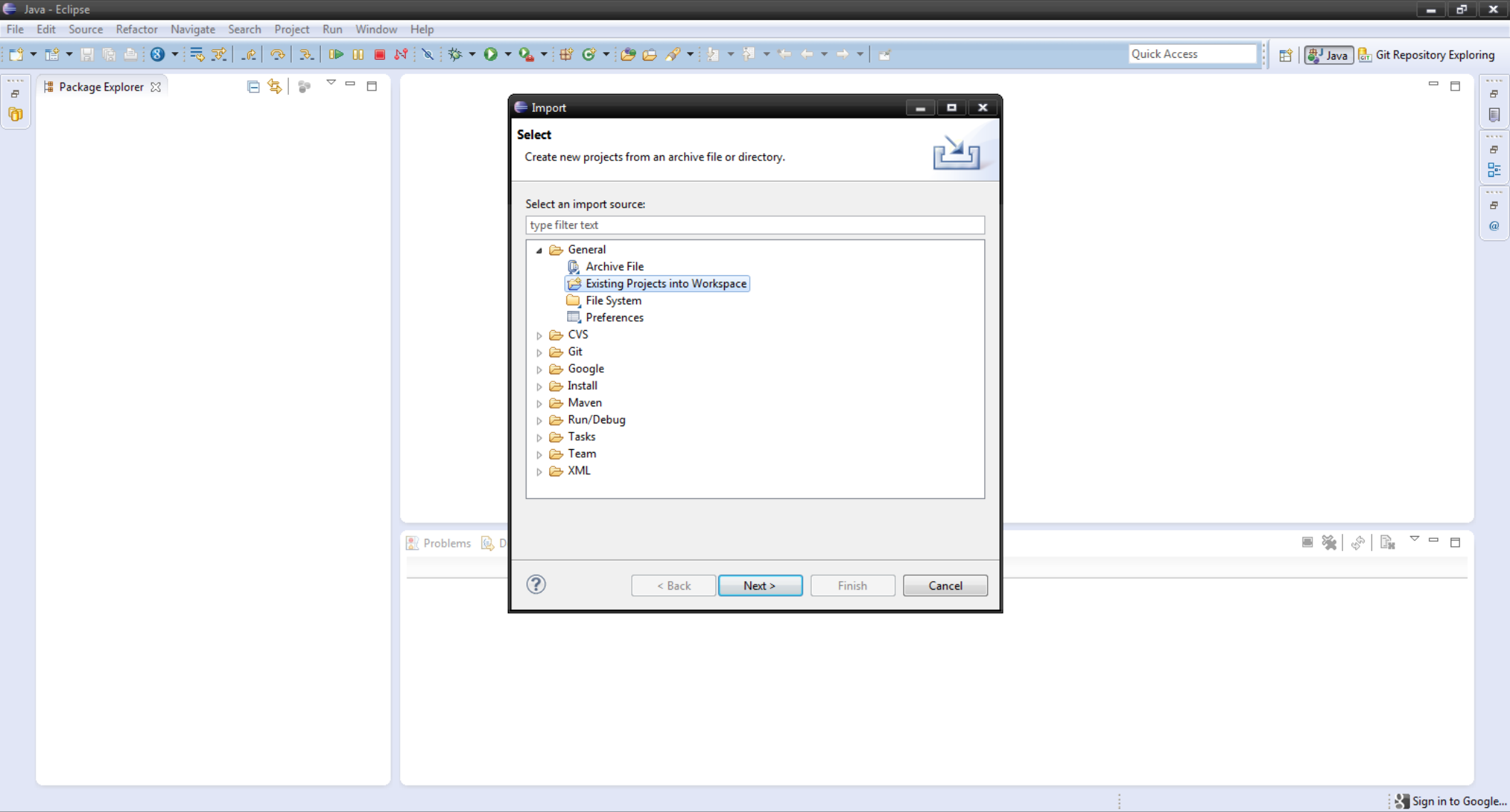
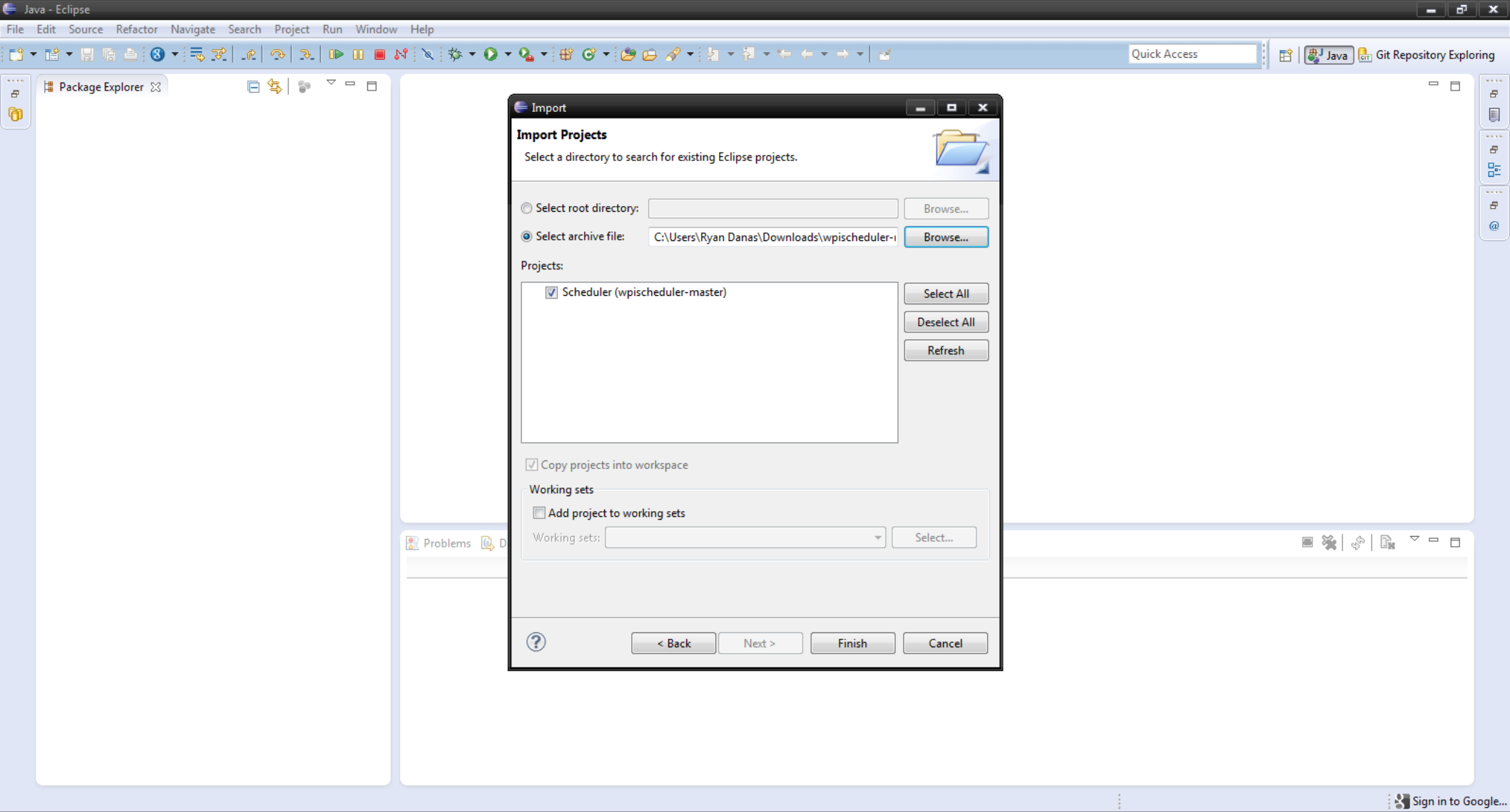


