

freeMule - A User's Manual

David Roden <drozen@gmail.com>

\$Id: manual.tex 129 2006-02-18 18:18:18Z bombe \$

Contents

1	Introduction	2
1.1	The FreeNet Project	2
1.2	<i>freeMule</i>	3
1.3	Java	3
1.4	This Document	4
2	The User Interface	4
2.1	The Main Window	4
2.2	The Toolbar	5
2.2.1	The "Connect" Button	5
2.2.2	The "Disconnect" Button	5
2.2.3	The "Node" Button	5
2.2.4	The "Preferences" Button	5
2.3	The Node Dialog	5
2.4	The Preferences Dialog	6
2.4.1	The Identity Panel	6
2.4.2	The Shared Directories Panel	6
2.4.3	The Network Panel	7
2.5	The Download Panel	7
2.5.1	The "Clean Up" Button	7
2.5.2	The "Cancel" Button	8
2.5.3	The "Stop" Button	8
2.5.4	The "Resume" Button	8
2.5.5	The "Properties" Button	8
2.6	The Upload Panel	8
2.7	The Search Panel	8
2.7.1	The "Download" Button	9
2.7.2	The "Show Comments" Button	10
2.7.3	The "Show Filenames" Button	10
2.8	The Known Clients Panel	10
2.9	The Shared Files Panel	10
2.10	The Logging Panel	11
2.11	The Service Threads Panel	12
2.12	The Message List	12
2.13	The Download Key Textfield	13

<i>LIST OF FIGURES</i>	2
3 Frequently Asked Questions	13
3.1 Some of my files can not be hashed! Why?	13
4 Technical Information	13
4.1 Clients	13
4.1.1 Announcements	14
4.1.2 File Lists	14
4.2 File Download	14
4.2.1 Upload Requests	15
5 freeMule Development	15
6 License	15
6.1 <i>freeMule</i>	15
6.2 This Document	15
7 About the Author	16
7.1 Contact Information	16
7.2 Public Key	16

List of Figures

1 The Node Dialog	5
2 The Download Panel	7
3 The Search Panel	9
4 The Shared Files Panel	11
5 The Logging Panel	12

1 Introduction

This section describes what *freeMule* is about, and why it is what it is right now.

1.1 The FreeNet Project

The [FreeNet Project](#) has been around for quite some time now. *FreeNet*—in short—is a large, distributed, encrypted hard disk, spread over the internet.

Whenever a document is inserted into *FreeNet* it is spread over several *FreeNet* nodes¹, creating redundant copies. When a document is then requested it is further spread across the nodes in the request chain.

When a *FreeNet* node runs out of local disk space it forgets the oldest document² it has in its store. Thus not-requested content tends to “drop out” of *FreeNet* after a while. This is the only way content can “disappear” from *FreeNet*: there is no way to specifically delete a file.

Other features inherent in the design of *FreeNet* are “inserter anonymity” and “requester anonymity” (based on “plausible deniability”). Though it is possible to trace a request (be it insert or retrieval) to your node, nobody can

¹A *FreeNet* node is the program you run on your computer to access *FreeNet*.

²More exactly, the document that was not accessed for the longest time.

determine for sure that you inserted or requested the document in question. There is always the possibility that your node merely forwarded the request for some other node.

A disadvantage of *FreeNet*'s design is the impossibility to search content, or to create bidirectional real-time communication (comparable to a TCP connection).

1.2 *freeMule*

Because *FreeNet* can not be searched for files locating a file you want to download is pretty difficult. When using "The Freedom Engine"³ you have to search every freesite⁴ for a link to the file you are interested in. This can be a lengthy process and might not result in what you want. Also, as described earlier, files that are not requested tend to "drop out" of *FreeNet*. As a solution to these problems I came up with the idea of *freeMule*.

freeMule (as can already be deduced by its name) is a file-sharing application modelled after the popular eDonkey-network client *eMule*. The key in downloading a file with *eMule* is that as soon as you're downloading a file, you're also uploading it to other clients who are interested in the same file.

Obviously, *eMule* works with TCP (and UDP) connections, allowing bidirectional real-time communication between clients and servers. *freeMule* does not have this possibility so some mechanisms used in *eMule* had to be adapted to the "*FreeNet* way." Fortunately, *FreeNet* also offers some advantages over classical file-sharing: content has to be stored in *FreeNet* so as soon as a block of data is uploaded to *FreeNet* it is available to all users interested in that block at the same time and at no cost for the original inserter.

Searching for a file in *eMule* works by sending a query string to the server one is connected to. The server has all a list of files of all clients that are connected to it. As you don't connect to a server in *freeMule* the clients store their file lists in *FreeNet*. Other clients can then download these file lists and search them locally.

1.3 Java

Some people will always find something to argue: the most prominent point of concern of many people is the language a program was written in. No matter what language you write it in, there will *always* be people that ask the developer(s), "why wasn't <insert name of program> written in <insert name of language>?"

Well, I for myself chose Java for one reason alone: it is the language I am most proficient in. I know the language details, I know the libraries to a large extent, I think in a "Java way." That makes Java the preferred language for me when it comes to actually writing a program.

If you think *freeMule* should be written in C, C++, Modula, Pascal, Quick Basic, 6502 Assembler—stop bullshitting me! Go ahead and do it! The source is available—you have no excuse.

