

Managing Your iPod With Smartlists

April 2007 Edition



Charles Hannum Jr
2004-2007

PART I: INTRODUCTION

What this document does:

This document describes a system of linking smartlists that allow you to create self-updating music mixes according to weights and rules you decide. The particular examples used are derived from my own creations, but, if you understand what I've done, you can easily tailor things to suit your tastes.

Why you might care:

iTunes' smartlist system is what truly separates the iPod from the rest of the market. Forget the style, forget the simplicity of use (which largely arises from the iTunes software anyhow) and think about taking greater control of what you listen to in a way that takes very little effort on your part. Once you understand how to link smartlists to create automatic music mixes, you can craft self-generating playlists that turn your iPod into a series of cyber DJs that play music tailored to you without very little to no further effort on your part.

In addition to the "cyber DJ" aspect of smartlists, they can also serve an important function for those who have more music than fits on their iPod. If this describes you, then you know you will need some sort of system to rotate music on and off your iPod. Sure, you could just manually manage everything, but wouldn't it be nicer if some, most, or even all of this rotation was automated?

Damn, this is a lot of reading; why not just use the "Shuffle Songs" feature of the iPod?

The "Shuffle Songs" feature, though idiot proof, is relatively weak. Although there have been some improvements, notably the ability to set a "Skip while shuffling" flag for files you don't want shuffled, there are still shortcomings to the "Shuffle Songs" feature that aren't eliminated simply because you can now tell the iPod not to shuffle your mp3 audiobooks.

1. Shuffling is blind, deaf, and dumb. When you shuffle songs, the iPod simply goes through all of the audio files eligible for shuffling in a more or less random manner. Some filetypes are automatically excluded from shuffling, but these are hardcoded by Apple and barely useful. All other exclusions from shuffling must be manually flagged in iTunes. This binary yes/no to shuffling is the beginning and end of your primary control of the music mix. If you want to hear more Eminem than Mozart, too bad, what gets shuffled is at the mercy of Apple's shuffling algorithm.

Now, you can take control what gets shuffled by carefully limiting what's available in the first place by either only shuffling within a user defined playlist or carefully controlling the ratio of what gets put on your iPod (e.g. 20% Jazz, 40% Classic Rock, 10% 1980s Pop, 30% Hip Hop & Rap). Those aren't terrible solutions, but they are laborious to set up and maintain. The beauty of this document is it explains how to automate such a process and polish it far beyond anything you can do manually.

2. *Shuffling is cognitively impaired.* In spite of whatever Apple wants to claim to the contrary, shuffling is not purely random and has some sort of weighting in the algorithms for how it chooses music. This would be fine except that it lacks a persistent memory state: it resets its shuffle state back to zero after every reset or sync. The result is that you hear some songs repeatedly while never hearing others. You can wind up hearing mostly the same 400 songs in slightly different arrangements again and again while never hearing the bulk of the other 4000 songs on the iPod.

A minor rant before I get to the good stuff:

For some reason, there are people out there paying good money so they can downgrade their iPod with software like Anapod. These people are sold on the idea that it's better to manually manage your music via dragging and dropping versus the allegedly inefficient way iTunes "forces" you to manage your music. Worse, they've fallen for the scam of paying someone else to do something iTunes already does for free.

There isn't much efficient or desirable about drag and drop management, particularly if you have a large music collection and a smaller iPod. Show me someone who thinks it's fun to manually select a different 3.5 GB of music every week for their mini or nano and I'll show you someone with a ball-gag in the nightstand drawer.

Concept: Dragging and dropping is only desired because you've become accustomed to using DAPs with garbage software that forced you to adapt to the lowest level of management. In a similar vein, file-tree browsing is not a feature, it's a kludge from designers too lazy to write good software. Apple could certainly improve upon the browsing options with the iPod, but that's hardly an argument to go all the way back to the cave.

There is very little iTunes forces you to do in regards to managing your music beyond allowing iTunes to create an index of it. A nice feature of the iPod/iTunes marriage is that Apple caters to all three types of control schemes. The total control freak that only wants to move files on and off of the iPod manually can do that - and drag and drop is fully supported from within iTunes if that's your thing. At the same time, iTunes fully supports the "don't think, plug it in and go" crowd. Last, iTunes caters to those in the middle, that's where this document comes in.

The simplest explanation of what you are going to do is set your iPod to sync to a certain set of playlists, some smart, some regular, and then all you will have to do is plug the iPod in, update, and, voila, music will be moved on and off of your iPod according to the rules you use.

Admittedly, you do have to use iTunes to accomplish anything really useful with the iPod, but if that's your hang up, stop reading now. I have tried many music organising programs out there, and while they range from the awful to pretty good, not a one of them touches the functionality of the iTunes/iPod combo. The very reason I bought an iPod and continue to buy iPods is iTunes. Yes, it sucks for CD ripping, volume normalisation, and is even a bit clunky for music playback, but for music and player organisation, it can't be touched in my opinion.

PART II: MANAGING AN IPOD USING THE SMARTLIST FUNCTIONS OF ITUNES

Configuring the iPod:

In order to manage an iPod with playlists you will need to set the options on the Music tab for the iPod to sync to selected playlists. The reason for this is two-fold. One, Apple provides no way to sync ratings and play data from iPod to iTunes other than auto-syncing (bad design) and ratings and play data are an integral part of smartlist management. Two, the idea is to get iTunes to manage at least some of your music without your thought (good design).

Some playlist basics:

There are two types of playlists, regular and smart. A regular playlist is just a list of songs that you manage directly by dragging or copying genres, artists, albums, or songs over to the playlist. These are the lists with the sheet and musical note icons in the left hand panel of iTunes.

A smart playlist (smartlist for short) is a rules-based list that is dynamically generated by iTunes based upon rules that you set when the list is created or later edited. These are the lists with the blue sheet and asterisk icon in the left hand panel of iTunes.

There are two uses for regular playlists in managing your iPod. The first is as a list of songs you directly manage that you want on your iPod. This can be a special playlist, e.g. “Jane’s Birthday Party”, or it can be an album that you want intact on your iPod. You can use as many or as few lists as you want for this aspect. The downside to extensive use of regular playlists is that they are negating the automation part that makes iTunes so nice for the iPod. The second use is as a list of files that a smartlist will pull from (more on this later).

Smart playlists, however, are where it’s at – with a push of the ‘+’ button you add a new rule. With the many drop down options you can filter for anything that iTunes keeps track of: playcount, rating, artist, genre, composer, date added, when it was last played, bitrate, etc. You create the conditions and iTunes instantly finds every single song in your library that matches them. Further, by chaining smart playlists, you can perform some fairly sophisticated selection of music.

