

LimeWire Peer to Peer File Sharing Software

Researched by FE Thad Winkelman

Assisted by FE Dean Brown, FE Mark Johnson,
and FE Shawn Healy

Heart of America Regional Computer Forensic Laboratory

This paper was written about the LimeWire File sharing program, on version 4.10.9 using Java Platform 5.0 update 6. I have not tested prior versions of the software but I believe that they all act similar, due to the observations of the previous version in a live case. The file structure was the same and the forensically valuable files contained the same information and was formatted the same. I have agreed to release this research to the IACIS forensic community at the bequest of CFCE David Melvin, ICE. If additional research is added I encourage others to add to and amend this as needed and ask them to send me a copy so I can amend mine. Believe me the offenders are sharing information on how we operate and we will be behind the curve if we don't share.

HISTORY

LimeWire

From Wikipedia, the free encyclopedia

LimeWire is a free and open source peer-to-peer file sharing client for the Gnutella network. It is released under the GNU General Public License. The program allows users to share files using the Gnutella peer-to-peer protocol. It was the first file sharing program to support firewall-to-firewall file transfers, a feature introduced in version 4.2, which was released in November 2004.

LimeWire is written in Java and hence runs on any computer with the Java virtual machine installed. To facilitate installation for casual users, the developers release installation packages for Microsoft Windows, Mac OS X, and for Linux, in RPM format. Classic Mac OS (Mac OS 9 and before) support has been dropped with LimeWire 4.0.10.

LimeWire uses the SHA-1 and Tiger tree hash cryptographically secure hash functions to ensure that downloaded data is uncompromised. Although researchers have identified possible vulnerabilities in the SHA1 algorithm, because LimeWire does not rely on SHA1 alone these vulnerabilities do not have many adverse implications for LimeWire's verification of downloaded files.

The Windows version of LimeWire installer includes a stripped-down version of Sun's Java installer which will download and install version 1.5 of the Java Runtime Environment (JRE) if it detects the machine doesn't have Java, or has a version of Java below version 1.4.1.

Lime Wire LLC, the developer of LimeWire, distributes two versions of the program; a basic, free version, and an enhanced version sold for a small fee, which the developers claim offers faster downloads. Prior to April 2004, the free version of LimeWire was distributed with a bundled program called "LimeShop" (a variant of TopMoxie), which was considered by computer security experts to be spyware. Among other things, LimeShop monitored online purchases in order to redirect sales commissions to LimeWire LLC. Uninstallation of LimeWire would not remove LimeShop. With the

removal of all bundled software in LimeWire 4.0 (released May 14, 2004), these objections were addressed.

Being open source, LimeWire has spawned several forks, including LionShare, an experimental software development project at Penn State University, FrostWire, and Acquisition, a popular Macintosh-based Gnutella client with a proprietary interface.

Researchers at Cornell University developed a reputation management add-in called Credence that allows users to distinguish between "genuine" and "suspect" files before downloading them.

According to a 28 June 2005 report in *The New York Times*, LimeWire LLC may stop distributing LimeWire due to the outcome of *MGM v. Grokster*. However, new versions are being released (4.9 released on August 26) with smarter search results, optimized downloads and other features. On September 25, it was reported that Lime Wire LLC was working on a version of the program which will refuse to share files that lack valid license information.

An October 12 2005 report states that some of the LimeWire open source contributors have forked the project and called it FrostWire. The FrostWire project has a beta release, which is claimed to be equivalent in power to the nonfree version of Limewire. The FrostWire developers emphasize that they will never place any sharing-related restrictions on the client.

General Installation Information

When fully installed, LimeWire installation program builds folders for its' use. It built a folder to store incomplete download files and populated the LimeWire folders. It created a shared folder named **C:\Documents and Settings\[User Profile]\Shared** and another folder was installed to handle partial downloads, **C:\Documents and Settings\[User Profile]\INCOMPLETE**. LimeWire marks these folders by default to share files contained inside of them. It will share partially downloaded files as well as downloaded files. Some of these files have forensic value.

Determining File Sharing on a LIVE SYSTEM

To find out which files are being shared there are many options. One of the ways to determine the shared files on a live system is from within the program. This can be accomplished by mounting the user's hard drive (or an image of the hard drive) in a virtual machine or by restoring the drive and booting it in the suspect's machine. This is a time consuming means but effective, although information is circulating that these options will soon be available directly from within FTK 2.0 or EnCase.

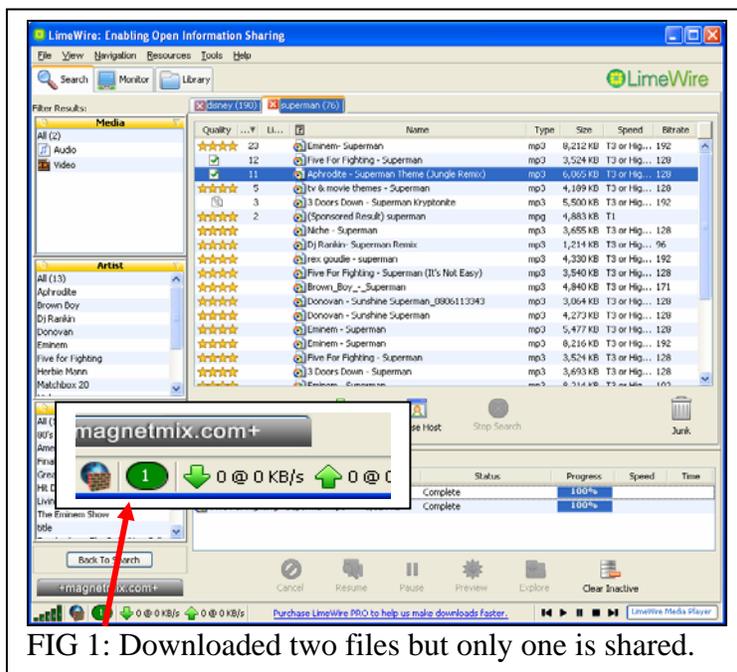


FIG 1: Downloaded two files but only one is shared.

Once in the program, the library tab, it will show the folders and the individual files shared.

The program has an indicator in the bottom left side of the status bar which shows a number surrounded by a green circle that lets the user know how many files they are sharing.

In this example, two files were downloaded using LimeWire. One file was downloaded to a shared directory and the other was downloaded to a non-shared

directory with the setting “do not share downloaded files” selected. The result was that only the file in the shared folder was shared. When the option was changed to include all downloaded files the count remained one which indicated that past downloads were not shared when this option is invoked. When another file was downloaded to the previously tested non-shared folder, the file count changed to two. The existing file in that non-shared folder was not automatically shared (due to the entry in the LIBRARY.DAT file. See the section on LIBRARY.DAT files).

Determining File Sharing on a drive (Forensically)

Another, more forensically sound place to determine file sharing that is by examining the LimeWire configuration files. By using the configuration files LIMEWIRE.PROPS, LIBRARY.DAT, and the FILEURNS.CACHE, the files available to be shared using the Gnutella Network can be identified for THE LAST TIME THE PROGRAM WAS ACTIVE. This is important to note because, if the files were shared the last time the LimeWire program was started but have since been deleted, and the LimeWire program was not restarted, file names (and sha1 values) may appear that are no longer on the system.

Using the Sysinternals (<http://www.sysinternals.com>) program Filemon, it was discovered that when files are unshared (or the reverse) in a directory listed in the LIMEWIRE.PROPS file as a shared directory, LimeWire writes to a file called LIBRARY.DAT. The LIBRARY.DAT file lists the exceptions to the blanket sharing or non-sharing of a directory under a header that is easily interpreted. The file is identified by name and path in this file. It is what the program uses to generate the exceptions to LIMEWIRE.PROPS shared directory list.

Another phenomenon was observed during the testing. Audio files appear and disappear in the audio.xml folder when shared and unshared. When items were shared and unshared the name and SHA1 value of the file would appear and disappear in the *.SXML files. These files are named by file type (i.e. audio.xml, video.xml). (A note of caution to users LimeWire reports the sha1 value in base32 while FTK reports it in base16. Senior FE David Loveall, FBI (HARCFL) wrote a quick program in Python to convert the values.)

The LimeWire program also adds the letters “sr” to the end of the sha1 for the first entry in *.SXML file and then adds the letter “q” to the remaining. At the time of this writing the significance of these letters being added was not discovered. The speculation was that the “sr” designation may be the programmer’s notation to start the record. I have asked the assistance from other examiners from the lab to help discern the significance of these designations. Although it should be noted that the *.SXML phenomenon did not reappear with later java updates installed (update 6) and a later version of LimeWire.

The reliable conclusion discovered was that all of the files in the folders listed in the LIMEWIRE.PROPS file after the text: `DIRECTORIES_TO_SEARCH_FOR_FILES` are available for sharing to other Gnutella users when the program is running and connected to a network (the internet) *unless* noted as an exception in the LIBRARY.DAT file.

To determine what files were available for sharing the last time the program was started, the examiner should look at the FILEURNS.CACHE file. The FILEURNS.CACHE file populates each time the program starts. It will contain the actual files that were shared the last time the LimeWire program was active. Items listed in this file are listed by name and SHA1. They can be compared to the files discovered on the system (including renamed or recovered deleted files) to confirm the contents. These files will respond to queries from other Gnutella users.

Search Terms

With the assistance of FEs Mark Johnson, Dean Brown, and Shawn Healy research was conducted of the live memory using the SPADA Linux Boot CD (v3.0). The research yielded that although the search terms were located in plain text in the live memory, the user searches from the local user could not be distinguished from external queries from other users when the user was connected as an ultra peer.

Further search of the PAGEFILE.SYS file showed that if the terms are cached from memory they will appear in the paging file (pagefile.sys). The first term was headed with the label SEARCH TERMS followed by the search term with a “:urn” following directly. In hex view we observed a 10 character string that was consistent before each search term began. 00h 80h 00h 06h [variable] 00h 00h 00h [variable] 00h was the hex pattern discovered to be consistent prior to the search terms. It was discovered that when the system is changed the second character is also changed but consistent with that search. We found it easier to search by the “:URN” and then further identifying them by the 10 hex character set. By searching this pattern we were able to call up the search terms in the HIBERFIL.SYS and PAGEFILE.SYS files. Research is ongoing to determine if there is a way requests entered by the user can be identified. Senior Forensic Examiner David Loveall reviewed the Java programming and reported that the program does not make a designation between the internal and external searches. This means that if the subject was an ultra peer or received queries then these terms cannot be relied on as user entered searches by the user of THAT computer.

After the latest Java update (update 6) the signature changed. Further research is needed to distinguish incoming search terms from inquiry terms.

FILE BY FILE

The following text contains the observations about the individual LimeWire configuration files by their file names.

INSTALLATION.PROPS

```
#LimeWire installs file
#Sat Mar 25 09:12:00 CST 2006
SAVE_DIRECTORY=true
FIREWALL_WARNING=true
SPEED=true
LAST_EXPIRE_TIME=1143299482781
START_STARTUP=true
LANGUAGE_CHOICE=true
SCAN_FILES=true
```

FIG 3 Contents of the installation.props file.

The **INSTALLATION.PROPS** file appears to be of little use other than to determine the last user settings. It may shed light on the last use as LimeWire’s installation date and time change in this file every time the program starts.

It does not display the name of a save directory but indicates that one is selected with the entry **SAVE_DIRECTORY=true**. It also notes the following:

- the presence of a firewall (and assumed lack) of
- the setting to auto-start the program

- the scanning of the files for additional media shares.

LIMEWIRE.PROPS

The second file examined was the **LIMEWIRE.PROPS** file. My original assumption

```
#LimeWire properties file
#Sat Mar 25 09:12:00 CST 2006
DIRECTORIES_TO_SEARCH_FOR_FILES=C:\\Documents and Settings\\Demo\\Shared
LAST_EXPIRE_TIME=1143299481546
DIRECTORY_FOR_SAVING_FILES=C:\\Documents and Settings\\Demo\\Shared
COUNTRY=
CLIENT_ID=5ED721F802436C59677269F067679D00
PORT=13674
INSTALLED=true
MAX_SIM_DOWNLOAD=8
CONNECTION_SPEED=350
```

FIG 4 – Contents of the LIMEWIRE.PROPS file with sharing.

```
#LimeWire properties file
#Sat Mar 25 10:46:26 CST 2006
FILTER_HASH_QUERIES=true
LAST_EXPIRE_TIME=1143299481546
LAST_ACCEPTABLE_BUG_VERSION=4.10.10
DIRECTORY_FOR_SAVING_FILES=C:\\Documents and Settings\\Demo\\Shared
UPDATE_DOWNLOAD_DELAY=10000000
COUNTRY=
MIN_CONNECT_TIME=7
CLIENT_ID=5ED721F802436C59677269F067679D00
SHARE_DOWNLOADED_FILES_IN_NON_SHARED_DIRECTORIES=false
IDLE_CONNECTIONS=2
TOTAL_UPTIME=1509
PORT=13674
AVERAGE_UPTIME=1509
UPDATE_GIVEUP_FACTOR=24
UNSET_FIREWALLED_FROM_CONNECTBACK=true
FLUSH_DELAY_TIME=256
LAST_FWT_STATE=true
INSTALLED=true
MAX_SIM_DOWNLOAD=8
LAST_GWEBCACHE_FETCH_TIME=1143304879109
UPDATE_DELAY=252000020
CONNECTION_SPEED=350
```

FIG 5 - LIMEWIRE.PROPS file with sharing disabled.

based on text postings was that this folder contained a listing of the location of the shared folder, notice if share was disabled or not, creation time and date of the file, and the indication of local or global sharing. The settings shown in FIG4 are the default installation. Disabling the file sharing adds several lines of text to the file.

Note the line in FIG 5. **SHARE_DOWNLOADED_FILES_IN_NON_SHARED_DIRECTORIES=false**, this is the only reference to sharing (when sharing is enabled) that I found. Seeing this entry tells the examiner that the user has to disabled the option that all downloaded items are to be shared regardless of their location. The absence of this line means it is enabled (there is only a statement that shows what is turned off not what is turned on). All files within the shared folder are shared regardless of this setting (for individual file information see the section on library.dat). Individually shared files can be enabled/disabled in the library tab of the user interface. By alternatively clicking on the file in the viewer window the user can opt out of sharing individual files.

Using the Sysinternals (<http://www.sysinternals.com>) program Filemon, I was able to determine that when files are unshared in a directory listed in the LimeWire.props file as a shared directory writes to a file called library.dat. The library.dat file lists the exceptions under the appropriate headers header. The same holds true for files shared in non-shared directories. **BY DEFAULT ALL SHARING IS ON.**

The absence of the line regarding sharing the files downloaded to non shared directories and the presence of text indicating a path to “search for files”

(DIRECTORIES_TO_SEARCH_FOR_FILES=C:\\Documents and Settings\\Demo\\Shared) is the key to determining if there is a shared folder (default). If additional directories are shared they will appear in this line.

The lack of this line in FIG 5, combined with the above entry for individual file share, indicates that file sharing in FIG 5 is completely turned off.

To validate and illustrate the impact of these observations, I enabled file sharing again and experimented with different configurations within the program then inspected this file.

Another place that is more accessible without mounting the drive is using the LimeWire.props file and the library.dat file. Using the Sysinternals (<http://www.sysinternals.com>) program Filemon, I was able to determine that when files are unshared in a directory listed in the LimeWire.props file as a shared directory writes to a file called library.dat. The library.dat file lists the exceptions to the blanket sharing or non-sharing of a directory under the appropriate header. The file is identified by name and path in this file. It is what the program uses to generate the shared list.

I also saw the files for audio appear and disappear in the audio.xml folder. When items were shared and unshared the name and SHA1 value of the file would appear and disappear in the *.SXML files. These files are named by file type (i.e. audio.xml, video.xml). A note of caution to users LimeWire reports the sha1 value in base32 while FTK reports it in base16. FE David Loveall wrote a quick program in Python to convert the values. I printed the conversion results into a PDF file for inclusion in the FTK report for the actual case. The LimeWire program also adds the letters "sr" to the end of the sha1 for the first entry in *.SXML file and then adds the letter "q" to the remaining. At the time of this writing I was not able to identify the significance of the "q" or "sr" designation. The "sr" designation is always the first entry and may be the programmer's notation to start the record. I have asked the assistance from other examiners from the lab to help discern the significance of these designations. I can say that all of the files in the folders listed in the LimeWire.props file after the text: **DIRECTORIES_TO_SEARCH_FOR_FILES** are shared unless noted as an exception in the library.dat file.

DOWNLOADS.DAT

The **DOWNLOAD.DAT** file is a listing in the incomplete folder that lists the files that are not completely downloaded. This file contains the partial downloads and any previews that were done. When an item is previewed during the download it will append the title to include a prefix **PREVIEW-T-[file name]**. The file does keep track of the address of the host that it was last downloading from as a reference to try and start with the same location should an interruption occur. There is an option in the program is to delete partial downloads after a set time period. These files are shared by default but this setting may be opted out by the user in the options tabs.

VISTA UPDATE

LimeWire hasn't changed but Windows certainly has. The new default locations for the configuration files (fileurns.cache, LimeWire.props, Library.dat):

%SYSTEMROOT%\Users\[PROFILE]\AppData\Roaming\LimeWire

Default Shared directory:

