



Kardia / Centrallix VM Appliance Quick Reference

**Version 1.0 Beta 3
11-May-2011**



(c) 2011 LightSys Technology Services, Inc.
<http://www.LightSys.org/>
Redeeming Technology... For God's Kingdom.

Overview...

The Kardia/Centrallix VM Appliance is designed to allow you to quickly get started in helping to develop the Kardia and Centrallix open-source software systems, or to simply allow you to test-drive the bleeding-edge versions of Kardia and Centrallix.

We've created this VM Appliance to make it easier for you to try out these technologies, without having to install Linux, download the source code for Kardia, or install a number of different packages.

About Centrallix...

Centrallix is an open-source HTML/Javascript/AJAX based rich internet application platform featuring data and application abstraction, a SQL engine for joining diverse data sources, declarative (not scripted) development, reports, components, and style templates. Development first began on Centrallix in 1998 as LightSys was being conceived at Operation Mobilization's US headquarters in Tyrone, Georgia.

Centrallix was a pioneer in the world of rich internet applications (RIA's), using AJAX type technologies over half a decade before the term "AJAX" was even coined. As a declarative domain-specific language (D-DSL), Centrallix is aimed at meeting the need for a declarative development platform for community-developed business applications among organizations not having dedicated software development departments.

About Kardia...

Kardia is a free, open-source ministry administration software system (donations, finance, payroll, mailings, etc.) built on the declarative web-based Centrallix app platform. It is aimed at the need for a free and flexible ministry administration database system that does not require expensive computing platforms or development tools, but which instead promotes the local customization of the system by ministry staff without requiring extensive knowledge of how modern development languages work, and allowing such customizations without breaking compatibility with the upstream product provided by LightSys.

Download...

The 1.0 Beta 3 version of the VM Appliance can be downloaded at:

http://www.lightsys.org/public/releases/KardiaVM_1_0_Beta3.zip

The MD5 Checksum is:

51823a988e79ad44228d55ffb8e4f66f

The VM is built on the 32-bit x86 version of CentOS 5.6 Linux. The VM does *not* include the source code for Kardia or Centrallix. Instead, the very latest source code is downloaded and built when you run the VM.

System Requirements...

- Windows, Linux, or x86-based Mac
- 2.5 GB free hard disk space (4.5 GB if you are using Hyper-V)
- 2 GB memory (more if you run lots of applications in addition to Kardia/Centrallix)
- Virtualization Software (VMware Player, VirtualBox, Hyper-V, and Xen have been tested)
- Firefox 3.x or Firefox 4.x
- An Internet connection

Installation...

To install, download the .zip file and extract it. The .zip contents will expand to roughly 2GB.

Then point your VM software at the extracted files. If you are using VMware, point it at the .vmx file. If you are using VirtualBox or other VM products, create a new virtual machine and point the machine at the extracted KardiaVM-s001.vmdk raw disk image file, and configure the virtual machine for the following:

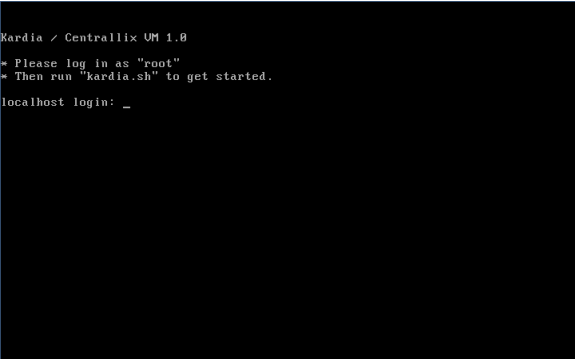
- 512MB of memory
- IDE disk controller (VirtualBox in particular defaults to SATA)
- Time UTC offset of +0 (virtual machine hardware clock set to GMT/UTC)
- Networking to “NAT” (so that the VM has Internet connectivity).

If you are using Hyper-V, you will need to first convert the vmdk file to Hyper-V's VHD format.

The VM image only has about 500MB of free disk space. If you wish to install a lot of packages or import large amounts of data, you will need to grow the disk image and filesystem.

Running the VM...

