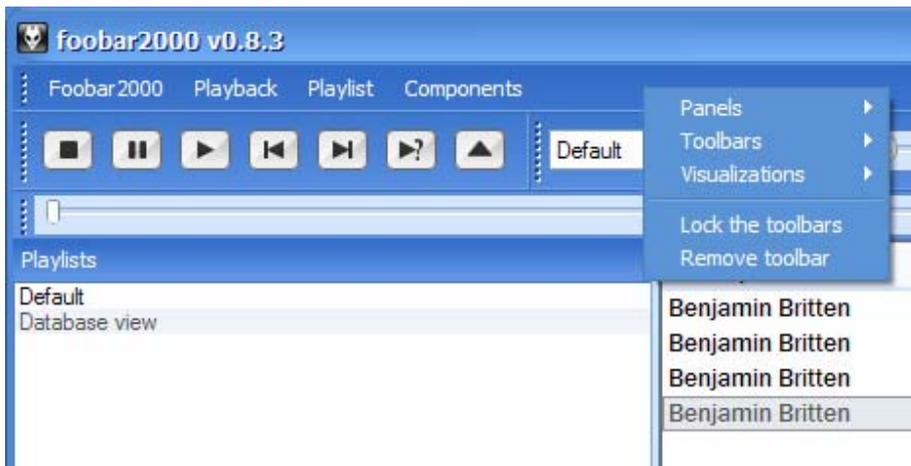
To move panels above the main play area, click *Foobar2000* -> *Preferences*, and find the *Display* -> *Columns UI* items in the options tree on the left of the screen. Go to the **Other** tab:



The option you need to enable is the List all toolbars one, as I've shown here. Doing so will cause the following warning dialog to appear:



Although this sounds rather drastic, you can ignore it. Just click [Yes] to continue. Once enabled, the option means that panels can be embedded onto the main program toolbar, rather than just in the left-hand strip we've been using till now. So, you can click [Apply] and then [Close], and you should immediately be able to right-click anywhere along the top toolbar:

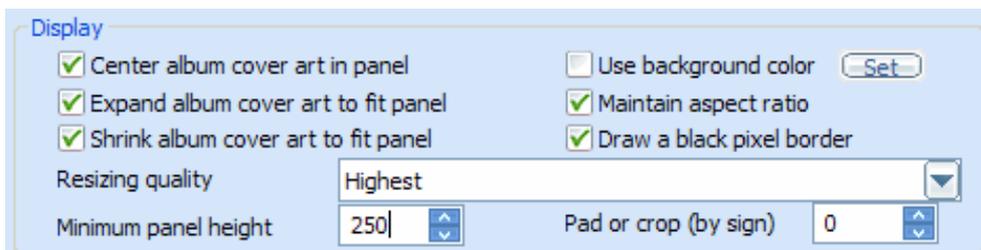


Once you can do this, you can proceed to switch on each of the panels as before, only this time they will now appear along the top of the program window, rather than the side:

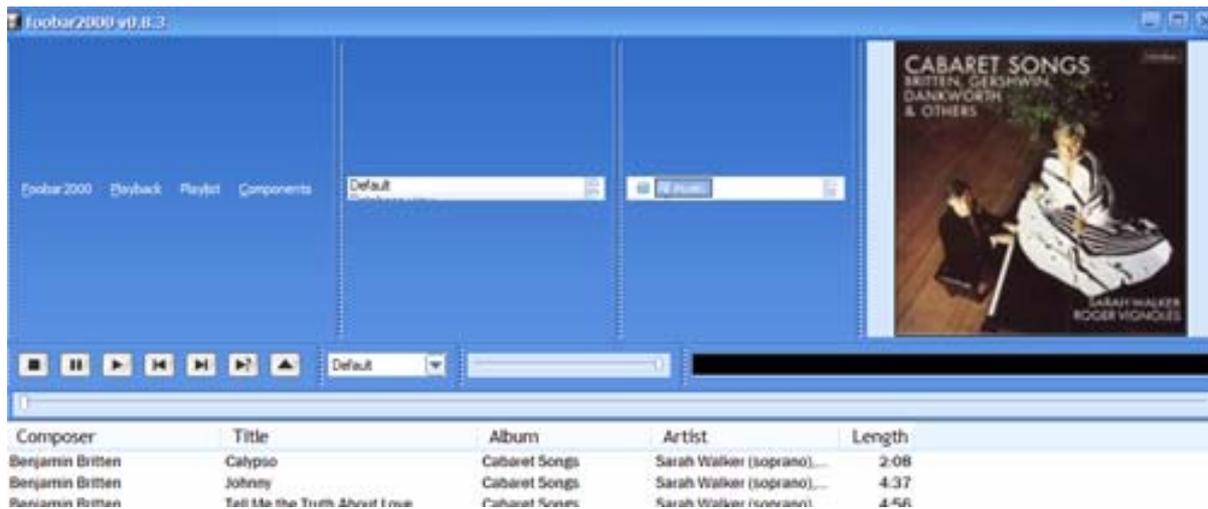


You might not recognise them immediately, but here are the *Playlist Switcher*, *Album list* and *Album Art* panels, running horizontally along the top of my Foobar2000 program window! What makes them so ridiculously difficult to spot, of course, is that they have been created at a daft size... and that needs fixing if this 'horizontal panel' capability is to be really useful.

Well, that's relatively easily done by going back to the *Foobar2000* -> *Preferences*. Find the *Components* -> *Album Art Panel* item in the options tree, and specify a minimum height of, say, 250 pixels:



As soon as you do this, the album art will be resized to the specified minimum height, and with a bit of jiggling around with panel widths, you'll end up with something that resembles this:



Now, clearly this is just a start! The album cover art might be looking a reasonable size these days, but none of the other panels do... and they've made the main toolbar so high, the poor old menu options look stranded!

Well, let's deal with that last problem first: just click-and-drag the Album Art panel down to beneath the toolbar, but above the playlist, and all should be well:



And whilst that's not particularly attractive either, it is at least a start! At least you now know how to re-position panels above the main playlist area.

