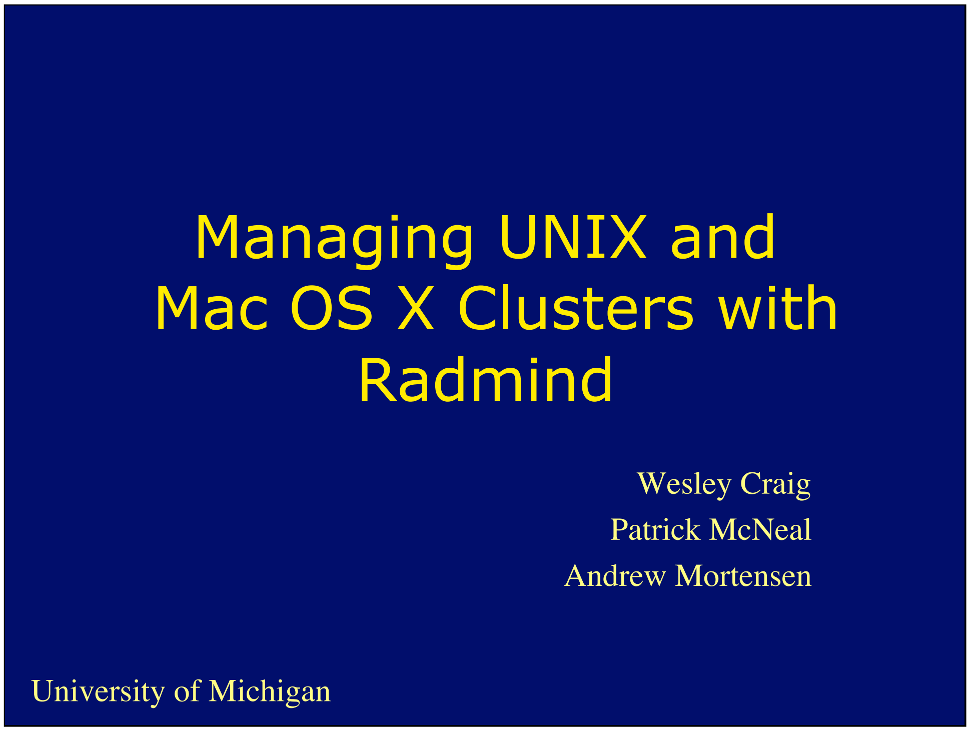
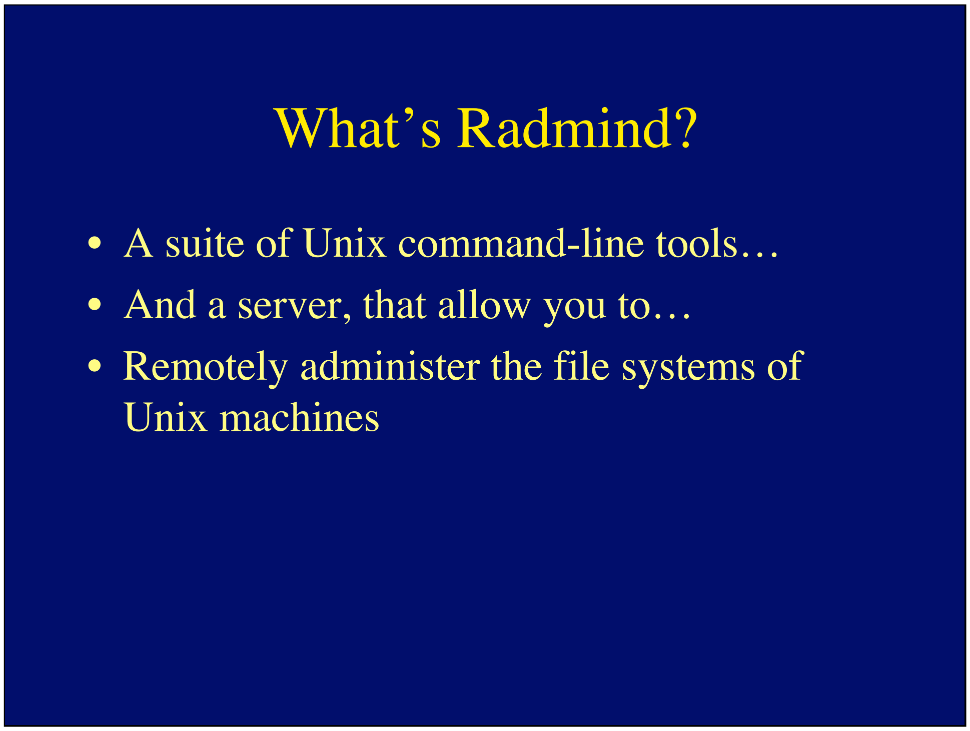


Managing UNIX and Mac OS X Clusters with Radmind

Wesley Craig
Patrick McNeal
Andrew Mortensen

University of Michigan

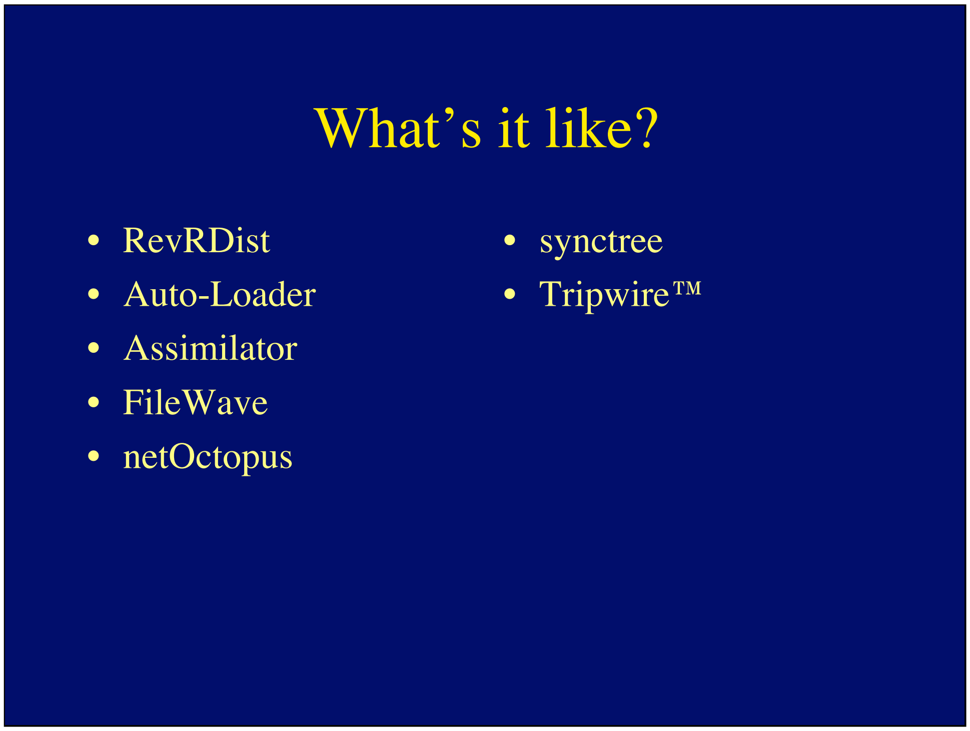
Research Systems Unix Group of the University of Michigan



What’s Radmind?

- A suite of Unix command-line tools…
- And a server, that allow you to…
- Remotely administer the file systems of Unix machines

“radmind” is the name of both the server and the package as a whole. While it stands for “remote administration daemon”, it’s pronounced “rad-mind”. Developed by members of the Research Systems Unix Group at the University of Michigan. Freely available under a BSD-style license.