³Gateway website to find other websites in *FreeNet*

⁴A website in *FreeNet*

So, there still is my choice of Java 1.5 (or J2SE, Version 5 as [SUN](#) calls it) for *freeMule*. The answer is equally simple though not exactly the same. In the beginning I wanted to check out the new features of Java 1.5 (generics, enhanced for-loop, auto-(un)boxing, enumerations) in some “real-life application.” Soon I realized that I actually liked those extensions so I decided to stick to them and make *freeMule* a Java 1.5 application.

Sure, *freeMule* could be re-written (or rather adapted) not to use these new extensions. Most probably it wouldn't even take very much time, I think I could be done with that in one or two days. I also thought of creating a 1.4-branch of *freeMule* but for the moment I won't bother. If you desperately need a 1.4 version of *freeMule*—and you'd better have a good reason (“Java 1.5 sucks” is *not* a good reason, “there is no Java 1.5 VM for my Amiga yet” might be a one)—contact me (see [Contacting the Author](#)) and I'll see what I can do for you.

1.4 This Document

This document may contain terms and command line examples that are (more or less) specific to UNIX-like environments as development completely took place under the free operating system [Linux](#). Though I am aware that the majority of users will probably using a non-free, proprietary operating system from [Microsoft](#) I used terminology that most of these users do not have a clear concept of. Also, in some cases it's pretty hard to describe a simple procedure as importing a public key (see [Public Key](#)) when you do not really have a command line interface available.

It boils down to this: this document does not describe certain operating system specific procedures in detail; all given command lines are examples only. Also, I can not be held responsible for whatever you manage to break.

2 The User Interface

The internal code structure and mechanisms allow for any kind of interface. The only interface I bothered to implement so far is the Java Swing GUI you are presented with when *freeMule* starts.

I have tried to make the user interface as sensible and intuitive as possible but I am only a programmer, not a user interface designer. So, if the interface is flawed in some way, forgive me (and submit a patch :).

2.1 The Main Window

freeMule's main window is what you see when you start *freeMule*. In the upper half it contains the menu bar, a toolbar, and a tabbed panel. In the lower half you can see the messages window and the key download textfield.

The following sections will describe the functions in the menu and the toolbar and all panels that can be seen by selecting the corresponding tab from the tabbed panel.

2.2 The Toolbar

The toolbar has contains a couple of functions that can also be accessed from the menu. As those functions behave exactly the same way regardless from where they are executed no distinction between the source of the action is made.

2.2.1 The “Connect” Button

This button starts *freeMule*. More exactly, all the services are started.

2.2.2 The “Disconnect” Button

This button causes *freeMule* to stop. All running services are stopped which may take some time (e.g. the download service finishes all blocks it is currently requesting first).

2.2.3 The “Node” Button

Pops up the node dialog (see [The Node Dialog](#)). This button is only activated when *freeMule* is not running.

2.2.4 The “Preferences” Button

Pops up the preferences dialog (see [The Preferences Dialog](#)). This button is only activated when *freeMule* is not running.

2.3 The Node Dialog

This dialog lets you configure the address and port number of your local *FreeNet* node. The default for the node’s address is localhost, default for the port number is 8481. This is what *FreeNet* ships with so you only need to change this if your node is running on a different machine or if you configured *FreeNet* to use another port number for client connections.

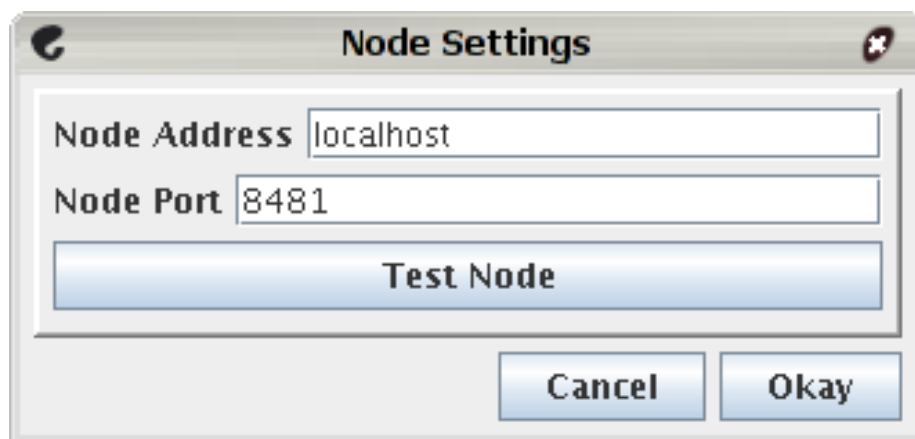


Figure 1: The Node Dialog

Note that if your node is running on a different machine you have to change the *FreeNet* config file to allow connections from your local machine!

The dialog also contains a “Test Node” button. This button will check if there is a *FreeNet* node reachable with the current settings. If there is a node, it’s name and version are shown. Otherwise you will see an error message.

2.4 The Preferences Dialog

In the preferences dialog you can change a couple of settings that are relevant to your local client.

The preferences dialog is split up into different panels. Also, it can only be shown in *freeMule* is currently not running.

2.4.1 The Identity Panel

The identity panel lets you choose a username that your client shows to other clients. In theory one could simply use the cryptographic public key of your client to identify it but a name chosen by the user can be more convenient in many places. The default name consists of random characters.

