

Welcome to FoxPlay 98!

Welcome to FoxPlay 98, the ultimate in music player frontends for Windows. FoxPlay allows you to play a virtually unlimited number of music files in random order (or in the order you want) and distribute them over the Internet Relay Chat (IRC).

FoxPlay 98 includes the following features:

- **Intuitive user interface.**
- **Supports playback of MIDI, MOD (.MOD, .S3M, .XM and .IT) and MP3 files.**
- **Use your favorite MOD and MP3 players automatically with FoxPlay.**
- **Organizes all of your files into an easy-to-use Playlist that stores all sorts of information about them.**
- **No known limit to the number of files you can support.**
- **Easy-to-use Editor and Auditor systems make modifying your playlist and creating new ones quick and simple.**
- **Built-in support for mIRC allows you to send and receive music files any time you want!**
- **True randomization! Unless you specifically tell it not to, FoxPlay will always pick a different file at random.**

For more information, keep reading! Also check out the official FoxPlay webpage at <http://www.descent2.com/dorsola/fplay.htm> for the latest news and releases.

Please see the following topics for more information:

- **[What is FoxPlay? \(a brief history\)](#)**
- **[Notes to users of FoxPlay 6.22](#)**
- **[Shareware and Legal Information](#)** (Please read!)
- **[System Requirements](#)**
- **[Setting up FoxPlay](#)**

- **[The Playlist](#)**
- **[The Queue](#)**
- **[Playing Files with FoxPlay](#)**
- **[Editing File Information](#)**
- **[Loading and Saving Files](#)**
- **[FoxPlay Options](#)**
- **[The Playlist Editor](#)**
- **[Using FoxPlay with mIRC](#)**

- **[Frequently-Asked Questions](#)**
- **[Credits](#)**

What is FoxPlay? (a brief history)

Hi there. This is Matt Kellner, from sunny San Luis Obispo, California. Many of you know me as Dorsola the Dolphin. I have a keen interest in music, and have become a big-time collector of some of the Internet's best music files.

I've been on the Internet for about three or four years now. When I first got my own access, one of the first things I started doing, naturally, was collecting MIDI files. Soon I had a nice collection, but no real good way to play my files. (Windows provides its own Media Player, but I didn't like having to sift through my files and double-click on each one to play it.) I wanted a program that would let me put a bunch of files together in a list and play them at random.

I went through probably about a dozen programs trying to find the right one for my needs. Unfortunately, every MIDI player I found had some sort of limitation – some were slow, a few wouldn't play my files correctly, one of them totally screwed up my sound card, and most of them put some pretty big limitations on how many files I could play. Seeing as how my collection was well over 200 files in size at the time (at this writing, it's close to 600), this just wouldn't do. And almost all of the players required some sort of registration before they became fully functional – something I was rather turned off about.

I was about to resign myself to this fate when I came across mIRC, the leading Windows-based Internet Relay Chat client. mIRC provides a relatively simple, intuitive scripting language with support for multimedia files (namely WAVs and MIDIs). One of mIRC's strengths has always been intuitiveness of design - the \$findfile identifier automatically searches ALL subdirectories of a given directory for the particular file or file mask you're searching for. This flexibility allowed me to put together a quick list of all the files in my collection into a simple text file, and still preserve the organization method I was using for them. A related command, \$read, allowed me to then read lines at random from this list.

So, I wrote a simple script that would just read a random line out of my playlist file and play the file, then repeat itself when the file ended. Not too long after that, it occurred to me that I could make a second copy of the list and actually remove the entries from it as I played them. That way, I could have "true randomization" that would prevent files from repeating. After that, I made it possible to queue a list of files so that I could listen to files in a particular order (and/or loop them) if I so wished.

As you can imagine, I continued to add little features here and there, and soon I had a script that did everything I wanted. But then I was in the process of setting up the original #midifiles (which has since then become #musicfiles) on the IRC, and I wanted to provide people with a way to get the files in my collection. So I made the first version of PlayMIDI, which simply broadcast the files to the channel as I played them. I wrote a simple bot script that would stay active and play files, and provide a set of file servers for people who wanted to browse my multimedia. PlayMIDI in turn interacted with the file servers and allowed people to request the files as they were played.

But then I realized that I needed a way to describe the files, so that people would know what the files were, who the songs were written by, how I had rated them, etc. So I created a more complete playlist format that included all of this information, and modified PlayMIDI to broadcast the information. Soon I had people asking me for a copy of the script so they could play around with it.

So, I created a reasonable user interface and released PlayMIDI 2.0. Soon after that, I added support for MOD and MP3 files to my own copy of the script. This support really just consisted of a command to run an external player (MODPlug for MODs and WinAmp for MP3s) and a timer that would trigger at about the same time as the file ended, to keep the script running. Because this support was custom-tailored to my system, I didn't include it in the release version.

PlayMIDI 3.0 implemented support for MODs and MP3s directly by simply directing mIRC to run the MOD/MP3 file in question, relying on Windows to run the appropriate program given a set of file associations. This meant that you had to have a separate program REGISTERED in Windows in order to

play these files. Most people found this to be no problem, so they downloaded it, and *BAM* PlayMIDI took off like a rocket.

Since then, PlayMIDI evolved into a full-service script sporting a nice-looking and reasonably intuitive GUI, an Auditorer for listening to, saving and deleting new files, and an easy-to-use Queue Manager. It also included a transparent messaging system that allowed users to share parts of their playlist with each other.

Around the time that I was getting ready to release PlayMIDI 5.0, it was suggested to me that "PlayMIDI", with its support for MODs and MP3s, was too limiting a name for the script. My script seemed to be in direct competition with WavGet and a host of miscellaneous MIDI and MP3 player scripts. It needed something more general. Something that wouldn't make people think it was just a MIDI player, but at the same time something that would inspire curiosity and make it stand out among the rest. I also had (and still have) a fully-functional bot named FoxTail running my IRC channel, and it has several subsystems such as "FoxMail" and "FoxServe." To keep in line with this naming scheme, I decided to rename my script "FoxPlay."

FoxPlay continued to evolve even more. Around version 6.0, I made a major change to its playlist format, and then some serious problems started to arise. I had programmed so much into my script that it actually spanned FOUR separate .MRC files (each with a limit of around 28K of text), 9 bitmaps for the various parts of the GUI, and an entire webpage to pass as a help file. The script was slow, difficult to use and even more difficult to keep updated and error-free. New FoxPlay users had a lot of trouble setting it up - and this is no fault of their own. I'll be the first to admit that FoxPlay for mIRC is not very easy to use. Older users started seeing errors and problems where they hadn't before. And with the recent release of mIRC 5.4, all hell has broken loose for FoxPlay users. Basically, what I've run into is a program whose functions and requirements were exceeding the capabilities of mIRC.

Well, in May of 1998, a friend of mine here in San Luis (who goes by the nickname StarFire) offered to help me convert FoxPlay into a full Windows program. I had just picked up a copy of Microsoft Visual C++ 5.0, and he had his own copy, so we decided to collaborate on the project. He helped me get started with Windows programming (which I had very little experience with), and soon FoxPlay for Windows was well on its way. StarFire eventually bowed out of the project to take his job in Yellowstone Park for the summer, but by that time I had a clear sense of direction and was able to finish the project on my own.

After about two months of work, FoxPlay for Windows (which has been redubbed FoxPlay 98) is finally ready to venture out into the world. It provides MAJOR-class improvements over FoxPlay 6.22 (the last mIRC script version), including being VERY fast and much more flexible than ever before. The mIRC script has been completely rewritten to act as a simple interface between mIRC and FoxPlay, and because of the rewrite, FoxPlay is now both forward and backward compatible with mIRC versions 5.1 and up. And, being a Windows program, FoxPlay can function entirely on its own without requiring that you have mIRC installed. (The purpose of keeping mIRC in the picture is to make FoxPlay more than just a simple player.)

People have asked me why I decided to make this program freeware/shareware, as opposed to requiring registration of some sort. Well... to make a long story short, I really don't want to deal with the hassle of keeping registration information for everyone who uses my program.. Also, the spirit of this program is to provide a simple player that everyone can and will want to use. I have a strong dislike for small, simple programs that require that you pay for them, so it's part of my philosophy to make my programs free to everyone. (Larger, more versatile applications made by professional companies are a different story.)

I do appreciate monetary support, however, in exchange for the time and effort I've put into this program. You aren't required to send me anything for FoxPlay, but anything you do send (I humbly ask for \$20 US) will help me pay the ISP and electric bills that it takes to keep all of this running, and it will contribute toward future projects. Please see the **Shareware Section** for more information.

Notes to users of FoxPlay 6.0 to 6.22

If you're upgrading from earlier versions of FoxPlay, you will need to set up your options and playlist from scratch. FoxPlay 98 supports conversion between the .FPL files that FoxPlay 6.x uses and its own new playlist format, so you can tell it to use your old list when you set up the new FoxPlay. However, your options will be lost and will need to be recreated, since there was no easy way to convert those options. (That's okay, though – it's really easy now.)

You will also be presented with an entirely different method for telling FoxPlay where your files are. Instead of specifying a separate base directory for MIDI, MOD and MP3 files, FoxPlay will ask you for a LIST of directories, which can contain any type of multimedia file. This new system is MUCH more flexible and easier to understand, and it can span multiple directories and even multiple drives if you want it to! It also ensures that directories you do NOT want included won't be included, and your Incoming Directory will never interfere with your directory list.

