

GSLIS INFORMATION TECHNOLOGY SERVICES

COMPUTER LAB MANAGEMENT: ANALYSIS AND RECOMMENDATIONS

Prepared for Dr. Mary-Lynn Rice-Lively,
Coordinator of Information Technology Services

by Students
in the Fall Semester 1997 Class
Systems Analysis and Evaluation (LIS 387.5)

**Jon Drucker
Aimee Green
Dan Huo
James Irwin**

**Matt McGrievy
Kevin Rioux
Sybil Shearin
Magan Stephens**

under the direction of
Professor Ronald E. Wyllys

Graduate School of Library and Information Science
The University of Texas at Austin

1997 December 2

Those copyright items (if any) reproduced herein have been copied as fair use for the educational purpose for which the study treated herein was undertaken and is here reported.

Except for such otherwise copyright items, the entire contents of this report are copyright © 1997 by Ronald Eugene Wyllys for the benefit of the GSLIS Foundation, Graduate School of Library and Information Science, The University of Texas at Austin, Austin, Texas 78712-1276. Permission to copy, quote, reproduce, and/or use all or any part of the contents of this report (except those parts that are otherwise copyright) is granted to all persons and organizations, provided that due acknowledgment is made of the source of the material thus utilized.

Telephone: 512-471-3969/2742; 800-551-0294

Fax: 512-471-3971

Email: wyllys@gslis.utexas.edu

Graduate School of Library and Information Science

Sánchez Building 564

The University of Texas at Austin

Austin, Texas 78712-1276

TABLE OF CONTENTS

1 EXECUTIVE SUMMARY	1
2 PROJECT OVERVIEW	1
2.1 TEAM COMPOSITION	2
2.2 PROJECT DESCRIPTION.....	2
2.3 MISSION STATEMENT	2
3 INFORMATION TECHNOLOGY LAB: ANALYSIS	3
3.1 CURRENT SITUATION.....	3
3.1.1 <i>Present Capabilities</i>	3
3.1.2 <i>Historical Context</i>	3
3.1.3 <i>System Context</i>	4
3.1.4 <i>Operations and Organization</i>	4
3.1.5 <i>Constraints</i>	4
3.2 DESIRED CAPABILITIES.....	5
3.3 PROPOSED SOLUTION	6
3.3.1 <i>Summary</i>	6
3.3.2 <i>System Requirements</i>	6
3.3.3 <i>Hardware and Software Requirements</i>	7
3.3.4 <i>Security Requirements</i>	10
3.4 DESIGN AND IMPLEMENTATION ISSUES.....	10
4 CONCLUSIONS AND RECOMMENDATIONS	11
4.1 HARDWARE AND SOFTWARE RECOMMENDATIONS.....	11
4.2 TRAINING AND INSTRUCTIONAL RECOMMENDATIONS.....	13
4.3 SECURITY RECOMMENDATIONS.....	13
4.4 TIME FRAME	14
APPENDIXES	15
5 IT LAB SURVEY	16
5.1 METHODOLOGY	16
5.2 SURVEY INSTRUMENT	16
5.3 SURVEY RESULTS.....	20
6 LABMAN SITE VISIT DESCRIPTIONS	27
6.1 LEARNING TECHNOLOGY CENTER, COLLEGE OF EDUCATION.....	27
6.2 COMPUTER SCIENCE LAB, T.S. PAINTER HALL.....	28
6.3 ELECTRONIC INFORMATION CENTER (EIC), UNDERGRADUATE LIBRARY	30
7 TECHNICAL RESOURCES	31
7.1 CORNELL UNIVERSITY IMPLEMENTATION NOTES.....	31
7.1.1 <i>PC-Rdist sample distribution script</i>	34
7.2 KENT STATE UNIVERSITY IMPLEMENTATION NOTES.....	35
7.3 REV RDIST CONFIGURATION INFORMATION	38
7.4 REV RDIST ADMINISTRATOR'S GUIDE.....	43
7.5 USING ASSIMILATOR	60
8 REFERENCES	73
8.1 IT LAB TEAM RESOURCES	73
8.2 LABMAN AND RELATED PRODUCTS RESOURCES.....	73
8.3. TRAINING RESOURCES.....	74
8.4 REV RDIST LISTSERV INFORMATION	74

1 EXECUTIVE SUMMARY

At the outset of this project we identified a need for a Lab management system. Previous efforts by Lab staff had focused on pieces of software rather than a coherent system, and the user was missing from the equation. We resolved to make a structured effort to design a system that proceeded from both user needs and administrative / managerial concerns.

To achieve consistent services and an environment for all users we saw the need to gather data on who the Lab's users are and identify their information technology service and software needs, and on the hardware, software, and networking requirements for a system to meet those needs.

Our data gathering took three forms: a Lab user survey administered to over 100 GSLIS students, research into existing lab management tools, and interviews with managers of labs using such systems.

The purpose of computer lab management systems is to make more productive use of (often limited) staff, material and digital resources. Generally the main component of such a system is a suite of lab management software that handles access control and system security, the distribution of materials in the network and the monitoring and/or limiting of any pertinent quotas, replication of machine standards, and remote monitoring of system activities. Lab management software enables staff to perform tasks more effectively and efficiently, gathers usage data to produce statistics that can be used to inform planning decisions, manages software distribution to insure both compliance with licensing agreements and best access for users, and provides security by keeping user logs and controlling access on several levels. Access control may include access to individual workstations, access to shared resources, or establishing levels of access such as User, System Operator, and System Administrator, for example.

By standardizing startup and rebuilding procedures, and by monitoring machine status, hardware and software maintenance and upgrades can be centralized, coordinated, monitored, and streamlined. software resources also gain a measure of protection against both piracy and data corruption. Lab software systems can also coordinate user requests for assistance by sending help requests to a standard location and by identifying a description of the problem to insure that appropriate staff members respond. The result is a better quality of help for users and support time more efficiently spent.

Control of access, especially in regards to use monitoring, must be handled mindfully. Log files may contain information subject to privacy laws, university regulations, or organizational by-laws, and determining who is to have access to such discretionary information is a decision to be carefully considered and formalized in written policy.

Lab management software can also manage printing services in a lab. Printing charges may be a source of income, and many labs have little control over print jobs sent over a network. By linking jobs in a print queue to their sources a lab manager can insure that print services are paid for or even prevent jobs from going to the printer without prepayment.

The software we studied is a package developed here at the University of Texas at Austin, in wide use in UT campus computer labs. Technical support is abundant on campus, and upgrades are made and tested locally. One added advantage is that this software package is available at no charge to UT-Austin computer labs. In addition, a wide base of lab users translated to a wealth of experience by both system users and administrators that our project could draw upon.

2 PROJECT OVERVIEW

This report presents the results of a systems-analysis and -design project conducted during 1997 September 30 through 1997 December 2 by graduate students in the Graduate School of Library and Information Science (GSLIS) at the University of Texas at Austin (UT-Austin).

The original group of students identified themselves as the GSLIS Network Services Team. The team consisted of the following thirteen graduate students:

David Brown	James Irwin	Sybil Shearin
Mike Carlson	Matt McGrievy	Magan Stephens
Jon Drucker	Liqun Pan	Jennifer Thomas
Aimee Green	Kevin Rioux	Steven Williams
Dan Huo		

together with Dr. Ronald E. Wyllys, Professor of Library and Information Science. The students carried out the analysis as part of the requirements for course LIS 387.5, Systems Analysis and Evaluation, as taught in the Fall Semester 1997. In its first meetings this rather large group identified two distinct subprojects within the project and divided into two separate teams: one to address the original task of assessing administrative and faculty intranet network needs (the Intranet Team) , the other focusing on issues facing the Information Technology Services (ITS) Computer and Multimedia Labs (the Lab Team).. While these two groups coordinated much of their efforts, met together on several occasions, and shared personnel, resources, and information, the two acted as autonomous bodies and are presenting their results as separate reports. The words "we," "us," and "our" in this report refer to the student members of the Lab Team together with Dr. Wyllys.

2.1 Team Composition

The Lab Team consisted of the following personnel:

Jon Drucker	Aimee Green	Dan Huo	James Irwin
Matt McGrievy	Kevin Rioux	Sybil Shearin	Magan Stephens

One of the team members is on the Information Technology Lab staff; four team members are Lab volunteers. Kevin Rioux also spent a significant portion of project time working with the Intranet Team.

2.2 Project Description

The LabMan Team's main focus lay in analyzing possible management solutions for the two Information Technology Labs, the Computer Lab and the Multimedia Lab. We used earlier efforts by Lab staff concerning the implementation of laboratory management software as a starting point but departed from those efforts as they dealt with individual software applications rather than an entire system for Lab users. We focused on the Computer Lab (IT Lab) in our analysis but attempted to create a system that could be extended to the Multimedia Lab.

Our primary client in this project is Dr. Mary-Lynn Rice-Lively, co-chair of the GSLIS Technology Planning Committee, Coordinator of Technology Initiatives, and Academic Assistant to the Dean. This report will also be submitted to Luke Rosenberger, ITS Technical Manager.

2.3 Mission Statement

The mission of the Lab Team concerned the GSLIS Information Technology Laboratory (IT Lab). The IT Lab was in need of laboratory management software. The reason for this need was twofold: The IT Lab needed to provide consistent machine settings for the people who use the computers, so that users could move from one computer to another with a minimum of difficulty. In addition, Lab staff required the means to maintain the computers in the lab in an automated fashion, primarily in installing and configuring software, and to gather usage statistics.

The 1997-98 GSLIS Vision Plan states that part of GSLIS's Vision is "to prepare Masters and PhD graduates for leadership in the development, management, use, and evaluation of digital resources, including socioeconomic impacts." Based on that Vision one of the School's objectives is that

Faculty and students will be directly involved in practical research in the design and usability of digital resources throughout the GSLIS program. The School is committed to developing its research programs, especially to embrace further the broad areas of digital

(hypermedia) information resource management, intelligent search agents, and user environments (*italics added*).

Based on these statements the Lab Team sets forward its Mission:

“To make the Information Technology Labs more useful and usable by providing a standardized computing environment, and to improve management of the Lab’s information resources by creating a coherent, scaleable, and adaptable system that meets both user and administrative needs.”

3 INFORMATION TECHNOLOGY LAB: ANALYSIS

3.1 Current Situation

3.1.1 Present Capabilities

As of November 1997 the GSLIS Computing Lab (IT Lab) is equipped with 7 Unix machines: 5 Hewlett Packard Workstations running the HP-UX Unix operating system and 2 Sun UltraSPARC servers running Sun OS v.5.5.1. There are 16 PCs, all running Windows 95 (5 IBM PCs with Pentium processors, 8 Compaq 486's, and 3 newer Dell Optiplex models), and 15 Macintoshes [14 PowerMacs (one PM 7300, 2 PM 8500s, 11 PM 7200s) running Mac OS8, 1 Mac IIfx running Mac OS 7.6.1]. All machines are publicly accessible during normal hours except for the one mission critical Sun server, which is reserved for use by the System Administrator. One Compaq and one PowerMac 7200 are used as demonstration machines in classrooms. Peripheral devices in the Lab include 12 locally connected dot-matrix printers, one networked Hewlett Packard laser printer, and one flatbed color scanner (connected to a PowerMac). Four older Magitronic PCs that were removed from service during Summer 1997 are being installed with Linux OS in an attempt to restore them to use and also as a learning project for Lab staff.

The IT Lab is the primary computing facility for GSLIS students. The Lab is also a teaching laboratory, primarily through the use of the Lab’s classroom housing most of its Macintoshes (the Mac Annex). Lab resources are used for classroom assignments, with an emphasis on use of the Internet, online searching (using Web browsers, DIALOG, and other information retrieval tools), cataloging tools, graphics programs, HTML, and word processing.

The Lab also houses a small but extensive (estimated at 1000 volumes) collection of print materials including reserve items placed by instructors, reference books, instructional manuals, tutorials, cataloging references, and technical specifications. Some items are available for circulation, although loan policy is at best sketchy and often arbitrary. The collection is largely uncatalogued at this writing.

The Multimedia Lab (MML) currently has only one PowerMac 7300 at present, reserved for the use of the Media Lab Teaching Assistant. During the past year the MML has been equipped with additional Ethernet connections in preparation for more computers. Peripherals currently include one dot-matrix printer and a 35mm film / slide scanner.

All lab machines (as well as staff and faculty computers) are currently connected to UTNet, the campus fiber optic ring. As of this writing the Lab’s network connection has a capacity of 10 Mbits/sec.

3.1.2 Historical Context

During the Summer of 1997, GSLIS IT Services upgraded from Hewlett Packard Apollo 705, 710, and 715 stations that supported an aggregate of 5 GB to over 400 users to two Sun Microsystems UltraSPARC Enterprise I Internet servers, each with 143 MHz clock speeds and 2 GB of hard disk space. One Sun station (named Fiat, the mission-critical server) is connected to a Sun Scaleable Storage Array (SSA) of RAID (redundant array of inexpensive disks) storage with a total capacity of 27.3 GB. The SSA is scaleable to 70 GB.

GSLIS’s network connection is provided a single Ethernet interface on a Cisco AGS+ Router maintained by the College of Education (CoE), delivering network point-of-presence to GSLIS network users. In November 1997 the Cisco Router was upgraded from 10Mbps in capacity to 100Mbps as part of an

upgrade plan to bring CoE and GSLIS up to speed with the campus-wide 100 Mbps UTMNet FDDI Ring. By the start of the Spring 1997 semester the Lab should have full 100Mbps capacity.

3.1.3 System Context

In order for the computer system in the IT Lab to run smoothly and efficiently, certain controls must be exercised over the hardware and software. In the past, Lab users have been able to use the machines with no constraints. As a result, students frequently have downloaded software, reconfigured machines, and unknowingly sent commands to the laser printer. This has placed undue demands on the staff, forcing them to spend unnecessary time reconfiguring machines, helping users recover data, and rebooting frozen machines. Printing has been a particular problem. Dot matrix printing is free and laser printing costs per page. Students attempting to print to dot matrix printers have often sent print commands to the laser printer by mistake. Unfortunately, the Lab has had no way to recover costs of unwanted laser printer pages. In addition, no systematic restrictions have been placed on who is able to use the Lab. The GSLIS IT Lab is intended for students, faculty, and staff of the program, but currently the Lab has no way to enforce the policy.

The need for some control mechanism has become apparent as system problems have accumulated over time, reducing the efficiency of the machines and staff.

3.1.4 Operations and Organization

GSLIS IT services is coordinated and supported by two full-time staff (Coordinator of IT and Technical Manager), 9 part-time student employees (6 Teaching Assistants and 2 Graduate Research Assistants - all graduate students in GSLIS - plus one undergraduate Federal Work Study student), and 5 student volunteers (working an average of 3 hours per week each). The IT labs are available and staffed for full technical support 57 hours a week. Lab staff also provide training in the form of short courses on the use of computing and networking services as well as one-to-one technical assistance.

All IT policy, planning, implementation, training, and maintenance are coordinated by the GSLIS Coordinator for Information Technology and the Technical Manager of GSLIS Computing, with regular and frequent input from Lab staff.

Student access to the Lab has historically been flexible, with four Lab keys available for checkout for afterhours access. Students were also permitted to remain in the Lab after closing to continue work as long as they signed a statement of responsibility. At its November 18th meeting, GSLIS's Graduate Studies Committee (GSC) unanimously voted to restrict Lab access to the designated hour of operation (Mon-Thurs 9am-8pm, Fri 9am-4:30pm, Sat 12noon-6pm) due to concerns over Lab security. The GSC noted that Lab policy was not in compliance with University regulations on computer lab and building security, and that several assaults in the Sanchez Building had occurred over the past three years as well as a number of computer equipment thefts across campus in recent months. The Lab plans to conduct a security audit in the next few months. Due to the demand for access Lab staff agreed to temporarily add Sunday hours (12noon-4pm) and review hours of operation for the Spring semester.

3.1.5 Constraints

Presently, the major obstacle to implementing controls over the GSLIS computing system is a lack of homogeneity of hardware and software across the network. Current lab management software designed to aid in the administration of public computer labs requires a high level of uniformity across platforms and individual machines. With the impending purchase of several new computers (outlined in the next section), a greater degree of similitude will be achieved.

The IT Lab is also limited by its physical space. As GSLIS occupies only a small corner of the Sanchez Building and is regarded as sort of a "tenant" of the College of Education, little room for expansion existed short of the construction of a new building. The latter is unlikely in the short term. There are only so much space available for computers in the Lab and Annex, and most space is currently occupied. There are also building safety codes to consider.

The Multimedia Lab's wiring was recently upgraded to four Ethernet connections. At present this appears to be the maximum capacity for this Lab beyond future building-wide upgrades. The four machines planned for the MML would seem the capacity load for networked equipment. The MML must also use the IT Lab's networked laser printer for printing above dot matrix quality - something vital to the MML's potential graphic output.

At present both Labs have a finite number of network connections. The number of machines connected to the network is limited to this number. As the building's network apparatus was just recently upgraded, it seems unlikely that this limitation will change any time in the near future.

ITS's staffing budget also limits hours of Lab availability. In a time of budgetary downsizing the Lab has little prospect for increasing Teaching Assistant or Graduate Research Assistant staff hours, although it has been suggested that the Lab add work-study students to extend staffing abilities at a relatively low cost.

3.2 Desired Capabilities

As the Mission of the IT Lab is to serve student users, ITS wishes to maximize its facilities to fit coursework, research, and professional development needs of its users. The desired capabilities include up-to-date hardware, software, and telecommunications facilities, especially those that are likely to be present in students' future places of work as well as enable students to further the development of Library and Information Science as both an academic and a professional discipline.

Ideally the Labs should provide access to:

- Word processing, spreadsheet, presentation, and database software.
- Internet applications including Web browsers, telnet and FTP clients, and new communications technologies.
- Library applications such as OCLC access, LEXIS-NEXIS and Dialog, cataloging software.
- Access to UT Library Online databases
- Facilities for viewing and producing graphic, audio, and video materials, including appropriate multimedia applications and equipment.
- Reference and instructional materials; pathfinders to electronic and print resources.
- Facilities for new technologies as appropriate.

These facilities require several factors to make them useful to student users, as listed below:

- Consistency. Appearance and content should be standardized to the extent possible.
- Trained personnel. Staff needs to be knowledgeable, service-oriented, and flexible.
- Access. Facilities need to be available during hours of greatest demand.
- Educational programs. Instruction in information technologies at novice, intermediate, and advanced levels is essential.

Many of the IT Lab's machines are approaching obsolescence. Already 5 older PCs that had run Windows 3.1 have been removed from operation. Four of these are presently being retrofitted with the Linux operating system as an experiment by Lab staff. Several older Macintoshes have been moved to doctoral students' and other offices to replace their even more antiquated equipment. Although budgets remain tight, equipment upgrade continue to be planned, and recent "price wars" in the computer industry may provide further opportunities to keep Lab machines up-to-date in accordance with GSLIS's mission.

By the beginning of the Spring 1998 semester three to four PowerMacs will be moved into the Multimedia Lab to be used with various multimedia development applications. The machines will be moved from the Macintosh teaching classroom (the Mac Annex). The remaining six Macintoshes will move out into the main lab, to be replaced by nine new Dell Optiplexes along with the 3 existing Dells. The timeline for this switch is still undetermined but expected to occur in early 1998. These 12 machines will effectively become the Lab's teaching machines, as the room they will occupy (currently called the Mac Annex) is the Lab's primary classroom. (Some Lab staff have already begun to refer to the proposed classroom as the Dell Annex, although the prospect of naming it after a UT dropout may not appeal to GSLIS faculty.) The two desktop machines presently used for classroom demonstrations will be replaced with new laptops, allowing the current demo machines to return to Lab service. Plans to provide network access to laptop users have also been discussed, and one Ethernet socket is currently temporarily designated for portable computer access.

The Multimedia Lab is the subject of the next phase of ITS expansion plans. With the addition of 3 powerful computers designed for multimedia application, the MML stands to become the main site for such applications as graphics manipulation, teleconferencing, desktop publishing, and instructional materials production. At present no new computers are planned for purchase beyond those to be moved from the IT Lab, and the MML's plans to purchase a new high-resolution flatbed scanner present the only other major addition to the lab.

3.3 Proposed Solution

3.3.1 Summary

Efforts during Summer 1997 to adopt lab management software in the IT Lab faltered for a number of reasons that cannot be accurately assessed within the scope of this project. Two needs as perceived by the Lab Team were 1) a comprehensive study of existing facilities using lab management and desktop reconstruction software and 2) a survey of Lab use patterns and user needs.

Our study determined patterns of existing and desired use that for the most part fall in line with existing Lab facilities. The major areas requiring attention include:

- Improved management of certain lab resources under heavy demand.
- More training in certain software applications.
- Increased availability of multimedia applications.

We propose the implementation of LabMan lab management software along with disk reconstruction softwares as appropriate to the existing and planned future hardware in both the IT and Multimedia Labs. This implementation will require minimal hardware and software purchases and only moderate staff time. Once adopted, this system should alleviate much of the staff time put into machine maintenance and software upgrades.

LabMan together with associated disk restoration utilities will not only regulate the allocation of Lab resources but also gather usage statistics, provide a fair amount of security for the Labs as a whole, and make easier the reservation of teaching resources.

3.3.2 System Requirements

The Lab manager software (LabMan) was created by the Departmental Services division of the University of Texas at Austin Academic Computing and Instructional Technology Services (ACITS). It is intended to simplify day-to-day management of a microcomputer laboratory.

LabMan is capable of performing the following functions:

- Restricts workstation access to authorized users (departmental students, faculty, staff)
- Handles waiting lists and station assignments

- Usage logging and statistics reporting
- Workstation software configuration maintenance and updates
- Help system for summoning support personnel to workstations
- Logging of support requests
- Print Services

LabMan is composed around a central lab server running Microsoft Windows NT, and additional administrative stations to maintain LabMan's functions. Lab stations can contain a mix of Macintosh and IBM PC-compatible stations.

Each workstation requires access to a network file server. PC's require a "DOS requester" while Macintoshes can use any Apple Filesharing Protocol (AFP) compatible server and workstation software

LabMan requires an Ethernet network supporting TCP/IP Protocol. The server portion of the LabMan runs under Microsoft Windows NT. Any system running Windows NT can act as a server.

The only satisfactory alternative to LabMan is the University of Georgia's Lab Manager, a similar set of programs that work together to help manage and control networked computer labs. As the University of Georgia grew in size, a more efficient waiting list procedure was needed in the computer labs. When the labs were smaller the waiting lists could be reasonably managed by one person and a sign up sheet. Now with Lab Manager and a designated waitlist station, the lab staff is free from informing incoming users of the wait policies and free from patrolling the lab looking for an open computer. The waitlist station informs users of procedures as well as the length of the wait. Now the lab staff at University of Georgia has become more productive because they no longer need to police the lab and handle a manual waiting list.

