

Kawa Frequently Asked Questions

or The 11 Habits of Highly Effective Kawa Users

Do you know the following?

- We recommend that you open and run all projects from H drive, but **you** will need to copy them there initially (see **Section 1** for details on how to do this)
- In case you make a mess of your code and would like a fresh version of a particular file or project there is a backup of the original files that you can copy from (see **Section 1.2** for details)
- The Kawa program can be started through the Windows Start pop-up menu, or Windows Explorer (see **Section 2** for details)
- The Kawa interface contains windows for managing projects, editing Java files and looking at the text and errors that your program produces (see **Section 3** for details)
- Generally each Java Class is saved as a different Java file. As a program may require several objects (such as for example, ones for managing circles, squares and lines in a drawing program), Kawa Projects allow easy managing of all the associated files (see **Section 4** for details)
- Do you know how to Open, Compile and Run a particular project (see **Section 4.1** for details)
- There is a difference between open and registered projects that you need to be aware of in running and compiling code. That is, when you hit “compile” or “run”, Kawa does not necessarily use the Java files that you have open, but rather the ones associated with the currently open project (see **Section 4.2** for details).
- Do you know how to construct a new class or project (see **Section 5** for details)
- Do you know how to find compilation errors in your code (see **Section 6** for details)
- If you are installing Kawa at home you will need to specify the Classpath and the JDK binary directories for the program to work. On the Java Genesis disk, there is a file called “readme.txt” which tells you how to do this and details the rest of the installation process.
- There is a more detailed Kawa tutorial (called KawaRead.doc) contained on the disk that came with the Java Genesis book.

Section 1: Genesis File Facts


1.1 Using H Drive

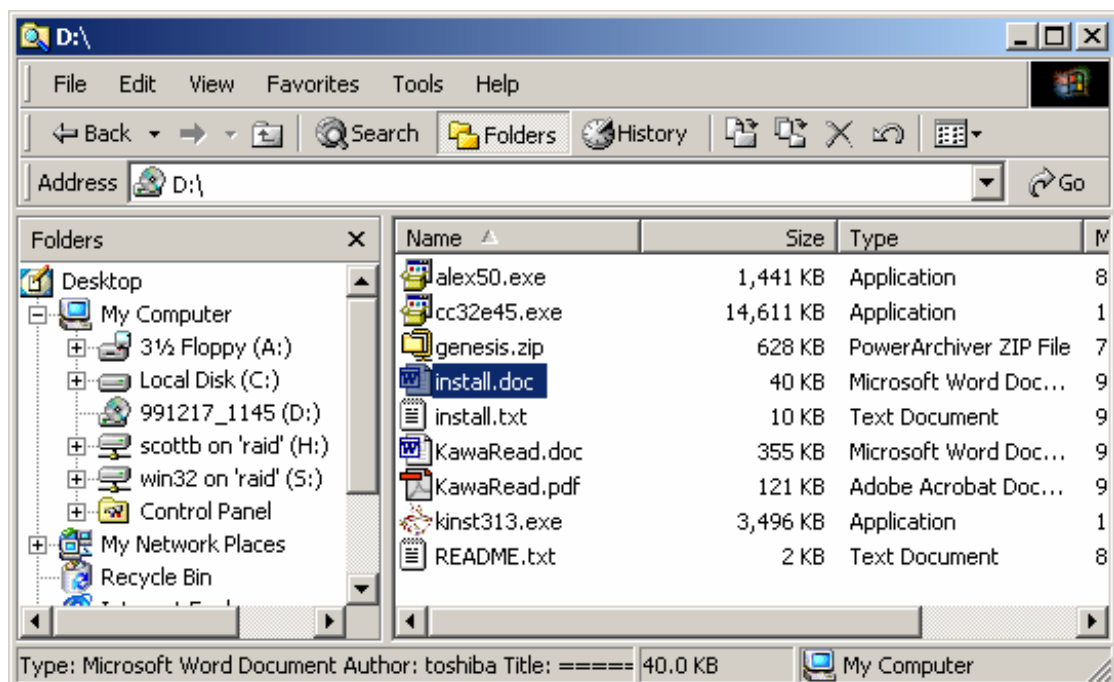
In COMP1500/7901, the genesis projects that you need to open and modify are initially stored on both the C drive and the G drive under the directory “genesis\projects...”. Although you can modify the code that is stored on C drive, these modifications **will be lost** when you log off. For this reason, we suggest at the beginning of the course, making a copy of the projects folder (and only the projects folder) on your H drive. The H drive is your personal student area, and any modifications that you make will be kept for future sessions. Thus, when using Kawa, make sure that you are opening the projects that have been copied to the H drive rather than C. If you are unsure how to use Windows Explorer to copy files or folders, see section 1.3

