

Linux system image propagation and backup instructions for the MBARI AVED project.

by:
Karen A. Salamy

The AVED project uses the System Installation Suite (SIS) software to propagate system images across various computers and to create image backups of those systems. SIS is an image-based installation tool for RPM-based Linux distributions. It is a collection of packages, including System Installer, SystemImager, and System Configurator which provide a total solution for Linux installation and maintenance.

The following seven steps outline basic usage of the system for the MBARI AVED project.

- 1) [Installation of SIS software on client computer system.](#)
- 2) [Creation of an RSA key login that will allow remote root login without the use of a password.](#)
- 3) [Creation of a backup of the koi_golden image prior to updates of koi or of other systems.](#)
- 4) [Performing system updates on koi using CVS repository and system testing.](#)
- 5) [Creation of new koi_golden image on SIS server.](#)
- 6) [Backup of client image prior to being updated with new koi_golden image.](#)
- 7) [Propagation of the koi_golden image to client systems on the AVED project.](#)

Step 1. Installation of SIS software on new computer. If you are not adding a new computer to the AVED image system configuration and are working on a system that has been set-up before to update its image to koi, skip Steps [1](#) and [2](#) and go to [Step 3](#).

NOTE: Anywhere *newcomputer* is seen, replace with the name of the new Linux system (e.g., scorpionfish).

Login into *newcomputer* as **root** (use system's <root password>) and create a folder to store all SIS installation files.

```
[root@newcomputer]$ mkdir /root/RPMs/SIS_RPMs
```

Next, ssh into koi as root and change to the SIS_Setup directory.

```
[root@newcomputer]$ ssh -l root koi Use koi's <root password>
[root@koi]$ cd RPMs/SIS_RPMs/SIS_Setup
```

Secure copy (scp) the following RPMs from koi to the new computer to be setup.

```
[root@koi]$ scp * root@newcomputer:/root/RPMs/SIS_RPMs
Use newcomputer's <root password>
[root@koi]$ exit (this will exit you back to newcomputer)
```

Next, change to the SIS_RPMs directory.

```
[root@newcomputer]$ cd /root/RPMs/SIS_RPMs
```

and install all of the rpms (use command: **rpm -ivh**) in the following sequential order).

1. db2-2.4.14-10.i386.rpm
 2. perl-modules-5.6.0-32cl.i386.rpm
 3. perl-Appconfig-1.52-6.noarch.rpm
 4. systemconfigurator-2.0.2-1.noarch.rpm
 5. systemimager-common-3.0.1-1.noarch.rpm
 6. systemimager-client-3.0.1-1.noarch.rpm
-

Step 2. Create RSA key on *newcomputer* and copy to *koi*. This will enable root login between *newcomputer* and *koi* without the need to type in a root password. This is a crucial step in propagating images across systems.

From *newcomputer* type the following:

```
[root@newcomputer]$ ssh-keygen -t rsa
```

Hit enter a few times using all defaults. Once the rsa key generator has been completed, cd into the root `.ssh` directory.

```
[root@newcomputer]$ cd /root/.ssh
```

Secure copy *newcomputer*'s rsa key to *koi*. Use *koi*'s `<root password>`.

```
[root@newcomputer]$ scp id_rsa.pub root@koi:/root/.ssh/newcomputer_id_rsa.pub
```

Next, ssh into *koi* as root and copy *newcomputer*'s key to the `authorized_keys2` file on *koi*

```
[root@newcomputer]$ ssh koi <root password>
[root@koi]$ cd /root/.ssh
[root@koi]$ cat newcomputer_id_rsa.pub >> authorized_keys2
```

Next, from the same directory on *koi* copy the already generated `rsa_key` from *koi* to *newcomputer*. Use *newcomputer*'s `<root password>`.

```
[root@koi]$ scp id_rsa.pub root@newcomputer:/root/.ssh/koi_id_rsa.pub
[root@koi]$ exit (exits you back to newcomputer)
```