More notes: The new Setup and Playlist Creation Wizards will make the process of setting up FoxPlay much easier than ever before. All of your options are now set in FoxPlay's **Options dialog**, and the "toggleable" options can be set via the **mIRC Popup Menu system**.

You will find that it takes next to no time at all to do major operations on your playlist, such as importing information and scanning for files. Some of these operations (particularly loading your playlist into memory and formatting the list when creating a new Playlist file) would take an excessively long time in FP 6.22 because of the limitations of mIRC's scripting language. These operations are all but instantaneous in the new FoxPlay!

Be sure to let me know if you have any problems. My e-mail address is dorsola@lightspeed.net.

Shareware and Legal Information

FoxPlay 98 is a **Shareware** program. No registration is required, and this program does not have an expiration time-limit set. No features are disabled (aside from features that simply haven't been programmed yet), and you don't have to register to get anything more out of the program. However, if you like this program and want to continue using it, I humbly ask that you send me \$20 US to help fund future projects. Your contribution will be greatly appreciated and will be put to good use. (I will do my best to provide prompt technical support and upgrade announcements to everyone.)

If you wish to contribute, please send a check or money order for \$20 US to:

Matt Kellner
1241 Johnson Ave. #223
San Luis Obispo, CA 93401

You may distribute this program to anyone you want, and permission is also granted to post the distribution ZIP file **IN ITS ORIGINAL FORM** on your website if you so desire. With the exception of *foxplay.mrc*, you may not decompile, reverse-engineer or otherwise modify any of the files included with the original FoxPlay distribution.

Permission is granted (and it's actually encouraged, too!) to modify *foxplay.mrc* for your own uses – if you wish to customize its behavior, you're certainly more than welcome to. A section is included in this Help file on the DDE interface between mlRC and FoxPlay. If you wish to include your modified *foxplay.mrc* file and any extra files and documentation specifically for these modifications in the distribution ZIP, please send me a copy of these modifications. My e-mail address is dorsola@lightspeed.net. You may not distribute these extra modifications without my permission. (If they're good mods that don't screw up everyone's computers and stuff, I'll be more than happy to grant permission for that.)

With FoxPlay being a shareware program, I need to make the following points clear. FoxPlay 98 is provided AS-IS and with no warranty whatsoever, either express or implied. While I've done my best to ensure that this program is bug-free, I can't guarantee that it'll work correctly 100% of the time for everyone. As with all programs, there are bound to be a few bugs lurking in the deepest, darkest corners. If you find such a bug, please take note of what it was and, if you can reproduce it, send me an e-mail at dorsola@lightspeed.net and I will do what I can to correct it.

Please note that you use this program **AT YOUR OWN RISK**. I am not responsible for any damage done to your computer, hardware, software, data, sanity, or the hamster wheel that's powering your system, as a result of using this program. I am also not responsible for any legal action taken against you in connection with using this program to distribute files – if you wish to distribute commercial music using FoxPlay, you do so **AT YOUR OWN RISK**, and it is expected that you are aware of the possible consequences.

System Requirements

In order to use FoxPlay 98, you must have the following:

- **A 486 DX2/66 or higher.**
- **Windows 95, 98 or NT 4.0 or higher.**
- **A soundcard (one with a MIDI Wavetable, such as the AWE64, is recommended).**

0 If you wish to play MODs and/or MP3s with FoxPlay, you will also need:

- **Most likely, a Pentium 100 or higher.**
- **A MOD Player for playing MOD, S3M, XM and IT files. (I recommend ModPlug Player 1.32 and higher.)**
- **An MP3 Player for playing MP3 files. (I recommend WinAmp 1.81 and higher.)**

Additionally, if you want to use FoxPlay over the IRC, you will need mIRC 5.1 or later. Previous versions of FoxPlay required mIRC 5.31 specifically – this limitation is no longer present, but FoxPlay does require mIRC 5.1 or up in order to display colored text and to communicate via the **DDE Interface**.

Installing and Setting Up FoxPlay

Installing and setting up FoxPlay is a simple process. Create a directory for FoxPlay (or you can use an existing one if you prefer) and unzip the contents of the *FoxPlay.zip* file to that directory. Then just run FoxPlay and you'll be introduced to the Setup Wizard.

The Setup Wizard will ask you if this is your first time using FoxPlay. If you select Yes and hit the Next button, you will then be taken through the Playlist Creation Wizard. If you select No, you will be prompted for the location of your existing playlist. If you've used FoxPlay 6.x and have a playlist from that, you can choose that list and FoxPlay 98 will automatically convert it for you.

In either case, you will also need to set up your Directory List, your Filetype-Specific Options, your General Options and your mIRC Options. Please see these topics for more information on how to set up these options.

Once you've finished with this process, you're ready to go! If you're setting up FoxPlay with mIRC, see the mIRC Setup section for more information.

The Playlist

The most integral part of FoxPlay is the **Playlist**. This is a simple, sorted list of files and their associated information. An entry in the Playlist contains the following fields:

- **Filename** (*can contain spaces!*)
- **Title**
- **Artist/Description**
- **Length in seconds**

Note: The Length field is filled in automatically for MIDI files, but it must be filled in manually for MOD and MP3 files, since FoxPlay is unable to automatically detect the length for these files. If FoxPlay plays a MOD or MP3 file that doesn't have a time set, it will issue a warning and will continue "playing" until you either fill in the time or tell FoxPlay to play something else.

Unlike its predecessors, FoxPlay 98 keeps all of this information in memory all the time while it's running. This information is saved to disk automatically when the user makes a change to the playlist or tells FoxPlay to save it. The new playlist format is binary and is therefore not modifiable via Notepad, DOS Edit or any other text editor.

If you wish to edit this information in a text editor rather than using FoxPlay's built-in editing system, you can Export your Playlist to a FoxPlay 6.x Playlist file, make your changes, and then Import the new information.

FoxPlay uses this Playlist as a source of information for the files it plays. When it's not playing files out of the queue, it picks files at random out of the list and plays them. Depending on how your options are set up and whether you're running mIRC, it may also send a **Public Broadcast** to the channels you're in and/or **echo the information locally**.

You can modify a single entry in your list by double-clicking on it and filling in the dialog box that pops up. You can also add and remove files, Import and Export information, and Audition new files via the Playlist Editor.

In the Main Player Dialog, you can elect to view your Playlist as a collection of filenames or as a list of titles and artists. Your files are sorted in the following order:

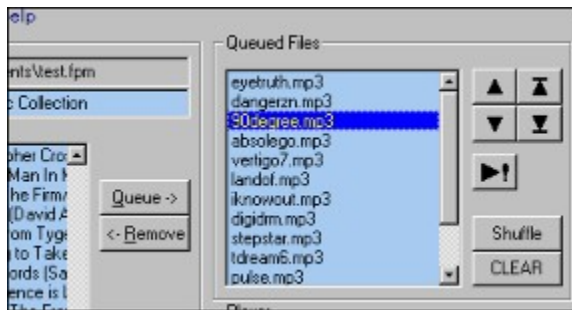
```
MIDIs (* .MID) : #-Z
MODs  (* .MOD) : #-Z      (Basically, all files
S3Ms  (* .S3M) : #-Z      are sorted first by this
XMs   (* .XM ) : #-Z      order of extensions, and
ITs   (* .IT ) : #-Z      then alphabetically.)
MP3s  (* .MP3) : #-Z
```

FoxPlay 98 currently does not have functions enabled to let you search and filter your

playlist as FoxPlay 6.22 did. I will be implementing these features at a later time.

The Queue

By default, FoxPlay randomly picks files from your Playlist when it plays them. If you wish to play these files in a specific order and/or loop them, a **Queue** is provided. This Queue lets you add files in whatever order you like, and will let you queue them as many times as you want. You can then listen to a certain set of your files over and over again without having to create a new playlist specifically for them (although you might find that to be more convenient in some cases.)



The Queue Manager is an intuitive, easy-to-use interface. Basically, all you have to do is select a file from your Playlist, hit the **Queue** button to add it to your list, and it's done! Repeat for all of the files you want. (You can select multiple files at once if you prefer.)

You can change the order in which the files appear. Select any of the files in the Queue and then use the **Up**, **Down**, **Move to Top** and **Move to Bottom** buttons to change the order of the files accordingly. The **Shuffle** button will rearrange your list in random order. If you want to take files back out of the list, hit the **Remove** button. Finally, if you want to clear your list and start over, just hit the **Clear** button.

You can save your Queue to a file (this is done automatically if you close FoxPlay with stuff still in the Queue). Using the menu, you can select "*Functions >> Queue >> Save Current Queue*", then specify the name of a file to save it to. By default, Queue files have the .CUE extension.

You can also load a saved .CUE file from disk. Go "*Functions >> Queue >> Load Queue File*", select the filename and its contents will be added to your current queue. **Note: Loading a .CUE file does NOT replace the contents of your current queue, if any.** This feature is provided so that you can load the queue multiple times if you want.

There are a few more things you can do with the Queue if you're using mIRC as an interface to FoxPlay. See the **Popup Menus** and **Text Commands** sections for more information.

Playing Files

Like most multimedia player systems, FoxPlay has a relatively simple set of commands for playing and stopping the player. It has a **play** button, a **stop** button, and a **repeat** button.

While FoxPlay is playing a file, information is displayed in the Player area describing the file being played. You can modify this information on the fly by hitting the **Edit** button, and you can also opt to **remove** the file from your Playlist if you decide you didn't like it.

