



ACM INTERNATIONAL COLLEGIATE PROGRAMMING CONTEST



California State University, Sacramento's

PC²
Version 9.6

Judge's Guide

<Last Update: 2018-02-06>

Table of Contents

1	Introduction	2
2	Logging in to PC²	3
	2.1 Login Screen	3
	2.2 Main Judge’s Screen	4
3	Judging Runs	5
	3.1 Selecting Judgments.....	5
	3.2 Manual Execution	6
	3.3 Test Case Viewing	6
	3.4 Test Result Options.....	8
4	Handling Clarifications.....	9
	4.1 Answering Clarifications.....	9
	4.2 Filtering Clarifications	10
	4.3 Generating Clarifications	10
5	Additional Options.....	11
	Appendix A – Log files	14

1 Introduction

PC² is a dynamic, distributed real-time system designed to manage and control Programming Contests. It includes support for multi-site contests, heterogeneous platform operations including mixed Windows and Unix in a single contest, and dynamic real-time updates of contest status and standings to all sites. In particular, it provides support for two distinct methods of *judging* such contests: *automated* (or *computer*) judging and *manual* (or *human*) judging. These two methods can be used independently or in conjunction with each other.

This Guide assumes the reader is familiar with the overall operation of a programming contest, and that the reader is also familiar with the steps involved in installing and configuring a PC² system (or else has another person acting as the Contest Administrator who can answer questions regarding such). For further information on PC², including how to configure the parameters which control and enable various judging operations, refer to the *PC² Contest Administrator's Installation and Configuration Guide*, which can be found at <http://pc2.ecs.csus.edu/>.

PC² operates using a client-server architecture. Each site in a contest runs a single PC² *server*, and also runs multiple PC² *clients* which communicate with the site server. One such client type is a *Judge*. A Judge client can operate in either *Auto-Judging* or *Manual Judging* mode. Auto-Judging (AJ for short) is normally set up by the Contest Administrator and happens automatically without human intervention once the contest starts, while Manual Judging takes place (once set up by the Contest Administrator) by having human judges log in to the system and interactively evaluate team submissions.

This Guide describes the steps involved in using manual judging in a contest using PC².

2 Logging in to PC²

2.1 Login Screen

In order to use a Judge client, there must first be a *judge's account* created by the Contest Administrator (again, see the *Contest Administrator's Guide* for details on doing this and all other admin functions). The Contest Administrator must provide each (human) judge with *login credentials*, consisting of a *login name* (also called an *account name*), and a *password*. You will need to obtain these prior to being able to use PC² for judging.

PC² login accounts for judges are by default named “judgeX”, where “X” is a number (the so-called *judge account number*). However, the names of client accounts are configurable by the Contest Administrator.

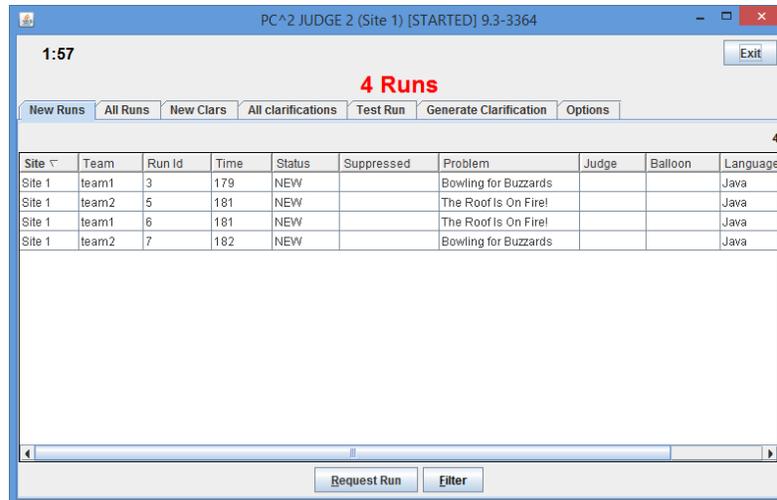
To login to PC², open a command prompt and change to the PC² installation folder. In that folder, type the command `./bin/pc2judge` (on a Linux or Macintosh machine) or `.\bin\pc2judge` (on a Windows machine). This will bring up the PC² Judge login screen, which appears as follows:



Enter the login (account) name and password as indicated, and press the Login button. This will start the judge login process. During startup, the judge client connects to the server and obtains the contest configuration. If the configuration is invalid (for example, required data files are missing), the judge client halts with an explanatory error message. Otherwise, the client completes the login process and continues to the main judge's screen.