1.2 The G Drive – What to do when you make a mess of the code

The G drive was set up so that if you make a mess of your code, you can make a fresh copy of the original files to use. The G drive, in contrast to C and H, contains a **read only version** of the files, meaning that you can load and copy but not modify the code that is stored on it. If you are unsure how to use windows explored to copy files or folders, see section 1.3.

1.3 Using Windows Explorer to Copy Files

In Windows 2000, Windows explorer can be run by clicking on the  icon on the toolbar at the bottom of the screen. Earlier versions of Windows may require you to run the program through the Start pop-up menu (it can usually be found in the Programs section). When run, a window like the following will appear:



This window is broken into two main sections, one containing the folders (to the left of the window), and one containing the corresponding files (to the right of the window).

To copy a file or a directory takes four steps:

(1) navigate and **select the file or folder** that you want to copy. For example, if you wish to copy all the Genesis Project files across to H:, select My Computer from the Folder section (by double clicking). When selected, the file information will expand, showing subfolders etc. underneath the selected folder. Then select "(G:)", followed by "Genesis." In the file section (in the right of the window), you should see the folders "Hints", "packages" and "projects." Select "projects" by single clicking on this item.

(2) From the "edit" pull-down menu in Windows Explorer choose "**copy.**" This will create a copy of the files that you have selected that can be "pasted" elsewhere.

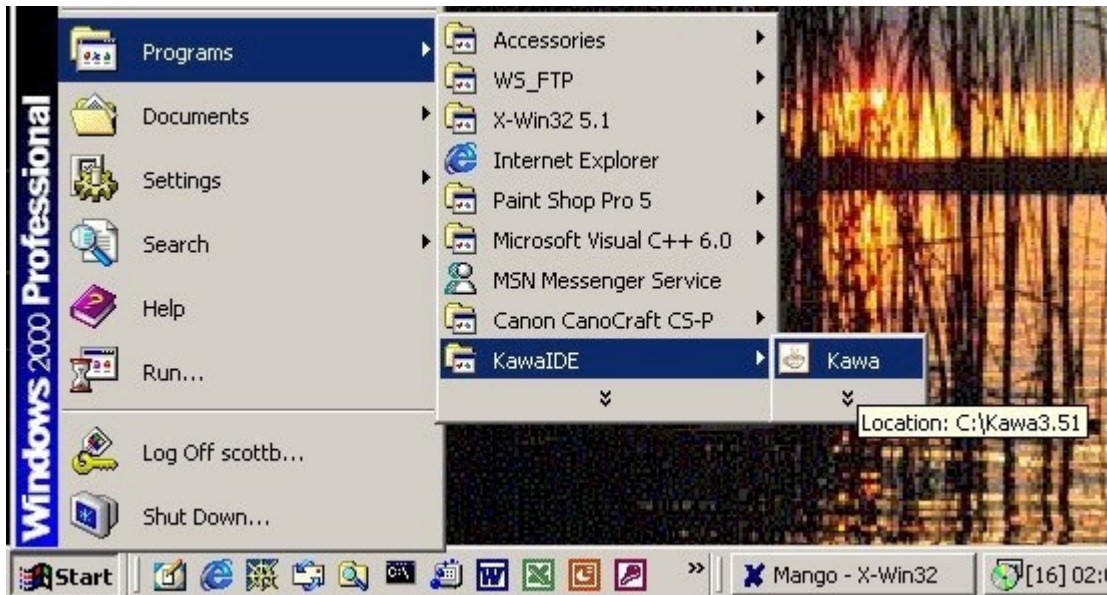
(3) **Select the destination** to where you want to place the files, in a similar way to step 1. If you are copying Genesis project files, you should select the "(H:)" directory.

