

**Extensions and Enhancements to the iLab Heat Transfer Project Site**

by

David P. Saylor

Submitted to the Department of Electrical Engineering and Computer Science  
in Partial Fulfillment of the Requirements for the Degrees of  
Bachelor of Science in Computer Science and Engineering  
and Master of Engineering in Electrical Engineering and Computer Science  
at the Massachusetts Institute of Technology

September 9, 2005

Copyright 1998 David P. Saylor. All rights reserved.

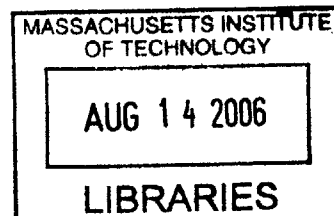
The author hereby grants to M.I.T. permission to reproduce and  
distribute publicly paper and electronic copies of this thesis  
and to grant others the right to do so.

Author \_\_\_\_\_  
Department of Electrical Engineering and Computer Science  
September 9, 2005

Certified by \_\_\_\_\_  
Clark K. Colton  
Thesis Supervisor

Accepted by \_\_\_\_\_  
Arthur C. Smith  
Chairman, Department Committee on Graduate Theses

**BARKER**



Extensions and Enhancements to the iLab Heat Transfer Project Site  
by  
David P. Saylor

Submitted to the  
Department of Electrical Engineering and Computer Science

September 9, 2005

In Partial Fulfillment of the Requirements for the Degree of  
Bachelor of Science in Computer Science and Engineering  
and Master of Engineering in Electrical Engineering and Computer Science

## **ABSTRACT**

The iLab Heat Transfer Project website started four years ago to enable web access to experiments related to movement of heat through transport processes. This thesis details improvements made to the site which extend and enhance the site prior to the project. Software improvements include giving teaching assistants the ability to add their entire class as users simultaneously and creating a method by which feedback data is stored as a full questionnaire instead of database entries. Hardware improvements include the addition of a webcam that streams video and audio of the experiment in real time and the integration of two new thermodynamic experiments complete with remote access. The final improvement is the administrator manual, which is intended to ease the burden on new staff members by bridging their knowledge with that of previous years.

Thesis Supervisor: Clark K. Colton  
Title: Professor of Chemical Engineering

# Acknowledgements

I would most like to thank my parents, David and Doris Saylor and my family for always being there for me and encouraging me to try to be the best I could be. Without them I would never have made it to MIT in the first place, much less be completing an advanced degree.

I would also like to thank my advisor, Professor Clark K. Colton for the numerous late nights spent drafting and revising the project files and all his support in the final month of my research assistantship.

Also, I thank the former research assistant Sid Sen for his knowledgeable advice and availability even after graduation, the class teaching assistant Anna Pisanía for her help in setting up the experiments, and to all the other people who have worked on this project in the past and helped it become a reality.

# Table of Contents

<b>List of Figures .....</b>	<b>5</b>
<b>1.0 Introduction .....</b>	<b>6</b>
1.1 Background .....	6
1.2 Problem Statement .....	6
1.3 Thesis Outline .....	7
<b>2.0 Overview .....</b>	<b>7</b>
2.1 Site .....	8
2.2 Software .....	9
2.3 Hardware .....	11
<b>3.0 Software Improvements.....</b>	<b>14</b>
3.1 Tool for Adding Student Accounts .....	14
3.2 Questionnaire Logging.....	16
<b>4.0 Hardware Improvements.....</b>	<b>18</b>
4.1 Addition of Webcam with Audio .....	18
4.2 Radiation Experiments .....	20
4.2.2 General Interface.....	23
4.2.3 HT14 Interface .....	23
4.2.4 HT16 Interface .....	27
4.2.5 Code .....	28
4.2.6 Web Access.....	30
<b>5.0 Administrator Manual .....</b>	<b>32</b>
<b>6.0 Impact and Future Work.....</b>	<b>32</b>
6.1 Impact.....	32
6.2 Future Work .....	33
<b>Appendix A – Administrator Manual.....</b>	<b>34</b>
<b>References .....</b>	<b>34</b>

# List of Figures

<b>Figure 1: The main interface of the iLab Heat Transfer Project .....</b>	<b>8</b>
<b>Figure 2: The Java chat interface .....</b>	<b>9</b>
<b>Figure 3: A simpleVI implemented in LabVIEW 7.0.....</b>	<b>10</b>
<b>Figure 4: The heat exchanger.....</b>	<b>11</b>
<b>Figure 5: The HT10XC’s front and back panels .....</b>	<b>13</b>
<b>Figure 6: The tool for adding groups of students to the site access.....</b>	<b>14</b>
<b>Figure 7: The code for parsing the “add users” input.....</b>	<b>16</b>
<b>Figure 8: Excerpt from the questionnaire .....</b>	<b>16</b>
<b>Figure 9: A sample questionnaire response.....</b>	<b>17</b>
<b>Figure 10: Code for <code>rblast_choice()</code> .....</b>	<b>18</b>
<b>Figure 11: Webcam Tracker Live, Main Window .....</b>	<b>19</b>
<b>Figure 12: Opening a network stream with VLC .....</b>	<b>19</b>
<b>Figure 13: Diagram of the HT14.....</b>	<b>21</b>
<b>Figure 14: Diagram of the HT16-REM .....</b>	<b>22</b>
<b>Figure 15: Instructions tab for both the HT14 and HT16 interfaces.....</b>	<b>23</b>
<b>Figure 16: The LabVIEW interface to the HT14 .....</b>	<b>24</b>
<b>Figure 17: The Data Table tab of the HT14 LabVIEW interface .....</b>	<b>25</b>
<b>Figure 18: Contents of a CSV file. ....</b>	<b>26</b>
<b>Figure 19: Administrator Settings tab of the HT14 interface.....</b>	<b>27</b>
<b>Figure 20: User Interface for the HT16.....</b>	<b>28</b>
<b>Figure 21: LabVIEW code for reading measurements .....</b>	<b>29</b>
<b>Figure 22: LabVIEW code for controlling equipment.....</b>	<b>31</b>
<b>Figure 23: Web interface for the HT14 .....</b>	<b>31</b>

# 1.0 Introduction

## 1.1 Background

The iLab Initiative, part of the iCampus Project was initiated to find an effective way to share experiments over the Internet to enhance the educational opportunities available to students. The ideal driving this goal is that students learn best through experimenting with actual equipment. For students, this would mean having access to large, expensive, or even dangerous experiments their college would be unable to provide otherwise.

Along with other web-based labs such as the Microelectronics Weblab and the Polymer Recrystallization Weblab, the Heat Transfer Project site endeavors to share experiments with other institutions, both national and international. The equipment has already been used by students at the University of Texas and the University of Cambridge in England. Additionally, there are plans to broadcast the site to students at the Obafemi Awolo University in Ile-Ife, Nigeria as well as universities in Brazil and Argentina.

The year-long project summarized in this thesis was aimed at improving the iLab Heat Transfer Project by both enhancing existing elements and extending the site with additional elements.

## 1.2 Problem Statement

The scope of this project was to solve several shorter-term goals which would improve the Heat Transfer Project site.

On the software side, there was no efficient way to add a whole class of users to the site's access at one time. If the teaching assistants could add all of their students simultaneously, it would save a considerable amount of explanation of signup procedures and allow the students to begin their experiments immediately.

An additional problem was with the feedback given by the students after their experiments were completed. They would answer an online questionnaire, but their answers were stored in a database, where it became extremely difficult to extract one particular set of answers. This problem became very evident when potential cooperators asked to see completed questionnaires from the students. A logging system needed to be created.

On the hardware side, there were two extensions to the site which had been discussed extensively. The first was the integration of a webcam and streaming server which would allow the students to see and hear the real-time response of

the experiments as they performed them. The second was to add two additional experiments to the offerings of the

Finally, the most significant problem left over from previous years was the repeated loss of knowledge of how to run the complex site and equipment as the staff of the project changed from year to year. The entire system would have to be relearned nearly from scratch by the incoming members. There was a desperate need for a way unifying the knowledge and skills from previous years with the present, and so an administrator manual was created this year.

## 1.3 Thesis Outline

First there will be a brief overview of the iLab Heat Transfer Project site as it stood at the beginning of this project, including the site, the backend software, and the hardware components to give a picture of the complexity of the system.

Next there will be discussion of the two software improvement tools made over the course of year. One which enables adding an entire class worth of students to the system in one action, and another which stores a copy of the questionnaires submitted by students to preserve the feedback which is essential to the future of the site.

Following the software improvements will be the hardware extensions. A webcam was added to give students a real-time link to the experiment they are performing, to make it seem more real. Also two additional experiments were added to the site, with interfaces created in LabVIEW and fully accessible by remote computer.

Last is the discussion of the administrator manual, which is intended to be useful to anyone who should happen to be on the Heat Transfer Project team, regardless of computer skill. The document is made to be complete and extensible as the system changes.

As a conclusion, the successful extensions and enhancements to the site will again be emphasized and a few possible future projects will be presented.

## 2.0 Overview

The iLab Heat Transfer Project site is intended to provide web access to physical experiments related to thermal processes. This section contains a brief description of the setup of the iLab Heat Transfer Project site and the supporting software and hardware.

## 2.1 Site

 iLab Heat Transfer Project

---

**Home**  
Friday, September 09, 2005 3:15:13 AM

**Home** **Welcome, this is the iLab Heat Transfer Project website.**

---

**Access**

- Login
- Register
- Class Teams**
- Schedule

---

**Website**

- About
- Team
- Publications
- Archive

---

**Experiments**

- Equipment
- Theory
- Instructions
- Perform Experiment**
- Questionnaire



To learn about the web site, click on [About](#).

To access all features of this web page, please [log in](#) or [register](#).

Web lab exercises currently taking place:

[10.302 - Flat Plate Heat Exchanger](#)

© 2003 MIT Department of Chemical Engineering, all Rights reserved. Last Modified November 2004.  
Contact the administrator at [dsaylor@mit.edu](mailto:dsaylor@mit.edu) for questions or comments about this website.

Figure 1: The main interface of the iLab Heat Transfer Project. All of the grey links are unavailable to users who have not registered and logged in. The site is available at <http://heatex.mit.edu/HeatexWeb/>. The grey links are not available until the user has logged in.

The main page for the iLab Heat Transfer Project is located at <http://heatex.mit.edu/HeatexWeb> and will look like the screen capture in Figure 1. When users first access the site, having not yet logged in, some links are not available to them and appear in grey. After they register and log in, these links will become available.

The pages on the site are grouped into three major categories. The Access links let users log in and out and see what other users are registered on the site and what times they have reserved to use the equipment. The Website links contain



information about the site, including a general description of the site and project, the current development team, the publications written about the site, and the archive of past assignments. The Experiments links relate to conducting the experiment. The equipment, theory, and instructions links should be read by users to familiarize themselves with the procedures before they conduct the experiments. When they are ready, the perform experiment link lets them access all of the interfaces currently available. After they have finished the experiment, the users submit the questionnaire to help with future development of the site.

## 2.2 Software

The site has several supporting software components. The server machine uses Windows 2000 Server Advanced's default IIS for serving the majority of the files on the site. Microsoft Visual Studio .NET is used to develop all of the web pages and run server-side code. Microsoft SQL Server is the database backend for storing users, passwords, courses, teams, and the responses to the feedback questionnaire.

The site also runs a simple instant messaging service implemented in Java Forte. This chat facility allows students in the class to collaborate from remote locations. A chat interface window is present at the top of every experiment page. Figure 2 shows the chat interface, with a successful login from a remote computer. The chat server is integrated directly with the database, so no additional login information is required to use it.

# Heat Exchanger Experiment

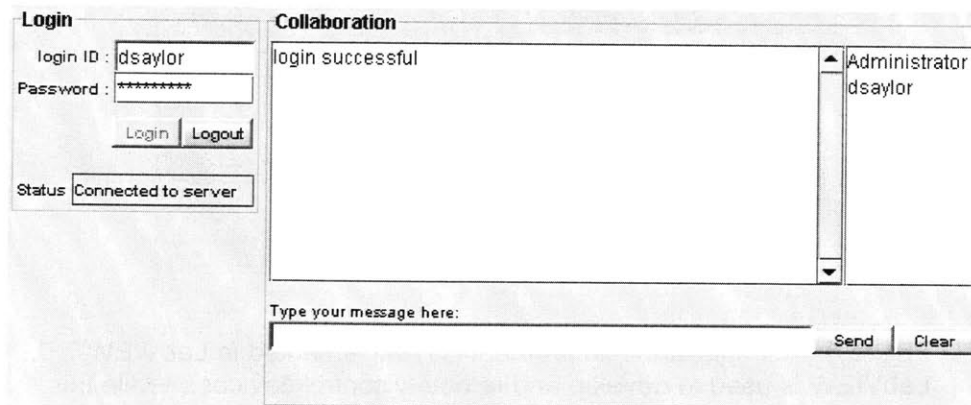


Figure 2: The Java chat interface. This applet window appears at the top of the page for every experiment which runs on the iLab Heat Transfer Project site. Using this collaboration facility, students can perform the experiment together without being in the room together. The logins and

passwords carry over from their account used to access the experiment interface, so no additional login information is required.

Finally, the principle application which enables experiments to be run remotely is LabVIEW, by National Instruments (<http://www.ni.com>). Because newer versions of LabVIEW are not fully backwards-compatible, the iLab Heat Transfer Project site uses version 7.0, even though the latest version is 7.1.1. LabVIEW allows users to create an instrument panel, called a Virtual Instrument (VI), graphically in the same way one would create it in Adobe Photoshop. Once all of the components are in place, the user switches to the block diagram view of the VI where the inputs and the outputs of the components are connected to form the functionality desired. This is the unique way in which LabVIEW panels are programmed, like wiring programming modules the same way one would chips on a bread board. Wires are colored according to data type: blue is for integers, orange is for floating points, green is for Booleans, and pink is for strings. In Figure 3, a simple VI has been created where a switch will toggle and LED on and off every time it is flipped. The block diagram is shown on the left and the panel is on the right. The devices used in the iLab Heat Exchanger Project have more complicated interfaces, owing to the interaction with mechanical devices instead of simulated ones, but the code is similar.

LabVIEW has its own internal web server which is capable of publishing VIs across the Internet. Any VI which is open in LabVIEW can be viewed and accessed remotely if the LabVIEW runtime environment plug-in is installed in the user's web browser. With the plug-in and the LabVIEW web server, users can interact with the VIs in real time.

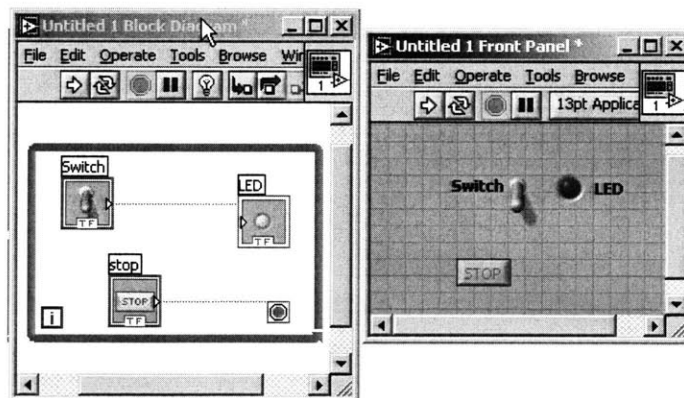


Figure 3: A simple virtual instrument (VI) implemented in LabVIEW 7.0. LabVIEW is used to develop and remotely control devices. While this interface is running, this switch will toggle the green LED on and off as the switch is flipped. The stop button will stop the interface from running.

## 2.3 Hardware

Prior to the addition of the two radiation experiments, there were four devices on which students could run experiments remotely. All of these are devices manufactured by Armfield, Ltd. (<http://armfield.co.uk>).

The most notable experiment, for which the site's domain is named, is the heat exchanger. A picture of the heat exchanger apparatus is given in Figure 4. The device was custom modified by Armfield for the MIT iLab Heat Transfer Project to include support for remote controlling of the experiment. The base of the apparatus is a service unit called the HT30XC which reads in data from the thermocouples and other sensors and transmits it to the computer via USB. The LabVIEW interface passes values across the USB to the service unit, which in turn controls the device directly.

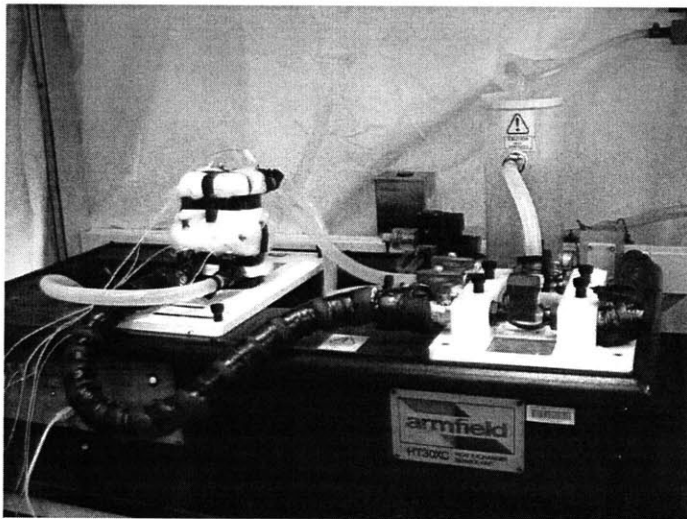


Figure 4: The heat exchanger. The base is the HT30XC service unit which communicates data from the various probes back to the LabVIEW interface running on the computer. The interface is also able to send commands to the service unit which then controls the equipment directly.

The other three experiments connect to a different service unit, the HT10XC. These experiments are heat conduction experiments, the HT11 Linear Conduction Experiment, the HT12 the Radial Conduction Experiment, and the HT15 Extended Surface Conduction Experiment. This service unit also connects to the computer via USB. The HT10XC service unit will be discussed in greater detail because the two experiments added to the iLab Heat Transfer Project for this thesis also use it as a service unit.

Figure 5 shows a diagram of the HT10XC's front and back panels. The back panel provides power to the heater in an accessory device. Devices with special controls such as fans attach to the auxiliary output jack in the center. The front panel has controls for manual operation of the accessory device. On the top left of the panel are the switches which control the power to the device and whether the device should accept input from a remote computer. The middle left is where several specialized probes can be connected to the unit. These include probes for thermal radiation (R), brightness (L), and air velocity (Ua). The dial in the upper right can select one of these probes to have its measurements displayed on the LCD next to it. The dial and LCD under them in the middle right display readings of the ten thermocouples at the bottom of the panel. The thermocouple connectors take K-type thermocouples. T10 is the only thermocouple capable of measuring up to 500°C, the rest can only measure up to 200°C. Devices which do not use T10 use the thermocouples starting at T1 and increasing. Devices which do use T10 use thermocouples starting at T10 and decreasing.