TODO: Insert Identity Panel Screenshot here

The public key is generated automatically on the first start of *freeMule*. You have no possibility to change it.⁵ The public key is also part of the address where your filelists are stored (see [Announcements](#)).

2.4.2 The Shared Directories Panel

In this panel you specify the temporary path for downloading files and the incoming directory (which is the directory completed downloads are moved to).

TODO: Insert Shared Directories Panel Screenshot here

But more important, you specify the directories you want to share in this panel. By pressing the “Add” button you can add new directories to share, by selecting a directory and pressing “Delete” you delete the directory from the list. Once a directory is listed in this panel, you can check the check box in front of the directory to specify if this directory should be scanned for files recursively⁶ or if you only want to share the files that are in this directory.

⁵As it is stored in plain text in the configuration file you are of course technically able to change it. But that is strongly discouraged.

⁶i.e. in this directory and all subdirectories

2.4.3 The Network Panel

This panel lets you adjust some network-specific settings, such as number of download and insert threads, and the HTL⁷ used by the different services.

TODO: Insert Network Panel Screenshot here

2.5 The Download Panel

This panel contains a table showing all files that are currently downloaded, including some information and a progress bar. When one or more downloads are selected, the buttons below the table are activated in correspondence with the number of downloads selected and their states.

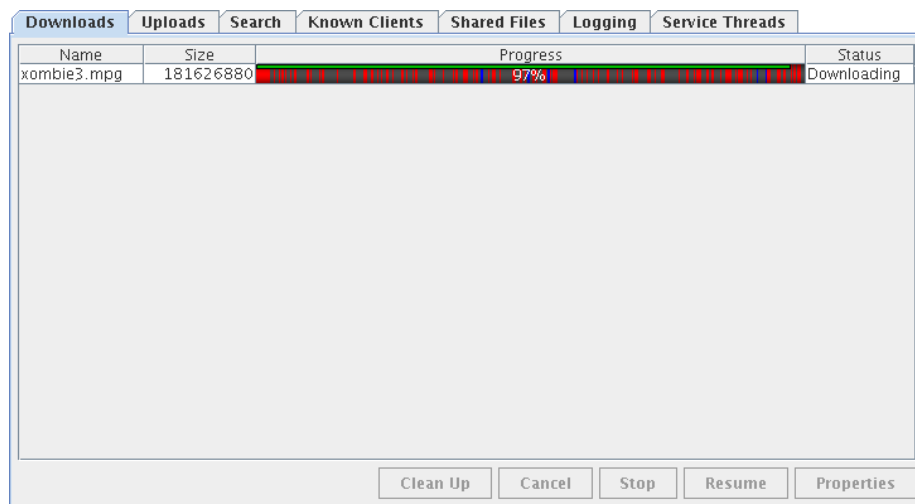


Figure 2: The Download Panel

By clicking the table with the right mouse button a context menu will pop up, giving you exactly the same possibilities as the buttons below the table do. For that reason only the buttons are explained below.

2.5.1 The “Clean Up” Button

This button removes downloads that are already finished. Those downloads are usually shown with an all-green progress bar and the status “Complete.”

This button is only activated when there actually are completed downloads in the list.

⁷Hops to live, i.e. how many nodes a request travels before it is allowed to finish. Note that this setting is not an absolute value as *FreeNet* uses random values in dependence of the HTL value to see if a request is forwarded further.

2.5.2 The “Cancel” Button

By pressing this button all selected downloads are cancelled. Cancelling a download means that all temporary files for it are deleted and the download immediately disappears from this list.

This button is only activated when one or more active downloads are in the download list.

2.5.3 The “Stop” Button

By pressing this button all selected downloads are stopped. Stopping a download means that no more blocks for that download are requested from *FreeNet* thus effectively halting the download for the time being.

This button is only activated when one or more downloads in the download list are active and running.

2.5.4 The “Resume” Button

This button is the opposite of the “Stop” button (see [The “Stop” button](#)). Pressing this button will resume a download that has been previously stopped.

This button is only activated when one or more downloads in the download list are currently stopped.

2.5.5 The “Properties” Button

Pressing this button will pop up a dialog window showing some properties of the currently selected download (like the filename in the temporary directory, some statistics on availability, etc.). There is not much you can do here. Press “Okay” to get rid of the window.

This button is only activated when exactly one download is selected.

2.6 The Upload Panel

This panel contains a table showing all files that are currently uploaded. It is determined at run-time which files are uploaded. *freeMule* scans *FreeNet* for upload requests (see [Upload Requests](#)) and inserts matching files on the fly.

TODO: Insert Upload Panel Screenshot here

Below the table you won’t find any buttons to control the uploads. Because downloading a file is not possible without anybody uploading the files in questions you are intentionally not given the possibility to cancel or pause an upload.

2.7 The Search Panel

The search panel is a big part of *freeMule*’s magic. Here you can search all the file lists your client collected.

