Williams' Notation

This notation was invented by the author to help solve some of the drudgery of note taking in Computer Science classes. The problem this notation solves is one familiar to most computer science students and practitioners. Often one must describe something that is a place-holder. Once very good example is describing something in BNF (Backus-Normal-Form if you have something against Mr. Naur; or Backus-Naur Form if you don't have anything against Mr. Naur [for a more complete discussion please see this Wikipedia entry). Or another is where the author wants to indicate that a file-name would be inserted into a particular position in a command line. For example:

"The student should type: `cat <insert_a_file_name_here>`".

The problem isn't the `insert_a_file_name_here` clause but rather telling the student NOT to type the left and right angle-brackets. Williams notation is to simply note the non-entered entities at the end of the sentence, enclosed by some meta-character of its own. So the above sentence would become:

"The student should type: **cat <insert_a_file_name_here>**. **/< >/**"

The notation **/< >/** indicates the angle brackets are not typed and do not belong to the clause **insert_a_file_name_here**. You can also indicate that a punctuation or phrase is simply intended for emphasis and not to be considered part of the sentence itself. For example:

"The student should type **cat <insert_a_file_name_here>**". **/< > " "/**

The space between the characters is intentional so that the characters are more easily spotted. This is called "visual acuity" and is something that the author wishes was more deeply considered by computer program language authors. Sometimes a brace and a parenthesis are indistinguishable in print or on a fading screen with glare: {(

Lastly in the notation the meta-character does not have to be a slash[1]. It may be any character you wish to use. Just use it in pairs.

How to install

Eclipse

To install just drage the com.sigh.jar file to your eclipse/plugins folder. This folder should be located in your eclipse install folder. On Mac OS-X the eclipse install folder is normally found in the /Applications folder.

Once you install the com.sigh.jar file you will want to know how to use the jar file in a project. So here is a little tutorial (This tutorial is from the point of view of Mac OS-X, but is easily extrapolated to *nix and/or other operating systems):

1. Open eclipse and close all your open projects (not necessary strictly speaking but this step cuts down on a great deal of confusion).

2. Close any open source files on display on the workbench (again, this cuts down on confusion).

3. Locate the following menu items: File->New->Project and select it.

---

1 Slash is the character "/". /" "/ More properly known as a virgule. It is NOT the reverse-virgule or "\". /" "/ This is another lunacy perpetrated by MS bigots. A / is a slash and a \ is a back-slash.

4. Locate the Java Wizard in the dialog box and click on the triangle next to the word Java to open the tree. Find the Java Project wizard and select it.

5. Click on the Next > button.

6. Name the Project "TestSIGH" /" "/ Note: If this is the first project you have built since installing eclipse then stop now and read the eclipse tutorials.

7. Click on the Next > button.

8. Click on the Libraries Tab.

9. Click on the Add External JARs... button.

10. From the pop-up file selector navigate to the plugins folder inside of your eclipse install folder where you dropped the com.sigh.jar.

11. Click on the com.sigh.jar file entry to select it.

12. Click on the Open button.

13. Click on the triangle next to the com.sigh.jar icon that appears.

14. Find the Javadoc location: (None) line and select it.

15. Click on the Edit button.

16. Click on the Javadoc in archive radio-button.

17. Click on the Browse button.

18. From the pop-up file selector find com.sigh.jar and select the entry.

19. Click on the Open button.

20. Click on the Open button next to the Path withing archive: text-entry box.

21. From the pop-up dialog click on the triangle next to the /Applications/eclipse/plugins/com.sigh.jar icon.

22. Select the doc folder icon.

23. Click the OK button.

24. Click the Validate button.

25. You should get a notification indicating the following text:

    Location is likely valid. Files 'package-list' and 'index.html' have been found. Open the location in your browser?

26. Click on the OK button.

27. You may have to repeat steps 24-26. Depends on your browser and eclipse.

28. You should now see the JavaDocs.

29. Read at your leisure.

30. Close the browser window.

31. Click on the OK button.

32. Click the Finish button.

33. You may or may not get the Open Associated Perspective? Pop-up dialog box. Do what you want to do if you do get the box but eventually click on the Yes button.

34. If it is not open, open the TestSIGH icon and select the src icon.

35. Control-click (or right-click) and select NEW->Package from the pop-up.

36. In the Name text-entry box enter "testSIGH" as the name. /" "/

37. Click the Finish button.

38. Control-click (or right-click) on the new testSIGH package icon and select NEW->File

39. In the pop-up dialog box's Text-Entry box for File name: enter "TestSIGH.java" (don't forget the ".java"). /" "/

40. Click on the Finish button.

41. Cut and paste the following source into the new source editor window.

```
package testSIGH;

import com.sigh.utils.strings.c_functions.Csprintf;
import com.sigh.utils.strings.c_functions.CsprintfExceptions;




public class TestSIGH {
    private int negOne = -1;
    private long negOneLong = -1L;
    private String aBevyOfPercents = "%% %% %%%%%%%%";
    private Object[] values = new Object [ 3 ];

    public TestSIGH() {
        Csprintf csf = new Csprintf();
        try
        {
            System.out.println ( Csprintf.sprintf ( "%s", aBevyOfPercents ) );
            values [ 0 ] = new Integer ( negOne      );
            values [ 1 ] = new Long      ( negOneLong );
            values [ 2 ] = new String ( aBevyOfPercents );
            System.out.println ( csf.doSprintf ( "%u %u %s", values ) );
        }
        catch ( CsprintfExceptions e )
        {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void main ( String[] argv ) {
        TestSIGH ts = new TestSIGH();
    }
}
```

42. Find the following menu items: Run->Run As->Java Application

43. Your should get the following results.

```
% % %%%%
4294967295 18446744073709551615 % % %%%%
```

44. To include in an existing project you select the menu item Project->Properties

45. From the pop-up dialog box select Java Build Path and then go to step 8 of these instructions.

The code is not great Java code. I tried to make it look familiar to VC++ users.

JavaDocs

To include the JavaDocs start by selecting Project->Properties and from the pop-up dialog box select the Libraries tab. Then continue on step 13 of these instructions.

Document End