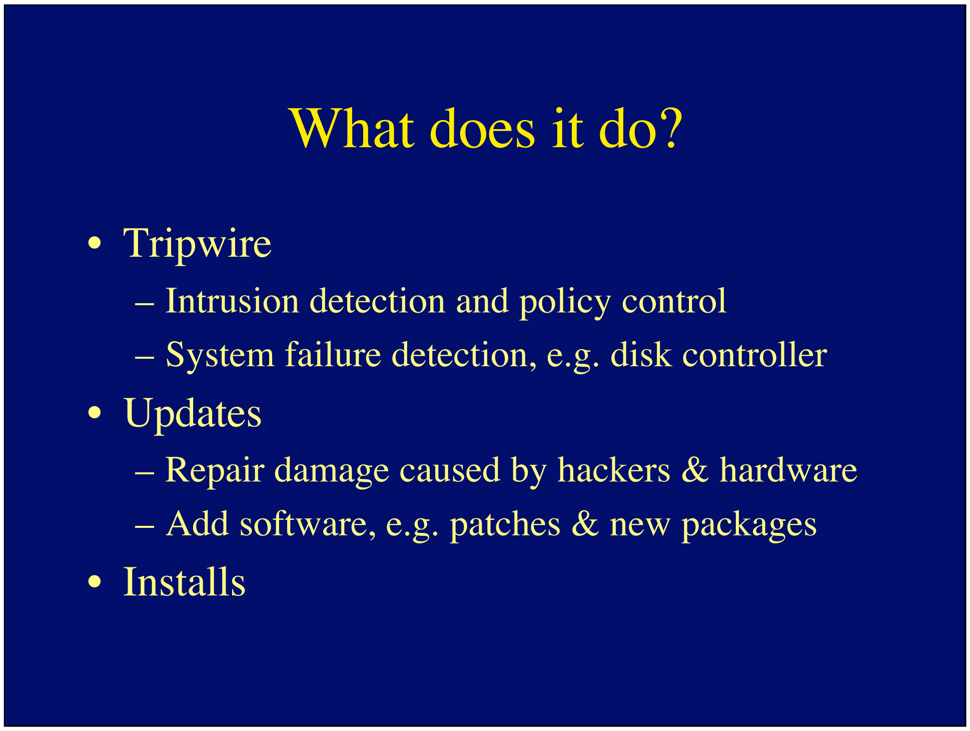
What’s it like?

- RevRDist
- Auto-Loader
- Assimilator
- FileWave
- netOctopus
- synctree
- Tripwire™

The packages in the left column are all for the Mac. RevRDist was developed at Purdue and may be updated for Mac OS X. The author is working on HFS+ support for rdist. Once that work is completed, he will consider whether a Mac OS X version of RevRDist is appropriate. Nothing is planned before Fall of 2002. Auto-Loader was developed at the University of Michigan College of Engineering. The other three are commercial. Assimilator is no longer under development. FileWave and netOctopus have both been ported to Mac OS X.

The packages in the right column are for Unix. Synctree was developed at the University of Michigan College of Engineering. Tripwire™ was developed by Dr. Eugene Spafford and Gene Kim.

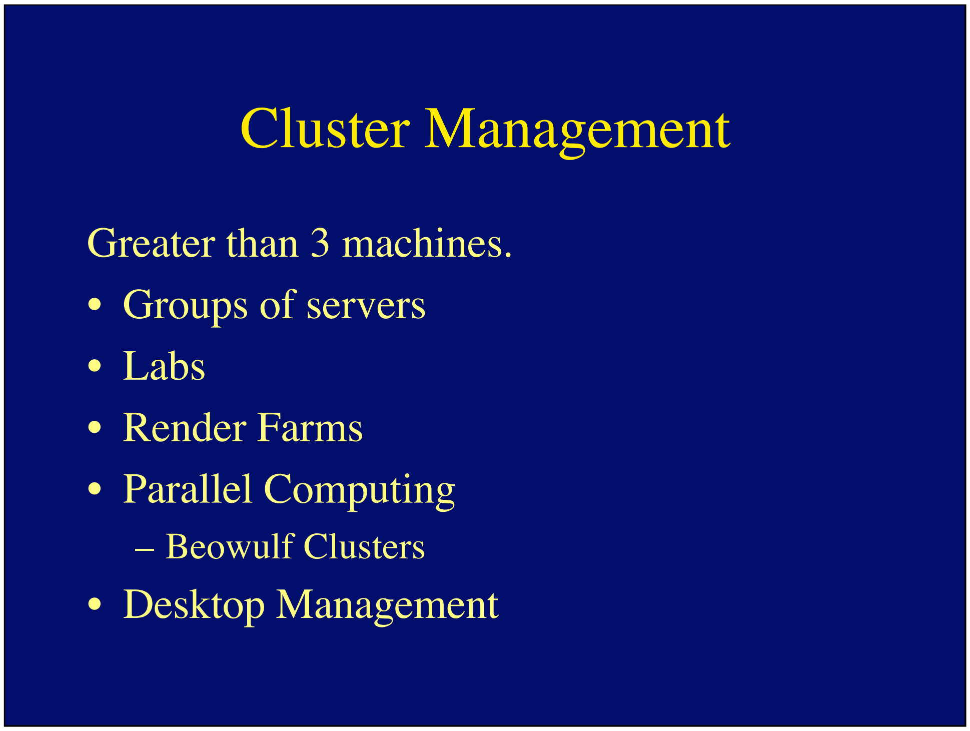
The design of radmind is “inspired” by both Auto-Loader and synctree, though the technology is very similar to Tripwire™ with the addition of management.



What does it do?

- Tripwire
  - Intrusion detection and policy control
  - System failure detection, e.g. disk controller
- Updates
  - Repair damage caused by hackers & hardware
  - Add software, e.g. patches & new packages
- Installs

On a fully managed filesystem, radmind can be used to do a full install. We use radmind this way for our Solaris servers. We have not (yet) developed a fully managed loadset for Mac OS X.

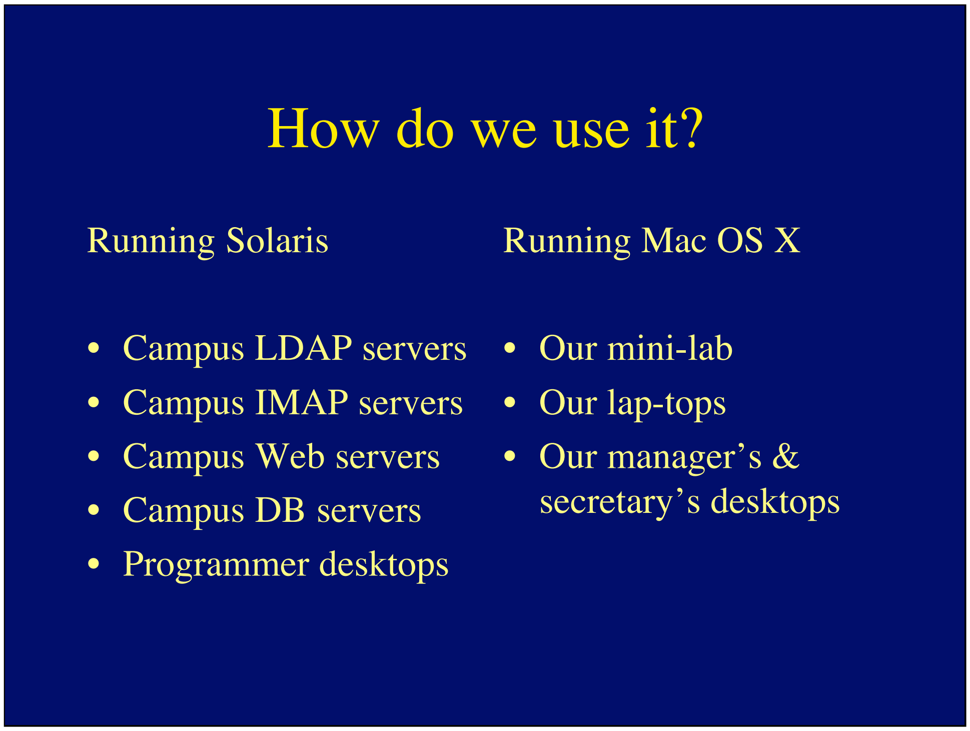


Cluster Management

Greater than 3 machines.

- Groups of servers
- Labs
- Render Farms
- Parallel Computing
  - Beowulf Clusters
- Desktop Management

Running three machines manually is about equivalent to running one server, one client, and radmind loadsets from the client. Unlike “imaging” software, once a base loadset has been created, the incremental cost of running many machines that are mostly similar is low.



How do we use it?