(4) From the "edit" pull-down menu, choose "**paste.**" This will then copy all the files across to the desired location.

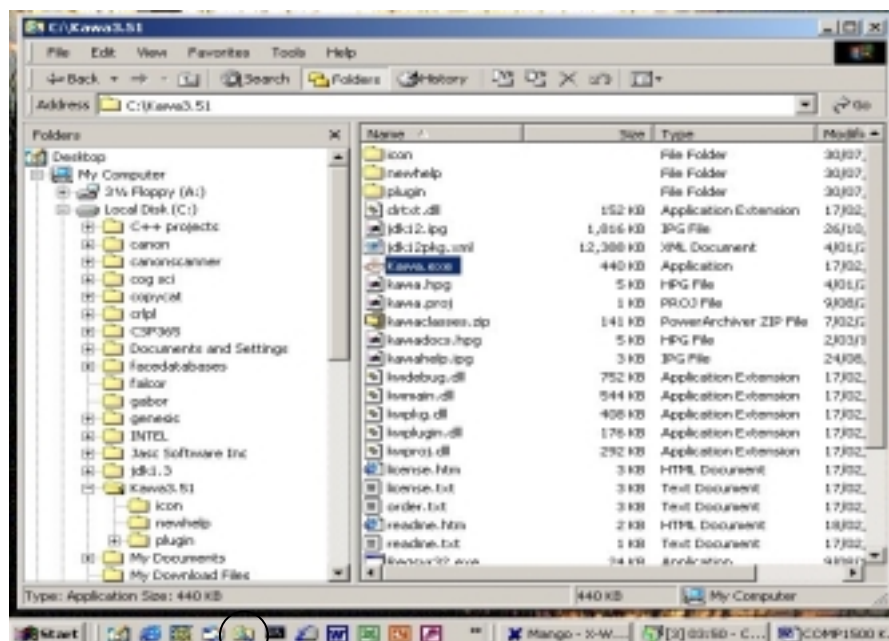
Section 2: Running Kawa

There are two simple ways in which to start the Kawa Program.

- (1) Kawa can be run by navigating through the Windows **Start pop-up menu**.
Click on Start – Programs – KawaIDE – Kawa