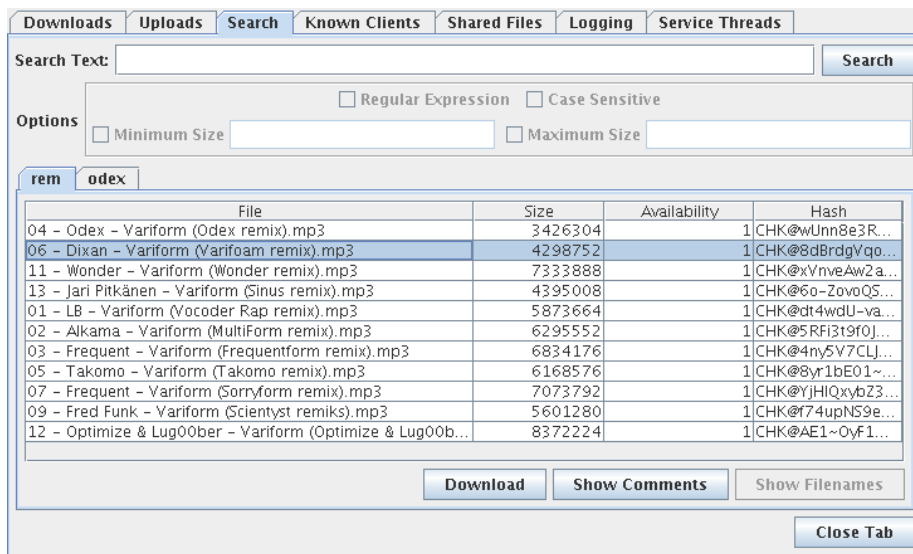


Figure 3: The Search Panel

The text you want to search for is entered in the box at the top of the panel. You can enhance your query by selecting some of the options below the box (though the options are currently disabled because I'm too lazy implementing them). Pressing the return key or the "Search" button will cause *freeMule* to start searching through all file lists it already has.

Two search results are open in the shown figure, one for the string "rem," the other one for the string "odex." In addition to the always present "Close Tab" button (which—surprise!—closes the active search result tab) you have three more buttons labelled "Download," "Show Comments," and "Show Filenames." These buttons are described below.

In the table you have several columns for the filename, the size of the file, the availability and the key.

The column for the size is the only one that needs absolutely no explanation.

The column for the filename lists only the first filename *freeMule* finds for the key. If more than one filename is available use the "Show Filenames" button to show them (see [The "Show Filenames" Button](#)).

The column labelled "Availability" tells you in how many file lists a file with this key was found, giving you a good estimate as to how many clients will eventually participate in helping you to get this file.

The column labelled "Key" shows the key of the file. There will be the possibility to copy the key to the system clipboard (to allow you to use another program for downloading)—somewhen in the future. :)

2.7.1 The "Download" Button

This button does almost exactly what it says: it adds the currently selected file (or files) to the list of download files. If a file is already being downloaded it is not added a second time, though.

2.7.2 The “Show Comments” Button

This button will only be activated if the currently selected file has one or more comments associated with it. On pressing a dialog will be opened that shows the client’s name, the rating, and the comment. Using this dialog it is quite easy to discover fakes⁸.

TODO: Insert Comments Dialog Screenshot here

2.7.3 The “Show Filenames” Button

This button will only be activated if the currently selected file has more than one source (i.e. its availability is greater than 1). On pressing it a dialog will show you how many clients prefer which name. Using this dialog can also help you in filtering out fakes.

TODO: Insert Filenames Dialog Screenshot here

2.8 The Known Clients Panel

The known clients panel contains a table listing all clients that *freeMule* was able to discover so far (see [Announcements](#)). For every client found its username, program and version number and the number of shared files will be shown.

TODO: Insert Known Clients Panel Screenshot here

Below the table you will find some statistics about the known clients.

Note that the number of shared files of a client changes whenever *freeMule* collects a new filelist for that client.

In the future I’m planing to implement a “Show all Files for Client” function but for now there is nothing you can do here.

2.9 The Shared Files Panel

This panel contains a table showing all the files you intend to share. Note that a shared file appears in this list only when its key has already been calculated, so on the first start of *freeMule* it might take a while for this table to get populated (file segmentation and key creation is a lengthy process as every *FreeNet* user can tell you). Unless all files in the shared directories have keys created for them no file list will be generated and uploaded. That means when you start *freeMule* for the first time depending on the number and size of files you want to share you will be listed in the “Known Clients Panel” (see [Known Clients Panel](#)) with 0 files.

⁸A “fake” is a file whose content is not matching the name.

