

SFU

SIMON FRASER UNIVERSITY Surrey
DEPARTMENT OF PHYSICS

SFU SURREY

Physics Studio

Technical Manual

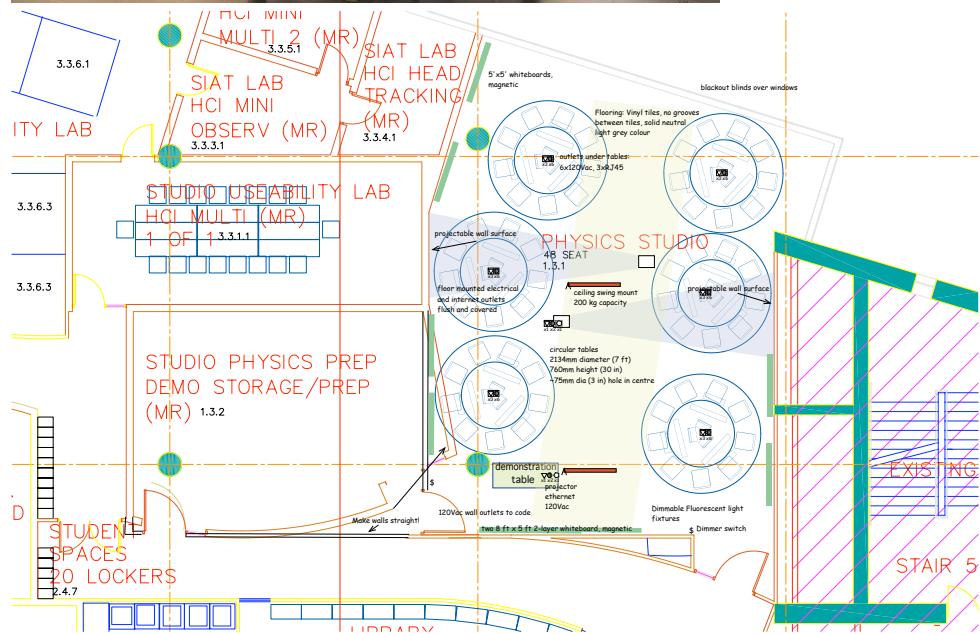
by
Neil Alberding

2008 July 24, Thursday

| | |
|--|-----------|
| Studio Physics Equipment | 1 |
| <i>Studio Room Layout and Features</i> | 1 |
| <i>Ceiling Support Bars</i> | 2 |
| <i>Instructor's Desk: Computer and A/V connections</i> | 3 |
| WetCT and WebDAV | 5 |
| <i>Uses of WebCT</i> | 5 |
| <i>Gradebook</i> | 6 |
| <i>WebDav</i> | 8 |
| <i>The WebDAV dialog from Hell (Mac only)</i> | 10 |
| <i>Using CGI scripts to collect and publish class data</i> | 10 |
| Unit 1: Introduction and Computing | 17 |
| <i>Session 1: Data Collection and Spreadsheet Use</i> | 17 |
| <i>Session 2: Measuring and Graphing Horizontal Motion</i> | 18 |
| Unit 2: Measurement and Uncertainty | 18 |
| <i>Session 1: Direct and Indirect Measurements</i> | 18 |
| <i>Session 2: Random and Systematic Variation</i> | 18 |

Studio Physics Equipment

Studio Room Layout and Features



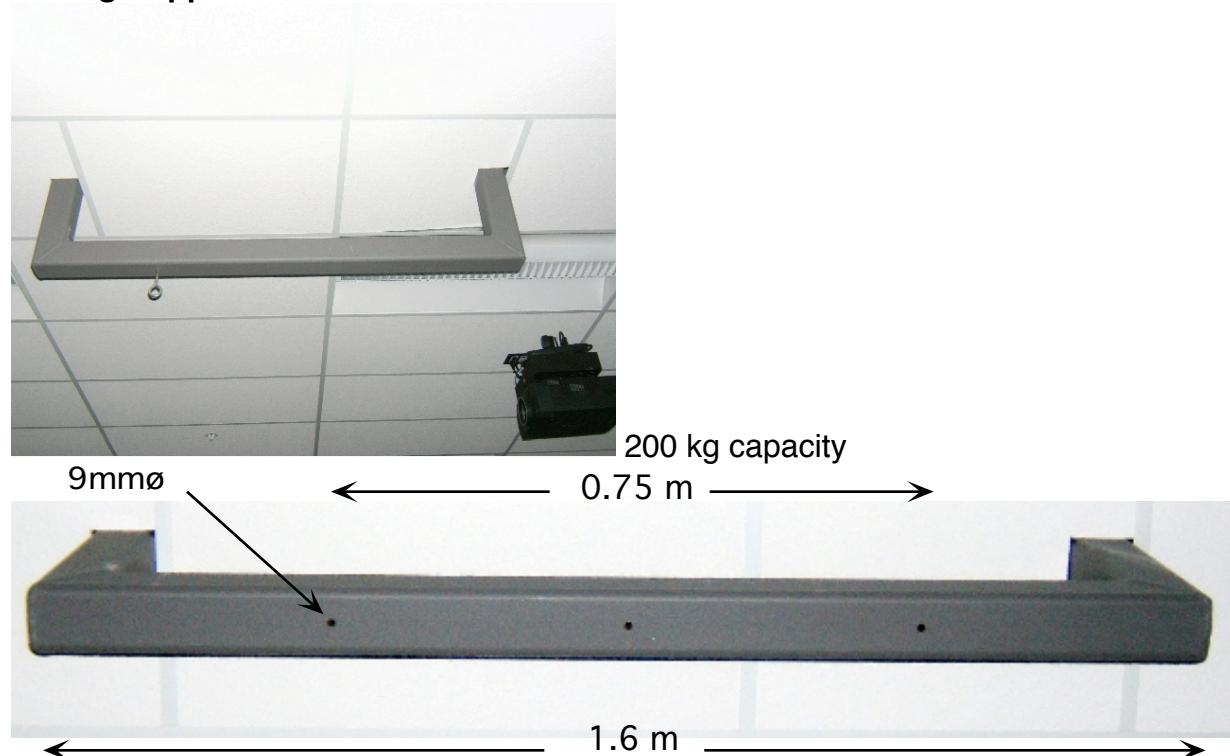
The studio room is approximately 144 square metres (1550 square feet) The front wall has two 8-ft double layered sliding white boards. An instructor's table with an intel iMac is in the front. The ceiling fluorescent lights can be dimmed by a controller near the students' entrance.