%SYSTEMROOT%\Users\[PROFILE]\SHARED

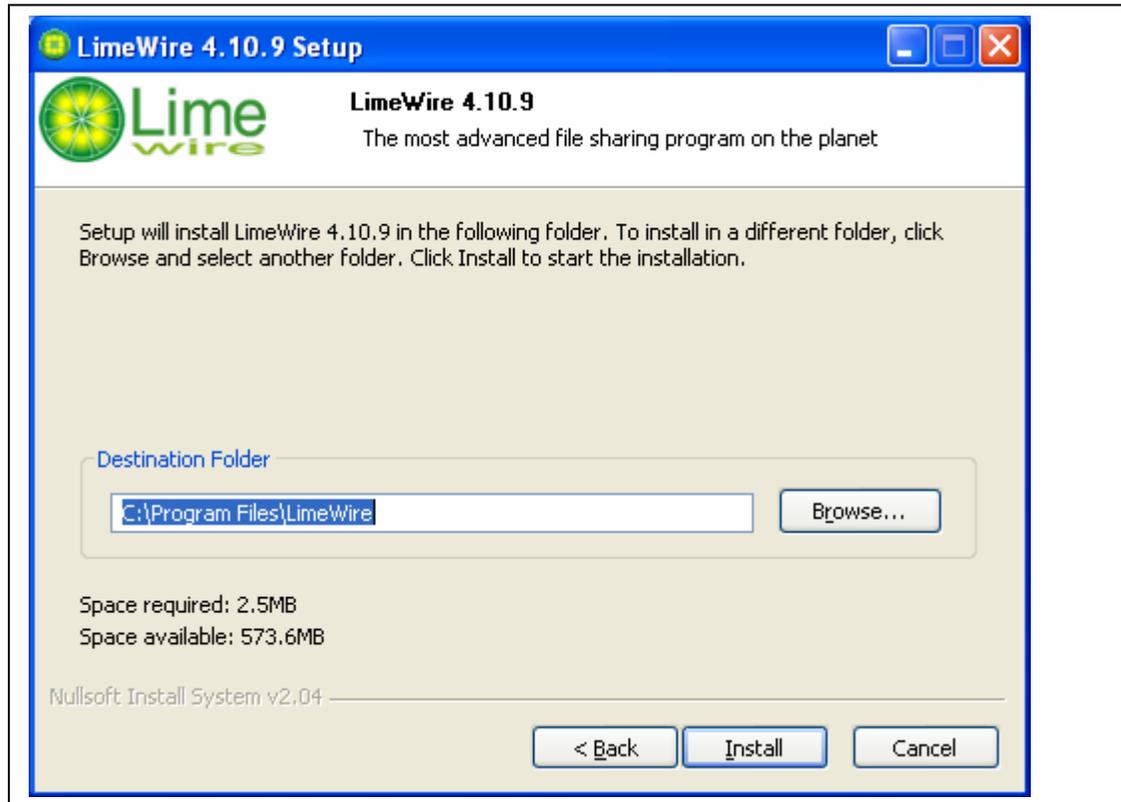
Default incomplete directory

%SYSTEMROOT%\Users\[PROFILE]\INCOMPLETE

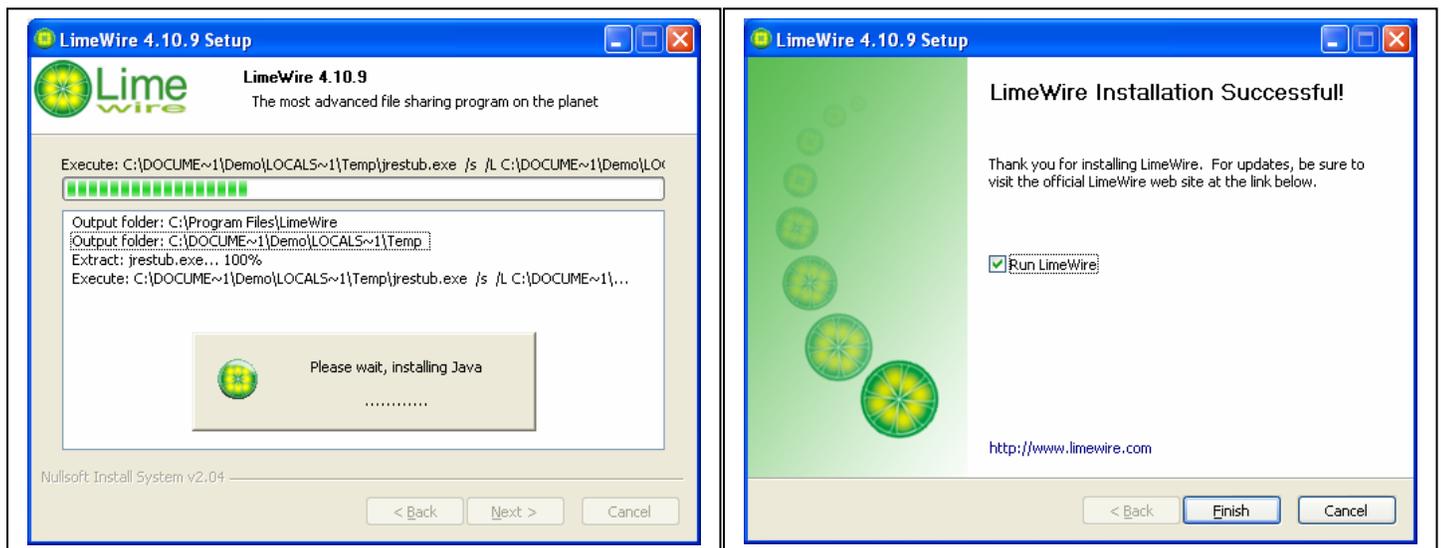
SETUP AND INSTALLATION

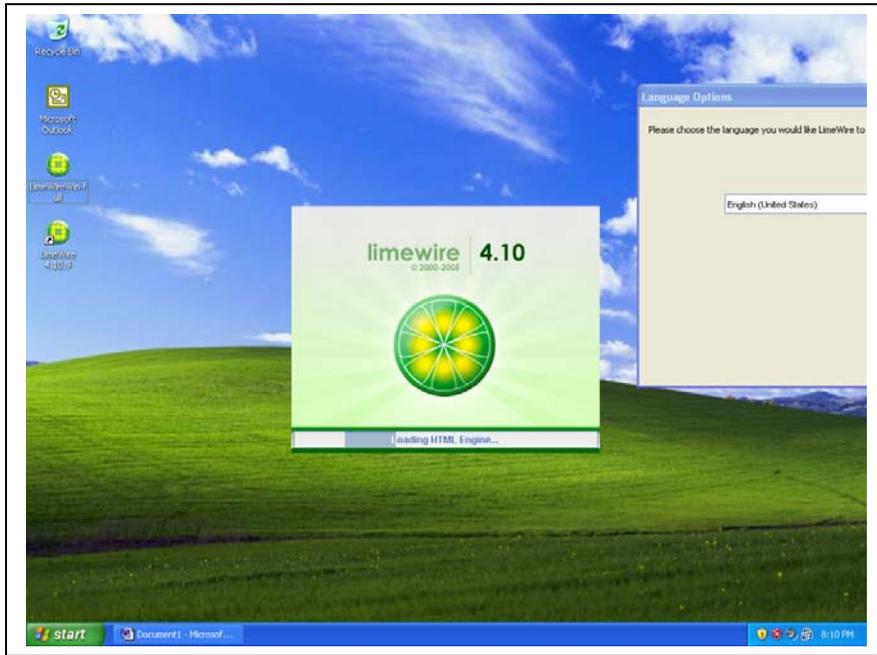
The following section is a recap of the installation process. It may lend itself to insight to what an investigator may look for in a case.

OPENING SCREEN



The first option in installation of the software is the ability to change the software location (like any other program). After going through the setup it asks to launch the software. At this point the software is **not fully installed**. This is evidenced by the limewire and incomplete directories not being installed. This occurs after the options are activated during the next phase of installation.

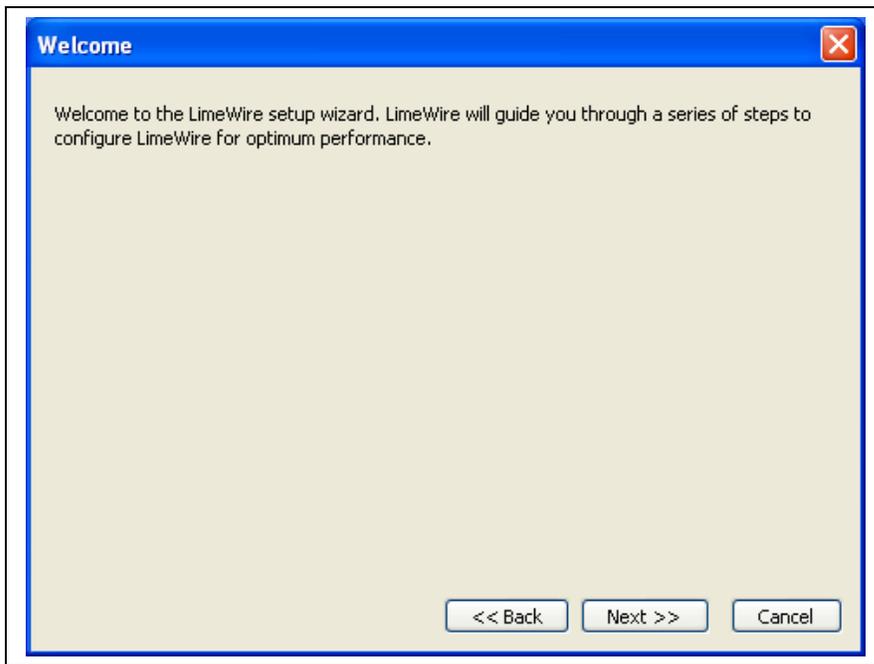




Once the program is started it loads the java platform and walks you through the install options. It quickly lists the starting processes in the bar below the LimeWire logo.



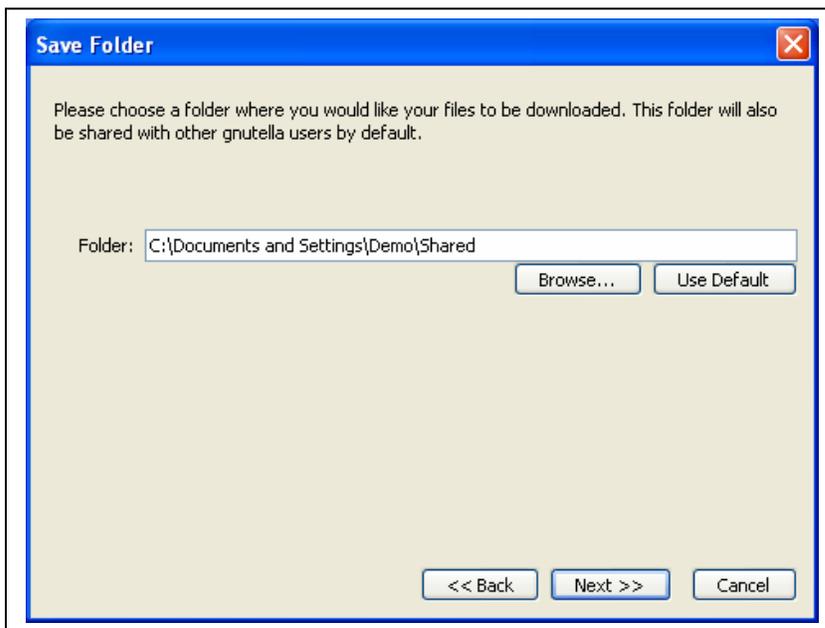
The first choice to make in this stage is language. According to one of the daily tips shown at the start of the program (after complete installation) boasts that the program supports several languages and even searches text translated in several languages.



Once the language is chosen, the next screen is the Welcome Screen. It notifies the user that this program is adaptive to the hardware it is running on. In my case it was a VM machine using the profile of DEMO.

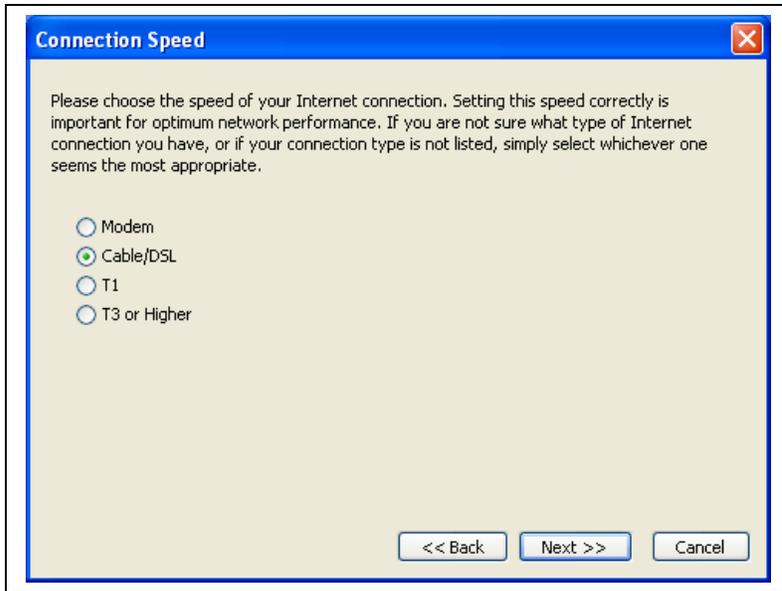
This is an important screen to note as the “restore default settings” button

may activate options not in the original default settings when first installed. An example of this will be later explained in the area discussing connection speed.



Following the notification of adaptive installation, the user is prompted for the designation of a saved folder for default download. You can set it to an existing folder or leave it as the default **C:\Documents and Settings\[User Profile]\Shared**. The text in this screen also notifies users that the “folder will also be shared with other gnutella users by

default.” This could lend itself to prove that the user was notified that they would be sharing files.



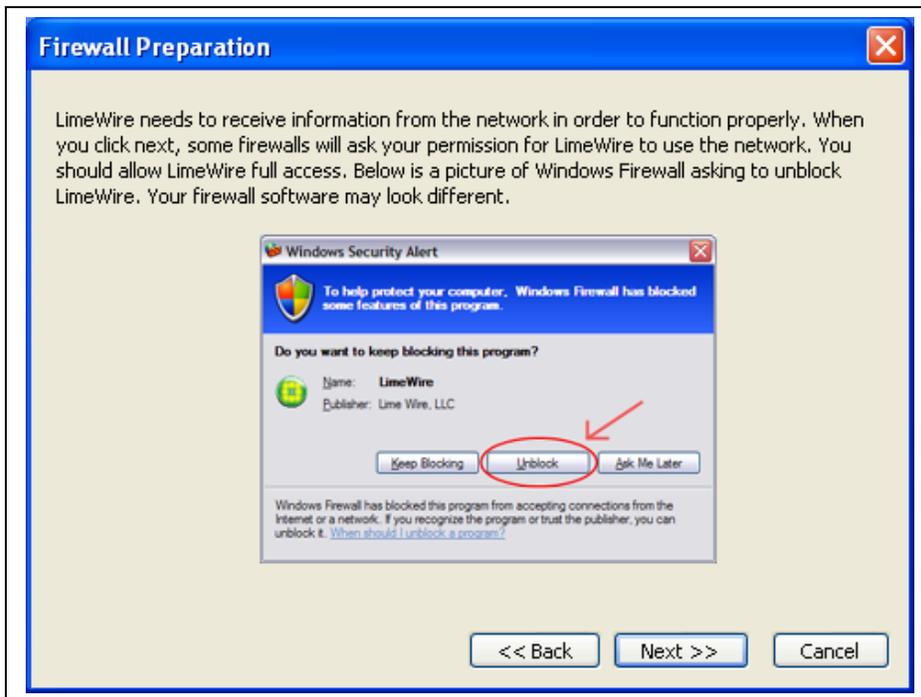
Once the save location is set, the user is prompted for the speed of the internet connection being used. It is an auto-detect function. In the sample case it defaulted to Cable/DSL on the virtual machine that was connected via wireless router on a DSL connection.

When investigating the options inside the program chose the reset default button and it

changed this setting to Modem. The auto-detect seems to be active only during the initial installation.

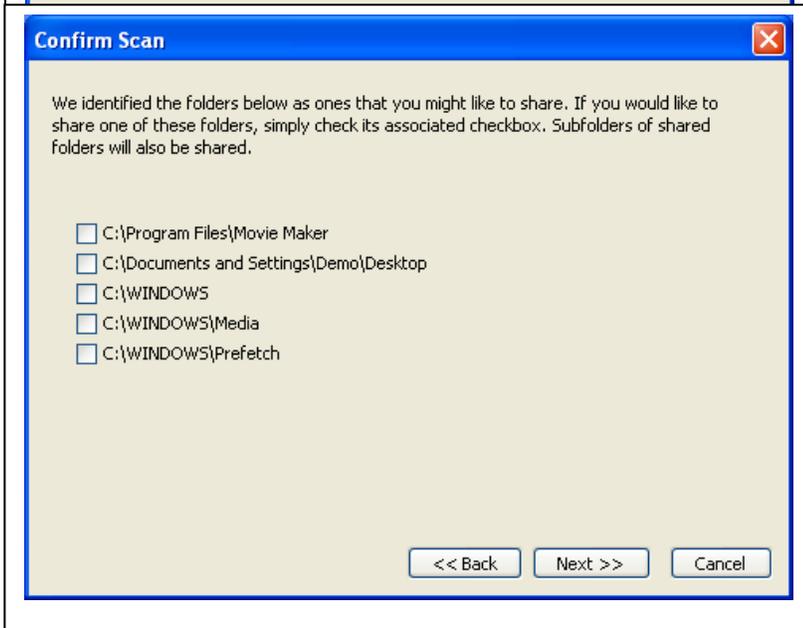


The user is then given the option to start up the program when they log on to the computer. It is set to automatically start by default setting.



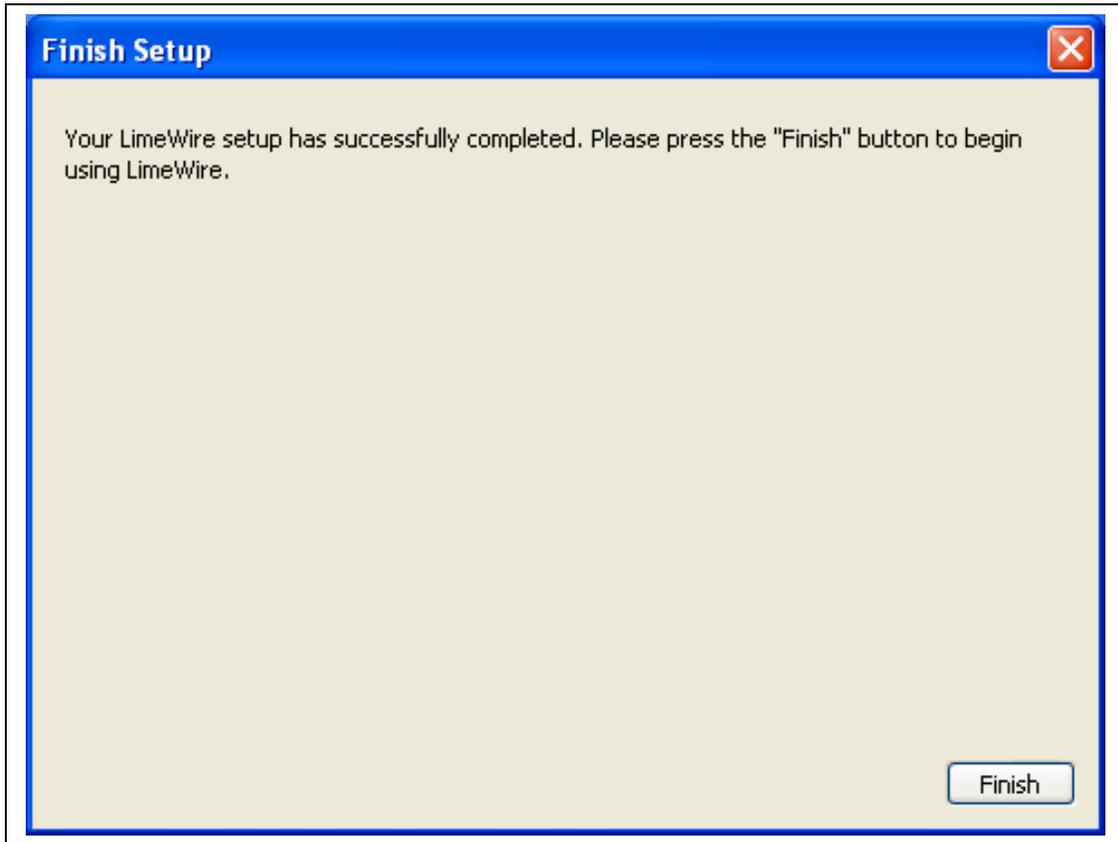
The next two screens deal with a firewall detection. It notifies the user that they are enabling an open port through the firewall to use the LimeWire network and “allow LimeWire full access”. In the user interface the bottom tray displays if a firewall is detected and active on the user’s computer.





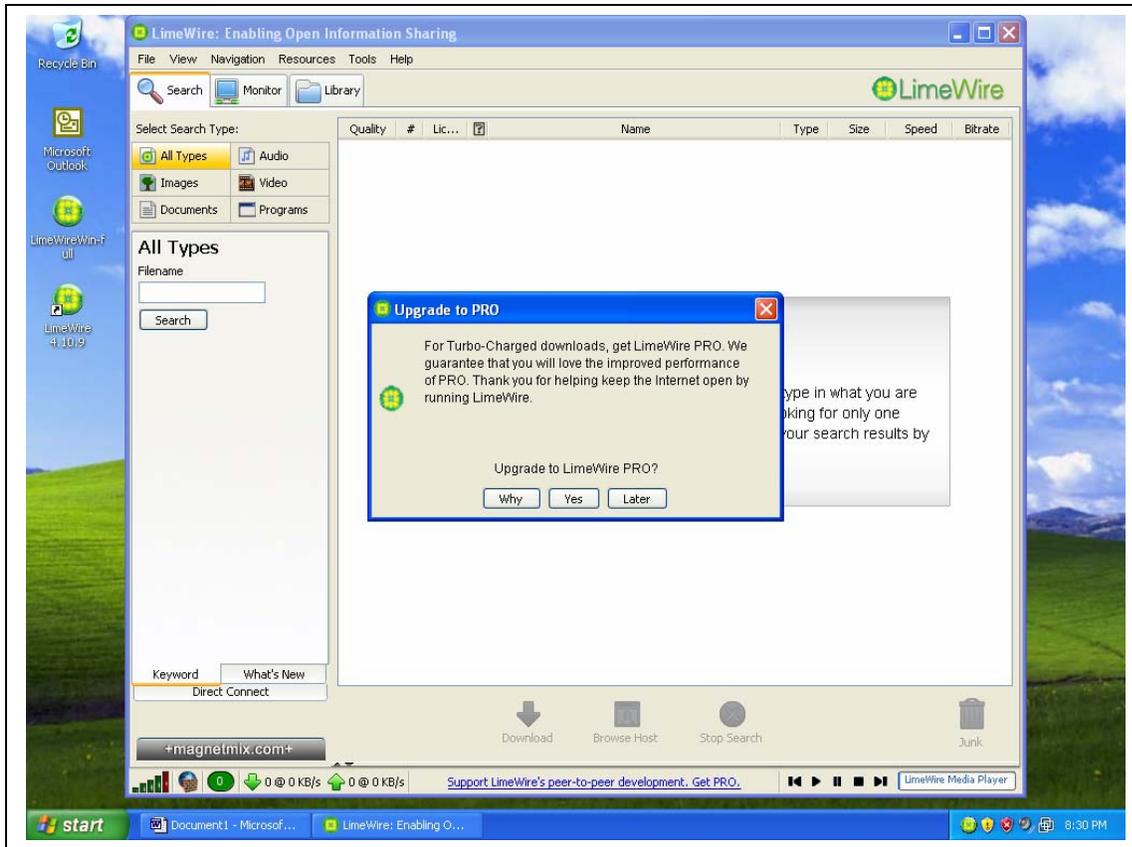
Once the firewall settings are complete and the program has access to the internet, the user is prompted to allow the program to scan the hard drive for other media. It notifies the user if a scan is allowed it will later give an opportunity to specify which folders you would like to share. After the scan it lists all drives that it found recognized media. They are **not shared** by default.

