

Information and specifications published here are current as of the date of publication of this document. Tridium, Inc. reserves the right to change or modify specifications without prior notice. The latest product specifications can be found by contacting our corporate headquarters, Richmond, Virginia. Products or features contained herein may be covered by one or more U.S. or foreign patents. © 2008 Tridium, Inc.

## Linux AX Supervisor Notes

Starting in AX-3.4, support was added for an AxSupervisor running on Red Hat Enterprise Linux 5 distribution on a specific PC platform (contact your Tridium sales channel for further PC platform details). This document provides *installation* details specific to this Linux AxSupervisor. For related details on NiagaraAX platform operation, see the *NiagaraAX Platform Guide* section “Linux-based AxSupervisor.”

**Note:** *Note that the station running on a Linux AxSupervisor is “owned” by a specially created user/group niagarad:niagarad, and therefore cannot bind to Linux “root owned” software ports 1-1024. This is not an issue for the conventional port (3011) used for a platform connection, but does affect the standard port used by the station’s WebService (Http Port), which cannot be used at the default port (80) setting. In addition, other software ports potentially used by various drivers must be adjusted above port 1024.*

The following main sections are in this document:

- “Installing the Linux AxSupervisor” on page 1
- “Interacting with the platform daemon” on page 5
- “Uninstalling the Linux AxSupervisor” on page 6
- “Common Linux AxSupervisor issues” on page 6
- “Document change log” on page 7
- Appendix: “Example installation notes on Red Hat machine” on page 7

## Installing the Linux AxSupervisor

Once you have obtained the Linux AxSupervisor image, via CD or download, you will initially have a file on your machine named similar to `Vykon_AX_Supervisor_for_Linux_x86-3.4.n.zip`.

Expand this file—for example, change to its parent directory and enter the following command:

```
root@<host>: /home/<user># unzip Vykon_AX_Supervisor_for_Linux_x86-3.4.n.zip
```

This command expands the image and produces the directory hierarchy:

dist	folder
docs	folder
install-data	folder
lexicon	folder
modules	folder
overlay	folder
install.sh	shell script
INSTALL	installation instructions
README	README document
setup-linux-x86.tgz	Tar archive

Please review the README and INSTALL. When you are ready to begin the installation process, type the following at the command prompt:

```
root@<host>: /home/<user># bash install.sh
```

This script steps you through the installation process by asking a series of yes/no questions, where the default choice (press ENTER) appears in capital letters, that is YES or NO instead of yes or no. Some of the more complicated questions include:

```
Would you like to configure which users can use Niagara AX [YES/no]:
```

If you answer yes to this question, any user you provide will be added to the new group 'niagarad', have permissions to start/stop/query the NiagaraAX platform service, and have Desktop and Menu Icons installed to their home directory. If you choose not to configure any users, you'll still be able to run 'niagarad' and NiagaraAX, but it can only be controlled as 'root'.

If you want to add users later, use the command `usermod -G -a niagarad UserNameYouWantToAdd`

Should Niagara AX users be allowed to accomplish certain root privileged tasks?  
This is limited to setting system time, date, timezone, reboot, TCP/IP and NTP  
settings [yes/NO]:

This is a very important question, as is it only asked once, and can not be re-configured after the installation completes (unless you uninstall, then install again). If you answer yes to this question, then a setuid binary called 'ndsupport' will be installed in `${NIAGARA_HOME}/bin`.

Through this binary the wb, station or niagara daemon will obtain root privileges and perform the desired operation. If you choose not to install this executable, or if you delete it for some reason after you have installed it, then you will not be able to set or execute any of the listed operations through 'niagarad', 'wb', or 'station'.

Should Niagara AX add necessary "/etc/sudoers" information for the group  
niagarad? [yes/NO]:

This is a convenience. If you choose yes, then the `/etc/sudoers` entry required to allow members of the group 'niagarad' to execute `/usr/bin/niagaradctl` as the user 'niagarad' is automatically generated and added to the file `/etc/sudoers`. If you choose no, then the necessary entry will still be generated and logged, you will just need to manually modify the `/etc/sudoers` file through vim or visudo.

The remainder of the questions asked by the installer should not require explanation. The installer will ask you to review your choices and then it copies files. If the installer fails at any step, please send an email to Tridium support that specifies your Linux distribution, version, and the `install.log` file.