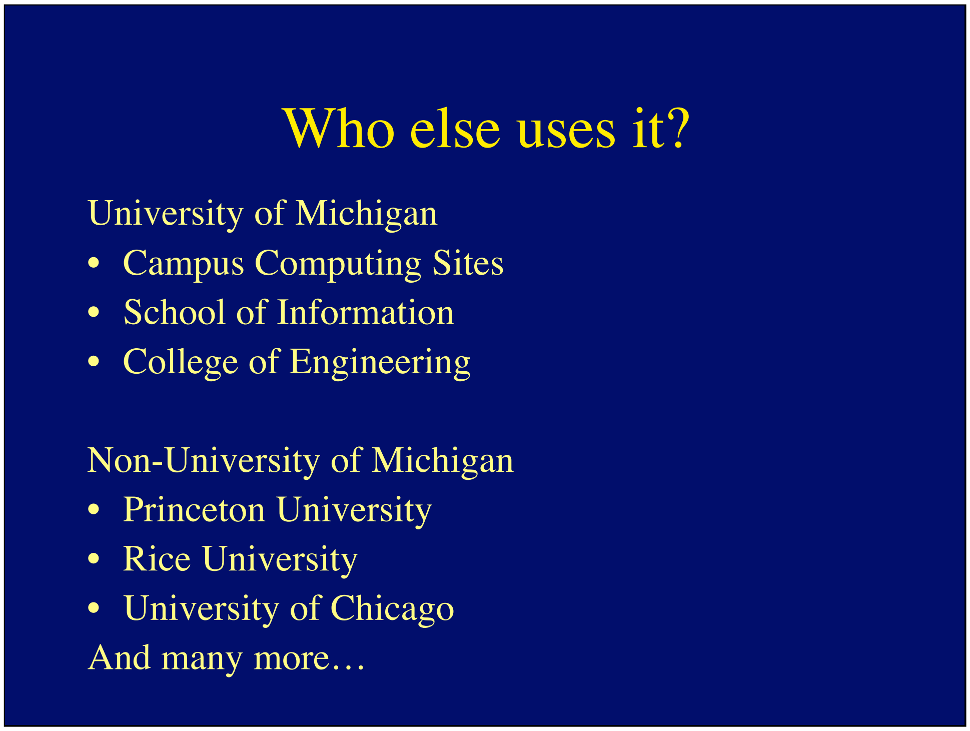
Running Solaris

- Campus LDAP servers
- Campus IMAP servers
- Campus Web servers
- Campus DB servers
- Programmer desktops

Running Mac OS X

- Our mini-lab
- Our lap-tops
- Our manager’s & secretary’s desktops

Under Mac OS X, one loadset works for all of these machines. This is not the case with Solaris.



Who else uses it?

University of Michigan

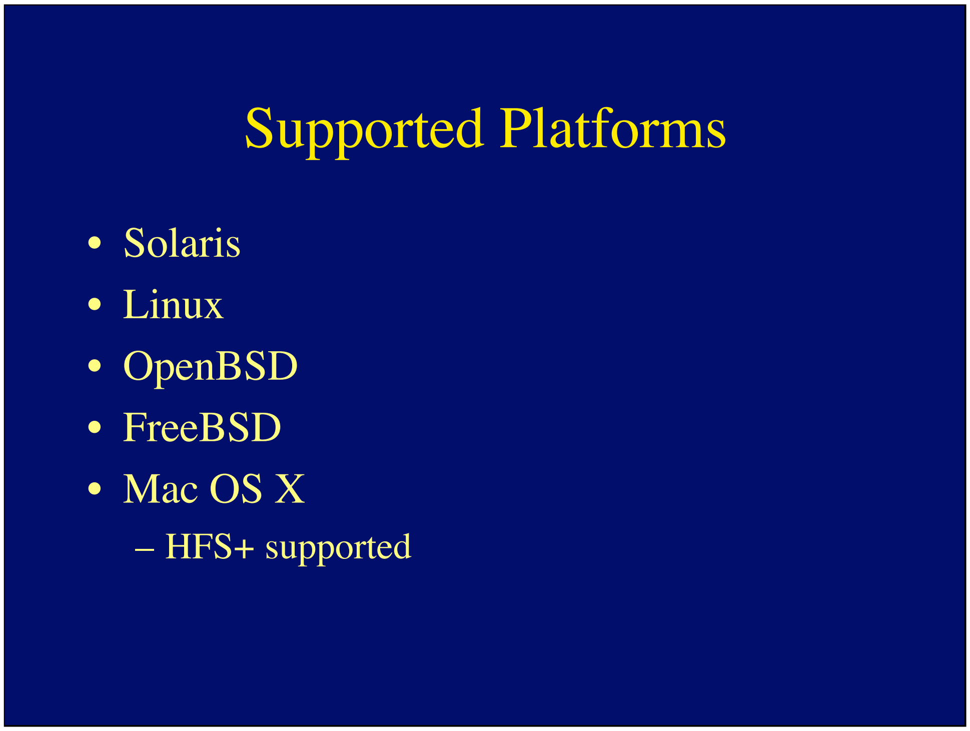
- Campus Computing Sites
- School of Information
- College of Engineering

Non-University of Michigan

- Princeton University
- Rice University
- University of Chicago

And many more…

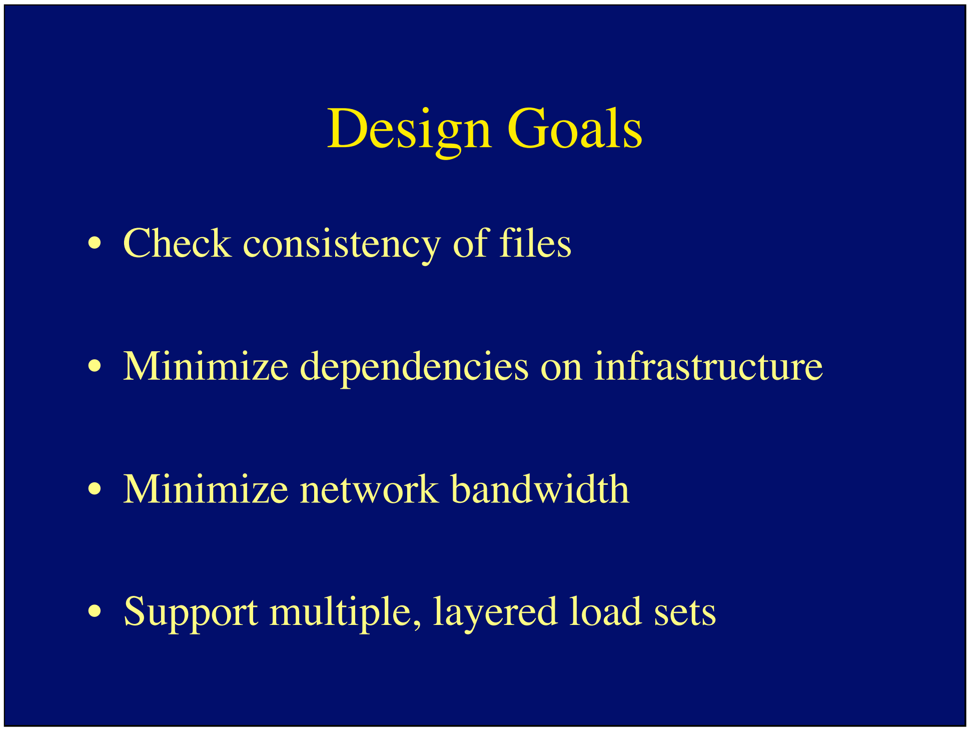
Disney and Apple have also indicated interest.