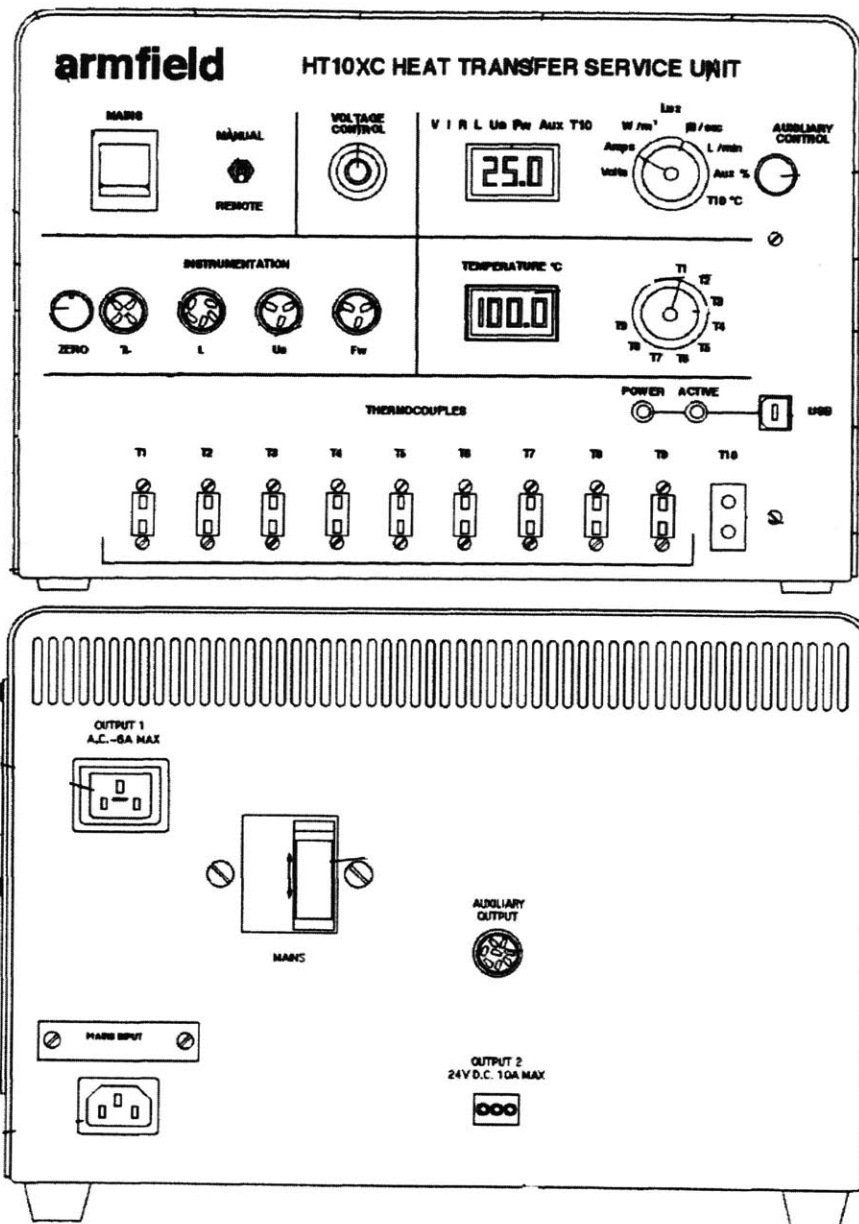


Figure 5: The HT10XC's front and back panels. This device is the service unit which the heat conduction and radiation experiments connect to. Sensors connect to the slots on the front panel and the device is powered from the back panel.

## 3.0 Software Improvements

### 3.1 Tool for Adding Student Accounts

In older versions of the site, the students were responsible for creating their own accounts to log in to perform the experiments. This setup resulted in students creating anonymous user accounts with invalid email addresses and other incorrect information. To solve this problem during the Fall 2004 offering of 10.302, the teaching assistant manually entered information for every student and created their accounts for them through the registration function of the web site. Although the problem of invalid information was fixed, this process took many hours of the teaching assistant's time.

The final solution is to have a tool to create many accounts at once. Figure 6 shows the interface to the tool. The first two text boxes have which course the set of students is being signed up under and a password which will be common to all of the accounts. The only things which need specified for each student are their name, email address, and team name, which is usually the day and time when they are performing the experiment, such as Thu12-2 for the group assigned to Thursday from 12:00 until 2:00.

The parser automatically generates a username, a globally unique identifier (GUID), and a timestamp for the account creation which are associated with their account.

The format for the large textbox is: <Full Name> <Email address> <Team>  
Spaces are permitted in the name of the team. Each entry must be on a separate line.  
Each entry will have the Course and Default Password specified.  
A maximum of 50 users can be entered simultaneously.

```
Course:10.302
Default Password:new10302user
David Saylor dsaylor@mit.edu Thu-3
Student #1 test1@mit.edu Fri-12
Cambridge Student #4 test4@cambridge.ed.uk Mon-9
```

Figure 6: The interface for the tool for adding large groups of students to the site access simultaneously. The accounts will be created all with the same course number and password.

The tool assumes that the first word which contains a @ character is the email address. Therefore it treats everything before the email address as the student's name and everything after it as the team they are assigned to. The same username of the email address will be used as their username for the site. The site is able to handle multiple accounts with the same username because it is the GUID which defines uniqueness in the table. The code used to parse each line of the text box, called "users" in the HTML code, is given in Figure 7.

```

private void Submit_Click(object sender, System.EventArgs e)
{
    string[] users = inputBox.Text.Split(new
    char[]{'\n'},50);
    foreach (string _line_ in users)
    {
        SqlConnection con;
        string sql;
        SqlCommand cmd;
        string[] tokens = _line_.Split(new char[] { ' '},
        10);

        string globalID = Guid.NewGuid().ToString();
        string firstname = tokens[0];
        int i = 1;
        string lastname = "";
        while(tokens[i].IndexOf('@') == -1)
        {
            lastname += tokens[i];
            i++;
        }
        string email = tokens[i];
        string team = "";
        while(i < tokens.GetLength(0))
        {
            team += tokens[i];
            i++;
        }
        string login =
        email.Substring(0,email.IndexOf('@'));
        string course = courseBox.Text;
        string passwd = passBox.Text;

        sql = "INSERT INTO [Students](UserID, Login,
        Password, FirstName, LastName, Email, Course, Team,
        Registered, IsAdministrator) VALUES (";
        sql += "\"" + globalID + "\", \"" + login + "\", \"" + passwd + "\",
        \"" + firstname + "\", \"" + lastname + "\", \"" + email + "\",
        \"" + course + "\", \"" + team + "\",
        \"" + DateTime.Now.ToString() + "\", \"" + 0 + "\"";
        con = new
        SqlConnection(ConfigurationSettings.AppSettings["He
        atex.ConnectionString"]);
        con.Open();

        try{
            cmd = new SqlCommand(sql, con);
            cmd.ExecuteNonQuery();
        }
        catch (Exception err) {
            error.Text = err.ToString(); }
        con.Close();}
    }
}

```

}  
 Figure 7: The code responsible for parsing the contents of the “users” text box. On every line, the first word containing an @ character is assumed to be the email address of the student. Everything before the email address is assumed to be the student’s first name and everything after to be their team assignment.

## 3.2 Questionnaire Logging

Feedback from students about the quality of education that they are receiving is an integral part of all courseware at MIT. The iLab Heat Transfer Project is no different. In prior years, an online questionnaire was developed to record students’ reactions to being subjected to the weblab. It is available online at <http://heatex.mit.edu/HeatexWeb/Questionnaire.aspx>. The questionnaire asks the students to respond with their level of agreement, ranking it 1 to 7, lowest to highest, to several statements about their experiences during the course of the experiment. Statements are grouped into subcategories of usability of the website, the remote panel interface, the Java chat feature, possible improvements for the site, accomplishing educational objectives, the overall experience, and how remote experiments compare to hands-on experiments. At the end of each section, there is a box for additional comments. An excerpt from the current questionnaire is shown in Figure 8.

7 = Strongly Agree, 1 = Strongly Disagree

**Usability when carrying out the experiment.**

1. The instructions were clear.  1  2  3  4  5  6  7

2. I had no problems operating the experiment.  1  2  3  4  5  6  7

3. I was able to control the electrical heater during the experiment.  1  2  3  4  5  6  7

4. I was able to make transient and steady state temperature measurements as required.  1  2  3  4  5  6  7

5. I was able to record and retrieve the data needed for the assignment.  1  2  3  4  5  6  7

6. It actually took me \_\_\_\_\_ minutes to complete the experiment.

*Comments on Usability:*

Figure 8: Excerpt from the questionnaire. This questionnaire asks students to indicate their level of agreement with several statements from 1 (lowest) to 7 (highest). There are also comments fields for every group of statements.



The approach used to store responses in previous versions of the site, however, was to simply enter all the responses into a database without any context as to the question the student was responding to. This situation was problematic when, after several revisions were made, it became necessary to examine feedback from previous years. Having only numbers, the data was meaningless.

To fix this problem, the simplest solution was to create a copy of the questionnaire each student submitted, as well as entering the information into the database. Although this means a possibly large number of files will be created, it also means the questionnaires will be immediately available when they are needed and any aggregate statistics that need calculated can be done through use of the database. A sample response is shown in Figure 9.

Submitted: Thursday, August 25, 2005 at 2:27:50  
 Course Name: 10.302  
 Team Name: Team 1  
 Your Name: David Saylor

Complete this questionnaire, and submit your results at the bottom of the page.

Note: You may only submit this questionnaire once.

7 = Strongly Agree, 1 = Strongly Disagree

**Usability when carrying out the experiment.**

- |  |  |
|--|--|
| 1. The instructions were clear.  | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input checked="" type="radio"/> 6 <input type="radio"/> 7 |
| 2. I had no problems operating the experiment.   | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 |
| 3. I was able to control the electrical heater during the experiment.                  | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input checked="" type="radio"/> 6 <input type="radio"/> 7 |
| 4. I was able to make transient and steady state temperature measurements as required. | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5 <input type="radio"/> 6 <input type="radio"/> 7 |
| 5. I was able to record and retrieve the data needed for the assignment.               | <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5 <input type="radio"/> 6 <input checked="" type="radio"/> 7 |
| 6. It actually took me 45 minutes to complete the experiment.                          |  |

*Comments on Usability:*

Interface was very easy to us due to nature of experiment. All that was need was to specify flow rates and one temperature, and the experiment would run itself.

Figure 9: A sample questionnaire response. An html file like this one is generated for every questionnaire which is submitted. These files, although numerous, provide immediate access to the student responses exactly as the student saw them.

The code used to generate these html files consists mainly of statements like this: `f.WriteLine("<TR><TD>1. The instructions were clear.</TD><TD><P>" + rblast_choice(Q1.SelectedItem.Value) + "</P></TD></TR>");`

Here, `Q1.SelectedItem.Value` represents the string value of the student's selection. The function `rblast_choice()` returns the HTML code for a list of radio buttons with one selected. The full implementation of this function is given in Figure 10.

```
private String rblast_choice(string selection) {
```

```

int s = Int32.Parse(selection);
string returnstring;
returnstring = "<table border=0 height=9 width=240><tr>";
for(int i=1; i<=7; i++)
    if (s==i)
        returnstring += "<td><input type=radio checked=true>"+
            i.ToString() + "</td>";
    else
        returnstring += "<td><input type=radio>" +
            i.ToString()+"</td>";
returnstring += "</tr></table>";
return returnstring;
}

```

Figure 10: Code for `rblast_choice()` which returns the HTML code for a list of seven radio buttons with one of them selected. This function is called for every question in the questionnaire with a 1-7 ranking choice.

## 4.0 Hardware Improvements

### 4.1 Addition of Webcam with Audio

In previous versions of the iLab Heat Transfer Project website, there had been a Logitech Quickcam Pro 4000 webcam set up which would broadcast a video stream of the experiment setup through a third party service called WebcamPlanet. This arrangement was unsatisfactory because there was no control over the broadcasting and because audio was not transmitted across the stream with the video.

After comparing several other services and server products, a licensed was purchased to use a program called Webcam Tracker Live produced by Sytaxis Software, <http://sytaxis.com/>. Figure 11 below shows the main interface window of Webcam Tracker Live, with the audio signal of a clap to show the audio capabilities of the program.

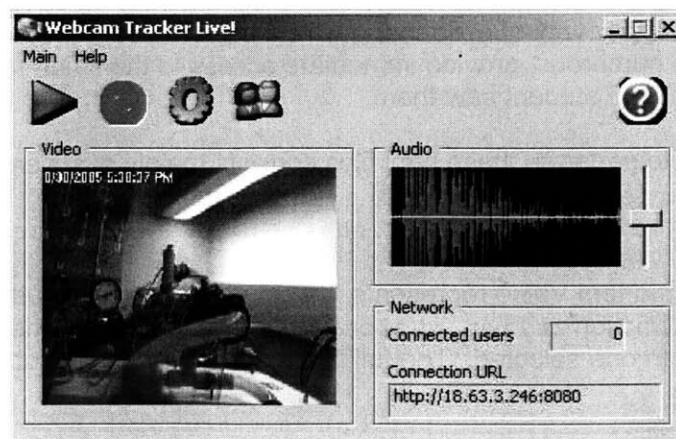


Figure 11: Webcam Tracker Live, Main Window. This is the main interface window to the software program we use to stream live video and audio of our experiment. The image shown is the shell and tube heat exchanger, and the sound of a clap is displayed in the audio panel.

In order to view the webcam feed, the most reliable client is the VideoLAN Client (VLC) media player, available at <http://www.videolan.org/vlc/>. Because the video and audio are both compressed using the Windows Media 9 Codec package, which is specific to the Windows operating system, it could be extremely complicated to watch the stream on Mac or Unix computers. VLC solves this problem. It does not require any external codecs to decode the stream. They are already included internally. Additionally, it is available for a wide range of operating systems, including Windows, Mac OSX, Red Hat Linux, and FreeBSD. By installing VLC, and entering the URL of the webcam stream into the Open Network function as shown in Figure 12, users on any of the supported operating systems will be able to watch and hear the experiment as it is conducted.

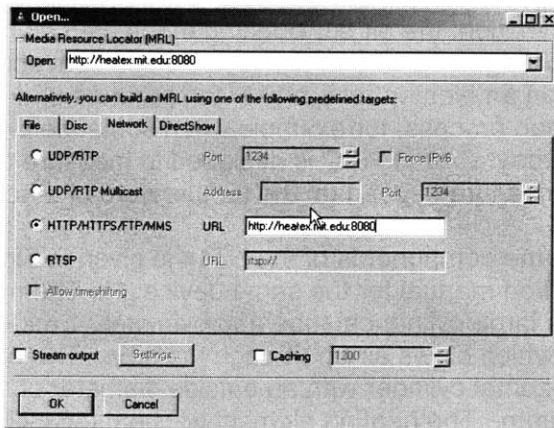


Figure 12: Opening a network stream with VLC. This media player is able to open video regardless of the codec used to compress it. Webcam Tracker Live uses a Windows proprietary codec, so having a media player which is able to decode it regardless of the operating system is a great advantage.

Webcam streams often take up a large amount of bandwidth, which can be a problem when trying to stream to places such as Obafemi Awolo University in Nigeria. They are one of several international contacts who have expressed interest in the iLab Heat Transfer Project and accessing its experiments for academic use. To accommodate users who have severe bandwidth limitations, we have configured the webcam stream to be extremely low quality. The video is set to a small resolution at very low quality, which still serves its purpose because the heat exchanger experiment has no moving parts. The audio is also set to a low quality, one-channel encoding. Although the presence of sound is extremely important to the interactive experience of using the heat exchanger, it

is only the sound of a motor and does not require a large amount of bandwidth to be heard.

## 4.2 Radiation Experiments

To expand the number of experiments offered on the iLab Heat Transfer Project site, two additional Armfield units were purchased, the HT14 Combined Radiation and Convection Accessory and the HT16 Errors in Radiation Measurement Accessory. These devices are meant to exhibit the effects of thermal radiation on temperature measurement. Although both devices can be operated manually without use of computer software, the additional software provides data tracking and remote operation which are not possible with the hardware alone.

### 4.2.1 Experiment Description

The HT14 device exhibits measures the properties of forced convection on a heated element. Without any outside influence, the process of free convection occurs. In free convection, the air surrounding the heated element gains heat through radiation, changes density, and rises. Forced convection is the process where air is blown on an element of a higher temperature and as a result cools it off. This is a common process, for example when someone blows on hot food or uses a fan on a hot day. The HT14C is designed to measure the dynamics of the effect of wind at a measured speed on the temperature of a heated object.

A diagram showing the components of the HT14 is given in Figure 13, taken from the Armfield instruction manual for the same device [1]. The main body of the device consists of a large cylindrical tube approximately 1 meter tall. At its base is a centrifugal fan which blows air up through the tube. The element at the top of the tube is a horizontal cylinder with an outside diameter of 10mm and a heated length of 70mm. The heating element will heat the cylinder to a maximum temperature of 500°C, for safety reasons. The temperature of the cylinder and the upward-flowing air are measured with K-type thermocouples connected to the HT10XC service unit, which was discussed in Section 2.3, as T10 and T9, respectively. The air velocity is measured with an anemometer, which also connects to the HT10XC at Ua.

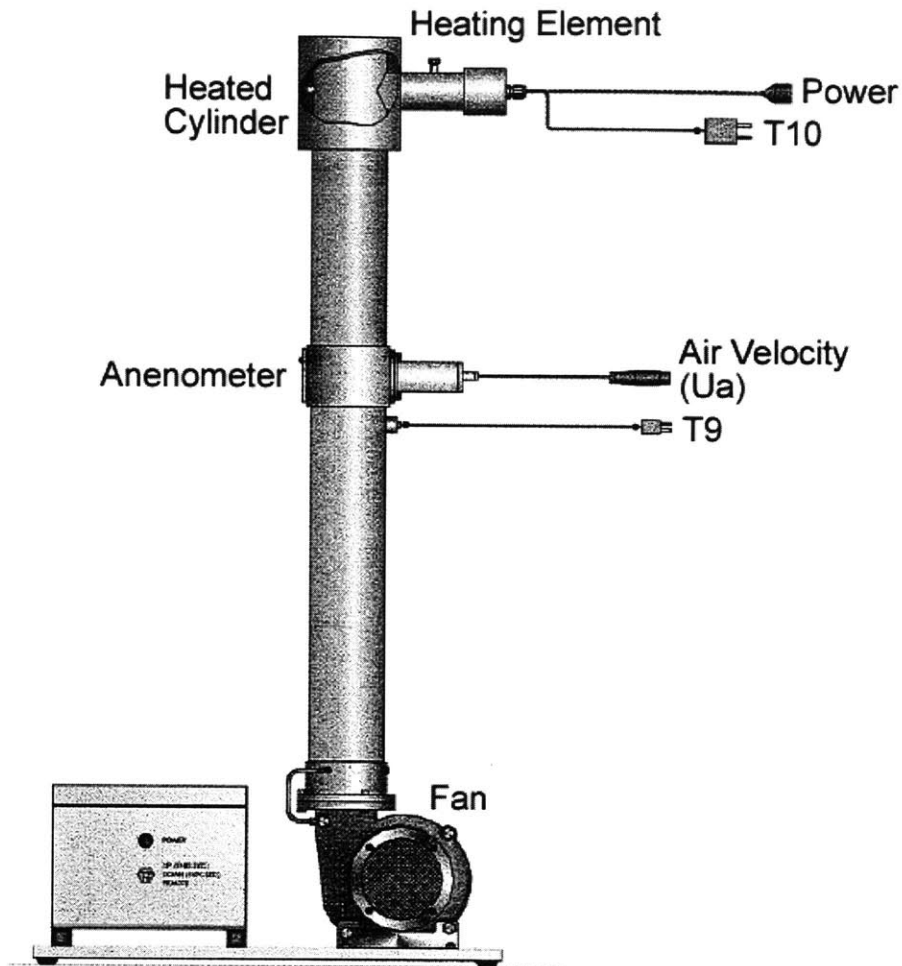


Figure 13: Diagram of the HT14. The fan at the base blows air up through the tube across the cylinder heated by the heating element. The anemometer measures the velocity of the wind. The temperatures of the upward air and the cylinder are measured by T9 and T10, respectively.

