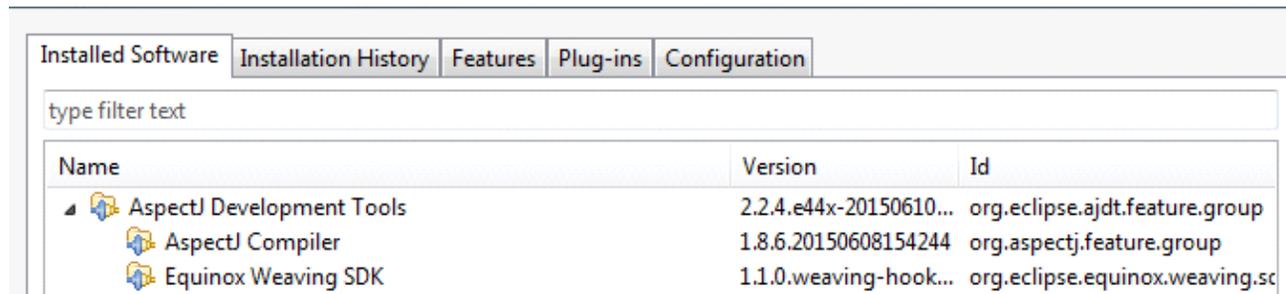


# AspectJ Tutorial

Thursday, March 28, 2019 10:10 AM

AspectJ Development Tools (AJDT) is the Eclipse plugin for AspectJ. Install it as per the instructions given at <https://www.eclipse.org/ajdt/>. Easiest way to install is through Help --> Install New Software menu on your eclipse IDE. After installation, you should see the AspectJ Compiler and Equinox Weaving SDK in the Installed Software tab.



We are now ready to write our first AspectJ code.

Create a new aspectj project in eclipse. Use File --> New --> Other --> AspectJ --> AspectJ Project.

Let us create a new class. Name it Student.java. Copy the following code into it.

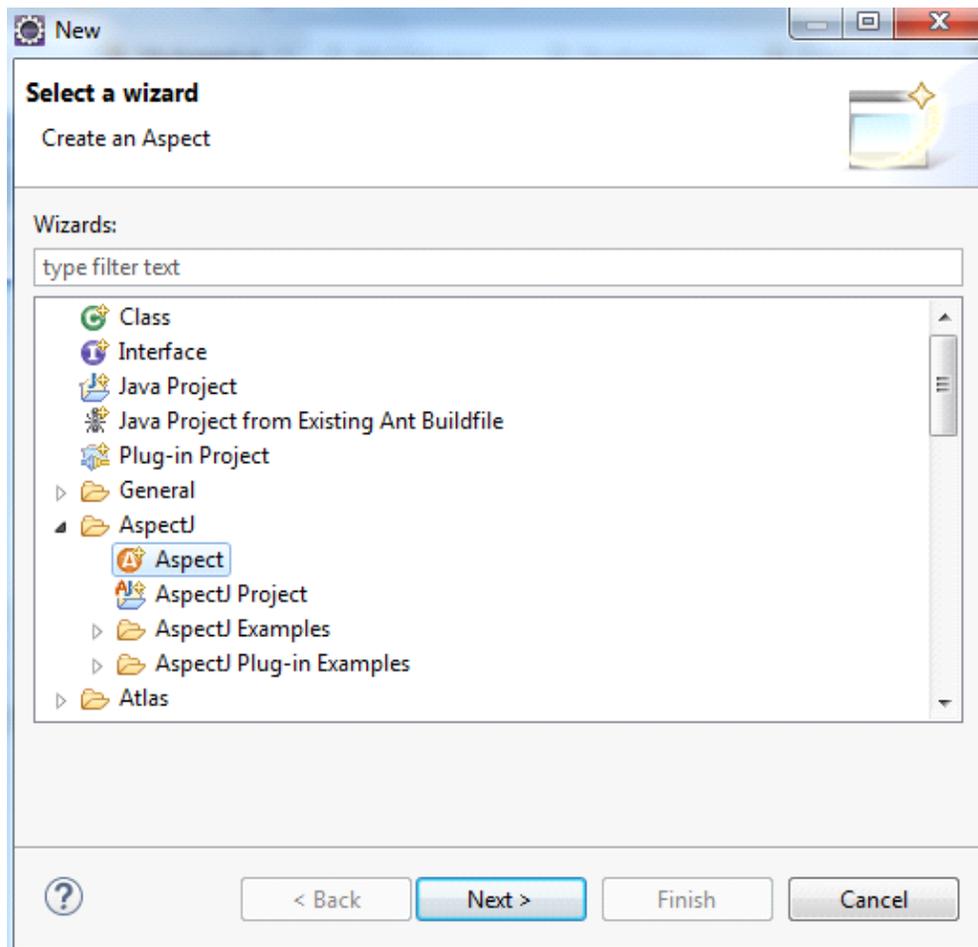
```
public class Student {
    void printName() {
        System.out.println("Venkatesh");
    }

    void printGrade() {
        System.out.println("9.0");
    }

    public static void main(String[] args) {
        Student student = new Student();
        student.printName();
        student.printGrade();
    }
}
```