Tip: If you shift-click the ‘+’ button (option-click on a Mac) in the editing window for a smartlist, the rule category of the line you clicked, as opposed to the default of ‘Artist’, will automatically be selected for the new rule.

One more very important thing about playlists: the order a playlist is in at the time you sync it to the iPod is the order that songs will appear in your iPod. If you go to all the trouble to create a mix but have it sorted by artist at the time of syncing, oops, it’s all in artist order. You have to take care to make sure that playlists are sorted or arranged exactly how you want them to be before any syncing operations, e.g. for an auto-generated mix, you should have it sorted according to the list track number, the left most column.

Smartlist Updating – Quirks and unresolved issues

When you create a smartlist, one of the options you can set is whether or not to enable “Live Updating”. A smartlist for which this option is turned off only fills itself up once and then behaves like a static normal playlist until you activate the “live updating” checkbox again (at which point, you can uncheck it again to “freeze” the song selection). Live updating, though, makes the list automated and dynamic. A smartlist with live updating enabled *should* automatically add or drop tracks from that smartlist whenever something it keeps track of changes. As an example, if you have a rule that says that all songs that are rated higher than 3 stars should be on a particular list and you rate a song 4 stars, it should be added to that list. If you later decide that song is really only worth 3 stars, it should then be removed from that list. Pretty simple conceptually but not quite so simple when it comes to iTunes or the iPod; here are the major quirks involving live updating that you need to know to predict how your smartlists will work:

1. In iTunes, updating of smartlists is immediate. If you’re listening to a list that only includes those songs higher than 3 stars in rating and you lower a song’s rating to 3 stars it will immediately stop playing and be removed from the list.

On the iPod, however, while the smartlist is modified in the background, an actively playing list won’t change so long as you are in that playlist. Once you back out of the list and go back in, though, the changes will have taken effect. This, to me, is the preferable behavior because you can change the rating on a song without everything immediately coming to a halt because the state of your smartlists has been changed.

2. A smartlist will not update on the iPod unless all dependent playlists are present. For example, if you have a smartlist, [SMARTLIST](#), of songs taken from a second smartlist, [SECOND SMARTLIST](#), but only [SMARTLIST](#) is synced to your iPod, it won’t live update on the iPod. This example is simple and obvious, but it becomes troublesome with the sort of chained systems this document is built around because if any smartlist in a chain isn’t sync’d to the iPod, none of them will live update. Basically, there are three options:

A. Sync ALL your playlists involved in your selection system to your iPod. This can be a bit unwieldly and has the secondary requirement that the total amount of music defined by the smartlists doesn’t exceed the capacity of your iPod. Depending on what you’re managing in iTunes and the particular smartlist system you are using, this may be impossible. For example, if you’re using an exclusion list as a key part of your system, it may be nearly as large or larger than your iPod, and that’s the stuff you don’t want on the iPod! There are also some major issues related to the processor capabilities of the iPod with attempting this that I’ll cover in more detail later.

B. “Cheat” by manually copying the contents of some intermediate smartlist(s) onto normal list(s) to get around this behavior. This is how I had to manage music on a 60GB iPod; I used a system of smartlists in iTunes to generate a music pool and periodically copied that onto a normal music pool list that was sync’d to the iPod. In turn, the on-iPod system of smartlists that resulted in various mix lists all referenced this normal music

pool list at the base of their chains. While not fully automated, it only involves the clearing and copying of contents between two lists periodically to achieve great results.

C. Just ignore live updating for on-the-go use of the iPod and rely on iTunes syncs for updating the smartlists on your iPod. If you design your system with this limitation in mind, it will be of little, if any, handicap. The first example later in this document for a playlist system uses this approach.

3. Not all of the things tracked by iTunes are supported on the iPod. If you attempt to use a smartlist on your iPod that incorporates a rule involving one of these elements, live updating for that list (as well as any lists that depend on it) will not work. Worse, iTunes usually won't inform you with the exclamation point icon like it does for most smartlist breaking errors. Some examples of known smartlist breaking elements are:

A. Referencing a folder. In iTunes, you can organise playlists using folders in the source pane and, even more conveniently, you can actually refer to an entire folder (even one containing nested folders) as if it were a giant playlist when using the "Playlist is/is not" rule. Unfortunately, folders are not supported on the iPod and any playlists that refer to a folder as a playlist will not live update on the iPod. The solution is to not reference folders as playlists.

B. Podcast is True/False. There is a smartlist rule just for whether an item is a Podcast. Unfortunately, it doesn't work right and any playlists that involve using this rule will not live update on the iPod. The solution is instead to use the {Genre IS/IS NOT Podcast} rule to include or exclude podcasts. All podcasts downloaded by iTunes will have this as their genre automatically anyhow so the Podcast flag rule is superfluous in addition to being poorly implemented.

C. Video Kind is XXX. Like the Podcast flag, this is another half-baked implementation that does not work right. You'll want to use the general {Kind IS XXX} or {Genre IS XXX} rules for including or excluding video content on the iPod with smartlists.

D. Last Skipped rule on older iPods. iTunes recently began tracking if a track was skipped in the first 20 seconds. This can be a handy way of getting a track to drop off of playlist temporarily if you're not in the mood for it just now but might be tomorrow. Unfortunately, being a recent addition to iTunes, it's an incompatible rule with older iPods (e.g. I know it won't work with the 1G iPod mini, which means it probably doesn't work with anything before the 5G and nanos).. The solution here is to not to use this condition unless you know your iPod supports it.

There are probably more conditions that can break live updating on the iPod, but these are four that I know about. If a smartlist *should* live update on the iPod, i.e. you know it doesn't reference any playlists not on the iPod, live updating is checked, and it still fails to live update, it's probably because of an iPod incompatible rule. You will have to engage in a little trouble shooting to figure out the source of your problem.

A brief warning:

Once upon a time, I nearly finished a computer science degree before leaving to go back to biology. There are some concepts that are obvious to me that may or may not be obvious to you. Two major ones that come into play for this document: Functions and Boolean Logic.

When you write code, you break it out into functions. Each function may be thought of as a chapter in an instruction manual or a tool that does a particular thing. You then have a main function – named `main()` of all creative things – and it pulls all the other minor functions together and organises the way they work together. Conceptually, this is how I approach smartlists. It should be clear enough, but if you find yourself scratching your head at how I approach things, this is why.

