

NodeShark



User's Guide

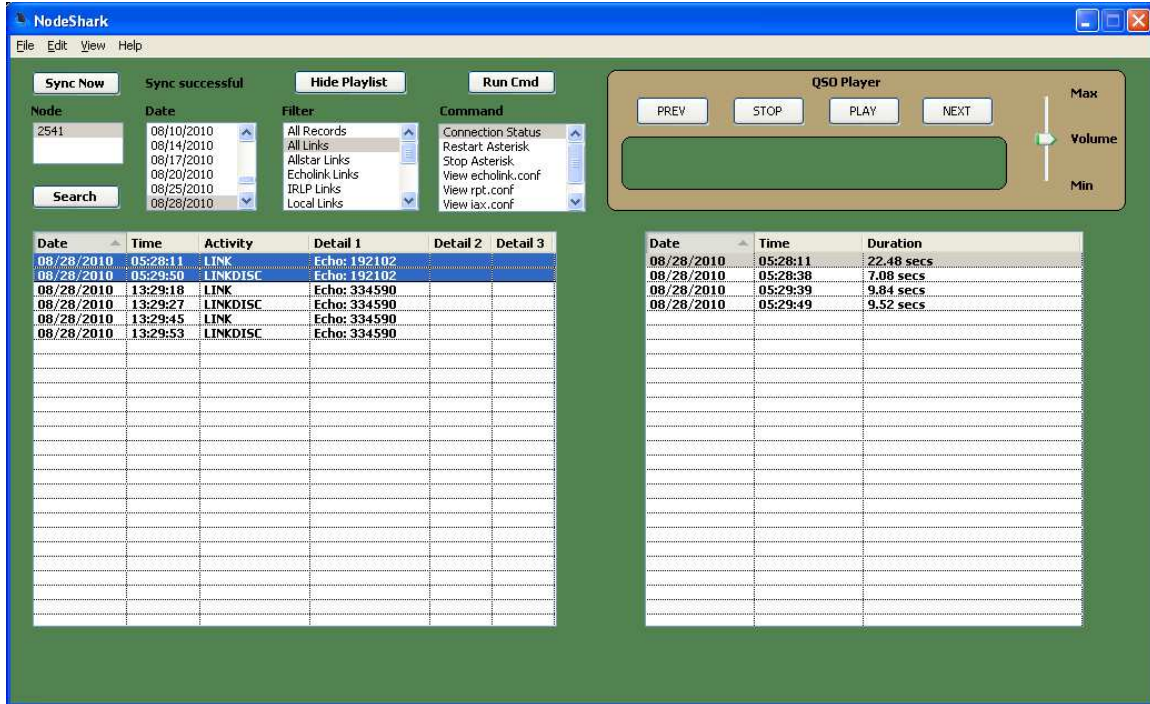
v0.95

Keith Williamson
KF7DRV

Page Intentionally Left Blank

1 Overview

1.1 What it is



NodeShark is a simple application that allows you to easily cull through and read selected, formatted versions of AllStar logfile records. You can then select ranges of those records and playback associated recordings of node transmissions.

NodeShark also makes it easy to do full and incremental backups of your AllStar node's logs and wav files.

Finally, new with v0.95, NodeShark includes a convenient and customizable remote command execution facility which allows the user to create, name, save, view, edit and launch shell scripts from a simple pull down list in the main application window.

1.2 Intended Audience

NodeShark is intended for (and really only useful to) AllStar Node owners and operators.

1.3 System Requirements

1.3.1 Windows

NodeShark is currently supported on a number of Windows variants.

It should work on any of the following:

- Windows 2000
- Windows XP 32-bit and 64-bit
- Windows Vista 32-bit and 64-bit
- Windows 7 32-bit and 64-bit

I have tested it on the following:

- Windows XP 32-bit
- Windows Vista 64-bit
- Windows 7 32-bit
- Windows 7 64-bit

With a few changes and a significant amount of testing, I can build a Linux version but I'll wait to see if there is any demand for that.

1.3.2 PuTTY

In order for NodeShark to retrieve a local backup of the "archivedir" directory, NodeShark uses PuTTY's PSCP (PuTTY Secure Copy). NodeShark also depends on PuTTY's Plink remote execution command to enable the remote shell script feature.

PuTTY is a very popular, FREE and stable implementation of SSH, SCP, Telnet, etc for Windows and I'm sure many of you already have it installed on your Windows platform to allow you to SSH into your node. If you haven't installed it, you can download it for free here:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

1.3.3 AllStar Node Requirements

1.3.3.1 Network Addressable

For normal operation of NodeShark, the target AllStar node must be network reachable (via SSH) from the system NodeShark is installed on. This can be either directly, if NodeShark is installed on the same LAN as the AllStar node, or via routing through a public IP address if on separate networks.

1.3.3.2 Archivedir Enabled

In order for NodeShark to be of much value, you must have enabled the "archivedir" directive in AllStar's rpt.conf file. The "archivedir" directive causes app_rpt to write out daily log files showing detailed information about inbound and outbound Internet connection attempts (e.g AllStar, Echolink, IRLP, and local IP phones), keyups, node restarts, DTMF key processing, etc) along with compressed wav files of each incoming and outgoing transmission. This information is great for maintaining your awareness of the usage of your node and gives you the ability to listen to the sound quality and monitor your node for inappropriate use.

The rpt.conf file is typically in the /etc/asterisk directory on the node. To enable the "archivedir" feature, just uncomment that line in rpt.conf and add the full pathname of the parent directory you want the node's (or nodes') log files and wav files to be stored (e.g. archivedir = /etc/asterisk/qso). Then simply do a Linux "mkdir" command to create the directory. The "archivedir" feature embraces the possibility that you may be controlling multiple nodes with a single instance of AllStar so, to keep multiple nodes' log files separate, you must create subdirectories of the "archivedir" directory for each node and name those subdirectories using each node's number. Even if you are only controlling a single node, you must follow that convention. For example, in my case, I created a parent directory, /etc/asterisk/qso, and then created a subdirectory below that of 2541 (my AllStar node number).

After making the above changes, restart asterisk to enable the feature by doing an "astres.sh".

1.4 Features

1.4.1 Formatted display of log records

If you've ever used "vi" to view an AllStar log file, you'll know that each record is made up of a comma-separated list of fields starting with a 14-digit date/time field. It's a very efficient and structured approach to storing log information but it's not easy for us humans to read. Fortunately, that nice structure is perfect for software to parse and then reformat into a much easier form for human eyes. NodeShark formats and displays log records in adjustable width columns and translates the record fields into more human friendly terms.

1.4.2 Flexible display-filtering of logs

AllStar's logging feature stores a great deal of detailed operational information, much of which you may not be interested in at any given time. To allow you to just display the kinds of records you're interested in, NodeShark uses display filters. NodeShark provides a number of pre-configured filters (e.g. All links, AllStar-only links, Echolink-only links, etc) but also provides a simple facility to create your own custom filters.

