ARIZONA STATE UNIVERSITY

FLETCHER LIBRARY AT THE WEST CAMPUS

Version 2.0

DRAFT

March 9, 2005

Forward

This document is a draft of the "Build Station" component of the E3 project version 2. The three component documents consist of the Build Station, Development Station, and Boot Server. Information contained within this document may include errors and inconsistencies which will be edited for the final version. Additions, changes, and deletions may occur before the final version.

A forthcoming development portal Web site is scheduled to go online during the summer of 2005. This site will allow other institutions to download, contribute, and exchange information, files, and contributions to the project.

Version 1 project documentation can be retrieved from http://wlibinfo.westlib.asu.edu under past projects.

Perry Horner
Coordinator of Library Technology Support & Development
ASU Fletcher Library at the West Campus
p_horner@asu.edu

March 9, 2005

ASU West Library Public Workstation Version 2

The following outlines the creation of a client image for network booting. Some areas may be more detailed than others.

Create From Scratch

Version 2 uses the stock Fedora Core 2 distribution as the building block for our custom system. We begin by installing onto a fresh hard drive using the stock anaconda installer.

1. When it comes to creating drive partitions, do not use up all remaining space. Make the partitions the sizes for the client RAMdisk image. Install only the packages you want to use on the client image. We will call this machine the BUILD-STATION.
2. Perform a second installation on a different machine which uses the same identical hardware. This installation you will want to load all of the development utilities and libraries to build custom software for the client image. You can use the same machine for both if you have a removable hard drive tray installed in the case of the system. We will call this machine the DEV-STATION.
3. Build a third machine that does not have a hard drive in it. This will be a copy of what a client workstation would be. We will call this machine the CLIENT-TEST-STATION.

After both installations are complete, set the `/etc/yum.conf` file to point to the latest yum archive and perform a `yum update` on both systems. This will update both systems to the latest versions of their respective installed software.

NOTE: *Any script that you modify you should include a unique word in a comment field. We use"* E3-*" for remarks, comments, and script prefixes. This helps when you want to* `grep` *for what has been customized.*

DEV STATION

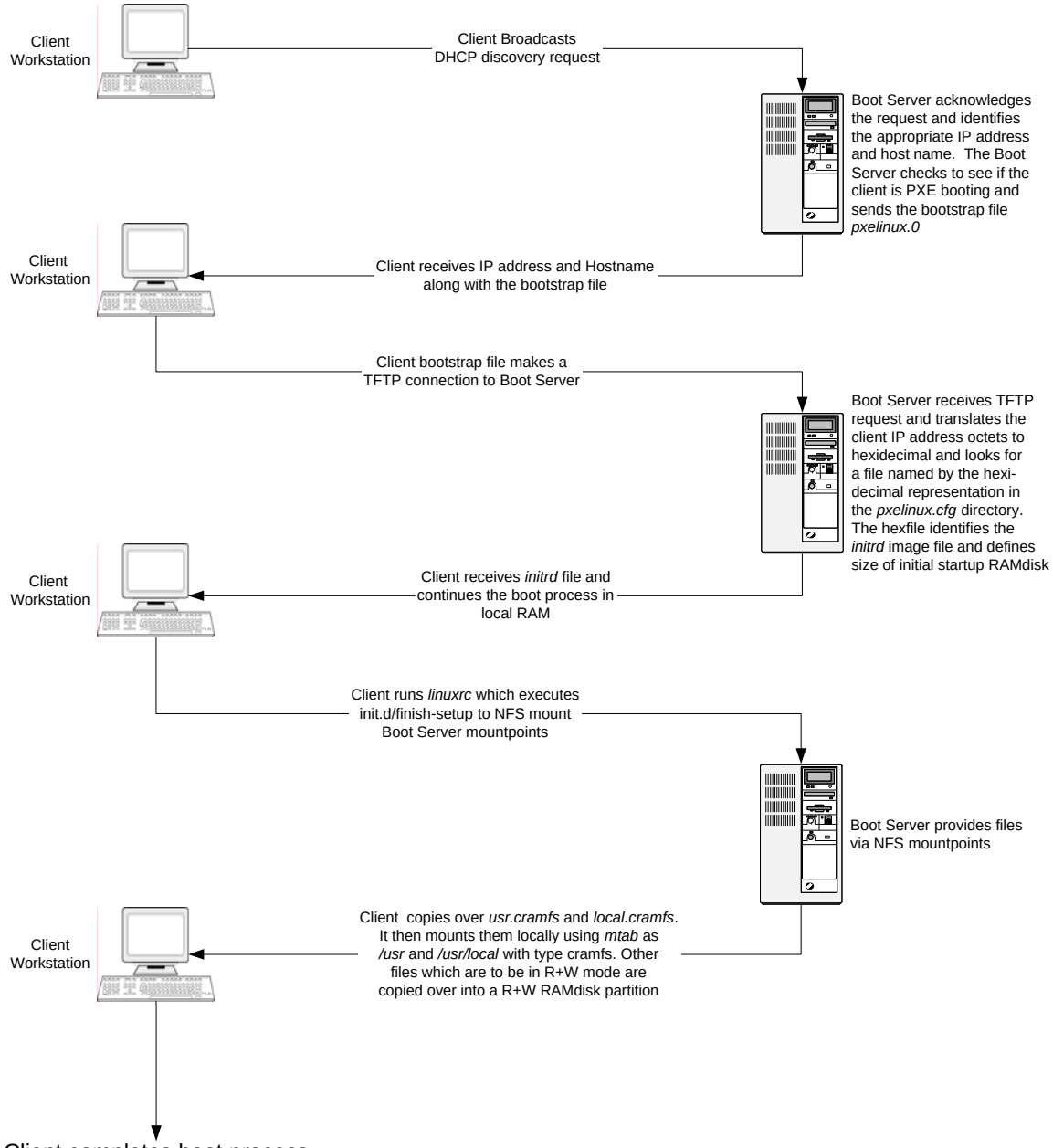
This station is to be used to compile custom apps and source code distributions. These applications can then be copied over to the Build Station.

BUILD STATION

The Build Station contains all of the components of the Client Station with the addition of a hard drive and dvd rom drive. This station has special scripts which are used to create the boot image served by the boot servers.

You will want to perform all of your software installations using the *root* account. Any preferences you set will be stored in resource files in the `/root` directory. These preferences can then be copied over to `/etc/skel` which will be copied to each new workstation user's `/home` directory for default settings.

Client Boot Process



Changes & Enhancements

Fletcher Library
Public Machine Image Version 2.0
Spring 2005

Version 2.0 of the Library's PAC image has been developed from the ground up. TS&D decided not to build upon the previous 1.5 image since no documentation existed covering its development and specifications. Version 2.0 may look similar to version 1.5, but under the hood, it is a completely different system. TS&D has created the development documentation at every step during the build process. The amount of time invested in its development is almost half that of the Version 1. This is a milestone in our process. Version 2 took into account all issues and enhancement requests placed during the past 9 months as well as feedback from end users through the "Talk Back" feature on each workstation.

The following is a descriptive of what Version 2.0 has to offer. An item identified as NEW indicates something that was not on previous versions. UPDATED indicates the item has been upgraded to a new version or a bug has been fixed. REMOVED identifies an item that was on previous images but is not included in Version 2. UPGRADED indicates hardware and device enhancements.

Area	Type	Version 1.x	Version 2.0	Comment
Hardware	UPGRADED	MSI motherboard	ASUS motherboard	Better motherboard
	UPGRADED	1GB PC133 RAM	2GB PC2100 DDR RAM	More and faster RAM
	UPGRADED	Generic Power supply	High Temp rated power supply	Lasts longer and is more quiet
	UPGRADED	8MB video	64MB video	Faster rendering
	UPGRADED	1ghz AMD Duron CPU	AMD Athlon 2200+ XP CPU	Faster processor
	NEW		Multimedia Reader	Read/write to micro devices
	NEW		Sound	Listen through headphone jack
Operating System	UPDATED	2.4.x kernel	2.6.x kernel	Better Linux
	UPDATED	KDE 3.1.x	KDE 3.3.x	Desktop environment
	UPGRADED	Automounter	Submount	Better mounting of removable media
Web Browser	UPDATED	Firefox 0.93	Firefox 1.0	Stable final release
	UPGRADED	Cute JR Firefox theme	Qute 3 Firefox theme	Stable with version 1.0
	UPGRADED	SearchThis context menu	ConQuery context menu	Better search tool
	NEW		Image Toolbar	Context menu for images
	NEW		PrintIt!	Context menu for printing
	UPDATED	Translate	Translate	Language translator
	UPDATED	User Agent Switcher	User Agent Switcher	IE, Netscape, Firefox, etc.
	-	Tab X	Tab X	Close tabs
	NEW		MAF	View and save in the Mozilla archive file format
	UPDATED	PrefBar	PrefBar	Provides bar for allowing popups, etc.
	UPDATED	Java 1.4.2	Java 1.5	

Area	Type	Version 1.x	Version 2.0	Comment
Helper Applications	UPDATED	DjVu	DjVu	
	UPDATED	Flash Player 6	Flash Player 7	
	UPDATED	Real Player 7	Real Player 10	
	UPDATED	Acrobat 5.05	Acrobat 5.09	
		Microsoft Word Viewer	Microsoft Word Viewer	
		Microsoft PowerPoint Viewer	Microsoft PowerPoint Viewer	
		Microsoft Excel Viewer	Microsoft Excel Viewer	
	NEW		Quicktime	
	NEW		Microsoft Windows Media Player	
	NEW		Shockwave	
	NEW		AuthorWare Player	
	NEW		Ebrary Reader	Not exactly stable
	NEW		iPIX viewer	
	NEW		Scorch viewer	Does not output MIDI
	NEW		divX codecs	
Login	UPDATED	Crystal	Keramik	New login screen
	UPDATED	Blue background	Textured background	Background screen flows from login to session
	UPDATED	Library background	Textured background	
Desktop	UPDATED	AFS Network space icon	New AFS icon	
	NEW		News Ticker	RSS fed tattle tape linked to Web articles
	UPDATED	Library background	Textured background with hardware guide	So users know where to put their devices

Allowed Pop-up Sites

0-journals.iucr.org
web6.epnet.com
www.fofweb.com
brs.newsbank.com
publish.aps.org
www.chemnetbase.com
www.tpcincweb.com
web5.silverplatter.com
humanities.uchicago.edu
www.euromonitor.com
www.jstor.org
www.library.uiuc.edu
www.update-software.com
aio.anthropology.org.uk
arba.odyssi.com
www.asme.org
www.netlibrary.com
www.edrs.com
0-*
www.kcdlonline.com
www.netadvantage.standardandpoors.com
www.biologists.com
www.alexanderstreet2.com
web.lexis-nexis.com
ejournals.ebsco.com
www.booksinprint.com
app.harpweek.com
www.biomedcenter.com
www.grovemusic.com
premium.hoovers.com
rdsweb2.rdsinc.com
arjournals.annualreviews.org
www.hbcpnetbase.com
www.aip.org
proquest.umi.com
www.groveart.com
www.emeraldinsight.com
www.dgbiblio.unam.mx
www.edpsciences.org
sanborn.umi.com
library.cqpress.com
journals.cambridge.org
portal.acm.org
lion.chadwyck.com
juno.emeraldinsight.com
www.gpoaccess.gov
onlineedition.culturegrams.com
ca1.csa.com
poolesplus.odyssi.com
www.stat-usa.gov
infotrac.galegroup.com
www.ciaonet.org
gateway.ut.ovid.com
www.asee.org
accuweather.ap.org
www.engineeringvillage2.org
www.datagold.com
www.blackwell-synergy.com
www.accessible.com
www.cios.org
www.artstor.org

Evaluation Checklist

Login Screen

- ❑ Fonts readable
- ❑ Heading instructions clear
- ❑ Background and theme are pleasant and inviting
- ❑ Transition from login to desktop session is smooth

Desktop

- ❑ Background screen appropriate
- ❑ Icon placement satisfactory
- ❑ Icon labels satisfactory
- ❑ AFS icon appears with ASURITE login
- ❑ AFS icon does not appear with generic login
- ❑ Right click on desktop context menu disabled
- ❑ Right click on kicker bar context menu disabled
- ❑ Cannot delete desktop items
- ❑ Icons appropriate
- ❑ Drag and drop files between disks and storage devices
- ❑ Screensaver initiates after 10 minutes of no keyboard or mouse activity
- ❑ Screensaver can be canceled by moving the mouse or pressing a key

Kicker Bar

- ❑ Kicker bar buttons correctly placed
- ❑ Kicker bar buttons appropriate and understandable
- ❑ News ticker connects to alert
- ❑ Clock is readable
- ❑ Clock time correct
- ❑ Talk Back button goes to feedback Web page
- ❑ Open Web button goes to acceptance page for ASURITE
- ❑ Open Web button goes to guest user info page for generic user
- ❑ Disk Formatter button opens disk formatter application
- ❑ Logout Button logs out session
- ❑ Kill button terminates all Windows applications

Applications

- ❑ Disk Formatter application formats disks
- ❑ Firefox defaults with all themes and extensions loaded
- ❑ Adobe Acrobat should default save to user Home directory

Firefox Web Browser

- ❑ Default saves to user HOME directory space
- ❑ Print command launches Pharos client for naming print job
- ❑ Standard documents launch appropriate helper application
- ❑ Web sites can be accessed using the User Agent Switcher
- ❑ ASU Blackboard behaves correctly
- ❑ Printed items default with footer location id

Multimedia

- ❑ Audio and video files launch appropriate players
- ❑ Audio is at a comfortable listening level
- ❑ Audio can be heard
- ❑ Video plays back with audio

Web Services

- ❑ Java applications work
- ❑ Blackboard works
- ❑ Subscription databases work

File System

- ❑ Cannot save to anyplace but HOME directory and devices
- ❑ Cannot execute binary applications [local or remote]
- ❑ Cannot traverse file system
- ❑ Filling up file space does not crash system

Installation PART 1

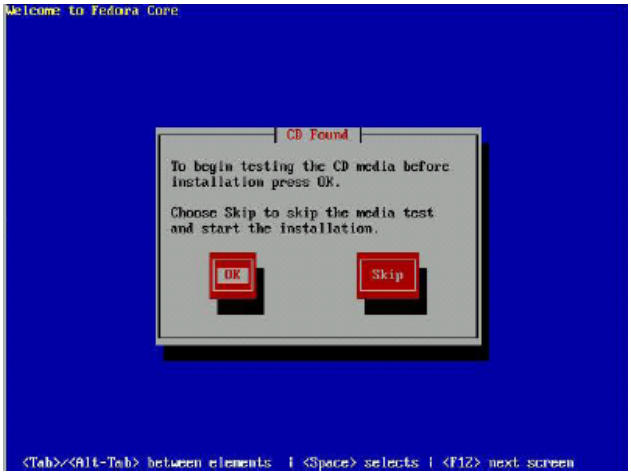
INSTALLATION CDs
Install *Fedora Core 2*

Installation Procedures

Insert the Fedora Core 2 disk 1 CD or DVD
Boot the machine

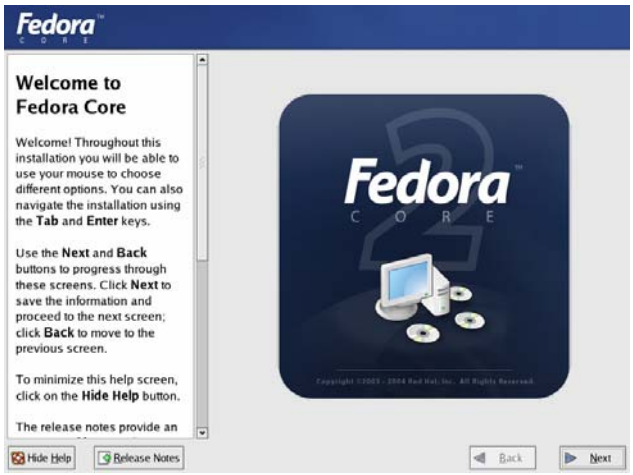
At the installer boot prompt, press the ENTER key to start installation in a graphical environment.

Select SKIP to skip the CD media testing (this will take a long time to test if you just select OK)



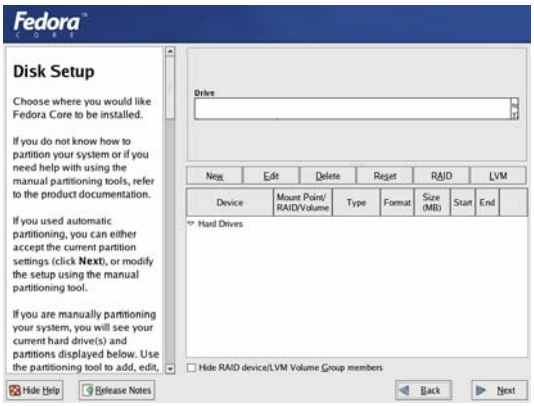
Follow the prompts for setting your keyboard, mouse, location.

Select *Custom Install*



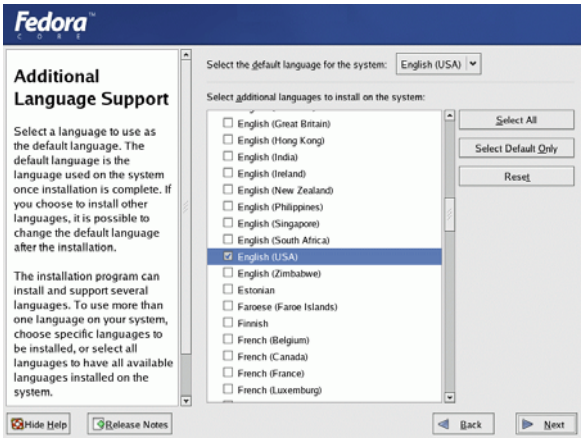
Partitions (on a 20GB drive)

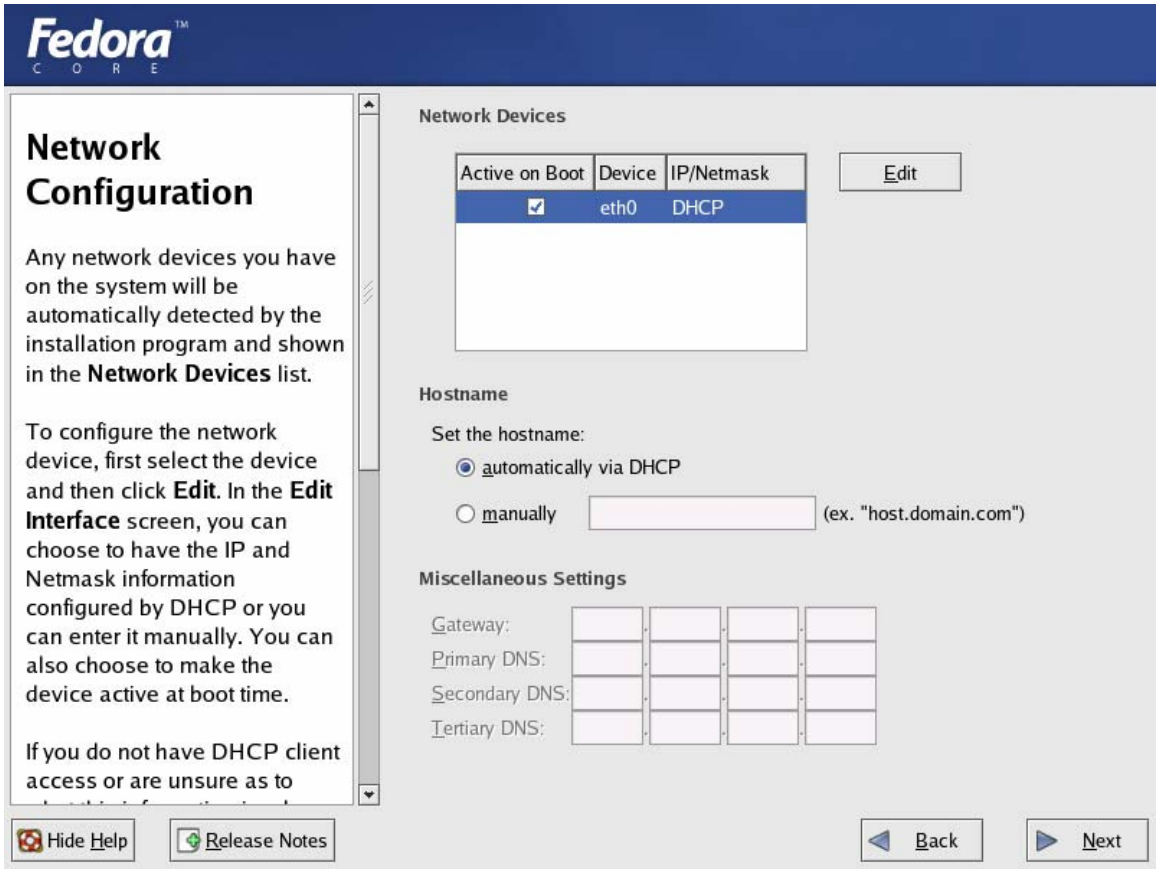
`/boot`	`200MB`
`/swap`	`4000MB`
`/`	`15000MB`



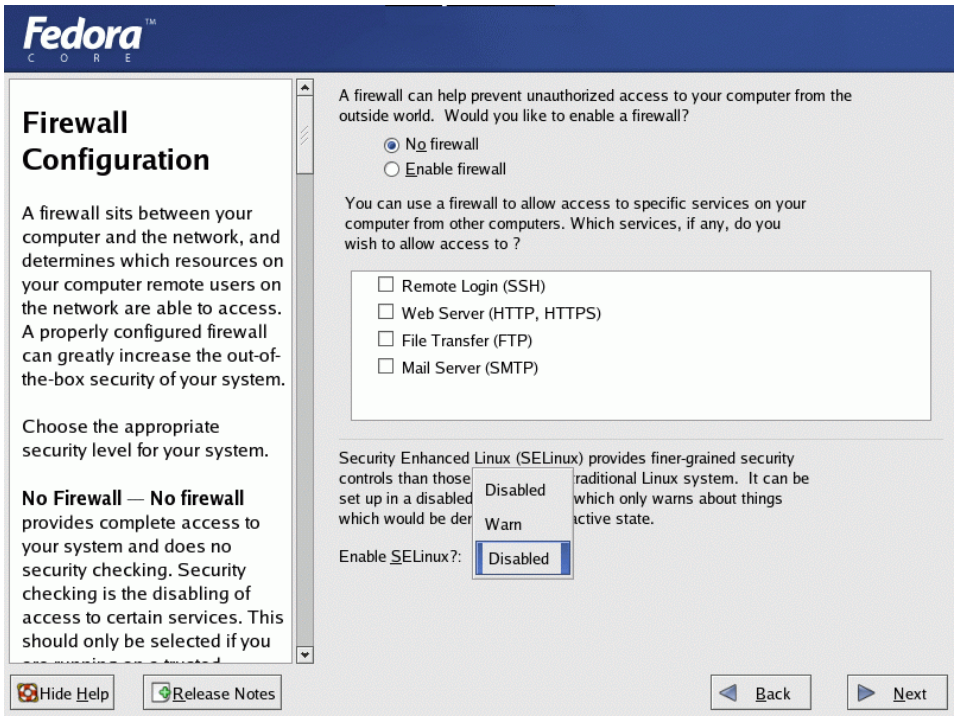
Additional Languages [each takes up a lot of space. Choose wisely!]:

- Arabic (Egypt)
- Chinese (P.R. of China)
- English (USA)
- French (France)
- German (Germany)
- Greek
- Hebrew (Israel)
- Hindi (India)
- Italian (Italy)
- Japanese
- Korean
- Russia
- Spanish (USA)
- Swedish (Sweden)
- Vietnamese

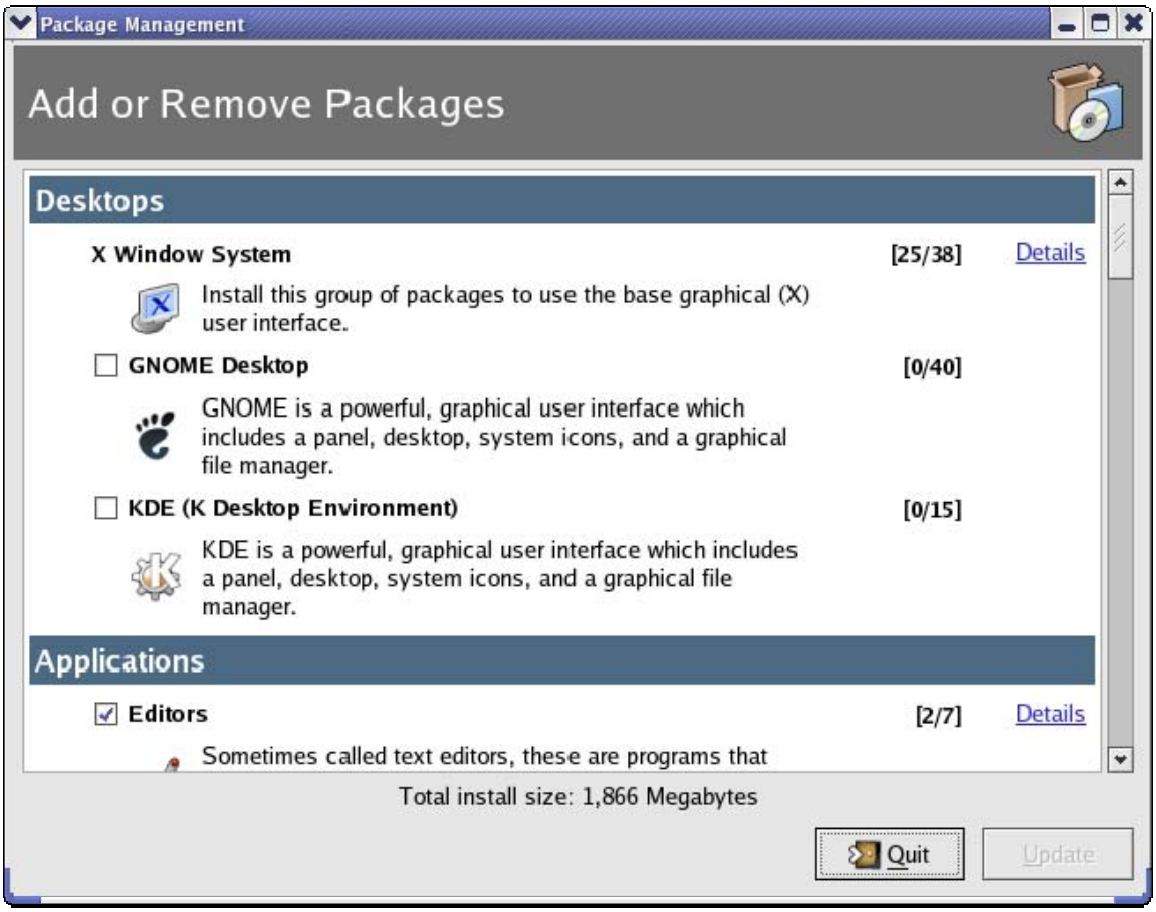




Select DHCP for network configuration and NO firewall.