**Caution**

*Important information is included at the end of the installation script. Please take the time to read this information and complete any additional steps it may ask you to do.*

Once you have installed NiagaraAX, the following is in `/opt/Niagara/Niagara-3.4.n/`

bin	folder
certificates	folder
docs	folder
install	folder
install-data	folder
jre	folder
lexicon	folder
lib	folder
modules	folder
security	folder
stations	folder
sw	folder
uninstall	folder
workbench	folder
install.log	application.log

## Linux AxSupervisor's !/bin contents

In the AxSupervisor's `Niagara-3.4.n/bin` folder, find the following:

### axlauncher

This file is the shell script responsible for launching NiagaraAX applications. It is used primarily by the Desktop and Menu entries as a means to automate the creation of a suitable runtime environment (that is, env variables); although, a user could use this script from the command line if they choose.

If this file is deleted, Desktop and Menu shortcuts will no longer work. It is not advised to change the contents of this file.

Default attributes are: `-r-xr-x-- niagarad:niagarad`

## **build**

This executable is used to build NiagaraAX modules. It corresponds to the `build.exe` file in a Windows environment.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **lib\*.so**

These files are runtime libraries used by NiagaraAX. They correspond to the `*.dll` files in a Windows environment. Deletion of any of these files will result in NiagaraAX run time failure.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **jikes.122 / jikes**

These files are responsible for compiling Java code into NiagaraAX modules. 'jikes' is a wrapper script that forces command line expansion for compiling entire directories of java source. 'jikes.122' is a binary that corresponds to the 'jikes.exe' file in a Windows environment. Deletion of these files will prevent the successful completion of the 'build' executable, and you will not be able to create Program objects in NiagaraAX.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **.niagara**

This file sets up the run time environment needed to run any NiagaraAX binary. It will properly initialize several important variables, such as `LD_LIBRARY_PATH` and `NIAGARA_HOME`. This script can be sourced at the command line to allow you to launch NiagaraAX binaries, such as `wb`, from the command line at a later time. Deletion of this file will prevent Desktop and Menu shortcuts from functioning properly, as well as [axlauncher](#) and `niagaradctl`. Do not modify the contents of this file as it is custom generated at install time for NiagaraAX.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **niagarad**

This file is the niagara daemon, aka platform daemon. It corresponds to the file 'niagarad.exe' in a Windows environment. Do not execute this file manually, as it should only be controlled (e.g, started/stopped) by the script 'niagaradctl'. Deletion of this file will prevent the 'niagaradctl' script from functioning properly and will prevent you from making platform connections to your host in 'wb'. Do not delete or modify this file in any way.

Default attributes are: `-r-x----- niagarad:niagarad`

## **niagaradctl**

This file is the control script for 'niagarad'. It is used to start, stop, and query as to the status of 'niagarad'. Symbolic links from `/etc/init.d` and `runlevel` directories are made to this file at service installation time. This script may be executed by any users who are configured to use niagarad (done at installation time) to manually start, stop, restart or query the niagarad service. Deletion of this file will prevent 'niagarad' from running as a service, and will prevent 'niagarad' from starting at boot time, and stopping at restart/reboot time. Do not delete or modify the contents of this file in any fashion.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **niagarad\_generic**

This file is a cache of the original code used to generate 'niagaradctl' used by the 'install.sh'. The contents of this file are concatenated to 'niagaradctl' at it creation following the generation of platform dependent code during 'install.sh' and 'install\_service.sh' runtime. Deletion of this file, or more than trivial modification will prevent you from successfully reinstalling the platform service, i.e. 'install\_service.sh'.

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

## **niagaradlog**

Executing this script allows you to see any output from 'niagarad'. 'niagarad' is run as a service and therefore his standard output is redirected to an internal logging service. This script will locate the logs generated by 'niagarad' and print them to the screen. Deletion of this file will prevent you from seeing 'niagarad' output externally from NiagaraAX applications (that is, you'll still be able to see it in 'wb').

Default attributes are: `-rwxr-xr-x niagarad:niagarad`

### **nre**

This file is responsible for launching the Niagara Runtime Environment (NRE). It corresponds to the 'nre.exe' file in a windows environment. Deletion of this file will prevent you from being able to query the NRE externally from other NiagaraAX applications, for example for hostid.