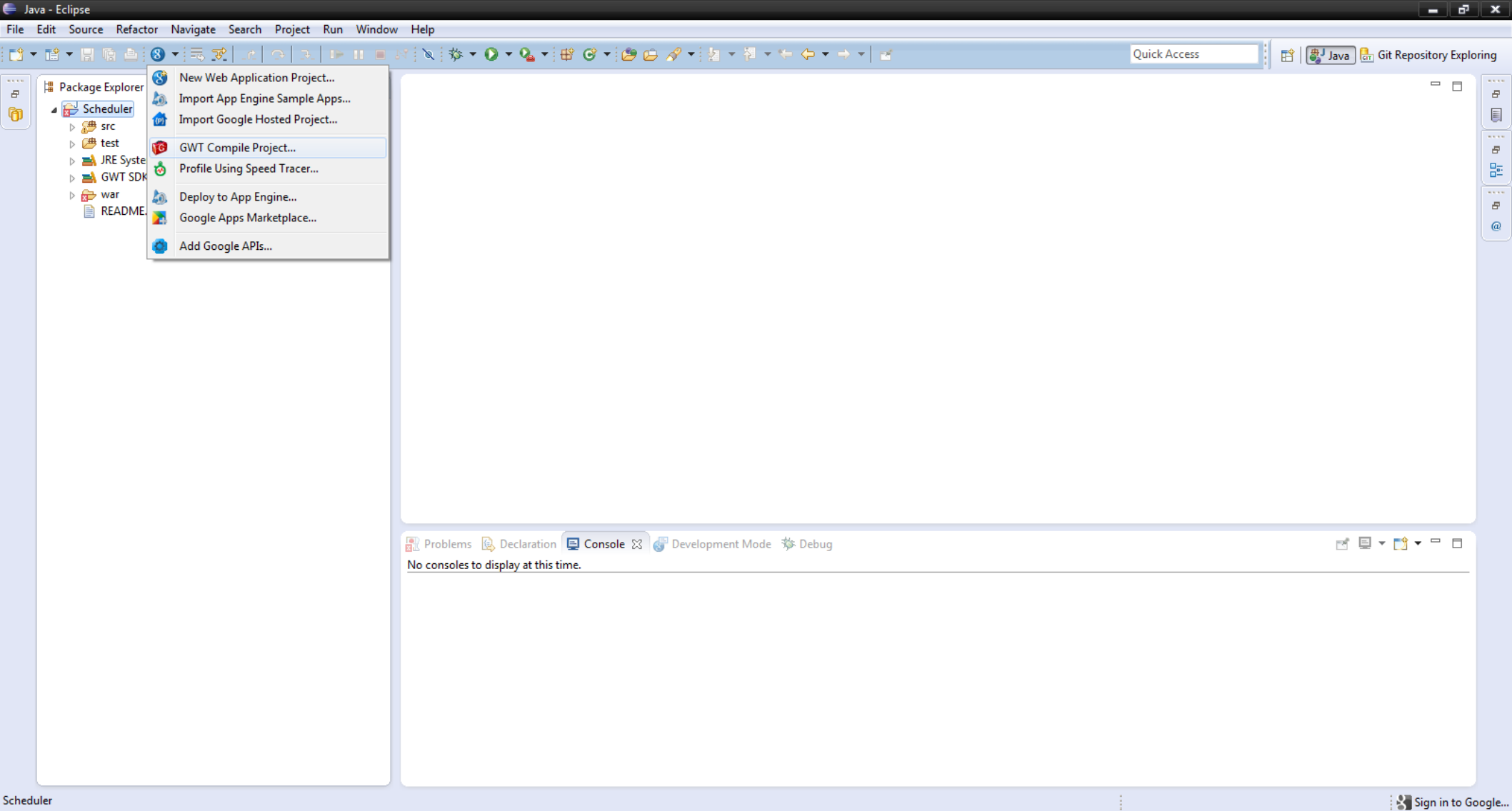
This Deployment Guide is intended to give explicit instructions on how to host WPI Scheduler on your own web server. In order to do so, you will need the Eclipse IDE (<http://www.eclipse.org/>), Google Web Toolkit plug-in for Eclipse (<http://www.gwtproject.org/usingeclipse.html>), and a copy of this project's source code (either from GitHub or attached from this report). Once you have Eclipse and the GWT plug-in installed, begin importing the source code as a new eclipse project.