Select Packages to Install:



X Windows System

- Bitstream-vera-fonts
- Firstboot
- Openssh-askpass
- Rhgb
- System-config-* (all)
- Xterm

Gnome

- File-roller
- Gtk-engines
- Gtk2-engines

KDE

- Kdeaddons
- Kdeadmin
- Kdeartwork
- Kdegraphics
- Kdemultimedia
- Kdenetwork
- Kdeutils

Editors

- Vim-common
- Vim-enhanced

Graphical Internet
- (none)

Text-based Internet
- (none)

Sound and Video
- Kdemultimedia
- Mikmod
- Vorbis-tools
- Xmms

Authoring and Publishing
- (none)

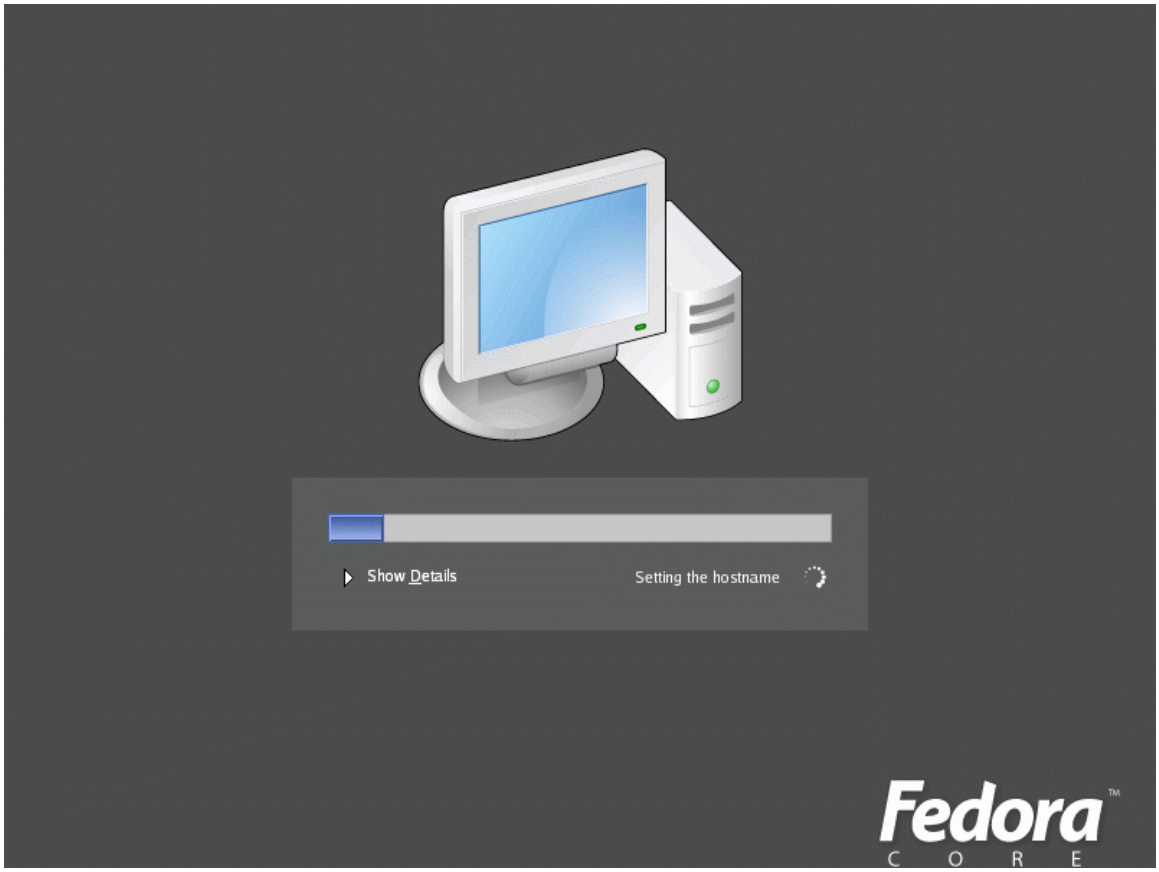
Graphics
- Imagemagik
- Kdegraphics

Games and Entertainment
- (none)

Administration Tools
- System-config-* (all)

Printing Support
- A2ps
- Cups
- Enscript
- Hpijs
- Tttprint

After installation is complete, select the REBOOT button to restart the BUILD machine.



End of PART 1

Installation PART 2

After the final REBOOT from PART 1:

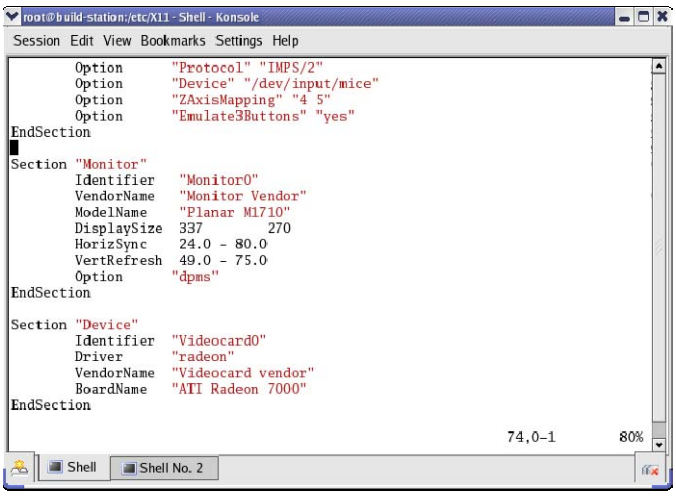
Login as `root` at the greeter prompt
Open a terminal window (console, xterm, etc) to get a shell prompt
Execute `switchdesk kde` to set the default desktop to KDE

Initial Package Removal
We need to remove some packages that were installed during PART 1. Do the following in the terminal window:

```
rpm –e firstboot
rpm –e desktop-printing
rpm –e gnome-utils
rpm –e gnome-session
rpm –e control-center
rpm –e --nodeps nautilus-cd-burner
rpm –e nautilus
rpm –e gnome-ufs2-smb
rpm –e eog
```

X Server Configuration
Now we need to set the `xorg.conf` file with the settings for our public machine monitors. Do the following in the terminal window:

```
cd /etc/X11
vi xorg.conf
```



Using the arrow keys, scroll to where the monitor definition is located and press `i` to go into "insert mode" and edit the settings to look like the following [make adjustments to match the monitor of your public machines]:

```
    Section “Monitor”
        ModelName      “Planar M1710”
        DisplaySize    337  270
        HorizSync      24.0-80.0
        VertRefresh    49.0-75.0
        #option        “dpms”
```

If a *dpms* option exists, comment it out.
Add a new section with the following entries. Options that are enabled are in bold.

```
Section "ServerFlags"
# Uncomment this to disable the <Crtl><Alt><Fn> VT switch sequence
# (where n is 1 through 12).  This allows clients to receive these key
# events.
    Option     "DontVTSwitch"

# Uncomment this to disable the <Crtl><Alt><BS> server abort sequence
# This allows clients to receive this key event.
    Option     "DontZap"

# Uncomment this to disable the <Crtl><Alt><KP_+>/<KP_-> mode switching
# sequences.  This allows clients to receive these key events.
    Option     "DontZoom"

# Uncomment this to disable tuning with the xvidtune client. With
# it the client can still run and fetch card and monitor attributes,
# but it will not be allowed to change them. If it tries it will
# receive a protocol error.
#    Option     "DisableVidModeExtension"

# Uncomment this to enable the use of a non-local xvidtune client.
#    Option     "AllowNonLocalXvidtune"

# Uncomment this to disable dynamically modifying the input device
# (mouse and keyboard) settings.
#    Option     "DisableModInDev"

# Uncomment this to enable the use of a non-local client to
# change the keyboard or mouse settings (currently only xset).
#    Option     "AllowNonLocalModInDev"

# Set the basic blanking screen saver timeout.
    Option     "blank time"    "0"    # 0=no screensaver

# Set the DPMS timeouts.  These are set here because they are global
# rather than screen-specific.  These settings alone don't enable DPMS.
# It is enabled per-screen (or per-monitor), and even then only when
# the driver supports it.
#    Option      "standby time"  "20"
#    Option      "suspend time"  "30"
#    Option      "off time"      "60"

EndSection
```

Now press the "ESC" escape key to get out of insert mode and you should be able to type wq at the bottom of the vi editor window to "write and quit". If all goes well, you should be back at the terminal window with a shell prompt.

Yum Update

Now we can update our installation with the latest updates to the kernel and all our installed packages. This is done through the use of the yum application. We have a server that has a default yum.conf file available for download which has all the settings necessary to facilitate the update.

In the terminal window, do the following, where *validuser* is an account name which has been granted access to the server:

```
cd /etc
sftp validuser@wlibtris.westlib.asu.edu
```

This will connect to the WLIBTRIS server in secure ftp mode. Type in the correct password when prompted.
Once connected, enter the following:

```
cd ..
cd web
cd html
cd fedora
cd 2
get yum.conf
exit
```

You now have downloaded the current yum.conf file to be used by the yum updater.
In the terminal window, do the following:

```
yum update
```

This will start yum downloading all the available header files for what is on the yum repository. It will then check the list against what is loaded on the BUILD workstation. When it is done checking and identified any additional packages you need downloaded, it will present you with a list of packages that can be updated.

Answer Y to start the updates.

It will then start the update process. When this process is complete, logout of your session and have the system REBOOT by selecting this option under the *Menu* pull-down menu from the login (greeter) screen.

Additional Actions

After the system reboots, logon as `root` and open a terminal window. We will now turn of some services. Type the following at the shell prompt:

```
chkconfig kudzu off
chkconfig isdn off
chkconfig rhnsd off
chkconfig pcmcia off
chkconfig cpuspeed off
chkconfig mdmpd off
```

We also need to disable IPV6. IPV6 comes as a default in FC2 but has been known to slow down DNS queries. To disable this, edit the `/etc/modprobe.conf` file:

```
cd /etc
vi modprobe.conf
```

Use the arrow keys to move to the end of the last line and press "`i`" to go into insert mode. Insert a new line with the following statement.

```
alias net-pf-10 off
```

This will turn off IPV6. Press the "`ESC`" escape key and type "`wq`" to write and quit editing the file.

New Directories

Now we will make some new directories to be used later. At the shell prompt, type the following:

```
cd /
mkdir e3image
mkdir e3custom
mkdir packages
```

We now should have three new directories in the root `/` directory. `e3image` is where the actual disk image files will be stored. `e3custom` is where any custom files and settings will be stored for building the image. `packages` is where any rpms or tarballed files that are installed will be placed.

It is time to once again, logout of this session and REBOOT the machine from the login screen Action item selection.

Final Cleanup

When the system has restarted and presents you with a login prompt, login as `root` and open up a terminal window and type the following:

```
cd /var/log
cat messages
```

You can use the page up and page down keys or scrollbar to view the `messages` file. Look for any problems encountered and address them.

```
cat dmesg
```

Examine the `dmesg` content to see if any issues were encountered during boot and address them (or ignore them).
If all is well, then we can remove them so that they will be fresh when the system boots the next time. Type the following at the shell prompt:

```
rm /var/log/dmesg
rm /var/log/messages
```

We also need to rid ourselves of the Redhat Network Update stuff. Type the following at the shell prompt:

```
rm /usr/bin/rhn_register
rm –rdf /etc/sysconfig/rhn
```

Another service we do not need is the cron scheduling of logwatch. Do the following:

```
rpm –e logwatch
```

Now we need the locater database to reflect all of the changes we have made so far. To do this, type the following at the shell prompt:

```
updatedb
```

Since the kernel has been updated to the latest version, and since the DEVELOP station will need to compile some applications against this kernel version, we need to edit the yum.conf file to exclude all new kernel updates.

```
cd /etc
vi yum.conf
```

Under `[main]` add the entry:

```
exclude=kernel*
```

Do this also on the DEVELOP station.

Now all should be well and we are ready to duplicate the hard drive so we have a fresh backup copy. Logout and perform a shutdown of the system. Take the hard drive out and use our hard drive duplicator to make a copy to the "master" BUILD station drive.

End of PART 2

Configuration PART 1 – Initial configuration

Images and Icons

We need to use some custom images for backgrounds screens, etc. We will want to place these custom items into the `e3custom` directory. To start with, we need to make some new subdirectories.

```
cd e3custom
mkdir usr
cd usr
mkdir local
cd local
mkdir e3
cd e3
mkdir wallpaper
mkdir icons
```

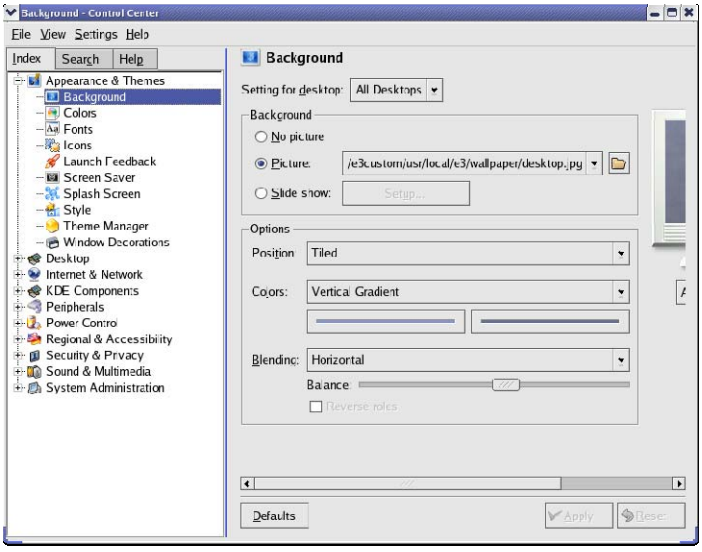
Now you can place your custom pictures in the `wallpaper` directory and custom icons in the `icons` directory.

Desktop Customizing Using KDE Control Center

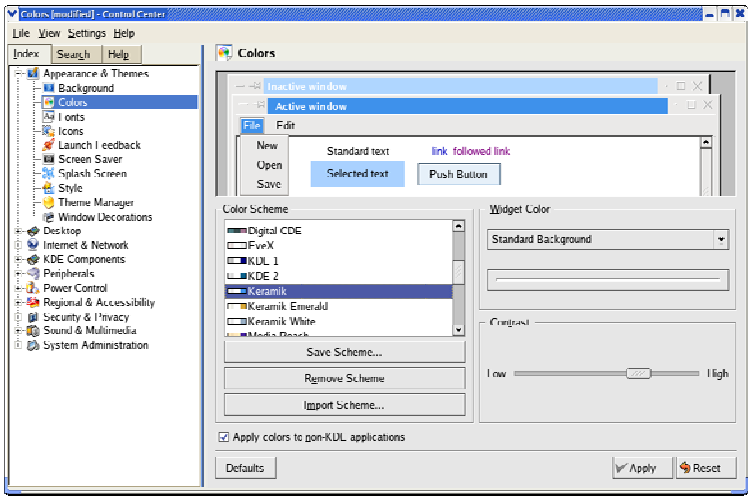
The KDE Control Center application allows us to customize the user interface and desktop behaviors. You will start by customizing the `root` user desktop environment where we will later export these settings into the global environment for all users. Below is a list of setting types and what should be set.

Appearance & Themes

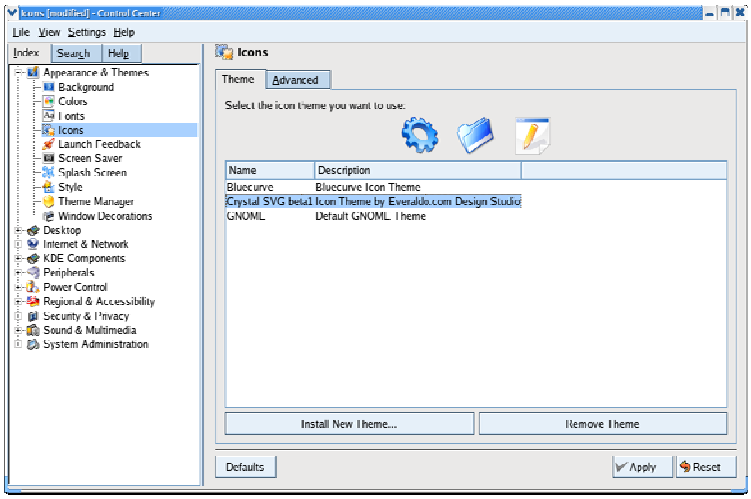
Background Set the background to a picture file you placed in the `e3custom/usr/local/e3/wallpaper` directory.



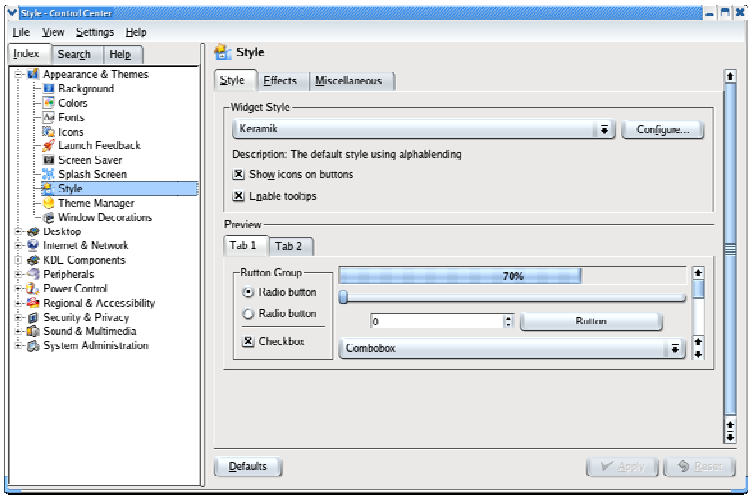
Colors Set the colors to Keramik.



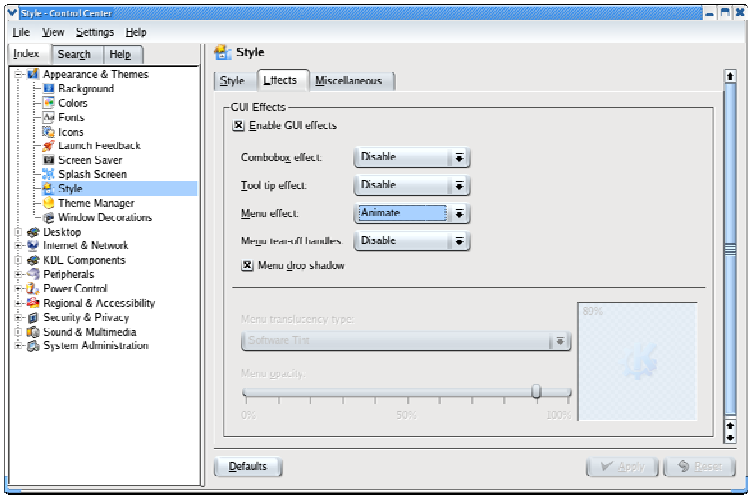
Icons Set the icons to Crystal SVG.



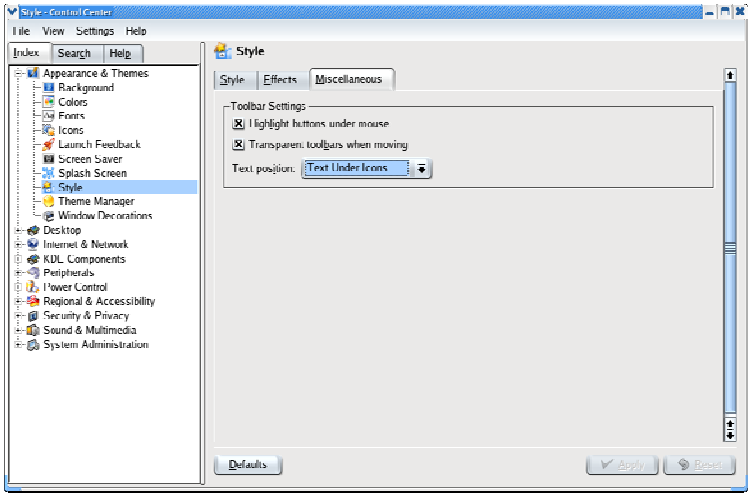
Style Set the Style widget to Keramik



Style Under the Effects tab, enable GUI Effects.

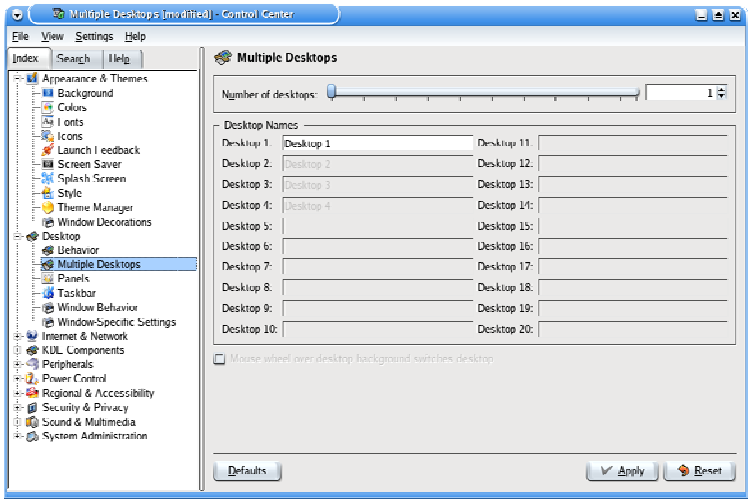


Style Under the Miscellaneous tab, set Text Position to Text Under Icons.