When you launch the VM, it will boot up CentOS 5.5:



```
Kardia / Centrallix VM 1.0
* Please log in as "root"
* Then run "kardia.sh" to get started.
localhost login: _
```

At this point, log in with username “root” and password “vmtest”. At the root prompt, type “kardia.sh” to launch the menu-based setup and management utility. We did not include the GUI on the Kardia VM, in order to keep the download small. When kardia.sh starts, it will ask if you want to run the initial Setup Guide:



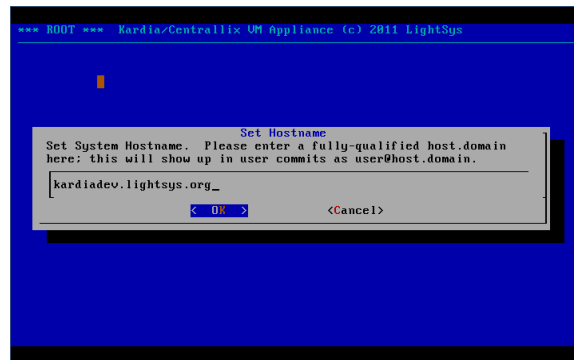
Press [ENTER] to launch the Setup Guide. The Setup Guide will take you through nine or ten necessary steps to set up the VM, download and build Kardia and Centrallix, and run it so you can connect to it using Firefox.

Root Password...

Choose a root password that is secure enough for the purpose you want to use the Kardia / Centrallix VM. Type it and confirm it when prompted.

Hostname...

You should give the Kardia/Centrallix VM a unique host name, particularly if you intend to do any software development on Kardia and/or Centrallix. Then press [ENTER] to continue.



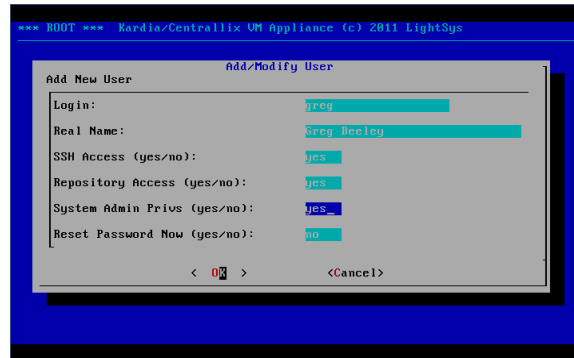
You can change the hostname later by running kardia.sh, going to the Config menu, and selecting the Hostname option.

Timezone...

You'll want to select the time zone that you are located in. On the first screen, you'll be presented with a list of world regions. Select the world region that you are in, and then you can select the city or local area, such as "America/Denver" or "US/Pacific".

Adding One or More Users...

Next, you need to add at least one user. When prompted, select `NewUser`, and then fill in the form. Use the Up/Down arrow keys to move between the form fields, and use `Tab` to move to the OK button when you're done (our apologies for the Tab/Arrow thing; we would design it to always use `Tab` to navigate, but this is how the "dialog" utility, used to create these menus, works). Unless you really know what you are doing, you should type "yes" for the first three yes/no options. Then press `[ENTER]`. You'll be asked to enter a password for the new user account.

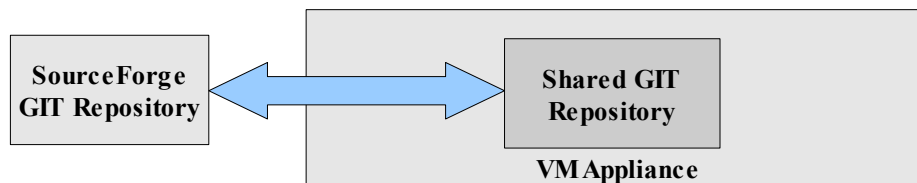


Later, you can always add or modify users by selecting the `Users` option from the `kardia.sh` menu. When you're done adding users, simply press `[ESC]` to return to the Setup Guide (or, tab to the `Back` button and press `ENTER`).

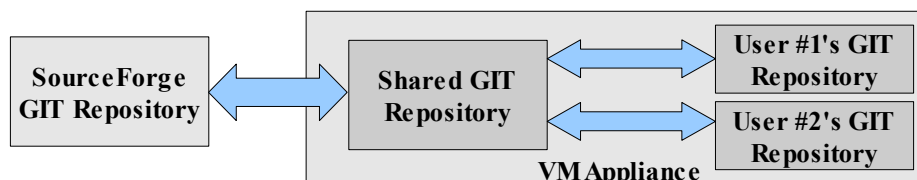
Workflow Mode...

One of the wonderful things about the GIT source code management tool is that it enables a variety of ways of running a software project. This VM Appliance supports three workflow modes, which are also explained by the Setup Guide.