Supported Platforms

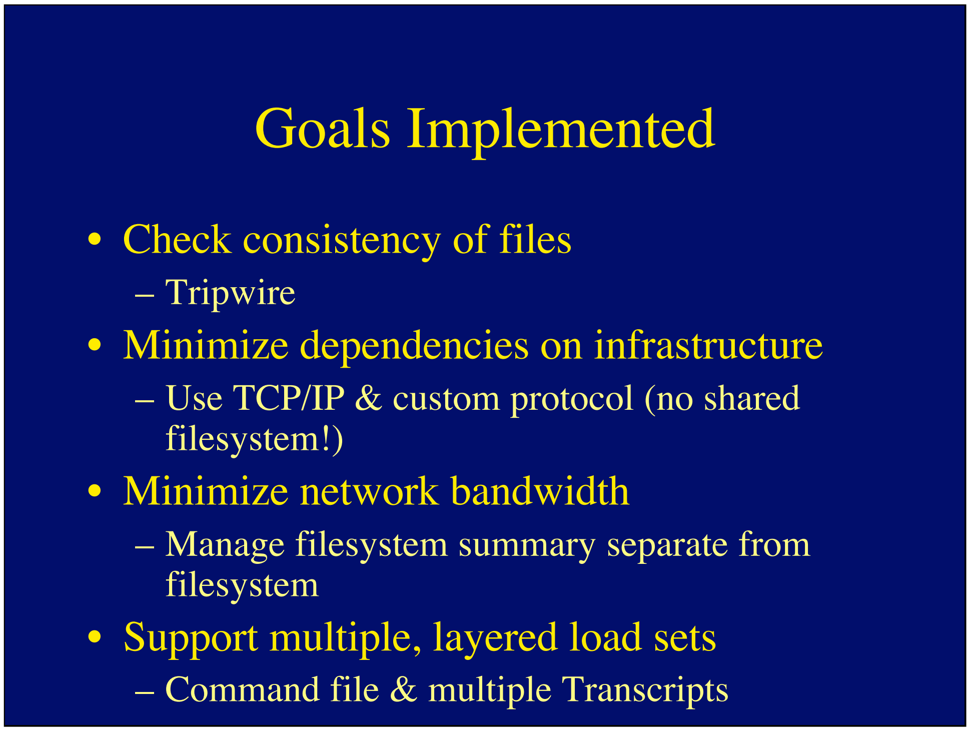
- Solaris
- Linux
- OpenBSD
- FreeBSD
- Mac OS X
  - HFS+ supported

The developers are actively using radmind to manage Solaris and Mac OS X machines. Other UMich people are managing Linux machines. It compiles and seems to run on *BSD, but we’re not aware of anyone using it on these platforms in production.



Design Goals

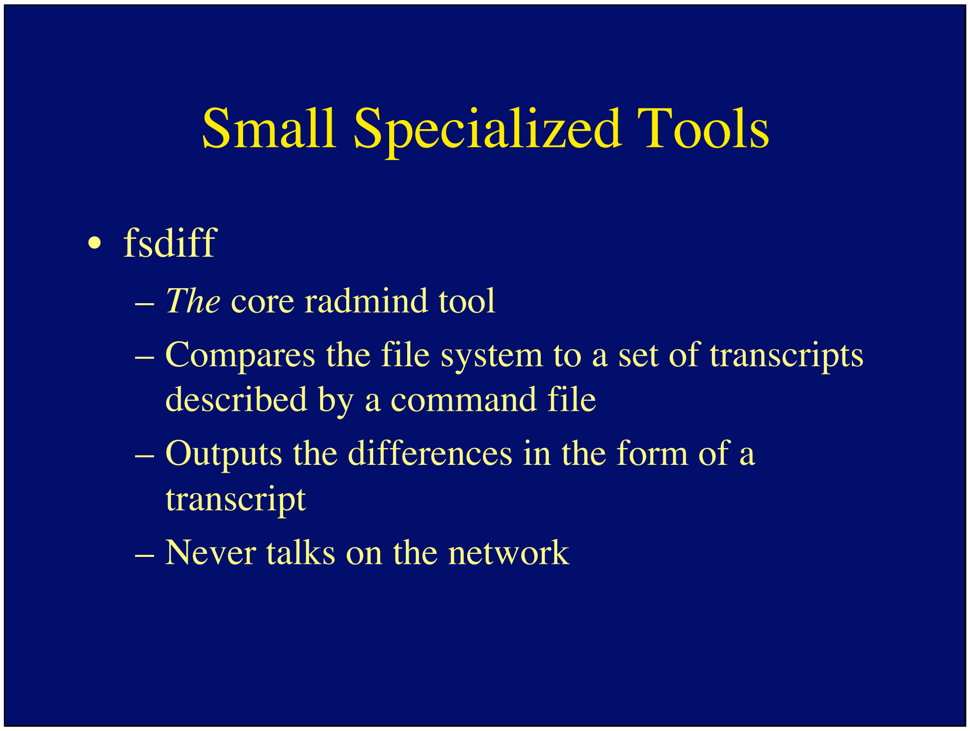
- Check consistency of files
- Minimize dependencies on infrastructure
- Minimize network bandwidth
- Support multiple, layered load sets



Goals Implemented

- Check consistency of files
  - Tripwire
- Minimize dependencies on infrastructure
  - Use TCP/IP & custom protocol (no shared filesystem!)
- Minimize network bandwidth
  - Manage filesystem summary separate from filesystem
- Support multiple, layered load sets
  - Command file & multiple Transcripts

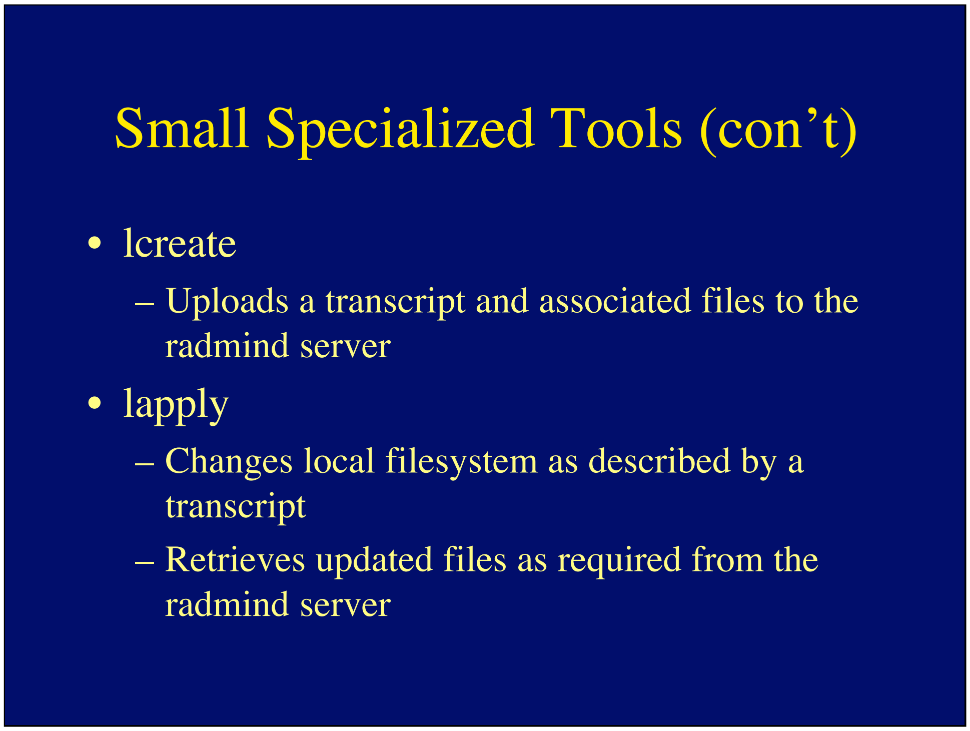
As noted earlier, radmind is “inspired” by Auto-Loader and synctree. Auto-loader doesn’t check the contents of files, only time stamps and size. Both Auto-Loader and synctree use a shared filesystem rather than a server. Auto-Loader does uses direct access to the shared filesystem, rather than filesystem summaries. Both Auto-Loader and synctree allow multiple, layered load sets.