The **Play** button has multiple functions. If it appears as a triangle with a bar in front of it, hitting this button will play the next file in the Queue. (This will only be the case if you have something in the Queue.) If it has a question mark (?) next to it, hitting this button will play a random file.

There are two other Play buttons in the system, both triangles followed by exclamation points (!). These are known as "**Play now!**" buttons and let you play a selected file from either the Playlist or the Queue immediately. (Note: if you play a file from the Queue, it'll be removed from the Queue at that time.)

When a file is finished playing, FoxPlay will do the equivalent of hitting the Play button to start playing another file. If there are files in the Queue, it'll go through them one-by-one, pulling from the top, until there aren't any more. Once the Queue is empty, it will pick files at random until you stop the player. The only times this is NOT the case are when you have **Remote-Control Mode** turned on and/or **Random-Pick Files** turned off. See the **General Options** section for more information.

To play **MIDI files**, FoxPlay uses the Windows Multimedia System (technically referred to as the MCI device driver for the MIDI sequencer). While it doesn't auto-detect the length of a MIDI ahead of time, Windows notifies FoxPlay automatically when a file is finished playing, so that it can store the elapsed time and move on to another file.

To play **MOD files** with a MOD player, and **MP3 files** with an MP3 player, FoxPlay runs programs designed for playing these files. You have to tell FoxPlay where these players are before FoxPlay can work with files of these types. Unlike its predecessors, FoxPlay 98 actively runs and shuts down these external programs in order to play the files, rather than running the files themselves and depending on Windows to run the proper programs. This eliminates the need for the dummy files *FP_STOP.MOD* and *FP_STOP.MP3*, which were created by older versions of FoxPlay.

Note: FoxPlay needs to know how long your MOD and MP3 files are! FoxPlay is unable to automatically detect the length of these files. If you have MOD and/or MP3 files and want to play them, you must specify their length (which you can usually get from their associated players) so that FoxPlay will know when to stop playing them and move on. If you do not specify this length, the "elapsed time" bar in the Player will flash at you and FoxPlay will issue a warning. Since it won't know how long the file is,

it will continue to “play” the file (basically, it’ll sit there counting time) until you either tell it how long the file is or tell it to do something else.

Once it knows how long the file is, FoxPlay will attempt to close down the MOD/MP3 player. For the most part, this works, but I’ve seen a few instances where it’s unable to close down WinAmp for some reason. I don’t know exactly why this works some times but not others, but I don’t appear to have the same problem with MODPlug or with any other programs I’ve tested FoxPlay with.

In the case of WinAmp, the good news is that if this happens, it won’t open multiple instances of itself when FoxPlay tells it to play more MP3s. The bad news is if your MP3 is longer than you told FoxPlay it was, you could have an MP3 and a MIDI playing at the same time, or you could even see error messages from your MOD player indicating that it wasn’t able to start playing because someone else was already using the soundcard.

Note that FoxPlay doesn’t need you to specify the lengths of the MIDI files in your Playlist, since the Windows Multimedia System takes care of this automatically.

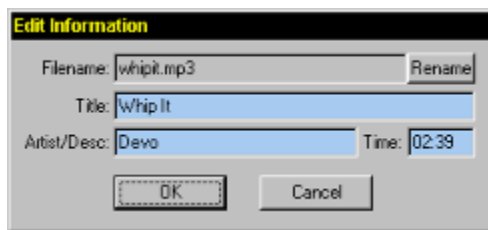
When FoxPlay is finished playing a file, it pauses for about 6 seconds before playing the next file. This pause is meant to make the music seem less rushed and more comfortable – kinda like a jukebox. The only exceptions to this rule are in the cases I described above (where FoxPlay stops itself altogether) and when you have a particular file set to loop a certain number of times. If you have, say, *cantaloop.mid* queued five times in succession, FoxPlay will not pause before playing the file again. This comes about as close as possible to truly looping the file. (I will attempt to code true loop/repeat support for MIDIs into a later release).

Finally, hitting the Repeat button will allow you to restart the file that’s currently playing or to replay the file that last played if the player is currently stopped.

Editing Files

Editing files in the main player is relatively simple. All changes you make affect the main Playlist. With the exception of when you rename a file in your Playlist that shows up in the Queue, you will not need to go back and change anything else. You also don't need to "Reinitialize the Randomizer" like you did in FoxPlay 6.22 to keep your information current..

To edit a single file's information, you can select it and hit the **Edit** button in the Playlist area, or you can simply double-click on it. You can also double-click on a file in the Queue to modify its information, and you can hit the Edit button in the Player area to modify the information for the file that's currently playing.



Doing any of these things will bring up a small dialog box that will let you modify the file's information. You can hit the **Rename** button to bring up another dialog that will let you type in a new name for your file, giving you active feedback on whether or not it can save the file with the new name.

Just type in your new information and hit **OK**. If you wish to change the time for a file, you can simply type it in as either a count of seconds (for instance, 159 seconds is 2 min 39 sec), or in MM:SS format (2:39). FoxPlay will translate the value either way.

When you hit OK, your Playlist will be updated and saved with the new info.

To make more extensive changes to your playlist itself (for instance, removing files or adding new ones), use the Playlist Editor.

Loading and Saving Playlists

Loading and saving FoxPlay Playlists is really just like loading and saving any other type of file – just use the *File Menu* to do so. By default, FoxPlay saves its Playlist files with a **.FPM** extension – short for “**F**ox**P**lay **M**aster playlist” – although you can change this extension if you prefer. (Changing the extension for your playlist file will make it more difficult to locate in the future, however. It is recommended that you use FoxPlay’s default settings.)

Your Playlist is saved automatically when you make changes to it. If you wish to change its filename, simply go “*File >> Save As*” and enter your new filename.

You can also **Export** your Playlist to a simple text file, which contains organized summaries of all the files in your List, or to a FoxPlay 6.x PlayList (***.FPL**). Please see this topic for more information.

The FoxPlay Options Screen

FoxPlay 98 provides a place for you to setup your options the way you want them. These options are divided into the following categories:

- **Filetype-Specific Options**
- **General Options**
- **The Directory List**
- **mIRC Options**

For more information on these options, click on one of the topics above.

Filetype-Specific Options



FoxPlay divides your files into three distinct groups – MIDIs, MODs (which include **.MOD**, **.S3M**, **.XM** and **.IT** files) and MP3s. The above “Option Grid” allows you to customize FoxPlay’s behavior for these three filetypes. When a button appears “pushed in”, that option is enabled. Here’s a detailed explanation on how these options work:

Allow Playback

Simply put, if you have this option checked for a filetype, FoxPlay will be allowed to play files of this type. This means that the file can be picked at random, and you can also direct FoxPlay to play it immediately or via the **Queue**.

This is a free toggle for MIDI files, but if you want to play MODs and MP3s, you must specify a **player** for each. Until you do, the **MOD** and **MP3** buttons are disabled.

You can type in the path and filenames of your MOD and MP3 player programs in the fields provided, or you can hit the question-mark button (?) to locate the files. Once a **valid** filename is entered into these fields, the **Play MODs** and **Play MP3s** options will be available.

Auto-Request

If you’re using **mIRC**, you can have FoxPlay automatically request files that you don’t have when other people play them. This way, you can start building a collection! When this option is enabled for a certain filetype, and someone plays a file of that type that you don’t have, FoxPlay will send a private message through mIRC requesting the file via the standard **!nick filename** system (for instance, “!*Dorsola heyyou.mid*”). In most cases, people will automatically send the requested file to you. If the other person is also using FoxPlay 98, he/she will automatically send you information on the file when you receive it.

When you receive a file from another user, that file will go in your **Incoming Directory**. By default, this directory is your default mIRC “DCC Get” directory, where files of any generic type will go when they’re received. For info on how to set this directory yourself, see the **General Options** section.

Once you’ve received a file, you can **audition it** and decide whether or not you want to

keep the file. See the [Auditioner](#) topic for more info.

If you prefer to request only certain files (rather than all the files of a particular type), I recommend leaving Auto-Request turned off. You can use the [Popup Menu](#) to manually request the file in the same manner, or you can simply type the request. See the section on [Requesting Files](#).

Show to Public

This option is also known as **Public Broadcast**. Basically, when you play a file, you can have mIRC advertise it for you. A message will go out to the channel(s) you're on (unless you've specified a specific channel or set of channels) that will look similar to this:

```
<FoxTail> FoxPlay 98: Splintering Heart (Marillion) -> 64.82 KB  
(06:52) <- To get: !FoxTail splinter.mid
```

Note the **In Color** toggle next to the "Show To Public" buttons. This option lets you specify whether or not FoxPlay should include mIRC colors in its broadcasts. It's recommended that you pay CAREFUL attention to the rules of the channels you're in. If you get kicked because you're sending out colors, you should either modify your [channel list](#) or turn **In Color** off.

If you aren't on any channels, FoxPlay won't send out a message.

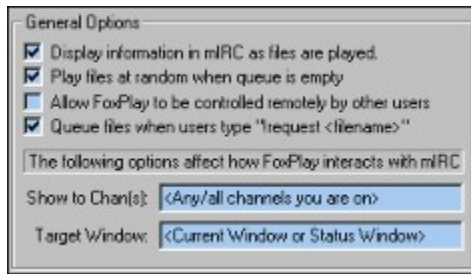
NOTE: You will not see your own Public Broadcasts! Unless you're running multiple clients in the same channel, you won't be able to see your own messages to the channels when you play files. This is because FoxPlay is designed to not clutter your screen with your play messages. If you want to see what you're playing when you play it, you can turn on [Local Display](#) so that your files will show up on your screen.