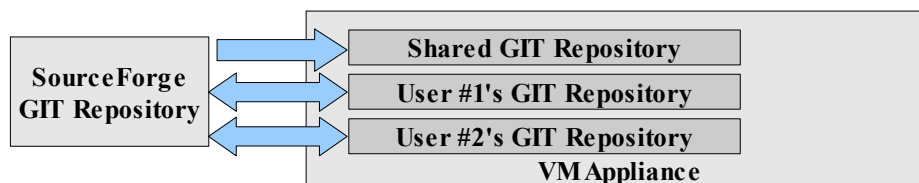
- **Shared:** Select this option if you are just taking Kardia/Centrallix for a test spin, or if you will always be the only user developing on the VM. In this workflow mode, all users develop on a common shared source code repository. This is simplest, but it also means users can easily get in each others' way. It is best for just one user, or for two or three users who are working exceptionally closely together in tight coordination.



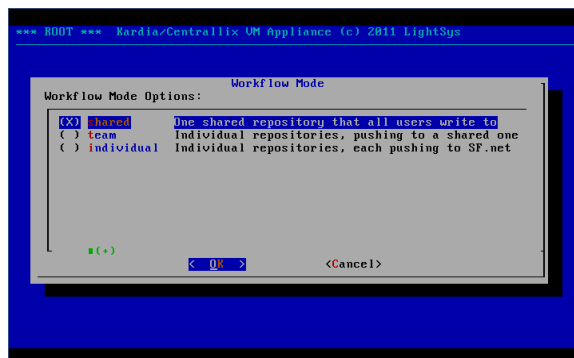
- **Team:** Users develop from their own private source repositories, but when they commit and push changes, those changes are pushed to a common shared repository. From there, changes can be pushed to Sourceforge.net. This is ideal for a team of users who want to review their changes as a whole team and coordinate the pushing of their modifications to Sourceforge.net.



- **Individual:** Users develop from their own private repositories, and commit and push changes directly to Sourceforge.net. A common shared repository still exists, but it pulls changes directly from Sourceforge.net. This is ideal for users who are working largely independently from each other, on separate, distinct, subprojects.



If you don't know which option to pick, choose Shared. On the below screen, use the arrow keys to move to the option you want to use, press [SPACE] to select it, and then press [ENTER] to accept the choice and continue.



Development Mode...

Next, you need to select a Development Mode. This chooses whether the Centrallix server (which underlies Kardia) runs as root, or whether multiple copies of Centrallix run as separate users. If you are just trying out Kardia, or if you want to actually use Kardia with multiple users, select Root here. If you are going to be doing development, you can select either option depending on what you need. If you want each developer to have their own Centrallix server process (to debug, etc.), then select Users here. If you need to develop the production multi-user features of Centrallix, you should select Root. You can always change this option later, via going to the Config menu in kardia.sh, and selecting DevMode.

Run Mode...

Next, you will select the run mode for the Centrallix server, which underlies Kardia. There are three options for the run mode (or, only two options if you selected Users for the Development Mode):

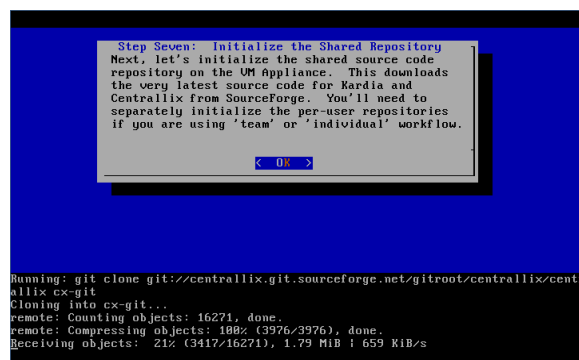
- **Service.** Select this option if you're just trying out Kardia/Centrallix. This runs Centrallix as a normal background service, as root. This options is only available when Root is selected for the Development Mode.
- **Console.** Run Centrallix in a console, which you can access in order to view the screen output of Centrallix (such as error messages). For you Linux folks, this mode runs Centrallix under the “screen” utility.
- **GDB.** This runs Centrallix in a console, using the GDB debugger. This allows you to debug the Centrallix server, and provide us with better bug reports if there is a crash. Select this option if you intend to do a lot of development on Centrallix.



On the above screen, use the arrow keys to move to the selection you want, press [SPACE] to select it, and then press [ENTER] to accept your selection and continue. You can always change the Run Mode by going into kardia.sh, in the Config menu, and selecting RunMode.

Initializing Source Code Repository...

You will need a copy of the Kardia and Centrallix source code in order for the VM to build and run Kardia and Centrallix. You can re-initialize the shared source code repository at any time using the Config menu in kardia.sh, via the SInit option. Note that re-initializing the repository will overwrite any changes you've made that you have not pushed back to Sourceforge.