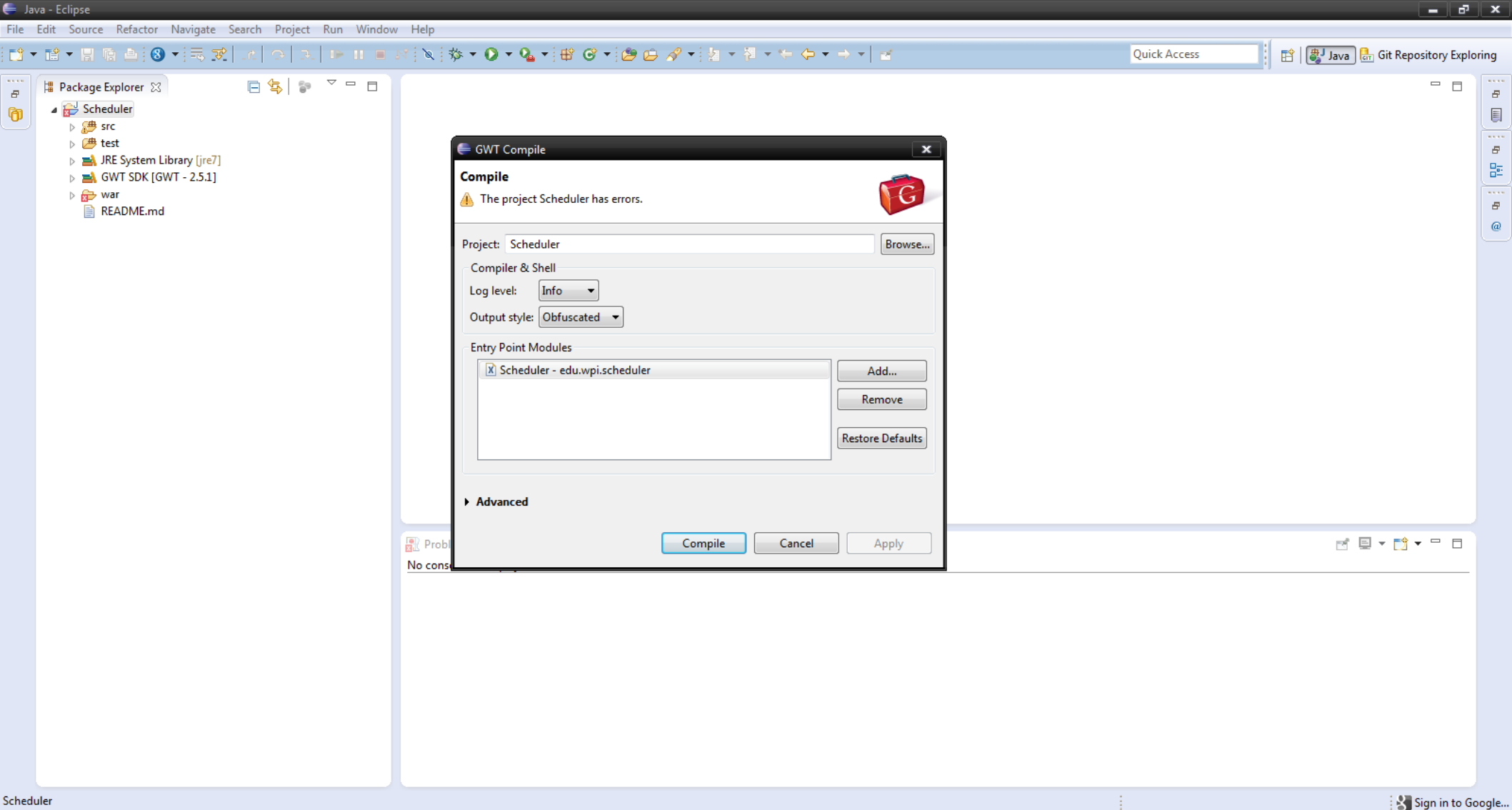
After going to the import menu, select import existing project into workspace.