2.2 Main Judge's Screen

Successfully logging in to the system will bring up the *Judge's Main Screen*, as shown below.



The tabs across the top allow you to display various sets of judging-related information, as follows:

New Runs: shows a grid of all runs which are waiting to be judged (notice that in the above view there are several runs waiting).

All Runs: shows a grid of all runs which have been submitted in the contest so far, whether judged or not.

New Clars: shows a grid of all Clarification Requests which have been submitted by teams but have not yet been answered by the Judges.

All Clarifications: shows a grid of all Clarification Requests which have been submitted by teams, both those that have been answered already and those that have not.

Test Run: allows the Judge to execute a program in the same way it would be executed if a team had submitted it.

Generate Clarification: allows the Judge to submit a clarification request as if the request had come from a team.

Options: allows the Judge to change various options including the account password and the way in which certain information is displayed

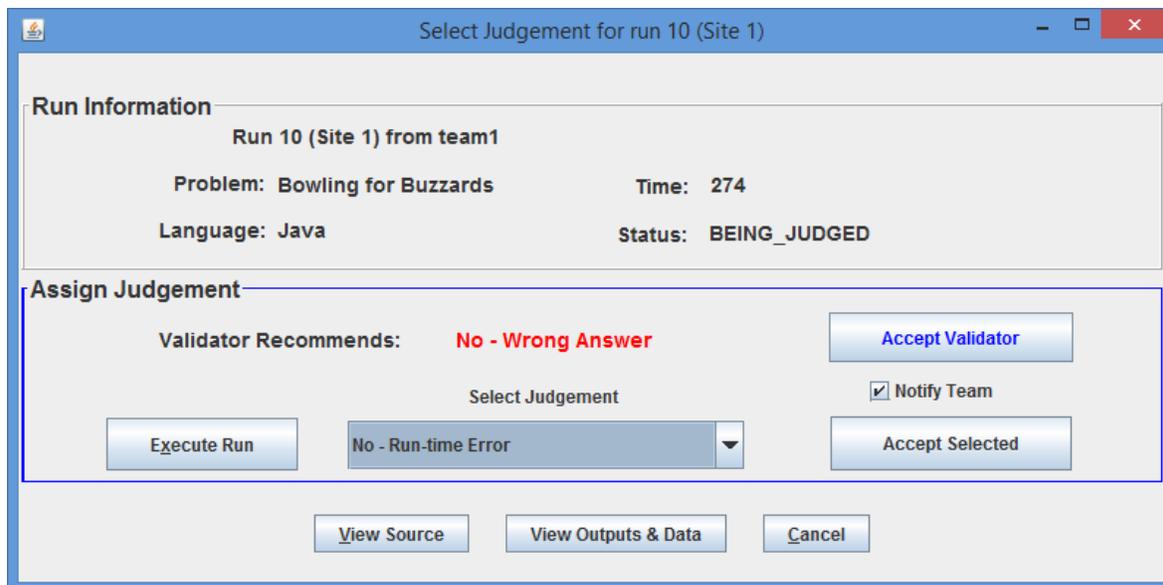
The current contest clock (the time remaining in the contest) is displayed in the upper left corner of the judge's screen. The "EXIT" button at the upper right allows the judge to exit (log out of) PC².

3 Judging Runs

In order to judge a run you must first request the run from the server. This is done by selecting the run (clicking on its row in the grid), and then pushing the “**Request Run**” button. (Double-clicking on a row in the grid also automatically issues a run request to the server.)