This option downloads the latest (bleeding edge) copy of the Kardia and Centrallix source code from Sourceforge.net. From the Setup Guide, just press [ENTER] when prompted, to initialize the shared repository.

Set Sourceforge.net Username...

To be able to commit changes back to Sourceforge.net from the shared (common) repository, you'll need to set up a Sourceforge.net account and let the Kardia VM know what your Sourceforge.net username is. If you do not have a Sourceforge.net username, you can leave this blank. In that case, you will still be able to receive software (source code) updates from Sourceforge, but will not be able to contribute your changes and enhancements back to Sourceforge.

Here, enter *just* your Sourceforge username. Do not enter a full email address.

Build and Run Centrallix and Kardia...

This option during the Setup Guide does several things:

- **Compile Centrallix.** This takes the Centrallix source code and compiles it into an executable program. It builds both Centrallix and its supporting library Centrallix-lib.
- **Install Centrallix.** Centrallix is installed, either in /usr/local (for running as Root), or in users' home directories in the "cxinst" subdirectory (for running as Users).
- **Build Kardia Database.** This runs several SQL scripts that installs the Kardia data model into a MySQL database.
- **Install Test Data.** Next, a small test data set is installed into the Kardia database.
- **Start Centrallix.** This launches the Centrallix application platform server, which will allow you to connect and start using Kardia.

It is normal at this time for there to be some errors displayed while building the database.

You can always re-compile and install Centrallix by using the Build option on the Devel menu in kardia.sh. You can re-install the Kardia database and test dataset by using the DBBuild option in the Devel menu. The DBBuild option will attempt to preserve existing data in the database, but if the data model has changed significantly (such as removed columns), some data may not be able to be re-loaded.

Centrallix can be started and stopped via the Start and Stop options in the Devel menu.

If you need to access the GDB debugger (when running in GDB Run Mode), you can select the Console option in the Devel menu in kardia.sh.

If you are running as Root, then the port Centrallix will be listening on is port 800. If you are running as Users, then each user will have their own Centrallix instance, and the port will be some number greater than 8000, such as port 8500. The Setup Guide will tell you what port Centrallix is running on, and what URL you should use to connect to it:



Connecting to Centrallix and Kardia...

At this point, just start Firefox and enter the URL shown in the dialog box above. You'll be presented with a welcome screen that has a few options on it:



Just click the first button on the list, labeled **Kardia**. This will take you to the main Kardia screen.

Developing using the Kardia VM Appliance...

One of the major reasons that we created the VM appliance was to help developers begin work on Kardia and Centrallix more easily, regardless of whether they run Windows or Linux or Mac, and regardless of their level of knowledge of how to install packages and package prerequisites on a Linux system.

To edit the source code on the VM Appliance, you have two choices:

- **Login with SSH.** If you're a command line and “vi” type person, you can log into the Kardia VM using SSH and edit files using “vi”. The directories you'll want to use are:

Shared Repository:

`/usr/local/src/cx-git`
`/usr/local/src/kardia-git`

User Repository (for Team or Individual workflows):

`/home/username/cx-git`
`/home/username/kardia-git`

- **Connect to File Shares.** If you have a nice IDE development environment that you like to use, such as Eclipse, the Kardia VM Appliance shares the source code repositories using Windows-type file sharing (CIFS / Samba). Just browse to the IP address of the VM Appliance and there should be either two or four shares. In **Shared** workflow mode, there will only be two shares, called “kardia” and “centrallix”. In **Team** or **Individual** workflow mode, there will be four shares, two each for Kardia and two each for Centrallix, with two being the shared repository and two for the user's repository.

If you connect via the file shares, you can usually use the GIT client that is built in to your development IDE instead of having to log into the Kardia VM Appliance and use the GIT push/pull/commit options that are in the menu system. **However**, the exception is pushing from an individual repository to the shared repository, or vice versa, in **Team** workflow mode. In that case, you should either use the `kardia.sh` menus to do the pushes and pulls, or you will need to change the origin URLs in the individual repository to point to the file share (rather than using the VM Appliance's internal directory names such as `/usr/local/src/cx-git`). For instance, you might set the origin to `\\192.168.244.132\centrallix-shared`. *Please note that if you do that, you can no longer push from the individual repository to the shared repository using the Kardia VM Appliance's menus (`kardia.sh`) until you change the origin URL back to what the VM Appliance expects.*

Another important note: If you are using the **team** workflow mode, whenever an individual team member pushes to the shared repository, it overwrites any changes that were made in the shared repository's working

directory. Normally you wouldn't be modifying the shared repository directly in **team** mode, so this shouldn't impact normal use.