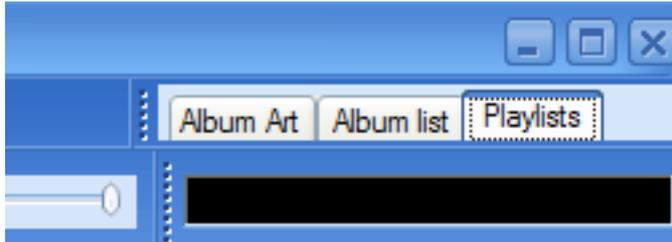
9.1 Stacked Tabs

To fix the problem of the other tabs being so small, we need to download and install one more Foobar2000 component: this time, one called **foo_uie_tabs.dll**.¹³ Somewhat uniquely as far as this article is concerned, the download is not a zipped file, so can be saved directly into the Foobar2000 components directory. If you then shutdown and re-start Foobar2000, you can right-click on the main toolbar as before, and this time you should have a new *Layout* menu option available to you in the pop-up menu: *Tabbed panel stack*. Switch that on by selecting it, and another new panel will appear on the main toolbar. It won't look very exciting, but that's because it needs a bit of configuring before it warms up!

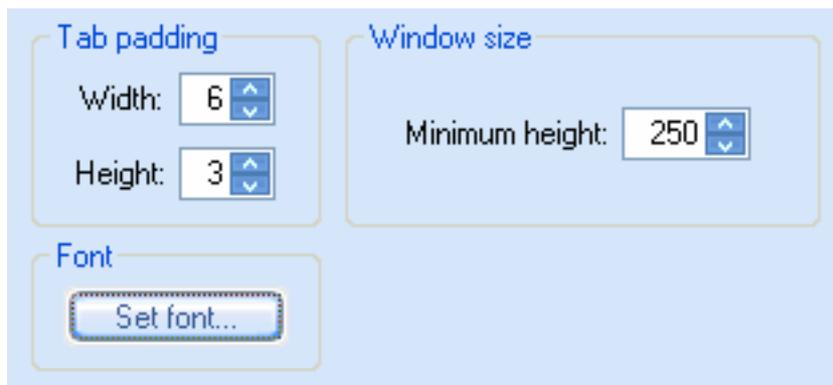
¹³ Available from http://vallara.cs.rmit.edu.au/~kspear/foobar/foo_uie_tabs.dll

First, let's undo all the work we've done so far (it's OK, I planned it this way!). Right-click the main toolbar, and switch off the *Album Art*, *Album list* and *Playlist Switcher* panels. That means we're starting with a clean slate once more. Now right-click the main toolbar once more, and select the new menu options that are available to you because of the installation of the new component: *Layout* -> *Tabbed panel stack*. Things still won't look very exciting just yet, but we're nearly there!

Right-click inside the new panel that has appeared in the main toolbar, and select the *Add tab* -> *Panels* menu options. Click on one of the items now listed (for example, the *Album Art* one), and that will be added as a tab into the new toolbar panel. Repeat the process until all three panels are added into the tab:

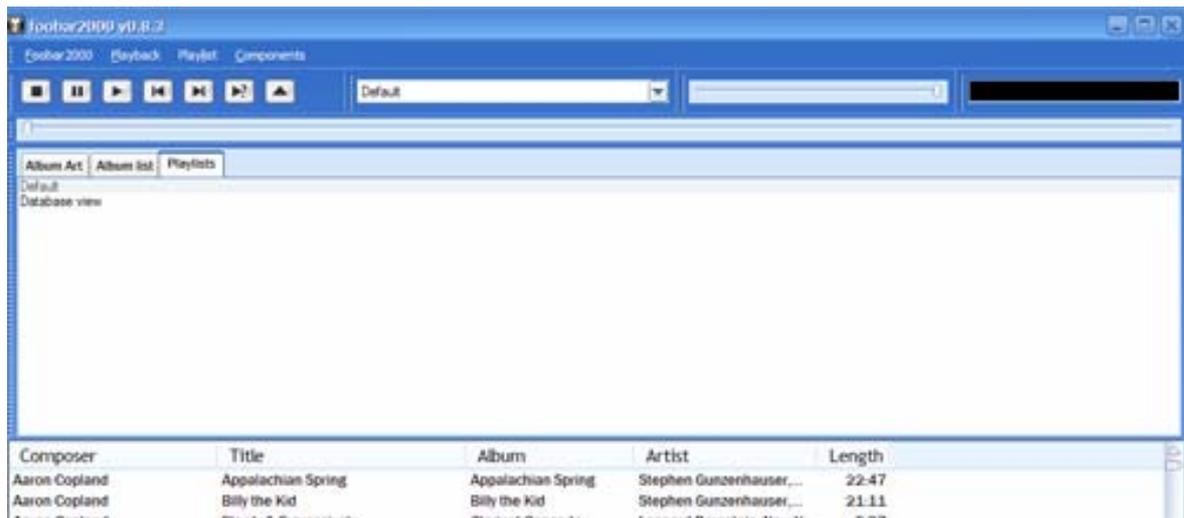


Next, we need to establish a minimum height for the tabbed panel, just as we did earlier for the *Album Art* panel on its own. From the *Foobar2000* -> *Preferences* menu options, you'll find an item for configuring the *Tabbed panel stack* under the *Components* branch of the options tree:



Specify a minimum height here of, once again, about 250 pixels, and you might also like to take the opportunity to set a font for the tabs (I again used Franklin Gothic Medium for my examples here).

As before with the *solo* *Album Art* panel, you'll find that the new minimum height for the tabbed panel causes layout grief for the other program components and menus. But you also now know how to fix that: just grab the new panel and drag it downwards until it's where it needs to be. Mine ended up beneath the play slide-bar, looking like this:



By clicking on the various tabs, you can switch the contents of one large display area between album lists, album art and playlists –and because it’s one large panel that’s doing duty for all three functions, there’s no longer an issue of lack of space for the information it contains.

Of course, eagle-eyed observers will note that I now can’t select an item in my Album list panel and have the corresponding album cover displayed in the Album Art panel... because by displaying the list, I must be hiding the art. Well, there’s method in my madness: you’ll now learn how to *remove* something from the tabbed panel and put it back as a standalone panel!!

If you right-click the Album Art tab itself, you’ll discover an option to *Close tab*. Take that option now, and the tab is effectively removed from the stack. Now right-click in the main toolbar area, and select the *Panels* menu option. Select the Album Art panel once again, and album art will again be displayed, but this time as a standalone panel which you can drag down to sit alongside the tabbed ones:



You didn’t have to adjust the height of the Album Art panel this time, because the configuration you previously did was still there and still usable. So, whilst still far from perfect, I think you will agree that this is starting to look seriously usable... and certainly something of an improvement over what we started with.

