



essentials

the newsletter of
Computer Systems Support
March 2003, v. 4 No. 1

Disk Quota and Disk Space

CSS increased disk quota for student, staff, faculty, and class accounts over spring break. Both the previous and the new soft quota are shown below. The number in parentheses is the hard quota.

engineering students: 200MB -> 300MB (350)
faculty: 600MB -> 800MB (850)
staff: 325MB -> 450MB (500)
class: 200MB -> 350MB (400)

CSS tries and does regularly increase disk quota for engineering account holders. In April 2002 we increased disk quotas by 50% because of increased capacity on the file server. This increase is due to the work CSS did at the winter break and analyzing disk usage data to determine what disk space is available.

Read also the article "Increased Mail Quota" in this issue. Mail quota was increased at the beginning of March. Disk quota is the amount of space you have to store your Unix and Windows files, which are stored in a single user account that can be seen from either type of computer. Mail quota, separate from disk quota, is used only by mail messages and their attachments.

Overflow and Project Disk Space Coming

CSS is working on a plan to implement semester-long access to a very large overflow disk and group project space. This disk space will be available to anyone with an engineering account and is intended to help students who need more disk space for specific classes than individual account quotas allow and for group project space when classes require team work.

The space will **not** be backed up and will be deleted at the end of every semester. There will be no quotas implemented on this space. We do not have the capacity to backup this extra disk space; otherwise we would simply incorporate it into our quota system.

Our current plan is to have this in place by fall 2003. The plan requires purchasing and configuring the disk storage hardware, configuring lab workstations for easy backup, developing administrative procedures, and testing the final product. Will you use this space? Do you have comments?

Need Additional Space?

Students can now purchase extra disk space at the rate of \$5.00/100MB/month.

Disk Space Costs Decreasing

Beginning 1 July 2003, the cost for additional (beyond your quota) email and file server disk quota increases will decrease to \$5.00/200MB/month.

~ Susan Beckett



In This Issue

Disk Quota and Disk Space	1
How Software Gets There	2
EASY Improvements	4
LISTSERV	5
Class Accounts Archive	5
Software Funding	5
Increased Mail Quota	6

How Software Gets There



No jargon summary: *Installing software on Windows XP is an arcane and not completely quantifiable process. This is probably a self-serving article, but the new reality of installing Windows XP software is still rather like science fiction to us.*

The steps to installing Windows XP software are many and often iterative. If you don't read about the entire process, notice the number of steps.

If you read the *Software Funding* article, you know some of the process that gets software purchased. Below is a description of what CSS does to get a piece of software from the consideration stage to installed. You might be interested in noting this often circuitous route if you have ever thought that you could have something installed much faster than it appears CSS can.

Requesting Software

When CSS knows about a software request, CSS staff collect requirements — anticipated use of software, number of copies needed, etc. — from the requester. Then they contact the vendor to get licensing information: are there quantity discounts? the cost to purchase the software and yearly maintenance versus leasing the software? site license available? (This is our preference because a site license means CSS can install the software on all machines.)

Susan finds the best pricing option, outlines the cost share for each option (see the “Software Funding” article), and gives that information to the requester, who decides if funds are available. Sometimes one department does not have the funds to purchase desired software but thinks another department or individual might be interested in a joint purchase. Upon request CSS gives the purchase information to interested departments.

If funded, CSS orders the software. It arrives. Depending on which platform the software will run on there are some or many things that must happen before that software appears on the menu and is usable.

Unix Installation

1. Susan installs the software on the application server and the license on license server. The software runs. Susan asks the requester to test the installation, and if functions are missing, she fixes the installation. The Commercial Apps menu is updated and the software usable.

Windows:

All software that runs on Windows XP must be repackaged for network deployment. (See “Why Repackage?” below to understand why.) Because the steps below can be iterative, I've numbered them for easy reference. *[Editor's note: A diagram of this process might illustrate it better. But I didn't know a good way to illustrate all the iterative loops.]*

1. Install software on a test machine on the network. If installation settings are required, CSS may ask the requester to test the test install. If this installation is successful, move on to step #2. If the installation fails, this software won't run with Windows XP and cannot be installed, but this has not happened yet.