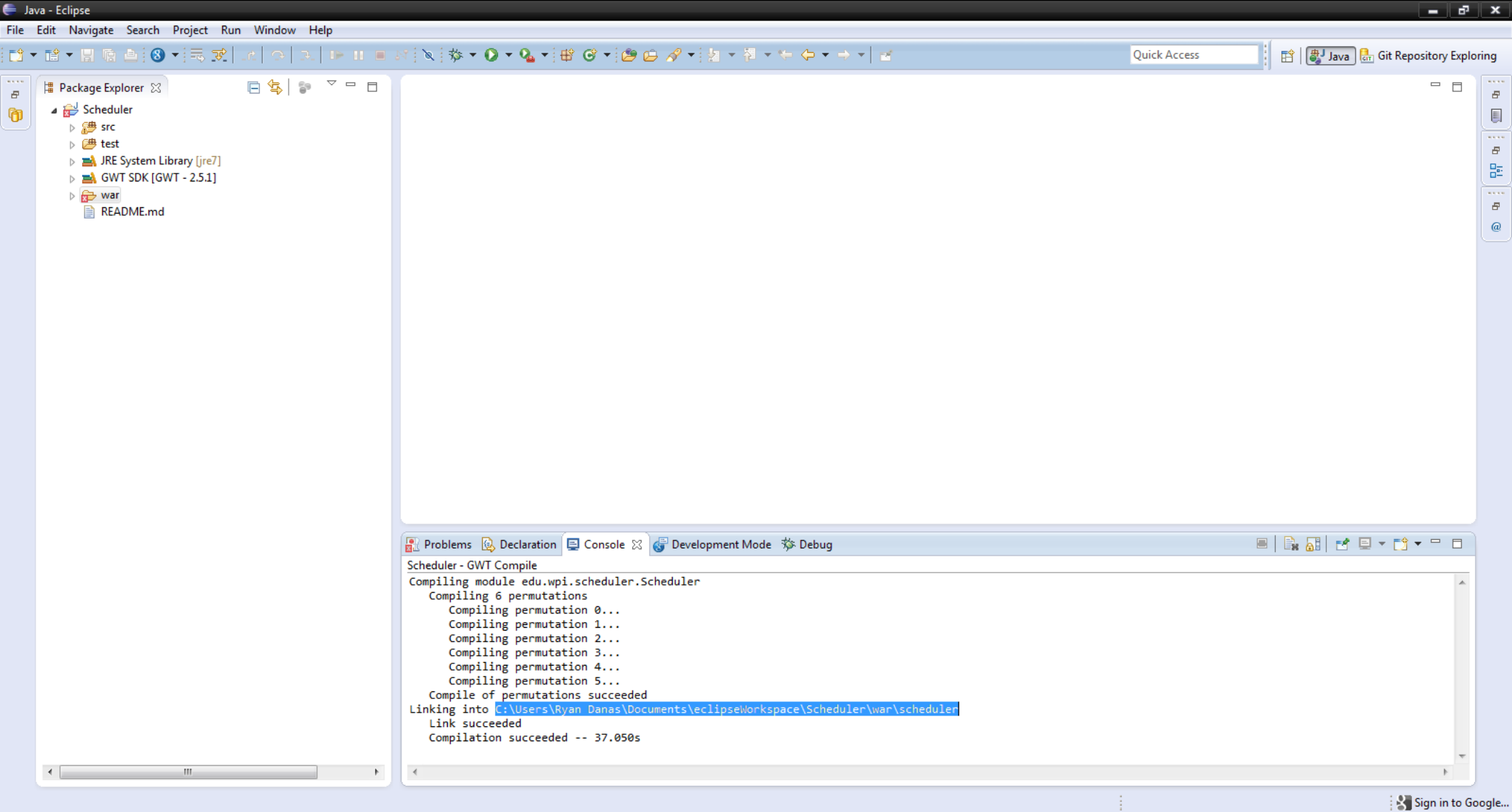
Import the archive file included with this report, or a copy you downloaded from GitHub. You should see the Scheduler project pop up as being available for import. Import the project into your workspace.