It's very important that you keep this in mind when using FoxPlay with mIRC. Since you can't see your own messages, you may inadvertently join a channel where mIRC colors are not allowed, or some other rules are in place against playing music files. In this case, you will want to either [change the channels that FoxPlay broadcasts to](#), or turn off Public Broadcasting for all three file types. (It's also suggested you do either one of these things if/when people ask you to stop playing in their channel.)

People who see your files can request them in the same manner as was described above. Unless you have another script that overrides FoxPlay's responses to **!nick filename** requests, FoxPlay will send the requested file to the user. Therefore, to keep people from hogging your bandwidth, you may want to turn off Public Broadcasting.

The next section, [General and mIRC-specific Options](#), covers things such as Local Display, Remote-Control Mode, your Incoming Directory and Randomization.

General and mIRC-specific Options



These options further modify FoxPlay's behavior, but do not pertain to any of the different types of files (MIDIs, MODs and MP3s) in particular. Here's how they work:

Display Information in mIRC

If you're using mIRC with FoxPlay, you can have it display information to your screen as it plays files. This is handy if, say, you want to keep track of what's going on without having to switch back and forth between mIRC and FoxPlay. This information does not get sent out over the IRC – it simply gets echoed to your screen. By default, it echoes this information to whatever window happens to be active at the time. If no window is active, or the window that IS active doesn't support text echoes, the text will be sent to your **Status Window** instead.

You can specify a particular window for this information to go to. This window can be any valid name of an open window (such as a particular channel, or a **custom @Window**) or it can be one of the following:

- a = Equivalent to "Active window"
- s = Forces text to go to the Status Window at all times.

Basically, whatever you input into this field will be inserted into an mIRC **/echo** command. If nothing is specified, **-a** is used by default.

Note: This option is independent of **Public Broadcasting** – it need not be active in order for you to show your files to the public, nor do you have to be broadcasting in order to see the files locally. In fact, you can turn both of these options off and let FoxPlay act as a completely transparent player! (Note that this is automatically the case if mIRC is not running.)

Play Files at Random

FoxPlay's default behavior is to basically act as a virtual DJ of sorts – it plays a large number of files in random order, and it keeps going until you tell it to stop. A **Queue** is provided so that you can play files in a specific order (including looping them). But if you want to make it so FoxPlay ONLY plays files that are queued, you can turn this option off to cause the player to stop once your Queue is empty. Some users prefer to

have FoxPlay stop as sort of a signal that more files need to be queued.

If this option is on, FoxPlay starts playing files randomly based on your **Allow Playback** options. When this option is turned off, FoxPlay will “go to sleep” when there’s nothing in the Queue. You can tell it to play a single random file by simply hitting the **Play** button.

Allow FoxPlay to be Controlled Remotely

This option is otherwise known as **Remote-Control Mode**. It basically puts FoxPlay into a “standby” mode which waits for other users over the IRC to play files that you have. Depending on your **Allow Playback** option settings, FoxPlay will automatically start playing any files you have when other people play them.

If someone plays one of your files while you’re already playing something, it adds the file to the **Queue** and plays it in sequence after the current file’s done. This is both to make it so FoxPlay doesn’t override what’s currently playing, and also to allow multiple users to control FoxPlay without causing problems.

Note: Remote-Control Mode does NOT provide a method of access to your computer. It simply responds to the standard **CTCP SOUND** requests that other users send out over the IRC. Therefore, you don’t have to be worried about this allowing someone to hack into your computer or anything like that.

While Remote-Control Mode is active, FoxPlay will do the following:

- It will NOT display anything to the public, despite your settings. (This is to avoid cluttering the screen with redundant information.)
- It will NOT pick files at random, but will instead wait for more Sound Requests to come in. This overrides **Play Files at Random**.
- It will play files out of the Queue same as normal, and will add more files to the Queue if it’s already playing something when a recognized Sound Request comes in.

If you aren’t using mIRC with FoxPlay, you should leave this option turned off.

Queue Files When Users Type !request

FoxPlay also provides a sort of “jukebox” function to other IRC users. If you have this option enabled, people can type “*!request <filename>*” in the channel you’re in (or as a private /msg to you) to queue files that you have. The syntax for this command is:

!request <file1> [numtimes] [<file2> [numtimes] ...]

For example:

!request anyway.mid 3 splinter.mid iridium.mod 4

If you type that in my channel and I have all three of the specified files, I will have queued **anyway.mid** 3 times, **splinter.mid** once, then **iridium.mod** 4 times. No single file can be requested more than 10 times via this method, so someone can't deliberately try to crash your copy of FoxPlay by telling you to queue something half a million times.

In the case I outlined above, you would have received the following response from me:

-Dorsola- (FoxPlay) I've queued the following: anyway.mid (3 times), splinter.mid, iridium.mod (4 times)

With the exception of responding with a list of files that were queued, this function works exactly the same as if the person on the receiving end of the **!request** command had typed **/queue <file list>**. Please see the [mIRC Text Commands](#) section for more information.

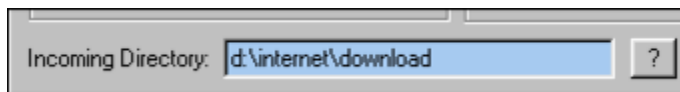
Show to Chan(s)

By default, when you **broadcast files to the public**, your broadcast will go out to any and all of the channels you're on (if any). However, you can use this field to specify a particular channel or list of channels that you want to broadcast to, and a broadcast will be sent only if you are on those channels. (If you're on a channel that's not in this list, that channel will NOT receive a broadcast.)

Target Window

This field specifies a particular mIRC window to which FoxPlay should echo song information when it plays a file. By default, this is whatever window happens to be active at the time, or the Status Window if no window is active or FoxPlay is unable to display to that window (for instance, if the active window is a picture window). You can specify the particular window as described above in the **Display Locally** option.

Incoming Directory



This directory is where FoxPlay stores new music files that come in from the IRC. When someone sends you a MIDI, MOD or MP3 file and you have [foxplay.mrc](#) loaded, FoxPlay will store this file (moving it if necessary) to your **Incoming Directory** and will also store some information about the file.

By default, this directory is whatever happens to be your mIRC **DCC Get Directory** – the directory in which any generic file would be stored. You can change the destination

for your new multimedia files by changing this value. FoxPlay will automatically update the options in its script portion so that the changes will take effect.

If you attempt to set this directory to one that does not exist, it will be set back to the default directory as provided by **mIRC** (if it's running). If you set your Incoming Directory to one that already appears in your **Directory List**, that directory will be removed from your Dir List so that the two won't interfere with one another.

A file named *fp_new.fpi* will be created in this Incoming Directory when you receive a file from another user. This file contains information about the files coming in. It also stores any changes you make to their information via the **Auditioner**, which you can use to listen to them and evaluate them before making them part of your collection. Please see that section for more information.

The Directory List

Unlike FoxPlay 6.22 and earlier, which used distinct “base directories” for each type of file, FoxPlay 98 uses a list of directories, which can contain any combination of filetypes. This directory list organization method is both more flexible and easier to understand.

In the Options Dialog, you can see your Directory List. To make changes to it, either double-click on the list or hit the **Modify List** button. This will take you to another dialog where you can add or remove directories to/from your list.

The left pane of this window shows your whole filesystem. You can select a directory in this pane (expanding and collapsing the various directory trees as needed) and hit the **Add** button to add that directory to your list. Your list shows up in the right-hand pane.

Alternatively, you can hit the **Add Subdirs** button, and the directory you have selected and ALL of its subdirectories will be added all at once. This is useful if, say, you have a directory for your MIDIs with subdirs for different artists, or organized in alphabetical order.

To remove a directory, simply select it in the right-hand list and hit the **Remove** button.

FoxPlay searches through the directories in this list whenever it tries to locate files for playing or sending information. It also iterates through these directories when **scanning for new files** to add to your Playlist. **It does NOT automatically search subdirectories!** Therefore, you need to make sure that any and all subdirectories that you want included in your Playlist are present in your Directory List.

Part of the reason for this change is so that your **Incoming Directory** doesn't interfere with your main Directory List. In the past, it's been difficult for some people to manage their files because they would have a “new” directory within the same base directory as the rest of their files, and FoxPlay would traditionally pick up on these files without the user's knowledge or consent. The new Directory List system eliminates this problem.

If/when you add a directory (or set of directories) that coincides with your **Incoming Directory**, the Incoming Directory will be set back to its default value. This eliminates any possible conflicts between the two. (See the section on **General/mIRC Options** for more information.)

NOTE: FoxPlay 98 does not currently have directory lists associated with particular Playlist files. I plan on adding this functionality in a future release. For the time being, if you wish to change Playlists often, and these Playlists cover a different set of directories, you will need to manually change your Directory Lists accordingly.

Tip: You can create separate playlists for separate directories all at once, then set your Directory List to include ALL of those directories and just call upon the directory-specific Playlists. So long as you don't have any duplicate filenames between the different

directories, these separate playlists will remain separate and will eliminate your need to change directories each time.

