CONTROL TECHNOLOGY CORPORATION

Model 5200 Web Development Guide

Model 5200
Web Development Guide

Copyright 2004 © Control Technology Corporation
All Rights Reserved.

WARNING: Use of CTC Controllers and software is to be done only by experienced and qualified personnel who are responsible for the application and use of control equipment like the CTC controllers. These individuals must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and/or standards. The information in this document is given as a general guide and all examples are for illustrative purposes only and are not intended for use in the actual application of CTC product. CTC products are not designed, sold, or marketed for use in any particular application or installation; this responsibility resides solely with the user. CTC does not assume any responsibility or liability, intellectual or otherwise for the use of CTC products.

The information in this document is subject to change without notice. The software described in this document is provided under license agreement and may be used and copied only in accordance with the terms of the license agreement. The information, drawings, and illustrations contained herein are the property of Control Technology Corporation. No part of this manual may be reproduced or distributed by any means, electronic or mechanical, for any purpose other than the purchaser's personal use, without the express written consent of Control Technology Corporation.

The information in this document is current as of the following Hardware and Firmware revision levels. Some features may not be supported in earlier revisions. See www.ctc-control.com for the availability of firmware updates or contact CTC Technical Support.

Model Number	Hardware Revision	Firmware Revision
5200	All Revisions	> 5.00.30

TABLE OF CONTENTS

Blank

CHAPTER 1

Introduction

This document provides information on the functionality and structure of the 5200 internal web server. This information will be important to those attempting to develop custom web applications.

The web server is a means by which to provide files, stored on the 5200 internal disk, to a browser, via a TCP HTTP connection on port 80. Any type of file can be served up as it is simply read from the disk and transmitted to the requestor. Advanced functions, such as CGI scripting, are not supported. The content of files typically consist of pages with HTML text, jpegs, gif, or png images, css style sheets, xml, etc… The web server simple reads the file and sends it, it does not process the data.

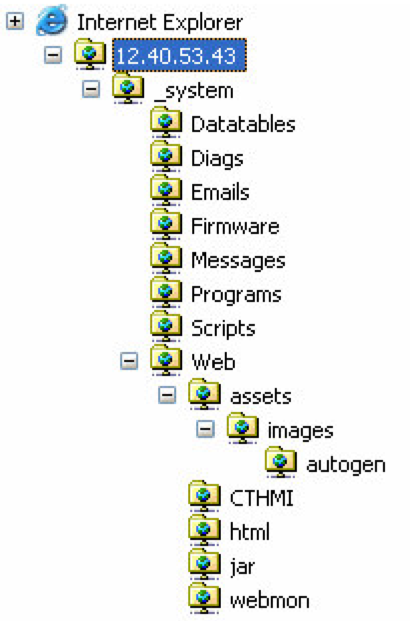
Those wishing to do advanced features, such as that implemented by WebMON, may use Java applets. An applet is typically embedded within a web page to provide added functionality. When using Java the applet may present information as part of the page, or in the case of WebMON, as a separate framed window. This document introduces a program called “CTCSample” which provides sample code to read and write a single register and/or monitor numerous registers (scan), using Java. The 5200 web server, combined with the Scripting Language, provides an extremely powerful data collection and presentation environment. Reference the WebMON manual for some examples of the power of Java when combined with the 5200.

Blank

CHAPTER 2

Web Server File System

The 5200 internal web server is resident on the main flash disk and follows a strict directory structure. The root directory for HTTP access is always /_system/Web. This is where the default 'index.html' file must reside when the IP Address of the controller is entered within a browser, and no path is given. If a path is supplied, user defined directories may be referenced. The default file folder structure for a factory installed WebMON environment is as follows:



The only directory that can be accessed by the web server is '/_system/Web' which is known as root ('/') to the browser. Other directories can be referenced via virtual directories but that is beyond the scope of this document (*see Document #951-520001, Remote Administration Guide*).

With regards to WebMON, its 'index.html' file is within the /_system/Web/webmon directory and the /_system/Web directory. This means the application can be invoked by either “Controller IP Address” or “Controller IP Address”/webmon, within a browser. Directly referencing an ‘html’ file is also possible, as it is with most web servers.