1.4.3 Point-and-click playback of AllStar node transmission recordings

AllStar's transmission recordings are kept in the same directory as the log files. The transmission recordings use the WAV container format and are named using a 14-digit date/time (YYYYMMDDhhmmss.WAV). Finding a desired transmission recording (or range of recordings) is tedious using tools like "ls" and "grep". NodeShark makes this very easy. Simply select a displayed log record or a range of displayed log records and NodeShark will quickly generate a playlist of all of the transmission recordings that correspond to that selected range of log records. Now just hit the "Play" button on the integrated QSO Player to listen to the transmissions. Or click on any of the recordings on the generated playlist to skip to that particular recording. You can pause, resume, skip forward to the next recording or skip back to the previous recording. The integrated QSO Player also features a slider to change the volume during playback.

1.4.4 Full and incremental backups of your AllStar node logs and transmission recordings

NodeShark operates on a local copy of your node's log file and transmission recording archive. This provides fast, consistent performance when filtering through the logs and playing back transmissions. It also ensures that you have a backup of your logs and transmissions should the node's disk drive ever fail. In addition, some AllStar nodes use small flash memory-based filesystems that discourage their operators from enabling the "archivedir" feature. With NodeShark, these operators can enable the archiving feature

and use the backup (or "Sync") feature to regularly retrieve the latest archived files and then delete them from the node's filesystem.

When you install NodeShark and perform the first "Sync", it can take quite some time to complete depending on how long your node has had the feature enabled and the network throughput between your Windows system and the node. Subsequent "Sync"s are very fast though since NodeShark only backs up files on the node that are newer than those in the local NodeShark archive.

NodeShark also provides a Command Monitor window that allows you to view the progress of the backup and an optional "verbose" setting that allows you to see more detailed information in the Command Monitor for connection debugging.

1.4.5 Remote execution of user-defined commands

This feature, which is new with version 0.95, allows the user to easily select and remotely execute commands on the AllStar system. These command scripts are created by the user with the new custom command editor. The custom command editor window allows the user to view, create, name, edit and delete command scripts. These can be any non-interactive shell scripts which are valid for the AllStar CentOS 5 based system. Some example use cases would be scripts which stop Asterisk, restart Asterisk or run Asterisk CLI commands to retrieve node status, etc (e.g. asterisk -rx "rpt stats 2541").

1.4.6 Customizable NodeShark window colors, sizes, and screen positions

Everybody has different tastes in colors, so NodeShark provides several different pre-installed color themes to choose from. If none of those is to your liking, you can create a custom color scheme. You can also move and resize many of the NodeShark windows to your liking and then do a File->Save Preferences so that the next time you fire up NodeShark, it will be sized, placed, and colored the way you like it.

1.4.7 Very simple, light-touch installation that doesn't touch your Windows registry

Although this initial release of NodeShark is for Windows only, I wrote it using a cross-platform development tool that can generate binaries for Windows, Linux, and Mac. There is only a single binary file and a folder that contains three DLL files. I wasn't interested in writing a Windows installer since 1) it wouldn't be portable across platforms, 2) I'm not interested in mucking with people's Windows registry and 3) there are so few files a formal installer isn't necessary. This way it's also very easy to uninstall..just delete the binary and the NodeShark folder with the three DLLs.

1.4.8 Very responsive Windows native binary executable

I developed NodeShark using Real Studio 2010. Real Studio has been around since 2003 and was initially known as Real Basic. It started as a cross-platform answer to Microsoft's Visual Basic and has grown over the years into a top-notch Integrated Development Environment for creating GUI applications that are event-driven, object-oriented and cross-platform. It produces efficient, x86 native binaries that perform exceedingly well. I briefly considered using an interpreter-based language like Java or Python but decided that the end application would likely not be fast and responsive enough.

2 Installation

2.1 PuTTY

2.1.1 Installing PuTTY

You will need to install the current v0.60 version of PuTTY and I highly recommend the Windows installer version (putty-0.60-installer.exe).

<http://the.earth.li/~sgtatham/putty/latest/x86/putty-0.60-installer.exe>

Simply download and run the installer. Afterwards, go into the Windows Control Panel-> System -> Advanced -> Environment Variables, and add the install location to the PATH spec (e.g c:\Program Files\PuTTY). This way, the PSCP feature can be run from any local directory on the Windows system.

2.1.2 Testing PuTTY

After installing PuTTY, you'll want to test it by using PuTTY's PSCP to transfer a few files from the AllStar node to the Windows system. This ensures that your network connectivity is sufficient, that you are using the correct SSH/SCP port number expected by the AllStar node, that the SSH/SCP shared keys for your Windows system are stored on the AllStar node, and that you are using the correct AllStar login and password to access the archive directory on the node.

When you configure NodeShark for the first time, you'll tell it where on the Windows system the archive files will be found. If the folders don't yet exist on the Windows system, now is the best time to create them. Remember, you'll need to create a top level folder where all of the archives will be kept (you may be controlling more than one AllStar node with a single AllStar system). You can create this top-level folder wherever you please. I create a folder called "qso" at the root of my C: drive but again, you can create it wherever you like and call it whatever you like. After creating this folder, create a subfolder for each of your nodes below that top-level folder. I only have a single node, 2541, so I create a subfolder of C:\qso called "2541".

Now, to test PSCP, run CMD.exe and change directory to one of your node folders (e.g. C:\qso\2541). You can now test PSCP by typing (as an example):

```
C:\qso\2541> pscp -P 222 root@192.168.1.44:/etc/asterisk/qso/2541/20060822* . (note the trailing space and dot).
```

If this is the first time you have attempted to connect to your AllStar node from that particular system, the SSH/PSCP client will prompt you for permission to store the shared key for your AllStar system. Type "Yes" to allow it and it will then prompt you

for login followed by password. If those are accepted, any files starting with 20060822 (in the example above) will be transferred to your local archive. There probably aren't any that old but you'll have proven that your network connectivity is sufficient, that you are using the correct SSH port and IP address, and will have cached the shared key on your Windows system for the AllStar system you'll be accessing.

In my case, since I like to access my node from outside of its local network, I actually specify the public IP address for the network my node is on (my home router's WAN IP address) and a port number that I have setup in my home router that is NAT forwarded to the internal IP address and port number expected on my node.

Once you have successfully completed this initial PSCP test, you are ready to install and configure NodeShark. One more thing though; for NodeShark to be useful, you'll need to pull over the entire archive from your Allstar node. You can do that right now using PSCP or you can use NodeShark to do it.

If you want to do the initial backup using PSCP, just follow the steps you used above to test PSCP but change the file spec in the PSCP command from 20060822* to simply "*". If you have a lot of files in your AllStar node archive and/or your network throughput between your Windows system and your node is limited, you may want to kick off this PSCP command and take a long break or go to bed while it runs.