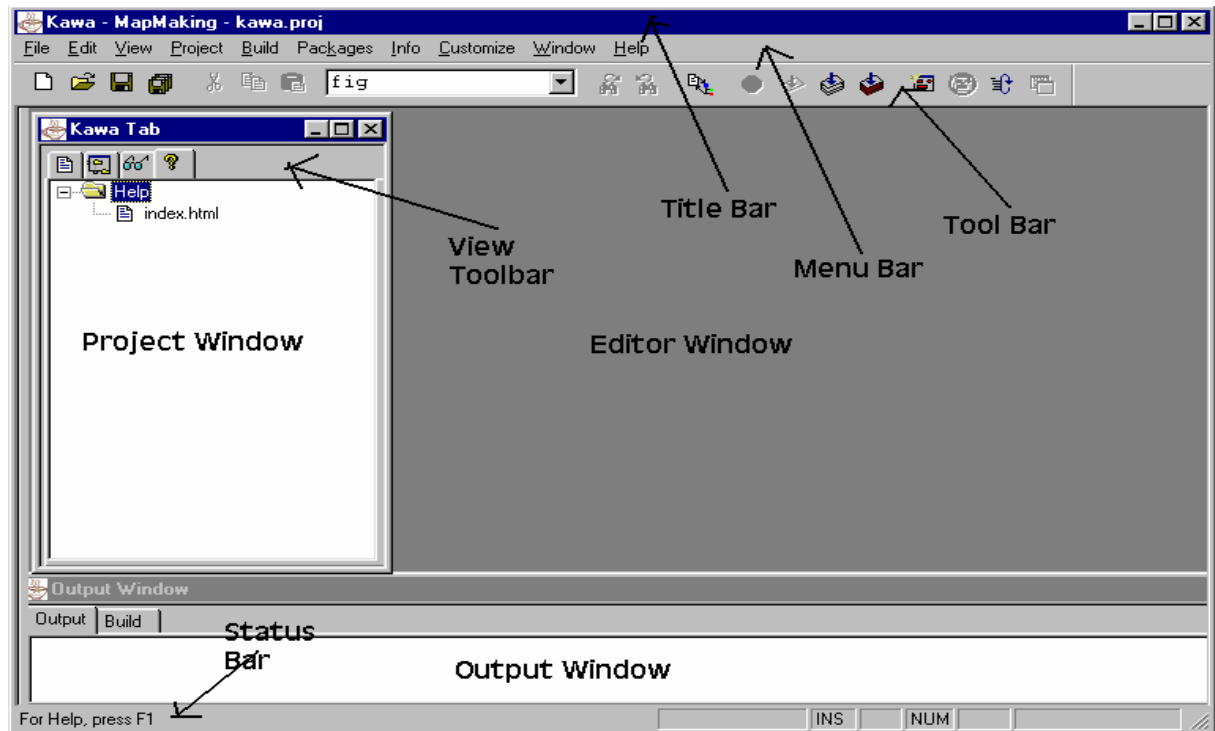


- (2) If for some reason, Kawa does not appear in this menu system, Windows Explorer can be used to run the file Kawa.exe.



Click [here](#) to open windows explorer (see section 1.3 if this doesn't exist). Open the "My Computer" folder in the left part of the window (by double clicking) followed by "Local Disk (C:)", and "Kawa3.51." Double clicking on Kawa.exe in the right section will then start the program.

Section 3: The Kawa Interface at a glance



Below are the main sections of the Interface that you should become familiar with:

Project Window

In writing Java code, it is common to save the code for each different Java Class as a separate file. As projects may need to contain several Classes (for example, a drawing program may need separate classes for circles and squares etc.), Kawa allows you to keep track of what different files are included in each project. Furthermore, Kawa allows you to keep track of several different projects that you are working on.



The Kawa Project window contains a list of Projects that you may be currently working on. Double clicking on the name of the project will give you a list of the Java files that the project contains (you can double click on the name again to remove this list). Double clicking on the name of any of the Java files will open them in the editor window.

It is important to note that this window also indicates which of the projects is currently active (the one that is in bold). When you hit the "run" button in the Tool Bar, this is the project that will be executed. Right clicking on the name of a Project and selecting "open" from the pop-up menu will change this to the active project. More information about opening and creating projects can be found in **Section 4**.

Menu Bar

The *Menu Bar* provides us with a selection of menus allowing you to accomplish such tasks as opening projects and creating new Java files (see **Sections 4-5** for details).

Tool Bar

The *Tool Bar* appears below the Menu Bar. It contains buttons for the most frequently used commands from the menus such as compiling () and running () java projects.

Editor Window

The entire region below the Tool Bar is the *Editor Window*. This window can contain many other windows, in particular the *Project Window*, the *Output Window* and text windows displaying the contents of opened Java files.

Output Window

The *Output Window* displays various types of information. When we compile our Java programs, the Output Window will display compiler information and list the errors that were found during compilation and the line numbers in the Java file where the errors occurred. We can also write our programs so that when they are executed they output information to the Output Window.

Section 4: Projects in Kawa

It is typical practice to save each Java Class as a different Java file. However, many programs will require the use of several objects. For example, in creating a drawing program, you may require a selection of objects such as circles, rectangles and lines. For this reason Kawa allows us to organise our Java programs by creating *projects*. Kawa creates special project files to keep track of which Java files belong to which projects (Project files have a *.kawa* extension.) We can create new projects or open existing ones previously created with Kawa. The following sections indicate how to load existing projects, the difference between registered and open projects, how to open Java files contained in a project, and how to compile and run projects. **Section 5** tells you how to create your own Java Classes and projects.

Section 4.1: Opening an Existing Project

Frequently throughout *Java Genesis*, you will be instructed to open a folder with a specific name. For each problem in each chapter in the text, there is a distinct folder containing a prepared Kawa project. We store within each project folder all the files required for that problem as well as the Kawa project file for Kawa's use.