10.0 Playlist Formatting

At the moment, the main playlist area is pretty dull:

Composer	Title	Album	Artist	Length
Savage Garden	Affirmation	Affirmation	Savage Garden	4:56
Savage Garden	Hold me	Affirmation	Savage Garden	4:51
Savage Garden	I knew I loved you	Affirmation	Savage Garden	4:12
Savage Garden	The best thing	Affirmation	Savage Garden	4:19
Savage Garden	Crash and Burn	Affirmation	Savage Garden	4:41
Savage Garden	Chained to you	Affirmation	Savage Garden	4:08
Savage Garden	The Animal Song	Affirmation	Savage Garden	4:40
Savage Garden	The lover after me	Affirmation	Savage Garden	4:50

It's pretty basic –and you wouldn't, for example, know from this display which of the possible tracks it was that was actually playing. It also seems a bit daft to repeat the album name 15 times, just because it happens to be made up of 15 tracks, and the same can be said for what I've called the 'Composer' here, but which you may recall is actually the ARTIST ID3 tag.

Well, we saw quite a while ago how to alter the column names, positions, and fonts... can we use the same sort of thing to resolve these sorts of issues? And, as you might have expected, the answer is 'yes' –but I think it fair to say that it is extremely tricky to get right!

The secret is to configure 'formatting strings' for each of the playlist columns and for the playlist globally –and format strings use the sort of syntax you saw earlier when we were configuring a sort/display order for the Album List dialog. You might recall that such syntax is very powerful, but rather obscure and difficult to read and interpret correctly.

10.1 Setting Playlist Colours

So let's start very simply: how about simply changing the colours of our text? To achieve that, you have to know that in the colouring format string, the colours are specified like so:

text-colour|selected-text-colour|background-colour|selected-background-colour|dividing-line

...with the | character distinguishing each colour from the next. Knowing that, you can perhaps now make sense of this formatting string, which is the default one provided in the Columns UI setup:

```
000000|555555|$if(%_isplaying%, $rgb(220,220,220), $rgb(255,255,255))|$if(%_isplaying%, $rgb(204,208,212), $rgb(230,234,238))|$if(%_isplaying%, $rgb(204,208,212), $rgb(240,244,248))
```

It's not easy, is it?! But it means the text colour is black (that's what 'ooooo' gets interpreted to mean), and the selected-text-colour is 55555, which (if you have Photoshop handy and can check RGB values) is a sort of dull, muddy dark grey. So my track titles start off jet black, but if I click on one, its text turns dark grey-ish, and you might be able to make that out in this screenshot:

Composer	T	Title
Aaron Copland	6	Help us, O Lord
Aaron Copland	7	Have mercy on us, O my Lord
Aaron Copland	8	Sing ye praises to our King

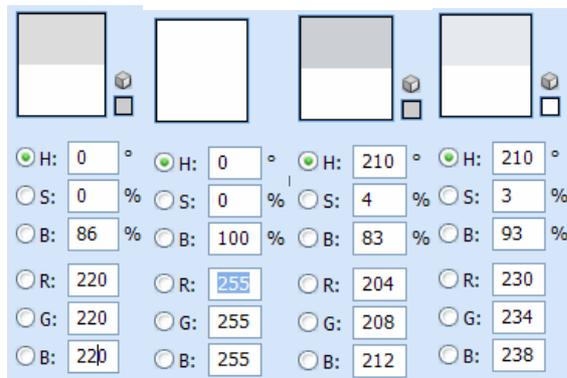
Now, of course, something else is happening here too: the selected track has a light-grey background, whereas the unselected ones have a white one. Well, that's what the rest of the formatting string I showed you earlier means. Remember, colours are specified in this order:

text-colour|selected-text-colour|background-colour|selected-background-colour|dividing-line

000000|555555|\$if(%_isplaying%,\$rgb(220,220,220),\$rgb(255,255,255))|\$if(%_isplaying%,\$rgb(204,208,212),\$rgb(230,234,238))|\$if(%_isplaying%,\$rgb(204,208,212),\$rgb(240,244,248))

Now, we already know that text-colour = 000000, and selected-text-colour is 555555. That means **\$if(%_isplaying%,\$rgb(220,220,220),\$rgb(255,255,255))** must be the normal background colour, and **\$if(%_isplaying%,\$rgb(204,208,212),\$rgb(230,234,238))** is the selected background colour.

Again, using Photoshop, I can show you what colour each of these RGB values represents:



So, basically, we have three shades of grey and a pure white. Why so many colours? Well, the logic of the formatting string is 'I am setting the background colour: if this track is actually being played right now, I will set it to mid-grey, otherwise it's bright white; and if the track is actually selected/highlighted *and* being played, then I'll set it to darkish grey, otherwise to quite light grey'.

Now, knowing all of that, how would I turn the currently-playing song a shade of blue, rather than grey? Well, first define your colours! When doing this, you'll need to be able to use some colour selection tool that allows you to specify RGB values and see their meaning. I use Photoshop, but Foobar2000 gives you access to the standard Windows colour chooser: just click the Tools button on the *Foobar2000* -> *Preferences* -> *Display* -> *Columns UI* -> *Globals* page.