As I said, you can do this with NodeShark also. If you want to use NodeShark to do it, delete any local files that might have been pulled over during the initial PSCP testing since NodeShark only backs up files newer than the newest file in the local archive. This is how NodeShark automatically does incremental backups but you want to do a first-time full backup.

2.2 Installing Nodeshark

2.2.1 What's installed

The NodeShark zip file contains the application binary (NodeShark.exe), and a NodeShark folder containing three DLLs. These DLLs were generated by the development tool to provide the support needed for certain classes used in Nodeshark.

2.2.2 Where to install it

On 32-bit Windows systems, the NodeShark archive can be unzipped to wherever you like. After unzipping, you can right-click on NodeShark.exe and do "Send to Desktop" to create a desktop icon for it.

On 64-bit Windows systems, it's necessary to unzip the archive into Window's "Program Files (x86)". By installing NodeShark there, Windows knows that it's a 32-bit application.

2.2.2.1 Where the preferences file is stored

After NodeShark has been installed and you've configured any NodeShark settings, you'll save those settings using File->SavePreferences. When you do that, NodeShark writes your preferences into an XML file called NodeShark.cfg and saves it in the Windows "AppData" folder (sometimes "ApplicationData" folder) for the user you're running as.

2.2.3 Initial NodeShark Setup

NodeShark uses a local folder structure to store a backup of the archive folder that lives on the AllStar node system. Prior to running NodeShark for the first time, you should create that folder structure wherever you want it on your Windows system. You'll want to have this in place prior to running the first NodeShark "Sync" operation. If you followed the procedure above for testing PSCP, you probably already created the folder structure. If not, review the procedure (Testing PuTTY), and create that folder structure now.

Having done that, you can start NodeShark and do the initial setup. When NodeShark starts up without a configuration file, it comes up in default colors (the "Forest Service" theme) and the listboxes for Node and Date will be empty. Those list boxes are populated dynamically based on the node folders and log file dates found in the local archive. Since NodeShark doesn't know where those are yet, the boxes are empty. To configure NodeShark, go to Edit -> Preferences. The Preferences window will open and will initially be in the Target panel. In the Target panel, type in the fully qualified pathname of the top of your local archive folder structure (e.g. C:\qso).

Now click on the Backup tab. This panel contains the settings used for the backup or "Sync" operation. The fields are fairly self-explanatory:

IP Address is the IP address of your AllStar node relative to your Windows system (either local IP or public IP if you'll be going through a router).

Port is the SSH port number for your AllStar node. AllStar's default is 222, I believe, but you might have changed it. Also, if you are coming in from a router, you may have set up a port translation to going from the public (WAN) side of the router to the private side (LAN).

Login is a login on the AllStar node that allows read access to the "archivedir" directory. Frequently, folks just use "root".

Password is the password corresponding the login specified above.

Remote Path is the fully-qualified pathname on the AllStar node to it's top-level "archivedir" directory (should be the same as in rpt.conf).

Local Path is the fully qualified pathname on the local Windows machine that you setup for the archive backups (e.g. c:\qso). Don't include the node number folder as NodeShark will automatically discover that and add it whenever it needs to.

At this point, initial setup is complete. To save the configuration, click the OK button on the Backup tab and then do File -> Save Preferences. Now do File -> Exit and restart NodeShark. Once NodeShark has restarted, it should now show your node number (or numbers) in the Node list box. The first node number in the list will be selected automatically.

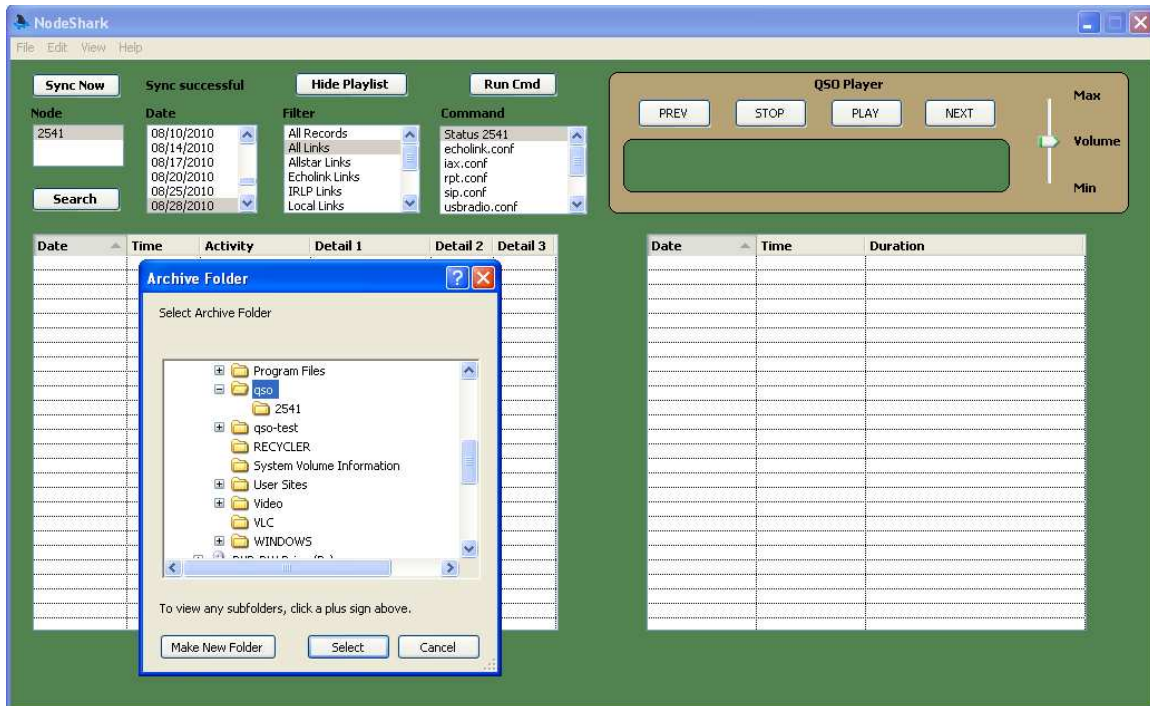
If you previously did a backup of your archive files using the procedure in the "Testing PSCP" section, you should also see a list of dates in the Date listbox with the most recent date selected. You can scroll the Date listbox top to bottom to see all of the dates of log files discovered in the local archive.

If you haven't done that first backup yet, the Date listbox will be empty and you can go ahead and start a Sync now if you like. If you want to monitor the progress of the Sync, go to View -> Command Monitor and enable it. The Command Monitor window will open. If you want detailed PSCP dialog information, you can go to Edit -> Preferences, and in the Backup tab, check the "Enable Verbosity" box. You need not do File -> Save Preferences unless you always want Verbosity set for archive Sync sessions.