The Playlist Editor

The Playlist Editor is the part of FoxPlay that allows you to make any sort of extensive changes to your Playlist. From here, you can perform the following operations:

- [Edit Your Playlist](#)
- [Add or Remove Files From Your Playlist](#)
- [Import and Export Playlist Information](#)
- [Create a New Playlist](#)
- [Audition New Files](#)

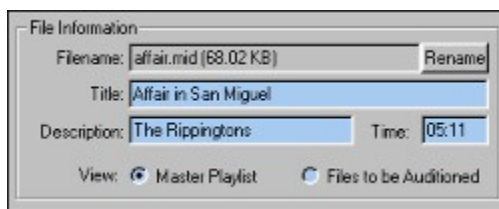
Click on one of the topics above for more information.

From this window, you can also [Modify your Options](#) and [Load and Save Playlists](#).

Editing Your Playlist



This section of the Playlist Editor allows you to select a file and modify its information. To edit a file, simply select it in the list. Its information will show up in the **File Information Area**, shown below:



Here, you can view and modify the info for the currently selected file in your Playlist. Just place your cursor in the desired field and type in your new information. If you wish to change the time for a file, you can type it in as either a count of seconds (for instance, 159 seconds is 2 min 39 sec), or in MM:SS format (2:39). FoxPlay will translate the value either way.

If you want to **rename** the file, hit the **Rename** button and enter your new filename. FoxPlay will give you feedback as to whether or not it can successfully change the file's name.

You can also **Add New Files** and **Remove Existing Files** here. See these topics for more information.

Any changes you make here will be stored to your Playlist. If you wish for the **Main Player** to pick up on these changes without closing the Editor, you can hit the **Apply** button. Otherwise, hit **Done** to close the Editor. FoxPlay will then update the Player window with the new information.

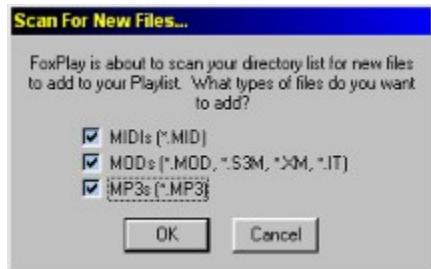
NOTE: If you are making extensive changes to your Playlist, it is highly recommended that you **Save your Playlist** (perhaps with a different name) before you exit. That way, if you find you didn't like the changes you made, or if you're doing this to split up your Playlist into separate sections, you will not lose your original Playlist.

Note in the dialog box above that you can select "Files to be Auditioned" as well as "Master Playlist." This basically switches the editor between looking at your Playlist and the Auditioner when you're editing file information. This is mainly to keep both

itself and you from getting confused while working with files.

Adding and Removing Files

You can add files to your Playlist in a number of ways. The first (and easiest) way to do this is to tell FoxPlay to **Scan For New Files**. Hit the **Scan For New Files** button, and you will be presented with this prompt:



Simply check the types of files you want to have FoxPlay add to your Playlist (if it finds them). When you hit **OK**, FoxPlay will search through your **Directory List** for any files that don't already appear in your Playlist. These new files will be added automatically to your list with "default" information (you will need to either **Edit their info** or **Import Information from another Playlist**).

In order for this to work, your **Directory List** must include all of the directories in which your music files reside.

An alternative method of adding files is to **Add Them Manually**. Use the menu to select "*Edit >> Add Specific Files*", then find and select all of the files you want to add (you can only add from a single directory at a time), and hit **OK**. If the directory that contains these files isn't in your **Directory List** already, it will be added.

Note that if you attempt to add a file that's already in your Playlist, it will not get added a second time. Each entry in your Playlist must have a unique filename, so if you have duplicate files on your hard drive, only one of them will be counted. I plan on adding a feature in a future release that will find and point out duplicates in your directory structure.

You can also use the **Auditioner** to add files. See that section for more information.

If you want to remove files from your Playlist, simply select them in the right-hand list and hit the **Remove** button. You will not be prompted for this action. You can also tell FoxPlay to **Clean Up Your Playlist** by selecting "*Clean Up List*" in the Edit menu. When you tell FoxPlay to clean up your list, it attempts to locate all of the files in your Playlist, searching your **Directory List** and removing any entries it can't find. This way, you can avoid error messages such as "Unable to locate thisfile.mid".

FoxPlay is capable of deleting files from your hard drive as well. This is a convenient feature for the **Auditioner**, but you can also use it to manage your Playlist. As a security feature, I've made it so you can only delete one file at a time via FoxPlay. If

you want to delete a large number of files, you would be best advised to use Windows Explorer to delete them, then **clean up your list** as described above.

Simply select the file you want to delete, then hit the **Delete** button. You will be asked if this is really what you want to do – if you hit Yes, the file will be permanently deleted. (It will NOT be sent to the Recycle Bin!) Once the file's gone, its entry in your Playlist will also be removed.

Importing and Exporting Information

You have the ability to import information from one Playlist to another, or to export your current Playlist to a text file or a FoxPlay 6.x Playlist File (.FPL).

Let's say you just installed FoxPlay for the first time, or you just got a whole ZIP full of MIDI files from someone over the IRC. When you add these new files to your Playlist, you won't have any information on them (other than their filenames). Rather than type in all of this info yourself, you can **import** this info from another playlist.

In the **Playlist Editor**, go "*File >> Import Info From...*" and select the Playlist file to import from. FoxPlay can import information from other FP98 (.FPM) and FoxPlay 6.x (.FPL) Playlist files. (It will not support playlists that were created with any version of FoxPlay earlier than 6.0.)

Note: FoxPlay will prompt you if it finds an entry in your Playlist that has info that doesn't match what's in the Playlist you're importing info from. You can opt to accept the new information or keep the old info. You also can tell FoxPlay to automatically accept or reject all remaining info for files such as this one.

If you want to share your Playlist with other users who are using older versions of FoxPlay (or may not be using FoxPlay at all), you can **export** your Playlist either as a FoxPlay 6.x Playlist File (.FPL) or as a text file (.TXT). Go "*File >> Export To*" and select the type of file you want to save, then specify a name and location for the new file.

When you export a file as a FoxPlay 6.x FPL file, people using FoxPlay 6.x can then read this Playlist, use it to help them create their own Playlists, etc. When saved as a text file, your Playlist will appear as a collection of summary lines showing the file's name, title, artist/description, size and length (if known). Each line will basically look similar to the line people see when you **Broadcast Your Files** over the IRC.

Keep in mind that the new Foxplay 98 .FPM format is a binary format that cannot be easily read or edited in a text editor. If you wish to make a bunch of changes to your Playlist using *NotePad* or *DOS Edit*, it is recommended that you **export** your playlist to an FP 6.x .FPL file, make your changes, then **re-import** this Playlist with the modified information.

For reference, the FP 6.x .FPL format is as follows:

FPL#54_FORMAT

```
filename.ext \ title \ artist-desc \ length
filename2.ext \ title2 \ artist-desc2 \ length2
...
```

The first line MUST read **FPL#54_FORMAT**. This is how FoxPlay 6.x can tell that this file is a Playlist file.

The only other requirements for **.FPL** files are as follows:

- If the list is to be used in FoxPlay 6.x, filenames must not contain spaces.
- “title” should not be blank. If you don’t know the title of the song, it is recommended that you use the file’s name as its title.
- “artist-desc” can be blank, but a space must separate the backslash (\) characters.
- “length” must be an integer indicating the length of the file in seconds. Specifying 0 or -1 will tell FoxPlay that the length of the file is unknown. (This field can be blank.)
- The entries in your list need not be in alphanumeric order.

Creating a New Playlist

FoxPlay has a Playlist Creation Wizard which will guide you through all the steps of setting up a new Playlist. To access it from the Playlist Editor, go “*Playlist >> New*”, then simply follow all of the prompts.

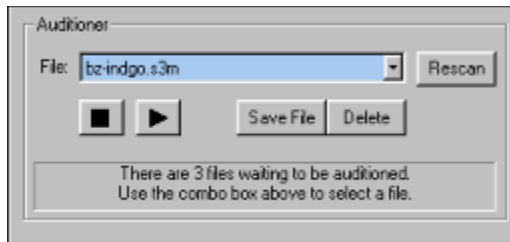
Creating a new Playlist is divided up into the following steps:

- 1) Giving your new list a title.
- 1) Setting up your Directory List to include the directories you want.
- 1) Scanning for new files**
 - 1) Optionally importing information from another Playlist
 - 1) Setting up your Filetype-Specific Options
 - 1) Setting up your General and mIRC-Specific Options
- 1) Saving your new Playlist.

See the underlined topics for more info on those particular items. Although the layout of the Wizard isn't the same as in the Options Dialog itself, the options all mean the same thing.

The FoxPlay Auditorer

In the Playlist Editor, you have what's referred to as an **Auditioner** – a subsystem of FoxPlay that lets you listen to and evaluate new files before you add them to your Playlist.



The Auditorer works with your **Incoming Directory** to determine what new files are ready to be listened to. Use the “combo box” to select a file, then hit the **Play** button to listen to the selected file. New music files that you receive over the IRC will be stored in this directory as they come in, if you have **set up FoxPlay to work with mIRC**.

NOTE: You need to have a **MOD and MP3 Player** set up in order to audition MOD and MP3 files (respectively). Otherwise, you will be given an error message when you attempt to play the files.

If you don't have an **Incoming Directory** set up, or if there aren't any files waiting to be auditioned, this area won't do much.