3.1 Selecting Judgments

When a run is fetched from the server, the following **Select Judgement** window is displayed:



The top portion of the window displays information about the run, while the middle section provides the place for the Judge to assign a judgment to the run. Note that in this example the problem has been configured with a *validator*¹, and the screen displays the information that the validator recommended that the judgment for the submission be “No – Wrong Answer”. Pressing the **Accept Validator** button will cause this judgment to be assigned to the submission and will close the **Select Judgement** window.

Alternatively the Judge can use the **Select Judgment** dropdown list and choose any of the available judgments which have been configured into the system by the Contest Administrator. In the above example, “No – Run-time Error” has been chosen by the Judge. Subsequently pressing the **Accept Selected** button will cause the selected judgment to be assigned to the submission and the window will be closed.

¹ A *validator* is a program that takes as input the output of an execution of the submitted program, and compares that output with the *judge's answer file* corresponding to the *judge's input data file* used as input to the team program during that execution, returning an indication of whether the team's output matches the judge's output according to some criteria (for example, simple diff, although many other approaches are possible). See the Contest Administrator's Guide for details about how validators are created and loaded into PC².

If the validator recommendation and the Judge’s selection differ when the Judge pushes either the **Accept Validator** or the **Accept Selected** button, the system will prompt for confirmation regarding which one should be applied. In either case, if the **Notify Team** checkbox is checked then the system will send a message containing the chosen judgment to the team. Before Version 9.6 the **Notify Team** checkbox would remember the last checkbox setting. In Version 9.6 and after the **Notify Team** checkbox will always be checked by default.

3.2 Manual Execution

Frequently it is desired to execute the program and examine the results manually before selecting a judgment (this is what is meant by *manual* or *human judging*). To do this, press the **Execute Run** button on the **Select Judgment** pane. This has several effects. First, it will execute the submitted program once for each judge’s test data set that has been configured into the problem by the Contest Administrator. As each execution takes place a small “timer” window will be displayed indicating either “Compiling”, “Executing”, or “Validating” (the latter only appears when the Contest Administrator has configured the problem with a validator).

After the program has been executed against all the configured test cases, the following **Test Results** window automatically appears, showing the results of each test case execution:

Select	Data Set#	Result	Time (ms)	Team Output	Compare Outputs	Judge's Output	Judge's Data	Validator Output	Validator STDERR
<input type="checkbox"/>	1	Pass	0	View	Compare	View	View	View	View
<input type="checkbox"/>	2	Pass	0	View	Compare	View	View	View	View
<input checked="" type="checkbox"/>	3	Fail	0	View	Compare	View	View	View	View
<input type="checkbox"/>	4	Pass	0	View	Compare	View	View	View	View
<input type="checkbox"/>	5	Pass	0	View	Compare	View	View	View	View
<input checked="" type="checkbox"/>	6	Fail	0	View	Compare	View	View	View	View
<input checked="" type="checkbox"/>	7	Fail	0	View	Compare	View	View	View	View
<input checked="" type="checkbox"/>	8	Fail	0	View	Compare	View	View	View	View

In this case the Contest Administrator has configured 8 “test cases” for the problem. As shown above, 4 of those cases passed while 4 failed. At this point one option is to recognize that if *any* test case failed then the submission should receive a “NO” response, and simply close the window and go back to the **Select Judgment** screen (above).

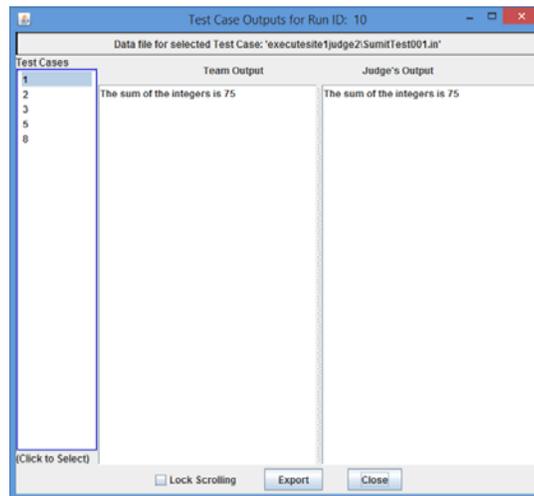
3.3 Test Case Viewing

The Test Results screen allows the Judge to perform a variety of useful viewing operations related to each test case. Each independent test case configured into the problem corresponds to a row in the Test Results grid. Across each test case row is a series of links; clicking on a link in the row invokes either a “View” or a “Compare” operation. For example, clicking on a “View” link in the “Team Output” column pops up a display window showing the team’s output for that test

case²; similarly, clicking the “View” link in the “Judge’s Output” column displays the Judge’s output (Judge’s Answer File) for that test case, and so forth.