The HT16 exhibits the errors in temperature measurement which can be introduced by a heat source near the measuring instrument. The outside heat source affects the temperature which the thermocouple sees, even though it is a localized temperature difference. This is especially true when the temperature of a gas is being measured. The HT16 measures the effect of radiation on the temperature readings of three different kinds of thermocouples.

A diagram showing the different parts of HT16 is given in Figure 14, taken from the Armfield instruction manual of the same device [2]. Like the HT14, the main body of the device is a long cylindrical tube with a centrifugal fan at the base. Also like the HT14, the velocity of the upward air velocity is measured using an

anemometer which connects to Ua on the HT10XC. At the top of the tube are three thermocouples, each with a different coatings on the beads used for temperature measurement. Each of these thermocouples connects to the HT10XC service unit. The T7 thermocouple has a normal-sized thermocouple bead of 0.5mm diameter. The bead and leads are polished, which means they are completely exposed. The T8 thermocouple has the same size bead, but its bead and leads are coated with a black, heat-resistant paint which raises their emissivity. The T9 thermocouple has a large, 3mm bead which is coated with the same paint as the T8 thermocouple. The leads are insulated. The temperature of the wall is measured with the T10 thermocouple. The unique feature of the HT16 is the remotely-controllable radiation shield. When the shield is raised, it stands between the thermocouples and the heated wall.

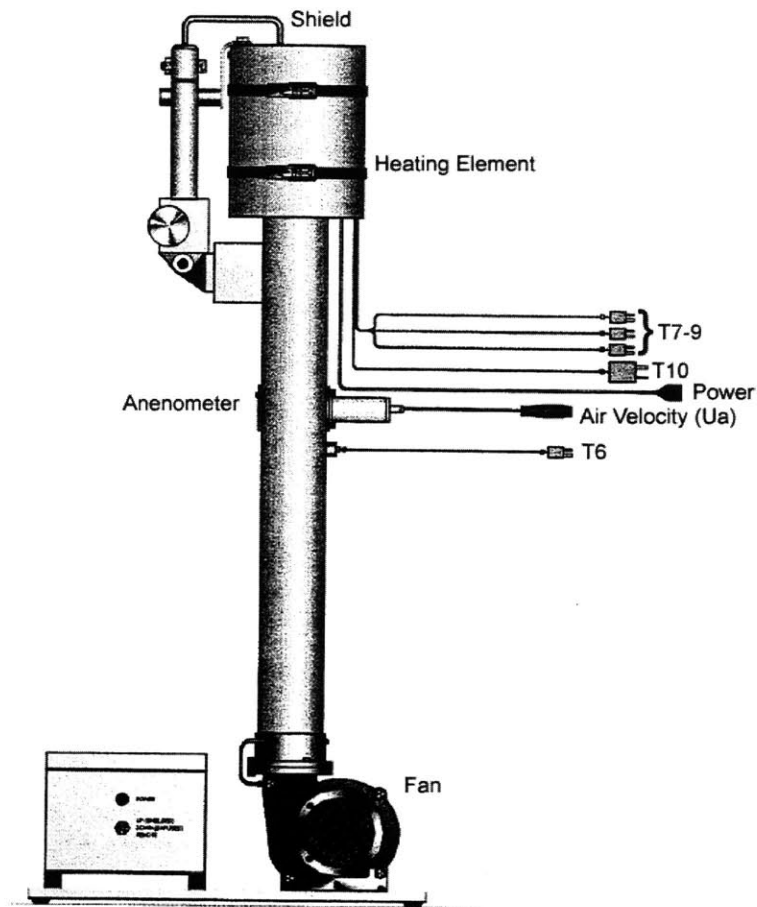


Figure 14: Diagram of the HT16-REM. Like the HT14, the fan at the base blows air up the tube, the velocity of which is measured by an anemometer. At the top are three thermocouples with three different types of heat shielding on them which also measure the temperature of

the air, but are subjected to the presence of an external heating element on the wall directly adjacent to them. T10 measures the wall temperature, T7-9 measure the three thermocouples temperatures, and T6 measures the true temperature of the upward air.

## 4.2.2 General Interface

To be able to run these experiments remotely, there are two LabVIEW interfaces with all the controls necessary to run the controllable aspects of the equipment. The interfaces for both the HT14 and HT16 include an Instructions tab which describes the general procedure for gaining control of the remote panel before running the experiment. This tab is the first one seen by a student logging in and is shown in Figure 15.

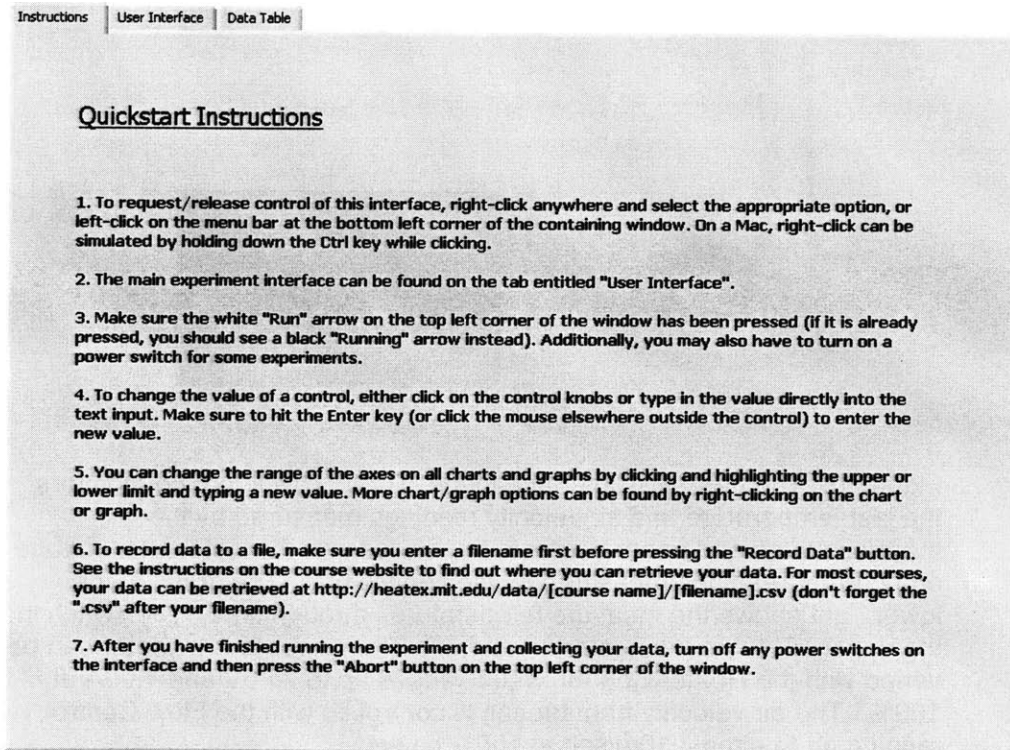


Figure 15: Instructions tab for both the HT14 and HT16 interfaces. These instructions describe how to gain control of the interface once the VI has been loaded.

There are three other tabs, two visible and one hidden, in each interface which are unique to the specific experiment.

## 4.2.3 HT14 Interface

The LabVIEW interface for the new HT14 is shown in Figure 16. This is the interface which students will see when they log into the experiment remotely. The interface reports the last temperature measurements made by the thermocouples, the last air velocity measurement made by the anemometer, and the last voltage and current measurements made by the service unit. The graph shows in the lower right tracks the temperature profile through time.

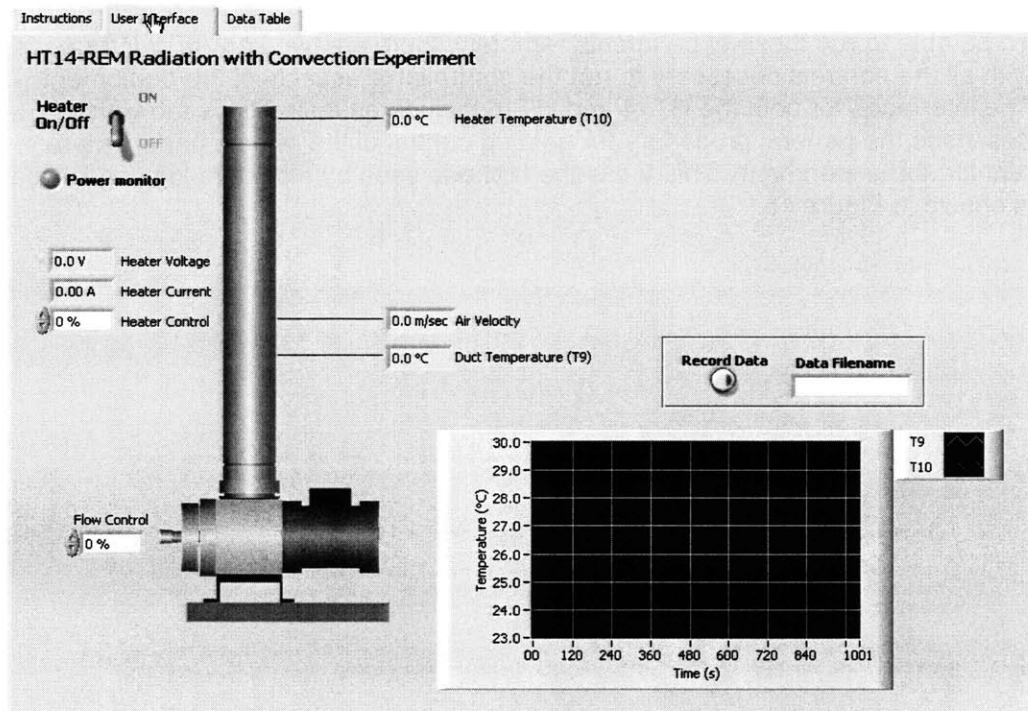


Figure 16: The LabVIEW interface to the HT14. This interface monitors the last temperature and air velocity readings measured by the thermocouples and anemometer in the device and the last heater voltage and current readings measured by the service unit. The graph in the lower right shows the measure temperatures through time. The switch in the upper left turns the heater on and off. The power to the heater can be varied with the Heater Control, which ranges up to 24.0V and 4.00A at 100%. The air velocity from the fan is controlled with the Flow Control, ranging up to almost 10m/sec at 100% power.

The controls on the interface control the actual device remotely. The switch in the upper left turns the power to the heater on and off. The amount of power the heater draws can be controlled with the Heater Control, which ranges from 0-100%. At maximum power, the heater draws around 24V and 4.00A. The Flow Control controls the power to the fan and therefore the speed of the upward-flowing wind. At maximum power, the wind from the fan will reach slightly less than 10m/sec.



The box above the temperature graph is how the students record data. While the Record Data button is not pressed, all the incoming data is graphed and then discarded. When there is a valid Windows filename present in the Data Filename field, and while the Record Data button is depressed, the data will be written out to that filename with a .csv extension in addition to being graphed. CSV stands for “comma separated values”, which is like a text version of a spreadsheet, with rows of numbers and commas separating the column values. An example of the data found in CSV files generated from this interface is given in Figure 18. CSV files can be read by Microsoft Excel and OpenOffice which can be used for data analysis.

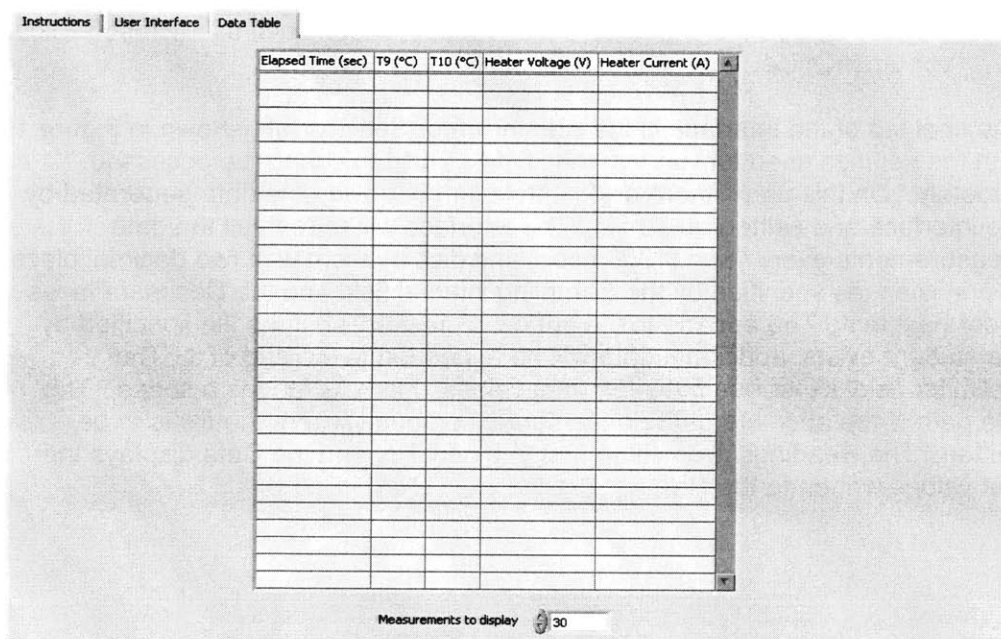


Figure 17: The Data Table tab of the HT14 LabVIEW interface. This table holds the last 30 measurements reported by each of the instruments along with the time at which it was reported. Any previous measurements can be viewed by scrolling up through the table. If the Record Data button on the User Interface tab is pressed, the data will be written out to a file in addition to being displayed in this table.

```
Elapsed Time (s) , T1 (°C) , T2 (°C) , T3 (°C) , T4
(°C) , T5 (°C) , T6 (°C) , Heater Voltage (V) , Heater
Current (A) , Qc (L/min)
121.81 , 23.55 , 69.86 , 7.19 , 1.18
122.81 , 23.45 , 70.10 , 7.18 , 1.18
123.81 , 23.45 , 70.35 , 7.21 , 1.17
124.81 , 23.64 , 70.59 , 7.18 , 1.18
125.81 , 23.55 , 70.59 , 7.19 , 1.18
126.81 , 23.84 , 70.35 , 7.18 , 1.17
```

```
127.81 , 23.84 , 70.35 , 7.19 , 1.17
128.81 , 23.74 , 70.35 , 7.19 , 1.18
129.81 , 23.94 , 70.10 , 7.21 , 1.17
130.81 , 23.94 , 69.86 , 7.19 , 1.18
131.81 , 23.84 , 70.10 , 7.19 , 1.17
132.81 , 23.84 , 69.61 , 7.18 , 1.17
133.81 , 23.84 , 69.61 , 7.18 , 1.17
134.81 , 23.74 , 69.61 , 7.18 , 1.17
135.81 , 23.74 , 69.61 , 7.18 , 1.17
136.81 , 23.84 , 69.37 , 7.19 , 1.18
```

Figure 18: An example of the contents of a comma separated values (CSV) file. This is the format which the LabVIEW interface records data in. It is a text version of a spreadsheet. The rows are separated by newline characters and the columns are separated by columns. This data can be readily be opened with a spreadsheet application such as Microsoft Excel or OpenOffice.

The final tab of the interface is the Administrator Settings tab, shown in Figure 19 with the settings used for running the HT14. This tab cannot be accessed remotely. On this panel there are controls for controlling the data generated by the interface and written out to file. The interface will refresh all the data measurements every 1sec (1000msec) and display them with two decimal places of precision, as specified by the Sampling interval field and the Decimal Places slider next to it. The append to file button is pressed, so if the file specified by the student exists, additional data will be appended to the end of it. The Delimiter field is inserted between data values. Here, \s means a space. The File path is the absolute path on the server computer where the file is to be written. The Headings are written first to a new file, and the Data displays the last values written to the file.

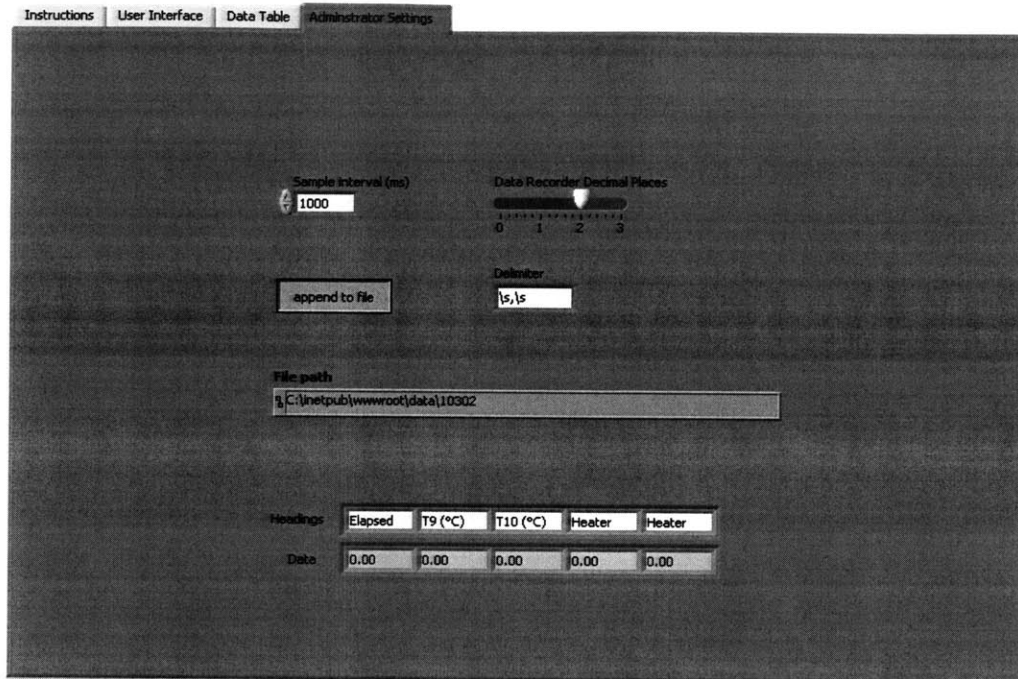


Figure 19: Administrator Settings tab of the HT14 interface. This tab is hidden and cannot be accessed remotely. The Sample interval controls how many milliseconds the interface waits before resampling the instruments. The sliding bar to the right controls how many decimal places of precision are given to measurements. The Append to file button determines if an existing file will be overwritten or appended with new data. The Delimiter is written between measured values. The Headings are inserted at the beginning of a new file and the Data is the last set of measured values.

