

Manhattan 3.2 Administrator's Reference

Steven Narmontas

Manhattan 3.2 Administrator's Reference

Steven Narmontas

Published 08 December 2006

Copyright © 2003-2006

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the appendix entitled "GNU Free Documentation License" (*Appendix B, GNU Free Documentation License [135]*).

Table of Contents

1. Introduction	1
History	1
What is Manhattan?	1
Your role as administrator	3
Where to get Manhattan	3
2. Installation and Testing	4
Introduction	4
FreeBSD needs gmake	4
STEP 1: Select/create a user account	5
STEP 2: Unpack the Manhattan distribution.	5
STEP 3: Test for 'make' and 'gcc'.	6
STEP 4: Do you have sqlite3 installed?	7
Step 5: Type make clean	8
STEP 6: Install and test the ClearSilver library.	8
STEP 7: Create and edit the custom.h file	9
STEP 8: Compile the programs	9
STEP 9: Configure Apache	10
STEP 10: Set file permissions.	11
STEP 11: Perform some final tests.	12
Step 12: Configure your Super User account	13
3. Managing Admin accounts	14
Creating the super-user account	14
Logging in and the administrator's menu	15
You can't share your super-user account!	16
Creating additional administrator accounts	17
Deleting an administrator	19
Modifying an administrator's privileges	19
Forgot your password?	20
4. A tutorial for administrators	22
Create courses	22
Normal, Template and Standalone Courses	22
Creating normal courses	24
Logging in as Prof. Einstein	27
Create a Course Template for Prof. Einstein	31
Create a standalone course	33
Configure courses	35
Changing a group's description	37
Accessing a course's configuration menu	38
Configuring Course Templates	40
On your own	41
Changing a course's access rights	41
View/modify/delete users	44
The search users form	45
A quick list of all users	45
Viewing Prof. Einstein's profile	46
About a user's "Personal Info"	47
A few notes about users	48
On your own	49
Manage chat	49
Start the Melange chat service	50
Enter Chat	50

View/delete Melange service log	51
Stop the Melange chat service	52
Recommended periodic chat management	52
On your own	53
Login Logs	53
How the logs get rotated	54
All logs vs. individual logs	55
Admin logins	56
Viewing raw log data	57
Login Graphs	58
On your own	60
Surveys	60
The Administrator's Survey Lists	60
Creating a New Survey	61
Delivering a survey	65
Prof. Einstein takes the survey	70
Viewing survey results	71
How to learn more	72
Miscellaneous commands	72
Who is on Manhattan?	73
Lock Server	74
Delete temporary files	75
Purge Post Office Messages	77
Disk free	78
Course Usage Reports	79
On your own	80
Delete courses	80
Step 1 - Get a list of courses to choose from	81
Step 2 - Select courses from the list	82
Step 3 - Confirm your decision	82
Cleaning up	83
Deleting the empty course groups	83
Removing users without classes	84
5. Creating Courses - details	86
ID numbers, usernames and passwords	86
More about the roster text file	87
Choosing a group name	88
Choosing an "Internal Course Name"	90
Name Mismatch	90
6. Creating Courses with XML	92
The Alternative XML format	92
The XML format by example	92
Using an XML file to create courses	97
Skipping the XML Upload step	101
The XML Roster Manager	101
You must enable this feature!	102
Using the XML Roster Manager	103
The Request Classroom feature	105
7. Configure courses - details	109
Basic classroom configuration	109
View course roster	110
Get course roster	111
View student login logs, class access stats	111
Recent logins	112

Event Calendar	113
Event List	114
Disk Usage	114
Add a student	115
Add a teacher	117
Delete a student or a teacher	118
Reset a person's password	119
Change a person's name, team, or alias	119
Assign teams	120
Clipboard Viewer	121
8. Surveys - additional details	122
The big picture	122
Questions and answers	124
A. How to translate Manhattan	129
Introduction	129
Example: the 'doorstep' program	129
Things to note	131
Date and time formats	131
Some lang files affect more than one program	131
Watch out for JavaScript!	131
Other things to translate	133
Keeping up with new versions	133
B. GNU Free Documentation License	135
PREAMBLE	135
APPLICABILITY AND DEFINITIONS	135
VERBATIM COPYING	136
COPYING IN QUANTITY	136
MODIFICATIONS	137
COMBINING DOCUMENTS	138
COLLECTIONS OF DOCUMENTS	139
AGGREGATION WITH INDEPENDENT WORKS	139
TRANSLATION	139
TERMINATION	139
FUTURE REVISIONS OF THIS LICENSE	140
ADDENDUM: How to use this License for your documents	140

Chapter 1. Introduction

This brief chapter provides an overview of “The Manhattan Virtual Classroom”. You’ll learn a bit about its history, and get a broad view of how it’s used by teachers and students.

Most importantly, you’ll get an idea of what your role as system administrator will be. That is, you’ll learn why you need to read this manual.

History



Founded in 1919, Western New England College is a private, non-sectarian college located in Springfield, Massachusetts. The College enrolls approximately 4,600 students: 2,000 full-time undergraduates, 500 in full and part-time programs in the School of Law, and approximately 2,500 in part-time undergraduate and graduate programs offered on campus and at sites throughout the Commonwealth. Western New England College has 30,000 alumni around the world.

In February 1997, Steven Narmontas, who currently is the Director of the Educational Technology Center at the College, introduced a small group of faculty members to a software system he had worked on as a spare time project. Dubbed "The Manhattan Project", because it was largely developed in secret, the software enabled teachers to post files to a web site for their students to read. The earliest version of "Manhattan" also supported a few discussion groups and private messaging. Several pioneering professors made good use of the system immediately and Steve continued to develop the software. Today The Manhattan Virtual Classroom is very much a normal part of teaching and learning at the College supporting over 500 course sections per semester.

Manhattan was released on the Internet as free software under the GNU General Public License in October of 2000. Since then, the system has been downloaded thousands of times, and is in use around the world. You can obtain a free copy of The Manhattan Virtual Classroom and other supporting materials, including the latest version of this and other documentation from <http://manhattan.sourceforge.net>

Western New England College is committed to a developing a culture of collaboration, to developing innovative programs and alternative learning formats, to providing its students and faculty with a responsive technological environment, and to establishing itself as a leader regionally and nationally. Supporting the open source development and dissemination of Manhattan fits well within the goals we have set for us in our Strategic Plan.

I truly hope you find Manhattan useful, and that you’ll visit us on the web at: <http://www.wnec.edu>

What is Manhattan?

The Manhattan Virtual Classroom (or simply "Manhattan") is a web based course management system. Manhattan can be used to add an online component to a traditional face-to-face course, or it can be used to support distance learning courses that only meet online.

MAN 101 01 - Spring 2006
Introduction to Manhattan - Prof. Narmontas



The Manhattan software itself runs on a Linux (or possibly other Unix-like) server. Since it is a web based application, all of Manhattan gets installed on the server and no special software needs to be installed on the computers of teachers and students using the system. From their point of view, Manhattan is a web site. To use the system, teachers and students basically need computers connected to the web and accounts on a Manhattan server. When they enter their username and password correctly on Manhattan's login screen, they will be entering a private environment where teachers can:

- Provide their students with handouts, notices, lecture materials, interactive self-tests, and web sites to visit.
- Assign homework for students to complete, receive the work they do in response to those assignments, and provide feedback.
- Issue multiple-choice and short answer exams.
- Exchange private messages with their students.
- Host discussions with the entire class, or with teams of students.
- Keep students apprised of their grades.
- Issue anonymous surveys and collect the results.
- Engage in live online "chats" with their students.
- Track which students are using the system and when.

Manhattan was developed by Steven Narmontas and was first used at Western New England College in Springfield, Massachusetts in 1997. In October of 2000, the software was released in its entirety on the Internet for free under an unusual software license called the GNU General Public License. Today, Manhattan is in use around the world, and continues to be actively developed.

Your role as administrator

This is a reference manual for system administrators - people responsible for the daily care and feeding of a web server running The Manhattan Virtual Classroom. While you'll need some Linux/Unix experience to install Manhattan, once the system is up and running you'll find that you can do almost anything you need to do from a simple web-based interface.

Your job as Manhattan administrator includes:

- Creating Manhattan courses and populating them with an initial roster of students and teachers.
- At least occasionally adding or removing individual students or teachers from courses.
- Deleting courses when they are finished.
- Starting Manhattan's chat service, and keeping an eye on the logs.
- Assisting users who have forgotten their password.
- In general, keeping an eye on things like disk space and login logs.

Fortunately, Manhattan is quite stable. In general once you get past the flurry of activity of creating courses at the beginning of a school year, running an even very busy Manhattan server will take only a few minutes of your time a day.

A *Teacher's Reference* is available at <http://manhattan.sourceforge.net>. It's not absolutely necessary for a system administrator to know how to use Manhattan from a student's or teacher's perspective. However having at least a broad understanding of what Manhattan looks like to an end-user is useful. This manual makes no attempt to cover the basics of Manhattan from that perspective, so you might want to at least browse through the other documentation.

Where to get Manhattan

The Manhattan Virtual Classroom is free open source software covered by the GNU General Public License. Manhattan's website is <http://manhattan.sourceforge.net>. There you'll find the latest release of Manhattan along with this and other documentation.

Chapter 2. Installation and Testing

This chapter provides detailed instructions for installing Manhattan 3.2 on a Linux server. I also provide some hints for FreeBSD servers. (If you don't yet have the software, visit <http://manhattan.sourceforge.net> to download a copy.)

Installing Manhattan isn't difficult. How long it takes depends largely on how much experience you have with Linux. Complete “newbies” may struggle with basic tasks like copying or editing a file, that more experienced users have done thousands of times. After you've installed Manhattan once, you can do it again if you need to within 20 minutes.

Introduction

The Manhattan Virtual Classroom is a web application written in the C programming language.

Manhattan was developed under Linux, and most people run it under that operating system. However, Manhattan should also compile cleanly under FreeBSD.



Currently *MANHATTAN WILL NOT COMPILE ON A 64 BIT OPERATING SYSTEM*. Also, there are issues moving classroom data between 64-bit and 32-bit systems. Some have been successful compiling the system on a 32 bit system, and then moving the binaries over to their 64-bit system, but that's not recommended. If you purchased a 64-bit system for the sole purpose of running a Manhattan server, you can install a 32 bit version of Linux on that hardware. Support for 64 bit systems is on our to-do list, but it is not a high priority. Software like Manhattan will not run faster on a 64 bit system.

Many people who would like to install Manhattan are relatively new to Linux. In addition there's many Linux distributions, each of which does things a little differently. Because of these issues, these INSTALL instructions have been recently changed to provide more hand holding and testing than previous versions.

FreeBSD needs gmake

Earlier releases of Manhattan were were successfully compiled and installed on a PC running FreeBSD 6. Unfortunately I no longer have FreeBSD installed anywhere, so I have not yet experimented with installing this version of Manhattan under that OS.

One thing is certain: If you are running FreeBSD, you'll have to set things up so you use 'gmake' instead of FreeBSD's standard 'make'. This worked for me:

1. Install gmake from the ports collection.
2. Move FreeBSD's standard make out of the way:

```
cd /usr/bin  
mv make make.sav
```

3. Create a symbolic link make --> gmake

```
ln -s /usr/local/bin/gmake make
```

Now the following INSTALL instructions should work fine for you, since every reference to **make** will actually be running **gmake**

REMEMBER TO UNDO THIS after installing Manhattan like this:

```
cd /usr/bin
rm -i make
mv make.sav make
```

There may be other (better) ways to substitute gmake for make on a FreeBSD system. The bottom line is that when you type:

```
make --version
```

You should get a response that looks something like this:

```
GNU Make 3.80
Copyright (C) 2002 Free Software Foundation, Inc.
```

The actual version number is unimportant, as long as you have any version of gmake installed, you'll do fine. If the above command results in some version information for GNU make, you're ready to install Manhattan.

STEP 1: Select/create a user account

Manhattan will be installed in the home directory of an ordinary user. You can either use an existing user account (such as your own), or you can create an entirely new user account for this purpose.

At times you must become **root** during the installation process. The user account you select should have that capability, either by using the **su** command together with the root password, or by having the privilege of running the **sudo** command. In either case, know how to execute commands as **root** before moving on!

If you do create a new user account for your Manhattan installation, I recommend you use the username **manhat**. *THESE INSTRUCTIONS WILL ASSUME THE USERNAME YOU'LL USE IS:*

```
manhat
```

If you use another username, adjust the following instructions accordingly.

NOW LOGIN AS THE USER **manhat** and from **manhat**'s home directory, enter the command:

```
chmod 711 .
```

Note the "dot" at the end of the command. This command gives the Apache web server limited access to **manhat**'s home directory.

STEP 2: Unpack the Manhattan distribution.

Copy the Manhattan distribution file into user **manhat**'s home directory, and unpack it with:

```
tar -xvzf manhat-x.x.x.tar.gz
```

Replace **manhat-x.x.x.tar.gz** with the actual name of the distribution you downloaded. (The **x.x.x** part of the file name represents the Manhattan version number.)

By user `manhat`, I mean whatever user account you have decided to use to install Manhattan. (This is the last time that will be mentioned!)

You'll notice that as the file unpacks, it creates a new directory. That's the installation directory you'll use in place of `manhat-x.x.x` in the remainder of these instructions.

STEP 3: Test for 'make' and 'gcc'.

Change to the `src` subdirectory of the `manhat-x.x.x` directory that was just created:

```
cd manhat-x.x.x/src
```

Now type the command:

```
src> make hello
```



Don't type the `src>` part! I'll use that to represent your shell prompt. Hopefully the `src>` prompt will remind you to be in the `src` directory of your Manhattan installation.

You should see something like:

```
rm -f sqlite3_test.o hello.o clearsilver_test.o
gcc -O2 -Wall -Iinclude -c hello.c
gcc -O2 -Wall -Iinclude hello.o -o hello
src>
```

Now run the `hello` program by typing:

```
src> ./hello
```

You have 'make' and 'gcc' working! Test passed.

```
src>
```

As shown above, the program simply prints a line of text on your screen. If you completed the above steps without errors, move on to Step 4 below.

Still here?

If you get any "command not found" or similar errors, you probably don't have GNU `make` or the `gcc` compiler installed on your system.

You can test for these individually by typing:

```
src> make -v
```

and

```
src> gcc -v
```

`make` and `gcc` are tools used to create programs under Unix-like operating systems. Sometimes these tools are not automatically installed when you install Linux. Use your version of Linux's package management system to find and install `gcc` and `make`. Often `make` and `gcc` will be found in a software category with a name that contains the word **Development**.

For some distributions, including Ubuntu, you might get the strange message: "C compiler can't create executables", or similar. If this is the case, you need to install the GNU C Development Libraries and Header files. This package will usually have a name like `libc6-dev`, and might be found in the "Development" or "Libraries" category.

There's no point continuing until your system can pass this simple test. If you get stuck, consult a forum for your particular version of Linux. Tell them you can't get a simple "hello world" program written in C to compile and run.

STEP 4: Do you have `sqlite3` installed?

Starting with version 3.2, Manhattan requires a "shared library" called `sqlite3`. *MAKE SURE YOU ARE IN THE `src` DIRECTORY OF YOUR INSTALLATION*, then type:

```
src> make sqlite3_test
```

In a perfect world, you'll get something like:

```
rm -f sqlite3_test.o hello.o clearsilver_test.o
gcc -O2 -Wall -Iinclude -c sqlite3_test.c
gcc -O2 -Wall -Iinclude  sqlite3_test.o  -lsqlite3 -o sqlite3_test
src>
```

If this is the case (that is, you received no error messages), run the program by typing:

```
src> ./sqlite3_test
```

You should get something like:

```
Creating a database...
Creating a table...
Inserting some data...
Performing a query...
12345 Hello world! 1 0
sqlite3 tests passed!!
src>
```

If the above steps completed without errors, move on to *Step 5 [8]*

Still here?

You need to install the `sqlite3` and/or the `sqlite3 development` packages for your version of Linux. Use your Linux version's package management system (whether that's RPM, YaSt, Synaptic Package Manager, aptget, or whatever) and look for the "sqlite3" AND the "sqlite3 development" (it may be called "sqlite3-dev") packages, and install them on your system. Then try the above test again.

For many Linux distributions, you'll find that `sqlite3` is already installed, but the `sqlite3 development` package is not. (This seems to be the case with Ubuntu Linux.)



You'll probably find packages for "sqlite" as well as "sqlite3". There's no harm installing both, but Manhattan (and this test) will work only if you have "sqlite3" installed.

The website for SQLite is: <http://www.sqlite.org>

Step 5: Type `make clean`

Be sure you are in the `src` directory of your Manhattan installation, then type the command:

```
src> make clean
```

You should see many lines scroll by, as the Manhattan distribution is made ready for installation. There's not much that can go wrong here, since you already checked that you have the 'make' program installed in an earlier step.

STEP 6: Install and test the ClearSilver library.

Another shared library that must be installed is called `libneo.so`. This provides the ClearSilver html template library that plays a central role in Manhattan 3.x. The website for ClearSilver is: <http://www.clearsilver.net>.

ClearSilver's `libneo.so` library is newer and much less common than the `sqlite3` library. Unless you've already had a previous 3.x version of Manhattan installed on this server, it's unlikely you have this installed. To test, *MAKE SURE YOU ARE IN THE `src` DIRECTORY OF YOUR INSTALLATION*, then type the following:

```
src> make clearsilver_test
```

If you have the ClearSilver library properly installed, you'll see something like:

```
rm -f sqlite3_test.o hello.o clearsilver_test.o
gcc -O2 -Wall -Iinclude -IClearSilver -c clearsilver_test.c
gcc -O2 -Wall -Iinclude clearsilver_test.o -lneo -o clearsilver_test
src>
```

If there are no errors, you can try running the `clearsilver_test` program by typing:

```
src> ./clearsilver_test
```

If everything works perfectly, you'll see something like:

```
ClearSilver.Test = Passed
src>
```

If you can reproduce the above actions without error, move on to the next step: *Create and edit `custom.h` [9]*

Still here?

I'm not surprised, since the ClearSilver library (named `libneo.so`) is much less common than the `sqlite3` library. It's unlikely you'll have it installed unless you've already installed an earlier edition of Manhattan 3.x on this system.

To install the ClearSilver library, follow these steps carefully.

First, make sure you are in the 'src' directory of your Manhattan installation, then type the command:

```
src> make clearsilver
```

You'll see lots of lines scroll by on your screen as the library is created. At the end, you'll be returned to your shell prompt. Type the following command to list the files in a newly created `lib` directory:

```
src> ls lib
libneo.so
src>
```

The `libneo.so` file needs to be installed on your system. *TO DO THAT YOU MUST BECOME root.* (See the comments in *Step 1 [5].*) As `root` do:

```
src> cp lib/libneo.so /usr/lib
src> ldconfig
```



If you get a "command not found" while typing the second command, try the command (while `root`) as: `/sbin/ldconfig`



IF YOU ARE RUNNING FreeBSD, the second command should be:

```
ldconfig -m /usr/lib
(or maybe /sbin/ldconfig -m /usr/lib )
```

Once you've installed the `libneo` library using `ldconfig`, repeat the test listed at the top of this step to make sure `ClearSilver` is working.

STEP 7: Create and edit the `custom.h` file

Be sure you are in the `src` directory of your Manhattan distribution, then type the command:

```
src> cp default_custom.h custom.h
```

This copies the default configuration file into a file named `custom.h`, which is required to install Manhattan.

Use your favorite text editor to view and probably modify the `custom.h` file. Extensive comments within that file explain each option. If you don't understand what an option does, you can generally leave it alone.

Don't skip the step of reviewing the `custom.h` file, *BUT DON'T SPEND HOURS ON IT!* Most of the options are fine just the way they are, but *YOU WILL WANT TO CHANGE THE FIRST ENTRY*, which is:

```
#define EXIT_URL "http://manhattan.sf.net"
```

If you don't, you'll find that users will be directed to Manhattan's home page when they logout of the classroom. Change this entry to point to one of your own websites.

STEP 8: Compile the programs

Be sure you are in the `src` directory of your Manhattan installation, then type the command:

```
src> make install
```

The compile process involves converting a number of C source files into binary executable files. You should see dozens of lines scroll by on your screen as the process completes. This can take a few minutes, especially on slower computers. WARNINGS can generally be ignored. If any errors occur the process will stop before you see the word "Done!".

STEP 9: Configure Apache

You have to become `root` to do this step!

I assume you already have the Apache web server installed and running on your server. If not, get that done.

Now add the following lines to the appropriate place within your Apache configuration files. *BE SURE TO REPLACE ALL SIX OCCURRENCES OF `manhat-x.x.x` WITH THE NAME OF YOUR MANHATTAN DISTRIBUTION*. The text `/home/manhat/manhat-x.x.x` below should be the full path to the directory where Manhattan is installed:

```
ScriptAlias /manhat-bin/      /home/manhat/manhat-x.x.x/bin/

<Directory "/home/manhat/manhat-x.x.x/bin/">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
</Directory>

ScriptAlias /manhat-sbin/     /home/manhat/manhat-x.x.x/sbin/

<Directory "/home/manhat/manhat-x.x.x/sbin/">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
</Directory>

Alias /manhat-images/        /home/manhat/manhat-x.x.x/images/

<Directory /home/manhat/manhat-x.x.x/images>
    Options FollowSymLinks
    AllowOverride None
    Allow from all
</Directory>
```

The "appropriate place" is wherever your installation of Apache likes to see `ScriptAlias` directives. This might be in a file named `httpd.conf` or could be in another file Included into your `httpd.conf` file.

If you don't know how to do this, maybe this explanation (ok, rant) will help. Traditionally, Apache's configuration is controlled by a text file that's usually named `httpd.conf`. Where this file is located seems to be at the whim of the designer of your Linux distribution, slightly influenced by whether you are running Apache 1.3 or some 2.x version. You need to find the `httpd.conf` file. It's been known to live in any one of these places:

```
/usr/local/apache/conf/httpd.conf or  
/etc/httpd/conf/httpd.conf or  
/etc/httpd/httpd.conf or  
/etc/apache2 or  
/usr/local/etc/apache/
```

The commands **locate httpd.conf** or **whereis httpd.conf** or **find | grep httpd.conf** can help you find the file.

Found it? Good. You'd think that would be enough, eh? There was a time when all of the configuration directives were placed in this one text file. That might still be the case for your installation. However, now it seems to be trendy to add other configuration files, sometimes many other configuration files, by using Include statements within the httpd.conf file. So, the `<ScriptAlias>` directives we're looking for could be almost anywhere. Start looking in the `httpd.conf` file. If it's there, great, otherwise look in files that are Included from the `httpd.conf` file, or maybe even in files that are stored somewhere near the `httpd.conf` file.

After changing your Apache configuration file(s), restart the Apache web service. You can often restart Apache by typing (as root) **apache2ctl restart** for Apache 2 or **apachectl restart** if you're running Apache 1.3.

As a specific example, on my Ubuntu 6 installation, I was able to create a new file called simply `manhattan` containing nothing but the above directives. I placed the file in the directory: `/etc/apache2/conf/conf.d` To restart the Apache service I used: **sudo apache2ctl restart**

Ubuntu is based on Debian, which may or may not use a similar configuration scheme.

STEP 10: Set file permissions.

From any computer, use a web browser to visit:

```
http://YOUR_SERVER/manhat-bin/apache_group
```

Replacing `YOUR_SERVER` with the hostname of your server.

If everything is set up correctly so far, you'll get a page that looks something like this:

```
Apache is running as group: www
```

```
To properly set the permissions for your Manhattan installation:
```

1. cd to the `src` directory of your Manhattan installation
2. Become root (or use **sudo** to execute the next command as root)
3. Type the command:

```
./set_dirs www
```

Assuming the page displays, follow the instructions shown in your web browser. This involves running a **set_dirs** script with an argument, which will set up all of Manhattan's data directories with the proper file permissions. The above example shows that "Apache is running as group: `www`", and therefore the command you need to run from the `src` directory as `root` is

```
src> ./set_dirs www
```

The Apache group is often `www`, but your installation might be different. Ubuntu, for example, seems to use the group `www-data`. In all cases, follow the instructions on your web browser's screen.

You'll see something like:

```
Setting up ../sbin and ../bin
Setting up ../tmp
Setting up ../templates
Setting up ../clipboards
Setting up ../surveys
Setting up ../images
Setting up ../chat
Setting up ../auto_update
Setting up ../courses
Setting up ../users
```



MAKE SURE YOU RUN THE COMMAND AS ROOT, or the permissions will not be set properly.

The whole point of visiting the `apache_group` URL is to learn the Group setting of Apache, so you can run `./set_dirs` with the correct argument. If you can't visit the web page, then there's probably something wrong with the Apache configuration you did in the previous step: *Configure Apache [10]*

You should only need to run the `set_dirs` script once, when you are first installing Manhattan. The file permissions will stay correct forever, unless you mess around with the file system. That said, it's safe to run `set_dirs` whenever you think the file permissions may be wrong, provided you include the correct argument, which is the Group setting of your Apache installation.

STEP 11: Perform some final tests.

From any computer, use a web browser to visit:

```
http://YOUR_SERVER/manhat-bin/config_test
```

Replacing `YOUR_SERVER` with the hostname of your server. This page does a few more tests, such as making sure the spell checker for Manhattan is installed and working. The page includes some information on how to set things straight if they are not already working.

Manhattan Configuration Tester

Is the 'images' alias set correctly?

You should see an image below. If not, the 'manhat-images' Apache alias may be incorrect. Check the Apache configuration settings step in the INSTALL instructions.



The URL used for the above image is: /mh3-images/theme_0/manhattan.gif

Is Apache's ServerName set properly?

The ServerName setting for this Apache server is: **empire.wnec.edu**

The server string Manhattan will construct is: **http://empire.wnec.edu**

An Apache ScriptAlias pointing to Manhattan's 'bin' directory should be called: **mh3-bin**

An Apache ScriptAlias pointing to Manhattan's 'sbin' directory should be called: **mh3-sbin**

A link to a teacher/student login prompt: <http://empire.wnec.edu/mh3-bin/doorstep>

A link to an Administrator's login prompt: http://empire.wnec.edu/mh3-sbin/super_doorstep

Be sure both of the above links work!

If they don't it's most likely because Apache's ServerName setting (listed above) is incorrect. One of your Apache configuration files contains a ServerName entry. The ServerName setting is used by Manhattan to redirect users back to certain Manhattan programs. Often this setting is filled in with some useless default value.

The correct value for your ServerName setting is the complete name someone would type into a web browser to visit a page on your server. An example is:

Step 12: Configure your Super User account

If all of the above worked out for you, Manhattan should be ready to go. Visit:

http://YOUR_SERVER/manhat-sbin/super_conf

To create a "Super User" account. (see *Creating the super-user account* [14] in the next chapter)

Chapter 3. Managing Admin accounts

Manhattan uses a web-based administration system for creating and deleting classrooms, starting and stopping the chat service, and for managing user accounts. System administrators login to the administrative system using a username/password.

Creating the super-user account

After installing Manhattan, the next step is to create an account that will be used to enter Manhattan's web-based administration system.

Point your web browser to:

```
http://YOURSERVER.edu/manhat-sbin/super_conf
```

Replace `YOURSERVER.edu` with the name of your server. Use the form you see to create the super user account:

Create the super user account

Username for the super-user:

Super-user password:

Super-user password (again):

You'll need to enter a username and password for this super user. A good username might be your first initial followed by your last name, all in lowercase letters with no spaces. A good password is, well, a good password.

Think a bit when you select the username. Whatever you select you'll be using from now on. The password can be changed as often as you wish.



After submitting the form, you will have created the super-user account, and the above URL will no longer work. For the best protection, you should delete the `super_conf` CGI program, so it can never be run again from a web browser. Login to the server using the `manhat` shell account, then:

```
cd manhat-x.x.x/sbin
rm super_conf
```

You'll be shown a page summarizing the super-user account information. At the bottom of the page is a link that you can click on to login to the administrative system for the first time.

Wed 12/07/2005 01:56 PM

Username:	holiday
Granted privileges	
	Create courses
	Delete courses
	Configure courses
	Change course access
	Archive courses
	Surveys
	View/modify/delete users
	Who's on Manhattan?
	Login logs
	Disk space
	Manage chat
After running the super_conf program, you'll be shown the URL of the administrator's login.	
http://manhattan.wnec.edu/gnt-sbin/super_doorstep	



If, when you click on the link shown above to login to the Administrative system, your web browser cannot locate the web site, it's likely that the `ServerName` is set improperly in Apache's `httpd.conf`. You should revisit the configuration test page as described in *Perform some final tests* [12]

Logging in and the administrator's menu

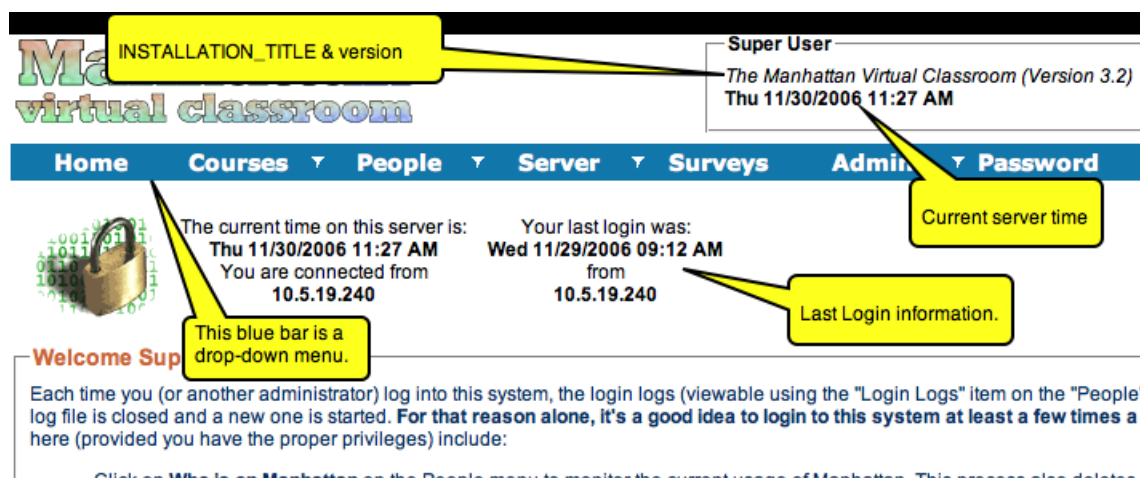
Point your web browser to:

`http://YOURSERVER.edu/manhat-sbin/super_doorstep`

Again, replace `YOURSERVER.edu` with the name of your server. Enter your super-user username and password, and you should soon be to the administrative menu shown below.

The above URL is your entry point into Manhattan's administrative system. You probably should NOT post it to a web page, since there's no need to advertise the URL to the world.

The administrative system allows you to pretty much manage the entire Manhattan server from a web browser. The blue bar with the words "Home, Courses, People, Server ..." is your drop-down menu system.



Most of this manual is devoted to explaining the use of the commands available from this menu. There's a few things to note about this page before we move on.

When you login, the top of the page shows both the current time on the server, and the time of your last login. Pay attention to what you see here. The server time IS important, since the messages students and teachers post to their classrooms are marked with the time sent, time read, etc. If your server's clock is off by more than a few minutes, you should correct the problem. Secondly, the time of your last login, along with the information regarding your current and last host addresses, can help you determine if someone else has stolen your super-user username and password.

The **Password** menu item is used to change the super-user's password, and the **Logout** menu item (not shown above) is used to log out of the administrative system.

Notice the line labeled "INSTALLATION_TITLE and version" in the above screenshot. The words "The Manhattan Virtual Classroom" are picked up from this entry:

```
#define INSTALLATION_TITLE "The Manhattan Virtual Classroom"
```

in the `custom.h` file found in the `src` directory of your distribution. Changing this `custom.h` value can become useful when you manage more than one installation of Manhattan, since it can be used to remind you of which version you are administering at the moment. (see *Create and edit custom.h* [9] for info about the `custom.h` file. After modifying the file, you must type: **make clean ; make install** to recompile the programs that make up Manhattan.)

Immediately next to the installation title is the version of Manhattan you are running, which happens to be "Version 3.2" in the above screenshot.

You can't share your super-user account!

You cannot share your super-user account with another person. The moment someone else logs into Manhattan using your username and password, you will get logged out. If you are working within Manhattan's administrative system and someone else logs in as you, the next time you click on a Manhattan command button or other link you'll get a message like "Sorry, but you have been logged out of the classroom. It's possible that someone else is using your account!"