Now, near the upper left corner of the main NodeShark window, click the "Sync Now" button. You should see PSCP start running in the Command Monitor window and be able to follow its progress. If you successfully tested PSCP as described earlier and entered everything correctly in the Preferences -> Backup tab, everything should work fine. If this is your first backup, it may take quite some time, so you may want to go do something else.

3 Usage

3.1 Archive Selection

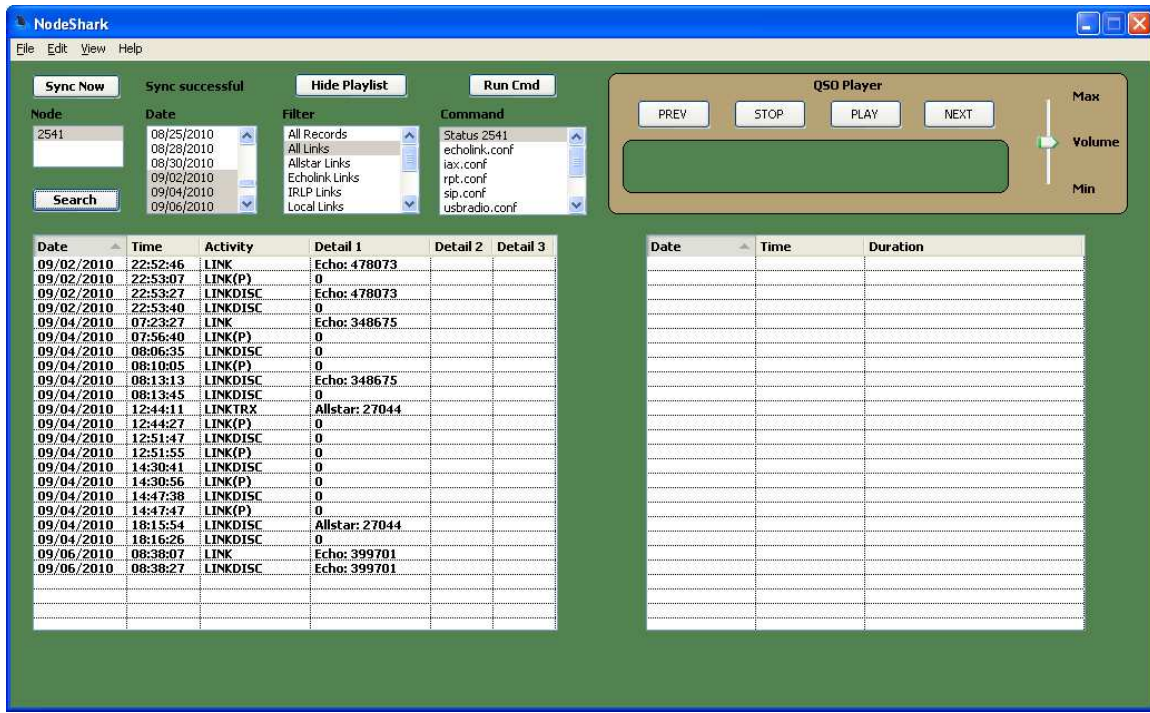


NodeShark can operate on any backup archive stored on the local Windows system. Besides having a backup of your primary AllStar node archive, you may have other archives stored on your system. Perhaps you are helping another AllStar node operator to debug an issue or perhaps you have split your archive into multiple archives based on calendar year. Whatever the case, you can select the current archive you want to investigate by going to File -> Open and then browse to the top level folder of the alternate archive. NodeShark will then open that archive, enumerate the nodes within that archive and populate those node numbers into the Node selection listbox. NodeShark will automatically select the first node listed in the Node selection listbox and then enumerate the log file dates and populate those dates into the Date selection listbox.

Opening an alternate archive with File -> Open will also populate that archive path into the File -> Preferences -> Target tab. As long as you don't then do a File -> Save Preferences, the next time you start NodeShark, it will open your usual archive. Of course if you did do a File -> Save Preferences, upon restart, NodeShark will automatically open that alternate archive. In that case, to revert to your primary archive, just do File -> Open, browse to your primary archive, and do a File -> Save Preferences.

Alternately, you can go to Edit -> Preferences -> Target tab and type in the path to your primary archive. Either way will work.

3.2 Analyzing Logs



When NodeShark has a valid archive open, the Node selection listbox will have one or more nodes to select from and the Date selection listbox will have some number of log file dates to choose from.

The first thing to do when you want to view logs is to select one or more dates in the Date selection listbox. You can click on a single date, select multiple dates by CTRL clicking each desired date, or select a range of dates by clicking on a start or end date and then dragging to select a range.

Once you have selected the dates you want to investigate, you can now optionally choose a filter. By default, the "All Records" filter (no filtering) is selected. Alternatively, you can narrow down the log records that will be displayed by choosing one of the pre-defined filters. Currently there are the following prebuilt filters:

All Records (no filtering done)

All Links (link connect, link disconnect, link failure, link monitor for all network types)

AllStar links (same records as above but just AllStar network)

Echolink Links (same records as above but just Echolink network)

IRLP Links (same records as above but just IRLP network)

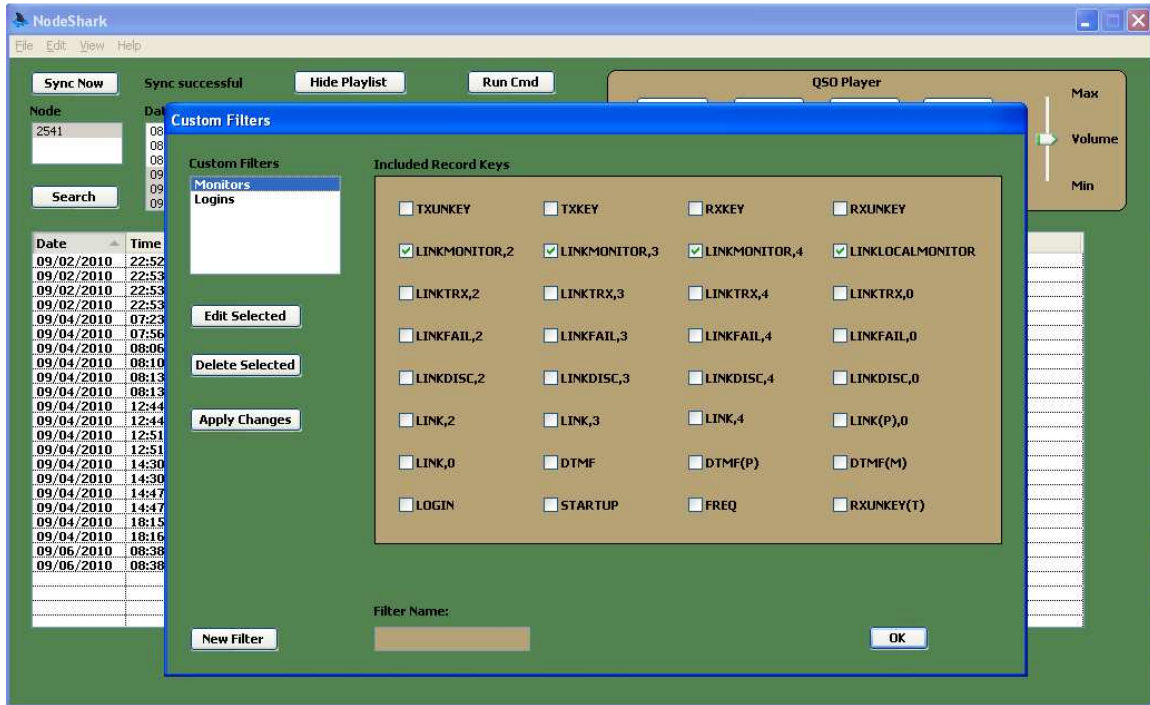
Local Links (same records as above but local VoIP calls only)

