

# Automatic Class Scheduler

## Administrator's Guide

### Table of Contents

● Introduction	1
● What is this document?	1
● What is this document not?	2
● For whom his this document?	2
● Prerequisites	2
● Installation	2
● Configuration	3
● Server Control / Maintenance	4
● Warranty / Disclaimer	4
● Credits / Responsibilities	5

### Introduction

The course "Software Design" is co-listed at the State University of New York at Oswego as a course in the Computer Science Bachelor program, CSC480, as well as a course in the Human Computer Interaction Master program, HCI521.

Throughout the class, students are required to design, plan and implement a software framework in a tight time frame with conditions that are close to real-world development situations. In the spring semester of 2007, this class project was a scheduling software for academic courses of SUNY Oswego. The goal was to develop a framework that would allow any student of SUNY Oswego to search for classes and get automatically generated schedules on the basis of personal constraints.

### What is this document?

This document is the administrator's guide to install, configure, and maintain the Automatic Scheduler. In a step-by-step process, it allows the user to set up the whole software framework from scratch. Also, this document contains important information on who should read this document and who shouldn't and tries to give background information on fixing installation problems. Lastly, this document is intended to keep a short list of every member of the development team and states their responsibilities. Although the Automatic Scheduler is designed to run on all platforms, this document assumes that you are running a Unix-based operating system and will only rarely point out the MS Windows equivalent, if at all.

## What is this document not?

This document will not cover information on how to debug Java code. Also, this is not a tutorial on Apache Tomcat, but merely a quick reference on how to set up pre-configured servers to run your custom scheduler. Also, introductions on how to administer Unix and Windows are not covered. This document is lastly also not intended to explain how to deploy the Automatic Scheduler to a Tomcat server other than the pre-configured one. If you seek information on those issues, please feel strongly encouraged to google. This document will also not cover information on how to apply this software to another campus. Most of the program code has been written merely for SUNY Oswego so explaining what to change goes beyond the scope of this document. If you want to use this software on a campus other than SUNY Oswego, please contact the project administrators, the project supervisor or any of the development team members.

## For whom is this document?

This document assumes that you are running a Unix-based operating system like Linux, Solaris or alike. Furthermore, you should be fairly familiar with basic administrative shell commands on Unix as well as very familiar with file manipulation. **You do not need a root account** to the machine you want to run your Automatic Scheduler on, in fact, you are encouraged **not** to run this software as **root**. Also, you might need knowledge about how to change a few simple lines of code in a Java source file and you should be able to recompile Java code.

## Prerequisites

The machine you want to deploy your Automatic Scheduler to should have a Java runtime environment installed, at least version 1.4.2 or higher. Also, you should have downloaded the Code Pack for the Automatic Scheduler, including all resource files, as well as the pre-configured Tomcat server. You should be able to obtain these files from the same source you received this document from.

If you choose to run the Automatic Scheduler using your own custom Tomcat or some other code processing http server, make sure it supports Java 1.4.2 or higher. In case of Tomcat, this would be version 5.

## Installation

Installing the Automatic Scheduler is fairly easy. You just have to copy all files into the correct locations. See below for details.

1. unzip the contents of the pre-configured Tomcat server pack to a directory of your choice to your hard disk
2. unzip the contents of the Automatic Scheduler Code Pack. You will find four directories in the archive: frontend/, backend/, lib/ and res/
  - a) copy the contents in frontend/ into webapps/ROOT/ in your tomcat directory
  - b) copy the contents in backend/ into webapps/ROOT/WEB-INF/classes/ in your tomcat directory. Make sure to preserve the full directory tree.
  - c) copy the contents in lib/ into webapps/ROOT/WEB-INF/lib/ in your tomcat directory
  - d) copy the contents in res/ anywhere you like. It is recommended to copy them

into the db folder in the directory tree of the backend classes. In any case, make sure to remember the location of these files as you need to change some code in the Java files later with regard to this location.

Remember that jess.jar is only a 21-day trial version of the expert system Jess. Make sure to contact Sandia for a suitable license.