Like LabMan, Lab Manager compiles use statistics. Lab Manager records who has logged in and logged off a computer and which applications they have used. The user has also been added to a statistical database to monitor usage trends. These statistics can be exported to a text file to be analysed by a spreadsheet or statistics program.

Since Lab Manager performs more or less the same functions as LabMan, the latter's local development and support proves a clear advantage. For this reason we have discounted Lab Manager as an alternative solution.

3.3.3 Hardware and Software Requirements

PC Requirements:

Pentium compatible processor
 32MB RAM
 MS Windows NT 3.51 (Workstation or Advanced Server)
 ISA 16-bit Network Interface Card
 TCP/IP protocol stack (Winsock, Novell, Microsoft TCP/IP)

Mac Requirements

68030, 68040 or Power PC processor
 32MB RAM
 System 7.1 or later
 Thread Manager 2.x (built into system 7.5)
 Mac TCP 2.0.6 or Open Transport TCP/IP

Currently LabMan is being implemented in some Macintosh labs running OS8 with few but significant documented problems.

For either platform lack of RAM seems to be the biggest problem. 32MB is recommended if a single Windows NT server is used for both LabMan services and as a file server.

ACITS claims that Windows NT Advanced Server is capable as the Network Operating System (NOS), which supports DOS, Windows, Windows for Workgroups, Windows 95, Windows NT and Apple

Macintosh clients. As LabMan is designed for Windows NT, it is wise to consider NT Advanced Server as your lab file server

The Mac clients rely upon RevRdist, a "demand-pull" utility developed by Purdue University. RevRdist is a Macintosh compatible hard disk management utility for computer labs. It is designed to restore individual hard disks to a prescribed configuration.

3.3.3.1 RevRdist

RevRdist is an instructional computer lab hard drive management utility for Macintosh computers written by Dale Talcott of the Purdue University Computing Center. It is designed to run in a Macintosh lab environment to restore individual hard drives to a configuration specified by the lab administrator. RevRdist compares the local hard drive volume to a master folder located on a network server. Where differences are found, RevRdist can update the local file or folder from the master copy, simply ignore the local file or folder, delete them or move them to a "junk" folder where they will be reexamined or deleted later.

RevRdist does not prevent users from modifying the contents of the hard drive. It does not affect the normal operation of the Macintosh computer in any way. Instead, you run RevRdist at startup, shutdown, or at any other convenient time and RevRdist will restore the local hard drive volume to a desired configuration.

RevRdist is named after the UNIX Rdist program, but operates in a "reverse" manner. Rdist, "remote distribution," distributes files from a central site to a remote UNIX system. While RevRdist allows individual Macintosh computers to "pull" desired files from a central server. It is important to point out that RevRdist is a Macintosh application that runs independently on each Macintosh computer just like any other Macintosh software application. RevRdist logs on to a file server and copies files from the server to the client as specified.

No part of the RevRdist program actually runs on the file server. The file server simply stores the files that RevRdist accesses. Contrary to what has been reported in the past, RevRdist is not a program designed for UNIX servers. RevRdist is a Macintosh application designed to run in a Macintosh environment.

RevRdist requires that the system support networked clients running system 7.1 or greater. LocalTalk (PhoneNet) is an option, but a 68030, 68040 or PowerPC hard drive with at least 8MB RAM, Thread Manager 2.x (built into System 7.5+) and MacTCP 2.0.6 or Open Transport TCP/IP are recommended. Any system capable of running Windows NT 3.51.1 or 4.0 can act as a server. A Pentium or compatible processor with at least 16MB RAM and a ISA 16-bit Network Interface Card is recommended.

The RevRdist program and its source code are now part of the public domain. Therefore, a lab administrator can acquire RevRdist by simply downloading the software from Purdue University. This site also maintains useful "user-contributed question and answer" links and provides a list of links to other sources of Macintosh lab management software. The Purdue University Computing Center also provides a listserv dedicated solely to solving RevRdist related problems.

3.3.3.2 PC-Rdist

PC-Rdist is a software distribution utility for Windows and DOS. It is used to maintain a standard set of files and settings on multiple computers across a network. To accomplish this, PC-Rdist runs on the client machines and compare files and settings on the local computer to master copies located on a server.

To effect standard settings across a network, PC-Rdist replaces missing or modified files, moves files added by users to a temporary directory (with the option of deleting them), updates the local registry, and restores system settings.

Elapsed rebuilding time for PC-Rdist is approximately 2.5 to 3 minutes.

Because PC-Rdist is a component of the LabMan software package, system requirements will be similar. If PC-Rdist were to run by itself on the network, system requirements would be reduced slightly. Currently the absolute minimum hardware requirements for running PC-Rdist (Win32 version) are:

<u>Client</u>	<u>Server</u>
386 Processor	Pentium or compatible processor
Networked machine running Win95	Microsoft Windows NT 3.51 (Workstation or Advanced Server)
8MB RAM	32MB RAM
ISA 16-bit Network Card	ISA 16-bit Network Interface Card

In order for PC-Rdist to run at optimal speed and performance, system requirements covered in the LabMan section (4.3.4) are recommended.

A shareware version of PC-Rdist is available to evaluate for 21 days. After 21 days, a license must be purchased. Single user licenses cost \$10.00, departmental licenses cost \$400.00, and organizational licenses cost \$1000.00. Licenses are distributed on a per-client basis; a single computer lab using PC-Rdist would require the departmental license, while a university would require an organizational license. Departmental licenses cover an unlimited number of computers owned, operated, and maintained in a lab and organizational licenses cover an unlimited number of departments in the university. Yale University's PC-Rdist mailing list is also a valuable source of information for discussions of both general PC network maintenance and PC-Rdist.

In the past, system administrators had the choice of using either PC-Rdist or Rdist4PC, another piece of rebuilding software. Rdist4PC is no longer maintained nor documented, so PC-Rdist is the best option for Windows 95 machines in the GSLIS Computing Labs. Because the software is already well integrated into the LabMan package, running PC-Rdist requires minimum manipulation on the part of IT Lab system administrators. In addition, PC-Rdist is well-documented and a number of other university computer labs are already using the software. Many computer labs have posted useful resources on the web, including trouble-shooting and configuration guides. These sites are listed in Section 7.

3.3.3.3 Assimilator

Assimilator was designed by Peter N. Lewis to make the hard drive of a large number of Macintosh computers in an educational laboratory environment conform to the specifications of a lab administrator.

When Assimilator runs it will access an Appleshare file server and then make the client hard drive look identical to a pre-specified source folder on the server. It does this by comparing each file in the source folder to the corresponding file on the destination hard drive. If the file on the destination is missing, or has been changed, then it is thrown away (into the Trash) and a new copy of the file is downloaded from the source. Any extraneous items on the destination are also thrown away. Finally aliases, icon positions, folder views, finder flags and so forth are all set to match those in the source folder.

Assimilator makes no attempt to secure its files from malicious users. If a lab has malicious users, then the lab administrator must take additional steps to protect the system. This might include installing lab management software or making the System Folder invisible. Assimilator also requires System 7.0 or later.

Assimilator is a Shareware program made available by Kagi Shareware of Berkeley, CA. This requires the lab administrator to pay a licensing fee of \$10 per machine. Assimilator is licensed on the basis of the number of Macs it is running on, not on the number of copies of Assimilator Admin used.

If the lab administrator wishes to run Assimilator on 0 machines or more he/she should consider purchasing a site license for \$00. A site license covers usage of Assimilator on an unlimited number of machines owned by the licensed organization that are within 100 miles of a given point designated by the licensed organization. Presently, the University of Texas plans to purchase the rights to Assimilator and install it on the

Macs in the Computer Science Department computer labs. This means that the use of Assimilator would be free for all other departments on campus.

3.3.4 Security Requirements

3.3.4.1 Online Authentication/Authorization

Lab TA Lance Hayden is constructing a firewall feasibility study for the Unix servers in GSLIS. Upon consultation with the Lab Team he has extended the study to include the proposed intranet and NT server, as well as the lab management system.

Two issues to consider regarding network security are authentication (logging in who gets onto a computer) and authorization (how much access a user has on a computer). LabMan handles session authentication by requiring the user to enter a login name and password before accessing system resources. Disk rebuilding software such as PC-Rdist, Assimilator and RevRdist can be used to standardize individual workstations after each session so that students cannot make permanent changes to the system configurations, effectively limiting their authorization.

A security policy must be developed which takes into account who the users are, their current and anticipated access needs, the level(s) of access to be provided, and the means of access (e.g., remote or local, via client or console, on which server, etc.). Additionally, if the IT lab management system and the proposed administrative intranet are to share servers, security efforts must be coordinated.

In any network there are three tiers of users to be supported. These are arranged in a hierarchy by degree of access. In general these groups (in order of increasing levels of access) are the general public, an internal group of users, and system administration. In the case of the IT lab the levels correspond to GSLIS student users with the least access, system operators such as Lab staff and volunteers as well as any users with expanded or enhanced access such as faculty or members of the GSLIS Web Team in the middle, and the system administrator(s) with priority access privileges at the top of the hierarchy.

Security audits add another measure of protective maintenance to a lab's domains. Security audit software capable of analyzing a networked station, a server, or an NT domain is available. Currently an evaluation copy of one such application, Intrusion Detection, Inc.'s Kane Security Analyst for Windows NT, is installed on one of the Lab's PCs as part of Lance Hayden's project mentioned above.

3.3.4.2 Physical Plant Security

The IT Lab has only had a rudimentary written security policy in the past, with the recent decision of the Graduate Studies Committee bringing the IT Lab into compliance with University policy on lab and building safety. Access to the lab after hours had previously been given to students based on a key checkout system; this policy was found to be in violation of the University's security policy for computer use (memorandum 6.302, Office of the President, September 10, 1993). The Lab is now only accessible during the 61 staffed hours per week, which have been expanded from 57 hours. The 4 additional hours are on Sunday, a day the Lab had previously been closed.

Lab staff, faculty, and doctoral students are issued keys to the Lab. With the change in policy, and also due to a few keys still in circulation, locks will be changed as soon as possible; doctoral students may not be issued new keys. Previously students were required to sign a statement of responsibility on an afterhours log stating their name, time entering and time leaving the lab. Whether this statement would have been legally binding is questionable.

The Lab is considering security devices such as fiber optic cable locks, which attach securely to equipment and must be deactivated by a passcode before disconnecting. If disconnected without the passcode the optical signal alerts Campus Police. Other possible additions include door alarms or motion detectors.

3.4 Design and Implementation Issues

LabMan is constantly being upgraded and improved. To meet the needs of the many University departments implementing LabMan, a reevaluation of the accounting functions is necessary. The upgraded

design calls for a reduced number of user lists with associated values indicating a particular user's validation options. These options would be customizable, to fit the criteria by which you wish to restrict or provide access to your lab. This criteria could restrict the lab to Graduate Students or to staff members.

Unfortunately, a redesign may require substantial time to accomplish. Labs wishing to use Computation Center account validation using the Print Relay Services (PRS) system may find it easier to get started using LabMan. Users would simply log in using IF (Individually Funded) account numbers. The Print Relay Service enables lab users to print to any of ACITS's output facility printers. Charges are per page and automatically billed to students' IF accounts.

Benefits to Using Current PRS System:

- Validation functions are currently designed for PRS validation, providing simple ways for users to change passwords and set spending limits for PRS printing.
- Nearly 75% of University students already have IF account numbers
- Account numbers are free. Charges are only accrued for printing to Computation Center output sites and other Computation Center services.
- Users can take advantage of Computation Center support facilities such as the Help Desk for acquiring forgotten passwords and setting up new accounts.
- User access is consistent across labs. There's no need for a user to remember more than one account number or password.
- The system works. In two years of operation, the Student Microcomputer Facility (SMF) saw over 1.3 million logins, with few glitches.
- The Computation Center can provide lab administrators with a list of IF Accounts which belong to a given set of University major codes, allowing you to restrict access to your lab to members of your department or college.

PRS requires an output of approximately 50,000 print jobs per year to be of use. Some labs have chosen either to not use print services at all (e.g., the College of Education's Learning Technology Center) or to use a third-party print management solution such as Uniprint (e.g., some branches of the General Libraries).

Uniprint is the primary alternative to PRS at UT-Austin. Uniprint uses a server attached to a laserprinter and an Allcard (printing debit card; reader the same type used on photocopiers across campus). Users insert their Allcard, select their print job(s) on the server's terminal screen, the card reader deducts the print charges from the card, and the job is sent to the printer. The advantages of this system are:

- No further accounting or billing is necessary.
- Annual job loads less than 50,000/year can be handled
- Uniprint servers are owned and maintained by a third party, CDP.
- Print jobs sent accidentally can be canceled before any pages are "burnt".

The disadvantage to using Uniprint are that print jobs are not tracked from start to finish, and it is possible for users to inadvertently (or intentionally) print jobs other than their own.

4 CONCLUSIONS AND RECOMMENDATIONS

4.1 Hardware and software recommendations

Given the preceding analysis, the Lab Team recommends that LabMan software be implemented as a management solution for IT Lab computers. LabMan is a well-tested technology, with high quality, local technical support, and it is distributed free of charge.

Although present official documentation for LabMan is somewhat sparse, a great deal there of informal documentation is available through the LabMan User Group. Official documentation for LabMan is due to be released in August 1998.

LabMan's requirements are relatively modest. It will require a dedicated machine that will be located in IT Lab itself (due to a lack of space in the Technical Manager's office). This machine will need to be a Pentium-based PC running Windows NT Server 3.51 or later, with a minimum speed of 133 MHz, a minimum RAM of 48 (with an optimal RAM of 64) and a minimum of 1 GB of storage space. One of the IBM Pentium machines currently owned by GSLIS may be configured to be this server. Furthermore, we recommend one machine as a backup server. If feasible this backup should be the Micron server Jeep, currently housed in the Technical Manager's office.

A signup station for IT Lab users is also recommended. During times of heavy usage it will be necessary for students to reserve machines or sign up on a waiting list administered by LabMan. A single public workstation (either Mac or PC) in the IT Lab will serve this function. This machine should be able to support the same version of NT Server as the LabMan server but will require less storage space.

In conjunction with LabMan, we recommend the latest version of PC-Rdist for the management of PC desktops and hard drives. PC-Rdist disk images can be housed on the same server as LabMan.

For Macintoshes, we recommend that the Lab administrator choose one or both of the currently available disk and desktop reconstruction clients for Macintoshes, which also would run on the LabMan server. At this point, the Lab administrator may choose between RevRdist or Assimilator. The choice may not be easy for several reasons.

There are a number of issues that a lab administrator must address before making a choice between RevRdist and Assimilator software. What documentation is available, what information is available on past incompatibility problems, what is the potential for system incompatibility if a new operating system is introduced, is there a current up to date version of the program available, what level of support is there, and what is the cost of implementation.

RevRdist - Purdue University has not provided a release version in a number of years. They have updated the software and put out beta versions, but with each new beta version a few bugs are corrected and a new beta version released. The documentation available for RevRdist is old and incomplete. The most current information was updated in the Spring of 1996. However, Chris Cooley manages the Macintosh computers at the University's student microcomputer facility where RevRdist is used and he is very accessible and could be consulted if problems arise. It is important to remember RevRdist is part of the public domain and therefore free to use.

Assimilator - The Peter N. Lewis group introduced release version 1.1 in February 1997 and has provided complete documentation. One main concern with Assimilator is its failure to operate outside of a homogeneous environment. Tjorn Smith, the Computer Science computer lab manager, has found ways to work around some of these problems. He also points out that the new Assimilator version is much faster and much easier to work with when compared to RevRdist. He hopes to have the remaining bugs worked out and documentation on his efforts available in the near future. Tjorn states he is on the verge of purchasing a site license for Assimilator which would allow all other departments on campus to use it free of charge.

One primary concern that has not been addressed in the available literature is the compatibility of this software with Macintosh's OS8. It appears Chris Cooley and his staff at the student microcomputer facility have worked out all the problems they initially had running RevRdist with OS8. There is very little information available on Assimilator's compatibility with this new operating system. It is important to understand there will be numerous compatibility problems when running both of these lab management utilities with software and operating systems that will be released in the future. It is an absolute necessity for the lab administrator to have access to strong support resources.

As a RevRdist user the University's Student Microcomputer Facility provides a high level of access and support. For this reason we tentatively recommend the GSLIS IT Lab install RevRdist as the Macintosh

hard drive management utility. However, the lab administrator should pay close attention to the efforts being made by Tjorn Smith in the Computer Science computer labs. If he is able to work out all of Assimilator's problems with various software and hardware, develop the documentation, and acquire a site license then the speed and ease of use of Assimilator may be reason enough for change.

Print services may be handled by either LabMan's Print Relay Services (PRS) or by CDP's Uniprint. Again, we recommend that the Technical Manager and System Administrator, along with Lab staff, make the decision to use one or the other. The major factors in both systems are:

- Does the Lab produce enough print jobs per year to warrant PRS?
- Does the Lab have sufficient space to house the Uniprint server along with an Allcard vending machine?

We recommend the Lab undertake a further analysis of these two questions in order to determine which printing solution would best serve Lab users and be most effective for Lab administration and operations.

4.2 Training and instructional recommendations

If this system is to be implemented, staff will need to be trained in its use and administration. David Cook of ACITS has proposed introductory training seminars for labs new to LabMan; the IT Lab should take advantage of these sessions. Training of new and existing Lab staff needs to be codified for consistency across personnel and time.

Education of Lab users on the new system is of paramount concern. Orientation sessions, informational handouts, and one-on-one training will be required. Finally, advance publicity announcing plans for the system and a timeline of implementation will ease the transition for most users.

4.3 Security recommendations

In light of the efforts of both Lance Hayden and the Intranet Team, we recommend that any decisions to implement firewalls or other network security protocols or systems use their recommendations as their bases. The combination of LabMan and the rebuilding programs provide sufficient authentication and authorization procedures for user access to resources. Additional network security recommendations are outside the scope of this project.

Physical security, while not an original focus of this project, stands out like a sore thumb in this Lab environment. We **strongly** recommend the addition of physical security devices in the IT Lab. Fiber optic loop alarms on individual workstations - especially mission critical and high value machines - should be installed at the earliest possible date. University maintenance have extensive experience installing such devices in campus labs, but have a long waiting period. The sooner the Lab decides to implement, the sooner they can be installed. Other security measures such as door alarms and / or motion detectors should also be seriously considered. In a best case scenario, we recommend both.

Fire detection and control devices are also essential. At the very least, smoke detectors should be installed as soon as possible. Other disaster preparedness and recovery planning procedures need to begin now. Finally, property and liability insurance are needed for the Lab. It is unlikely that any insurance quote can be given until a security audit of the Lab is conducted. This audit will likely be made by a representative of an insurance company or an outside consultant. If some security measures are undertaken and firmly implemented before such an audit begins, it will strengthen the Lab's ability to become sufficiently insured.

As with new hardware and software systems, Lab staff will need to be trained in opening, closing, and maintenance procedures once security measures are out into effect. Once staff is trained (and continues to be trained), they can better educate Lab users on security policies, procedures, and the rationale behind them.

4.4 Time frame

A 6-8 week timeframe is recommended for implementing lab management software. LabMan is easily installed, and testing it can begin as soon as possible. Tests of PC-Rdist and one or both Mac clients should start on at the same time, and will require one machine apiece. Once the client applications are tested and debugged, the remainder of the IT Lab's PCs and Macs may be added to the system.

Security implementations may take longer, depending on installation scheduling, security auditing, and training. For this reason we recommend starting as early as possible, and allowing an entire semester for the process. Simple physical security measures such as loop alarms should receive priority scheduling.

APPENDIXES

IT Lab Survey

Site visit reports

Technical resources

References

5 IT LAB SURVEY

5.1 Methodology

This study was undertaken to assist Dr. Mary Lynn Rice-Lively and her team in understanding user trends in the Information Technology (IT) Lab.

Between October 13 and 17, 129 IT Lab users were interviewed, both in the IT Lab and in several GSLIS classes, using a prepared pen-and-paper survey instrument. The survey sample represented 25% of GSLIS's enrollment (and hence, potential lab users).

The survey instrument focused on:

- ⊙ frequency of lab use;
- ⊙ platform preferences and reasons behind them;
- ⊙ waiting times to use equipment;
- ⊙ perceived ease-of-use of LabMan or similar systems;
- ⊙ scanner use;
- ⊙ applications used regularly by individual respondents;
- ⊙ areas of opportunity in learning or increasing application proficiency.

This survey illustrates current trends in IT Lab user opinion, and is supplemented with verbatim responses. Caution should be used when using these data to inform long-term planning or to generalize to a larger group.

5.2 Survey Instrument

1) How often do you use the IT Lab? (Check one.)

1. Every day 2. 2-3 times a week 3. About once a week
 4. 2-3 times a month 5. Hardly ever

2) When you work in the Lab, which computer platform do you prefer? (Check one.)

1. Macintosh 2. Windows 3. No preference

3) I have a list of possible reasons why you might prefer a particular platform. Please let me know which items on this list are applicable to you: (Check all that apply.)

1. I'm used to this platform 2. I learned to use this platform first
 3. I use this platform at work 4. I just don't like other platform
 5. I think more software is available on my preferred platform

4) Have you ever had to wait for your preferred type of machine in the IT Lab? (Check one.)

1. Yes 2. No

5) Below we've listed several software applications currently available in the IT lab. What applications do you regularly use in the Lab? Are there any that you would like to be able to use? (Check all that apply.)

	Use Now	Would Like to Use
Word Processing		
1. Corel WordPerfect (Win)	<input type="checkbox"/>	<input type="checkbox"/>
2. Corel WordPerfect (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
3. MS Word (Win)	<input type="checkbox"/>	<input type="checkbox"/>
4. MS Word (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
5. ClarisWorks (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
6. ClarisWorks (Win)	<input type="checkbox"/>	<input type="checkbox"/>
7. Other (please specify) _____		
Spreadsheet / Statistical		
8. MS Excel (Win)	<input type="checkbox"/>	<input type="checkbox"/>
9. MS Excel (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
10. SPSS (Win)	<input type="checkbox"/>	<input type="checkbox"/>
11. Other (please specify) _____		
Presentation		
12. MS PowerPoint (Win)	<input type="checkbox"/>	<input type="checkbox"/>
13. MS PowerPoint (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
14. Other (please specify) _____		
Database		
15. MS Access (Win)	<input type="checkbox"/>	<input type="checkbox"/>
16. FilemakerPro (Mac)	<input type="checkbox"/>	<input type="checkbox"/>
17. AskSam (Win)	<input type="checkbox"/>	<input type="checkbox"/>
18. Visual dBase 5.5 (Win)	<input type="checkbox"/>	<input type="checkbox"/>
19. FoxPro (Win)	<input type="checkbox"/>	<input type="checkbox"/>
20. Other (please specify) _____		

Productivity

21. MS Schedule (Win)
22. Other (please specify) _____

Internet Clients

23. Eudora (Win)
24. Browser: Netscape (Mac)
25. Browser: Netscape (Win);
26. Browser: MS Internet Explorer (Mac)
27. Browser: MS Internet Explorer (Win)
28. UTCAT/ UTAcess (either platform)
29. NewsWatcher (Mac)
30. Other (please specify) _____

Web Graphics Tools

31. L-View (Win)
32. Transparency (Mac)
33. Image map tools: WebMap (Mac)
34. Image map tools: MapEdit (Win)
35. GIF Builder (Mac)
36. Image Assistant (Mac)
37. Other (please specify) _____

HTML/Text Editors:

38. BBEdit Lite (Mac)
39. Simple Text (Mac)
40. HTML Editor (Mac)
41. HTML Pro (Mac)
42. Hot Dog Web Editor (Win)
43. Other (please specify) _____

Multimedia

- 44. Adobe Acrobat Writer (Mac)
- 45. Adobe Illustrator (Mac)
- 46. Adobe Premiere (Mac)
- 47. Adobe Photoshop (Mac)
- 48. MS Multimedia Viewer (Win)
- 49. Other (please specify) _____

Desktop Publishing

- 50. Pagemaker (Mac)
- 51. Other (please specify) _____

Library Software

- 52. Dialog Link (Win)
- 53. LEXIS-NEXIS (Win)
- 54. LEXIS-NEXIS (Mac)
- 55. Mitinet MARC (Win)
- 56. OCLC Passport (Win)
- 57. Other (please specify) _____

5a) Scanning

- 58. Flat-bed scanner (in the Mac Annex)

If yes, do you use it for: graphics text / OCR scanning

6) Are there any other applications you would use if they were available in the Lab?