In the studio are six student tables. Each table is seven feet in diameter and can seat up to nine students in groups of three. Each group of three has a complete set of equipment for the activities. There is a set of permanent apparatus on each table as follows:

1. Mac mini (most have ppc processor, two are intel)
2. 17" Dell monitor with 4 usb ports (on side and bottom)
3. Power bar (8 outlets, 3 of which are widely space for wall-warts.)
4. LabPro interface
5. Mac wired keyboard
6. Microsoft optical mouse with wheel
7. Keyboard tray

In addition to the permanent equipment on the table each station has a rolling drawer tower for each course (Phys140 and Phys141) containing other equipment used in those courses.

Ceiling Support Bars



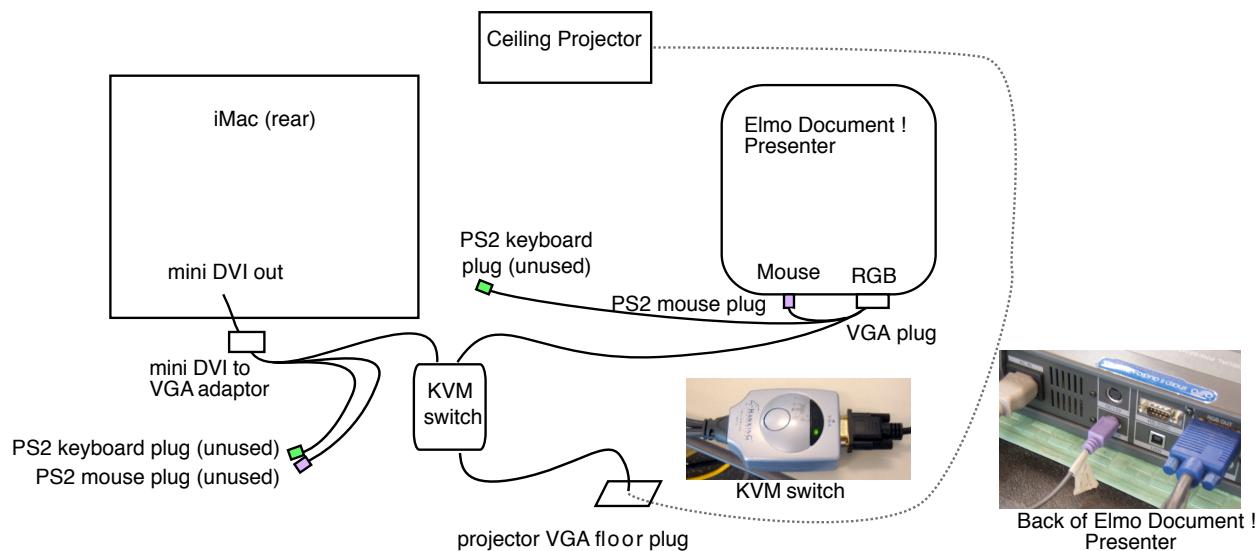
There are two bars on the ceiling for hanging demonstrations such as pendula, swings, spinning bicycle wheels, etc. The load limit is 200 kg (at least that's what was

requested). The dimensions are 1.6 m long, and about 95 mm square. The three vertical holes have a 9 mm diameter and are 375 mm apart (centre-to-centre). A 3/8" x 8" eyebolt can be inserted through the holes to suspend ropes as shown in the picture..

Instructor's Desk: Computer and A/V connections

The equipment at the instructor's desk includes one computer (intel iMac) and an Elmo Document Presenter. Both the computer and the Elmo are connected to the room's projectors through a KVM switch which plugs into the VGA socket in the floor. The switch chooses between the iMac video and the Elmo video by pressing the button.

The interconnections among these items are illustrated in the following diagram:



The miniDVI output on the back of the iMac must be connected to a miniDVI to VGA adapter.

The miniDVI to VGA adapter is plugged into one input of the KVM switch.

The RGB output of the Elmo is plugged into the other input of the KVM switch.

The PS2 mouse plug of the KVM switch must be plugged into the mouse input of the Elmo in order to provide the KVM switch's power.

The keyboard PS2 plug is not used.