#### 4.2.4 HT16 Interface

The interface for the HT16 experiment is shown in Figure 20. The HT16 is fundamentally similar to the HT14, and there are obvious similarities between their two interfaces. The HT14 interface had controls for the heater and fan power, a thermocouple for the upstream air temperature, a temperature graph, and a box for specifying a filename, the same as on the HT16 interface. The HT16, however, has additional readouts for measuring the three thermocouples affected by the wall temperature, as well as a readout for the temperature of the wall itself. There is also a switch to control the vertical position of the radiation shield.

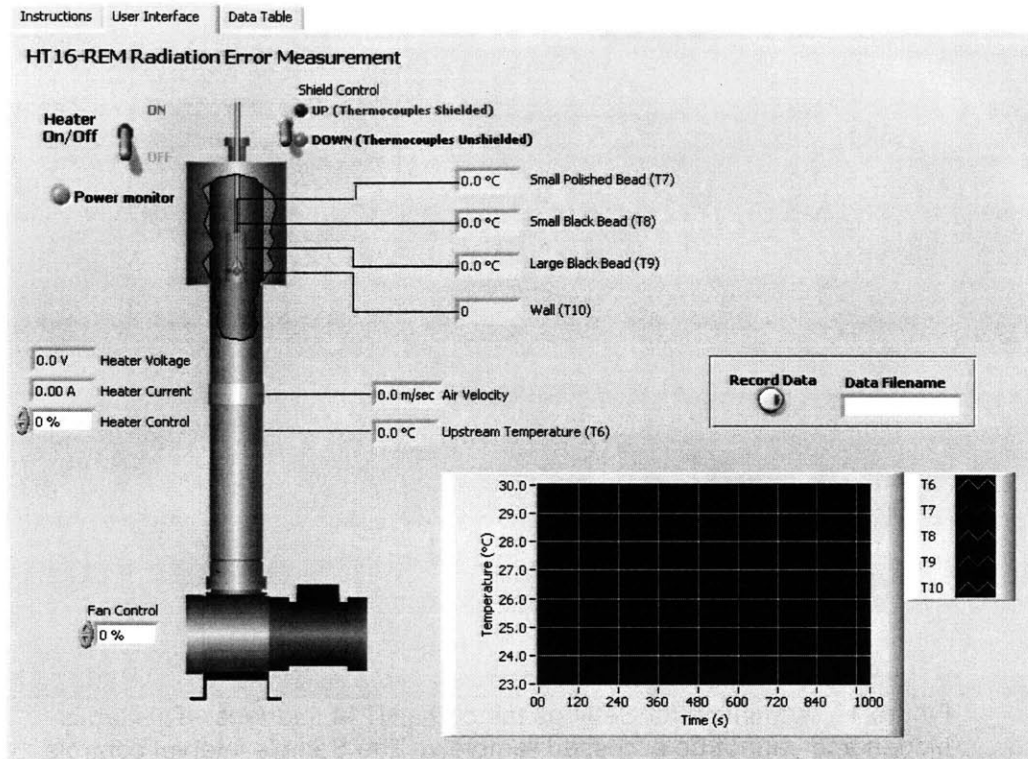


Figure 20: User Interface for the HT16. Like the HT14, it has controls for the heater and the centrifugal fan, a graph of the temperature profile, and a write to file function. Unlike the HT14, it has additional thermocouple readouts and a control for the radiation shield.

#### 4.2.5 Code

Like the interfaces, the code behind the LabVIEW panels is extremely similar for both the HT14 and HT16. The LabVIEW code shown is from the HT16. Remember that LabVIEW “code” consists of a block diagram wired together like chips on a breadboard. Figures 21 and 22 show the two major aspects of interacting with the equipment, measured values and control, respectively. In Figure 21, Read Analog code components read values out of the data stream of the USB port. This value is always between -2048 and +2048, no matter what sensor is reporting it, so it must be scaled appropriately. All of the data values then are connected to their respective indicators as well as the temperature profile graph in the center and the Data Table tab in the upper right. The region of pink wires on the right side is where data is written out to the CSV files for the students. Figure 22 has digital controls which turn the heater power on and off and raise and lower the shield, and also analog controls which are used to control the amount of power to the heater and fan. The grey rectangles around the code is a representation of a while loop. The code will continue to execute until the loop condition is false. In this case the code will run continuously.

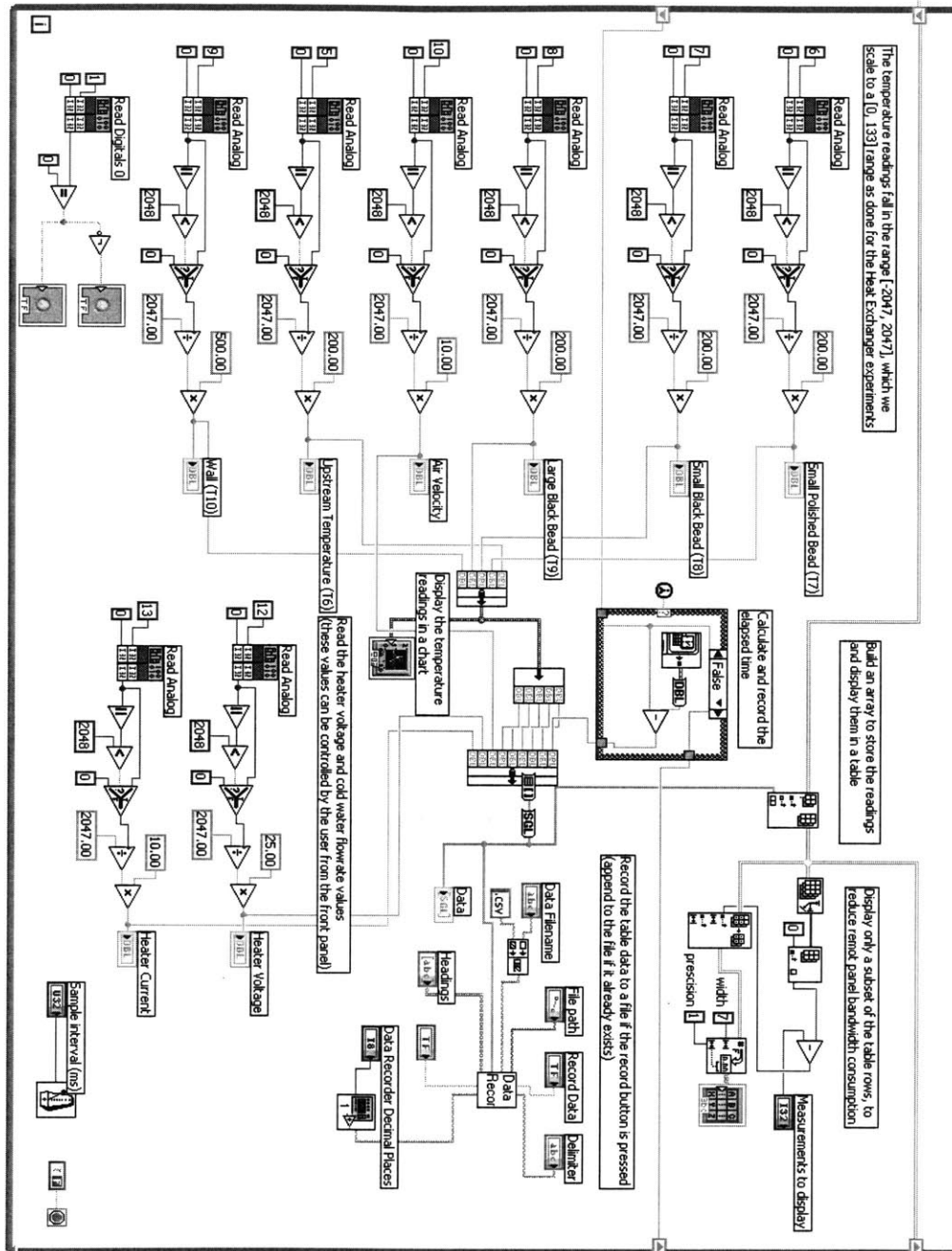


Figure 21: LabVIEW code for reading measurements. Each of the Read Analog functions returns the value of one channel on the USB port, which then gets scaled to its actual measurement value. Blue wires represent integers, orange ones are floating points, green is for Booleans, and pink is for strings.

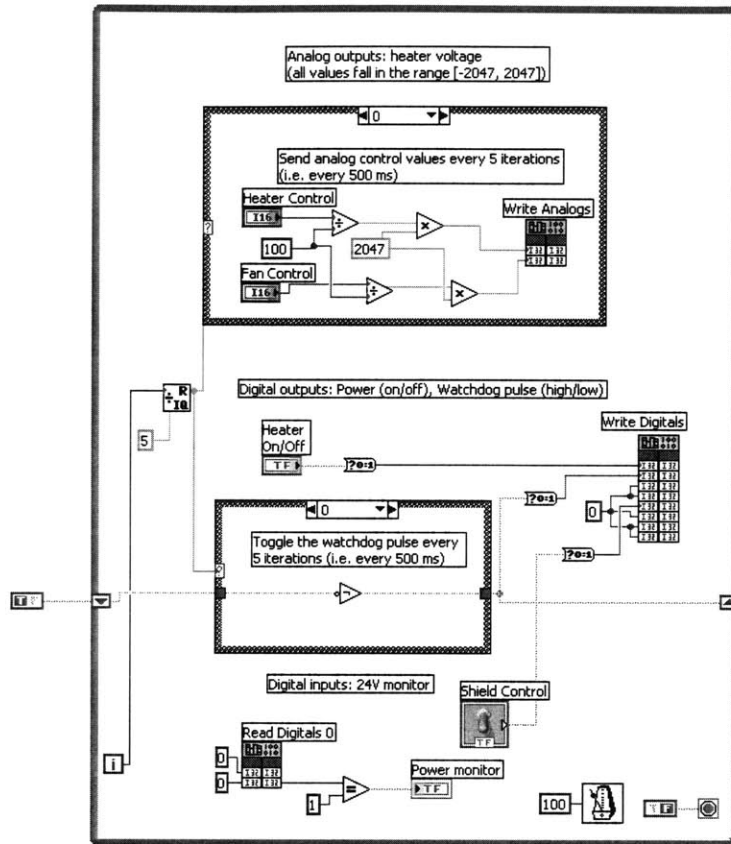


Figure 22: LabVIEW code for controlling equipment. The Write Digitals function outputs Boolean values, such as the position of the radiation shield

#### 4.2.6 Web Access

Both interfaces can be accessed via links on the iLab Heat Transfer Project website at <http://heatex.mit.edu/HeatexWeb>. When the students access the page and perform the experiment, they will see something similar to Figure 23. The Collaboration window at the top of the screen is a Java applet that allows teams of students to communicate and perform the experiment simultaneously from different locations. In the figure, the heater was turned on at 30% power until the cylinder reached 70°C. At that time, the fan was turned on at 40% power and the temperature began to exponentially decline toward an equilibrium value. The data was being written to the file radiation\_test001.csv, which would be available at [http://heatex.mit.edu/data/10302/rad\\_test001.csv](http://heatex.mit.edu/data/10302/rad_test001.csv) assuming this experiment was performed as part of MIT course 10.302. The HT16 web interface is nearly the same as the HT14 web interface in Figure 23 and is not shown.

## Radiation with Convection Experiment

**Login**

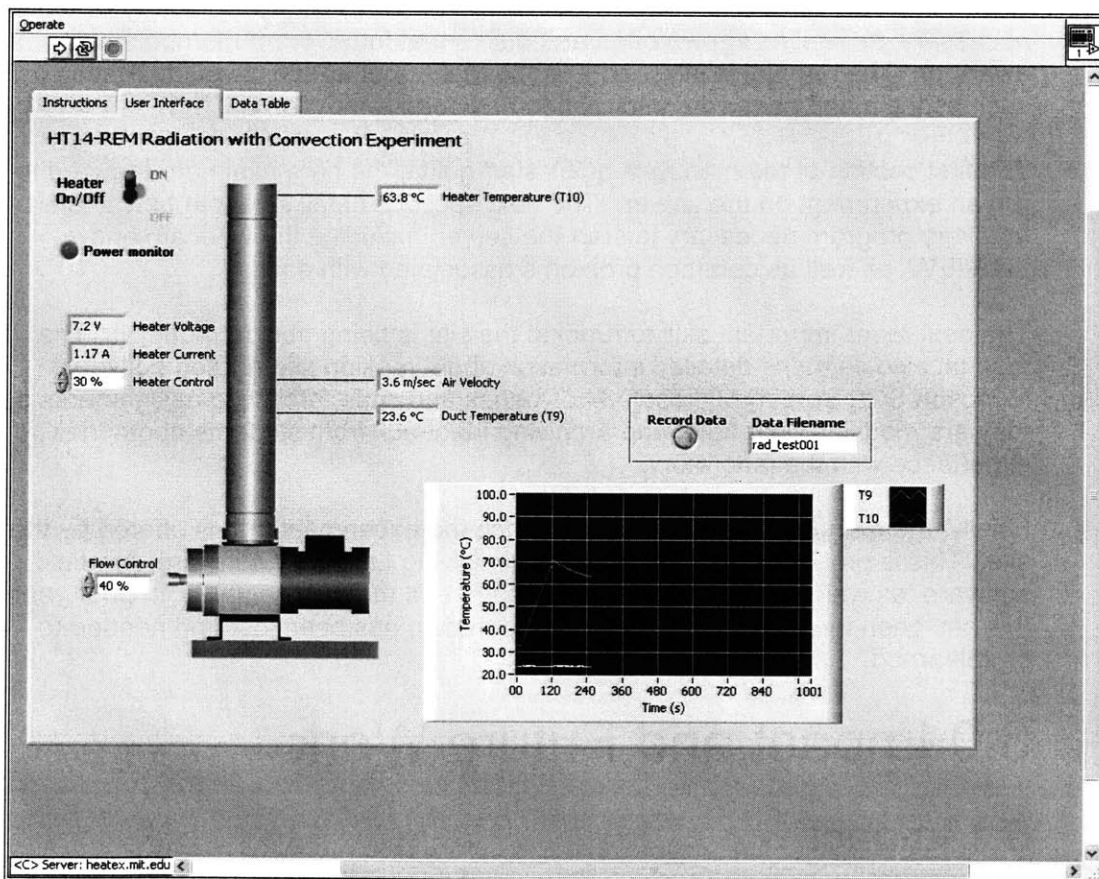
login ID :

Password :

Status :

**Collaboration**

Type your message here:



Experiment not working?  
 email [dsaylon@mit.edu](mailto:dsaylon@mit.edu)

© 2005 MIT Department of Chemical Engineering, all Rights reserved. Last Modified August 2005.  
 Contact the [administrator](#) for questions or comments about this website.

Figure 23: Web interface for the HT14. This figure shows the state of the interface after the heater has been left on at 30% until the temperature of the cylinder reached 70°C, at which point the fan was activated at 40% power. The resulting temperature curve is exponentially approaching the equilibrium value.

## 5.0 Administrator Manual

The team working on the iLab Heat Transfer Project site changes every year. This creates a problem because several components of the system, notably the LabVIEW interfaces and the Armfield service units. In the past this has led to the need for new team members to essentially relearn what was already known in years past.

As a solution to this problem, an administrator manual has been created which contains a full explanation of how to run the site under normal conditions. It also contains details of how to make modifications to the system should they become necessary, as well as known bugs and quirks that future team members must be aware of. The manual is included in Appendix A, but all sensitive information such as login and password information has been removed.

The first section of the manual a quick start guide, the bare minimum required to run an experiment on the server. The next sections detail in length how to start up every program necessary to start the server, including the webcam and LabVIEW, as well as common problems associated with each.

The next more important skill to running the site is being able to modify it. The manual also contains detailed information about making site-related edits for Microsoft SQL Server, Microsoft .NET pages and code, archiving assignments as they are modified over time, and archiving feedback from students about their experience with the laboratory.

Finally, the manual discusses how to switch the experiment that is offered by the site. This is probably the most complex, involving LabVIEW software, Armfield software, as well as settings on the USB port and the service units. In prior years, this has been the primary area where information has been lost and needed to be relearned.

## 6.0 Impact and Future Work

### 6.1 Impact

Over the past year, several significant developments have been made on the iLab Heat Transfer Project with relation to this thesis. These are:

- A tool for adding an entire class of students to the site as users simultaneously was created, giving students immediate access to their experiments and simplifying the explanation required from teaching assistants and professors.



- Additional functionality was added to the online questionnaire where on every successful submission, a whole and complete copy of the questionnaire would be saved, reflecting the student's responses. The questionnaires can be easily retrieved and printed out as a hard copy.
- A webcam was added to the site via a Logitech QuickCam Pro 4000 and Webcam Tracker Live server, allowing the students to see and hear the experiment equipment as they interact with it, even in low-bandwidth areas.
- Two additional experiments, the HT14 and HT16 radiation experiments, were added to the project. They are both fully functional in terms of the remote access to the devices and only need assignments before they are ready for use in a class.
- An administrator manual was created to unify the knowledge of the past years and the present system. Even if the members of the site change every year, the manual is designed to allow people with minimal computer skills to run and edit the site.

## 6.2 Future Work

As part of the iLab Initiative, the Heat Transfer Project would ideally be integrated into their software architecture. This would allow the site and experiments to reach more universities and more students and improve the quality of sharing experiments.

There was an attempt made to integrate the iLab Heat Transfer Project into the iLab architecture during the past year, but unfortunately their development team was unable to make sufficient progress with their service broker prototype for integration to be feasible and the process had to be delayed. The service broker may be ready in the coming year for integration.

Another potential for future work would be fully integrating Clipstream, a different kind of webcam server into the site. Unlike Webcam Tracker Live, which requires a third party client to view the stream, Clipstream plays through a Java window, which is platform-independent. This functionality allows the stream to be embedded into a page. This software had not been working properly on the site due to USB bandwidth issues with the experiments.

# APPENDIX A – Administrator Manual

## Administrator Manual for MIT iLab Heat Transfer Project ([heatex.mit.edu](http://heatex.mit.edu))

Written by  
David Saylor

September 8, 2005

# 1.0 Introduction

This manual is for research assistants (RAs) working under Clark K. Colton for administration of the server used to conduct remote experiments in MIT class 10.302-Transport Processes. The server mentioned in this document is the Heatex server, accessible at <http://heatex.mit.edu>.

This document assumes a basic knowledge of understanding written code and syntax in HTML and Microsoft Visual C++, or it will be difficult to maintain and expand the existing applications.

Additional skills that will be necessary are CSS style sheets, ASP web forms, and SQL querying. Here are some basic tutorials which may be of help. Details about how these relate to the site will be covered later. There are many more tutorials available through Google searches, so please seek out additional material if necessary.