Startups (startup and frequency log records)

DTMF (DTMF digit processing log records only)

Having now selected the node, a range of dates, and a filter, just click on the "Search" button to find and format the relevant log entries.

3.2.1 Custom Filters



NodeShark also allows you to create an unlimited number of custom filters.

To create, edit, or delete custom filters go to Edit -> Custom Filters. The Custom Filters editor window will open. This window has a Custom Filters selection listbox that will be populated with all custom filters you have already defined. If you have any defined, the top most one will be selected by default and the Included Record Keys selection box will show which record keys you selected for that custom filter.

To create a new custom filter, click the New Filter button. The Filter Name edit window will be filled in with a default filter name of CustomX where X is a number that represents the next available ordinal number position in the Custom Filters selection listbox. You can certainly use that name for a quick and dirty filter definition, but you probably want to double-click on that name and enter a more descriptive name for the filter. Now to define the filter, just click on the desired record keys in the Included Record Keys box to enable those keys for that filter. When you are satisfied, click on the Apply Changes button and NodeShark will populate the Custom Filters selection listbox with the new filter and will also immediately add it to the Filter selection listbox on the main NodeShark window. The new filter can be used immediately.

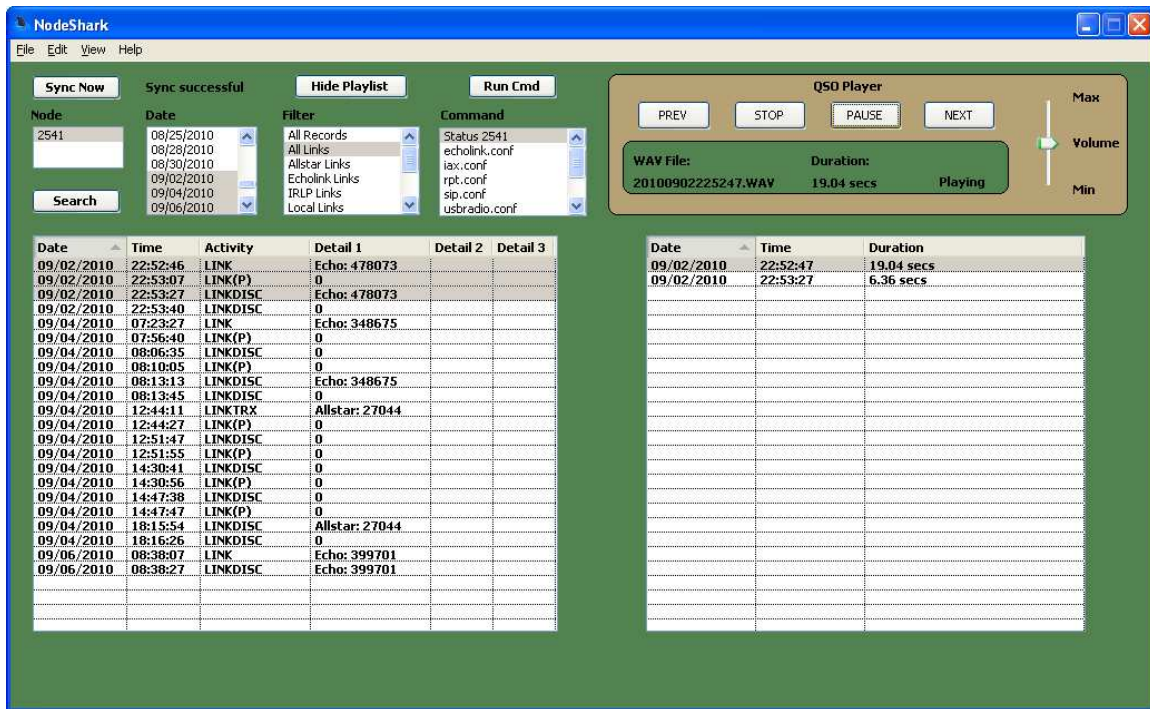
If you want the newly defined custom filter to be saved persistently, do File -> Save Preferences before shutting down NodeShark.

You can also review and edit existing custom filters you have defined. Simply select the custom filter you want by clicking it in the Custom Filter selection listbox. You will be in Review mode. The definition of the filter will be reflected in the Included Record Keys box (the included keys are the ones checked).

If you want to edit the definition of the selected filter, click on Edit Selected. You can now click or unclick the record keys to be included. When you are done, click on Apply Changes. Remember that to keep the changes persistently across NodeShark executions to do File -> Save Preferences.

Finally, to delete a custom filter, simply select the desired filter in the Custom Filters selection listbox and then click Delete Selected. The deletion will be immediate. To make the deletion persistent across NodeShark executions, remember to do File -> Save Preferences in the main NodeShark menu.

3.3 Playback of QSO Sound Files



A major feature of NodeShark is the ability to easily playback archived transmission recordings. The recordings are selected by first searching for and then displaying one or more log records that relate to the date/time and node activity of interest. To discover the transmissions that occurred during the creation of those log entries, you simply select a series of log records on the formatted log display. You can select these records by clicking on individual records or clicking and dragging across multiple log records. When you select one or more log records, NodeShark automatically generates a playlist of transmission recordings that occurred during the time period those log records were made. To view the generated playlist, click on the Show Playlist button on the main NodeShark window or go to View -> Playlist.

If there were transmission recordings during the date/time range of the log records selected, you'll see an ordered list of recordings with their date, time, and recording duration. To play them all, just click on the PLAY button on the QSO Player.

You can pause playback at any time. You'll notice that when you click the PLAY button and playback begins, the button's label changes to PAUSE. Now you can click it to pause the recording. As soon as you pause the playback, the button's label changes to RESUME. Click it to resume. As soon as you resume playback, the button's label reverts to PAUSE. When playback ends, the button's label will change back to PLAY.