Small Specialized Tools

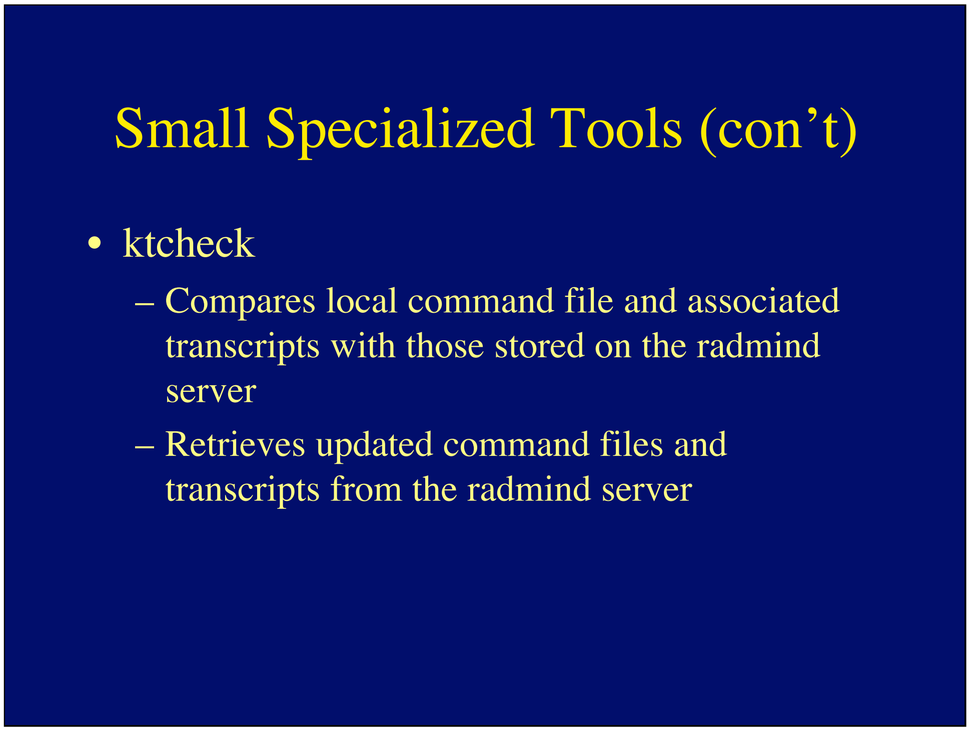
- fsdiff
  - *The* core radmind tool
  - Compares the file system to a set of transcripts described by a command file
  - Outputs the differences in the form of a transcript
  - Never talks on the network

A “transcript” is a list of filesystem objects, e.g. files, directory, links, and their attributes. A “command” file contains a list of transcripts in order of precedence.



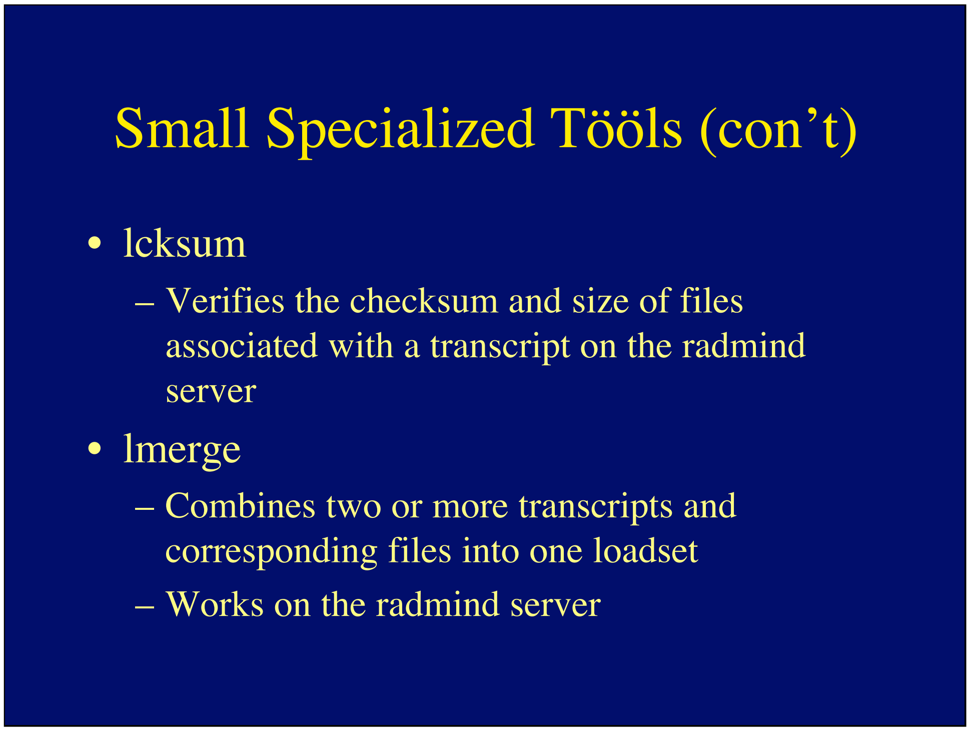
Small Specialized Tools (con’t)

- lcreate
  - Uploads a transcript and associated files to the radmind server
- lapply
  - Changes local filesystem as described by a transcript
  - Retrieves updated files as required from the radmind server



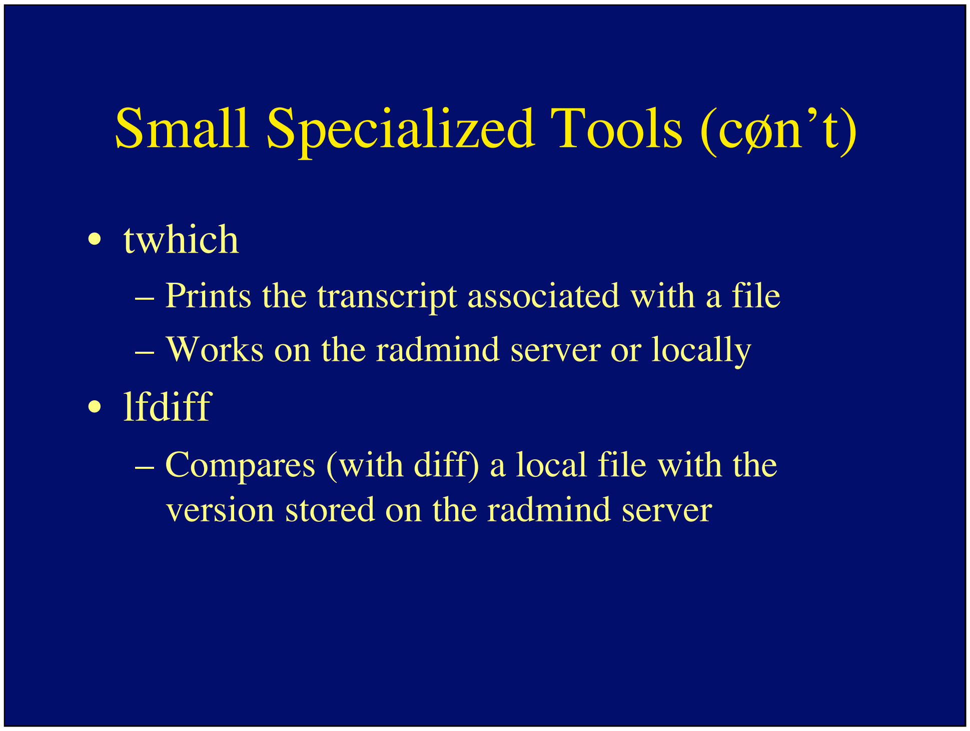
Small Specialized Tools (con’t)

- ktcheck
  - Compares local command file and associated transcripts with those stored on the radmind server
  - Retrieves updated command files and transcripts from the radmind server



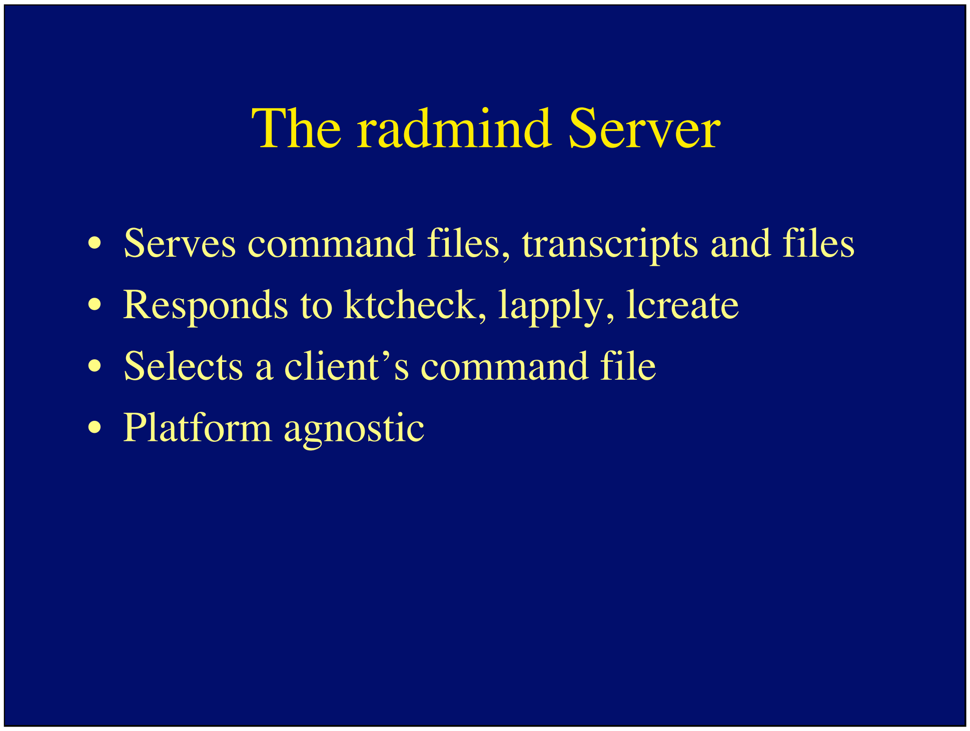
Small Specialized Tööls (con’t)