-W3Schools CSS Tutorial <http://www.w3schools.com/css/default.asp>

-W3Schools ASP Forms Tutorial [http://www.w3schools.com/asp/asp\\_inputforms.asp](http://www.w3schools.com/asp/asp_inputforms.asp)

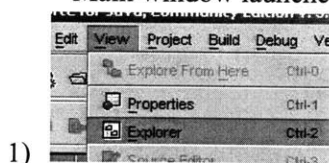
-W3Schools SQL Tutorial <http://www.w3schools.com/sql/default.asp>

## 2.0 Starting the Server

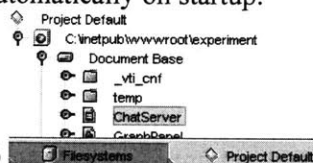
First, this is the quick start section, the minimal amount of work that needs done to set up the computer to have an experiment ready for the students. If anything goes wrong at any stage of this, see the more detailed startup instructions in that software's specific startup in this section.

### Quick Start:

- Java Forte
- \* Main window launches automatically on startup.



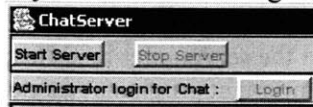
- 1) View → Explorer



- 2) Project Default tab → Highlight ChatServer



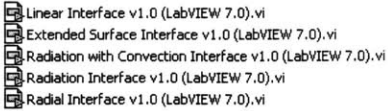


- 3) Press Execute button or F6




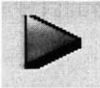
- 4) Press Start Server and then Login

- LabVIEW



- 1)   Choose interface's folder on the Desktop
- 2)  Double click on the right interface

- Webcam

\*There are two webcam servers installed. You can only have one running.

- 1)  Click Webcam Tracker Live icon on Desktop
- 2)  Click Start

OR

- 1)  Click the Clipstream Live icon on the Desktop
- 2)  Click the START/STOP button

## 2.1 Getting into the room

The server is located in the sub-basement of Building 66. Take the elevator to SB. The server room is the chemistry lab directly to your right after you step out of the elevator (not the conference room with blue chairs). The combination to the door is XXXXXX as of September, 2005. The combination may change from time to time. Contact Steve Wetzel, the Facilities Manager (see section 6.0) to get the most recent combination. Important: Full-length pants and protective eyewear are required at all times while in the room, as it is a chemistry lab. This includes times when you are alone in the lab and no experimental equipment is in use. Failure to do so will definitely result in complaints by those conducting experiments in the adjacent chemistry labs.

## 2.2 (Re)Starting the server

The server takes a considerable amount of time to start up. After the startup is finished, Windows should launch automatically.

If you are planning to restart the server after it has loaded Windows, you must make sure that no one is running any important programs remotely before shutting down the server. You can see what other processes are running by right clicking on the Taskbar and selecting **Task Manager**. You can see all the users who are logged in, including remote logins via the **Control Panel**→**Administrative Tools**→**Terminal Services Manager**. If there is someone logged in, get permission from them before restarting.

## 2.3 Logging In

The login screen should ask for you to press Ctrl+Alt+Del in order to log into the server. Do so to bring up the login screen. As of the date this manual was written, the username and password were as follows:

Login: [REDACTED]

Password: [REDACTED]

If this is not the correct information, contact Clark Colton in 66-452 for the new login information.

## 2.4 Starting the chat server

Java Forte should load by default when Windows starts. If it does not, access it from the Start Menu. Sometimes the program starts overloading the computer while it tries to load, for unknown reasons. If the interface does not load for a long time, right click on the Taskbar and select **Task Manager**. Find the process **runide.exe** and **End Process**. Then start Java Forte manually.

Select **View**→**Explorer**, as shown below. Under the **Project Default** tab, find the **ChatServer** application, and hit the **Execute** button (single green arrow, F6). This will bring up the chat interface. Press the **Start Server** button and the application will now be active. This is all shown in Figure 1.

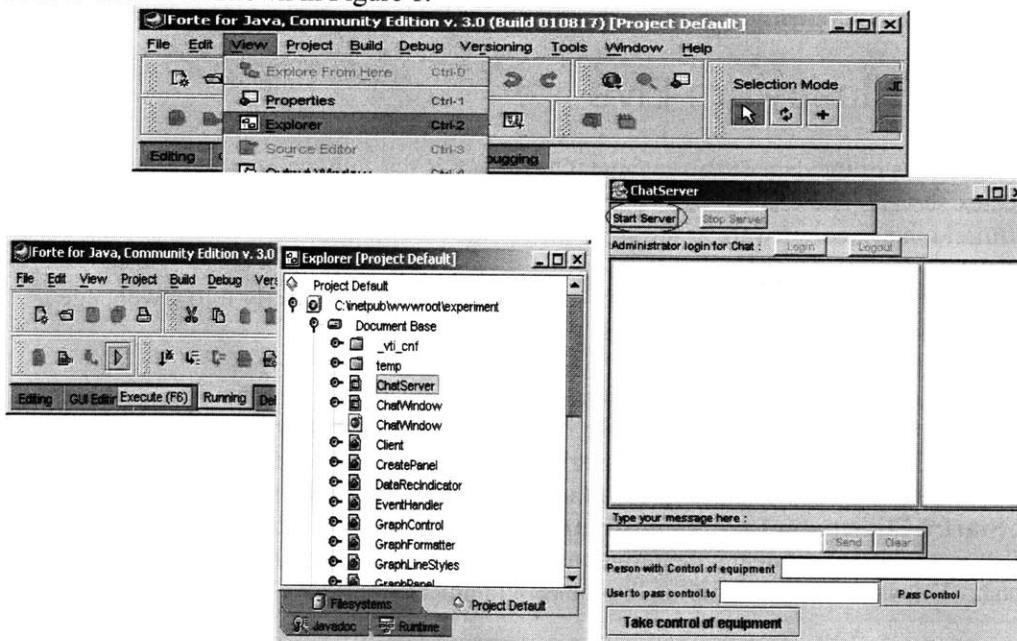


Figure 1. The steps to starting the Java chat server through Forte.

## 2.5 Starting LabView VIs

After starting up the Java chat server, the LabVIEW virtual instrument (VI) interfaces must be started. Once the VI is loaded, LabVIEW's internal web server will be active and the experiments will be available remotely through the website.

The VIs are in two folders on the desktop:

- **Heat Exchanger Files 7.0** contains two of the interfaces:
  - **Shell & Tube Heat Exchanger v1.4 Live (LabVIEW 7.0).vi**
  - **Flat Plate Heat Exchanger v1.4 Live (LabVIEW 7.0).vi**
- **Heat Conduction Files** contains the other five interfaces:
  - **Linear Interface v1.0 (LabVIEW 7.0).vi**
  - **Extended Surface Interface v1.0 (LabVIEW 7.0).vi**
  - **Radial Interface v1.0 (LabVIEW 7.0).vi**
  - **Radiation Interface v1.0 (LabVIEW 7.0).vi**
  - **Radiation with Convection Interface v1.0 (LabVIEW 7.0).vi**

## 2.6 Starting the WebCam

We use a Logitech QuickCam Pro 4000 to stream video and audio of the experiment as a webcam feed. There are currently two webcam stream servers installed on the site, Clipstream and Webcam Tracker Live. Clipstream is a more advanced program and allows integration of a Java window to watch the stream. Webcam Tracker Live gives much more control over the video and audio settings, which is a major advantage when streaming internationally to low-bandwidth clients.

### 2.6.1 Clipstream Live

\*\*\*Note: As of the date of this manual, Clipstream is not fully compatible with the Heatex server. There is not sufficient USB bandwidth to run both the webcam and the Armfield interface software at the same time. One potential solution which has not yet

Clipstream Live allows the integration of the webcam stream directly into a webpage without use of a client or extra codecs.

First, make sure the webcam is positioned to get a good shot of whichever experiment is being run. The cord is not that long, so it may be necessary to move the equipment around. The webcam software is Clipstream Live by Destiny Media Technology (<http://www.clipstream.com>). To start the program, use either the shortcut on the Desktop or **Start→Clipstream Live→Clipstream Live**. If you get an error saying that DirectX 8.1 is not installed, it means the program was unable to initialize the webcam, and it will crash. This is poor error handling by the programmers.

To configure the webcam source, select **Settings→Capture Devices** and set both fields to be the Logitech QuickCam Pro 4000. Under **Settings→Server Settings**, the field

Specify IP for Codebase should be set to the IP of the server, and the port should be set to 83, as shown in Figure 2. As of the date of this manual, the IP of heatex.mit.edu was statically assigned to 18.63.3.246, but to get the IP at any point in time, click on Start→Run and enter **cmd** and press Enter, and at the prompt that comes up type **ipconfig** and Enter, and then find the field IP Address. Also under the Server Settings is an option to log the stream of the webcam, but there is not sufficient room on the server for that kind of storage, so leave it unchecked.

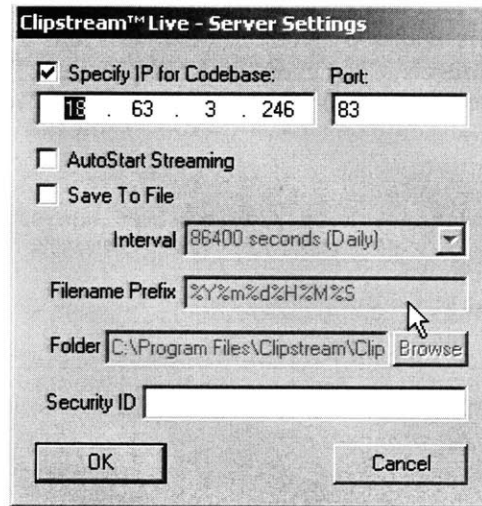


Figure 2. The Server Settings for Clipstream Live. The IP for Codebase should be the server's IP and the Port is set to 83 by default.

The **Settings→Advanced Server Settings** panel is an extremely difficult form of IP restriction. Using this panel, it is possible to either restrict access to the IPs on the list, or deny access to the IPs on the list. Do not alter these settings unless you know what you are doing. With no entries in the list, all users are accepted. The IP system this panel uses is CIDR, which does not support the wildcard (\*/? ) format more common in general use. In CIDR wildcards, you say how many of the top bits have been specified, and leave the rest of the numbers as 0. For example, the IP of the server is 18.63.3.246, which is a full IP address of 32 bits under IPv4, so it is equivalent to 18.63.3.246/32 under CIDR since the top 32 bits are specified. To specify 18.63.3.\*, you only want to specify the top 24 bits, thus it is 18.63.3.0/24 under CIDR. Likewise, 18.63.\*.\* would be 18.63.0.0/16. CIDR does not support expressions such as 18.\*.3.246 that have a wildcard in the middle with a single expression.

Two useful IP might be:

- 18.0.0.0/8            All of MIT
- 131.111.0.0/16      All of University of Cambridge
- 18.63.3.246         Only the server
- 18.63.0.0/16        Only computers in Building 66

To configure the server to accept connections from High, Medium, and/or Low Bandwidth users, use the **Live Streams** pull-down menu.

Clicking the START/STOP button shown in Figure 2-1 will start the webcam feed if there are no errors. Our license only supports five simultaneous users watching the stream, so make sure the students are aware of that. If there are too many, they may be denied access.

To integrate the webcam feed into a webpage, you must insert the generated HTML code into the page. The code is available if you click **Live Streams**→**View Web Page** and selecting **View**→**Page Source**. Copy everything between the `<!-- begin` and the `<!-- end` tags into Notepad, and add the attribute `codebase="http://heatex.mit.edu:83/"`, as shown in the code below:

```
<!-- begin Clipstream Live applet -->
<Applet MayScript codebase="http://heatex.mit.edu:83/"
  cabbase=clipstreamlive.cab archive=clipstreamlive.zip
  code=clipstreamlive.class
  alt=ClipstreamLive name=clipstreamlive
  width=320 height=256>
<param name='AutoBitrate' value='true'>
<param name="VideoURL1" value="High,555,/live256.vcs">
<param name="VideoURL2" value="Medium,150,/live128.vcs">
<param name="VideoURL3" value="Low,0,/live044.vcs">
<param name="AutoPlay" value="True">
<param name="VideoBufferSize" value="5">
<param name="BackgroundColor" value="#FFFFFF">
<param name="TextColor" value="#000000">
<param name="SeekEnable" value="false">
<param name="PanelEnable" value="True">
<param name="PanelImagesURL" value="buttons.gif">
<param name="PanelLayout" value="PMC">

<param name="PanelImageWidth" value="51,1,52,1,1,1,1,20,75">
<param name="PanelFlash" value="2">
<param name="PanelPopup" value="False">
<param name="PanelPosition" value="Bottom">
<param name="VideoTitleImageURL" value="title.gif">
<param name="VideoEdgeColor" value="#FFFFFF">
<br><A HREF=http://www.clipstream.com/help/visitorhelp.shtml>Clipstream
Live requires Java enabled in your browser or email. Click here for more
information.</A><br></Applet>
<!-- end Clipstream Live applet -->
```

Figure 3. HTML code to insert Clipstream Live's feed into a webpage. This code is automatically generated by the Live Streams→View Web Page function, but the codebase attribute, shown in bold, must be typed in manually.

Any page this code is inserted to will display the webcam feed. As a reminder, the LabVIEW interface HTML files are in C:\Program Files\National Instruments\LabVIEW 7.0\www.



## 2.6.2 Webcam Tracker Live

This software needs a third party client to view the stream, but offers more customization.

First, make sure the webcam is positioned to get a good shot of whichever experiment is being run. The cord is not that long, so it may be necessary to move the equipment around. The webcam software is Webcam Tracker Live, by Sytaxis Software (<http://www.sytaxis.com>). The launch shortcut for this is **Start→Webcam Tracker Live→Webcam Tracker Live** as well as on the Desktop.

Click on the gear icon, shown in Figure 4, to be sure the configuration is correct. The configuration window may take some time to load. The Video input should be set to “Logitech QuickCam Pro 4000”, and the Audio input to “Logitech Microphone (Pro 4000)”. Under the Source button of the Video input, the Frame Rate should be set to 5.000 and the Output Size (the resolution) to 160x120. We do not need any quality in the feed, nothing will be moving, and the only thing they will be hearing is the sound of a motor. This will greatly conserve bandwidth on the USB port as well as when streaming to places such as Nigeria or Argentina. The Audio format should be set to 8kbps, 8000kHz, mono (single microphone input), again to conserve bandwidth use.

Interacting with the configuration seems to take an extremely long time, especially when setting an audio source. We do not know why. When all the settings are in place, press the Start Server button and you should see both video and audio signals displayed. The URL to access the webcam feed is <http://heatex.mit.edu:8080> and the server IP and port will be displayed in the bottom right of the main tracker window. We suggest that the students use VLC, available at <http://www.videolan.org/vlc/>, because it can decode our stream on any OS, and is available for free download.

If you get a message saying there is not enough bandwidth available on the USB port, you should close the applications which use the USB, including LabVIEW and the webcam server, and then restart them. The webcam is usually greedier with bandwidth so it may be prudent to restart LabVIEW first.

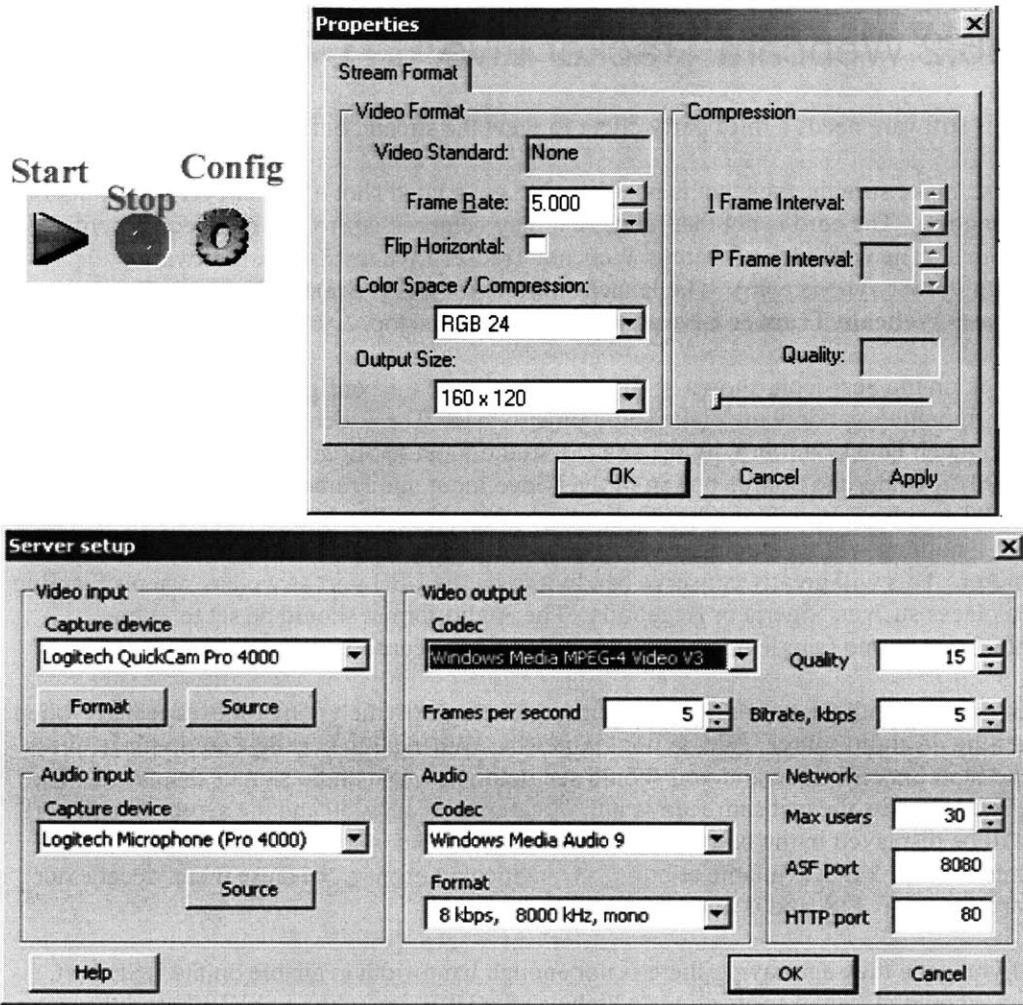


Figure 4. Webcam Tracker Live software interfaces. At the top left are the Start/Stop/Configure buttons, then the Audio Source configuration on the top right, and finally the overall Configuration settings on the bottom. These settings are designed to minimize the outgoing bandwidth because several bandwidth-limited foreign countries will be trying to access it.

### 2.6.3 Advanced Problems

The drivers for the camera we are using have shown themselves to be unstable. Sometimes the drivers do not load on startup. If you start up the webcam server and do not see a value for the Video or Audio input Capture device, the driver did not load. Try disconnecting and reconnecting the camera and any other USB devices and then starting up the Logitech QuickCam software from the Desktop. If you still can't get any signal from the camera, you may have to try reinstalling the drivers from

<http://www.logitech.com/index.cfm/downloads/software/US/EN,CRID=1794,contentid=6023> and then restarting the computer. This should fix the problem, but remember to reset the configuration for the software.