Finally, add the `koi_id_rsa.pub` key to the `authorized_keys2` file on *newcomputer*

```
[root@newcomputer]$ cat koi_id_rsa.pub >> authorized_keys2
```

Next, verify that the changes have worked. From *newcomputer* type:

```
[root@newcomputer]$ ssh koi
```

This should automatically log you in as root on *koi*. Then, type the following from *koi*:

```
[root@koi]$ ssh newcomputer
```

This should log you automatically in to the *newcomputer* as root. If this step does not work, do not proceed with image updates as they will not work until this is successful. If all else fails, you may re-generate the `koi_id_rsa.pub` file on *koi*, however, be aware that that will "break" the `rsa_key` login capabilities for all other AVED systems (i.e., they will all need to be updated with the new `koi_rsa_id.pub` in their `/root/.ssh/authorized_keys2` file).

Step 3. Create a backup of the koi_golden_image prior to updates of koi or of other AVED systems.

Backing up working koi_golden_images are critical to the maintenance and support of the AVED imaging system. Before a new image can be made, an old image must be backed up to disk. Older versions of the koi_golden_image are burned onto data DVDs in order to maintain an accurate record of previous images for the project and act as rescue disks should a serious problem occur during an upgrade.

For AVED, data DVDs are created with the k3b CD Kreator GUI application. While easy to use and extremely useful, one of the few limitations of this (and all linux DVD burning software) is that it is based on the 'mkisofs' and 'growisofs' utilities. 'mkisofs' and 'growisofs' are both used to create an ISO9660 filesystem for burning CDs and DVDs, however, they are limited to recording only 2GB files at any time. Any file size larger than 2GB will produce a burn error at this time. With technology moving ever-forward, future updates of the 'mkisofs' and 'growisofs' utilities should rectify this problem but for the time being, it limits AVED system backups from koi.

Limits are the following: The total size of the koi_golden_image is ~5GB which requires some compression (TAR) to fit all on one convenient DVD which holds 4.4GB. Any TAR file created can NOT exceed 2GB (due to 'mkisofs' and 'growisofs' limitation listed above). In the past, usually two or sometimes three *.tar files have been made to get all of the filesystem files on one DVD. Every system update to koi will change previous file system sizes and thus affect file compression and TAR creation. Thus, the ultimate compilation of files into *.tars will be up to the person maintaining the system. This document merely provides one way of doing it based on the current system configuration.

1) Determine image size and create compressed image files to burn to DVD

To get started, perform a disk usage call to determine how large each directory is within the old koi_golden_image. First, log into koi as root *<root password>* and determine the disk usage of image directories.

```
[root@koi]$ cd /AVED_images/images/koi_golden_image
[root@koi]$ du -hs *
```

A typical listing may look like this:

```
4.0K  AVED_images
7.0M  bin
5.2M  boot
436K  dev
16M   etc
4.0K  home
4.0K  initrd
46M   lib
4.0K  lost+found
20K   mako
4.0K  misc
20K   mnt
4.0K  opt
4.0K  proc
291M  root
15M   sbin
52K   tftpboot
2.7M  tmp
8.0K  tornado
2.6G  usr
1.8G  var
```

NOTE: mako/ and tornado/ are mounted file systems and should not be included in the backup of the koi_golden_image. The AVED_images/ directory should also be excluded.

As you can see from the disk usage display, the majority of the directories are small, adding up to 400 MB at most. Two directories stand out however. These include the usr/ and the var/ directories which are both at or over the 2GB file size limit for burning DVDs. As a result, several *.tar files will need to be created in order to get all files onto a DVD.

The usr/ directory is usually TAR'ed into its own file due to its size, while the var/ directory can often be TAR'ed and GZIP'd with the rest of the directories and will generally result in one file just under the 2GB limit. Again, updates change these dynamics and the final decision on how to compress these files will be ultimately left up to the person in charge of the image maintenance for the project.