The repackage process starts here.

Why Repackage?

CSS repackages software to install applications on several hundred machines and to maintain a consistent environment. Repackaging allows software to repair itself if files get deleted accidentally. It also allows CSS to apply a transform file (for patches and upgrades) to an existing MSI for deployment on top of an existing installation. It is the only reasonable way to handle installing software on the number of machines CSS manages. The alternative is assigning staff to go to each machine, log in as administrator, and install each package. This method would be like having a small crew paint the Golden Gate Bridge; by the time the crew reached the far end, the starting end would need repainting.

In the future some software will be distributed as an MSI. This will reduce the number of installation steps and allow CSS to use those distributed MSIs to deploy the software and eliminate the need to repackage in some cases.

Back to the Process

2. On a clean machine (a machine with the OS [Operating System] and a network connection but no additional software) take a picture of the machine. This picture is made with software that looks at the registry settings and other pieces of the OS to record the machine's state before installation.

3. Install the software.

4. Take another picture. This picture records differences in the beginning and final states of the registry, OS, etc. Occasionally the repackager fails to complete the analysis step. Failure means go back to step 2, build on a clean machine, take the picture, and so on.

5. Import the 2nd picture into software that converts the picture information, i.e., the legacy application, into a database that represents installation data (shortcuts, registry keys, logic about installation of software). This is the project file.

6. Clean the project file. This process must be done manually. Determine if some of the information in the project file can or should be removed. This is an imperfect process, though we are getting better with more experience. It is also an iterative process.

7. Use the cleaned up project file to create an MSI (Microsoft software installer; .msi is the extension of this file; this is Microsoft's windows installer package). An MSI is a file that can be installed silently, that is, installed network wide without any user interaction. An MSI installs the application as well as adding a shortcut to the Programs menu or other locations. (From msdn.microsoft.com: "An MSI installation can also provide other services such as installing files into preferred locations, setting up a local database, placing items into a user's Favorites folder, and so on.")

8. If the MSI builds with errors*, correct it as you are building it. If that doesn't work, go back to the original installation on the clean machine and start from that point, #2 above. You may also have to look up specific registry key values on the test machine to compare with those in the MSI.

* Errors occur when the legacy application (step #5) does not follow the Windows installer rules or the conversion process misinterprets data in the picture. The strings can be quite complex and the rules for parsing them complicated.

9. Install the MSI on a test machine that includes the lab build [the applications installed on the Windows computers in Hering and Elder labs].

10. Try basic software functions (open, save, help...). If basic functions work, go to #11. If not, return to the project file, steps #7 and 8. Review the specific component information and compare that with the actual installation to try to figure out what needs to be corrected.

11. Ask requester to test software. Because CSS is able to test only the basic functions, it is often only at this point that CSS can tell that things don't work as needed for class use. One common reason for failure is that the software expects to be able to write anywhere on the C: drive. (There are other reasons for failure). If there is no failure, go to step #13.

12. If some functions fail (here's another iterative part),

a. Search the project file and look at the original installation for registry keys and the structure of file system for reasons for failure.

b. If you can, fix the project file, recreate the MSI, and retest. If success, go to #13. If not, return to #12a. If you cannot fix the project file, then look for another way to install the package, possibly an unattended install.

c. If needed, call vendor for help tracking down details about the legacy package requirements. Use this information to fix the MSI.

d. Some fixes cannot be implemented in the MSI. For those, write a Perl wrapper script to handle things like group policies and file permissions.

e. Reinstall on test machine and test or have requester test function that failed. If no failure, go to step #13.

Continued on page 4...

EASY Improvements

Based on feedback from both faculty and staff who used the course Evaluation and Assessment Survey (EASY) system, CSS made major improvements to EASY. Faculty should find it easier to create and test a survey, and then to evaluate the results. Students should find surveys easier to take. More surveys should be written and taken. And CSS should spend less time and money maintaining EASY. Why? Because EASY is now web based.