If the user chooses to share these folders (or later chooses to share these folders on the options tab) the entry **DIRECTORIES_TO_SEARCH_FOR_FILES=C:\Documents and Settings\Demo\Shared** in the **LIMEWIRE.PROPS** file will be amended to include that path in addition to the default folder.



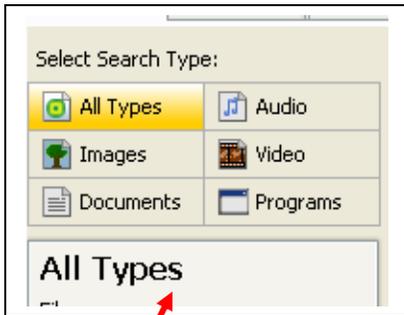
The final screen in the setup process is the acknowledgement that the LimeWire program has setup successfully. It then allows the user to open the program to the user interface screen.

THE GUI



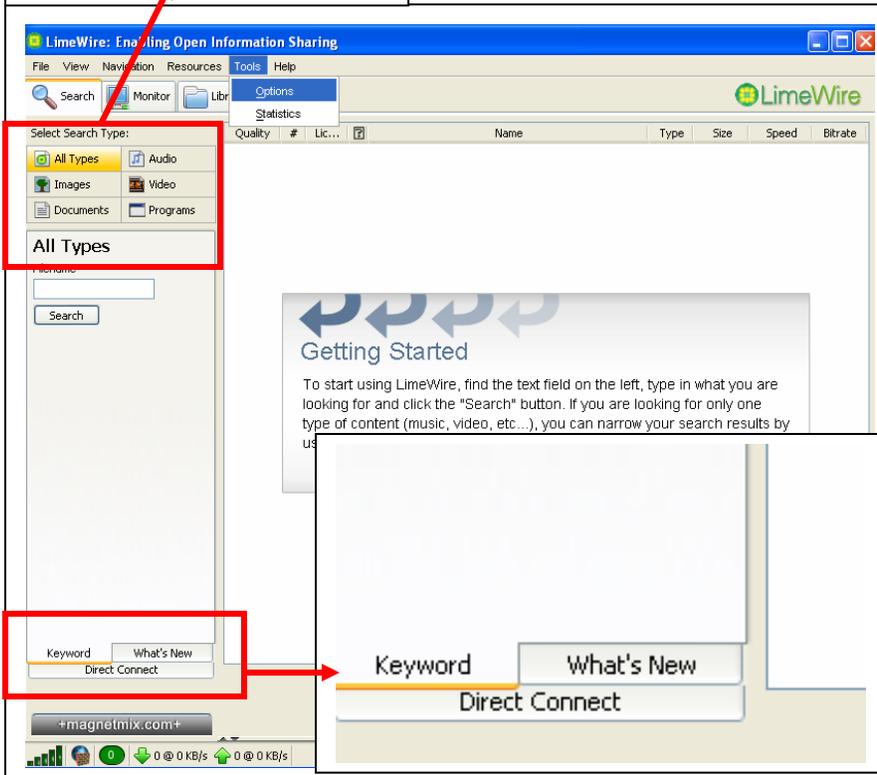
The LimeWire GUI is a window with three tabs. The search tab, monitor tab, and library tab all have features that may need to be examined.

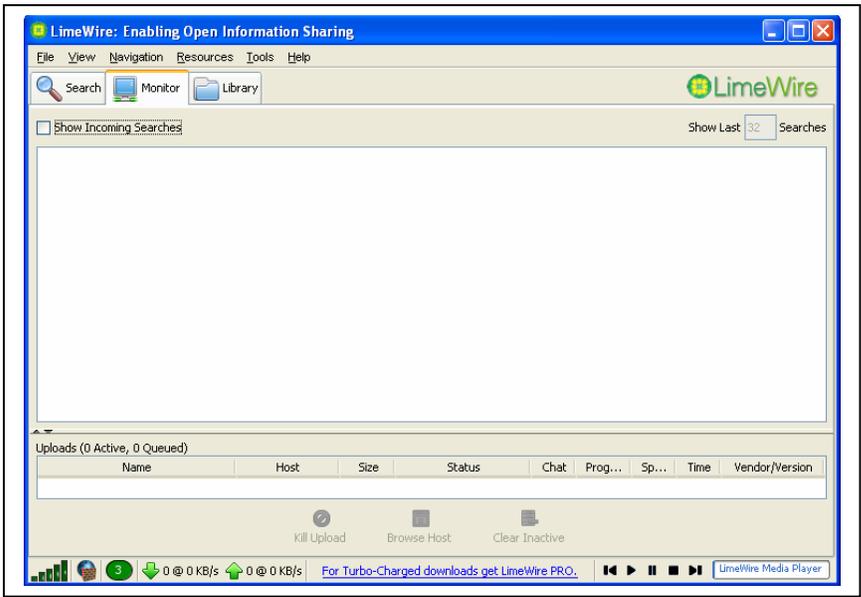
Every time the program is started with the free mode it asks to upgrade to LimeWire PRO and varies the location of the three choices, (**Why**, **Yes**, and **Later**).



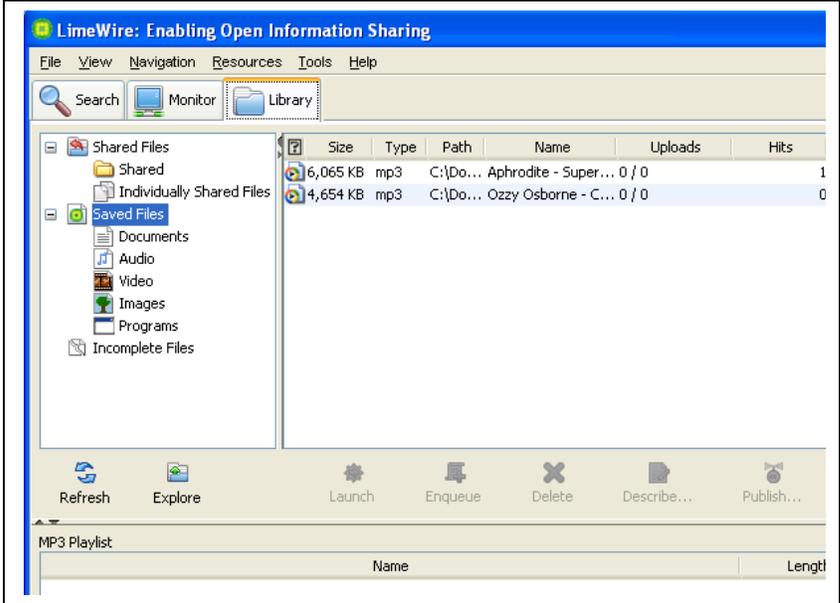
LimeWire's search functions can be limited to the type of program you wish to search for or a general search for all types of media meeting the search terms entered. The user can even search for the newest items available with the What's New submenu. The user is also allowed to exchange in a direct connection if the IP address of the host computer is known. This feature listed that I was not able to establish a direct connection due to a firewall being enabled. Search terms are

cached into the system and will reappear in auto complete form until the program is turned off or the computer is powered down. At the time of this writing members of the lab were still trying to locate the storage of the search terms in live memory.

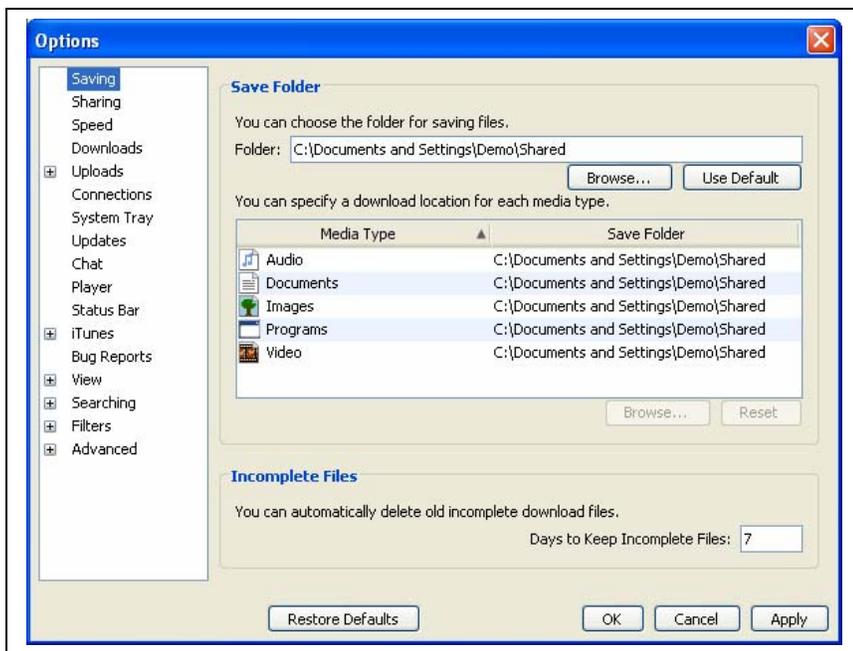




The monitor tab tracks incoming queries and uploads. By default this is disabled, but could be activated with a check mark.

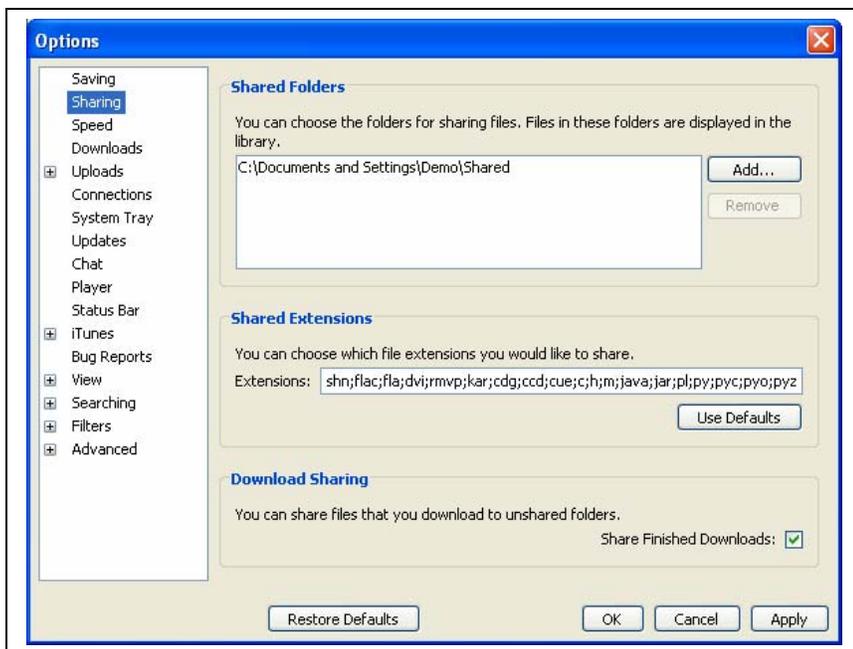


The library tab is where you can find the shared folders and files. In this image the contents of the shared directory is displayed in the right pane with the left pane showing the different divisions recognized by the program.



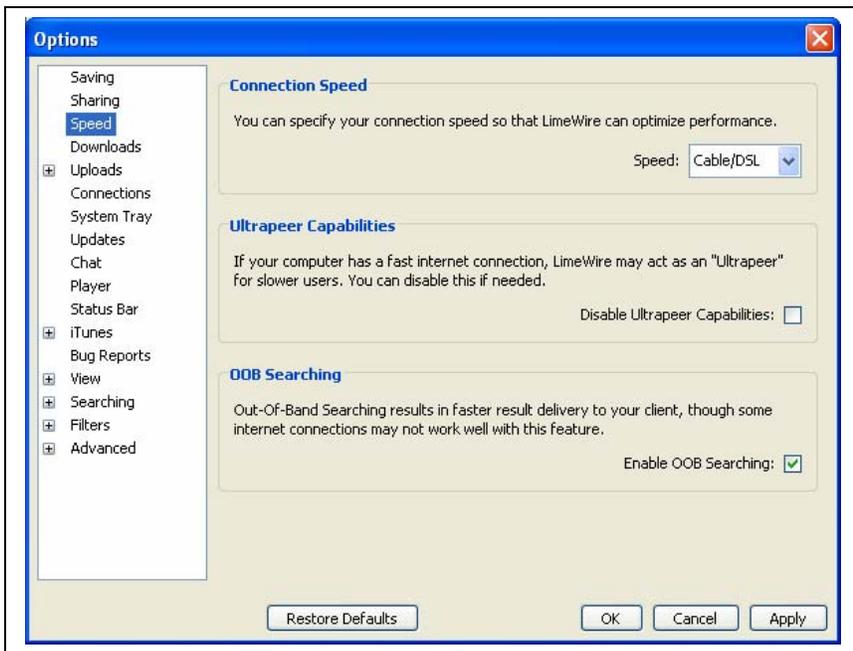
The options tab allows the user to customize several options that are of interest to the investigator. The saving tab shows the default saving directory and allows the user to change the settings for individual types of media or the entire group. Another item of interest to the forensic investigator is that incomplete files and downloads are purged automatically after a period of time set by the user. This is extremely important to consider if the suspect file

was only partially downloaded from the suspects computer. All incomplete files are shared in the partial state in support of the swarming downloads. This option can be disabled in the Uploads tab. The files that reside in the incomplete folder are shared by default.

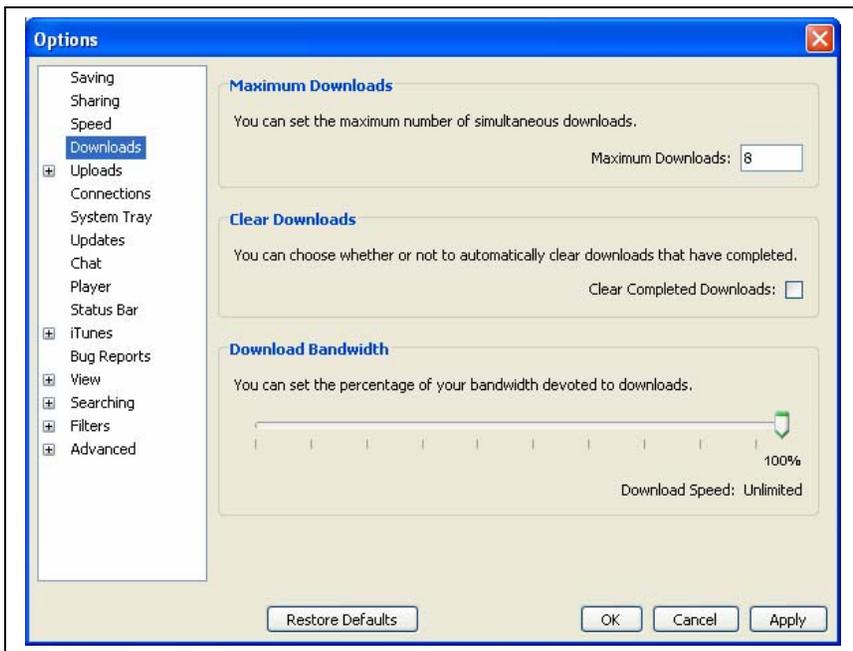


Shared folders tab holds the settings for all shared directories. It also shows the default extensions to share. This can be limited or expanded by the user. This is the screen where the user can share additional entire directories and also can determine to share downloaded files. This check box adds or deletes the entry in the **LIMEWIRE.PROPS** file **SHARE_DOWNLOADED_FILES_IN_NON_SHARED_DIRECTORIES=false** which can determine if

sharing is active or not on downloaded files. The files saved into the shared folder are shared regardless of this check box setting. The **LIMEWIRE.PROPS** only tracks the current settings. If this is changed it does not delete prior shares just shares downloaded with the option disabled. By default all downloaded files are shared.

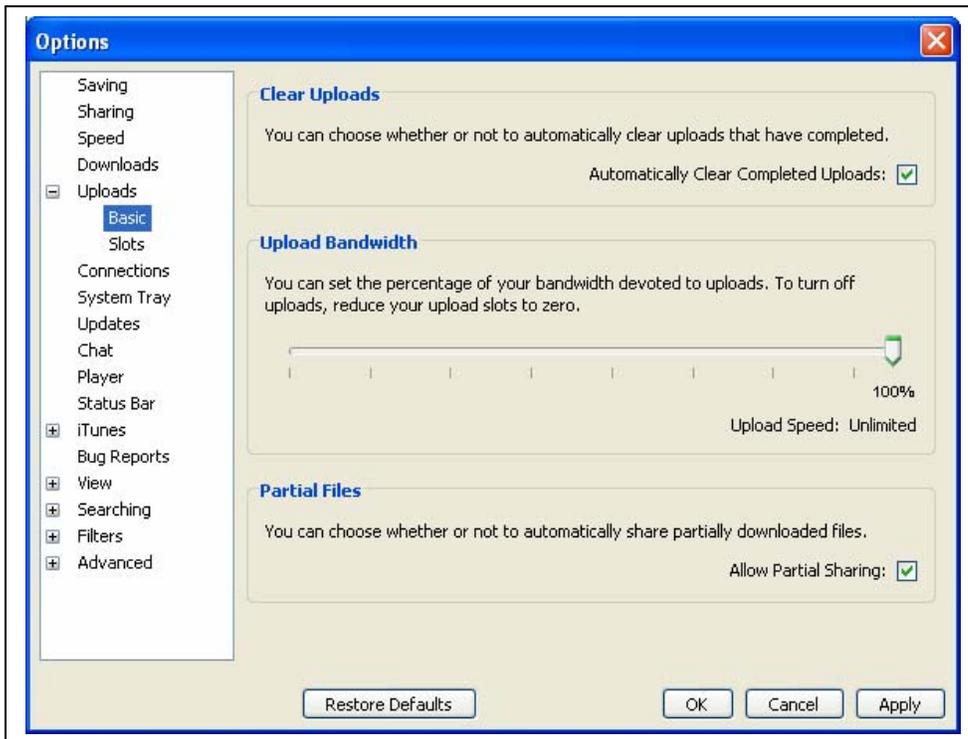


The speed tab contains an option to allow limewire to use your connection and bandwidth to act as an Ultrappeer. This is enabled by default but may need to be checked. Out of band searches can also be disabled in this tab.

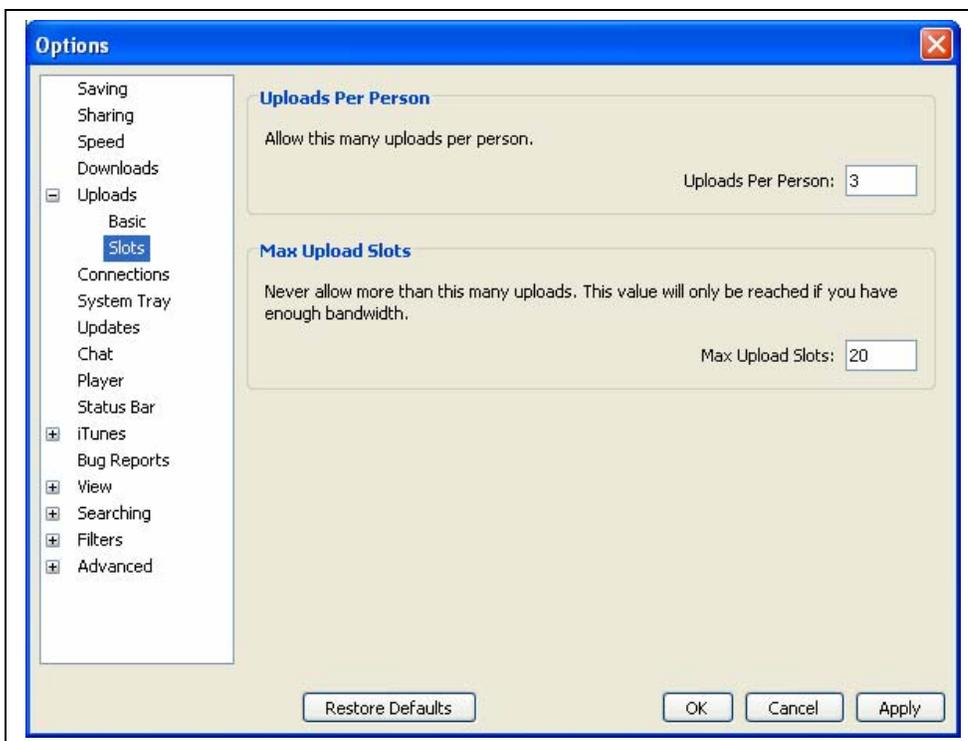


The download tab allows the user to limit the amount of downloads that are simultaneously downloaded. In this instance it was limited 8 downloads but when I used the Restore Defaults button it changed the limit to 10.