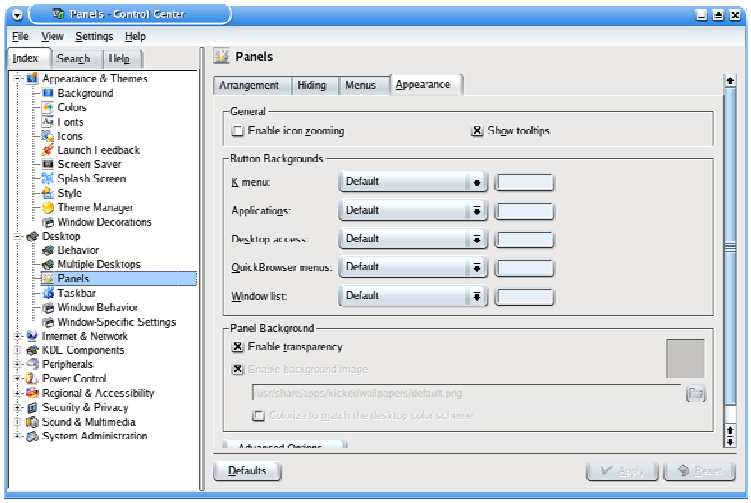


Desktop

Multiple Desktops Set to 1 virtual desktop



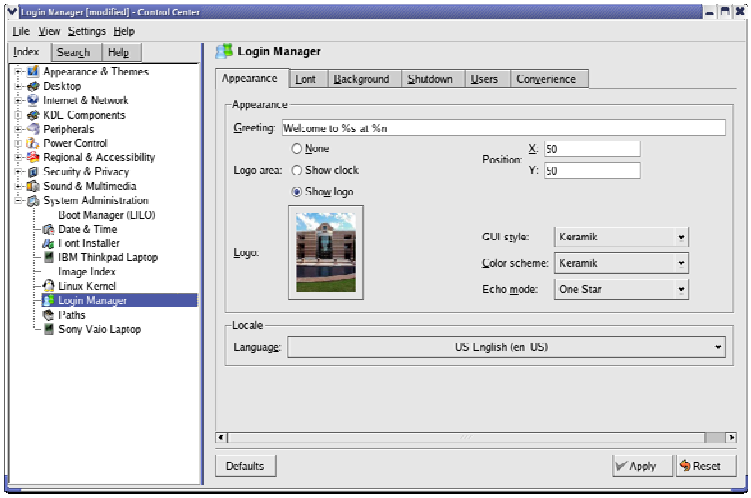
Panels Under the *Appearance* tab, select to Enable Icon Zooming and Enable Transparency.



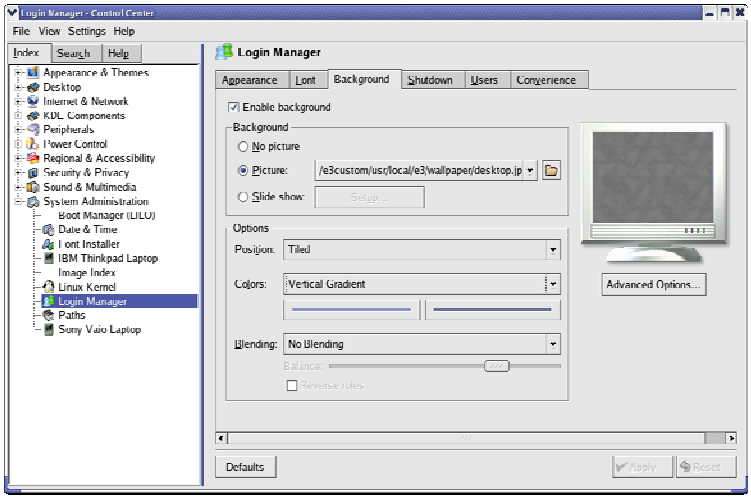
System Administration

Login Manager

Select the *Appearance* tab and change the logo to an image file in the `e3custom/usr/local/e3/wallpaper` directory.
Change the default language to US.
Change the GUI style to Keramik.
Change the Color Scheme to Keramik.



Select the *Background* tab to change the login background screen.



Also verify that the desktop theme is set to default.

More on Background Screens

When a user initiates a login, the background normally will change to a solid color until the desktop is loaded. Then our desktop background picture is loaded. Since we want to ensure a smooth login transition process, we want the login background picture and desktop background picture to remain the same as well as the transition screen. To do this, we need to edit the `Xsession` file.

```
cd /etc/X11/xdm
vi Xsession
```

Move down to where there is an entry for `xsetroot`. It should be listing `–solid` and some hex number for the color. Remove this entry and add one for `display`. Change this line to look like the following:

```
display –window root "/usr/local/e3/wallpaper/logindesktop.jpg"
```

This is now pointing to the same background picture that is used for the KDE login screen.

Configuration PART 2 – Custom Software

OpenAFS
This is an optional component. To enable AFS network storage mounting we need to install OpenAFS. To do this, we will use yum to install our custom OpenAFS packages:

```
yum install openafs
yum install openafs-client
```

Submount
Submount is a Sourceforge project to replace automounter for realtime access to removable media devices. This system allows one to insert a device, read or write to it, and then immediately remove it without having to perform a mount/umount command. To install submount, type the following:

```
yum install submount
```

This should have installed the kernel module into `/lib/modules/kernelversion/fs/subfs` and started it.
Type `lsmod` to see if it is running. It should be listed as `subfs`. If not, type:

```
modprobe subfs
```

Now we need to set up the mountpoints for all of our removable media. Since we use a 10-in-1 media reader, as well as an IDE zip and floppy, we need to make the new mountpoints. Open a terminal shell and type:

```
cd /mnt
rm –rdf *
mkdir floppy cdrom memory-stick usb-key compact-flash
secure-digital secure-media zip
chmod 666 *
```

This will allow the Build Station to have access to these devices. Now we need to set up `fstab`. This is a bit tricky. We need an `fstab` for the Build Station but also an `fstab` for the Public Machine (PAC) image. To facilitate this, do the following at a command prompt:

```
mkdir /e3custom/etc
```

Now we need to edit our `fstab` files to enable the *submount* system:

```
cd /etc
vi fstab
```

Make the `fstab` file look like the following. This is specific to your hardware so you might want to first `cp` (copy) your current `fstab` file to `fstab.old` as a reference.

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/boot             /boot                   ext3    defaults        1 2
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /dev/shm                tmpfs   defaults        0 0
none                    /proc                   proc    defaults        0 0
none                    /sys                    sysfs   defaults        0 0
/dev/hdb3               swap                    swap    defaults        0 0
/dev/cdrom              /mnt/cdrom                  subfs fs=cdfss,ro,umask=0 0 0
/dev/fd0                /mnt/floppy             subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdb1               /mnt/memory-stick       subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sde1               /mnt/usb-key            subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sda1               /mnt/compact-flash      subfs fs=vfat,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdd1               /mnt/secure-digital     subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdc1               /mnt/secure-media       subfs fs=vfat,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/hdb4               /mnt/zip                subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,umask=0 0
```

Remember to press the `escape` [esc] key and type `:` then `wq` to write the new `fstab` file in `vi`.

Now go into the `/e3custom/etc` directory and make a new `fstab` file to look like this:

```
cd /e3custom/etc
vi fstab
```

```
LABEL=/                 /                       ext3    defaults        0 0
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /dev/shm                tmpfs   defaults        0 0
none                    /proc                   proc    defaults        0 0
none                    /sys                    sysfs   defaults        0 0
/dev/fd0                /mnt/floppy             subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdb1               /mnt/memory-stick       subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sde1               /mnt/usb-key            subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sda1               /mnt/compact-flash      subfs fs=vfat,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdd1               /mnt/secure-digital     subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/sdc1               /mnt/secure-media       subfs fs=vfat,rw,nodev,nosuid,procuid,noexec,sync,umask=0 0 0
/dev/hdb4               /mnt/zip                subfs fs=floppyfss,rw,nodev,nosuid,procuid,noexec,umask=0 0 0
none                    /home                   tmpfs     size=60M,nodev,noexec,nosuid 0 0
none                    /tmp                    tmpfs     size=60M,noexec,nosuid,nodev 0 0
```

We set the `umask` directive to mode 666 to allow only read/write to these devices. We do not want users to be able to execute from them.

Applications

Printing

Now we need to set up the printing environment for the public machine. The following is instructions ONLY if you use the Pharos UniPRINT system for your print server. If you do not, you probably should stick with using CUPS as your print engine and skip this section.

Get `libcom_err.so.3.0` from an existing Redhat 9 installation. We need this file since Fedora does not ship with it and it will be required by `lpr`. Copy this file to the `/usr/lib` directory.

Now we need to make a symbolic link:

```
cd /usr/lib
ln –s libcom_err.so.3.0 libcom_err.so.3
```

Next we need to install the Kerberos workstation software, remove CUPS (KDE default print system) and install LPRng so we can print to the Pharos UniPRINT system.

```
yum install krb5-workstation
yum remove cups
yum install LPRng
```

The RPM spec file for LPRng has a switch in it to `enable-kerberos` or `disable-kerberos`. Our RPM build has Kerberos disabled.

Now we can remove any traces of the CUPS system we just removed by doing the following:

```
rm –rdf /etc/cups
```

Just in case things get screwy, we should create a `/etc/printcap` file with a default entry. By rights, we should never have to reference this since we are using the *`pharosdialog`* application. But for posterity, here is what you can enter using `vi /etc/printcap`:

```
atrium:
       :lpr_bounce
       :lp=atrium1@asuw4.west.asu.edu
       sd=/var/spool/lpd/atrium
```

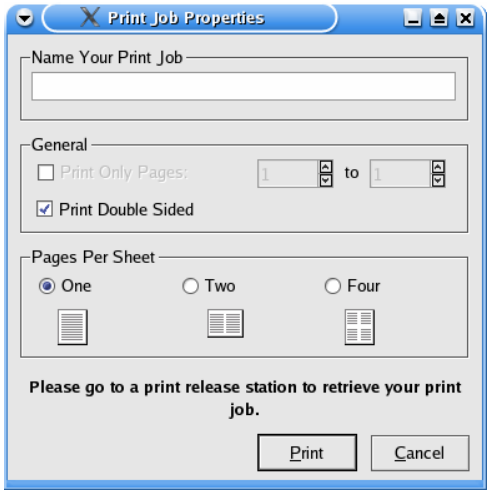
Pharos Dialog Application

Now we are ready to install the *`pharosdialog`* application. Make a directory called `pharosdialog` in `/etc`

```
mkdir /etc/pharosdialog
```

Now you can yum install the pharosdialog RPM package:

```
yum install pharosdialog
```



The pharosdialog installation places the application into the `/usr/local/bin` directory and creates an initial configuration file named `pharosdialogrc` in the `/etc/pharosdialog` directory. The contents of the `pharosdilaogrc` file looks something like this, where each entry has the server queue name @ server.x.x.x for the UniPRINT server.:

```
[General]
PharosServerPublic=atrium2@asuw4.west.asu.edu
PharosServerValid=atrium1@asuw4.west.asu.edu
```

Now we need to do some cleanup. Do the following:

```
cd /usr/bin
cp lpr lpr.LPRng
rm lpr
rm lp
```

Now we need to create some symbolic links so `lpr` will work with the *`pharosdialog`* application:

```
ln –s /usr/local/bin/pharosdialog lpr
ln –s /usr/local/bin/pharosdialog lp
ln –s lpr lp.LPRng
cd /usr/local/bin
ln –s pharosdialog lpr
```

Now to make the duplex printing option work, we must install the dpr application. This application is available pre-compiled in our package distribution or can be obtained from http://www.muquit.com/muquit/software/dpr/dpr.html and compiled on the DEV STATION. Because it uses its own version of a2ps, we will also want to remove what Fedora installed default:

```
rpm –e a2ps
```

Now we can install `dpr`. Put the `dpr-1.1.tar.gz` file which includes our new compiled `dpr` binary into `/tmp`. Then we can extract it and put it where it needs to go:

```
cd /tmp
```

```
tar –zxvf dpr-1.1.tar.gz
cd dpr
cd dpr1.1
ls –l
./dpr
cp dpr /usr/local/bin
mkdir –p /usr/local/lib/dpr
cp libs/* /usr/local/lib/dpr/
```

Later in this section we will install CodeWeaver's CrossOver Office. To make duplex printing work with it, there will be an edit made to the generic printer ppd file. See the Crossover installation for details.

Custom Disk Formatter

Our library uses its own custom application for users to format their Iomega Zip and floppy disks. It is a simple GUI interface called *floppyformatter2*.
The disk formatting application uses ***`mtools`*** and ***`mkdosfs`*** applications to build the filesystem and format the disks. You will need to edit the `/etc/mtools.conf` file to only enable the following. All other entries should be commented out or removed from the `mtools.conf` file. The device locations for drive a: and drive z: may be different then shown below so you should look at the `/etc/fstab` file to see what they really are and reflect that in the below edit.

```
cd /etc
vi mtools.conf
```

Make the following edits/entries:

```
drive a: file="/dev/fd0" exclusive mformat_only
drive z: file="/dev/hdb4"
```

Now we need to prep things for installing the actual application. Do the following:

```
cd /etc
mkdir diskformatter2
```

This is where the `diskformatter2rc` file will be located. It contains the actual device paths for both zip and floppy drives. You can see what your device paths are by viewing the `/etc/fstab` file and make the appropriate changes. Do this **AFTER** you perform the following step:

```
yum install diskformatter
```

This placed the `diskformatter2` application in `/usr/local/bin` and added the `diskformatter2rc` file to `/etc/diskformatter2`. You can now edit the `/etc/diskformatter2/diskformatter2rc` file. If you do not have a particular drive installed, just leave the entry blank after the = sign. The following is an example of the `diskformatter2rc` file:

```
[General]
Floppy=/dev/fd0
Zip=/dev/hdb4
```

Custom Logout Screensaver

This is an optional component. If you want to have a screensaver kick in after x number of minutes and warn the user to press a key or move the mouse while having a visible timer count down to zero, you can install the `logoutsaver` screensaver to have the workstation automatically logout and present the login prompt. To do this, do the following:

```
cd /usr/X11R6/lib/xscreensaver
rm –rdf *
cd /usr/share/control-center/screensavers
rm –rdf *.xml
```

We have now cleaned out all screensavers that were installed during our initial installation. Now we can install the custom logout screensaver:

```
yum install logoutsaver
```

After the installation is complete, the `logoutsaver` application was installed into `/usr/bin` and the `logoutsaverrc` and `logoutsaver.desktop` files were installed. Now, do the following:

```
cd /usr/share/applnk/System/ScreenSavers
chmod 755 *
```

You can change the time used for the countdown using the admin interface of the `logoutsaver` application. You can do this by entering the following:

```
logoutsaver –setup
```

The other entry shows the `logoutsaver` will execute a KDE `dcop` command to the `ksmserver` telling it to logout immediately and close all applications.

Inuseclient
This is an optional component. The *inuseclient* application is used to connect to the *inuseserver* telling it that someone has either logged on to a public machine or logged off. This information is stored in a MySQL database. To install the client:

```
yum install inuseclient
```

The *inuseclient* application is placed in the `/usr/local/bin` directory.

Java
Now we need to install the Java 2 Runtime Environment. Download this from Sun.com or use the yum installation.

```
yum install jre
```

This will have created the `/usr/java/jre1.5.0` directory and subdirectory files. NOTE: The file `rt.jar` needs to be read/write/executable in order to run Java applications (III Millennium).

Firefox

We will use the Mozzilla.org Firefox Web Browser on our public machines. We now need to install it. Get firefox-1.0.installer.tar.gz (or later) from www.mozilla.org and save it to the `/root` directory. Now you need to decompress it:

```
gzip –d firefox-1.0.installer.tar.gz
tar –xf firefox-1.0.installer.tar
```

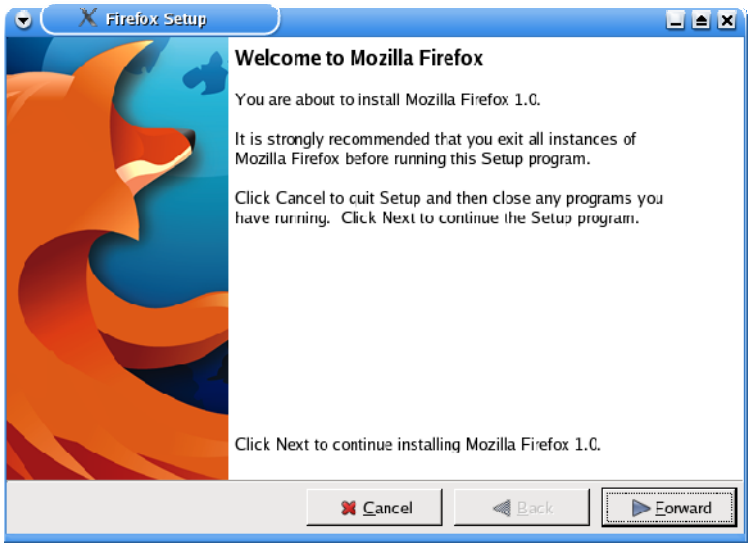
This makes the subdirectory `firefox-installer`. Go into that directory.

```
cd firefox-installer
```

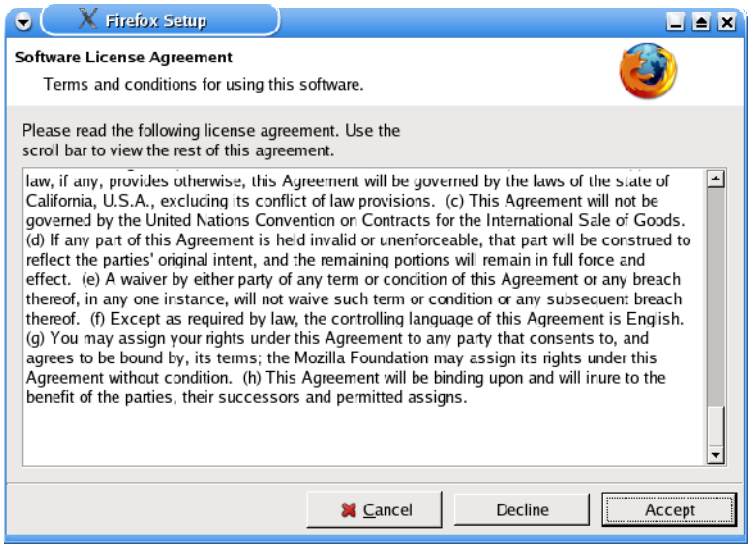
And execute the installer application:

```
./firefox-installer
```

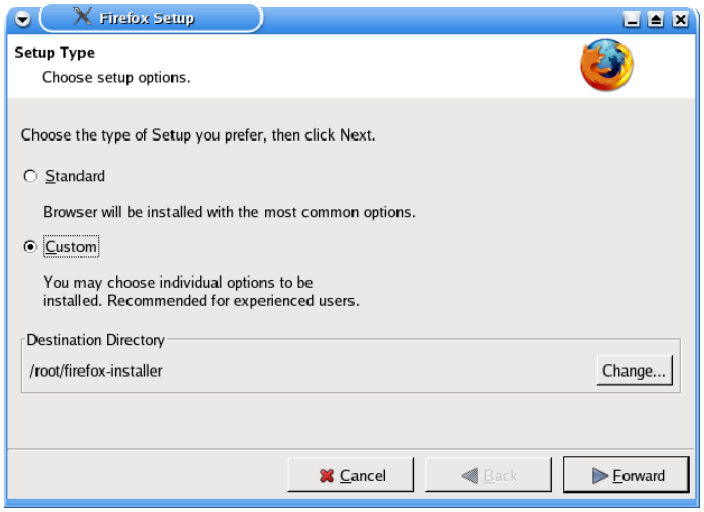
You should now see the setup screen. Click on the **Forward** button.



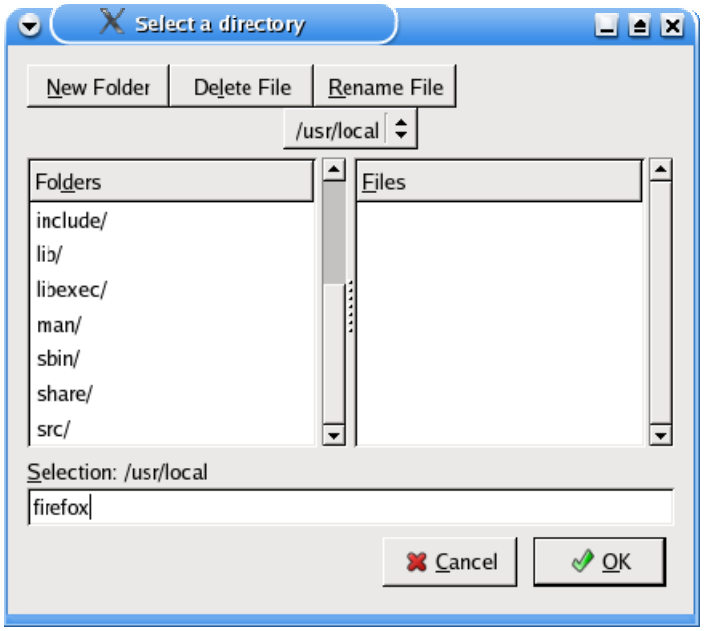
Select **Accept** for the Software License Agreement.



Select the **Custom** radio button.

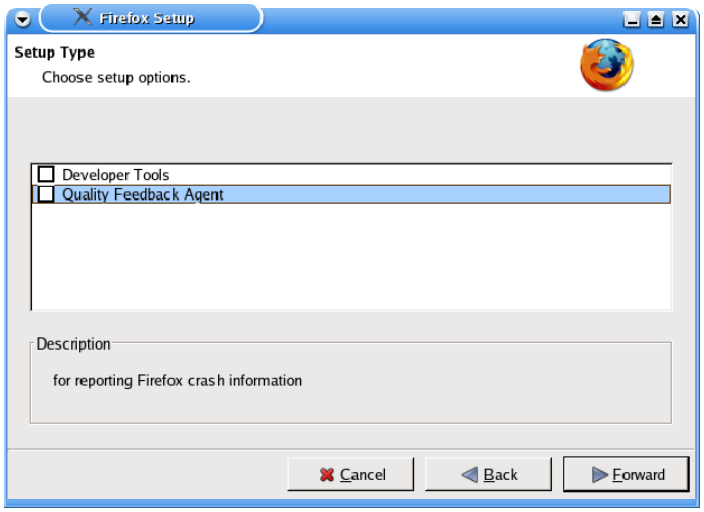


Select the **Change** button for the Destination Directory and navigate to `/usr/local` and change the Selection to `firefox`.

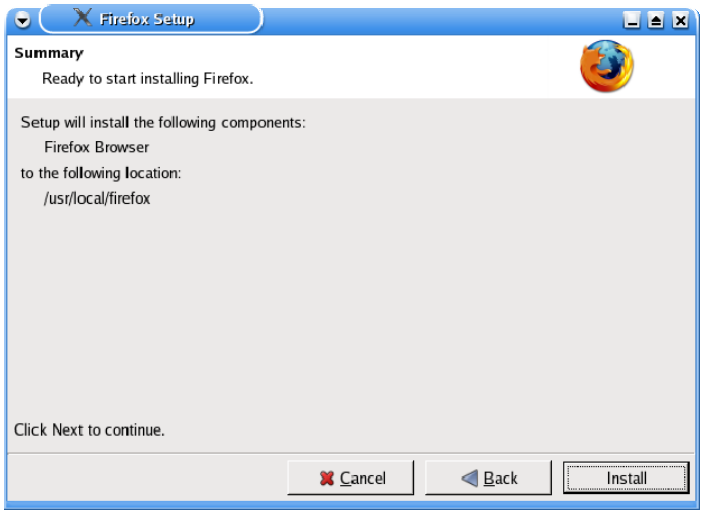


Answer **YES** to creating the new directory.

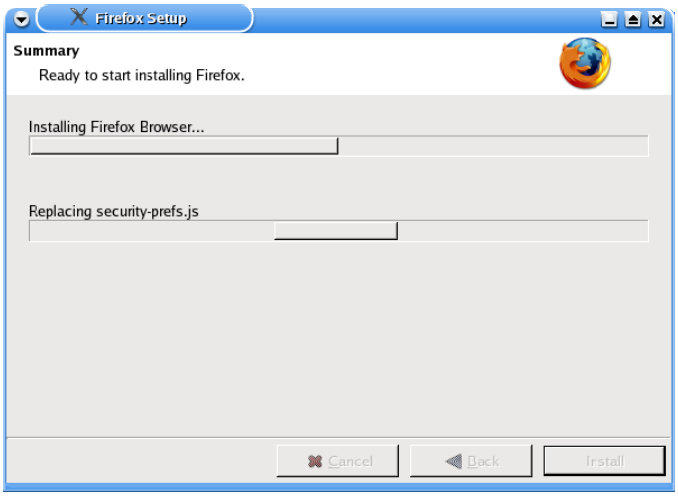
Uncheck the Quality Feedback Agent box.



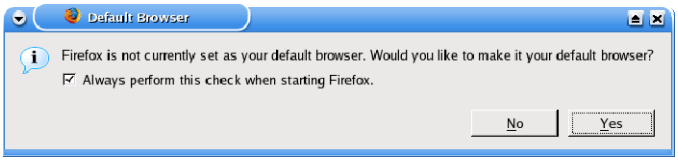
Select **Install** to start the installation.



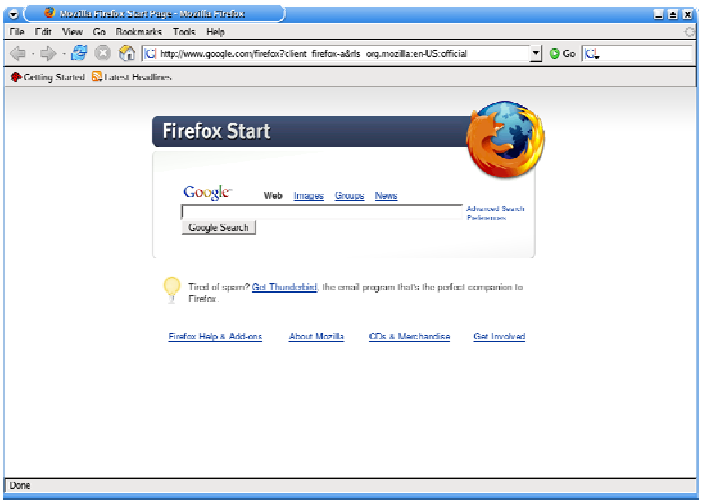
Installing…



Answer **YES** to make Firefox the default Web Browser.



Firefox should now be installed and working.



Firefox Helper/Viewer Applications
There are a few applications and plugins that we need to install for viewing our library resources using the Firefox browser. The following describes the installation procedures.

DjVu
To install the DjVu plugin do the following at a command prompt:

```
yum install djvulibre
```

Now you need to associate the viewer plugin with Firefox:

```
cd /usr/local/firefox/plugins
```

Now you need to create a symbolic link to the DjVu plugin resource:

```
ln –s /usr/lib/mozilla/plugins/nsdejavu.so nsdejavu.so
```

To verify that the plugin works with Firefox, do the following:

```
cd /usr/local/firefox
./firefox about:plugins
```

You should see an entry for *DjVuLibre*.

Java Runtime Environment (JRE)
To enable Java to work with Firefox, do the following at a command prompt:

```
cd /usr/local/firefox/plugins
```

Here is where we create the symbolic link to the Java plugin resources:

```
ln –s /usr/java/jre1.5.0/plugin/i386/ns7/libjavaplugin_oji.so libjavaplugin_oji.so
```

To verify that the plugin works with Firefox, do the following:

```
cd /usr/local/firefox
./firefox about:plugins
```

You should see an entry for *Java JRE*.

Macromedia FlashPlayer 7
Macromedia provides a native Linux Flash plugin for Firefox. Get `install_flash_player_7_linux.tar.gz` from www.macromedia.com and download it to `/root`. Now, at a command prompt, do the following:

```
cd /root
gzip –d install_flash_player_7_linux.tar.gz
tar –xf install_flash_player_7_linux.tar
cd install_flash_player_7_linux
./flashplayer-installer
```

1. Press [ENTER] when prompted
2. Enter `/usr/local/firefox` when prompted for an install path

```
cp /usr/local/firefox/plugins/flashplayer.xpt /usr/local/firefox/components
```

To verify that the plugin works with Firefox, do the following:

```
cd /usr/local/firefox
./firefox about:plugins
```

You should see an entry for *Flash/Shockwave/FuturePlayer*.

Real Player 10

For audio and video playback of Real formatted files, we need to install the RealPlayer. Go to www.real.com and download the RealPlayer 10 for Linux application and save it to `/root`. Then at a command prompt, do the following:

```
cd /root
chmod 777 RealPlayer10GOLD.bin
./RealPlayer10GOLD.bin
```

1. When prompted for the path to install, enter: `/usr/local/RealPlayer`
2. Answer [Y]es to create symbolic links
3. Answer `/usr` for symlinks

```
cd /usr/local/firefox/plugins
ln –s /usr/local/RealPlayer/mozilla/nphelix.so nphelix.so
cd /usr/local/firefox/components
ln –s /usr/local/RealPlayer/mozilla/nphelix.xpt nphelix.xpt
```

To verify that the plugin works with Firefox, do the following:

```
cd /usr/local/firefox
./firefox about:plugins
```

You should see an entry for *RealPlayer*.

Adobe Acrobat 5.09

Go to www.adobe.com and download the latest version of the Acrobat Reader for Linux (linux-509.tar.gz). Save the file to `/root` and enter the following at a command prompt:

```
cd /root
gzip –d linux-509.tar.gz
```

Adobe does not tar their application into a separate directory so we need to create a temporary one to decompress to:

```
mkdir acrobat
mv linux-509.tar ./acrobat
cd acrobat
tar –xf linux-509.tar
```

Now we can start the install script:

```
./INSTALL
```

1. Type [`accept`] to accept the license agreement
2. Install to: `/usr/local/Acrobat`

```
cd /usr/local/Acrobat
./acroread
```

Accept the license agreement . Now we need to create a symbolic link to the reader:

```
cd /usr/local/bin
ln –s /usr/local/Acrobat/bin/acroread acroread
```

We now need to create a script that will be used by Firefox and other applications to view pdf files. Create a file in /usr/local/bin called pdfview by using vi, and enter the following:

Script: pdfview

```
#!/bin/sh
#
# script to launch pdf viewer
#
# the export language is needed by Adobe Acrobat
export LANG="EN"

TEMPFILE=`date '+%H%M%S'`
FULLTEMPFILE=$HOME/$TEMPFILE-RENAME-ME.pdf

cp "$1" $FULLTEMPFILE

####Launch reader
cd $HOME
acroread $FULLTEMPFILE

####cleanup##########

if [ "${1:0:5}" == "/tmp/" ] ; then \
                        rm "$1"
                   rm $FULLTEMPFILE
else \
                   rm $FULLTEMPFILE
fi
```

After saving the pdfview file, do the following:

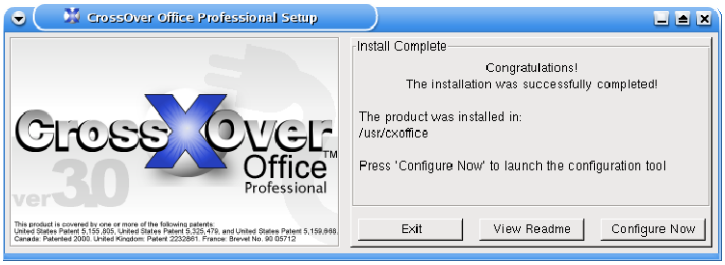
```
chmod 755 pdfview
```

CrossOver

We will use the version of WINE from www.codeweavers.com for native Microsoft Windows helper applications using the product called CrossOver. Start installing by running the CrossOver installation script.

Set the default installation Path to:
`/usr/cxoffice`

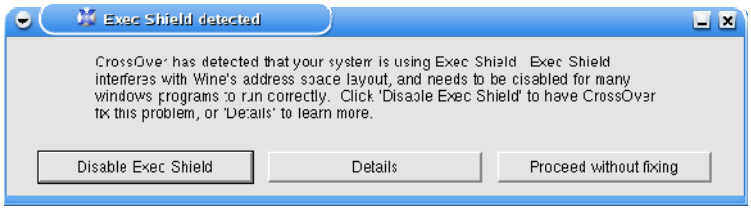
Select the Configure Now button



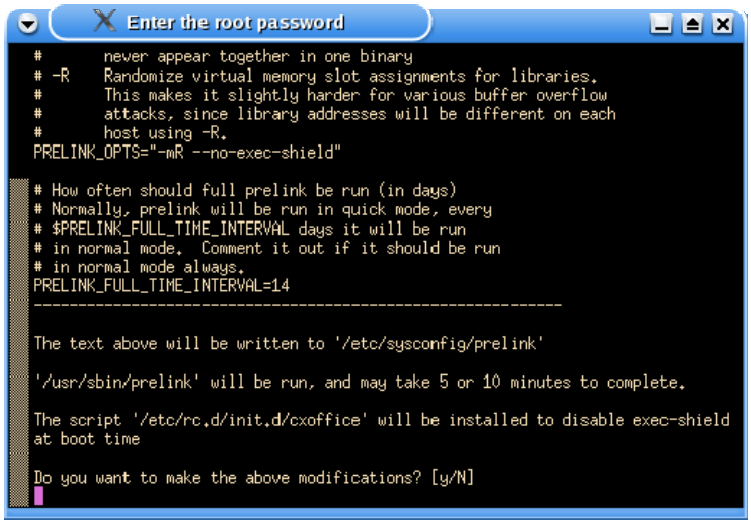
Select Setup Mode:
Managed Multi-user

Ignore errors

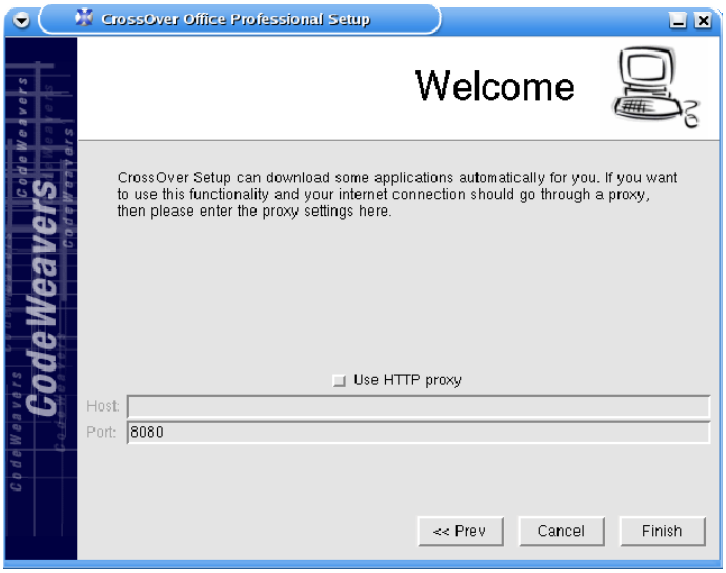
Disable Exec Shield



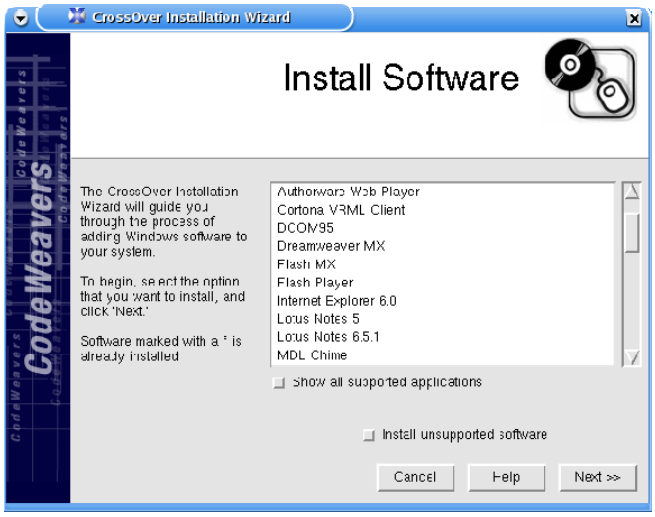
Answer [Y]es to allow modifications



Press Finish to complete the initial installation

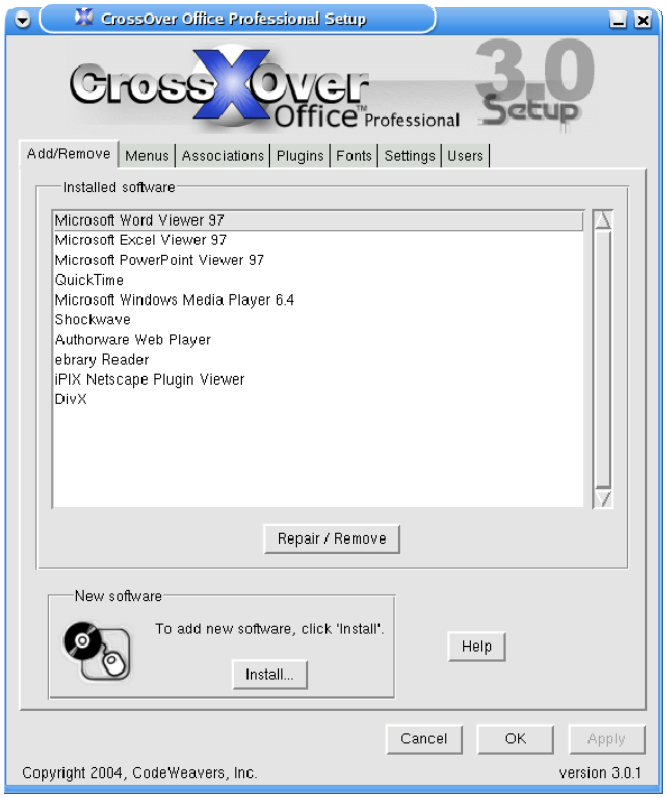


Select the *Plugins* tab and change the plugins path to include: `/usr/local/firefox/plugins`
You can do this by adding a `;` to the already included path lists



Under "New Software" select Install to start installing helper applications. We install the following:

- `Microsoft Word Viewer`
- `Microsoft Excel Viewer`
- `Microsoft PowerPoint Viewer`
- `Quicktime`
- `Microsoft Windows Media Player 6.4`
- `Shockwave`
- `Authorware Player`
- `ebrary Reader`
- `iPIX Viewer`
- `DivX codecs from www.divxmovies.com`



After you have installed all helpers, go into the Settings tab and make sure that the `/usr/local/firefox/firefox` applications is identified as the default browser. Then select *OK* to exit CrossOver.
CrossOver created a directory `/usr/cxoffice` and placed all necessary files in it.

Edit the `/usr/cxoffice/support/dotwine/config` file. Change the device entries to reflect the current hardware configuration for the public machines.

Also, edit the `/usr/cxoffice/support/dotwine/generic.ppd` file. We will want to set duplex printing for MS Windows applications as default for our ***PharosDialog*** application:

Change the line:

```
*DefaultDuplex:  None
```

To look like this:

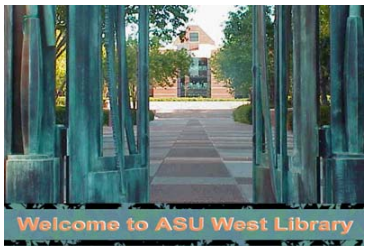
```
*DefaultDuplex:  DuplexNoTumble
```

Kicker

Kicker is used to display the toolbar at the bottom of the desktop window. Edit the `/usr/share/config/kickerrc` file. Remove `kminipagerapplet` and `lockoutpanelapplet`.

Login Splash Screen

We want to change the picture used by KDE on the splash screen during login to use our custom `SplashWelcome1.png` picture located in `/e3custom/usr/local/e3`. To do this:




```
cd /usr/share/apps/ksplash/Themes/Default
rm splash_top.png
ln –s /e3custom/usr/local/e3/SplashWelcome1.png splash_top.png
```

Login Greeter

The login greeter is the dialog box that is presented as the login screen. To change the message, fonts, etc. edit `/etc/X11/xdm/kdmrc`

KnewsTicker Applet

This is a small hyperlinking RSS news reader that runs as a ticker in the Kicker bar. Right click on the Kicker (task) bar and ADD APPLET KnewsTicker. Right click on the now applied newsticker and configure it to point to an RSS feed. Move it to a good spot on the Kicker bar.

More Firefox Tweaking

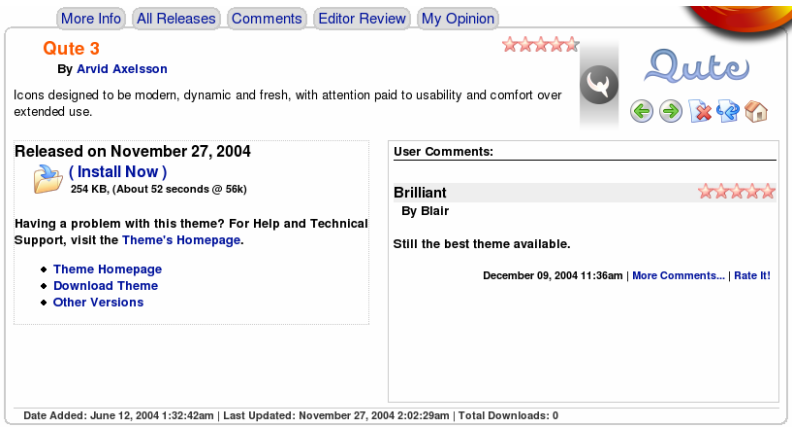
There are many things we can do to Firefox. The following are what is applied to our public image.

Theme

To change the theme, launch Firefox and select under Tools the Themes item.

Select the link to get more themes

Install the Qute 3 theme



Go back to the Tools menu, select Themes, and select the new Qute 3 theme and press the Use Theme button.

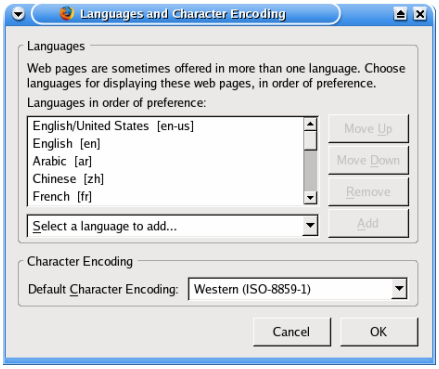
Now quit Firefox and launch it again. We should be seeing the new theme implemented.

Languages

To add additional languages, launch Firefox and select under Edit the Preferences item.

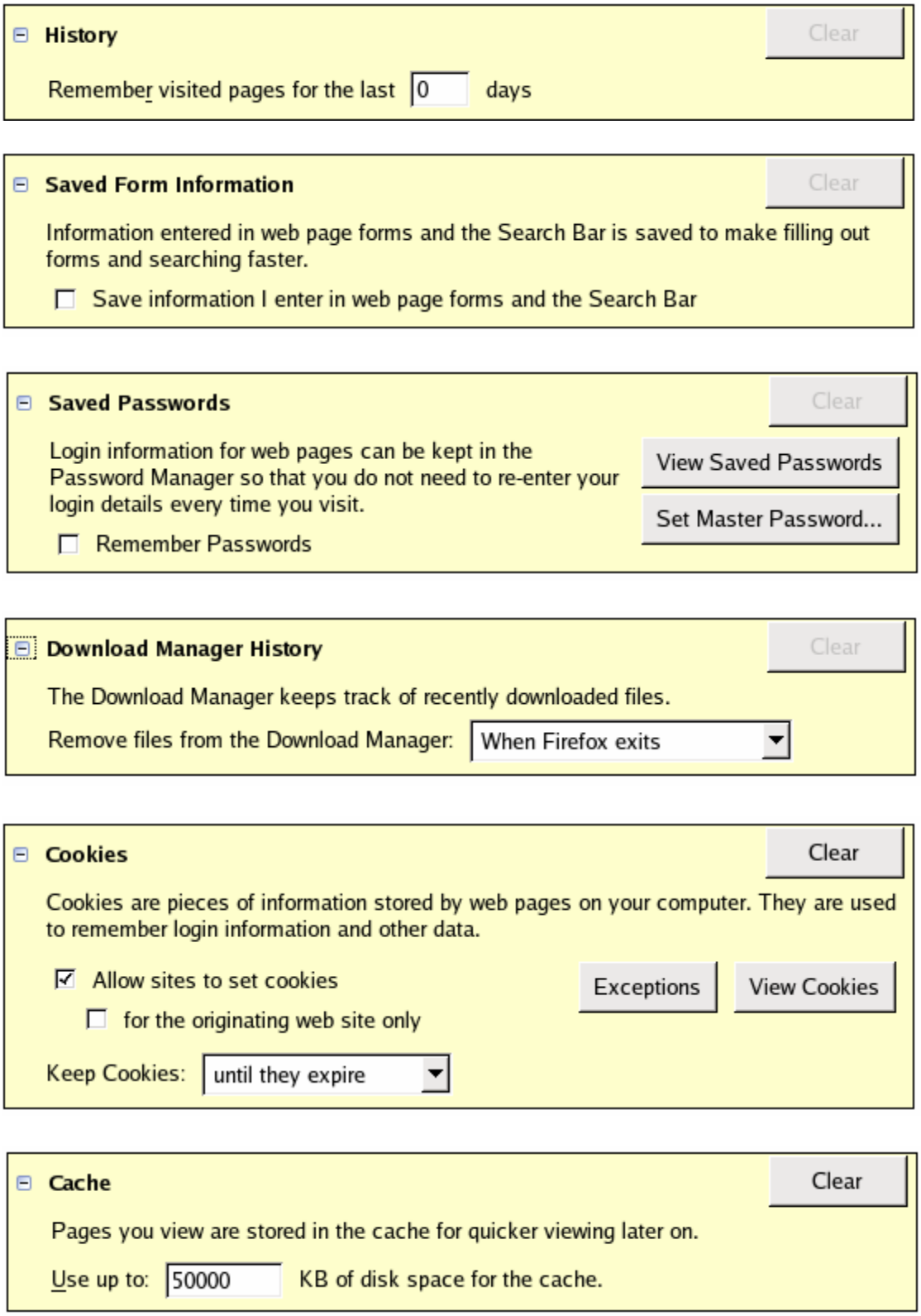
Select the General icon and then Languages

Add languages

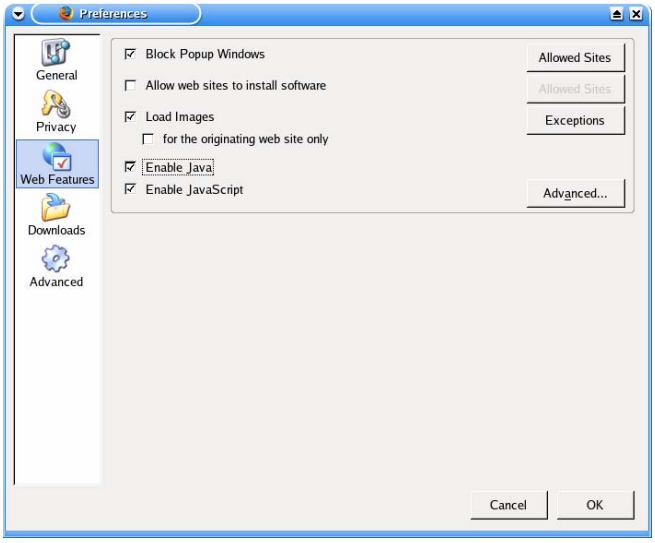


Privacy

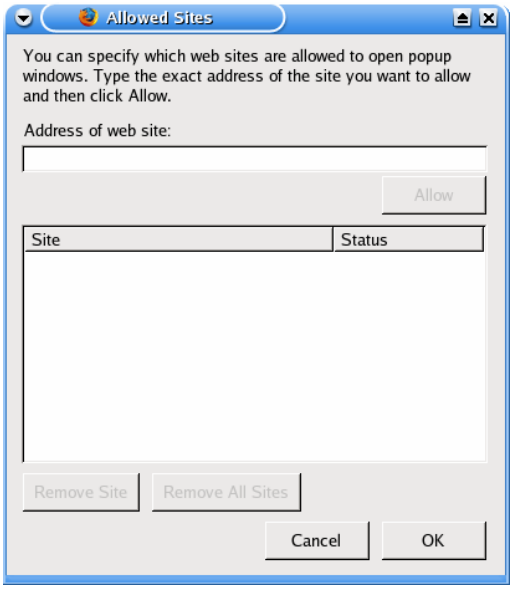
Here is where you can change the historical record and behavior of Firefox:



Web Features



Under the Web Features icon we need to uncheck the "Allow web sites to install software" and check "Enable Java" to enable Java applications.



We can also add sites to the white list of Allowed Sites that will not have their popups blocked. This list is stored as plain text in a file called `hostperm.1`

The following page lists the sites on our "white list"

Allowed Sites

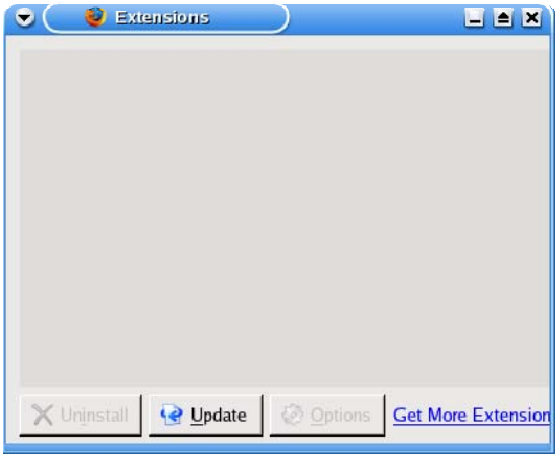
0-journals.iucr.org
web6.epnet.com
www.fofweb.com
brs.newsbank.com
publish.aps.org
www.chemnetbase.com
www.tpcincweb.com
web5.silverplatter.com
humanities.uchicago.edu
www.euromonitor.com
www.jstor.org
www.library.uiuc.edu
www.update-software.com
aio.anthropology.org.uk
arba.odyssi.com
www.asme.org
www.netlibrary.com
www.edrs.com
0-*
www.kcdlonline.com
www.netadvantage.standardandpoors.com
www.biologists.com
www.alexanderstreet2.com
web.lexis-nexis.com
ejournals.ebsco.com
www.booksinprint.com
app.harpweek.com
www.biomedcenter.com
www.grovemusic.com
premium.hoovers.com
rdsweb2.rdsinc.com
arjournals.annualreviews.org
www.hbcpnetbase.com
www.aip.org
proquest.umi.com
www.groveart.com
www.emeraldinsight.com
www.dgbiblio.unam.mx
www.edpsciences.org
sanborn.umi.com
library.cqpress.com
journals.cambridge.org
portal.acm.org
lion.chadwyck.com
juno.emeraldinsight.com
www.gpoaccess.gov
onlineedition.culturegrams.com
ca1.csa.com
poolesplus.odyssi.com
www.stat-usa.gov
infotrac.galegroup.com
www.ciaonet.org
gateway.ut.ovid.com
www.asee.org
accuweather.ap.org
www.engineeringvillage2.org
www.datagold.com
www.blackwell-synergy.com
www.accessible.com
www.cios.org
www.artstor.org

The popup allow list is located in the `./mozilla/firefox/default` directory called `hostperm.1` Here is an example of what it looks like:

```
# Permission File
# This is a generated file! Do not edit.

host            popup           1     0-journals.iucr.org
host            popup           1       web6.epnet.com
host            popup           1       www.fofweb.com
host            popup           1      brs.newsbank.com
host            popup           1       publish.aps.org
host            popup           1     www.chemnetbase.com
host            popup           1      www.tpcincweb.com
host            popup           1    web5.silverplatter.com
host            popup           1   humanities.uchicago.edu
host            popup           1     www.euromonitor.com
host            popup           1        www.jstor.org
host            popup           1    www.library.uiuc.edu
host            popup           1   www.update-software.com
host            popup           1   aio.anthropology.org.uk
host            popup           1       arba.odyssi.com
host            popup           1         www.asme.org
host            popup           1     www.netlibrary.com
host            popup           1        www.edrs.com
host            popup           1             0-*
host            popup           1     www.kcdlonline.com
host            popup           1       www.netadvantage.standardandpoors.com
host            popup           1     www.biologists.com
host            popup           1  www.alexanderstreet2.com
host            popup           1     web.lexis-nexis.com
host            popup           1     ejournals.ebsco.com
host            popup           1    www.booksinprint.com
host            popup           1      app.harpweek.com
host            popup           1    www.biomedcenter.com
host            popup           1     www.grovemusic.com
host            popup           1     premium.hoovers.com
host            popup           1     rdsweb2.rdsinc.com
host            popup           1       arjournals.annualreviews.org
host            popup           1     www.hbcpnetbase.com
host            popup           1         www.aip.org
host            popup           1      proquest.umi.com
host            popup           1      www.groveart.com
host            popup           1    www.emeraldinsight.com
host            popup           1     www.dgbiblio.unam.mx
host            popup           1     www.edpsciences.org
host            popup           1       sanborn.umi.com
host            popup           1     library.cqpress.com
host            popup           1    journals.cambridge.org
host            popup           1        portal.acm.org
host            popup           1      lion.chadwyck.com
host            popup           1   juno.emeraldinsight.com
host            popup           1      www.gpoaccess.gov
host            popup           1       onlineedition.culturegrams.com
host            popup           1         ca1.csa.com
host            popup           1    poolesplus.odyssi.com
host            popup           1       www.stat-usa.gov
host            popup           1    infotrac.galegroup.com
host            popup           1       www.ciaonet.org
host            popup           1     gateway.ut.ovid.com
host            popup           1         www.asee.org
host            popup           1     accuweather.ap.org
host            popup           1       www.engineeringvillage2.org
host            popup           1      www.datagold.com
host            popup           1  www.blackwell-synergy.com
host            popup           1     www.accessible.com
host            popup           1         www.cios.org
host            popup           1       www.artstor.org
```

Firefox Extensions

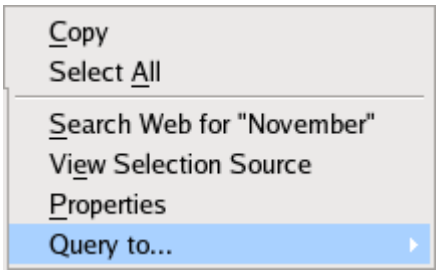
We will want to load some extensions to the Firefox browser to enhance the user's experience and productivity. The following are some Extensions we want to load for our library users.



In Firefox, under Tools, select Extensions. Click on "Get More Extensions" link in the Extensions dialog box.

ConQuery

Install ConQuery. This extension allows you to highlight text and right click to a context menu so you can select from a list of Web resources to query the text to. After ConQuery is installed, you will need to restart Firefox to activate the extension. To configure the ConQuery extension, go to `/usr/local/firefox/searchplugins`. You will see various files with a `.src` extension. You can add, delete, or modify existing `.src` plugins or create your own. You will want to keep these in this directory. If there is a tiny icon graphic you want ConQuery to present in the context menu for a particular `.src` plugin file, you can place a `.png` icon with the same filename as the `.src` file.



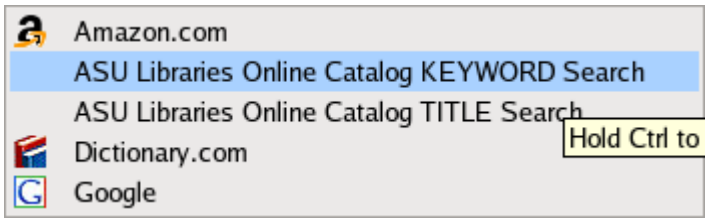


Image Toolbar

This extension provides additional context entries for viewing images.

PrintIt!

This extension provides an addition to the standard context menu for printing and print previews.

Translate

This extension allows you to highlight text in a Web page and send it to an online language translator to translate the selection from a foreign language to English.

User Agent Switcher
This allows you to select the message sent from Firefox to a Web server identifying what browser is being used. This is helpful for some sites that only allow Microsoft Internet Explorer browsers access.

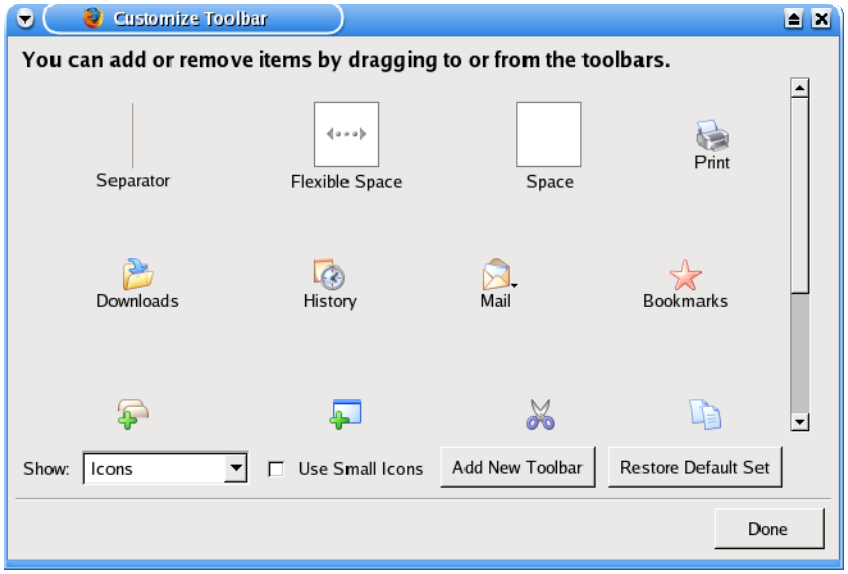
Tab X
This places an X next to each browser tab to allow you to easily close the tab.

MAF
Maf is the Mozilla Archive File extension that allows you to right-click and save a page in the .MHT archive format. It also makes Firefox able to read .MHT files. Found at http://maf.mozdev.org

PrefBar
This is a set of additions for Firefox that allows you to place various functions on the browser window. After installation, the file `prefbar.rdf` is created in the local `.mozilla/firefox/default` folder. You can use the GUI interface to PrefBar from the Firefox Tools menu to configure. It places these changes `in prefbar.rdf`. We use the "Block Popups", "Clear Cookies", and "Kill Flash" options. If you edit the prefbar.rdf file, you can change the LABEL text to say whatever you want for each option.

Customizing the Toolbar

Right click on the Firefox toolbar and select Customize. You then will be presented with a set of icons that you can drag over and place on the Firefox browser.



Associating File Types in Firefox

When you click on a URL that references a file, Firefox will try to launch the viewer application for that file. To associate CrossOver resources in Firefox, an easy way is to create a Web page with URLs to each type of file format you want to access. Then when you click on the URL, Firefox will present you with a dialog box asking you what you want to do.

Select the application in the `/usr/cxoffice/bin` directory. The application script files are actually hidden "dot" files.

You can see these by going to `/usr/cxoffice/bin` in a command shell and use a `ls –al .*` to get a listing of them. You can `cat .`*`filename`* to view each of them to see what application each launches.

Then in Firefox, when it asks for the application to use to view, go to the /`usr/cxoffice/bin` directory and type the name of the dot file.

Then in the Firefox dialog box, check the box that says to use this application from now on without prompting.

Do this to all filetypes you want to use browser helper applications for viewing (ie. Word docs, AVI, mp3, etc,) Since we loaded the Linux Adobe Acrobat and created the *pdfview* script `in /usr/local/bin`, you can use the *pdfview* script as the application to view pdf documents in Firefox.

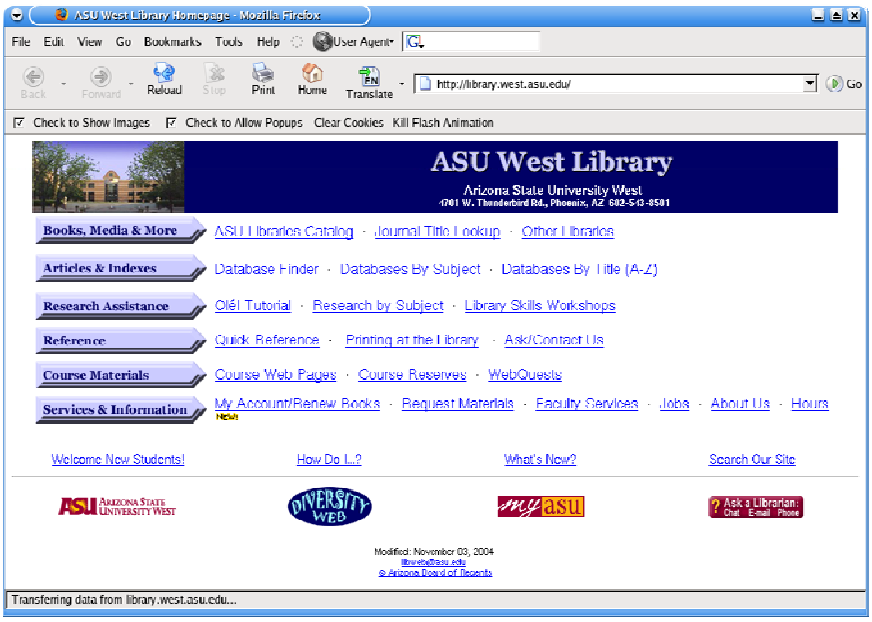
Firefox Preferences File

Firefox preferences are stored in the `prefs.js` file located in the `/root/.mozilla/firefox/default` directory. The preferences we want enabled can be edited and are listed in the sample `prefs.js` file below:

```
# Mozilla User Preferences

/* Do not edit this file.
 *
 * If you make changes to this file while the browser is running,
 * the changes will be overwritten when the browser exits.
 *
 * To make a manual change to preferences, you can visit the URL about:config
 * For more information, see http://www.mozilla.org/unix/customizing.html#prefs
 */

user_pref("app.update.autoUpdateEnabled", false);
user_pref("browser.download.lastDir", "./");
user_pref("browser.download.manager.closeWhenDone", true);
user_pref("browser.download.manager.behavior",2);
user_pref("browser.download.manager.retention", 0);
user_pref("browser.download.save_converter_index", 0);
user_pref("browser.download.useDownloadDir", false);
user_pref("browser.urlbar.clickSelectsAll", true);
user_pref("browser.formfill.enable", false);
user_pref("browser.history_expire_days", 0);
user_pref("browser.preferences.lastpanel", 1);
user_pref("browser.search.selectedEngine", "Google");
user_pref("browser.shell.checkDefaultBrowser", false);
```

```
user_pref("browser.startup.homepage", "http://library.west.asu.edu");
user_pref("browser.startup.homepage_override.mstone", "rv:1.7.5");
user_pref("conquery.Main.HTTPIcon", false);
user_pref("conquery.Main.HideFromTools", true);
user_pref("conquery.Main.OpenResults", "tab_fg");
user_pref("conquery.Main.PlugsDir", "/usr/local/firefox/searchplugins");
user_pref("conquery.Main.UseHotMenu", false);
user_pref("dom.disable_open_during_load", false);
user_pref("dom.disable_window_open_feature.status", false);
user_pref("dom.disable_window_status_change", false);
user_pref("extensions.disabledObsolete", true);
user_pref("extensions.lastAppVersion", "1.0");
user_pref("extensions.update.autoUpdateEnabled", false);
user_pref("general.skins.selectedSkin", "Qute");
user_pref("imagetoolbar.altButton", 1);
user_pref("imagetoolbar.buttonArray", "0,0,0,0,0");
user_pref("imagetoolbar.ctrlOverrideSize", false);
user_pref("imagetoolbar.display.copy", false);
user_pref("imagetoolbar.display.folder", false);
user_pref("imagetoolbar.display.mail", false);
user_pref("imagetoolbar.display.print", false);
user_pref("imagetoolbar.display.relativeToMouse", true);
user_pref("imagetoolbar.display.save", true);
user_pref("imagetoolbar.display.smallIcons", false);
user_pref("imagetoolbar.extensionsToIgnore", "");
user_pref("imagetoolbar.imageFolder", "/home/pacsym");
user_pref("imagetoolbar.minHeight", 0);
user_pref("imagetoolbar.minWidth", 0);
user_pref("imagetoolbar.offsetX", 30);
user_pref("imagetoolbar.offsetY", 0);
user_pref("imagetoolbar.overwrite", false);
user_pref("imagetoolbar.popupDelay", 2500);
user_pref("imagetoolbar.promptFilenameIncrement", true);
user_pref("intl.accept_languages", "en-us, en, ar, zh, fr, de, he, it, ja, ko, la, ru,
es, th, vi");
user_pref("intl.charsetmenu.browser.cache", "windows-1250, ISO-8859-15, ISO-8859-1, UTF-
8");
user_pref("maf.general.useragent", "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.5)
Gecko/20041107 Firefox/1.0");
user_pref("maf.postsetup.0.4.3.complete", true);
user_pref("maf.savearchive.filterindex", 3);
user_pref("network.cookie.prefsMigrated", true);
user_pref("prefbar.version", 20041003);
user_pref("print.printer_PostScript/default.print_command", "lpr ${MOZ_PRINTER_NAME:+'-
P'}${MOZ_PRINTER_NAME}");
user_pref("print.printer_PostScript/default.print_in_color", true);
user_pref("print.printer_PostScript/default.print_paper_height", "279.40");
user_pref("print.printer_PostScript/default.print_paper_name", "Letter");
user_pref("print.printer_PostScript/default.print_paper_size_type", 1);
user_pref("print.printer_PostScript/default.print_paper_size_unit", 0);
user_pref("print.printer_PostScript/default.print_paper_width", "215.90");
user_pref("print.tmp.printerfeatures.PostScript/default.can_change_num_copies", true);
user_pref("print.tmp.printerfeatures.PostScript/default.can_change_orientation", true);
user_pref("print.tmp.printerfeatures.PostScript/default.can_change_paper_size", true);
user_pref("print.tmp.printerfeatures.PostScript/default.can_change_plex", false);
user_pref("print.tmp.printerfeatures.PostScript/default.can_change_spoolercommand",
true);
user_pref("print.tmp.printerfeatures.PostScript/default.has_special_printerfeatures",
true);
user_pref("print.tmp.printerfeatures.PostScript/default.orientation.0.name", "portrait");
user_pref("print.tmp.printerfeatures.PostScript/default.orientation.1.name",
"landscape");
user_pref("print.tmp.printerfeatures.PostScript/default.orientation.count", 2);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.0.height_mm", 210);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.0.is_inch", false);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.0.name", "A5");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.0.width_mm", 148);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.1.height_mm", 297);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.1.is_inch", false);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.1.name", "A4");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.1.width_mm", 210);
```

```
user_pref("print.tmp.printerfeatures.PostScript/default.paper.2.height_mm", 420);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.2.is_inch", false);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.2.name", "A3");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.2.width_mm", 297);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.3.height_mm", 279);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.3.is_inch", true);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.3.name", "Letter");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.3.width_mm", 215);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.4.height_mm", 355);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.4.is_inch", true);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.4.name", "Legal");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.4.width_mm", 215);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.5.height_mm", 254);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.5.is_inch", true);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.5.name", "Executive");
user_pref("print.tmp.printerfeatures.PostScript/default.paper.5.width_mm", 190);
user_pref("print.tmp.printerfeatures.PostScript/default.paper.count", 6);
user_pref("print.tmp.printerfeatures.PostScript/default.plex.0.name", "default");
user_pref("print.tmp.printerfeatures.PostScript/default.plex.count", 1);
user_pref("print.tmp.printerfeatures.PostScript/default.supports_orientation_change",
true);
user_pref("print.tmp.printerfeatures.PostScript/default.supports_paper_size_change",
true);
user_pref("print.tmp.printerfeatures.PostScript/default.supports_plex_change", false);
user_pref("print.tmp.printerfeatures.PostScript/default.supports_spoolercommand_change",
true);
user_pref("security.OCSP.URL", "");
user_pref("security.OCSP.signingCA", "Builtin Object Token:IPS CLASE1 root");
user_pref("security.warn_entering_secure", false);
user_pref("security.warn_leaving_secure", false);
user_pref("security.warn_submit_insecure", false);
user_pref("signon.rememberSignons", false);
user_pref("translate.displayContextMenu", true);
user_pref("translate.displayToolMenu", false);
user_pref("translate.userlanguage", 0);
user_pref("useragentswitcher.1.appname", "Netscape");
user_pref("useragentswitcher.1.appversion", "5.0 (X11; en-US)");
user_pref("useragentswitcher.1.description", "Firefox");
user_pref("useragentswitcher.1.platform", "Linux i686");
user_pref("useragentswitcher.1.useragent", "Mozilla/5.0 (X11; U; Linux i686; en-US;
rv:1.7.5) Gecko/20041107 Firefox/1.0");
user_pref("useragentswitcher.1.vendor", "Firefox");
user_pref("useragentswitcher.1.vendorsub", "1.0");
user_pref("useragentswitcher.2.appname", "Netscape");
user_pref("useragentswitcher.2.appversion", "7 [en] (Windows NT 5.1; U)");
user_pref("useragentswitcher.2.description", "Netscape 7 (Windows XP)");
user_pref("useragentswitcher.2.platform", "Win32");
user_pref("useragentswitcher.2.useragent", "Netscape/7 [en] (Windows NT 5.1; U)");
user_pref("useragentswitcher.2.vendor", "");
user_pref("useragentswitcher.2.vendorsub", "");
user_pref("useragentswitcher.3.appname", "Microsoft Internet Explorer");
user_pref("useragentswitcher.3.appversion", "4.0 (compatible; MSIE 6.0; Windows NT
5.1)");
user_pref("useragentswitcher.3.description", "Internet Explorer 6 (Windows XP)");
user_pref("useragentswitcher.3.platform", "Win32");
user_pref("useragentswitcher.3.useragent", "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1)");
user_pref("useragentswitcher.3.vendor", "");
user_pref("useragentswitcher.3.vendorsub", "");
user_pref("useragentswitcher.menu.hide", true);
user_pref("useragentswitcher.reset.onclose", true);
user_pref("useragentswitcher.user.agents.count", 3);
user_pref("xpinstall.enabled", false);
user_pref("xpinstall.whitelist.add", "");
```



Finished product

System Behavior

Login Screen Sessions

The KDE login screen has a button called "Menu". Under that there is a selection for "Session Type". We do not want the default list of session types to be available for our users. To fix this, we need to mark all but KDE as hidden. Go to a command shell prompt and do the following:

```
cd /usr/share/apps/kdm/sessions
```

Here if you list this directory you will see a lot of `.desktop` files. These are for all of the known desktop managers. We will need to keep the ones that are listed in the unmodified session selector and delete the rest to save disk space. Do the following:

```
cp kde.desktop metacity.desktop mwm.desktop twm.desktop /tmp
rm –rdf *
cp /tmp/kde.desktop .
cp /tmp/metacity.desktop .
cp /tmp/mwm.desktop .
cp /tmp/twm.desktop .
```

Now use vi to edit all these .desktop files EXCEPT `kde.desktop`. Add the following to each .desktop file:

```
Hidden=true
```

Now when you are at the login screen and select Menu and Session Type, you should only see KDE, default, custom, and failsafe.

Client Default Settings

All settings and actions in applications you run as the *root* user are primarily stored in the `/root` directory somewhere. You will want to make sure that when you are logged in as *root*, the applications are behaving the way you want them to for the public client. Once that is done, we can start to move these settings files to a location that will make them default for all users.

First, a quick brief on the skel. Fedora Linux uses a directory to serve files to every user who logs onto the workstation. This "profile" is stored in `/etc/skel` and is placed in the `/home/`*`loginname`* directory when someone logs in for the very first time. Since our public machine users will be served a clean home directory every time they log in, the files stored in the `skel` will be copied to them. We also do not want certain configuration files to be used by default on our *Build* station (ie. root account) so we will place those files in a different `skel` directory that will be overlayed during the image build.

/etc/skel

Default Environment and Application Settings
We need to copy the necessary files from `/root` (the *root* user's custom settings) to the default and custom `skel`.

Now we can copy over the rest of the files.