It's not required that you audition these new files. The Auditorer is mainly provided as a convenient method by which you can filter out the files you don't like.

When you select a file in the combo box, its current information will pop up in the **File Information Area**, where you can modify it on the fly. If you decide you want to keep the selected file and add it to your playlist, hit the **Save File** button and select a directory for the file to be stored in. The file will then be moved from your **Incoming Directory** to the directory you selected, and its information will be added to your Playlist.

If you don't want to keep the file you selected, you can hit the **Delete** button to get rid of it. After answering “Yes” to the prompt that appears, the file will be deleted permanently.

Finally, hit the **Rescan** button to have FoxPlay re-analyze your **Incoming Directory** and check to see if any new files have shown up. The combo box will be updated accordingly.

Using FoxPlay with mIRC

FoxPlay 98 provides built-in support for **mIRC**, the leading **IRC** client for Windows. The file *foxplay.mrc* which is included with FoxPlay 98 is basically an interface between **mIRC** and FoxPlay.

FoxPlay 6.22 and below were written entirely in **mIRC-script** – the scripting language that **mIRC** provides. As a result, these earlier releases had several drawbacks: They were slow (particularly in setting up your Playlist), they **REQUIRED** that you have **mIRC**, and they often didn't work properly or at all on some peoples' systems, or had adverse interactions with other scripts people would have installed. With the new release, however, all of the functions governing Playlist management, playing files and manipulating options are now done within the FoxPlay executable, and the **mIRC script** is now a simple interface between the two programs.

foxplay.mrc, when loaded in **mIRC**, intercepts **CTCP SOUND** requests from IRC channels and sends the information on to FoxPlay via what's referred to as the **DDE Interface**. FoxPlay then looks at how your **Options** are set up and what's in your **Playlist**, then reacts accordingly.

This script file also allows you to share parts of your Playlist with others, and also lets you queue files and display information via a series of **Text Commands**.

See the following topics for more information:

- **Setting up mIRC to work with FoxPlay**
- **Text Commands**
- **Popup Menus**
- **Requesting files from other people**
- **Sharing Playlist information**
- **Modifying Options**

Many of FoxPlay's **Options** define how FoxPlay interacts with **mIRC**. Therefore, please make sure you read the Options section of this Help File so you understand how everything works.

And if you're interested in working directly with *foxplay.mrc*, or creating your own script to work with FoxPlay 98, take a look through the **DDE Technical Reference** for instructions on how to communicate with FoxPlay through mIRC's **DDE interface**.

Setting up mIRC to work with FoxPlay

FoxPlay 98 comes with a script file, *foxplay.mrc*, which allows **mIRC** to work with the program. This file handles keeping track of FoxPlay's options, passing playlist information between the player and the outside world, and giving you the ability to do some more advanced things, such as queue multiple files at once.

If you're already using FoxPlay 6.22 (or another version), you can simply load this new *foxplay.mrc* over your existing one in the **mIRC Remote**. The script will take care of the rest.

To set up **mIRC** for use with FoxPlay (and vice versa), simply load *foxplay.mrc* in your **mIRC Remote**. (To do this, go "*Tools >> Remote*", then "*File >> Load >> Script*", find the new *foxplay.mrc* and select it. This file will then be loaded into your **Remote**, and once you hit **OK**, FoxPlay will set itself up.

You will most likely be asked whether or not **mIRC** should allow the script to initialize itself. Answer **Yes** to this question. You will then be asked where *foxplay.exe* is. Locate your executable (should be in the same directory as the *.mrc* file). The script will then execute *foxplay.exe*. If you haven't already set up FoxPlay itself, FoxPlay will guide you through the setup process.

That's really all there is to it! Unlike previous versions of FoxPlay, FP98's setup is fast, simple and should be easy to understand. Remember that all of the processing takes place within FoxPlay itself, so you no longer have to be worried about **mIRC** locking up or spending a lot of time processing your Playlist.

Text Commands

Once you've loaded *foxplay.mrc*, you will be able to use the following text commands:

/foxplay

Tells FoxPlay to play a random file, or the next one in the queue if applicable.

/foxplay filename

Tells FoxPlay to play the specified file. This filename can contain spaces, but it must match a file in your Playlist. If FoxPlay cannot find the specified file, it will issue a warning and ignore the filename given.

/playmidi ...

Same as **/foxplay** – this command is a holdover from when FoxPlay was known as **PlayMIDI**.

/foxstop

Stops the player.

/stopmidi

Same as **/foxstop** – this command is a holdover from when FoxPlay was known as PlayMIDI.

/fp_startup

Use this to “kick-start” FoxPlay in case it stops talking to mIRC. If you're playing files but not seeing FoxPlay interact with mIRC like it should be, use this command to get the two programs talking to each other again. If that fails, try closing down and restarting FoxPlay.

If FoxPlay isn't currently running, typing this command will execute FoxPlay.exe. If you want a particular action to occur as soon as FoxPlay starts up, type **/set %fp_action <action you want performed>** - when FoxPlay finishes loading up, the script will then perform the specified action.

/queue file1 [numtimes] [file2 [numtimes] ...]

Tells FoxPlay to add the specified file(s) to the Queue. FoxPlay will only add files it can find in your Playlist. The filename(s) given can contain spaces, but they must be followed up by a valid file extension. You can specify multiple filenames (each with its own repeat number) if you want.

numtimes specifies the number of times to repeat the file. Let's say you wanted to loop “*crush.mp3*” 4 times in a row. Instead of typing **/queue crush.mp3** four times, or **/queue crush.mp3 crush.mp3 crush.mp3 crush.mp3**, you could just say **/queue crush.mp3 4**.

/cue ...

Same as **/queue**

/repeat [numtimes]

If **numtimes** is specified, this command tells FoxPlay to queue the currently-playing file (or the last one that was played if FoxPlay's not playing anything currently) the specified number of times. This is equivalent to typing **/queue <current_filename <numtimes>**

If you just type **/repeat** by itself, FoxPlay replays the current or most recent file immediately. This is equivalent to typing **/foxplay <current_filename>**

/clearqueue

Directs FoxPlay to clear its Queue, if anything is in it. You will be prompted as to whether you really want to do this.

/clearcue

Same as **/clearqueue**

/desc filename

This alias allows you to display information on the specified file. You will "say" the file's name and any information your Playlist contains on it when you type this command in a channel, query window or DCC CHAT window.

!request filename [numtimes] [filename2 [numtimes] ...]

This works similarly to the **/queue** command (uses the same syntax), except that it is typed in a channel, or as a /msg to another user. This function allows you to "remotely" queue the specified files on the other person's system if he/she has them. The option **Accept Jukebox-Style Requests** controls whether or not a user responds to this command. If you type it in a channel, all FoxPlay users in the channel (except for yourself) will intercept and respond to this message.

!nickname filename

Sends a manual request for the specified filename from the specified user. If the person you're requesting the file from has the file, he/she will (usually) send it to you automatically. (Example: **!Dorsola enigma.mod**)

You can, of course, write your own aliases to do additional things with FoxPlay 98. If you're familiar with mIRC scripting, you should take a look at the **DDE Technical Reference** to find out how to communicate with FoxPlay, and to get a good grasp of how the interface works. Also check out the **mIRC Help File** – particularly the topics on writing aliases and using the **/dde** and **\$dde** commands.

Popup Menus in mIRC

In addition to text commands, FoxPlay also provides a set of **popup menus** to give you access to common functions and options, without having to switch back and forth between programs.

There are two popup menus. The first one is accessible from any channel or query window, and from the Status Window. Just right-click in any of these windows, then select **FoxPlay 98**, then any of the menu choices you see there. The top half of the main menu here allows you to play files, repeat the current file, stop the player and clear your queue, as well as starting FoxPlay when it's not running. The bottom half gives you access to your main Options.

The other popup menu is accessible from the nickname list in a channel window. From this menu, you can send a file to the currently-selected user, or you can choose one of the last five files this person has played and:

- Request the file
- Request information about the file (if you have it)
- Suggest your info about the file (if you have it)

In future releases of FoxPlay, these popup menus will have increased functionality and intelligence. For now, this is the extent of what they can do.

Requesting Files From Other Users

There are several ways that you can request files from other users. First of all, FoxPlay provides an **Auto-Request Feature**, which allows you to collect any files that people are playing and you don't already have. FoxPlay needs to be running in order for this feature to work.

You can also **Manually Request** a particular file from someone by:

- Typing **!nickname filename** in the channel or to that person as a private /msg. (For instance, **!Dorsola inferno.mod**)
- Selecting the file in the **Nickname-List Popup Menu** under the user's nickname, and then selecting "*Request this file*". (You can select from the last five files this person has played.)
- Of course, you can always just ask the person to send the file to you.

Any music files that people send to you will be placed automatically in your **Incoming Directory**, which is by default your mIRC "Get Directory". The script file takes care of this, so you don't necessarily need to have FoxPlay running at the time. However, in order to change your **Incoming Directory**, you will need to have FoxPlay running so you can modify that option via the **Options Screen**.

If the person who sent you the file is also running FoxPlay 98, he/she will automatically send you information about the file once the file transfer is completed. This information (which includes the proper name of the file if it has spaces in it) is stored in a file called *fp_new.fpi* in your **Incoming Directory**. This file contains information about all of the files waiting to be **Auditioned**, and can be edited via Notepad or any other text editor.