Now, of course, we're into deep artistic territory at this point (which rather rules me out of the game!) So you might well end up with a different choice of colours than me, but I decided I'd go for:

text-color 00 00 00
selected-text-color 52 61 99
background-color 112 161 211
selected-background-color 192 216 240
dividing-line 00 00 00

text-colour|selected-text-colour|background-colour|selected-background-colour|dividing-line

Meaning that my colouring string should read:

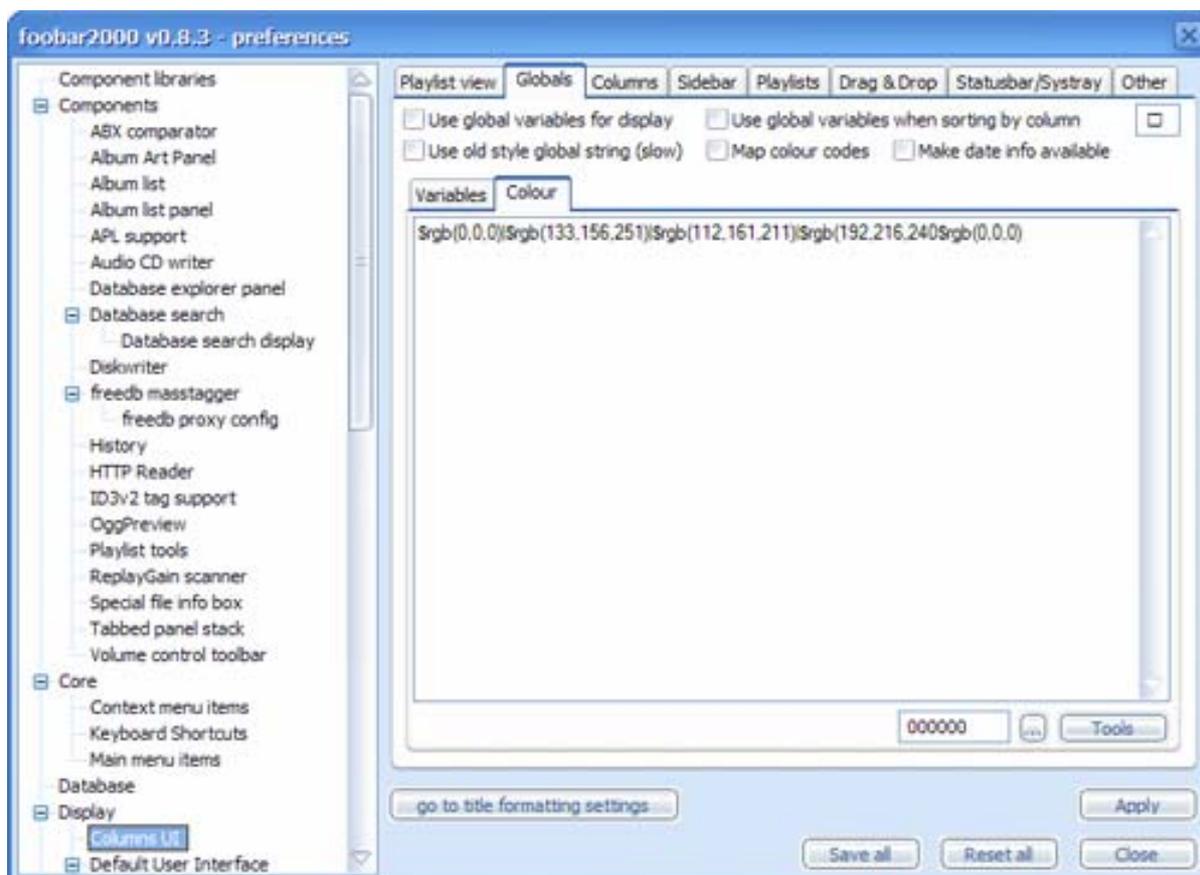
\$rgb(0,0,0)|\$rgb(133,156,251)|\$rgb(112,161,211)|\$rgb(192,216,240)|\$rgb(0,0,0)

...which results in a display similar to this one:

Composer	T	Title	Album	Artist	Length
Aaron Copland	4	Prologue	Music for the theatre	Leonard Bernstein, New Y...	6:27
Aaron Copland	5	Dance: Allegro molto	Music for the theatre	Leonard Bernstein, New Y...	3:29
Aaron Copland	6	Interlude: Lento moderato	Music for the theatre	Leonard Bernstein, New Y...	5:47
Aaron Copland	7	Burlesque: Allegro vivo	Music for the theatre	Leonard Bernstein, New Y...	3:40
Aaron Copland	8	Epilogue: Moderato - Molto moderat...	Music for the theatre	Leonard Bernstein, New Y...	4:14

...which is not brilliant, but at least it's a start... or, at least, it would be if I'd told you where to paste in the new colour string! (And before you do this, you might well want to take a fresh backup of your Columns UI configuration, because this is the point at which things can start to go horribly wrong!)

Well, I'm pasting them in here:



You get here by clicking the *Foobar2000* -> *Preferences* -> *Display* -> *Columns UI* -> *Globals* tab, and then selecting the 'Colour' sub-tab (which, I am happy to note, is spelled correctly!). Paste in the new string there, and click [Apply], and you'll see the effects of your changes instantly (there's no need to re-start Foobar2000, for example).

Now we might like to refine this a little: having one colour for the background can make it difficult to read across the lines of the playlist. So how about alternating the colours a bit? That is easily done by working out whether the playlist is displaying an even or an odd-numbered line, and picking a different colour depending on the answer.

Now: your programming skills might be as bad as mine are, but you might nevertheless somehow know that $\text{mod}(x,2)$ will determine whether a number is odd or even by returning the remainder of the division of that number by 2: even numbers will return a 0, and odd numbers won't. And, if you read the *titleformat_help.html* document in your foobar2000 main directory, you'll discover that there is a special field, `%_playlist_number%`, which will return the line

number of the current playlist occupied by each track listed in it. So, thinking this through from first principles, you can imagine a function that says, in pseudo-code:

```
if mod(%_playlist_number%,2) = 0 then
    background-colour = (x,y,z)
else
    background-colour = (a,b,c)
end;
```

Well, that is precisely the logic of this statement, written in proper 'Foobar2000-speak':

```
$if($strcmp($mod(%_playlist_number%,2),0),$rgb(x,y,z),$rgb(a,b,c))
```

And so we substitute my earlier choice for the background colour in the first of those RGB values, and pick a rather lighter version of that colour for the second:

```
$if($strcmp($mod(%_playlist_number%,2),0),$rgb(112,161,211),$rgb(207,227,247))
```

That gives me two possible values for a background colour, so I now need to embed that entire string into the appropriate place within the colour formatting string I devised earlier:

```
$rgb(0,0,0)|$rgb(133,156,251)|$if($strcmp($mod(%_playlist_number%,2),0),$rgb(112,161,211),$rgb(207,227,247))|rgb(192,216,240)|$rgb(0,0,0)
```

It wraps horribly, but I hope you can nevertheless see that this is merely the earlier string of five colours with an even-or-odd test wrapped up inside it for one of the five colours. Pasting that into the same place as I did before, I get this result:

Composer	T	Title	Album	Artist	Length
Johann Sebastian Bach	2	Da jesus diese Rede vollendet hatte	Matthäus-Passion	Karl Munchinger, Stuttgar...	2:10
Johann Sebastian Bach	3	Da versammelten sich die Hohenpri...	Matthäus-Passion	Karl Munchinger, Stuttgar...	0:45
Johann Sebastian Bach	4	Da nun Jesus war zu Bethanien	Matthäus-Passion	Karl Munchinger, Stuttgar...	1:13
Johann Sebastian Bach	5	Da das Jesus merkte	Matthäus-Passion	Karl Munchinger, Stuttgar...	2:06
Johann Sebastian Bach	6	Du lieber Heiland du	Matthäus-Passion	Karl Munchinger, Stuttgar...	1:06
Johann Sebastian Bach	7	Buss' und Reu'	Matthäus-Passion	Karl Munchinger, Stuttgar...	5:15

...and you can see here the alternating 'striping' effect that makes it a bit easier to match up the right length time with the right track title. The dark black bar is the one that is currently highlighted.

Again, you might well look at that screenshot and go "Ugh!", and that is your right and privilege! I'm not making claims for great colour coordination efforts here, and I have no doubt that you can cook up better colour schemes for yourself. It is all a matter of just experimenting with appropriate combinations of colours. But the real point is that you at least know how to start experimenting for yourself now. The quality of the results is merely a matter between you, your conscience and your innate good taste!

10.2 Suppressing Duplicates

The other thing people seem to want to do quite a lot is to switch off the repetitious display of the artist or album name within a playlist. This can be done, but it's not the easiest thing in the world to achieve!

Again, it helps to start with a basic example, and establish the basic logic, which is really fairly straightforward, once you think about it (though it helps to be coolly rational and logical when you do. That's why we have developers and programmers, of course!)

Basically, you want Foobar2000 to walk through a playlist, and ask itself, 'Are you track 1, because if you are, I'll display a divider in your Album name. If you are track 2, I'll display the

artist name as normal. If you're track 3, I'll display the album name, and if you're anything other than track 3, I'll display a blank'.

Putting that into pseudo-code, we have:

```
if %tracknumber%=1 then
    set %album%="-----"
else
if %tracknumber%=2 then
    set %album%=%artist%
else
if %tracknumber%=3 then
    %album%
else
    set %album%=''
end;
```

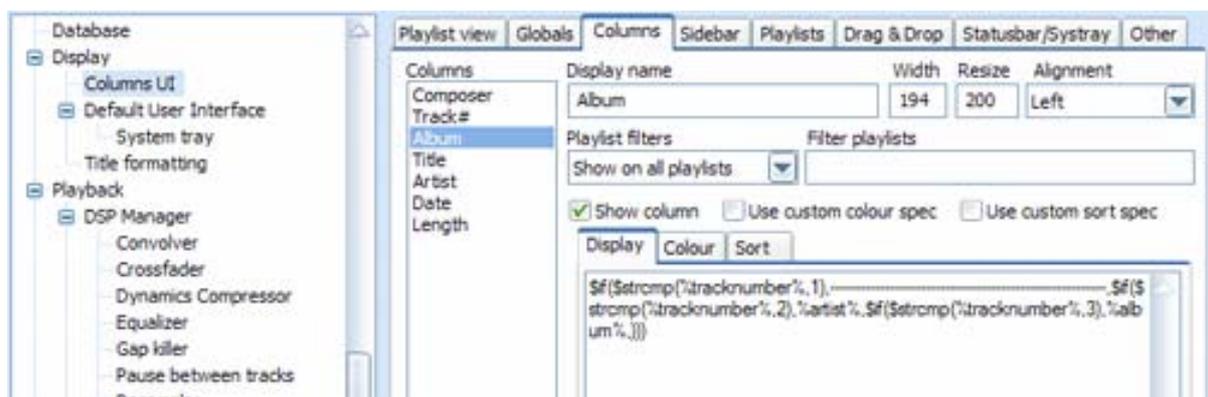
Now you already know how to do IF statements using Foobar2000 formatting syntax:

```
$if(something, do-this, else-do-this)
```

And we have seen previously one example of where we can test whether a tag has a specific value: we use the **\$strcmp** function ("string comparison"). So the 'Foobar2000-speak' way of writing this would be something like this:

```
$if($strcmp(%tracknumber%,1),-----
,$if($strcmp(%tracknumber%,2),%artist%,$if($strcmp(%tracknumber%,3),%album%
,)))
```

Once again, it's a nightmare to read, because of the nested functions, but if you can follow the basic logic, then all we have here is three nested if-then-else statements, and the thing doesn't become quite so hard to read after all. To take effect, this code needs to be pasted as the display value for the album column in the playlist. Do that by right-clicking the Album column header, and then selecting the *Preferences* menu option:



You should already be opened at the Columns tab, and you simply paste that code into the Display sub-tab, as you can see me doing here. Once you click the [Apply] button at the bottom of the screen, you'll see this sort of thing:

Composer	T	Album	Title
Johann Sebastian Bach	1	_____	Kommt,ihr Toc
Johann Sebastian Bach	2	Johann Sebastian Bach	Da Jesus diese
Johann Sebastian Bach	3	Matthäus-Passion	Da versammel
Johann Sebastian Bach	4		Da nun Jesus v
Johann Sebastian Bach	5		Da das Jesus r
Johann Sebastian Bach	6		Du lieber Heila

And, as you can see, whilst the results are certainly not polished, the code is at least doing its work correctly. In point of fact, I struck lucky with that screenshot, because different people tag track numbers differently, and sometimes my track numbers are not '1', '2' and '3', but '01', '02' and '03', meaning that the logic breaks down and nothing in my formatting string ever evaluates to true. To get round that, I modified the string a little:

```

$if($strcmp($num(%tracknumber%,2),01),-----
-----
, $if($strcmp($num(%tracknumber%,2),02),%artist%, $if($strcmp($num(%tracknumb
er%,2),03),%album%,)))

```

By nesting a `$num` function in there, I force all track numbers to be represented consistently as two-character fields (I know my music, and know ahead of time that I have no albums with more than 99 tracks on them, so I didn't need to force the strings to be three characters in length!) Naturally, the string can get an awful lot more complicated than this, too... but I hope you see that no matter what functions might get nested in here to make its operation consistent and desirable, the basic logic is unaffected.