7) Have you ever used a computer that required you to log in to the machine itself (such as those in the Student Microcomputer Facility)?

1. Yes 2. No

If so, was it: 1. Easy to use 2. No different than others 3. Difficult to use

5.3 Survey Results

Number of IT Lab Users surveyed: 126

Number of valid surveys: 119

Part One: General Questions

1. Frequency of Lab Use

Daily	2-3/week	1/week	2-3/month	Rarely
16	35	22	22	24

2. Preferred Platform

Mac	Win	None
27	81	10 (one respondent preferred UNIX)

3. Reason for Preference

Used to it	Learned it 1st	Use at work	Like it better	More SW
88	47	45	14	28

4. Waiting in IT Lab

Yes	No
33	83

5. Used LabMan or equiv. Before

No	Yes
52	63

(if yes) Ease of Use

Easy	No different	Hard
31	25	4

6. Scanner Use

Use Now Want to Learn

Scanner Used For

Graphics	Text/OCR
34	14

Part Two: Software Use

Word Processing Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
Corel WordPerfect (Win)	16	10	1	1	0	0
Corel WordPerfect (Mac)	1	3	2	3	0	0
MS Word (Win)	65	2	8	0	9	0
MS Word (Mac)	9	2	25	1	4	0
ClarisWorks (Mac)	3	4	9	3	1	0
ClarisWorks (Win)	2	10	1	1	0	0

Spreadsheet/Statistical Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
MS Excel (Win)	39	16	5	2	6	1
MS Excel (Mac)	4	2	13	7	3	1
SPSS (Win)	3	3	2	2	1	1

Presentation Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
MS PowerPoint (Win)	17	37	4	3	3	5
MS PowerPoint (Mac)	2	5	12	10	2	4

Database Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
MS Access (Win)	12	15	1	3	2	2
FileMaker Pro (Mac)	2	6	8	6	0	3
AskSam (Win)	1	7	0	1	0	2
Visual dBase 5.5 (Win)	2	8	0	1	1	1
FoxPro (Win)	2	8	0	0	0	0

Productivity Apps	Windows	Macintosh	Other
--------------------------	----------------	------------------	--------------

	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
MS Schedule (Win)	3	4	0	1	0	0

Internet Client Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
Browsers						
Netscape (Mac)	18	1	26	1	9	0
Netscape (Win)	66	5	9	0	9	0
MS Internet Explorer (Mac)	1	2	0	3	2	1
MS Internet Explorer (Win)	14	7	0	1	2	0
Other						
Eudora (Win)	29	10	7	2	3	1
UTCAT/ UTAcess (either)	65	3	20	0	8	0
NewsWatcher (Mac)	1	1	4	2	0	0

Web Graphics Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
L-View (Win)	8	6	0	2	0	2
Transparency (Mac)	1	2	2	6	0	2
Image map tools: WebMap (Mac)	2	3	2	6	0	3
Image map tools: MapEdit (Win)	4	12	0	3	0	3
GIF Builder (Mac)	1	5	1	7	0	2
Image Assistant (Mac)	0	4	0	4	0	0

HTML/Text Editor Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
BBEdit Lite (Mac)	2	1	5	4	0	1
Simple Text (Mac)	1	1	10	1	0	0
HTML Editor (Mac)	1	6	2	6	0	1
HTML Pro (Mac)	0	7	0	5	0	2
Hot Dog Web Editor (Win)	2	12	0	2	0	0

Multimedia Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
Adobe Acrobat Writer (Mac)	1	9	4	7	0	1
Adobe Illustrator (Mac)	1	8	2	9	0	2

Adobe Premiere (Mac)	0	5	2	5	0	0
Adobe Photoshop (Mac)	8	12	11	8	0	4
MS Multimedia Viewer (Win)	1	14	0	3	0	0

DTP Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
Pagemaker (Mac)	6	15	7	5	0	4

Library Apps	Windows		Macintosh		Other	
	Use Now	Like to Use	Use Now	Like to Use	Use Now	Like to Use
Dialog Link (Win)	32	16	8	4	6	1
Lexis-nexis (Win)	43	12	6	3	4	1
Lexis-nexis (Mac)	3	3	12	4	4	1
Mitinet MARC (Win)	10	7	0	5	5	1
OCLC Passport (Win)	17	11	7	3	3	2

Other Apps (Not on Survey)	Use Now	Like to Use
MS FrontPage	2	
Macromedia Director	2	
SecureShell	2	
Photoshop (win)	3	2
Eudora (Mac)	3	

Other Equipment Requests (verbatim)

- ⊙ QuickCam & software
- ⊙ PC scanner
- ⊙ Copy machine
- ⊙ Dot-matrix printers for the Mac annex
- ⊙ Programming packages

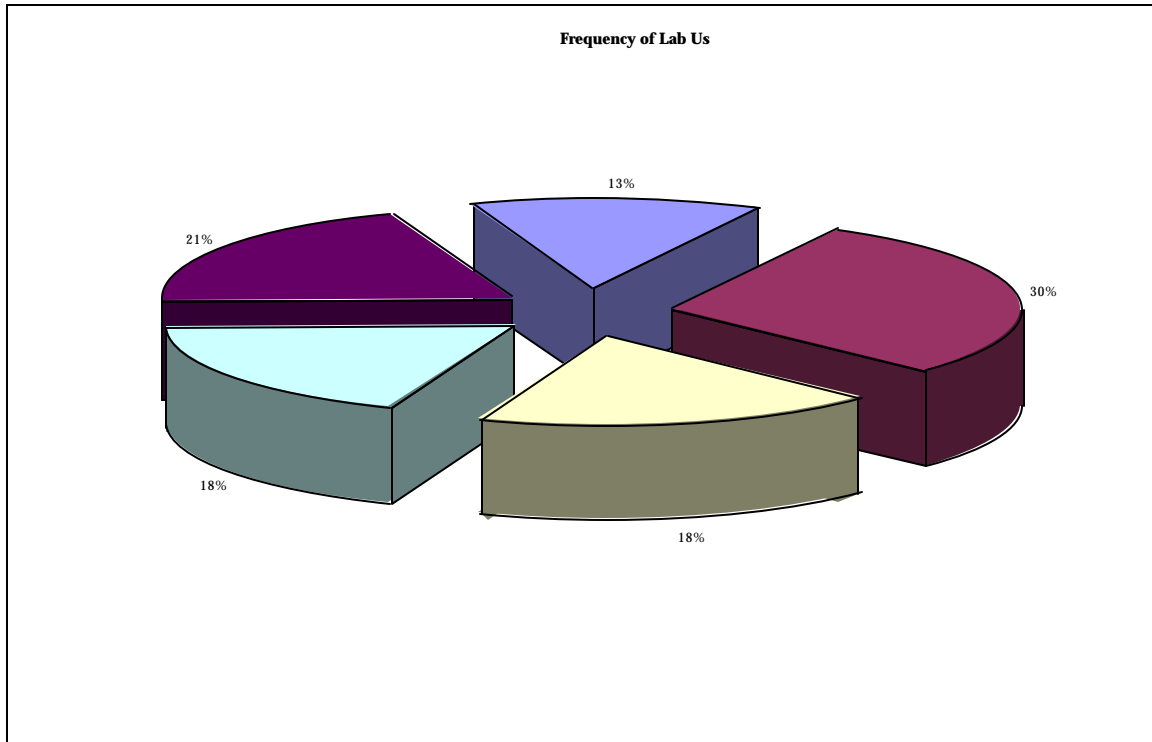


Figure 1: Frequency of Lab use. 43% of respondents use the Lab at least twice a week; 21% use it only “rarely.” Future surveys may want to ask how long students use the Lab each visit.

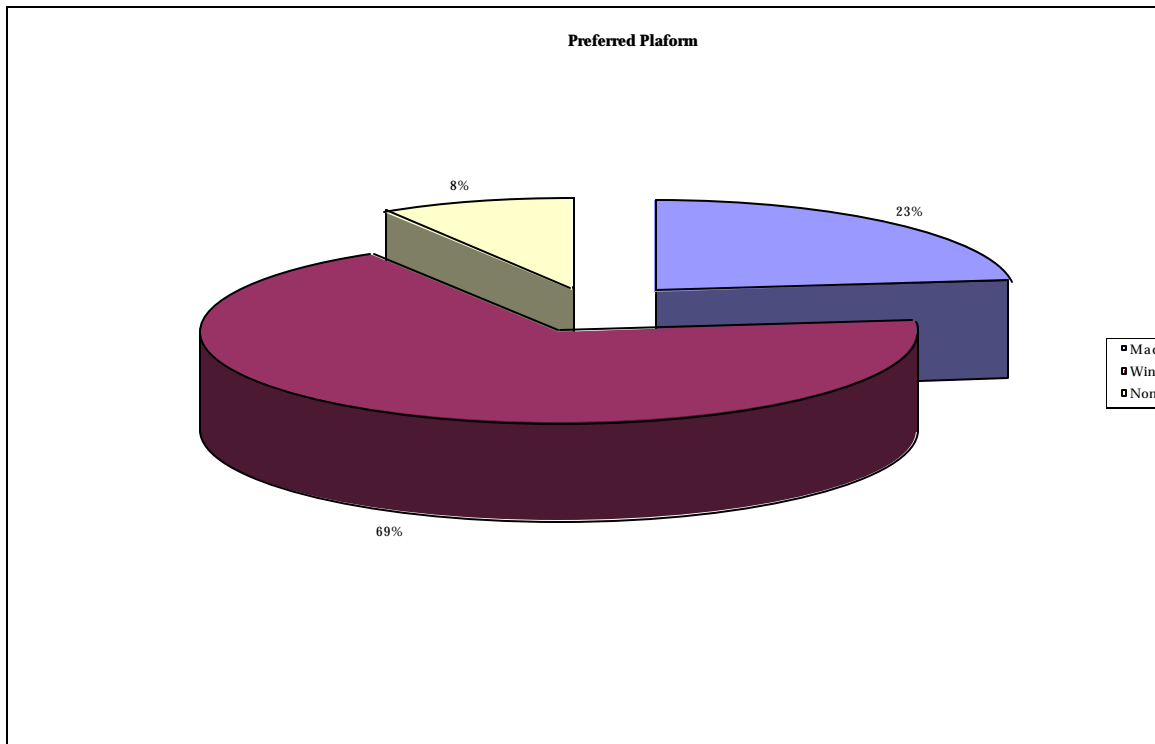


Figure 2: Preferred platform. 69% prefer Windows compared to 23% preferring Macintosh. Reasons for preference are not included in this graphic.

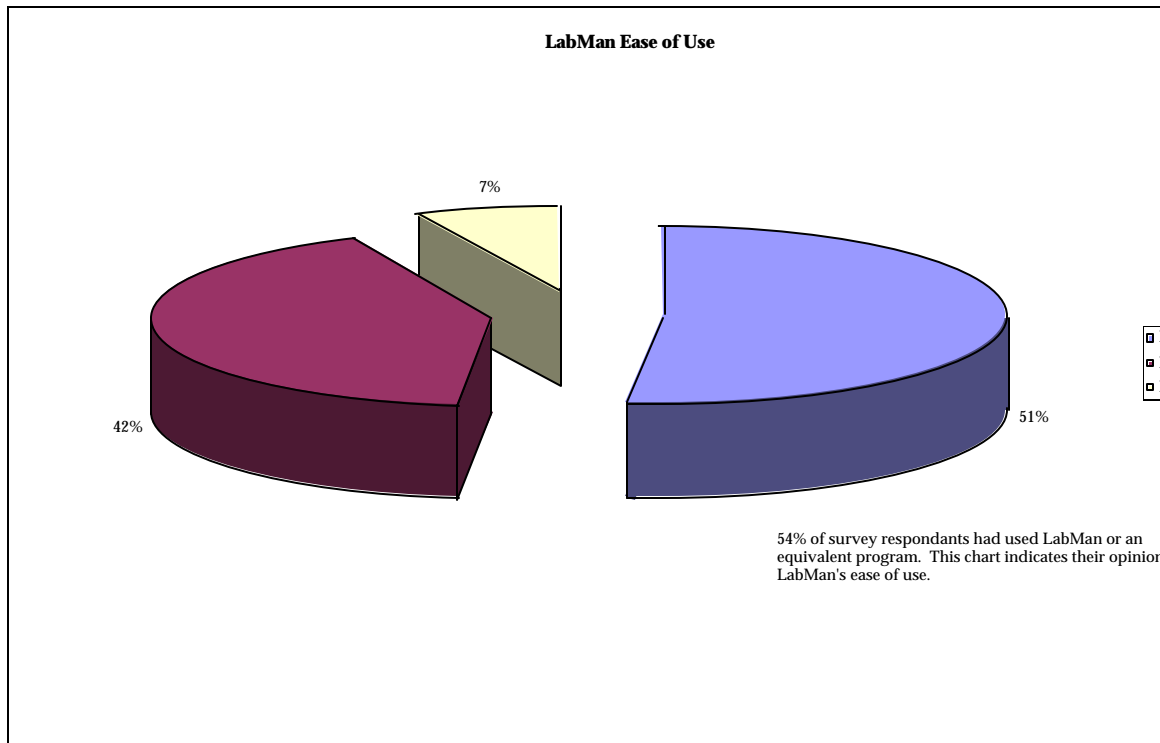
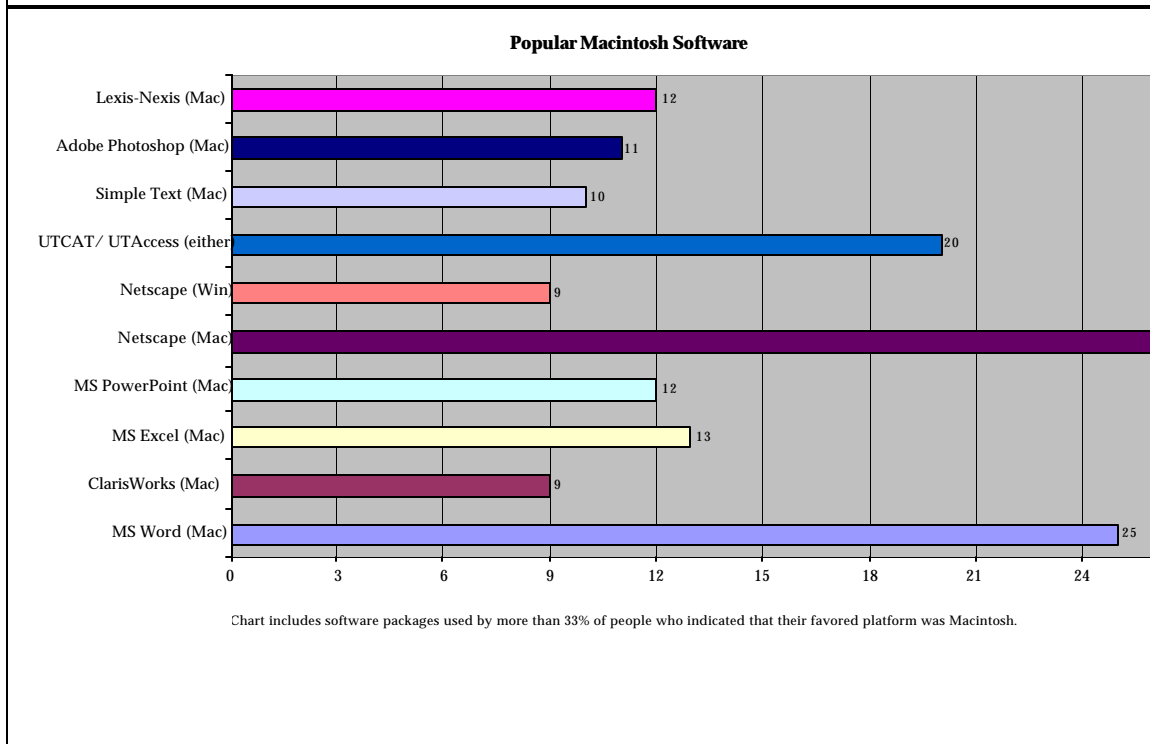
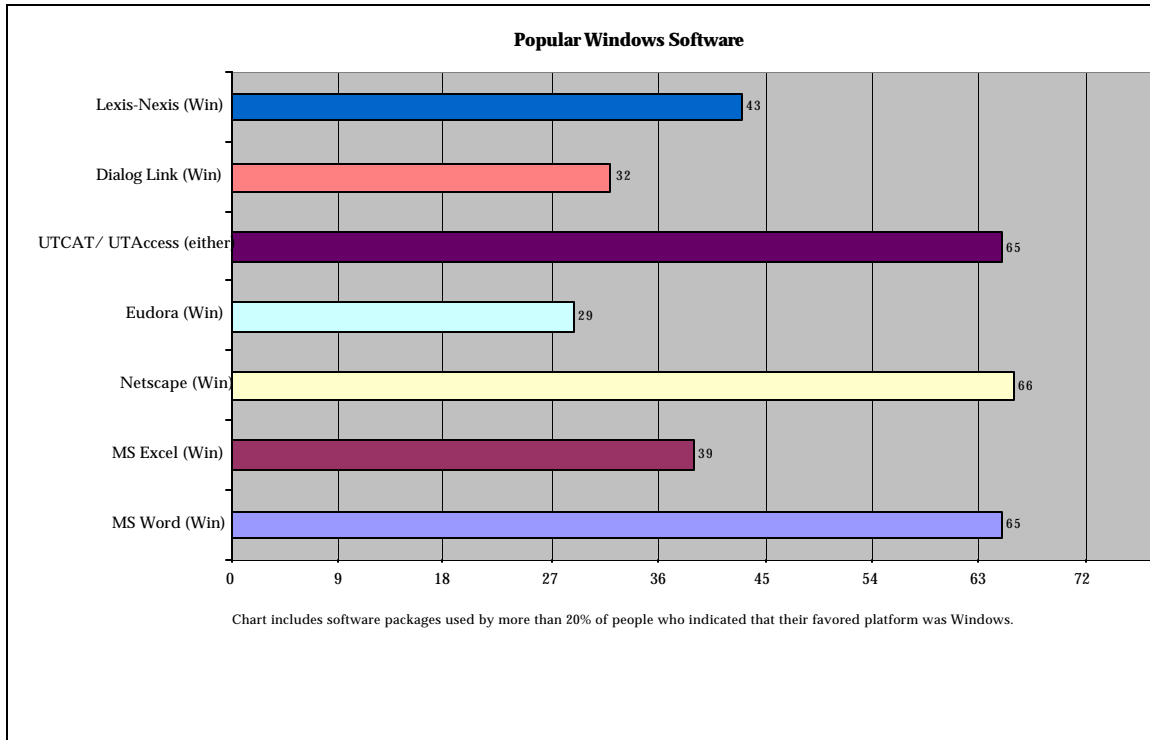


Figure 3: LabMan ease of use. Only 7% of responses indicated any difficulty with LabMan or similar systems.



Figures 4 and 5: Popular software for Windows and Mac. Netscape, Word, and UTCAT/UTACCESS were by far the most widely used on both platforms.

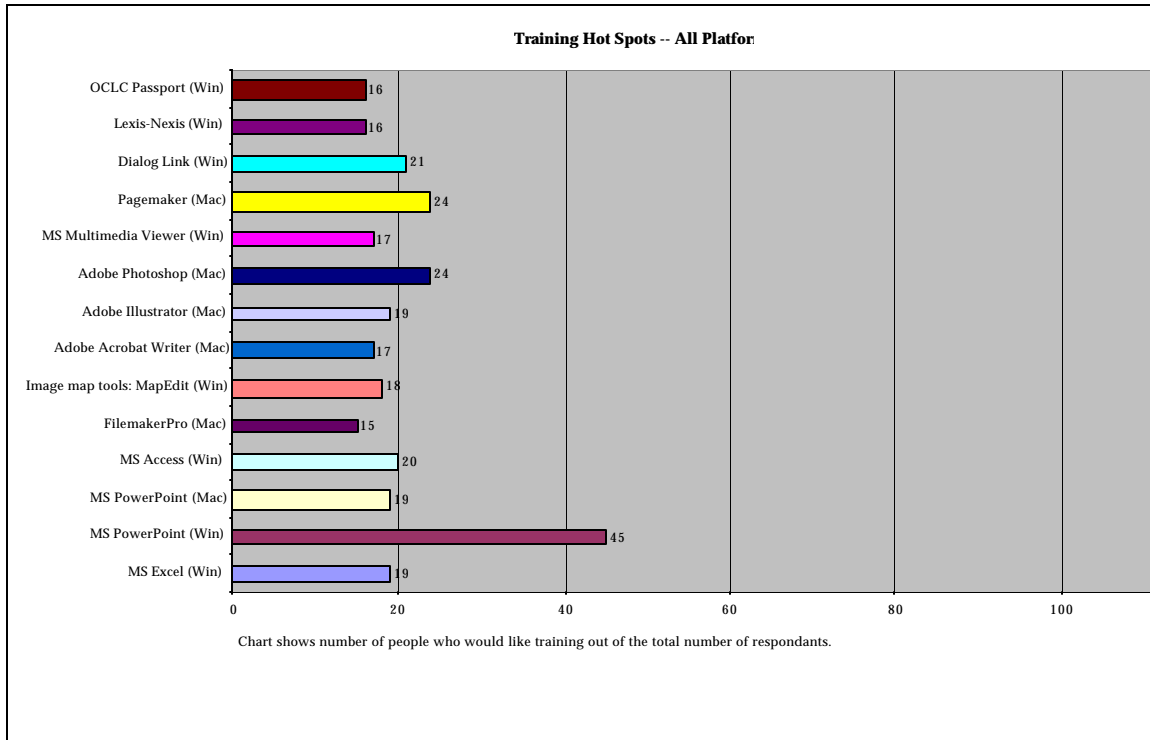


Figure 5: Training hotspots. Over half of the respondents indicated a desire for Powerpoint training, approximately one quarter indicated database training, other heavy responses included graphics applications, Web editors, and PageMaker.