Downloads	Uploads	Search	Known Clients	Shared Files	Logging	Service Threads
Filename						
01 - LB - Variform (Vocoder Rap remix).mp3	5873664	audio/mpeg	Com...	Rating	Key	...
02 - Alkama - Variform (MultiForm remix).mp3	6295552	audio/mpeg		Not rated	CHK@5RFi3t9...	0 0
03 - Frequent - Variform (Frequentform remix).mp3	6834176	audio/mpeg		Not rated	CHK@4ny5V7...	0 0
04 - Odex - Variform (Odex remix).mp3	3426304	audio/mpeg		Not rated	CHK@wUnn8e...	0 0
05 - Takomo - Variform (Takomo remix).mp3	6168576	audio/mpeg		Not rated	CHK@8yr1bE...	0 0
06 - Dixan - Variform (Varifoam remix).mp3	4298752	audio/mpeg	Great!	Excellent	CHK@8dBrdg...	0 0
07 - Frequent - Variform (Sorryform remix).mp3	7073792	audio/mpeg		Not rated	CHK@YjHIQxy...	0 0
08 - Pinza - Variform (Horn-y House Cut).mp3	7286784	audio/mpeg		Not rated	CHK@veM-t0...	0 0
09 - Fred Funk - Variform (Scientyst remiks).mp3	5601280	audio/mpeg		Not rated	CHK@74upN...	0 0
10 - Gargaj - Variform (Surface mix).mp3	6408192	audio/mpeg		Not rated	CHK@azryXaL...	0 0
11 - Wonder - Variform (Wonder remix).mp3	7333888	audio/mpeg		Not rated	CHK@xVnveA...	0 0
12 - Optimize & Lug00ber - Variform (Optimize & ...	8372224	audio/mpeg		Not rated	CHK@AE1-0y...	0 0
13 - Jari Pitkänen - Variform (Sinus remix).mp3	4395008	audio/mpeg		Not rated	CHK@6o-Zov...	0 0
14 - Mel Funktion & Little Bitchard - Variform (Beta)...	9103360	audio/mpeg		Not rated	CHK@TOJHOu...	0 0
15 - Mel Funktion & Little Bitchard - Variform (Instr...	9164800	audio/mpeg		Not rated	CHK@MXH5a...	0 0
Number of Files: 15, Number of Bytes: 97636352 Bytes (93.1 MiB)						

Figure 4: The Shared Files Panel

Below the table you will find some statistics about your shared files.

The only thing for you to do here is to change the rating or comment of a file. Double click a file and a dialog will pop up that lets you enter a comment and select a rating. Though you can also select a mime type here this is *only* recommended if the mime type could not be automatically determined from the file's extension.

On confirming the changes you made a new file list is created and uploaded, and a new announcement is inserted as well (see [File Lists](#)).

There's nothing more you can do here. If you want to change the list of shared files, either put some new files or directories into one of the directories you're sharing (the change will be picked up automatically), or you can stop *freeMule* and change your shared directories in the preferences.

2.10 The Logging Panel

The Logging Panel lets you adjust the logging properties of *freeMule*.⁹

The currently implemented logging mechanism knows two primary targets for log messages: the message window (see [The Message List](#)) and a logfile. As destination for the messages that go to a logfile the user can choose between *stdin* (the standard output, messages are printed to the console), *stdout* (the standard error channel), and a user-specified logfile.

The main purpose of this separation is that you don't want to clutter the message window with useless information but you might want to help the developer fixing some bugs. If the latter is the case the developer (that is me) will highly appreciate if you can supply a log file that shows all events that have been generated by the *freeMule* backend. Fixing bugs will be hell of a lot

⁹Logging is a property of the user interface; the *freeMule* backend does not know squat about logging.

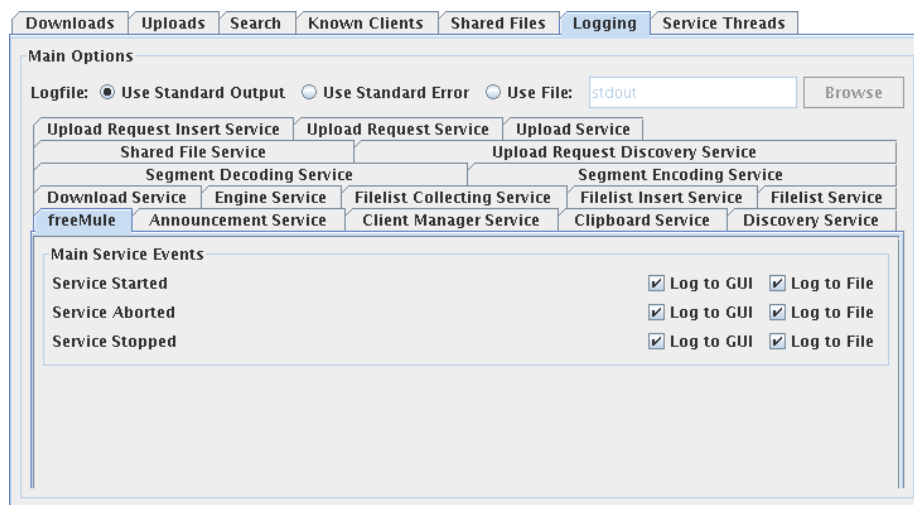


Figure 5: The Logging Panel

easier this way.

Anyway, for each service that the *freeMule* backend runs you can configure where the generated events go. You might want to keep the number of events shown in the message window down to a minimum to reduce scrolling on the screen (which is CPU-intensive). Also, some events don't necessarily have to be shown in the user interface at all.

In case you don't want to log the messages to a file it is not recommended to deactivate the file messages for all events. Instead, set the logfile to *stdout* so that all messages go the standard output. This way you can choose to ignore them (by running the application from your desktop or using a shortcut) or you can still have them available if you need them (by starting *freeMule* in a minimized console window).

2.11 The Service Threads Panel

This panel gives you an overview about the running services and what each of their threads are doing right now.

TODO: Insert Service Threads Panel Screenshot here

By unfolding a folder with a service name you will get a list of threads that service has spawned. Each thread will be listed with its name, its number, and what it is currently doing.

Again, this panel has only informational value so you can't do anything here.