```
cd /root
cp –r .adobe /etc/skel
cp –r .acrobat /etc/skel
cp .bash_logout /etc/skel
```

Note that you might not have all of these depending on what applications you have installed and the settings you have made.

```
cd /root
cp .djvurc /etc/skel
cp .dmrc /etc/skel
cp .gtkrc* /etc/skel
cp .mime.types /etc/skel
cp .realplayerrc /etc/skel
cp –r .icons /etc/skel
cp –r .kde /etc/skel
cp –r .loki /etc/skel
cp default.arts-env /etc/skel
cp .fonts.conf /etc/skel
```

We will change some of the files in these directories later. For now, we just need to get things started.

Associating Konqueror File Types
When a user double clicks on a file, based on its type and extension, the appropriate application is launched. Since KDE likes to use its embedded viewers for most file types, we need to change this. As *root*, go into the Home directory icon on the desktop. This will open a Konqueror window. Under the Settings menu bar, select Configure Konqueror. This will open up a settings window. On the left side, there is a list of setting types. Select File Associations. This will present a cascading list of file types. Opening up one of these, like Images, we can select on GIF. On the right side will list the applications and behavior for GIF file viewing. Remove the applications listed and add our `/usr/local/bin/openweb` application. This will make GIF files open in the Firefox Web Browser. Also select the tab for embedded. You will want to remove embedded applications and select the radio button for launching in a separate application. Do this for all known file types. When you select APPLY or OK, KDE will update the associations. This info is then stored in `/root/.local`

We now need to get this info to our default user profile:

```
cd /root
cp –r .local /etc/skel
cd /etc/skel
chmod –R 444 .local
cd .local
```

Go into each subdirectory and remove all applications but the `openweb.desktop` and any other application you want to keep for viewing. The `openweb.desktop` file should look similar to this:

```
[Desktop Entry]
Categories=Application
Comment=Browse the Internet
Encoding=UTF-8
Exec=/usr/local/bin/openweb %u
Icon=/usr/local/e3/icons/start.png
InitialPreference=1
MimeType=application/xhtml+xml;image/jpeg;image/png;text/html;text/rdf;
text/x-bibtex;text/x-chdr;text/x-c++hdr;text/x-diff;text/x-java;text/x-
log;text/x-objcsrc;text/x-pascal;image/gif
Name= Web Browser
GenericName=Web Browser
StartupWMClass=Firefox
X-KDE-SubstituteUID=false
Terminal=false
Type=Application
X-KDE-StartupNotify=true
```

Also you may want to include our custom pdf reader script for Adobe Acrobat. If so, go into the same directory as `openweb.desktop` and do the following:

Vi pdfview.desktop

Make it look like this:

```
[Desktop Entry]
Encoding=UTF-8
MimeType=application/pdf;application/postscript;image/x-
eps;application/x-gzpostscript;application/illustrator
InitialPreference=6
Exec=/usr/local/bin/pdfview
Icon=/usr/local/e3/icons/pdf.png
Type=Application
GenericName=Adobe Acrobat
Name=Adobe Acrobat
Name[en_US]=Adobe Acrobat
Terminal=0
ServiceTypes=
X-KDE-StartupNotify=true
```

Open With…

We need to create a special config directory to store custom KDE context menus. The following is for when a user selects an item and right clicks to bring up a context menu, then selects *Open With…* they will see relevant applications that are defined here.

```
mkdir /etc/skel/.config
mkdir /etc/skel/.config/menus
```

Now we will copy over the default *Open With* menus:

```
cp /etc/xdg/menus/applications.menu.kde /etc/skel/.config/menus
```

Now we can edit the menu to reflect our helper applications:

```
cd /etc/skel/.config/menus
vi applications.menu.kde
```

Change this file to look like the following:

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
 "http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd">

<Menu>

  <Name>Applications</Name>
  <Directory>Applications.directory</Directory>

<!-- Read standard .directory and .desktop file locations -->
  <DefaultDirectoryDirs/>

  <!-- Read in overrides and child menus from applications.d -->
  <!-- Merge in these other files as submenus -->
  <Menu>
    <Name>Preferences</Name>
  <Deleted/>
</Menu>

  <Menu>
    <Name>System Settings</Name>
  <Deleted/>
</Menu>

  <!-- Menu items to include in the toplevel -->
  <Include>
    <And>
      <Category>Core</Category>
      <Not>
        <Category>Merged</Category>
      </Not>
    </And>
  </Include>

  <!-- Menu items to exclude from the toplevel -->
  <Exclude>
    <Filename>gnome-search-tool.desktop</Filename>
  </Exclude>

  <!-- Accessories submenu -->
  <Menu>
    <Name>Accessories</Name>
```

```
  <Deleted/>
</Menu> <!-- End Accessories -->

  <!-- Development Tools -->
  <Menu>
    <Name>Development</Name>
  <Deleted/>
</Menu> <!-- End Development Tools -->

  <!-- Editors -->
  <Menu>
    <Name>Editors</Name>
  <Deleted/>
</Menu> <!-- End Editors -->

  <!-- Edutainment -->
  <Menu>
    <Name>Edutainment</Name>
   <Deleted/>
</Menu> <!-- End Edutainment -->

  <!-- Games -->
  <Menu>
    <Name>Games</Name>
  <Deleted/>
</Menu> <!-- End Games -->

  <!-- Graphics -->
  <Menu>
    <Name>Graphics</Name>
  <Deleted/>
</Menu> <!-- End Graphics -->

  <!-- Internet -->
  <Menu>
    <Name>Internet</Name>
    <LegacyDir>/usr/local/bin/applications</LegacyDir>
  </Menu>   <!-- End Internet -->

  <!-- Multimedia -->
  <Menu>
    <Name>Multimedia</Name>
  <Deleted/>
</Menu>   <!-- End Multimedia -->

  <!-- Office -->
  <Menu>
    <Name>Office</Name>
  <Deleted/>
</Menu> <!-- End Office -->

  <!-- System Tools-->
  <Menu>
    <Name>System</Name>
  <Deleted/>
  </Menu>   <!-- End System Tools -->

  <!-- Other -->
  <Menu>
    <Name>Other</Name>
  <Deleted/>
</Menu> <!-- End Other -->

<!-- File Created by CrossOver Office cxmenu -->
<Menu>
  <Name>Applications</Name>
<LegacyDir>/usr/cxoffice/support/xdg-legacy-
menus/Windows_Applications/Programs</LegacyDir></Menu>
</Menu>
</Menu> <!-- End Applications -->
```

To make sure Konqueror uses these settings also, you need to copy the `applications.menu`:

```
cp /etc/xdg/menus/applications.menu /etc/skel/.config/menus
```

Edit the `/etc/skel/.config/menus/applications.menu` to look like this:

```
<!DOCTYPE Menu PUBLIC "-//freedesktop//DTD Menu 1.0//EN"
 "http://www.freedesktop.org/standards/menu-spec/1.0/menu.dtd">

<Menu>

  <Name>Applications</Name>

  <!-- Read standard .directory and .desktop file locations -->
  <DefaultAppDirs/>

  <!-- Menu items to exclude from the toplevel -->
  <Exclude>
    <Filename>gnome-search-tool.desktop</Filename>
  </Exclude>

  <!-- Accessories submenu -->
  <Menu>
    <Name>Accessories</Name>
 <Deleted/>
</Menu> <!-- End Accessories -->

  <!-- Development Tools -->
  <Menu>
    <Name>Development</Name>
 <Deleted/>
</Menu> <!-- End Development Tools -->

  <!-- Editors -->
  <Menu>
    <Name>Editors</Name>
 <Deleted/>
</Menu> <!-- End Editors -->

  <!-- Edutainment -->
  <Menu>
    <Name>Edutainment</Name>
 <Deleted/>
</Menu> <!-- End Edutainment -->

  <!-- Games -->
  <Menu>
    <Name>Games</Name>
 <Deleted/>
</Menu> <!-- End Games -->

  <!-- Graphics -->
  <Menu>
    <Name>Graphics</Name>
 <Deleted/>
</Menu> <!-- End Graphics -->

  <!-- Internet -->
  <Menu>
    <Name>Internet</Name>
    <LegacyDir>/usr/local/bin/applications</LegacyDir>
  </Menu>   <!-- End Internet -->

  <!-- Multimedia -->
  <Menu>
```

```
    <Name>Multimedia</Name>
 <Deleted/>
</Menu>   <!-- End Multimedia -->

  <!-- Office -->
  <Menu>
    <Name>Office</Name>
 <Deleted/>
</Menu> <!-- End Office -->

  <!-- System Tools-->
  <Menu>
  <DontShowEmpty/>
  <Name>System</Name>
    <Include>
      <Category>System</Category>
        <Filename>kfmclient.desktop</Filename>
        <Filename>kfmclient_dir.desktop</Filename>
    </Include>

    <Exclude>
      <Category>TextEditor</Category>
      <Category>Utility</Category>
      <Category>TerminalEmulator</Category>
      <Category>X-KDE-More</Category>
      <Category>X-Red-Hat-Base</Category>
      <Filename>kuser.desktop</Filename>
      <Filename>konsole.desktop</Filename>
      <Filename>konsolesu.desktop</Filename>
      <Filename>kpackage.desktop</Filename>
      <Filename>krfb.desktop</Filename>
      <Filename>ksysv.desktop</Filename>
      <Filename>krandrtray.desktop</Filename>
      <Filename>kcron.desktop</Filename>
      <Filename>ksysguard.desktop</Filename>
      <Filename>redhat-authconfig.desktop</Filename>
      <Filename>system-config-proc.desktop</Filename>
      <Filename>system-config-kickstart.desktop</Filename>
      <Filename>system-config-display.desktop</Filename>
      <Filename>system-config-date.desktop</Filename>
      <Filename>system-config-keyboard.desktop</Filename>
      <Filename>system-config-language.desktop</Filename>
      <Filename>system-config-mouse.desktop</Filename>
      <Filename>system-config-nfs.desktop</Filename>
      <Filename>system-config-packages.desktop</Filename>
      <Filename>system-config-rootpassword.desktop</Filename>
      <Filename>system-config-securitylevel.desktop</Filename>
      <Filename>system-config-services.desktop</Filename>
      <Filename>system-config-soundcard.desktop</Filename>
      <Filename>system-config-users.desktop</Filename>
      <Filename>redhat-neat.desktop</Filename>
      <Filename>redhat-printconf-gui.desktop</Filename>
      <Filename>redhat-rhn-up2date-config.desktop</Filename>
      <Filename>redhat-switchdesk.desktop</Filename>
      <Filename>redhat-switchmail.desktop</Filename>
      <Filename>redhat-switch-printer.desktop</Filename>
    </Exclude>
  </Menu>   <!-- End System Tools -->
  <!-- Other -->
  <Menu>
    <Name>Other</Name>
 <Deleted/>
</Menu> <!-- End Other -->
<Menu>
 <Name>Windows Applications</Name>
<LegacyDir>/usr/cxoffice/support/xdg-legacy-
menus/Windows_Applications/Programs</LegacyDir></Menu>
</Menu> <!-- End Windows Applications -->

</Menu> <!-- End Applications -->
```

After you have edited the files, do the following to make them read only:

```
chmod 555 *
```

Some KDE applications will still show up on our custom Open-With so we need to tell those applications not to show up:

```
cd /usr/share/applications/kde
```

Edit using the vi editor each of the following files by adding an entry `Hidden=true` to each of them:

```
krfb.desktop
ksysguard.desktop
kpackage.desktop
kuser.desktop
krandrtray.desktop
sysv.desktop
kcron.desktop
```

Now you will need to add items for the custom E3 menu we just created:

```
mkdir /usr/local/bin/applications
```

Now any .desktop file that we place in `/usr/local/bin/applications` will be available to the user to *Open With*.

Default Konqueror Embedded Display

Konqueror has various mimetypes associated with different filetypes. An html file when double clicked on will open up in Konqueror. We do not want these files to open using the embedded viewers in Konqueror. To change this behavior, we need to do the following:

```
cd /etc/skel/.kde/share
mkdir mimelnk
cd mimelnk
mkdir application
mkdir audio
mkdir image
mkdir text
```

Now we need to populate these directories with some KDE instruction files.

```
cd application
vi pdf.desktop
```

Make it look like this:

```
[Desktop Entry]
Hidden=false
Patterns=*.pdf;*.PDF
X-KDE-AutoEmbed=false
```

Now for audio:

```
cd ..
cd audio
vi basic.desktop
```

Make it look like this:

```
[Desktop Entry]
Hidden=false
X-KDE-AutoEmbed=false
```

Now for images, we will have four files…

```
cd ..
cd image
vi gif.desktop
```

```
[Desktop Entry]
Hidden=false
Patterns=*.gif;*.GIF
X-KDE-AutoEmbed=false
```

```
vi jp2.desktop
```

```
[Desktop Entry]
Hidden=false
X-KDE-AutoEmbed=false
```

```
vi jpeg.desktop
```

```
[Desktop Entry]
Hidden=false
Patterns=*.jpg;*.JPG;*.jpeg;*.JPEG
X-KDE-AutoEmbed=false
```

```
vi png.desktop
```

```
[Desktop Entry]
Hidden=false
```

```
Patterns=*.png;*.PNG
X-KDE-AutoEmbed=false
```

And finally we will do text:

```
cd ..
cd text
vi html.desktop
```

```
[Desktop Entry]
Hidden=false
Patterns=*.html;*.HTML;*.htm;*.HTM;*.shtml
X-KDE-AutoEmbed=false
```

```
vi plain.desktop
```

```
[Desktop Entry]
Hidden=false
Patterns=*.txt
X-KDE-AutoEmbed=false
```

```
cd /etc/skel/.kde/share/mimelnk
chmod –R 555 *
```

Now files of these types should not launch the embedded Konqueror viewer.

`.desktop` Files

Before we go on, a brief explanation of KDE's `.desktop` format is in order. A desktop file contains entries describing the application and the behavior in which it functions and is displayed. The following is line by line description of each component. Not all or necessary and some are optional.

Label		Required?	Values	Description
`[Desktop Entry]`	←	●		Beginning header for file
`Encoding=`	←		UTF-8	Encoding type
`Type=`	←	●	Application Directory	What is this file referencing?
`Exec=`	←		*Filename*	What file to execute (use full path when needed)
`Icon=`	←		*Icon filename*	What icon file to use (normally located in /usr/share/icons)
`Name=`	←	●	*Application/directory name*	Name of the application or directory
`GenericName=`	←		*Application name*	Application name to display in menus
`Categories=`	←		*Application categories separated by semicolons with no spaces*	*Application categories separated by semicolons with no spaces*
`Hidden=`	←		`true` `false`	Make the icon or menu entry visible or not

More `.desktop` Files

We still need a `.desktop` file for Adobe Acrobat Reader, so do the following:

```
cd /usr/local/bin/applications
vi pdfview.desktop
```

Make the `pdfview.desktop` file look like this:

```
[Desktop Entry]
Encoding=UTF-8
MimeType=application/pdf;application/postscript;image/x-
eps;application/x-gzpostscript;application/illustrator
InitialPreference=6
Exec=/usr/local/bin/pdfview
Icon=/usr/local/e3/icons/pdf.png
Type=Application
GenericName=Adobe Acrobat
Name=Adobe Acrobat
Name[en_US]=Adobe Acrobat
Terminal=0
ServiceTypes=
#X-KDE-Library=libkghostviewpart
X-KDE-StartupNotify=true
```

While we are at it, let's make Firefox an application they can open with:

```
vi openweb.desktop
```

Make it look like this:

```
[Desktop Entry]
Comment=Browse the Internet
Comment[en_US]=Browse the Internet
Encoding=UTF-8
Exec=/usr/local/bin/openweb %u
GenericName=Web Browser
GenericName[en_US]=
Icon=/usr/local/e3/icons/start.png
InitialPreference=1
MimeType=application/xhtml+xml;image/jpeg;image/png;text/html;text/rdf;
text/x-bibtex;text/x-chdr;text/x-c++hdr;text/x-diff;text/x-java;text/x-
log;text/x-objcsrc;text/x-pascal;image/gif
Name=Web Browser
Name[en_US]=Web Browser
StartupNotify=true
StartupWMClass=Firefox
Terminal=false
Type=Application
X-KDE-StartupNotify=true
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Kicker Bar Buttons

Now we need to create some more `.desktop` files. These are for our custom application launching from the *kicker* (task bar) menu buttons. You will want to create these in the `/usr/share/applications` directory.

```
cd /usr/share/applications
vi home-init.desktop
```

This file will create the users special desktop when they log on. The `home-init.desktop` file for our library looks like this:

```
[Desktop Entry]
Encoding=UTF-8
Comment=
Comment[en_US]=map afs home space
Exec='/usr/local/bin/home-init.sh'
Name=map-afs
Name[en_US]=map-afs
Terminal=false
Type=Application
Categories=Application
X-KDE-SubstituteUID=false
```

Next is the logout button.

```
cd /usr/share/applications
vi logout.desktop
```

```
[Desktop Entry]
Comment[en_US]=Logout of this session
Encoding=UTF-8
Exec=/usr/bin/dcop ksmserver ksmserver logout 0 0 0\s
Icon=/usr/local/e3/icons/logout.png
MimeType=
Name[en_US]=Logout
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
Categories=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
X-KDE-StartupNotify=false
```

Next is the button to open the Web browser.

```
cd /usr/share/applications
vi openweb.desktop
```

```
[Desktop Entry]
```

```
Encoding=UTF-8
Name=openweb
Name[da]=Weblï¿½er
Name[de]=Web-Browser
Name[el]=?????????? ??????????
Name[es]=Navegador de Web
Name[fr]=Navigateur Web
Name[is]=Vafri
Name[it]=Browser Web
Name[ja]=Web????
Name[ko]=? ????
Name[ms]=Pelungsur Web
Name[nl]=Web Bladerprogramma
Name[no]=Nettleser
Name[pt]=Navegador Web
Name[ru]=???-???????
Name[sv]=Webblï¿½are
Name[tr]=Web ?stemcisi
Name[zh_CN]=??????
Name[zh_TW]=?????
Comment=Browse the Internet
Comment[da]=Surf pï¿½Internet
Comment[de]=Im Internet browsen
Comment[el]=????????? ??? ?????????
Comment[es]=Navegar por Internet
Comment[fr]=Naviguer sur Internet
Comment[is]=Flakka um neti
Comment[it]=Esplorare Internet
Comment[ja]=??????????
Comment[ko]=??? ??
Comment[ms]=Lungsur Internet
Comment[nl]=Over het Internet surfen
Comment[no]=Nettleser
Comment[pt]=Navegue na Internet
Comment[ru]=???????? ????????
Comment[sv]=Surfa pï¿½Internet
Comment[tr]=Internet'e Gzat
Comment[zh_CN]=?????
Comment[zh_TW]=??????
Exec=/usr/local/bin/openweb %u
Icon=/usr/local/e3/icons/start.png
Terminal=0
Type=Application
Categories=Application
X-KDE-StartupNotify=true
StartupWMClass=Firefox
```

```
MimeType=text/html;application/xhtml+xml;image/gif;ima
ge/jpeg;image/png
```

Our library also uses a Web survey for feedback, so this button is created.

```
cd /usr/share/applications
vi survey.desktop
```

```
[Desktop Entry]
Encoding=UTF-8
Name=survey
Comment=Take the Library Survey!
Exec=/usr/local/bin/openweb "http://urltosurvey"
Icon=/usr/local/e3/icons/survey3.png
Terminal=0
Type=Application
Categories=Application
X-KDE-StartupNotify=false
```

And since we are using Codeweaver's CrossOver product which is a modified implementation of WINE, we will want to create a "Kill" button. This will let the user terminate a "hung" MS Windows application (which sometimes happens) without affecting the rest of the user's session.

```
cd /usr/share/applications
vi winekill.desktop
```

```
[Desktop Entry]
Encoding=UTF-8
Name=WineKill
Comment=Terminates stalled Windows Viewer Applications
Exec=/usr/local/bin/winekill
Icon=/usr/local/e3/icons/kill.png
Terminal=0
Type=Application
Categories=Application
X-KDE-StartupNotify=false
```

KDE Login Scripts

Now we need to create the shell scripts that are called by the `.desktop` files we just created. First will be a script we will use later to clean out the /tmp directory.

```
cd /usr/local/bin
vi cleantmp
```

```
#!/bin/bash
# E3-script  Cleantmp script
# Cleans out tmp directory of all files owned by a user
with a UID over 500
#
# Jul 11 2003 -- DPB -- Initial Version

pushd /tmp
stat -c "%u|%n" * | gawk 'BEGIN {FS="|"} {if ($1 >= 500)
system (sprintf("rm -fr \"%s\"", $2));}'
popd

pushd /var/tmp
stat -c "%u|%n" * | gawk 'BEGIN {FS="|"} {if ($1 >= 500)
system (sprintf("rm -fr \"%s\"", $2));}'
popd

#remove crossover WINE files - Jan 6 2004 -- PCH --
pushd /tmp
if [ $UID > 500 ]; then
                        rm -rf .wine*
fi
popd
```

KDE Autostart

KDE has a default autostart folder where `.desktop` files reside to execute when a user logs in. Some of the default `.desktop` files we do not want to use. We remove the following but it is up to you:

```
cd /usr/share/autostart

rm kgpg.desktop
rm irkick.desktop
rm korgac.desktop
```

We also want to lock down the desktop so users cannot save to it. To do this, we will create a file in `/usr/share/autostart` which refers to a script in `/usr/local/bin`:

```
cd /usr/share/autostart
vi lockdesktop.desktop
```

Make it look like this:

```
[Desktop Entry]
Comment=
Comment[en_US]=
Encoding=UTF-8
```

```
Exec='/usr/local/bin/lockdesktop.sh'
GenericName=
GenericName[en_US]=
Icon=
MimeType=
Name=lockdesktop
Name[en_US]=lockdesktop
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Now we will make the script it refers to:

```
cd /usr/local/bin
vi lockdesktop.sh
```

Make it look like the following:

```
#!/bin/bash

# E3-script
#  lock the desktop from being written to

if [ $UID -gt 0 ] ; then
        chmod 500 $HOME/Desktop
        chmod 500 $HOME/.kde/altdesktop
fi
```

Now when a user logs on, the desktop will be locked to read-only mode.

User Home Desktop

Since we want to make available on the user's desktop some icons to directories, you need to create the desktop links. KDE has a special place for default desktops (KDE does not use the `/etc/skel` for this!) located in the `/usr/share/apps/kdesktop/DesktopLinks` directory. We will make the standard `.desktop` files here (actually, we do not need to label them with the `.desktop` suffix, but you can if you want). First we will make the user's *Home* directory:

```
cd /usr/share/apps/kdesktop/DesktopLinks
vi Home
```

Make the file `Home` look like this:

```
[Desktop Entry]
Comment=
Comment[en_US]=
Encoding=UTF-8
Exec=konqueror --profile pac.profile
GenericName=Personal Files
GenericName[en_US]=Personal Files
Icon=/usr/local/e3/icons/kfm_home.png
MimeType=
Name=Home – My Home Space
Name[en_US]=Home – My Home Space
StartupNotify=true
Terminal=false
TerminalOptions=
Type=Application
X-DCOP-ServiceType=
X-KDE-SubstituteUID=false
X-KDE-Username=
```

```
cp Home /usr/share/applications
```

Next we will create a link to the AFS network filespace:

```
ln –s /usr/share/applications/map-afs.desktop map-afs
```

Now we need to change the permissions on these files:

```
chmod 644 *
```

That is it for the Desktop. Now we need to create the file that will link to the *Disks & Media* directory in the user's *Home* directory.

```
cd /usr/share/applications
vi Disks & Media
```

Make the file *Disks & Media* look like this:

```
[Desktop Entry]
Encoding=UTF-8
GenericName=Removable Media
Icon=/usr/local/e3/icons/3floppy_unmount.png
Name=Disks & Media
Name[en_US]=Disks & Media
Type=Link
URL=$HOME/Disks & Media
```

To make Disks & Media available to each user, we need to put it in the /etc/skel. To do this, all we need to do is create a symbolic link to the actual file:

```
cd /etc/skel
ln –s /mnt /etc/skel/”Disks & Media”
```

We are now finished with the user’s desktop items.