For the listing above, to create two files <2GB that will fit onto one DVD, one option is:

```
[root@koi]$ tar czf /home/koi_golden_image_date_part1.tar /bin boot/ dev/ etc/ home/  
initrd/ lib/ lost_found/ misc/ mnt/ opt/ proc/ root/ sbin/ tftboot/ tmp/ var/
```

```
[root@koi]$ tar czf /home/koi_golden_image_date_part2.tar usr/
```

where **date** is the date the image was created. (e.g., koi_golden_image_2-3-2005_part1.tar).

!!!NOTE: Even though these files are temporary and will only exist until they are burned onto DVD, it is best to create them in the /home directory which has sufficient space to store them on the system. /root simply does not have the space. This is why it is strongly suggested that the above files be saved as /home/koi_golden_image.

Once the new *.tar files are created, verify that both are <2GB. Otherwise, you will need to come up with a new file compression configuration as they will not burn properly to DVD.

2) Burn newly compressed files to DVD with k3b GUI.

NOTE: For more in-depth usage directions on k3b, please see the document "How To Burn DVDs in Linux using the 'k3b' utility".

As root, open up the k3b DVD burner GUI application on koi:

```
[root@koi]$ k3b
```

Set up writing speeds of CD burner (if asked) on start (40X is standard).

The next step is to build a new Project. Do this on the top menu:

File → New Project → new Data DVD Project

This will create a new project in the bottom half of your screen with browser windows in the upper half of the screen. To choose files to burn to DVD, simply browse the file tree structure and drag and drop specific files to the new Data DVD Project icon in the bottom left of the screen.

Once you have dragged and dropped all of the files that you want into your new DVD project, set up the project's properties in the following manner starting at the top menu:

Project → Properties

This should open a second window entitled: "DVD Project – k3b"

Click on the **"Writing"** menu tab. The following should be set:

Burning Device: (e.g., SONY DVD RW DRU-530A)

Speed: Auto

Writing Mode: Auto

Options: [*"On the Fly"* and *"Remove Image"* options should be checked. Leave others unchecked.]

Next, click on the **"Settings"** menu tab.

"No Multisession" should be checked.

Next, click on the **"Volume Desc"** menu tab. This is an arbitrary section for Disk information and is dependent on the user and the data being stored.

Volume Name: Whatever the user wants to call it (e.g., *koi_golden_image_02_03_2005*)

Volume Set Name: Whatever the user wants (e.g., *Part 1 of 3*) – Usually blank.

Volume Set Size: 1 (default)

Volume Set Number: 1 (default)

Publisher: Whatever the user wants (e.g., *MBARI*)

Preparer: k3b-Version 0.11.6 (default)

System: Linux (default)

Application: k3b The CD Kreator Version 0.11.6 ©2003 Sebastian Trueg and the k3b Team (default)

Next, click on the **"Filesystem"** menu tab. The following should be set here:

Under **Filesystems:** Click on **"Generate Rock Ridge Extensions"**

Under **Symbolic Links:** Do **Not** click on either option.

Under **Permissions:** Click on **"Preserve File Permissions (backup)"**

Under **Whitespace Treatment:** Click on **"Replace with [_]"** This will put an underscore in for all white spaces within filenames.

Finally, click on the **"Advanced"** menu tab.

Here, click only on the following box:

"Allow untranslated filenames"

Save the Setup when done.

The project title should reflect the one that was given in the setup now. It is good to verify this.

Next, you are ready to begin burning the DVD. Insert the DVD-R into the system DVD burner and then go to:

Project → Burn

The system will start to burn the files to the DVD as requested. This process will take some time depending on the speed of the burner and the number of files.

While burning, the GUI will show you a time-bar to indicate its process, speed, and estimated time until completion.

When the system is through burning the files to DVD, the GUI will indicate that it was successful and eject the DVD. **NOTE:** It will also tell you if it failed (usually due to files being > 2GB). If a failure of this kind does occur, you will need to use a new DVD as the old one will have been written to during the initial burn step and it will appear to be a “used” DVD. Once a DVD has been burned successfully, you are then welcome to begin another project or to quit. It is okay to discard unsaved data if the DVD was burned successfully and it is the only copy of those files that you wish to make.