- lcksum
  - Verifies the checksum and size of files associated with a transcript on the radmind server
- lmerge
  - Combines two or more transcripts and corresponding files into one loadset
  - Works on the radmind server



Small Specialized Tools (cøn’t)

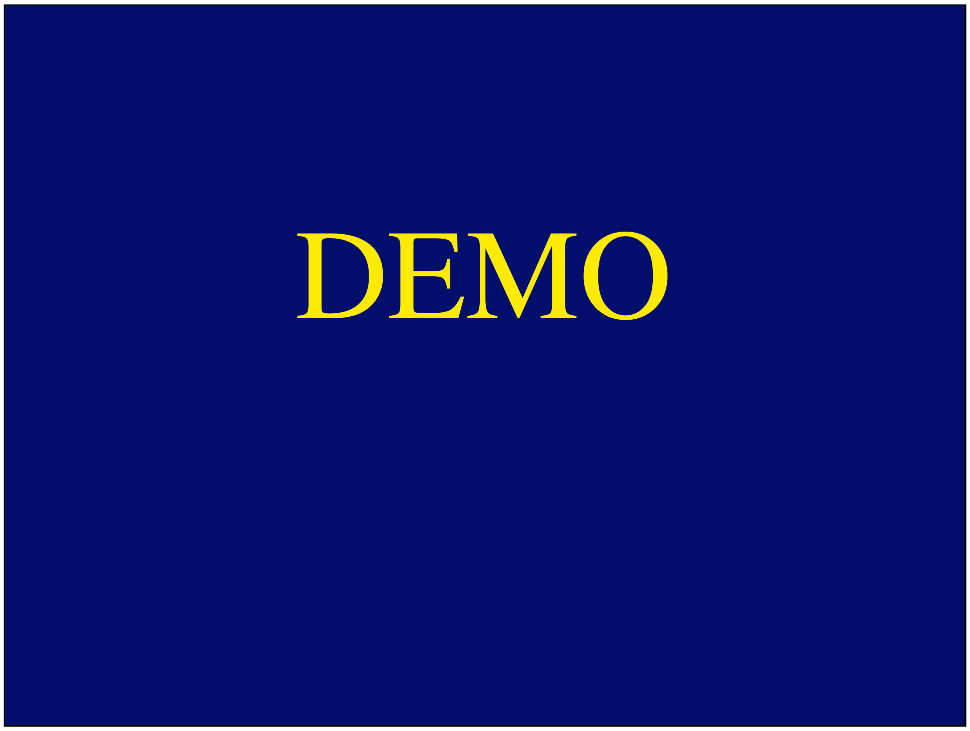
- twhich
  - Prints the transcript associated with a file
  - Works on the radmind server or locally
- lfdiff
  - Compares (with diff) a local file with the version stored on the radmind server



The radmind Server

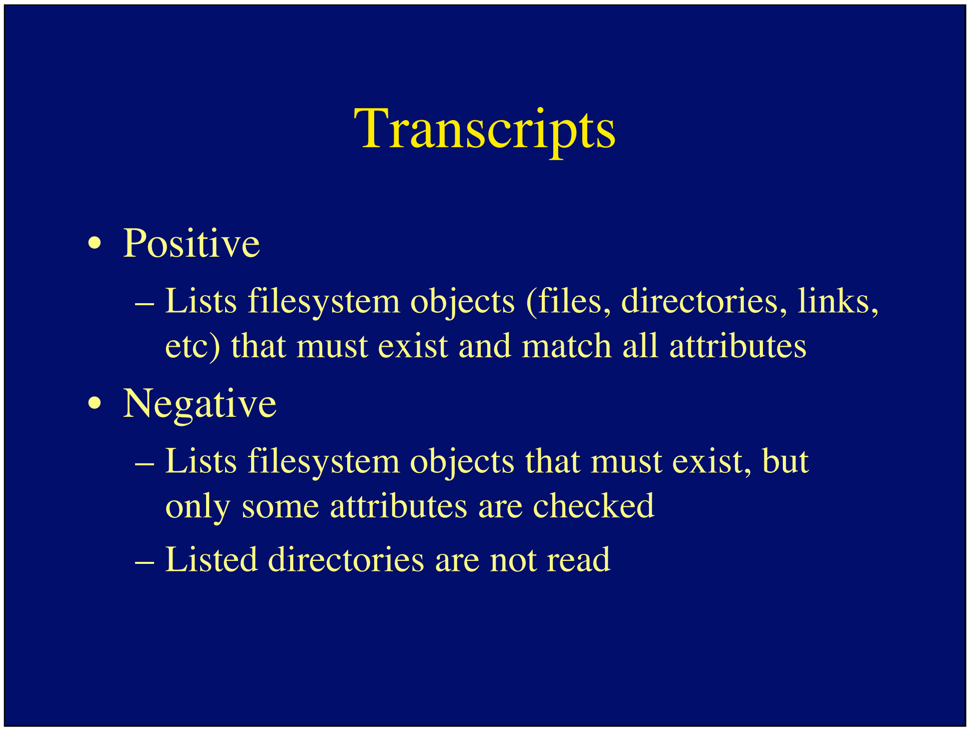
- Serves command files, transcripts and files
- Responds to ktcheck, lapply, lcreate
- Selects a client’s command file
- Platform agnostic

Esoteric file types on Mac OS X are stored as AppleSingle encoded files on the server.



DEMO

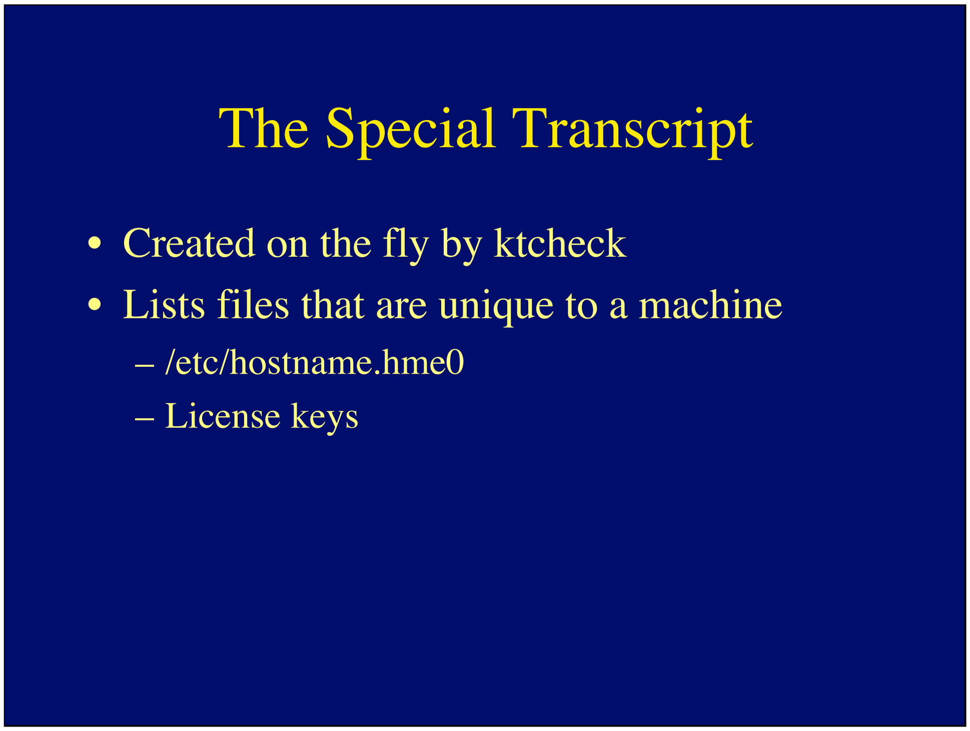
Patrick begins with a fully up-to-date machine. As administrator, he removes the Chess application, the transcripts, and the command file. He then runs ktcheck to get updated versions of the transcripts and command file. He then runs fsdiff -A to produce a transcript which, when run through lapply, downloads and installs all of the components of the Chess application. He run the Chess application.