6 LabMan Site Visit Descriptions

6.1 Learning Technology Center, College of Education

Interviewee: **Merry Wolf**, Microcomputer Applications Specialist (merryw@mail.utexas.edu)

November 12, 2-3 pm

Purpose of the Interview

Discussion of the LTC's experience implementing LabMan as well as network and physical security issues.

Background of the LTC

The lab fulfills the diverse needs of its student, faculty, and CoE staff users. A variety of applications are available in the LTC including statistical software (SPSS), collaborative software or "groupware" (Daedelus), multimedia applications for presentations and instructional materials (Hyperstudio). Internet clients and word processing software are available on every lab computer. Use is limited to the College of Education.

LTC and LabMan

The LTC planned their implementation of LabMan during Spring 1997. This laid the grounds for its summer session 1997 installation. This allowed time between semesters to test and debug the system.

LabMan, operating on a dedicated Dell Optiplex XL5100 model running Microsoft Windows NT Server v.3.51, manages usage of the LTC's approximately 80 Macintosh computers, models ranging from older 68K processor models to late model PowerPCs. The older Macs run Mac OS 7.1, the later models run OS 7.5.5. The LTC plans to upgrade the operating system to OS8 on those Macs capable of supporting it in the future, but no schedule has been set. The LTC's 15 PCs are not yet managed by LabMan, although their

addition to the system is in planning. RevRdist clients on all Macs handle desktop management, utilizing disk image mirrors on two servers (one Mac Quadra and one Apple Workgroup Server 8150).

LabMan has proven to save the LTC enormous amounts of time by allowing users to log themselves in autonomously. Before adopting LabMan LTC staff checked students in manually.

The LTC also uses Sassafras Key Server software to distribute certain applications with concurrent licenses. Sassafras meshes well with LabMan, managing the number of licensed copies of applications as LabMan manages computer usage.

Implementation problems and solutions

One problem that was encountered and solved during implementation involved the Mac Energy Saver (ES) Extension. During Mac System upgrades older versions of ES remained on machines, causing extension conflicts. As the ES cannot be removed from the System, the solution reached was to set the ES Control Panel to never activate ES's low power mode, effectively disabling the extension and solving the conflict.

Merry recommended against using CRON, a feature of LabMan which tracks users' idle time and displays. CRON is believed to have caused several minor but annoying problems while in use in the LTC. Since disabling this feature these (unspecified) problems have disappeared.

Print services

LTC does not use LabMan's Print Relay Services function. LTC does not produce enough laser print jobs to make PRS worth their time. Merry has not yet looked at the usage statistics generated by LabMan, but plans to do so "once she get five free minutes".

Security issues

CoE does not have a written security policy. The lab is equipped with intrusion alarms on the doors, fiber optic loop alarms on the computers, and motion detectors in all rooms of the LTC complex. Only LTC staff have keys to the labs. Access to the LTC is restricted to the hours of operation; even staff must leave after hours since the motion detectors and alarms are armed once the lab is closed.

Merry stated that the vast majority of their users are "low level" and therefore not high security concerns. The LTC has no network security protocols such as firewalls, since software access is regulated by LabMan, RevRdist, and Sassafras. Access to the lab's servers is restricted to staff, and users may not make changes to the server.

6.2 Computer Science Lab, T.S. Painter Hall

Interviewee: **Tjorn Smith** (tbone@cs.utexas.edu)

November 17, 2-4 pm

Purpose of the Interview

Luke Rosenberger had suggested visits to laboratories currently running LabMan, to get their perspective and allow them to recount anecdotal evidence of LabMan's success or failure as laboratory management software.

Luke had a list of information that he wanted us to compile, including:

- ⊙ hard facts about each lab -- number of users, number of support staff, hardware configuration, server specifications
- ⊙ history of the lab's experience with LabMan, for example: how long has LabMan been in use, why did the lab install it, how easy/difficult was it to configure, experiences with troubleshooting
- ⊙ information about the specific LabMan configuration used, and whether there were any LabMan features not used

- ⊙ similarities and differences in the machines controlled by LabMan
- ⊙ the types of users who would be using the lab and the general knowledge level expected
- ⊙ any advanced or esoteric applications of LabMan's functionality that have been implemented in the labs we visit

The Computer Science Laboratory

The Computer Science Laboratory (CS Lab) has roughly 130 machines, half Macintoshes running OS7 and half PCs running Windows 95. The Macintoshes are split into two homogeneous groups: PowerMac 7200 and PowerMac 9500. The PCs are also split, into three homogeneous groups: 15 486/66MHz machines, 20 Pentium 120MHz machines, and 35 Pentium 133MHz and 166MHz machines. Within each group, Tjorn has tried to make sure that the machines contain identical hardware and identical software. (Note: This is difficult in the best of times, as two machines with identical configurations bought from different suppliers can have different hardware and drivers "under the hood.").

The CS Lab also has four servers: one Macintosh, again running OS7, and three PCs running Windows NT 3.51. The Macintosh server is used as a license server for Macintosh software. The PC servers are divided as follows:

- One functions entirely as an "assignment" server, that is, students in the various classes submit their projects electronically, and they are stored and made available from this server.
- One functions as a LabMan server, and also controls the "rebuild" process for Macintoshes.
- One functions as a print server, and also controls the "rebuild" process for PCs.

The CS Lab will be getting two more servers in the near future, which will be used (among other things) for mirroring critical data. Tjorn has stated that the network that supports the CS Lab cannot be "down" for more than half a day.

Manpower seems to be scarce -- Tjorn is *the* full time employee responsible for the CS Lab. There are also seven part-time User Assistants, one of whom is always on-duty during lab hours. The lab itself is split into four separate rooms in Painter Hall.

CS Lab Users

The CS Lab is available for the 2400+ students taking undergraduate computer science courses. The user base is extremely technologically literate, and the nature of their assignments requires that they have the freedom to install and run new software, change system settings, add user libraries to existing programs, and in general, alter the system in ways that LabMan usually doesn't allow.

As an interesting note, the CS Lab is exclusively for undergraduates. The graduate-level students in Computer Science have their own lab, which consist mostly of UNIX workstations.

LabMan in the CS Lab

In many ways, the CS Lab is in a unique position. They need more flexibility in how LabMan is configured, since their users primarily tend to be writing and running programs. LabMan can restrict the programs that can be run. The CS Lab users need to be able to run many different programs, including ones that can potentially be used to reconfigure the system, such as ResEdit on the Macintosh.

On the other hand, the CS department provides no lack of talent and resources to test new LabMan configurations for potential "loopholes." Tjorn has set up a series of machines which are used to run test configurations, and he has several volunteers who try to break the system.

Tjorn finds several benefits in using LabMan. Primarily, he says, "it makes the CS Lab possible." Software installations and upgrades alone would take more time than he has if LabMan were not installed. Second, LabMan's logging functions allow the CS Lab to establish accountability should any untoward

incidents occur. Third, LabMan's administrative functions allow Tjorn to control each machine's status from his office.

The CS Lab's unique set of resources have allowed a greater degree of freedom in "fine-tuning" the operation of LabMan and associated disk restoration utilities. Tjorn commented that he has been "tweaking Windows and PC-Rdist and RevRdist to work fully together." In addition, the CS Lab is experimenting with advanced applications of LabMan's functionality, such as creating disk image overlays.

LabMan Drawbacks

1. The disk restoration utilities require machines whose software and hardware are quite homogeneous. This effort toward homogeneity was mentioned in the CS Lab description on page 1.
2. By default, LabMan ID and password validation are tied to UT's Print Relay System (PRS). This default can be bypassed, but the bypass is complex and non-intuitive. Since the CS Lab has rigorous requirements for continuous availability and PRS validation happens via a network connection, Tjorn has found it necessary to bypass the default validation scheme and establish their own.
3. The LabMan interface itself has many shortcomings. Everyday tasks require non-intuitive sets of actions, and lab-specific tasks are more complex yet.

Disk Restoration Utilities in the CS Lab

LabMan has no capabilities for disk restoration, although that is an important part of having an adequately managed computer lab. It does, however, allow the administrator to specify a third-party disk restoration utility. Tjorn commented on three of these utilities: PC-Rdist, RevRdist, and Assimilator.

PC Software

According to Tjorn, PC-Rdist is *the* choice for PC disk restoration. The only other option, has not been supported in some time, and no new development has been scheduled to make it competitive with PC-Rdist.

The new version of PC-Rdist has the capabilities of a programming language, as it is extensible with JavaScript. It promises much more flexibility in situations where the basic functionality of PC-Rdist is not sufficient.

Macintosh Software

The choice of Macintosh disk restoration utilities is not quite so clear-cut. The utility most often used in other software labs around campus, RevRdist, is also in use at the CS Lab. Tjorn found a number of design issues with RevRdist, and commented in the interview that the product is "slow and flaky." RevRdist also does not restore the Desktop database (a part of the Macintosh system that determines file associations, extension settings, and the appearance of the Mac Desktop). LabMan has been enhanced to work around this limitation in RevRdist.

Tjorn is currently evaluating another utility, Assimilator, as a possible replacement for RevRdist. He has found, however, that Assimilator has its own set of quirks. It is more usable than RevRdist and runs cleanly on the new OS8, but it provides less functionality than RevRdist.

One problem that RevRdist and Assimilator share is ongoing file system corruption. These utilities only replace the files in their restoration activities; they do not replace the file system. Periodic maintenance is needed; the administrator must boot each Macintosh from a floppy to rebuild the file system.

6.3 Electronic Information Center (EIC), Undergraduate Library

Interviewee: **Damon Jaggars**, Head of Reference and Information Services (jaggars@mail.utexas.edu)

November 17, 3-4 pm

Purpose of the Interview

We hoped to answer a few lingering questions about hardware and network configurations, print service alternatives, and insights into Mac OS8 implementations.

EIC Background

The Electronic Information Center serves as a research laboratory for users of the Undergraduate Library. Macintosh (OS 7.6.1) and PC (Win95 and 3.1) computers are available for public use. Consoles for staff use run Windows NT Workstation. Applications include Internet clients and the General Libraries' CD-ROM server. Word processing software is intentionally omitted to keep the workstations for research rather than as general purpose computers (which are available elsewhere in the UGL and upstairs at the SMF).

LabMan at the EIC

EIC has been using LabMan, PC-Rdist, and RevRdist for about one and a half years. LabMan runs on a dedicated 133MHz Pentium machine running Windows NT Server in a back office. The server also houses the PC and Mac disk image mirrors. Damon explained that the disk images took up very little hard drive space even though several of the EIC's machines used different monitor drivers. The differences were handled by station-specific registry files rather than complete disk images.

One caveat about the LabMan server is that it needs to be dedicated to lab management and nothing else. In many cases a server could fulfill other functions such as file or print server for a network, but the frequency of two-way communication between Mac clients and the server can present problems including slow responses of the LabMan system and even system crashes.

The rebuilding clients have worked with little or no difficulty. In fact, the Windows 95 implementation of PC-Rdist runs almost seamlessly, and the EIC hopes to migrate or replace the last of its Windows 3.1 stations to take advantage of this. Despite its occasional "clunkiness" RevRdist was chosen over Assimilator. When the Computer Science Labs' success with Assimilator was mentioned Damon commented that Tjorn was able to tweak the source code of Assimilator, something beyond his or any of the EIC staff's capabilities.

When asked if problems with the CRON feature of LabMan had caused them to disable it, Damon answered "yes and no." The EIC had set up CRON in such a way that it did not notify the user of idle time but still tracked it. The only aspect of CRON needed in the EIC is the function that tracks use time on the EIC's designated 30-minute "express" stations; it then passes that information off to another customized feature that notifies the user of their elapsed time at 25 minutes, 28 minutes, and then begins notifying the user every 30 seconds after the 30 minutes have expired. We decided to call this "management by annoyance."

Print Services

The EIC uses Uniprint rather than PRS. Print jobs are handled by a (rather large) print server running Windows 3.1 and connected to a Hewlett Packard LaserJet 5si similar to the one used by the GSLIS IT Lab. Damon explained that photocopying revenues had dropped in recent years, but since installing Uniprint much of those revenues have been restored. The advantage of Uniprint here is lower overall cost per print job over PRS.

Staffing issues

Until recently Damon shared functions of administration of the EIC's network with Darby Syrkin. Darby has left the EIC, and Damon is at present doing two jobs.

7 Technical Resources

7.1 Cornell University Implementation Notes

(Taken from the ATS Lab homepage, by Pat Washburn, <http://128.253.152.16/windows95/Details.htm>)

This section will detail exactly what I am doing in my labs. There are certainly several other possibilities.

The steps to implement PC-Rdist in a lab:

1. Set up a new Master Image
 - ⊙ Server Issues
 - ⊙ Creating the Master Image
 - ⊙ Creating the Master Registry
 - ⊙ Copying the Master Image to the Server
2. Configure PC-Rdist
 - ⊙ Distribution Files
 - ⊙ Custom Registry Files
3. Add Clients
4. Update the Master Image as needed
5. PC-Rdist Tips and Tricks
6. Recover from client problems - Visit the Windows95 Problem Resolution Guide.

Setting up a new Master Image

1. Server Issues: Most of the server issues are pretty obvious -
 - ⊙ The server must have sufficient drive space for your Master Image, your Master Registry, and the additional custom registry files that will be required.
 - ⊙ The server must support Long File Names. On NetWare servers, OS2 name space must be added to the volume that will contain your Master Image.
 - ⊙ The server must be available for the client to logon. (I know, that sounds really basic, but our NetWare/IP issues have made me more aware of just being able to log on to the server as an issue.)
2. Creating the Master Image:
 - ⊙ Simple - just make 1 machine look EXACTLY like you want all of them to look. Load all of the software, make all of the correct file associations, set up all of the hardware, logon to the server as the generic user, set up the Start Menu, arrange windows and icons as you want them to appear, etc.
 - ⊙ Watch out for those .PWL files! Windows95 will make a .PWL file by default whenever you logon as a new user. If you log on as a Supervisor or Admin user, Windows95 may save the password, and use that password to automatically login if someone else tries to login as Supervisor.
 - ⊙ Apply any Service Packs and Updaters to the Operating System and Applications.
 - ⊙ Launch each application to clear any license or registration screen , and to disable any automatic tutorials.
 - ⊙ If any network components need to be deleted, make certain to reboot before adding any others. The order in which network components are added will affect their location in the Registry, and can cause problems when you set up other machines to run from this Master Image and Master Registry.
3. Creating the Master Registry: Again, this is pretty easy.
 - ⊙ Start REGEDIT from a DOS prompt. This keeps Regedit out of the Start Menu's Run... list.
 - ⊙ From Regedit, export the registry to the Desktop, and name it Master.reg (or whatever you prefer).
 - ⊙ Quit Regedit.

4. Copying the Master Image to the Server: I have a Win95 folder that contains a Registry folder and an Image folder.
 - ⊙ Log on to the server as a user with Write access to the PC-Rdist volume. Since you have already copied the Master Registry, it will not contain any information on your Supervisor login ID.
 - ⊙ Move the Master.reg from the Desktop to the Win95\Registry folder.
 - ⊙ Set View... Options... View tab to display ALL files.
 - ⊙ Copy all files and folders from the C: drive to the server volume EXCEPT the Windows folder.
 - ⊙ Once that copy is completed, Create a Windows folder on the server manually. Open the Windows folder on the C: drive, and Select All. Press the Ctrl key, and click on the Windows Swap file (Win386.swp) to de-select it. Drag all remaining selections to the server's Windows folder.

Configuring PC-Rdist

Distribution Files: PC-Rdist has an astounding number of options that can be configured through the distribution files. I create a custom distribution files (called, in my case, logout.dst) for each machine. Others create a single distribution file that works for all machines. Both methods have their good and bad points. This is a copy of the distribution file that I use on all of the machines that I am administering. The only changes that I make are for the lab name and machine number in the paths to the Master Image, Master Registry, custom Logout.Dst, and custom Registry files.

Custom Registry Files: All items in Italics will need to be modified for each machine. There may be other items that you will need to deal with. This is the minimum set of registry keys that I need to customize for each machine.

Note: the exact path of the last key will vary depending on the network components that you are using, and the order in which they are loaded.

REGEDIT4

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\ComputerName\ComputerName]
"ComputerName"="machinename"
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MSTCP]
"HostName"="machinename"
```

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Class\NetTrans\0003]
"IPAddress"="IP address"
```

Updating the Master Image: Adding new software to the Master Image is relatively easy. (At least it beats adding new software to 80+ machines by hand....) The best technique that I have come up with is to run PC-Rdist on a machine to clean up any irregularities. Once PC-Rdist is complete, run a virus check, and clean up any problems that may exist.

Then, install the new software, and run PC-Rdist again using the -n switch. To do this, start a command prompt, switch to your pcrdist directory, and type: pcrdist distfile -n, where distfile is the name of your distribution file, logout.dst in my case.

Now, if you print out the Distribution Log file, you will have an exact record of all files, folders and registry keys that PC-Rdist modified. Several keys and files are replaced every time that PC-Rdist runs, so you will learn to ignore them. I usually edit the log file before I print it out to remove any extra junk that I don't need to deal with.

With the Distribution Log in hand, create a new Master Registry if the software adds or changes any Registry keys. Then, log in to the server, rename the old Master Registry and copy up the new Master Registry, if needed, and copy all of the files and folders into the correct locations on the server's Master Image.

Run PC-Rdist on another machine, and test the new software to make sure that it works correctly. If it doesn't you may have missed a file. Review the Distribution Log, and add any files that didn't get copied up the first time. Once everything works on the second machine, logout and PC-Rdist the first machine, and turn it back over to the users.

PC-Rdist Tips and Tricks:

- ⊙ Start Regedit from a DOS shell so it doesn't appear in the Start Menu's Run... list.
- ⊙ Associate .DST files with NotePad on your office machine. Updates are much easier, and you won't accidentally launch PC-Rdist.
- ⊙ Make frequent backups of your office PC's Registry in case you accidentally double click on one of those .REG files trying to edit it.
- ⊙ Edit .REG files by Right Clicking on them and choosing Edit from the Pop up menu.
- ⊙ Policy Editor (POLEDIT) can do some useful things, check it out.
- ⊙ Microsoft's Power Toys (especially TweakUI) can also be helpful.
- ⊙ Restart Windows quickly by choosing Restart from the ShutDown Menu, and holding down the Shift key when you click on OK.
- ⊙ Holding the Shift key when Windows starts up will bypass the StartUp Programs.

7.1.1 PC-Rdist sample distribution script

Refer to the PC-Rdist documentation for more information.

```
#Set up the configuration
tempdir=c:\temp
expire=1 days
logfile=c:\temp\distlog.txt
log=all
freespace=10240
timestamp=c:\pcrdist\pcrdist.tsp
flatten=no
filetime=mtime
makeroom=oldest
boot=warm
cbreak=no
logmax=0
#mount the server volume
#(Insert your server name and volume name)
+mount \\server\volume j:
#Update the file system
> c:\j:\win95\image\lab : m s t j b
> windows
# Ignore the registry files and Windows Swap file
| user.dat : ic is
| system.dat : ic is
```

```

| win386.swp : ic
<
> pcrdist
#copy the custom logout.dst file for this machine
| logout.dst j:\win95\registry\lab\#\#\logout.dst : m s t w
<
> tmp
#delete the Kerberos ticket, if it exists
| Ticket.krb : d
<
> bawin
  > winba
    > netscape
      #delete all files from the NetScape Cache
        > cache : d
          <
          <
          <
          <
          <
          <
#Update the Registry, ignoring the Enum section
#which contains Plug and Play information
regpath=j:\win95\registry\lab;j:\win95\registry\lab\#\#
> registry master.reg ##.reg : m s c d
  > HKEY_LOCAL_MACHINE
    > Enum : i
    <
    <
    <

```

7.2 Kent State University Implementation Notes

(Taken from Steve Tapp, <http://act.kent.edu/stapp/win95rem.htm>)

Using PC-RDIST to synchronize the desktops

As stated previously, PC-RDIST is a W95 program that synchronizes the files on a local hard drive with a copy of the files on a file server. It works by comparing the directory information on both machines and copying or deleting files as needed to insure that the file system on the workstation exactly matches the files on the server. By running PC-RDIST in the NetWare login script, the desktop file maintenance can be automated. How does one set this up? Let's get started:

- ⊙ Manually setup a typical lab machine with W95, network client, applications, TCP/IP stack, screen savers, virus scanners, printer drivers, desktop settings, policies, etc. Start with the base install of W95 you created earlier. You can run applications from the server or install them on the local hard drives. I use a mix in our lab. It is helpful to dual boot with DOS and Novell VLMs as

was done with the Ghost image. It comes in handy later if W95 get trashed and the entire machine must be rebuilt. I found it instructive to keep track of the files installed and registry changes made by each application. Make a note of any settings that are different for each machine such as IP address and login name.

- ⊙ Prepare the Novell file server by adding long file name support. Add OS2 (NetWare 3.x) or LONG (NetWare 4.x) name space to the NetWare volume to hold the desktop image.
- ⊙ Run PC-RDIST to copy the files from the workstation to the server. Note that the registry files and swap files are ignored.

Custom registry settings for each desktop

Some additional notes on the registry are in order. There are two issues: how to copy the registry files and how to update them. As mentioned above, only the backup registry files are copied to the server (*.DAK). The autoexec.bat file on the workstation looks for and runs the bat file regupdat.bat. This bat file copies the backup registry files over the actual registry files before the W95 GUI loads and customized the registry for the particular machine. There is a separate registry update file for each machine. PC-RDIST has commands to update the registry as well if you prefer a different method.

The registry updates are located in the machine.reg file customized for each machine. The DOS version of regedit applies the updates after the reg files are copied and before W95 loads. One of the big reason for the registry updates is the IP address. If you have a DHCP server, you can configure Microsoft's TCP/IP stack to use the DHCP server instead. I've used both methods with equal success. The DHCP server is one more thing to break and I was relying on a server not under my control. Just call me paranoid. Other useful registry edits include the machine name, workgroup, and default user name.