Once you verify that the files were successfully burned to the DVD, remember to delete them in the /home directory to preserve disk space on the system.

Step 4. Update koi system with changes from CVS repository and test, test, test.

Normally this step is preceded by an e-mail from the project's lead software engineer indicating that changes have been made to the vision software and the updates need to be updated to the koi system for creating a new koi_golden_image. This step is important as not only is the golden image created from the koi system but it is always a good idea to thoroughly check the new update on the koi system before creating the image and then propagating it to the other AVED systems. Often, software update steps will be outlined in detail to you by the lead software engineer. For example, often changes are only made to a few directories on the system (e.g., `aved/mbarivision` and `aved/scripts`) requiring only a few configuration steps and not a full upgrade of the entire system – information which should be provided to you in an e-mail or in a meeting prior to attempting an update of the koi system.

Steps to update koi to the CVS repository (this can be done as any user):

First, login to the CVS repository on moonjelly (usually in your `/home` directory (*user*)).

```
[user@koi user]$ cd /home
[user@koi user]$ cvs -d :pserver:user@moonjelly:/home/cvs login
<enter your cvs password>
```

and checkout the code to your directory if you haven't done so already. This should install the `aved` files in your `/home` directory.

```
[user@koi user]$ cvs -d :pserver:user@moonjelly:/home/cvs checkout aved
```

If you have already checked out the `aved` module, update your files to match those in the repository:

```
[user@koi user]$ cvs -d :pserver:user@moonjelly:/home/cvs update aved
```

To update only specific directories (e.g., `scripts`) rather than the entire project, do the following:

```
[user@koi user]$ cvs -d :pserver:user@moonjelly:/home/cvs update aved/scripts
```

To update a directory to a specific release (e.g., `release-2004-00_01_00`), do the following:

```
[user@koi user]$ cvs -d :pserver:user@moonjelly:/home/cvs checkout -r
release-2004-00_01_00 aved/mbarivision
```

Once the new repository files are added to the system, perform all requested update steps to the system and then rebuild it.

!!Follow all directions listed by the software engineer and do not be afraid to ask questions regarding the update process. You will need to perform a 'make clean' and then a 'make' on all updated applications. Again, all changes should be described in detail to you by the person who made the system changes to the repository files. This is an important step and worth discussion if you have questions.

Next, test the system thoroughly by running video clips through the vision software and verify that all changes took effect and work properly. Only once you have performed a thorough system test should you generate a new `koi_golden_image`. So test, test, test.

Step 5. Delete old and generate new koi_golden_image on SIS server.

System images (koi or otherwise) are generated through the 'tksis' utility. You should be root on koi when you use this GUI application for file permission reasons.

To bring up the tksis GUI, type the following:

```
[root@koi]$ tksis &
```

tksis will bring up a server window listing all images created and placed on the server.

NOTE: The server location is /AVED_images. Perform a 'df -h' to view the amount of space available on the server before deleting and creating the new image as image creation will fail should you run out of space. The koi_golden_image is currently ~5GB.

In order to create a new koi_golden_image, you MUST delete the old one. To do this, click on the **koi_golden_image** listing and then Select: **Delete**. The average size of the golden image is around 5GB so deletion may take a while.

NOTE: You must use the tksis GUI for changes (deletion or image creation) to show up in the server list. If images do not show up in the server list, other systems will not be able to see them.

Once the image has been deleted, it will disappear from the tksis server window listing.

Next, to create a new image, Select: **Fetch Image**

This brings up a separate window. Input the following:

```
Image Name:      koi_golden_image
Client Name:      koi
SSH User Name:    root
***Exclude File:  /etc/systemimager/koi_backup.exclude
IP Assignment Method: replicant
Post Installation Action: beep
```

Then Select: **Fetch**

*** The koi_backup.exclude file is a list of files NOT to be copied over to the new image by the system and is read by tksis before creating the new image file. This file contains all /home files and mounted systems which are unwanted in the new image. These files are system-specific and should be set-up for the client system and should not be overwritten during an upgrade by the koi system files.