AFS Network Mapping

The next script is unique to our university's network. This will map user AFS network filespace to a drive on the desktop. You can customize it to meet your needs.

```
cd /usr/local/bin
vi map-afs.sh
```

```
#!/bin/bash
#E3-script
# This is a script to create a symlink and a desktop shortcut to an ASU user's
# AFS Space.
#
# Jun 11 2003 -- DPB -- Initial version
# Jul 11 2003 -- DPB -- Added code to lock desktop
# Jan 10 2004 -- PCH -- Added altdesktop entry
# The minimum UID to try this with (everything less than this is not an ASU
# affiliate in TS&D's implementation.
MINIMUM_UID=20000
KDE_DESKTOP_FILE_NAME="$HOME/Desktop/afs.desktop"
KDE_ALT_DESK_FILE_NAME="$HOME/.kde/altdesktop/afs"
KDE_DESKTOP_FILE_DESC="ASU AFS Network Space"
KDE_DESKTOP_FILE_ICON="/usr/local/e3/icons/afs_mount.png"

# Make sure we have a home directory at this point
if [ ! -d $HOME/Desktop ] ; then
                                          exit 1
fi

#all non student account users (generic logins) are served
#via LDAP a UID greater than 1000 but less than 20000
#so we will not give them access to AFS space
if [ $UID -lt 20000 ]; then
        exit 1
fi

#if the UID is not root then create the AFS items
if [ $UID -gt 0 ] ; then
        chmod 700 $HOME/Desktop
        chmod 700 $HOME/.kde/altdesktop
fi

# Check to see if we have a valid UID, and the USER variable is set and is
# an appropriate length
if [ $UID -ge $MINIMUM_UID -a -n "$USER" -a ${#USER} -ge 4 ] ; then \
         AFS_PATH="/afs/asu.edu/users/${USER:0:1}/${USER:1:1}/${USER:2:1}/$USER"

               # Looks good!  Make the symlink.
               pushd $HOME
               ln -s $AFS_PATH afs
               popd

#Now, we are going to create the link for konqueror for the side bar. - JJF
               echo "[Desktop Entry]" > $KDE_ALT_DESK_FILE_NAME
               echo "Encoding=UTF-8" >> $KDE_ALT_DESK_FILE_NAME
               echo "Name=$KDE_DESKTOP_FILE_DESC" >> $KDE_ALT_DESK_FILE_NAME
               echo "Icon=$KDE_DESKTOP_FILE_ICON" >> $KDE_ALT_DESK_FILE_NAME
               echo "URL=$AFS_PATH" >> $KDE_ALT_DESK_FILE_NAME
               echo "Type=Link" >> $KDE_ALT_DESK_FILE_NAME
fi

if [ $UID -gt 0 ] ; then
                                    chmod 500 $HOME/Desktop
        chmod 500 $HOME/.kde/altdesktop
fi
```

The button placed on the Kicker bar needs to execute a script to check if the user actually has AFS access. To make this script, do the following:

```
cd /usr/local/bin
vi afsinit
```

```
#!/bin/bash
# E3-script

if [ "$UID" -lt "20000" ]; then
              kdialog --error "AFS Network File Space requires you to login to this
workstation with your ASURITE ID"
              exit 1
fi

MYDATE=`date +%m-%d-%Y`

AFS_PATH="/afs/asu.edu/users/${USER:0:1}/${USER:1:1}/${USER:2:1}/$USER"

if [ -e "/tmp/${USER}-${MYDATE}" ]; then

              konqueror -mimetype inode/directory file:$AFS_PATH

else

              MYPASS=`kdialog --password "Enter AFS Password"`
              echo "$MYPASS" | klog ${USER}@ASU.EDU -pipe
              RES=`tokens | grep -i asu.edu | wc -l`
              echo "this is the result: $RES"

              if [ "$RES" == "0" ]; then
                        kdialog --error "AFS Password Incorrect"
              else
                        echo "Success for ${USER}@ASU.EDU" > /tmp/${USER}-${MYDATE}
                        /usr/local/bin/map-afs.sh
                         konqueror -mimetype inode/directory file:$AFS_PATH
              fi
              echo $RES
fi
```

To make this script execute via a kicker button, we need to create a `.desktop` file. Do the following:

```
cd /usr/share/applications
vi map-afs.desktop
```

Make it look like this:

```
[Desktop Entry]
Comment=Check and map AFS network volume to desktop
Comment[en_US]=Check and map AFS network volume to desktop
Encoding=UTF-8
Exec='/usr/local/bin/afsinit'
GenericName=AFS Network Filespace
GenericName[en_US]= AFS Network Filespace
Icon=/usr/local/e3/icons/afs_mount.png
MimeType=
Name= AFS Network Filespace
```

```
Name[en_US]= AFS Network Filespace
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
Categories=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Now we need to change the permissions of the `map-afs.desktop` file:

```
chmod 644 map-afs.desktop
```

AFS should be set to load when authenticated during logon. To verify this, we will check the PAM module loading for KDE:

```
cd /etc/pam.d
vi kde
```

It should look something like this:

```
#%PAM-1.0
auth       required           pam_stack.so service=system-auth
auth       required           pam_nologin.so
auth       sufficient         pam_timestamp.so
account    required           pam_stack.so service=system-auth
password   required           pam_stack.so service=system-auth
session    required           pam_stack.so service=system-auth
session    required           /lib/security/pam_mkhomedir.so
session    optional           /lib/security/pam_krb5afs.so
session    optional           pam_timestamp.so
session    optional          pam_selinux.so
session    optional          pam_console.so
```

Custom Web Homepage

To allow our users to open a Web browser (Firefox) we will want to create a script that checks the user id (UID) served from the LDAP server identifying the user as either a valid library/university account (UID>=20000) or a public account with the machine name (UID<20000). Based on the UID a separate Web page can be served describing the services available for that user.

```
cd /usr/local/bin
vi openweb
```

```
#!/bin/bash
#
# E3-script
# A simple script to make Mozilla launch "nicely"
#
# Jun 25 2003 -- DPB -- Initial version
# Jul 11 2003 -- DPB -- Enabled checking to see if Mozilla is already running
# Jul 11 2003 -- DPB -- Allright, Mozilla's script just sucks too bad.  Rewriting this to
do everything we need...
# Aug 12 2004 -- PCH -- Updated to use Firefox in /usr/local/firefox instead of mozilla

# Constants that we may change
AFFILIATE_URL="http://valid-user.html"
GUEST_URL="http://guest-user.html"

# if we don't have any options, choose the appropriate URL
if [ ".$1" == "." ] ; then \

     if [ $UID -ge 20000 ] ; then \
                                    URL="$AFFILIATE_URL"
     else
                                      URL="$GUEST_URL"
     fi

else
# Trap options
if [ "${1:0:1}" == "-" ] ; then \
    echo "If you want to pass options, don't call me.  Call /usr/local/firefox/firefox"
     exit 1
fi

URL="$1"
fi



# Check to see if Firefox is running
ALREADY_RUNNING=`ps ax | grep firefox-bin | grep -v grep`
echo "$ALREADY_RUNNING" >> /tmp/fire.tmp
echo "$URL" >> /tmp/fire.tmp
if [ ! -z "$ALREADY_RUNNING" ] ; then \
        exec /usr/local/firefox/firefox -width 1024 -height 680 -a firefox -remote
"openURL("$URL",new-window)"

else
              exec /usr/local/firefox/firefox -width 1024 -height 680 "$URL"

fi
```

Frozen Windows Applications

The script for our "Kill Application" MS Windows application terminator uses *CrossOver*. If you are using a vanilla version of *WINE*, you will want to replace the call to `wine-pthread` with a call to `wine-kthread` and remove the `-9` switch.

```
cd /usr/local/bin
vi winekill
```

```
#!/bin/sh
# E3 version 2 12-15-2004
#First, we are going to make sure that there are Wine Apps running.

EXISTENCE=`ps ax | grep wine-pthread | grep -v grep`

if [ -z "$EXISTENCE"]; then
kdialog --error "There are no stalled Windows applications to terminate."

#JJF - Here we are going to confirm that the user really wants to kill all of the wine
apps.

else

kdialog --title "Kill Stalled Windows Applications?" --warningyesno "Are you sure you
want to kill all Stalled Windows Applications?" || exit 0

#if the user confirms that they want to kill wine apps
#then the killall is issued to all of the wine threads
#this leaves the winserver running so it doesn't take too
#long to restart that app.

killall -9 wine-pthread
fi
```

/etc/skel/.mozilla

We now want to take what we have done as user `root` in Firefox and make it default for all users.

```
cd /root
cp –r .mozilla /etc/skel
```

We have to modify a profile file that our public machine users will use defining where Firefox will access settings. Do the following:

```
cd /etc/skel/.mozilla/firefox
vi profiles.ini
```

Change the settings to reflect the following. Take note that the `h7udmkmu.default` entry may be different on your machine. If you `ls` the `/etc/skel/.mozilla/firefox` directory use the directory name that has `.default` as the suffix.

```
IsRelative=0
Path=/home/pacsym/.mozilla/firefox/h7udmkmu.default
```

Now we need to remove all of the embedded path entries referencing the `root` user directory `/root` from all Firefox resources in `/etc/skel/.mozilla`. We need to replace these entries with a path to `/home/pacsym`. Remember `pacsym`? This is why we created it. This way Firefox will always think the user's `home` directory is `/home/pacsym` but `pacsym` is a symbolic link to the real user's `home` directory. It makes things easy. We can make these changes by using global search and replace commands in the `vi` editor. Do the following:

```
cd /etc/skel/.mozilla
grep –r "/root" *
```

This will list the files that have a reference to the `/root` directory. These files are what we need to edit. Change directory (`cd`) into where the first file is to edit. Use the `vi` editor to open the file. While in `vi`, execute the global search and replace command by typing the `:` colon and in the command line type:

```
%s/root\/.mozilla/home\/pacsym\/.Mozilla/g
```

All references `to /root/.mozilla` should now be replaced with `/home/pacsym/.mozilla`. Do this to all files until `/root/.mozilla` does not exist in any of them. You can execute the `grep` for root command just to verify.

Default User Desktop Items

Now that we have created the buttons for the Kicker bar, we need to configure the Kicker to use them and make them default for all users. Since we copied over some directories from root's `/root` directory, we will need to clean up that which we do not want to keep in `/etc/skel`.

```
rm /etc/skel/.kde/share/config/clock*
rm /etc/skel/.kde/share/config/kickerrc
rm /home/.directory
rm –rdf /etc/skel/.kde/share/config/session
rm –rdf /etc/skel/.kde/share/config/kresources
rm –rdf /etc/skel/.kde/share/config/kpilot*
rm –rdf /etc/skel/.kde/share/config/conquerorrc
rm –rdf /etc/skel/.kde/share/config/dockbar_panel*
rm –rdf /etc/skel/.kde/share/config/dummy
rm –rdf /etc/skel/.kde/share/config/email*
rm –rdf /etc/skel/.kde/share/config/kate*
rm –rdf /etc/skel/.kde/share/config/kcalcrc
rm –rdf /etc/skel/.kde/share/config/kchar*
rm –rdf /etc/skel/.kde/share/config/kcontrolrc
rm –rdf /etc/skel/.kde/share/config/kcookie*
rm –rdf /etc/skel/.kde/share/config/klipperrc
rm –rdf /etc/skel/.kde/share/config/kmail*
rm –rdf /etc/skel/.kde/share/config/konq*
```

```
rm –rdf /etc/skel/.kde/share/config/konsolerc
rm –rdf /etc/skel/.kde/share/config/kpgp*
rm –rdf /etc/skel/.kde/share/config/ksms*
rm –rdf /etc/skel/.kde/share/config/ksnap*
rm –rdf /etc/skel/.kde/share/config/kuick*
rm –rdf /etc/skel/.kde/share/config/kwrite*
cd /etc/skel/.kde
rm cache*
rm socket*
rm tmp*
rm –rdf Autostart
```

uiserverrc

We also need to edit the `uiserverrc` file to look like the following:

```
[$i]
[UIServer]
ShowList=false
ShowStatusBar=false
ShowSystemTray=false
```

kioslaves

The KDE io slave resource files need to be edited so that when a user opens the filemanager/konqueror there are no cached information presented which may give a false display. Do the following

```
cd /etc/skel/.kde/share/config
vi kio_httprc
```

Make it look like the following:

```
MaxCacheSize=5120
UseCache=false
cache=Refresh

[$Version]
update_info=kioslave.upd:kde2.2/r1,kioslave.upd:kde2.2/r2
```

Now we need to edit the main io:

```
vi kioslaverc
```

Make it look like the following:

```
MaxCacheSize=5120
UseCache=false
cache=Refresh

[$Version]
```

```
update_info=kioslave.upd:kde2.2/r1,kioslave.upd:kde2.2/r3
```

Kickerrc

Now we can customize the Kicker bar.

```
cd /usr/share/config
vi kickerrc
```

Edit the `kickerrc` file to look like the following:

```
[$Version][$i]
update_info=kickerrc.upd:kde_3_1_sizeChanges

[Applet_1]
ConfigFile=klipperrc
DesktopFile=klipper.desktop
FreeSpace=1
WidthForHeightHint=30

[Applet_2]
ConfigFile=taskbar_panelappletrc
DesktopFile=taskbarapplet.desktop
FreeSpace=0.00431965
WidthForHeightHint=46

[Applet_3]
ConfigFile=clockappletrc
DesktopFile=clockapplet.desktop
FreeSpace=1
WidthForHeightHint=75

[Applet_4]
ConfigFile=knewsticker_panelappletrc
DesktopFile=knewsticker.desktop
FreeSpace=0.847619
WidthForHeightHint=180

[Applet_5]
ConfigFile=kmix_panelappletrc
DesktopFile=kmixapplet.desktop
FreeSpace=1
WidthForHeightHint=64

#E3
#
#REMOVE KMenuButton for final image - PCH 12-2004
#
[General][$i]
Alignment=0
Applets=KMenuButton_1,ServiceButton_1,ServiceButton_4,ServiceButton_2,A
pplet_2,ServiceButton_3,Applet_4,ServiceButton_6,ServiceButton_5,Applet
_1,Applet_5,Applet_3
AutoHideDelay=3
AutoHidePanel=false
```

```
AutoHideSwitch=false
BackgroundHide=false
BackgroundTheme=/usr/share/apps/kicker/wallpapers/default.png
ColorizeBackground=false
CustomSize=64
ExpandSize=true
HideAnimation=true
HideAnimationSpeed=40
Position=3
ShowToolTips=true
ShowLeftHideButton=false
ShowRightHideButton=false
Size=4
Transparent=true
UnhideLocation=6
UseBackgroundTheme=true
XineramaScreen=0
SizePercentage=100
TintColor=191,191,191
TintValue=42
FadeOutAppletHandles=false
HideAppletHandles=true

[KMenu]
UseSidePixmap=false

[ServiceButton_1]
DesktopFile=/usr/share/applications/openweb.desktop
FreeSpace=0

[ServiceButton_2]
DesktopFile=/usr/share/applications/diskformatter2.desktop
FreeSpace=0.00483092

#[ServiceButton_3]
#DesktopFile=/usr/share/applications/map-afs.desktop
#FreeSpace=0.810945

[ServiceButton_3]
DesktopFile=/usr/share/applications/Home
FreeSpace=0.810945

[ServiceButton_4]
DesktopFile=/usr/share/applications/survey.desktop
FreeSpace=0

[ServiceButton_5]
DesktopFile=/usr/share/applications/logout.desktop
FreeSpace=1

[ServiceButton_6]
DesktopFile=/usr/share/applications/winekill.desktop
FreeSpace=1

[button_tiles][$i]
EnableBrowserTiles=false
EnableDesktopButtonTiles=false
```

```
EnableKMenuTiles=false
EnableURLTiles=false
EnableWindowListTiles=false

[buttons]
EnableIconZoom=true
EnableTileBackground=false

[menus][$i]
DetailedEntriesNamesFirst=false
DetailedMenuEntries=false
MaxEntries2=30
RecentVsOften=false
ShowHiddenFiles=false
UseBookmarks=false
UseBrowser=false
UseRecent=false
```

Desktoprc

The desktoprc file contains defaults for desktop behavior. We need to modify this so a user cannot right click on the desktop to perform actions. Also we will want to enable our custom screensaver.

cp /usr/share/config/desktoprc /etc/skel/.kde/share/config
cd /etc/skel/.kde/share/config
vi desktoprc

Add the "`Mouse Buttons`" entry as shown below, along with the screensaver entries:

```
[Desktop0]
Pattern=
Program=
Wallpaper=/e3custom/usr/local/e3/wallpaper/desktop.jpg
WallpaperList=
WallpaperMode=Tiled

[Mouse Buttons]
WheelSwitchesWorkspace=false
Left=
Middle=
Right=

[ScreenSaver]
DPMS-dependent=false
Enabled=true
LockGrace=false
Saver=/usr/share/applnk/System/ScreenSavers/logoutsaver.desktop
Timeout=600

[Version]
KDEVersionMajor=3
KDEVersionMinor=3
KDEVersionRelease=1
```

Konqueror

Konqueror is used by the public image as a file manager. There are some default menu items in Konqueror that we do not want our users to have access to. We can make these changes by doing the following.

```
cd /etc/skel/.kde/share/apps
rm –rdf *
mkdir konqueror
cp /usr/share/apps/conqueror/conqueror.rc .
```

Now you can edit out what you do not want a user to see in the file interface, like “run” “open terminal” etc.

```
vi konqueror.rc
```

Delete out the lines for menus you want removed. Here is an example:

```
<?xml version="1.0"?>
<!DOCTYPE gui SYSTEM "kpartgui.dtd">
<gui name="Konqueror" version="52">
<MenuBar>
 <Menu name="file" noMerge="1"><text>&amp;amp;Location</text>
  <Action name="new_window"/>
                                  <Action name="newtab"/>
  <Action name="duplicate_window"/>
  <Action name="open_location"/>
  <Separator/>
  <Separator/>
  <Merge/> <!--Includes termination separator-->
  <Action name="print"/>
  <DefineGroup name="print" />
  <Separator/>
  <ActionList name="openwith" /> <!--Includes termination separator-->
  <Action name="quit"/>
 </Menu>
 <Menu name="edit" noMerge="1"><text>&amp;amp;Edit</text>  <!--Here is what applies to a
selection-->
  <Action name="undo" />
  <Action name="rename"/>
  <Action name="trash"/>
  <ActionList name="delactions" />
  <ActionList name="operations" /><!-- copy files and move files, when two directory
views -->
  <Separator/>
  <WeakSeparator/>
  <Merge/>
 </Menu>
 <Menu name="view" noMerge="1"><text>&amp;amp;View</text> <!--Here is what applies to the
active view-->
  <ActionList name="viewmode" />
  <Separator/>
  <Action name="reload"/>
  <Action name="stop" />
 </Menu>
 <Menu name="go"><text>&amp;amp;Go</text>
  <Action name="up"/>
  <Action name="back"/>
  <Action name="forward"/>
  <Action name="home"/>
 </Menu>
 <Action name="bookmarks"/>
```

```
 <Menu hidden="true" name="tools" noMerge="1"><text>&amp;Tools</text>
  <Action name="findfile"/>
 </Menu>
 <Menu hidden="true" name="settings" noMerge="1"><text>&amp;Settings</text>
  <Action name="fullscreen"/>
 </Menu>
 <Menu hidden="true" name="window" noMerge="1"><text>&amp;Window</text>
 </Menu>
 <Menu hidden="true" name="help" noMerge="1"><text>&amp;Help</text>
 </Menu>
 <Merge/>
</MenuBar>
<ToolBar fullWidth="true" name="mainToolBar" newline="true"><text>Main Toolbar</text>
 <Action name="up"/>
 <Action name="back"/>
 <Action name="forward"/>
 <Action name="home"/>
 <Separator/>
 <Action name="reload"/>
 <Action name="stop"/>
 <Separator/>
 <Action name="print"/>
</ToolBar>
</gui>
```

konquerorrc

To lock down actions within the konqueror interface even more, we need to edit `konquerorrc` in the skel. We will use the KDE directive `[$i]` to make some entries immutable- users cannot change them. We also want to implement some rules for file and directory access. The below example does not allow a user to traverse, list, execute any file locations other than their home directory, afs network storage space, and the mounted storage devices. We also do not want them to access Microsoft networks via `smb://` directive nor ssh connections via the `fish://` directive.