Default attributes are: -rwxr-xr-x niagarad:niagarad

### **station**

This file is responsible from launching NiagaraAX stations on your platform. It corresponds to the 'station.exe' file in a Windows environment. Deletion of this file will prevent you from launching any stations on your local platform, pretty much rendering your AxSupervisor useless.

Default attributes are: -rwxr-xr-x niagarad:niagarad

### **test**

This file is responsible for running predefined test cases against the NRE. It corresponds to the 'test.exe' file in a Windows environment. This file is not critical, but deletion of this file will prevent you from testing the Framework.

Default attributes are: -rwxr-xr-x niagarad:niagarad

### **wb**

This file is responsible for launching wb, that is Workbench. It corresponds to the file 'wb.exe' in a Windows environment. Deletion of this file will prevent you from launching wb.

Default attributes are: -rwxr-xr-x niagarad:niagarad

## **Linux AxSupervisor's !/install contents**

In the AxSupervisor's Niagara-3.4.n/install folder, find the following:

### **install.sh**

This file is a copy of the script used to initially install the Linux AxSupervisor, it currently serves no purpose and is going to be removed. Default attributes are: -rwxrwxr-x root:root

### **install\_service.sh**

This script is responsible for installing the NiagaraAX Platform Service, aka 'niagarad'. You will need root privileges to successfully run the script. It can be run as a stand-alone installer and does not require or rely on 'install.sh' at all. When run, the script will install (but *not start!*) the platform daemon of that version of NiagaraAX, overwriting any other NiagaraAX daemon currently installed on that platform.

This means you can use this script to switch between installations of NiagaraAX on your platform. Once completed, you may start the service from the command line or simply reboot. The script relies on 'niagarad\_generic', and as such, might fail to function properly if that file is significantly modified. If deleted, you will be unable to switch the platform daemon between multiple NiagaraAX installations on your machine. Aside from the fact that this script does not start the daemon upon completion, it corresponds to the 'Install Platform Daemon' command available in a Windows environment. For details, see ["Installing the platform daemon"](#) on page 5.

Default attributes are: -r-xr-x--- niagarad:niagarad

## **Linux AxSupervisor's !/uninstall contents**

In the AxSupervisor's Niagara-3.4.n/uninstall folder, find the following:

### **uninstall\_service.sh**

This script is responsible for uninstalling the NiagaraAX Platform Service, aka 'niagarad'. You need root privileges to successfully run the script. It can be run as a standalone uninstaller, and will not remove the complete NiagaraAX installation from your platform, just the service. This script is provided as a convenience for users who would like to prevent 'niagarad' from starting at boot-time, or would like to disable the NiagaraAX installation temporarily, rather than remove it completely. Modification of this file may prevent it from successfully completing. For details, see ["Uninstalling the platform daemon"](#) on page 6.

Default attributes are: -r-xr-x--- niagarad:niagarad

### **uninstall.sh**

This script is responsible for completely removing the Linux AxSupervisor from your platform. In a Windows environment, this script corresponds to 'uninstall.exe'. You will need root privileges to successfully run the script. It steps you through the uninstallation process via a series of yes or no questions. In order to safely and efficiently remove NiagaraAX, it uses a configuration file generated at installation time

named 'uninstall.conf'. For details, see “Uninstalling the Linux AxSupervisor” on page 6. However, if this file is missing, the uninstaller can still be executed successfully (but it may ask unnecessary questions, and/or report incorrect information). If this file is significantly modified, you may be unable to remove NiagaraAX via an automated process—and you just have to do it manually file by file.

Default attributes are: -r-xr-x--- niagarad:niagarad

### uninstall.conf

This file is created at installation time based on the configuration of NiagaraAX (install service, doc, dist, etc.) you chose to install. The key/value pairs in this file are used to efficiently remove NiagaraAX when running the script 'uninstall.sh' by asking questions tailored for your platform. Modification or deletion of this file will not prevent you from uninstalling NiagaraAX, but it might cause the script 'uninstall.sh' to ask questions, and attempt to remove items, not tailored for your platform.

Default attributes are: -rw-r--r-- root:root

## Interacting with the platform daemon

Once NiagaraAX has been installed, you will be able to control the platform daemon, 'niagarad', through the control script 'niagaradctl'. This script can be found in the bin directory of your NiagaraAX Installation or from the symbolic link '/usr/bin/niagaradctl'.