2.12 The Message List

The message list window shows in text form what *freeMule* is currently doing. Every event generated by the *freeMule* backend may generate a line here,

depending on how you configured the logging (see [The Logging Panel](#)).

TODO: Insert Message List Screenshot here

By pressing the “Clear Messages” button you can clear this list.

2.13 The Download Key Textfield

freeMule can also be used to download splitfiles normally. Enter the key, optionally followed by a slash (‘/’) and a filename, into the download key textfield. Press the return key or the “Download Key” button to add this file to the download list.

TODO: Insert Download Key Textfield Screenshot here

3 Frequently Asked Questions

3.1 Some of my files can not be hashed! Why?

Most probably you do not have set the correct charset for your environment. If you are using special characters (like German umlauts) in your filenames the charset in your LC_CTYPE environment variable has to match the charset of the filenames in question. Try

```
# export LC_CTYPE=de_DE
```

before starting *freeMule*. Of course, adjust de_DE to whatever country you’re in and whatever regional settings might be required.

4 Technical Information

This section is aimed at anybody who wants to know a bit more about how *freeMule* works without actually reading (and trying to understand) the source code.¹⁰ It explains the techniques used in client announcement and discovery, as well as how files are up- and downloaded.

4.1 Clients

Because *FreeNet* lacks the ability for bidirectional communication *freeMule* has to follow a kind of P2P¹¹ approach. “A kind of” P2P approach because the clients also can not communicate directly. Instead, data has to be inserted into *FreeNet* by own client and download by other clients.

¹⁰I’d say that my source code is actually pretty clean and well-structured but other people’s source is *always* hard to understand, no matter what.

¹¹peer-to-peer, i.e. clients communicate directly with each other without some kind of server

In the design phase it has been taken into account that a request for a file that does not exist in *FreeNet* strains the *FreeNet* network much more than a request for a file that does exist (due to nodes having to forward the request to a number of other nodes until finally being able to conclude that the data is not there opposed to being able to abort a search prematurely due to having found the data). In theory the files a client inserts are requested by almost every other running *freeMule* client so that the data should spread very quickly through the *FreeNet* network, in turn allowing nodes to find it almost immediately. Of course that is only valid if a sufficient number of nodes are running (and the network is working well).

4.1.1 Announcements

When a client starts it knows nothing about other clients. In order to compensate the lack of a server telling the client about other clients, clients insert "Announcement Information." This information contains some client-related properties, like the name the user choose, the identity of the client and the latest version of it's file list.

Announcements are inserted to `KSK@freeMule/announcement/date/index`. *date* is a string containing the current time (with a 1-hour resolution), *index* is a running number, starting at 0. If a client can not insert its announcement at an *index* (due to a key collision) it increments *index* and tries again, until the announcement succeeded. This process is repeated every hour so that a client can assume that clients that haven't been heard from (by announcement) for the last two hours have gone away.

4.1.2 File Lists

File lists are inserted by the clients to enable other clients to search for file. File lists contain the local filename, the *FreeNet* key (without the "freenet:" prefix), the size, the number of blocks, and the user-given rating and comment.

File lists are inserted to `SSK@private key/freeMule/filelist/index`. Insertion to an SSK key ensures that the file lists can not be manipulated by other clients (or malicious parties). *index* is incremented every time a client updates its list of shared files (by adding or removing files or directories, or by adding or changing a rating or comment of a shared file).

When a file list has been inserted, new announcement information containing the now updated *index* will be inserted. Thus, by continuously scanning for announcement information by other clients a client can keep up with changes in file lists of other clients.

4.2 File Download

Downloading a file can be accomplished in two ways: by using the Download Key Textfield (see [The Download Key Textfield](#)) or by selecting a file in the search results panel and clicking the "Download" button. Both actions will add the file to the list of download files and it will appear in the download panel.

For files added via the textfield the size of the file is unknown (resulting in a displayed size of -1). When a file is added to the list of download files its “splitfile manifest”¹² is requested from *FreeNet*.

As soon as the splitfile manifest has been downloaded, *freeMule* starts downloading random blocks for the file. *freeMule* assembles segments¹³ as soon as there are enough blocks to do so. Finally, when all segments have been decoded the final file is assembled in the temporary directory and moved to the incoming directory.

4.2.1 Upload Requests

Upload requests are inserted into *FreeNet* by a client who is downloading a file. When a client receives an upload request and has the asked for file in its list of shared files it starts uploading the file.

Upload requests are inserted in about the same way as announcements (see [Announcements](#)), i.e. all upload requests (one for each downloading file) are collected and inserted once an hour to `KSK@freeMule/uploadrequest/date/index`. Also, the upload requests of a client are updated when the client downloads a new file or finishes downloading a file.

5 *freeMule* Development

Development of *freeMule* has started in May 2005. Since then I have spent quite some time implementing, designing, and documenting. Currently *freeMule* is developed by me only: David ‘Bombe’ Roden (see [The Author](#)). I originally had the idea and wrote all of the code.

If you would like to participate in taking *freeMule* further you should get in contact with me (see [Contacting the Author](#)).

Also, if you find any bugs (or if you can even create a patch fixing that bug) it would be best to contact me as soon as possible.

6 License

6.1 *freeMule*

freeMule has been placed by its author under the GNU Public License, Version 2 (or any later version, if you prefer).