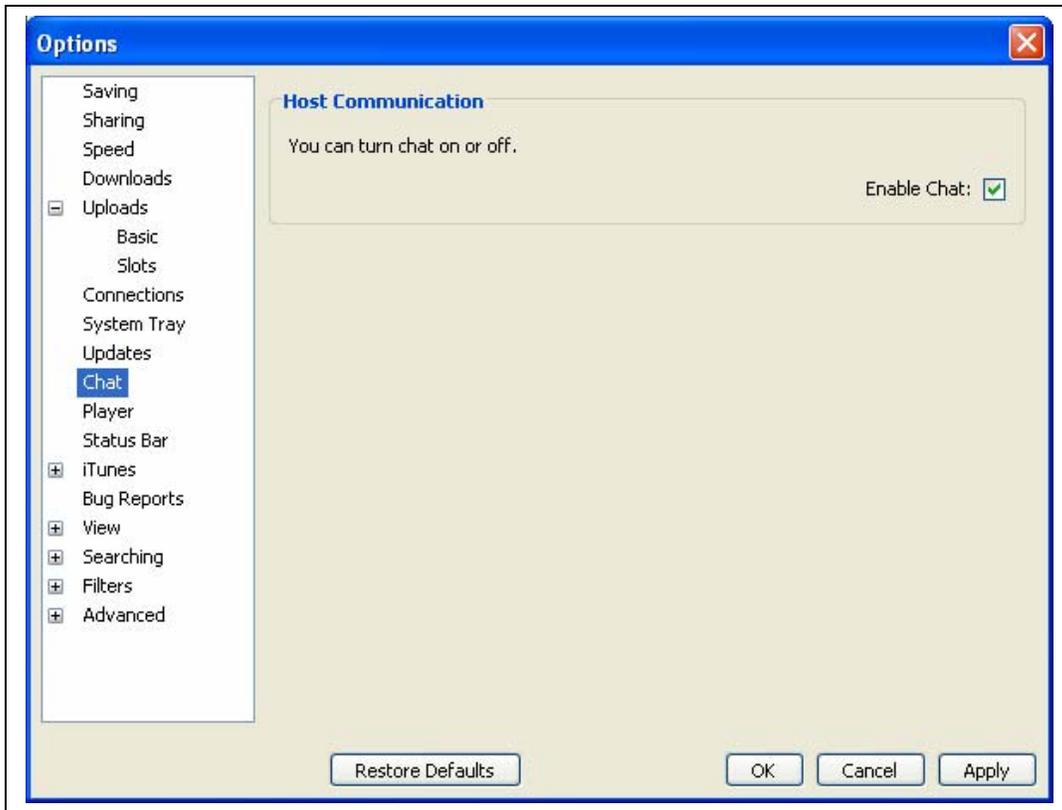
Another option for the user is to set the percentage of bandwidth to devote to downloads.



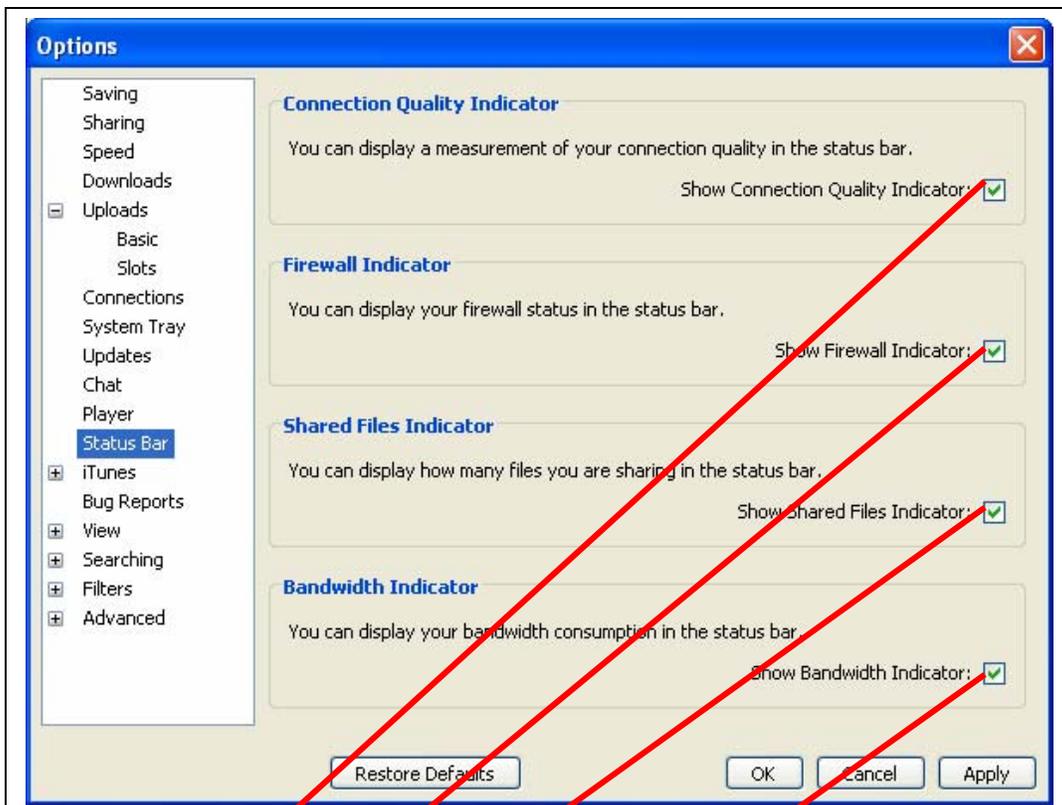
Uploads tab is broken into two submenus. In the Basic submenu the user has the option to limit the bandwidth devoted to uploads. In addition the user is able to disable partial downloads from being shared.



The second submenu in this setting is the slots. This limits the number of uploads per person (at a time) and the number of slots.



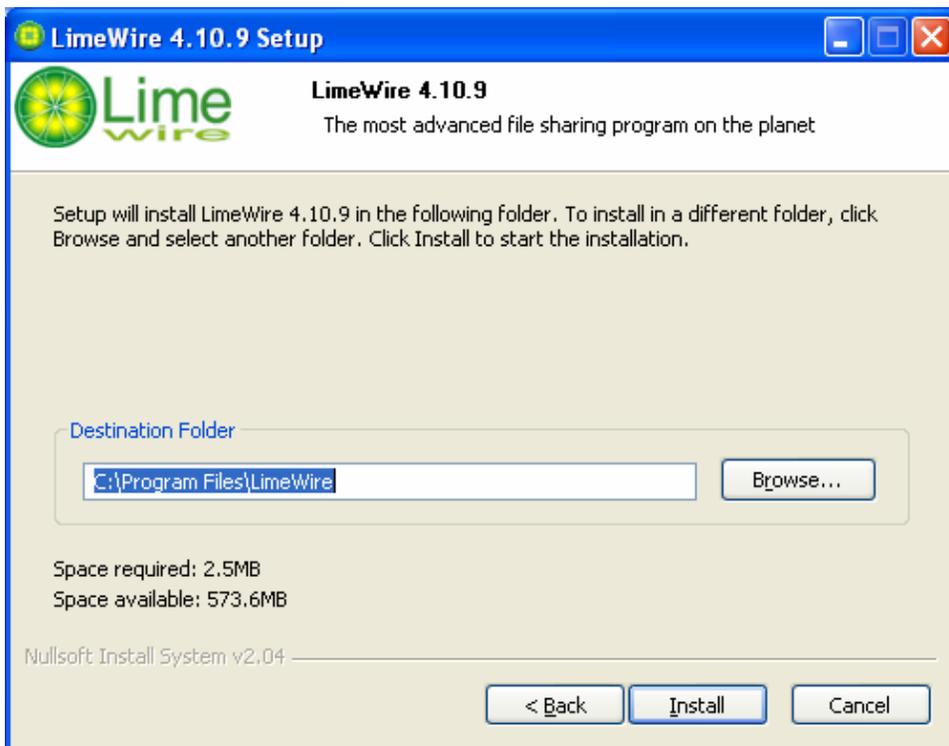
Sets chat on or off. On is default.



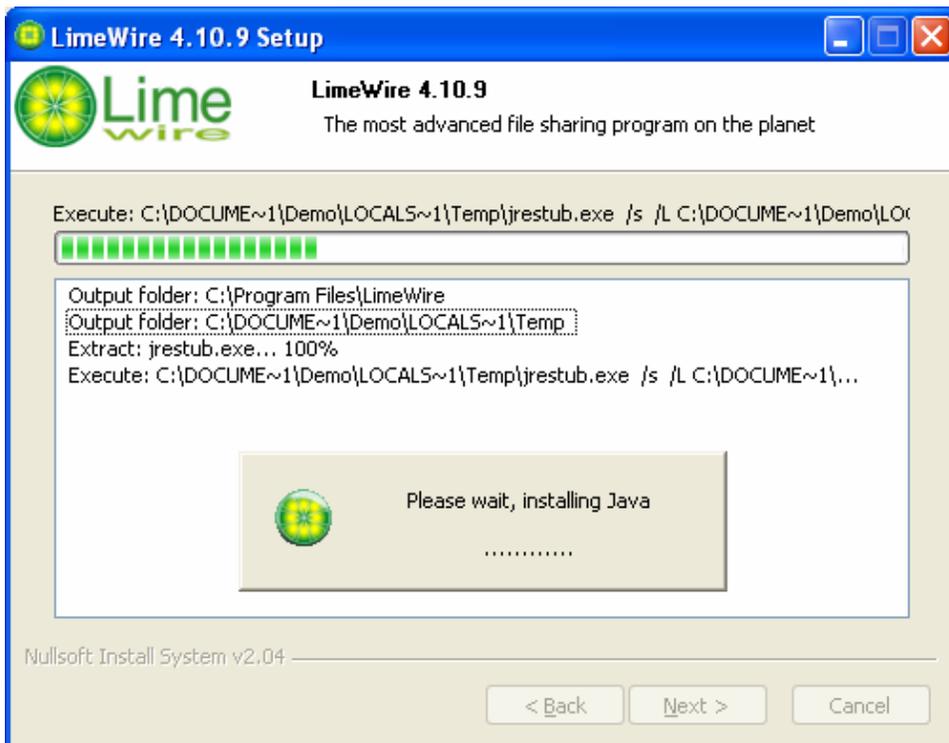
This tab allows the settings of the indicators in the bottom of the limewire screen these default setting produce the result shown below.

The remaining settings are shown in appendix B.





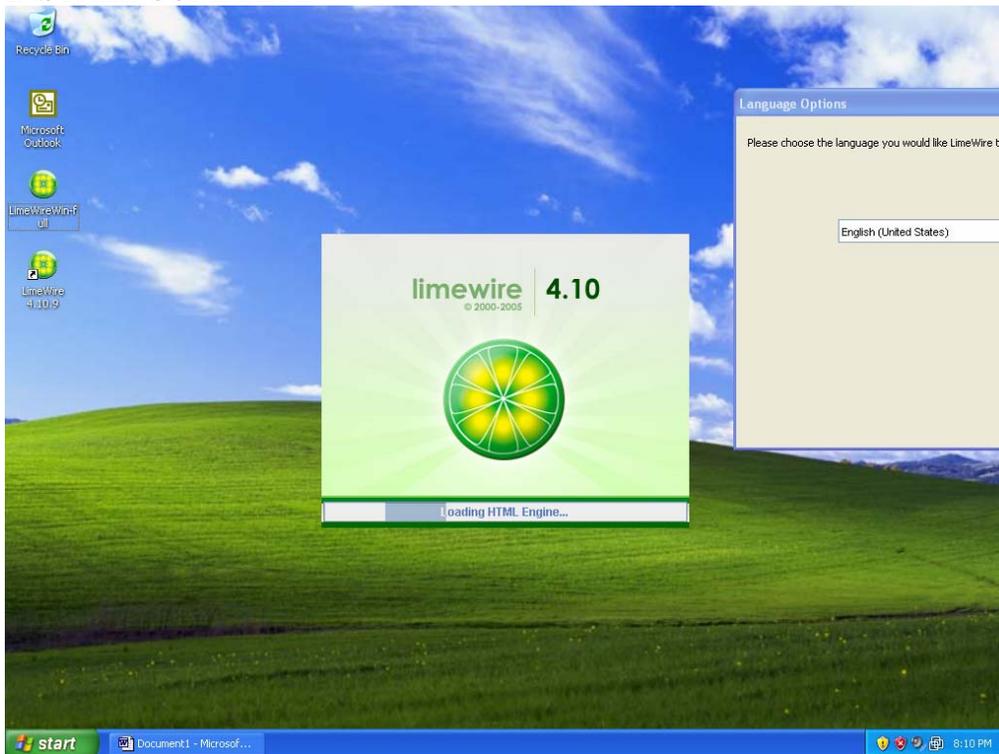
OPENING SCREEN



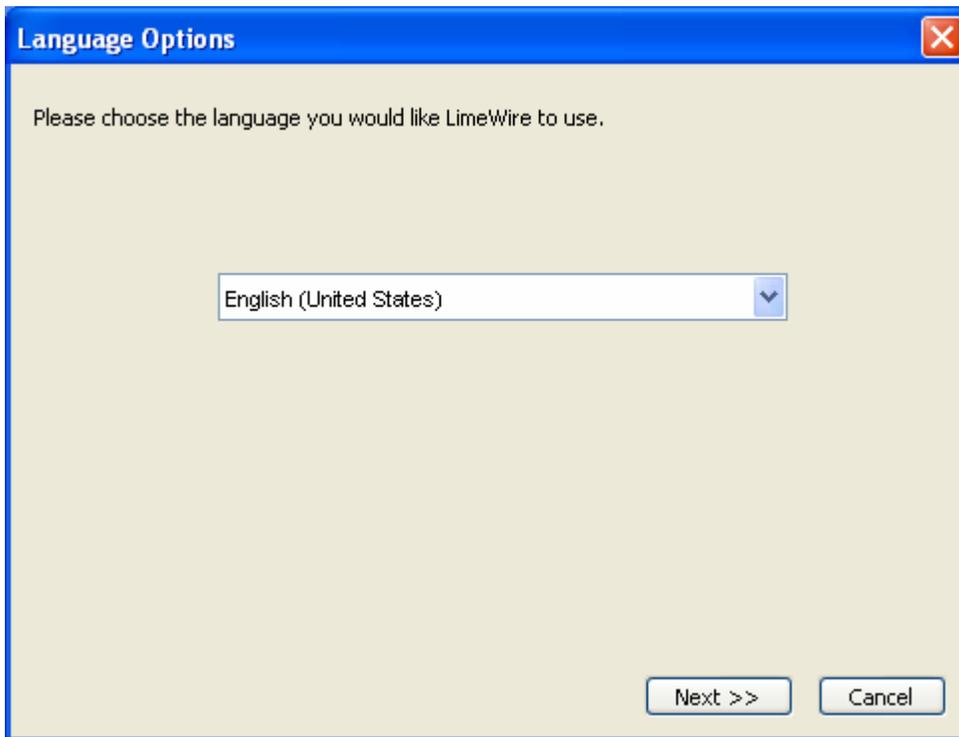
LOADING SCREEN



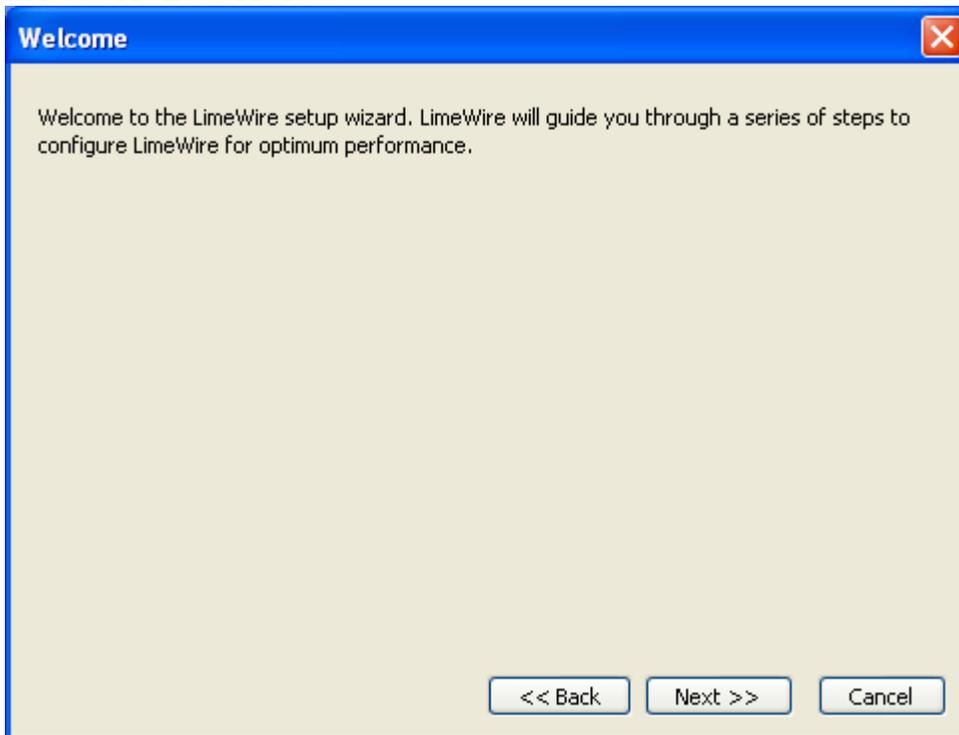
INSTALL COMPLETE



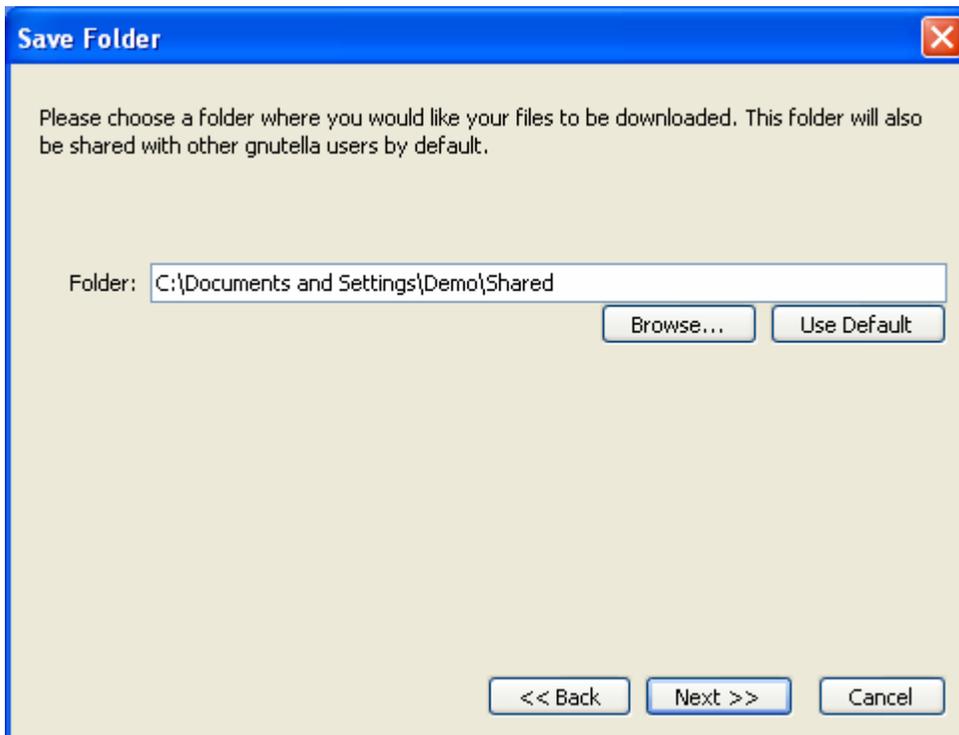
Starts loading HTML.