Transcripts

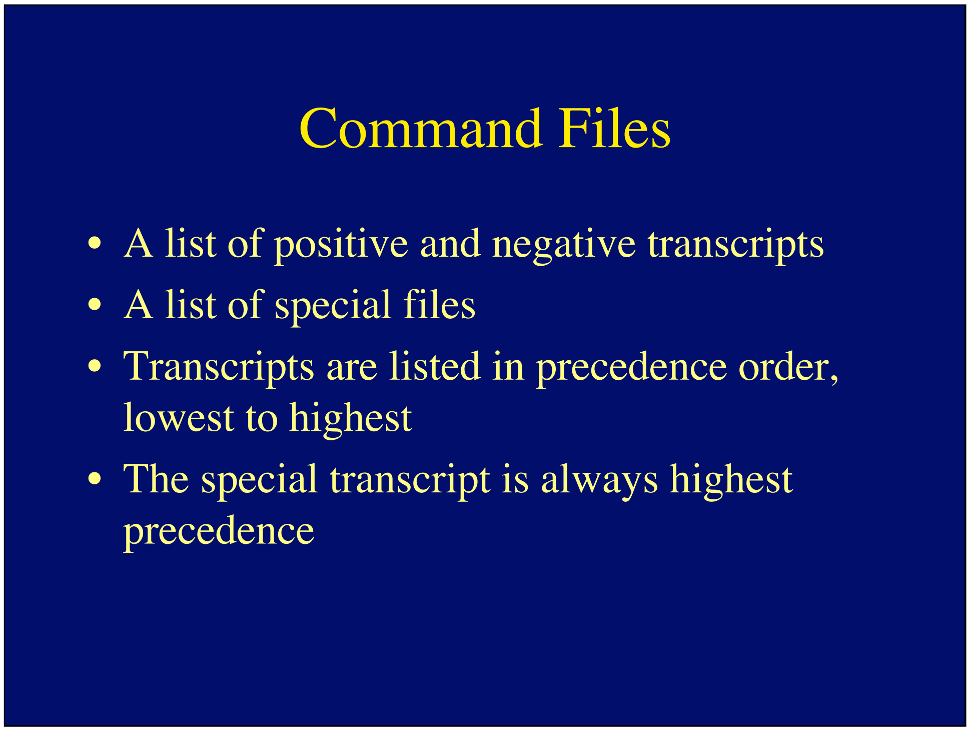
- Positive
  - Lists filesystem objects (files, directories, links, etc) that must exist and match all attributes
- Negative
  - Lists filesystem objects that must exist, but only some attributes are checked
  - Listed directories are not read

Prefixing a transcript line with the minus sign ‘-’ means that the filesystem object must not exist.



The Special Transcript

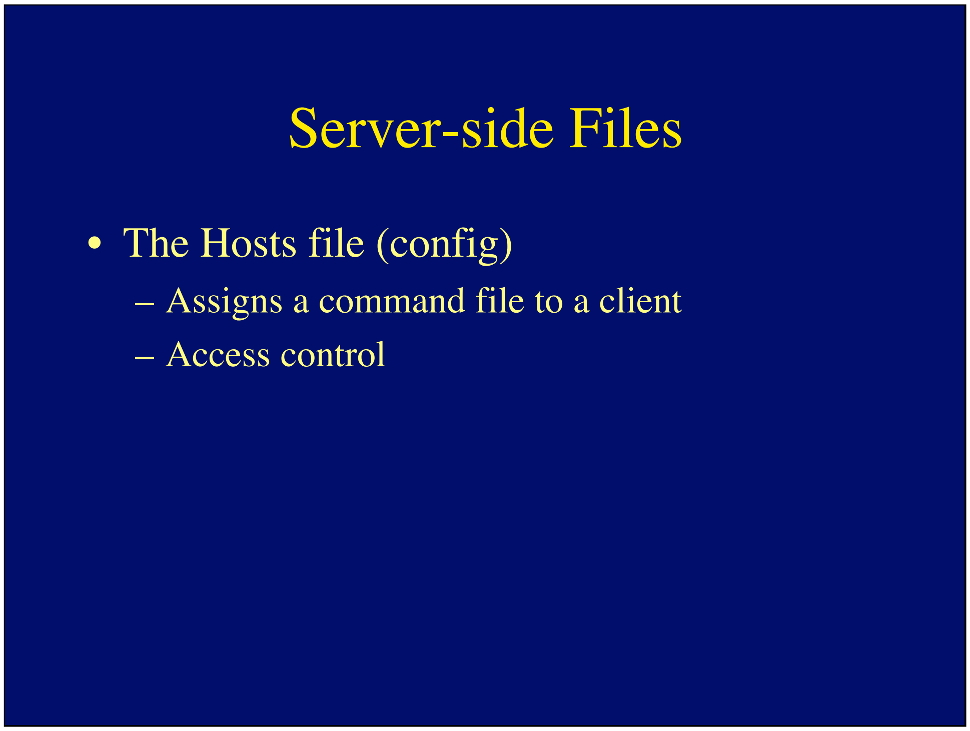
- Created on the fly by ktcheck
- Lists files that are unique to a machine
  - /etc/hostname.hme0
  - License keys



Command Files

- A list of positive and negative transcripts
- A list of special files
- Transcripts are listed in precedence order, lowest to highest
- The special transcript is always highest precedence

On the client machine, the command file is always known as command.K while on the server, command files have arbitrary names.



Server-side Files

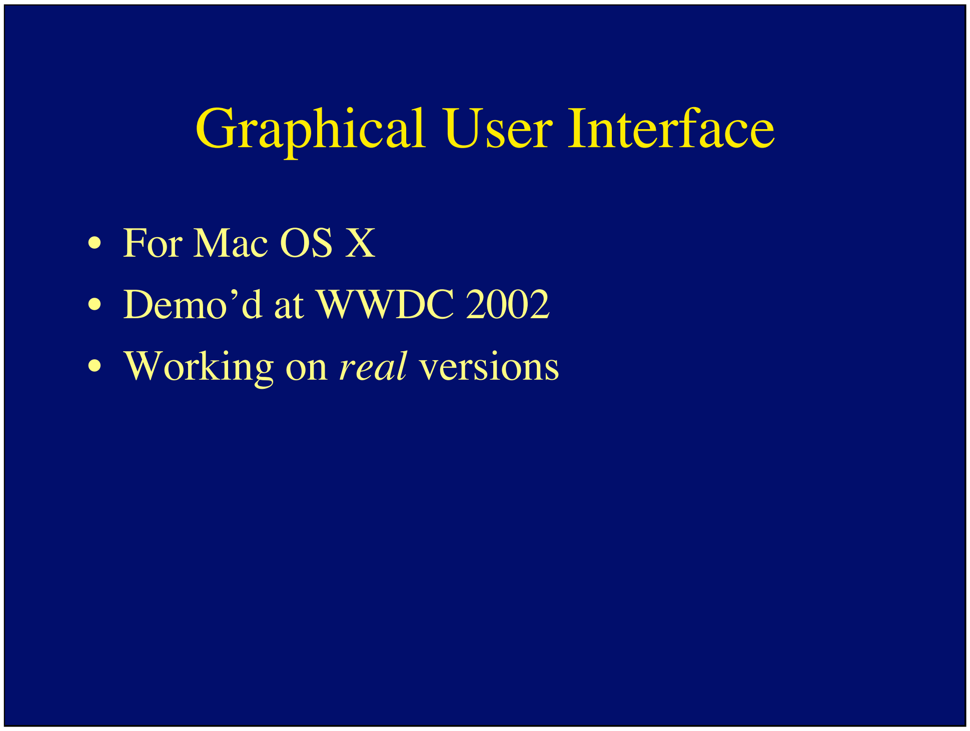
- The Hosts file (config)
  - Assigns a command file to a client
  - Access control

Hosts are identified by the DNS hostname or their IP address.