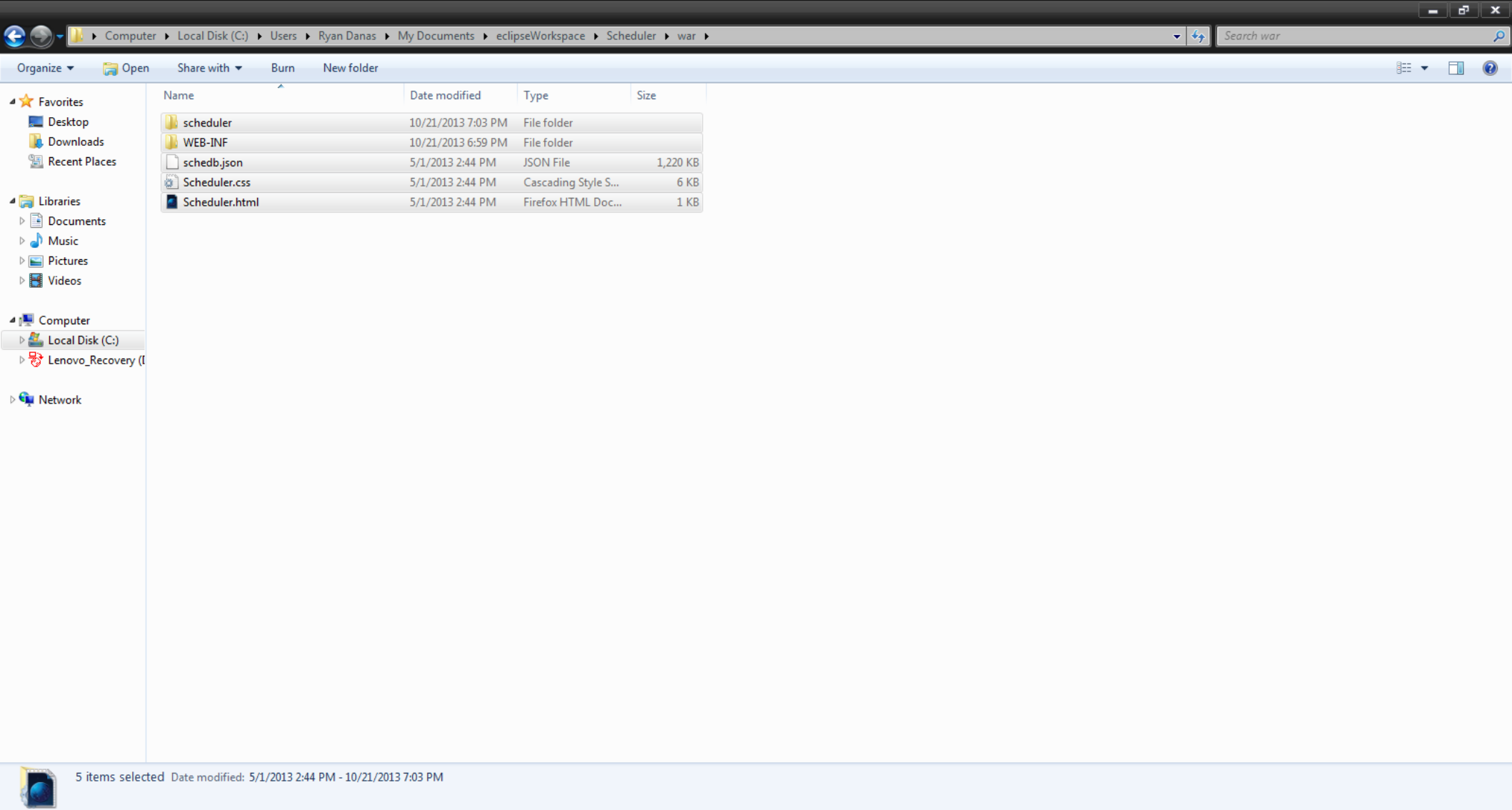
Once the project is imported, select it. From the GWT plugin menu, select GWT Compile Project



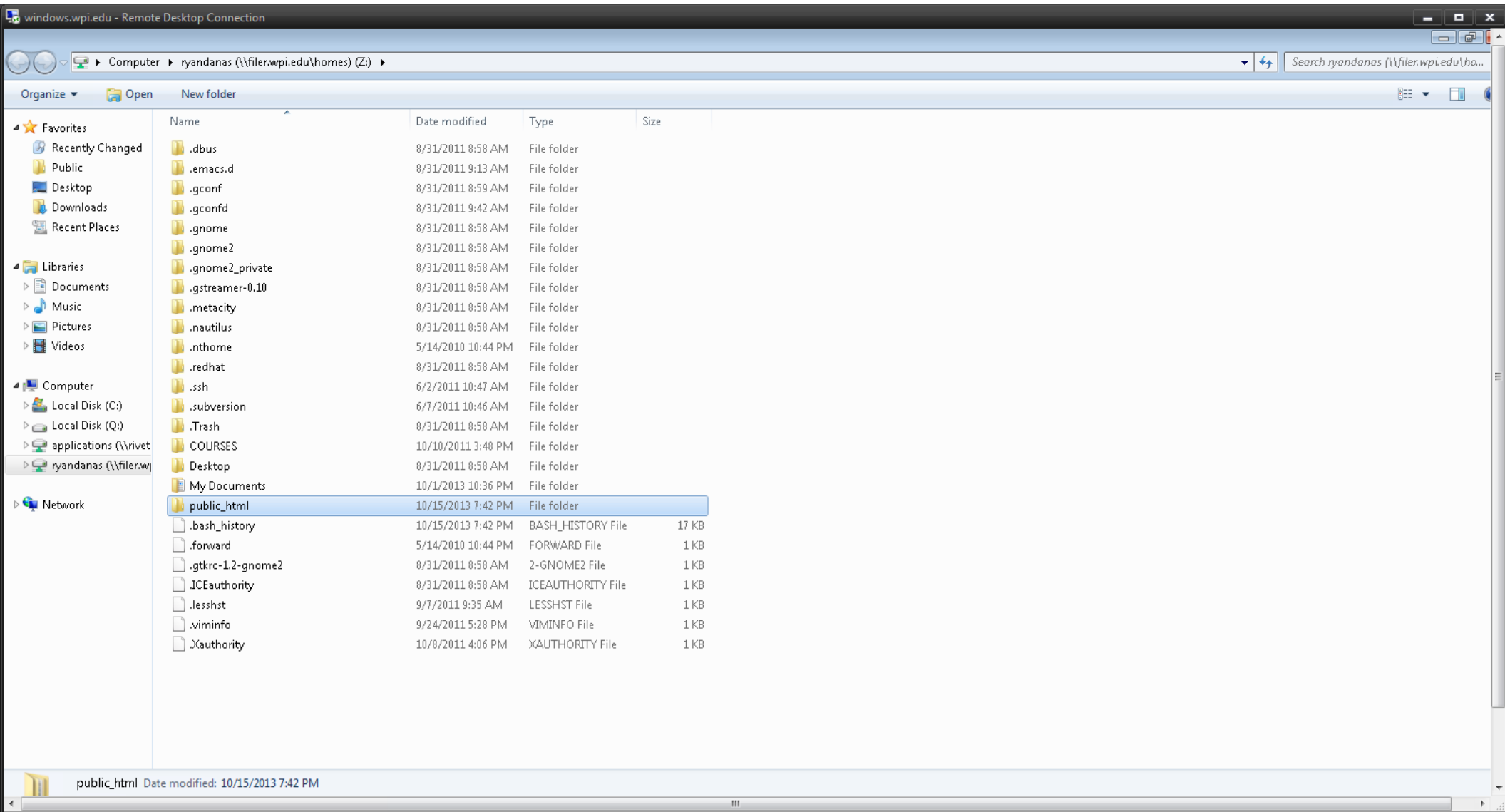
There are some files that are not automatically generated unless you edit and run the project; however, these missing files do not affect the project from being compiled properly, so the warning is safe to ignore. The default compilation settings are fine for hosting purposes. There is no need to edit them unless you have a particular reason to.