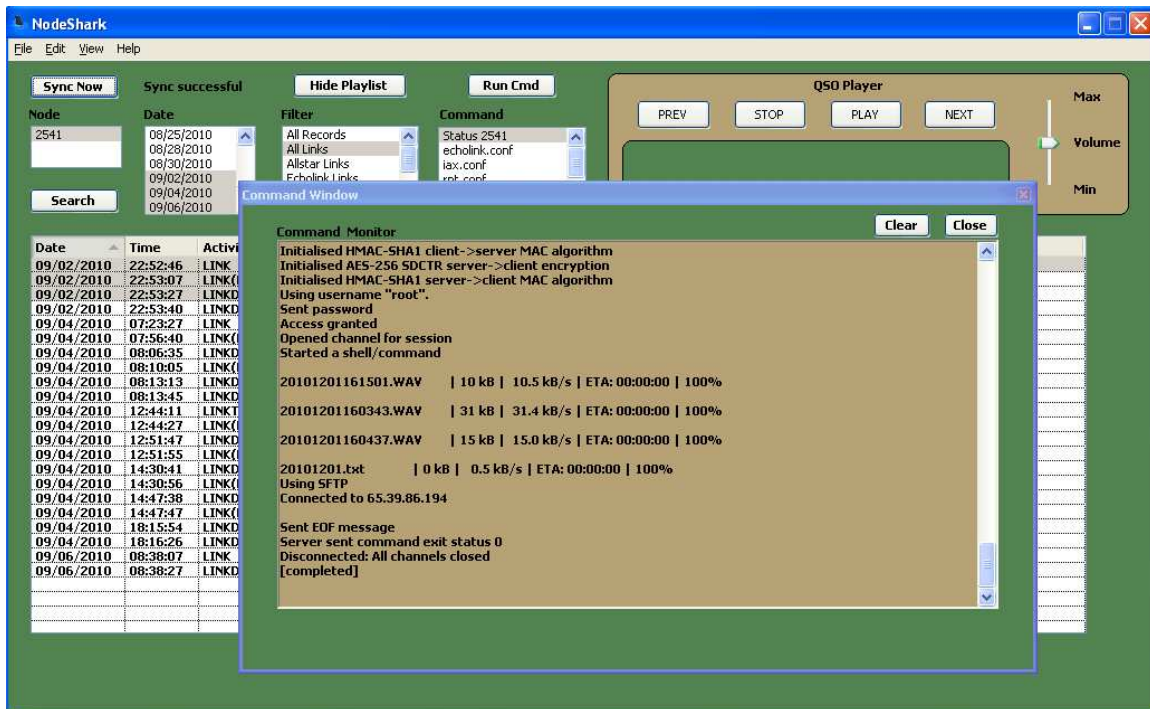
Playback will stop when either the end of the playlist is reached or you click the STOP button.

You can also skip forward a track at a time in the playlist by clicking the NEXT button or skip backward through the playlist by clicking the PREV button.

Additionally, you can skip directly to any track in the playlist by simply clicking on the track.

Finally, you can adjust the playback volume anytime using the Volume slider.

3.4 Synchronizing Archives



The Sync feature of NodeShark provides an easy, point-and-click way of keeping up-to-date local copies of your AllStar archives of log files and transmission recordings.

For performance reasons, NodeShark only operates on local copies of the archive so it's pretty much useless without them.

A side benefit is that you'll have off-node backups of these files should the disk drive on the node fail.

After the initial Sync is done, NodeShark always does incremental Syncs. NodeShark only backs up files on the AllStar node whose dates are more recent than the most recently backed up log file in the local archive. NodeShark doesn't use file creation dates or modification dates but just goes by the date built into the names of the log files and transmission recording files.

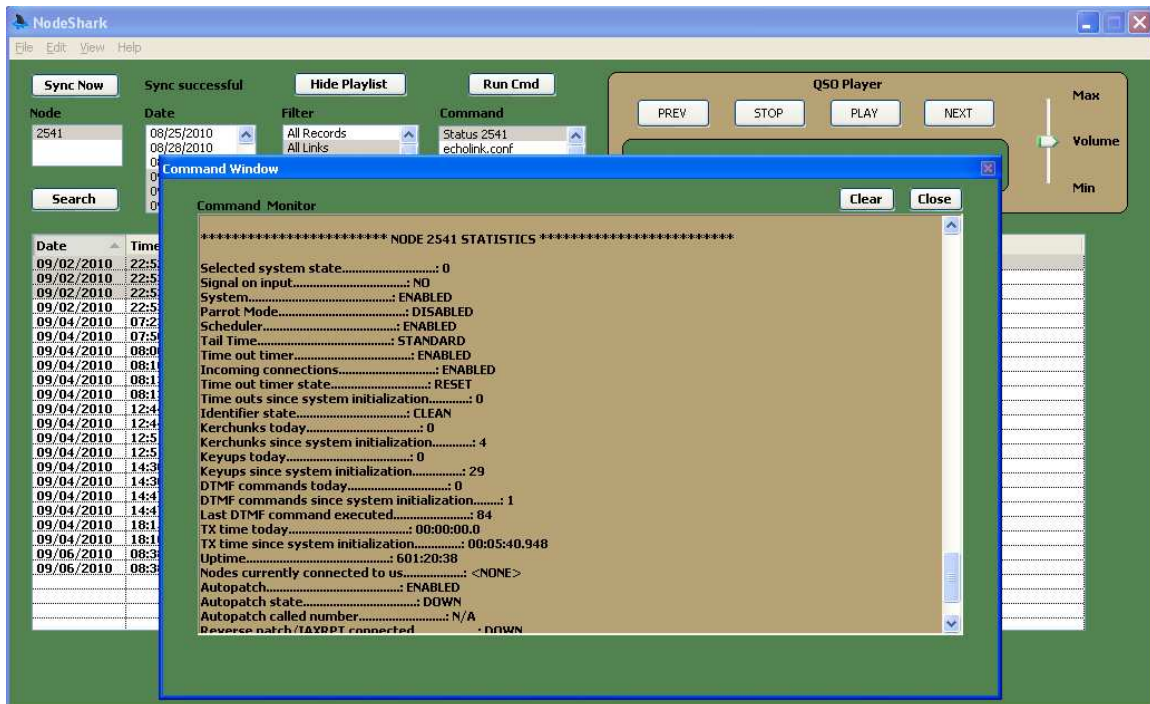
NodeShark Syncs can be done from multiple computers since there is nothing maintained in the AllStar node's archivedir that indicates whether that archive has been backed up by NodeShark. Additionally, you can Sync across any network that allows you to SSH into your AllStar node. I run NodeShark at work and Sync though our corporate firewall and also run NodeShark at home and Sync though a home router.