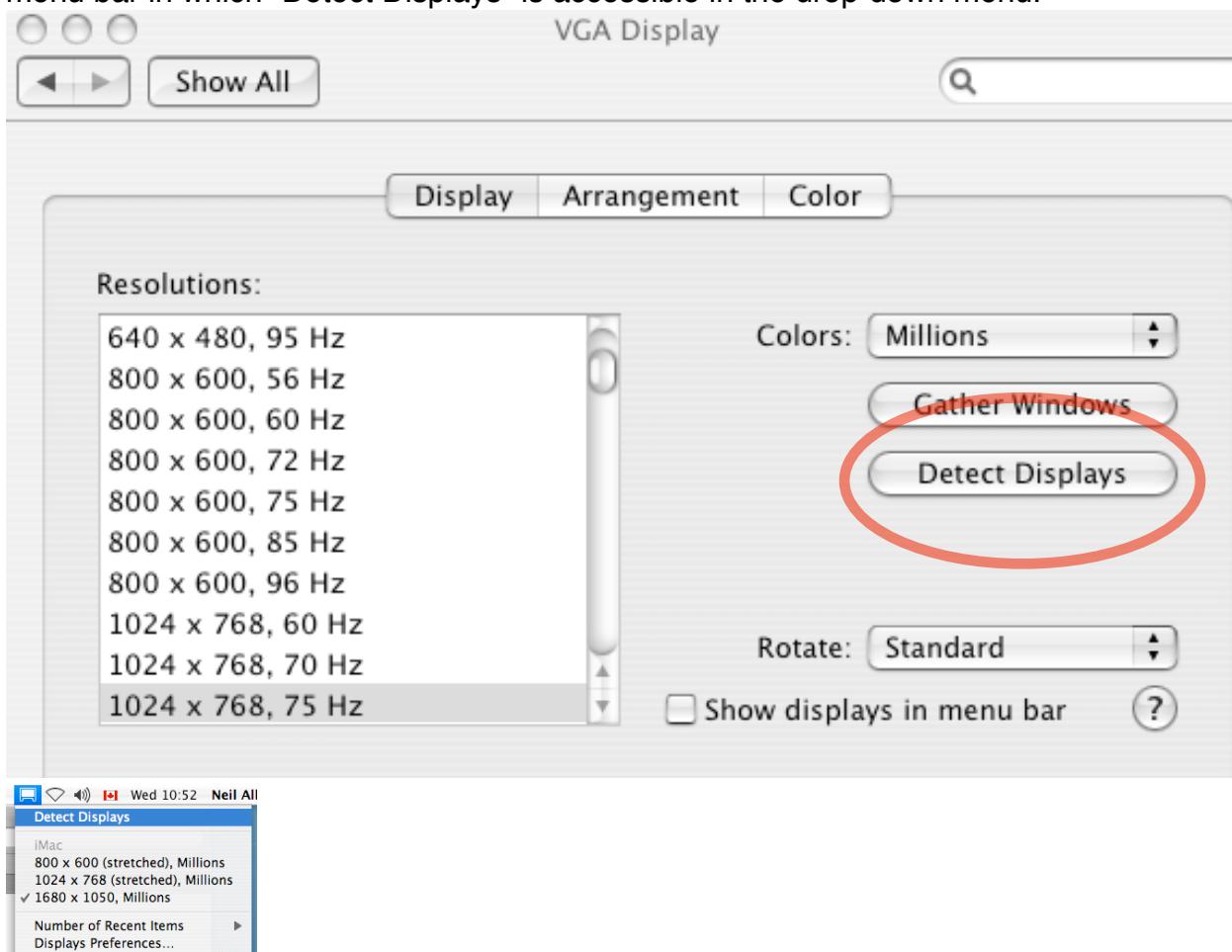
The output of the KVM switch is then plugged into the VGA jack in the floor which feeds the overhead projectors. A separate VGA-VGA cable is necessary here.

There are two projector input locations in the floor. One is in the front of the room near the whiteboards. The other is near the middle of the room to be used when the

instructor wishes to teach from a more central position in the room. These inputs are selected by switching between input1 and input2 in the projectors' input menu.

When the projectors are turned on they usually adjust to the iMac's screen resolution (or vice-versa) only if the Elmo is powered on and the KVM switch connects the projector to the iMac. If you find that the iMac display is not being projected, you get a blue screen when you switch from Elmo to iMac, then you should open the "Displays" control panel and press the "Detect Displays" button. This should result in the iMac screen and the projector establishing communication.

You should also check that the "Mirror" mode is selected under the "Arrangement" tab. The check box "Show displays in menu bar" allows you to put a Display icon in the menu bar in which "Detect Displays" is accessible in the drop-down menu.



WetCT and WebDAV

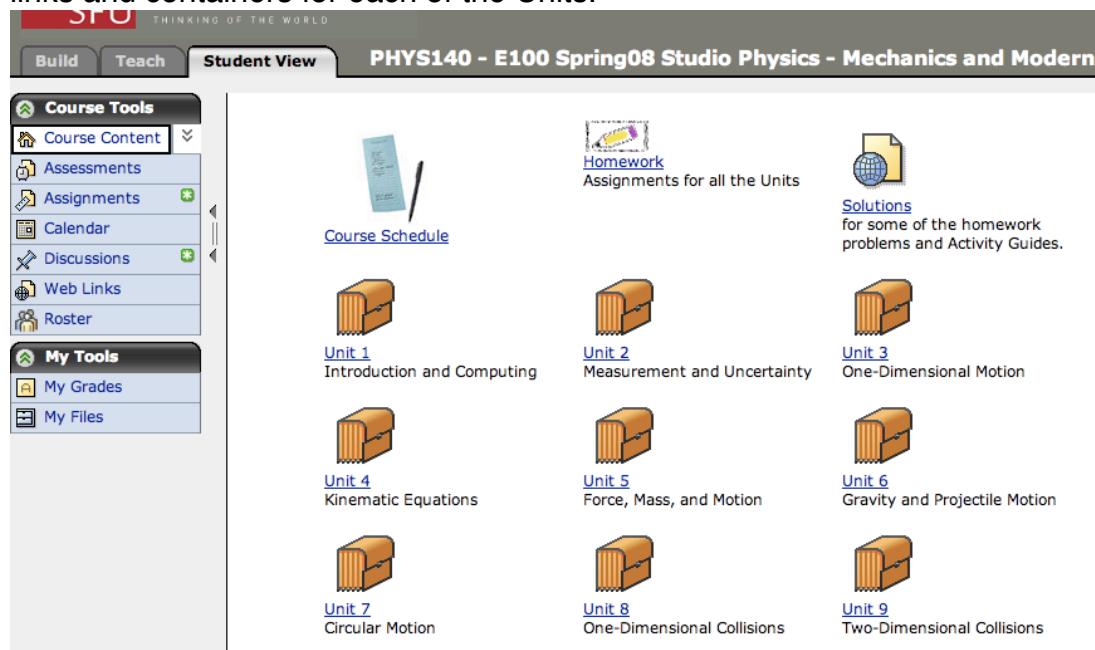
Uses of WebCT

A principle organizing tool for the course is WebCT. WebCT provides the following functions

- Gradebook
- Links to Activity Guides and other handouts in pdf format
- Course Schedule
- Facility to upload computer output for assignments such as spreadsheets and computer-generated graphs
- Links to homework solutions
- links to CGI scripts for collecting class data
- Links to external sites of interest

Registered students are automatically entered into the class roster and can access the Physics 140 WebCT by using their university computer ID and password. Students who need to participate before being officially registered can be entered manually as an Auditor via the gradebook. If they are later officially registered their “role” will be changed from auditor to student. TAs need to be entered manually in the gradebook as well.