Clicking on a “Compare” link in a test case row invokes a comparator tool that shows some form of “comparison” (e.g., side-by-side windows) between the team and judge’s output for that test case.

Another way to compare team output with judge’s output is to utilize the checkboxes on the left side of the Test Results window to select an individual test case, or a set of test cases; the **Select All** and **Unselect All** buttons can be used to speed up selection. Pressing the **Compare Selected** button invokes a tool that allows side-by-side display of the team and judge outputs for any of the selected test cases. The following screen shows the result of selecting test cases 1,2,3,5, and 8 on the Test Results screen and then clicking **Compare Selected**:

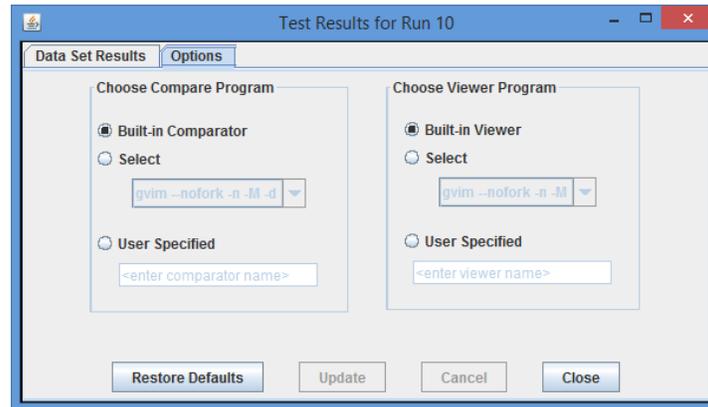


Clicking on an entry in the “Test Cases” column causes the Team Output and Judge’s Output panels to display the corresponding test case outputs. If either output is too long to fit on the screen then a scroll pane is applied, in which case the **Lock Scrolling** checkbox can be used to keep the two output displays in sync. Clicking the **Export** button allows saving the currently-viewed outputs to a file.

² In the current build of Version 9.3, the “Team Output” listing also contains the results sent to “*stderr*”.

3.4 Test Result Options

Clicking on the **Options** tab on the Test Results window causes the following display to appear:



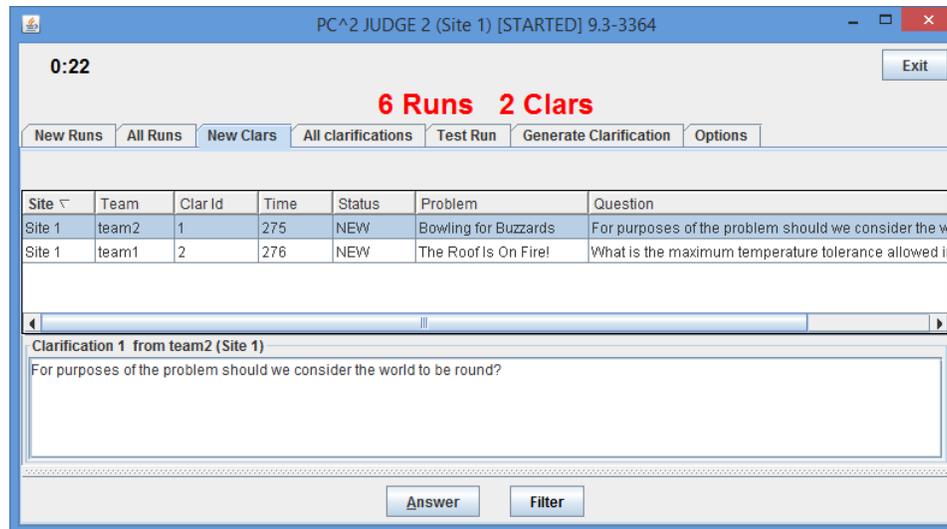
This screen allows selection of the program(s) used to perform the viewing and comparison operations between team and judge's output for different test data cases. By default a simple PC² built-in tool is used both for viewing output files and for comparing them. However, if other more complex viewers/comparators are configured into the system then those can be selected on this screen.³ As a third alternative, an arbitrary user-defined program can be externally invoked to do the viewing/comparison.