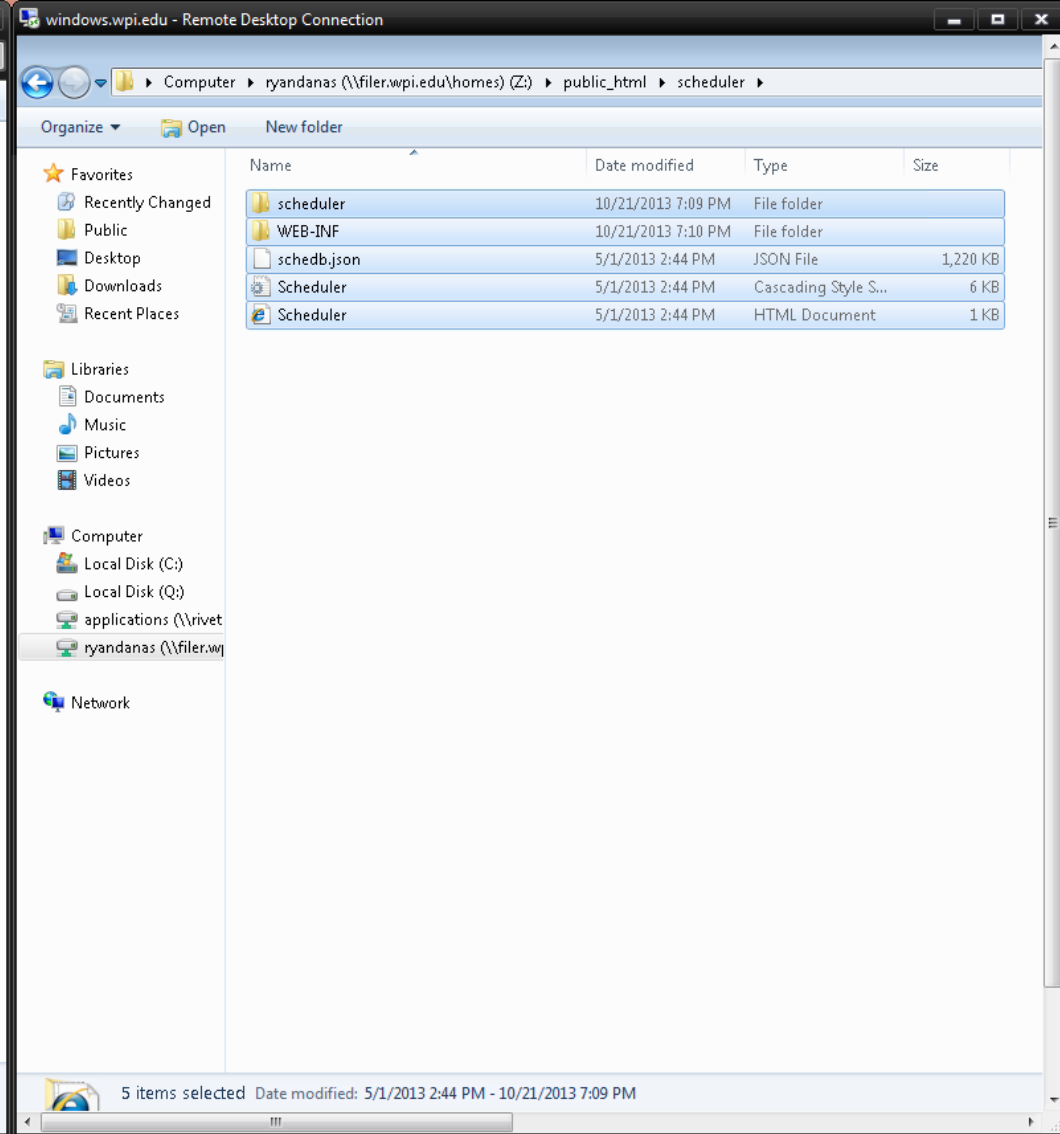
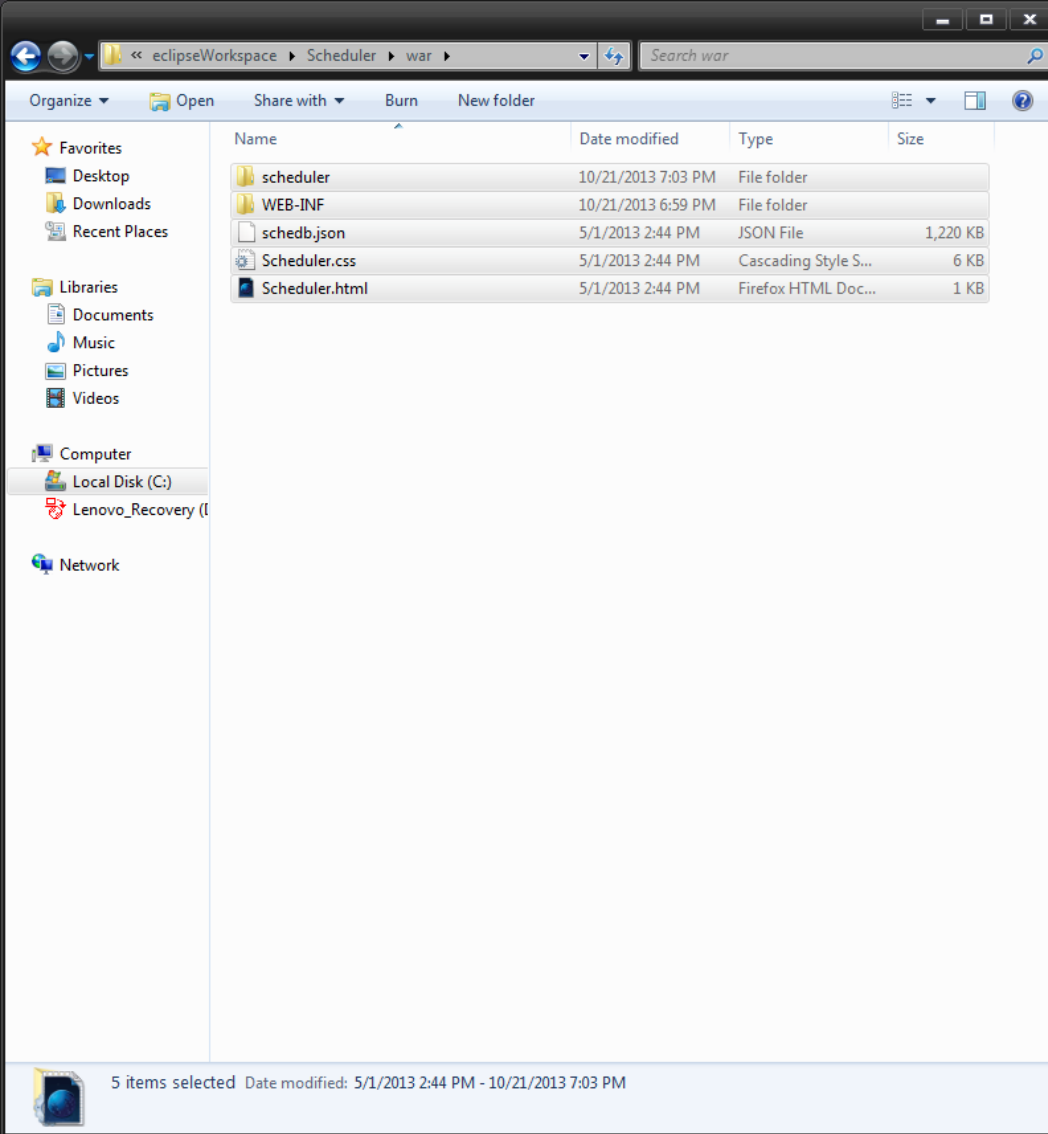
Once you select to compile the project, GWT's progress will be output to the console. Once the compilation is complete, take notice of the directory where the compiled files are located. It should be a directory similar to the one highlighted above.