Boolean logic is simpler. It's the rules that define how a set of logical statements linked by conditional statements (and, or, equal, not equal, etc.) are evaluated. Fortunately, the smartlist system is simplistic: Every smartlist has a setting, "Match All" or "Match Any" at the top of it. "All" means to think of each rule in that smartlist as linked by a Boolean AND conditional and "Any" means to think of each rule in that smartlist as linked by a Boolean OR conditional. The simplest explanation is that if you have a smartlist set to "Match All" then every rule you define must be true for every audio track or it won't be pulled in by that list. If you have a smartlist set to "Match Any" then, logically enough, if any one of the rules is true for any track, it will be pulled in by the smartlist. You can read more on Boolean logic here:
http://en.wikipedia.org/wiki/Boolean_logic

Concept: Although each individual smartlist is either a series of all AND or all OR statements due to the MATCH ALL or MATCH ANY setting global to each smartlist (with each rule line being one statement), you can also think of each playlist as a set of Booleans enclosed by parentheses. As such, in spite of some people's erroneous belief, there is no practical limit to the complexity of filtering possible through linking playlists, just limitations from what conditionals you can use in the first place (and the limitations of your patience).

PART III. EXAMPLES OF PLAYLIST SYSTEMS

This document will describe and discuss three systems of using smartlists for organising music on your iPod:

1. The first system does not take advantage of live updating on the iPod and relies upon syncing with iTunes for all updates of the playlist contents. Conceptually, it is the easiest to understand so we'll start there.
2. The second system is more elegant and takes advantage of live updating on the iPod for generating dynamic playlists. However, it requires an intermediate normal playlist as a source. Its advantage is that it gets around the problems of having to sync every playlist involved in a smartlist chain to the iPod.
3. The third system is, I believe, the most elegant. Everything is automated with smartlists. The disadvantage is that this only works with a relatively small amount of music on the iPod, so it's best suited for small capacity iPods like the nano.

System One: The Beginner

This system is a very simple one that could be used to either manage a small capacity iPod or to supplement whatever management scheme you have for a larger capacity iPod (I used a nearly identical system once for an iPod mini). The basic idea is to generate some number of pre-shuffled music mixes. In this particular example, we'll set the length arbitrarily at 50 songs.

Concept: iTunes is not truly random. If you've used the "Shuffle Songs" or any other randomizing feature you probably already figured that out. There are certainly random elements to iTunes, but there is definite weighting going on in the choices due to whatever is the underlying algorithm. The upside is that it's not bad at simulating the job a human does when going through a list of music and picking and choosing – there is a certain amount of seemingly logical clumping and concentration of artists within any particular generated list. The downside is that, without your help, the shuffling and random selection features of iTunes will do little more than shuffle through the same music you've already heard while managing to ignore most of your library. Fortunately, like most Apple products, although out of the box it's geared more towards trained chimps, power users can use iTunes if they'll put a little thought into things.

These music mixes are built around the following categories:

1. Highly rated music – who doesn't want to hear their favorite music?
2. A random selection from the whole library – this gets things mixed up.
3. Music not played in a while – makes sure that everything is rotated.
4. Music you're in the mood for now.

Parts 1, 2 & 3 are the general mix, part 4 is more specialised; it's that new album that just came out, or that artist you have a hankering to immerse yourself in. I'll talk about how to generate the general part of the mix first.

First thing you need to do is define what can be shuffled and what can't. We're going to create a smart playlist called **EXCLUSION**. Set it to MATCH ANY and set conditions for anything you

don't want shuffled. For example, {Genre IS Audiobook} and {Genre IS Comedy} are some of the rules I used in my own system. If you use ratings (this example does), also set it to match any ratings you don't want shuffled (for me, that's the 1 & 2 star rated songs). The end result is a list of all the files you don't want to be shuffled by iTunes. The way you use this list is that in the future whenever you generate another smart playlist you can make {Playlist IS NOT **EXCLUSION**} one of the mandatory requirements and it will never pull in anything you don't want. Once you've done that, you're ready to move on.

Concept: One thing mentioned above and that you need to keep in mind is that iTunes is positively awful at hitting all your music without you forcing it to do so. If you do not create a condition that forces it to play all your music then the sun might very well be cold by the time iTunes does so left to its own devices. Some people choose to take the simplest route of forcing full rotation by always having playlists select according to least recently played. However, that's a very artificial way of listening to your music as you only hear things once per library rotation, bleh. There are ways to force this full library rotation without resorting to pure brute force, and I'm going to tell you about them.

Concept: Before I get to that, the key related element is that we need some way to get these auto-generated lists to cycle music at all. When you mess around with the smartlists, you will see that there are a huge variety of temporal and quantitative methods to limit music. You can have iTunes only select those songs played less than a certain amount of times, not played in the last 7 days (or 7 weeks), you can have iTunes only select music added in the past month, and so on. These are the keys to shuffling music through these playlists. If you have a smartlist with the limitation that music must not have been played in the past week and you do play it, poof, it drops off the list and something else is pulled in to replace it the next time you sync.

Addendum: When using a non live updating system like this, if you use a "not played in the last..." time window to cycle a particular smartlist, the time window chosen to drop music from a playlist must be longer than whatever your typical iTunes sync/refresh interval. If you set a playlist to shuffle out songs played in the past week but only sync/refresh every two weeks then nothing will be rotated on and off of your iPod.

Final Note: This system (and the other examples in this document) make extensive use of ratings for weighting music mixes. My particular rating scheme (and the one used in this document) is:

- 1 star = garbage, don't care if I ever hear it again.
- 2 star = doesn't work as a shuffled song but fine if listening to the whole album.
- 3 star = meat and potatoes song; good but not great.
- 4 star = really good song but not the best.
- 5 star = my absolute favorites.

It is not necessary to use ratings but they are my preferred way of weighting mixes so that they always contain a certain percentage of my favored songs.

The Playlists:

Now we are ready to describe the notion of making the general mix. For this system, the goal is a pool of 500 songs to make the mix lists from. Going back to the three categories described above (highly rated, random, & least recently played), you'll need to choose how much of each you want. We'll use the following percentages for this example: 20% highly rated (100 songs), 45% random (225 songs), & 35% oldest played (175 songs).

Playlist GOODSTUFF (the 4 & 5 star rated songs) is put together like so (this is a good example of how to chain playlists so pay attention):

First, you need to decide on a split of 5 versus 4 stars; in this example, we'll go with a 60:40 split. Second, you need to decide on the minimum interval between hearing songs; in this example, we'll use an interval of 7 days for 5 star songs and 14 days for 4 star songs.

Now, here's where forcing full library rotation comes in: if you just stuck with selecting according to {Last Played NOT IN THE LAST X days} you'd wind up hearing the same songs repeatedly for the most part like some Clear Channel radio station. Here's a basic way of getting around that behavior, create 4 playlists: **4 STAR OLD**, **4 STAR RANDOM**, **5 STAR OLD**, & **5 STAR RANDOM**. By selecting some of the highly rated stuff at random and some by least recently played this forces iTunes to shuffle through all your high rated music while still being more or less random to all appearances.