To allow you to follow the progress of a given Sync session, NodeShark provides a Command Monitor window. To access it, go to View -> Command Monitor. When the

Sync session is complete, go to View -> Command Monitor again to deselect it or simply click on the Close button in the Command Window.

For debugging, you can get additional detail on the PSCP session startup dialog by going to Edit -> Preferences -> Backup and click on "Enable Verbosity". This is especially handy when you are testing a new set of Backup settings. You'll be able to tell immediately if you have configured the wrong IP address/port number or have mistyped your login or password.

3.5 Remote Command Execution



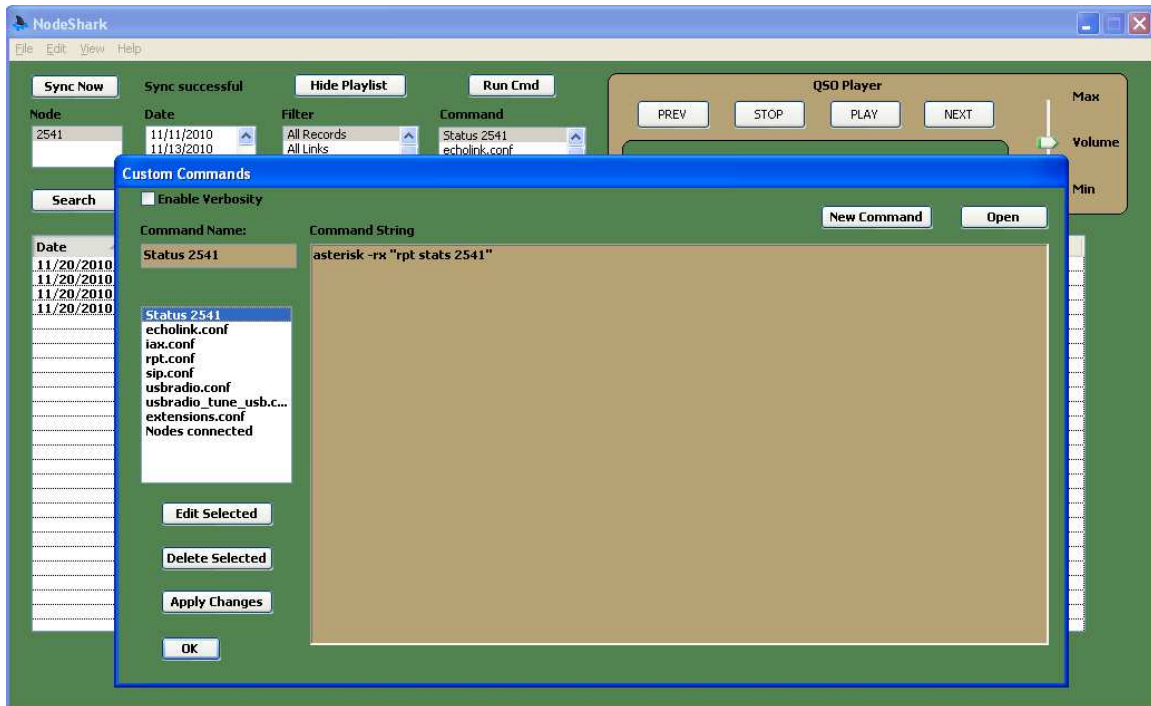
Executing a remote command script (assuming you have created at least one) is as simple as selecting the command name in the Command selection list box and then click on the Run Cmd button. The Command Window will immediately open and you will see that the command has been launched. When the command execution completes, you will see the command output in the Command Monitor box of the Command Window.

The output of each command you run will be appended to the output from the previous command in the Command Monitor box. If you wish to erase the output from previous commands before running a new command, simply click the Clear button in the Command Window.

If you wish to see more verbose output related to the progress of remotely launching a command, you can click on the Enable Verbosity checkbox in the Custom Commands editor window (see the following section on the Command Editor).

When you are finished, click the Close button to close the Command Window.

3.5.1 Command Editor



The Command Editor window allows you to view, create, name, edit, and delete command scripts. To access it, from the main menu, select Edit-> Custom Commands.

NodeShark doesn't ship with any predefined command scripts. I felt it would be too presumptuous (and risky) to create scripts that run with a single touch of a button on someone else's AllStar system. So, for this feature to have any value, you need to create your own.

To create a new command script, click on the New Command button. This will cause the NodeShark Command Editor to enter "Create Mode", fill in a default name for the command, and enable editing within the command editor text box.

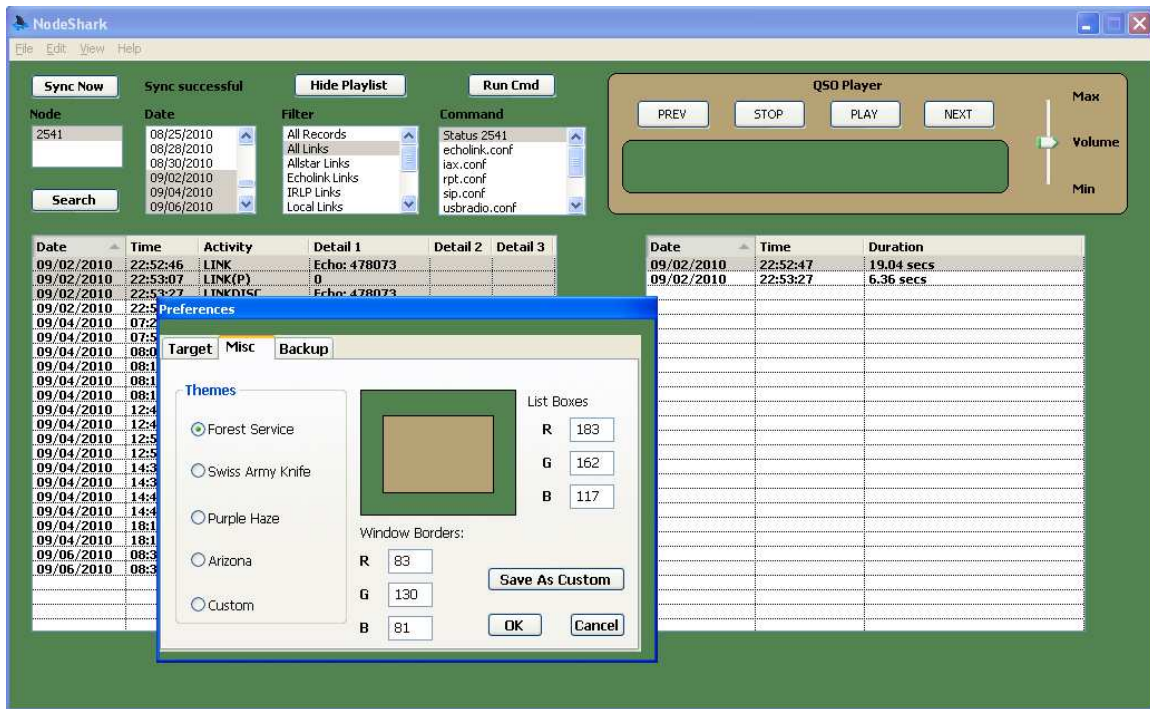
At this point, you can click in the Command Name text box and change from the default name to whatever name you want to use for this command script. You can actually rename this command at any time prior to clicking on Apply Changes. Next, click the mouse in the Command String window and begin typing in your custom command script. Remember, any set of commands that are valid on the AllStar system and are non-interactive, are valid for this remote command execution facility. If you already have created a command script outside of NodeShark (using a text editor such as Notepad), you can browse to where you have it saved by clicking the Open button and then click OK within the Open dialog box. The script will be read into the Command String box where you can optionally edit it.