## Configuration

The configuration process is rather simple, too, although a little more challenging than the installation, because the code base needs to be re-compiled. There are two steps necessary: configure the code base and configure Tomcat. Just follow the steps below.

### Code re-compilation

In order to deploy the Automatic Scheduler to another server or another location on the same server, it is necessary to change the path to certain resource files in the Java code. This is easily done by simply changing one variable in a few files. Below is a list of all files and the respective variables that need to be changed.

**Please note that in all cases, the path must be changed to the full path name from root to the resource files.** This means, if your Tomcat is running in /usr/httpd/ and you placed the source files in the WEB-INF/ directory of the Tomcat webroot folder, this would be the path to the resource files: /usr/httpd/webapps/ROOT/WEB-INF/ .

All files in the below table are in the package edu.oswego.cs.csc480AutomaticScheduler

<i>File name</i>	<i>Variable name</i>
engine.ObjectLoader.java	prefix
db.GenEd.GenEds.java	fileLocation
db.UniversalMaps.java	fileLocation
db.HTMLParserReturn.java	fileLocation

After you have changed the path in the variables in the above files, you need to re-compile the code. Make sure to point to the correct classpath. The classpath must point to the directory where you copied the \*.jar files from the lib/ directory from the Code Pack.

### Tomcat configuration

1. navigate into the directory conf/ in your tomcat directory
2. locate and open the file server.xml
3. locate the tag "Connector" and change the port from 8080 to something else. This part is optional – tomcat can run on any port. Make sure the port number is larger than 1024 so that you don't need root access to run the server.
4. If you plan on running multiple Tomcats, give every Tomcat an own port number
5. If you have multiple Tomcats running, also change the port in the tag "Server" to something else for each tomcat.

## **Server Control / Maintenance**

As mentioned above, you do not need root access to the machine you are running the Automatic Scheduler on. Simply follow the below instructions to start and stop the server.

1. navigate into the directory bin/ of your Tomcat installation.
2. To start the server, type `./startup.sh` . On Windows, simply double-click the `startup.bat` file.

The server can then be accessed through your browser by typing in the machines IP address or DNS name, followed by a colon and the port you have configured. Example:

<http://moxie.cs.oswego.edu:2684/>

3. To stop the server, type `./shutdown.sh` . On Windows, simply double-click the `shutdown.bat` file. Note, that this will close all running tomcats that have the same port configured in the "Server"-tag in the config file.

The server will run under the user name under which you have started it. At all times, you can view the log files (which are owned by the user that started the server) in the logs/ directory. The files beginning with "catalina" (the internal reference name of your Tomcat) contain the output that the Automatic Scheduler has written to stdout (i.e. the console) and the files beginning with "localhost" contain runtime information from Tomcat – i.e. what Tomcat itself has written to stdout.

If something goes wrong with the Automatic Scheduler or unexpected behaviour of either the code or Tomcat occurs, you are encouraged to refer to the log files for error fixing.

## **Warranty / Disclaimer**

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

© 2007, CSC480 / HCI 521 class of spring 2007.

Tomcat is a product of the Apache Software Foundation. © 1999-2007 All rights reserved.

## **Credits / Responsibilities**

Here is a list of all people that worked on this project and their responsibilities. This listing is in no particular order.

project supervisor:	Dr. Douglas Lea
project administrators:	Denis Barrett, Bastian Tenbergen, Alex Lim
Database Team:	Denis Barrett, Ryan Mann
Gen Ed Research:	Sem Phillippe, Kathi Dutton
Engine Framework:	Bastian Tenbergen, Chris Nagle
GUI / GWT:	Rick Buck, Joe Samuels, Dan Christensen, John Vint
User Requirements / HCI evaluation:	Ben Hance, Patrick Aguilar
Team Coordination / SVN supervision:	Rick Buck, Patrick Aguilar
Code Testing:	Tom Capetta, Alex Lim, Bob Free
Documentation:	Ben Hance, Bastian Tenbergen