As one more example: it's a bit messy spelling out all those hyphen characters to display if the track is track 1. Instead, why not re-write the function like this:

```

$if($strcmp($num(%tracknumber%,2),01), $repeat(-,35), $if($strcmp($num(%track
number%,2),02),%artist%, $if($strcmp($num(%tracknumber%,2),03),%album%,)))

```

The `$repeat` function nested in there now will return the first character listed in the brackets the number of times specified as the second element within the brackets. It makes it a bit tidier, but again it doesn't really alter the fundamentals of what this code is trying to do.

Some of you might already have spotted an obvious problem looming with my logic, though: what if the guy has albums with less than 4 tracks? Well, of course, the logic falls flat on its face and important bits of information would simply never get displayed. CD singles would be a particular nightmare, for example, being displayed as nothing more than a sequence of anonymous dividers! To counter this, I could write a new tag physically into such tracks that indicates whether the track is a single or not. Then I could wrap the entire existing logic within yet another `$if` test: if the track identifies itself as a single (or, in general, as being part of an album with less than X tracks), then don't do anything clever at all, but just display the proper tag names.

But that is a mere detail. If you've understood what I've done so far, then you are well enough versed in the mysteries of formatting strings to start experimenting for refinements like these of your own, and you will probably swiftly discover better and more efficient ways of implementing even the logic that I have outlined here. I certainly hope so, at any rate, because I don't claim any expertise when it comes to writing this sort of thing, and I know ahead of time that others can write it better.

Then, too, consider that most people won't be faffing around writing their own formatting strings in any case, but will instead simply get their media player working by borrowing `.fcs`

files prepared by others, in which the formatting strings are already beautifully and perfectly formed... but at least if you join them in doing that, I hope now you'll be able to recognise some of what those files contain, and understand what they are doing to achieve their effects.

11.0 Final Refinements

Well, this article could run and run, and we'd still only be scratching the surface of what's possible with Foobar2000. So I'd like to wrap up just by touching on a few other customisations I like to make to my own Foobar2000 configuration, and in the process hopefully show you that there's lots more to explore.

11.1 Additional Components

Hunting through 12,000 CD tracks for the particular one you want is not fun. So I like to install the **foo_playlistfind.dll**¹⁴ component. If you display your entire music collection in a playlist (which you can do by creating a new playlist, and then dragging the 'All music' item from the Album list panel to it), then you can start typing what you remember of a track or album's title, and the component will jump to the first item that matches. Type some more, and it narrows down the number of matches. Configure it in the usual *Foobar2000* -> *Preferences* panel, and with a simple down- or up-arrow and you can find the next or previous matches to a given search. With the number of tracks I have, it's pretty essential.

Slightly less essential for me, but useful in case I get lost wondering which track it is that is actually playing at any given time, the **foo_uie_trackinfo**¹⁵ displays as much information about the currently-playing track as can be determined from the track's tags. The only thing limiting the amount of information displayed is how good you are at working with Foobar 2000's formatting language –which, by now, should hopefully not be a problem!

11.2 Custom Sorts

You might have noticed in section 5, when I was talking about how to tag and rip your music tracks, I mentioned that when I have a multi-CD boxed set, I use the DISCNUMBER tag to distinguish between track 1 on CD1 and track 1 on CD2. (Incidentally, some people prefer to use just the DISC tag, but I like to be consistent: if it's *tracknumber*, then it ought to be *discnumber*. But you don't have to agree with that!)

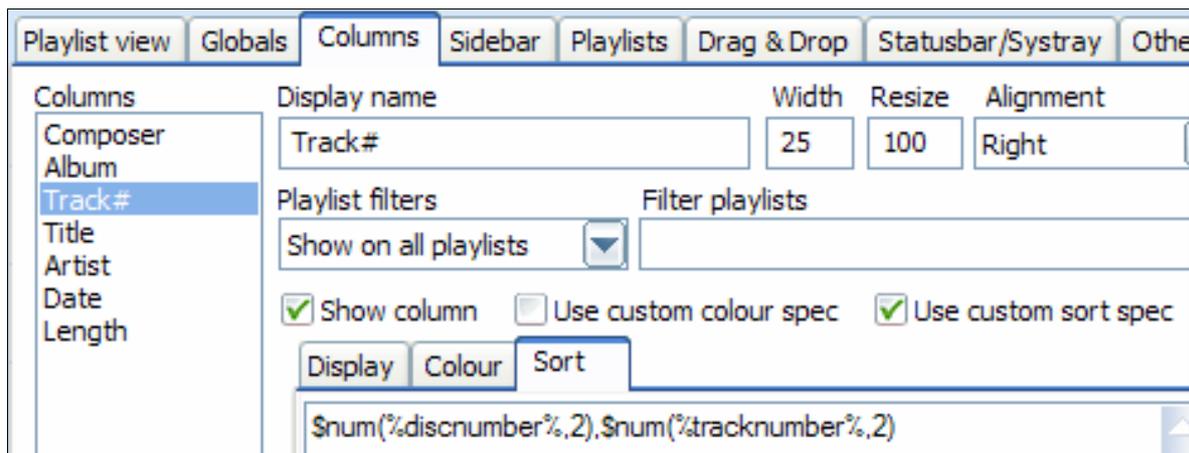
It follows that when a multi-CD album is displayed in a playlist, I want all the disc 1 tracks listed before all the disk 2 ones –but, within each disc, the ordering should be by track number.

Now it's easy enough to sort a playlist: you just click on a column heading once to sort it ascending by that column, and again to sort it descending. So I could just edit the playlist so that it displays the disc number column (see section 4), and then sort by that. That would certainly get things sorted by disc number... but within a given disc number, all bets are off. And if I click on the tracknumber column to establish *that* as the sort order, then it's goodbye sorting by disc first!