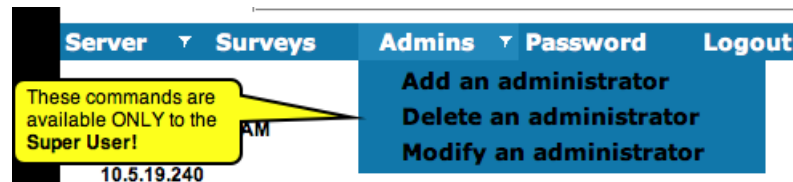
Sorry, but you have been logged out of the classroom.
(It's possible that someone else is using your account!)

[Back](#)

To further safeguard the administrator accounts, *you will be automatically logged out if you have not clicked on a Manhattan command button or link for four hours.*

Creating additional administrator accounts

The super-user can create additional administrator accounts, using the **Add an administrator** selection on the **Admins** menu.



After clicking on **Add an administrator**, you'll get the form:

Add an administrator

Username:

Password:

Password(again):

First name:

Last name:

Select the privileges

Courses

- ☐ Create courses
- ☐ Delete courses
- ☐ Configure courses
- ☐ Change course access
- ☐ Archive courses
- ☐ Surveys

People

- ☐ View/modify/delete users
- ☐ Who is on Manhattan?
- ☐ Login logs

Server

- ☐ Disk free
- ☐ Course usage reports
- ☐ Manage chat
- ☐ Lock server
- ☐ Delete temporary files
- ☐ Purge post office msgs

Add this administrator

Filling in the form is straightforward. You are deciding on the username and initial password for an additional administrator account. By checking the boxes, you are also determining which of the commands on the administrative menu will be available to this new administrator. (You can change the privileges later.)

For example, you might want to give someone at your school's help desk access to the **View/modify/delete users** command, so they can help students by resetting passwords and by looking up usernames. You might not want that person to be able to “Create courses” or “Delete courses”.



It is not necessary to create additional administrators. Many schools will do well with only one Manhattan administrator - the super-user.

After creating a new administrator, that person can login using the same URL used by the super-user:

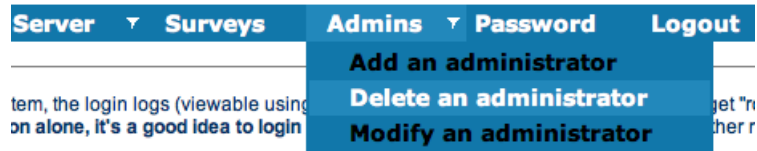
`https://YOURSERVER.edu/manhat-sbin/super_doorstep`

They'll see the same menu that the super-user sees, except only the items the super-user has allowed will appear on the menu.

The commands the super-user sees under the Admins menu: **Add an administrator**, **Delete an administrator**, and **Modify an administrator** can *never* appear on another administrator's menu. These commands are reserved for the super-user, and are what makes the super-user different from other administrators.

Deleting an administrator

The super-user can remove any administrator he has created by selecting **Delete an administrator** from the **Admins** menu.



Deleting an administrator account is as straightforward as can be. Simply select the user from the list, and click the **Delete this administrator** button.

Modifying an administrator's privileges

The **Modify an administrator** command allows the super-user to change the commands a particular administrator can access. After clicking the link, the super-user will see a list of administrators. Select one from the list, click the submit button and you'll see:

Grant/revoke administrator's privileges -

Courses

- ☐ Create courses
- ☐ Delete courses
- ☒ Configure courses
- ☐ Change course access
- ☐ Archive courses
- ☒ Surveys

People

- ☒ View/modify/delete users
- ☒ Who's on Manhattan?
- ☐ Login logs

Server

- ☐ Disk space
- ☒ Manage chat
- ☐ Lock server
- ☐ Clean up

As for when you created the administrative account for the first time, each “privilege” corresponds to an item on the administrator's menu. Use the checkboxes to add or remove privileges and submit the form.

Forgot your password?

If an administrator other than the super-user forgets his password, the only remedy is for the super-user to:

1. Delete the administrator using the **Delete an administrator** command.
2. Re-create the administrator account using the **Add an administrator** command.

If the super-user forgets his password, things are a little more complicated. First you must login to the Manhattan server using the `manhat` shell account. Change to the `manhat-x.x.x/users/admin` directory:

```
cd manhat-x.x.x/users/admin
```

Next, rename the file named `index.txt`:

```
mv index.txt index.txt.orig
```

It doesn't matter what you call the renamed `index.txt`, the important thing is that there no longer is a file named `manhat-x.x.x/users/admin/index.txt`

Now you can repeat the process for creating a super-user account for this installation, as described in *Creating the super-user account [14]*. If you followed my advice in that section and have deleted the `super_conf` program, you must recompile it from source. While logged in to manhat's shell account on the server:

```
cd ~/manhat-x.x.x/src
make super_conf
mv super_conf ../sbin
```



- Be 100% certain you are renaming the correct `index.txt` file. The one you want to rename is in the `users/admin` directory of your installation. There's another `index.txt` file in the `users` directory, *which you should not touch!*
- This is an exceptional procedure to handle an exceptional accident - the super-user forgot his password. The procedure works, but you should never have to use it!
- You'll find that this procedure deletes all administrative accounts. The super-user will have to re-create them.
- Again, you should consider deleting the `super_conf` program from the `sbin` directory of your installation as soon as you are done re-creating the super-user account.

Chapter 4. A tutorial for administrators

The intent of this chapter is to provide the first time system administrator with a tour of the capabilities of Manhattan's administrative system. Follow this tutorial and you'll:

- Create two *Normal* courses.
- Create a *Course Template* that the teacher can use to store course materials for re-use.
- Create a *Standalone Course* for the Physics Club.
- Login as a teacher to see his view of Manhattan.
- Get an overview of how you, as system administrator, can:
 - Configure courses
 - Work with user accounts
 - Delete courses and users

When you are done with the tutorial your new Manhattan distribution will be exactly where it was when you first installed Manhattan, and you'll be ready to start creating real classrooms for your school.



Throughout this chapter we'll refer to URL's like this:

`http(s)://YOURSERVER.edu/...`

In each case, replace `http(s)` with `https` if your server supports SSL, or with `http` if you are not yet using SSL. Of course, you'll need to replace `YOURSERVER.edu` with the name of your server.

Create courses

This part of the tutorial demonstrates how you can create Manhattan courses using a simple roster text file, by selecting Courses → Create courses → ...from roster file



Normal, Template and Standalone Courses

Before we start the tour, you should know there are three types of Manhattan classrooms:

Normal

These are classrooms that are tied to the centralized login system. Students and teachers login using a single username/password form and then are presented with a “My Manhattan” page that lists all of their courses. They can then enter any course on the list.

Standalone

These classrooms are *NOT* tied to the centralized login system

- A standalone classroom must be accessed via its own URL. Typically the administrator (that's you) will maintain a list of course links at a website somewhere. Logging into a standalone course gains entry to that course only.
- Changing a password in a standalone course changes it for that course only.
- The teacher (or system administrator) of a standalone course can freely add students or teachers to the course using fictitious names or Student IDs, since these names/IDs are completely independent of the centralized system.

Templates

A *course template* is intended to be a place for a teacher to “permanently” store course materials for re-use. Manhattan allows the teacher, using Manhattan's internal “Copy and Paste” feature, to copy a message and all of its attachments from one classroom to another. This feature can be used to copy course materials to and from a course template.

- A course template is like a “normal” course in that it's tied to the centralized login system. When a teacher logs into Manhattan, their “My Manhattan” page will show all of the courses they're teaching, plus their course templates.
- Course templates should only have one member - the teacher who owns the template. However, there's nothing to prevent you (or the teacher) from adding additional students or teachers.
- Certain of Manhattan's *modules* are disabled in course templates. These include the Post Office, Chat, Team and Anonymous discussion.
- Course templates use a different background image on the classroom's *Main Menu* to remind teachers where they are.
- Course templates happen to be stored in a different directory on the server, in the `templates` directory instead of in the `courses` directory, which is where “normal” and “standalone” courses are stored.



Most of the time, you will be creating “normal” courses. “Standalone” courses are useful to support non-academic activities like clubs or special courses with students who are not part of your normal population.

You'll understand the value of “course templates” when you get to the point where you've got 100 teachers running 400 courses on your server, and no one wants you delete any courses from previous semesters because they contain too much valuable material! It's impractical to keep hundreds (or thousands) of old courses on a server, just so the people who taught those courses can re-use materials they have posted there. Instead you can create a number of course templates for each teacher, and ask them to move course content from their old courses into the course template. Then you can delete (perhaps after backing up) the old

courses. In this way, you can save many megabytes of disk space since many of the posts made to a completed course are not from the teacher, but from the students in the class.

Creating normal courses

In this section, you'll start by creating a Manhattan course "Introduction to Physics" taught by Albert Einstein. You'll then create an English Literature course for another professor.

Create the Introduction to Physics course

Open your favorite text editor (Notepad is fine if you're working with Windows) and either type or copy/paste the following lines into a new file.

```
PHY 101 01
Introduction to Physics
Spring 2003
Prof. Einstein
X34322  Albert Einstein
X343888 Neils Bohr
X347332 Isaac Newton
X394032 Michael Jordan
X334321 Steven Stanton
X394452 Cindy Slattery
X485734 Robert Frost
```

Briefly, the first four lines provide a general description of the course - its *course-code-section*, *title*, *semester*, and the *title and name of the teacher*. The fifth line contains the teacher's ID number and the teacher's first and last names. The remaining lines describe students in the course. We arbitrarily chose the ID numbers, e.g. X34322 for Albert Einstein, in this tutorial but normally you'd get the ID numbers from your school's student information system. The format of this file is described fully in *More about the roster text file* [87].

Save the file. You can call the file whatever you wish, but remember what you called it and where you put it on your hard disk. Then complete the following steps:

1. Visit

```
http(s)://YOURSERVER.edu/manhat-sbin/super_doorstep
```

and enter your super-user username/password to login to Manhattan's administration system.

2. From the menu, select **Courses**, then **Create courses**, then **..from roster file**.
3. Next to the label "Group:" is a selection list. If this is the first time you're creating a course, there will be only one choice: "<New Group>". Select <New Group>, and type the word **try** in the text box next to the "<New Group>" selection list. (See the screenshot below.)
4. In the "Internal course name" text box, type: **phy10101**. The "Internal course name" will be the name of the directory on the server where this course will live. You should always select a name that reflects the course-code-section of the course you are creating.
5. Click the **Browse** button and use your web browser's Choose File dialog to find and open the text file you just saved on your PC.

6. Check out the “Course type:” drop-down list. You use the same form to create “normal”, “standalone”, and “template” courses. (See *Normal, Template and Standalone Courses* [22] for a discussion of these course types.) Select “Normal”.
7. Click the **Create this course** button.

Here's a screenshot of the form after following the above instructions. Of course, the “Roster file:” location will differ, depending on where you saved the text file on your PC:

Use roster file

Group:

Internal course name:

Roster file:

Course type:

Group
All courses on the server belong to a 'group'. Internally, a group is a director within that group are stored. You can either select an existing group from the text box.

After submitting the form, you'll should get a page stating that the course has been created. All of the people should be listed as “New Users”.

1 Normal and Template Courses

Group:	try
Internal Course Name:	phy10101
Course Code:	PHY 101 01
Course Title:	Introduction to Physics
Semester:	Spring 2003
Course Type:	Normal
Status:	New Course
Success?	Yes

Full Name	Id	Username	Central Account	User Type	Course Status	Central Status
Albert Einstein	X34322	ae4322	New User	Faculty	Add Successful	Success
Neils Bohr	X343888	nb3888	New User	Student	Add Successful	Success
Isaac Newton	X347332	in7332	New User	Student	Add Successful	Success
Michael Jordan	X394032	mj4032	New User	Student	Add Successful	Success
Steven Stanton	X334321	ss4321	New User	Student	Add Successful	Success
Cindy Slattery	X394452	cs4452	New User	Student	Add Successful	Success
Robert Frost	X485734	rf5734	New User	Student	Add Successful	Success
Total Users: 7						

You've just created your first Manhattan classroom!

Create an English Composition I course

Now create a second course for Prof. Jenkins' "English Composition I" course. Use a text editor to create a second file with the following contents:

```
ENG 101 01
English Composition I
Spring 2003
Prof. Jenkins
X34321 Julius Jenkins
X34388 Neils Bohr
X347389 Robert Fuller
X394096 Thomas Edison
X334377 Richard Taylor
X394478 Marie Curie
X485753 Janet Jackson
```

Save the file on your PC, and again select Courses → Create courses → ...from roster file from the menu. This time:

- The drop-down selection list contains the "Group" called `try` you created when you set up the Physics 101 course. At the moment it says the group `try` has "(No Description)" - you'll learn how to add a description to a course group later. *Select the `try` group for this new course.*
- Type **engl10101** as the "Internal Course Name" As mentioned earlier, the "Internal Course Name" is actually the name of a directory on the server where this course's data will be stored. The name you choose should reflect the course-code-section of the course.
- Click the **Browse** button and open the roster file you just saved on your PC. (Be sure you don't select the file for the Physics course by accident!)

Your filled-in form will look something like the screenshot below. The location of the "Roster file:" will probably be different, since it depends on where you saved the text file on your computer.

Use roster file

Group:	try(No description)	
Internal course name:	engl10101	
Roster file:	C:\roster.txt	Browse...
Course type:	Normal	
Create this course		

Group
All courses on the server belong to a 'group'. Internally, a group is a directory on the server. All courses within that group are stored. You can either select an existing group from the list, or enter a new group name in the text box.

Click the **Create this course** button to create the course. The output of this command should be:

1 Normal and Template Courses

Group:	try
Internal Course Name:	engl10101
Course Code:	ENG 101 01
Course Title:	English Composition I
Semester:	Spring 2003
Course Type:	Normal
Status:	New Course
Success?	Yes

Full Name	Id	Username	Central Account	User Type	Course Status	Central Status
Julius Jenkins	X34321	jj4321	New User	Faculty	Add Successful	Success
Neils Bohr	X343888	nb3888	Existing User	Student	Add Successful	Success
Robert Fuller	X347389	rf7389	New User	Student	Add Successful	Success
Thomas Edison	X394096	te4096	New User	Student	Add Successful	Success
Richard Taylor	X334377	rt4377	New User	Student	Add Successful	Success
Marie Curie	X394478	mc4478	New User	Student	Add Successful	Success
Janet Jackson	X485753	jj5753	New User	Student	Add Successful	Success
Total Users: 7						

Note that the table lists Neils Bohr as an “Existing User”, which simply means that he already has at least one other Manhattan course. In fact, he is also enrolled in Prof. Einstein's Physics 101 course. You should now have two courses on your Manhattan server.

Logging in as Prof. Einstein

Now visit:

`http(s)://YOURSERVER.edu/manhat-bin/doorstep`

This is the login page used by students and teachers. Note the differences between the URL used for the administrator's login and that used by students and teachers:

Administrators use: `http(s)://YOURSERVER.edu/manhat-sbin/super_doorstep`

Students and teachers use: `http(s)://YOURSERVER.edu/manhat-bin/doorstep`

Prof. Einstein's username and password are:

Username: ae4322

Password: ae4322



Manhattan derives a person's username by adding the last four digits of their ID number to their initials (see *ID numbers, usernames and passwords* [86] for details). A person's password is initially the same as their username. They are forced to change their password the first time they log in.

Since this is the first time Prof. Einstein is logging in, you should be looking at the “Change your Password” page now.

Change Your Password

Thu 11/30/2006 12:26 PM

You logged in correctly, but you cannot continue to use your initial password. Use this form to change it.

When selecting a new password, keep in mind:

- Passwords must be between 6 and 15 characters long and may contain only letters of the alphabet a-z and A-Z, the digits 0-9 and the special characters +=-\$.^*
- Passwords are case sensitive. For example, if you type a new password in UPPERCASE, then you must always type it in uppercase.

Username:	ae4322
Real Name:	Albert Einstein
Current Password:	
New Password:	
New Password (again):	
Change My Password	

Enter:

Current Password: ae4322
 New Password: relativity
 New Password (again): relativity

You should now see Prof. Einstein's “My Manhattan” page, which lists all of his courses. There's only one course there at the moment:

[Password](#)
[Personal Info](#)
[Logout](#)

Albert Einstein

Red stars in this group mark classes with unread messages

Spring 2003		
	PHY 101 01	Introduction to Physics <i>Prof. Einstein</i>

Click on the link to “PHY 101 01” to enter that course. The first time the teacher enters a newly created course, he's brought to a page that asks him to select the *modules* to be used in that course. Scroll down and enable all of Manhattan's modules (you can select either of the two “Enable this module” checkboxes for modules that give you a choice) and click the **Update Configuration** button to submit the form.

Welcome to your virtual classroom!

Before you can begin using this system, you must decide which 'modules' you'd like to use for this course. You can also use this form to adjust details such as the title of the course, the semester, etc. The decisions you make now can easily be changed at any time. To do so, return to this screen by clicking on the 'Configuration' button on your classroom's Main Menu and select the first item in the Configuration Menu. Note that you must enable at least one module by selecting it from the list below.

This options on this page are divided into **four sections** - scroll down to see them all. For your convenience, each section has a "Update Configuration" button. Clicking on any of the "Update Configuration" buttons **will submit the choices you made in all four sections** and deliver you to the classroom's main menu.

The first time a teacher enters a course, he must select the "modules" he'd like to use.

Scroll down, enable all of the modules, and click on one of the **Update Configuration** buttons.

Section 1: Which modules do you want to use? You can enable or disable a module whenever you wish. Disabling a module does NOT delete the messages within the module!

Module	Disable completely	Enable, and allow users to hide/unhide topics h?	Enable, don't allow users to hide/unhide topics h?
--------	--------------------	---	---

You'll next see a page that lists the students who have been added to your course:

PHY 101 01
Introduction to Physics
Spring 2003 *Prof. Einstein*

For your information, the following students have automatically been added to this classroom. (Click the Continue button AT THE BOTTOM OF THE PAGE after reviewing the list.)

StudentId	Username	Realname	Added time
X343888	nb3888	Neils Bohr	Thu 12/08/2005 09:06 AM
X347332	in7332	Isaac Newton	Thu 12/08/2005 09:06 AM
X394032	mj4032	Michael Jordan	Thu 12/08/2005 09:06 AM
X334321	ss4321	Steven Stanton	Thu 12/08/2005 09:06 AM
X394452	cs4452	Cindy Slattery	Thu 12/08/2005 09:06 AM
X485734	rf5734	Robert Frost	Thu 12/08/2005 09:06 AM
Continue			

This page is shown to teachers whenever an administrator runs the create course process to either create a new course, or update an existing course. Click the **Continue** button, and you'll be brought to the Main Menu for this course:

You should now be at the "Main Menu" for this course:



PHY 101 01 - Spring 2003
Introduction to Physics - Prof. Einstein

This classroom is locked - **students cannot enter.**
The option to unlock the classroom is under "Basic classroom configuration" on your Configuration menu.



How to effectively use Manhattan as a teacher is well beyond the scope of this administrator's manual - look for the *Manhattan Teacher's Reference* at <http://manhattan.sourceforge.net>. For now, click on the **Configuration** button. You'll get to a page listing things the teacher can do to administer this Manhattan classroom. Make a mental note of what the teacher can do using the commands on this menu. As you'll soon learn, as the system administrator, you can get to the exact same configuration menu for any course on the server.

PHY 101 01
Introduction to Physics
Spring 2003 Prof. Einstein

Basic classroom configuration

Go here to enable and disable modules. You can also change basic information about the course (including the course title and teacher's name), or lock the classroom to keep students out. Message options allow you to enable or disable the spell checker and choose whether or not to use 'emoticons' or fixed-width font etc.

View course roster

Provides a complete list of members of the class and shows their usernames, id numbers, and teams.

Get course roster (select type):

TAB-delimited CSV - Comma Separated Values

Provides a TAB-delimited (or Comma Separated Values) text file of Student IDs and names. This file can be imported into most spreadsheets and grade management packages.

Fixed Width Fields

Provides a fixed width field text file of Student IDs and names. There's one student line. The first column is 35 columns long. The second column is 15 columns long. The roster imports into the commercial Grade Management package - see <http://www.manhattanclassrooms.com>

Scroll down and take a look at the items on the teachers Configuration menu.

As the server administrator, you can access these controls for any classroom on the server.

View student login logs, class access statistics, etc.

Answer questions like: When was the last time Jane logged in?

If you'd like, you can spend time playing around with Prof. Einstein's Physics course. In particular, you might want to:

- Unlock the classroom to allow students to enter. Click on the **Basic classroom configuration** link, found on the **Configuration** menu for the course. Scroll down to "Section 2", check the "Unlock this classroom" radio button, and click the **Update Configuration** button. Manhattan classrooms start off in a locked state, to allow the teacher time to prepare course materials before students enter.
- Click on the **View course roster** link, found on the **Configuration** menu for the course. Print out this list of students in the class. You can use this list to log in as students, if you wish. Remember that a person's initial password is the same as their username.
- If you want, you can try adding a few messages to the classroom.
- When you're done experimenting, click on the **Exit Classroom** button on the classroom's **Main Menu**. You'll get back to the "My Manhattan" page, which lists Albert Einstein's courses. Now click on the **Logout** link in the upper right corner of the page. You can control where users are sent when they logout of Manhattan by changing the EXIT_URL setting in the custom.h file and do a make clean; make install See *Create and edit custom.h* [9].

When you're ready to learn more about the administrative system, move on to the next section.

Create a Course Template for Prof. Einstein

As described in *Normal, Template and Standalone Courses* [22], a "course template" is intended to be a place to permanently store course materials for re-use. We set up an active classroom for Prof. Einstein's "Introduction to Physics" course. As the semester progresses, he's likely to post content like the syllabus

and lecture notes that he'll want to re-use for the next semester. As the system administrator, you will sooner or later be telling Prof. Einstein that you would like very much to delete his old Physics courses to make room on the server. By creating a course template for "Introduction to Physics", you'll be giving Prof. Einstein a safe place to put his lecture notes.

Log back into Manhattan's administration system as the super user. Again, the URL looks something like this:

```
http(s)://YOURSERVER.edu/manhat-sbin/super_doorstep
```

Use your favorite text editor to create a new file with these lines:

```
PHY 101
Introduction to Physics
Course Templates
Prof. Einstein
X34322  Albert Einstein
```

(Again, the format of this file is detailed in *More about the roster text file* [87].) Select Courses → Create courses → ...from roster file from the menu.

Open the drop-down list next to the "Group:" label, and note again that the group we created earlier named "try" is now on the list. (You'll see how to give this group a description later in this tutorial.) It turns out that since we are about to create a course template, it doesn't matter which group you select.



When creating a course template, Manhattan ignores the group you select on the "Create courses" form. Course templates are automatically placed in a special group for each teacher on the system.

Type **phy101** into the "Internal Course Name" field. Click on the "Browse" button and find and open the text file you just saved on your PC. *Be sure to select "Template" for the "Course type"*. After filling in the form, it should look like the image below. Of course, the path to the roster file will be different, depending on where you saved it on your PC.

Use roster file

Group:	<New Group>	Browse...
Internal course name:	phy101	
Roster file:	C:\roster.txt	Browse...
Course type:	Template	
Create this course		

Click the **Create this course** button to submit the form and create the course template. You should see something like the following:

1 Normal and Template Courses

Group:	Tae4322
Internal Course Name:	Tphy101
Course Code:	PHY 101
Course Title:	Introduction to Physics
Semester:	Course Template
Course Type:	Template
Status:	New Course
Success?	Yes

Full Name	Id	Username	Central Account	User Type	Course Status	Central Status
Albert Einstein	X34322	ae4322	Existing User	Faculty	Add Successful	Success
Total Users: 1						

Notice that Manhattan has automatically put the template into a group named Tae4322 (ae4322 is Albert Einstein's username), and that it has inserted an uppercase 'T' (for Template) in front of the "Internal Course Name" you specified on the form.

Now login as Prof. Einstein again, by visiting [http\(s\)://YOURSERVER.edu/manhat-bin/doorstep](http(s)://YOURSERVER.edu/manhat-bin/doorstep) His username is: **ae4322** If you followed this tutorial, his password is now: **relativity** You should see his course template listed in addition to his active Physics 101 course. Once Prof. Einstein learns how to use Manhattan's **Copy and Paste** commands, he can freely move course materials to and from his course template. Someday, as the system administrator you will delete the active Intro to Physics course with the students, but you'll promise the teachers they can keep their course templates "forever".



As I said before, the use of Manhattan from a teacher's perspective is covered in a separate manual, look for it at: <http://manhattan.sourceforge.net> Feel free to experiment all you wish with the sample courses you create in this tutorial. You'll be completely deleting all of the courses and users at the very end, leaving your installation fresh for your real courses.

Create a standalone course

Prof. Einstein assists students with a Physics club. The club is open not only to students in your school, but also to students, teachers, and parents associated with any school in your town. He'd like to use Manhattan as a way to keep club members informed of meetings.

A Manhattan "standalone course" is the best choice where there are students/teachers from outside of your school involved in the "classroom". As described in *Normal, Template and Standalone Courses* [22], the user information for standalone courses is completely separate from the central store of usernames/passwords on your server. This makes it possible for the teacher or system administrator to freely add students and other teachers using fictitious ID numbers, or even fictitious names if desired.

Since the username/password information is not centralized for a standalone course, a standalone course won't show up when a person logs into Manhattan in the normal way, that is, by visiting [http\(s\)://YOURSERVER.edu/manhat-bin/doorstep](http(s)://YOURSERVER.edu/manhat-bin/doorstep). Instead, they must visit a separate URL for each standalone course. Nevertheless, you'll find that the system administrator can manage standalone courses via the administrative program, much like normal and template courses.

To create a standalone course for the Physics Club, create a text file with the following lines:

```
Phy Club
The Physics Club
2003-2004
Albert Einstein
X34322 Albert Einstein
000001 John Bellows
000002 Michael Adams
000003 Sara Smith
```

The above roster text file includes Albert Einstein as the teacher of the class. We used his real ID number X34322 in the 5th line, the same one used to create his “normal” course and his “course template” earlier in this tutorial. We did this only to make his username for this standalone course the same as for his “centralized” courses, but we didn’t have to. The other three people listed in the file are founding members of the club. Since they don’t go to this school, we have no idea what their ID numbers are (since they aren’t in our school’s information system), nor do they have other Manhattan accounts. Therefore, we just invented their ID numbers. (We could have used, say, the last four digits of their phone number, or any other string of characters that contains at least one digit 0-9.)

Now login to the administrative system as super-user, click the **Create courses** link, and fill in the form as follows:

- Select “<New Group>” and type **clubs** as the name of the new group.
- Use **physics** as the “Internal Course Name” of the course.
- Click **Browse** to open the “Roster file ” you just created on your PC
- Be sure to select **Standalone** as the “Course type”

Use roster file

Group: <New Group> ▼

Internal course name:

Roster file:

Course type: Standalone ▼

As usual, the location and name of the “Roster file” will vary according to what you called the file and where you saved it on your PC. After clicking the **Create this course** button, the standalone course will be created and you’ll get a page like this:

1 Standalone Courses

Group:	clubs
Internal Course Name:	physics
Course Code:	Phy Club
Course Title:	The Physics Club
Semester:	2003-2004
Course Type:	Standalone
Status:	New Course
Success?	Yes

Full Name	Id	Username	Central Account	User Type	Course Status	Central Status
Albert Einstein	X34322	ae4322	N/A	Faculty	Add Successful	N/A
John Bellows	000001	jb0001	N/A	Student	Add Successful	N/A
Michael Adams	000002	ma0002	N/A	Student	Add Successful	N/A
Sara Smith	000003	ss0003	N/A	Student	Add Successful	N/A
Total Users: 4						



Although we created a new group named `clubs` to hold this standalone course, we could have stored the course in the existing course group named `try`. “Normal” and “standalone” courses can be intermixed within a course group.

Now login as Prof. Einstein again, by visiting `http(s)://YOURSERVER.edu/manhat-bin/doorstep` His username is: **ae4322** If you followed this tutorial exactly, his password is: **relativity**. You will *not* see the standalone Physics Club classroom you just created! As I said, standalone courses, well, “stand alone”. Standalone courses always have their own URL which takes the form:

`http(s)://YOURSERVER.edu/manhat-bin/doorstep?class=GROUP/INTERNAL_COURSE_NAME`

where `GROUP` is the name of the “course group” named when the course was created, and `INTERNAL_COURSE_NAME` is the “internal course name” given to the course when it was created. Generally the system administrator, i.e. you, will maintain a list of links to standalone courses at a web site. (A link to the Physics Club standalone course would look nice on the Physics Club website, wouldn't it?) As a specific example, to login to the Physics Club classroom, visit:

`http(s)://YOURSERVER.edu/manhat-bin/doorstep?class=clubs/physics`

This is the first time Prof. Einstein is logging in to this standalone course, so his password is the same as his username: **ae4322**. After logging in for the first time, you'll be forced to select another password. While you might change Prof. Einstein's password to **relativity** to keep it the same as his other courses, keep in mind that the password for this standalone course is entirely independent from the password for his centralized courses. A person could have a different password for each of their standalone courses.

Configure courses

Now that we've created a few courses, we can continue our tour by looking at the features available by selecting `Courses → Configure courses` from the menu.



Log in to the administrative system as the super-user and make the menu selection shown in the above screenshot. You'll see something like this:

This form asks the question “Which courses do you want to work with?” The text box is used to specify all or part of the “Internal course name” given to the course when it was created. Leaving this field blank will search for all of the courses.

While you can leave the text box empty, you must specify which group or groups you are interested in. If you're following the tutorial, there will be three groups available:

1. The `<Templates>` group is always available. Every course template you create will be located within a sub-group, one for each teacher, under the `Templates` group.
2. We created the `clubs` group when we created the standalone Physics Club classroom. (See *Create a standalone course* [33])
3. We created the `try` group when we created Prof. Einstein's “Introduction to Physics” class. (See *Creating normal courses* [24])



You can select more than one group from the selection list by holding down the **Ctrl** key as you click on the groups.

Both the `clubs` and the `try` groups have the words (No Description) next to them in the selection list. We'll add a description in a moment. Click on the `try` group, leave the Internal course name text box empty, and click on the **Search** button. You'll get something like this:

Configure Course -search results

Search string	
within course group(s):	
	try (Click Here to describe this group.)
Include ___ courses:	Central and standalone courses
Matches must search string:	Contain
Matching records found:	2

Open All | Close All

Configure Course -search results

- 2 central(normal) courses
 - try (No description)
 - e
 - ENG 101 01 English Composition I Spring 2003 Prof. Jenkins (try/engl10101)
 - p
 - PHY 101 01 Introduction to Physics Spring 2003 Prof. Einstein (try/phy10101)

Changing a group's description

Since we left the search box blank, the page lists all of the courses within the try group - there are two of them at this time. Note that they are sorted by the first letter of the "Internal Course Name" specified when the course was created. Let's start by giving the try course group a description. Click on the either of the two links shown below:

Search string	
within course group(s):	
	try (Click Here to describe this group.)
Include ___ courses:	Central and standalone courses
Matches must search string:	Contain
Matching records found:	2

Open All | Close All

Configure Course -search results

- 2 central(normal) course(s)
 - try (No description)
 - e
 - ENG 101 01 English Composition I Spring 2003 Prof. Jenkins (try/engl10101)
 - p
 - PHY 101 01 Introduction to Physics Spring 2003 Prof. Einstein (try/phy10101)

Click on either of these links to give the try group a description.

Type in the following, and submit the form:

Edit Description for Course Group tryDescription:

Notes:

```
This course group is just for practice and will be deleted at
the end of the tutorial.
```

Now the group `try` will have the description “For the tutorial” wherever it appears throughout the administrative system. You can change the description (or view any notes you’ve added) by returning to this page and repeating the process.



Since you can add a short description and extensive notes to a course group, you should resist the urge to use long names when you create new groups as you create new courses. In our example `try` is internally the name of a directory on the server. Group names should be kept short! See *Choosing a group name [88]* for more information.

Accessing a course's configuration menu

The **Configure courses** item on the **Courses** menu is the route to the *same Configuration menu* the teacher sees when he clicks on the **Configuration** button from the course's main menu.