Because the image will be several GB in size, this process will take quite a bit of time to complete. Give it an hour and check back. When the image creation has completed, tksis issues a new pop-up window stating that it was successful (a window will be produced if there was a problem too). Keep in mind, if the ssh-keygen doesn't work the image will not be fetched and an error will be produced. For routine back-ups of the koi system itself, koi must be able to login to root@koi without a password similar to any other client system (see [Step 2](#)). In addition, the new image will appear in the server window listing on the tksis main page.

Step 6. Create a backup of the image you are about to update.

In addition to back-up old koi_golden_images, it is equally as important to back-up working client system images as well in the event you need to recover from a unfortunate upgrade issue.

To do this, burn the existing image onto DVD (see [Step 3](#)), delete the old image from the server list, and then create a new one (see [Step 5](#)). The only difference in the image file name should be the creation date for the image (e.g., mako_image_07_08_04). The creation date for the koi_golden_image is not listed in its filename but it is for all other system images, including the backup of the koi system itself.

To see a previous naming list of standard image name formats, type the following:

```
[root@koi]$ ls -l /AVED_images/images
```

Step 7. Propagate koi_golden_image to other systems on the AVED project.

Propagation of the koi_golden_image to another system on the AVED project is usually performed at the terminal of the computer to be upgraded (client) for ease and convenience (which is how this section is written), however, updates can be performed remotely from koi.

NOTE: In the following text, references to **clientname** indicate the name of the system that is to be upgraded with the new koi_golden_image.

1) Start systemimager on koi

The first step is to ssh into koi and make sure that systemimager services are started:

```
[root@clientname]$ ssh koi (should automatically log you in if ssh-keygen worked)
[root@koi]$ /etc/rc.d/init.d/systemimager restart
```

2) Review, edit and/or create an updateclient.local.exclude file on koi

Next, look at the computer system's updateclient.local.exclude file on koi. This file lists specific computer system files on the client that are **NOT** to be overwritten by the koi_golden_image. Many system files are edited by Linux specifically for that system, it's hardware, etc, so an update.exclude file must be created and tailored for each. In general, many of the files will be the same between systems (e.g., /etc/fstab). It is important to list all files that need to be excluded in this file before performing an update. Should some important boot or system files be overwritten, the system may not boot. If you are building an updateclient.local.exclude file for the first time, please refer to another file for a similar system as a template before running update on the client. All updateclient.local.exclude files can be found at /etc/systemimager (e.g., the updateclient file for scorpionfish is located at etc/systemimager/updateclient.local.exclude.scorpionfish) on koi.

```
[root@koi]$ cd /etc/systemimager/
[root@koi]$ ls -l updateclient.local.exclude.* (list all previously created files)
[root@koi]$ cat /etc/systemimager/updateclient.local.exclude.clientname
```

Once an updateclient.local.exclude file has been created and tested successfully on a system, it rarely needs to be changed again.

3) Copy updateclient.local.exclude file from koi to client system.

Once an updateclient.local.exclude file is either created or edited on koi, the file must be copied (removing reference to the clientname in the filename when doing so) to the client system. **NOTE:** The update client script only looks for the filename updateclient.local.exclude when performing an update. The reference to the client name on each file stored on koi is for the system manager's benefit.

```
[root@koi]$ scp /etc/systemimager/updateclient.local.exclude.clientname
root@clientname:/etc/systemimager/updateclient.local.exclude
```

ssh into the client and verify that the updateclient.local.exclude file was copied successfully.

```
[root@koi]$ ssh clientname
[root@clientname]$ cd /etc/systemimager
[root@clientname]$ ls -l updateclient.*
```

4) Run prepareclient

Run prepareclient. This utility will prepare the client system to rsync with the image server.

```
[root@clientname]$ prepareclient
```

Accept all defaults listed.