Playlist	Conditions	Selection Limitation
4 STAR OLD	Match ALL {My Rating IS 4 stars} {Playlist IS NOT EXCLUSION }	20 songs selected according to least recently played
4 STAR RANDOM	Match ALL {My Rating IS 4 stars} {Last Played IS NOT in last 14 days} {Playlist IS NOT 4 STAR OLD } {Playlist IS NOT EXCLUSION }	20 songs selected at random
5 STAR OLD	Match ALL {My Rating IS 5 stars} {Playlist IS NOT EXCLUSION }	30 songs selected according to least recently played
5 STAR RANDOM	Match ALL {My Rating IS 5 stars} {Last Played IS NOT in last 7 days} {Playlist IS NOT 5 STAR OLD } {Playlist IS NOT EXCLUSION }	30 songs selected at random

From those four sublists you put together your 100 songs of **GOODSTUFF**:

Playlist	Conditions	Selection Limitation
GOODSTUFF	Match ANY {Playlist IS 4 STAR OLD } {Playlist IS 4 STAR RANDOM } {Playlist IS 5 STAR OLD } {Playlist IS 5 STAR RANDOM }	100 songs selected at random

Concept: The reason for limiting the 100 songs of the 4 sublists to 100 songs at random may not be immediately obvious: iTunes does not have any way to randomize a smart playlist unless its selection is limited. So, the list is limited so that the songs from the 4 lists will be further mixed up, otherwise they'd just be put in the same order they appear in the 4 sub-lists and we want things as random as possible to try and foil the iTunes patterning in "random" selections.

Next step is to create the random playlist, this one is much simpler:

Playlist	Conditions	Selection Limitation
RANDOM	Match ALL {Playlist IS NOT GOODSTUFF } {Playlist IS NOT EXCLUSION } {Last Played IS NOT IN THE LAST 21 days}	225 songs selected at random

Concept: By telling **RANDOM** that it can't have anything that appears on **GOODSTUFF**, duplicates are avoided and, therefore, all songs are unique. You will notice this was also used in the related lists such as **5 STAR OLD** & **5 STAR RANDOM**. This theme will be repeated throughout these playlists.

Next, create the playlist for the oldest played music:

Playlist	Conditions	Selection Limitation
OLD STUFF	Match ALL {Playlist IS NOT GOODSTUFF } {Playlist IS NOT RANDOM } {Playlist IS NOT EXCLUSION }	175 songs selected according to least recently played

Then put it all together:

Playlist	Conditions	Selection Limitation
GOOD, OLD, & RANDOM	Match ANY {Playlist IS GOODSTUFF } {Playlist IS OLD STUFF } {Playlist IS RANDOM }	500 songs selected at random

Now, that is indeed a 500 song, auto-updating pool, but what we're going to do in this case is dope it with the 4th category of music, the music you're in the mood for now. This step is purely optional, but it's an example of one way you can get a focus on music you want to hear sooner than later.

Create a regular playlist, we'll call it **HERBS**. Onto **HERBS** drag any music that you want to hear more often than a random cycling through the library will accomplish. There's no limit to what you can put on this list as it is only a source for another smart playlist that will be limited. After you have your **HERBS**, create a smart playlist, we'll call it **SPICES**:

Playlist	Conditions	Selection Limitation
SPICES	Match ALL {Playlist IS HERBS } {Last Played IS NOT IN THE LAST 14 days}	100 songs selected at random

I could make things more complex, but **HERBS** is a list that is managed manually and constantly changing so there's no need to make it any more complicated than this. There is no need to involve **EXCLUSION** because nothing goes on **HERBS** that you don't intentionally put there.

If you've been paying attention you should almost be able to predict what comes next: you "cut" the 500 song pool you made, **GOOD, OLD, & RANDOM**, with some random number of these 100 **SPICES**:

Playlist	Conditions	Selection Limitation
MIXPOOL	Match ANY {Playlist IS GOOD, OLD, & RANDOM } {Playlist IS SPICES }	500 songs selected at random

OK, that's how you generate a pool of "randomized" songs that will actually sound like something you would have picked on your own if you had the time to go through everything in your library for hours out of the day and put together a 500 song playlist and constantly update it. At this point, you could just slap **MIXPOOL** on your iPod and play it, or even shuffle it if you're so inclined. What we're going to do is generate 50 song playlists from the pool for the iPod because it makes it easier to remember where you were.

The only trick here is to exclude the list that came before to avoid duplicates:

Playlist	Conditions	Selection Limitation
PLAYLIST_I	Match ALL {Playlist IS MIXPOOL }	50 songs selected at random

Playlist	Conditions	Selection Limitation
PLAYLIST_II	Match ALL {Playlist IS MIXPOOL} {Playlist IS NOT PLAYLIST_I}	50 songs selected at random
PLAYLIST_III	Match ALL {Playlist IS MIXPOOL} {Playlist IS NOT PLAYLIST_I} {Playlist IS NOT PLAYLIST_II}	50 songs selected at random

And so on...

Tip: There is a quirk with the order that iTunes processes data sync'd from the iPod: songs are only moved on or off of the iPod according to the state of the playlists in iTunes immediately before the sync. So, after the initial data syncing that occurs when you plug the iPod in, right-click on the iPod icon in the source pane and select "Sync". Although you shouldn't have to do this second sync, you do since iTunes does not update smart playlists based upon the iPod play data until after the initial sync. If you do not do this second update, the smartlists on your iPod will always be one update out of step with the ones in iTunes.

System Two: The Intermediate

This next system is a more advanced version of the first one. In this system you're going to take advantage of on the go live updating, probably the iPod's greatest feature as no other player comes close at matching this ability. Using on the go live updating you can have multiple uniquely filtered playlists that refresh and refill themselves automatically from a weighted mix list similar to MIXPOOL from the first example .

I originally devised a version of this system for music management on a 5G 60GB iPod. There are a lot of different ways you could use so much space with smartlists, but this example isn't intended to do that. Instead, it is based upon a music pool of "only" 1000 songs. This example system serves mainly to demonstrate how you would create an auto-updating set of lists on the iPod. In the end, it's just a tweaked and advanced version of the first example, so understand this and you'll be able to do just about anything with smartlists.

The basis of this system is, again, to build your EXCLUSION playlist first since almost everything else stems from it. For example, in my library, EXCLUSION would be all one and two star rated songs, non-musical genres, and all non-mp3 filetypes (KIND in iTunes).

In the previous example, we broke categories down into random and least-recently-played sub-categories. For this system, we'll introduce a third sub-category: least-often-played. By adding this category there will be a bit of nudge for more recently added music until they catch up on playcount with older songs. It's a bit of a tradeoff, by adding in this category, your weighting for ratings, genre, etc. will be watered down somewhat, but, in turn, you will gain a more even exposure to your music library.