If something should happen that one of the webcam servers need reinstalled, download the locked, trial versions from their individual sites, listed above. Our keys to unlock the full versions are:

Clipstream Live: License Key: [REDACTED]

Webcam Tracker Live: Name: [REDACTED] Reg. code: [REDACTED]

## 2.7 Remote Access

The main method of remote access to administrating the server remotely is the **Remote Desktop Connection** application available in the **Start→Accessories→Communications** folder of Windows 2000 and XP. Connect to the host **heatex.mit.edu** with the same user and password you use to log in:

Login: [REDACTED]  
Password: [REDACTED]

There is also a VNC server (TightVNC) which loads automatically on the server. Download the client from a site at [http://prdownloads.sourceforge.net/vnc-tight/tightvnc-1.2.9\\_x86\\_viewer.zip?download](http://prdownloads.sourceforge.net/vnc-tight/tightvnc-1.2.9_x86_viewer.zip?download) and extract only the .exe file. This client does not require installation. Enter **heatex.mit.edu** as the VNC server and enter the same password as above. Important: VNC will only work if there is a login active on the console. Otherwise it will give a “VNC authentication failed” error, which is the same message it reports for an incorrect password. This message is shown in Figure 5. If you are typing the password correctly and getting this message, you must go down to the server console and log in to create an active login session.

You can do any tasks remotely that you would normally do at the actual console, but do not start a LabVIEW experiment interface for the website, or the chat server, or the webcam on a Remote Desktop Connection session to be used by the students. Multiple instances of any of these programs can cause instability in the system. When you logout, be sure to choose **Log out Administrator** from the pull-down menu, and not Shutdown.

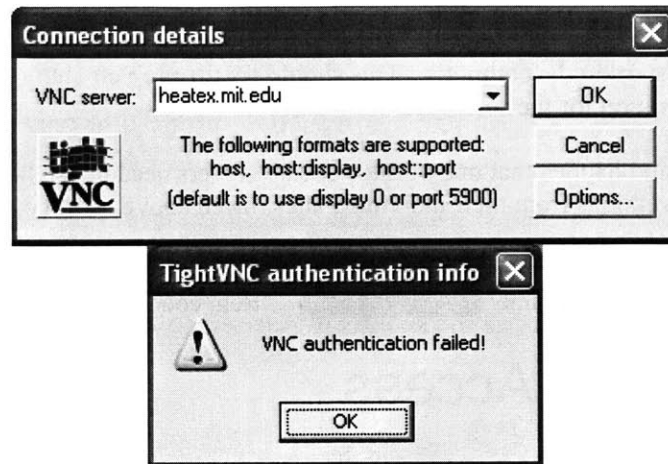


Figure 5. The Tight VNC client interface and the error message received when there is not a session active on the console.

## 3.0 Overview of the site

This section will discuss briefly the organization of the site and the support and service applications we have running.

### 3.1 The Menu Bar

This is the main way to access individual pages of the site. Depending on the user who is logged in, there are several links which will be grey, indicating that a login is required to access them. If the link is beige, it does not require a login to view. If someone who has not logged in attempts to access one of the grey pages directly through the URL, they will be asked to log in. This is handled through .NET. The menu bars seen by a guest user, a logged-in user, and an administrator are shown in Figure 6.

<b>Home</b>	<b>Home</b>	<b>Home</b>
<b>Access</b>	<b>Access</b>	<b>Access</b>
Login	Edit Registration	Edit Registration
Register	Log out	Log out
Class Teams	Class Teams	Class Teams
Schedule	Schedule	Schedule
<b>Website</b>	<b>Website</b>	<b>Website</b>
About	About	About
Team	Team	Team
Publications	Publications	Publications
Archive	Archive	Archive
<b>Experiments</b>	<b>Experiments</b>	<b>Experiments</b>
Equipment	Equipment	Equipment
Theory	Theory	Theory
Instructions	Instructions	Instructions
Perform Experiment	Perform Experiment	Perform Experiment
Questionnaire	Questionnaire	Questionnaire
		<b>Admin Functions</b>
		Edit Courses
		Edit Timeslots
		View All Users
		Add Users
		Admin Manual

Figure 6. The Menu Bars seen by a guest user, a logged-in user, and an administrator, respectively.

## 3.1.1 User Pages

These are the pages which can be viewed by students and any other person who has an account to use the website.

### 3.1.1.1 Home

At the top of the menu bar is a link back to Home. This page is also the one most frequently used for redirects such as logging in and logging out.

### 3.1.1.2 Login/Logout

This link appears as Login if the user is not yet logged in (a “guest” user), or as Logout if they have already logged in. Login will first take them to a page which requests their

username and password, and then return to the home page. Logout will first pop up a dialog box confirming that the user wants to log out, and then return to the home page with the login session removed.

### 3.1.1.3 Register/Edit Registration

This link appears as Register to guest users and Edit Registration to users who are logged in. Regardless of whether the user is logged in or not, this link leads to **Secure/NewUser.aspx**. The code behind this page automatically checks if they are logged in. If they are, the page displays their current information and allows them to change it. If not, it lets them create a new account. After a new account is created, it will ask them to log in with their new information.

### 3.1.1.4 Class Teams

This page requires a login to view. This page will list all of the teams currently registered under a particular course. It does not list the current members of those teams. These teams are added through the Edit Courses link, in section 3.1.2.1.

### 3.1.1.5 Schedule

This page requires a login to view. The page lists the current times available to run the experiment and allows students to sign up for specified time periods to run the experiment. The time slots which are available can be changed via the admin Edit Timeslots link in section 3.1.2.2. There is no correlation between being signed up for a time slot and being able to run the experiment. This page exists only as a logistical tool. It is still possible for people to access the interface at times which they are not scheduled for if they enter the URL of the experiment interface directly.

### 3.1.1.6 About

This page gives visitors and users an overall description of this project, which is currently called the iLab Heat Transfer Project, under the iLab initiative.

### 3.1.1.7 Team

This page lists the current and former website team members who worked on this project.

### 3.1.1.8 Archive

This page is the archive for every assignment performed via the Heatex website under the year, the term, and course during which it was completed. Classes which performed more than one experiment during a term will have multiple entries in the archive for that term

### 3.1.1.9 Equipment

This contains general descriptions of all the heat transfer devices and service units we have associated with the iLab Heat Transfer Project. There is a QuickTime movie demonstrating the heat exchanger. There are also links to the manuals, which are available from Clark Colton, or under the Manuals folder on the Desktop.

### 3.1.1.10 Theory

This page contains links to derivations and solutions for theoretical problems that are useful in the analysis of some assignments that are not typically found in textbooks.

### 3.1.1.11 Instructions

This page contains the instructions that apply to running the experiments. The first section contains general instructions that apply to all of the interfaces, such as installing Java and the LabVIEW plug-in and using the webcam feature. The second section contains instructions specific to the individual graphical user interfaces for each of the different pieces of equipment.

### 3.1.1.12 Perform Experiment

This page requires a login to view. It contains only redirecting links to the individual interfaces. This is the extent of security on the Heatex site. Because a login is required to access this page, and because the links to the interfaces are not published anywhere else, only people with accounts will be able to access these direct links. These links are displayed in the address bar once the user accesses the interface. Take care not to make these URLs public.

### 3.1.1.13 Questionnaire

This is the means by which the project receives its feedback as part of educational assessment. After they perform the experiment, each student is responsible for filling out a questionnaire.

## 3.1.2 Administrator Pages

These are the pages which only administrators are allowed to view. Users in the database have the IsAdministrator column which is 0 if they are a normal user or 1 if they are an administrator. If someone who has not logged in attempts to access one of these pages directly, they will be asked to log in, as they were for the restricted user pages. If someone who has already logged in who is not an administrator attempts to access one of these pages directly by typing in the URL, they will be redirected to the home page.

### 3.1.2.1 Edit Courses

This page is where new courses can be added to access the site. When you add a new course, you can specify how many teams it should have. They will be numbered Team 1, Team 2, etc. This page also lets you add and remove teams from courses already on the site. The names can only be changed manually through accessing the Teams table in the Database, which is discussed in section 4.2.

### 3.1.2.2 Edit Timeslots

This page allows the administrator to disable time slots from being available to the Schedule page. The time slots are one hour in duration. It is important to remember, however, that the schedule has no security influence at all and will not be able to stop people from accessing the experiments at times they are not allowed to.

### 3.1.2.3 View All Users

This page lists all of the users currently in the Students table in the Database. This is where you can see someone's e-mail address or change an account to be an administrator. It is difficult to use this page in Firefox due to the way it handles div tags. All the functionality of this page can be performed directly on the table in the Database as well.

### 3.1.2.4 Add Users

This page is used for adding multiple users at the same time. By specifying several students' full name, email address, and team each on a separate line, you can add the whole class at one time.

### 3.1.2.5 Admin Manual

This page contains links to this document in Microsoft Word DOC and Adobe Acrobat PDF formats.

## 3.2 The Database Server

The database used for this site is Microsoft SQL Server. This application is located at **Start→Programs→Microsoft SQL Server→Enterprise Manager**. The data tables for the class are located in **Console Root→Microsoft SQL Servers→SQL Server Group→(local) (Windows NT)→Databases→HeatexData→Tables**. The important tables in this are: Courses, CourseTeams, Questionnaire, Students, and Teams. What each table contains should be fairly obvious. At this point in time, there is very little relation between the tables. The Students table contains all of the information relevant for the day-to-day working of the website. The Questionnaire table also contains critical



data that is used as a feedback agent for the website. The other tables listed are only used by administrators, but they may be useful for future applications.

## 3.3 Windows IIS

This is the default web server which runs on port 80. As of the date of this manual, it serves files out of the C:\inetpub\wwwroot\ directory. That means to access the file C:\inetpub\wwwroot\mmc.jpg you use the URL To change this directory go to **Start→Control Panel→Administrative Tools→Internet Services Manager**. Under the “\* heatex” site definition, right click on Default Web Site, go to the Home Directory tab, and change the directory path there. Remember to go back through the site and make sure that any other links which used that directory are still valid in the new directory.

## 3.4 Visual Studio .NET

We use Microsoft Visual Studio to maintain the entirety of the project files and as the primary web server for the site. This software enables us to run server-side, event-driven code in the backend of the web site. The application can be launched from **Start→Programs→Microsoft Visual Studio .NET→Microsoft Visual Studio .NET** or the shortcut on the Desktop. The project file is called **HeatexWeb** and should show on the Start Page once the application loads.

The MS .NET programming language is similar to C-style languages, including C, C++, Java, Python, and Perl. If you know one or more of these languages, you should not have any difficulty reading the syntax. The language uses C-style comments with // and /\* \*/. Some additional functionality is the ability to hide regions of code to reduce the amount of code you have to view at any one time. This is done automatically for blocks of code enclosed with { }, and you can do it manually by setting #region and #endregion lines.

Our page consists of ASPX pages, each of which has a C-code CS page associated with it. Any time you make changes to the ASPX files, you only have to Save them for the changes to be shown on the site. If you make changes to one of the CS code pages, you must select **Build→Rebuild HeatexWeb** from the menu bar before you will see the changes. Rebuilding the site will automatically save all the modified files. There are several other configuration options listed in the **Web.config** file which control the built-in web server included in .NET. Changes to this file do not require a rebuild.

By default, .NET can only access files inside its directory. To give it permission to modify other files or directories, you must right click on the target file or directory and select Properties. Under the Security tab, press the **Add** button and select the **ASPNET** user. Give that user Read, Write, and/or Full Control depending on what you want .NET to do.

## 3.5 Cascading Style Sheet (CSS)

Our site uses a CSS located at `C:\inetpub\wwwroot\HeatexWeb\Style\iestyle.css`. This defines the style attributes that will be applied to every page on the site, without having to retype the code.

The **body** name in the `iestyle.css` defines the attributes to be used on every instance of a `<body>` tag. Here, it defines an off-white background, a size 10 font, and a Tahoma font face. This saves having to use the tag

```
<body bgcolor=#F0F1F6 style="Font-Size: 10; Font-Family=Tahoma, Verdana, 'Times New Roman'">
```

on every page, as well as trouble of updating all of them for a minor change. The same is true for the `<table>` tag, which is defined below the body.

Names starting with a '#', such as **#button**, name a specific tag, in this case the side menu. Attributes listed there will only be applied to the single tag with that name, which happens to be `<div id="button">` for us. This cannot be applied to multiple tags.

Finally, the names starting with a '.' such as `.TextBox` are class names. If you create a textbox, such as

```
<asp:textbox id="Name" cssclass="TextBox"></asp:textbox>
```

it will carry the attributes defined under the `.TextBox` class. This is the middle ground between the # which apply to only one element and the names without a prefix which apply to all elements. With the '.' class, you can pick and choose which elements will receive the style attributes.

## 3.6 LabVIEW 7.0

Our LabVIEW Virtual Instrument panels (VIs) are programmed for LabVIEW 7.0, a product of National Instruments. Due to some rather poor software development practices by National Instruments, VIs programmed for version 7.0 will not run under version 7.1 or higher. Perhaps this will change in a later version, but for now all LabVIEW applications must be nothing else except version 7.0.

## 3.7 LabVIEW Web Server

LabVIEW itself also has a web server that it uses to serve VIs, which normal web servers like Windows IIS are unable to do. The web server is only active while LabVIEW is running. There is documentation available at <http://heatex.mit.edu:82/> when LabVIEW is

open. The web server is configured to run on port 82, which means any address for a LabVIEW page must contain :82 after the server name when entered into an internet browser, as the documentation URL does. When no port is specified, the browser will automatically assume port 80, which is the default HTTP port, which is what Windows IIS, the web server for the rest of the site, runs on. The web server's default directory is **C:\Program Files\National Instruments\LabVIEW 7.0\www**. This is where you will find the pages that serve the experiment interfaces. To access, for example, the file `radiation_interface.html` in `C:\Program Files\LabVIEW 7.0\www`, you would access the URL [http://heatex.mit.edu:82/radiation\\_interface.html](http://heatex.mit.edu:82/radiation_interface.html).

## 4.0 Website Administration

This section will describe how you are to interact with the various aspects of the web site. There will also be an example of adding to the web site a new page that interacts with the database which builds continuously through each section. To be made an admin for the website, you must create an account at <http://heatex.mit.edu/HeatexWeb/Secure/NewUser.aspx> and then have an existing admin, either Clark or one of the previous administrators, upgrade your account.

### 4.1 New Users

Right now the most efficient way to add students to the database is to use the Add Users page under the Admin Functions. It is possible for the students to sign themselves up, but this tool gives more control to the admins over making sure everyone has an account. Using the Add Users tool page, everyone will have the same set password and their own Athena username as their login. But either way, each user will be assigned a globally unique identifier (GUID) which will be used as their primary key in the database. The GUID lets us restrict the users' actions, such as not allowing them to submit the Questionnaire multiple times.

### 4.2 The Database

If you want to extract data from the database, it is fairly simple. Open Microsoft SQL Server as described in Section 3.2. Right click on the table whose data you want and select **Open Table→Return all rows**. You can either select the data you want (it must be in continuous rows), or you can perform an SQL query using the [SQL] button in the menu bar. After you select the data you want, you can copy it and paste it into Excel.

To add a table to the database, right click on the **Heatex Data→Tables** icon and select **New Table....** You have to give a name for every column in the table, as well as a data type, and a maximum length. Strings are represented by the `varchar` data type, a variable-length character array. The other typical data types are `int` and `float`, but there are many others. Checking the Allow Nulls value for a column means null values

can be inserted into the table. A benefit of not allowing nulls in all the columns is that any SQL query with an incorrect number of arguments will be immediately caught and not create a row with entries that do not make sense. Most of the tables for the Heatex site are done in this manner. For example, try making a table called Test with two columns, TimeStamp and Response. TimeStamp should be a varchar of 50 characters and Response a varchar of 5. This table will be used by our example page later.

If you need to create a new table with the same column settings as an existing one, click on the Diagrams icon above Tables on the left. On the right-hand side, right click and select New Database Diagram. Use the wizard to select the table you want to duplicate and then hit Finish. You will now have a new diagram with the existing table and all of its columns listed. Right click again and select New Table, and give it a name. Now select all of the column names from the existing table, copy them, and paste them into the new table. Now press the Save button and give a name to this diagram. It will then create the new table with the same column names and data values as the original. You may delete the diagram without affecting either of the tables.

## 4.3 Adding New Pages

To add a completely new page to the site, first open the .NET project. To add the new file, click on **File→Add New Item**, and then select **Web Project Items** from the left and **Web Form** from the right. Be sure to type in the name of the file at the bottom before creating it, or you will have to change this name in several places after it is created. This will create two new items on the server, an ASPX page and a CS class. Suppose we created a page called `TestPage.aspx`. The following sections will detail how to incorporate this new page into the site.

### 4.3.1 Changing Access Permissions

By default, any pages you add will require a login to view them. This can be changed by adding the page to the `Web.config` file's list of public pages like this:

```
<location path="TestPage.aspx">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>
```

Now this page does not require a login to be viewed.

### 4.3.2 Page Headers

You must do some additional setup before this page will express the CSS style sheet properly. Click on the HTML view and replace all of the <meta> tags in the header with the following (or you can copy and paste from another ASPX file):

```
<LINK href="Style/iestyle.css" type="text/css" rel="stylesheet">
<meta content="Microsoft Visual#9;Studio 7.0" name="GENERATOR">
<meta content="C#" name="CODE_LANGUAGE">
<meta content="JavaScript" name="vs_defaultClientScript">
<meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
```

The first line is what defines the CSS style sheet to be used in displaying the page. This will make the page display with the same color and size settings of all the other pages on the site. The content of the style sheet will be covered later.

### 4.3.3 Changing Inheritance

Next, right click on the view of the ASPX file and select **View Code**. We have to change the inheritance of this new page's class so that it will integrate with the code running for the rest of the site. Specifying `AdminBase` for the base class will require an administrator login to view it. Specifying `HeatexBase` will allow anyone who logs in to view it. We'll assume `TestPage.aspx` will be a public page. Where it says:

```
public class TestPage : System.Web.UI.Page
```

change it to read:

```
public class TestPage : HeatexBase
```

Doing this will add the side menu bar as well as the login functionality when the page is accessed, i.e. users will only be able to access the page if they are logged in. The page will now look and behave like the rest of the pages.

### 4.3.4 Writing Pages

Static pages containing only text and images are simple to write if you know HTML. Simply add in the text using the proper formatting, and remember that you must use <br> tags instead of carriage returns to start a new line.

#### 4.3.4.1 Dynamic Pages

If you are going to be making a page which will accept input, there are a few things which will make it easier to write the code on the server. Every time you add a web form item, like a text box or a radio button list, be sure to add it to the variables defined in the CS file. So for example, if we insert a radio button list called "rblast" with three options into `TextPage.aspx`:

```
<asp:RadioButtonList ID="rblast" RepeatDirection="Horizontal"
Runat="server">
    <asp:ListItem Value="1">Yes</asp:ListItem>
    <asp:ListItem Value="2">No</asp:ListItem>
    <asp:ListItem Value="3">Maybe</asp:ListItem>
</asp:RadioButtonList>
```

You will get a list with three choices: Yes, No, and Maybe. After you have entered this HTML, click on the Design View of the page and .NET will automatically make the following declaration for you:

```
protected System.Web.UI.WebControls.RadioButtonList rblist;
```

Now, every time you reference `rblist` in the code, it will interact with that specific control. For example, `rblist.SelectedItem.Value()` will return a 1, 2, or 3 (as a string) depending on the choice.