Two other directories, ‘assets’ and ‘html’, were simply created by a web publishing package and contain normal html pages, style sheet information, and graphical images. You do not have to follow this structure just note that all file and folder names are case sensitive, similar to UNIX.

There are two reserved directories, ‘CTHMI’ and ‘jar’. The CTHMI directory must be present for use with CTHMI since this is where projects are published. The ‘jar’ directory is where all CTC jar files presently reside. In a custom environment, neither of these directories are required to exist, although, if WebMON functionality is to be maintained, the current directory structure must be preserved.

CHAPTER

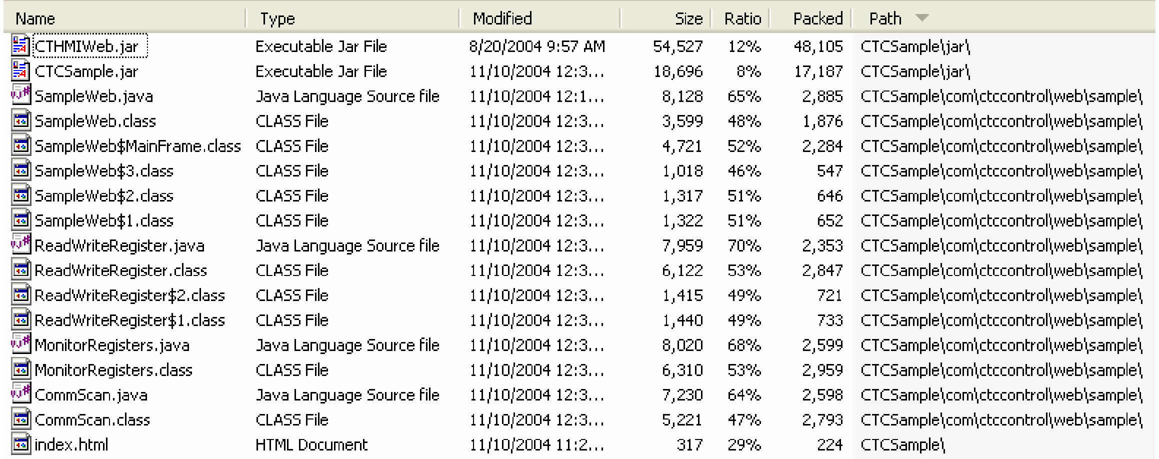
3

Sample Java Applet

A sample Java applet environment is available via download from the Control Technology Corporation web site (www.ctc-control.com). This applet is a very simple Java implementation, presenting methods to simply read/write registers and periodically scan a list. With the provided software additional functionality may be added. The full power of the *Script Language Guide, Document #951-520003*, is available.

Applet Installation

The applet is distributed as a zip file, “CTCSample.zip”. Both the source and run-time class files are available.



Name	Type	Modified	Size	Ratio	Packed	Path
CTHMIWeb.jar	Executable Jar File	8/20/2004 9:57 AM	54,527	12%	48,105	CTCSample\jar\
CTCSample.jar	Executable Jar File	11/10/2004 12:3...	18,696	8%	17,187	CTCSample\jar\
SampleWeb.java	Java Language Source file	11/10/2004 12:1...	8,128	65%	2,885	CTCSample\com\ctccontrol\web\sample\
SampleWeb.class	CLASS File	11/10/2004 12:3...	3,599	48%	1,876	CTCSample\com\ctccontrol\web\sample\
SampleWeb$MainFrame.class	CLASS File	11/10/2004 12:3...	4,721	52%	2,284	CTCSample\com\ctccontrol\web\sample\
SampleWeb$3.class	CLASS File	11/10/2004 12:3...	1,018	46%	547	CTCSample\com\ctccontrol\web\sample\
SampleWeb$2.class	CLASS File	11/10/2004 12:3...	1,317	51%	646	CTCSample\com\ctccontrol\web\sample\
SampleWeb$1.class	CLASS File	11/10/2004 12:3...	1,322	51%	652	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister.java	Java Language Source file	11/10/2004 12:3...	7,959	70%	2,353	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister.class	CLASS File	11/10/2004 12:3...	6,122	53%	2,847	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister$2.class	CLASS File	11/10/2004 12:3...	1,415	49%	721	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister$1.class	CLASS File	11/10/2004 12:3...	1,440	49%	733	CTCSample\com\ctccontrol\web\sample\
MonitorRegisters.java	Java Language Source file	11/10/2004 12:3...	8,020	68%	2,599	CTCSample\com\ctccontrol\web\sample\
MonitorRegisters.class	CLASS File	11/10/2004 12:3...	6,310	53%	2,959	CTCSample\com\ctccontrol\web\sample\
CommScan.java	Java Language Source file	11/10/2004 12:3...	7,230	64%	2,598	CTCSample\com\ctccontrol\web\sample\
CommScan.class	CLASS File	11/10/2004 12:3...	5,221	47%	2,793	CTCSample\com\ctccontrol\web\sample\
index.html	HTML Document	11/10/2004 11:2...	317	29%	224	CTCSample\