³ For example, many systems support the *gvim* program, which can be used to compare two files by collapsing similar sections of the files into empty space, leaving only the sections which are different to be examined (which is quite handy for judging). PC² has a built-in understanding of how to invoke *gvim* to do this (although of course it depends on *gvim* being installed on the judge's machine).

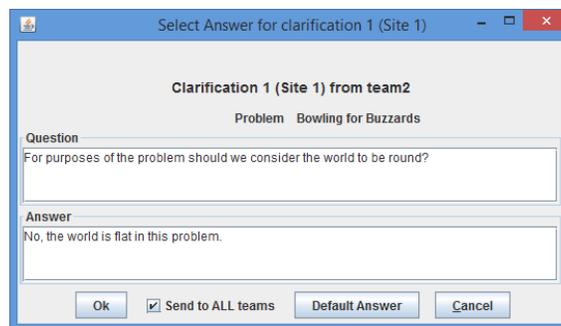
4 Handling Clarifications

4.1 Answering Clarifications

When a team submits a Request for Clarification (also known as just a “Clarification” or “Clar”), it shows up in a grid on the **New Clars** tab on the Judge’s main screen, as shown below:



Selecting a row in the grid causes the complete text of the team’s clarification request to be displayed in the bottom portion of the window, as shown above. Clicking the **Answer** button pops up the following window which is used to send an answer to the team:



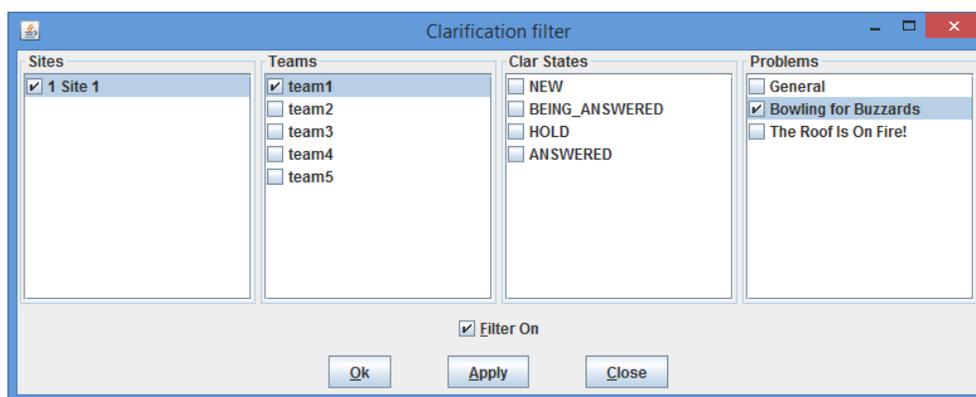
Any desired answer can be typed into the “Answer” box; subsequently pressing the **Ok** button will send the answer to the team which originated the clarification request. If the **Send to ALL teams** checkbox is checked *before the Ok button is pushed*, then the text of the clarification request as well as the judge’s answer will be sent to ALL teams in the contest.

Since many contests end up having teams submit questions that require the same answer (for example, “We’re not going to answer that; read the problem statement”), PC² supports the concept of a configurable *default answer*. The text of the default answer is configured into the system by the Contest Administrator (usually in consultation with the judges); thereafter, pressing the **Default Answer** button will automatically send the current default answer string back to the team (and to all teams if **Send to ALL teams** is checked).