Sequence of operations during a complete rebuild

- ⊙ PC boots to DOS (Dual boot or floppy)
- ⊙ Autoexec.bat loads VLMS
- ⊙ Login to Novell server
- ⊙ System login script exits to BUILD95.BAT
- ⊙ Build95 detects DOS and runs Ghost
- ⊙ When Ghost finishes, Build95.bat reboots machine
- ⊙ PC reboots and load base install of Win95
- ⊙ Login to server with 32 bit network client
- ⊙ System login script exits to BUILD95.BAT
- ⊙ Build95 detects Win95 and runs PC-RDist
- ⊙ PC-RDist copies final image and registry updates
- ⊙ PC-RDist reboots PC when finished
- ⊙ Autoexec.bat runs REGUPDAT.BAT
- ⊙ Regupdat copies and updates the registry, renames itself
- ⊙ Final image of Win95 loads
- ⊙ PC logs into file server
- ⊙ System login script exits to BUILD95.BAT
- ⊙ Build95 detects Win95 and runs PC-RDist
- ⊙ PC-RDist detects it just ran and exits

- ⊙ Lab users begin to trash machine...

Note that each morning PC-RDist only has to touch up the desktop image and the entire process completes in under 5 minutes.

Using PC-RDist with multiple image files

The lab manager with a single lab equipped with identical machines should count his/her blessings daily. For those with multiple labs and numerous hardware platforms to support, storing complete images for each configuration will consume large amounts of server disk space.

One approach is to install and run applications from the server. This reduces the size of the workstation image substantially. However it also creates network traffic, loads the server, and creates a central point of failure. If you have a fast and/or lightly loaded network, this is a good option.

Another approach is to copy files common to all machines (such as application files) to a shared image directory. Create separate image directories for each platform containing only machine specific files. The Root, DOS, and windows directory should contain all system specific files.

During a rebuild, PC-RDist can download the machine specific files and the shared application files creating a complete desktop image. This will save disk space and time uploading images.

A Note to the Lab consultants on rebuilding the W95 desktop

As you know, the lab machines rebuild themselves each morning or when you click the rebuild icon under system tools. What this does is synchronize the files on the local hard drive with a copy of the files on the server. The program PCRDIST copies down any files that have been erased or modified and erases any files on the local drive not on the file server. This works very well but fails in two situations:

- ⊙ The file contents are corrupted. PC-RDist works by comparing the directory info only (file name, size, date, time, attributes). It does not look at the file contents. Therefore if a file gets corrupted but the directory info stays the same, PC-RDist will not recopy the file. The fix is to erase the offending file and run rebuild which will recopy the missing file. The trick is to know which file is corrupted and needs replaced. A common culprit is WordPerfect. If you just cant get WP to run, try erasing the c:\office directory and running rebuild. I've also seen corrupted hp printer drivers. If you get printer errors when trying to print from windows programs, erase c:\windows\hp*. * and run rebuild. Fortunately corrupted files don't happen that often. Why doesn't PC-RDist scan the file contents? It can but the scan takes as long as recopying all the files and is not practical. It would take the machines over a half hour each to boot in the morning not to mention the load on the network and file server.
- ⊙ The PC will not start windows and login to the file server. PC-RDist is a windows program. The PC must start windows and login to the file server before PC-RDist can do its work. Occasionally, a lab machine will be hosed to the point that windows will not start. We've had users format the hard drive, erase the windows directory, etc. If windows will not start, then its time for a scorched earth rebuild.

A scorched earth rebuild completely reloads the hard drive from scratch. The hard drive is reformatted and all files recopied. It is important not to skip any of the steps described below.

1) *Boot the PC with DOS*

Boot the PC to the DOS prompt. The lab machines are setup to dual boot DOS or Win95. Turn on the PC and when you see "Starting Windows 95..." press the F4 key. DOS will start and load the network drivers to login to the file server. If the hard drive is reformatted or something equally as sinister, get a copy of the rebuild diskette and boot the machine from the floppy. This will require a CMOS change to allow booting from floppy. Don't forget to change the CMOS settings back when finished!

2) *Load a base install of windows.*

Once DOS and the network shells are loaded, login to the file server using the account assigned to the machine. The boot diskette logs in automatically without needing a password. Ghost runs from the network login script and copies a minimal install of Win95 to the hard drive. This should take 2 or 3 minutes. When Ghost is finished, it will beep. The machine will log off the network and reboot.

3) *Reboot and use PC-RDist to copy the rest of the files*

Once rebooted, the machine will load Win95 and pop up the network login screen. Login with the usual account for the machine. Now the normal rebuild (pcrdist) will start and copy all the files needed to complete the desktop. This copy takes around 15 minutes to complete during a rebuild compared to 3 minutes normally. .

4) *Reboot and load final Win95 desktop*

When the copy is finished, PC-RDist will reboot the machine. During this last reboot, the registry files are updated, the final desktop is loaded and the machine is ready for use. Is this cool or what?!

7.3 RevRdist Configuration Information

This information came from a Hahnemann University lab administrator's guide designed for setting up and using RevRdist, including initial configuration strategies, testing, debugging, and maintenance. The information was downloaded from <http://www.purdue.edu/revrdist/contrib/hahnemann-revrdist-guide.html>.

Utilities Used in the Lab

ShowStartup

This is an extension that shows the Startup Items folder. The Finder will not run items included in the Startup Items folder if the folder is set to 'hide.' This extension will briefly make the folder visible until, HideStartup, it's companion application that resides in the Startup Items folder hides it again.

`ftp://ftp.cc.purdue.edu/pub/mac/Hide_Show.sea.hqx`

RRHelper

Is a RevRdist related extension that keeps PRAM settings, default printer, volumes mounted at startup with password, etc.

`ftp://ftp.cc.purdue.edu/pub/mac/testing/Build_RR_Helper1b42.hqx`

ZapRRtempFiles

This is a system extension that removes RevRdist temp files on startup. It is located at:

`ftp://ftp.cc.purdue.edu/pub/mac/testing`

NiceStopShutdown

Is an extension that Restarts the Mac if a user selects Shutdown.

RealEject

Is a control panel that traps the Eject command in the Specials menu replacing it's function with Put Aways function. This avoids the 'ghost' disk image on the desktop and "Please insert disk..." alert. (Shareware.)

QuitIt

This is a control panel that warns users with an alert box if they close the last application window. (Shareware.)

Making Changes

Listed below is a strategy used in configuring RevRdist. This short section explains how to make software changes to Macs running RevRdist.

1. To add software that will run on all lab Macs add it to the Master2 folder on the rArchive server.
2. To add software on a particular family of Macs, add the item to the Specials folder on the rArchive server and call it from that Mac family's dist file using the renaming string after the action for the item. Here's an example:

| some application :A. =rArchive:Specials:SpecialApps:some application

3. The same routine will hold if you want to remove an item from lab Macs. Remove it from the Master2 folder (for all) or delete or 'comment-out' (preface line with '#') the referencing line in the dist file of one that is copied from the Specials folder.

Strategy

It seems that many lab administrators create unique System Folders for each Mac type and configuration or call all machines by the same name.

Create a Master Folder (Master2) on the rArchive server using the minimum number of applications. That is, all the lab machines should have these applications. The System Folder also should have control panels, extensions, preferences, etc. for the minimum applications. The System and Finder are from the most prevalent machine in the lab.

Create a Specials folder which contains all the special applications, Systems, extensions, control panels and RevRdist items (app, prefs, RR Helper.) The renaming function is used to copy these items to specific Macs. This is the same technique used to copy a different System and Finder while ignoring/deleting the one in the Master2 folder.

The advantages of this method include the ability to install a new hard drive or machine and start up with a network startup disk which will allow you to drag over the items in the Master2 folder. A specific System and Finder (and Enabler, if necessary) would replace the minimum System Folder items.

Families of computers using the same Ethernet driver, which is installed as a resource in the System file, need to copy the relevant System file. When resources are installed directly into the System file you need to have that System file copied uniquely or the destination Mac will not find pertinent resources like Ethernet drivers and default printers.

If you are using RR Helper (settings and PRAM system extension) this is also true. If the proper System hasn't been copied over in a RevRdist run it will complain that a resource wasn't found "contact a consultant." It must have a unique System (and RR Helper ext.) to boot correctly.

Unique configurations of applications require unique Dist Files. Ideally, there should be unique RR Helper INITs for each lab machine if servers need to be mounted with unique usernames and passwords. Most lab administrators seem to call their machines Macintosh HD and log on to servers with the same name and password. RR Helper works well for this. The method used here is to copy an RR Helper extension to an individual Mac and mark the dist file with CiHs (ignore it if it is on a client but not on the master and hide it.) RRHelper extensions are located in an archive folder in the Specials folder.

Protection of Settings

Since RevRdist and its prefs are located in the Startup Items folder they are vulnerable to deletion or tampering. There are two utilities (Show Startup-INIT and Hide Startup-APP) that are used to protect the RevRdist Prefs and application. The extensions (RR Helper and !Show Startup) can get 'Hs' included in their action codes in a Dist File to hide them. The Finder requires that Startup Items be visible to launch. The combination Show/Hide will make the folder briefly visible, items will launch and the folder will be hidden.

Dist File Read Me Section

First, trash or disable any automatic virus checking programs on the client Macs. Resources are changed and files/extensions are modified in the setup process.

Quick Start

Set up a lab Mac to your specifications. Pick one that will get the most applications, extensions, etc. and is a prevalent Mac.

- 1) Disconnect from network.
- 2) Install System 'for Any Macintosh'.
- 3) Install all desired applications, extensions, etc..
- 4) Make the application aliases for Apple Menu, etc. (in order for RevRdist to resolve aliases they must be local, hence the network disconnection.)
- 5) Create a Master folder on a server available to lab client Macs and copy the client Mac folders to this folder.
- 6) Create a floppy disk with RevRdist, Build RR Helper and a RevRdist Prefs file for demand running of RevRdist and creating the RR Helper extension.
- 7) Run RevRdist on test client Mac in "Verbose/List only, don't do" mode to watch what may occur in the Activity window. Instructions for running RevRdist follow:

Flags Dialog Window

Open RevRdist on a local machine and the Flags Dialog window will appear. For test runs use the "List only, don't do" flag and other defaults. (Note: flags are saved in RevRdist application, not in a prefs file.)

For a "demand run," drag a fresh copy of RevRdist (with default flags) and a generic Prefs file (which connects to the Masterarchive specifying partition - rArchive - and a distfile) to the Mac desktop from a floppy. The flags for a real test run are Flags Dialog; Quit other application; and uncheck "use prefs mod time" and all other flags. Some lab administrators just enter the Mac username and password in the Prefs file. See Debugging Note below for making sense of the List Only Activity window items.

The copy of RevRdist placed in the Startup Items folder of the Master System Folder is set to "Quit Other Programs; Use prefs mod time; Restart Mac When Done; and Run unattended." The last flag is set automatically when placed in the Startup Items folder.

Setting Prefs Window

After flags are chosen and saved the OKAY button will open a Prefs setting window with defaults. A unique prefs file will only open if it is named "RevRdist Prefs" and is at the same

level as the RevRdist application. Items are entered in the RevRdist prefs screen of each lab machine. Both RevRdist and the prefs file reside in System 7's Startup Items folder for automatic updates.

AppleTalk zone:	LibLRC
AppleShare server:	LRCsvr
AppleShare username:	varies by machine
Password:	same for all lab Macs

Distribution files

This file will vary by setup. The appropriate DistFile will be called from the client if the lab has a wide variety of Mac models. For example:

rArchive:distfiles:DistFile40scan2	#16
rArchive:distfiles:DistFile80std2.HD17	#17
rArchive:distfiles:DistFile80std2.HD18	#18
rArchive:distfiles:DistFile80lsr2.PATH19	#19
rArchive:distfiles:DistFile80lsr2.HISTO20	#20
rArchive:distfiles:DistFile80lsr2.PATH21	#21

```

rArchive:distfiles:DistFile80lsr2.HISTO22    #22
rArchive:distfiles:DistFile80av2            #31-37
rArchive:distfiles:DistFile80cd2.A11        ADAM 11
rArchive:distfiles:DistFile80cd2.A12        ADAM 12
rArchive:distfiles:DistFile80DeskMac2       DeskMac
Master Folder:          rArchive:Master:
Junk Folder:            ×Lost and Found (default)

```

Running and Debugging RevRdist

The 'greater than' and 'less than' action list delimiters refer to the folder itself and the contents respectively. The following action list would copy a server folder to a client (Su) but ignore the contents (Si).

```

> folder name      :SuE- # denotes folder actions
< folder name      :Si   # denotes folder contents actions

```

If you want to ignore copying a folder the appropriate action code is:

```

> folder_name      :SiCd
< folder name      :SiCd

      or

> folder_name      :Ai  # ignore folder
< folder name      :Ai  # ignore folder contents

```

OR

```

> folder_name      :Ad  # deletes folder nomatterwhat on client
< folder_name      :Ad  # deletes folder contents nomatterwhat

```

If you want to ignore copying a file the appropriate action code is:

```

| file_name        :Ai

```

Note: if you use only the "pipe" tag to denote an action to ignore a folder you will see "Control file does not match Master for folder ..." in the Activity window. Using the two-line specification with the "greater than/less than" tags eliminates the problem.

If you use the renaming function be sure to mark the Master file for deletion in the dist file or it will be copied twice. Once for the renaming routine and once by itself. The Mac won't boot correctly if it's the System file. (This technique is used for multiple System files using differing Ethernet drivers installed as resources. For example:

```

|      System          :A.O- =System.Apple660av
|      System.Apple660av :SiCd

```

When doing a "List only, don't Do/Verbose" session show the Activity Window when complete. The following notes are helpful when reading the file:

- ⊙ "Working on..." and "Done with..." are start/stop markers for actions on folders.
- ⊙ "Ignoring..." is the marker for items inside a folder.

The log may also show something like:

```

"Used CxD (CSD) Cd for some_file"

```

A filename can be located in three places (C)lient, (S)erver and (D)ist File. These locations are checked in sequence; 1) a literal check and 2) a renamed file check (in parenthesis.) The line immediately above tells

you if it was found in each location and if a different name was specified in the action list. That is why the (C)(S)(D) appear in the parenthesis. It signifies finding the file on the second pass through an action. The code following the parenthesis in the line above is from the action list for the file:

```
| some_file :Cd =a_different_name|
```

Look for the following markers when debugging or viewing an Activity window:

```
"Missing..."  
"Discarding..."  
"Copying..."  
"Cannot update...", etc.
```

These need to be resolved.

Final Setup

Once RevRdist has been debugged and is running successfully, the application and prefs (named 'RevRdist Prefs') are placed in the Startup Items folder of both the client and the server. RevRdist flags are set to "Use prefs mod time", "Restart when done" and "Quit other applications." The "Run unattended" flag will be set by virtue of placement in the Startup items folder. The appropriate RR Helper extension will be created, placed on the server and in the client System Folder and the machine will be restarted.

Maintenance

Once RevRdist is set up and debugged things should go smoothly. There are some steps to take to ensure smooth operation.

1. RevRdist can cause trouble by fragmenting, etc. due to daily over copying and file deletion. Run Norton Utilities or similar utility monthly.
2. Occasionally alias files may lose their target. The only sure-fire method seems to be to delete the alias and run RevRdist to replace it. It will retarget the new copy.
3. After running RevRdist daily on lab machines for four months RevRdist slowed considerably (15 minutes vs. 1 minute.) I replaced the RevRdist app with a fresh copy and configured it appropriately. (It is copied over to each client on every run.) It now runs as before. Maybe replace it each quarter-year...

In this lab the On Demand Disk contains all RevRdist Prefs files and RR Helper extensions as well as RevRdist, ResEdit, Build RR Helper (stack), an alias to Master server and the control panels Sharing Setup and Users and Groups.

This lab also has a System 7.5 Network Access Disk which is a startup disk used when a Mac is in serious trouble. Using this the lab administrator can access the Master archive which also includes a set of System Install images and ShrinkWrap (a much-improved Disk Copy available on the Internet.) Drag them to the local drive and mount the images and reinstall the System. Follow this with a RevRdist on-demand run.

Read Me for RR Helper

RR Helper Note: These system extensions can be used in 'families.' This INIT/extension modifies the System file (and itself) the first time it runs. Create the INIT then copy it to Master before running on any Mac.

The following options are selected for creating the RR Helper system extension:

- ⊙ Workstation Number A (may be used in formulas)
- ⊙ AppleShare Startup Options (from Mac where INIT made)
- ⊙ AppleTalk Node Number (from formula)
- ⊙ Beep Sound (Sound c/pnl)

- ⊙ Computer Name "#";TEXT V (Sharing Setup;'Macintosh name')
- ⊙ Desktop Pattern (General Controls)
- ⊙ General PRAM Settings (General Controls)
- ⊙ Mouse Controls (Mouse c/pnl)
- ⊙ Network Selection (Network c/pnl)
- ⊙ Owner name "LRC #";TEXT V (Sharing Setup)
- ⊙ Printer Selection (Chooser)
- ⊙ Volume name "HD #";TEXT V
- ⊙ Additional initialization (clear the field)
- ⊙ Lookup tables (clear extra lines)

'V' in a formula will take the numeric portion of a drive name and use it.

'A' is the current LocalTalk node number.

7.4 RevRdist Administrator's Guide

RevRdist

Administrator's Guide

By Dale Talcott

Purdue University Computing Center

aeh@cc.purdue.edu

Introduction

RevRdist is a "demand-pull" Macintosh equivalent of the Berkeley Unix program rdist (1). Both programs exist to keep copies of files synchronized among systems in a network. Where rdist normally runs on a source system and distributes updated files to destination systems, RevRdist reverses this by running on a client and copying information from the server system.

The intended setting for RevRdist is a group of Macintosh workstations, each with a local hard disk, connected to an AppleShare server. The server maintains the master copies of folders and files. Client workstations have local copies of (a subset of) these files, as well as additional files that may be unique to the workstation. RevRdist compares the client copies of folders and files against the master copies. Then, based on options and criteria specified in a distribution control file, RevRdist can do the following:

- Discard unwanted files/folders from the client.
- Move other files/folders to a "junk" folder on the client. They are then discarded after a selected interval.
- Copy files/folders from the server to the client.
- Make client files/folders invisible from the Finder.
- Set the Finder window positions of client files/folders to match those of the master copies on the server.

Actions can be specified explicitly for individual files or folders and default actions can be set on a per folder basis for contents of folders that have no explicit action.


Notice: RevRdist does not go to great lengths to protect itself or its files against malicious users. If you have destructive users, you need more than RevRdist.

RevRdist requires the hierarchical file system and has been tested with versions of the Macintosh operating system from System 6.0.3 to 7.0.1.

The document describes options as relevant to “System 6” or “System 7.” What this really means is that RevRdist optionally uses some operating system features that became standard in System 7. If you are running an enhanced version of System 6, RevRdist will take advantage of it.

The pieces

The RevRdist package uses five kinds of files: the RevRdist application, an INIT to selectively invoke RevRdist at reboot, a preferences file, a distribution control file, and occasional temporary files. These files have the following icons, types, and creators:

Icon	Type	Creator	Function
.	APPL	Dist	Application
.	INIT	Dist	Boot time INIT
.	RDsp	Dist	Preferences file
.	TEXT	varies	Distribution control file
	Junk	Dist	Temporary file

The INIT is optional. So is the preferences file, as RevRdist has a set of default preferences. However, these preferences are unlikely to be useful, so you will want to create a preferences file. The distribution control file is required.

Under System 6, the application, INIT, and preferences files normally reside in the System folder. Under System 7, the application and preferences go in the Startup Items folder inside the System folder. The INIT is not used. The files can go anywhere if you do not need RevRdist to run automatically.

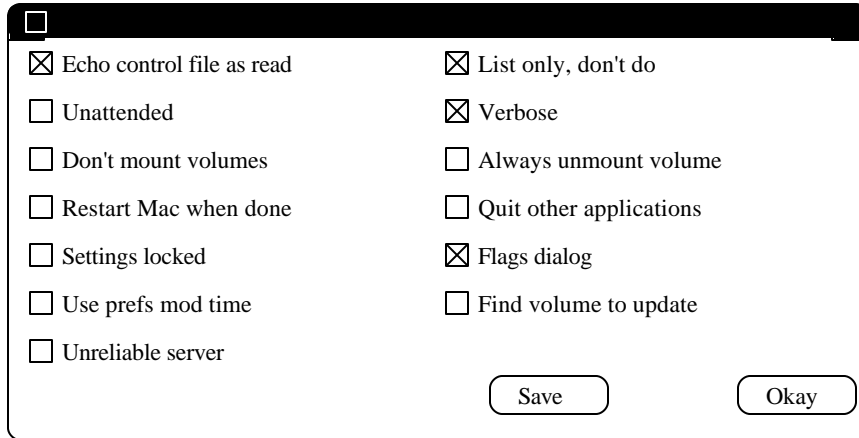
The distribution control file can reside on the server and several clients can share the same control file. The preferences file is private to the client and must reside on the client disk.

There are three different sources of configuration information for RevRdist: processing options, file/folder location preferences, and the distribution control file. RevRdist supplies dialogs to set the processing options and the preferences. The processing flags are saved within RevRdist itself. The preferences go into preferences files. The control file is created and modified outside RevRdist by an external text editor.Ü

Processing flags

If you hold the mouse button down while starting RevRdist, or if the “Flags dialog” flag is set, RevRdist presents a dialog as follows:

†Since there is currently no tool explicitly for creating the control file, you will have to create it with any Macintosh text editor, e.g. MacWrite, Word, miniWriter. Thus, its icon will be the icon appropriate for the editor used, not the one given above.



The checkboxes have the following meanings:

Echo control file as read	As each line is read from the distribution control file it is listed to the Activity window. This option might be used when RevRdist complains about the control file, and it is not clear which line causes the problem.
List only, don't do	RevRdist lists each action it would perform to the Activity window, but does not actually do the action. Used to test preference and control files without risking the contents of the client disk. This flag is also set internally by RevRdist if the client volume is read-only.
Unattended	Set internally when RevRdist detects it was started by the INIT under System 6 or by the Finder at startup under System 7. Some user interactions are disallowed (e.g., changing the preferences), but you can still Pause and Continue, display the Activity window, and Quit.
Verbose	When set, RevRdist lists in the Activity window the action for every file or folder, even if it is to just ignore the file or folder. RevRdist runs noticeably more slowly when this flag is set, so you should turn it off once you have debugged your preferences and control files.
Don't mount volumes	RevRdist normally mounts and unmounts the volume containing the master folder as needed. This option disables that.
Always unmount volume	When set, RevRdist unmounts the volume with the master folder even if RevRdist did not mount it.
Restart Mac when done	If this flag is set when RevRdist finishes, it restarts the Macintosh.
Quit other applications	Under System 7, if this flag is set, RevRdist terminates the Finder, desk accessories, and any other applications before mounting the master volume. The flag has no effect under System 6.

Settings locked	<p>If this flag is set when the “Save” button is pressed, the running copy of RevRdist alters itself so neither the Flags dialog nor the Preferences dialog can be brought up, thus locking the current settings for both.</p> <p>You might want to set this flag if you install RevRdist on client hard disks in an environment where users (e.g., students) run programs just to see what they do.</p> <p>Note that this is protection against only the curious: dedicated hackers will have no trouble getting around this flag.</p>
Flags dialog	Bring up this dialog window automatically on startup. If this flag is clear, and the “Settings locked” flag is not set, you can show the Flags dialog by holding the mouse button down when starting RevRdist.
Use prefs mod time	When set, RevRdist uses the last modification time of the preferences file to decide whether it is time to run.
Find volume to update	Requests RevRdist to scan the mounted volumes looking for writeable, non-floppy, non-AppleShare volumes. The first such volume is assumed to be the volume needing updating.
Unreliable server	Ignore. This flag is for use at Purdue.