What I need to fix this, in fact, is a custom sort order for the track number column. I can create one of those by using the *Foobar2000* -> *Preferences* -> *Display* -> *Columns UI* options, and then calling up the **Columns** tab:

¹⁴ http://yallara.cs.rmit.edu.au/~kspear/foobar/foo_playlistfind-0.8.7.zip

¹⁵ http://yallara.cs.rmit.edu.au/~kspear/foobar/releases/foo_uie_trackinfo.dll



What this screen says, in effect, is: if the user sorts by the 'Track#' playlist column, actually sort by the results of the custom sort specification. And, as you should be getting used to by now, such a custom sort specification is written using the sort of configuration commands, functions and syntax we've seen many times before. In this case, my sort specification reads:

```
$num(%discnumber%,2), $num(%tracknumber%,2)
```

...which means: take the contents of the discnumber tag, and convert it into being a 2-digit number. Also take the tracknumber tag, and convert *that* into a 2-digit number as well. And then concatenate the two together to produce a single 4 digit number by which you can then sort the tracks.

So, I might have disc 1, track 16 and disc 2, track 1... but the formula converts those into 0116 and 0201, and sorts accordingly. That's a single column sort, therefore (I only have to click the TRACK# column to get it performed), but it results in a primary-secondary sort on more than one column.

Another example of breathtakingly-good functionality. I know of no other media player that can be persuaded to do this sort of thing –and, given the way I tag my music, it's a must for me!

11.3 Switching off Playlist Column Headings

This is a very minor tweak, but I don't actually like seeing the headings for my playlist columns (I can still sort tracks without them, by using the *Playlist -> Sort* menu options). You can switch them off easily enough by using the *Foobar2000 -> Preferences -> Display -> Columns UI* options, and then calling up the **Playlist view** tab. At the top left of the screen-full of options, you should find a check-box called **Show columns header**: uncheck that, and the column headings are gone!

Given that column headings are needed to resize columns using the good, old-fashioned click-and-drag-till-it-looks-ok method, you might want to make sure all your columns are the right size and position before switching them off! You can, of course, alter sizes and positions of columns in the Columns UI options screen, as we saw back in section 4.

11.4 Auto-resizing Columns

Another minor issue is that if I resize the Foobar2000 main window, I don't like to see all my playlist columns slowly disappear off the right-hand side. That's what happens when the column widths are specified 'absolutely', which is the default. But in the Columns UI preferences screen,

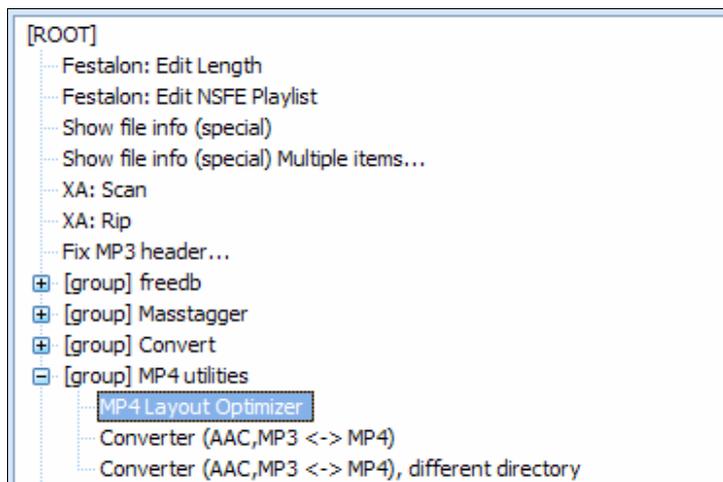
under the **Playlist view** once more, you can enable a check-box labelled *Auto-resizing columns mode*, and if you do so, columns shrink and expand as necessary when the main program window is re-sized. The results can be unpredictable at times, so you might well need to switch back and forth between 'absolute' and 'auto' resizing modes from time to time, but it's still a good option to have.

11.5 Custom Menus

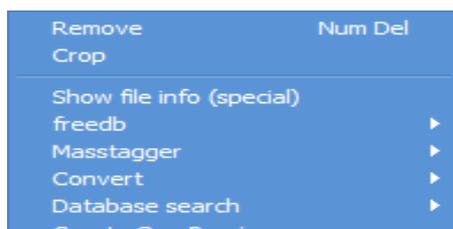
This one's quite tricky, and can wreck a Foobar2000 configuration if you're not careful, so backup before you start!

But the default ordering of options in pop-up menus can sometimes be a bit awkward. For example, when you're tagging and ripping music, you'll make use of a right-click pop-up menu (known in the trade as a *context-sensitive menu*) which has the options **freedb**, **Masstagger**, and **convert**. And those three options will appear at the top, bottom and middle of the context-sensitive menu! Since you use all three options to achieve the one task, why can't all three appear next to each other?

Well, they can. In *Foobar2000* -> *Preferences* -> *Core*, you'll find a configuration item called **Context menu items**. The main panel will then display the structure of context menus, and you can click-and-drag any of them to re-position them. You can also right-click any menu item and move it up or down by selecting from the resulting pop-up options, which is a rather more controlled way of achieving the same thing! Either way, you might end up with something like this:



Here, I've dragged the freedb, Masstagger and Convert menu *groups* so that they sit together at the top of the list of options. (They're groups because they contain sub-menu items. They can be re-positioned too, if you like). When I right-click on a track in a playlist now, I therefore see this:



...and you'll notice how the three items appear nicely together, and at the top of the menu where I can most easily get to them. That's a real productivity bonus for me!

Of course, by right-clicking on a Context menu item, you can also choose to rename a menu item, or delete it altogether, and you can also add in completely new menu options, if the mood takes you. I tend not to go that route myself, but the option's there if you need it.

