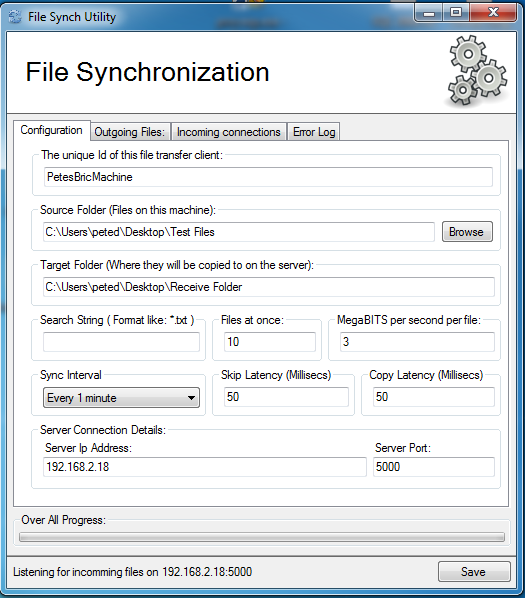
**AutoSync** (the File Synchronization utility)

AutoSync, or the File Synchronization utility was developed to overcome the problem of collecting massive amounts of video generated by BVS Branch servers in the field. After testing free and commercially available file synchronization tools, it was determined that a more powerful and robust solution was required.

AutoSync is a massively concurrent file transfer and synchronization tool which allows the user to finely tune which files, how many at once, and how fast each file will be transferred - as well as the more commonly expected aspects of file synchronization tools such as synchronization frequency, source and target file locations, etc.

AutoSync integrates client and server functionality into a single utility, to make two way synchronization and transfer of files to the remote branches possible. This document will outline the configuration of AutoSync.



On the configuration tab, we have the client configuration. This is where we will configure how this utility will connect to the remote server, which files will be sent, and where they will be placed on the remote machine.

\* There is a lot of room for improvement here. As of this writing, only one source folder can be selected, and one target folder. A massive improvement would be to change this from a single source and target folder, to the ability to add source and target folder combinations, or synchronization JOBS. This way we could specify multiple source file folders, and where each set of files should be copied TO on the server.

**The configuration Tab**

**Unique ID**: At the top of the configuration tab, a Unique ID for this machine must be specified. The name must be unique on the *remote server*.

**Source Folder**: Beneath that, your source files folder must be specified. This is the location of the files on this machine that must be copied to / synchronized with files on the remote server.

**Target Folder**: Next, we specify the target folder. This is a folder on the remote server where these files will be copied to / synchronized with. Files copied to this folder will be stamped with the identical creation and modification date as the source files, and every time AutoSync runs these stamps and the size of the file will be compared. If the remote file has a different creation or modification stamp then the source file, or it is a different size, it will be overwritten with the file in the source folder.

**Search String**: Enter one or more search strings here as you would on a command line. Separate them with the or (|) symbol. Each search string you add will INCLUDE that set of files in your set of files to be sent. For example, the search string "\*.txt|\*.exe" entered into the search string text field ***without the quotes*** will cause all .txt files and all .exe files to be included in the list of files to be synchronized / sent to the server, and all others to be ignored. Leaving this field blank is the same as entering the search string "\*.\*".

**Files At Once**: Enter the number of files you would like to synchronize at once, or concurrently. While choosing the value for this setting, you need to consider how much bandwidth you want to allow for each transfer and how much CPU you want this application to consume. Setting this higher than 10 will generally consume more CPU resources then necessary, and may cause increased CPU usage on the server also.

**Megabits per second**: This value is used to control how much bandwidth each transfer engine is permitted to use while transferring a file to the server. Values should only be whole numbers between 0 and 1000 (gigabit). Remember that this is the amount of bandwidth each transfer engine is permitted to use, so when you are calculating how much bandwidth you will allow the file transfer tool to consume you much multiply this value by the number of concurrent transfers. For example say you wanted to allow the file transfer tool to use only as much as 10 Megabit to transfer everything. You could configure Files at once to be 2 and Megabits per seconds to be 5, allowing a max of 10 total megabits. Or Files At Once could be set to 5 and Megabits set to 2 for the same max bandwidth. Each scenario has potential benefits. Scenario #1 (FilesAtOnce = 2 and MegabitsPerSecond = 5) will copy large files faster, but take longer to work through a large set of files. It will also take longer to work through files that have already been transferred (checking with the server to see that they have been transferred and that they have the correct file size and time stamp). Scenario #2 (FilesAtOnce = 5 and MegabitsPerSecond = 2) will work through a large list of small files much faster, and work through a large list of files that have already been transferred much faster also.

**Sync Interval**: This value determines how long the transfer tool will wait between synchronization events. It is NOT how often files are synchronized. For instance, if this value is set to "Every 60 minutes", the file sync tool will wait 60 minutes between synchronizations - no matter how long it takes to do the synchronization.