You may extract this file to a disk area of your choice. The files in the CTCSample\jar directory should be placed in the 5200 /_system/Web/jar sub-directory. In a standard installation CTHMIWeb.jar will already exist. In this instance the one included with the zip file will not be required. CTCSample.jar is the file that was created from the source code. Compiling/Editing was done using Eclipse, www.eclipse.org. With CTCSample.jar loaded into the 5200 jar subdirectory, the ‘index.html’ file will also need to be loaded via ftp. This file can either be placed in a sub-directory of your choosing, such as /_system/Web/CTCSample or the web server root directory if WebMON functionality is not required as the default page that appears when the IP address of the controller is entered within a web browser.

All source code and class files are contained within the package com.ctccontrol.web.sample, referencing the JApplet Swing class, SampleWeb. The provided "index.html" file, used to invoke the sample program, is extremely simple, opening a blank page:

```
<html><head><title>Sample CTC Applet</title></head>
 <BODY>
<applet
  name="_5200WebDevelopment"
  code="com.ctccontrol.web.sample.SampleWeb"
  codebase="/jar/"
  archive="CTCSample.jar, CTHMIWeb.jar"
  alt="Your browser cannot run the CTC SampleWeb Applet!"
  height=1 width=1
>
</applet>
</body></html>
```

As with, WebMON, the page containing the applet is standard html. Links to other web sites, pdf files, jpegs, gifs, etc., may also exist.

In summary recommendations are as follows:

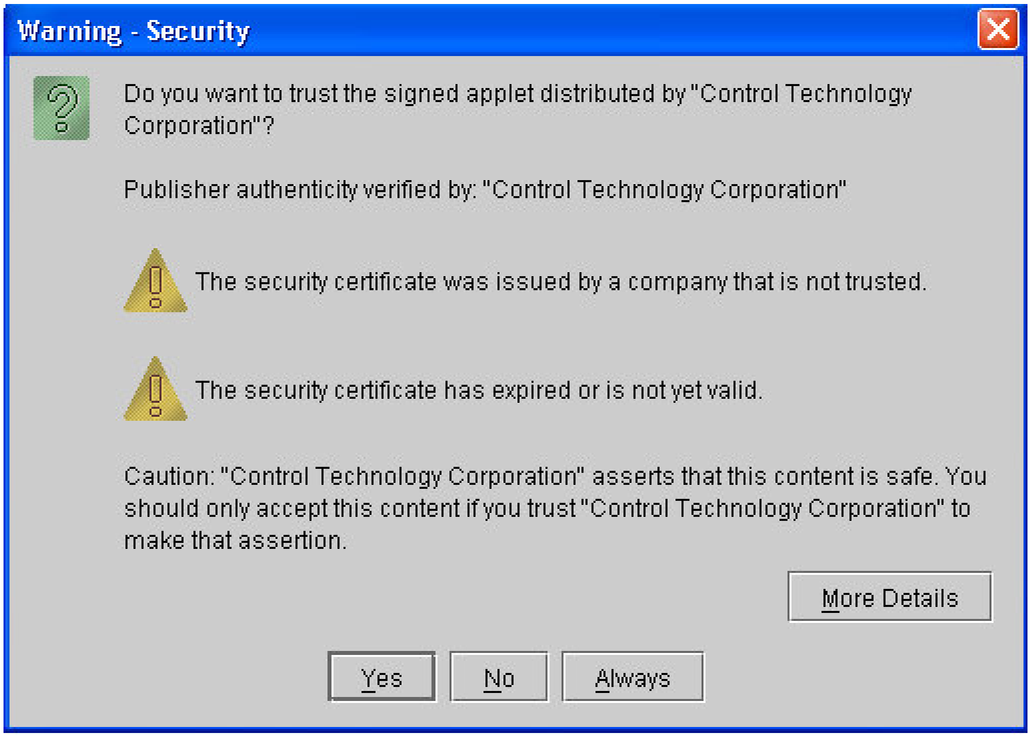
index.html -> /_system/Web/CTCSample
CTCSample.jar -> /_system/Web/jar
CTHMIWeb.jar -> /_system/Web/jar   (optional as should already exist)