More about the flags

If both the “Echo control file” and “Verbose” options are set, RevRdist displays the applicable criterion and action for each file or folder examined. (See later for the meanings of “criterion” and “action”.) This combination is useful in debugging control files when RevRdist does something other than what you expect.

The “Unattended” flag implies that RevRdist is running automatically, without an attendant user. It has the following effects: if the “Use prefs mod time” flag is also set, RevRdist quits unless the update interval has elapsed since the last time it ran; if RevRdist finds newer versions of itself in the same folder, the older versions quit instead of running (this is helpful under System 7 to install updates to RevRdist); RevRdist starts updating the client immediately rather than displaying the preferences dialog and waiting for a Go request; RevRdist ignores a request to quit during the copying of a file from the server to the client; and instead of interacting with the user on errors, RevRdist quits or ignores the error.

The “Don’t mount volumes” option exists to provide backward compatibility with previous versions of RevRdist. Originally, RevRdist relied on the INIT or the AppleShare preferences to mount the server volume.

The “Always unmount volume” flag is provided for sites that mount the server volume outside of RevRdist, but still want it unmounted when RevRdist completes.

Anytime RevRdist is set up to run automatically, either from the INIT under System 6, or because it is in the System 7 Startup Items folder, you should probably set the “Restart Mac when done” flag. This is so any cleanup RevRdist needed to do is available to the next user of the machine. For example, if RevRdist had to copy or replace an INIT, that INIT has no effect until a system restart. By restarting after RevRdist finishes, you guarantee the system initializes the way you intended.

You should also select the Restart option if you select the “Quit other applications” option and you have any applications or desk accessories besides RevRdist in the Startup Items folder. Further, restarting is needed in the uncommon case that you automatically connect to the master server using some account other than the guest account and that account does not have access to the master folder. In this case, RevRdist

has to unmount all volumes from the server before it can mount the volume with the master folder. When it finishes, RevRdist will try to remount the volumes it unmounted, but will not succeed unless it can do so as guest.

If Restart is set, you don't need "Always unmount volume," because restarting the Mac unmounts all volumes anyway.

Be careful when you set the Restart option that you don't get into a loop. That is, if RevRdist runs automatically every time the system restarts, and if it restarts the system every time it runs, then the Mac is not going to do much else! You avoid this looping by setting a long enough update interval in your preferences file (see below). Under System 7, you must also select the "Use prefs mod time" option.

The "Quit other applications" flag is useful when you have files on the master volume that you want to hide from curious eyes. RevRdist causes the Finder, desk accessories, and other applications to exit before it mounts the master volume. Without these, it is tricky (but not impossible) for users to do anything with the master volume. If the AppleShare account you run RevRdist under has a password, you should probably set this flag. If you don't care whether the clients can access the master folder, then leave this box unchecked so the users can put RevRdist into the background while they do other work. If you set up RevRdist to run automatically and you don't check this option, then don't check the Restart option either. Otherwise, users might start work while RevRdist is running, only to have RevRdist reboot the system out from under them.

If RevRdist is the only program running when RevRdist finishes, the Mac automatically restarts the Finder. You don't have to restart the Mac just to get the Finder back.

The "Use prefs mod time" flag has two effects: if the Unattended flag is also set, then RevRdist checks the modification time of the preferences file. If this time is more recent than the interval specified in the preferences, then RevRdist quits without taking any other action (including restarting). Second, if RevRdist does run, and if it completes without being asked to quit prematurely, then it sets the preferences modification time to the current time.

The "Find volume to update" flag is used to create a "one-button" restore hard disk floppy, described later in the section on running RevRdist on demand.

The "Unreliable server" flag causes RevRdist to perform additional consistency checks. When something seems wrong, RevRdist alerts the user. If it is running unattended, RevRdist quits immediately on inconsistencies. If your master folder is on AppleShare, you don't need to set this flag. If it is on InterCon's NFS/Share, you must set it (at least as of September, 1992).

As distributed, the "Echo control file as read," "List only, don't do," "Verbose," and "Flags dialog" flags are set. Under System 6, once you have your preferences and control files debugged, you should create a copy of RevRdist saved with no flags selected and use that for on-demand running. Another copy of RevRdist should be created with only "Restart Mac" and "Settings locked" selected. This is the version that the clients run automatically. It should go in the master copy of the System folder, along with the INIT and a copy of the preferences file.

Under System 7, your on-demand copy of RevRdist might also have the "Quit other applications" flag selected. The automatic copy should have only "Restart Mac," "Quit other applications," "Settings locked," and "Use prefs mod time" selected. It goes in the master copy of the Startup Items folder, inside the System folder, along with a copy of the preferences file.

The "Save" button changes RevRdist to make the current checkboxes the defaults.

The "Okay" button dismisses the dialog and lets RevRdist continue.

Setting the preferences

RevRdist is designed to run automatically and unattended. In normal operation, it gets controlling information entirely from the preferences file and the distribution control file specified by the preferences file.

The preferences file is created and modified using the “Preferences Ö” item on the “Windows” menu. This brings up a dialog window:

AppleTalk zone	
AppleShare server	
AppleShare username	
Password	
Distribution file	DistFile
Master folder	AppleShare:Clients:Master:
"Junk" folder	Lost & found
Update interval	23:00
"Junk" lifetimes - minimum	1 00:00
maximum	7 00:00
Disk space threshold	5000000

Information maintained in the preferences includes:

AppleTalk zone	The AppleShare server zone. Leave blank to use the local zone.
AppleShare server	The AppleShare server holding the master folder. This name should appear in the Chooser “Select a file server” window when you click on the AppleShare icon.
AppleShare username	The username to log on to the server as. This user must have “see files” and “see folders” access to the master folder.* RevRdist does not protect the password well. It is best to restrict the RevRdist account to only “see files” and “see folders” access to the master folder.
Password	The AppleShare user password.
Distribution file	The complete path to the distribution control file.
Master folder	The complete path to the master folder on the server.
“Junk” folder	The name of the folder on the client to hold mis-placed files temporarily. If a student leaves a file on a local hard disk, RevRdist moves the file to this folder, where it stays for a while before being discarded.
Update interval	Minimum time between automatic RevRdist runs. Because automatic runs happen only when the system is restarted, the actual interval between runs can be much longer.
“Junk” lifetimes - minimum	Junk files newer than this are not discarded. The last modification time is used to determine age.

* For some reason, Novell servers also require “Write To File” permission in order to read Finder comments from the server. Without this permission, RevRdist still copies files, but it cannot copy their comments.

maximum	Junk files older than this are always discarded. Similarly, junk files with modification times at least this far in the future are always discarded.
Disk space threshold	If free space on the client disk is less than this size (in bytes), junk files with ages less than the maximum, but greater than the minimum, are discarded.

The sample dialog above shows the default values built into RevRdist. Paths to files and folders must be entered explicitly, with colons (":") after each folder name.

Times are entered as the number of days, one or more spaces, and then hours, minutes, and seconds in whatever form is standard for the country. For the U.S.A, this means a time of two days, four hours, and 20 minutes is entered as "2 4:20."

On older versions of System 6, times must be entered in seconds. Some common values are 1 hour = 3600 seconds; 8 hours, 28800 seconds; 1 day, 86400 seconds, and 1 week, 604800 seconds.

The default values come out to an update interval of 23 hours, a minimum junk lifetime of one day, and a maximum of one week.

Preferences settings can be used by pressing the "Go !" button. They can be saved to a file with the standard "Save" and "Save as Ö" items on the "File" menu.

You can open a different preferences file with the "Open Ö" item, also on the "File" menu.

Creating the distribution control file

The distribution control file is a plain TEXT file, created and edited with any program that can read and save "text-only" files. Its format, however, is quite obscure and ugly to work with. Eventually, there will be a utility, either part of RevRdist or a HyperCard stack, to simplify creating the file.

The control file is comprised of lines terminated by carriage return or line feed. Empty lines, or lines that are just whitespace (blanks or tabs), are ignored.

The first non-whitespace character of a line determines the type of information on the line:

- # The line is a comment and is ignored.
- | The line names a client file and what should happen to it.
- * Like a "|" line, except it identifies files by type and/or creator instead of by name.
- > The line starts information about a client folder.
- < The line ends information about a folder and supplies the default actions for the contents of that folder.

Except for comment lines (which won't be discussed further), each line then has either two or three fields. The first field is a file or folder name. This field ends at a colon (":"). The second field is an action list and ends at whitespace or end of the line. The optional third field is a substitute file or folder pathname. When present, this field extends to the end of line. Whitespace between fields is ignored.

Macintosh file and folder names can contain leading or trailing spaces. To protect these spaces from being treated as whitespace, put a backslash ("\") before the first leading space in a name or before the last trailing space. Also put a backslash before any real backslash in the name. For example, a file name consisting of two spaces, an "A", a "\", a "B", and two more spaces at the end would be entered as:

```
\_A\_B\_
```

where spaces here are indicated by “_”. To be clearer about spaces, you could put backslashes before each leading or trailing space:

```
\_ \_ A \ \ B \_ \_
```

Even better would be to avoid names with leading or trailing spaces.

Spaces in the middle of names don’t need any special treatment.

Backslashes also introduce characters that do not have printable representations and so must be given by the decimal, ASCII value of the character. For example, the name used by the System 7 Finder for the invisible files containing icons is “Icon” followed by a carriage return. Because carriage return has the ASCII value 13, the file name can be specified as “Icon\13”.

On “*” lines, the type and creator appear instead of a file name. The format is TYPE(CREATOR), where TYPE and CREATOR are four-character identifiers assigned to files by the Macintosh operating system. Either TYPE or (CREATOR) may be omitted, meaning any type or creator is acceptable. A type or creator of “????” stands for either itself or binary zero. As with names, use “\” to escape spaces, tabs, “\” itself, and unprintable characters.

For example, assume we have a control file that looks like:

```
# Distribution list for normal clients
>root          :mumble
  | Desktop    :mumble
  > Applications :mumble
    | Microsoft Word :mumble
    | SuperPaint  :mumble
  < Applications :mumble
  > Lost and found :mumble
  < Lost and found :mumble
  > System folder :mumble      =Clients:Client system
  < System folder :mumble
<root          :mumble
#end of distribution list
```

The mumbles are action lists, described later.

This file explicitly describes four items in the root folder: one file (“Desktop”) and three folders (“Applications,” “Lost and found,” and “System folder”). It also describes two items within the Applications folder. There may be additional files and folders on the client or within the server master folder. The actions taken for such unnamed files are determined by the default actions for the folders that contain them.

The file says the folder called “Client system” within folder “Clients” on the server is the master copy of the folder known as “System folder” on the client.

Action lists

Action lists are strings of character pairs. Loosely, the first character in each pair indicates a criterion and the second indicates an action to take when the criterion is satisfied. By convention, the criterion character is an uppercase letter and the action is lowercase, but case is actually irrelevant.

The criteria are:

Letter	Mnemonic	Criterion or attribute
C	Client	File/folder exists on client, but not on server.
S	Server	File/folder exists on server, but not on client.
V	Version	File on server has a different creation date from copy on client.
N	Newer	File on server has later modification date from file on client. That is, the server file is newer than the client file.
O	Older	File on server has earlier modification date from file on client. That is, the server file is older than the client file.
Z	siZe	File on server is different size from file on client.
E	Else	None of the above applies.
W	Window	Client Finder window coordinates.
H	Hide	Client file/folder visibility from Finder.
L	Locked	Client file/folder write lock.
B	Backup	Maintain backup copy.
R	Restart	Restart if this file is updated.
A	All	Shorthand to set the same action for all criteria and most attributes. "Ax" is the same as "CxSxVxNxOxZxExWxHxLx".

C, S, V, N, O, Z, and E are criteria. W, H, and L are attributes of the client copies of the files or folders. B and R alter RevRdist's processing for specific files. "A" is not a criterion itself—it is only an abbreviation to assign the same action to all criteria and attributes (except B and R).

Note that V, N, O, and Z criteria apply only to files and are ignored when examining a folder.

If a criterion appears more than once in an action list, only the last action applies. E.g., CxCyCz is the same as Cz alone.

Action characters, the criteria they apply to, and their meanings are:

Criteria	Char	Action
any	. (period)	Whatever action is the default for the folder containing this file/folder.
any but E	-	Ignore this criterion.
E	(minus)	Ignore if file. Substitute "S" action if folder.
CSVNOZE	i	Ignore file/folder.
W		Don't change Finder window position.
H		Don't change file/folder visibility.
L		Don't change file/folder lock.
CSVNOZE	d	Discard file/folder from client.
H		Make client file/folder visible on client.
L		Unlock client file/folder on client.
CSVNOZE	j	Move file/folder to junk folder. If there is already a junked file with the same name, the existing file is renamed using variations of the suffix ".old".
CSVNOZE	u	Update file by copying it from the server to client, replacing any existing file.
CSE		Update folder by comparing the list of items in the client folder with those on the server. Create client folder if not present.
WHL		Set Finder window information, visibility, or lock on client to match that of master copy on server.
H	s	Make client file/folder invisible on client.
L		Lock client file/folder on client.
B		Make backup of file before replacing it.
R		Restart Macintosh if this file is updated.

The default action for a criterion is given by the default actions for the folder containing the item. This can be made explicit with the "." action character.

The C, S, V, N, O, Z, and E criteria are tested in that order. The first criterion that applies and whose action is not "-" completely determines the action taken. Even if two or more criteria might be met, only one action is taken: the action of the first matching criterion.

If the action performed is "u" or "i," then the W, H, and L attributes are examined and their actions taken.

Further, when the action is "u", causing a file to be copied from the master, then the B and R attributes are examined. If the B action is "s", then the old client copy of a file is moved to the junk folder to serve as a temporary backup copy of the file. If the R action is "s", then the Restart flag is set. You might use this if you normally run without the "Restart Mac when done" flag, but nonetheless want to restart when certain files (e.g. the System file or some extensions) are replaced.

The client and server might both have a file with the same name, but with different types or creators. These are treated as two different files, first matching the C criterion, then the S. Note a possible confusion if the C action says to ignore the file, but the S action says to update it. RevRdist junks the client file first in this case.

Similarly, a file name on the client might conflict with a folder name on the server. Again, the two are handled separately, first matching the C and then the S criterion. The same note about junking the client copy applies.

RevRdist never changes the master folder. All of its actions affect only the client. This means some criterion/action combinations make no sense. For example, "Sd" would mean "delete this file from the client if it exists only on the server". Nonsense actions are ignored.

Action examples

Assume we have several workstations in a student laboratory. We want all client workstations to be mostly the same. If we find extra files or folders on clients, we move them to the junk folder (which we name "Lost and found").

At the root level our distribution control file looks like:

```
>root           :Au
      |         Desktop      :AiHs
<root           :CjSuVuNuOuZuE-WuHuLu
```

The first line says we want to update the volume from the master. (The name "root" is arbitrary and ignored.) The next line makes sure the Finder's Desktop file is invisible, but otherwise left alone. The last line moves anything else only on the client to the junk folder ("Cj"). Everything else in the server master folder is copied to the client if it is missing or different ("SuVuNuOuZuE-"). The "E-" option says we don't need to copy files if they are the same, but we should update folders. The last action list could be abbreviated to "AuCjE-".

This example is too simple because it does not account for files in the System folder. These should not be exactly identical for all workstations, so we need to expand the file to something like:

```
>root           :Au
      |         Desktop      :AiHs
      >        System folder :Au
      |         RevRdist prefs :AiSuVuWu
      <        System folder  :AuCdE-
<root           :AuCjE-
```

(System 7 control files are more complicated. For now, we restrict our examples to System 6. Later sections describe additions for System 7.)

The "Cd" on the "< System folder" line says if we find strange files in the System folder, we want to throw them away directly (we have no use for "donated" INITs, etc). Nearly everything else is kept synchronized with the master copies on the server. You may or may not want to synchronize the RevRdist preferences file. If you run RevRdist automatically, you certainly don't want to, because the modification time is used to remember when RevRdist ran last. In the example, we update the preferences file only if a new version is created on the server or if it is missing entirely from the client.

HyperCard stacks pose another problem. These files are modified just by looking at them. Further, they are big. If we tried to keep them synchronized with master copies, they would be copied much more often than necessary.

Assume we decide not to copy HyperCard stacks unless we install new master versions. There are also some specific stacks to move to clients whenever we change them on the server. We add the following lines among the entries for the root level:

```

> HyperCard folder :A.
* STAK(WILD) :VuN-O-Z-
| HyperCard :AuEi
> HyperCard Stacks :A.
| Home :VuNuO-Z-
| Address :VuNuO-Z-
| Phone :VuNuO-Z-
< HyperCard Stacks :A.
< HyperCard folder :A.

```

The “*” line ignores changes to stacks unless a new version is installed on the server. The line for HyperCard itself keeps it the same as the master copy (getting rid of XCMDs that users’ stacks might have installed).

Many system extensions (INITs) and control panels are like HyperCard stacks in that they are modified every time they are used. The same trick of using an action list including “VuO-Z-” could handle these cases. For INITs and control panels, though, it might be better to see if they still work when the file is locked (“Ls”). If so, this is a better solution because it prevents students from changing the settings for those INITs or control panels. With some INITs and control panels, the user can change the current settings even if the file is locked, but the settings revert to their locked values when the system is restarted. Others simply won’t work, or don’t work reliably, when locked. For these, if their settings are important, they must be updated from the master every time.

We also include a line to ignore the contents of the junk folder.

```

> Lost and found :AiW.
< Lost and found :Ai

```

Files and folders are processed alphabetically within each folder, so if you specify an action other than to ignore the junk folder, files already junked by the time the junk folder is processed are examined on the current run of RevRdist. Files junked after the junk folder is processed will not be handled until the next RevRdist run.

You might want to discard any applications that end up in the junk folder immediately, rather than the next time RevRdist runs. You can do this by naming your junk folder “xLost and found” and putting these lines in the control file:

```

> xLost and found :AuW.
* APPL :Cd
< xLost and found :AuE-

```

The “x” starting the name puts the folder at the end of the alphabet. “x” is typed by pressing <option><shift>V.

More on type/creator lines

The “*” lines are more complicated than heretofore described. “*” lines don’t apply to files explicitly appearing on name (“|”) lines. Rather, “*” lines act like folder-level defaults restricted to specific kinds of files.

Type lines are inherited by nested folders. In the HyperCard example above, the STAK(WILD) line applies to “HyperCard folder” as well as all folders within “HyperCard folder” and folders within those folders, etc. If we placed the line directly under the root folder, it would apply to stacks anywhere on the client.

When an inherited “*” line contains a “.” action character, the action used is the default for the nested folder, not that of the folder containing the “*” line.

More than one “*” line could apply to a given file. In that case, the most recently appearing line applies. For example, take the following control file:

```

>root          :mumble
*      APPL    :AuCdEi
...
>      programming :mumble
*      APPL    :Cj
...
<      programming :mumble
...
<root          :mumble

```

With this, any application appearing anywhere on the client is discarded unless it also appears on the server. Any difference in dates or sizes between client and server causes the application to be restored from the server. An exception is made for client-only applications found in a “programming” folder: these are moved to the junk folder instead.

The previous example also illustrates a potential trap. The root level APPL line applies to all applications that don’t have explicit lines of their own, no matter what the defaults are for the folders the applications appear in. It might be better to use:

```

>root          :mumble
*      APPL    :Cd
...

```

System 7 considerations

Under System 6, only the Desktop file and the System folder need special consideration. With System 7, Apple adds several special files and folders. You must configure the control file to leave most of these alone.

Here is a suggested excerpt from a System 7 distribution control file.

```

> root          :Au
*      alias      :O-Z-
*      icons      :Cd
>      \3\2\1Move&Rename :AiHs
<      \3\2\1Move&Rename :Ai
|      AppleShare PDS    :AiHs
|      Desktop          :AiHs
|      Desktop DB       :AiHs
|      Desktop DF       :AiHs
>      Desktop Folder  :A.Hs =DesktopFolder
<      Desktop Folder  :A.
>      DesktopFolder   :Ad
<      DesktopFolder   :Ad
>      NetWork Trash Folder :AuHs
<      NetWork Trash Folder :Ad
>      Trash           :AuHs
<      Trash           :Ad
|      VM Storage       :Ai
Ö (local folders and junk folder)
>      System Folder    :A.
>      Apple Menu Items :A.
<      Apple Menu Items :A.
>      Control Panels   :A.
<      Control Panels   :A.
>      Extensions       :A.
<      Extensions       :A.
>      Preferences      :A.

```

```

<           Preferences      :A.
>           PrintMonitor Documents      :Ai
<           PrintMonitor Documents      :Ai
>           Startup Items      :A.
*           APPL(Dist)          :Cd
*           RDsp(Dist)         :CdO-
<           Startup Items      :A.
<           System Folder      :Cd
< root      :AuCjE-

```

The “alias” and “icons” entries are explained later. I do not know what the “\3\2\1Move&Rename” folder is for, so it’s left alone. Similarly, I am not sure what the “AppleShare PDS” files is for, but believe it is created when you enable file sharing on a client.

The “Desktop DB” and “Desktop DF” files are the System 7 replacement for the “Desktop” file from earlier systems. Like the “Desktop” file, these should be left alone.

The “Desktop Folder” is where items appearing on the desktop, outside any folder window, are kept. It needs to be invisible on the clients. However, the System 7 Finder does not let you create a folder named “Desktop Folder.” So, we call the master folder “DesktopFolder” (without the space) and use the renaming function to match it up with the client’s “Desktop Folder.”

Almost any time you use a renaming line, you will also want an entry saying to ignore the master copy of the file or folder under its other name. This is the reason for the lines for “DesktopFolder.” Without these lines, “DesktopFolder” is copied to the clients twice, once as “Desktop Folder” and once as itself.

The “Trash” folder holds items that appear in the desktop trash can. This, too, should be invisible. The folder default of “Ad” empties the trash when RevRdist runs. Another reasonable choice would be “Ai”.

The “VM storage” file is used for virtual memory paging. If you do not run with virtual memory enabled, change the “Ai” to “Ad”.

When file sharing is enabled on a client Macintosh, the “NetWork Trash Folder” holds items located on this Macintosh’s hard disk that have been discarded from another Macintosh. Treat this folder the same as the Trash folder.

The “PrintMonitor Documents” folder holds files for background printing that has not completed yet. It should be left alone.

The entries inside the “Startup Items” folder identify RevRdist and the RevRdist prefs files, but not by name. See the section on installing new versions of RevRdist for why it is done this way.