6.2 This Document

This manual has been placed by its author under the GNU Free Document License, Version 1.2 (or any later version, if you prefer).

¹²“Splitfile” is the name established by *FreeNet* for files that are inserted with error-correcting information and “split” up into blocks; the “splitfile manifest” contains the keys for all blocks and information about how to assemble the file, given enough blocks can be downloaded.

¹³Files larger than 1 mebibyte (1 mebibyte = 2^{20} bytes = 1048576 bytes) are split up into segments of 128 mebibytes; each segment is inserted with 50% additional check blocks for redundancy, and each segment can be assembled independent of other segments.

7 About the Author

David Roden is currently 29 years old and lives in a small village some dozens kilometers away from Hannover, Germany. Due to an illness he is currently quite incapable of working for someone who will actually pay him so he spends a lot of free time at home, coding away at will. One of the results is this program: *freeMule*.

Having started programming at about the age of 9, David has quite some experiences in solving problems on different platforms. His first computer was an ATARI 130XE, an old 8-bit machine which was later exchanged for an ATARI 1040STE on which the first steps in programming a high-level language (in the case C) were taken. Even later a 386SX was bought, and Pascal was learnt. After school Java caught David's interest and has kept it till today.

7.1 Contact Information

The best way to contact the author is via email: it is fast, pretty reliable nowadays, can be encrypted and offers long-term storage on the client side. Though there are other possibilities to contact the author, email is the preferred way.

Email	droden@gmail.com
ICQ	3017094
IRC	Bombe (IRCNet)

7.2 Public Key

If you choose to send an email to the author, encrypted emails are preferred. Copy the following public key and import it into your public keyring. Under UNIX-compatible systems (Linux, Mac OS X) with GnuPG (a free PGP clone) use the following command line to import a public key:

```
# gpg --import
```

Now paste the copied text and press the EOF character (usually Control-D). GnuPG should happily import the public key and say so afterwards. Read the manual of your key-management software to get more information.

As soon as the public key was imported into your keyring you can use your mail client to write encrypted mails (given a supporting mail client).