Running Applet

To invoke:

"Controller IP Address"/CTCSample
Example: 12.40.53.43/CTCSample

Since the CTHMIWeb.jar file is a signed applet the CTC notification will appear. The user may select "Always" so that this box never appears again:



Do you want to trust the signed applet distributed by "Control Technology Corporation"?

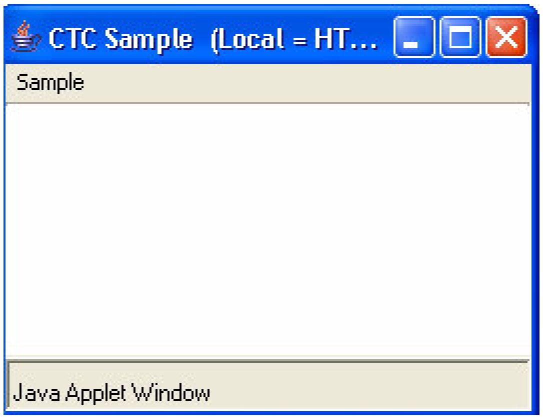
Publisher authenticity verified by: "Control Technology Corporation"

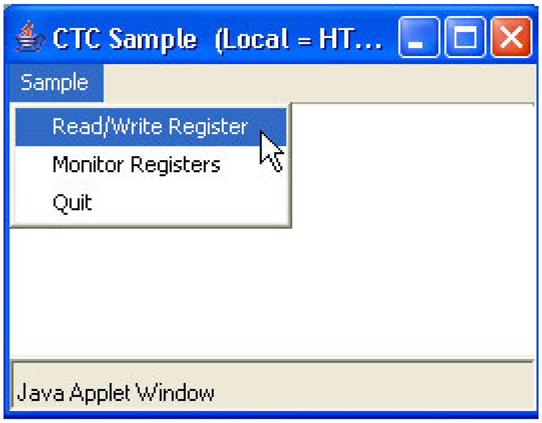
The security certificate was issued by a company that is not trusted.

The security certificate has expired or is not yet valid.

Caution: "Control Technology Corporation" asserts that this content is safe. You should only accept this content if you trust "Control Technology Corporation" to make that assertion.

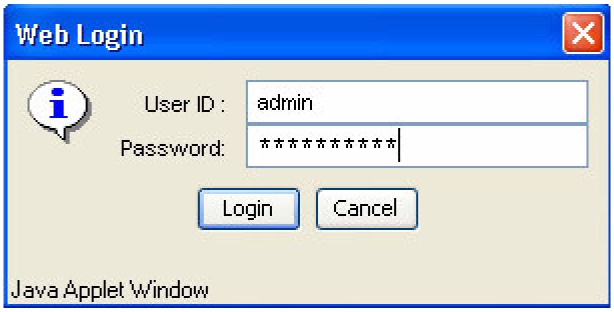
The CTCSample.jar file is provided, unsigned. Once the Security notice is responded to a JFrame window will appear:





Two menu options are available under ‘Sample’, ‘Read/Write Register’ and ‘Monitor Registers’.