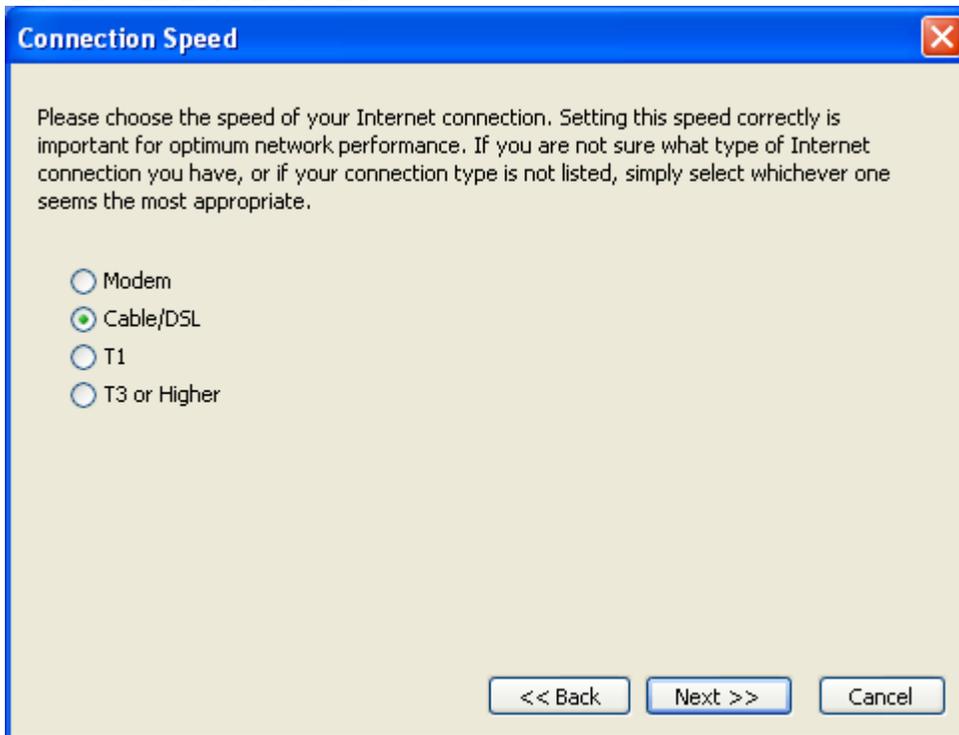
LANGUAGE OPTIONS



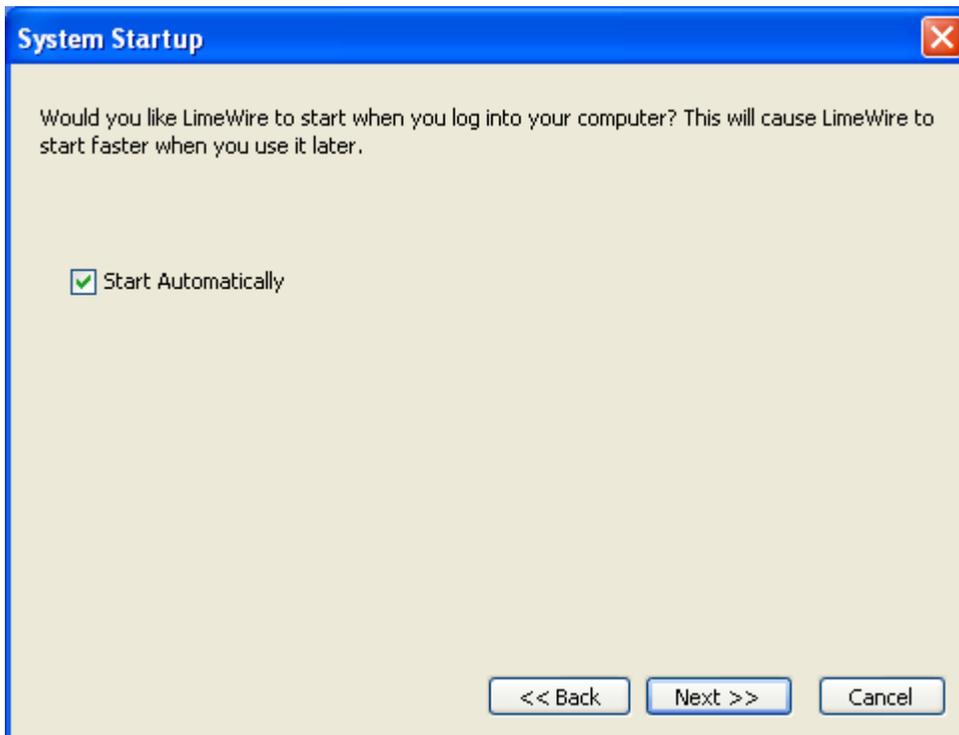
WELCOME SCREEN



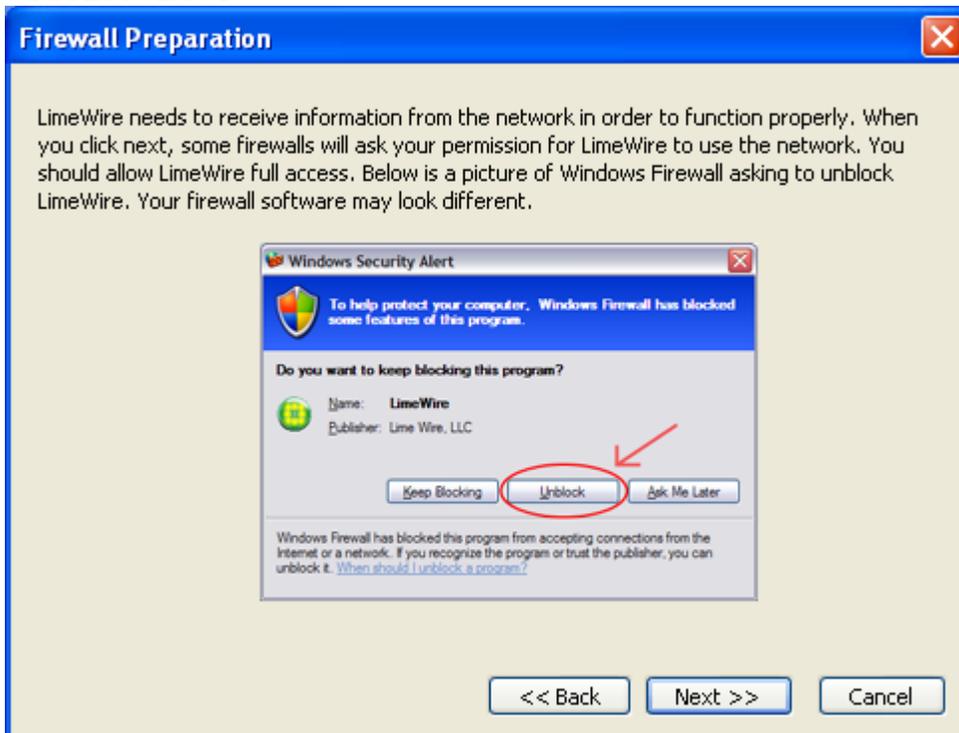
SAVE FOLDER DEFAULT



CONNECTION SPEED



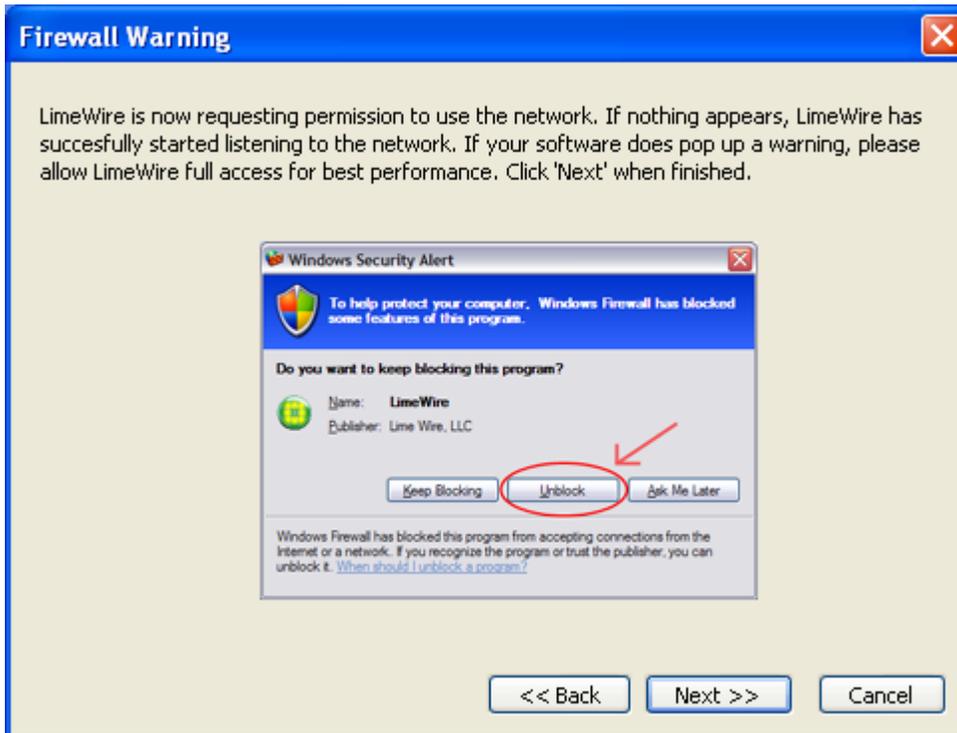
SYSTEM SETUP



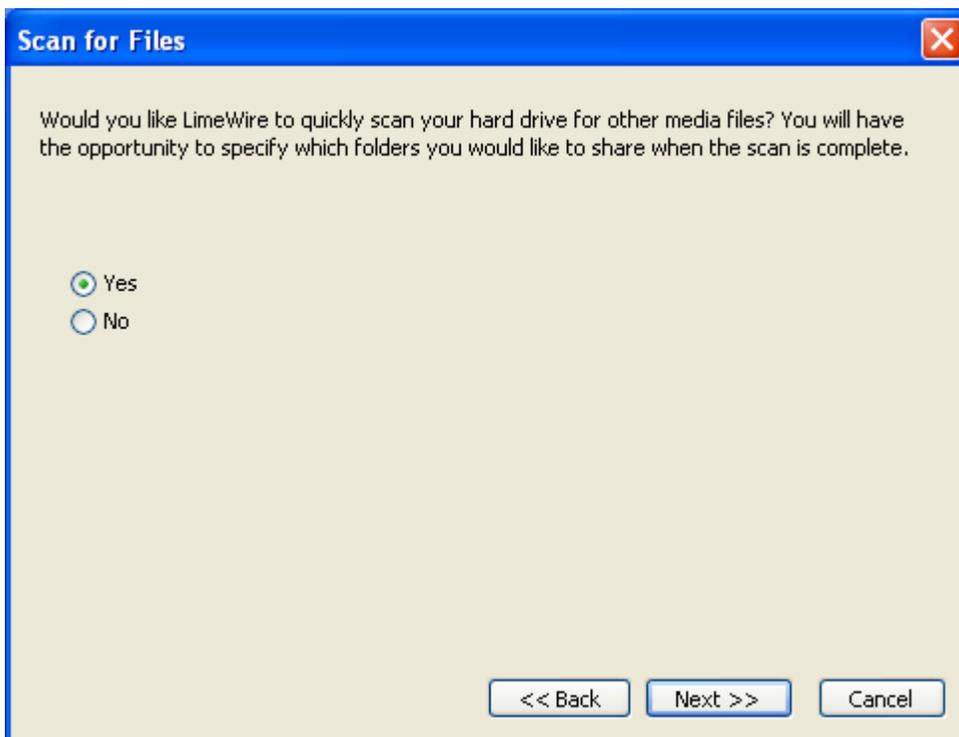
FIREWALL REQUEST



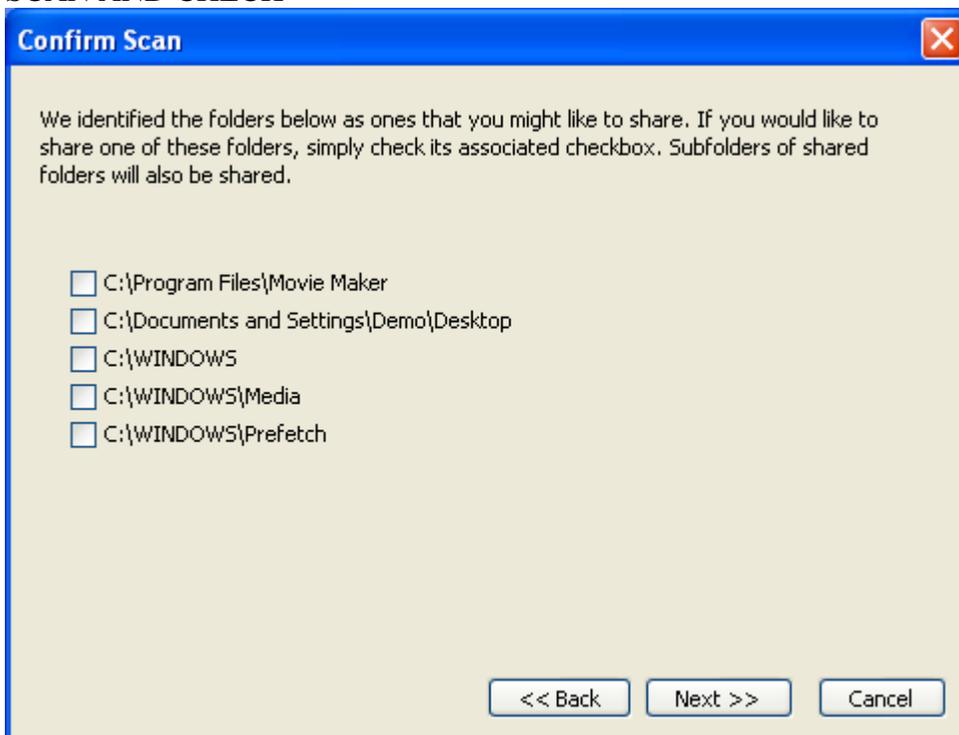
WINDOWS FIREWALL ALERT



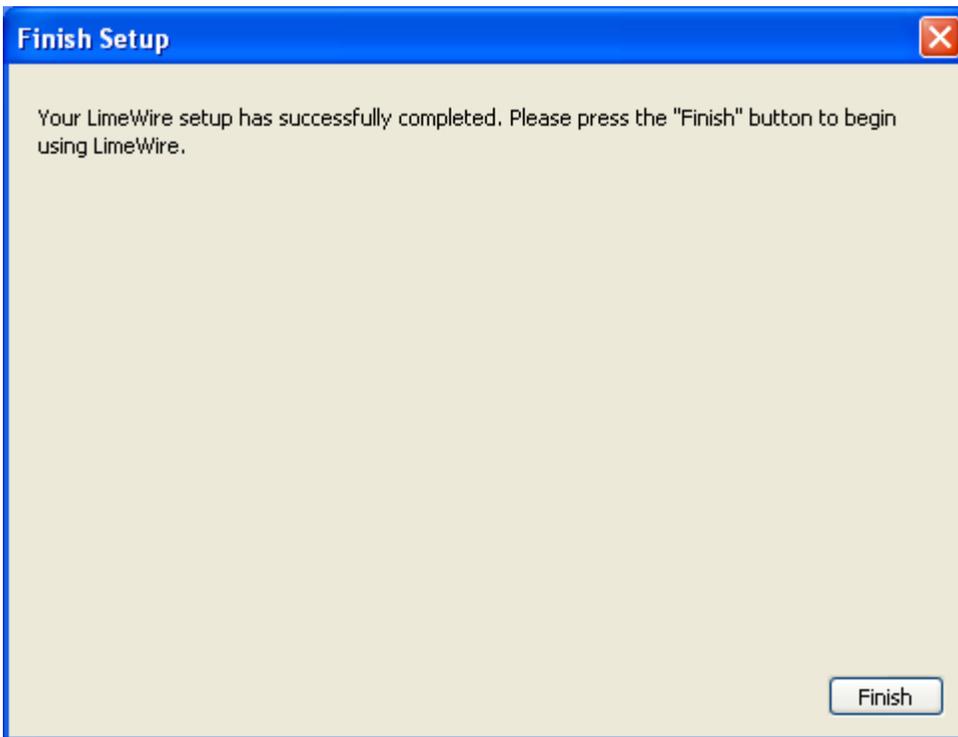
RETURNS TO THIS SCREEN



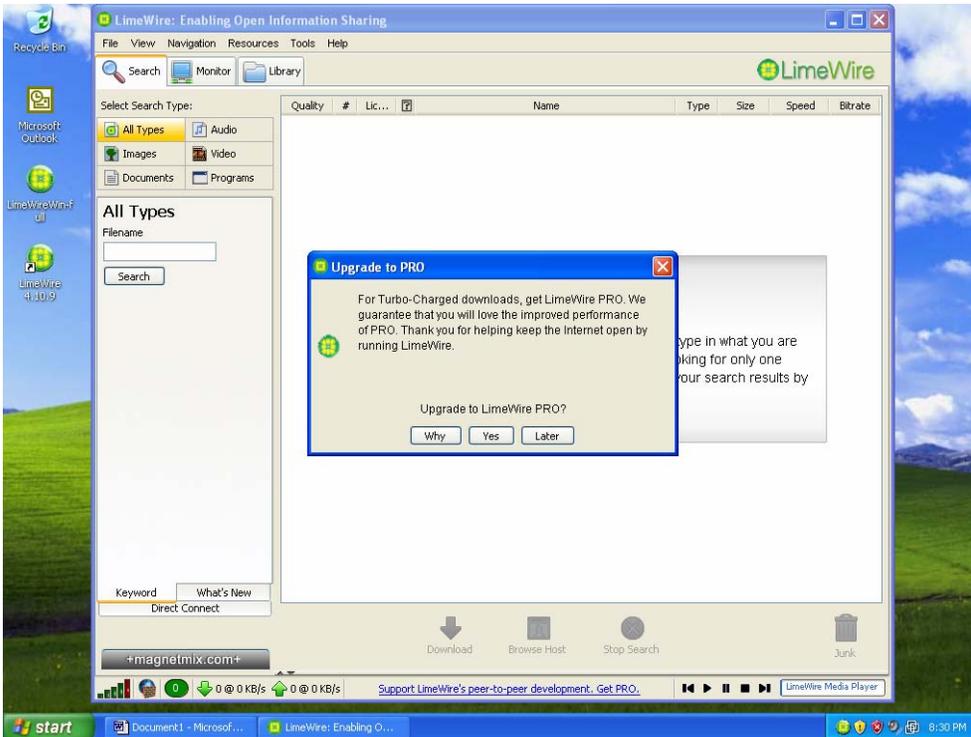
SCAN AND CHECK



MEDIA SCAN RESULTS

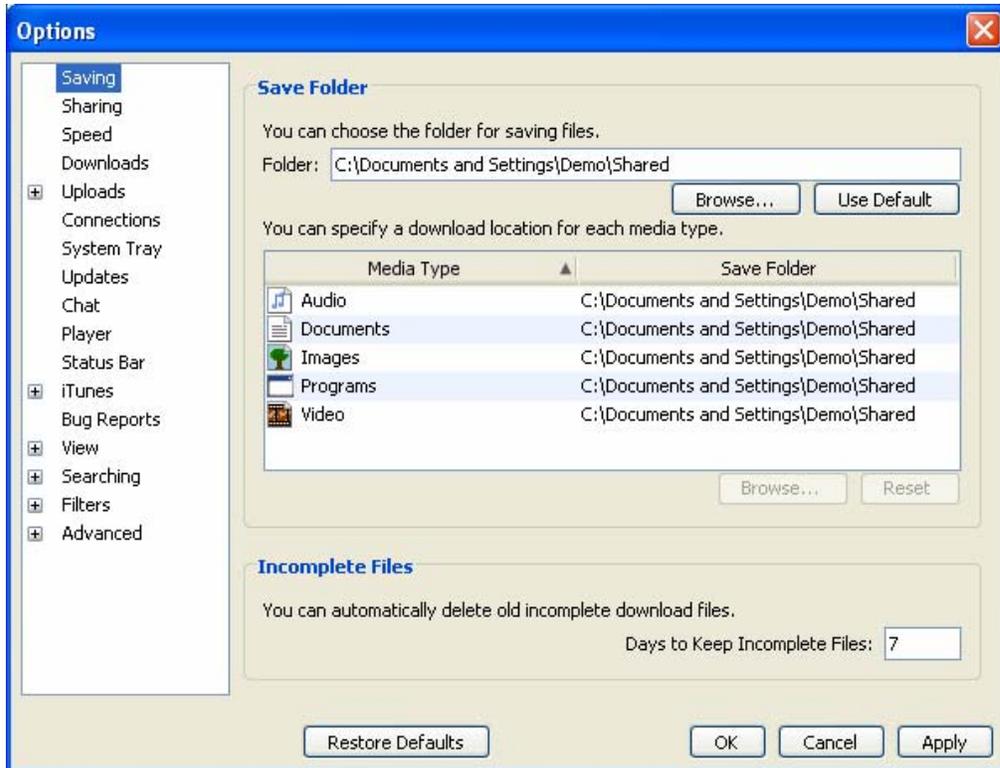


SETUP COMPLETE

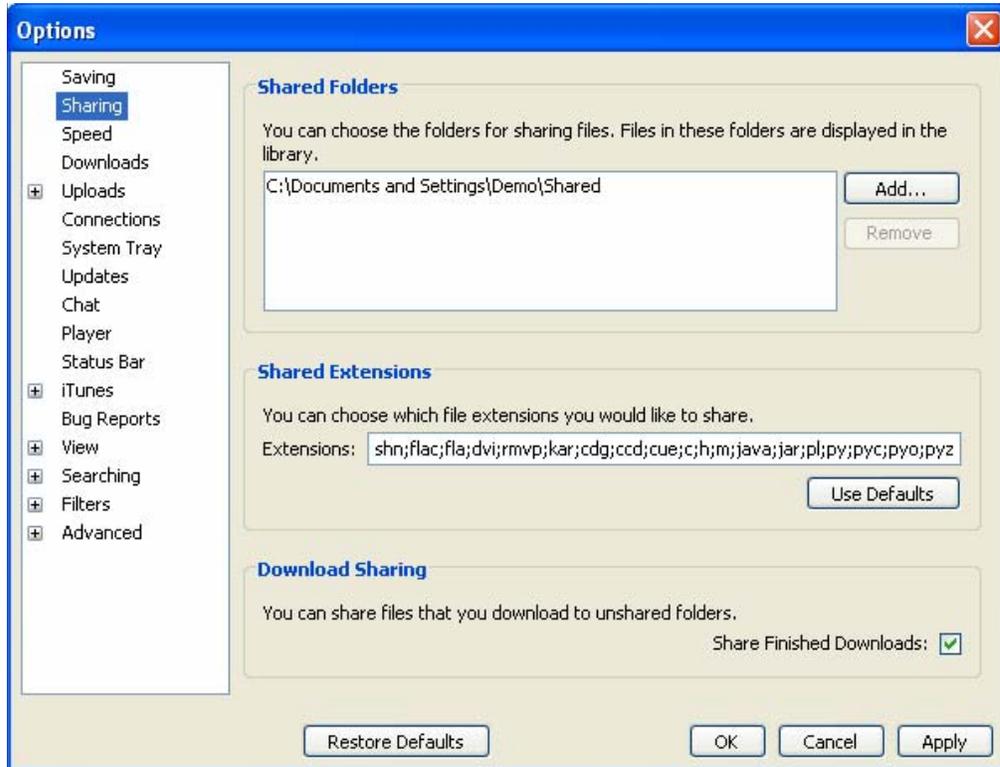


START UP

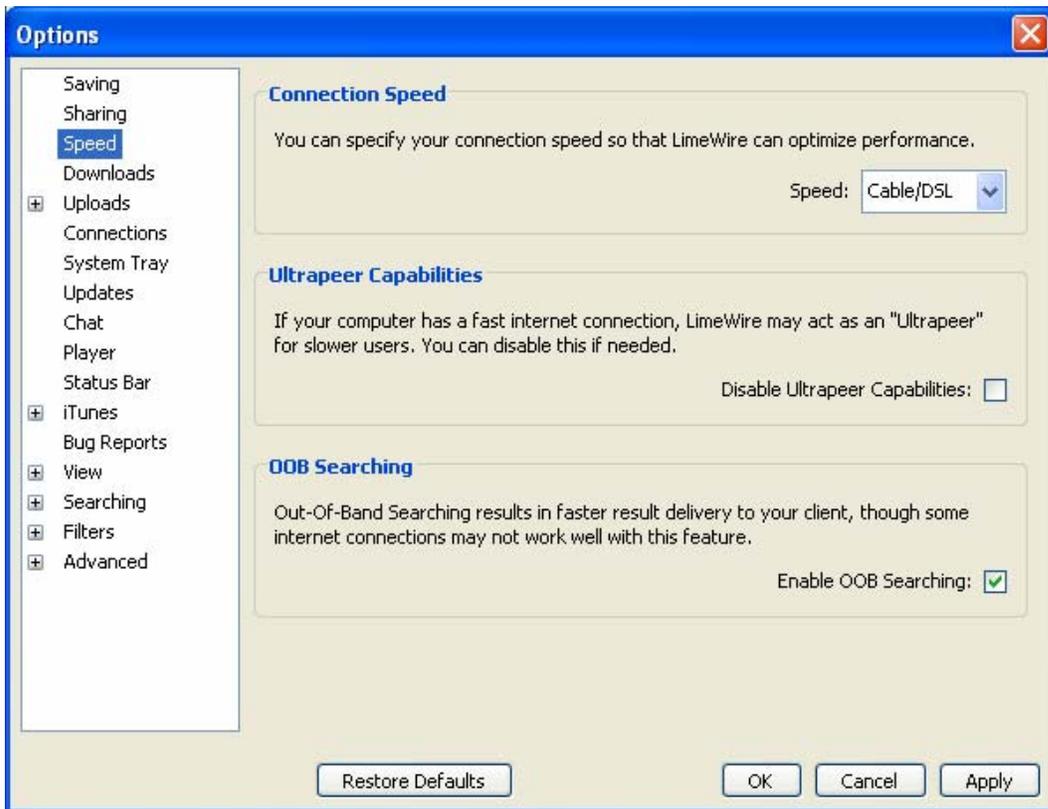
APPENDIX B – OPTIONS WITH DEFAULT SETTINGS