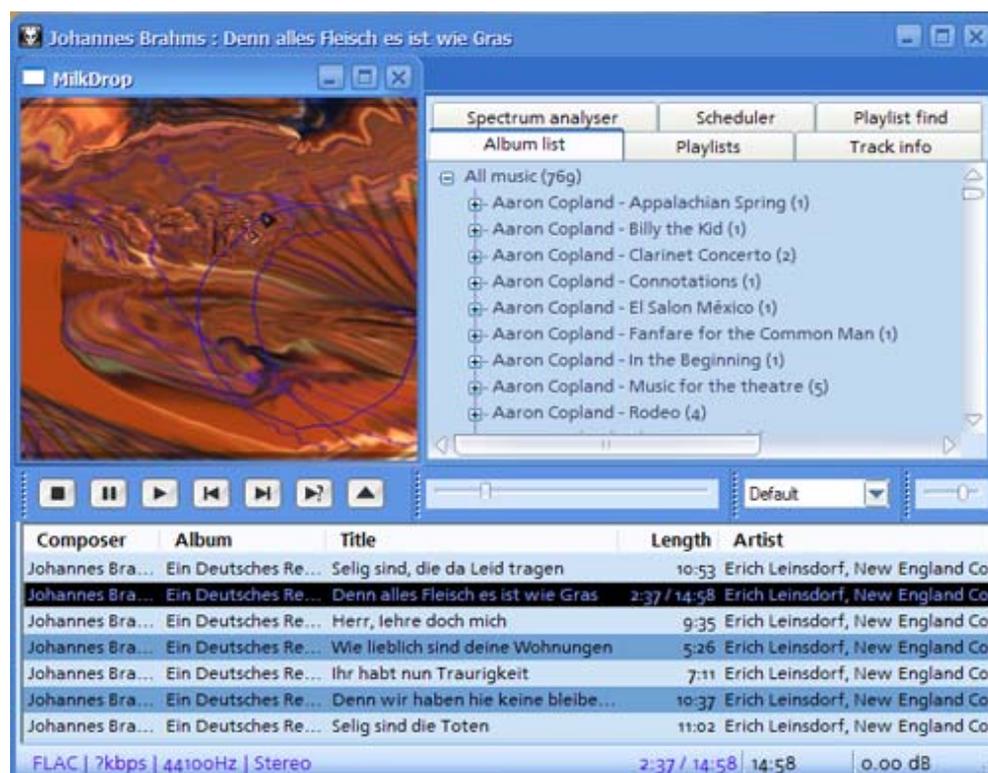
You can also do the same sort of re-construction job on the main program menus (the ones that drop down from the main toolbar's *Foobar2000*, *Playback*, *Playlist* and *Components* menus.

Name me another media player where the very structures and layouts of the menus you use to accomplish anything productive are capable of so much user-configurable control!

11.6 Visualisations

As you read this article, you might have noticed that my music tends to be mostly of the classical variety. So you might imagine that I don't have a lot of time for flashy visualisations when playing my music... and you'd be absolutely right. In fact, one of the main reasons I first adopted Foobar2000 as my media player was that it *didn't* have all the flashy (and hugely distracting, from my point of view) visualisation bells and whistles that (say) Winamp or Windows Media Player ship with.

Instead, Foobar2000 ships with a (frankly) rather inadequate spectrum analyser, and that's about it. Suits me, as I say. But if you are desperate for some decent visualisations, then here's proof you can have them:



That is Foobar2000 running a Winamp visualization plug-in called **Milkdrop**, and it can do this because I downloaded and installed the **foo_vis_bacon** component.¹⁶ The website has excellent installation instructions: better than any I could write here, so follow them closely, and you too can expend CPU cycles whilst *not* listening (closely) to your music!

¹⁶ From <http://www.msu.edu/~singergr/bacon.html>

12.0 Now go back and do it all over again!

Way back when this article started, I encouraged you to download the 'Special' edition of Foobar2000. And when you installed it, I suggested you should select to install all the optional components (see page 4, if you've forgotten that far back). The reason giving you those two pieces of advice was simple: it meant that I would know pretty much for certain what you had installed and what its capabilities were, which made it easy for me, throughout the rest of the article, to know what would work without an additional download or install, and what wouldn't.

But installing all those extra components does mean Foobar2000 runs 'fatter' than it can, consuming more memory and other system resources for pieces of functionality that you might never use (though I myself have no problem with Foobar2000's fairly modest demands, even at its fattest!)

So, now you've got this far in the article, and can consider yourself something of a Foobar2000 'old hand', can I suggest that you take a new .fcs backup (so your Columns UI configuration is safe for future use), uninstall Foobar2000 completely... and then re-install it with only the minimal or default components installed? As you re-configure your new Foobar2000 installation, you can add in just those bits of functionality that you really need. You can always re-import your saved .fcs settings to get the look and feel right, but you'll end up with a leaner, meaner media player than before.

13.0 Conclusion

There's lots more we could cover here, but at 54+ pages, this is already not so much an article as a thumping great tome. My aim was to give you the basics necessary to get Foobar2000 looking and working more-or-less the way you want it in a matter of a few hours, and if I keep on going, it'll be a matter of a few days or more!

So, for the sake of sticking to the original deal, that's it for now!

I hope you find Foobar2000 as much fun, and as much a pleasure, to use as I do. The fun is in the configuration: from having a program behave the way you want it to behave, and not the other way around. That's such a refreshing change from most software these days! But the pleasure is in knowing that you can take configuration as much or as little as you want: if you choose to stick with the default look and feel, it's a damn fine music player anyway.

Well, to sum it all up: If this little guide has helped you discover one smidgen of that fun and pleasure, I've done my job.

13.1 Acknowledgements

I couldn't have written this article without the help of many of the posters who inhabit the official Foobar2000 forum¹⁷. And it wouldn't have been worth writing had countless developers spent time developing the many components which make Foobar2000 such a highly functional and configurable media player. But, of course, my cheeriest acknowledgements must go to Peter Pawlowski, who wrote the thing in the first place.

¹⁷ <http://www.hydrogenaudio.org/forums/index.php?showforum=28>

14.0 Legal Matters

14.1 Copyright

This article is copyright © Howard J. Rogers, 2005. It is licensed under a *Create Commons Attribution-NonCommercial-ShareAlike 2.1 Australia* license, meaning that you are free to copy, distribute and display the article and to make derivative works, but if you do so, you must include full attribution for the original work to me, Howard J. Rogers. You must not use this article for commercial purposes. And if you alter, transform or build upon this article, you may only distribute the resulting work under a license identical to this one. The full conditions of the license may be viewed at <http://creativecommons.org/licenses/by-nc-sa/2.1/au/legalcode>.

Application for permission to waive any of these conditions can be made to me, Howard J. Rogers, at hjr@dizwell.com.

14.2 Errata and disclaimers

This article may contain errors of fact or interpretation. Notification of such error should be made to me, Howard J. Rogers, at hjr@dizwell.com. Any and all responsibility for loss, damage or destruction of data or any other property which may arise from relying on the information contained in this article is disclaimed. I will in no case be liable for any monetary damages arising from such loss, damage or destruction.