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.2 (GNU/Linux)
```

```
mQGIBDz60EMRBACAr6VJyfDrmoZndreQ09SDftTX6TtTcwSCCbhWnyTeeKeMdTmI
syuk6pNe8ZgmqWfFa2/t73ZGHt7ZjdWuYWyYYzq/wfTeLkVEDTAQ/id7wnL5z9y
6CeN2WVwKTDuI6qIyIicy6jOB/tooaezn+3+JR1W89nEovg3IVLJp6w97wCgxA6y
w5+NjA2rSjGkVA3COyKAFjUD/A91NEcLypAEdhDM/Zz1GJwLNE9RuUEIu+z0+uQ
i04pr63xRqYY+NFKU35ts0Xb5AqZNDcVCBPKmBdwC/h6gqUY0sP90vzybo2m0G4y
bWz8LlV4ryUvRb2ciKEC12cfxJZITVc9UK3BEJMY7Px0xXSYEoncJ52gaGYinrBT
iwB1A/95Ej0M1E/MWAMCBLsBf4xDAX/0KQp1mMV10B8CtyyWLBPRrtXXWgFMQ1Mh
```


S1/VTvHj6k74DHWPJUauN+aP7+9gNxW2BFHQDTL.fTIwaOU4Hrj07GtwFBL1x4BX2
I7VzJ9hdArIalH01naBz6xsp6GjlfXsquBkFpW9aDvWiuLKGPbQwRGF2aWQgJ0Jv
bWJlJyBSb2R1biA8Ym9tYmVAdG9kZXNiYXVtLmR5bmRucy5vcmc+iGIEExECACIC
GwMHCwkIBwMCAQMVAgMDFgIBAh4BAheABQJCAtmBAhkBAAoJELIfB4KeU8JD0BwA
n1AHiv0knvblfDMem4EFg+EQcQe6AJ4/PQ1FiIcwwdr6nUPtES/F87DpjELQoRGF2
aWQgJ0JvbWJlJyBSb2R1biA8Ym9tYmVAdG9kZXNiYXVtLmR1PohZBBMRagAZBASh
AwIDFQIDaxYCAQIEAQIXgAUCQgLGzQAKCRCyHweCnlPCQzGfAJ0fedYbTyR0cDSh
6xcU2AUke6v1DACfQPufpEryy/oQAg1kudnWl8htUpGIRgQTEQIABgUCQGMqAAK
CRBJT5bKMa4V4gh0AJ9ORWLI29okpe2M3LtnftdcguuxgCftjBQwkVsJlFfoVFP
N47CWlQcb0+IXAQTEQIAHAQLBwMCAxUCAwMwAgEChgECF4ACGQEFajz60EQACgkQ
sh8Hgp5TwkOebACfZpEx0Rf6SJCfaaWeeiqlz4hZX08AoJUCzInayhtYvlorG8xs
WHkGk5xbtDsuZHIgLyAuZmFyYnJhdXNjaCAuY29uc3VtZXIgLmNvbnN1bHRpbmcg
PGRyQGZhcmltcmF1c2NoLmR1PohcBBMRagAcBQI8+t2DAhsDBAShAwIDFQIDaxY
AQIEAQIXgAAKRCyHweCnlPCQ6TKAKCP0oh1fUR26fcJ45sy7TmrWj/2dQCfYAu5
7fnwdfKWAHLDemTx7EakBbWIRgQTEQIABgUCQGMqWAKCRBJT5bKMa4V4gJRAKCG
vRtqBXbD80CpZT0vf6xhxh8+gQCgndhe4vXhz7feUz1T70+5KRUsity0JUJvbWJl
IC8gVGVrbG9yZHMgPGJvbWJlQHRla2xvcMzLm9yZz6IXAQTEQIAHAUCPPzBbgIb
AwQLBwMCAxUCAwMwAgEChgECF4AACgkQsh8Hgp5TwkMx4ACgsfv4Qs2h+FCB3Ht0
cUWwQMndQMAAoL/bGluEl+YXs5ZBnwHgDS203GLdiEYEEExECAAYFAkBgDKsACgkQ
SU+WYjGuFeIqTgCgqkFLf9bpd+Y2J8VZGQWR6jX9eAsAn2HosRpecjS/XS6SnmAM
Nb+MO/5itB1EYXZpZCBSb2R1biA8ZHJAdG9kZXNiYXVtLmR1PohcBBMRagAcBQI9
b6hHAhsDBAShAwIDFQIDaxYCAQIEAQIXgAAKRCyHweCnlPCQ6bGAKCi2vSSI+GG
5N9nNU9XK5cUAvmK5QCePdZFdThhF3DemoF5YaeKuCp98LSIRgQTEQIABgUCQGM
qWAKCRBJT5bKMa4V4qBJAJ9xZiTI7yiI66h/N+FrMG/LF5BcbgCghunwNvq1bWfo
F1Wwcmz3iMK1WYC0HkRhdmlkIFJvZGVuIDxkcm9kZW5AZ21haWwY29tPohgBBMR
AgAgBQJDX0ieAhsDBgsJCAcDagQVAggDBBYCAwEChgECF4AACgkQsh8Hgp5TwkOs
ZQCcCm66Pr6L2c3w+7ay2SiMZ8sAnkAAAn2RKfbSv43LujrWDakdoUzwGYV/3uQIN
BDz60GEQCACUROQiefme1EMqTc4DJNfluYvZFGdmydnoBgs95RSH1dtz2/lb/C1w
muKDZgqjoNcQqnQMXzffAhmnwWQ9rXR7UTj15j8T/8kj1rVbFKN3dLjkXeT9gBM
vZ6qYrQLAaGV6b0eobqoNvv9s0zQkokeSRgjEtyc1Jgxm0ZDiExQnt8oU+XFxK3i
SBtWmMEJFbmuSm2HeHCabjkj53Kff4RTmxFkNz+2l98GVrg1Xl5ZmMpqU6w3p2n8
E5inz05NXDwpMP1998A4Fkk0MJUzAIatJs0/NJjzCBQ074V4ECxnSmdbfVRYz635
k+7jdoACvoqpPLLKOD4bYSRbOVZa6mtfAAMFB/90N7viuwpXPqCCDyajwnNFAFTq
+4A5hp978+0FpV3crBRYkASEtKq1BTPVnyVdzM3001u1FGiTMkp+M7zmZ50RuVS9
oW4mEMvQYh2AKX1UKoN5eozB5+9a+dZFW4F37GkQxocfheFKAnrKouqTZNC2AO2N
DXn4yvBy9TaW1+3XI9sUd/A03hyh448DqaKLAQoVR+4USuJrqdsW2TBMVP70cv/B
ittK+fIybbRyAvxxUkgJXBrK2PC+fRl00pisjcG0deX1/ABwaB0spypdq9SzwM2
SXqQJirbRRPLLVv+F2G25/9d8uUhEn8amJtd07QdWfJW1di177gmbGNuFb7KiEYE
GBECAAYFAjz60GEACgkQsh8Hgp5TwkN91QCeJbrdy+8GB5ME1bWcZMQGittJwTo8A
nRYL1RLH9ccSvCvd2o1B5IRci7Wn

=d7U7

-----END PGP PUBLIC KEY BLOCK-----

Index

- Announcement, 12
- Announcemnt, 9
- Anonymity, 1

- Cancel, 6
- Clean Up, 6
- Client, 12
- Comment, 9
- Comments, 8
- Connect, 3

- Disconnect, 4
- Download, 8, 13
- Download Key Textfield, 11, 13
- Download Panel, 6

- Fake Files, 8, 9
- File List, 9, 13
- File Lists, 5, 9
- FreeNet, 1, 12

- Identity, 5

- Java, 2

- Known Clients Panel, 9

- License, 14
- Linux, 3
- Logging Panel, 10

- Main Window, 3
- Menu, 3
- Message List, 11

- Network Settings, 5
- Node, 1, 4

- Preferences, 4
- Preferences Dialog, 4
- Properties, 7
- Public Key, 3, 15
 - importing, 15

- Rating, 8, 9
- Resume, 7

- Search Panel, 7
- Service, 3

- Service Threads Panel, 11
- Shared Directories, 5
- Shared Files Panel, 9
- Show Comments, 8
- Show Filenames, 9
- Splitfile, 11, 13
- stdin, 10
- stdout, 10, 11
- Stop, 6

- Toolbar, 3

- UNIX, 3
- Upload Panel, 7
- Upload Request, 7, 14
- User Interface, 3