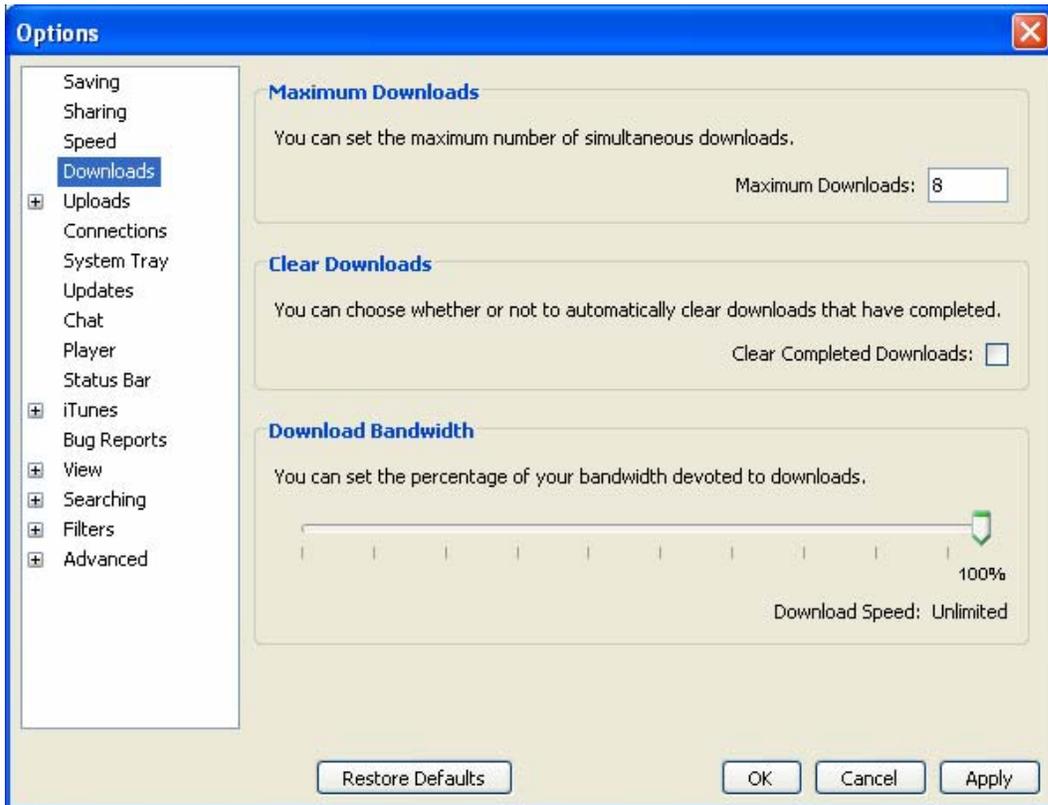
OPTIONS - SAVING



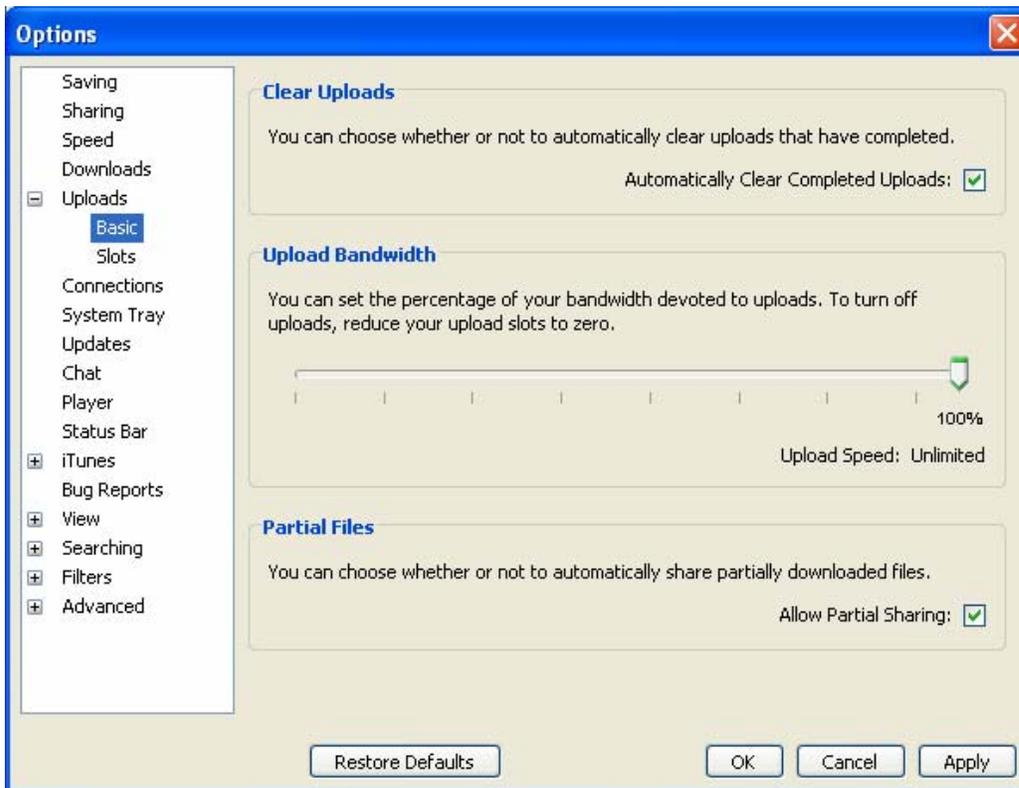
OPTIONS-SHARING



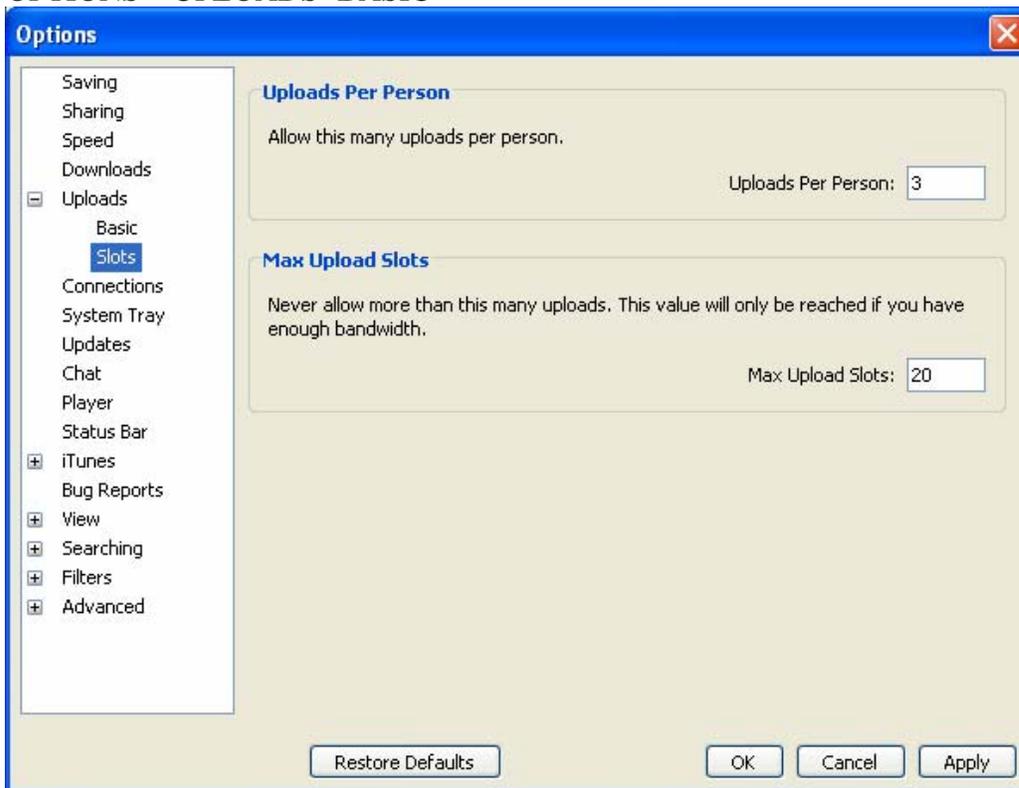
OPTIONS – SPEED



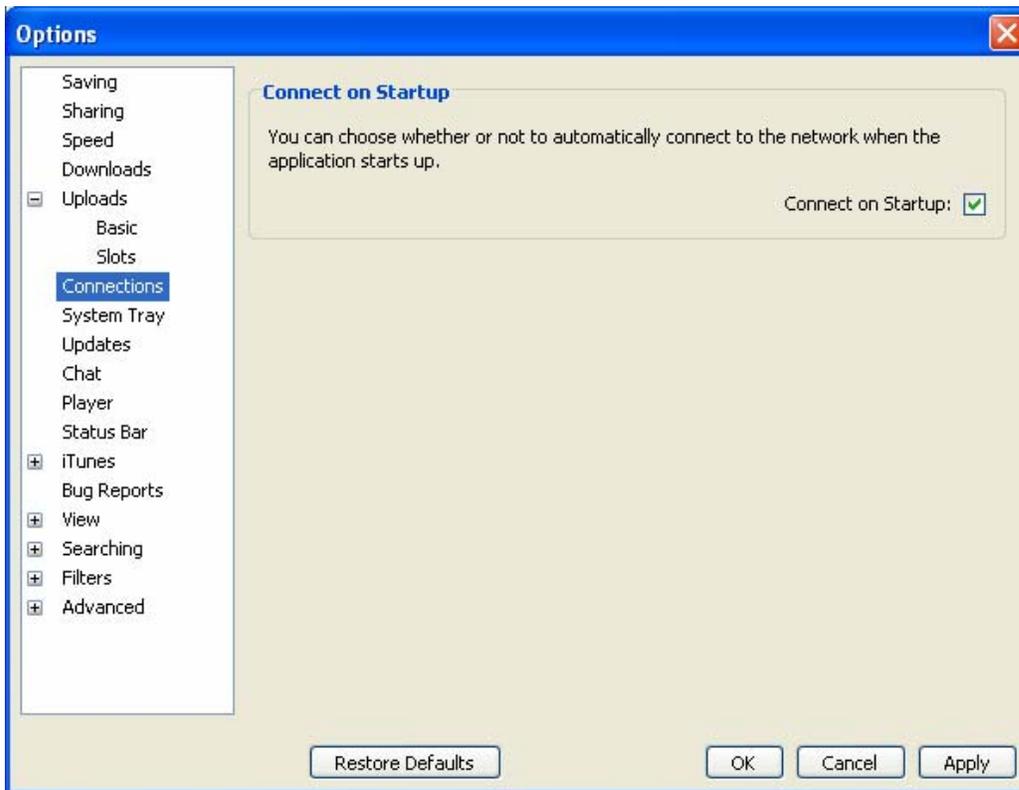
OPTIONS – DOWNLOADS



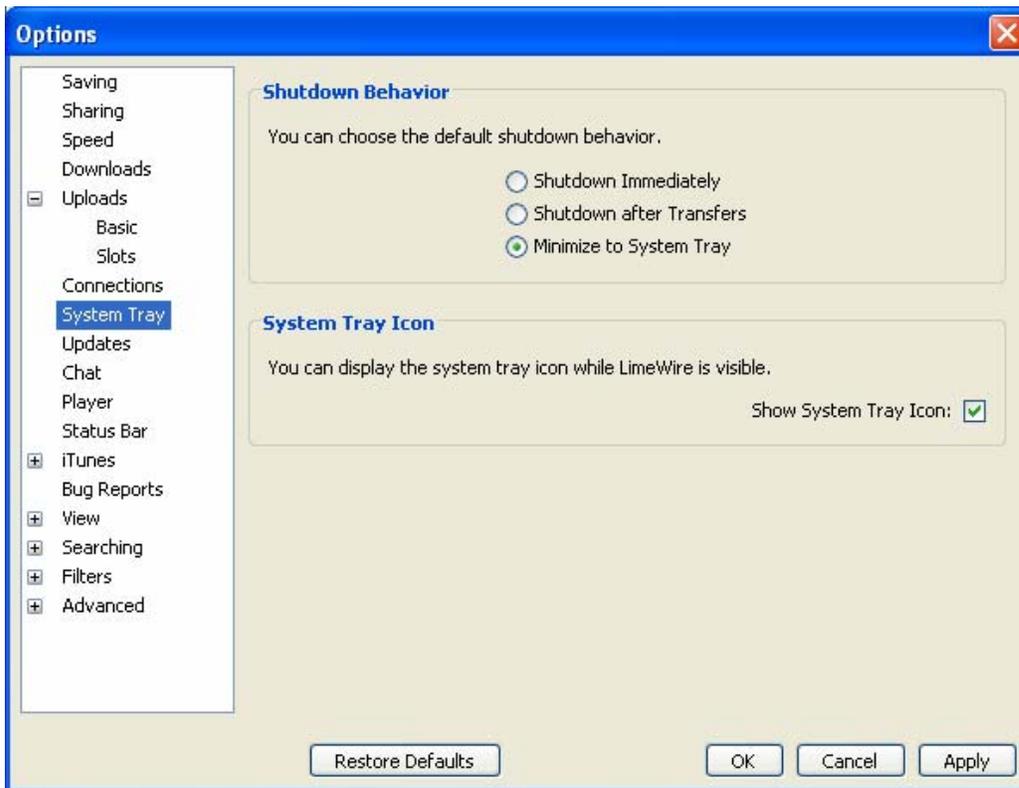
OPTIONS – UPLOADS- BASIC



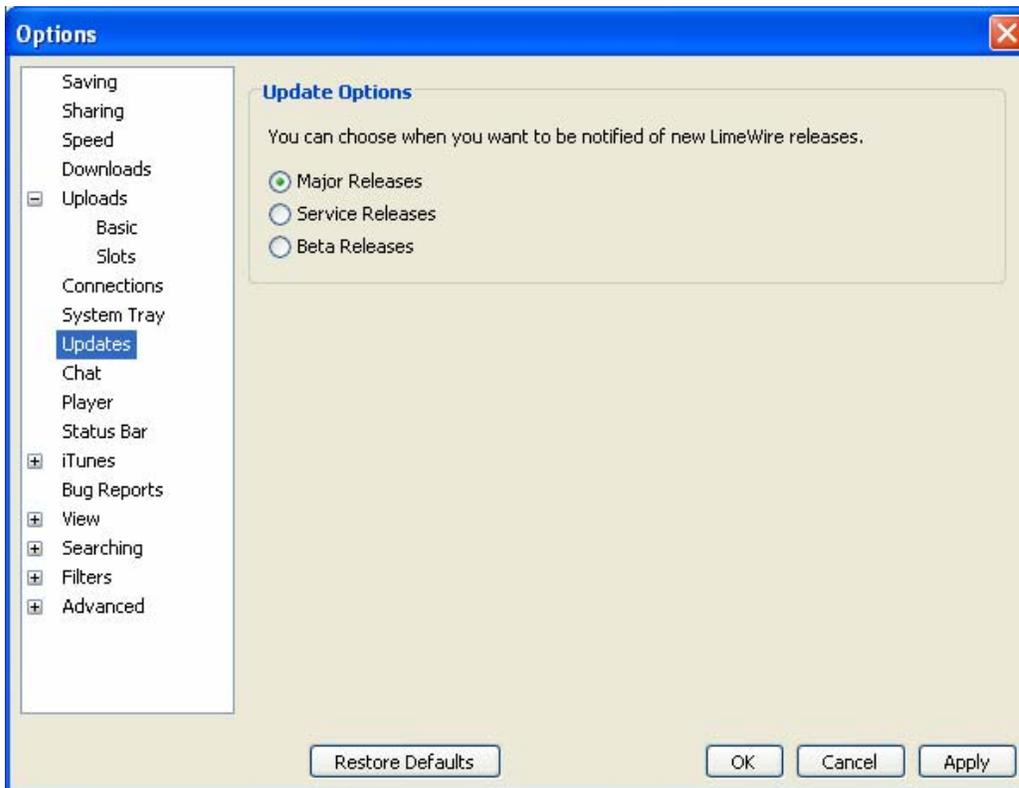
OPTIONS – UPLOADS – SLOTS



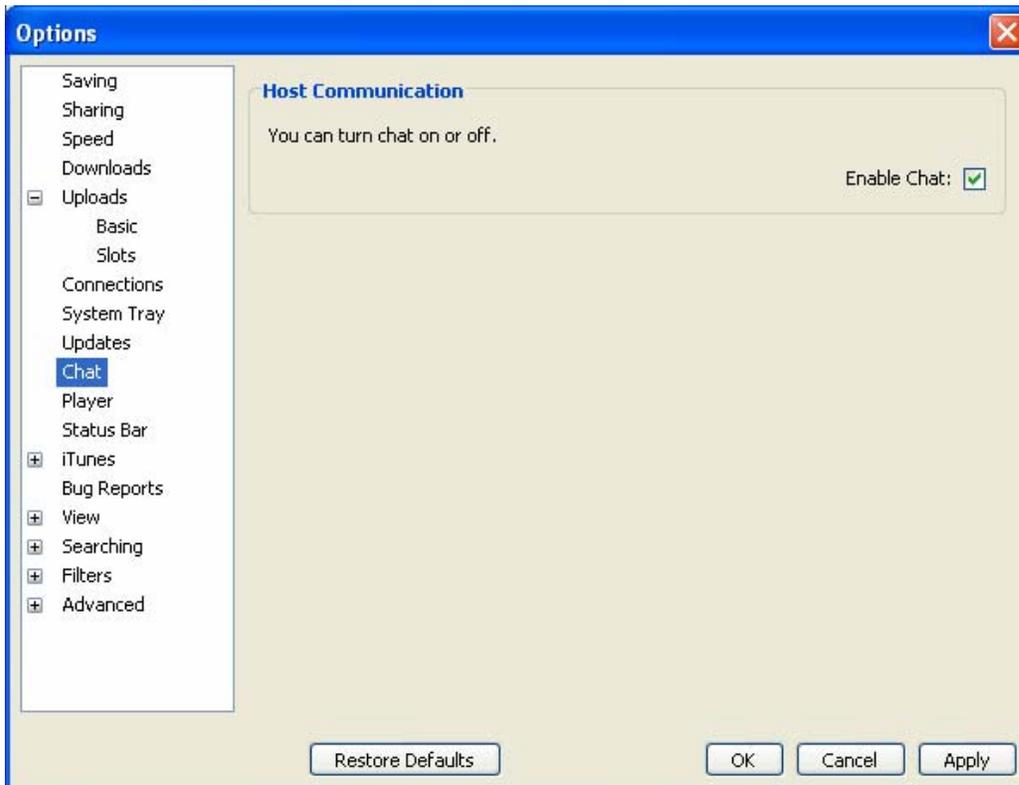
OPTIONS – CONNECTIONS



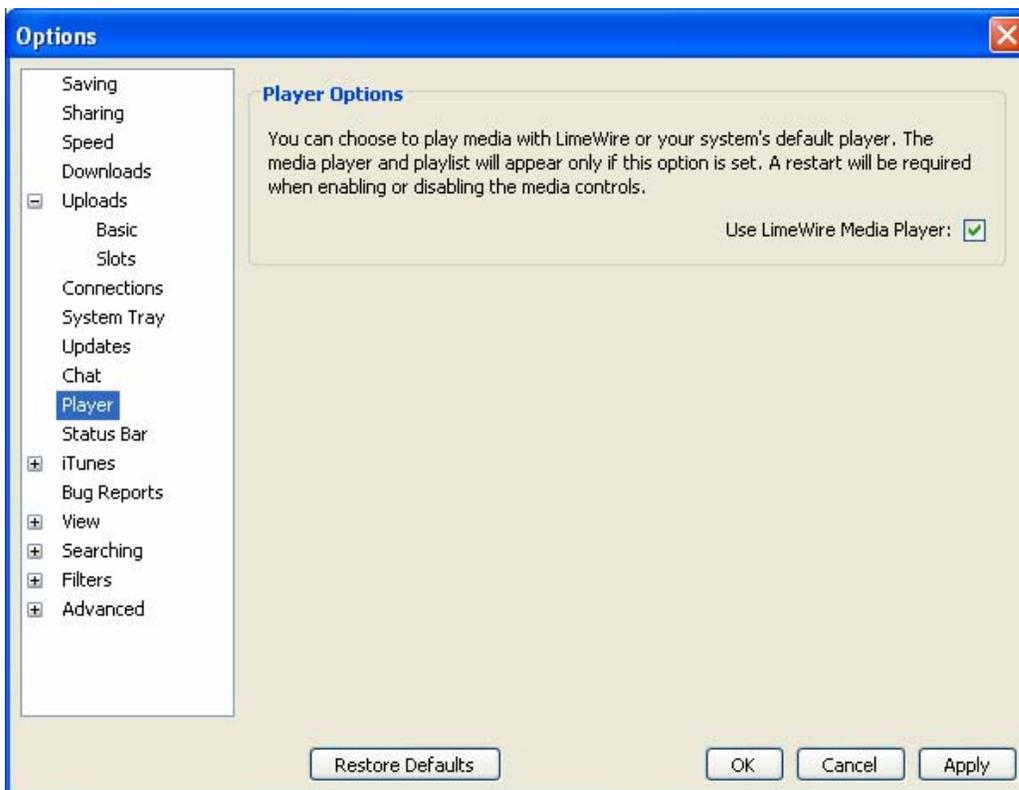
OPTIONS – SYSTEM TRAY