The main page contains the class schedule, link to homework, solutions and outside links and containers for each of the Units.



The screenshot shows the WebCT student view for the course PHYS140 - E100 Spring08 Studio Physics - Mechanics and Modern. The top navigation bar includes 'Build', 'Teach', and 'Student View'. The main content area is titled 'PHYS140 - E100 Spring08 Studio Physics - Mechanics and Modern'. On the left, there are two vertical menus: 'Course Tools' (Course Content, Assessments, Assignments, Calendar, Discussions, Web Links, Roster) and 'My Tools' (My Grades, My Files). The main area displays a grid of nine units, each represented by a folder icon and a title. The units are: Course Schedule, Assignments for all the Units, Solutions for some of the homework problems and Activity Guides; Unit 1: Introduction and Computing, Unit 2: Measurement and Uncertainty, Unit 3: One-Dimensional Motion; Unit 4: Kinematic Equations, Unit 5: Force, Mass, and Motion, Unit 6: Gravity and Projectile Motion; Unit 7: Circular Motion, Unit 8: One-Dimensional Collisions, Unit 9: Two-Dimensional Collisions.

Each Unit container has a list of tasks for that unit and also contains links pertaining to the particular unit. The following figure shows the Unit 1 contents.

Your location: Home Page > Unit 1

What to do before Session 2:

- Read the section in the front of the Activity Guide entitled [Workshop Physics Philosophy, Policies, and Procedures](#) and come to class prepared to ask any questions you might have.
- Learn a few letters in the Greek Alphabet. Beware! There may be a quiz! Physicists and mathematicians tend to run out of symbols to use in mathematical equations. You should memorize some of the most common letters borrowed from the Greek alphabet that are used in physics equations. These are shown below.
- Do Problem SP1-1 (under "Homework" below) for more spreadsheet practice and submit it to WebCT.

| Letter | Capital | Lower Case |
|---------|-----------|---------------|
| Alpha | A | α |
| Beta | B | β |
| Gamma | Γ | γ |
| Delta | Δ | δ |
| Epsilon | E | ε |
| Theta | Θ | θ |
| Lambda | Λ | λ |
| Pi | Π | π |

 [Philosophy, Policies and Procedures](#)

 [Unit 1 Activity Guide](#)

 [Appendix A](#)
Using Excel

 [Appendix E 1](#)
Mathematical Modelling to Fit Data

 [Submit height data](#)

 [Get Class Data](#)
View the data submitted by all the class members

 [Modelling Worksheet](#)

 [Modelling Tutorial](#)

 [Submit Bowling Data](#)
Submit your group's time for the bowling run.

 [Get Bowling Data](#)
Retrieve all the data submitted for the bowling activity.

 [Modeling Worksheet for Open Office](#)
An open document format file that works with free Open Office

 [Modeling Tutorial for Open Office](#)
This is an ods file which works with the free office suite Open Office

Gradebook

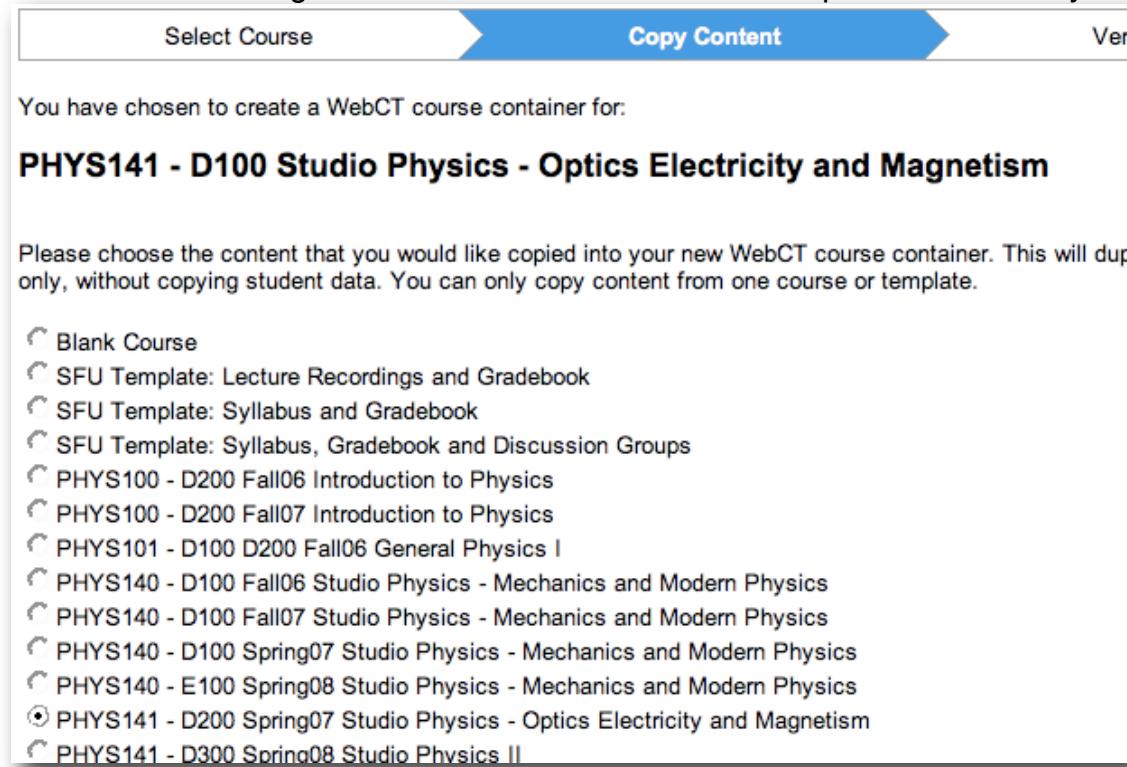
Grades in the gradebook can be entered individually using a web browser. For large classes it is convenient to upload grades from a spreadsheet. Grades for a single item or multiple items can be uploaded in one fell swoop. The file of grades to be uploaded should be exported from the spreadsheet in csv (comma-separated values) format. It must contain a column of the students "User ID"s (the students' e-mail address without the "@sfu.ca"). The other columns should contain the grades to be uploaded. It is advisable to create the columns in the gradebook before upload. If not, then the column will be created as text, not numeric. The "out of" value has to be entered separately in the gradebook via "column options". Each column in the file to be uploaded must have a header with the same column name as it has in the gradebook.