Sharing Playlist Information

FoxPlay provides a convenient Information Sharing system that lets you send and receive information about files that you and another user may have in common. Here are a couple of scenarios:

Say you and “Bob” both have *splinter.mid*. You see him play it and notice that it just comes up as:

```
<Bob> FoxPlay 98: splinter.mid -> 64.82 KB <-
```

You have the following information on the same file:

```
<Joe> FoxPlay 98: Splintering Heart (Marillion) -> 64.82 KB  
(06:52) <-
```

You can “suggest” your information to him by highlighting his name in the **nickname list** in the channel you two are in, right-click to bring up the **Nickname-List Popup Menu**, then select *FoxPlay 98*, then *splinter.mid*, then “*Suggest info from my copy*”. You will send your info to Bob. He will be presented a dialog telling him about your suggestion, and if he chooses to accept the suggestion, he’ll send you back a quick note thanking you for the suggestion. His Playlist will be updated accordingly, so next time he plays the file, his information will match yours.

This is a handy tool for helping other people form their playlists. However, if you know that Bob has most of the same files that you do, and he could use a lot of suggestions, you should send him a copy of your Playlist and have him **Import the info** from it. This will save you both a lot of time.

Okay... now, take the same scenario as I just described and reverse your roles. Bob now has the detailed information on *splinter.mid* and you don’t. When he plays the file, you can use the **Nicklist Popup Menu** to go “*Bob >> FoxPlay 98 >> splinter.mid >> Request info for my copy of this file*”. This will send an “information request” to Bob, and he will send back his info on the file same as if he had suggested it to you in the same manner as above. You will be presented with the “suggestion” dialog and given the opportunity to accept the info, and your Playlist will be updated if you accept it.

Note that you can only do this for files that both you and the other user have. The filenames have to match. If FoxPlay determines that the file you’re trying to suggest or request information about doesn’t exist in your Playlist, it will give you an error message.

Modifying your Options via mIRC

The **Popup Menu** allows you to modify some of your **Options** on the fly, without having to switch over to FoxPlay. These are the options that can be toggled on or off. When you pull up the **Popup Menu** (right-click in a channel, query or status window) and examine your options, you will see either **(ON)** or **(OFF)** next to each one. This indicates the current status of the specified option. To toggle the option, simply click on the menu item, and FoxPlay will update the option accordingly. (FoxPlay must be running in order for this to work.)

The options you can change this way are:

- **Random-Pick Files**
- **Local Info Display**
- **Remote-Control Mode**
- **Accept Jukebox-Style !requests**
- **Allow Playback** (MIDI, MOD, MP3)
- **Auto-Request** (MIDI, MOD, MP3)
- **Show to Public** (MIDI, MOD, MP3)
- **Broadcast in Color**

If you wish to change any other options, you will need to do so in FoxPlay itself. I plan on adding more of these option toggles to *foxplay.mrc* in future releases.

FoxPlay DDE Technical Reference

FoxPlay provides a system for communicating with **mIRC** and passing information back and forth between the two programs. This system is known as the **DDE Interface** (DDE stands for Dynamic Data Exchange). Currently, FoxPlay only supports **mIRC**, but it may support other IRC clients and/or other programs in the future.

If you wish to make modifications to *foxplay.mrc* or write your own script to interface with FoxPlay (perhaps an automated **!describe** command or something), you should read through this DDE Reference to find out how it works.

For information on using mIRC's DDE system, see the **mIRC Help File**, under **DDE Commands**.

A DDE Command (XTYP_POKE) is sent to FoxPlay via mIRC's **/dde** command, which uses the following syntax:

```
/dde foxplay <command_name> "" <data>
```

...where **<command_name>** is the name of the command to be sent to FoxPlay (see below), and **<data>** is the accompanying data, if any. The quotes between **<command_name>** and **<data>** are simply a filler, but it **MUST** be there so that mIRC can interpret the **/dde** command correctly.

A DDE Request (XTYP_REQUEST) is sent to FoxPlay via mIRC's **\$dde** identifier, which uses the following syntax:

```
$dde foxplay <request_name> <data>
```

or

```
$dde foxplay <request_name> ""
```

...where **<request_name>** is the name of the request being sent to FoxPlay (see below), and **<data>** is a single word of additional data. This identifier is usually used as part of a variable assignment or in the middle of another command, and the return value from this command is substituted in its place.

Note: The **\$dde** identifier **MUST** have three parameters, as shown above. No more, no less. If you need to send multiple words to FoxPlay (such as a filename that has spaces in it), replace the spaces in **<data>** with **ASCII value 127**. If you have no data to send (just the request by itself), you must insert the quotes in place of **<data>** so that mIRC can properly interpret the identifier.

In all cases where **<dde_name>** is specified, **<dde_name>** should match your current mIRC servicename. Your DDE server must also be turned on in order for FoxPlay to be able to communicate with mIRC. To retrieve your current DDE servicename, use **\$readini \$mircini dde servicename** – and to turn your DDE Server on, type

/ddeserver on [servicename] (where servicename is optional).

FoxPlay recognizes the following **commands**:

/dde FoxPlay FoxPlay "" <filename>

Directs FoxPlay to play the specified file. <filename> cannot be blank, but it can contain spaces.

/dde FoxPlay FoxPlay "" NULL

Directs FoxPlay to play the next file in the Queue, or a random file if the Queue is empty. Use this if you don't want to specify a filename to play.

/dde FoxPlay stop "" ""

Tells FoxPlay to stop playing, if it's playing something.

/dde FoxPlay UpdateOptions "" <dde_name>

Tells FoxPlay to send all of its current option settings to mIRC. This is useful for keeping the two programs in sync with each other.

/dde FoxPlay SetServiceName "" <dde_servicename>

Tells FoxPlay who it's supposed to be talking to. This is mainly for when FoxPlay attempts to broadcast something publicly, display information locally, and do other things automatically through mIRC. Use this only when you change your DDE servicename, or if communications between the two programs go awry.

/dde FoxPlay CTCPsOUND "" <dde_name> <nickname> <filename>

Use this to pass on information to FoxPlay about a CTCP SOUND event that occurs in a channel or to you personally via another user. <dde_name> must be mIRC's DDE service name (see above). FoxPlay will look at the information provided to it, compare the filename against your Playlist and your Options, and react accordingly.

/dde FoxPlay Queue "" <dde_name> <filename>

Tells FoxPlay to queue the specified file if it exists.

/dde FoxPlay QueueRepeat "" <dde_name> <filename> <numtimes>

Tells FoxPlay to add the specified file to the queue the specified number of times. Useful for looping.

/dde FoxPlay QueueDone "" <dde_name>

Use this at the end of a batch queueing operation, such as is done in *foxplay.mrc*, to provide a summary of the queueing operation or to do something else. To see the files that were queued in the entire operation, look at the variable **%fp_queueutemp** – it's recommended that you /unset this variable once you're finished.

When you send this command to FoxPlay, FoxPlay sends back the following

command to mIRC: **/fp_queuedone** – you need to modify or replace this alias in *foxplay.mrc* in order to make use of this command.

/dde FoxPlay ClearQueue “” “”

Tells FoxPlay to clear its queue.

/dde FoxPlay Suggestion “” <dde_name> <nickname> <FPL_Info>

Use this command to notify FoxPlay of a **suggestion** that someone’s sent you. <dde_name> needs to follow the same rules as described above. <FPL_Info> is a string in the following format:

filename \ Title \ Artist or description \ length in seconds

For example: splinter.mid \ Splintering Heart \ Marillion \ 412

At the very minimum, you must supply the file’s name.

FoxPlay will present you with a dialog box asking you if you want to accept the new information (if it can find the specified file in your Playlist). If you accept this info, FoxPlay will then update your Playlist and send a notice back to the user who sent you the suggestion, thanking him/her for the update.

/dde FoxPlay UserInfoReq “” <dde_name> <nickname> <filename>

Use this command to have FoxPlay send an automatic suggestion back to <nickname> when they request information about a particular file. If FoxPlay can find the specified file in your Playlist, it’ll send back the requested info as a suggestion.

/dde FoxPlay SetDirIncoming “” <path>

Use this command to set the **Incoming Directory** in FoxPlay. This command is used in *foxplay.mrc* to set the **Incoming Directory** to mIRC’s default “Get Directory” if a directory is not already set.

The path you send to FoxPlay must be a valid path on your hard drive. It must **not** contain a backslash “\” at the end.

/dde FoxPlay SetOption “” <dde_name> [noupdate] <option_name> <new_value>

Sets the specified option in FoxPlay. <new_value> can be either **0** or **1** (for OFF and ON, respectively). Normally, FoxPlay will automatically update all of its options in mIRC when you set one via this command. To prevent this from happening when setting a large number of options at once, specify **noupdate** between <dde_name> and <option_name>. Then either use **UpdateOptions** at the end or don’t use **noupdate** in the last **SetOption** command.

Valid <option_name> values are:

- **play midis** (Allow Playback)

- **play mods**
- **play mp3s**
- **areq midis** (Auto-Request)
- **areq mods**
- **areq mp3s**
- **show midis** (Show to Public)
- **show mods**
- **show mp3s**
- **localdisplay** (Show information locally)
- **randompick** (Pick files at random)
- **remotecontrol** (Remote-Control Mode)
- **acceptrequests** (Accept !requests)
- **showcolor** (Broadcast in color)

Example: **/dde FoxPlay SetOption "" show midis 1** – this tells FoxPlay to broadcast MIDI files as it plays them.