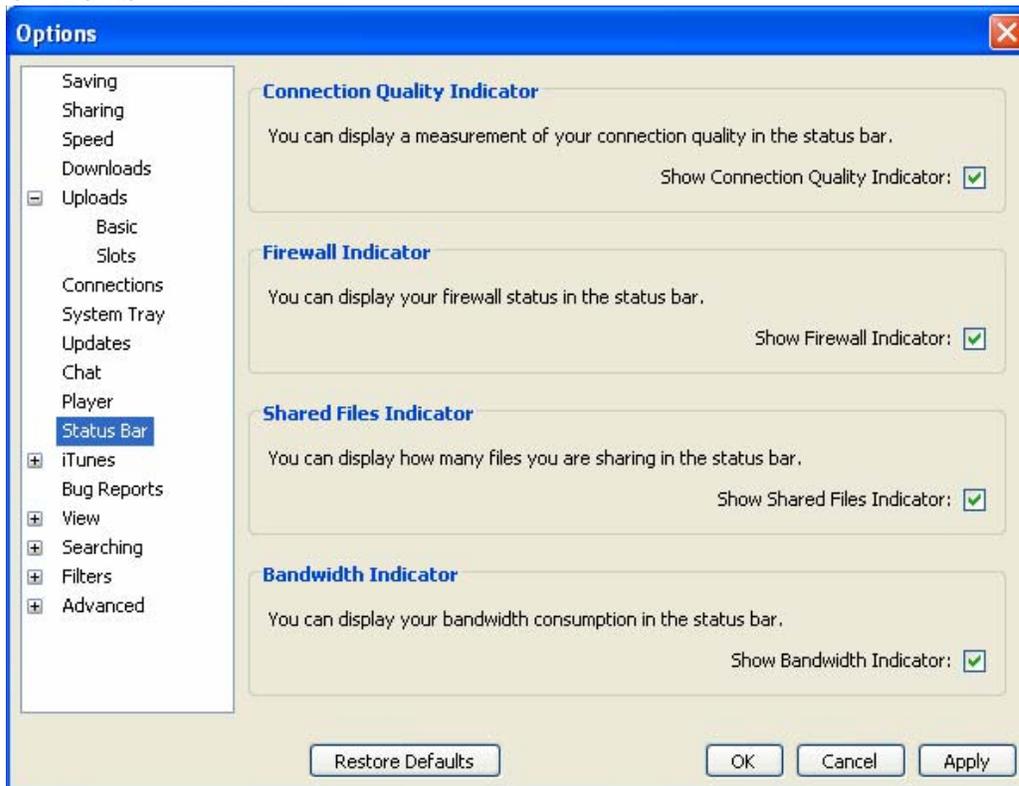
OPTIONS – UPDATES



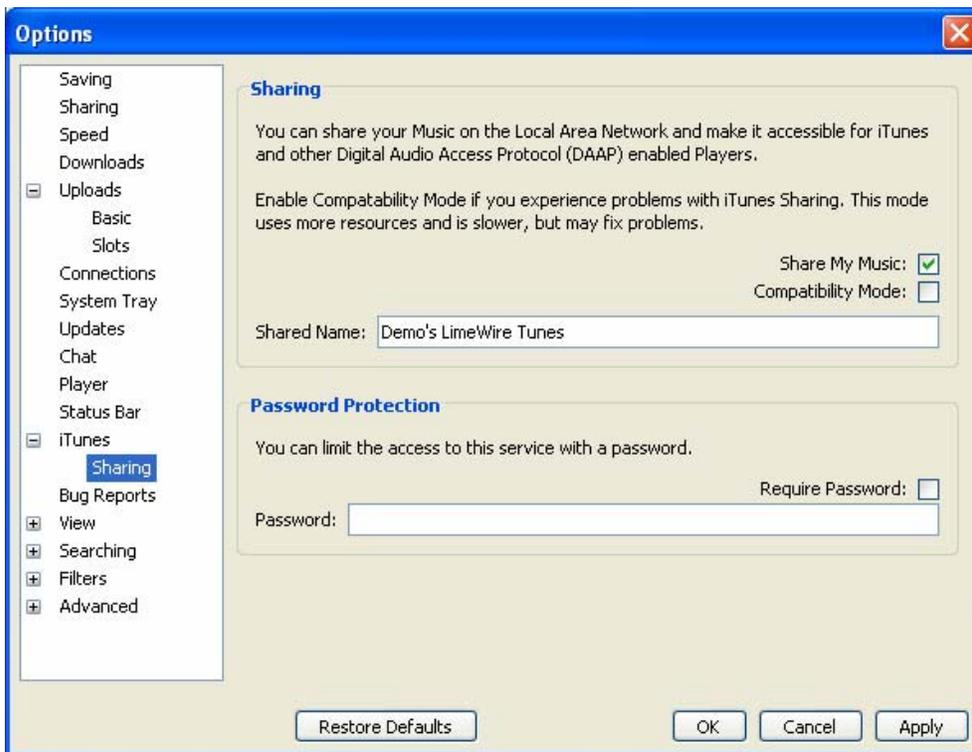
OPTIONS – CHAT



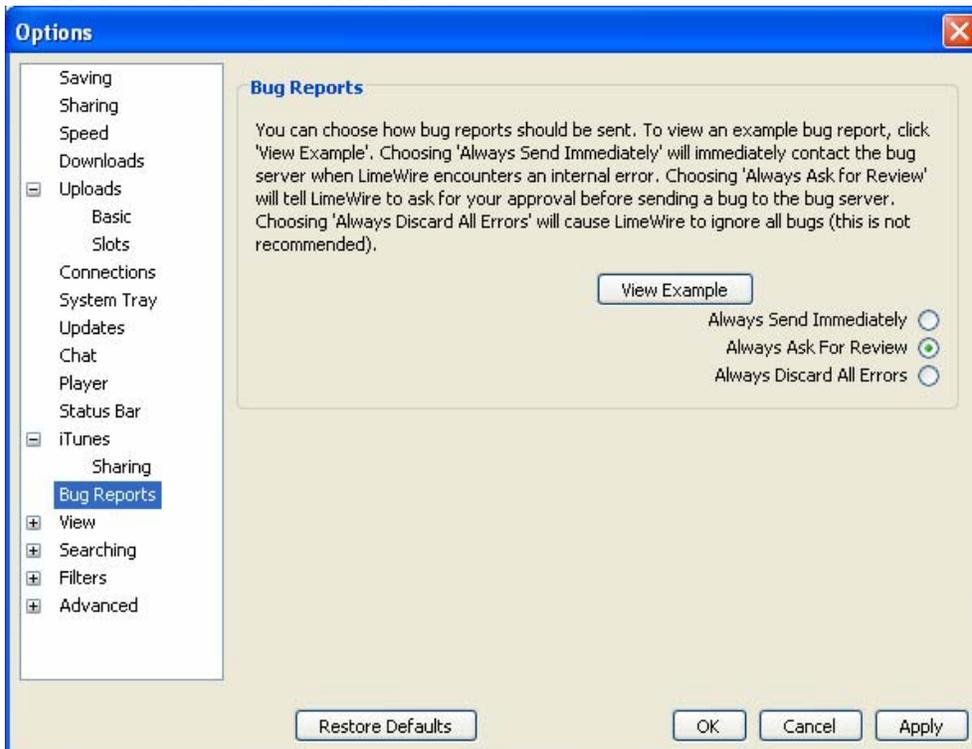
OPTIONS – PLAYER



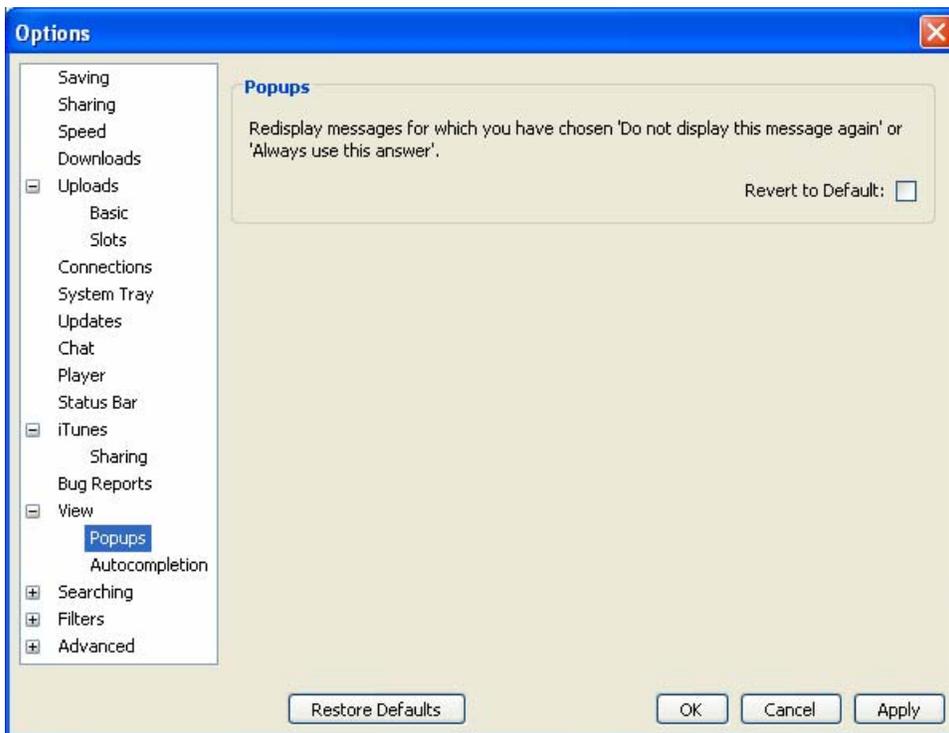
OPTIONS – STATUS BAR



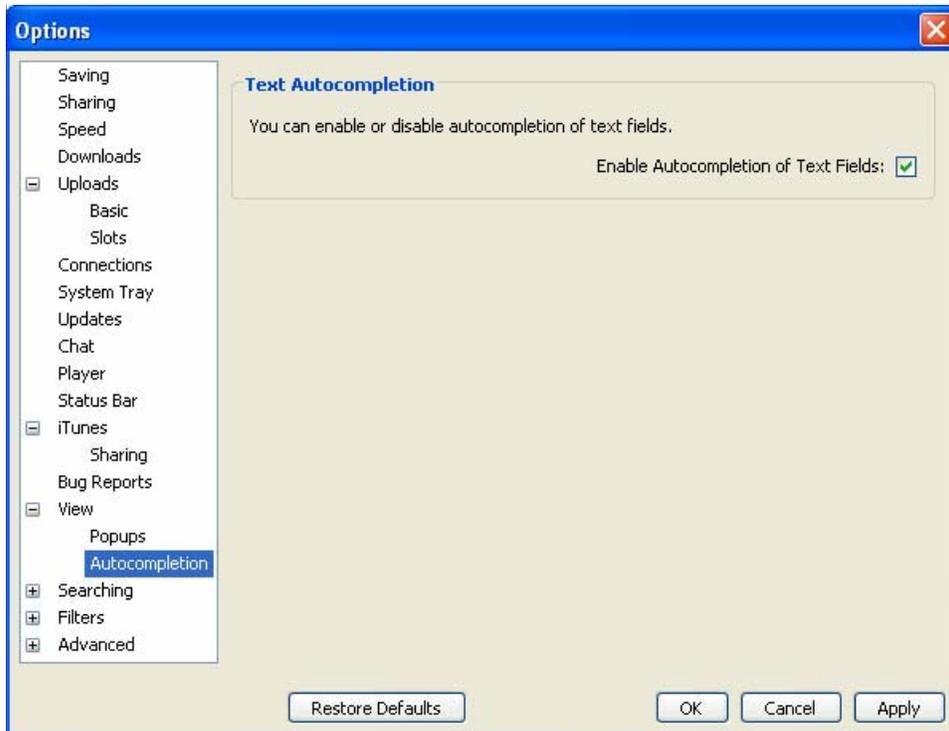
OPTIONS – iTUNES – SHARING



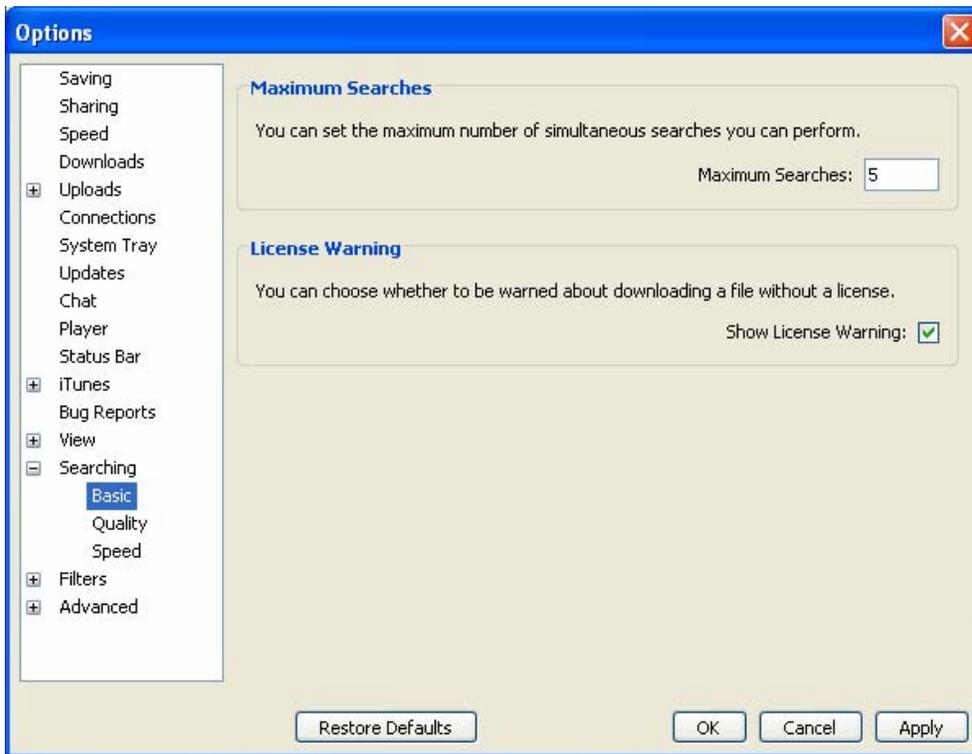
OPTIONS – BUG REPORTS



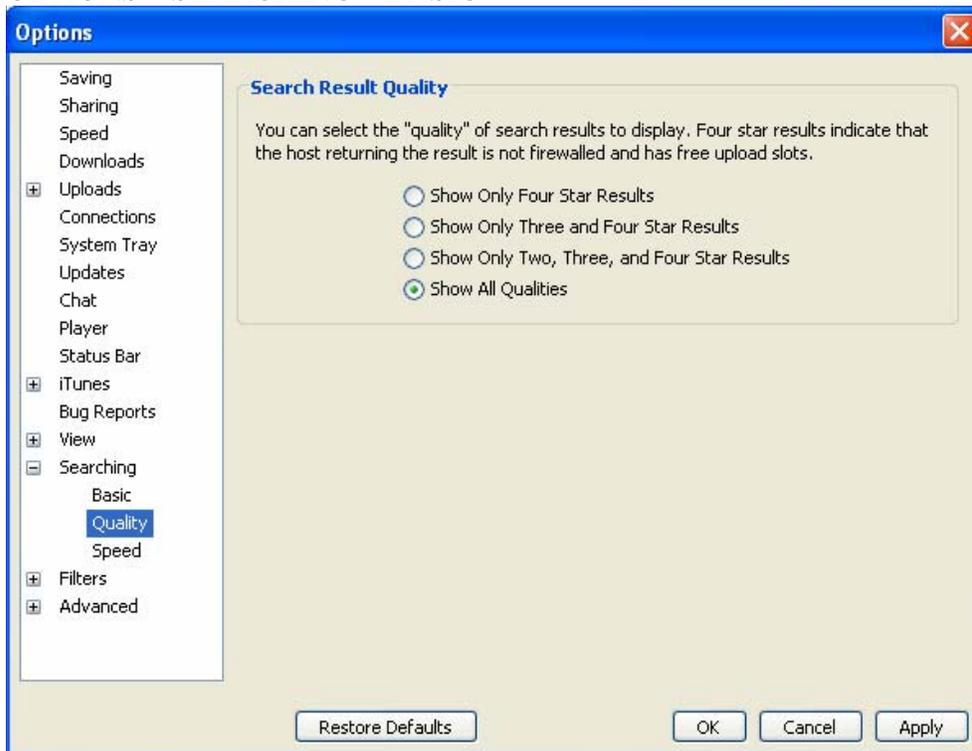
OPTIONS – VIEW – POPUPS



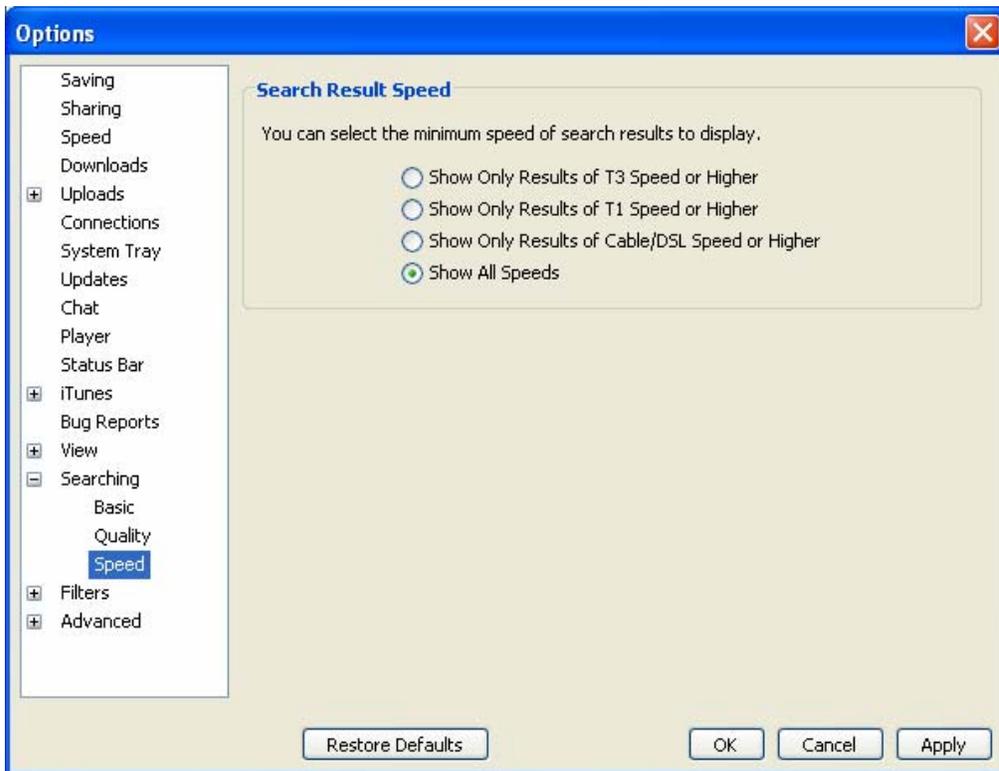
OPTIONS – VIEW – AUTOCOMPLETIONS