Once you are satisfied with your command script and ready to test it, click on Apply Changes. Your new command will appear in the Command selection box on the main NodeShark window. If you want the new command to be persistent across NodeShark executions, you need to click on File -> Save Preferences before exiting the app.

To edit an existing command, select it in the command list in the Custom Command window by clicking on it and then click Edit Selected. Now you can edit the script to your liking. Once you are satisfied with it, click Apply Changes to commit the changes. Remember to do File -> Save Preferences prior to exiting NodeShark if you want the changes to be persistent across NodeShark executions.

To delete an existing command, select it by clicking on the command in the command selection box and then click Delete Selected. Again, you must do File -> Save Preferences if you want the command deletion to be permanent.

3.6 Customizing NodeShark's Color Scheme



NodeShark ships with a default color scheme called "Forest Service". You can change it to one of three other pre-installed color schemes or create your own color scheme. To change it or to create your own theme, go to Edit -> Preferences and click on the "Misc" tab.

In the Misc panel you can select one of the other color themes or create a custom color theme. To create a custom theme, you can start from any of the standard themes and then modify each of the RGB numbers (0 - 255) under List Boxes and Window Borders. As you change the RGB values, NodeShark gives you instant feedback on what the resulting colors will look like. If you like a pair of colors that you create, you can click on the "Save As Custom" button. Your theme will be automatically selected and will take effect immediately. To save this theme across NodeShark restarts, remember to go to File -> Save Preferences.

4 Troubleshooting

4.1 Windows Installation Issues

Installation on supported 32-bit Windows platforms should really be trouble free. The NodeShark executable along with the NodeShark DLL folder can be extracted to any Windows folder desired. To make access easier (hmmm..where did I put that thing?), I recommend immediately creating a Desktop shortcut (right click on the executable and choose "Send to Desktop").

On 64-bit Windows platforms, installation location is more critical. The 64-bit Windows OS variants require that 32-bit Windows applications be placed in the "Program Files (x86)" folder. So if your Windows version is 64-bit, I recommend creating a folder called NodeShark below "Program Files (x86)" and extract the NodeShark files there. It will also work to extract directly to "Program Files (x86)" but you'll notice that the convention is to contain each application in a sub-folder that is usually named the same as the application.

To check whether you have a 32-bit or 64-bit version of the Windows OS, you can go to Control Panel -> System.

4.2 Synchronization Issues

There can be many reasons that "Sync Now" can fail but the prime reason is connection failure to the target AllStar node.

4.2.1 Incorrect IP Address

The IP address specified in the Backup Preferences tab must be correct relative to the network that the NodeShark PC runs on. Some operators will run on a PC that is within the same LAN that the AllStar node is on. In that case, it is probably a local, non-routeable address of the form 192.168.x.y. This is the easiest and least error-prone configuration since no routers are involved. However, it is also the most restrictive configuration. If you are configured this way, you can test your connectivity by pinging the AllStar node IP address from the PC. If that works, you can also test by trying to connect to the AllStar node using "SSH" (i.e. PuTTY).

If your AllStar node and your PC running NodeShark are on separate networks, you will need to specify the Public IP address of the AllStar node in the Backup tab of Preferences. Unless your AllStar node has a direct public IP interface on it, the packets will have to transit through a small router that provides NAT (Network Address Translation). This usually means you need to setup port forwarding on that router so that the NAT forwarding engine within the router will know what internal IP address to pass packets to for the port you have configured for SSH/SCP.

4.2.2 Incorrect Port Number

Typically, AllStar nodes use TCP port 222 instead of the standard port 22. If your NodeShark PC is on the same LAN as your AllStar node, you would specify that port (222) directly in the Backup tab of your Preferences. If instead your NodeShark PC and AllStar node are on separate networks, you would specify the TCP port number you have set on the public side of the NAT forwarding entry you created in your router. This may be the same TCP port number as above or you may have further obfuscated it by mapping a different port (e.g. port 8020) for SSH/SCP on the public side of the router to the internal port (222).

You can test this outside of NodeShark by running PuTTY SSH from your NodeShark PC.

To test from within NodeShark, you can try a "Sync Now" after setting the "Enable Verbosity" option within Preferences->Backup and clicking on View->Command Monitor. This will allow you to watch the detailed dialog between PSCP (launched by NodeShark) and the SSH/SCP daemon on the AllStar node.

4.2.3 Incorrect Login/Password

Obviously, you need to specify the correct login and password for the AllStar node in order for PSCP to be able to retrieve archive files from it.

You can test your settings using the "Enable Verbosity" option and watching the Command Monitor window while doing a "Sync Now".

4.2.4 Incorrect Remote Path

The correct setting for the Remote Path is the same path as you set in the rpt.conf file on your AllStar node. As such, it would exclude the node number since that is added automatically to the path depending on which node number you're syncing.

4.2.5 Incorrect Local Path

The correct setting for the Local Path is the full pathname of the top-level folder you created for backing up your archives. As such, the local path would NOT include the node number folder.

4.2.6 SSH/PSCP Keys not yet cached

The first time you connect via SSH or PSCP to a remote node from a local system, the local SSH/PSCP client prompts you as to whether you want to cache the generated keys for the remote system. You type in "Yes" or "No". You need to type in "Yes" so that you won't be prompted each time you connect to that particular remote system. This is critical to the operation of the Sync feature in NodeShark, since I didn't program NodeShark to

do that for you (seems too presumptive to me). You only need to do this once using either PuTTY SSH or PSCP from a command line and it will be cached.

4.3 Other Issues



For any other issues you run into that the above troubleshooting tips fail to resolve, feel free to email me at the email address in the Help -> About window. Please provide as complete a description as possible including the version of NodeShark you are running.

5 Disclaimers

NodeShark is provided for free, as-is, with no warranty or guarantee of support. I wrote NodeShark to try to contribute back, in a small way, to the awesome work done by Steve Rogers, Jim Dixon, and all others who helped to create the AllStar software and network infrastructure.