Try typing 'niagaradctl' at the command line. You will see:

```
<user>@<host>:/usr/bin# niagaradctl
Usage: naxd {start|stop|restart|status}
```

Where 'naxd' stands for “NiagaraAX Daemon”, and it is the name presented to the operating system to associate with this service (niagarad). As you can see, with this script, you will have the ability to start, stop, restart and query the platform daemon.

### Starting the platform daemon

To start the platform daemon on a Linux AxSupervisor, type the following at the command line.

```
<user>@<host>:/usr/bin# niagaradctl start
Starting Niagara Daemon: [ OK ]
```

To verify that the daemon has started, type the following at the command line.

```
<user>@<host>:/usr/bin# niagaradctl status
niagarad (pid 4815162342) is running...
```

The platform daemon is now running on your platform and can accept platform connections.

### Stopping the platform daemon

To stop the platform daemon on a Linux AxSupervisor, type the following at the command line.

```
<user>@<host>:/usr/bin# niagaradctl stop
Stopping Niagara Daemon: [ OK ]
```

To verify that the daemon has stopped, type the following at the command line.

```
<user>@<host>:/usr/bin# niagaradctl status
niagarad is stopped
```

**Note:** Typically you rarely stop the platform daemon (and thus station) on any AxSupervisor, except immediately before installing a new point release, meaning an upgrade.

### Installing the platform daemon

To reinstall the platform daemon on a Linux AxSupervisor for a particular version of NiagaraAX you will have to execute the script 'install\_service', found in NiagaraAX's "install" directory. It is not necessary to execute this script after a successful NiagaraAX installation, as that process will automatically install the platform daemon.

So the main reasons you will be executing this file are either:

1. You have accidentally broken your NiagaraAX environment and you want to repair it, or
2. You have two versions of NiagaraAX installed on your platform, and you want to change which platform daemon is started/controlled by '/usr/bin/niagaradctl'.

You need to be acting as root to accomplish this task. The first step to installing a particular version's platform daemon is to change to the target release's install directory:

```
root@<host>:/usr/bin# cd <targetNiagaraReleaseDirectory>/install
```

Then, you issue the “install\_service.sh” command. For example, if installing the platform daemon from `opt/Niagara/Niagara-3.4.25/install`:

```
root@<host>:/opt/Niagara/Niagara-3.4.25/install# bash install_service.sh
```

The platform daemon (service) will be installed, but not started. See “Starting the platform daemon”.

### Uninstalling the platform daemon

To disable the platform daemon at start up, that is remove the NiagaraAX scripts in “/usr/bin” and “/etc/init.d”, you have to execute the script ‘uninstall\_service’, found in NiagaraAX’s “uninstall” directory. It is not necessary to execute this script prior to running NiagaraAX’s main uninstallation script, as it will uninstall the service, and more, on its own.

You need to be acting as root to accomplish this task. The first step to uninstalling a particular version’s platform daemon is to change to the target release’s `uninstall` directory:

```
root@<host>:/usr/bin# cd <targetNiagaraReleaseDirectory>/uninstall
```

Then, you issue the “uninstall\_service.sh” command. For example, if uninstalling the platform daemon from `opt/Niagara/Niagara-3.4.25/uninstall`:

```
root@<host>:/opt/Niagara/Niagara-3.4.25/uninstall# bash uninstall_service.sh
```

The platform daemon (service) is then uninstalled. The service can be safely reinstalled using the ‘install\_service.sh’ command. See “Installing the platform daemon”.

## Uninstalling the Linux AxSupervisor

To completely uninstall a version of NiagaraAX from your Linux AxSupervisor machine, you have to execute the script ‘uninstall.sh’, found in NiagaraAX’s “uninstall” directory.

You need to be acting as root to accomplish this task. The first step to uninstalling a particular version’s platform daemon is to change to the target release’s `uninstall` directory:

```
root@<host>:/usr/bin# cd <targetNiagaraReleaseDirectory>/uninstall
```

Then, you issue the “uninstall.sh” command. For example, if uninstalling Niagara-3.4.25 from your machine, from `opt/Niagara/Niagara-3.4.25/uninstall`:

```
root@<host>:/opt/Niagara/Niagara-3.4.25/uninstall# bash uninstall.sh
```

The uninstallation procedure is very similar to that of the Windows environment, and the script steps you through the process. Once finished, please make sure you read and following any additional instructions that were not automated by the script.