Here is an example of the first few lines of such a file. (the quotation marks are optional unless commas need to be included in the uploaded text.).

```
User ID,Midterm 2
"yma14","36"
"aia9","32"
"tna8","21"
"tsb1","22"
"jmb13","27"
"pwc6","29"
"amc20","28"
"acc25","22"
```

Links to pdf files can be inserted onto the WebCT page via the WebLink option. I usually prefer to link to files on the webdav server rather than upload them to webct. This makes it easier to revise handout by merely replacing the file on the webdav server. (It is possible to mount the WebCT file space as a webdav server too, but the name is somewhat unwieldy.) Another advantage of using an external file space is that the locations of these files remain the same from semester to semester.

Every semester a WebCT “container” is created for each section. If one instructor is teaching multiple sections it might be more convenient to create one container for those multiple sections. When requesting the container via the course request page, <http://toolkit.sfu.ca/webct/courseRequest/index.cgi>. The new request should specify that the content be copied from a previous course container. Then all one has to do is update changes for the new semester. In particular, any computer assignment submission dates must be changes and well as dates for the on-line quizzes and surveys.



The screenshot shows a step in the SFU Toolkit course request process. The top navigation bar includes 'Select Course', 'Copy Content' (which is the active step, highlighted in blue), and 'Verify'. The main content area displays the message: 'You have chosen to create a WebCT course container for: **PHYS141 - D100 Studio Physics - Optics Electricity and Magnetism**'. Below this, instructions say: 'Please choose the content that you would like copied into your new WebCT course container. This will duplicate the content only, without copying student data. You can only copy content from one course or template.' A list of available courses to copy from is shown, with the last item, 'PHYS141 - D200 Spring07 Studio Physics - Optics Electricity and Magnetism', selected (indicated by a radio button with a dot).

- Blank Course
- SFU Template: Lecture Recordings and Gradebook
- SFU Template: Syllabus and Gradebook
- SFU Template: Syllabus, Gradebook and Discussion Groups
- PHYS100 - D200 Fall06 Introduction to Physics
- PHYS100 - D200 Fall07 Introduction to Physics
- PHYS101 - D100 D200 Fall06 General Physics I
- PHYS140 - D100 Fall06 Studio Physics - Mechanics and Modern Physics
- PHYS140 - D100 Fall07 Studio Physics - Mechanics and Modern Physics
- PHYS140 - D100 Spring07 Studio Physics - Mechanics and Modern Physics
- PHYS140 - E100 Spring08 Studio Physics - Mechanics and Modern Physics
- PHYS141 - D200 Spring07 Studio Physics - Optics Electricity and Magnetism
- PHYS141 - D300 Spring08 Studio Physics II

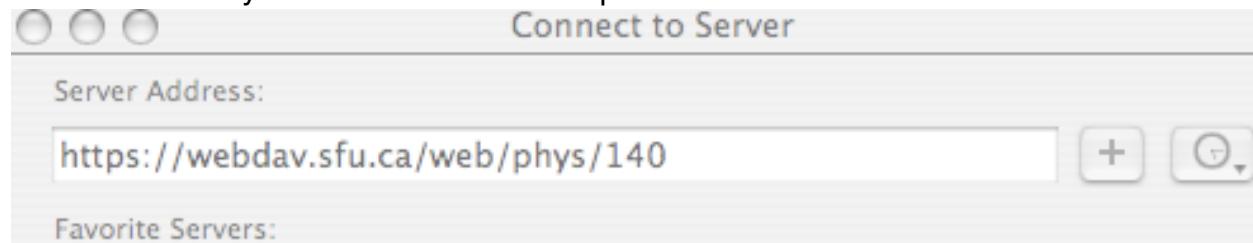
WebDav

WebDAV is a file-sharing protocol available through the webserver. WebDAV directories can be mounted on your computer's desktop and can then be accessed as if they were external drives on your own computer. Furthermore, the files on the WebDAV server can be accessed read-only via a web browser. Files coded in html will be displayed as webpages when accessed by a web browser. Normally WebDAV directories are accessible to everyone without restrictions. However, it is possible to install password protection to secure the information in the directories.

For example, the root directory for Physics 140 is accessed by the url “<http://www.sfu.ca/phys/140>”. Because this directory contains an index.html file, it is displayed. Subdirectories which do not contain an index.html file will display as a file listing.

This same directory can be mounted on a personal computer file system via the url <https://webdav.sfu.ca/web/phys/140>

The procedures for mounting vary for macs and pc's running Windows or Linux. On a mac you choose “Connect to server” from the finder's “Go” menu, Then enter in the url and then your SFU username and pssword.



On Windows you go to “My Network Places” and navigate the “Add new network place” wizard. On Windows you usually have to re-enter username and password three times before access is granted.

Access is controlled by two files named “DONT_DELETE_ACCESS.txt” and “DONT_DELETE_USERS.txt”. It's important not to disturb these files unless you really know what you're doing because any mistake will completely lock-out anybody's access until you get ACS to fix your mistake. For example you can add user id's to the “DONT_DELETE_USERS.txt” file using the mac editor Textwrangler (or BBEdit, but for some reason, using another editor causes the file to be deleted before it is saved and locks out out).

Read-only access through a web-browser is similary controlled by the invisible files .htaccess and .htpasswd. These file have the same format as the “DONT_DELETE...” files but only affect access via a web browser.

If no access files exist in subdirectories, the subdirectory inherits the access privileges of its mother. This goes for read/write access (DONT_DELETE_ACCESS.txt) as well as read-only access (.htaccess).