As a result, there's now a new distribution of song types:

- 20% 5 star songs*
- 20% 4 star songs*
- 20% random songs
- 20% longest since played
- 20% least played

* *these are also broken out into random, least recent, and least played*

So, the most basic lists are as follows:

Playlist	Conditions	Selection Limitation
4 STAR - LEAST	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 4 stars} {Last played IS NOT IN THE LAST 9 days}	67 songs by least often played
4 STAR - OLD	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 4 stars} {Last played IS NOT IN THE LAST 9 days} {Playlist IS NOT 4 STAR – LEAST}	67 songs by least recently played
4 STAR - RANDOM	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 4 stars} {Last played IS NOT IN THE LAST 9 days} {Playlist IS NOT 4 STAR – LEAST} {Playlist IS NOT 4 STAR – OLD}	67 songs by random

Playlist	Conditions	Selection Limitation
5 STAR - LEAST	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 5 stars} {Last played IS NOT IN THE LAST 6 days}	67 songs by least often played

Playlist	Conditions	Selection Limitation
5 STAR - OLD	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 5 stars} {Last played IS NOT IN THE LAST 6 days} {Playlist IS NOT 5 STAR – LEAST}	67 songs by least recently played
5 STAR - RANDOM	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 5 stars} {Last played IS NOT IN THE LAST 6 days} {Playlist IS NOT 5 STAR – LEAST} {Playlist IS NOT 5 STAR – OLD}	67 songs by random

Playlist	Conditions	Selection Limitation
LEAST	Match ALL {Playlist IS NOT EXCLUSION} {Last played IS NOT IN THE LAST 12 days} {Playlist IS NOT 4 STAR – LEAST} {Playlist IS NOT 4 STAR – OLD} {Playlist IS NOT 4 STAR – RANDOM} {Playlist IS NOT 5 STAR – LEAST} {Playlist IS NOT 5 STAR – OLD} {Playlist IS NOT 5 STAR – RANDOM}	200 songs by least often played
OLD	Match ALL {Playlist IS NOT EXCLUSION} {Last played IS NOT IN THE LAST 12 days} {Playlist IS NOT 4 STAR – LEAST} {Playlist IS NOT 4 STAR – OLD} {Playlist IS NOT 4 STAR – RANDOM} {Playlist IS NOT 5 STAR – LEAST} {Playlist IS NOT 5 STAR – OLD} {Playlist IS NOT 5 STAR – RANDOM} {Playlist IS NOT LEAST}	200 songs by least recently played

Playlist	Conditions	Selection Limitation
RANDOM	Match ALL {Playlist IS NOT EXCLUSION } {Last played IS NOT IN THE LAST 12 days} {Playlist IS NOT 4 STAR – LEAST } {Playlist IS NOT 4 STAR – OLD } {Playlist IS NOT 4 STAR – RANDOM } {Playlist IS NOT 5 STAR – LEAST } {Playlist IS NOT 5 STAR – OLD } {Playlist IS NOT 5 STAR – RANDOM } {Playlist IS NOT LEAST } {Playlist IS NOT OLD }	200 songs by random

These base lists are then combined into a pooled list, **BASE POOL**:

Playlist	Conditions	Selection Limitation
BASE POOL	Match ANY {Playlist IS 4 STAR – LEAST } {Playlist IS 4 STAR – OLD } {Playlist IS 4 STAR – RANDOM } {Playlist IS 5 STAR – LEAST } {Playlist IS 5 STAR – OLD } {Playlist IS 5 STAR – RANDOM } {Playlist IS LEAST } {Playlist IS OLD } {Playlist IS RANDOM }	1000 songs by random

A change you might notice from the first example is that these basic lists are no longer being pulled into intermediate lists such as **GOODSTUFF**. The reasoning behind the old method was to increase randomization of track order. Because tracks from **BASE POOL** will undergo at least three more randomized selections before reaching your ears, we’re opting for a cleaner approach.

This example also abandons the manually managed **HERBS** & **SPICES** for “cutting” the main mix. In this example, we will instead place the emphasis on new and unrated music automatically:

Playlist	Conditions	Selection Limitation
SPICES & HERBS	Match ALL {Playlist IS NOT EXCLUSION } {Date Added IS IN THE LAST 60 days} {Last played IS NOT IN THE LAST 12 days}	250 songs by random

Playlist	Conditions	Selection Limitation
UNRATED	Match ALL {Playlist IS NOT EXCLUSION} {My Rating IS 0 stars} {Last played IS NOT IN THE LAST 12 days}	250 songs by random

Note: There is no attempt to avoid duplication with either anything from BASE POOL or these two lists because the goal is to favor these two groups of songs.

Then the three lists are mixed together to get to the intermediate list:

Playlist	Conditions	Selection Limitation
MUSIC POOL[OFF]	Match ANY {Playlist IS BASE POOL} {Playlist IS SPICES & HERBS} {Playlist IS UNRATED}	1000 songs by random

Now, here is where the part about needing all dependent playlists on the iPod to achieve live updating comes in. If I simply put MUSIC POOL[OFF] onto the iPod, any smart lists that reference it won't live update, so what you do is copy the contents of MUSIC POOL[OFF] to a normal playlist: MUSIC POOL[ON]. You do this every time you want to refresh the music in your music mixes (i.e. about every 3-5 days), being sure to clear the contents of MUSIC POOL[ON] first.

Concept: The idea here is that as far as smart playlists on the iPod are concerned, MUSIC POOL[ON] is the entire library as every single one of them will point to it or a playlist "upstream" of it instead of the library or iPod contents in general. This is not an ideal solution, but because of the way Apple has programmed the iPod OS to treat chained smartlists, it's the only way to achieve live updating without having to carry your entire library contents on your iPod (which those of you with more iPod space than music have the option of doing that and can modify this system appropriately).

So far, so good, but now we have a new problem. The reason you need to copy your music pool songs onto a normal playlist is that you don't want to have to fit all of the dependent lists (e.g. EXCLUSION) on the iPod itself. You still need somehow of making sure that once a song is played or rated unfavorably that it won't play again, though. You could just essentially recreate all of the playlists you used to create MUSIC POOL[ON] in the first place (only this time pointing to MUSIC POOL[ON]), but, frankly, that is overkill – this list is meant to be refreshed about every 3-5 days so there's no reason to make things that complex. The solution is to make a smartlist that pulls everything not excluded from MUSIC POOL[ON]:

Playlist	Conditions	Selection Limitation
DYNAMIC POOL	Match ALL {Playlist IS MUSIC POOL[ON] } {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars} {Last played IS NOT IN THE LAST 6 days}	1000 songs by random

The final step is to create auto-updating playlists for daily use. You could simply play **DYNAMIC POOL** but you can also modify things further. These are some examples of playlists for specific situations that I have used:

Playlist	Conditions	Selection Limitation
MIX: ALL OF IT	Match ALL {Playlist IS DYNAMIC POOL }	25 songs by random

This is my general list, it includes everything I like to listen to.