The **All Clarifications** tab on the judge’s main screen displays a grid similar to the **New Clars** tab, the only difference being that only *new (unanswered)* clarification requests are shown on the **New Clars** tab while every clarification that has been submitted (whether answered yet or not) is displayed on the **All Clarifications** tab.

4.2 Filtering Clarifications

Pressing the **Filter** button on either the **New Clars** or **All Clarifications** screen pops up a dialog which can be used to filter (control) what is displayed in the corresponding list box:



Selecting the checkbox next to an item means that item will be *included* in the corresponding list box. However, a filter is not applied unless the **Filter On** checkbox is also checked. In the above example, the Judge has chosen to look only at clarification requests from Team 1 at Site 1 which are about the “Bowling for Buzzards” problem.⁴

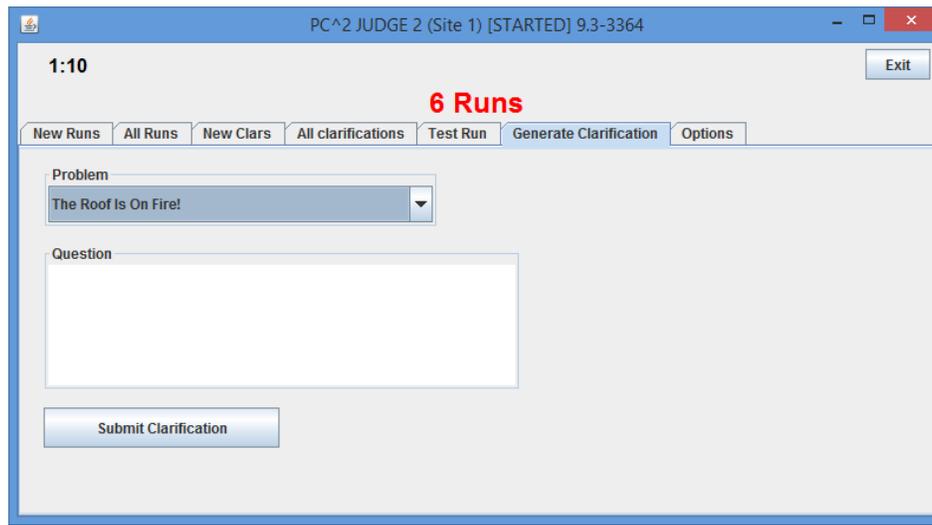
4.3 Generating Clarifications

It is frequently desirable for Judges to be able to send a notice to either a particular team or to all teams. Examples of such notices are corrections for an error in a problem statement, a notification to a team regarding a change of status of something, or a general announcement regarding the contest.

PC² does not have a “built-in messaging system” to use for such notices or announcements. However, it does allow the “Clarification system” to be used for this purpose. To send an announcement, a Judge basically *generates a clarification request*, and then turns around and

⁴ There is a drop-down list on the team’s GUI which allows them to identify the problem about which a clarification is being requested.

issues an answer to his/her own clarification request. This is done via the **Generate Clarification** tab on the judge's main screen, which looks like this:



The Judge uses the **Problem** dropdown list to choose one of the contest problems (the broader category “General” is also available), types a question in the box, and pushes the **Submit Clarification** button. For example, suppose the Judges wanted to make an announcement that lunch was available. They could do this by submitting a clarification question such as “When is lunch going to be available?” The normal clarification answering process (described above) can then be used to “answer” this question with something like “Lunch is available now in the Lobby” and sending that clarification/answer to all teams. A similar approach can be used if the Judges wish to issue an unsolicited clarification about one of the contest problems.

5 Additional Options

The Judge's interface provides several additional options, described in this section. Several of these are accessed through the **Options** tab on the main judge's screen:



The **Show PC² Logs** button pops up a visual display showing the contents of the log which the Judge constantly generates as events transpire in the system. The display updates dynamically whenever a new message is written to the judge's log; these log messages can be quite useful in diagnosing things when a problem occurs in the system. See the Appendices for more information about the PC² logs.

The **Change Password** button allows the Judge to, um, change the password for the currently logged-in account.

