Backup Methods for your BBB or RPi2 Node

PLEASE NOTE THESE INSTRUCTIONS APPLY STARTING WITH BBB 1.2 AND RPi2 Images

It is very important to do periodic backups of your node especially if you make lots of changes. Although you can always just reload the stock image without a backup you would lose all changes you made like configuring your node, etc.

Once you loaded the image and configured your node you should make an image backup of the SD card. This backs up the entire system so any and all changes you have made will be stored.

Note – if you are still using BBB version 1.0 which is now over a year old (you should upgrade!) please see the instructions at the end of this document.

RPi2 and BBB version 1.2 or greater Backup and Restore Scripts

Once you have your Allstar system up and running and configured you should do a complete image backup (see image-backup.sh below). You also should use one of the new backup methods detailed below to backup you configs and local directory so you can carry them to the future releases of the image.

The RPi2 and BBB Version 1.2 has five new scripts to accomplish this, backup.sh, restore.sh, usb-backup.sh, usb-resotre.sh and image-backup.sh. Here are descriptions of each file.

backup.sh

This script will backup the main directories that will be required for restoring the Allstar BBB setup. These scripts use the local file system to backup and restore. The file system directory that is used for backup/restore is /node_backup.

The backup script saves all files in the /etc/asterisk and /var/spool/cron directories. The included and excluded files and directories to be backed up can be configured in the script if desired. Information about the system it was stored on is also saved

These files and directories are saved as compressed tar file in the /node_backup directory. Multiple backup can be made, each is time and date stamped. These files can be copied to a different system for restoration using sftp, scp, winscp or other methods

restore.sh

The restore script reads the /node_backup directory and supplies a list of backup files located there to the user. When you select a file it is untar'ed to the original directories. Typically you would copy the file(s) from another BBB or RPi2 to a new installation in the /node_backup directory and run restore.sh but you also might want to store multiple configurations that you could change at will on the same system. If the restore script detects a restore to a different system then it was backed up on it asks if you want to restore. This avoids over writing a system with improper data. In most cases it would be alright to do this but it asks just in case.

Backup File Names are in the following format::

<hostname_yyyy-mm-dd-hhmm>.tgz

for example:

node1998_2014-09-22-1408.tgz

Yet another reason why you should name your system with your node number for identification!!!

usb-backup.sh

An even better and somewhat easier method to backup your Allstar setting is usb-backup.sh

This script uses a USB memory stick to store and transfer your files. The RPi2 and BBB Version 1.2 have been configured to mount a stick and the usb-backup.sh script will recognized the stick and back up your files in a flash. Simply run the script and follow the directions. If you have a functional USB hub you can plug the stick in with other devices without any problems. If you do not have or you are not using a hub you can pull your URI or sound FOB and install the stick directly to the BBB.

The script uses devmon to auto-mount and stores the compressed tar file to the stick.

The script has the same include and exclude options as backup.sh and saves with the same file names and system information.

usb-restore.sh

The usb-restore.sh script simply works in reverse of usb-backup. Run the script and follow the directions. The user will be supplied with a files or list of files if more than one was created and is given a choice of what backup file to restore. The files are then restored to their original locations on the current system. The user is prompted if the system does not match the backup system and asked to continue.

image-backup.sh

The image-backup.sh script creates a complete image of your current SD card and writes it to a USB stick. Typically images sizes are 4GB assuming you did not expand the file system. **Expanding the file system is not recommended.** So you must have a USB stick that is at least this size preferably greater. A recommended size would be 16G or 32G. A 32G USB stick is currently (9/2014) available for about $19 at Walmart.

Like when using usb-backup.sh you have the option of plugging the USB stick directly into the BBB or you can use a USB hub. The RPi2 has four ports so a hub should not be necessary. When you run image-backup.sh you should either have the USB stick inserted or insert it when instructed after starting the script.

The USB stick as purchased is probably FAT32 formatted and this will NOT work. You must reformat the stick with either EXFAT or NTFS format. This can easily be done in Windows. Insert the Stick and then select START, Computer, to view the mounted drives. Left click the stick once to select it then right click and select format. The format window will open. Select either EXFAT or NTFS under files system. Check quick format and start. The format is very quick. Your stick is now ready to be inserted into the BBB.

The script will assign a file name to the backup, verify that a stick is present and mounted and then proceed to create the file systems and copy the image to the USB stick. Note that this process can take as long as 10 to 20 minutes or more and at some points it will tell you to wait with no feedback. When done it will tell you it has completed and that you can remove the USB stick.

Since you undoubtedly will be running this from a remote computer using Putty or whatever program you ssh in with, that session will be tied up while doing the image backup. You can always open a second ssh window to your system to carry out other administrative things while the backup is taking place.

Image File Names are in the following format::

<hostname_yyyy-mm-dd-hhmm>.img

for example:

node1998_2014-09-22-1408.img

The initial backup takes longer as it creates the file systems and does a complete backup of the SD card. Subsequent backups only save changes using rsync and should be much quicker often just several minutes.

When you re-enter the script it will ask you if you want to create a NEW (complete) image or you can select a previous image if one exists to update.