After browsing to the directory, go up a folder into the root of the 'war' folder. This folder contains every file that must be copied to your web server in order for the scheduler application to properly run.



A common place for WPI students to easily host web pages is their own public html webspace. You can find it at the root of your Filer drive. For more information on how to access this drive, please contact WPI's Information Technology department.



Simply copy the files from the root of the war directory to the particular place you would like to host Scheduler. Depending on your web server setup, you may need to edit permissions to allow the public to have access to the newly copied files.

Finally, you must generate a JSON file containing the most recent class data. This can be done by downloading the XML file provided by WPI from <http://wpischeduler.org/run/wpi.schedbR> and passing that file as input to the included xmltojson.jar. The resulting output from that program, schedb.json, should also be placed in your public_html directory. For more specific instructions, please see the README.txt included with the xmltojson utility

The XML file is updated every 4 hours. It is recommended to set up a scheduled job to update your installation periodically.

These instructions are current at the time of this report. However, the availability of the XML file is contingent on WPI continuing to provide it in the same location. In the future, there may be a different location to access the XML file, and the XML file may be updated to contain more information. The application will continue to work regardless of which version is provided.

XML File Specification

There are 5 complexTypes allowed in the current XML file. All fields are required unless otherwise specified. Formal schema is available at

<http://wpischeduler.org/schedb.xsd>

1. Schedb

1. A sequence of dept objects, representing the different departments (see below)
2. String field “generated”, representing the date / time the file was created
3. Integer field “minutes-per-block”, representing the smallest duration of time that a period can last. (This field is currently not used by our implementation)

2. dept

1. A sequence of course objects, representing different courses (see below)
2. String field “abbrev” representing the abbreviation for this department
3. String field “name” representing the full name for this department

3. course

1. A sequence of section objects, representing different sections (see below)
2. (Optional) A sequence of notes, each of which must be a string
3. String field “number” representing the course number
4. String field “name” representing the course name
5. Double field “min-credits” representing the minimum number of credits that can be earned in this course
6. Double field “max-credits” representing the maximum number of credits that can be earned in this course
7. (Optional) String field “grade-type” representing the grading method used for this course
8. (Optional) String field “note” containing any additional information

XML File Specification (continued)

4. section

1. A sequence of period objects, representing different periods (see below)
2. (Optional) A sequence of notes, each of which must be a string
3. Long field “crn” representing the CRN for this section
4. String field “number” representing the number for this section
5. Integer field “seats” representing the total number of seats for this section
6. Long field “availableseats” representing the number of available seats (that is, those which have not been registered yet) for this section
7. String field “term” representing the year and semester of this section
8. String field “part-of-term” representing whether this section takes place in the first or second half of this semester
9. (Optional) String field “note” containing any additional information

5. period

1. String “type” which must be one of the following:
 - ind-study, lab, Lab, lecture, Lecture, conference, Conference, practicum, Practicum, recitation, Recitation, seminar, Seminar, studio, Studio, video tape, Video Tape, Video tape, web, Web, Blended Learning, blended learning, other, Other
2. String field “professor” representing the name of the professor teaching this period
3. String field “days” representing the days on which this period occurs
4. String field “starts” representing the time this period starts
5. String field “ends” representing the time this period ends
6. String field “building” representing the building where this period takes place
7. String field “room” representing the room where this period takes place

JSON File Specification

As there is currently no formal JSON Schema format, no fields are technically required. These are the objects and associated fields that our application will interpret. Any additional data will be ignored, the program can function with missing data. Currently, not all of these fields are provided by WPI's XML file.

1. Root
 1. String value "generated" representing the date the original XML file was created
 2. Number value "minutesPerBlock" representing the smallest duration of time a period can last
 3. Array of department objects "deparments", containing all of the departments
2. department
 1. String value "abbrev" representing the abbreviation for this department
 2. String value "name" representing the name of this department
 3. Array of course objects "courses", containing all the courses offered by this department
3. course
 1. String value "number" representing the course number
 2. String value "name" representing the course name
 3. String value "description" representing the course description
 4. String value "minCredits" representing the minimum number of credits that can be earned in this course
 5. String value "maxCredits" representing the maximum number of credits that can be earned in this course
 6. String value "gradeType" representing the grading method used for this course
 7. String value "note" containing any additional information
 8. String value "cross_list" representing all other courses that this course is cross-listed with
 9. Array of section objects "sections", containing all sections offered for this course

JSON File Specification (continued)

4. section

1. String value "crn" representing the CRN for this section
2. String value "number" representing the number for this section
3. String value "seats" representing the total number of seats for this section
4. String value "availableseats" representing the number of available seats (that is, those which have not been registered yet) for this section
5. String value "max_waitlist" representing the maximum capacity of the waitlist for this section
6. String value "actual_waitlist" representing the number of waitlist slots already taken
7. String value "term" representing the year and semester of this section
8. String value "partOfTerm" representing whether this section takes place in the first or second half of this semester.
9. String value "note" containing any additional information
10. Array of period objects "periods" containing the periods this section occupies

5. period

4. String value "type" representing the type of period
5. String value "professor" representing the last name of the professor / instructor
6. String value "professor_sort_name" representing the full name of the professor / instructor (Last, First, Middle)
7. String value "professor_email" representing the email address of the professor / instructor
8. String value "days", representing the days this period takes place. This should be a comma separated list containing any of the following in any order: mon,tue,wed,thu,fri,sat,sun
9. Number value "starts" representing, in 24hr time, the time that the period begins
10. Number value "ends" representing, in 24hr time, the time that the period ends
11. String value "location" representing the building and room where this period occurs.

Now that Scheduler is on your web server you should be able to access the application from any web browser. You may have to refresh once for everything to load properly. Other than that, Scheduler should now be hosted properly and accessible for anybody to use.