My practice is to allow universal access to all subdirectories restrict specific directories as needed. For example every semester I create a subdirectory for items specific to that semester. Eg, <http://www.sfu.ca/phys/140/1081> contains data for the spring semester of 2001. Every semester I create a “secure directory” for use by the instructors and is not permitted to students. Solutions to homework etc are restricted to class members and instructors.

Here is an example of the .htaccess and .htpasswd files to restrict access to the subdirectory “extras” in a Physics 120 class.

```
.htaccess
AuthType Basic
AuthName "Phys120"
AuthUserFile /web/phys/120/051/extras/.htpasswd
require valid-user
```

```
.htpasswd
+mrastan
+rudd
+!phys-120-d200
+!phys-fac
```

Here the access is restricted to users mrastan, rudd and anyone on the mailing lists phys-120-d200 and phys-fac.

Here is an example of restricting access to the subdirectory 1081_secure to instructors.

```
.htaccess
AuthType Basic
AuthName "Phys140"
AuthUserFile /web/phys/140/1081_secure/.htpasswd
require valid-user
```

```
..htpasswd.
+mrastan
+sjohnson
+rudd
+exafs
+kchanv
```

The following files control read-write access to the root /phys/140 directory. Most subdirectories inherit these privileges unless they contain their own “DONT_DELETE_ACCESS.txt” files.

DONT_DELETE_ACCESS.txt

```
AuthType Basic
AuthName "Phys140"
AuthUserFile /web/phys/140/DONT_DELETE_USERS.txt
require valid-user
```

DONT_DELETE_USERS.txt

```
+!webdav-admin
+mrastan
+rudd
+sjohnson
+exafs
+kchanv
+kurn
```

The WebDAV dialog from Hell (Mac only)

Occasionally one attempts to access a WebDAV directory for which one doesn't have permission to access. Then the authentication dialogue box keeps attempting to authenticate and on Mac OS it keeps returning even when “Cancel” button is chosen. It seems impossible to get out of a vicious loop of repeated authentications unless one knows a valid username and password.

There are ways out of this trap without pulling the plug. Try one of the following:

1. Press option-command-esc to bring up a task list. Choose “Finder” and click “Relaunch”
2. Open “Terminal” and get a process list by typing
`ps -aux`
Find the process that's causing the problem, usually the last “mount” process. Kill it.

Using CGI scripts to collect and publish class data

Some activities require that individual groups collect data. Then the data from all the groups are distributed to the whole class so all students can statistically analyse the data. For this purpose it is useful to have a web form so that each group can submit data. The file of all class data is then easily accessed by class members by a web browser and can be copied and pasted into an Excel worksheet.

For example, in the first session the class may go to the park and pitch baseballs to measure each individual's pitching speed. The following form allows everyone to submit their data, the distance and time for their pitch.

Send data form

Send your own pitching data

SFU email address for yourself: (each student should submit their own average.)
@sfu.ca

Your pitching distance = m,

Your average pitching time = s,

This form is produced by the following html code, vcollect.html:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>Send data form</title>
<meta content="N Alberding" name="author">
<meta content="students can send their data to me" name="description">
</head>
<body>
<h1>Send your own pitching data</h1>
<br>
<form action="http://sc3980-inst.surrey.sfu.ca/cgi-bin/vcollect.cgi"
method="post"> <input value="yes" name="nonempty" type="hidden">
<p>SFU email address for yourself: (each student should submit their own average.)<br>
<input maxlength="30" size="10" name="from" type="text">@sfu.ca<!--<textarea
name="comments" rows="1" cols="30"></textarea>-->
</p>
Your pitching distance = <input maxlength="50" size="15" name="d" type="text">m, <BR>
Your average pitching time = <input maxlength="50" size="15" name="t" type="text">s, <br>
<p> <input value="Send Data" type="submit"></p>
</form>
<br>
<br>
<br>
</body>
</html>
```

This form submits data to the cgi program, vcollect.cgi, written in perl, which accepts the data, reformats and writes it to a file.

```
#!/usr/bin/perl
#vcollect.cgi
# collect values of v and their error from students.
require('forms-lib.pl');

#Readinput from the form
%input = &GetFormInput();
$d = $input{'d'} ;
$t = $input{'t'} ;
$sender = $input {'from'};

open(GFILE, ">/Library/WebServer/Documents/vfile.txt");

print GFILE "\n";
print GFILE "$sender \t $d \t $t";
close(GFILE);