While the purpose of the script is to save images in the case of failure the .img files created can be used to create new SD cards which are then clones of your current system. You would write the .img file to an SD card just like you would any .img using win32diskimager or in linux dd or whatever image writer you may choose. Note you have to do an image copy not just copy files!

For most users once you have your Allstar system configured there will be only minor changes to the system. So a monthly or even longer run of this script to do an incremental backup would be sufficient. I would recommend a complete backup prior to making more extensive changes in the system or at periodic intervals so you can go back to prior versions if something should go wrong. Incremental backups are OK but they could also perpetuate a system problem so it is good practice to make a complete backup at least every few months.

Remote BBB's could be backed up using ssh to run the script with a USB stick permanently installed on a hub. A 32G stick would store at least seven complete backups.

The image-backup method allows a user to copy an SD card on a live system without removing the power, interrupting its operation (Allstar), or removing the SD card. Here are the steps to do that:.

'putty' into your system.

Use the image-backup script to create an image on a removable USB stick.

Use scp or winscp to copy that image from the USB stick to a PC

Use win32diskimager or Linux dd to copy the image to a SD card inserted in a reader/writer on the PC.

These operation would take about an hour requiring minimal user interaction.

The user has the option of using the image-backup script to backup or clone the SD card or the usual method of powering down and removing the SD card and reading it on an external reader into an image files on another computer. Either method works, this script just makes it a little more convenient.

Also note that creating a new system with the great setup scripts we now have takes minimal time so if you do not make a lot of changes other than the normal setup in some cases it might be just as easy to just recreate the system rather than doing backups should you need to. Just keep a copy of your setup parameters like node, node PW, simpleusb.conf settings, and RX and TX levels. If you have everything handy you could get a new system up from scratch in less than 10 minutes. Obviously if you make a lot of changes and additions like personal scripts, configurations, etc. you will want to back them up.

Many thanks to Chris, W0ANM for writing these backup scripts to make our life easier!!!

The following instructions only apply if you are doing a backup on BBB version 1.0 !!!

The easy way to do this is to shutdown the system -

shutdown -h now

When the four LED's to the right of the Ethernet connector on the BBB turn off remove power and then remove the SD card. Insert it in a PC card reader and using DD (linux) or win32diskimager (Windows) read the card and create a backup file. This file will be stored as a backup should you need it. Do this whenever you make significant changes to the BBB system that you would not like to have to recreate should there be a failure.

Since most all of the changes you would normally make to the system reside in /etc/asterisk that directory and those below it can be tar'ed and transferred to another system for safe keeping. This backup would also be used should you want to update to a new image in the future.

NOTE – just copying file using winscp DOES NOT preserve Linux file permissions. Therefore it is strongly recommended that you use tar on your BBB computer to transfer files.

If you are upgrading to version 1.2 please skip to the special notes below.

To backup using tar do the following:

Change directories to wherever you want to store the tar file on the bbb.

cd /root

then execute:

tar cvzf asterisk-backup.tgz /etc/asterisk

This will create the file /root/asterisk-backup.tgz with all of the files from /etc/asterisk inside.

Then use a file transfer program like winscp to transfer the file to you PC.

If you need to recover the file in the future to a new BBB install copy the backup file back to the root base directory (/ NOT /root) of the BBB again using a file transfer program like winscp. Then untar it:

cd /

tar xvzf asterisk-backup.tgz

Now all of those files will replace those in /etc/asterisk.

While winscp is fairly easy to use on the Linux side on you r BBB scp is a command line directive.

Here are two examples of how to use scp to transfer a file to another BBB server or linux computer.

The first example requires interaction. It asks for your password.

The second example uses sshpass to pass the password in the command line.

NOTE - this is not secure but for local transfers and even for any transfers unless you are paranoid this should be OK. We aren't dealing with national security here!!!

Obviously replace with your username, IP, port, and file names and for sshpass with your password.

```
scp -P 222 root@192.168.0.132:/etc/asterisk/local/halt.wav halt.wav
```

This will ask for your PW.

```
sshpass -p "password" scp -P 222 root@192.168.0.132:/etc/asterisk/local/change_vol.sh change_vol.sh
```

This will execute immediately as the PW is part of the command. **Note sshpass is only available in version 1.2 or greater.**

The following nodes apply to the RPi2 and BBB versions 1.2 or greater only! Read these instructions carefully and formulate a plan for your upgrade!

Starting at version 1.2 the **/etc/asterisk/local** directory is reserved for user scripts and misc. user storage. All global script files and files not generally modified by the user have been move elsewhere. Because of this it is recommended that you **NOT** copy your old local directory to your new 1.2 install. You may copy files that you may have placed in the local directory over to the new install but nothing else.

Unless you want to start over and setup the config files from scratch you will need to copy them from your old version. You will only need to copy the files in /etc/asterisk that you have changed. Typically the following files -

extensions.conf
iax.conf
rpt.conf
simpleusb.conf
echolink.conf < Only if you configured it in the prior version
simpleusb_tune_usb.conf < to preserve your radio level settings

DO NOT WHOLESALE COPY THE /etc/asterisk/local DIRECTORY!!!

You can use scp or winscp to copy these files to another computer and then after installing version 1.2 back to the same directory overwriting the stock files that come with 1.2 **Do not copy or alter other files unless you know what you are doing!!**