For an example, let's open the project *BubbleSortDemo* in *Chapter 5*.

- In Kawa, select the Project Menu and choose *Open*.
- Navigate to the *Chapter 5* folder (**make sure it is on H drive if you have copied the files as suggested**), thence to the *Bubble Sort Demo* folder. At this point, a file *BubbleSortDemo.kawa* will appear in the list of Kawa files in this folder.
- Click on this file and then press open. Our Project Window changes to include the files contained in this project. Its appearance is now as below:



We can see that this project is made up of three Java files. Studying the colours of the file icons indicates that the files have been compiled earlier and that the file **BubbleSortDemo.java** is the main class (the entry point to the program) as it has a red box around its icon.

Section 4.2: Registering, Opening and Closing Projects

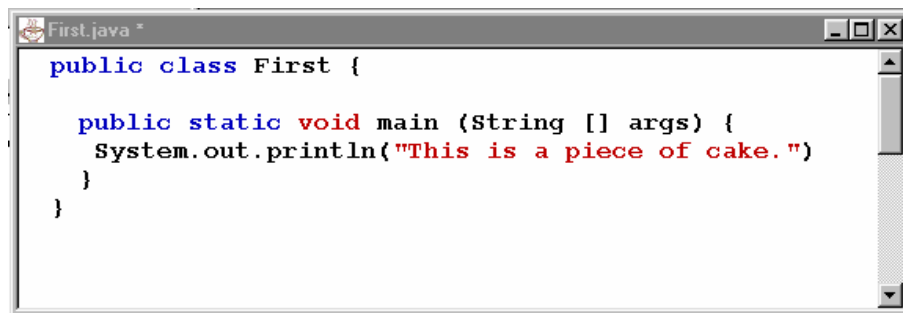
As can be seen from the previous figure, the Kawa Tab window can contain information about several projects. The projects contained in this list are said to be “registered.” However, it is important to note that **only one of these projects can be open at a time**. When the commands “run” or “compile” are given, it is only the open project that these refer to. That is, even if there are open Java files in the Editor Window, these may not be the actual files that are run or compiled.

To “open” a registered project, simply right click on the name of the project in the Kawa Tab window, and select the “open” option. The project name will then appear at the top of the list, with the names of the associated Java files appearing underneath as in the previous figure.

If too many projects are contained within the Kawa Tab menu, and you wish to delete some, simply right click on the project name, and select “delete.” This will only remove the item from the list, but will not actually delete any files from the file system as perhaps the name (misleadingly) suggests.

Section 4.3 Opening a Java File Contained within a Project

Double clicking on the name of the Java file in the Kawa Tab window will open a new text editor window containing the Java code in that file.







```
First.java *
public class First {
    public static void main (String [] args) {
        System.out.println( "This is a piece of cake. " )
    }
}
```

This code can be edited by clicking where you want the cursor and typing on your keyboard in the usual way.

Section 4.4 Compiling and Running a project

There are three options contained within the Kawa Tool Bar for compiling code:

-  Compile – compiles **the Java file** that is open and active (has a highlighted title in the Java code window).
-  Rebuild All – compiles all the files in the **active (i.e., currently open) project**
-  Rebuild Dirty – compiles all the files in the **active project** that have been changed since the previous compilation.

To run the active project simply click on . **Note:** the active project may not correspond to the active Java code window. So, if you are modifying Java code and trying to run the corresponding program, make sure you have the correct project open!

Section 4.5 Saving Modified Code

Many examples in the Java Genesis book ask you to modify existing code. Remember, you should have already made a copy of the Genesis projects folder on your H drive, and you should be opening projects from there. To save the code that you have modified, there are two main options in the Tool Bar:



- Clicking on this option **saves all the files** that are contained in the open Kawa project.



Clicking on this option **saves the active file**. The active file is the Java file that is open and has its window bar highlighted.