```
cd /etc/skel/.kde/share/config
vi konquerorrc
```

Make it look like this:

```
[Java/JavaScript Settings]
EnableJavaScript=true
EnableJSDebugOutput=false
EnablePlugins=true

[HTML Settings]
AutoDelayedActions=true
AutoLoadImages=true
ChangeCursor=true
FormCompletion=true
HoverLinks=true
MaxFormCompletionItems=10
ShowAnimations=Enabled
UnderlineLinks=true
MediumFontSize=11
MinimumFontSize=9

[FMSettings]
DisplayFileSizeInBytes=false
ShowFileTips=false
StandardFont=Sans,10,-1,5,50,0,0,0,0,0
UnderlineLinks=false
WordWrapText=true
```

```
[Reusing]
AlwaysHavePreloaded=false
MaxPreloadCount=0
PreloadOnStartup=false
SafeParts=

[KFileDialog Settings]
Automatic Preview=false
ShowPreviews=false

[KDE Action Restrictions][$i]
shell_access=false
lock_screen=false
run_command=false
start_new_session=false
action/file_new=false
action/file_print=false
action/file_print_preview=false
action/kdesktop_rmb=false
action/menuedit=false
action/options_show_menubar=false
action/options_show_toolbar=true
action/options_show_statusbar=true
action/options_save_options=false
action/options_configure=false
action/options_configure_keybinding=false
action/options_configure_toolbars=false
action/options_configure_notifications=false
action/help=false
action/help_report_bug=false
action/options_configure=false
print/system=false
user/root=false
action/bookmarks=false
lineedit_text_completion=false
action/history=false
action/save_default=false
action/save_sessions_profile=false
action/options_configure_notifications=false
action/options_configure_keybinding=false
action/options_configure=false
action/settings=false
action/menuedit=false

[KDE URL Restrictions][$i]
rule_count=11
rule_1=open,,,,file,,,false
rule_2=open,,,,file,,$HOME,true
rule_3=open,,,,smb,,,false
rule_4=open,,,,file,,/afs,true
rule_5=open,,,,fish,,,false
rule_6=open,,,,file,,/mnt,true
rule_7=list,,,,file,,,false
rule_8=list,,,,file,,$HOME,true
rule_9=list,,,,file,,/afs,true
rule_10=list,,,,file,,/mnt,true
rule_11=open,,,,file,,$TMP,true
```

kdeglobals

The kdeglobals file contains information on the KDE interface which is shared by all applications. We need to set this up with action restrictions and settings that will be default for any user who logs on. Note that any additions to this file that exists in `/usr/share/config/kdeglobals` will merge with our custom file, while anything we have that is the same will override the `/usr/share/config/kdeglobals` file. You could copy this file to the `skel`

location and edit it for the changes instead if you want to, otherwise, we will do it from scratch:

```
cd /etc/skel/.kde/share/config
vi kdeglobals
```

Make it look like the following:

```
[$Version]
update_info=kwin.upd:kde3.2Xinerama,klippershortcuts.upd:04112002,kded.upd:kde3.0,socks.u
pd:kde3.0/r1,kaccel.upd:kde3.1/r3,kaccel.upd:kde3.3/r1

[DesktopIcons]
ActiveColor=169,156,255
ActiveEffect=togamma
ActiveSemiTransparent=false
ActiveValue=0.7
Animated=true
DefaultColor=144,128,248
DefaultEffect=none
DefaultSemiTransparent=false
DefaultValue=1
DisabledColor=34,202,0
DisabledEffect=togray
DisabledSemiTransparent=true
DisabledValue=1
DoublePixels=false

[General]
XftHintStyle=hintmedium
alternateBackground=238,246,255
background=234,233,232
buttonBackground=230,240,249
selectBackground=169,209,255
selectForeground=3,3,3
widgetStyle=keramik

[Global Shortcuts]
Activate Window Demanding Attention=default(Alt+Ctrl+A)
Defaults timestamp=Oct 20 200403:26:51
Desktop Screenshot=default(Ctrl+Print)
Enable/Disable Clipboard Actions=default(Alt+Ctrl+X)
Halt without Confirmation=default(Alt+Ctrl+Shift+PageDown)
Kill Window=default(Alt+Ctrl+Escape)
Lock Session=none
Logout=default(Alt+Ctrl+Delete)
Logout without Confirmation=none
Manually Invoke Action on Current Clipboard=default(Alt+Ctrl+R)
Mouse Emulation=default(Alt+F12)
Next Taskbar Entry=none
Popup Launch Menu=none
Previous Taskbar Entry=none
Reboot without Confirmation=none
Run Command=none
Show Klipper Popup-Menu=default(Alt+Ctrl+V)
Show Taskmanager=default(Ctrl+Escape)
Show Window List=default(Alt+F5)
Switch One Desktop Down=none
Switch One Desktop Up=none
Switch One Desktop to the Left=none
Switch One Desktop to the Right=none
Switch to Next Desktop=none
Switch to Next Keyboard Layout=default(Alt+Ctrl+K)
Switch to Previous Desktop=none
Toggle Showing Desktop=default(Alt+Ctrl+D)
Toggle Window Raise/Lower=none
Walk Through Desktop List=default(Ctrl+Tab)
```

```
Walk Through Desktop List (Reverse)=default(Ctrl+Shift+Tab)
Walk Through Desktops=none
Walk Through Desktops (Reverse)=none
Walk Through Windows=default(Alt+Tab)
Walk Through Windows (Reverse)=default(Alt+Shift+Tab)
Window Above Other Windows=none
Window Below Other Windows=none
Window Close=default(Alt+F4)
Window Fullscreen=none
Window Grow Horizontal=none
Window Grow Vertical=none
Window Lower=none
Window Maximize=none
Window Maximize Horizontal=none
Window Maximize Vertical=none
Window Minimize=none
Window Move=none
Window No Border=none
Window On All Desktops=none
Window Pack Down=none
Window Pack Left=none
Window Pack Right=none
Window Pack Up=none
Window Raise=none
Window Resize=none
Window Screenshot=default(Alt+Print)
Window Shade=none
Window Shrink Horizontal=none
Window Shrink Vertical=none
Window to Desktop 1=none
Window to Desktop 10=none
Window to Desktop 11=none
Window to Desktop 12=none
Window to Desktop 13=none
Window to Desktop 14=none
Window to Desktop 15=none
Window to Desktop 16=none
Window to Desktop 17=none
Window to Desktop 18=none
Window to Desktop 19=none
Window to Desktop 2=none
Window to Desktop 20=none
Window to Desktop 3=none
Window to Desktop 4=none
Window to Desktop 5=none
Window to Desktop 6=none
Window to Desktop 7=none
Window to Desktop 8=none
Window to Desktop 9=none
Window to Next Desktop=none
Window to Previous Desktop=none

[Icons]
Theme=crystalsvg

[KDE]
EffectAnimateCombo=false
EffectAnimateMenu=true
EffectAnimateTooltip=false
EffectFadeMenu=true
EffectFadeTooltip=false
EffectNoTooltip=false
EffectsEnabled=true
InsertTearOffHandle=0
colorScheme=Keramik.kcsrc
contrast=7

[KFileDialog Settings]
Automatically select filename extension=true
Height 768=278
LocationCombo Completionmode=5
```

```
PathCombo Completionmode=5
Recent URLs=
Separate Directories=false
Show Preview=false
Show Speedbar=true
Show hidden files=false
Sort by=Name
Sort case insensitively=true
Sort directories first=true
Sort reversed=false
View Style=Simple
Width 1024=657

[Locale]
Country=us
Language=en_US
TimeFormat=%l:%M:%S %p

[MainToolbarIcons]
ActiveColor=169,156,255
ActiveEffect=none
ActiveSemiTransparent=false
ActiveValue=1
Animated=false
DefaultColor=144,128,248
DefaultEffect=none
DefaultSemiTransparent=false
DefaultValue=1
DisabledColor=34,202,0
DisabledEffect=togray
DisabledSemiTransparent=true
DisabledValue=1
DoublePixels=false
Size=22

[PanelIcons]
ActiveColor=169,156,255
ActiveEffect=togamma
ActiveSemiTransparent=false
ActiveValue=0.7
Animated=false
DefaultColor=144,128,248
DefaultEffect=none
DefaultSemiTransparent=false
DefaultValue=1
DisabledColor=34,202,0
DisabledEffect=togray
DisabledSemiTransparent=true
DisabledValue=1
DoublePixels=false

[Paths]
Trash=$HOME/Desktop/Trash/

[PreviewSettings]
BoostSize=false
MaximumSize=548576
UseFileThumbnails=false
ar=false
audiocd=false
camera=false
devices=false
file=false
fish=false
floppy=false
fonts=false
ftp=false
imap=false
imaps=false
lan=false
ldap=false
```

```
ldaps=false
mac=false
man=false
newimap=false
newimaps=false
nfs=false
nntp=false
pop3=false
pop3s=false
print=false
printdb=false
programs=false
rlan=false
settings=false
sftp=false
sieve=false
smb=false
smtp=false
smtps=false
start-here=false
system=false
tar=false
webdav=false
webdavs=false
zip=false

[SmallIcons]
ActiveColor=169,156,255
ActiveEffect=none
ActiveSemiTransparent=false
ActiveValue=1
Animated=false
DefaultColor=144,128,248
DefaultEffect=none
DefaultSemiTransparent=false
DefaultValue=1
DisabledColor=34,202,0
DisabledEffect=togray
DisabledSemiTransparent=true
DisabledValue=1
DoublePixels=false
Size=16

[Toolbar style]
Highlighting=true
IconText=IconTextBottom
TransparentMoving=true

[ToolbarIcons]
ActiveColor=169,156,255
ActiveEffect=none
ActiveSemiTransparent=false
ActiveValue=1
Animated=false
DefaultColor=144,128,248
DefaultEffect=none
DefaultSemiTransparent=false
DefaultValue=1
DisabledColor=34,202,0
DisabledEffect=togray
DisabledSemiTransparent=true
DisabledValue=1
DoublePixels=false
Size=22

[WM]
activeBackground=62,145,235
activeBlend=62,145,235
activeTitleBtnBg=220,220,220
frame=234,233,232
handle=234,233,232
```

```
inactiveBackground=175,214,255
inactiveBlend=175,214,255
inactiveForeground=255,255,255
inactiveFrame=234,233,232
inactiveHandle=234,233,232
inactiveTitleBtnBg=220,220,220

[KDE Action Restrictions]
action/kdesktop_rmb=false
action/kicker_rmb=false
editable_desktop_icons=false
editable_system_desktop_icons=false
action/movable_toolbars=false
shell_access=false
icon=false
action/run_command=false
action/lock_screen=false
action/custom_config=false
action/run_desktop_files=false
```

Default Login Services

We use some custom services and applications during the login and logout process. One of these is the IsInUse service. In order for us to track login/logout times for our users, we will want to make this available during login.

```
cd /etc/X11/xinit/xinitrc.d
vi isinuse.start
```

Make this file look like the following:

```
#!/bin/bash
#
#Logs user into IsInUse workstation tracking system
#
#E3-script
#June 19, 2003 DPB Initial Version
#
#
#check for user id to see if not root
if [ $UID -gt 0 ] ; then \
  inuseclient 149.169.192.249 1299 $(hostname | cut -f 1 -d
. ) $UID 1
fi
#
#this sends to the isinuse server the hostname, userid, and
#the value 1 for checkin
```

Now we need to change the permissions on the file…

```
chmod 755 isinuse.start
```

Next is the script to clear the desktop.

```
vi nodesktop.sh
```

```
#!/bin/sh
chmod 400 ${$HOME}/Desktop
```

```
chmod 755 nodesktop.sh
```

When someone logs off, we use the Xreset script to put things back to normal.

```
cd /etc/kde/kdm
vi Xreset
```

```
#! /bin/sh
#Xreset - run as root after session exits
#E3-script /etc/kde/kdm/Xreset
#
# Reassign ownership of the console to root, this should
disallow
# assignment of console output to any random users xterm
# This is not required if you use PAM, as pam_console
should handle it

#clear out HOME and TMP directories - DPB April 18, 2003
rm -rf /home/* &> /dev/null
/usr/local/bin/cleantmp &> /dev/null

#kill WINE servers
killall wineserver

#check reboot/shutdown-on-logout flag to see if we need to
reboot
if [ -f /.reboot-on-logout ] ; then \
     reboot
fi

if [ -f /.shutdown-on-logout] ; then \
     poweroff
fi
```

```
chmod 755 *
chmod 444 Xresources
chmod 644 xdm-config
cp –f Xreset /etc/kde/kdm
```

kdm

Now we need to configure the KDE desktop manager.

```
cd /etc/X11/xdm
vi kdmrc
```

```
[General]
ConfigVersion=2.1
PidFile=/var/run/kdm.pid
Xservers=/usr/share/config/kdm/Xservers

[Shutdown]
HaltCmd=/sbin/poweroff
RebootCmd=/sbin/reboot
UseLilo=false

[X-*-Core]
AllowNullPasswd=true
AllowRootLogin=true
AllowShutdown=Root
AutoReLogin=false
Reset=/usr/share/config/kdm/Xreset
Resources=/usr/share/config/kdm/Xresources
Session=/usr/share/config/kdm/Xsession
SessionsDirs=/etc/X11/sessions,/usr/share/xsessions,/usr/share/apps/kdm
/sessions
Setup=/usr/share/config/kdm/Xsetup
Startup=/usr/share/config/kdm/Xstartup

[X-*-Greeter]
AntiAliasing=true
ColorScheme=Keramik
DefaultUser=pac
EchoMode=OneStar
FaceSource=AdminOnly
FailFont=Sans,18,-1,5,75,0,0,0,0,0
#E3-script change 12-2004
#change FocusPassword to false
FocusPasswd=false
GUIStyle=Keramik
GreetFont=Sans,14,-1,5,75,0,0,0,0,0
#E3- modified GreetString - PCH 12-2004
GreetString=ASU Students, login with ASURITE ID and password.\nGuest
users, login with '%n' for both username and password.
GreeterPos=50,50
HiddenUsers=
Language=en_US
LogoArea=Logo
LogoPixmap=/e3custom/usr/local/e3/wallpaper/loginpic.jpg
MaxShowUID=65000
MinShowUID=500
PreselectUser=None
SelectedUsers=
ShowUsers=NotHidden
SortUsers=true
StdFont=Sans,14,-1,5,50,0,0,0,0,0
UseBackground=true
UserCompletion=false
```

```
UserList=false
#E3- add numlock on -PCH 12-2004
NumLock=On

[X-:*-Core]
AllowNullPasswd=true
AllowRootLogin=true
AllowShutdown=Root
NoPassEnable=false
NoPassUsers=

[X-:*-Greeter]
#LoginMode=DefaultLocal

[X-:0-Core]
#AutoLoginEnable=false
#AutoLoginUser=pac

#E3-script change 12-2004
#make Xdmcp Enabled=false
[Xdmcp]
Enable=false
Willing=/usr/share/config/kdm/Xwilling
Xaccess=/usr/share/config/kdm/Xaccess
```

```
cd /etc/kde/kdm
vi backgroundrc
```

```
[Desktop0]
BackgroundMode=Flat
BlendBalance=100
BlendMode=NoBlending
ChangeInterval=60
Color1=138,148,198
Color2=104,112,150
CurrentWallpaper=0
LastChange=0
MinOptimizationDepth=1
MultiWallpaperMode=NoMulti
Pattern=
Program=
ReverseBlending=false
UseSHM=false
Wallpaper=/usr/local/e3/wallpaper/desktop.jpg
WallpaperList=
WallpaperMode=Tiled
```

```
chmod 644 backgroundrc
```

Now we need to edit what happens when X starts up…

```
vi Xstartup
```

```
#! /bin/sh
# Xstartup - run as root before session starts

PIDFILE=/var/run/kdmdesktop-$DISPLAY.pid
if [ -f $PIDFILE ] ; then
    kill `cat $PIDFILE`
fi
#exit 0

# By convention, both xconsole and xterm -C check that the
# console is owned by the invoking user and is readable
# before attaching
# the console output.  This way a random user can invoke
# xterm -C without causing serious grief.
# This is not required if you use PAM, as pam_console
# should handle it.
#
#chown $USER /dev/console

#exec sessreg -a -l $DISPLAY $USER

# E3-script addition
# we need to create a symbolic link to our public machine
# user's home directory for mounting
ln -s /home/$USER /home/pacsym
```

Konqueror Profile

When Konqueror launches, it can be set to a default format for display. We will enable this by doing the following:

```
cd /usr/share/apps/konqueror/profiles
rm –f *
vi pac.profile
```

```
#E3-script for duel pane filemanager with removable drives
[Main Window Settings Toolbar bookmarkToolBar]
Hidden=true
Index=3

[Main Window Settings Toolbar extraToolBar]
Hidden=true
Index=0

[Main Window Settings Toolbar locationToolBar]
Hidden=true
Index=2

[Main Window Settings Toolbar mainToolBar]
Hidden=false
```

```
Index=1

[Profile]
Container0_Children=View1,View2
Container0_Orientation=Horizontal
Container0_SplitterSizes=142,557
Container0_activeChildIndex=1
Container0_docContainer=true
FullScreen=false
Height=503
Name=pac.profile
RootItem=Container0
View1_LinkedView=false
View1_LockedLocation=false
View1_PassiveMode=false
View1_ServiceName=konq_iconview
View1_ServiceType=inode/directory
View1_ToggleView=false
View1_URL=file:$HOME/.kde/altdesktop
View2_LinkedView=false
View2_LockedLocation=false
View2_PassiveMode=false
View2_ServiceName=konq_iconview
View2_ServiceType=inode/directory
View2_ToggleView=false
View2_URL=file:$HOME
Width=705
```

```
chmod 644 pac.profile
```

To use Konqueror for viewing, now that we have a custom profile, we need to create the directory for the second pane.

```
mkdir /etc/skel/.kde/altdesktop
```

Copy over or create all removable media `.desktop` files to the `altdesktop` directory. Be sure to remove the `.desktop` extension in the filename or else it will get displayed in the Konqueror interface with that label. So if you want the floppy disk to show up, name the `floppy.desktop` file as `floppy`. Konqueror will know that it is a `.desktop` file even without the extension and use it as such.

Samba SMB Configuration

Since we do not want users to map to smb shares (they could then execute arbitrary code), we will edit the `pam_smb.conf` file and remove all entries.

```
vi /etc/pam_smb.conf
```

Building the Image

Init Scripts
When the client machine boots, it will be served an image that has been locked down by disabling and customizing various aspects of `/etc/rc.d` files. We are now going to create the necessary init files for our custom image. These will be integrated during the build process.

Before we begin, we need to stop a few services that are not needed:

```
chkconfig senmail off
chkconfig crond off
chkconfig anacron off
```

Since our boot image will be served via nfs, we need to enable the nfs protocol:

```
chkconfig nfs on
```

`rc.d`
Now we can create our rc.d directory and files:

```
cd /e3custom/etc
cp –R /etc/rc.d .
```

We now need to create some symbolic links:

```
ln –s rc.d/init.d init.d
ln –s rc.d/rc rc

ln –s rc.d/rc0.d rc0.d
ln –s rc.d/rc1.d rc1.d
ln –s rc.d/rc2.d rc2.d
ln –s rc.d/rc3.d rc3.d
ln –s rc.d/rc4.d rc4.d
ln –s rc.d/rc5.d rc5.d
ln –s rc.d/rc6.d rc6.d

ln –s rc.d/rc.local rc.local
ln –s rc.d/rc.sysinit rc.sysinit
```

We need to create our custom `rc.local` file:

```
cd /e3custom/etc/rc.d
vi rc.local
```

Make it look like this:

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
#umount /mnt/nfs
#rmdir /mnt/nfs

# Since we may or may not shut down properly... - DPB
hwclock --systohc

# for lm_sensors - DPB - Oct 15 2003
# I2C adapter drivers
modprobe i2c-isa
# I2C chip drivers
modprobe it87
modprobe i2c-proc
# end lm_sensors

# Enable Wake on LAN on the next startup
/sbin/ethtool -s eth0 wol g

# Setup AFS space to not honor special files
#mount -o remount,noexec,nosuid,nodev /afs

# Close orphan isinuse sessions
/usr/local/bin/inuseclient xxx.xxx.xxx.xxx 1299 $(hostname | cut -f 1 -d . ) 0 2


# Setup hosts file so that LPRng works correctly
echo "127.0.0.1               localhost.localdomain localhost" > /etc/hosts
echo "xxx.xxx.xxx.xxx         client-test.xxx.xxx.xxx client-test-station" >> /etc/hosts
```

Now we need to change the permissions for these:

```
chmod 666 fstab
chmod –R 755 *
```

Remove Unneeded Files

There are many files on our build station that are not needed. We can trim these a couple ways. One is to physically remove them. The other is to exclude them in the image build script. Let's start by physically removing some files we know are not needed by our build station nor the client image.

Picture Files

```
cd /usr/share/apps/kdm/pics
rm –rf *
```

Splash Screens

Now we need to remove the splash screens we will not use.

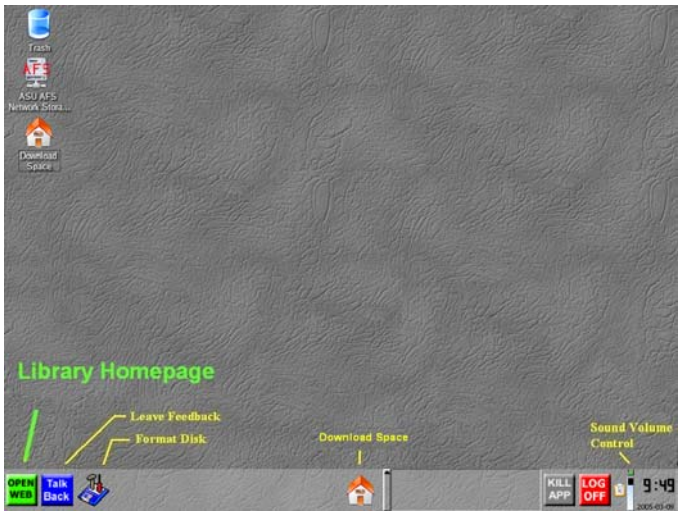
```
cd /usr/share/apps/ksplash/Themes
```

```
rm –rf blue-bend
rm –rf BlueCurve
rm –rf Redmond
rm –rf Standard

cd /

rm –rf /usr/share/rhn
rm –rf /usr/share/themes

cd /usr/share/apps/kstyle/themes
remove all themes but Keramik.themerc

cd /usr/share
rm –rf wallpapers
rm –rf sounds
rm –rf pixmaps
rm –rf xscreensaver
rm –rf libgphoto2
rm –rf backgrounds

cd /usr/share/pilot-link
rm –rf pix

cd /usr/share/xmms/skins
rm –rf *
```

Finished Product

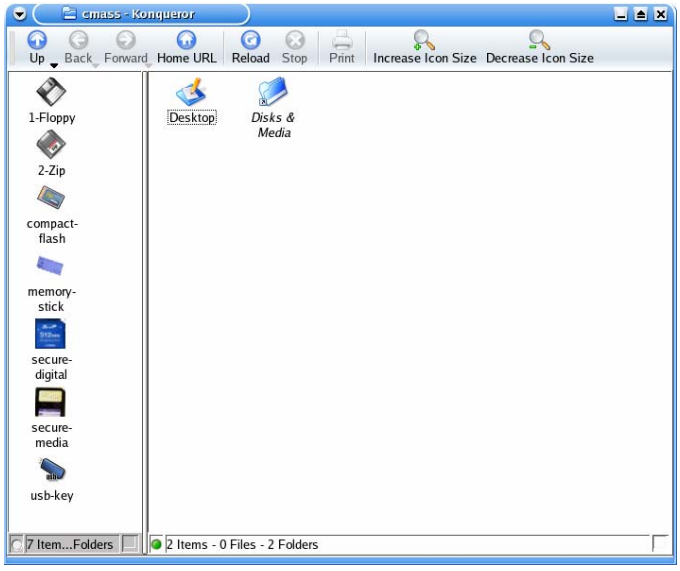
Desktop

When a user logs on to our workstation they are presented with a desktop that includes our custom background screen which describes the various buttons available. The desktop displays the trashcan, network file drive, and home directory. The Kicker Bar provides access to the Web, feedback agent, disk tools, home directory, alert ticker, Windows killer, logout , clipboard, volume control, and clock.



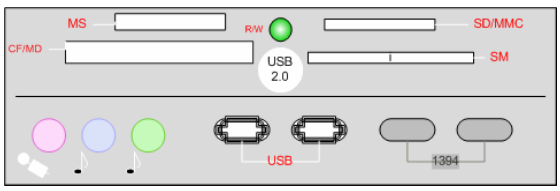
Home/Download Space

The Home directory contains a left menu displaying all available media drives that can be accessed. The main menu display contains readable and writable, but not executable 60MB of temporary storage space to save and move files. It also includes a symbolic directory link to the removable media drives so a user can drag and drop between the left menu and right main menu a file from one media device to the other.



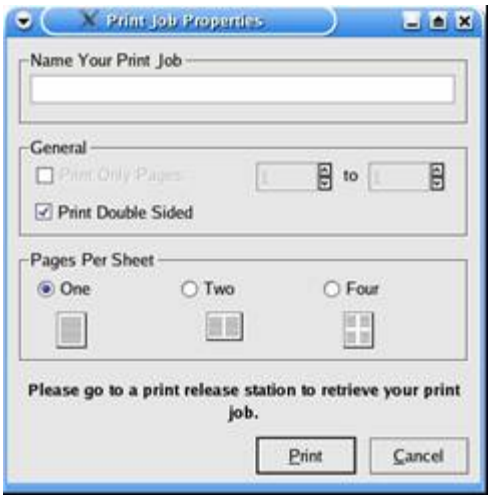
Removable Media Devices

Except for the floppy and Zip drives, all other removable media are available using a built-in media interface in the physical workstation case.



Printing

Printing is facilitated through our custom KDE print dialog to allow naming of the print job for the Pharos UniPrint system.

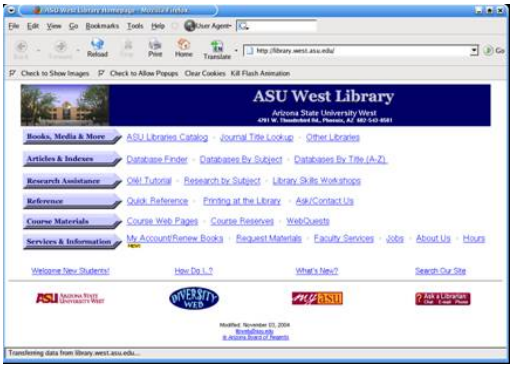


Disk Formatting

Disk formatting, or in our case disk erasing, is performed by our custom disk formatter application. It takes a floppy or Zip disk and erases all files and directories on it while preserving the VFAT filesystem.

Web Browser

The Firefox Web browser has been customized to meet our needs and provides our users with ease in access to information and additional tools to aid in their processing of information.



Logout Screensaver
After 10 minutes of inactivity from the mouse and keyboard, our custom logout screensaver kicks in with a countdown warning.

Kicker Bar Alert RSS
Custom RSS newsfeed using Knewsticker on the Kicker Bar alerts users to problems or special timely announcements.

Windows Killer
Custom application that terminates stalled Windows reader applications if they freeze up (due to a file format or versioning issue which WINE or the reader application cannot handle.)