Playlist	Conditions	Selection Limitation
MIX: EXPLORATION	Match ALL {Playlist IS DYNAMIC POOL } {My Rating IS 0 stars}	20 songs by random

This is my list for when I feel like listening to all new or at least previously unrated songs.

Playlist	Conditions	Selection Limitation
MIX: POKER NIGHT	Match ALL {Playlist IS DYNAMIC POOL } {Playcount IS GREATER THAN 0} {My Rating IS GREATER THAN 2 stars} {Genre DOES NOT CONTAIN <a copy of this rule for everything my buddies hate>} {Artist DOES NOT CONTAIN <a copy of this rule for everything my buddies hate>}	4 hours worth of music by random

This is my list for when I'm playing poker with my friends. It avoids stuff like classical, movie soundtracks, new age, world, and J-Pop. Additionally, because it can only pull in songs that I've listened to at least once and rated 3 or higher stars, it avoids any untested songs that might not be very good.

Playlist	Conditions	Selection Limitation
MIX: SOME OF IT	Match ALL {Playlist IS DYNAMIC POOL} {Genre DOES NOT CONTAIN <a copy of this rule for everything my wife hates or my daughter shouldn't hear>} {Artist DOES NOT CONTAIN <a copy of this rule for everything my wife hates or my daughter shouldn't hear>}	20 songs by random

This is my list for when I'm with my wife and/or daughter. It avoids stuff like heavy metal, hard rap, punk, and J-Pop.

Then all I have to do is set the iPod to sync to the following playlists:

MUSIC POOL[ON]
DYNAMIC POOL
MIX: ALL OF IT
MIX: EXPLORATION
MIX: POKER NIGHT
MIX: SOME OF IT

That's it, this gives you self updating playlists on your iPod that almost automatically pull from your entire library according to a series of rules you create that controls the precise percentages of what kind of music goes into the general pool. Because the lists are self updating, whenever you reach the end of a given playlist, merely go back into it from the Music menu and it will be completely refreshed. Similarly, there is now no need to remember where you were in a list because if you go and play another playlist, album, podcast, etc., and come back, it will automatically have dropped all of the previously listened-to tracks.

System Three: Your Final Exam

The final example of a linked smartlist system we'll go over is my favorite but, depending on your iPod and what you're trying to manage, may be more of a proof of concept than anything practical. This system is completely automated, there is no copying of intermediate lists, there is no problem of too large EXCLUSION lists, it just works once you set it up (with some caveats).

If you think about it long enough, you might realise that in any list that your were able to use the {Playlist IS NOT EXCLUSION} rule for eliminating unwanted tracks, you could have accomplished the exact same thing by instead putting NOT versions for each condition that is set as TRUE in your EXCLUSION list.

For example, the playlist, RANDOM, shown in the first playlist system might look something like this:

Playlist	Conditions	Selection Limitation
RANDOM	Match ALL {Playlist IS NOT GOODSTUFF } {Last Played IS NOT IN THE LAST 21 days} {Genre IS NOT Audiobook} {Genre IS NOT Comedy} {Genre IS NOT Holiday} {Genre IS NOT Podcast} {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars}	225 songs selected at random

And once you've realised that, it's only a little jump in thought to realise that no matter how big your iTunes library, here's a way to put this whole system on any iPod no matter what size and have everything 100% automated without ever needing to copy a thing or wait for an iTunes sync to update playlists since you can have all of the lists on the iPod and updating live and on the go. Sure, that's a lot of rules to set for each smartlist, but isn't the full automation going to be worth it?

Maybe, or maybe not; your iPod is not much of a computer. Once you move to a fully automated system, the entire contents of your iPod are fair game. Every time something tracked by any smartlist changes, the state of all linked smartlists must be updated. This can result in ten or even twenty thousand item long lists being sorted and resorted again and again in the meager memory space of your iPod. Trying to run a multiply linked playlist system like this may not be so glorious. I know because I tried this with a fully loaded 60GB 5G iPod and wound up with 90 second or more delays when moving between smartlists because of the amount of computations that had to be performed (not to mention the intense amount of hard drive activity while it did all of this).

On the other hand, while it's not going to work for managing a fully loaded full sized iPod, it does work very well with nanos, minis, or an iPod with only a 1,000 or so songs (which, conveniently, is what this example uses). I offer this bit of information as a tantalising morsel, but use the concept wisely. If you've got a nano, the system is brilliant with delays of only a second or two when moving between smartlists. If you've got a 30 or 60 GB iPod and they're fully loaded with media, don't even think about it, you will probably not be happy with the results.

Before I get to the playlists themselves, since this system involves numerous playlists that have to be on your iPod, most of which you will never want to access directly, now is a good time for...

Tip: Playlist organisation – although iTunes has the very useful feature of being able to organise playlists in folders, both the iPod and the smartlist interface lacks any such affordance. As such, it is in your best interest to adopt some sort of naming convention for playlists and folders to make it easier to navigate. You might have noticed that some of the playlists given for the second example system have prefixes in the playlist name, e.g. **MIX: SOME OF IT**. Since the iPod

displays all playlists sorted alphabetically, this gives you a way of organising numerous playlists. Allowing us to create folders on the iPod the same way we can in iTunes would be fantastic, but since that is a future (if ever) improvement, I recommend that people give something akin to this a try. Some of the prefixes I have used:

- AB** = Audiobook
- MIX** = Music playlist
- POD** = Podcast
- W** = Whole album (these are normal playlists, incidentally)
- XX** = Playlists used purely for organisation, e.g. **MUSIC POOL[ON]** & **DYNAMIC POOL** were actually titled **XX: MUSIC POOL[ON]** and **XX: DYNAMIC POOL** on my iPod so they appear below all my audiobook, music, podcast, and whole album playlists
- *** = Folder

In addition, I utilise spaces, #, \$, etc. characters as the first character in playlist names to affect how playlists sort in the flat list of the iPod or smartlist rule selection. With a bit of trial and error, you can come up with a naming convention that will let you keep things organised on your iPod no matter how many playlists you have.

Tip: If you have a lot of playlists, you will notice that iTunes only shows as many as will fit on one screen when selecting for the {Playlist IS/IS NOT} rule. Use the Page Up & Page Down buttons to move to playlists off screen.