The Users file

Lists privileged users

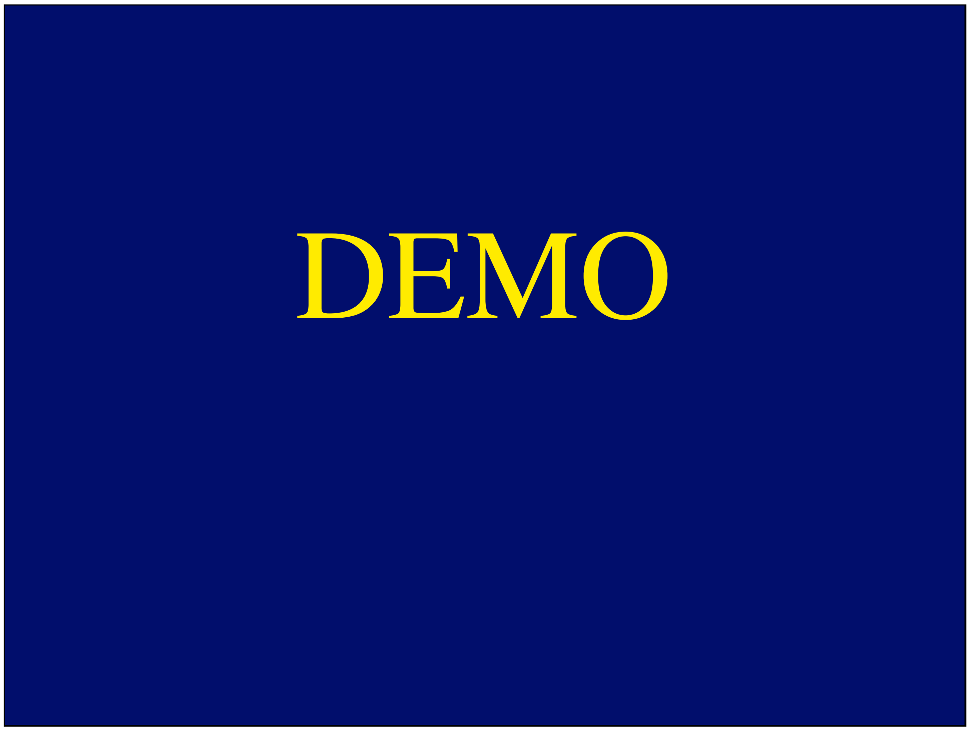
Doesn’t exist, yet



Graphical User Interface

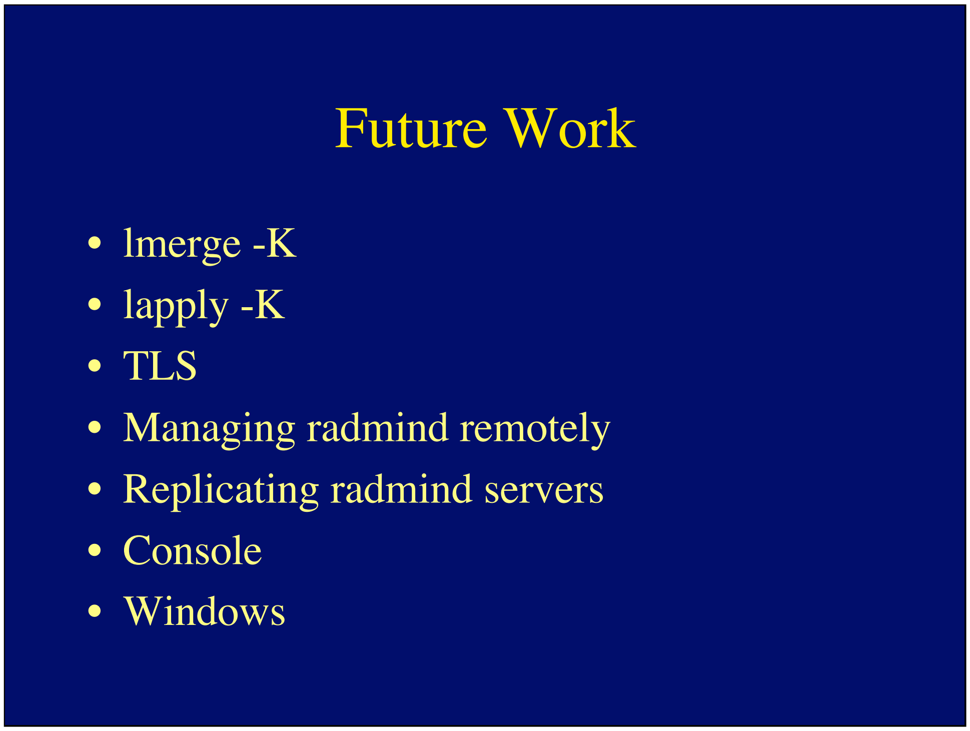
- For Mac OS X
- Demo’d at WWDC 2002
- Working on *real* versions

Real versions of the client-side GUI tools look a lot like a wizard. Radmind Assistant.



DEMO

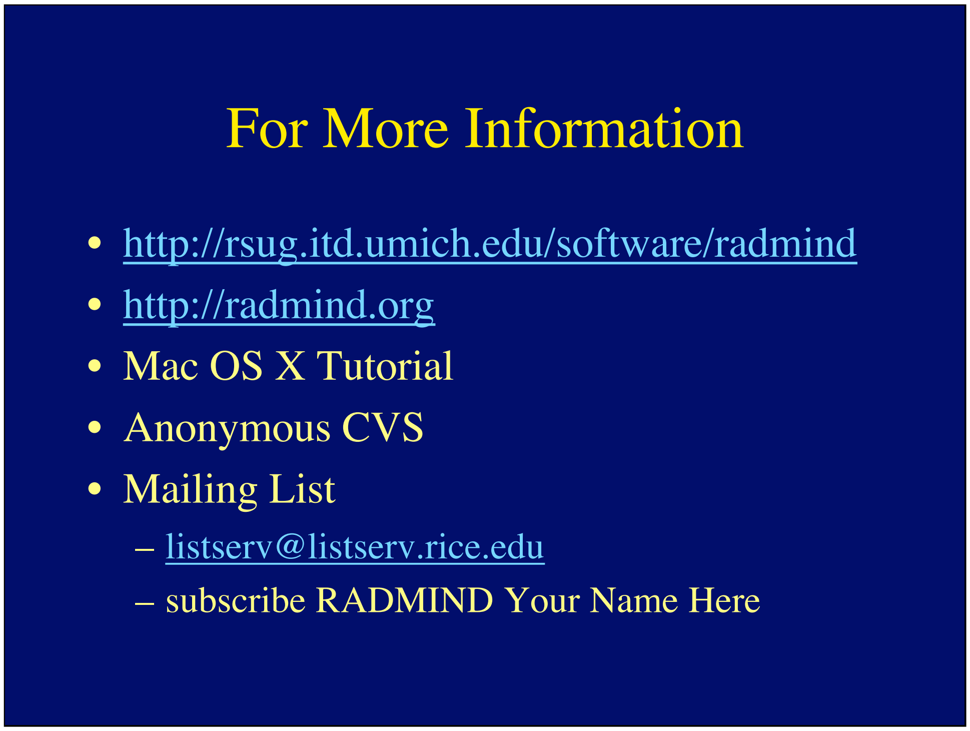
Andrew begins with a fully up-to-date machine. He installs Mulberry, using the installer's defaults. He notices that Mulberry has been installed in the root filesystem. He uses Radmind Updater to remove the errant copy of Mulberry. He starts the installer again, this time he chooses to install Mulberry in /Applications. He uses Radmind Loadset Creator to produce a create-able transcript which he reviews in the Radmind Transcript Editor. Radmind Loadset Creator stores the Mulberry loadset on the server. Andrew moves over to the server and runs the Radmind Manager. (Patrick runs Radmind Updater to remove Mulberry.) Andrew shows the server environment, and updates the command file for the client to include the newly created Mulberry loadset. Back on the client, Andrew runs the Radmind Updater to load Mulberry on the client. He launches Mulberry.



Future Work

- lmerge -K
- lapply -K
- TLS
- Managing radmind remotely
- Replicating radmind servers
- Console
- Windows

Support for privileged users, pipelining, wildcards in the hosts file, subdirectories in the special directory, directory finder info.



For More Information

- http://rsug.itd.umich.edu/software/radmind
- http://radmind.org
- Mac OS X Tutorial
- Anonymous CVS
- Mailing List
  - listserv@listserv.rice.edu
  - subscribe RADMIND Your Name Here