Configure Course -search results

Search string	
within course group(s):	try (For the tutorial)
Include ___ courses:	Central and standalone courses
Matches must search string:	Contain
Matching records found:	2

Open All | Close All

Configure Course -search results

2 central(normal) course(s)

try (For the tutorial)

e

ENG 101 01 English Composition I Spring 2003 Prof. Jenkins (try/engl10101)

p

PHY 101 01 Introduction to Physics Spring 2003 Prof. Einstein (try/phy10101)

Click on a course to get to that course's Configuration menu.

This leads to the same menu the teacher of the course sees when he clicks on his **Configuration** button from the course's main menu:

PHY 101 01 - Spring 2003
Introduction to Physics - Prof. Einstein



If you're following this tutorial, you've already visited the Configuration menu for the Physics 101 course when you were logged in as Prof. Einstein. There are many options behind this menu. Briefly, you can:

- Add and remove students and teachers from the classroom.
- Get a list of students/teachers in the classroom.
- Enable/disable Manhattan's modules and change the course title.
- See who has logged into this particular class, when they've logged in, and what they did while they were there.
- Organize students into teams for use in Manhattan's Team and Team/Teacher modules.



Although the system administrator can enter any course's Configuration menu, he can *not* enter other areas of the course. The administrator can *not* read any messages, or view grades or chat transcripts!



Although 99% of the administrator's capabilities within this Configuration menu are exactly the same as the teacher's, there's one subtle difference. The administrator always has the capability to add completely new Manhattan users to the system (i.e. people who do not already have other Manhattan classrooms) via an option under the **Add a student** and **Add a teacher** links. This option may or may not be available to the teacher, depending on how the custom .h was configured at the time Manhattan was installed. See *Add a student* [115]. Similarly, administrators get an extra option to “Quickly remove students” that is not available to teachers. See *Delete a student or a teacher* [118]

PHY 101 01
Introduction to Physics
Spring 2003 Prof. Einstein

Basic classroom configuration

Go here to enable and disable modules. You can also change basic information about the course (including the course title and teacher's name), or lock the classroom to keep students out. Message options allow you to enable or disable the spell checker and choose whether or not to use 'emoticons' or fixed-width font etc.

View course roster

Provides a complete list of members of the class and shows their usernames, id numbers, and teams.

Get course roster (select type):

TAB-delimited CSV - Comma Separated Values

Provides a TAB-delimited (or Comma Separated Values) text file of Student IDs and names. This file can be imported into most spreadsheets and grade management packages.

Fixed Width Fields

Provides a fixed width field text file of Student IDs and names. There's one student line. The first column is 1 and is 35 columns long. The second column is 36 and is 15 columns long. The roster imports into the commercial Gradebook management package - see <http://www.manhattan.edu>

Scroll down and take a look at the items on the teachers Configuration menu.

As the server administrator, you can access these controls for any classroom on the server.

View student login logs, class access statistics, etc.

Answer questions like: When was the last time Jane logged in?

Configuring Course Templates

Let's now take a look at the course template we created earlier in the tutorial. On the Configure Courses search form, select <Templates> from the course group list, and click the **Search** button:

Internal course name: within course group(s):

Include ___ courses: ☒ Standalone and Centralized ☐ Only Standalone ☐ Only Centralized

must ___ search string: ☒ Contain ☐ Start with ☐ End with

<Templates>
 clubs (No description)
 try (For the tutorial)

Search

The page you'll see has the same controls as the one described earlier. A difference is that course templates are further sub-divided into one group per teacher. When you created a course template for Prof. Einstein's course in *Create a Course Template for Prof. Einstein [31]*, Manhattan automatically created the group Tae4322 to hold the template. Any additional templates created for this teacher will also be automatically placed under this group.

Configure Course -search results

Search string	
within course group(s):	
Include ___ courses:	C
Matches must search string:	C
Matching records found:	1

Open All | Close All

- Configure Course -search results
- 1 template course(s)
 - Tae4322 (No description)
 - T
 - PHY 101 Introduction to Physics Course Template Prof. Einstein (Tae4322/Tphy101)

The course <Template> group is further subdivided into a group for each teacher. These subgroups are created automatically.

The name of the group will be the letter T followed by the teacher's username.

Notice the link "(No Description)", next to the group name. As for your try group, you can, and should, give this template group a description. **Click on the No Description link**, type in something like the following, and submit the form:

Edit Description for Course Group Tae4322

Description:

Notes:

On your own

Before you move on, you should spend a few minutes exploring the simple search capabilities of the **Configure courses** section of the Administrative system. In particular, note:

- You are not limited to displaying courses within one course group at a time. Try **Ctrl-clicking** on the <Templates>, try, and clubs groups and then click the **Search** button. This will display all of the courses on the server.
- Although standalone courses are not connected to Manhattan's centralized login system, the system administrator still has access to them. Try using the Search Courses form to find the one standalone course on your server.

Changing a course's access rights

Home	Courses ▾	People ▾	Server
Welcome Su Each time you (c log file is closed here (provided y - Click on - are left w	Create courses Delete courses Configure courses Change course access Archive courses		

The **Change course access** item on the **Courses** menu allows the administrator to control who can see a course (students, teachers, or both) and whether they can just read the messages or whether they can also

add new messages, or otherwise change the content of the classroom. After making the menu selection, you'll see a course search form:

Internal course name (leave blank to select all): within course group(s):

<Templates>
 clubs (No description)
 try (For the tutorial)

Include ___ courses: ☒ Standalone and Centralized ☐ Only Standalone ☐ Only Centralized

must ___ search string: ☒ Contain ☐ Start with ☐ End with

As for the **Configure courses** command, this form allows you to select the courses you're interested in working with. For this tutorial, leave the text box blank, and select our *try (For the tutorial)* course group, and click the **Search** button. You'll see:

Search string	
within course group(s):	
	try (For the tutorial)
Include ___ courses:	Central and standalone courses
Matches must search string:	Contain
Matching records found:	2

Select the courses for which you'd like to change access rights from the list below.

[Open All](#) | [Close All](#) | [Check All](#) | [Uncheck All](#)

Change Course access -search results

- 2 central(normal) courses
 - try (For the tutorial)
 - e
 - ☐ ENG 101 01 Spring 2003 Prof. Jenkins (try/engl10101) **All:** Full access
 - p
 - ☐ PHY 101 01 Spring 2003 Prof. Einstein (try/phy10101) **All:** Full access

For the course(s) selected above, change the access rights to:

- ☐ All: Full access
- ☐ All: No access
- ☐ Teachers: Read-only, Students: No access
- ☐ All: Read-only
- ☐ Teachers: Full access, Students: No access
- ☐ Teachers: Full access, Students: Read-only
- ☐ Capture mode (experimental)

The options are listed at the end of the list of courses. They are:

All: Full access

This is the normal setting. The course will appear on the My Manhattan page for both students and teachers, and both students and teachers can add messages to the course.

All: No access

The course will *not* appear on anyone's My Manhattan page. Students and teachers simply cannot enter the course

<i>Teachers:</i> Read-only, <i>Students:</i> No access	The course is not listed on students' My Manhattan pages. Only teachers can enter the course, but the command buttons used to add new messages, or otherwise change the course are disabled.
<i>All:</i> Read-only	Both students and teachers can enter the course and read messages, but the commands used to add new messages or otherwise change the course are disabled
<i>Teachers:</i> Full access, <i>Students:</i> No access	Teachers have full read/write access to the course. The course doesn't exist as far as students can tell.
<i>Teachers:</i> Full access, <i>Students:</i> Read-only	Teachers have full read/write access to the course. Students can enter the course, but cannot add any new messages.
Capture mode (experimental)	This experimental mode is intended to allow the system administrator to “capture” a snapshot of the course using an offline browser program. When fully developed, the administrator will be able to use this feature to the snapshot of the (finished) course on a CD for archival purposes. Both students and teachers are unable to enter the course while in capture mode. (The administrator can enter the course using a special URL.) In addition, the course will behave in such a way to make it possible for the offline browser package to efficiently capture the course. This feature is not documented as of this writing, so please don't use it.



For “standalone” courses the behavior is similar to that listed above. However, when the access rights are set to “no access” for an individual, he'll get a “Request denied” message when he logs in.

The screenshot shows a web interface titled "Change Course access - search results". It displays a tree view of courses under the heading "2 central(normal) course(s)". The tree includes a folder "try (For the tutorial)" which contains two sub-items: "e" (ENG 101 Spring 2003 Prof. Jenkins) and "p" (PHY 101 01 Spring 2003 Prof. Einstein). The "p" item is selected. To the right of the tree, a list of access rights is shown for the selected course(s). The list includes: "All: Full access", "All: No access", "Teachers: Read-only, Students: No access" (which is selected), "All: Read-only", "Teachers: Full access, Students: No access", "Teachers: Full access, Students: Read-only", and "Capture mode (experimental)". At the bottom of the list is a button labeled "Change the access setting for selected course(s)".

Annotations in the image:

- A yellow box labeled "Select a course." points to the "p" item in the tree view.
- A yellow box labeled "Select the access rights." points to the "Teachers: Read-only, Students: No access" option in the list.
- A yellow box labeled "Click the button." points to the "Change the access setting for selected course(s)" button.

If you're following the tutorial, try changing the access rights for Prof. Einstein's “Introduction to Physics” class as shown in the above illustration. Then log in as Prof. Einstein. You should know the URL for teachers and students on your installation by now! (If you've forgotten, see *Logging in as Prof. Einstein* [27].) I will remind you that Albert's username is **ae4322** and his password is **relativity**. The access rights of the course are clearly shown on his My Manhattan page:

Albert Einstein

Red stars in this group mark classes with unread messages

Course Template		
 PHY 101	Introduction to Physics	Prof. Einstein

[Check for unread messages](#)

▲ **Spring 2003** (This group is NOT being checked for unread messages.)

PHY 101 01	Introduction to Physics	Prof. Einstein	Teachers: Read-only, Students: No access
----------------------------	--------------------------------	----------------	--

Now, on your own:

- Enter the Physics 101 course as Professor Einstein, and see for yourself that he cannot create new messages - the **New Item** buttons within each module are disabled, for example. Also click on the **Configuration** button and notice that most of the configuration commands you saw earlier in this tutorial are disabled. You can't add or delete students from the class, for example.
- Try logging in as one of Prof. Einstein's students. For example, Robert Frost's username and initial password are both: **rf5734**. Although he's a student in Prof. Einstein's Physics class, the class isn't listed on his My Manhattan page, since you set the access rights to the course as “*Teachers: Read-only, Students: no access*”.
- When you are done experimenting, log back in to the administrative system as the super user, and change the course access rights back to “*All: Full Access*”

View/modify/delete users

The next stop on our tour is the **View/modify/delete users** area of the administration system.

Home	Courses ▾	People ▾	Server ▾	Surveys
Welcome Super User Each time you (or another administrator) log file is closed and a new one is started here (provided you have the proper permissions)		View/modify/delete users Who is on Manhattan? Login logs		

This frequently used feature allows the administrator to search the centralized list of users on the server. For any user you can:

- Look up their username, real name, and ID number.
- View and edit their “Personal info page”
- See what courses they belong to, and whether they are a student or a teacher in each course.
- Remove them from a course, if they are a student.
- Make minor changes to their real name.
- Reset their password to be the same as their username.
- View their login records.
- Completely remove them from the system, if they no longer are a member of any classroom.



You cannot work with user accounts of students and teachers in “standalone” courses from this area of the administration system. Recall that “standalone” courses are not connected to the central user lists, which is searched by these utilities. You can, however, do many of the same things for standalone course users, by entering the course via the **Configure courses** choice on the administrator's menu.

The details in this section assume you have followed the tutorial from the beginning. At this point we have created a few courses and have two teachers and a dozen or so students on the system.

The search users form

Select People → View/modify/delete users from the menu. You'll get the following form:

Search for (leave blank to select all):

Search data fields: ☒ All fields ☐ Username ☐ Id Number ☐ Real Name

Include: ☒ All users ☐ Only students ☐ Only teachers ☐ Users without classes

Report type: ☒ Quick list ☐ Full records

Search is case sensitive: ☒ No ☐ Yes

Sort by: ☒ Last names ☐ Usernames ☐ Id numbers ☐ First names

Matches must __ search string: ☒ Contain ☐ Start with ☐ End with

[List users without classes for quick deletion](#)

The form has a text box where you can type in all or part of something that identifies a person, including their username, their real name, or their ID number. As for searching for courses, if you leave that text area blank, you will include all users.

Whatever you type in the text box (if anything) is filtered by the selections made with the radio buttons on the rest of the form. Take a moment to review what your choices are. We won't go over every possibility, but instead we'll do some basic searches.



The “List users without classes for quick deletion” link is not really part of the search form. We'll use that at the very end of the tutorial to remove all users from your installation, after we've deleted their courses.

A quick list of all users

Let's start with the easiest search. With the form set just like it's shown in the previous section, click the **Search** button. This provides a “quick” list of all users on the system (again, except those in “standalone” courses).

The "quick list" shows the users that matched your search criteria.

You can click on a username to see that person's full record.

The "Personal info" column shows whether the person has entered anything into their "Personal info" page.

You can click on a "Yes" link to view/edit that person's Personal Info.

Search string:	
Search data fields:	All fields
Include:	All users
Search is case sensitive:	No
Sort by:	Last names
Matches must __ search string:	Contain
Matching records found:	13
Showing records:	1-13

Username	Personal info?	Name	Id
nb3888	No	Neils Bohr	X84022
mc4478	No	Max Planck	X84022
te4096	No	Thomas Edison	X394096
ae4322	Yes	Albert Einstein	X34322
rf5734	No	Robert Frost	X485734
rf7389	No	Robert Fuller	X347389
jj5753	No	Janet Jackson	X485753
jj4321	No	Julius Jenkins	X34321

Viewing Prof. Einstein's profile

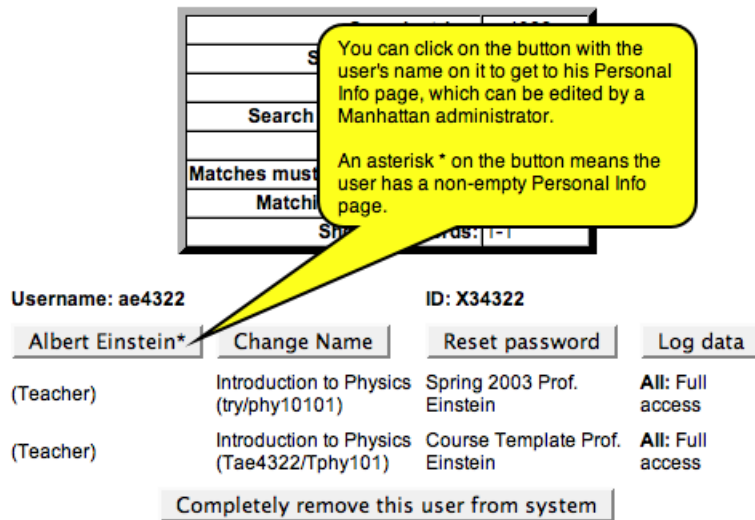
From the "quick list" of users, click on Albert Einstein's username to get to his full profile:

Search string:	
Search data fields:	All fields
Include:	All users
Search is case sensitive:	No
Sort by:	Last names
Matches must __ search string:	Contain
Matching records found:	13
Showing records:	1-13

Username	Personal info?	Name	Id
nb3888	No	Neils Bohr	X84022
mc4478	No	Max Planck	X84022
te4096	No	Thomas Edison	X394096
ae4322	Yes	Albert Einstein	X34322
rf5734	No	Robert Frost	X485734
rf7389	No	Robert Fuller	X347389
jj5753	No	Janet Jackson	X485753
jj4321	No	Julius Jenkins	X34321

Click on Albert Einstein's username to view his full record.

After clicking Albert Einstein's link, you'll see this:



The information on this page should be largely self-explanatory. Nevertheless, here's a quick rundown of the items. Feel free to experiment by trying the various commands.

- Prof. Einstein is a teacher in the PHY 101 01 course, stored internally in the course group try, and with the internal course name of phy10101
- He also owns one course template for his Physics course stored internally as Tae4322/Tphy101
- If Albert tells you that he forgot his password, you could click on the **Reset Password** button. After saying OK to a confirmation prompt, his password will be reset to be the same as his username again, ae4322. He can use that password to login, and Manhattan will force him to select another password.
- You can click on the **Log Data** button to view a history of Prof. Einstein's logins. The events recorded in the logs are described in *Login Logs* [53].
- If for some reason, Albert prefers to be known as “Al”, you can click the **Modify Name** button to make the change.
- The button labeled **Completely Remove this User from System** won't do anything for Prof. Einstein but give you a message “You cannot remove a user until you have deleted him from all of his classes.”. Because every classroom must have a teacher, you are not able to remove teachers from a classroom using this form. (If a course has more than one teacher, you can remove one of the teachers by entering the configuration menu for the course by searching for it under the “Configure courses” option of the administrative menu. See *Delete a student or a teacher* [118].)

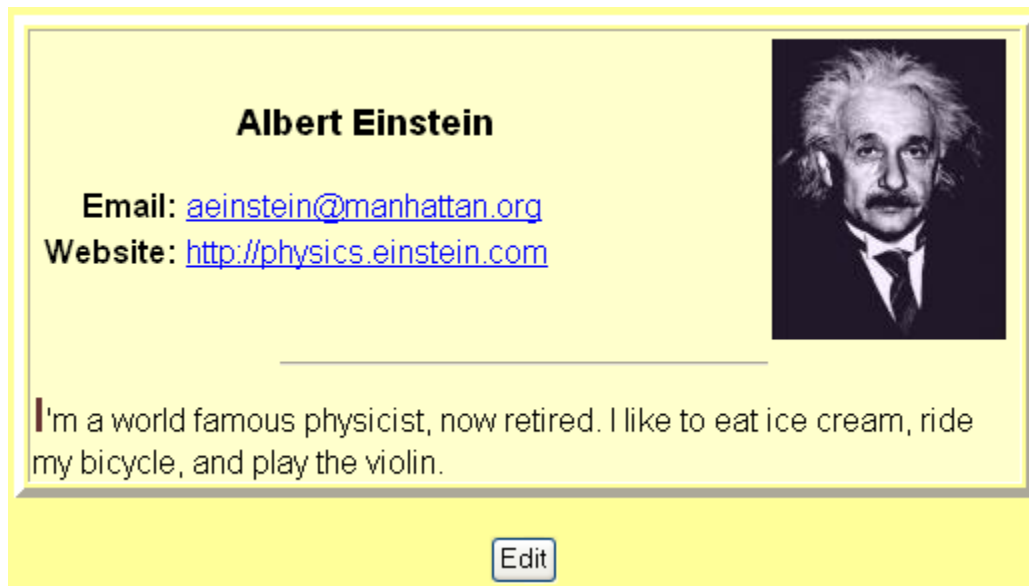
About a user's “Personal Info”

As described in *Manhattan Teacher's Reference*, Manhattan allows users to create and maintain a simple “Personal Info Page”. They can update their personal info to list their email address, web site, and a number of user defined fields for listing things like phone numbers, an office address, etc. A person's personal info can also include a photograph and a narrative. If a teacher enables the People module within a Manhattan classroom, students can see each other's personal info pages. In addition, all Manhattan messages then will include links to the personal info of the sender.

As the server administrator, you can view and edit the personal info of any user on the system (except for those in standalone courses.) Having the ability to view the personal info of a user can come in handy when

providing help desk support, if the person has included some contact information such as their email address. The ability to edit anyone's personal info was given to the administrator for two reasons. First, it allows administrators to help users update their personal info. Second, it allows the administrator to delete objectionable photos posted by immature students.

To view or edit Professor Einstein's personal info, you can click on his name while viewing his full record (see the above screenshot). Here's what Prof. Einstein's personal info might look like:



Albert Einstein

Email: aeinstein@manhattan.org
Website: <http://physics.einstein.com>

I'm a world famous physicist, now retired. I like to eat ice cream, ride my bicycle, and play the violin.

Edit

If you'd like, you can experiment by creating a personal info page for Prof. Einstein - click the **Edit** button while viewing his personal info page.

A few notes about users

Here's some general comments about working with users:

- This section of the administrative system can only give you information about users who belong to “normal” courses, or “course templates”. Users in standalone courses are not listed, because, well they are standalone courses! You can however, still do things like reset a user's password in a standalone course by accessing the course from the **Configure courses** link on the administrative menu.
- A person is defined as a “teacher” if they are a teacher in one or more courses. A person can be a teacher in one course and a student in another.
- As you'll see at the end of this tutorial, *a person retains their Manhattan account even when they no longer are a student or a teacher in any classroom on the server*. The search form has an option for listing “Users without classes” and you can also use the link **List users without classes for quick deletion** below the form to see who these users are, and to delete them from the server if you wish.
- The following tasks can be done either from the user's full record under the **View/modify/delete users** option, or from the course's configuration menu accessible to you via the **Configure courses** option on the administrator's menu. Since these commands are available on a course's configuration menu, they can be performed by teachers as well as system administrators:
 - Resetting a person's password to be the same as their username.

- Removing a student from a course. (Removing a teacher from a course can *only* be done from the course's configuration menu and only if there is more than one teacher in the course.)

On your own

Feel free to experiment with the options available in this section of the administrative program. A few things you can try include:

- List only teachers, then only students.
- Search for all users with “mar” as part of their name.
- Reset Albert Einstein's password. Log in as him, and change his password back to **relativity**. Back in the administrative system, look at his login records. What gets recorded?
- Delete Marie Curie from all of her classrooms (actually there should be only one). Then search for users without classes. Completely remove her from the system.
- Search for your super-user account. Is it there? (No - system administrators are not included among the users.)

Manage chat

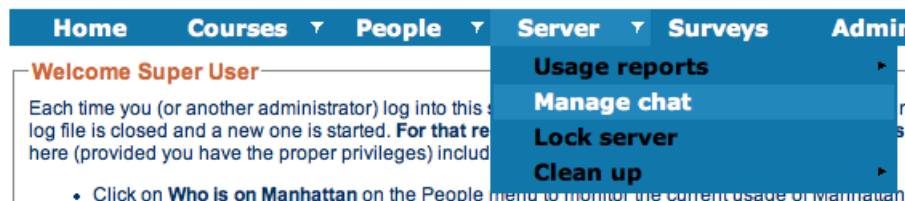
Manhattan's chat module allows students and teachers to exchange short text messages in real time. The use of Manhattan's chat module is fully discussed in its own chapter in Manhattan's *Teacher's Reference*, which you should read for a full understanding of chat. As the system administrator, your responsibilities are to:

- Start the chat service - no one can use chat until the system administrator takes this step.
- Periodically check to make sure the service is still running, and perhaps stop and re-start the service.
- Periodically view/delete the system logs associated with chat, to make sure they don't grow too large.



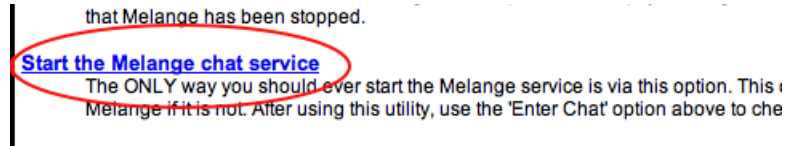
Manhattan uses a modified version of Christian Walter's Melange chat. If you search the web, you may find a security advisory warning of a “buffer overflow” exploit that's possible with Melange chat. **THAT ADVISORY DOES NOT APPLY TO MANHATTAN'S IMPLEMENTATION OF MELANGE!** Those programming errors were removed from Manhattan's chat long ago. Manhattan's chat server maintains the name “Melange” to give credit to its original author, and to describe its heritage. Much work has been put into Manhattan's implementation to increase security and stability.

The tools for controlling the Melange chat service are under Server → Manage chat on the administrator's menu:



Start the Melange chat service

No one can use chat until the system administrator starts the chat service. After selecting **Manage chat** link from the **Server** menu, you'll come to a page with a list of links, where each item is described in some detail. The third item down allows you to start the chat service:



Click on the link now, and you'll get a simple confirmation that a command was issued to start the service.



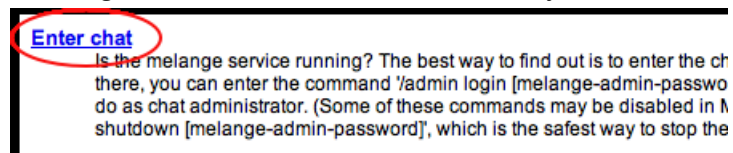
The melange chat service runs on a particular port. The port for your installation is controlled by the `MELANGE_PORT` #define in `custom.h` and is set when you install Manhattan. The default port is 6661. If your network is protected by a firewall, and if you want students and teachers to be able to use the chat module, you must open up access to whatever port chat is running on.



If you are running more than one installation of Manhattan on a server, each installation *must* use a different port number for chat.

Enter Chat

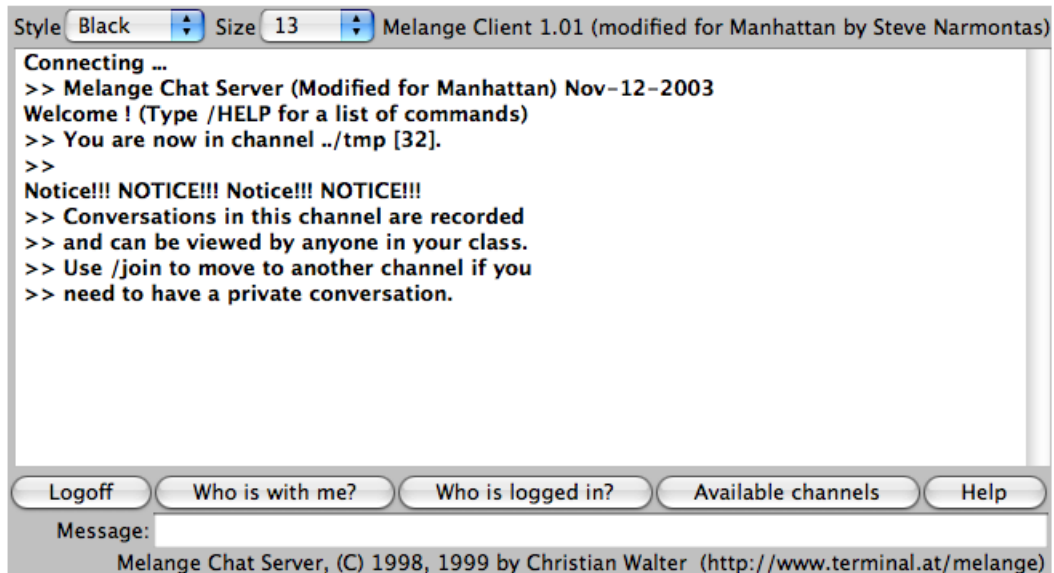
The first item on the Melange Chat Administration menu allows the system administrator to enter chat:



The primary reasons for doing this are:

- To verify for yourself that chat is working.
- To make sure no one is using chat before you stop (and then possibly restart) the chat service.

When all is well with chat, you'll get a window like this:



You can click on the **Who is logged on?** button to see if others are using chat at this moment. You should also try entering some text in the **Message:** area and hitting your Enter key to prove to yourself that chat is working properly. When you are done, click the **Logoff** button, then close the chat window.

A few notes:

- Your web browser must have a recent Java Virtual Machine (JVM), preferably downloaded from Sun at <http://java.sun.com>, installed in order to use chat. The same applies to students and teachers. Consult the chapter on chat in Manhattan's *Teacher's Reference* for details.
- Besides not having a JVM installed (or an old or otherwise incompatible JVM installed) the second most common reason for chat problems is that your firewall is blocking traffic on the chat port. The port chat runs on is listed at the top of your “Melange Chat Administration” menu. It's also told to you when you use the **Start the Melange chat service** option.

After starting the chat service, try logging in as Prof. Einstein. Enter his “Introduction to Physics” course, click on the **Chat** button on the course's Main Menu, then click the **Enter Chat** button within the module. Type a few messages to convince yourself that chat works for this teacher.

View/delete Melange service log

Melange keeps a service log, which is stored as text in the file `melange.log` within the `chat` directory of your Manhattan installation. The **View/delete Melange service log** option allows the system administrator to view or delete that log file:

[View/Delete the Melange service log](#)

Melange maintains a log of system activity. Use this option to view or delete the log file.

[Change the Melange administrator's password](#)

The Melange Administrator's password can be used to become the chat administrator after

Click on the link to view the log. Note the **Permanently Delete this Log File** button at the top. As the system administrator, you should come here at least once a week to review and then delete this log file:

Thu 11/30/2006 02:56 PM

Permanently Delete this Log File
Close this window

```

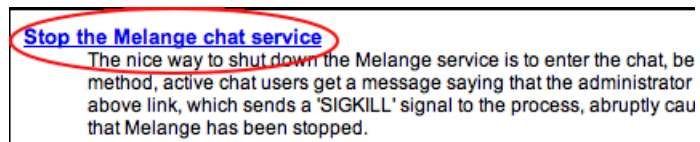
Mon 10/30/2006 08:58:32 AM: NEW CONNECTION: name unknown:71.232.241.111 (c-71-232-24
Mon 10/30/2006 08:58:32 AM: LOGIN:Scott-eph (i206/cjso34951):host c-71-232-241-11
Mon 10/30/2006 09:00:01 AM: LOGOUT:Scott-eph:host c-71-232-241-111.hsd1.ma.comcas
Mon 10/30/2006 09:01:06 AM: NEW CONNECTION: name unknown:71.185 (c-24-34-191-
Mon 10/30/2006 09:01:06 AM: LOGIN:David-eph (i206/cjso34951):host c-24-34-191-
Mon 10/30/2006 09:03:58 AM: NEW CONNECTION: name unknown:71.111 (c-71-232-24
Mon 10/30/2006 09:03:58 AM: LOGIN:Scott-eph (i206/cjso34951):host c-71-232-241-11
Mon 10/30/2006 09:09:32 AM: NEW CONNECTION: name unknown:24.62.83.252 (c-24-62-83-25
Mon 10/30/2006 09:09:32 AM: LOGIN:Randy-DeMello (i206/cjso34951):host c-24-62-83-252
Mon 10/30/2006 09:12:35 AM: NEW CONNECTION: name unknown:72.72.91.185 (pool-72-72-91
Mon 10/30/2006 09:12:35 AM: LOGIN:John-Ouellette (i206/cjso34951):host pool-72-72-91

```

You should come here periodically to review and possibly delete this log file.

Stop the Melange chat service

The second item on the Melange Chat Administration menu allows the administrator to stop the Melange service:



When you click the link, a “kill” signal is sent to the Melange service, telling it to shut down. As described on the page, you can then try to enter the chat to make sure the service has been stopped (in our experience, it always works!):



Recommended periodic chat management

The Melange chat service is different from the other programs that make up Manhattan because it runs as a “daemon” or background process on your server. Once you start the chat service, it continues to run for days, weeks or months (or years?) handling messages typed in by students and teachers on your server. While in our experience the service has been highly reliable, it's a good idea to periodically stop and restart

the service, and to delete the system log. Depending on how heavily your installation is used, I would recommend doing the following “every now and then”.

1. Enter the chat to make sure it's working. While you're there, see who's logged on by clicking on the **Who is logged on?** button. We've noticed that sometimes Melange will incorrectly report that someone is on, who is not. (You can use *Who is on Manhattan?* [73] to help you determine who's really on chat.) This is all the more reason to give Melange a fresh start.
2. Once you're sure you won't be ruining someone else's chat, stop the chat service.
3. Review the chat service log. You can use your browser to print or save a copy of the file if you wish. Once you've reviewed (and optionally saved) the log, delete it.
4. Start the chat service again. Enter the chat to see that it's working, and take a second peek at the service log. You've just given Melange a fresh start!

On your own

Go ahead and play with chat. The full end-user documentation exists as its own chapter in Manhattan's *Teacher's Reference*. As the system administrator, you'll want to make sure that chat is available both on your intranet, and from off-campus. Check the port number assigned to Melange in your `custom.h` file used to compile Manhattan, and your firewall settings.

I didn't tell you about the item on the Melange Chat Administration menu that allows you to set **Change the Melange administrator's Password**. Try setting the password. Then shut down and restart the chat service so the new password takes effect. Next, try the following:

1. Enter the chat.
2. In the message text box, type **/admin login your-password**, where your-password is the password you just set, and hit Enter. You are now logged in as the chat administrator.
3. Now type in the message **/admin help** for a list of things the administrator can do, but READ THE FOLLOWING:



Manhattan uses a modified version of Melange. Some of these administrator commands either have not been tested, or are simply a BAD IDEA for a classroom chat.