Create, Test, Take a Survey

The first version of EASY required that CSS install software on faculty machines for creating surveys or analyzing their results. It also required students to take the survey on a college-administered machine that had the survey software installed. Feedback from many people was that they would prefer to take surveys on the web. Now only a browser is needed. Students can run EASY in secure mode from a browser. Faculty can easily create, preview, and test a survey for their course, using the same program run from a browser as the students use. Thus, faculty will see what their students see when the survey is run.

So far, more students have used the web-based EASY than used the first version. The new version was used for all course evaluation surveys in fall 2002 and for a pilot test of another application for electronic surveys. 48% of the 5,743 surveys offered were taken. This compares with 39% of 3,772 surveys offered in spring 2002 being taken.

Analyze the Results

Analyzing survey data will be easier soon using new web-based, data base report application server software and a CSS-written front end from which to run data base reports. This new system allows faculty to easily obtain reports from their own data as needed, and removes the need for Carrie Hogarty to run the reports and distribute them for each faculty. Anyone with web access will be able to securely connect to a web site and obtain a report as long as the person enters the appropriate login. As above, this new system means that CSS doesn't have to install specialized software on faculty and staff machines.

... Continued from page 3

13. Copy software package (a directory that includes the MSI file plus all the pieces related to the software) to network server for deployment.

14. Matt creates the group policies needed to install software at the weekly (Sunday) reboot.

15. Ready to use. The next time you login in after the Sunday reboot, you see the software listed in Start | All Programs | ...

Time

Besides that the entire process takes time, at many points there can be iterative loops of trying to fix one thing that may result in finding another thing to fix. And there can be nested iterative loops of fixing one thing that requires going back to an earlier step with a fixed file and going through the entire process. And at any point in the repackaging process, we may need to call the vendor. Vendors don't always immediately know why an error is generated and so have to do their own research. And vendors don't always reply to questions immediately.

End Result

The process of getting software ready to deploy is long and sometimes tortuous. The final result, though, is a consistent environment and software that is easy to patch and upgrade. As noted above, we are getting better at knowing what keys and files to look at when there are problems in installation, but installing Windows software is not the clean and well-understood process that we are used to with Unix software.

-Diana Harris with lots of help from Susan Beckett



The new web-based EASY systems will be easier for students, faculty, and staff to use, which in turn will provide the college with more course assessment and other data.

-Jim Cramer

LISTSERV



ITS chose to replace its majordomo list software with LISTSERV. The conversion is in progress. This may be of interest to you if you want to create a new mailing list, for a class or a group of people to whom you regularly send email. Use LISTSERV. See the ITS web pages for information: <http://www.its.uiowa.edu/cs/email/listserv/>

LISTSERV® by L-Soft software allows users to create and maintain mailing lists via e-mail or through a web interface. LISTSERV® supports a variety of email types of lists including announcement only and moderated and unmoderated discussion groups. It also will interact with ODBC-compliant databases such as Oracle. Every list can be maintained through a simple web interface. The list archives are available on-line.

If you have a majordomo list, you have already received email from ITS about the transition to LISTSERV.



Class Accounts Archives

Every semester we back up the class accounts. Due to a corruption of the backup media, class account archives for the spring 2002, summer 2002, and fall 2000 semesters are not available. We have archives for: December 1999, May 2000, May 2001, August 2001, December 2001, and December 2002.

Beginning in May 2003, CSS will archive class accounts to DVD rather than to tape. So long as CSS has equipment to read those DVDs, the archives will be kept and available if needed.

-Susan Beckett

Software Funding: The New Model

Inspired by how software is paid for at other Big 10 engineering colleges, CSS and the Dean's office began last year to create a new model. That new model for software funding and acquisition has been in place for more than a year, but you may not have heard about it or understand what it offers and to whom. Faculty should be interested because this policy describes what educational software is on the Engineering Computer Network and how it gets there. Student should care because student computing fees pay for software capital purchases.