**Skip Latency**: An amount of milliseconds to wait after a transfer engine has asked the server if it should transfer a file, and the server has said "No, I have that one already". Typically, this tool will need to work through a large number of files that have already been transferred on a regular basis. A transfer engine can send hundreds of transfer requests receiving that "No, I have that one already" response from the server before finally hitting upon one that needs to be sent. If we don't inject a short wait period after these polite rejections, the transfer tool will do this very quickly and use up a significant amount of CPU recourses during the process. The higher the value of Files At Once, the more CPU is used and the faster the file sync tool will work through a large list of files that have already been sent. *Set this value higher to control the CPU usage, and lower to use more CPU but process synchronization jobs faster*.

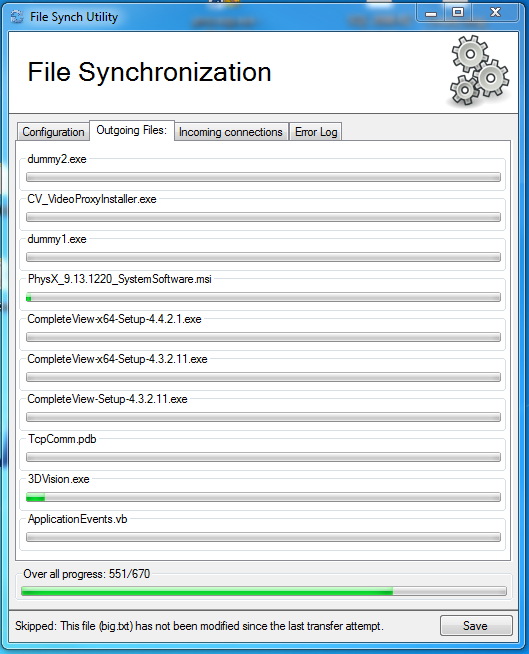
**Copy Latency**: This is the amount of milliseconds to wait after a file has been transferred. Setting this value higher will reduce CPU usage while transferring large numbers of small files, but slow down the overall transfer. Setting it higher will speed up transfers of large numbers of small files while using more CPU resources.

**Server Connection Details**: Settings in this group box pertain to the REMOTE server that this tool will attempt to connect to.

**Server Ip Address**: The IP Address of the remote server containing the target folder that files will be copied to.

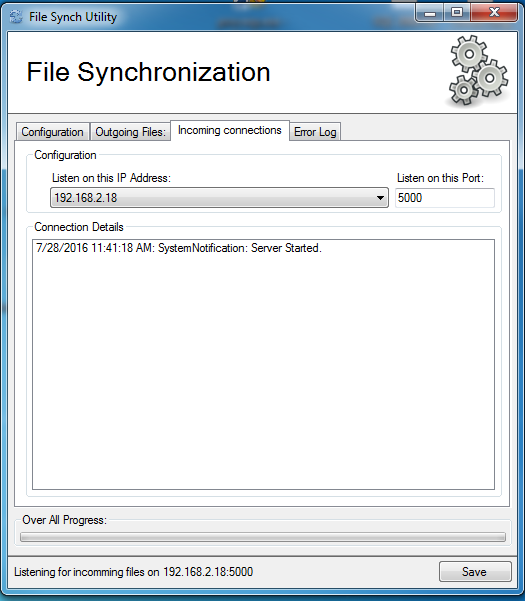
**Server Port**: The port number (between 1 and 65535) that this tool will attempt to use when connecting to a remote server to send files.

**The Outgoing Files Tab**



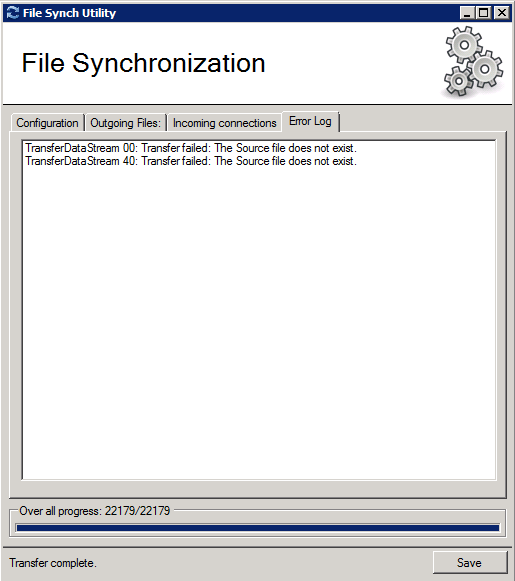
This tab will be blank unless the file sync tool is configured to send files, and is actively sending files to a server. If it is sending files, a progress bar will appear for each transfer engine (determined by the value set in Files At Once on the configuration tab), containing the name of the file being transferred and the amount of progress transferring it represented by a progress bar. Skips (files that the server has determined do not need to be transferred) will happen very quickly, and will show no progress.

**The Incoming connections tab**

****

This is where configuration for the integrated file synchronization *server* can be set, and information about other file synchronization tools that have have connected to this one can be found. Other synchronization tools will have a unique id, and when they connect to send files a time stamped entry will be seen in the Connection Details window. When they complete, a disconnect entry will be seen also.

**The Error Log Tab**

****

The error log tab lists all errors generated while attempting to send files to the server. Right now, you may see errors like failed transfers because the source file does not exist (deleted by salient), or transfer stream disconnected errors. You may see file corrupt errors during transfers also. These are errors due to bugs that have been caught, and the transfers retried.

At the moment, no known bug will cause the synchronization / transfer to fail - only files to be retried.