4. Despite the above warning, the one admin command that will work is **/admin shutdown your-password**, which provides another way to cleanly shut down the Melange service. Try that now.

Login Logs

Our next stop on our tour is the **Login Logs** item found under **People** on the administrator's menu:



Manhattan's login logs are stored as text files in the `users/logs` directory of your Manhattan installation. The following events are recorded:

- Successful logins - The user typed in both their username and their password correctly.
- Unsuccessful logins - Someone tried to login using a username that exists on the server, but they got the password wrong.
- Password changes by the user - The user changed their own password by clicking on one of the **Change your Password** buttons within Manhattan, or they were forced to do so because their password was the same as their username. (e.g. it was the first time they logged on, or someone just reset their password)
- Password was reset by a teacher or a system administrator - In Manhattan, any teacher can reset the password for a student in their class. As system administrator, you can reset the password for any student or teacher on the system. In either case the password reset event is recorded in the logs, including information on who performed the reset.

The **Login Logs** item on the administrator's menu allows you to view and delete these log files. You'll get a page that looks like this:

Login Logs						
All log files:						
Graphs Log Data Admin Logins						
	File name	Last modified	File size			
Current log	log.txt	Mon 09/29/2003 04:02 PM	284	Graphs	Log Data	
Older logs	log_29Sep2003_0926	Mon 09/29/2003 09:26 AM	1603	Graphs	Log Data	Delete

Your page will look slightly different (there will probably only be a "Current log" listed) if you're working through the tutorial.

How the logs get rotated

The login logs are stored as text files in the `users/logs` directory of your installation. In order to keep the size of the current log file reasonable, the file is automatically "rotated". This means that periodically a backup copy of the current active log file is made, and a new empty log file is started. Here's how the rotation works:

- Whenever you or another administrator logs in to Manhattan's administrative system, the size of the current log file, named `log.txt` is checked.
- If the current log file is greater than the default value of 250,000 bytes, the file is renamed and a new, empty `log.txt` file is created. (The value of 250,000 bytes can be changed via the `custom.h` file during installation.) The name of the "rotated" file includes the day, month, year, hour, and minute of the time the file was rotated.

Login

All log files:

The Current log is listed first. It's always named **log.txt** and is the log file that records are currently being added to as people log into Manhattan.

	File name	Last modified	File size			
Current log	log.txt	Mon 09/29/2003 04:02 PM	284	Graphs	Log Data	
Older logs	log_29Sep2003_0926	Mon 09/29/2003 09:26 AM	263200	Graphs	Log Data	Delete

A (possibly long) list of older logs will appear below the current log.

 These are log files that have been rotated to prevent the current log from getting too large.

Whenever the system administrator (that's you) logs in, the size of the current log file is checked. If it's larger than 250,000 bytes, the file is copied and a new, empty current log file is started.

A **Delete** button allows you to permanently delete the oldest log file on the list.

You can delete the log files one at a time by clicking on the **Delete** button. This **Delete** button only appears on the table row for the oldest log file, to make sure you can only clean up by throwing out the oldest log files first.



You or another administrator must login for the log rotation to occur. Although as system administrator you should login frequently, no harm is done if the log file isn't frequently rotated. Also, if you know something about Unix **cron** and shell scripting, you can have your system automatically rotate the login logs at a particular time of the day or night if you prefer.

All logs vs. individual logs

As shown in the screenshot below, the buttons labeled *All log files:* work on all of the log files, that is, the current log plus any and all older logs. The buttons on the rows of the table act only on that single log file.

Login Logs

The buttons along the top act on all of the log files - the current log plus and all "rotated" logs.

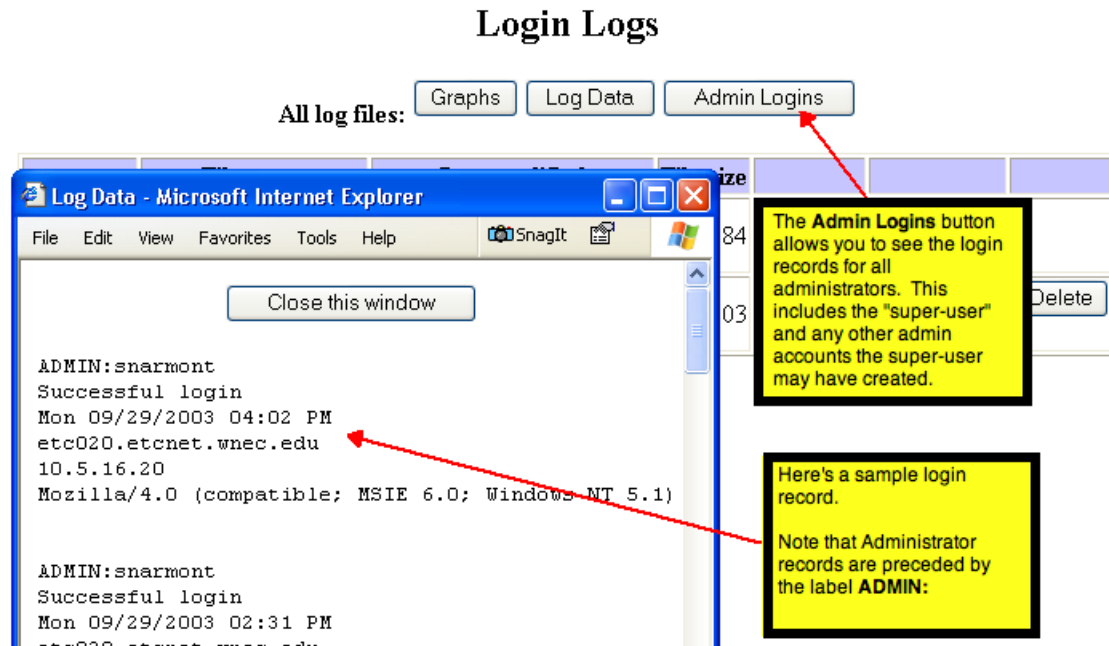
All log files:

	File name	Last modified	File size			
Current log	log.txt	Mon 09/29/2003 04:02 PM	284	Graphs	Log Data	
Older logs	log_29Sep2003_0926	Mon 09/29/2003 09:26 AM	263200	Graphs	Log Data	Delete

The buttons in a table row act only on the records in that particular log file.

Admin logins

The button labeled **Admin Logins** provides you with a plain text listing of the log data for system administrators. (System administrators are those with access to the administrative programs you are reading about now - including you.) Each record written to the log files by an administrator has the prefix **ADMIN:** before the username. This is needed because it is possible for an administrator to have the same username as a student or teacher.



Protecting the security of Manhattan administrative accounts is enormously important. One way to protect those accounts is to check the **Admin Logins** frequently to see whether an intruder is attempting to login as one of the administrators, or if he already has access to an administrative account.

Viewing raw log data

Login Logs

All log files: [Graphs](#) [Log Data](#) [Admin L](#)

Click here to view the combined log data for all log files.

	File name	Last modified	File size			
Current log	log.txt	Mon 09/29/2003 04:02 PM	284	Graphs	Log Data	
Older logs	log_29Sep2003_0926	Mon 09/29/2003 09:26 AM	263200	Graphs	Log Data	Delete

Click here to view the raw log data contained within this particular log file.