## Common Linux AxSupervisor issues

The following are some issues previously encountered when installing or running a Linux AxSupervisor:

- **error while loading shared libraries**  
This error can happen with any of NiagaraAX’s shared libraries: `libnre.so`, `libplatform.so`, or `libjvm.so`. Although these libraries are probably installed on your machine, the shared library loader can not locate them. This mostly likely means you have not sourced the script ‘.niagara’. As user `root`, from your installation’s `/bin` directory, type this:  

```
root@<host>:/opt/Niagara/Niagara-3.4.25/bin# ./niagara
```

  
Then try whatever command you used to produce this error again (`wb`, `nre`, `niagarad`, `station`).
- **java.net.BindException**  
This error can happen when starting a station. In the Linux OS, ports 1-1024 are “root ports,” meaning only an application running with `root` privileges may bind to them. Since ‘station’ is running as the user ‘niagarad’ (who does not have root privileges), it can not bind to port 80. If you have not configured your station to start on a port above 1024, you will see this error.  
To fix this error, configure your station to start its WebService bound to port 8080, or another port above 1024.
- **[workbench.auth] Cannot save credentials list**  
This error can happen while connecting to the local platform daemon from Workbench. In the Linux OS, an item owned by `root:root` usually can not be modified by any users other than `root` itself. If `root` owns the `users` directory, or if the permission of the `users` directory has been modified somehow to something other than `0775`, new user credentials can not be created.  
To fix this error, as user `root` from your installation’s directory, enter these commands:



```
root@<host>:/opt/Niagara/Niagara-3.4.25# chown -R niagarad:niagarad users
root@<host>:/opt/Niagara/Niagara-3.4.25# chmod 0775 users
```

In errors involving not being able to write or create something associated with a file, always try modifying a file, or directory, so that it is owned by `niagarad:niagarad`, as this is most likely the cause of the error.

## Document change log

Updates (changes/additions) to this *Linux AxSupervisor Notes* engineering notes document are listed below.

- Updated: September 30, 2008  
Following the first introductory paragraph, added a [Note](#): about port usage. Added a new appendix [“Example installation notes on Red Hat machine”](#).
- Draft: May 28, 2008  
Initial “Engineering Notes” type document.

## Example installation notes on Red Hat machine

The following notes include steps recorded by Tridium Systems Engineering (Test department) in the initial software setup of a Dell Precision Workstation ordered installed with Red Hat Linux 5 Client, and also the subsequent installation of the NiagaraAX Linux Supervisor.

- [Initial Dell workstation setup example](#)
- [Linux AxSupervisor installation example](#)

## Initial Dell workstation setup example

Following the unpacking of the Dell Precision Workstation, interconnection of keyboard, mouse, display, Ethernet, and power (following Dell documentation), the system was powered on. The following steps were taken and recorded on June 3, 2008:

1. Pressed Enter at EULA screen.
2. Selected **English (USA)** as Language, and pressed **Forward**.
3. Clicked **Yes, I agree** on License Agreement, and pressed **Forward**.
4. Selected **US English** for Keyboard layout, and pressed **Forward**.
5. Set root password and confirm password fields to “niagara”. Pressed **Forward**.
6. Pressed **Change Network Configuration**, selected the `eth0` device, and clicked **Activate**. (The Ethernet card was shipped deactivated).
7. Under the **Firewall Configuration**, clicked drop-down for **Other Ports**, and added ports 3011 and 80 for *both TCP and UDP*. Pressed **Yes** to override the existing firewall configuration.
8. Left SELinux at its default **Enforcing** setting. Pressed **Forward**.
9. Left **Enable Kdump** unchecked. Pressed **Forward**.
10. Set time zone to **Eastern Time**. Pressed **Forward**.
11. Under **Date and Time**, selected the **Network Time Protocol** tab and checked **Enable Network Time**. Pressed **Forward**.
12. The software update failed because the system could not detect an active network connection. Pressed **Forward**.
13. Under **Create User**:
  - Set username = `tridium`
  - Set Full Name = `tridium`
  - Set password = `niagara`
  - Set confirm password = `niagara`
  - Pressed **Forward**.
14. Pressed **Forward** under **Sound Card**.
15. Pressed **Finish** under **Additional CDs**.
16. Entered `tridium/niagara` at login screen.
17. Rebooted the machine. This corrected the inactive network connection issue detected at step 12. The initial setup had failed to acquire a gateway for the network connection.