For simplicity's sake, this system will be based around the same music distribution as the intermediate system. Onto the playlists:

Playlist	Conditions	Selection Limitation
XX[A]: 4 STAR - LEAST	Match ALL {My Rating IS 4 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 8 days}	67 songs by least often played
XX[A]: 4 STAR - OLD	Match ALL {My Rating IS 4 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 8 days} {Playlist IS NOT 4 STAR – LEAST }	67 songs by least recently played

Playlist	Conditions	Selection Limitation
XX[A]: 4 STAR - RANDOM	Match ALL {My Rating IS 4 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 8 days} {Playlist IS NOT 4 STAR – LEAST} {Playlist IS NOT 4 STAR – OLD}	67 songs by random
XX[A]: 5 STAR - LEAST	Match ALL {My Rating IS 5 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 6 days}	67 songs by least often played
XX[A]: 5 STAR - OLD	Match ALL {My Rating IS 5 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 6 days} {Playlist IS NOT 5 STAR – LEAST}	67 songs by least recently played
XX[A]: 5 STAR - RANDOM	Match ALL {My Rating IS 5 stars} {Genre IS NOT Holiday} {Last played IS NOT IN THE LAST 6 days} {Playlist IS NOT 5 STAR – LEAST} {Playlist IS NOT 5 STAR – OLD}	67 songs by random

Since these examples are based on my own systems, and I only rate music, the 4 & 5 star related lists are relatively simple as requiring a given rating eliminates everything except music automatically. The next playlists will be more detailed.

Playlist	Conditions	Selection Limitation
XX[A]: LEAST	Match ALL {Last played IS NOT IN THE LAST 11 days} {Playlist IS NOT XX[A]: 4 STAR – LEAST} {Playlist IS NOT XX[A]: 4 STAR – OLD} {Playlist IS NOT XX[A]: 4 STAR – RANDOM} {Playlist IS NOT XX[A]: 5 STAR – LEAST} {Playlist IS NOT XX[A]: 5 STAR – OLD} {Playlist IS NOT XX[A]: 5 STAR – RANDOM} {Kind IS MPEG audio file}* {Genre DOES NOT CONTAIN Audiobook} {Genre DOES NOT CONTAIN Comedy} {Genre DOES NOT CONTAIN Holiday} {Genre DOES NOT CONTAIN Podcast} {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars}	200 songs by least often played
XX[A]: OLD	Match ALL {Last played IS NOT IN THE LAST 11 days} {Playlist IS NOT XX[A]: 4 STAR – LEAST} {Playlist IS NOT XX[A]: 4 STAR – OLD} {Playlist IS NOT XX[A]: 4 STAR – RANDOM} {Playlist IS NOT XX[A]: 5 STAR – LEAST} {Playlist IS NOT XX[A]: 5 STAR – OLD} {Playlist IS NOT XX[A]: 5 STAR – RANDOM} {Playlist IS NOT XX[A]: LEAST} {Kind IS MPEG audio file}* {Genre DOES NOT CONTAIN Audiobook} {Genre DOES NOT CONTAIN Comedy} {Genre DOES NOT CONTAIN Holiday} {Genre DOES NOT CONTAIN Podcast} {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars}	200 songs by least recently played

Playlist	Conditions	Selection Limitation
XX[A]: RANDOM	Match ALL {Last played IS NOT IN THE LAST 11 days} {Playlist IS NOT XX[A]: 4 STAR – LEAST} {Playlist IS NOT XX[A]: 4 STAR – OLD} {Playlist IS NOT XX[A]: 4 STAR – RANDOM} {Playlist IS NOT XX[A]: 5 STAR – LEAST} {Playlist IS NOT XX[A]: 5 STAR – OLD} {Playlist IS NOT XX[A]: 5 STAR – RANDOM} {Playlist IS NOT XX[A]: LEAST} {Playlist IS NOT XX[A]: OLD} {Kind IS MPEG audio file}* {Genre DOES NOT CONTAIN Audiobook} {Genre DOES NOT CONTAIN Comedy} {Genre DOES NOT CONTAIN Holiday} {Genre DOES NOT CONTAIN Podcast} {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars}	200 songs by random

** As mentioned earlier, I only have .mp3 files for music, so I can eliminate videos and any definitely non-music file simply by requiring that {Kind is MPEG audio file}. If you have a mixed music library, e.g. some protected AAC, some mp3, some unprotected AAC, and an ALAC file or three, then you will have to expand this selection to include a rule that eliminates every definitely non-music file type in your library (if you have any).*

With these base level playlists, you build the first of the second tier playlists:

Playlist	Conditions	Selection Limitation
XX[B]: BASE POOL	Match ANY {Playlist IS NOT XX[A]: 4 STAR – LEAST} {Playlist IS NOT XX[A]: 4 STAR – OLD} {Playlist IS NOT XX[A]: 4 STAR – RANDOM} {Playlist IS NOT XX[A]: 5 STAR – LEAST} {Playlist IS NOT XX[A]: 5 STAR – OLD} {Playlist IS NOT XX[A]: 5 STAR – RANDOM} {Playlist IS NOT XX[A]: LEAST} {Playlist IS NOT XX[A]: OLD}	1000 songs by random

This system returns to the notion of weighting the music mixes towards recent and/or unrated music and these categories make up the last two playlists of the second tier :

Playlist	Conditions	Selection Limitation
XX[B]: SPICES & HERBS	Match ALL {Date Added IS IN THE LAST 3 months} {Last Played IS NOT IN THE LAST 6 days} {Kind IS MPEG audio file} {Genre DOES NOT CONTAIN Audiobook} {Genre DOES NOT CONTAIN Comedy} {Genre DOES NOT CONTAIN Holiday} {Genre DOES NOT CONTAIN Podcast} {My Rating IS NOT 1 star} {My Rating IS NOT 2 stars}	300 songs by random
XX[B]: Unrated	Match ALL {My Rating IS 0 stars} {Last Played IS NOT IN THE LAST 9 days} {Kind IS MPEG audio file} {Genre DOES NOT CONTAIN Audiobook} {Genre DOES NOT CONTAIN Comedy} {Genre DOES NOT CONTAIN Holiday} {Genre DOES NOT CONTAIN Podcast}	300 songs by random

Finally, these three second tier lists are combined into the final playlist:

Playlist	Conditions	Selection Limitation
XX[C]: FINAL POOL	Match ANY {Playlist IS XX[B]: BASE POOL} {Playlist IS XX[B]: SPICES & HERBS} {Playlist IS XX[B]: UNRATED}	1000 songs by random

When you make your mix playlists, you point them to XX[C]: FINAL POOL like you did with DYNAMIC POOL in the intermediate system. That's it, the end of the third system. As long as you remember to sync the second time, everything with your playlist system will be 100% automated and current with your iTunes library.