The lines for other folders inside the System Folder (“Apple Menu Items,” “Control Panels,” “Extensions,” and “Preferences”) could all be omitted, because their action lists are just “A.”. They are included in the example as placeholders to remind you where to put entries later if you discover an INIT or desk accessory that needs special treatment.

“Desktop Folder,” “NetWork Trash Folder,” and “Trash” are reserved names that Finder will not let you use. This does not matter for the latter two names, because we do not need master copies of them. However, in order to get RevRdist to examine the contents of these folders (e.g. to discard the contents), the folder start (“>”) line must have a client-only (“C”) action of update (“u”).

Aliases

System 7 added a special file called an alias that is just a pointer to a different file or folder. Aliases require a little care. Suppose we create an alias on one client pointing to a file on that client. Now we copy that alias to the master folder and then RevRdist it to a different client. If you then try to use the alias on the second client, you won’t be able to, even if the hard disk of the second client looks exactly like the hard disk of the original client. The problem is that the second Mac can tell that the alias did not refer to its own hard disk.

In a student lab, however, you want to be able to do this. To this end, RevRdist handles alias files specially. When RevRdist updates an alias file, if the “O” and “Z” criteria are both “-”, RevRdist examines the target of the alias. If it looks like it should be a file or folder on the client, and if a file or folder with that name exists on the client, then RevRdist changes the client’s copy of the alias file to point to the client’s disk.

This adjustment changes the modification time and size of the alias file, which is why RevRdist updates aliases only when “O-Z-” is specified.

Aliases that point to files or folders on a server can be copied from client to client with no problem.

To RevRdist, all alias files have a type of ‘alis’ and a creator of ‘MACS’. In truth, the type and creator of an alias is the same as the file the alias refers to. RevRdist pretends otherwise so you can refer to them implicitly on type/creator lines as:

```
*      alis(MACS)      :mumble
```

As a further convenience, RevRdist recognizes “alias” as a shorthand for “alis(MACS).” Looking back at the beginning of the System 7 example, the second line tells RevRdist that we want to fix up aliases to the client volume.

Note that all aliases are files, even when the target of the alias is a folder.

Icon files

The System 7 Finder allows you to customize the icon for any file or folder. The Finder stores these custom icons in an invisible file called “Icon\13”. (The “\13” is the ASCII for a carriage return.) For a file, this icon file goes inside the same folder as the file. For a folder, the icon file goes inside the folder itself.

Students will undoubtedly play with this feature to confuse the appearance of the workstation desktops. Discarding the icon files restores the original appearance.

To make it easy to refer to any “Icon\13” file, without needing to name each explicitly, RevRdist pretends the icon files have the type ‘icon’ and creator ‘MACS’.

As with alias files, there is a shorthand form to designate the files of icons: ‘icons’. Thus, the third line of the example control file

```
*      icons          :Cd
```

says to delete any icon files that appear only on the client.

Typical setup

RevRdist runs in three typical modes: initial installation, on demand, and automatically.

Initial installation

The initial installation involves creating the master folder on the server and creating and testing preferences and distribution control documents. You should plan on completely erasing the local hard disk of a test client workstation at least once, either by accident or deliberately to check that the master folder contains everything needed by a client.

The master folder

The simplest way to create the master folder is to set up a client hard disk just the way you want it. Files or folders that will be made invisible later should be left visible now. Create the master folder on the server. Copy everything from the client to the master folder.

You should set the master folder so it is readable and seeable by everyone, but not writeable. If access by everyone is too open, you might consider a dummy account that has see and read access to the master folder. Do not give this dummy account “make changes” access. The password stored in plain text in the preferences file, so it is simple for someone to discover it.

The distribution control file

Create the distribution control file using any Macintosh text editor that can save files as “text-only.” The contents were described above. In general, the less you specify in the control file, the more flexible it is. You will need explicit lines for files or folders that you want to be invisible or that have differing names between the client and the server.

If you have clients with different functions, you may want different control files for each function. Suppose some client workstations have a scanner or other specialized equipment attached that needs special software. The control file for these clients should be set up to update the special software, but the control files for the other clients should ignore or discard the specialized files.

You will probably want to keep the control files in the master folder on the server.

The preferences file

Once you’ve created the master folder and the distribution control file, you are ready to test them and create a preferences file. Copy RevRdist to the client disk and double-click it. For now, click the “Okay” button on the Flags dialog. You should be presented with the “Preferences” dialog. Fill in the locations of the master folder and control file for this client. You must supply absolute paths, using colons “:” to separate folder names. Also, set the junk folder name and junking parameters.

As distributed, RevRdist is configured to only list what it would do, but not actually do anything, and to be verbose about what it would do. While you’re experimenting with the control and preferences files, it is safest to leave it this way.

Save the settings to the client disk (and maybe to a floppy disk, in case something goes wrong). Select “Activity” from the “Windows” menu so you can watch things happen, then switch back to the Preferences dialog (again on the Windows menu). Press the “Okay” button. If there is something wrong with your preference values, RevRdist displays an alert with an explanatory message and goes back to the Preferences dialog for you to make corrections.

If RevRdist finds no problems with your preferences, it tries to parse the distribution control file. If that fails, it drops you back to the Preferences dialog. Examine the Activity window for further clues about what RevRdist did not like. Correct the control file (which may involve quitting RevRdist if you aren’t running under MultiFinder) and try again.

When RevRdist is happy with the control file, it scans the client volume and compares it to the master folder on the server. The activity window lists each action RevRdist would take. You should examine this list to see if it agrees with what you intend. There is no way to save or print this list, and it is flushed when it exceeds 32,000 characters, so you may want to Pause RevRdist every so often while you check its recent activity.

If you are testing RevRdist from the client used to create the master folder, then you should be suspicious of any action RevRdist proposes. You created the master folder from that very client, so if RevRdist finds anything needing updating, then you are probably updating too much. One common mistake is to have the “Update” (u) action for the “Otherwise” (E) criterion. You almost always want “E-”.

Quit RevRdist once you are satisfied with what it proposes to do. Backup the preferences file with your settings to the server or a floppy. Start again by double-clicking on the preferences file still on the client. Turn off all the checkboxes when you get to the “Processing flags” dialog. Press the “Save” button. RevRdist should run as before, but without showing the Preferences dialog. This time, it is actually doing whatever it proposed earlier. When it finishes, select “Activity” from the “Windows” menu. This should be a much shorter list than before (because “Verbose” is turned off). In fact, it should show that RevRdist did nothing.

After this step completes satisfactorily, you should test your configuration when there is real work to do. Create extraneous files and folders on the client and remove files or folders that should be there. Double-click on the preferences file again. Because you changed RevRdist’s processing options when you pressed “Save” the last time, you won’t see the Processing flags dialog. Examine the Activity window when

RevRdist finishes to be sure it did what you expected. Then quit RevRdist and check that it really did what it said it did.

Do not be concerned if you see errors about busy files in the Activity window. This just means that RevRdist could not delete or update a file because it was in use. This can happen with the System file, RevRdist itself, or almost any file when MultiFinder is active. You can boot from a floppy if you need to update the client copy of the System file, RevRdist, Finder, etc. See the next section.

Running on demand

Although RevRdist is designed to run automatically at restart, there are situations where it is run on demand. One example might be if a client's hard disk is erased, perhaps because it failed and was replaced. To reload the client with the full set of files, you should keep a bootable floppy containing a copy of RevRdist and the preferences file appropriate for the client. Boot from the floppy. Copy the preferences file from the floppy to the client hard disk. Double-click on the preferences file on the client. Assuming your distribution control file is similar to the examples, the client disk should be completely restored when RevRdist finishes. You may want to give the client its Chooser name back.

You can simplify this somewhat if your clients have only one local hard disk volume. Again, start with a bootable floppy. Copy RevRdist and a preferences file to it. Make sure the preferences is named "RevRdist prefs." Configure this copy of RevRdist with the Unattended, Restart, and Find volume flags set and the others cleared. Selecting the Settings locked flag is optional. If the floppy runs System 6, select RevRdist and then use the "Set startup..." option on the "Special" menu to make RevRdist the startup application. If the floppy runs System 7, move RevRdist and the preferences file into the Startup Items folder inside the System folder.

Thereafter, you need only restart a client from the floppy. RevRdist runs automatically to update the local hard disk. When it finishes, it restarts the Mac from the restored hard disk, ready for use.

The same steps work to clean up a disk sooner than specified by the update interval of the INIT, or if the INIT is not used.

One case where you might not want to use the INIT is if the master folder on the server is not readable from an account without a password. Because the password is kept in clear text in the preferences file, and because the INIT requires the preferences file be on the client, so there is no way to adequately protect a password and still use the INIT.

Running automatically

Once your preferences and distribution control files are debugged, you can run RevRdist automatically every so often. To do this under System 6, move or copy RevRdist, RevRINIT, and the preferences file into the System Folders on the clients. Rename the preferences file to "RevRdist prefs". Under System 7, copy RevRdist and RevRdist prefs to the Startup Items folder within the System folder.

Next, you need to add to the preferences you set earlier. Start RevRdist by double-clicking on it. (Don't click on the preferences file this time.) You should get the Preferences dialog. Fill in the AppleShare server that holds the master folder and a user name and password that give access to that folder. You should supply these even if the client AppleShare normally mounts the volume with the master folder. This lets RevRdist run even if a user discards or changes the "AppleShare prep" file.

Still on the Preferences dialog, set the "Update interval" to the desired minimum time after one automatic RevRdist run until the INIT runs RevRdist again. If you make it small, the INIT will run RevRdist essentially every time the system is rebooted. Other typical values might be after eight hours or after a day.

Note that the INIT runs only when the Macintosh is restarted, so if a Mac is never rebooted, the INIT never gets a chance to run RevRdist.

If you make the interval too small (less than about two minutes), the interval will expire while RevRdist is running. This could cause an infinite cycle of rebooting, seeing it's time to run RevRdist, running RevRdist, rebooting, Ö

Finally, save the new values back to the “RevRdist prefs” file and quit RevRdist. Start RevRdist again, holding the mouse down until you get the Flags dialog. Turn on the “Settings locked” checkbox and then press the Save button. Quit RevRdist. You should now copy both RevRdist and RevRdist prefs to the master copy of the System folder (Startup Items folder for System 7).

Installing new versions of RevRdist or RevRINIT

You can install a new version of RevRINIT just by putting the new version in the master copy of the System folder. The next time RevRdist runs on a client, the new version will replace the old one. The next reboot after that will actually use the new INIT.

Installing a new RevRdist is only slightly more difficult. Again, you just put the new version on the master in place of the old one, but you must give it a different name. For example, you could number each version, RevRdist1, RevRdist2, etc. To install a sixth version of RevRdist, you remove RevRdist5 from the master, copy RevRdist to the master, and then change its name to RevRdist6. It doesn't matter what name you give the new version, so long as it is different from the names of the old versions on the clients. (Under System 6, the name cannot be longer than fifteen characters.)

The explanation is simple: the Mac won't let you delete or replace a running application. So one version of RevRdist cannot be used to replace itself. By giving the new RevRdist a different name, we let two copies of RevRdist exist on the client for a while. The old version copies the new version to the client. It also tries to delete itself, but this fails. Under System 6, the next time RevRINIT detects the update interval has expired, the INIT finds the most recent copy of RevRdist to start. Under System 7, at the next system restart, both the new and old copy start running, but the older version detects that there is a more recent version and quits before it does any updating. In both cases, the new RevRdist tries to delete the old version. This time it succeeds because the old version is no longer running.

Note: Starting with version 1.4, RevRdist can replace running or busy files (including the System file and RevRdist itself), but only if there is disk space on the client to keep two copies of the file. The busy copy is moved to the junk folder where it can be deleted by a later RevRdist run.

7.5 Using Assimilator

What Comes in the Package

The following files are included in the Assimilator distribution:

README!

A warning about the dangers of running Assimilator without reading the documentation.

Assimilator Admin

The main program that controls Assimilator.

Documentation

This document.

Register

A program that you should use when you decide to register Assimilator.

Programs

A list of our other public domain, freeware, shareware and commercial products.

You'll notice that there is no "Assimilator" program contained in the distribution. This is because you use the Assimilator Admin program to generate Assimilators that are customized for your environment.

Creating a Customized Assimilator

When you run Assimilator Admin it creates a new, untitled customized Assimilator and displays it in a window. The window contains five areas:

Source

This panel lets you specify the source folder for the Assimilator. You must fill out this before you can create an Assimilator that performs any useful action. Creating a suitable source folder is discussed in the next section.

Kind

The controls in this panel specify what type of file this Assimilator will be saved as. Understanding this is critical to understanding how Assimilator works and it is discussed later in this section.

Miscellaneous

This area lets you set various sundry preferences. It is fully described in a later section.

Label Actions

This area lets you associate specific action with specific Finder labels. These actions are described fully in a later section but normally the default actions are OK.

Edit Database

Each Assimilator also holds a database that it consults when it wants to set machine specific parameters. The database is discussed in a later section.

The key to understanding Assimilator is the Kind panel . Assimilator Admin works on documents - that is, customized Assimilators -- which are themselves either documents, applications or extensions. When you save a customized Assimilator from within Assimilator Admin it creates the appropriate kind of file, depending on the settings in the Kind panel.

If you save your customized Assimilator as an application then it will start Assimilating whenever it's launched. You can put it in the Startup Items folder (to run at system startup), the Shutdown Items folder (to run at system shutdown) or in a convenient place, so that users can run it manually. If you run Assimilator as a Shutdown Item you also need to put the Assimilator Helper extension in the Extensions folder (the extension tells Assimilator if the machine is being shutdown or restarted).

One problem with putting Assimilator in the Startup or Shutdown Items folders is that it won't run if it is invisible. [The Finder refuses to launch any invisible items.] To get around this you can save Assimilator as a system extension. Assimilator loads a small nub at startup time (like any other system extension) and waits for startup time to finish. It then automatically launches itself, which then assimilates the system.

Note: Assimilator cannot run at Shutdown time if it is a system extension.

Finally you can save the Assimilator as a document , which contains just the configuration information. This is a useful way of storing and transporting a configuration without any danger of it accidentally being executed. Also, at some points Assimilator Admin will only let you save a configuration as a document because the configuration does not contain enough information to be a valid Assimilator.

Creating a Source Folder

Once you have created your new untitled Assimilator, you need to start filling out some of the fields. The first step is to create a source folder . This is a folder on a file server that acts as the master folder for all the assimilated clients.

The best way to create a source folder is to set up a single client machine just the way you like it and then copy that machine's entire hard disk up to the file server. While there are some caveats with this method, it is a good way to start.

When you copy up the hard disk to the file server you will notice that all of the aliases on the server still point back to the hard disk. This is obviously incorrect. You need to fix those aliases so that they point to the appropriate item inside the source folder. That way, when Assimilator copies the alias down to the destination disks, it will fix the aliases.

Assimilator User and Source Folder Permissions

When Assimilator runs on the client machines, it needs to mount the server that contains the source folder. To do this it needs a username (and password) that has sufficient privileges to see the source folder. It is strongly recommended that you give Assimilator its own dedicated username (normally "Assimilator" but you can give it any name you like), and that this user be highly restricted. Specifically the Assimilator user should not have any permissions to write to any volumes on the file server. Also the user's password should not be the same as any of your administration passwords.

Note: The Assimilator user's password is inherently insecure because it is stored inside the Assimilator on each of the lab machines. Although the password is scrambled, to prevent idle snooping, it is not effectively encrypted.

You should now set the permissions on the source folder so that the Assimilator user can see all of the enclosed files and folders. The best way to do this is to make yourself (the administrator) owner of the folder -- with See Folders, See Files and Make Changes privileges -- and make the Assimilator user the group of the folder -- with only See Folders and See Files privileges. Don't forget to make all of the enclosing folders the same as this, otherwise you'll end up with a half-assimilated machine!

Now that you've created the source folder and the Assimilator account you should fill out the items in the Source panel of the customized Assimilator's window. In preparation for doing this, mount the file server using the Assimilator user's account and password and check that the user can see all of the items within the source folder. Then click button in the source panel and use the standard directory dialog to set the source folder. Then you should fill in the Username and Password fields with the Assimilator user's name and password.

Note: If you leave the username field blank then Assimilator will attempt to log in to the file server as a guest.

Ready To Run

At this stage you can save your customized Assimilator as either an application or an extension and use it in action. In fact, it might be a good idea to try it out. Just take a copy of the Assimilator to a lab Mac and run it. It should automatically mount the file server, synchronize the Mac's hard disk with the source folder and then restart the machine. If not you should consult the check list at the end of this section.

Note: Assimilator virtually always restarts the machine at the end of an assimilation. This is because it has almost always downloaded new versions of some files and the Mac must restart in order for these to take effect.

WARNING: If you save this customized Assimilator as an application then running it is extremely dangerous, because it will automatically start Assimilating the local hard disk. Do not double click it to edit it, because that will run it! Instead, use Open from Assimilator Admin's File menu to open the Assimilator.

After you have tested your newly created Assimilator, you might want to customize it further. To do this, use the Open command on Assimilator Admin's File menu to open the Assimilator. There are a number of extra configuration options you might want to explore. These are discussed in the next section.

Check List

These are the steps required to create a minimal customized Assimilator. If you have troubles getting it working then make sure that you have performed each of these steps.

1. Create a source folder by dragging a copy of the lab machine's hard disk on to the file server.
2. Create an Assimilator user on your file server.
3. Set the privileges of the source folder so that the Assimilator user can see all of the folders and files within the source folder.
4. Mount the server volume containing the source folder using the Assimilator username and password.

5. Check that you (that is, the Assimilator user) can see the source folder.
6. Run Assimilator Admin and create a new untitled Assimilator.
7. Set the source folder by clicking on the button in the source panel.
8. Set the Assimilator username and password using the items in the source panel.
9. Set the Kind to Application.
10. Save the Assimilator.

Further Configuration

Once you have got a working Assimilator you can continue to configure it. There are a number of configuration options you might want to explore in order to gain better security and efficiency.

Abort Password

When Assimilator is assimilating a client machine, it displays a dialog with a Stop button. It's obvious that a student in the laboratory should not be allowed to stop the assimilation, so the operation is password protected. You can set the abort password in the Miscellaneous panel of the Assimilator's configuration window.

When Assimilator is running and a user clicks the Stop button, they are presented with a dialog asking for the abort password. If they enter the correct password then Assimilator stops. If they enter the incorrect password then the assimilation continues. Finally the password dialog times out after 30 seconds, so the assimilation continues if no password is entered.

Minimum Free Space and Trash Management

The Miscellaneous panel also holds an item where you can set the minimum free space that Assimilator should guarantee. At the end of the assimilation, Assimilator will delete items in the Trash to ensure that this amount of space is free on the destination disk. Normally you would set the minimum disk free space such that there is enough space for standard user operations, such as editing a large word processor document. Assimilator defaults to a setting of 2MB.

Leaving the minimum free space low is good because, if a user accidentally saves a file on the local hard disk of a machine, it will stay around in the Trash for a while. On the other hand, if you are a particularly fascist system administrator, you can set the minimum disk free space to a very large number, thereby ensuring that any files thrown away by Assimilator are automatically deleted.

The exact trash management scheme it uses is described in the following paragraphs.

Under normal circumstances, Assimilator puts any files it wishes to discard into the Trash. Obviously this process cannot continue indefinitely; sooner or later the disk will fill up. So Assimilator is forced to make decisions about which files to delete. It does this by deleting the oldest files first. To do this it uses a sophisticated Trash management scheme.

When it commences an assimilation, Assimilator creates a folder in the trash called "Assim ddmmmyy hhmm". It then puts any file that it needs to throw away into that Assimilator trash folder. This means that all the files in the Trash are recorded by the time and date when they were assimilated.

When Assimilator discovers that the disk is too full it starts deleting files in the Trash. It progressively deletes the files in the oldest Assimilator trash folder in the Trash until there is enough space free. If deleting all of the Assimilator trash folders fails to yield enough disk free space then it starts deleting files at the top level of the Trash; these are normally the files that users have discarded.

Minimum Rerun Time

Just like a television station, Assimilator has a minimum time before it will run the same program again (: You can set this time minimum rerun time , in minutes, in the Miscellaneous panel of the Assimilator's configuration window. The default is 10 minutes.

This rerun time has a number of consequences. If you have Assimilator set to run at startup, either by putting an Assimilator application in the Startup Items folder or by putting an Assimilator extension in the System folder, then you must be careful to set the minimum rerun time to longer than the time taken for the machine to assimilate and then restart. If the time expires before Assimilator has had time to run, assimilate and restart the machine, then it will notice that the time has expired and automatically run again, looping forever, or until an exasperated system administrator fixes it.

At times it may be useful to set the minimum rerun time to 0 minutes. One common technique for setting up a lab is an Assimilator extension that runs at startup time to clean up the machine when it restarts and to have an Assimilator application (with an obvious name like "Clean Hard Disk") on the desktop. This way, if a user encounters a machine that is broken they can fix it by running the "Clean Hard Disk" application, somewhat more obvious than restarting.

You could also imagine a situation where you don't even have a copy of Assimilator in the Extensions folder and all hard disk cleaning is done under user control.

If you hold down the control key while launching Assimilator, it will run regardless of the minimum rerun time. This is particularly useful for testing changes to your source folder.

Note: The time of the most recent assimilation is stored in the Backup date of the Assimilator application or extension.

If you use a backup program that sets this date, Assimilator may run more than required. Also, if Assimilator downloads a new copy of itself (typically because you modified the version in the source folder), then the new version will not have been run and its backup date will be unset. This means it will always assimilate, even if it's just to set its own assimilation time. Under certain circumstances it is possible for Assimilator to run three times before things settle down.

The Database of Deviancy

The goal of Assimilator is to make all of the Macintoshes in the lab look identical. This is a worthy goal but, as we all know, too much of the same thing can be bad. Some settings, such as the Sharing Setup Macintosh Name, must be different for each machine. This section describes the database that Assimilator uses to set machine specific parameters on each lab machine.

Database Concepts

There is a database stored in each Assimilator document, application and extension. You can edit the database by opening the Assimilator with Assimilator Admin. The database comprises a number of records, with each record containing the following fields:

Identify By

This is the key field. It determines whether Assimilator should consider this record when it's assimilating a machine. It contains either an Ethernet address, a hard disk creation date or a machine ID file.

Machine Name

This field contains whether Assimilator should set the Machine Name and, if so, the value to which it should be set. The Machine Name is normally set by the Sharing Setup control panel and stored in the System file and the Users & Groups Data File.

Owner Name

This field contains whether Assimilator should set the Owner Name and, if so, the value to which it should be set. The Owner Name is normally set by the Sharing Setup control panel and stored in the System file and the Users & Groups Data File.

Volume Name

This field contains whether Assimilator should set the name of the destination volume. There are three options here: Assimilator can preserve the existing name, set it to the name of the source folder or set it to some value held in the database.

TCP IP

This field contains whether Assimilator should set the MacTCP or OpenTransport IP number. Normally the IP number is set via the MacTCP or TCP/IP control panel and the value is stored in the appropriate preference file.