As shown in the above screenshot, you can view the raw log data for either all of the log files combined, or for one particular log file. When you click one of the **Log Data** buttons, a separate window opens displaying the raw data:

```
ADMIN:snarmont
Successful login
Sat 09/27/2003 07:40 AM
0-1pool196-162.nas27.somerville
63.159.196.162
Mozilla/4.0 (compatible; MSIE

ae4322
User changed their password
Fri 09/26/2003 04:21 PM
0-1pool212-111.nas31.somerville1.ma.us.da.qwest.net
63.159.212.111
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

ae4322
Successful login
Fri 09/26/2003 04:21 PM
0-1pool212-111.nas31.somerville1.ma.us.da.qwest.net
63.159.212.111
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

Each record shows:

- username
- event
- when
- user's host info (if available)
- user's IP address
- browser/OS info

The records are sorted with the most recent events at the top of the list. Recall from *Viewing Prof. Einstein's profile* [46] that if you want to look up a particular user's login records, you can use the **View/modify/delete users** option from the administrator's main menu. The "full record" for a user includes a **Log Data** button that shows, for all available log files, all of the records that match their username:

Albert Einstein

Username: ae4322
ID Number: X34322

(Teacher) try/phy10101 Spring 2003 Introduction to Physics Prof. Einstein

(Teacher)

When viewing a user's "full record" (under "View/modify/delete users") this button shows the login logs for this user, for all available log files.

Completely Remove this User from System

The ability to quickly check someone's login records as shown above is very useful when providing phone or email support for students and teachers. You can see when they've logged in (successfully or not), and if or when their password was reset by one of their teachers. You can also get an idea of the Internet service provider, web browser, and operating system they have used.

Login Graphs

Login Logs

All log files: [Graphs](#) [Log Data](#) [Admin L](#)

	File name	Last modified	File size			
Current log	log.txt	Mon 09/29/2003 04:02 PM	284	Graphs	Log Data	
Older logs	log_29Sep2003_0926	Mon 09/29/2003 09:26 AM	263200	Graphs	Log Data	Delete

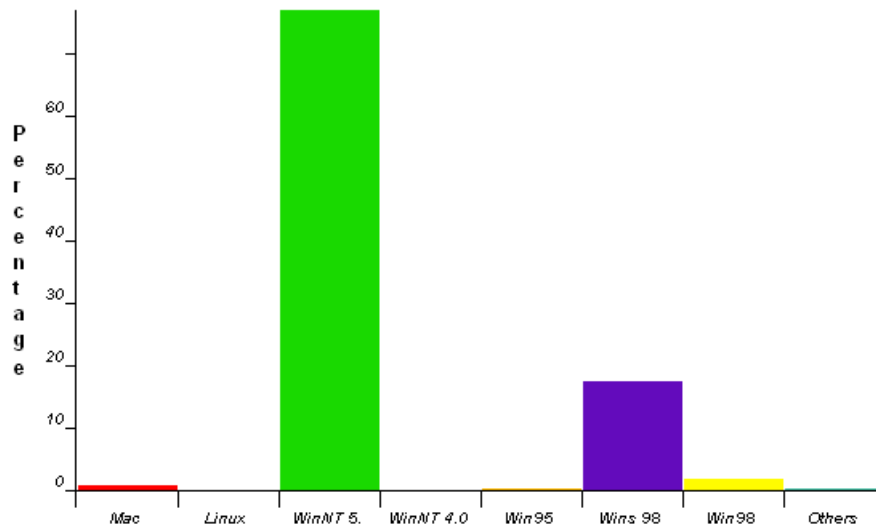
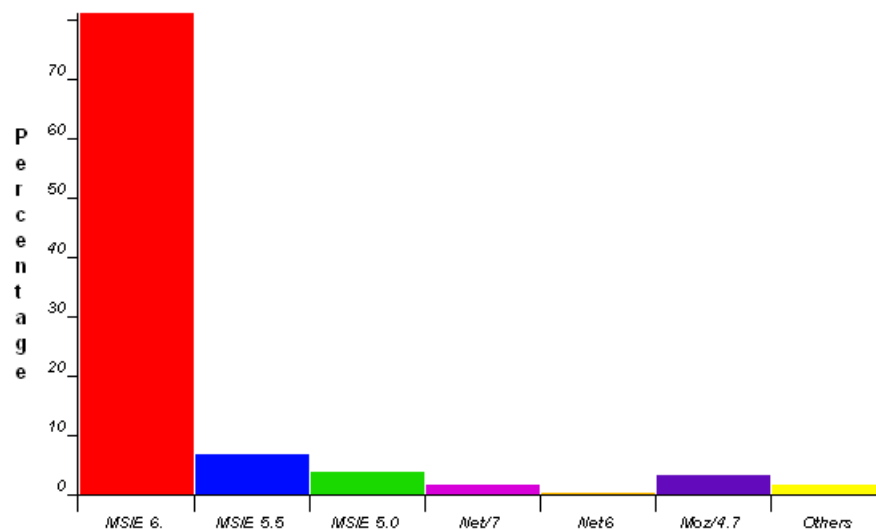
Click here to view the graphs for all log files combined.

Click here to view the graphs for this particular log file.

The login graphs are really a “work in progress”. At this time the buttons shown above open a separate window showing a graphical representation of two data items.

- The browsers (e.g. Netscape, Internet Explorer, other) preferred by your users.
- The operating system (e.g. Windows, Linux, Mac) they use.

Here's sample graphs from an active server:

Operating Systems:**Browsers:**

The above screenshots show graphs for over 80,000 logins. The majority of users run Windows NT 5 (which is Windows XP) and use MS Internet Explorer as their browser. The graphing programs try to parse the lines in the raw log file, which look like this:

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

This data is provided to Manhattan by the web browser a person uses when they connect. Unfortunately, there is no firm standard for the data a browser must provide. We've done the best we can to parse the data into reasonable graphs. This portion of Manhattan will have to be updated as Microsoft and others come out with new versions of operating systems and browsers. Still, this graphical representation will help you understand what users are running when they connect to their classrooms, and help teachers know which browsers they need to support for any special content they provide within their classrooms.



The graphs only include data on “Successful Logins” and exclude logins for system administrators.

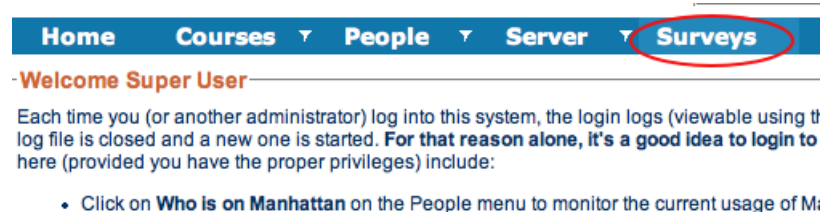
On your own

If you're following along with the tutorial, feel free to experiment with the login logs. In particular, you may want to try to cause the following events to happen:

- The super-user repeatedly logs in with the wrong password (but correct username).
- A teacher resets a student's password.
- A student (or teacher) changes their password.

Surveys

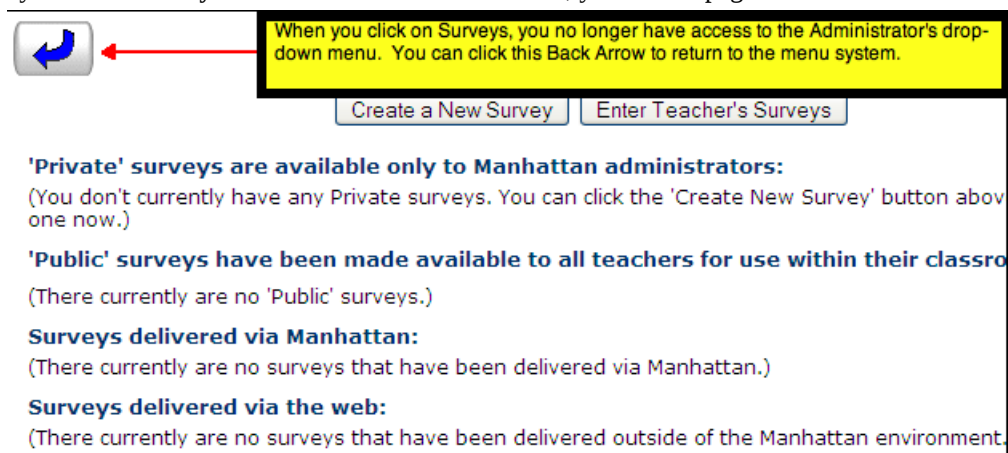
Let's take a look at what's available under **Surveys** on the administrator's menu:



Manhattan has rather complete capabilities when it comes to the design and delivery of surveys. For this part of the tutorial, we'll only briefly touch on a few of the things you can do as system administrator. In particular, we'll design a simple survey and deliver it to all of the students and teachers on your server. See *Surveys - details [122]* to learn more about surveys and Manhattan.

The Administrator's Survey Lists

When you select **Surveys** from the menu for the first time, you'll see a page like this:



Once you've created and delivered some surveys, you'll find them listed here, but initially your list will be empty. You'll see what's behind the **Create a New Survey** button shortly. Manhattan allows teachers to create and manage their own surveys as well. As discussed in *How can I take a copy of a teacher's Private survey? [126]*, the **Enter Teacher's Surveys** button allows the system administrator to see the list of surveys a particular teacher has designed.

Your surveys are divided into four groups:

Private Surveys	These are surveys that can be seen only by you and other server administrators. Your list of Private Surveys will include both surveys you are in the process of designing, and “finalized” surveys that are ready to be delivered or made available to teachers.
Public Surveys	Contains a list of surveys you have chosen to make “public”. Any teacher of a normal, centralized course (see <i>Normal, Template and Standalone Courses [22]</i>) will see these surveys on their own public list when they enable and enter the Survey module for any classroom. Teachers can choose to deliver the survey you made available as-is, or they can copy it into their own Private list and modify it.
Surveys delivered via Manhattan	Any teacher can deliver a survey to his students within the confines of a Manhattan classroom. As you'll see shortly, a Manhattan administrator can also deliver surveys, either to one classroom, to several classrooms, or to every classroom on the server. The “Surveys delivered via Manhattan” list contains surveys that you, as system administrator, have delivered in this manner.
Surveys delivered via the web	In addition to delivering surveys to students and teachers enrolled in Manhattan classrooms, the system administrator can also deliver surveys that are fully accessible to anyone on the Internet. This makes it possible to use Manhattan to create surveys for any purpose, for any population. Surveys delivered in this way will appear under the “Surveys delivered via the web” heading.

Creating a New Survey

When you click on the **Create a New Survey** button, a new window will open with this form:

Specify Survey Title, Heading, and Background Color

Survey Name:

Survey Heading :

Background Color: [Color Chooser](#)

The three elements on the form are:

Survey Name	This is a simple name used to identify the survey within your Private list. While there is a generous size limit, you should keep it relatively short. The only purpose of the Survey Name is to identify it on your
-------------	--

Private list - students will never see what you type here. Do NOT use any HTML codes in this field, and the title can only contain letters of the alphabet and spaces - no punctuation marks.

Survey Heading

Whatever you type here will appear at the top of the survey when it is opened. Essentially, you should simply type a title for the survey. Your title will be centered on the page and displayed in a boldfaced font. Although you can put a few HTML tags here, there really is no need to. Just type a one line title!

Background Color

This specifies the background color of the survey, using standard HTML color codes. The default is #FFFFFF, which gives you a white background. You can click on the Color Chooser link to select a color from a palette. Note that the text color will always be black, so you should definitely avoid selecting dark background colors!



The choices you make on this form can be changed at any time.

If you are following the tutorial, make the following entries then click the **Submit** button.

Survey Name: Computer Ownership

Survey Heading: Computer Ownership Survey

Background Color: #FFFFCC

After clicking **Submit**, you'll get a page like this:

This is the form used to design your survey. First, notice the button **End this editing session** in the upper left corner of the page. Click that button when you are done making changes to the survey.

The page is divided into sections contained by a rectangular border. Each section represents an element of your survey that can be modified. The first section, which is shown in the above screenshot, always represents the title, heading and background color of your survey. The other sections, which you'll add shortly, represent either a question or can be "Text/Instructions" entry where you can enter instructions for the people taking the survey, or any other text you'd like.

At the bottom of each section, you'll find these buttons, a few of which can be "grayed out" if the command does not apply at the moment, or completely missing if the command never applies to a particular section:

Modify

Brings you to a form that allows you to edit the item.

Add Below

Brings you to a page that allows you to add a new question or a custom entry. The new item will appear below the current item.

Copy	Copies the current item to an internal “clipboard”, replacing the last item placed in this clipboard. The item can be “pasted” elsewhere in the same survey by clicking on the Paste button.
Cut	Performs the same as the Copy button, except the current item is deleted from the survey.
Paste	Inserts the last item that has been Copied or Cut below the current item.

The **Paste** button will be grayed-out until you click on one of the **Copy** or **Cut** buttons. Also note that the first item in a survey is special in that it represents the title of the survey, some heading information, and the background color used in the survey. Since there can only be one element of this type in any survey, the **Copy** and **Cut** buttons will never appear for the first item. (see the above screenshot.)

To add the first question to the survey, click the **Add Below** button. You'll get the following page:

Select a Question Type:

[Text/Instructions]

Text/Instructions

A Text/Instruction field is not a question. Instead, it's a place where you can enter any text you'd like, such as instructions for the person taking your survey. The text you type can (should) be formatted with HTML tags. One useful HTML tag for this purpose is `<p>`, which inserts a blank line. For example, typing:

This is the first line. <p> This is line 2.

will look like this in your survey:

This is the first line.

This is line 2.

Text/Instructions:

The row of buttons along the top show you the elements you can insert into a Manhattan survey:

Text/Instructions	“Text/Instructions” elements are not questions at all. Instead a Text/Instructions element can be used to insert HTML formatted text into the survey. Typical uses are to insert general instructions at the top of the survey, or text that describes the purpose or instructions for a particular group of questions, or a thank you note at the very bottom of the survey.
Likert Question	A “Likert scale” question consists of a question or statement, followed by a set of from 2 to 8 radio buttons arranged horizontally. Unlike a multiple-choice question, however, only the left-most and right-most

	choices are labeled. An example might be “This course met my expectations.” Strongly Disagree _____ Strongly Agree
Multiple Choice	This is your typical multiple-choice question. The respondent must select from a list of responses, by clicking on a radio button next to each choice. If you check the “Multi-select” box while creating the question, respondents can select more than one answer.
Short Answer	The “Short Answer” question type provides the respondent with a simple one-line text entry field. They can type whatever they wish into this field. As an example, you might use this question type if you want respondents to enter their name.
Essay/Comments	The “Essay/Comments” question type is similar to the “Short Answer” question type, except the respondent is presented with a rectangular text box where they can type as much as they want. These are typically used to provide a way for respondents to type general comments related to the survey.

Let's start by adding a multiple-choice question to the survey:

- Click on the **Multiple Choice** button near the top of the page.
- Enter the following text into the fields (do not check the Multi-select option), and click the **Submit** button:

Question: How many computers do you own?
 1: None
 2: One
 3: Two
 4: More than two

After clicking the **Submit** button, you'll see the new question added to the survey:

End This Editing Session

Title, Heading, and Background Color
Computer Ownership Survey
Background color: #FFFFCC
Modify Add Below [Paste]

Multiple Choice
How many computers do you own?
☐ None
☐ One
☐ Two
☐ More than two.
Modify Add Below Delete Copy Cut [Paste]

Notice in the above screenshot, that the “Paste” button is not available. That’s because you haven’t yet used “Cut”, or “Copy” to place something in this survey’s clipboard.

- Click the **Copy** button on the multiple-choice question.
- When the page refreshes, click the now-enabled **Paste** button.

After following the above two steps, you’ll now have two identical questions in your survey. Let’s use the **Modify** button to change the second question:

- Click on the **Modify** button on the last question of the survey.
- Use the form that appears to change the question to: “How many computers do you wish you had?” and click the **Submit** button.

On your own

Using the survey creation program is pretty straightforward. You should take the time to experiment by adding questions of the various types to your survey. Here’s a few additional notes that will help:

- You can change the order of the questions by creatively using the **Cut** and **Paste** features.
- Manhattan expects “Text/Instructions” items to be formatted using HTML tags. This is the one place within Manhattan where knowing just a few HTML tags can come in handy.
- Questions and responses can also include HTML tags. However, you should probably not get too creative here. Judicious use of bold, italics, and underlining is usually sufficient.

When you are done adding a few more questions to your survey, click on the **End this editing session** button on the upper left corner of the page. This will return you to your list of surveys, which we’ll explore next.

Delivering a survey

If you’re following the tutorial, you’ve now created your first survey. It’s listed in a table under the “Private Surveys” heading:



Create a New Survey

Enter Teacher's Surveys

'Private' surveys are available only to Manhattan administrators:

						Share		Deliver	
Computer Ownership Survey	Preview	Modify	Finalize	Info	Delete	<i>Make Public</i>	<i>Give a copy to...</i>	<i>Deliver via Manhattan</i>	<i>Deliver via web</i>

'Public' surveys have been made available to all teachers for use within their classrooms:

(There currently are no 'Public' surveys.)

Surveys delivered via Manhattan:

(There currently are no surveys that have been delivered via Manhattan.)

Surveys delivered via the web:

(There currently are no surveys that have been delivered outside of the Manhattan environment.)

As mentioned in *The Administrator's Survey Lists* [60], this page lists an administrator's surveys under four headings. "Private Surveys" are available only to server administrators. They are generally a list of surveys under development, or that have been completely developed and are ready to be delivered.

A number of commands appear as simple links on the row corresponding to the survey you have created:

Preview	Displays the survey as it will be seen by the respondents once it is delivered. You should always preview the survey before delivering it, especially if you've made extensive use of HTML tags.
Modify	Allows you to continue editing a survey that has not been "Finalized".
Finalize	Marks a survey as completed. You must "Finalize" a survey before it can be delivered. Although you can't modify a survey once it's been Finalized, you'll see that you can make a copy of a finalized survey at the click of a mouse.
Info	Provides basic information about the survey, including its original author, and when it was created.
Delete	Used to permanently delete a survey. You'll find you get an opportunity to change your mind after you click the link!

Also notice the "grayed out" links under the *Share* and *Deliver* headings. These words will become live links once the survey is "Finalized".

If you're following along with the tutorial, click the **Preview** link to see how the survey will be displayed to users when it is delivered. The only difference between what you see and what they'll see is that the **Close this Window** button will be replaced with a **Submit** button once the survey is delivered. If you notice something that needs fixing, you can click the **Modify** link to edit your survey.

When you are satisfied with the survey, click the **Finalize** link. Once a survey has been "Finalized" a few things change in the list of available commands:

- The **Modify** link is grayed-out. You cannot change a survey once it has been finalized. However, you can easily make a copy which can be modified by clicking the **Copy** (see below)
- The **Finalize** link is replaced with a **Copy** link. When you click on this link, a copy of the survey is added to your Private list. You can then **Modify** that survey in any way you wish.
- Four additional links become active These are explained further in *Questions and answers* [124]:
 - **Make Public** moves the survey to the "Public" area. Surveys marked Public by the administrator are available to any teacher on the system via a classroom's Surveys module. Teachers can deliver the survey to their own students as-is, or can Copy the survey to their own Private list, where it can be modified to their liking.
 - **Give a copy to** moves the survey to any teacher's Private list of surveys, which is available within a classroom's Surveys module
 - **Deliver via Manhattan**, which we'll explore next, allows the server administrator to deliver a survey to users in one or more Manhattan classrooms.
 - **Deliver via web** allows you to survey people outside of the Manhattan environment. Anyone with web access can answer a survey that has been delivered in this way.

If you haven't already, click the **Finalize** link. When the page refreshes, your available links will change as described in the list above. Now click the **Deliver via Manhattan** link. You'll get a page like the following:

Step 1 of 2
Who will take the survey and when will it be made available?

Who should see and be able to take the survey?

☒ All - Everyone can view/answer the survey
☐ Students - Teachers can view the survey, but only student responses are recorded
☐ Teachers - only teachers can view/answer the survey
☐ Students only - only students can view/answer the survey

Should submissions list the user's courses?

☒ Yes - record the courses a respondent is enrolled in, since it may help with data analysis.
☐ No - don't include the courses a respondent is enrolled in. (Helps keep surveys anonymous.)

When should the survey be made available?

Click to Select a Date 12/09/2005 02 : 24 PM

When should the survey be removed from user pages?

Click to Select a Date 12/12/2005 02 : 24 PM

When should you be able to view the results?

Click to Select a Date 12/09/2005 02 : 24 PM

form below to narrow the list of courses to choose from, then click the Search button to continue to Step 2.

Name (leave blank to select all): within course group(s):

Include ____ courses: ☒ Standalone and Centralized ☐ Only Standalone ☐ Only Centralized

Matches must ____ search string: ☒ Contain ☐ Start with ☐ End with

clubs (No description)
 try (For the tutorial)

There are two steps involved in delivering a survey. The first is to fill out the above form, then click the **Search** button to get a list of Manhattan classrooms to choose from. Step 2 involves actually picking the courses from the list generated from the search done in step 1. Here's the scoop on the choices you need to make on the above form:

Who should see and be able to take the survey?

The choices here are self-explanatory:

- All - Everyone can view/answer the survey
- Students - Teachers can view the survey, but only student responses are recorded
- Teachers - only teachers can view/answer the survey
- Students only - only students can view/answer the survey

Should submissions list the user's courses?

Suppose you deliver a survey to all of the 500+ Manhattan classrooms on your server. A student, Jane Doe, is currently enrolled in 3 Manhattan courses. The question is, on the table that lists the raw data generated by the survey, should there be a column showing which courses a particular respondent is a member of? That is, should the (otherwise

anonymous) data provided by Jane Doe when she answered the survey show that the responses were from someone who was in engl10151, phy10102, and bio13203?

Turning this feature on is useful if you're issuing a survey to a large population of Manhattan users and hope to do some cross-tabulation on the data based on what courses a student is enrolled in. However, in some cases turning this feature on can reduce the anonymity of the surveys, since it may be possible to deduce who was responsible for a particular set of responses.

Don't agonize over a decision here. Select "Yes" unless you are extremely concerned about running an anonymous survey. (If you are only surveying one Manhattan classroom, it doesn't matter what setting you use, since you KNOW who you sent the survey to!)

When should the survey be made available?

The survey will not appear on the "My Manhattan" pages of users until after the date and time you specify. (Use the **Click to Select a Date** button to select a date from a pop-up calendar.

When should the survey be removed from user pages?

Specify a date and time when you want to end the survey. After the date and time specified, the survey will be removed from the "My Manhattan" pages of users who have not yet taken the survey. (A survey is automatically removed from a user's login page the moment they submit their answers to the survey. Manhattan surveys can only be taken once.) You can use the **Click to Select a Date** button to select a date from a pop-up calendar.

When should you be able to view the results?

Specify a date and time after which Manhattan will allow the administrators to see the results of the survey. The setting is useful when you issue an end-of-course student evaluation of instruction. Even though Manhattan surveys are anonymous, students want to know that their teacher won't see the results until after final grades are submitted. By selecting a date that is beyond the due date for final grades, students can be assured that *nobody* can see the survey results until then.

(Actually, the Surveys module used by teachers - see the *Manhattan Teacher's Reference* - within a Manhattan classroom is a better place to issue end-of-course evaluations. Within that module, students are explicitly shown the date and time their teacher will be able to see the results of the survey and the results can be seen only by the teacher, not server administrators.)

The bottom line is that *no one* will be able to see the survey results until after the date and time specified on the form. Most often you will pick the current date and time, so you will be able to see the results as soon as they come in.

The bottom portion of the page displays a Search form that's similar to the one used by the Configure Courses section of the Administrative system. (See *Configure courses* [35] for details on using the form.) The idea is that after setting the conditions listed above, you use the Search form to get a list of courses to choose from.

If you are following the tutorial, make the following selections:

- Who should be able to take the survey? Select "All"
- Should submissions list the user's courses? Select "Yes"
- When should the survey be made available? Enter the current date and time
- When should the survey be removed from user pages? Enter some date/time that's at least several hours into the future to give you time to experiment. You can enter a date/time that's far into the future if you wish.
- When should you be able to view the results? Enter the current date and time so you can see the results immediately.

Leave the search text box blank, click on the *try For the tutorial* group, and click the **Search** button. (See the screenshot above.) You'll be brought to the following page:

Search string	
within course group(s):	
	try(For the tutorial)
Include ____ courses:	Central and standalone courses
Matches must search string:	Contain
Matching records found:	2

Step 2 of 2: Select the courses that will receive the survey

Open All | Close All | Check All | Uncheck All

 Survey Course List

-  2 central(normal) courses
 -  try (For the tutorial)
 -  e
 -  ENG 101 01 English Composition I Spring 2003 Prof. Jenkins (try/engl10101)
 -  p
 -  PHY 101 01 Introduction to Physics Spring 2003 Prof. Einstein (try/phy10101)

Submit

At this point, you need to select the courses you want to deliver the survey to. For our tutorial, select both courses, click the **Submit** button, and click **OK** when asked to verify your selections. Your survey will be delivered, and you will be returned to your list of surveys, which now looks like this:

[Create a New Survey](#)

[Enter Teacher's Surveys](#)

'Private' surveys are available only to Manhattan administrators:

(You don't currently have any Private surveys. You can click the 'Create New Survey' button above to create a new Private survey.)

'Public' surveys have been made available to all teachers for use within their classrooms:

(There currently are no 'Public' surveys.)

Surveys delivered via Manhattan:

						Survey Results		
Computer Ownership Survey	Preview	Copy to your Private list	Info	Delete		Bar Graphs	Table	CSV

Surveys delivered via the web:

(There currently are no surveys that have been delivered outside of the Manhattan environment.)



Notice that the survey now appears under the *Surveys delivered via Manhattan* heading, and is no longer listed under the 'Private' surveys heading. You can quickly copy the survey back to your Private list by clicking the **Copy to your Private list** link next to any survey that appears in any heading on this page.

Delivered surveys have three new options:

Bar Graphs

Displays the results of the survey as a bar graphs.

Table

Displays the raw data collected by the survey as an HTML table.

CSV File

Sends you the raw data collected by the survey in a Comma Separated Values text file. A CSV file can easily be imported into a spreadsheet or statistics program for further analysis of the data.

If you click on either of the *Survey Results* links mentioned above, you'll get a message like "No one responded to this survey." Let's fix this by having someone take the survey.

Prof. Einstein takes the survey

Login as Prof. Einstein. His username is ae4322 and his password should be *relativity*. You should now see a link to the survey on his "My Manhattan" page:

Please take the following survey(s)

[Computer Ownership Survey](#)

Albert Einstein

Red stars in this group mark classes with unread messages

Spring 2003

[PHY 101 01](#)

Introduction to Physics

Prof. Einstein

[Check for unread messages](#)

▲ **Course Template** (This group is NOT being checked for unread messages.)



[PHY 101](#)

Introduction to Physics

Prof. Einstein

Click on the "Computer Ownership Survey" link, and the survey will be displayed:

[No thanks, I don't want to take this survey](#)

Computer Ownership Survey

1. How many computers do you own?

☐ None
☐ One
☒ Two
☐ More than two.

2. How many computers do you wish you had?

☒ None
☐ One
☐ Two
☐ More than two.

[Submit](#)

Note the **No thanks, I don't want to take this survey** button at the top of the page. This button allows the user to discard the survey without answering any questions. (You can click it to see a JavaScript alert, but then click **Cancel**.) Answering the survey questions is straightforward.

If you are following the tutorial, answer the questions and click the **Submit** button at bottom of the page. When you return to your My Manhattan page, the link to the survey will be gone.

Viewing survey results

Now log back in as the super-user, and select **Surveys** menu. The **Bar Graphs**, **Table** and **CSV File** links are used to view the results of the survey:

delivered via Manhattan:

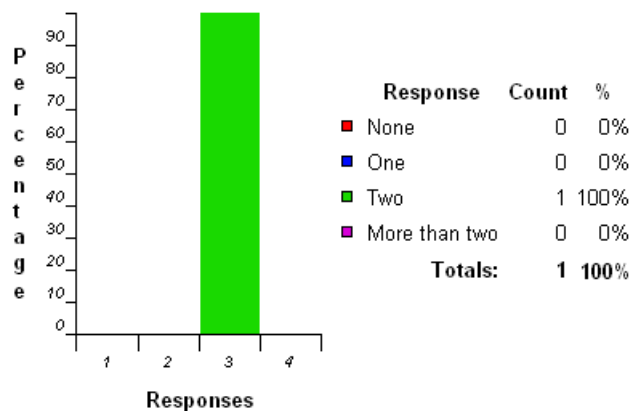
Survey Results						
Computer Ownership Survey	Preview	Copy to your Private list	Info	Delete	Bar Graphs	Table
					CSV File	

When you click on the **Bar Graphs** link, you'll get graphs showing the responses to each question:

Computer Ownership Survey

There were 1 responses. The survey was delivered to 12 people.

1. How many computers do you own?



When you click on the **Data** link, you'll get the raw data printed in an HTML table:

CLOSE WINDOW X

Number of responses:	1
Number of potential respondents:	11
Rate of Participation:	9.1%
Survey type:	Delivered via My Manhattan

[Download results as CSV File \(for spreadsheet\)](#)

	1	2
try/phy10101;	3	1

Note the link near the middle of the above screenshot, which sends you a Comma Separated Values file containing the same data found in the table. You can also download the same file by clicking on the **CSV file** link on the survey menu. CSV files allow you to import the data into a spreadsheet for further analysis.

How to learn more

This tutorial just scratched the surface of Manhattan's survey capabilities. To learn more, read *Questions and answers [124]* in this manual, and consult the *Manhattan Teacher's Reference* to learn how the Surveys module works within a Manhattan classroom.

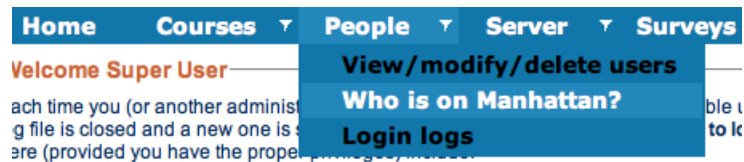
Miscellaneous commands

This section of the tour discusses a group of miscellaneous commands available on the administrators menu.

- People -> Who is on Manhattan?
- Server -> Lock Server

- Server -> Clean up -> Delete temporary files
- Server -> Clean up -> Purge post office msgs
- Server -> Usage reports -> Disk space
- Server -> Usage reports -> Course usage reports

Who is on Manhattan?



The **Who is on Manhattan?** command, found under the **People** menu, does two things:

- It gives you a list of people who are currently logged in.
- It deletes “stale keys”, which are small session files left on the server by users who are long gone, but never logged out properly.

The output of the **Who is on Manhattan?** provides two lists, one for “Normal” or centralized courses, and one for “Standalone” courses. Here's the top few lines of the centralized listing for a busy server:

Central Login Courses Tue 12/05/2006 09:32 AM
 Number of examined central keys: 678
 Number of keys in use: 613
 Number of deleted expired keys: 10
 Number of deleted expired sessions: 56

	Course(s)	Real Name	Minutes since last command
1	2 course(s)	Justin Gorski	0
2	4 course(s)	Joseph Balog	0
3	8 course(s)	Adrienne Dominie	0
4	5 course(s)	Daniel Liquori	0
5	3 course(s)	Ava Kleinmann	0
6	16 course(s)	Olalekan Adeoyin	0
7	5 course(s)	Stephanie Bellizzi	0
8	4 course(s)	Jeremy Belida	0
9	2 course(s)	Kristopher Gallant	0

The text at the top of the report says that 678 central “keys” were examined, and that 613 were in use. A “key” is a small file that's used to track a user's activity within Manhattan. When a user logs in, a new key is created. The key is deleted when the user properly exits Manhattan by clicking the **Log Out** button. Thus, the number of “keys in use” is technically the number of people who are logged into Manhattan at this moment. That would be information you can count on *if users always logged out of Manhattan properly*. In practice, many, many Manhattan users don't bother to log out. (This does no harm for those working from their own personal computers.)

Next, note the lines that read “Number of deleted expired keys” and “Number of deleted expired sessions”. This is the result of clean-up work that the **Who is on Manhattan?** command does whenever it's run. For users who haven't touched a command in 240 minutes, the keys and “session files” (small files created when they enter a particular classroom) are deleted, thus freeing up a bit of disk space. It turns out that a

user's "expired" keys and session files are automatically deleted the next time he logs in anyway, so you need not worry too much about them accumulating on the server. It's not a bad idea, however, to see "Who is on Manhattan?" whenever you login to the administrative system.

A student or a teacher can be a member of several Manhattan courses. When they log in, they are shown a "My Manhattan" page that lists all of their courses. The courses are organized by "semester" on that page, and initially the first semester listed is automatically checked for unread messages. The second column in the above screenshot simply shows the number of courses the user has thusfar checked for unread messages.

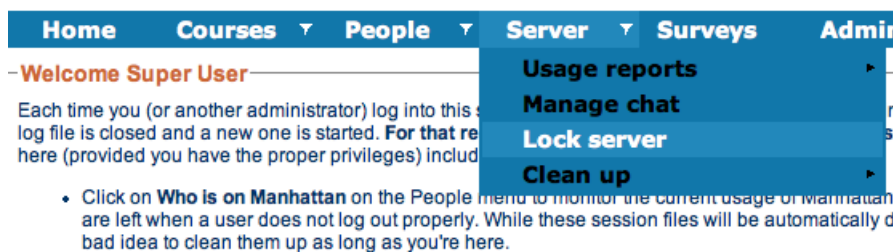
The "Minutes since last command" column shows the number of minutes that have elapsed since the user last clicked on a command button or link within Manhattan. This value gets reset to 0 minutes only when a person clicks on a button or link, such as **Reply** or **Send Message**. The timer does not get reset when a person is using their keyboard to type a message, or doing an activity like browsing through an attached website. The list is sorted with the most active users at the top - those with the lowest "Minutes since last command" appear at the top of the list. The next screenshot shows the bottom few lines of the same "Who is on Manhattan?" list:

607	4 course(s)	Sheldon Barnes	176
608	2 course(s)	Daniel Goodwin	181
609	5 course(s)	Stephanie Bates	184
610	5 course(s)	Crysta Lemon	186
611	9 course(s)	Mary Vollaro	202
612	2 course(s)	David Morrissey	235
613	1 course(s)	Peter Banach	237

Note that the last person on the list has been inactive for 237 minutes. It turns out that by default (you can change this setting in `CUSTOM.h`) users are automatically logged out after 240 minutes. You should never see a value higher than 239 minutes since last command on these lists. If we waited a few minutes, then refreshed the page, we'd find the last user is no longer on the list.

So how many people are actually logged in? The best you can do is use your own judgement based on the "minutes since last command". Certainly people with 0 or 1 or 2 minutes since their last command are actively using Manhattan, but even a person with 30 minutes since their last command could be working on an exam or reading material posted by the teacher. The moment they click a button or Manhattan program link, they'll be moved to the top of the list.

Lock Server



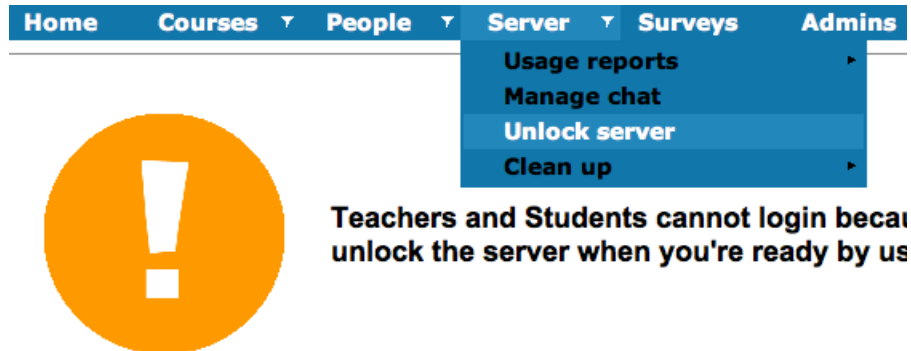
The purpose of the **Lock server** command is to allow you, as system administrator, to perform maintenance on either the physical server or perhaps on the Manhattan software (e.g. to upgrade to a new version). When you select **Lock server** from the **Server** menu (and after responding OK to an "Are you sure?" prompt):

- Teachers and students will be prevented from logging in. When they do enter a correct username/password, they'll get a message stating that the server has been locked for maintenance.

- Users that are already logged in can continue to work, however they'll be logged out automatically (with a message stating that the server has been locked for maintenance) the next time they visit the "My Manhattan" page, or the Main Menu for any one of their classes.

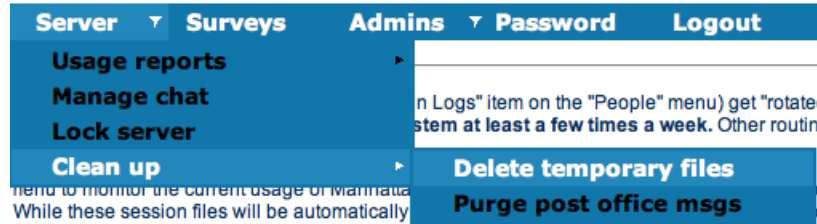
How do you know when everyone is really off the server? You should repeatedly use *Who is on Manhattan?* [73] to see who is still actively using Manhattan. As discussed in that section, it can be difficult to know with 100% certainty whether or not everyone is off. You'll have to use your judgement.

When the server is locked, the command on the **Server** menu changes to **Unlock server** and a prominent notice appears on the "Home" page of the administrative system to remind you that no one can log in to their accounts:



Selecting **Unlock server** from the **Server** menu returns the installation to normal so users can log in again.

Delete temporary files



There are three ways temporary files might be left on the server:

- A user clicked their **Stop** button, or otherwise caused their browser to stop sending data to the server in the middle of posting a new Manhattan message.
- A user was viewing an "Attached website" within a Manhattan classroom, and never returned to the classroom.
- A teacher copied a Manhattan message to Manhattan's internal clipboard, so they could reuse it in another module or classroom.

When a Manhattan message is sent, the raw data is first stored in the `tmp` directory of your Manhattan installation. The name of the file starts with `CGI_DATA_`. Once this data is parsed and stored as a true Manhattan message, this temporary file is deleted. Unfortunately it often happens that users somehow stop the process in the middle of sending a message. Most frequently, they are trying to send a large attachment and have underestimated how long it should take using their slow Internet connection. Whatever the reason, in rare circumstances an aborted send leaves the `CGI_DATA_*` file in the `tmp` directory within your Manhattan installation. The **Delete temporary files** command deletes any `CGI_DATA_*` files in the `tmp` directory that's more than 240 minutes (4 hours) old.



The `tmp` directory I'm referring to is within your Manhattan installation. It is not the same as the `/tmp` directory that's part of the standard Linux/Unix filesystem. Manhattan never writes any files outside of its own installation directory.

A second type of temporary file deleted by the **Delete temporary files** command are symbolic links found in the `images/links` directory of your Manhattan installation. Whenever a user views an "Attached website" (see the *Teacher's Reference* for details) within a Manhattan classroom, a symbolic link is created for that user within the `image/links` directory of your installation. If that user never returns to the classroom from that attached website, the symbolic link remains. While the **Delete temporary files** command also deletes any stale symbolic links it finds, it so happens that these symbolic links are also somewhat self-cleaning. The next time the user logs in to Manhattan, any stale symbolic links he left on the server during his last visit are automatically removed.

Another opportunity to reclaim disk space involves Manhattan's "clipboard" feature. As described in the *Teacher's Reference*, teachers can easily repost any Manhattan message, including its attachments. They do this by clicking on a **Copy to clipboard** button while reading the message they want to reuse. Internally, at that moment Manhattan places a copy of the message and all of its attachments in a `clipboards` folder within your Manhattan installation. When the same teacher clicks a **Paste from clipboard** button, the message data is read from the `clipboards` directory and posted as a new Manhattan message.

In the original design, a message copied to the clipboard remained on the server until it was replaced by another **Copy to clipboard** action taken by the same teacher. (Teachers can delete the content of their clipboard using the *Clipboard Viewer* [121] on their **Configuration** menu, but in practice this option is rarely used.). This seemed reasonable, until we noticed that with nearly 150 teachers on the system, and with some of the messages containing 100+ Megabyte attachments, it might be prudent to do some clean-up of unused clipboard content. For that reason, the **Delete temporary files** now automatically deletes any clipboard content that's over 3 days old.

Delete abandoned temporary files**Files deleted from Manhattan's tmp directory**

The 'tmp' directory of your Manhattan installation is used to temporarily store files as they are being uploaded by automatically by Manhattan, but under certain conditions unneeded files in the 'tmp' directory can get left on the directory (not your system-wide /tmp directory!) that is more than four hours old.

Mon 12/04/2006 11:36 AM

Number of deleted files: 0

Total size of deleted files: 0

Symbolic links deleted from Manhattan's images/links directory

Each time a user views an attached website that was included with a message, a special file called a symbolic link is created. These symbolic links are automatically deleted when the user leaves the message containing the attached website to Manhattan after viewing an attached website, their symbolic link gets left in the images/links directory. This utility deletes any symbolic link that is more than four hours old.

Mon 12/04/2006 11:36 AM

Number of deleted files: 0

Directories deleted from Manhattan's clipboards directory

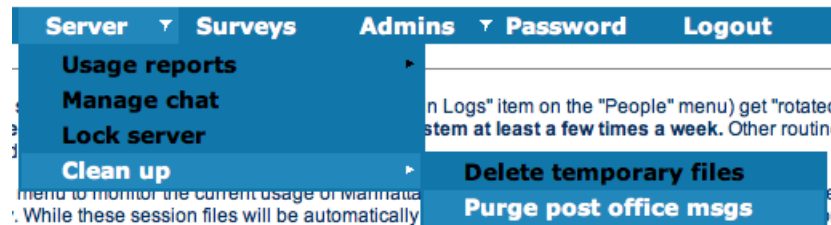
When a teacher clicks on a 'Copy this message to my Clipboard' button to use Manhattan's Copy/Paste feature, a directory called 'clipboards' on the server. That user can then use the 'Paste Message from Clipboard' button to classroom. A teacher can delete the contents of their clipboard (using an item on the class' Configuration menu) this utility deletes any clipboard directory that is more than 3 DAYS old.

Mon 12/04/2006 11:36 AM

Clipboards examined: 26

Clipboards deleted: 0

Purge Post Office Messages



The purpose of the **Purge Post Office messages** routine is to reclaim disk space on the server by permanently deleting Post Office messages sent by students to students. The Post Office allows users to send each other private messages in a manner similar to ordinary Internet email. Students are free to send each other messages, which may include large file attachments, that their teacher will never see. This process should only be run:

- If you are concerned that private student-to-student Post Office messages are consuming too much disk space on your server.
- On classrooms that aren't used any more, especially by students, usually because the course is long over.
- On classrooms where students have already been denied access using "Change course access" (see *Changing a course's access rights* [41]) on the **Courses** menu.

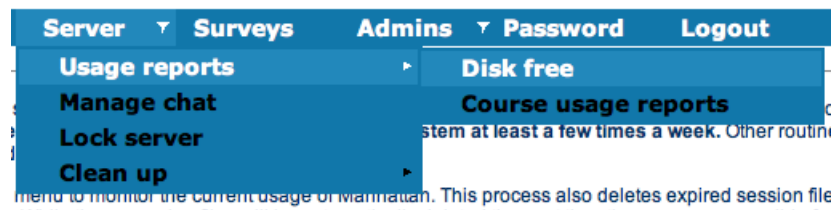
There are several steps to the procedure, and you'll be asked for your password to complete the process. The steps should be familiar since they follow the same pattern as elsewhere in the administrator's system. Once you commit to the process after entering your administrator's password, ALL Post Office messages sent BY students TO students in the selected classroom(s) will be PERMANENTLY deleted. Note this affects only the Post Office module. Messages posted by students to other Manhattan modules are completely unaffected. Also note the only messages that are deleted are messages sent by students to other students. Post Office messages sent by a student to a teacher, from a teacher to a student, or from a teacher to another teacher will NOT be deleted. In general, a teacher's view of the classroom will be completely unchanged by this process. A student's view of the classroom will be changed in that any Post Office message they have sent to other students will be deleted from their Post Office Outbox. A student's Post Office Inbox will contain only messages sent to them by the teacher(s) of the class.

The sole purpose of this procedure is to reclaim disk space on the server by deleting student-to-student private Post Office messages, that probably would never be seen again because you have closed students out of those classrooms anyway.

No log records are kept on if/when a classroom has had its Post Office purged. You might want to run this from a computer that has a printer so you can print the final page for your records.

This routine does not report back on how many messages were deleted, or the total disk space reclaimed. If you want a rough idea of disk space reclaimed, you can use *Disk free* [78] before and after running the procedure.

Disk free



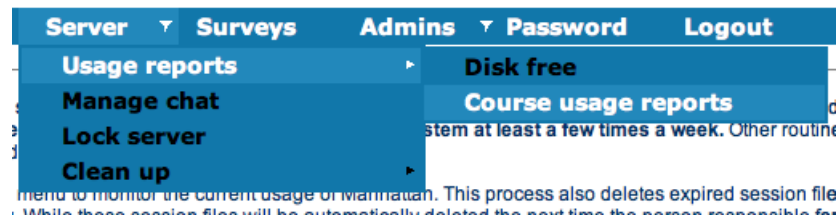
The **Disk free** command simply shows you the output of the standard Linux/Unix `df` ("disk free") command. Example output might be:

The 'disk free'(df) command on the server yields the following report.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	13G	3.9G	8.2G	32%	/
tmpfs	1014M	0	1014M	0%	/dev/shm
/dev/sdb6	918G	34G	884G	4%	/X
/dev/sda1	54M	17M	34M	34%	/boot
/dev/sda6	13G	39M	12G	1%	/usr/local
/dev/sdc5	1.2T	189G	981G	17%	/udd

The point, of course, is to allow you to see how much free disk space you have on your server. You will need to know something about your server's disks and how they are mounted to make use of this information. The above screenshot shows six filesystems. Only the system administrator knows which one is home for this particular installation of Manhattan.

Course Usage Reports



The **Course Usage Reports** give the administrator an idea of how Manhattan is being used at your school. The reports provide statistics on how much disk space is being used, how many messages were posted, which modules teachers are using, and so on.

When you select **Course Usage Reports** from the menu, you'll get a list of all of the course groups on your server:

Group: wks (for Manhat Workshops)	(Report Not Available)	Create a New Report
Group: Cert (Certificate Program - DCE)	(Report Not Available)	Create a New Report
Group: c104 (D&E Summer '04)	Report Created: Wed 05/03/2006 01:49 PM	Create a New Report
Group: c105 (D&E Summer '05)	Report Created: Tue 05/09/2006 09:05 PM	Create a New Report
Group: c106 (D&E Summer '06)	(Report Not Available)	Create a New Report
Group: c203 (D&E Fall '03)	Report Created: Mon 05/08/2006 11:27 AM	Create a New Report
Group: c204 (D&E Fall '04)	Report Created: Tue 05/09/2006 09:16 PM	Create a New Report
Group: c205 (D&E Fall 2005)	Report Created: Wed 05/03/2006 02:07 PM	Create a New Report

Because creation of a course usage report is a resource intensive process, the reports are not done "on the fly". Instead, once a report is generated, the results are stored on the server, where they can be viewed again. In practice, you may only need to generate a Course Usage Report once for each course group - that might be done a few weeks after the courses in that group are over.

In the above screenshot, you can see for example that the groups c104 and c105 have already had usage reports created. If you click on the **Create a New Report** link for either of these course groups, the new report will completely replace the existing report. The groups wks, Cert, and c106 have not yet had a report created for them.

Here's the top of a typical report:

Course Group:	c405 (D&E Spring 05-06)
Analysis Timestamp:	Thu 06/15/2006 10:27 AM
# Classrooms	461

Disk Usage Summary - bytes	
Total disk usage for this group:	32,699,650,112
Average per classroom:	70,931,996
Median for all classes:	9,235,718

Group Disk Usage by File Type - bytes (% of total)	
Typed-in messages:	75,037,310 (0.23%)
Attached files:	31,176,885,476 (95.34%)
Attached websites:	1,234,627,863 (3.78%)
Other (system overhead):	213,099,463 (0.65%)

How many ___ are there? - count --- average --- median	
Typed-in messages:	154307 --- 334.72 --- 188.00
Attached files:	70470 --- 152.86 --- 54.00
Attached websites:	8898 --- 19.30 --- 0.00
Chat messages:	60744 --- 131.77 --- 0.00
Student accounts:	8467 --- 18.37 --- 18.00

Module Usage - # classrooms w/module enabled (% of total classrooms)	
Assignments:	384 (83.3%)
Handouts/Notices:	422 (91.5%)
Class Discussion:	245 (53.1%)
Post Office:	407 (88.3%)

In addition to summary information, such as that shown above, the report includes data on every course in that course group.

On your own

If you're working through this chapter as a tutorial, take the time to experiment with these miscellaneous commands. In particular, it's useful to see how a "locked" server looks like to an end-user.

- Lock the server, then try to login as Prof. Einstein.
- Unlock the server, then login as Prof. Einstein and enter his Introduction to Physics course. Start composing a message in any module, but don't hit the **Send Message** button. From another browser window or computer, use the administrative system to lock the server. Switch back to Prof. Einstein's view and click the **Send Message** button. What happens? Did the message get sent?
- Extra credit. The online information within Manhattan explains that creating (or deleting) a file named `SYSTEM_LOCKED` within your installation's `tmp` directory has the same effect as using **Lock server** or **Unlock server** from within the web based administration system. Verify that this is true.

Delete courses

All courses eventually come to an end. The **Delete courses** link on the administrator's menu is used to permanently delete courses from your Manhattan installation:



You'll probably find that many teachers (and some students) won't want you to immediately delete courses once they are completed. There are a number of ways Manhattan can help you deal with an ever growing number of completed courses:

- You can use the features described in *Changing a course's access rights [41]* to mark the completed courses as “no access” or perhaps “read-only”. When a course is marked “no access” for students and optionally teachers, the course simply does not appear on their login page. When marked as “read-only”, users can view the materials but can not add any new messages.
- Judicious organization of courses into “groups” can help you as an administrator to know which courses have been completed. Since a course group corresponds to a subdirectory (under the `COURSES` directory) within your installation, it should be possible to move all of the courses in that group to a separate disk or partition on your server. If those courses are then marked as “read-only” it will then be possible to make one final archive backup of those courses, while they remain live on the server.
- You should create “course templates” for your teachers (see *Create a Course Template for Prof. Einstein [31]*), and encourage them to move (using Manhattan's “copy and paste” feature), all learning materials they plan to re-use into the appropriate template. You should never be asked to keep a completed course on the server only because it contains materials like the syllabus or lecture notes a teacher would like to re-use.

The remainder of this section demonstrates how to permanently delete the courses you've created as part of this tutorial. After deleting the courses, you'll delete all of the users you've created, so your Manhattan installation will be back to where it was when you started.

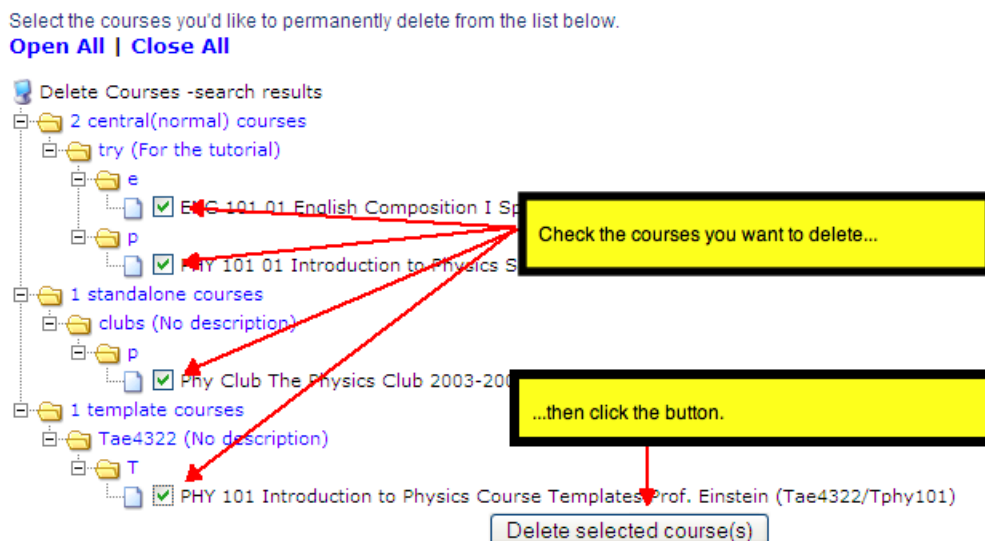
Step 1 - Get a list of courses to choose from

After clicking on the **Delete courses** link from the administrator's menu, you'll see a search form. This form behaves the same way as the one used when you **Configure courses**. It essentially asks “Which courses do you want to delete?”. Since we want to delete all of the courses created during this tutorial, hold your **Ctrl** key down as you click on the three available course groups: Templates, try, and clubs. Leave the search box empty to signify you want all of the courses in those groups, then click the **Search** button:

Delete Courses - search form

Step 2 - Select courses from the list

After selecting all of the course groups on the preceding form, you'll be presented with a list of all courses on the server. If you've been following the tutorial, your list will include the courses shown in the screenshot below:



As shown above, check the box next to each course on the list, then click the **Delete selected course(s)** button at the bottom of the page. You should get a JavaScript prompt asking you to confirm your decision. Click **OK**.

Step 3 - Confirm your decision

Permanently deleting a Manhattan course is a serious decision. In this step you are again shown a list of the courses you are about to delete, along with a form that requires you to enter your current administrator's password:

Delete Courses

Final step: Confirm your decision to permanently delete the selected courses.

Internal Course ID	Title/Semester/Teacher
try/engl10101	ENG 101 01 English Composition I Spring 2003 Prof. Jenkins
try/phy10101	PHY 101 01 Introduction to Physics Spring 2003 Prof. Einstein
clubs/physics	Phy Club The Physics Club 2003-2004 Albert Einstein
Tae4322/Tphy101	PHY 101 Introduction to Physics Course Templates Prof. Einstein

Password:

Permanently Delete the Listed Course(s)

Enter your administrator's password and click the button. Answer **OK** to one last “Are you sure?” prompt and the courses will be permanently deleted from the server. You'll get a listing of the deleted courses and the people who were in them.

Cleaning up

If you've been following along, you've just deleted all of the courses from your Manhattan installation. Even though there are no courses on the server, you'll find:

- The “try” and “clubs” course groups you created are still there.
- All of the students and teachers still have their accounts!

In this section you'll learn how to delete empty course groups, and completely remove users who are no longer associated with a Manhattan course.

Deleting the empty course groups

Deleting an empty course group is done the same way you change its description. When a course group is empty, the form used to change a group's description has an additional “delete” button.

First select Courses → Configure courses from the administrator's menu. You'll get the form used to search for courses. Hold down your **Ctrl** key as you click on both the `try` and `clubs` groups, then click the search **Search** button. Although the search should come up empty, you'll get a summary that looks like this:

Configure Course -search results

Search string	
within course group(s):	
	try (For the tutorial.)
	clubs (Click Here to describe this group.)
	: Central and standalone courses
	: Contain
	: 0

Click on the **For the tutorial** link as if you were going to change its description.

Click on the “For the tutorial” link to get to the form used to change a group's description and/or add notes. You'll find an additional button that allows you to permanently remove the empty course group:

Delete this empty course group try

Description:

Notes:

This course group is just for practice and will be deleted at the end of the tutorial.

[Delete this empty course group](#)

This button appears when the course group you are describing is empty. Click on it to delete the empty course group.

Go ahead and click the button to delete the empty `try` course group created for this tutorial.

On your own: Repeat the procedure just described to delete the `clubs` group.

Removing users without classes

By design, a student or teacher continues to have a valid Manhattan account even if they no longer belong to any classes. This could happen because the administrator has completely deleted their classes, or they may have been individually removed from all of their classes by their teachers or the administrator. When a person without any classes logs in, they'll simply see an empty list of courses. The only options they will have are to change their password, edit/view their personal info page, and logout.

It's easy for the administrator to find and optionally delete users without classes. First, select **View/modify/delete users** link from the **People** menu. You'll see the search form you've experimented with before in this tutorial:

Search for (leave blank to select all):

Search data fields: ☒ All fields ☐ Username ☐ Id Number ☐ Real Name

Include: ☐ All users ☐ Only students ☐ Only teachers ☒ Users without classes

Report type: ☒ Quick list ☐ Full records

Search is case sensitive: ☒ No ☐ Yes

Sort by: ☒ Last names ☐ Usernames ☐ Id numbers ☐ First names

Matches must __ search string: ☒ Contain ☐ Start with ☐ End with

[List users without classes for quick deletion](#)

First note the search option “Users without classes” underlined in red in the above image. You can always select that option when doing any user search to include only people who are “classless”. The quickest route to deleting these users, however, is the link circled in red in the above screenshot, labeled “List users without classes for quick deletion.”. Click that link and you'll get a list like this:

Username	Personal info?	Name	Id	Select
nb3888	Yes	Neils Bohr	X343888	☒
mc4478	Yes	Marie Curie	X394478	☒
te4096	Yes	Thomas Edison	X394096	☒
ae4322	Yes	Albert Einstein	X34322	☒
rf5734	No	Robert Frost	X485734	☒
rf7389	No	Robert Fuller	X347389	☒
jj5753	No	Janet Jackson	X485753	☒
jj4321	No	Julius Jenkins	X34321	☒
mj4032	No	Michael Jordan	X394032	☒
in7332	No	Isaac Newton	X347332	☒
cs4452	No	Cindy Slattery	X394452	☒
ss4321	No	Steven Stanton	X334321	☒
rt4377	No	Richard Taylor	X334377	☒

If you're following the tutorial, select all of the users (you can click the **Check All** button) and then click the **Delete User(s)** button. All of the selected user accounts will be deleted, and you'll get a summary page listing the deleted accounts.

Chapter 5. Creating Courses - details

In *A tutorial for administrators [22]*, you learned how to create normal, standalone, and template courses on your Manhattan server. This chapter is to provide more details about the format of the roster files and the course creation procedure.

ID numbers, usernames and passwords

A Manhattan user (i.e a student or a teacher) is uniquely identified by TWO data items: an *identification number* (or “ID number”) and a *username*.

Most schools have information systems that already identify students by some form of ID number. It's easiest to use that same number for their Manhattan accounts. Within Manhattan, ID numbers must meet the following conditions:

- They must be between 4 and 15 characters long.
- They must contain at least one digit 0-9.
- They must be unique. No two people can have the same ID number.

Teachers must also have an ID number that meets the same conditions as above. If your school doesn't already assign ID numbers to teachers, you can make them up. If you do need to invent teacher IDs, a suggestion is to include a particular letter, say a 'T', in a teacher's ID number to make it easy to identify teachers.

Students and teachers login to Manhattan by entering their *username* and a *password* into a web form. Manhattan automatically derives a person's username by the following recipe:

1. take the first letter of the first name
2. add the first letter of the last name
3. add the LAST FOUR CHARACTERS of the persons ID number
4. convert the entire username to lowercase letters

Examples:

Julie Jenkins has the student ID: D838993
Her username is: jj8993

Albert Einstein has the teacher ID: 839003
His username is: ae9003



A configuration item named `LAST_N_DIGITS_OF_ID` in the `custom.h` file can be modified at compile time to use fewer or more than 4 characters from the ID number when Manhattan derives a username. If you choose to change this value, you should do so before you start adding users to your server, so the usernames are consistent.



If you use the XML-based method for creating classrooms, described in *Creating Courses with XML [92]*, it is possible to use whatever username is stored within your student information system within Manhattan.



Just one more tip. If you know the C programming language, you can also modify the functions `derive_username()` and `derive_another_username()` in `src/manhattan/shared_authenticate.c`. These two functions determine how a person's username is derived, and how another username is derived when the usual username is already taken.

For most ID numbering systems, it is unlikely for there to be two users with the same initials and the same last four digits of their ID. However, conflicts can happen. Conflicts are resolved by taking additional letters from the last name.

Example:

```
Johnny Jones has the student ID: D788993
```

```
Manhattan detects that Julie Jenkins already has  
the username jj8993, so it takes an additional  
letter from Johnny's last name.
```

```
Johnny's username becomes jjo8893
```

A person's password is initially the same as their username. The first time they login, they'll be forced to select another password before they can continue to their course work.

More about the roster text file

As you know from the tutorial to create a new course, you need to start by creating a text file in a certain format. You can name the file anything you want, but be sure it's a plain (ASCII) text file. For Windows users, Notepad will work, for example. Here again is the sample roster text file we used in the tutorial:

```
PHY 101 01  
Introduction to Physics  
Spring 2003  
Prof. Einstein  
X34322  Albert Einstein  
X343888  Neils Bohr  
X347332  Isaac Newton  
X394032  Michael Jordan  
X334321  Steven Stanton  
X394452  Cindy Slattery  
X485734  Robert Frost
```

Line 1 of the roster file designates the `course code section` for the course. This information is displayed on various screens in Manhattan to remind the user what course they are working in. Beyond that, the string you type in the first line has no meaning - we could have typed "Bananas are yummy!" here. The maximum length is 20 characters. Both you and the teacher of the course can change whatever you type here at any point in the course. (The option is under the teacher's Configuration menu - see *Basic classroom configuration [109]*)

Line 2 of the roster text file is the `title` of the course. Like the `course code section` data, this information is shown to users while they work within the classroom to remind them of which course they are working with. Its maximum length is 40 characters. The course title can also be changed later by the teacher or system administrator using the course's Configuration menu.

Line 3 is the `semester`. This is another piece of information that is displayed on screens for no other purpose than to let the users know which course they are working with. The string "Ice Cream Sundae" is

just as valid as “Spring 2003”. However, you’ll find that when a person logs into Manhattan, their courses will be organized by the values of the Semester field. Be consistent! The ‘semester’ field can be changed by teacher or administrator at any time, again via the **Basic classroom configuration** option on the course’s Configuration menu.

Lines 4 and 5 both describe the primary teacher of the course. On line 4, list the name of the teacher as s/he will be referred to within this particular classroom. In our example Albert Einstein wants his name to appear as “Prof. Einstein” within this Physics 101 course. If he wanted to be “Mr. Einstein” you would type that on line 4. If he wanted to appear as “Albert Einstein” or “Bozo the Clown”, then that’s what you would type. Note that there’s nothing permanent about what is entered here.

On *line 5* type the ID number of the teacher followed by one or more spaces or tabs, followed by the teacher’s first name, followed by spaces or tabs, followed by the teacher’s last name. Here it *IS* important that you use the teacher’s correct, full name (with no middle initials), since Manhattan will use the full name to derive the teacher’s username. If you had typed “Prof. Einstein” instead of “Albert Einstein” on line 5, the teacher would wind up with the username pe4322.



When you create a new course using this method you can only specify one teacher. However after a course is created, additional teachers can be added to a course by entering the course’s configuration menu, by selecting **Configure courses** from the system administrator’s main menu - see *Add a teacher* [117]

The *remaining lines* of the file list the students in the class the same way the teacher’s information is listed on line 5.

Note that each line representing the teacher or a student is of the form:

```
ID<spaces/tabs>FirstName<spaces/tabs>LastName
```

There’s no accommodation for middle initials, and everything after the FirstName is treated as the LastName. For example, given the lines:

```
23423423      John Q. Public
93834383      Mary Van Smith
43423433      Martin Casey Jr.
```

John’s LastName is “Q. Public”, Mary’s is “Van Smith”, and Martin’s is “Casey Jr.” The last name used to derive the person’s username as described above, so John’s username would be: jq3423, which is probably not what you want. Also, lists of names displayed on various pages within Manhattan are sorted by the last name, so John’s name will be found under the Q’s.



Do yourself a favor and strip middle initials and suffixes like “Jr.” and “III” from the roster text file before you create a new course. The “Van” in Mary’s name in the above example, however is perfectly OK since it is truly part of her last name.

Choosing a group name

All courses on the server belong to a *group*. Internally, a group is a directory on the server (under the courses directory) where all courses within that group are stored.

```
manhat-x.x.x/courses <-- groups and courses are stored in the
                        'courses' directory
```

```
f03 <-- a 'group' is a subdirectory under 'courses'
```

```

engl10101
math13103 <--the f03 group has 3 courses -
hist35101      the 'internal course names' are also
                directory names

s04  <-- another course group
engl10102
cis40021
    
```

When creating a new course, you can either select an existing group from the list, or select <New Group> and type the name of the new group in the text box.

Group: <New Group> f03

Internal Course Name: psy32103

Roster file: Browse...

Course type: Normal

Create this course

When creating a new group:

- Use as few characters as possible. For example, f03 is a better group name than Fall_2003. Once a group is created, you can give it a more complete description using the **Configure courses** option on the administrative menu. (See *Changing a group's description* [37])
- Keep in mind that you are selecting the name for a directory on the server. No spaces, no dots, no slashes - just a-z, A-Z, 0-9
- Think about how you want to organize your classrooms. Other administrative programs, such as **Delete courses** and **Configure courses** allow you to search for courses within a group. A separate group for each semester will make sense for most schools. Also, consider a group called demo for classrooms used to demonstrate Manhattan, or maybe one called junk for courses that you know can be deleted without causing any problems.
- If you are creating a Course Template, your choice of group on this form is completely ignored. Instead, Manhattan will automatically use the group name T<username> where <username> is the username of the teacher who owns the template. (Course templates are created in the templates directory of your installation, instead of the courses directory.)

```

manhat-x.x.x/templates <-- course templates are stored in the
                        'templates' directory

Tae4322  <-- each teacher has their own 'group'.
                The group name is 'T'
                followed by their username

Tengl101
Tmath131 <--this teacher has 2 course templates.
                A 'T' is automatically
                added to the front of the
                "internal course name" you provide.
    
```

Choosing an “Internal Course Name”

Again, all of the “normal” and “standalone” courses on a Manhattan server are stored within sub-directories of the distribution's `COURSES` directory.

The internal course name is important. Choose a name that will let you, the server administrator, know exactly which course is stored in that directory. The best approach is to use the same `course-code-section` information that's used to identify the course within your school's information system. At my college, I know that the course “phy10101” within group “f03” is Physics 101 section 01 for the Fall 2003 semester, for example.

The programs that allow system administrators to work with courses allow you to search for courses by their group and internal course name

As for the group name, try to keep the internal course names short, say under 20 characters. Since you are specifying the name of a directory on the server, use only letters of the alphabet, the digits 0-9, and underscores (or dashes if you prefer) in the names. DON'T use spaces!



When you create a Course template, Manhattan will automatically insert an uppercase 'T' in front of the internal course name you specify, if one is not already there, as mentioned in the previous section for a normal or standalone course must NOT start with a 'T' - Manhattan will warn you if you make this mistake.

Name Mismatch

As we've said, the ID numbers of users on Manhattan must be unique. That is, no two people can have the same ID number. But suppose there's already a user with the following information:

```
ID number:  X34322
Username:   ae4322
Name:      Albert Einstein
```

Albert Einstein has his account because at one time in the past, you added him to a course either as a teacher or a student. Now you're about to create a new course and have prepared the following roster text file:

```
BSK 101 01
Intro to Basket Weaving
Fall 2003
Prof. Johnson
X34322 Michael Johnson <--- has Albert Einstein's ID number
123432 Johnny Student
234232 Laurie Learner
```

The problem is that a person in the roster file, Michael Johnson, has the same ID number as Albert Einstein. (Somewhere you've got a problem with your data!) When you create a course using the above file, you'll get these results:

Group	try
Internal Course Name	bsk10101
Course Code	BSK 101 01
Course Title	Intro to Basket Weaving
Semester	Fall 2003
Course Type	Normal
Status	New Course
Success?	Yes

BSK 101 01

Full Name	ID	Username	Central Account	User Type	Course Status	Central Status
Albert Einstein	X34322	ae4322	Name Mismatch	faculty	Add Successful	Success
Johnny Student	123432	js3432	Existing User	student	Add Successful	Success
Laurie Learner	234232	ll4232	Existing User	student	Add Successful	Success
Total Users: 3						

The “Name Mismatch” warning in the red-colored row warns you that the ID number supplied in your roster file belongs to an existing user, and that user's full name according to Manhattan's records *doesn't exactly match* the name found in your roster file. The course was created, but it is taught by the existing user with username ae4322, Albert Einstein, and not by Michael Johnson.

In the example we used, there clearly is a cause for concern, since your roster file says that Michael Johnson has the ID number X34322, and not Albert Einstein. However note that a “Name Mismatch” can also be more benign. Suppose for example the teacher was listed as “Al Einstein” in your roster file. In this case, the “Name Mismatch” is simply alerting you to the fact that “Al Einstein” and “Albert Einstein” *appear* to be different names, although in reality they are most certainly the same person.

Chapter 6. Creating Courses with XML

The Alternative XML format

The format of the roster text file described so far in this manual has been used within Manhattan since 1997. Its origins trace back to an early terminal-based student information system used at Western New England College. In that system, we were able to call up a course roster that would be displayed on the terminal in pretty much the same format as the roster file now supported by Manhattan. A quick “copy and paste” dropped the data into a text file that became input to Manhattan's create-a-course programs. The format continues to be supported, and is a reasonable choice because it's easy to type by hand, or to generate from a student information database.

Starting with version 2.1, Manhattan also supports a more flexible XML based file format for the creation of courses. While this text file format is difficult to type by hand, a capable programmer can write a script to generate the file from a database query. If you're able to find someone to write such a script, the XML format has a number of advantages:

- A single XML file can be used to create many courses at a time.
- Courses can be created with more than one teacher. While any Manhattan course can have more than one teacher, a limitation of the “legacy” file format is that you can only *create* the course with a single teacher. You (or that first teacher) must then add the additional teachers using an the “Add a teacher” option on the course's “Configuration menu.”
- If your student information database contains a username for each student and teacher, you can use that username instead of Manhattan's automatically generated usernames.

What is XML? That's a long story, well beyond the scope of this manual. Suffice it to say that an *XML* (eXtensible Markup Language) file is a text file marked up with tags enclosed in angle braces like this: `<tag>text between the tag</tag>`. (XML has some superficial similarities to HTML, the markup language used to create web pages.) One use of XML is to provide a way to move data from otherwise incompatible information systems, for example possibly your school's student information system and Manhattan. Manhattan understands a certain set of tags that must appear in a certain order, which we'll explain by showing you an example.

The XML format by example

Suppose, using the “legacy” roster file format, you're about to create a course on your server. You intend to have Manhattan store the course in:

```
Course Group: s03
Internal Course Name: phy10101
```

Both of the above data items are specified on the form you normally use to create courses within the administrative system - see *Creating normal courses* [24]. Here's the “legacy” roster text file you might normally prepare and attach to that same form:

```
PHY 101 01
Introduction to Physics
Spring 2003
Prof. Einstein
X34322  Albert Einstein
```

```
X343888 Neils Bohr
X347332 Isaac Newton
X394032 Michael Jordan
```

The same course can be described by the following XML file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>
  <course subdir="s03" id="phy10101">
    <course_no>PHY 101 01</course_no>
    <course_title>Introduction to Physics</course_title>
    <term>Spring 2003</term>
    <teacher_title>Prof. Einstein</teacher_title>
    <users>
      <user id="X34322">
        <first>Albert</first>
        <last>Einstein</last>
        <username></username>
        <group>faculty</group>
      </user>
      <user id="X343888">
        <first>Neils</first>
        <last>Bohr</last>
        <username></username>
        <group>student</group>
      </user>
      <user id="X347332">
        <first>Isaac</first>
        <last>Newton</last>
        <username></username>
        <group>student</group>
      </user>
      <user id="X394032">
        <first>Michael</first>
        <last>Jordan</last>
        <username></username>
        <group>student</group>
      </user>
    </users>
  </course>
</courses>
```



Obviously, that is not something a human would type. Using the XML format makes sense only if you can write a program to query your student information system and create the XML file.

Let's begin our analysis of the format with the first line:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

The first line of the file identifies it as an XML file and it *must be present exactly as it is displayed above*. The rest of the file describes the courses you are about to create. The entire body of the file is enclosed between a <courses> and a </courses> closing tag:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>

(XML describing the courses)

</courses>
```

Theoretically, an unlimited number of courses can be described between the required `<courses>` and closing `</courses>` tags. Sandwiched between those tags, a single course's description is enclosed between a `<course>` and closing `</course>` pair of tags. (Notice this is singular - the “courses” tags enclose, or contain, “course” tags.) Note in the sample below that the opening `<course>` tag has two required *attributes*. The `subdir=` attribute is used to specify the course group under which this course will reside - see *Choosing a group name [88]*. The `id=` attribute is used to specify the internal course name, which is the name of the directory where this course will live on your server - see *Choosing an Internal Course Name [90]*.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>
  <course subdir="COURSE_GROUP" id="INTERNAL_COURSE_NAME">

    (xml describing this particular course)

  </course>

  <course subdir="COURSE_GROUP" id="INTERNAL_COURSE_NAME">

    (xml describing this particular course)

  </course>

</courses>
```

Recall that the first four lines of the “legacy” roster text file described the course's course-code-section, its title, the semester, and the name of the teacher as she prefers to be known within the classroom. The remaining lines described the teacher and students in the class. (See *More about the roster text file [87]*.) This same information is recorded for each course in the XML version:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>
  <course subdir="COURSE_GROUP" id="INTERNAL_COURSE_NAME">
    <course_no>COURSE-CODE-SECTION</course_no>
    <course_title>COURSE_TITLE</course_title>
    <term>SEMESTER</term>
    <teacher_title>TEACHER_NAME</teacher_title>
    <users>

      (xml representing the teachers and students)

    </users>

  </course>
```

```
<course subdir="COURSE_GROUP" id="INTERNAL_COURSE_NAME">
    (xml describing this particular course - detail omitted for brevity)
</course>
</courses>
```

A list of students and teachers goes between the opening `<users>` and closing `</users>` tag. The following data is recorded for each user:

- Their ID number.
- Their first and last names.
- Whether they are a student or a teacher.
- Their username, if known.

Comparing this to the “legacy” roster file format, you’ll notice that in the XML format each user must be identified as either a student or a teacher. In the legacy format, the first person listed was assumed to be the teacher, and there can only be one teacher in the classroom.

Another difference is that the XML format allows you to specify a username for each person. This is of use to you only if your student information system has a username for each student/teacher at your school, or if the script that generates the XML can otherwise obtain a username for each person from another source. If a username is specified, it will be used as the person’s Manhattan username. If it is blank, Manhattan will derive the person’s username as it normally does - see *ID numbers, usernames and passwords* [86]

Here’s the XML framework again, with the real data we started with added to the data for the first course. Notice that:

- The data for each user is enclosed between an opening `<user>` and a closing `</user>` tag.
- A user’s ID number is specified as an `id=` attribute to the opening `<user>` tag.
- We included the `<username></username>` pair but with no content between the tags, since we want Manhattan to use its normal method of automatically deriving usernames
- The `<group></group>` tags are used to specify whether a user is a teacher (expressed with the value `faculty`) or a student (expressed with the value `student`).

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>
  <course subdir="s03" id="phy10101">
    <course_no>PHY 101 01</course_no>
    <course_title>Introduction to Physics</course_title>
    <term>Spring 2003</term>
    <teacher_title>Prof. Einstein</teacher_title>
    <users>
      <user id="X34322">
        <first>Albert</first>
        <last>Einstein</last>
        <username></username>
        <group>faculty</group>
```

```
</user>
<user id="X343888">
  <first>Neils</first>
  <last>Bohr</last>
  <username></username>
  <group>student</group>
</user>
<user id="X347332">
  <first>Isaac</first>
  <last>Newton</last>
  <username></username>
  <group>student</group>
</user>
<user id="X394032">
  <first>Michael</first>
  <last>Jordan</last>
  <username></username>
  <group>student</group>
</user>
</users>
</course>
<course subdir="s03" id="eng10101">
  <course_no>ENG 101 01</course_no>
  <course_title>English Composition I</course_title>
  <term>Spring 2003</term>
  <teacher_title>Prof. Fuller</teacher_title>
  <users>
    <user id="X349933">
      <first>Janet</first>
      <last>Fuller</last>
      <username></username>
      <group>faculty</group>
    </user>
    <user id="X348756">
      <first>John</first>
      <last>Narmontas</last>
      <username></username>
      <group>student</group>
    </user>
  </users>
</course>
</courses>
```

In the next section, you'll see how you can use the above XML file to create two courses on your server. Before we move on, here's a few additional details about this file format:

- ALL of the tags and attributes MUST exist, and be in the relative order shown in the samples above.
- All tags must be in lowercase letters.
- Only TWO of the tags can be empty - <teacher_title></teacher_title> and <username></username>
 - When <teacher_title></teacher_title> is empty, Manhattan will put the title "Prof." before the last name of the FIRST user it finds on the user list, and use that for the "teacher title".

- When `<username></username>` is empty, Manhattan will use its usual method for automatically deriving a person's username based on their initials and the last 4 digits of their ID number.
- Indentation and whitespace outside of tags has no meaning. It is NOT necessary to have your scripts generate neat looking XML code (although the small additional effort is worth the trouble since it makes it easier to debug your script.)

Using an XML file to create courses

The previous section, *The XML format by example* [92], describes a text file format that can be used to create Manhattan classrooms. In this section we'll use that method to create two Manhattan courses from a single XML file. Login to the administrative system and select Courses → Create courses → ...from single-use XML file



This leads to the page:

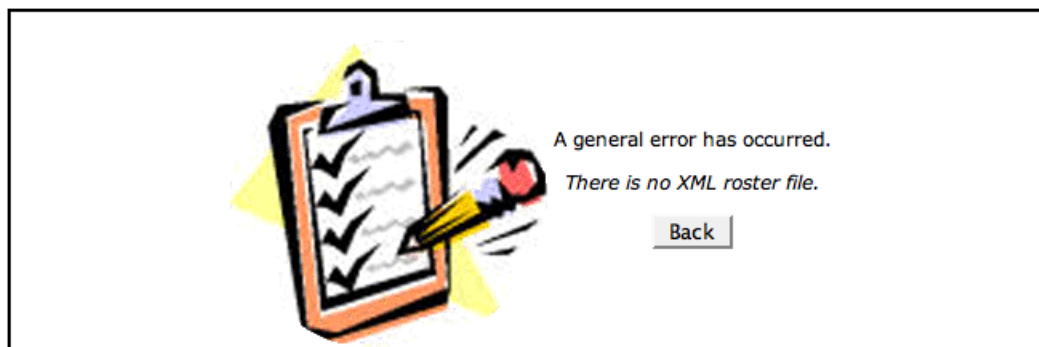
This utility allows you to create courses using an alternative XML format. You can either manually place this file on the server (perhaps by running a script of your own design), or you can use the above form to upload the file from your client PC.

The file must be placed in the 'tmp' directory of this Manhattan installation. The name of the file is based on the username of the person currently running this program. The format is 'CGI_DATA_<username>_roster.xml' where <username> represents your (administrator's) username. Specifically, at this moment, this program will upload files to, and create courses from, the file named:

../tmp/CGI_DATA_snarmont_roster.xml

The current contents of this file is shown in the frame below. You should always run this program using an XML-aware web browser such as Firefox or recent versions of Internet Explorer! With XML-aware browsers, a nicely formatted display will result if the XML file is 'well-formed'. If the XML is not 'well-formed', you'll get some indication of where the error occurred. Note that the 'well-formed' test simply checks things like whether the file is a valid XML file. It doesn't necessarily mean the file is correct in the context of this application!

Note that this file will be automatically deleted after it is used to successfully create courses. The 'Clean up' command on the Server menu will also delete this file if it is more than 3 hours old. These automatic deletions can only take place if Apache has write access to the file.



As shown in the screenshot above, the page consists of two sections. The top has a form that allows you to upload an XML file that describes the course or courses you want to create. After the XML file has been uploaded, on the **Create courses based on the displayed XML file** button to actually create the courses.

The bottom frame displays the contents of the XML file, which is waiting to be processed. When you first visit this page, there won't be an XML file and you'll get the "There is no XML roster file" message shown in the above screenshot.

To demonstrate, copy/paste the following XML into a new file using a text editor on your PC. (If you're reading a printed copy of this manual, visit <http://manhattan.sourceforge.net> for an HTML version of this manual.) Save the file.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<courses>
  <course subdir="s03" id="phy10101">
    <course_no>PHY 101 01</course_no>
    <course_title>Introduction to Physics</course_title>
    <term>Spring 2003</term>
    <teacher_title>Prof. Einstein</teacher_title>
    <users>
      <user id="X34322">
        <first>Albert</first>
        <last>Einstein</last>
        <username></username>
        <group>faculty</group>
      </user>
      <user id="X343888">
        <first>Neils</first>
        <last>Bohr</last>
        <username></username>
        <group>student</group>
      </user>
      <user id="X347332">
        <first>Isaac</first>
        <last>Newton</last>
        <username></username>
        <group>student</group>
      </user>
      <user id="X394032">
        <first>Michael</first>
        <last>Jordan</last>
        <username></username>
        <group>student</group>
      </user>
    </users>
  </course>
  <course subdir="s03" id="eng10101">
    <course_no>ENG 101 01</course_no>
    <course_title>English Composition I</course_title>
    <term>Spring 2003</term>
    <teacher_title>Prof. Fuller</teacher_title>
    <users>
      <user id="X349933">
        <first>Janet</first>
```

```

        <last>Fuller</last>
        <username></username>
        <group>faculty</group>
    </user>
    <user id="X348756">
        <first>John</first>
        <last>Narmontas</last>
        <username></username>
        <group>student</group>
    </user>
</users>
</course>
</courses>

```

Now click on the **Browse** button in the top frame, and use your browser's "find file" dialogs to locate and open the XML text file you just saved. Then click the **Upload roster XML file** button. The file will be transferred to the server and you'll be returned to the same page, which will look something like this:

Create courses based on the displayed XML file

This utility allows you to create courses using an alternative XML format. You can either manually place this file on the server (perhaps by running a script of your own design), or you can use the above form to upload the file from your client PC.

The file must be placed in the 'tmp' directory of this Manhattan installation. The name of the file is based on the username of the person currently running this program. The format is 'CGI_DATA_<username>_roster.xml' where <username> represents your (administrator's) username. Specifically, at this moment, this program will upload files to, and create courses from, the file named:

../tmp/CGI_DATA_snarmont_roster.xml

The current contents of this file is shown in the frame below. You should always run this program using an XML-aware web browser such as Firefox or recent versions of Internet Explorer! With XML-aware browsers, a nicely formatted display will result if the XML file is 'well-formed'. If the XML is not 'well-formed', you'll see an error message where the error occurred. Note that the 'well-formed' test simply checks things like whether the file is well-formed. It doesn't necessarily mean the file is correct in the context of this application!

Note that this file will be automatically deleted after it is used to successfully create courses. The **Remove** button on the Server menu will also delete this file if it is more than 3 hours old. These automatic deletions will only take place if Apache has write access to the file.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

- <courses>
- <course subdir="s03" id="phy10101">
  <course_no>PHY 101 01</course_no>
  <course_title>Introduction to Physics</course_title>
  <term>Spring 2003</term>
  <teacher_title>Prof. Einstein</teacher_title>
- <users>
- <user id="X34322">
  <first>Albert</first>
  <last>Einstein</last>
  <username/>
  <group>faculty</group>
</user>

```

After uploading the XML file, it will be displayed in the bottom frame.

Modern web browsers are “XML-aware”. When they encounter an XML file, they generally look for a “style sheet” that tells how to display the data. If there is no style sheet, the raw XML data is displayed in such a way as to show the structure of the XML file. XML-aware web browsers will also tell you whether or not the XML is “well-formed”. “Well-formed-ness” is a simple test which checks for example, that there is a closing tag for each opening tag. However, being “well-formed” is not a sufficient condition for Manhattan being able to create courses using the file. The following is “well-formed” XML, for example, but of course it could not be used by Manhattan to create courses:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<lunch>
  <hotdog>yech!</hotdog>
</lunch>
```

Nevertheless, the ability for XML-aware web browsers to check XML files to see if they are well-formed can be useful when you are first debugging your scripts that generate XML files to create Manhattan courses. The next screenshot shows the results in the Firefox web browser when a `</course_title>` tag was left out of an uploaded XML file:

XML Parsing Error: mismatched tag. Expected: </course_title>.
Location: http://empire.wnec.edu/manhat-sbin/super_upload_xml_ro
Line Number 38, Column 30:

`</course>`
 -----^

After uploading an XML file, and verifying that it is at least well-formed, you can click the **Create courses based on the displayed XML file** button to proceed to the next step:

You can change the **Group** and/or the **Internal Name** of the courses, from what was specified in the XML file.

Course Code	Course Title	Semester	Group	Internal Name	Instructors	Course Type
PHY 101 01	Introduction to Physics	Spring 2003	s03	phy10101	Albert Einstein	Normal
ENG 101 01	English Composition I	Spring 2003	s03	eng10101	Janet Fuller	Normal

Total Courses: 2

Create/update these courses

To combine two or more rosters into one classroom, make the **Group** and **Internal Name** the same!

As shown in the above screenshot, the next step allows you to:

- Manually edit the “course group” and/or the “internal course name” for each course represented in the XML file. (See *Choosing a group name* [88] and *Choosing an Internal Course Name* [90].)
- Decide whether you want to create a “normal”, “standalone”, or “course template”. (See *Normal, Template and Standalone Courses* [22].)



You can combine two courses into one at this point by manually changing both the “Group” and the “Internal course name” to be the same. This is useful when a teacher wants two or more sections of the same course in a single Manhattan classroom.

After reviewing and possibly modifying the elements on this form, click the **Create/update these courses** button to complete the process. The courses will be created and you'll get a results page listing the courses including a table of teachers and students who were added to the courses.

Skipping the XML Upload step

The previous sections described how to use the XML file format to create new Manhattan courses. One thing that should be clear about the file format is that while it is easy for humans to read, it is not at all easy for humans to type. *Use of the XML format makes sense only if you can develop your own programs or scripts to generate the XML file.* Typically, these custom programs/scripts will query your student information system to get the required fields such as course title, semester, and information describing each student and teacher in the course.

It is not always convenient to first download a script generated file to your client PC only so you can upload it to Manhattan via the mechanism described above. You can skip the upload step entirely if you can write your script so that it saves the file directly to the tmp directory of your Manhattan installation. The trick is to:

- Write the file to the tmp directory of your Manhattan installation. (This is NOT the same as the /tmp that's a standard part of your Unix filesystem - it's a directory named tmp found as something like /home/manhat/manhat-x.x.x/tmp.)
- Name the file `CGI_DATA_<YOUR_SUPER_USERNAME>_roster.xml` where you replace `<YOUR_SUPER_USERNAME>` with the username you use to login to Manhattan's administrative programs.
- Make sure the apache process (httpd) has read/write access to the file. If you've followed the installation instructions, this can be done by setting the group of the file to `www` and giving `rw-` access to members of that group.

After your script directly writes the XML roster file to Manhattan's tmp, directory you can simply select Courses → Create courses → ...from single-use XML file from the administrator's menu. If you've named the file correctly, the XML file will appear in the lower frame of the page that appears. Simply click **Create courses based on the displayed XML file** to process the file.

The XML file format is a big improvement over the simple roster text file format. However, the ultimate method for using the XML format to keep your Manhattan classrooms updated is not the “single use” method just described, but the “XML Roster Manager” described in the next section.

The XML Roster Manager

The section *The Alternative XML format [92]* described the advantages of using Manhattan's XML format for creating courses compared to the legacy roster text file method. A new feature introduced in Manhattan 3.2, called the **XML Roster Manager**, further simplifies and automates the creation and updating of Manhattan classrooms.

The XML files used by the **XML Roster Manager** have exactly the same format described in *The XML format by example [92]*. The difference is in which courses are represented in the XML file, and how long the XML file remains on the server as a data source. As described in *Using an XML file to create*

courses [97], when using the **...from single-use XML file** method, the XML file generated by your scripts should only contain data about the courses you want to add to the server (or existing Manhattan courses you'd like to update). Once the courses are created from that uploaded data file, the XML file is permanently deleted from the server.

In contrast, to make the best use of **The XML Roster Manager** tools, you'll want your XML file to contain information on every course offered at your school, whether or not the teachers of those courses currently intend to use Manhattan. You'll find you can upload more than one XML file to your server, perhaps to represent different divisions such as Graduate or Undergraduate courses, or possibly to represent more than one semester. The XML files stay on the server until you delete them, or replace them with a version containing up-to-date roster information. While the XML files are on the server they act as a data store, providing roster information to support the following:

- Any teacher can request that a Manhattan classroom be created for a course they teach, via a **Request Classroom** link that appears on their “My Manhattan” page shown to them when they login. The Manhattan administrator (that's you!) will see a list of course requests and can honor them by simply clicking a button. (It's also possible to set up a `cron` job to automate this process.)
- As the Manhattan administrator, you can create courses from the available XML files. Instead of the one-shot deal described in *Using an XML file to create courses* [97], you're able to select the courses you'd like to create, combining two or more courses into a single Manhattan classroom, if you wish.
- At the click of a button, you can update the rosters for every course on the server that was created from a particular XML file.
- You can easily create a Manhattan account for every student or teacher at your school, without adding them to any Manhattan classroom.

You must enable this feature!

When enabled, the XML Roster Manager is accessed by selecting Courses → Create courses → XML Roster Manager from the Administrator's menu. *There's a good chance this feature is not available in your installation, since it is turned off by default.* To activate the feature, use a text editor to open the file named `custom.h` found in the `src` directory of your installation. Look for the entry `ENABLE_FACULTY_COURSE_REQUEST` towards the bottom of the file:

```
#define ENABLE_FACULTY_COURSE_REQUEST 0
```

You'll need to change the `0` to a `1` as shown below and save the file:

```
#define ENABLE_FACULTY_COURSE_REQUEST 1
```

After saving the file, make sure you are in the `src` directory, and recompile Manhattan by typing the commands:

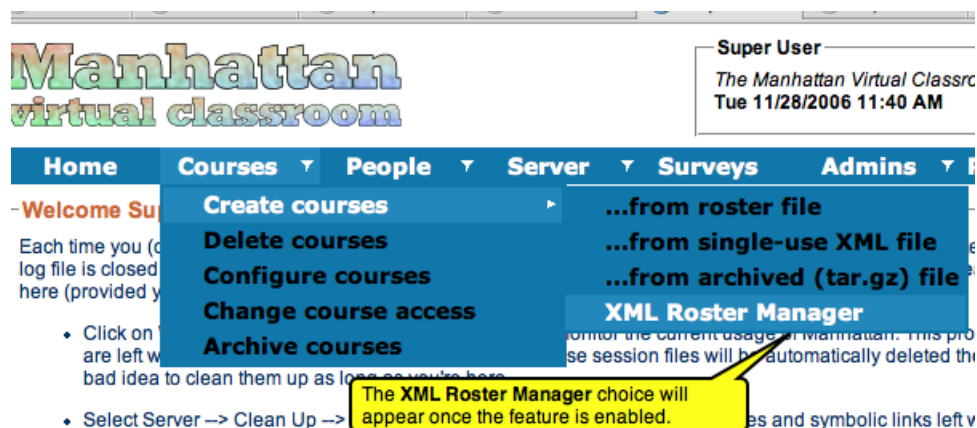
```
make clean
make install
```

For good measure, you should also *Test your Apache configuration and set your file permissions* [11] as described in the installation instructions.

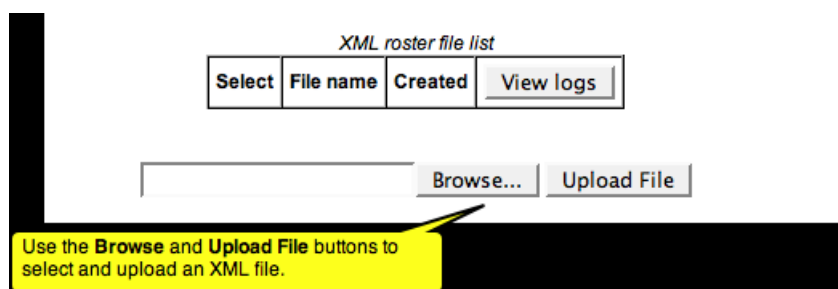
Once you've properly activated the feature two things will change. First, users will see an additional **Request classroom** link in the upper right corner of their “My Manhattan” page. We'll describe the use of that link shortly. Second, the Courses → Create courses → XML Roster Manager option, described next, will now be available on the administrator's menu.

Using the XML Roster Manager

Once the feature has been enabled (see above) you'll find the XML Roster Manager on the **Create Courses** menu:



Initially, the “XML roster file list” will be empty. Use the “Upload File” form to select and upload an XML file to the server:



As explained above, the uploaded XML file must have exactly the same format described in *The XML format by example* [92]. The file should represent all of the courses you might want to add to the Manhattan server.



When using the XML Roster Manager, the name of the file matters!

The name of the file matters, since it affects what teachers will see when they use the **Request Classroom** feature (described below) on their My Manhattan page. Use a dash - to separate the name of the Semester from the Academic Year. Underscores _ that appear to the left of the first dash - will be replaced by spaces. Some examples:

	What teachers see when they 'Request Courses'	
File name	Semester	Academic Year
Spring_Semester-2007.xml	Spring Semester	2007
Fall_and_Winter-2007-2008.xml	Fall and Winter	2007-2008
SloppyAdministrator.xml	SloppyAdministrator.xml	
this_and_that-2006_2007.xml	this and that	2006_2007



The uploaded files are stored in the `auto_update/rosters` directory of your Manhattan installation. You can write your own scripts to automatically upload XML files to that directory if you prefer.



If there's an existing file with the same name on the server, that file will be replaced by the newly uploaded file without warning.

Once you've uploaded one or more XML files, the XML roster list will no longer be empty, and you'll get some buttons that allow you to work with the roster data:

XML roster file list			
Select	File name	Created	View logs
☐	Semester1-2006.xml	Tue 11/07/2006 11:36 AM	View XML file
Delete file	Create courses	Create new faculty accounts	Create new student accounts
Update cours			

Here's a rundown of what the buttons do:

View XML file

This displays the raw XML in your web browser. Assuming your browser is “XML aware” (most are these days), this can be used to test whether the uploaded XML file is “well formed”. It also allows you to peek inside the XML file to see what courses and people it contains.

As mentioned in *Using an XML file to create courses* [97], “well formed” means simply that the XML file follows the basic rules that all XML files must follow. For example if there's an open tag `<foo>`, there must be a matching `</foo>` closing tag. This does not necessarily mean it's a valid Manhattan roster XML file!

View logs

Use this to view/delete the system logs that record when some of the options on this page were used. Examples of activities that are recorded in the logs include: a teacher requests a Manhattan classroom, an administrator honors that request, an administrator updates the classroom rosters on the server, an administrator creates faculty accounts from an XML file, and so on.

Delete file

As stated above the XML roster files remain on the system, where they serve as a data store. Use **Delete file** to permanently remove a file from the server.

Create courses

Lists the courses contained within the selected XML file. You can choose to create as many or as few courses as you wish from that list. The list also shows the courses that have already been created from the selected XML file.

See *Using an XML file to create courses* [97] for details, including a tip on how to combine two or more courses into one Manhattan classroom.

Create new faculty accounts

Automatically creates a Manhattan account for all teachers listed in the selected XML file who do not already have an account. Those

teachers can then login and use the **Request a Classroom** feature to select from a list of courses they teach.

Create new student accounts

Automatically creates a Manhattan account for all students listed in the selected XML file who do not already have an account. Teachers can then add any of those students to their own classrooms by using the **Add a student** option on the classroom's Configuration menu.

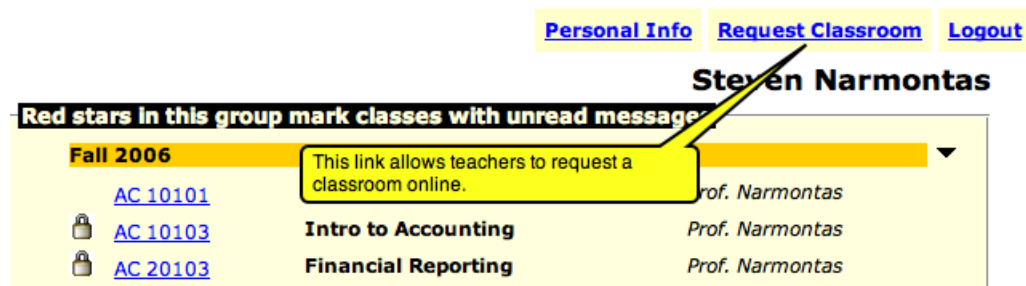
Update courses

Use this to synchronize Manhattan classroom rosters with the “official” and up-to-date roster information contained within the XML file. For each course in the selected XML file *THAT HAS ALREADY BEEN CREATED*, the current Manhattan roster is compared to the roster found in the XML file and the following actions are taken:

- Students listed in the XML file who are not in the Manhattan classroom are automatically added. The teacher will be notified of these added students the next time s/he enters the classroom.
- Any students who are in the Manhattan classroom, but who are no longer in the XML file will be flagged for possible removal from the classroom. In this case, the teacher will be given an option to delete the student, or to keep the student in the Manhattan classroom.

The Request Classroom feature

As mentioned above, when `ENABLE_FACULTY_COURSE_REQUEST` is set to 1 in the `custom.h` file, and Manhattan is recompiled, users will see an additional **Request Classroom** link in the upper right corner of their “My Manhattan” page:



When the link is selected, the user sees a list of the XML files, which were uploaded to the server by the Manhattan administrator as described above. Here's a screenshot of an installation that currently has five XML roster files available:

Please do NOT use this program if you want to have two or more sections combined into a single Manhattan classroom. Contact the Manhattan administrators if you want to combine sections into a single classroom!

Add course requests

Select		Academic year	Roster snapshot taken
☐	Law school Fall	2006-2007	Wed 11/29/2006 12:00 PM
☐	Day and evening Winter	2006-2007	Wed 11/29/2006 12:00 PM
☐	Off campus Fall	2006-2007	Wed 11/29/2006 12:00 PM
☐	Day and evening Fall	2006-2007	Wed 11/29/2006 12:00 PM
☐	11 week Fall	2006-2007	Wed 11/29/2006 12:00 PM

At this particular school, a separate XML file is used for Law classes, undergraduate classes, winter semester, and so on. As mentioned above, what a teacher sees in the table depends on the name of the file that was uploaded. The row Law school Fall with Academic Year 2006 - 2007 reads that way because the administrator uploaded an XML file named Law_school_Fall-2006-2007.xml.

In any case the teacher selects one of the roster files and clicks the **List My Courses** button. The user will be presented with a list of courses he teaches, which are listed in the selected roster file:

Please do NOT use this program if you want to have two or more sections combined into a single Manhattan classroom. You must contact the Manhattan administrator if you want to combine sections into a single classroom!

Add course requests

Courses you teach in Day and evening Fall 2006-2007
Check the courses you want to put on Manhattan

Select	Course Code	Course Title	Semester	Instructors	Status	Notes
☒	HIST11101	U.S. History To 1877	Fall 2006	Steven Narmontas	Not requested	
☒	HIST11103	U.S. History To 1877	Fall 2006	Steven Narmontas	Not requested	
☐	HIST11105	U.S. History To 1877	Fall 2006	Steven Narmontas	Not requested	
☐	HIST11115	U.S. History To 1877	Fall 2006	Steven Narmontas	Not requested	
☒	HIST35401	Civ War Reconstruction	Fall 2006	Steven Narmontas	Not requested	

Total Courses: 5

After the teacher selects one or more courses from the list and submits the form, he'll see something like:

Course request

You have requested the following course(s). Your course(s) will be automatically added within the next 12 hours.

Title	Course Code	Semester
U.S. History To 1877	HIST11101	Fall 2006
U.S. History To 1877	HIST11103	Fall 2006
Civ War Reconstruction	HIST35401	Fall 2006



The program says the Manhattan classrooms will be created within 12 hours. Unless the site administrator has set up a `crON` job to automate the process, the request will NOT be automatically be processed. That's the job of the Manhattan administrator!

Back to the administrator's point of view, when one or more teachers have requested a classroom, a section like the following appears on the **XML Roster Manager** page we've been describing in this section:

Course Creation requests			
Select	File name	Created	
☐	Day_and_evening_Fall-2006-2007.xml	Wed 11/29/2006 03:29 PM	View XML file
Process request			

This section lists the names of the XML files that have one or more pending classroom requests. If you click on the **View XML** button next to a listing, you'll see something like:

```
- <requests>
- <request>
  <faculty_realname>Steven Narmontas</faculty_realname>
  <faculty_username>sn5019</faculty_username>
  <sis_id>hist11101</sis_id>
  <subdir>c206</subdir>
  <new_id>hist11101</new_id>
  <course_code>HIST11101</course_code>
  <course_title>U.S. History To 1877</course_title>
  <course_term>Fall 2006</course_term>
  <time>Wed 11/29/2006 03:34 PM</time>
</request>
- <request>
  <faculty_realname>Steven Narmontas</faculty_realname>
  <faculty_username>sn5019</faculty_username>
  <sis_id>hist11103</sis_id>
  <subdir>c206</subdir>
  <new_id>hist11103</new_id>
  <course_code>HIST11103</course_code>
  <course_title>U.S. History To 1877</course_title>
  <course_term>Fall 2006</course_term>
  <time>Wed 11/29/2006 03:34 PM</time>
</request>
```

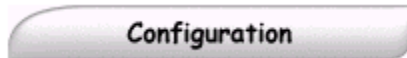
This XML format is actually different than the one used to describe courses and the roster of teachers and students that belong to that courses. It's not really important to understand the structure of this file, since it is used only internally, however it is easy to make out the who, what and when of the course requests represented by the file.

To honor the course requests, the administrator simply selects the request file and clicks the **Process request** button.

Again, keep in mind that while teachers can *request* Manhattan classrooms online, the classrooms are *NOT* created automatically. It is up to a Manhattan administrator to monitor and fulfill the requests as just described.

Chapter 7. Configure courses - details

A student's view of a Manhattan classroom is very similar to the teacher's. Manhattan doesn't have a special "designer mode" that is used by the teacher to make learning materials available to her students. For the most part, the teacher sees the same menus, buttons, and pages that a student sees, although often the teacher has a few more commands available than a student does. One very significant difference between a student's and teacher's view of Manhattan is contained within the main menu's



button. Only the teacher of the course has the Configuration button as the last item on the main menu.

Once a Manhattan classroom has been set up by the server administrator, the teacher is in full control. Using commands found within the *Configuration* section, the teacher can:

- Change details like the name of the course, the semester, and how your name appears throughout the system.
- Turn Manhattan's modules on or off.
- Reset student passwords, and make minor changes to their names.
- Add new students and additional teachers to the classroom.
- Remove students and teachers from the classroom.
- Get a roster for the course as a text file that can be imported into a spreadsheet or grade book program.
- Organize the students into teams.
- View logs that show when students have logged in, and what they have done while logged in.
- Get information about the amount of disk space used by the classroom.
- View the contents of their *Clipboard*, which is a special feature that allows teachers to reuse Manhattan messages.



This chapter explains the options available to the teacher via the **Configuration** menu. As the system administrator, you have access to the same capabilities via the **Configure courses** link on your administrative menu - see *Accessing a course's configuration menu* [38].

Basic classroom configuration

The first item in the configuration menu reads:

Basic classroom configuration

Go here to enable and disable modules. You can also change basic information about the course (including the course title and teacher's name), or lock the classroom to keep students out. Message options allow you to enable or disable the spell checker, and choose whether or not to use 'emoticons' or fixed-width fonts, etc.

Clinking on the link yields this a rather long form that is divided into four sections:

Section 1 - module selection	If you've followed the tutorial in this manual, you've already used this part of the form (see <i>Logging in as Prof. Einstein</i> [27].) This is where a teacher selects the modules he wishes to use in the classroom. Click on the help icons  in this section of the form to learn more about a module or a feature.
Section 2 - classroom locking	When a classroom is locked, students will see a message like "You've logged in successfully, but your teacher has not yet set up this classroom" when they enter. The teacher will be reminded that the classroom is locked whenever he visits that classroom's Main Menu. All Manhattan classrooms start off locked, to give teachers an opportunity to set things up before students enter.
Section 3 - course info	The text entered in the <i>Course-Code-Section</i>, <i>Course Title</i>, <i>Semester</i>, and <i>Instructor</i> fields are displayed in various places throughout the Manhattan classroom. The initial values of these fields are taken from the roster text file (or the XML file) you used to create the course. (See <i>More about the roster text file</i> [87]) Changing the <i>Semester</i> field changes the way the course will be listed on the "My Manhattan" page. Be consistent!
Section 4 - miscellaneous settings	This section allows you to choose settings for the Spell Checker, proportional or fixed font, emoticons, and the width of the textarea elements used to compose Manhattan messages.

View course roster

The second configuration item basically answers the question "Who's in this class?"

[View course roster](#)

Provides a complete list of members of the class and shows their usernames, id numbers, and teams.

Here's a screenshot of a typical class roster:

Total number of people in this class, including the teacher(s): 7

Username	Real Name	Type	Alias	Team	Id
nb3888	Neils Bohr	Student	anonymous	A	X343888
ae4322	Albert Einstein	Faculty	anonymous	Z	X34322
rf5734	Robert Frost	Student	anonymous	A	X485734
mj4032	Michael Jordan	Student	anonymous	A	X394032
in7332	Isaac Newton	Student	anonymous	A	X347332
cs4452	Cindy Slattery	Student	anonymous	A	X394452
ss4321	Steven Stanton	Student	anonymous	A	X334321

The "Username", along with a secret password, is what a user types to enter Manhattan. The "Name" column lists the person's real name, which can be modified using another choice on the Configuration menu (see *Change a person's name, team, or alias* [119]) If a person has created a Personal Info Page, their name will be a link to that page. This is the case for Albert Einstein in the above screenshot.

A person's "Type" can be either "student" or "faculty".

The "Alias" column lists how each person will be known in one particular Manhattan module, the *Anonymous Discussion* module, which is covered in Manhattan's *Teacher's Reference*. Usually a person's alias is left set to "anonymous". It is possible to change this to any string characters, for example if the teacher wants to be known by her real name within the *Anonymous Discussion* module.

Manhattan allows the teacher to divide a class into *teams* for the purpose of collaborating on projects. There are 52 possible teams, identified by the letters A through ZZ. The "Team" column lists a person's current team. When the server administrator first sets up a Manhattan classroom, all students are in team A, and the teacher is placed in team Z. Two commands on the Configuration menu allow you to change a person's team - see *Change a person's name, team, or alias* [119] and *Assign teams* [120].

The Team Discussion modules are discussed in Manhattan's *Teacher's Reference*.

As you know, each user must have an identification number, as listed in the "Id" column.

Get course roster

The third item on the Configuration menu allows you to retrieve a file containing a list of the students in the class:

Get course roster (select type):

[TAB-delimited](#) [CSV - Comma Separated Values](#)

Provides a TAB-delimited (or Comma Separated Values), quotes around text, text file of Student IDs and full names for import into most spreadsheets and grade manager programs.

[Fixed Width Fields](#)

Provides a fixed width field text file of student names and ID numbers. There's one student line. The name field starts at column 1 and is 35 columns long. The ID number field starts at column 36 and is 15 columns long. This format is provided for roster imports into the commercial GradeQuick grades management package - see <http://www.jacksonsoftware.com>

This feature may be of interest to the teacher, since it provides a way to import the class roster into a spreadsheet program such as Microsoft's Excel, for managing grades.

When you click on one of the links, Manhattan immediately sends you the text file in that format. A "Save File" dialog box will pop up on your web browser. The name of the file will be same as the "internal course name" used to identify the classroom on the server, followed by the extension .txt.

The format of the file allows teachers to easily create a spreadsheet file to upload to Manhattan's *Grades* module. The *Grades* module is covered in detail in Manhattan's *Teacher's Reference*, which includes an explanation on how to import a roster file into Microsoft Excel.

View student login logs, class access stats

[View student login logs, class access statistics, etc.](#)

Answer questions like: When was the last time Jane logged in? When do students tend to login? How much disk space does my class use? ... and more

Clicking on this menu item leads to another menu:

Recent Logins

View a table showing, for each student, the number of unread messages, the number of logins since the course began, and the dates and times of the last 10 logins. You can click on a person's name to view their 'event calendar'.

Event Calendar

Using a familiar calendar format, this option provides an at-a-glance view of when a student has logged in, and when he or she has read and sent messages. The calendar also displays when messages have arrived for each student. You can click on each symbol on the calendar to get more information.

Event List

This shows the same information as the "Event Calendar" (see above) in a table format.

Disk Usage

Displays, in kilobytes, the amount of server disk space used by this classroom and the free disk space remaining on the server.

Here's the big picture. Internally, Manhattan records, for both the students and the teacher, the date and time of each of the following transactions:

- when a user logs-in to the classroom
- when a user opens a message for the first time
- when a user sends a message
- when a user enters the *Chat* room

The first three items on the Statistics Menu allow you to view summaries of these transactions. The overall intent of this feature is to give the teacher a way to gauge the participation of her students within the virtual classroom. The last item, *Disk Usage* gives the teacher some additional information on the state of the Manhattan server. The following sections explore these options in the order they are listed in the menu.

Recent logins

Recent Logins

View a table showing, for each student, the number of unread messages, the number of logins since the course began, and the dates and times of the last 10 logins. You can click on a person's name to view their 'event calendar'.

This is perhaps the statistics feature you'll use most often. Clicking on the link yields a table similar to the following:

 Id/ Name	 Unread Msgs	 Number Logins	Last 10 Logins					
082453 Amber Bardliving	42	51	Fri 12/01/2006 02:01 PM	Thu 11/30/2006 01:45 PM	Wed 11/29/2006 05:53 PM	Wed 11/29/2006 03:50 PM	Tue 11/21/2006 10:55 AM	Mon 11/20/2006 03:19
084206 Morgan Bradley	0	202	Fri 12/01/2006 01:44 PM	Fri 12/01/2006 12:29 PM	Thu 11/30/2006 08:31 PM	Thu 11/30/2006 12:26 PM	Thu 11/30/2006 11:54 AM	Thu 11/30/2006 11:16
1275 Maurice Casey	20	90	Fri 12/01/2006 11:12 AM	Fri 12/01/2006 11:04 AM	Fri 12/01/2006 11:02 AM	Fri 12/01/2006 11:00 AM	Fri 12/01/2006 10:59 AM	Fri 12/01/2006 10:46
084929 Michael	0	169	Fri 12/01/2006 11:12 AM	Fri 12/01/2006 11:04 AM	Fri 12/01/2006 11:02 AM	Fri 12/01/2006 11:00 AM	Thu 11/30/2006 11:54 AM	Thu 11/30/2006 11:16

The table lists all of the people in the class and shows the date and time of the last ten times they have logged in to the classroom. The table also shows the total number of times they have logged in, as well as the number of unread messages they have waiting for them.

You'll note that each name is a link. Clicking on a name yields the *Event Calendar* for that person, which is also available via a separate choice on the statistics menu as described next.

Event Calendar

[Event Calendar](#)

Using a familiar calendar format, this option provides an at-a-glance view of when a student has logged in, and when he or she has read and sent messages. The calendar also displays when messages have arrived for each student. You can click on each symbol on the calendar to get more information.

Select a name from the list you'll see after making this selection and click the button to view that person's *Event Calendar*:

<< September 2006 >>						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6      	7 	8  	9
10	11	12  	13	14  	15 	16  

The event calendar shows, for each day within the range specified by the start and end dates of your class, the following activities:

- the time of each log-in (shown as a "smiley face")
- the time a new message became available for a person to read (shown as a yellow envelope)
- the time a message was read for the first time (shown as binoculars)
- the time the person composed a new message (shown as a hand addressing an envelope)
- the time the person entered the live Chat area (shown as a talking face icon)

Positioning your mouse over, or clicking on, one of the symbols will give you some detailed information about the event.

Two activities are NOT depicted on the calendar, or anywhere else within the statistics menus:

- when the student accessed the *Grades* module
- all activities within the *Anonymous Discussion* module.



Activity within the *Grades* module is available only to the teacher by entering the module from the course's main menu. Tracking activity within the *Anonymous Discussion* would make it possible to deduce who posted certain messages and so the information is hidden from all users.

Event List

Event List

This shows the same information as the "Event Calendar" (see above) in a table format.

Here's what an *Event List* looks like:

Time	Event	Module	Detail
Wed 08/02/2006 10:41 AM	Msg arrived	Lectures	Week 1 8-30-2006 (10:41 AM)
Wed 08/02/2006 10:42 AM	Msg arrived	Lectures	Week 2 9-06-2006 (10:42 AM)
Wed 08/02/2006 10:43 AM	Msg arrived	Lectures	Week 3 9-13-2006 (10:43 AM)
Wed 08/02/2006 10:44 AM	Msg arrived	Lectures	Week 4 9-20-2006 (10:44 AM)
Wed 08/02/2006 10:45 AM	Msg arrived	Lectures	Week 5 9-27-2006 (10:45 AM)
Wed 08/02/2006 10:45 AM	Msg arrived	Lectures	Week 6 10-4-2006 (10:45 AM)
Wed 08/02/2006 10:46 AM	Msg arrived	Lectures	Week 7 10-11-2006 (10:46 AM)
Wed 08/02/2006 10:47 AM	Msg arrived	Lectures	Week 8 10-18-2006 (10:47 AM)
Wed 08/02/2006 10:48 AM	Msg arrived	Lectures	Week 9 10-25-2006 (10:48 AM)

The *Event List* shows the exact same information about an individual's activity within the classroom as the *Event Calendar* (see *Event Calendar* [113]), except it is presented in a simple table.

Disk Usage

Disk Usage

Displays, in kilobytes, the amount of server disk space used by this classroom and the free disk space remaining on the server.

Here's a typical report:

Disk Usage as of Mon 12/04/2006 12:59 PM

This classroom is currently using 24462 kilobytes of disk space. (A kilobyte is 1,024 characters.)

The 'disk free'(df) command on the server yields the following report.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sdc5	1.2T	189G	981G	17%	/udd

Manhattan's *Disk Usage* command simply runs a program on your server called *df* (which stands for 'disk free'), and shows you the results. In the above example, we learn that the classroom is using 24,462K of disk space. The server has 981 Gigabytes of free space left.

This information is provided to the teacher only as a point of information. Some day they might wonder if there's enough room on the server for that huge streaming video file they'd like to make available for their students!

Add a student

Add a student

You can add a student anytime. The new student will receive a copy of all messages posted to the system except Post Office messages and Grades.

As the system administrator, you are usually able to pre-load a Manhattan classroom with the students who are registered for a course. Nevertheless, there will be occasions when you will need to add additional students to a classroom.

While most of the Configuration commands described within this chapter work the same for teachers and for system administrators, the **Add a student** and **Add a teacher** commands are exceptions. Recall that when you first installed Manhattan you had an option to edit a `custom.h` file that controls certain behaviors (see *Create and edit custom.h* [9]). Two of the items within the file:

```
#define ALLOW_TEACHER_CREATE_NEW_ACCOUNT 0
#define ALLOW_TEACHER_SELECT_USER_BY_COURSE 0
```

directly affect what *teachers* see in this section of the Configuration menu.

Teachers can always add a student or teacher to their classroom by performing a search, then selecting from a list of users who already have centralized Manhattan accounts. That's a default setting that can't be taken away from teachers by editing the `custom.h` before compiling Manhattan.

But suppose the teacher wants to add a student who doesn't already have a Manhattan account? Do you, as system administrator, want that teacher to create an entirely new account for that student? (In our experience the answer is "No" - invariably they'll make data entry errors, which you'll have to clean up later!)

If you set `ALLOW_TEACHER_CREATE_NEW_ACCOUNT` to '1' in `custom.h`, teachers will get a form that prompts them for the name and ID number of the entirely new person they want to add to the central database of users. *This form will appear only after they do a search for the student, and appears whether or not the search was successful.*

If this feature is turned off, by setting `ALLOW_TEACHER_CREATE_NEW_ACCOUNT` to 0, teachers will only be able to add existing users to their classrooms. They'll have to contact the system administrator - that's you - if they need to add an entirely new student or teacher to their classroom.

The second decision to make is whether or not you want to enable teachers to get a list of students/teachers in any other “normal” or “centralized” course on the server, so they can select students from that list to add to their classroom. This feature is useful in certain (typically K-12) environments. It allows a teacher, for example, to quickly add some or all students in Mrs. Smith's Biology class to their own class.

A good reason to disable this feature is to protect the privacy of students/teachers. When this feature is turned on, any teacher on the system can see who is enrolled in every Manhattan course on the system. That is, a teacher can browse through the rosters of courses other than their own.

Set `ALLOW_TEACHER_SELECT_USER_BY_COURSE` to 1 to enable this feature, and to 0 to disable this feature.



Both of these options are always enabled for the system administrator. When YOU “Add a student” or “Add a teacher”, both of these features will be enabled, regardless of the settings in `custom.h` file.



These settings apply only to courses tied to the central store of usernames/passwords - that is, “normal” courses and “course templates”. Teachers can always freely add students/teachers to standalone courses, even by inventing names and ID numbers. Teachers of standalone courses can NOT see any lists of centralized users.

When you click on the *Add a student* link, you'll see this form:

Add a Student

Both system administrators and teachers **ALWAYS** have the ability to search for students to add to the classroom.

The text entered in the search form will be matched against a person's ID number, username, and realname. The search handles partial matches. For example, simply typing 'mi' will list users who have an 'mi' in their real name, their username, or their ID number .

Search for:

Search

Instead of using the above search form, you can also click the button below to get a list of users who are currently enrolled in any centralized course on this server.

Look up users in courses

Administrators **ALWAYS** get this button. Teachers get this button **only if** `ALLOW_TEACHER_SELECT_USER_BY_COURSE` was enabled in `custom.h` when Manhattan was compiled.

To add a student, type in all or part of their name or ID number into the text box and click the **Search** button. You'll get a list of all matches, from which you can select a person to add to the classroom:



The list of matches (if any) appears at the *bottom of the page*. Scroll down!

1 Matches

		Real name	Id	Team
☐	jj5753	Janet Jackson	X485753	A

Add selected users to this class

Create a New Manhattan Account

First Name:

Last Name:

ID Number:

Team:

Add this new user

If any existing users match your search, they'll be listed at the bottom of the page.

You can select one or more from the list and submit the form to add them to the class.

This form, which allows you to add a completely new user to the class, is normally available only to system administrators.

As shown in the above screenshot, the results of the search also includes a form that allows you to add a completely new user to Manhattan. Thus, here are the steps that you, the system administrator, should follow to add a student to a Manhattan course:

1. Access the course's configuration menu via the **Configure courses** link on your administrator's menu.
2. Click on the **Add a student** link.
3. Enter the student's last name (or ID number) into the text box and click the **Search** button.
4. *Scroll down* to see the results of the search. If the person has been found, select them from the list and submit the form.
5. If you're sure the person doesn't already have an account you can use the "Create New Account" form to add the completely new user to the course, and to Manhattan's central store of user information.



The form that allows you to add a completely new user to the system appears only after you have performed a search for a user. This is to discourage you (and teachers if you enable them to use this feature) from attempting to create an entirely new Manhattan account before you are certain the person doesn't already have an account.

Add a teacher

Add a teacher

A class can have an unlimited number of teachers. All teachers have the same full access to Manhattan's features. The new teacher will receive a copy of all messages posted to the system except Post Office messages and Grades.

The mechanics for adding a teacher to your classroom are identical to those for adding a student - see *Add a student* [115].

Although the "Add a teacher" form asks you to enter a "Faculty ID" number, there really is no difference between a student ID and a faculty ID. A person can be a teacher in one class and a student in another.

All teachers in the classroom are equals in that they have access to exactly the same set of commands within Manhattan.

Delete a student or a teacher

The teacher or system administrator is also able to remove students from the classroom.

Delete a student or a teacher

You can safely delete students (or teachers) without deleting the messages they have posted to the classroom.

When you click on the *Delete a student or teacher* link, you'll see this:

Delete User

Neils Bohr
Robert Frost
Michael Jordan
Isaac Newton
Cindy Slattery
Steven Stanton

Both Admins and Teachers get this form.

Check here ☐ and here ☐
and if you really, really mean it

Click here

to remove the selected user from your classroom.

Any messages posted by a user will still be available after he/she has been removed from the classroom.

Quickly delete students

Check All | Uncheck All

Select	Username	Real Name	Id
☐	nb3888	Neils Bohr	X343888
☐	rf5734	Robert Frost	X485734
☐	mj4032	Michael Jordan	X394032
☐	in7332	Isaac Newton	X347332
☐	cs4452	Cindy Slattery	X394452
☐	ss4321	Steven Stanton	X334321

Delete

Any messages posted by a user will still be available after he/she has been removed from the classroom.

This one is seen only by Admins!

To delete a student or teacher from the classroom, first click on his name in the list. Next, click in both checkboxes, then click on the **Click Here** button.

If the deleted user never posted any messages to the classroom, his account is completely removed from the system. However, if the student or teacher did post one or more messages, those messages will still be available in the classroom.

You can restore a deleted student's or teacher's account by adding the person to the classroom using either the "Add a student" or the "Add a teacher" options in the Configuration menu. The person's account will be completely restored.



The system administrator can also remove a student, but not a teacher, from any classroom via a control on the user's "full record" accessible from the **View/modify/delete users** link on the administrator's menu.

Reset a person's password

[Reset a person's password](#)

Although you cannot look up a user's password, you can reset it to its original value.

No one, not even the system administrator, can look up a student or teacher's password. If a person forgets their password, the best anyone can do is to reset their password to its "default" or original value. As explained in *ID numbers, usernames and passwords* [86], when a new user is added to the system, their password is the same as their username. The first time they login, they'll be forced to change their password.

Thus resetting a person's password amounts to setting the password to be the same as their username. The next time they login, they'll be required to select another password.

Clicking on the "Reset a person's password" link yields the form:

Reset User's Password

Neils Bohr
*Albert Einstein
Robert Frost
Michael Jordan
Isaac Newton
Cindy Slattery
Steven Stanton

Click on a user in the list, then

to get a form that allows you
to reset his or her password.

Select a person and click the button to get a confirmation page:

Reset User's Password

Username:	in7332
Full Name:	Isaac Newton
Id:	X347332
Reset this User's Password	

After clicking the "Reset Password" button, the user's password will be the same as their username.



The system administrator can also reset a user's password via a control on the user's "full record" accessible from the **View/modify/delete users** link on the administrator's menu.

Change a person's name, team, or alias

[Change a person's name, team, or alias](#)

Go here to make minor changes to a person's name, to change their 'alias' used in the Anonymous Discussion, or to change user teams one at a time (also see the Assign Teams selection below).

When you select this item from the menu you'll see the following:

Change User's Name, Team or Alias

Neils Bohr
*Albert Einstein
Robert Frost
Michael Jordan
Isaac Newton
Cindy Slattery
Steven Stanton

Click on a user in the list, then
Click here
to get a form which allows you
to change his or her
information.

Use this form to select the person
whose info you'd like to change.

Change User's Name, Team or Alias

Username: in7332

Full Name: eg. correct the spelling, add or drop an initial, etc.

Alias: Used in 'Anonymous' discussion only. See notes below.

Team: Significant only if you are using the Team or Team/Teacher Discussion module. See additional notes below.

Make this change

After selecting a user, you'll get this form.

The most common reason for changing someone's Full Name is for example, when "Michael" prefers to be called "Mike".



This name change affects this classroom only. As the system administrator, you can change a person's "centralized name", that is, the name that appears on their login page, from the **View/modify/delete users** link on the administrator's menu.

The *Alias* setting determines how the user will be known within Manhattan's *Anonymous Discussion* module. Most teachers who use the *Anonymous Discussion* module leave this set to "anonymous" for all students. Another possibility is to change the teacher's *Alias* setting to their real name, so their messages within the *Anonymous Discussion* module are not anonymous at all. The use of the *Anonymous Discussion* module is explored in Manhattan's *Teacher's Reference*

Manhattan's *Team Discussion* and *Team/Teacher Discussion* modules depend on the ability to assign students to teams. This form, pictured above, allows you to change the team assignment for students one at a time. (Also see *Assign teams [120]*) A student can be assigned to one of 52, labeled A through ZZ. If the teacher does not plan to use teams, you can leave all students set to team A.

Assign teams

Assign teams

Use this form to place your students on a team, or to see the current team assignments at a glance.

Two of Manhattan's modules, the *Team Discussion* and *Team/Teacher Discussion* modules require that the students and teachers in the classroom be placed on teams. There are 52 possible teams, designated by the letters A through ZZ. The "Assign Teams" option is the best way to assign students to teams or to see the current team assignments at a glance. Clicking on the link yields a form like this:

Assign Teams**Number of people in this class (including the teacher): 7**

Username	Real Name	Group	Alias	Id	Team
nb3888	Neils Bohr	Student	anonymous	X343888	A ▾
rf5734	Robert Frost	Student	anonymous	X485734	A ▾
mj4032	Michael Jordan	Student	anonymous	X394032	B ▾
in7332	Isaac Newton	Student	anonymous	X347332	C ▾
cs4452	Cindy Slattery	Student	anonymous	X394452	B ▾
ss4321	Steven Stanton	Student	anonymous	X334321	C ▾
ae4322	Albert Einstein	Faculty	anonymous	X34322	Z ▾
Update teams					

See Manhattan's *Teacher's Reference* to learn more about the Team and Team/Teacher discussion modules.

Clipboard Viewer

Clipboard viewer

Your clipboard allows you to recycle course materials by using familiar "Copy and Paste" commands. Go [here](#) to view the contents of your clipboard, or to empty it.

Throughout Manhattan, the teacher will find command buttons that read "*Copy this message to my clipboard*" and "*Paste message from clipboard*". These Copy and Paste commands are Manhattan's mechanism for re-using course messages and course materials within your classrooms.

The *Clipboard Viewer* command found on the Configuration menu simply allows the teacher to see or delete the message that is currently in your clipboard.



The system administrator cannot view the contents of a teacher's clipboard. You'll find that this command is disabled on the Configuration menu.

Chapter 8. Surveys - additional details

The section called *Surveys [60]* within the tutorial covered the some of the basics of Manhattan's ability to design and deliver surveys, at least from the server administrator's perspective. The purpose of this chapter is to fill you in on “the rest of the story.” I won't go into detail on every mouse-click you can perform, instead we'll start with the big picture, then provide answers to several “How do I?” questions.

The big picture

A Manhattan server administrator can:

Design surveys	As you learned from the tutorial (see <i>Surveys [60]</i>) surveys can contain: • “Custom” items, which consist of HTML-formatted text. These are usually used to provide instructions to the person taking the survey. • Likert-scale questions. • Multiple-choice questions. • Short answer questions where the person can enter their response on a single line. • “Memo” field questions, where the person can type as much as they'd like into a rectangular text area.
Deliver surveys to one or more Manhattan classrooms	This was done in the tutorial. Surveys delivered in this manner appear at the top of a user's “My Manhattan” page when they first login.
Deliver surveys to the web	Surveys made “WWW accessible” by the administrator can be taken by anyone who is given the URL (web address) of the survey. This means you can issue surveys to people outside of the Manhattan environment.
Make surveys available to all Manhattan teachers	When the administrator marks a survey “Public”, that survey can be delivered by any teacher within his/her private Surveys module of a Manhattan classroom. Teachers can also make a copy of a Public survey, after which they can modify it to their liking.
Give a survey to a particular teacher	It is also possible for the server administrator to design a survey, and then hand over the survey to a particular teacher. The survey will appear only within that teacher's “Private” list of surveys within the Survey module of all of his classrooms.
Work with any teacher's Private surveys	Teachers can also create their own surveys, which will appear in their own “Private” lists. Manhattan administrators can enter that list of surveys and modify or delete them. Adminstrators can also copy a teacher's private surveys back to the administrator's “Private” list. Note that administrators can NOT see the results of surveys issued by teachers within the Surveys module. Administrators can only see the

questions teachers have designed - they can not see when or if the survey was delivered or what the results were.

The other part of the story is that Manhattan has a Surveys module, which can be used by a teacher to design and deliver surveys within their private classroom environment. While you'll have to read the *Teacher's Reference* for a complete description of the surveys module to see the teacher's perspective, here's what it enables a teacher to do:

Design their own surveys	Teachers use the same tools as administrators to design surveys. Like administrators, the surveys teachers design appear in their "private" list of surveys. They see the same private list in all of their classrooms.
--------------------------	---

Deliver surveys to their students	As shown in the tutorial (see <i>Delivering a survey</i> [65]), when an administrator delivers a survey to a class, it appears on the "My Manhattan" page of the students in the class. However, when a teacher delivers a survey to the students in his class, it appears as a message within the Surveys module of that classroom, and not on the "My Manhattan" page. Again you'll need to read the <i>Teacher's Reference</i> (and experiment with being a teacher) to see their perspective.
-----------------------------------	---

It's important to realize, however, that a teacher can choose to deliver either a survey of their own design (from their "Private" list) or a survey the Manhattan administrator has made "Public" (it will appear in a "Public" list for that teacher.)

Collect results of surveys - privately	Manhattan's surveys module gives the results of a survey directly back to the teacher. Only the teacher or teachers of that particular classroom can see the results. They are not available to the system administrator
--	--

How much you, as system administrator, have to get involved with surveys depends on your interest and also on the procedures for your school. At one end of the spectrum, it is possible that you don't need to get involved at all. Teachers have all the tools they need to design, deliver, and collect the results of surveys with the Surveys module of their private classrooms.

But what if there is a standard end-of-course student evaluation used at your school that all teachers should deliver to their students? In that case, you as system administrator, should create the survey then make it "public" so it becomes available to all teachers within their view of the Surveys module.

OK, but the Surveys module gives the results directly back to the teacher. The rules at your school are that the results need to go directly to the department chair/dean/principal. The only way to do that is for you, as system administrator, to deliver the surveys to the required classrooms using the method shown in the tutorial (see *Delivering a survey* [65].) Links to the surveys will appear on each students "My Manhattan" login page, and the results will go back to you and not the teacher. While this might be manageable for a small number of classes, it certainly will keep you busy if you've got several hundred Manhattan classrooms to be surveyed! That's because for an end-of-course evaluation, you'll have to deliver a separate survey for each classroom.

Now suppose Prof. Einstein has developed a survey that would be great to make "Public" so all teachers can use it. You can enter his "Private" list of surveys, copy the survey back to your own "Private" list, then make it "Public."

Or suppose Prof. Einstein is having difficulty designing a survey on his own, because he doesn't know enough HTML to make the "Custom" tag instructions look nice. You can enter his "Private" list and make the modifications for him.

And so on. The overall picture is that we've tried to design lots of flexibility to enable the system administrator help teachers design and share surveys. The next section will give you an idea of how to do many of these things.

Questions and answers

1. How do I deliver a survey to one or more Manhattan classrooms?

This was covered in the tutorial - see *Delivering a survey* [65]. Note that when an administrator delivers a survey to one or more Manhattan classrooms, the survey appears on the “My Manhattan” login page for the users being surveyed. (In contrast, when a teacher delivers a survey within a classroom's Surveys module, it appears as a message within that module.)

2. How do I make a survey available for use by all teachers?

You can move any survey from your “Private” list into the “Public” list by first clicking on the “Public” link:

Career Services Survey	Preview	Modify	Copy	Info	Delete	Make Public
Meal Points Survey	Preview	Modify	Copy	Info	Delete	Make Public
Engr 102/103 Evaluation	Preview	Modify	Copy	Info	Delete	Make Public

Any survey that has been made Public can be used by any teacher in any of their classrooms. They'll see the survey listed under the “Public” heading after clicking the **New Item** button within the Surveys module. Teachers can choose to deliver the survey as-is, or can copy the survey to their Private list for modification. See *Teacher's Reference* for the full story on the teacher's perspective.

3. How can I survey people outside of the Manhattan environment?

lick on the “Deliver via web” link.

Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan	Deliver via web
Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan	Deliver via web
Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan	Deliver via web

The survey will be moved into the “Surveys delivered via the web” section of your survey listing, and you will get a page like this, which lists the URLs associated with the survey.

Attitudes on Group Projects has been made WWW accessible

Here's the URL for the survey. (Give it to those you'd like to survey.)

http://manhattan.wnec.edu/beta-bin/survey_www_survey?id=toucsniukyg3m4

Here's URLs that lead directly to the survey results:

http://manhattan.wnec.edu/beta-bin/survey_www_bar_graph?id=toucsniukyg3m4&pass=766815

http://manhattan.wnec.edu/beta-bin/survey_www_table?id=toucsniukyg3m4&pass=766815

Others can also visit these URLs and enter the access code when prompted, to see the results.

http://manhattan.wnec.edu/beta-bin/survey_www_bar_graph?id=toucsniukyg3m4

http://manhattan.wnec.edu/beta-bin/survey_www_table?id=toucsniukyg3m4

Access code for this survey: 766815

There are three sets of URLs. The first contains the URL for the survey itself. Email it to those you'd like to take the survey, or post it on a web page. (It's possible to create a web page at a “nicer” URL,

e.g. “http://manhattan.wnec.edu/computer_survey.html” that automatically redirects the user's browser to the cryptic URL provided by Manhattan.)

The second section of the above screenshot lists two URLs that lead directly to the bar graph results, and the raw data expressed as an HTML table, respectively. You can give these out to anyone who you would like to see the results of the survey as they come in.

The third section of the above screenshot lists two “password protected” (well, sort of) URLs that lead to the results. When the URLs are visited, the user will be prompted to enter the access code (766815 in the above screenshot) before they are shown the results.

You can see these URLs again by clicking on the “Access Codes” link associated with the survey as shown in the screenshot below. Also, note the “Graphs” and “Data” links in the screenshot. As a Manhattan administrator, you can get to the results of the survey via those command links - you do not have to visit the URLs discussed above, those are for people besides you.

Sponsor Evaluation of Students	Preview	Copy to your Private list	Info	Delete	Access Code
Attitudes on Group Projects	Preview	Copy to your Private list	Info	Delete	Access Code
Spring Event 2007	Preview	Copy to your Private list	Info	Delete	Access Code

You should note that because surveys delivered via the web are presented outside of the Manhattan environment, there is no authentication done, and thus there's no way to prevent the same person from taking the survey several times.

4. When I deliver a survey or make it “Public”, the survey is removed from my “Private” list. How can I keep a copy of these surveys in my “Private” list?

Whenever you click on a **Copy to your private list** link associated with any survey, a copy of that survey is immediately added to your “Private” list, where it can be modified, or delivered again:

Peer Advisor Survey	Preview	Copy to your Private list	Info
Quality of Life Survey - Commuter	Preview	Copy to your Private list	Info
LBC Survey	Preview	Copy to your Private list	Info

5. How can I make a survey I designed available to just one teacher, so he can deliver it within the Surveys modules of his classrooms?

Design the survey - it will appear in your “Private” list. Click **Finalize** when the survey is the way you want it, then click **Give a copy to...**:

Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan
Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan
Info	Delete	Make Public	Give a copy to...	Deliver via Manhattan

Next, enter the username of the teacher in the text box, and click the **Submit** button:

Survey title: End of Term Attitudes on Group Projects

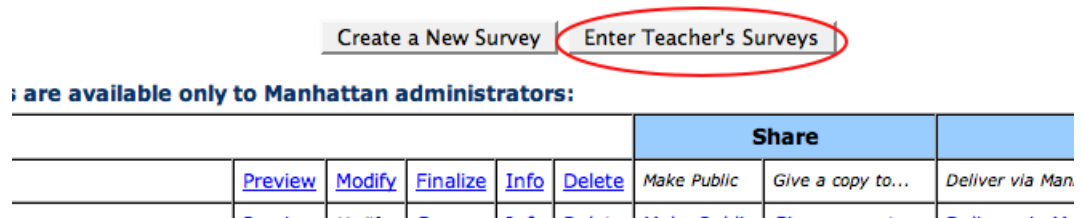
Give survey to teacher w/username: ae4322
Cancel Submit

The survey will be copied from your Private list into the Private list of the selected teacher. That teacher will see the survey in the Surveys module of all of their courses. They can deliver it as-is,

make copies of it, and modify it further. You can enter the Private list of the teacher to convince yourself that it has been copied successfully - see the next question.

6. How can I take a copy of a teacher's Private surveys?

Click on the **Enter Teacher's Surveys** button at the top of your list of surveys:



This leads to a page listing all of the users on the system who have created Private surveys. Click on a name to enter their survey list:

Manage Teacher's Surveys

Fri 02/13/2004 09:40 AM

[Albert Einstein \(ae4322\)](#)

[Julius Jenkins \(jj4321\)](#)

After selecting a teacher, you will be looking at the same list of surveys that teacher sees from within the Surveys module of any of their classrooms. Click the "Copy to Admin's Private list" link to get a copy of one of the teacher's private surveys::



The ability to move surveys between any teacher's private list and your private list opens up a lot of possibilities. Although there are sometimes several steps involved, you can:

- Design a survey and give it to one or more teachers.
- Take a survey designed by a teacher and make it Public so it can be used by any teacher.
- Take a survey designed by a teacher and copy it to one or more other teacher's private lists.
- Take a survey designed by a teacher and deliver it on the web outside the Manhattan environment.
- And so on...

7. How can I see the results of surveys a teacher has issued to his class within that class' Surveys module?

You can't. Or rather, you shouldn't. Surveys issued by a teacher within the Surveys module of a classroom are intended to be private. Of course, you can always ask the teacher for his password and login to get the results. Another approach is to temporarily add yourself as a second teacher to the classroom (hopefully with that teacher's permission), get the survey results, and remove yourself as a teacher - see *Add a teacher* [117]

8. Does this work with “Standalone” courses?

Mostly, no. Since a standalone classroom (see *Normal, Template and Standalone Courses* [22]) allows the teacher to invent users, it would be possible for a person to create a teacher that has the same username as an existing teacher using the “normal”, central login system. That fictitious teacher would have full access to the real teacher's surveys if they were linked together, so they are not.

A teacher of a standalone course can not create their own surveys, and therefore you, as system administrator cannot give them Private surveys or enter their private list of surveys. They don't have a private list of surveys! However teachers of standalone courses DO have access to any surveys you have made Public. Thus the only thing a standalone teacher can do is issue a Public survey to their students, without modification.

9. Where are the surveys stored on the server?

Surveys are stored as XML files within subdirectories of the `surveys` directory of your Manhattan installation. You'll find a `surveys/(username)` directory for each teacher who has created a Private survey. The `surveys/_admin_` directory holds the system administrator's surveys (and their results if delivered.) The `surveys/_www_` directory holds any surveys (and their results) that were made “WWW accessible” by the administrator.

Note that the results for surveys delivered by a Manhattan administrator are stored within the main `surveys` directory of your installation as described above. However, when a teacher delivers a survey using the Surveys module within a classroom, a copy of the survey (and the results) is written to that course's directory on the server. Look in, for example, `../courses/try/phy10101/people/ae4322/surveys` for any surveys delivered by our Prof. Einstein within his Physics course used in the tutorial.

You should not mess with these files outside of the web-based administration tools!

10. What are issues that I should be aware of?

With any system like Manhattan's surveys there's always room for improvement. The “known issues” that you should be aware of involve copying a survey to a Manhattan user that is not a teacher, and removing Private surveys that belonged to users who no longer have a Manhattan account.

When you “Give survey to teacher w/username:”, Manhattan doesn't check to see if the username you entered actually belongs to a teacher. That's because Manhattan doesn't make a firm distinction between whether a user is a student or a teacher. A person can be a student in some classes and a teacher in others. If a user doesn't happen to have any classes that he teaches at the moment, well he's not a teacher. The bottom line is that you might accidentally copy the survey to the Private list of a student (a person who is not teaching any Manhattan classes at the moment.) There's really no harm done, since that student can't possibly see or issue any surveys until he becomes a teacher. You can “Enter Teacher's Surveys” to delete the survey(s) you have copied if you realize your mistake.

You should also know that a user's list of private surveys is NOT deleted when that user account is completely removed from Manhattan. If you again add a user with the same username, he'll get all of his surveys back. For this version of Manhattan at least, if you want to delete all of the private surveys created by a user that has been completely removed from the system, you'll have to use the "Enter Teacher's Surveys" feature and manually delete them one-by-one

Appendix A. How to translate Manhattan

Introduction

This appendix explains how to translate Manhattan into a language other than English.

Translating Manhattan involves using a text editor to change a number of simple text files, which contain the English language strings used on Manhattan's web pages. Translating Manhattan also requires creating a new set of graphic buttons.

The following instructions will assume Manhattan has been installed in the directory called /home/manhat/manhat-x.x.x where x.x.x is Manhattan's version number.

Example: the 'doorstep' program

The simple program named doorstep is a good one to use as an example. Use a web browser to visit:

`http://YOURSERVER.edu/manhat-bin/doorstep`

The doorstep program simply displays the form that's used to login to Manhattan:



Next, use your web browser's View source command to view the underlying HTML that makes up the doorstep web page. (If you're using Firefox, select **View --> Source** from the menu.) The top of the HTML source contains some comments useful for translators:

```
<!--
  Manhattan version      : Version 3.2
  Program Name           : /beta-bin/doorstep
  ClearSilver Template:  ../images/theme_0/doorstep.cs
  Language File(s)       : ../images/theme_0/lang/doorstep.lang
-->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
<head>
<title>
  My Manhattan Login
</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="StyleSheet" href="/beta-images/theme_0/common.css" type="text/css" />
<script type="text/javascript" src="/beta-images/theme_0/form_validation.js"></script>
```

The HTML source for each Manhattan page starts with this information.

There are two items of interest to anyone interested in changing the look or language of Manhattan. First there's the line that reads:

ClearSilver Template: ../images/theme_0/doorstep.cs

Manhattan makes use of the ClearSilver HTML templating system. The **ClearSilver Template** is a text file that contains a mixture of HTML and ClearSilver templating language statements. You can change the entire look of the doorstep program by modifying its ClearSilver template file. To learn more, visit <http://www.clearsilver.net>

However, you don't need to edit the ClearSilver template to translate Manhattan into another language. Instead, you edit the **Language File(s)** associated with the web page. For the **doorstep** program, the language file is listed as:

Language File(s) : ../images/theme_0/lang/doorstep.lang

From a shell account change to the **images/theme_0** directory of your Manhattan installation. If you list the files in this directory, you'll find over 100 files with a **.cs** extension. These are the ClearSilver template files for Manhattan. The **.gif** and possibly **.jpg** and **.png** files you'll find in this directory are the images used within Manhattan. Since many of these images are buttons with English language phrases, you'll want to use a graphics program to edit them to suit your language.

Now change to the **lang** subdirectory. The full path will be something like: **/home/manhat/manhat-x.x.x/images/theme_0/lang**. Here you'll find what you're looking for - the text files, all of which end in **.lang**, that contain the English language phrases used throughout Manhattan. In particular, the **doorstep.lang** file contains the text used by the **doorstep** program:

```
lang.doorstep.username_prompt = Username :
lang.doorstep.username_prompt_help = Enter your username
lang.doorstep.password_prompt = Password :
lang.doorstep.password_prompt_help = Enter your password
lang.doorstep.submit_button = Enter Manhattan
lang.doorstep.submit_button_help = Enter Manhattan
lang.doorstep.mymanhattan = My Manhattan Login
lang.doorstep.super_login = Administrator's Login
lang.doorstep.super_submit = Submit
lang.doorstep.super_submit_help = Submit
lang.doorstep.warn_msg = Username and password fields cannot be left blank.
lang.doorstep.copyright = The Manhattan Virtual Classroom system...
```

A **.lang** text file consists of name/value pairs separated by an equal sign. For the first line above the name of the variable is **lang.doorstep.username_prompt** its value is currently **Username** :. Translating Manhattan involves changing the values for each item in all of the language files. For example, to translate the **doorstep** program into Spanish, you might change the file to read:

```
lang.doorstep.username_prompt = Usuario :
lang.doorstep.username_prompt_help = Ingrese su nombre de Usuario
lang.doorstep.password_prompt = Clave :
lang.doorstep.password_prompt_help = Ingrese su clave
lang.doorstep.submit_button = Ingresar
lang.doorstep.submit_button_help = Ingresar al Salón Virtual
lang.doorstep.mymanhattan = Ingreso a Mi Salón Virtual
lang.doorstep.super_login = Ingreso de Administrador
lang.doorstep.super_submit = Enviar
lang.doorstep.super_submit_help = Enviar
lang.doorstep.warn_msg = Campos de Usuario y Clave no se pueden dejar...
lang.doorstep.copyright = El sistema de Salón Virtual Manhattan...
```

The changes take effect the moment you save the `.lang` file. With the above changes, the doorstep page looks like this:

Ingreso a Mi Salón Virtual

Manhattan
virtual classroom

*El sistema de Salón Virtual Manhattan
tiene derecho de autor © 1997-2006 [Steve Narmontas](#)
y es distribuido bajo los términos de la
[Licencia Pública General de GNU](#) desde su página principal en
manhattan.sourceforge.net*

Version 3.2

Usuario :
Clave :

Things to note

Here's some to watch for as you translate Manhattan.

Date and time formats

The way dates and times are displayed within Manhattan is controlled by entries within the `custom.h` file found in the `src` directory. You should edit that file to change the way dates and times are displayed. If you do change `custom.h`, you must recompile Manhattan by typing **make clean ; make install** for the changes to take effect.

Some lang files affect more than one program

Usually, each Manhattan program has its own `.lang` file. However, that's not always the case. For example, the `doorstep.lang` file, shown in the previous section, is used by both the `doorstep` and the `super_doorstep` programs. (The `super_doorstep` program is used to enter the web-based administrative system.) After changing the `doorstep.lang` program, visit:

`http://YOURSERVER.edu/manhat-sbin/super_doorstep`

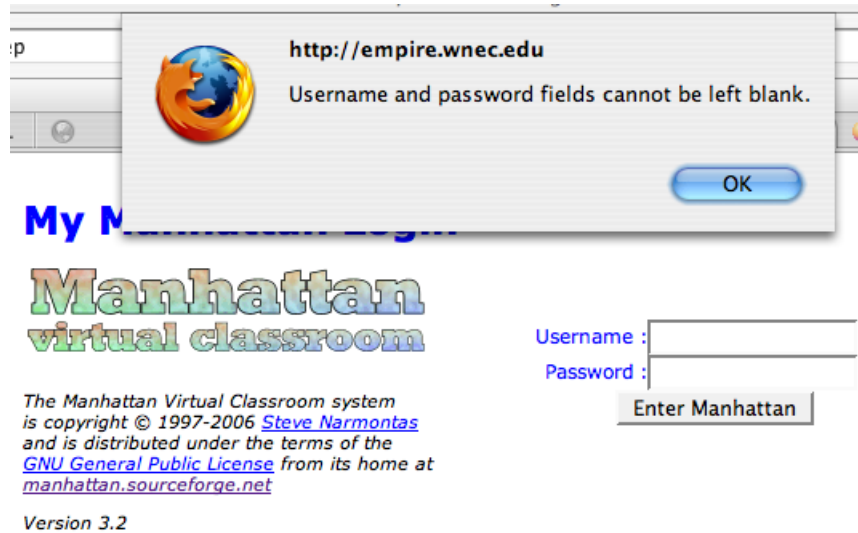
and you'll see your language changes appear there as well. If you use your web browser's **View Source** you'll see that `doorstep.lang` is correctly listed as the language file for the `super_doorstep` program.

Watch out for JavaScript!

Some of the English language phrases you'll be translating are used in JavaScript scripts on the web page. An example in the `doorstep.lang` file is:

```
lang.doorstep.warn_msg = Username and password fields cannot be left blank.
```

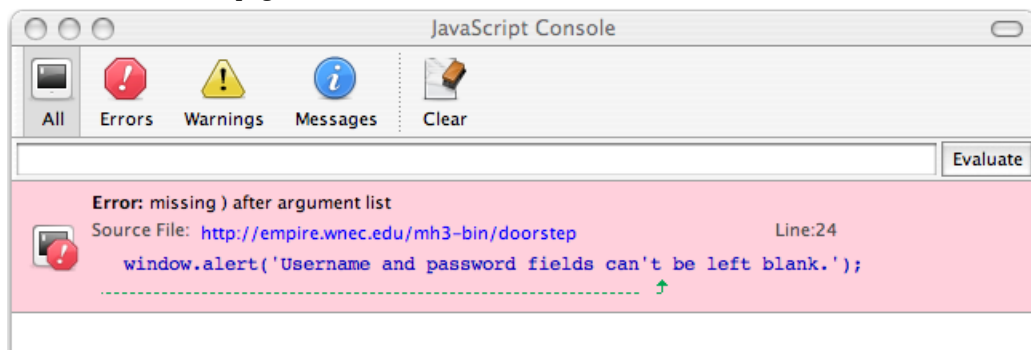
This string is shown to users within a JavaScript alert, when they click the Enter Manhattan button without entering both a username and a password:



Because the text is used within a JavaScript, you can't include certain characters. In particular, you must not use any form of quotes, or apostrophe's. As an example, suppose the word cannot is changed to can't in the `doorstep.lang` file:

```
lang.doorstep.warn_msg = Username and password fields can't be left blank.
```

Using Firefox, the page seems to simply skip the JavaScript alert when a user clicks the Enter Manhattan button without entering both a username and password. (The user's mistake is instead reported by the program that logs the user into Manhattan.) However, Firefox has a **JavaScript Console** found under its **Tools** that shows for this page:



The apostrophe in the word can't caused a syntax error in the JavaScript that is supposed to make sure the user has at least entered something into the Username and Password fields. Other browsers will show the error differently, perhaps with an "Error in page." message in the status bar at the bottom of the screen.

In this case the error is not serious and most users will never notice there's a problem. In a few rare situations, odd characters in strings used within JavaScripts can keep that particular program from working properly at all.

When we realized this might cause a problem for translators, we began including `javascript` as part of the name of an item within the lang files. For example the file `admin_module_form.lang` contains the entry:

```
lang.admin_module_form.javascript_submit_confirm = Save the choices made in all fo
```

Since the name of the variable (on the left of the = sign) contains the word `javascript`, you know the value (on the right of the = sign) must not contain quotes or apostrophe's.

Unfortunately, this potential problem did not occur to us until late in the development process, and there are a number of phrases used in JavaScripts that are not labeled as such. If you can't see a phrase you're translating on the web page, then it may be used within a JavaScript for that page. The easiest way to find these phrases is to View the HTML source for the page, and find the phrase you're looking for.

Other things to translate

In addition to changing the `.lang` files, a complete translation will include modifying the image files that contain English language phrases. These `.gif` files are stored in the `images/theme_0` of your Manhattan installation.

Also, there's a `help` directory found as `images/theme_0/help` that contains a number of `html` files. These `html` files are used to display context-sensitive help to users in areas of Manhattan such as the web-based grades module.

Keeping up with new versions

New versions of Manhattan are released periodically. New versions may have changed old `.lang` files or added new ones. A utility called `lang_diff` can be used to list the differences in the `.lang` files between two releases of Manhattan. Before you use the program, you must compile it.



The `lang_diff` utility may not have been included in all releases of Manhattan earlier than version 3.3. If you can't find it in your `src` directory, write me and I'll email you a copy.

Change to the `src` directory of your installation and type the command `make lang_diff`:

```
cd manhat-x.x.x/src
make lang_diff
```

The executable program is created in the `src` directory. You can move it anywhere you wish. If you run the program with no arguments, you'll get some usage information:

```
src> ./lang_diff
```

```
This program lists differences in the 'lang' files
between two releases of Manhattan.
```

```
It's useful for people who maintain translations.
```

```
Usage: ./lang_diff OLD_LANG_DIR NEW_LANG_DIR
```

```
Where:
```

```
    OLD_LANG_DIR is the FULL PATH to the 'lang' directory
                  of the older release of Manhattan
    NEW_LANG_DIR is the FULL PATH to the 'lang' directory
                  of the newer release of Manhattan
```

As an example, suppose you want to list the differences in the `.lang` files between versions 3.1 and 3.2 of Manhattan. You happen to have the lang files for version 3.1 installed in the directory: `/home/manhat/manhat-3.1.0/images/theme_0/lang` and the lang files for version 3.2 are in `/home/manhat/manhat-3.2.0/images/theme_0/lang`. You can write a report listing the language differences to a file name `differences.txt` by typing the following command ALL ON ONE LINE:

```
./lang_diff /home/manhat/manhat-3.1.0/images/theme_0/lang
/home/manhat/manhat-3.2.0/images/theme_0/lang
>differences.txt
```

You can then open the `differences.txt` file in a text editor to view the report:

```
Comparing 'Old' and 'New' Manhattan 'lang' files.
'Old' lang directory:
    /home/manhat/compare/manhat-3.1.0/images/theme_0/lang
    (found 156 .lang files)
'New' lang directory:
    /home/manhat/compare/manhat-3.2.0/images/theme_0/lang
    (found 164 .lang files)

+++++
The following lang file(s) are new.
They appear in the NEW PATH /home/manhat/manhat-3.2.0...
but not in the OLD PATH /home/manhat/manhat-3.1.0/ima...
+++++
grade_book_final.lang
main_menu_required_survey.lang
grade_book_release.lang
grade_book_default_letter_grades.lang
grade_book_categories.lang
super_auto_add_faculty.lang
grade_book_export.lang
grade_book_enter_grades.lang
grade_book_items.lang

etc ...
```

As shown above, the report first lists any `.lang` files that are entirely new. This is followed by a list of `.lang` files that are no longer used. For each of the remaining files, NEW ITEMS, OBSOLETE (no longer used) ITEMS, and CHANGED ITEMS are listed.

Appendix B. GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover,

and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the *Addendum* below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher

of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in *section 4* above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.