A Bit about GIT...

If you're familiar with other source code control tools, such as CVS or Subversion, here's a quick overview of how GIT works. This is not a complete guide to GIT – for that, see the GIT documentation online.

GIT has several basic operations that you'll want to be aware of:

- **Clone.** This is what initializes a new copy of a source code repository. The Kardia VM Appliance does a Clone operation to initially obtain the source code to Kardia and Centrallix from Sourceforge.
- **Pull.** A pull operation, like “update” in CVS or Subversion, updates the local repository with any new things (changes, new files, deleted files) that are in the origin (remote) repository. You generally always want to Pull before you Commit and Push, to keep things clean. Sometimes a Pull will result in conflicts, if someone else changed a file in the same spot that you changed it. In those cases, Pull will advise you that there were conflicts, and that you have to edit the affected files manually to resolve the conflicts.
- **Add.** This adds new files to the local repository, so that GIT is aware of them. The file must already exist. In the Kardia VM Appliance, you are presented with a list of potential new files. *PLEASE do not add control files or debug files that are specific to your development IDE! Other developers do not want to see those files in the Sourceforge repository!* Only add files that are new source code files for Kardia or Centrallix, or which are needed to compile/build the system.
- **Commit.** In GIT, the “commit” operation that was used in CVS and Subversion is broken down into two operations: Commit and Push. This is partly so that you can do commits while you are not connected to the internet. When you Commit, it only affects your local repository – no changes have been sent to Sourceforge yet. But you do enter the commit log message here.
- **Push.** The opposite of Pull, this takes all committed changes and sends them to the origin repository. (Actually, you can Push almost anywhere you want, not just to the “origin”. You can push to other users, you can push over email, etc. Those operations aren't supported by the Kardia VM Appliance menus, but GIT does enable such things).

Kardia VM Appliance Menu Reference...

The main menu:

Devel	Here you'll find developer tools, such as starting and stopping Centrallix, and committing and updating the source code repository.
Config	Here you'll find lots of configuration options, including those that you encountered when using the Setup Guide initially.
System	This menu is only visible if you are logged in as root, or if you were given system admin privileges when your account was created. This menu contains some general system administration functionality, such as OS updates and shutting down the VM.
Users	If you are root or have system admin privileges, you can add, modify, and remove users using this menu option.
Password	For non-root users, the password can be changed from the main menu. <i>Note – currently there is an bug that causes the system to fail to change the Centrallix password when the user changes their password via this menu item.</i>

Shell For non-root users, a shell (prompt) can be obtained via this menu entry.

The Devel menu:

Build This builds and installs Centrallix and Centrallix-lib. You'll need to do this any time you make changes to the Centrallix or Centrallix-lib source code. If Centrallix is running at the time you do the Build, then Centrallix will be stopped first, and then automatically restarted when the Build successfully completes.

DBBuild This rebuilds the Kardia database. Any data that was in the database before is preserved, if possible. If columns have been deleted, or new required columns added, then the data that was in the affected tables will not be preserved.

Status This option displays what uncommitted changes you have made to Kardia or to Centrallix.

Commit This option allows you to add, commit, and push. See "A Bit about GIT" for more information about these operations.

Update This operation does a Pull from the origin. See "A Bit about GIT" for more information.

Restart This restarts the Centrallix server.

Start This starts the Centrallix server.

Stop This stops the Centrallix server.

Console If you are using the Console or GDB Run Mode, then this option allows you to view the Centrallix console, where errors or other messages may have been shown. It also allows you to access the GDB debugger if you are using the GDB Run Mode.

Log This shows a log of what was output to the console, if in Console or GDB Run Mode.

The Config menu:

Hostname This lets you set the host name of the VM Appliance.

RunMode See "Run Mode" earlier in this guide. If you change the Run Mode and Centrallix is already running, it will be stopped and then restarted in the new mode.

DevelMode See "Development Mode" earlier in this guide.

WorkMode Sets the Workflow Mode. See "Workflow Mode" earlier in this guide.

SUser Sets the username that will be used to push commits from the shared repository up to Sourceforge. See also "Set Sourceforge.net Username".

SInit Initializes the shared source repository. This overwrites any existing code there with the copy of the Kardia and Centrallix source code from Sourceforge.

IUser Sets the username that will be used to push commits from individual repositories to Sourceforge. This is only available in "Individual" workflow mode, and only works

when logged in to the Kardia VM as a user, not as root.

llnit

Initializes the individual repository. In Team mode, this overwrites the individual repository with whatever is in the shared repository. In Individual mode, this brings down a copy of what is on Sourceforge.