When Assimilator runs, it walks down the records in the database looking at the Identify By key field. If this field matches the machine on which Assimilator is running then it uses that database record to set the various machine specific parameters. If no records match then Assimilator does not set any machine specific parameters.

Note: Assimilator sets most of these preferences by bashing the relevant files with great violence. A number of these bashes have certain compatibility drawbacks. Firstly, setting the Machine Name requires Assimilator to hack up the Users & Groups Data File, because a copy of the machine name is held in that file and there is no sanctioned way to modify it. Secondly, setting the IP number requires directly bashing the TCP preference files because there is no sanctioned way to do it otherwise. Assimilator will be revised to fix any compatibility problems that arise from future system software. Welcome to the world of MacLab hackery!

The Key to Identify By

There are three ways to identify a machine uniquely:

Ethernet Address

Every Ethernet card (or Macintosh with built-in Ethernet) must contain a 48 bit number that identifies the card uniquely. If your Macs have Ethernet then you definitely should use the Ethernet address to identify them.

Hard Disk Date

The creation date on the destination hard disk is a common way of identifying hard disks uniquely. It works because the creation date is stored in seconds and it's difficult to get two disks with the same creation date if you initialize them by hand. Unfortunately if you get a big batch of machines -- such as a new Mac laboratory for example -- you will often find that all the hard disks have the same creation date. If you reinitialize each of the hard disks manually then the creation dates will be unique and you will be able to use them as the key.

Machine ID File

The final, and least preferred, mechanism to identify a machine is by a machine ID file. This is a file in the Preferences folder whose name uniquely identifies the machine. The file must be of type 'Asim' and creator 'Asim', with zero length data and resource forks. Assimilator ignores such files when assimilating the hard disk. You must be extra careful to make sure that your machine ID files meet these requirements, otherwise they'll be assimilated.

Database Operations

Assimilator Admin contains all of the basic tools required to maintain the database. The Assimilator's configuration window contains an Edit Database button, which brings up a dialog showing each of the database records. You can click the Add button to add a new database record, the Delete button to delete the selected records, the Change button to edit a single record, and the Done button to dismiss the dialog.

When you add or change a database record you encounter the Add/Edit Entry dialog, which allows you to set the fields of the record. The meaning of each of the fields is described in a previous section.

Adding Entries Quickly

The most common problem you face as a lab administrator is having a whole bunch of machines whose Identify By keys (typically their Ethernet addresses) are unknown. Fortunately, Assimilator Admin has a shortcut for adding lots of individual entries easily.

The first step is to put a copy of Assimilator Admin on a floppy disk. Now put an Assimilator document, application or extension in the same folder as the Assimilator Admin program. Now go to a lab machine and run Assimilator Admin while holding down the control key . Assimilator Admin will bring up the Quick Add Entry dialog.

This dialog has the Identify By information set up in the best way possible. If the machine has Ethernet then the key is set to the Ethernet address of the Ethernet card. If the machine does not have Ethernet then the key is set to the hard disk's modification date but you can easily switch it to a machine ID file.

After you've confirmed the Identify By key field, you can simply type in the other information associated with this database record. When you click OK Assimilator Admin will automatically add this record to the database contained in the Assimilator document, application or extension in the same folder as it and then quit. You can now eject the floppy disk and take it to the next machine and repeat the operation.

Note: If you OK the dialog when the machine ID file option is set then Assimilator Admin will create the file for you.

The Import/Export Business

Because Assimilator Admin's facilities for editing the database are rather basic, it also provides a facility to export the database to a text file and then reimport a modified database from a text file. The format of this text file is rather hard to explain. If you're interested you should export a simple database and work it out from that.

Fine Tuning

Once you have got Assimilator working then you can spend considerable time fine tuning it for your particular installation. This section describes some of the useful mechanisms for doing this.

Assimilator Log

When Assimilator runs it creates a log of all its actions in a file called "Assimilator Log" in the Assimilator trash folder. You can look in this log for a description of the actions taken and the reasons for those actions.

Interpreting the log is the key to fine tuning Assimilator. Each line in the log starts with an action and ends with a reason. For example, imagine some has thrown away the ClarisWorks application on the hard disk. When Assimilator runs it sees that the ClarisWorks application is missing and downloads a new copy from the server. The log entry would look like:

Downloading "ClarisWorks" because it is missing.

The log actions are:

Startup

Assimilator records the date and time that it runs as the first item in the log.

Finish

Assimilator records the date and time that it finishes as the last item in the log. You can use this to calculate the time taken for assimilation.

Trashing

Assimilator is throwing away an item; the reason is given as part of the log entry.

Downloading

Assimilator is downloading an item; the reason is given as part of the log entry.

Blessing

Assimilator will automatically bless any System Folder that it downloads and record that action in the log. The exact criteria for a System Folder to be blessed is covered in a later section.

Deleting

Assimilator is deleting an item. The primary reason is given as part of the log entry. Items are only deleted if they cannot be held in the trash while maintaining the minimum disk free space requirement.

The log reasons are:

it is missing

The item is present in the source folder but missing on the destination hard disk.

one is a folder, one is a file

The item is a file on the source folder and a folder on the destination hard disk or vice versa.

moddate

The item on the destination volume has a different modification date from that in the source folder.

file type

The item on the destination volume has a different file type from that in the source folder.

file creator

The item on the destination volume has a different file creator from that in the source folder.

rsrc length

The item on the destination volume has a different resource fork length from that in the source folder. Basically this means that the file has changed size.

data length

The item on the destination volume has a different data fork length from that in the source folder. Basically this means that the file has changed size.

it is marked to always download

The item on the server has been marked as "Always Download" using a Finder label.

it is in the way of a download

The item on the destination volume is being discarded because a replacement item is being downloaded. This entry always has a matching entry that explains why the replacement item is being downloaded.

it should not be here

it should not be here (root)

The item on the destination volume is being discarding because it is not in the source folder. Don't ask me why there are two different messages for this!

we are zarching the entire hard disk

Assimilator is zarching the entire hard disk and thus it discarding all files.

the desktop folder should be empty

There are items on the destination volume's desktop but the desktop is supposed to be empty.

Labeling Your Source Folder

Once you have got your machines assimilating happily you should look at your log and see what actions Assimilator is taking and why. What you will find is that a number of files are being downloaded every time Assimilator runs. This is usually because the files are modified in the process of system startup. Unfortunately the most serious offender is the System file, which is also very big. So it's worth your time to analyze the Assimilator log and see how things can be improved.

The way you stop a file being downloaded every time is to label it (using the Finder) to indicate that to Assimilator that it is to take a special action. By default the labels and there actions are:

None -- Download if Modified

Essential -- Always Download

Hot -- Download if Different

In Progress -- Download if Missing

Cool -- Never Download

Personal -- Download if Modified & Make Invisible

Project 1 -- Always Download & Make Invisible

Project 2 -- Download if Different & Make Invisible

You can change these default actions using the instructions in the next section but normally there is no need to do so.

The default action, which corresponds to no Finder label, is to download a file if it has been modified. The definition of modified, as far as Assimilator is concerned, is that either the file has changed or the modification date has changed.

In the case of the System file, the modification date changes every time the machine restarts. You can tell Assimilator to ignore this modification date by labeling the file with the Hot label, which means the file is only downloaded if it is different -- where different means that the file size, type or creator has changed.

If, after you've labeled it Hot, the file is still being downloaded, then you can take more drastic action. If you label the file as In Progress then it will only be downloaded if it's missing, that is the file is present on the source but not on the destination. Although this is generally not recommended for use on files it is useful for recalcitrant files whose modification date and size change at system startup. This label acts slightly differently on folders: if a folder is marked as download if missing and the folder is present on the lab machine Assimilator does not search inside that folder, it just moves on. If the folder is missing normal Assimilation occurs and the folder, along with its contents, are downloaded according to the labeling. Very useful for creating User folders.

The other labels are useful too. The Essential label tells Assimilator to always download the file every time. This is useful for files like the MacTCP DNR file, which regularly get corrupted. Because the file is small, the load from downloading it every time is minor, certainly compared to the hassle of having to deal with a corrupted one.

The final action is achieved using the Cool label, which corresponds to the never download action. Assimilator will never download an item that is labeled Cool but it will also not throw it away. One possible use for this is to create a "Users" folder, where users can store their own personal files. If you label it Cool then Assimilator will not attempt to download the folder from the source (very sensible because it would be empty on the server) and it will not discard the folder on the destination.

The remaining Finder labels (Personal, Project 1 and Project 2) and analogues of some of the other labels except that they also tell Assimilator to render the item invisible after it has downloaded it. This is useful in scenarios where you want to project the System Folder on the destination disk from casual tinkering. While you can't make the entire System Folder invisible (the Finder gets upset) you can make the

Extensions folder, System and Finder invisible. This ensures that the casual tinkerer can't prevent a clean system restart, which in turns means that Assimilator will run and clean up the machine at that time.

Note: Assimilator will download invisible files in the source folder and render them invisible on the destination disk. The invisible labels are useful when you want the files to be visible on the server (so that you operate on them, for example installing new Fonts in the System file) but invisible on the lab machines.

One helpful hint is to change the name of the Finder labels (in the Labels control panel) to correspond with their Assimilator actions.

Action Reference

Assimilator supports the following actions:

Download Always

The item (and its contents if it's a folder) is always downloaded. Any existing item on the destination is always trashed.

Download if Modified

If applied to a file, the file is downloaded if the one on the destination is different from the one on the server or the file's modification date have changed. If applied to a folder this just causes Assimilator to recursively operate to items within the folder.

Download if Different

If applied to a file, the file is downloaded if the one on the destination is different from the one on the server. The definition of different is the file size, type or creator has changed. If applied to a folder this just causes Assimilator to recursively operate to items within the folder.

Download if Missing

The item is downloaded if it is in the source folder but not on the destination disk. If applied to a folder Assimilator does not recursively operate on items within the folder unless the folder is missing on the destination disk. As a special case, if there is a file in the source and a folder on the destination (or vice versa) the item is considered to be missing.

Never Download

The item is never downloaded. If the item exists on the destination then it is preserved. If it doesn't exist on the destination then nothing happens. If there's a file on one and a folder on the other then still nothing happens.

Any of these actions can be combined with the invisible action, which forces the item to be invisible on the destination disk.

Customizing Label Actions

There are 5 basic Assimilator actions:

- ⊙ Download if Modified
- ⊙ Always Download
- ⊙ Download if Different
- ⊙ Download if Missing
- ⊙ Never Download

There is also one modifier, the invisible checkbox, which determines whether an item is to be made invisible after it has been downloaded. This yields 10 (2 times 5) combinations. Eight of these combinations are associated with the eight Finder labels. By default the associations are:

None -- Download if Modified

Essential -- Always Download

Hot -- Download if Different

In Progress -- Download if Missing

Cool -- Never Download

Personal -- Download if Modified & Make Invisible

Project 1 -- Always Download & Make Invisible

Project 2 -- Download if Different & Make Invisible

You can change these associations using the popup menus and checkboxes in the Label Actions panel of the Assimilator's configuration window. Normally the default associations are correct and you don't need to change anything.

Miscellaneous

This section contains a whole pile of hints and tips that didn't fit anywhere else.

Desktop Database

In order to support double clicking documents, the Macintosh maintains a list of applications available in something called the desktop database. When it downloads or trashes an application (technically anything with a bundle), Assimilator attempts to update the desktop database. Normally this works fine and you do not need to worry about it but in certain circumstances it may be necessary to rebuild the desktop database completely. You can do this by holding down the command and option keys while the Finder starts up (usually at the end of the startup time).

The Desktop Folder

Under System 7, items that are "on the desktop" are actually held in a special, invisible folder called "Desktop Folder" that lurks at the root of each hard disk. You would think that the way to achieve this is to create a "Desktop Folder" inside the source folder and put the items you want on the lab machine's desktops in that folder. Unfortunately it's not that easy!

Firstly, the Finder won't let you create a folder called "Desktop Folder" because the name is "reserved by the system software". This is fairly easy to work around. If you create a folder called "Desktop Folder" (immediately inside the source folder) then Assimilator will treat it as if it's the desktop folder.

Secondly, it's quite hard to get the icons to appear in the right position on the desktop. There are a number of problems with this, not all of which are well understood. The best solution is to follow the advice given earlier in this document, whereby you create the source folder by dragging the entire hard disk of a prototype lab machine up to the file server. This will create a "Desktop Folder", including all of the icons in the right position. You should then modify that folder carefully, making sure to always keep it in the big Icon view.

Blessed are the System Folders

The Macintosh keeps track of the currently active System Folder on each disk because it needs that information at startup time. For historical reasons this folder is known as the "blessed folder" and the act of setting this folder is known as "blessing a folder". Blessing the wrong folder will prevent the Macintosh from starting. In order to guarantee that the correct folder is blessed, Assimilator will bless any folder that contains a System file. It recognizes the System file by its file type 'zsys' and creator 'MACS'. You should make sure that you only have one System file in your source folder and that it is the correct System Folder.

Assimilator adds a log entry whenever it blesses a folder. If you have blessing problems you should consult the log to find out what went wrong.

Special Files and Folders

Because Assimilator is fully System 7 aware it knows about, and deals correctly with, a number of special files that are maintained by System 7. These are documented here for completeness:

- ⊙ AppleShare PDS
- ⊙ Move&Rename
- ⊙ Network Trash Folder

These file and folders are found in the root directory of the hard disk and used by file sharing.

- ⊙ Desktop DB
- ⊙ Desktop DF
- ⊙ Desktop

These files are kept in the root directory of the hard disk and make up the desktop database.

Trash

This folder holds the items that are in the trash.

Shutdown Check

This file is created by the System 7.5 General Controls control panel shut down warning check.

VM Storage

This file is used by the System 7 VM system.

Aliases Explained

Assimilator attempts to correct any aliases that it downloads from the source folder. This process is a little tricky and it deserves an explanation. When it downloads an alias from the source folder, Assimilator resolves the alias. If the alias points to some item within the source folder then Assimilator corrects the alias so that it now points to the corresponding item on the destination hard disk. This is usually the Right Thing .

For example, lets say your source folder is called "Source Folder" and is held on a volume called "Source Volume" and you have an alias "Source Volume:Source Folder: Desktop Folder :SimpleText alias" that points to the application inside source folder, that is "Source Volume:Source Folder:Applications:SimpleText". Assimilator will download the alias to "Dest Disk:Desktop Folder:SimpleText alias" and the application to "Dest Disk:Applications:SimpleText". If Assimilator took no other action than the alias on the desktop folder of the destination disk would still point to the source folder and not to the copy of SimpleText on the hard disk. This is obviously the Wrong Thing . Instead Assimilator corrects the alias so that it points to "Dest Disk:Applications:SimpleText".

Whenever Assimilator fixes an alias, it adds a log entry describing its actions. In fact, it's a bit verbose in reporting it's actions, so it will list all the aliases it looks at and will show "errors" for the ones it decides not to fix (for example, aliases pointing to somewhere outside the source folder). The rule here is "don't be alarmed.Ó

Destination Disk Choice

Each time Assimilator runs, it's faced with the choice of which disk to assimilate. Surprisingly enough, it's actually quite hard to determine this. Assimilator uses the following algorithm: it assimilates the first disk that is larger than 2MB (this eliminates floppy disks), is not hardware locked (which eliminates CD ROMs) and is not an AppleShare volume (which means it won't try to assimilate your file server). You can use this algorithm to your advantage when building an Assimilator floppy, as described in a later section.

Time is Relative

It's important to remember that Assimilator uses the modification date of files to determine whether they've changed. If the times on your server, your administration machine and your lab clients are different then Assimilator may erroneously conclude that files have been modified even when they haven't. I recommend that you install a time synchronization utility to ensure that the time is the same on all of your machines. Time synchronization programs include LocalTime (freeware), Network Time (shareware), LabMaster (shareware) and KeyServer (commercial).

Backdoor Reference

Assimilator has a surprising number of backdoors, that is special key combinations that cause it to perform some special operation.

control key while launching Assimilator

If you hold down the control key while launching Assimilator, it will ignore the minimum rerun time run immediately. This is described better in the section "Minimum Rerun Time".

control and shift keys while launching Assimilator

If you hold down the control key and shift keys while launching Assimilator, it will 'zarch the destination disk'. This process involves throwing away all existing files on the disk and downloading all new copies from the source folder. Zarch does not delete the desktop database files or any of the other special files. Zarch is the best way to completely rebuild a suspect Macintosh.

control key while launching Assimilator Admin

If you hold down the control key while launching Assimilator Admin, it will display the Quick Add Entry dialog. This is described better in the section "Adding Entries Quickly".

Assimilator Boot Floppy

It is possible to build a single floppy disk that will boot any currently available machine and assimilate it. This task has been greatly simplified by the System 7.5 Network Access Disk that was recently released by Apple. At the time of writing, a disk image for this is available from:

<ftp://ftp.info.apple.com//Apple.Support.Area/Apple.Software.Updates/US/Macintosh/Utilities/Network_Access_Disk_7.5.sea.hqx>

There are two approaches you can take but both of them start with the Network Access Disk. The first approach merely requires you to boot the machine from the network access floppy, put it on the correct network using the Network control panel on that disk, then log into the file server using AppleShare (also on that disk) and run Assimilator from the file server.

The second approach is more automated but it's also more work to set up. First you start with a network access floppy. You then throw away the Finder on that disk, which frees up considerable disk space. Then take a copy of your customized Assimilator application and change its file type to 'FNDR' and its creator to 'MACS'. Rename it to "Finder" and put it in the System Folder of the network access floppy. Assimilator will now automatically run when you boot a machine from that disk. Because of the destination disk algorithm (discussed in an earlier section) it will choose to assimilate the machine's hard disk and not the floppy.

Note: If your machine is on Ethernet then it's possible that the above procedure won't work because the machine has accidentally reverted to LocalTalk. You can get around this either by using the previous procedure or by including a PRAM zapping extension on the boot disk. Such a utility is available as:
<ftp://redback.cs.uwa.edu.au//ComSci/LabUtilities/SetXParam.sit>

Even More Miscellaneous

Startup Delay

Assimilator waits for a few seconds at the beginning of each assimilation to give you time to abort it without damage. This is vitally important if you accidentally try to assimilate a non-lab Macintosh.

Force Quit

Assimilate disables System 7's Force Quit key combination (command-option-escape) while it is running.

Progress Bar

It is impossible to provide a decent progress bar for Assimilator without doubling the time it takes to assimilate. Please stop asking for one!

You Need More Power???

Assimilator, would you believe, is designed to greatly simplify the process of maintaining laboratory Macintoshes. To do this, I have avoided adding a number of features that could conceivably be useful for you. If you need more power for your Mac lab, try out RevRdist by Dale Talcott <aeh@cc.purdue.edu>. It's freeware (as opposed to Assimilator, which you are expected to pay for) and has tons of power (more power than I could ever hope to deal with). It is on the UMich archives in util/network/.

Conclusion

It is very easy to get confused when trying to set up Assimilator. The whole issue of maintaining Macintosh laboratories is rather complicated and Assimilator is a merely one step on the path to a solution, not the entire solution.

Assimilator is the product of years of experience running Mac labs and it works well if you set it up correctly. When you're confronted by problems please take the time to read this manual carefully. It contains a wide array of hints and tips for setting up your lab.

8 References

8.1 IT Lab Team Resources

Terry Zimmerman's GSLIS IT systems analysis:

<http://ccwf.cc.utexas.edu/~terryz/acad/systems/sys.html>

GSLIS 1997-98 vision plan: <http://www.gslis.utexas.edu/program/vision97.html>

UT Computer Security policy: <http://www.utexas.edu/admin/dp/computer.security/comsecurity.html>

Intrusion Detection, Inc.: <http://www.intrusion.com/>

8.2 LabMan and related products resources

ACITS: <http://www.utexas.edu/cc/>

Departmental Services: <http://www.utexas.edu/cc/ds/>

LabMan Home Page: <http://www.utexas.edu/cc/ds/labman/>

LabManager Labs at University of Texas: <http://www.utexas.edu/cc/ds/labman/lmlabs.html>

LabMan Users Group: <http://www.utexas.edu/cc/ds/labman/lug.html>

SMF Usage Data: <http://www.utexas.edu/smf/stats/stats.html>

PC-RDist: <http://www.pyzzo.com/>

<http://www.geneseo.edu/cit/helpdesk/refdesk/prdist/>

<http://fpg.uwaterloo.ca/projects/envs06/NT95labissues.htm>

<http://128.253.152.16/windows95/Details.htm>

<http://act.kent.edu/stapp/win95rem.htm>

Yale University's PC-Rdist listserv: To subscribe, send a message to:

pcrdist-request@cs.yale.edu

with the following in the body of the message:

subscribe pcrdist [youraddress].

PC-Rdist update listserv: Send an email to:

pcrdist-updates@pyzzo.com

with the following message in the body of the text:

subscribe pcrdist-updates [youraddress].

Threaded and chronological indexes from the listserv are linked off of the PC-Rdist mail list page at:

<http://www.pyzzo.com/pcrdist/list/index.shtml>.

In addition, a keyword search engine can also be used to access the mailing list archive.

RevRdist: <http://www.purdue.edu/revrdist/>

<http://www.utexas.edu/cc/micro/labman/training1/page4.html>

Assimilator: <http://www.stairways.com/assimilator>

<http://www.stairways.com/maillinglists/maclabmanager.html>

<http://www.stairways.com/assimilator/labs/labadmin.html>

<http://www.shareware.com/share/peterlewis/assimilator/index.html>

This includes Frequently Asked Questions, Documentation, Registration information, download links for updated versions of Assimilator and additional notes and links related to lab management.

Technical support information: support@stairways.com.au

University of Georgia's Lab Manager: <http://www.graviton.com/lm/>

8.3. Training Resources

ACITS: <http://www.utexas.edu/cc/faqs/> (ACITS Frequently Asked Questions list)

<http://www.utexas.edu/cc/rack/> (ACITS Document Rack)

Online tutorials: <http://www.utexas.edu/cc/training/handouts/tutorials.html>

Training Services' Handouts: <http://www.utexas.edu/cc/training/trsdocs.html>

The Help Desk's Help Resources: <http://www.utexas.edu/cc/docs/resources.html>

Other training sessions and tech support events: <http://www.utexas.edu/computer/tech/event.html>

8.4 RevRdist Listserv Information

The Purdue University Computing Center also provides a listserv dedicated solely to solving RevRdist related problems. To subscribe, contact the listserv at: listserv@quest.cc.purdue.edu, and type the following in the body of the message: *subscribe revrdist your name*

Once you have received confirmation of your list subscription via email, you can send messages to the list at: revrdist@quest.cc.purdue.edu.

To reach a person for help specifically related to the RevRdist list, send a message to: revrdist-request@quest.cc.purdue.edu

To reach a person for general questions about or problems relating to the listserv, send a message to:
listserv-request@quest.cc.purdue.edu