18. Attempted to load the self-extracting RPM Java plugin for Linux from java.com, however, the pre-installed Firefox (ver. 1.5.0.9) failed to detect the plugin link.

**Note:** *The Red Hat workstation came preconfigured with a pre-3.0 version of Firefox that does not detect the Java plugin previously mentioned. The fix is to upgrade to the latest version of Firefox. This is recommended if the "Web Workbench" (Workbench in a browser) will be run from the workstation.*

## Linux AxSupervisor installation example

Following the initial software setup of the Dell workstation (see previous section "[Initial Dell workstation setup example](#)" on page 7), the following steps were taken to perform a "fresh install" of the "AX Supervisor for Linux", build 3.4.42. These steps were recorded on September 30, 2008:

1. Downloaded the zipped image of build 3.4.*n* (3.4.42 as of September 30, 2008), saving it to the main desktop.
2. Created a folder on the main desktop named "builds", with a subfolder named "3 . 4 . 42".
3. Double clicked on the zip archive from the desktop to launch the extraction utility. Specified the "3 . 4 . 42" subfolder as the destination for extraction.
4. Following extraction, the "3 . 4 . 42" folder should contain the standard directory hierarchy for NiagaraAX.
5. Launched a new terminal session. The default prompt is the "tridium" user created during the original Red Hat installation.
6. Changed to the root user using the "su" command. Entered "niagara" when prompted for the password. The prompt switches to root.
7. Changed to the "3 . 4 . 42" directory created in Step 3. using command:  

```
cd /home/tridium/Desktop/builds/3.4.42
```
8. Ran the install.sh script by issuing command  

```
bash install.sh
```
9. Pressed Enter when prompted to continue by the script. Pressed "Q" to skip viewing the license agreement.
10. Typed "yes" followed by ENTER to accept the license agreement.
11. Pressed ENTER to accept the default absolute path for Niagara AX 3.4.42.
12. Pressed ENTER again to accept the prompt to create the directory, since it did not exist.
13. Pressed ENTER to accept the prompt to configure a new user.
14. Entered "tridium" as the username string when prompted by ENTER.
15. Typed "no" followed by ENTER to skip adding another user.
16. Typed "yes" followed by ENTER when prompted about whether users should be allowed to accomplish root privileged tasks.
17. Typed "yes" followed by ENTER when prompted to add niagarad to the /etc/sudoers file.
18. Typed "yes" for the prompts to install GNOME desktop and menu shortcuts.
19. Pressed ENTER when prompted to install lexicon files.
20. Pressed ENTER when prompted to install documentation.
21. Pressed ENTER when prompted to install as installation tool.
22. Verified the installation settings summary displayed, and pressed ENTER to accept and continue.
23. A warning message prompting the configured user to logout and log back in to workstation is displayed in blue.



**Caution** *At this stage of the installation, you must either:*

- *Close the terminal session, and log out and log back in, or*
- *Reboot the workstation*

*Prior to launching NiagaraAX (from the console or from desktop shortcuts). Otherwise, you will likely have permission denial errors.*

24. Verified the creation of desktop and menu shortcuts.
25. Exited the console.
26. Logged out of the workstation and logged back in (using credentials tridium/niagara).



**Note:** *The niagara daemon does not start up automatically when a user logs out and back in. It does however start up automatically when the workstation is rebooted. To start niagarad without rebooting:*

- Double-click the Niagara AX 3.4.xx Console icon to launch the NiagaraAX console.
  - Type “niagaradctl status” for the current status of the niagara daemon.  
It should say “niagarad is stopped”.
  - Type “niagaradctl start”  
The daemon starts up with an OK status.
  - Now if you type “niagaradctl status”, it should say “niagarad (pid xxxx) is running...”
27. Workbench can now be launched in a number of ways:
- From either of the Workbench icons on the desktop, or from the **Applications Menu** (the suggested way).
  - Running “wb” from the console.
28. Testing confirmed that all methods successfully launched Workbench, and could successfully open a connection to the local platform.