Jerry Sanders Design Contest PDA Judging Application
Developer's Guide

Table of Contents

Introduction

Each of the programs listed in this document need to be installed and configured properly on the machine you are using for development. You should start by doing the installation portion of each chapter. Then, create and initialize the database, and obtain the source code. After these steps, you should be able to run and test the application. You may want to read the User's Manual to get a better feel for how the pieces of the application work together.

CHAPTER 1: Microsoft Visual Studios .NET

1.1 Installation

The application is written in C#, and developed using Microsoft Visual Studios .NET. If you are a CS major at the University of Illinois, you can obtain a copy of this software for free from the University's MDNAA program. See http://msdnaa.cs.uiuc.edu/ for details.

1.2 Running the Application

We'll use the server as an example, but the following applies to the PDA interface, the Round Display, and the Match Display as well. It does not apply to the Database Library, which cannot be run.

Begin by opening JSDCServer.sln. This will open the server project and all of the code that goes with it. Then, in the Solution Explorer window, right click on the solution name JSDCServer, and go to Properties. There are two properties which may need to be set. The first is under Common Properties --> General --> Application. Set output type to 'Windows Application.' The second is under Configuration Properties --> Debugging --> Start Action. Set Debug Mode to 'Project.' With these two properties set, clicking on the blue triangle  will run the server and allow you to perform debugging actions such as setting breakpoints and stepping through a function.

If you do not want to debug the server, you can instead run it as a windows executable. Make sure the sever solution has been built with the properties above set,

and then run the server by double clicking on JSDCServer\JSDCServer\bin\Debug\JSDCServer.exe. Note that running the PDA interface through .NET will run it on a PDA emulator, which takes a long time to start up, so you'll probably want to just run the executable as often as possible.

If you're running something other than the server and having problems connecting, the problem is probably with the IP address. Make sure the IP addresses in the code is the IP address of the computer you're working on.

1.3 Running Automated Tests

We'll use the server as an example, but the following applies to the PDA interface, the Round Display, the Match Display and the DB Library.

Begin by opening JSDCServer.sln. Then, in the Solution Explorer window, right click on the solution name JSDCServer, and go to Properties. There are three properties which may need to be set. The first is under Common Properties General Application. Set output type to 'Class Library.' The second is under Configuration Properties Debugging Start Action. Set Debug Mode to 'Program.' Right below that, Start Application should be set to the nunit-gui.exe file. With these properties set, clicking on the blue triangle will run Nunit. You will then be able to run any or all of the tests found in the Tests folder under JSDCServer. See Chapter 2.2.

CHAPTER 2: Nunit

2.1 Installation

Nunit is a testing framework for all .NET applications. It is used to run automated tests, which are convenient when you want to make sure a change didn't break anything that worked before. You can download Nunit free of charge from http://www.nunit.org/download.html. The current (win) production release is fine.

You may have to add a reference to nunit.framework.dll

(located in the bin directory, wherever you installed Nunit). In .NET, in the solution explorer, right click on ‘References’ and select ‘Add Reference.’ Then go to Browse, and locate the nunit.framework.dll file.

If Nunit is not referenced correctly, you’ll get a build error similar to:

“The type or namespace name 'NUnit' could not be found (are you missing a using directive or an assembly reference?)”

2.2 Running Automated Tests

Run Nunit as described in chapter 1.3. Now you should see the Nunit GUI, with a list of available tests down the left hand side. You can run all of the tests by highlighting the top level and clicking Run, or a subset of tests by highlighting a lower level in the hierarchy and clicking Run. As the tests complete, the circle will turn green (test passed) or red (test failed).

Note that when you are writing tests, the order that they appear (and therefore, the order that they run) is NOT the order that they appear in the files, but rather alphabetical order. In the server, you can see that we used a numbering system to control the order the tests run in, since some of them rely on previous tests having passed.

If a test fails with a socket reuse error, the problem is probably the IP address in TestServer.cs (for example). Make sure all of the IP addresses in the code are the IP address of the computer you’re working on.

CHAPTER 3: Mysql

3.1 Installation

Mysql is used for the database behind the application. You can download mysql free of charge from http://dev.mysql.com/downloads/mysql/4.1.html (windows essential package). You’ll also probably want the query browser, which you can get from http://dev.mysql.com/downloads/query-browser/1.1.html.

3.2 Database Creation and Initialization

After you install mysql, you need to create the database, create the account that the application uses to access the database (cs427/jsdcpda), and create the tables. You can login as the root account that you set up during installation to do this.

The sql for creating the database is located in the setupdb.sql file, which can be found in your JSDCServer folder, or at http://cvs.sourceforge.net/viewcvs.py/jsdcpj/JSDCServer/.

In order for the server to start, the matches for the first two rounds need to already be in the scores table in the database. At some point, you should be able to get a list of the competing teams from the JSDC directors, and then you can create the first two rounds. Until then, you can use data from the previous year's competition, located in the initializedb.sql file, which can also be found at http://cvs.sourceforge.net/viewcvs.py/jsdcpj/JSDCServer/ or in your JSDCServer folder. An easy way to run this is to open the query browser, go to File Open Script, and open the initializedb.sql file, then click the Execute button right above where the script opens.

Before you run the server each time, you should delete any rows from the scores table that have a round number greater than two, and reset the scores from the first two rounds to 0.

```
delete from scores where roundNo > 2
update scores set score = 0
```

CHAPTER 4: TortoiseCVS

4.1 Installation

TortoiseCVS is a graphical interface to the CVS system provided by sourceforge. You can download it free of charge from http://www.tortoisecvs.org/.

4.2 Obtaining the source code

Although you don't need a sourceforge account to view the code, you will need it later when you want to update it. If you don't already have one, it's free! Signup at https://sourceforge.net/account/newuser_emailverify.php. Once you have an account, you need to be added to the project. Email an existing admin (https://sourceforge.net/project/admin/?group_id=123412) and he or she will add you.

After, you have an account and have been added to the project, create the folder where you want to store all of your source code. Then, right click on the folder and choose 'CVS Checkout.' Use the following options:

Protocol - Password server (:pserver:)
Server - cvs.sourceforge.net
Repository folder - /cvsroot/jsdcpj
User name – the account you just created
Module – . (just a period)

This will checkout all of the modules into the folder you selected. The modules that you will actually use are JSDCPDAGUI, MatchDisplay, RoundDisplay, JSDCServer and JSDCDBLIB. You can checkout each module separately if you'd rather not have the other unnecessary modules.

4.3 Committing changes

You should commit your code changes to the repository frequently, so that you have a lot of points to roll back to if problems occur. As a general rule, any time you have working code and have added a piece of functionality, you should commit it. To do this, right click on the file you changed (it should have an orange arrow icon, to indicate that it has been changed), and choose CVS Commit. If there is more than one person working on the source code, you should do a CVS Update on the file before you change it, to make sure you are changing the latest version. Then make your changes, and commit right away so other people don't make changes to an old version.