# Print out a confirmation message.
print "Content-type: text/html\n\n";
print <<"EndOfMessage";
<HTML>
<HEAD><TITLE>Comments Sent</TITLE></HEAD>
<BODY>
<H1>Your data have been sent</H1>
EndOfMessage
```

This script needs a library of perl programs for processing form data called forms-lib.pl.

It appends each submission to a tab-delimited file called vfile.txt which is located in the same subdirectory as the html code for the form and can be read with a web browser.

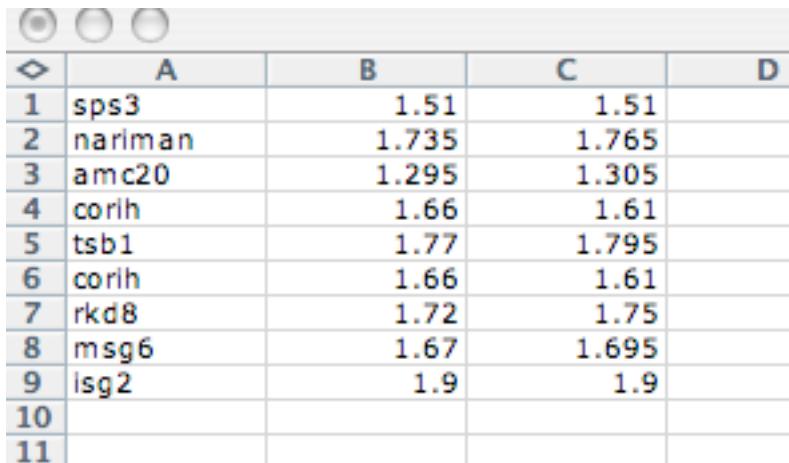
An example of the tab-delimited vfile.txt looks like this:

| | | |
|---------|-------|-------|
| sps3 | 1.51 | 1.51 |
| nariman | 1.735 | 1.765 |
| amc20 | 1.295 | 1.305 |
| corih | 1.66 | 1.61 |
| tsb1 | 1.77 | 1.795 |
| corih | 1.66 | 1.61 |
| rkd8 | 1.72 | 1.75 |
| msg6 | 1.67 | 1.695 |
| isg2 | 1.90 | 1.90 |

The instructor can edit this file using a text editor such as TextWrangler, vi or emacs.

Before class it should be cleared of data left over from a previous semester, and if there are several sections, it might be preferable for add a header lines labelling the section or make distinct files for each section by making custom html and cgi scripts for each section.

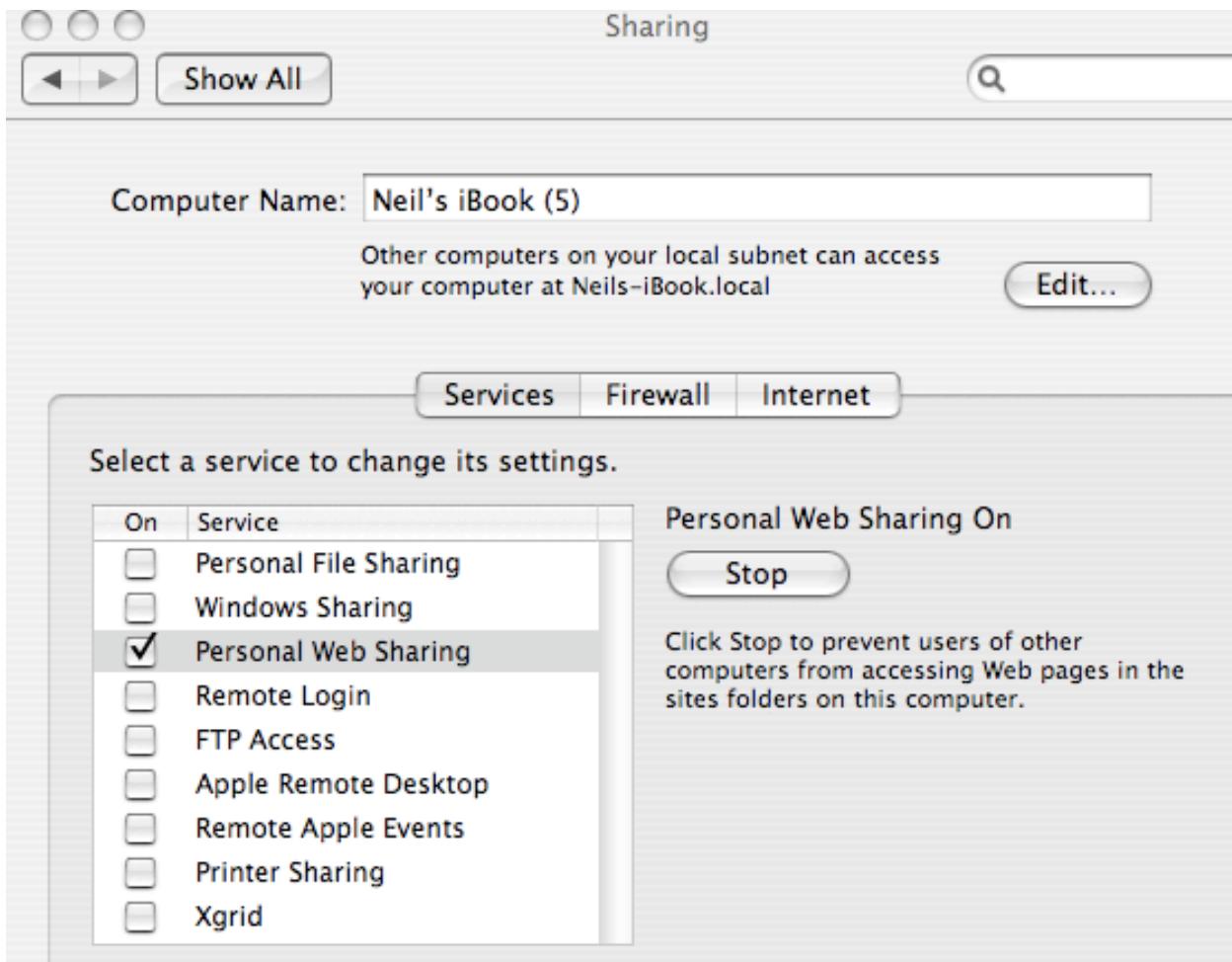
The data can be copied from a web browser and pasted into Excel for further analysis.



| | A | B | C | D |
|----|---------|-------|-------|---|
| 1 | sps3 | 1.51 | 1.51 | |
| 2 | nariman | 1.735 | 1.765 | |
| 3 | amc20 | 1.295 | 1.305 | |
| 4 | corlh | 1.66 | 1.61 | |
| 5 | tsb1 | 1.77 | 1.795 | |
| 6 | corlh | 1.66 | 1.61 | |
| 7 | rkd8 | 1.72 | 1.75 | |
| 8 | msg6 | 1.67 | 1.695 | |
| 9 | lsg2 | 1.9 | 1.9 | |
| 10 | | | | |
| 11 | | | | |

A convenient way to set up this system is to use the Apache webserver on the instructor's Mac in the Studio.

1. Put the vcollect.html file in /Library/WebServer/Documents
2. Put vcollect.cgi in /Library/WebServer/CGI-Executables. (Note that the subdirectory "CGI-Executables" is accessed through the url as "cgi-bin", as in "<http://sc3980-inst.surrey.sfu.ca/cgi-bin/vcollect.cgi>".)
3. Make vcollect.cgi executable by the terminal command: chmod 755 vcollect.cgi
4. create the data file in /Library/WebServer/Documents:
 - a. cd /Library/WebServer/Documents
 - b. cat - > vfile.txt (makes an empty file)
 - c. chmod 666 vfile.txt (necessary to allow the cgi to write to it.)
5. Turn on the webserver by opening System Preferences..Sharing panel and clicking on the box next to Personal Web Sharing. (Make sure that the computer is not set to go to sleep by the Energy Saver panel so that the students can access the data after hours.)



The instructor's computer uses a DHCP-assigned ip number, but the host name stays the same: sc3980-inst.surrey.sfu.ca.

When properly set up as described the data submission form can be accessed through a link to the following address:

<http://sc3980-inst.surrey.sfu.ca/vcollect.html>

and the data file can be accessed at

<http://sc3980-inst.surrey.sfu.ca/vfile.txt>

The same data collection system is used a number of times in the course:

| html form | cgi file | data file | comments |
|------------------|-----------------|------------|---|
| sdcollect.html | sdcollect.cgi | sdfile.txt | standard deviations |
| dropcollect.html | dropcollect.cgi | tfile.txt | bowling ball data |
| hcollect.html | vcollect.cgi | vfile.dat | (indoor height measurement alternative to the baseball pitch) |

The main index page accessed through <http://sc3980-inst.surrey.sfu.ca> summarizes the data collection pages and data files.

Studio Physics

Data Collection Pages

1. [vcollect](#) Unit 1, Activity 1-1: Baseball pitching speed
2. [tcollect](#) Unit 1, Activity 1-5: Bowling ball timing
3. [dropcollect](#) Unit 2, Activity 2-4 : 2-m ball drop timing
4. [sdcollect](#) Unit 2, Activity 2-5: Standard Deviations

Data Files

1. [vfile.txt](#) Unit 1,Activity 1-1: Baseball pitching speed
2. [file.txt](#) Unit 1, Activity 1-5: Bowling ball timing
3. [dropfile.txt](#) Unit 2, Activity 2-4: 2-m ball drop timing
4. [sdfile.txt](#) Unit 2, Activity 2-5: Standard deviations

Appendix: forms-lib.pl