FoxPlay responds to the following **Requests**:

\$dde FoxPlay version ""

Returns FoxPlay's current version number.

\$dde FoxPlay reqInfo <filename>

Returns a black-and-white text string containing information on the specified file. If the file contains spaces, you must replace the spaces with ASCII value 127. The filename will be interpreted within FoxPlay.

For example: **Splintering Heart (Marillion) -> 64.82 KB (06:52) <-**

\$dde FoxPlay reqInfoColor <filename>

Works the same as above, except that it returns the text string with mIRC colors.

\$dde FoxPlay reqInfoFPL <filename>

Works the same as above, except that it returns an FPL-formatted line with the file's information. The line will be in this format:

filename \ Title \ Artist or description \ length in seconds

For example: **splinter.mid \ Splintering Heart \ Marillion \ 412**

\$dde FoxPlay getPath <filename>

Returns the full path of the specified file.

For example: **E:\media\MIDIs\q-s\splinter.mid**

\$dde FoxPlay NextInQueue ""

Returns the name of the next file in the Queue. If the Queue is empty, this request

returns **\$null**.

\$dde FoxPlay CurrentFile “”

Returns the name of the file that's currently playing, or was most recently played if it's not currently playing something. Returns **\$null** if FoxPlay hasn't played anything.

\$dde FoxPlay Activity “”

Returns a brief summary of what FoxPlay is currently doing.

Note: You can use this and/or the **version** request to find out in your script if FoxPlay is currently running. If you execute ANY of these **\$dde** requests while Foxplay is NOT running, you will always get **\$null** as a response, but **Activity** and **version** are the only two that will never return **\$null** otherwise. This way, if you get **\$null** as a response to either of these, you can automatically run FoxPlay or provide the user with the option of doing so. Take a look at the code for the main **Popup Menu** in *foxplay.mrc* to see how this functionality can be put to good use.

Frequently-Asked Questions

What is a MIDI file?

MIDI stands for **Musical Instrument Digital Interface**, which is basically an interface between your computer and a synthesizer. A **MIDI file** is simply a recording of MIDI events, and is commonly referred to as the computer's form of sheet music. It contains data that directs your computer's sound card (which has a form of a synthesizer on it) to play a certain set of notes with a certain set of instruments at certain times, with certain effects, etc.

MIDI files are typically relatively small, and tend to sound different on different people's computers, depending on what type of soundcard you have. I recommend a soundcard with a high-quality **Wavetable** on it, which gives you realistic-sounding music.

What is a MOD file?

A "MOD file" (which, as far as FoxPlay is concerned, is any file with a **.MOD**, **.S3M**, **.XM** or **.IT** extension) is a type of music file that has some things in common with MIDI files, but also quite a few differences.

A MOD file contains sequencing information similar to a MIDI, but it also contains the actual samples for the instruments as well, and it requires a special player such as **MODPlug**. Large and highly complex MOD files often also require a fast processor.

What is an MP3 file?

MP3 stands for **Mpeg Layer 3**, and is a very highly-compressed form of a WAV file. Depending on how it's encoded, a typical MP3 file can be as little as 5% the size of the WAV file it was encoded from! Yet it retains nearly all of the quality of the original recording.

MP3 files also require a special player, such as **WinAmp**, and also typically require a Pentium 100 or higher processor. Even though they're extremely compact, MP3s are still fairly large – a typical 5-minute MP3 file is usually around 5 megs in size, although this depends on how the thing was encoded.

Note: Be forewarned that it is considered illegal in many states and countries to distribute MP3 recordings of commercial music over the Internet.

I know FoxPlay 6.22 didn't work with mIRC 5.4. Is the same true of FoxPlay 98?

Nope. FoxPlay 98 is both forwards and backwards compatible with ALL versions of mIRC 5.1 and up. (mIRC versions below 5.1 lack some of the features that *foxplay.mrc* depends on.)

The main reason for such a big difference in compatibility is the fact that FoxPlay 98 is now a Windows program. All of the heavy processing and GUI components are contained within that program, eliminating the need to do any of it through mIRC.

I used to use FoxPlay 6.22 or below. What files do I need in order to use FoxPlay 98?

Actually, none of them. All of the files that came with FoxPlay 6.22 are now obsolete. The only file you should save is your Playlist. You can replace *foxplay.mrc* and delete the rest of the .mrc files, and you can also eliminate all of the .bmp and .txt files, since they are no longer needed.

Why, even though I have Random-Pick Files turned on, won't FoxPlay keep playing anything?

- You have Remote-Control Mode turned on as well, which will override **Random-Pick**.
- You are not allowing FoxPlay to play anything. Check your Allow Playback options.
- FoxPlay is still “playing” a MOD or MP3 file that doesn't have a length set. Until you tell FoxPlay how long this file is, it won't be able to automatically advance.

Hey! FoxPlay didn't shut down my MOD/MP3 player when I told it to! What gives?

I've noticed this happens sometimes with the newer versions of WinAmp, and occasionally with other programs. Unfortunately, there's not a whole lot I can do about this, so if you see this happen, just shut down the program yourself and it should be fine after that.

People keep kicking me out of their channels because I used mIRC colors! This happens right after I play a file.

You are broadcasting files in color. Keep in mind that you cannot see your own broadcasts, so unfortunately it's difficult to tell if you are broadcasting at all, much less in color.

You can turn **Broadcast in color** on and off. If you're getting kicked from channels because you're using mIRC colors, turn this option off. You might also consider turning off **Public Broadcasting** entirely, or setting your Target Channels to not include the one(s) you're having trouble with.

What happened to the ability to search and filter my Playlist?

That's a feature that didn't make it into the first release of FoxPlay 98. I plan on

adding this feature in a future release. And you can bet it's gonna be a good one.

What is the file *fp_new.fpi* in my Incoming Directory?

This is a file that FoxPlay creates and uses to store information on the music files in this directory. The **Auditioner** uses this file to display information in the **Playlist Editor**. It is recommended that you only delete this file when there are no music files in this directory, or if the directory you find this file in is no longer set as your **Incoming Directory**.

What is the file *fp_temp.cue* in my FoxPlay directory?

This is a temporary file that FoxPlay creates when you shut it down. It stores the contents of your Queue so that the next time you start up FoxPlay, it can restore your Queue to its previous state.

Why do you say not to load *foxplay.ini* in mIRC?

Simply put, because *foxplay.ini* is not a script file. It is the file that FoxPlay uses to store and load ALL of its option settings. If you load this in mIRC, you will confuse the heck out of both programs.

What do I do if I have a question you didn't cover here?

Simplest thing to do is to e-mail me. My e-mail address is dorsola@lightspeed.net. I will be more than happy to answer your questions as best as I can.

Credits

I hate to sound conceited, but there really isn't a whole lot to say here. I have been developing FoxPlay for close to two years (although the development of FoxPlay 98 took only about two months). I wrote it from the ground up and have learned a LOT in the process.

However, I want to thank StarFire for helping me get started with Windows programming, and with his contributions to the beginning of the FoxPlay 98 project. I regret that he wasn't able to see it through to completion, but his contributions were invaluable.

I also want to thank MLamer, FireFox, ^b0rg^, Majjk and gizmo^ for their support, suggestions, feedback and contributions to the FoxPlay project. And to the whole wacky gang in #musicfiles on EFNet, for your constant support and encouragement both for this project and for lots of other things. I seriously doubt I would have had a reason to keep going with this project had it not been for you guys.

Thanks also to **Khaled Mardam Bey**, the creator of **mIRC**, without whom my beginnings with PlayMIDI/FoxPlay would probably have been much less likely. And I have to give credit to Microsoft for writing a good compiler. Visual C++ is by far the most user-friendly system I've ever used for programming, particularly in Windows.

Finally, thanks to all the great musicians out there who composed the MIDIs and MODs I've collected during the course of FoxPlay's development. Your songs have helped to make FoxPlay possible. Keep it up!

FoxPlay requires what's referred to as a MOD player in order to play MODs, and an MP3 player to play MP3s. These are programs that can play these types of files.

I recommend **MODPlug** for playing MOD files, and **WinAmp** for playing MP3s. However, you can use any Windows program to play these files.

A **Custom Window** in mIRC is a user-defined window that can be used with a script. It will ALWAYS have a name beginning with an @ sign. (Example: **@MyWindow**).

See the section on Custom Windows in your **mIRC Help File** for more information.

CTCP stands for **C**lient-**T**o-**C**lient **P**rotocol. It provides a method of passing non-text messages between IRC clients. The CTCP SOUND message is a “standard” being used by many IRC clients to play music and sound effects over the IRC.

foxplay.mrc is a script file for mIRC 5.1 and up. When you load this file in your **mIRC Remote**, this script will enable you to use FoxPlay with mIRC to share your files and Playlist information with other people. This file replaces all of the script files associated with FoxPlay 6.22 and below.

See the **Remote** section of your **mIRC Help File** for more information.

IRC stands for **Internet Relay Chat**, and is one of the largest means by which people can talk to one another over the Internet.

This character can be typed directly in mIRC by holding down the **CTRL** key and hitting **Backspace**. It appears as a square in most fonts. You can also use **\$chr(127)** within an mIRC script.

To replace spaces with this character in a text line, use the following:

```
/set %tempspc = $chr(32) | /set %tempchar = $chr(127)  
/set %temp Text with spaces  
/set %temp2 $replace(%temp, %tempspc, %tempchar)
```