This article summarizes the entire policy, which you can read at http://css.engineering.uiowa.edu/account_pol/software_acquisition_guidelines.html

All software purchase decisions reside with academic departments. Software is funded by a cost-share model. CSS, using student computer fee funds, provides 100% of purchase costs for new commercial educational software and contributes 25% to annual maintenance renewal costs. The department share for new software purchases is 75% of the annual maintenance cost. This cost-share model applies to the educational portion of any license agreement; research software must be paid for by requesting departments or faculty.

Because software installation is an art, and sometimes an opaque one at that (see article "How Software Gets There"), the policy includes guidelines that software must meet to be installed on the ECN and paid for using the cost-sharing model above. Such things as annual maintenance and support network licensing are included. These guidelines apply to purchased as well as donated or free software.

The process for renewing software and requesting new software begins in March. CSS has already sent software renewal notices to each department.

Adding software is one part of the process; removal is another. The policy also describes software removal.

-Susan Beckett

Increased Mail Quota

At the beginning of March, CSS increased mail quota for students by 66% (up to 25mb) and for faculty & staff by 30% (up to 130mb). Mail quota is separate and distinct from home directory quota. To see how much of your mail quota you have used, go to the Account Tools page

(<http://imap.engineering.uiowa.edu/mailtools#mstatus>) and check your Mail Status. Here is a sample response:

Mail space used: 31280K (31.28M) from 130000K (130.00M) allocation, with 98720K (98.72M) left

E-mail forwarding: turned OFF

Vacation Auto-reply: turned OFF

In this example, I have used 31280 kilobytes, which is 31.28 megabytes) of the allocated 130 MB and 98.72MB is left. The reply also tells you if email forwarding and vacation auto-reply are off or on.

To see your home directory quota from Windows XP, go to **Start | All Programs | COE Tools | Disk Quota**. From Unix, type the command **quota -v**. Information about disk quota and tips on managing it is available at http://css.engineering.uiowa.edu/tools/etudes/disk_quota_usage.html

-Diana Harris

Coming Attractions

CSS will install new wireless access points and so upgrade the wireless network. SSH2, more secure than SSH1, is on many Unix workstations. More information in the future.

Essentials is a publication of Computer Systems Support (CSS) in the College of Engineering at the University of Iowa. Essentials is published during the year whenever there is sufficient news to report to the user community. You have permission to reprint all or part of this newsletter for nonprofit purposes if you 1) acknowledge this source, and 2) send a copy of your publication with the reprinted material to: Newsletter Editor, College of Engineering, Computer Systems Support, 1249 SC, University of Iowa, Iowa City, IA 52242.

This issue:

Editor: Diana Harris. Contributors: Susan Beckett, Jim Cramer, Diana Harris



Jargon Rating: 0-2 ++. No image means there is no jargon and it should be easy to read and understand. Two plus signs mean lots of jargon. One plus falls between. All 2 plus articles include a no jargon summary at the top.

Images: © 1999-2003 www.barryscipart.com

Hours

Computer Labs — The Hering (1220 SC) and Elder (1231 SC) labs are open 24 hours/day, 7 days/week except for maintenance. With an access card, you can enter the building and the labs after the building closes.

Consulting, 1253 SC — M-Th 8am – 9pm; F 8am – 5pm; Sun 5pm – 9pm

CSS Main Office, 1256 SC — M-F 8am – noon, 1-5pm

Web Sites

This newsletter is published first to the CSS web pages at

<http://css.engineering.uiowa.edu/news/essentials>

The CSS pages are found at

<http://css.engineering.uiowa.edu>

The College of Engineering pages are at

<http://www.engineering.uiowa.edu>

The University of Iowa pages are at

<http://www.uiowa.edu>

CSS mission: to provide a secure and productive computer environment that supports the ability of the College of Engineering to achieve its educational mission.