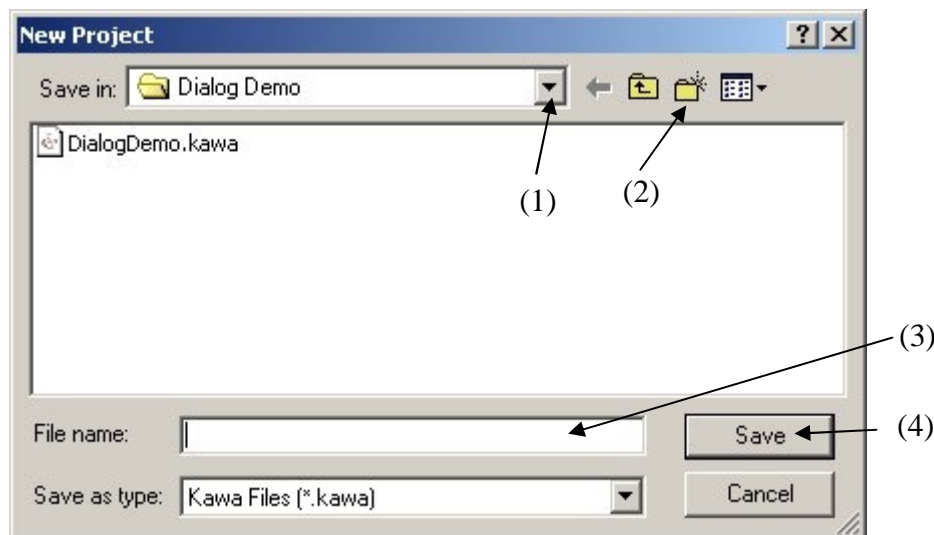
Section 5: Creating New Classes and Projects

In Java Genesis, occasionally you are requested to create a new class. Although classes can be run without creating an associated Project (if you know what you are doing), for consistency and ease of use we suggest creating a new project to place the new class in. So, in order to create a new class, you will firstly need to create a project (5.1) and then create the required Java file(s) to include in the project (5.2).

5.1 Creating a New Project

To create a new project in Kawa, we normally create a separate folder for the project so that the code for each project is in a different folder.

- Open the *Project* menu and select *New*. The *New Project* dialog box appears asking you to enter the name of your new project in the *File name* box:



At this point, you can create a new folder to place the project files in. This requires 2 steps:

- Simply, click on (1) and navigate the file system to select where you want the project to be stored.
- Clicking on (2) will create a new folder at that location. You will then need to type in the name of the folder, followed by enter.

Note: older versions of windows may not allow folders to be set up in this way, so you will need to create a folder (directory) in a way appropriate to the windows version you are using.

Once the new folder has been set up, you will need to use (1) to navigate into the new folder, and then type the name of the project into (3), and then click on “save” (4).

This will create a new open project in the Kawa Tab menu. For example, if you created a project called “First” the Kawa Tab menu may look like the following:



Note: as there are no associated Java files, even though the project is “open”, it is empty.

5.2 Creating New Java Files

We are now able to add files to this project – either existing files or new files that we can now create.

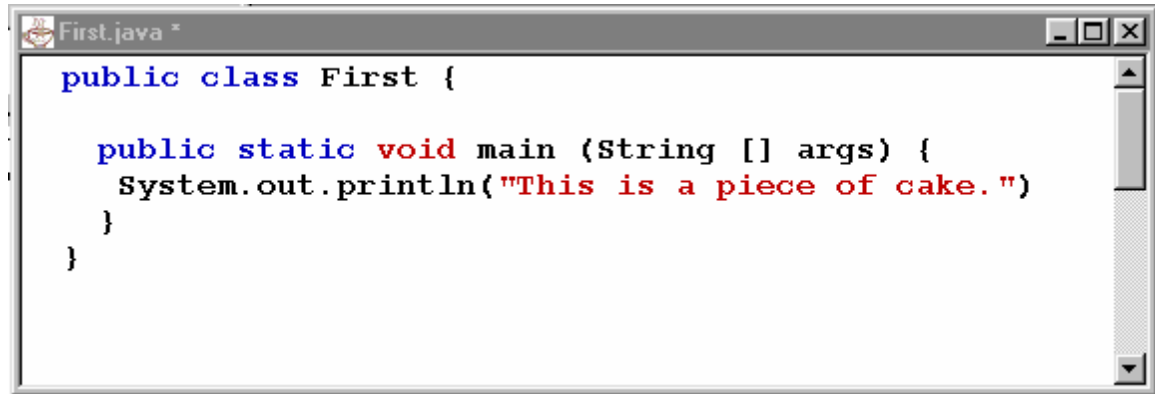
- To create a new Java file, go to the File Menu of the Menu Bar and select *New*. A text window with the tag *Untitled* appears.
- Place the cursor in the new window and type the code for your new class.
- Now we need to **save** this new file and **include** it in the project. Click on the File Menu on the Menu Bar. Select *Save As*. A dialog box appears.
- Navigate to the folder that was created for this project (if you are not already there).
- Type the name of the class that you have created in the *Name* box. This is the name of this new Java File.
- Click the *Save* box.
- When the file is saved, a dialog box appears asking if you would like the new Java file to be added to the project. Respond by clicking the **yes** button. The Project Window now includes the new Java file as below:



Section 6: Compiling in the Presence of Errors

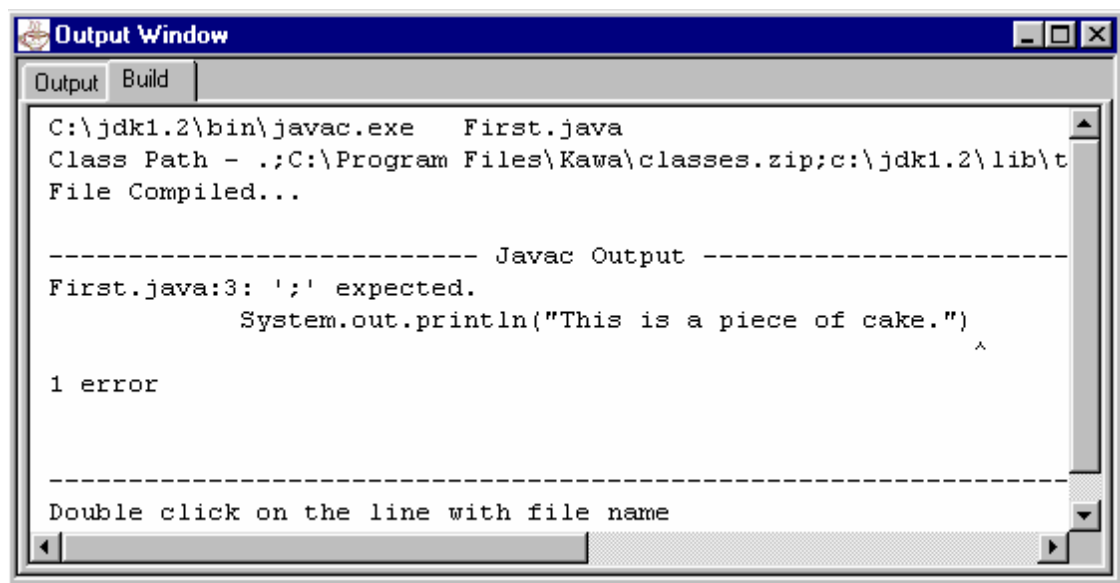
When errors are present in the code you have written, Kawa allows you to identify any syntax errors that you have made. For example, the following program is compiled without the required semi-colon at the end of the line:

```
System.out.println("This is a piece of cake.");
```



```
public class First {  
  
    public static void main (String [] args) {  
        System.out.println("This is a piece of cake.")  
    }  
}
```

When the program is compiled, we get the Output Window below.



```
Output Build  
C:\jdk1.2\bin\javac.exe First.java  
Class Path - .;C:\Program Files\Kawa\classes.zip;c:\jdk1.2\lib\...  
File Compiled...  
  
----- Javac Output -----  
First.java:3: ';' expected.  
        System.out.println("This is a piece of cake.")  
                                     ^  
  
1 error  
  
-----  
Double click on the line with file name
```

By double clicking on the line number in the Output Window, (i.e. clicking on First.java:3 above), a red arrow appears in the text window First.java at line 3. This arrow lies beside the offending line and allows us to easily identify the mistake.

The error is also given in the Output Window, giving some indication of the kind of mistake. For example, the text above tells us that a ';' is expected at the end of the line.