Both establish a telnet session with the controller and issue commands as defined in the 5200 Scripting Language. Selection of either will first invoke a UserName/Password dialog box:

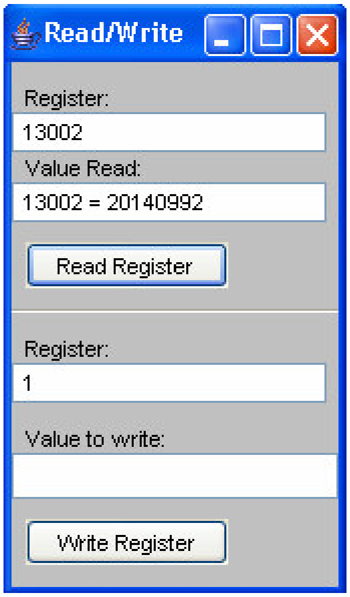


User ID : admin

Password: **********

Once entered any additional connections to the controller will invoke the same dialog although both the User ID and Password will be filled out, using the previous entry as a default. Each menu selection opens and independent window and separate TCP session with the controller.

'Read/Write' Register selection, click button for each read/write request:



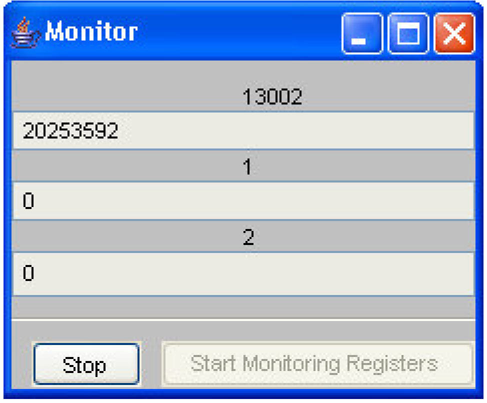
Register: 13002

Value Read: 13002 = 20140992

Register: 1

Value to write:

The 'Monitor Registers' selection will refresh every 500 milliseconds reading registers 1, 2 and 13002. It is left to the programmer for more advanced implementations:

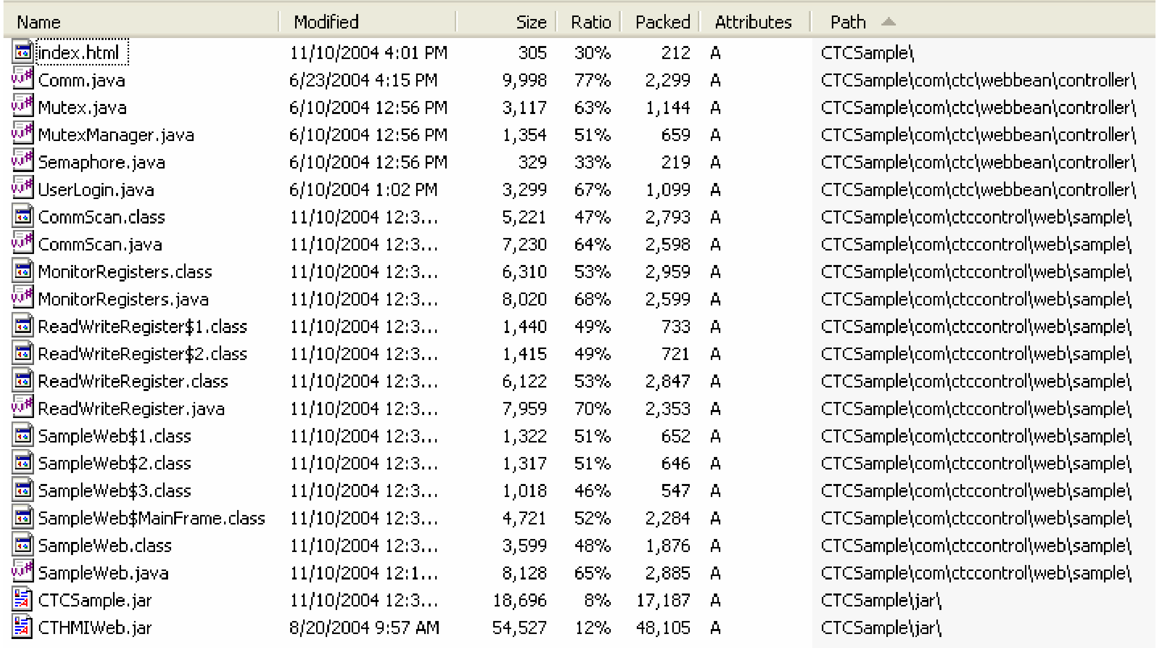


13002
20253592
1
0
2
0

Source Code

Source code for the SampleWeb applet is contained in CTCSample\com\ctccontrol\web\sample, for the main application. Additionally the required communication files which are referenced within CTHMIWeb.jar have been included within CTCSample\com\ctc\webbean\controller. The communication files are for reference only or building your own jar file. Modification of these files is not recommended without extensive testing but are provided to give the programmer a better understanding of the underlying architecture and also allow you to eliminate our signed jar file, CTHMIWeb.jar. Should you wish to sign your own jar reference Sun's web site for further details: http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/jarsigner.html.

CTCSample.zip



Name	Modified	Size	Ratio	Packed	Attributes	Path
index.html	11/10/2004 4:01 PM	305	30%	212	A	CTCSample\
Comm.java	6/23/2004 4:15 PM	9,998	77%	2,299	A	CTCSample\com\ctc\webbean\controller\
Mutex.java	6/10/2004 12:56 PM	3,117	63%	1,144	A	CTCSample\com\ctc\webbean\controller\
MutexManager.java	6/10/2004 12:56 PM	1,354	51%	659	A	CTCSample\com\ctc\webbean\controller\
Semaphore.java	6/10/2004 12:56 PM	329	33%	219	A	CTCSample\com\ctc\webbean\controller\
UserLogin.java	6/10/2004 1:02 PM	3,299	67%	1,099	A	CTCSample\com\ctc\webbean\controller\
CommScan.class	11/10/2004 12:3...	5,221	47%	2,793	A	CTCSample\com\ctccontrol\web\sample\
CommScan.java	11/10/2004 12:3...	7,230	64%	2,598	A	CTCSample\com\ctccontrol\web\sample\
MonitorRegisters.class	11/10/2004 12:3...	6,310	53%	2,959	A	CTCSample\com\ctccontrol\web\sample\
MonitorRegisters.java	11/10/2004 12:3...	8,020	68%	2,599	A	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister$1.class	11/10/2004 12:3...	1,440	49%	733	A	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister$2.class	11/10/2004 12:3...	1,415	49%	721	A	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister.class	11/10/2004 12:3...	6,122	53%	2,847	A	CTCSample\com\ctccontrol\web\sample\
ReadWriteRegister.java	11/10/2004 12:3...	7,959	70%	2,353	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb$1.class	11/10/2004 12:3...	1,322	51%	652	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb$2.class	11/10/2004 12:3...	1,317	51%	646	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb$3.class	11/10/2004 12:3...	1,018	46%	547	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb$MainFrame.class	11/10/2004 12:3...	4,721	52%	2,284	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb.class	11/10/2004 12:3...	3,599	48%	1,876	A	CTCSample\com\ctccontrol\web\sample\
SampleWeb.java	11/10/2004 12:1...	8,128	65%	2,885	A	CTCSample\com\ctccontrol\web\sample\
CTCSample.jar	11/10/2004 12:3...	18,696	8%	17,187	A	CTCSample\jar\
CTHMIWeb.jar	8/20/2004 9:57 AM	54,527	12%	48,105	A	CTCSample\jar\

Package com.ctcontrol.sample:

SampleWeb.java – JApplet sample program.
ReadWriteRegister.java – JInternalFrame creating a JPanel for reading/writing a register.
MonitorRegisters.java – JinternalFrame creating a JPanel that using CommScan to periodically scan for registers 1, 2, and 13002 for update in JTextFields.
CommScan.java – Communications thread to poll controller, updating JTextFields in MonitorRegisters.

Package com.ctc.webbean.controller:

Comm.java – Controls main connection to controller, logon handshake and read/write TCP packets.
Mutex.java - Middle level thread locking implementation.
MutexManager.java – Mutx management.
Semaphore.java – Low level thread locking implementation.
UserLogin.java – User dialog logon control.