PART IV. PODCASTS & AUDIOBOOKS

Although the podcatching functionality of iTunes is perfectly acceptable, when it comes to podcast management, iTunes is positively awful. There is only one possible way to display your podcasts (reverse added order), they are counted as played even if you only listened to 3 seconds of a 60 minute podcast, podcasts must be downloaded with iTunes or they won't appear on the Podcast menu, and so on and so on. Basically, the only way to achieve any acceptable degree of control over podcasts is to set the iPod options to update all podcasts but only keep checked episodes. This works, but still leaves you at the mercy of iTunes unintuitive reverse add order as well as having to monkey about with checking and unchecking podcast tracks every time you refresh your iPod contents. Fortunately, smartlists are ideal for getting around Apple's bad design.

The first step to taking back good management of podcasts is to leave all options for managing them on the Podcast tab of the iPod options off. If you have them turned on, I recommend turning them off unless you actually like the way iTunes handles podcasts. After stopping iTunes from managing your podcasts, it is very simple to do it with smartlists. I choose to have one smartlist per podcast, so some might look like this:

Playlist	Conditions	Selection Limitation
POD: DEMOCRACY NOW!	Match ALL { Album IS Democracy Now!} { Playcount IS 0}	5 songs selected by album
POD: ON THE MEDIA	Match ALL { Album CONTAINS On The Media from NPR/WNYC} { Playcount IS 0}	5 songs selected by least recently added
POD: SLATE MAGAZINE	Match ALL { Album IS Slate Magazine Podcasts} { Playcount IS 0}	5 songs selected by album

This method does require that podcasts are tagged with something consistently identifiable, such as a unique artist or album name. Fortunately, the vast majority of podcasts are tagged with such information. Similarly, selection limitation works best when track numbers or file names are consistently tagged (ordered selection can usually be accomplished via date added as well). However, even if you have to manually add some tag information to your podcasts to achieve better functionality, it's still better than the clumsy method that iTunes uses.

Tip: If you subscribe to a lot of podcasts, you may find yourself irritated by a bug (or intentional change in functionality, with Apple, who knows) introduced in recent iPod firmware versions. It used to be that if there were no files that matched a smartlist, it didn't appear in the Playlist menu of the iPod. This was very elegant and convenient. With the most recent firmware revisions, this is no longer the case: any playlist sync'd to the iPod will appear under the Playlist menu, empty or not. This can leave you entering and exiting several playlists just seeking something to listen

to. Either you can just accept it and hope Apple fixes this in a future update, or you can try something like I did:

I made each of the individual podcast smartlists as described above but named them with the bottom sorting naming conventions described in the third playlist system, e.g. [XX\[P\]: Battlestar Galactica](#), [XX\[P\]: Democracy Now!](#), [XX\[P\]: Scott Sigler](#), etc.

I then made aggregator smartlists for the broad categories of podcasts I listen to, e.g. [#POD: Fandom](#), [#POD: News & Talk](#), [#POD: Paudiobooks](#), etc. This still leaves me with some empty playlists, but reducing over two dozen podcast smartlists to less than a quarter of that is a huge improvement.

Audiobooks also lend themselves well to smartlists. Sure, the iPod has an Audiobook menu, but it's almost as useless as the Podcast menu. Only files of the type .m4b or .aa will be displayed under the Audiobook menu, yet the vast majority of audiobooks out there are .mp3s – even my local library has begun loaning out mp3 CDs of audiobooks. In the future, Apple might choose to do something as simple and logical as treating anything with the genre, “Audiobook”, as an audiobook (and the genre, “Podcast”, as a podcast), but we shouldn't be holding our breath. I've been dealing with Apple products going back to 1994; unfortunately, “intuitive” and “easy to use” are as much advertising hype as reality. For now, though, a D.I.Y. approach with smartlists does the trick nicely.

Audiobooks are handled exactly the same way as podcasts. You make smartlists based upon some unique identifier (usually Album, which is the book's title) and use the playcount criteria for rotating book sections off of your iPod.

Some example Audiobook playlists are given here:

Playlist	Conditions	Selection Limitation
AB: A Song Of Fire And Ice	Match ALL { Album CONTAINS A Song Of Fire And Ice} { Playcount IS 0 }	4 songs selected by album
AB: The Shining	Match ALL { Album IS The Shining [Unabridged] } { Playcount IS 0 }	4 songs selected by album

PART V. TIPS, FINAL WORDS, ETC.

Putting it all together:

The purpose of this document is not to show you the be-all, end-all method of smartlist organisation, but instead to give some detailed examples so you can really dig into the smartlist system in iTunes and learn how to do it on your own. Generate other playlists, modify these schemes, come up with your own, do whatever you want. Your final goal is to have some set of top-level playlists that reflect your needs, not mine. Once you come up with that, you simply set the iPod to be synced to those top-level playlists (along with any necessary supporting lists) and then listen and enjoy. With the ideas presented in this document, you should be able to come up with a playlist scheme that works for you while minimising the work needed on your part.

Some things you might want to consider when crafting your own smartlist systems:

- Using time units instead of number of songs for deciding how much a smartlist pulls in may make the total capacity defined by your smartlists more predictable. For example, in my music collection, over 90% of the music is encoded at VBR ~192 kbps, but the length of songs varies massively, ranging from 2 minute 60s pop songs to 45 minute Grateful Dead jams. It is more consistent in terms of disk capacity to limit the base playlists by the total time rather than number of tracks when bitrate tends to be more or less the same.
- iTunes uses really, really lazy tie-breakers for selection limitations on smartlists. Whenever two or more songs match the same smartlist limitation criterion, e.g. selected by least often played, iTunes simply picks them in the reverse order of their position in the library (sorted by Artist). As such, smartlist limiting criteria that produce lots of ties will result in some clumping of artists. This is something you should keep in mind depending on whether you view artist clumping as a good or bad thing for a particular system.

Final words:

For me, a lot of the early impetus to develop an efficient smartlist management system was that I had a 4 GB iPod mini and a huge library (currently exceeding 150 GB). However, even assuming that some future iPod could contain my entire library, I would still continue to use automated smartlists for the majority of my listening. First, because I would never want to expend the mental energy on always trying to think of something to listen to. Second, human nature being what it is, if you do always choose your music, you find yourself listening to the same stuff. A huge part of why I like digital music is the power it gives me to explore so much variety in music. Capacity issues aside, a non-human selector is a must, both for purposes of saving time and energy as well as getting me to explore everything in the library.

Hopefully this document gave you some ideas about what you can do with smartlists and how to do those things. This all started off as an email to some friends about using smartlists. In its original iteration, it was virtually unreadable, I trust these later versions are an improvement on that.

Copyright 2004-2007 Charles Hannum Jr

The *iPod* and *iTunes* are products and trademarks of Apple, Inc.

Special thanks to those who have provided feedback and suggested inclusions for this paper.
Further thanks to those who have found it useful enough to pimp it out to others.

I can be found at iLounge.com, user name “Code Monkey”.