5) Verify that the client can see the golden_client_image

Next, check to see if all of the images on the server can be seen by the client computer.

```
[root@clientname]$ lsimage --server koi
```

This should list all of the images in the server that were seen when running tksis earlier. The list should look something like the following:

```
[root@koi]$ lsimage -server koi
```

```
-----  
Available image(s):  
-----
```

```
beowulffish_image_12_31_03  
koi_golden_client  
koi_golden_image  
koi_image_01_13_05  
larvacean_image_01_08_04  
mako_image_07_08_04  
nematocyst_image_03_09_04
```

If not, try restarting systemimager on koi again (see [Step 7](#): part1)

6) Run the updateclient on the client system (i.e., the moment we have all been waiting for)

This step initializes the propagation of the koi_golden_image over to the client system. Make sure you have completed all previous steps before running this script as changes will be made to your client system.

Just to be on the safe side, you can perform a **-dry-run** and pipe it to a file to view all of the files that will be changed prior to making real changes. The system will automatically access the system's /etc/systemimager/updateclient.local.exclude file that you created / edited in [Step 7](#) (parts 2 and 3).

Here is how to run the updateclient in **-dry-run** mode.

```
[root@clientname]$ updateclient --server koi --image koi_golden_image --no-bootloader  
--dry-run > junk
```

NOTE: the **-no-bootloader** flag is important as it will not allow any of the client system's boot files to be overwritten by those on the koi_golden_image.

Review files to be updated on the client system for accuracy.

```
[root@clientname]$ more junk
```

Once you have reviewed the files to be edited / overwritten by updateclient in the 'junk' file and deem them fine, you can then run without the safety net. Correct any problems in the updateclient.local.exclude file on the client system before continuing should changes need to be made.

Running the updateclient for real:

```
[root@clientname]$ updateclient --server koi --image koi_golden_image --no-bootloader
```

If you know updates are only going to affect a specific directory (e.g., /usr/local) you can specify updates to just that directory by adding the **-directory=** flag in the following way:

```
[root@clientname]$ updateclient --server koi --image koi_golden_image --no-bootloader  
                                -directory=/usr/local
```

All that is left is to sit back (usually 20-30 minutes for the first update) and bite your nails – just kidding. The system uses 'rsync' to change files. Rsync does this by comparing each file for differences and updates files that differ than those in the koi_golden_image unless, of course, the files are excluded from this process in the /etc/systemimager/updateclient.local.exclude file. Subsequent updates will be much faster as there will be fewer files to change and update.

Once updateclient has completed, it is always prudent to first verify that important files you asked it to not overwrite are not overwritten (e.g., /etc/fstab, /boot, etc). If all looks well, reboot the system and run a few image clips with Vision before giving it your stamp of approval.

7) Recovery – if needed.

If the system does not boot properly – DO NOT PANIC. Remember you have backups for recovery on DVD. To use a backup, you will have to uncompress the files from DVD and transfer them to the /AVED_image/images folder. You may add the fully reassembled image to the image server on koi without tksis with the following command:

```
[root@koi]$ mksiimage -a --name <imagename> --nobuild --path (fully qualified path to image)
```

where **<imagename>** would be the old, newly reassembled image name from the DVD archive (e.g., scorpionfish_image_01_02_03).

Then type:

```
[root@koi]$ tksis &
```

and verify that the image was added to the server properly.

You may then have to carefully rebuild the client system (depending on where the problem lies and how bad the failure was). If the system does not boot to the proper kernel image, you may need to manually copy files over initially to log back in and re-run updateclient again. Maintaining a network connection is important for this however.

If all else fails, and the system will not even boot, you may need to rebuild the client system from scratch (Linux install disks), copy and install the SIS RPMs again, and then re-run updateclient to re-set all of the parameters back to the old image. Re-check the updateclient.local.exclude file. Chances are that you overwrote something crucial for that system. Again, use other system

updateclient.local.exclude files as templates. All crucial files are already accounted for in those files and will make the update process much smoother and faster.

Happy imaging!

Karen A. Salamy
April 20, 2005