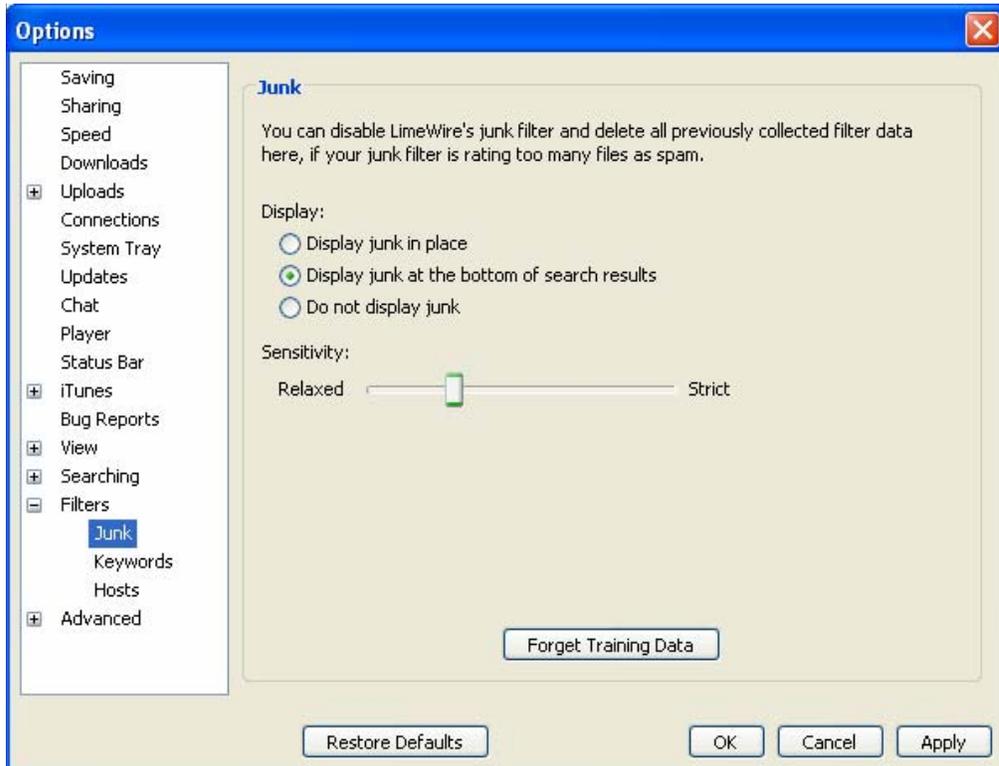
OPTIONS – SEARCHING – BASIC



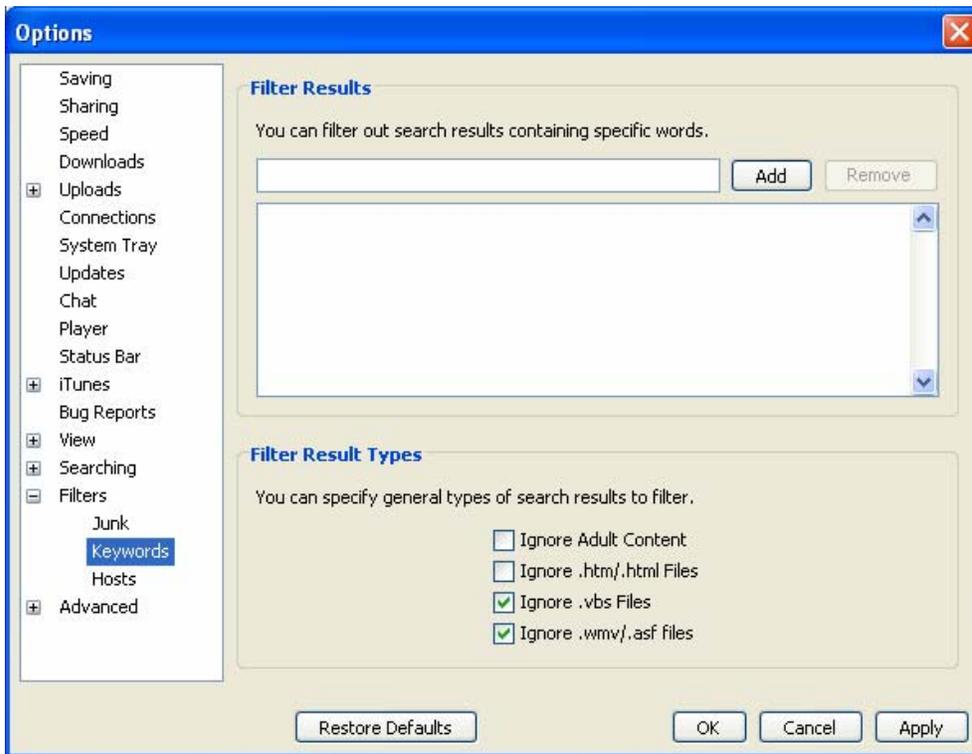
OPTIONS – SEARCHING – QUALITY



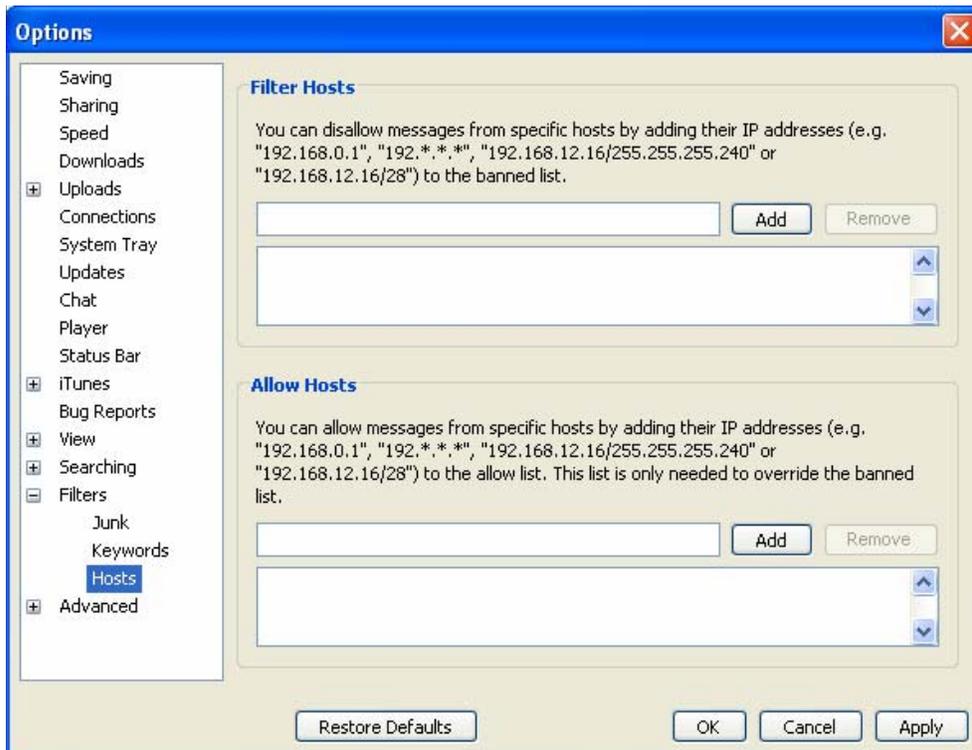
OPTIONS – SEARCHING – SPEED



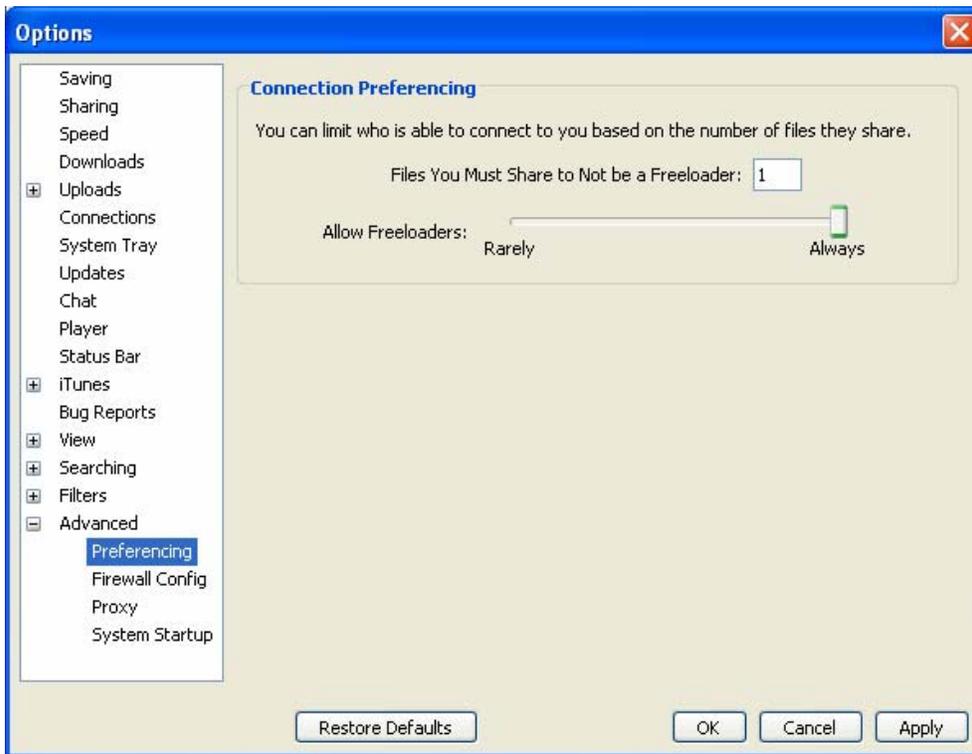
OPTIONS – JUNK



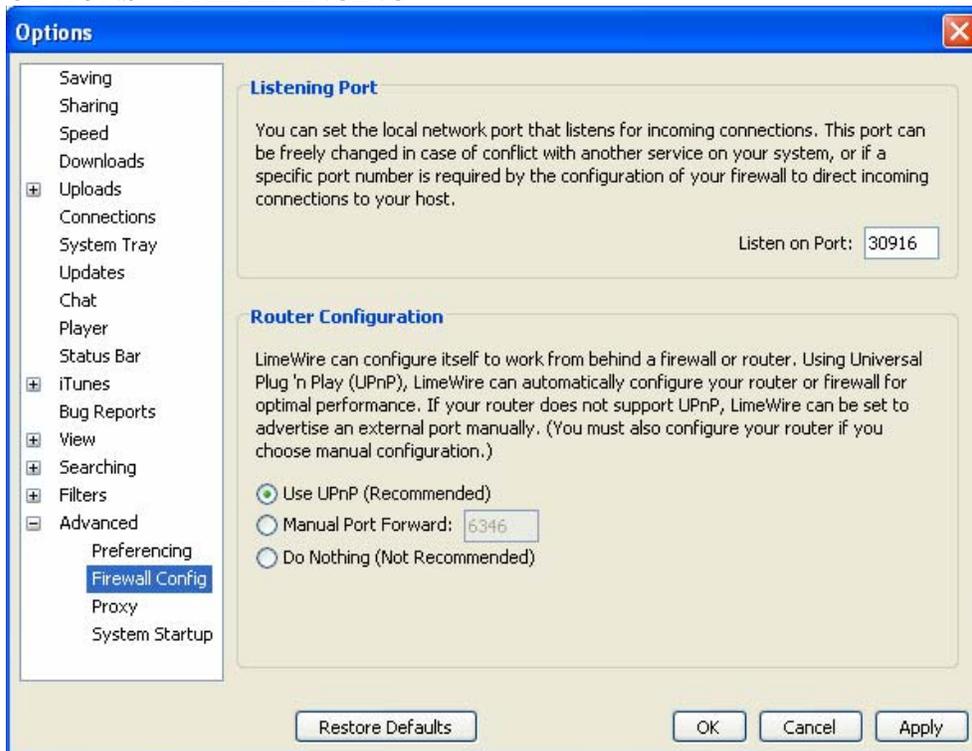
OPTIONS – KEYWORDS



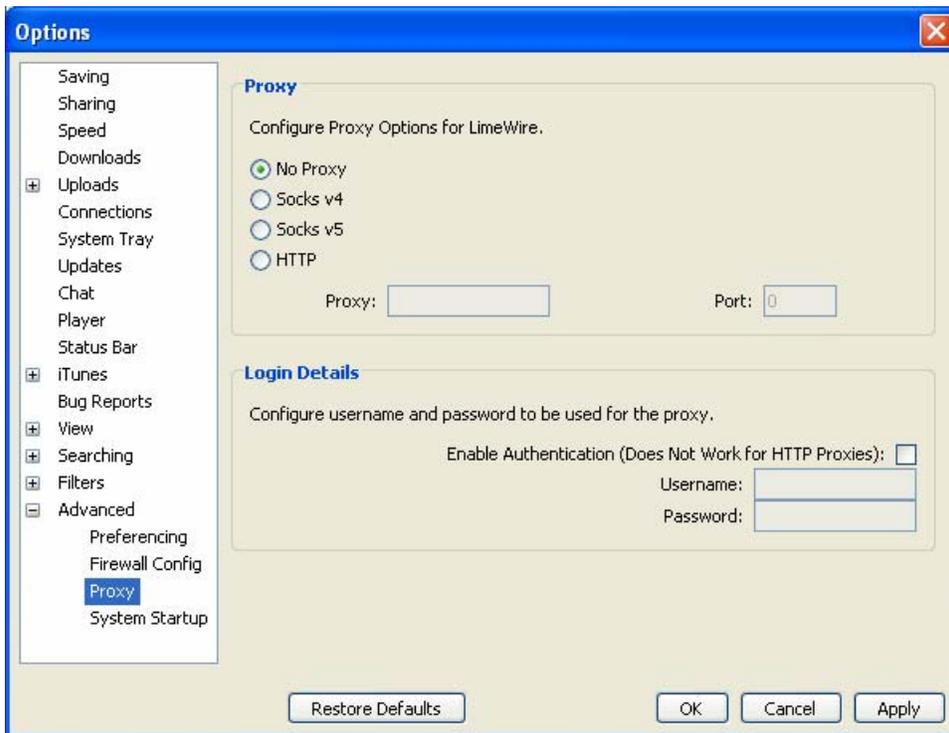
OPTIONS – HOSTS



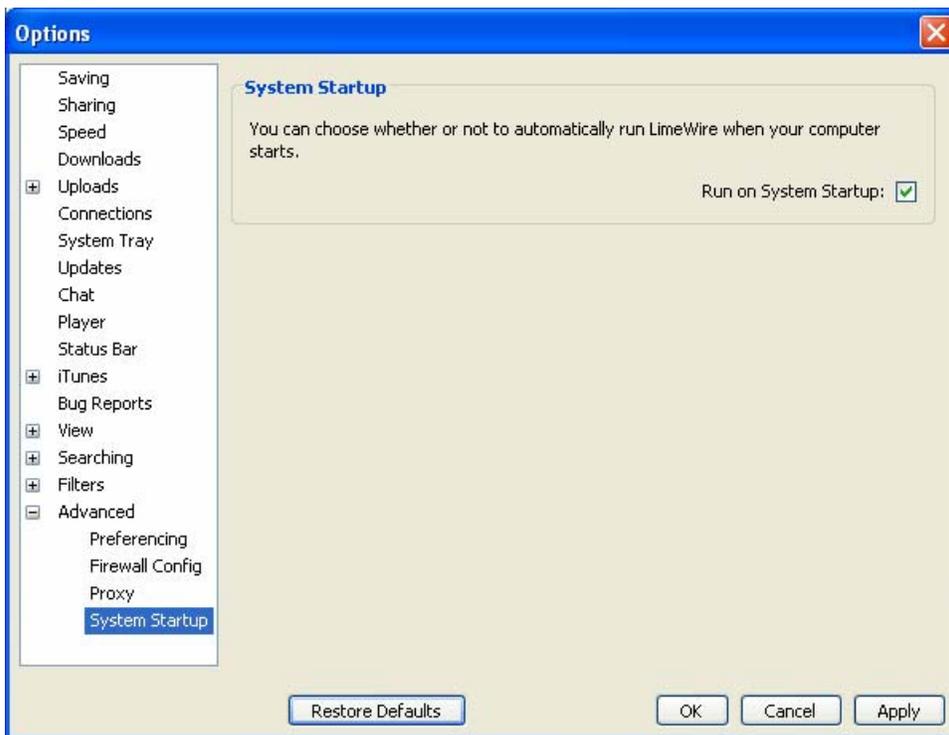
OPTIONS – PREFERENCING



OPTIONS – FIREWALL CONFIG



OPTIONS – PROXY



OPTIONS – SYSTEM STARTUP