If you are going to have a form for submitting, you need a submit button. We will add one which is labeled “Submit Answers”, and we will call it `SubmitButton`:

```
<asp:button id="SubmitButton" runat="server" cssclass="Button"
text="Submit Answers"></asp:button>
```

Now, go the Design View and double click on the button. This will create a function in the C code to handle the event where the button is pressed. Additionally, a `Button`-type variable with the name `SubmitButton` will be declared and a line of code will be added to the `InitializeComponents()` function inside the Web Form Designer generated code region. This new function, which will be called `SubmitButton_Click` is where you put whatever code you need to execute.

#### 4.3.4.2 Interaction with the Database

If we want to store or retrieve data from the database, there are a few simple steps to follow. You need to import the SQL libraries included in .NET as well as the Configuration variable libraries:

```
using System.Configuration;
using System.Data.SqlTypes;
using System.Data.SqlClient;
```

Then you need to declare an SQL connection, an SQL command, and a string to hold the SQL query string:

```
SqlConnection con;
string sql;
SqlCommand cmd;
```

Next these need to be initialized. For our purposes on the HeatEx site, there are some simple ways to initialize them:

```
con = new SqlConnection(ConfigurationSettings.
    AppSettings["Heatex.ConnectionString"]);
con.Open();
sql = "INSERT INTO [Test] ([TimeStamp], Response) VALUES
    ('Now', 'Yes')";
cmd = new SqlCommand(sql, con);
cmd.ExecuteNonQuery();
```

This code will open a connection to the `HeatexData` database in MS SQL Server, using a path defined within the .NET project code. The `sql` string contains the text that will perform the insertion, and the `cmd` is formed by applying the `sql` string to the `con` connection. Then the `ExecuteNonQuery()` instruction is called because the `INSERT` will not return a value. There is also an `ExecuteReader()` instruction which will return an object of type `SqlDataReader` in response to a `SELECT` statement, like this:

```
SqlReader reader;
Reader = cmd.ExecuteReader(CommandBehavior.CloseConnection);
```

Last, every page that performs a database action should have a try/catch block on every such command, as well as some <asp:Label> tags on the page to display the errors. For example, say we have a label with id “error”. The try/catch block would look like this:

```
try{ cmd.ExecuteNonQuery(); }  
catch (Exception err) { error.Text = err.ToString(); }
```

## 4.3.5 Final Page

Now we can put everything together and get the final result of the TestPage.aspx, which is shown in Appendix A. There are two additional features in the code which were not discussed. The code has a reference to DateTime.Now which is, of course, the current time; and the label named “SQLstring” will display the query string which is sent to the database, purely for illustrative purposes. You should be able to see what every line of code does toward creating the final page.

## 4.4 The Questionnaire

The Questionnaire is one of the most complex and most important pages on the site. It functions as a form, the same way as TestPage.aspx from the previous sections does, only larger. When a questionnaire is submitted, the code checks every answer to be sure it has been filled out with a reasonable answer. If every answer has been filled in with an appropriate value, a complete copy of the questionnaire is saved, in its HTML form via the printQuestionnaire() function, to the **C:\inetpub\wwwroot\HeatexWeb\QuestionnaireResults** directory to which the ASPNET user has been given full access. If the questionnaire is changed, the printQuestionnaire() function must also be updated so it outputs correctly. If you need to print out questionnaires, highlight all of the HTML files which need printed, right click and select Print.

The Questionnaire table in the Database does not allow any null entries, and has its primary key as the student’s GUID. It will return an error if any of the arguments are null, i.e. if the student did not fill them out, or if the student’s GUID already exists in the table, i.e. they are trying to resubmit the Questionnaire. The code handles both of these errors and reports them back to the user. If another, unexpected error, occurs, the code will print out the entire SQL string which was generated and the text description of the error. This may include the Database server being down or the Questionnaire table being the wrong format for the Questionnaire.aspx page.

## 4.5 Archiving

One of the most critical parts to running the site is the archiving of data and pages. These can be used to fall back to a stable site when some aspect of the code has been damaged beyond repair, or when old layouts or information needs to be reincorporated into the current site.

To archive an assignment, go the **C:\inetpub\wwwroot\HeatexWeb\archive** folder, and create a directory named for the term and course, for example **2005-Fall 10.302**. Move or copy the Word doc file and its corresponding pdf file into this directory. Next, make an index.html page inside this new directory with links to both the doc and pdf files. And finally, add a link to this index.html file in the Archive.aspx page.

To archive pages from the interface, to preserve code and layouts, make a directory named for the term or the date in **C:\inetpub\wwwroot\HeatexWeb\Admin Archive** and copy whatever files you want to back up into it. You may want to include any images that the page requires, but avoid backing up files which are large.

## 5.0 Equipment and LabVIEW Administration

This section will discuss how to connect an and administer experiment through LabVIEW.

### 5.1 Connecting an Experiment

\*\*\*Note: This section does not contain information about maintenance or cleaning of the equipment. Please seek help before performing any maintenance on the equipment, and if something needs replaced, contact [support@armfield.co.uk](mailto:support@armfield.co.uk).

To prepare an experiment for use through the HeatEx server computer, first you must make all the necessary connections for each device. The heat exchanger has its own built-in service unit, the HT30XC. The heat conduction, convection, and radiation experiments use an external service unit, the HT10XC. These service units are what the computer communicates with through a USB cable.

The Heat Exchanger must first have its hoses checked for kinks, and every valve, except the drain at the bottom of the tank, must be open. Then the four input hoses must be connected to the Heat Exchanger you are using, two for the cold water circulation and two for the hot water circulation. Then the tank must be filled with filtered tap water, or better distilled water. It may need additional filling if the hoses were initially empty. Finally, the four green thermocouple wires must be attached to the base at their proper positions. Figure 7 below shows the Flat Plate Heat Exchanger connected to the rest of the apparatus, with every valve in the open position (pointing along the length of the



hose). The hot water flows through the red valves, and the cold water flows through the blue valves.

The Shell and Tube Heat Exchanger, shown in Figure 6, will look extremely nearly when connected. The only difference is that the water hoses attach to the shell and tube differently than to the flat plate. Both heat exchangers use the same tank, circulation system, and service unit.

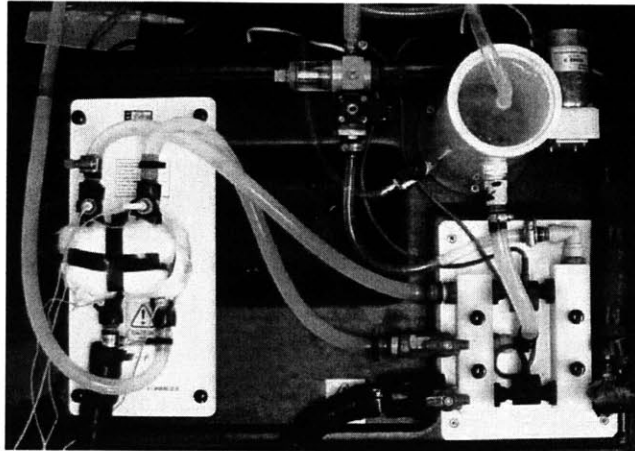


Figure 7. The Flat Plate Heat Exchanger, fully connected. The two red valves controlling the hot water flow rate are in the open position.

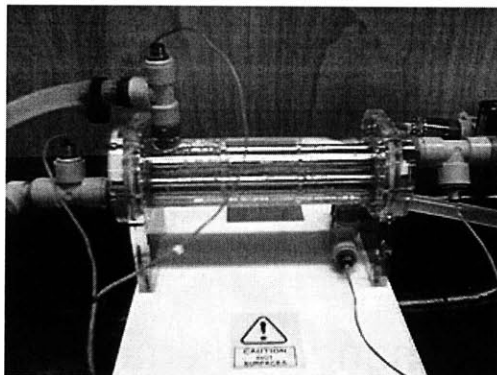


Figure 8. The Shell and Tube Heat Exchanger, one of the two Heat Exchangers used on the HT30XC.

The heat conduction and radiation experiments, including the Extended Surface, Linear Conduction, Radial, and the two Radiation experiments, need to be connected to their service unit, the HT10XC. Generally this involves connecting at least six thermocouples and the power to the heater. Some of the experiments may have additional necessary connections, such as the fan and anemometer for the Radiation experiments. Different devices use different thermocouple sockets as well. Check the device manual to be sure you are connecting the experiments properly. Occasionally when you have connected a

device to the HT10XC service unit and turn on the power, the service unit will short out. If this happens, turn off all the power, reach back and flip the white switch back to the upward position and turn the power back on. If this continues to happen, check to make sure only one interface is running

Every time you connect an experiment which was not previously plugged into the USB port, it is necessary to first run the experiment using the software provided from Armfield for the device, even if the computer had not been connected to another experiment. If this is not done, the LabVIEW interface will receive garbage readings and will be unable to control the experiment. We do not know why this is the case, but we suspect it has something to do with LabVIEW loading with old parameters that are left over from the previous experiment that was connected. The Armfield software is in two folders in the Start Menu: **Start→Programs→Armfield Heat Exchanger Software** and **Heat Transfer Software**. After you open it, all that needs done is to select **View→Diagram** and turn the Power On off and then on. Be sure to open the correct application for the experiment you are connecting. If even after you have opened the Armfield software you are still unable to use the interface, unplug the USB cable and plug it back in and repeat this process.

## 5.2 LabVIEW

After opening a VI in LabVIEW, you can edit the controls. If you want to change the default values of any of the controls (for instance the “file path” directory and the “PID Parameters-Custom/Automatic” control which are mentioned later), saving the VI will not be enough. You must click **Operate→Make Current Values Default**, and then Save the VI. There are several hidden panels containing controls that only the administrators should change. These hidden panels can only be accessed when no one is viewing the interface via the LabVIEW web server, because otherwise the remote clients would be able to see the administrator panels, which is a security risk. You can access a hidden panel by right clicking on a visible tab and selecting its name from the **Goto Page** menu. Currently, there are two types of interfaces, the two heat exchanger interfaces and the five heat conduction and radiation interfaces. The panels for both types of interfaces are included at the end of this document, in Appendix B.

### 5.2.1 Heat Exchanger Interfaces

The interface panels shown in Appendix B.1 are from the Shell and Tube Heat Exchanger interface, but the same panels are present on the Flat Plate Heat Exchanger interface as well.

#### 5.2.1.1 Controls Tab

This is the panel the students use to control the interface. The Power light and the Heater dial are not controls. They are only indicators. The setpoint is the temperature the device

will attempt to achieve on the hot water intake given the flow rates. The hot water and cold water flow rates can be adjusted using either the dials or the text box below each one and internally each of these values are treated as a setpoint. The further away from a setpoint the measured value is, the more drastic action the interface will take, either on the heater or on the flow controls. The countercurrent/cocurrent switch controls the direction the hot water flows. Cocurrent means the two streams will flow in parallel through the Heat Exchanger, and countercurrent means they will flow in opposite directions. The PID parameters that are displayed control the adjustments made to the heater power as feedback is received from the inlet hot water thermocouple. The PID parameters for the flow rates are fixed by an algorithm and are not shown on the front panel. The Record Data button will start the recording of data to the Data Filename specified, and will append new data to the end of the file if it already exists. Data will only be recorded after the button is pressed, and no previous data will be saved. The schematic device diagram has four indicators that report the temperatures recorded by each of the thermocouples. The same values are graphed in the chart below. The two red terminals T1 and T2 are the flow path for the hot water, and the blue terminals T3 and T4 are the flow path for the cold water. The flow rates of the hot and cold water are graphed in the lower right.

### 5.2.1.2 Data Table Tab

This tab shows the last several readings, 30 by default, generated from the thermocouples, regardless of whether the data is being recorded to a file or not. The columns marked N/A are data coming from ports that are not connected to thermocouples. There is no column for the elapsed time. The time period between samples can be set on the Hidden Controls tab in section 5.2.1.4.

### 5.2.1.3 Data Logging Tab

This is a hidden tab. The three sliders control the decimal precision of the results displayed in the Data Table, the Flow Rates of the hot and cold water, and the Temperature readings T1-T4. The write method used to write to the file is indicated in the grey box to be “append”, as opposed to overwriting existing files. The sampling time, measure in milliseconds, will be the time interval between samples taken from the thermocouples. The delimiter field is what will be inserted between every reading. The type of file we are writing to is Comma-Separated Values file, or CSV. Thus, we use a comma to delimit the values. The \s represents a space. The file path is the absolute path on the computer the file will be written. This can only take very simple paths, with no spaces, dots, or special characters. The directory for course 10.302 is **C:\inetpub\wwwroot\data\10302** and for the Cambridge experiments it is **C:\inetpub\wwwroot\data\CAMIIA**. For a new course, the folder must be created manually through Windows before LabVIEW can write to it. The heading of each column is given at the bottom, along with a log of the last value measured for each.

### 5.2.1.4 Hidden Controls Tab

This is a hidden tab. This is the administrator control panel to the Heat Exchanger. These will determine the initial screen which appears on the Controls tab when the interface is first launched. The PID Parameters switch is extremely important on the shell and tube interface. This determines whether the stored PID values on the PID Parameters tab will be used, or whether the students can change these values themselves. The low water level indicator says when the tank needs to be refilled.

### 5.2.1.5 PID Parameters Tab

\*\*\*Note: This section contains errors in its explanation of this panel. Do not attempt to make any changes to this panel based on the information contained here.

This is a hidden tab. These controls are all intended to set the response of the devices on the Heat Exchanger to errors measured by the interface. The heater cannot measure temperature directly, nor can the pumps measure the flow rate. So the interface has to adjust them based on the readings it receives from the sensors. The left side of the panel is all related to the heater. The Automatic PID Parameters are what will be used if the switch on the Hidden Controls tab is set to Automatic. The middle panel contains several controls which set the threshold values of when the readings are too high or too low, and what outputs are produced for each. The dt is the interval at which the error is checked. The rest are other minor parameters and options which can be set. The thermometers below indicate the current reading of T1, the sliding mean of the past number of loops of the temperature, and below them the resulting output of the calculation and the timestamp. The sliding bars at the bottom have been poorly labeled and their use is a mystery.

The hot and cold flow rates to the right operate in much the same way except they are not changeable by the students and are therefore much simpler on this interface. Below them are the setpoint which the device is to achieve and the Manual settings for when the PID Parameters button on the Hidden Controls is set to Controllable.

As described in Marc Knight's MEng thesis, P stands for proportional band, the gain that is applied to the error. P is dimensionless. I is the integral control that says how many seconds of previous measurements to include in the compensation response. I has units of seconds. D is the derivative control which can compensate for very sharp changes faster than the other two. D also has units of seconds. The error, e, is the difference of the measured value subtracted from the setpoint value.

The output of the PID control block will correspond to the equations:

$$e = X_{\text{setpoint}} - X_{\text{measured}}$$
$$u = P \cdot e + I \int e dt + D \frac{de}{dt}$$

Faye McNeill calculated values for the Heat Exchanger to be P=36.00, I=42.00, D=10.50. When University of Cambridge, England does their experiments on the shell and tube

heat exchanger, they will expect the values to be unstable so the students can carry out a control exercise. The parameters will not allow the temperature to converge to the setpoint and it will either oscillate or continue to heat up. Their assignment is to derive PID control parameters for the system.

If you need more detail about the PID parameters, or how to derive them, consult Marc Knight's 2003 MEng Thesis or the LabVIEW PID Control Toolkit Manual.

### 5.2.1.6 I/O Tab and Random Controls Tab

These are hidden tabs. They were most likely used for debugging the interface when it was initially programmed. The names of the controls do not give any intuition as to their purpose, and the controls do not seem to be useful in the current implementation, but it is probably best to leave them untouched.

## 5.2.2 Heat Transfer Interfaces

These devices are much less complex than the Heat Exchangers. And thus the interfaces are less complex as well. There are only four tabs compared to the seven for the Heat Exchangers, and there is no process control to worry about.

Again, pictures of the interface tabs can be found in Appendix B. The interface used is for the Extended Surface experiment, but the same tabs are also used on the Linear Conduction and Radial Conduction experiments.

### 5.2.2.1 Instructions Tab

This tab is simply a set of text labels explaining how to run the interface.

### 5.2.2.2 User Controls Tab

This tab is what the students use to control the experiment. The Heater On/Off switch will activate and deactivate the heater. The Heater Control is a scrolling value text box which takes values from 0 to 100%, in increments of 1%. As the percent goes up, so do the voltage and current flowing through the heater, which are reported in the Heater Voltage and Heater Current boxes. The temperatures at several points on the device are reported and graphed below. In this case T1-T9 are used, but some of the other experiments do not use that many. The Record Data button will start recording to the file specified in Data Filename as of the moment it is pressed, ignoring any prior data.


### 5.2.2.3 Data Table Tab

The Data Table reports all recorded measurements, including ones which are not written to the file. By default, it shows 30 such measurements at once, and scrolls as it receives new ones. Unlike the Heat Exchanger, this Data Table has a column for elapsed time.

### 5.2.2.4 Administrator Settings Tab

This tab is hidden. At the top you can control the time between samples (Sample interval) and the number of digits used to report the measurement (Data Recorder Decimal Places). The button on the middle left controls whether new data will be appended to the end of a file or if the file will be overwritten when the interface attempts to write to a file that already exists. The delimiter is what will be inserted between values. The \s means a space. The file path is the full path on the computer to where you want the file written. The Headings will be written whenever the interface creates a new file to write to, and the Data is the last set of measurements recorded.

## 5.3 Web Publishing

As discussed in Section 3.6, LabVIEW provides its own web server which is able to publish VIs on the web and respond to changes in real time. The web server's default directory which it will serve HTML files from is **C:\Program Files\National Instruments\LabVIEW 7.0\www**. Here you will find html files for the different interfaces. To access these files via the web, you must specify port 82 in your URL, like this: 

Each interface has its own HTML file, for example **flat\_plate\_interface.html** is for the Flat Plate Heat Exchanger. Each of the interface pages has a Java chat client included:

```
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
WIDTH = 600 HEIGHT = 250
codebase="http://java.sun.com/products/plugin/1.3/jinstall-13-
win32.cab#Version=1,3,0,0">
  <PARAM NAME = CODE VALUE = "ChatWindow.class" >
  <PARAM NAME="type" VALUE="application/x-java-applet;version=1.3">
  <PARAM NAME="scriptable" VALUE="false"><COMMENT>
  <EMBED type="application/x-java-applet;version=1.3" CODE =
  "ChatWindow.class" WIDTH =600 HEIGHT = 250 scriptable=false
  pluginspage="http://java.sun.com/products/plugin/1.3/plugin-
  install.html"> <NOEMBED></COMMENT> </NOEMBED>
  </EMBED>
</OBJECT>
```

This allows students to communicate directly from different terminals while performing the same experiment. If you are at the main server console, you can chat using the Administrator login.