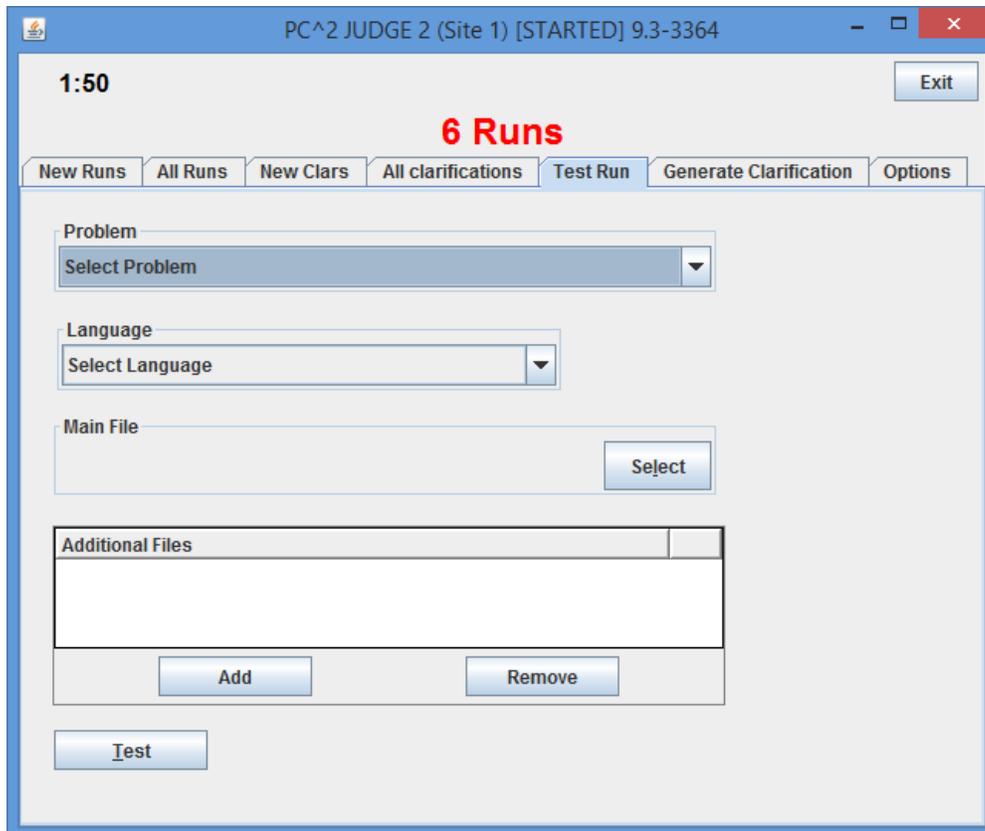
The **Show Unjudged Submissions Count** pops up a large display showing the number of unjudged submissions currently in the system:



This can be useful if a Judge is not able to stay close-by his/her system to monitor the “Runs Grid”; this display box can be easily read from across a room, giving the Judge a flexible way of monitoring the current status. The display updates dynamically as new runs are submitted, and it can be closed and reopened at any time without affecting any other portion of the system.

Another available option is the **Test Run** tab on the judge's main screen:

This screen allows the Judge to select a problem and language, and perform a “Test Run” by



submitting a program. Clicking the **Test** button will cause the system to prompt for the input data file to be used for the Test Run; the system will then execute the run as if it had been submitted by a team (however, such runs do not actually end up stored in the system; this facility is strictly for providing the judges with a simple testing mechanism).

Appendix A – Log files

PC² generates log files that track what is happening in the system and also record unexpected conditions. If something unexpected is happening, it is frequently useful to examine the log files to find out more detail about what is going on. Log files are stored under the **logs/** directory in the directory in which PC² was started.

There are 4 different log file types:

1. Startup log files – logging information before a module is logged in.
2. Module log files – logging information for a logged in module/client, **typically these are the files which should be checked for errors when running PC².**
3. Evaluations/judgments log – on server only, one line per judgment, see <http://pc2.ecs.csus.edu/wiki/Evals.log> for more details. These logs are not accessible (nor particularly useful) to a judge.
4. Security log files – logging security issues when they happen.