```
#!/usr/local/bin/perl
# Forms-lib.pl
# Processes form input.
# Reads the form contents into $input, decodes it,
# and returns it as an associative array.
sub GetFormInput {
local(%input);
    $input = &ReadInput();
    %input = &ParseInput($input) if $input;
    return(%input);
}
# Reads in the form contents and returns it as a string.
sub ReadInput {
    local($method, $input, $length);
    $method = $ENV{'REQUEST_METHOD'};
# 1. If the method is GET, read the searchpart of the URL from
#     the QUERY_STRING CGI environment variable.
    if ($method eq "GET") {
        $input = $ENV{'QUERY_STRING'};
    }
# 2. If the method is POST, read CONTENT_LENGTH characters
#     from standard input.
    elsif ($method eq 'POST') {
        $length = $ENV{'CONTENT_LENGTH'};
        read(STDIN, $input, $length);
    }
# 3. Return the string.
    return($input);
}
# Takes a URL-encoded string and returns an associative array.
sub ParseInput {
    local($input) = @_;
    local(@pairs);
# 1. Split into key=value pairs.
    @pairs = split('&', $input);
    foreach $pair (@pairs) {
# 2. Convert all plus signs to spaces.
    $pair =~ s/+/ /g;
```

```

# 3. Split into a key and a value.
    ($key, $value) = split('=', $pair, 2);
# 4. Convert hex numbers to alphanumeric.
    $key =~ s/%(..)/pack("c",hex($1))/ge;
    $value =~ s/%(..)/pack("c",hex($1))/ge;
# 5. To handle multiple values, separate by newlines.
    $input{$key} .= "\n" if defined($input{$key});
# 6. Associate keys and values.
    $input{$key} .= $value;
}
# 7. Return the associative array
return(%input);
}
# Displays the form from a template.
sub DisplayForm {
    local(%input) = @_;
    local($doc);
    $doc = &FillTemplateFile("../templates/commentsform.html", %input);
    print &FillTemplateString($doc, %ENV);
}
1; # Always return true at end of a library.

```

Unit 1: Introduction and Computing

Session 1: Data Collection and Spreadsheet Use

First day:

Hand out Activity Guides: Usually the activity guides are pre-paid with the course fees. There is usually some attrition during the first days of class so it is advisable to hand out only the Activity Guides for the first two or three units and the Philosophy and Principles handout on the first day of classes.

Sample Handout Schedule (Dates refer to Fall 2007):

First day: (Jan 7) Philosophy and Principles, Units 1, 2, 3 Activity Guides, Homework for Units 1 through 7

End of Second week (Unit 3 session 2, Jan 18)) Units 4 through 7 Activity Guides, 3-ring binder and tabs

Middle of 5th week (Unit 7, Session 3, Feb 13) Units 8 through 14 Activity Guides, Homework for Units 8 through 14, 3-ring binder and tabs.

Activity Guides should be stapled, by unit, and three-ring punched.

Equipment needed for each group:

Tote basket with stuff for baseball throwing measurements

1. 3 baseballs
2. stopwatch
3. 30-m tape measure

In addition there needs to be a number line on the floor marking every metre for the total distance of 12 m. The number line usually remains from the previous semester, but can be reconstructed by the students if necessary.

We start using Excel today. The students enter their data in an online form (vcollect.html) which collects all the class data in a file (vfile.txt) that everyone can access.

Session 2: Measuring and Graphing Horizontal Motion



Equipment needed:

one bowling ball

one stop watch for every member of the class (less two, the bowler and the catcher)

The Force Concept Inventory test is usually given during this session. The exam booklets are reused from year to year but new bubble sheets need to be on hand.

Unit 2: Measurement and Uncertainty

Session 1: Direct and Indirect Measurements

Equipment needed:

Activity 2-1:

A piece of paper or cardboard for each group to measure. One can use a piece of A4 paper cut in quarters. Rough cuts would be good for this exercise.

Activity 2-4:

A ball

A stop watch

A 2-metre stick (the 2-metre sticks are actually 2.4 m long pieces of wood moulding with marks on them every 10 cm.)

Session 2: Random and Systematic Variation

This unit