Each page also contains the information for loading a VI: This is the code which loads the Flat Plate interface:

```
<OBJECT ID="LabVIEWControl" CLASSID="CLSID:A40B0AD4-B50E-4E58-8A1D-
8544233807AB" WIDTH=970 HEIGHT=775
```

```

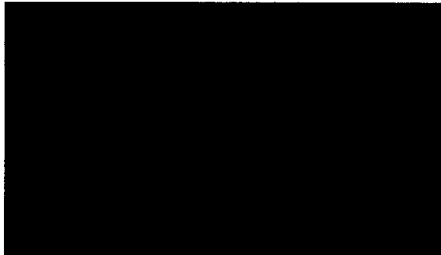
CODEBASE="ftp://ftp.ni.com/support/labview/runtime/windows/7.0/LVRunTime
Eng.exe">
<PARAM name="LVFPPVINAME" value="Flat Plate Heat Exchanger v1.4 Live
(LabVIEW 7.0).vi">
<EMBED SRC=".LV_FrontPanelProtocol.rpvi7" LVFPPVINAME="Flat Plate Heat
Exchanger v1.4 Live (LabVIEW 7.0).vi" TYPE="application/x-labviewrpvi7"
WIDTH=900 HEIGHT=760
PLUGINSOURCE="http://www.ni.com/webappdemos/lv"></EMBED>
</OBJECT>

```

The CODEBASE field indicates where the browser can download the plug-in to view the content if it is not currently installed. The PARAM shown indicates the name of the VI to be loaded. You do not need to specify the full path of the VI for these pages. When the LabVIEW server receives an object request like this, it will check if that particular interface is currently open in LabVIEW, and then display it.

At the bottom of the page is the contact information for the current web administrator, you.

## 6.0 Contact Information

Name	Office	E-mail (@mit.edu)	Phone (Extension)	Cell Phone
Clark K. Colton	66-452	ckcolton		
Barbara Driscoll	66-446	bhd		
Steve Wetzel	66-413	swetzel		
Rodney Graham	-	rgraham		
Michelle Hardiman	66-457	mth		
David Saylor	Japan	dsaylor@alum.mit.edu		
Sid Sen	Seattle	sidsen@microsoft.com		
Armfield Support	UK	support@armfield.co.uk		

# Appendix A - TestPage.aspx

## Appendix A.1 - Visual Studio .NET Code for TestPage.aspx

```
using System;
using System.Collections;
using System.ComponentModel;
using System.Configuration;
using System.Data;
using System.Data.SqlTypes;
using System.Data.SqlClient;
using System.Drawing;
using System.Web;
using System.Web.SessionState;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;

namespace HeatexWeb
{
    /// <summary>
    /// Summary description for TestPage.
    /// </summary>
    public class TestPage : HeatexBase
    {
        protected System.Web.UI.WebControls.Button submit;
        protected System.Web.UI.WebControls.Label error;
        protected System.Web.UI.WebControls.Label SQLstring;
        protected System.Web.UI.WebControls.RadioButtonList rblist;

        private void Page_Load(object sender, System.EventArgs e)
        {
            // Put user code to initialize the page here
        }

        #region Web Form Designer generated code
        override protected void OnInit(EventArgs e)
        {
            //
            // CODEGEN: This call is required by the ASP.NET Web
            Form Designer.
            //
            InitializeComponent();
            base.OnInit(e);
        }

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
    }
}
```



```

private void InitializeComponent()
{
    this.submit.Click += new
System.EventHandler(this.submit_Click);
    this.Load += new System.EventHandler(this.Page_Load);
}
#endregion

private void submit_Click(object sender, System.EventArgs e)
{
    SqlConnection con;
    string sql;
    SqlCommand cmd;

    con = new
SqlConnection(ConfigurationSettings.AppSettings["Heatex.ConnectionString
"]);
    con.Open();

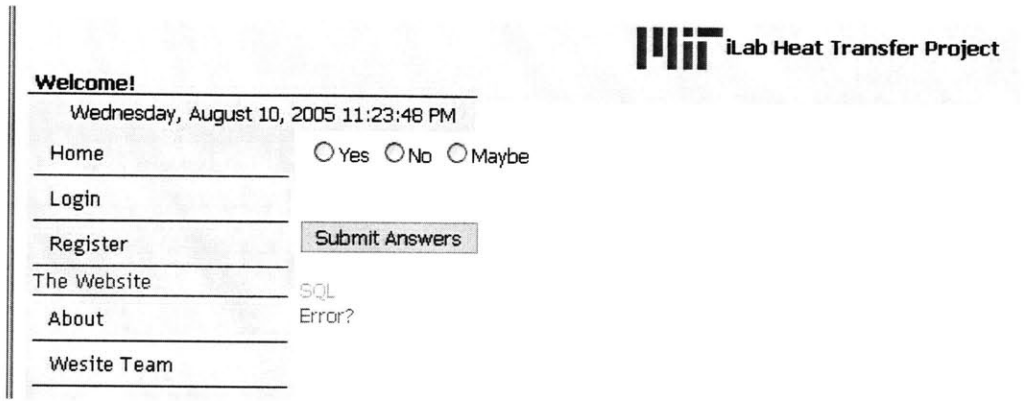
    try{
        sql = "INSERT INTO [Test] ([TimeStamp], Response)
VALUES (\'" +
            DateTime.Now.ToString() +
            "\',\'" +
            rblist.SelectedItem.Value.ToString() +
            "\')";
        SQLstring.Text = sql;
        cmd = new SqlCommand(sql, con);
        cmd.ExecuteNonQuery(); //Could fail
    }
    catch (Exception err) { error.Text =
err.ToString(); }
    con.Close();
}
}
}

```

## Appendix A.2 - HTML code for TestPage.aspx

```
<%@ Page language="c#" Codebehind="TestPage.aspx.cs"
AutoEventWireup="false" Inherits="HeatexWeb.TestPage" %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
  <HEAD>
    <title>Test Page</title>
    <LINK href="Style/iestyle.css" type="text/css"
rel="stylesheet">
    <meta content="Microsoft Visual&#9;Studio 7.0"
name="GENERATOR">
    <meta content="C#" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta
content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
  <body MS_POSITIONING="GridLayout">
    <form id="TestPage" method="post" runat="server">
      <asp:RadioButtonList ID="rblast"
RepeatDirection="Horizontal" Runat="server">
        <asp:ListItem Value="1">Yes</asp:ListItem>
        <asp:ListItem Value="2">No</asp:ListItem>
        <asp:ListItem Value="3">Maybe</asp:ListItem>
      </asp:RadioButtonList><br>
      <br>
      <asp:Button CssClass="Button" ID="submit" Text="Submit
Answers" Runat="server"></asp:Button>
      <br>
      <br>
      <asp:Label ID="SQLstring" ForeColor="#00ccff"
Runat="server">SQL</asp:Label><br>
      <asp:Label ID="error" ForeColor="#ff0066"
Runat="server">Error?</asp:Label>
    </form>
  </body>
</HTML>
```

## Appendix A.3 - Final Appearance of TestPage.aspx



The screenshot shows a web page with a navigation menu on the left and a 'Submit Answers' button on the right. The navigation menu includes links for Home, Login, Register, The Website, About, and Website Team. The 'Submit Answers' button is located next to the 'Register' link. The page also displays a 'Welcome!' message and the date and time: Wednesday, August 10, 2005 11:23:48 PM. The MIT iLab Heat Transfer Project logo is visible in the top right corner.

**MIT** iLab Heat Transfer Project

**Welcome!**  
Wednesday, August 10, 2005 11:23:48 PM

Home  Yes  No  Maybe

Login

Register

The Website SQL

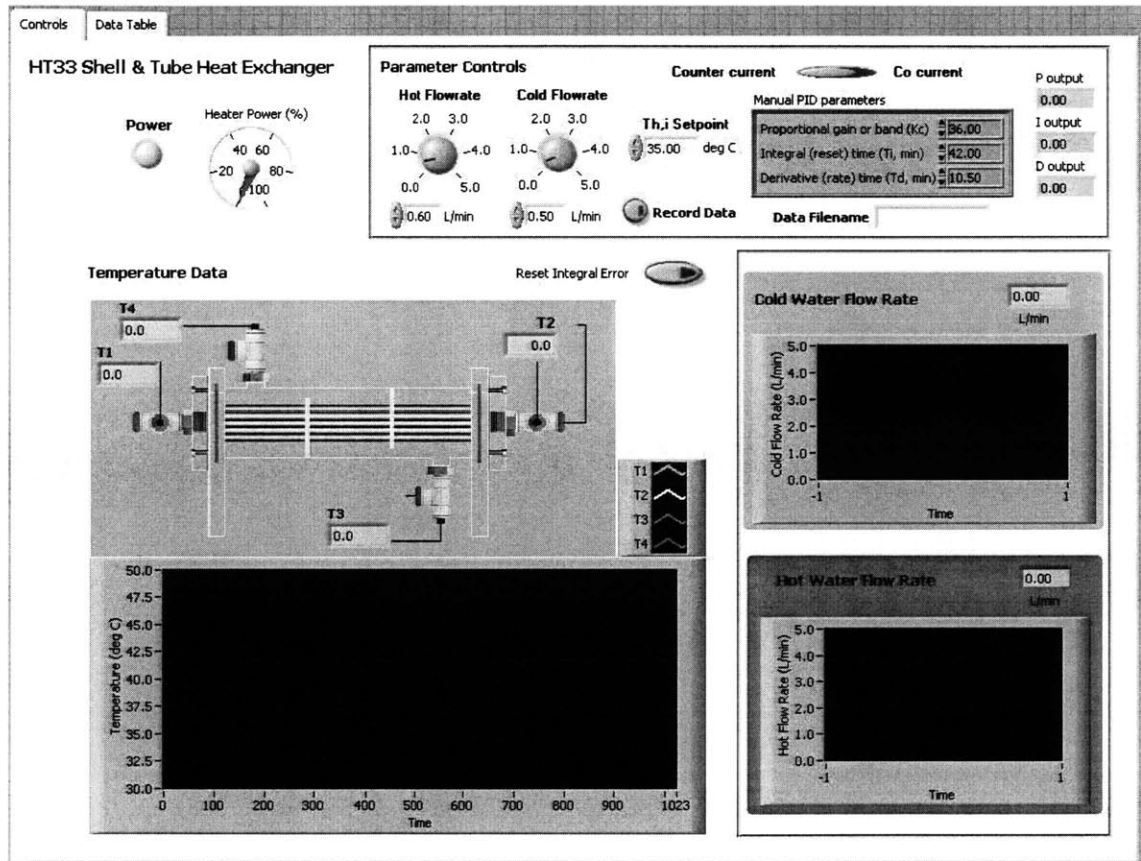
About Error?

Website Team

# Appendix B - LabVIEW Tabs

## Appendix B.1 - Heat Exchanger Tabs

### Controls Tab



## Data Table Tab

Controls
Data Table

Measurements to display

T1 (°C)	N/A	T2 (°C)	T3 (°C)	N/A (°C)	T4 (°C)	Qh (L/min)	Qc (L/min)	Th Setpoint (°C)	Heater Power (%)

## Data Logging Tab

Data Logging
Controls   Data Table

Data Recorder Decimal Places

Flow Rate D.P.

Temperature D.P.

Append to file

file path

Pause Between Data points

Delimiter

Headings 2

Headings

Log

Th (°C)	Thm (°C)	Tho (°C)	Ta (°C)	Tam (°C)	Tco (°C)	Qh (L/min)	Qc (L/min)	Th Setpoint (°C)	Heater Power (%)
Tho (°C)	Thm (°C)	Ths (°C)	Tai (°C)	Tcm (°C)	Tco (°C)	Qh (L/min)	Qc (L/min)	Th Setpoint (°C)	Heater Power (%)
16.61	18.56	12.12	13.68	0.71	0.11	35.00	100.00	16.61	16.61

69

# Hidden Controls Tab

Data Logging Controls Hidden Controls

POWER  Off  On

Watchdog Pulse  0

**INPUT CONSOLE**  
 Server  Client

Direction  
 Controllable  
 Co-current  Cold Flow PID Control  Heater Control

Set Point  
 Controllable  
 35.00 °C  Hot Flow PID Control  Manual  Heater Enabled (automatic)  Heater On (manual)

PID Parameters  
 Controllable  
 Automatic

Low Water level in hot water tank  Not active

Warning LED  Active

Pump drive  Active

Valve power [24V]  Active

**Client (Student) Settings**

Counter current  Hot Flowrate  Cold Flowrate  Data Recording 2

Thj Setpoint  deg C

Manual PID parameters 2

Proportional gain or band (Kc)	36.00
Integral (reset) time (Ti, sec)	42.00
Derivative (rate) time (Td, sec)	10.50

**Actual Settings In Use**

Flowrate - Hot fluid	Flowrate - Cold fluid	Temperature Setpoint - Hot fluid
3.00 L/min	40.00 L/min	0.00 deg C

Log to File  File name

# PID Parameters Tab

Data Logging Controls PID parameters

**Heater**

Automatic PID parameters

Proportional gain or band (Kc)	36.00
Integral (reset) time (Ti, sec)	42.00
Derivative (rate) time (Td, sec)	10.50

**options**

SP & PV range: sp low 20.00 sp high 60.00

Output range: out low 0.00 out high 100.00

dt (s) 1.00

Mean Temp  deg C

Temperature Setpoint  deg C

Manual PID parameters

Proportional gain or band (Kc)	36.00
Integral (reset) time (Ti, sec)	42.00
Derivative (rate) time (Td, sec)	10.50

**Cold Flow Rate PID Controller**

Proportional gain or band (Kc) 10.00

Integral (reset) time (Ti, sec) 0.04

Derivative (rate) time (Td, min) 0.00

SP & PV range: sp low 0.00 sp high 5.00

Output range: out low 0.00 out high 100.00

dt (s) 1.00

**Hot Flow Rate PID Controller**

Proportional gain or band (Kc) 20.00

Integral (reset) time (Ti, min) 0.0100

Derivative (rate) time (Td, min) 0.0000

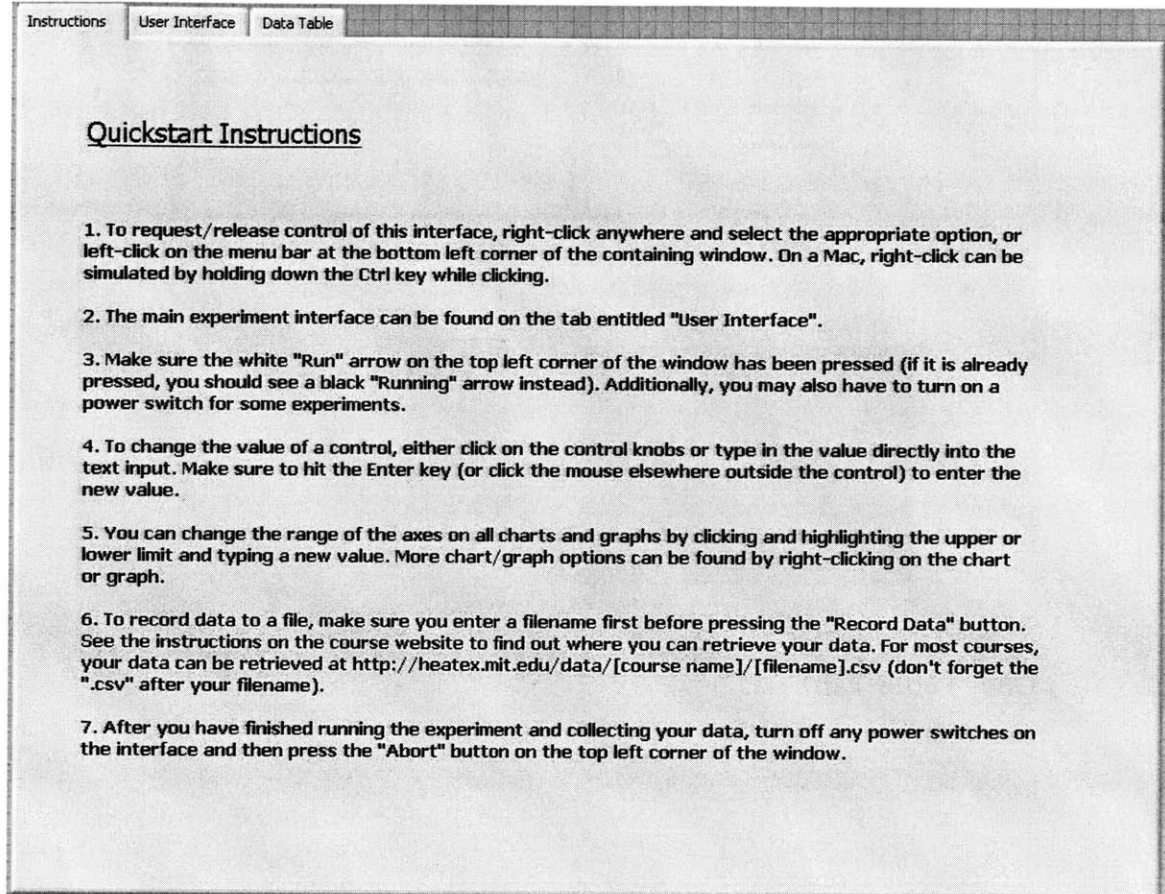
SP & PV range: sp low 0.00 sp high 5.00

Output range: out low 0.00 out high 100.00

dt (s) 1.00

# Appendix B.2 - Heat Transfer Tabs

## Instructions Tab



Instructions   User Interface   Data Table

### Quickstart Instructions

1. To request/release control of this interface, right-click anywhere and select the appropriate option, or left-click on the menu bar at the bottom left corner of the containing window. On a Mac, right-click can be simulated by holding down the Ctrl key while clicking.
2. The main experiment interface can be found on the tab entitled "User Interface".
3. Make sure the white "Run" arrow on the top left corner of the window has been pressed (if it is already pressed, you should see a black "Running" arrow instead). Additionally, you may also have to turn on a power switch for some experiments.
4. To change the value of a control, either click on the control knobs or type in the value directly into the text input. Make sure to hit the Enter key (or click the mouse elsewhere outside the control) to enter the new value.
5. You can change the range of the axes on all charts and graphs by clicking and highlighting the upper or lower limit and typing a new value. More chart/graph options can be found by right-clicking on the chart or graph.
6. To record data to a file, make sure you enter a filename first before pressing the "Record Data" button. See the instructions on the course website to find out where you can retrieve your data. For most courses, your data can be retrieved at [http://heatex.mit.edu/data/\[course name\]/\[filename\].csv](http://heatex.mit.edu/data/[course name]/[filename].csv) (don't forget the ".csv" after your filename).
7. After you have finished running the experiment and collecting your data, turn off any power switches on the interface and then press the "Abort" button on the top left corner of the window.





# Administrator Settings Tab

Instructions | User Interface | Data Table | Administrator Settings

Sample Interval (ms): 5000

Data Recorder Decimal Places: 0 1 2 3

append to file

Delimiter: ,s,ls

File path: %C:\inetpub\wwwroot\data\10302

Headings	Elapsed	T1 (°C)	T2 (°C)	T3 (°C)	T4 (°C)	T5 (°C)	T6 (°C)	T7 (°C)	T8 (°C)	T9 (°C)	Heater	Heater
Data	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

# References

[1] Armfield Limited. "Instruction Manual: HT14C—Computer Compatible Combined Convection and Radiation". Issue 2. March, 2005.

[2] Armfield Limited. "Instruction Manual: HT16-REM—Computer Controlled Radiation Errors in Temperature Measurement Accessory". Issue 1. July, 2004.