This code creates an instance of Student, and prints the name and grade.

Let us now create a new aspect. Use the File --> New --> Other --> AspectJ --> Aspect option. Name it as Grade.



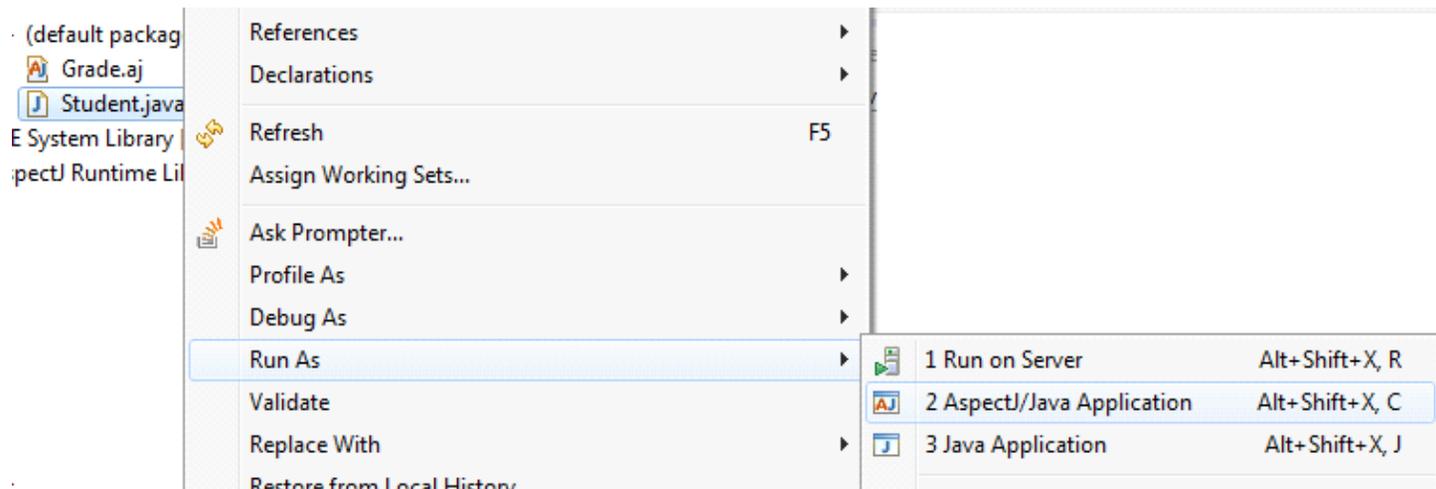
Eclipse creates a new aspect template and opens it in the editor.

We shall now create a pointcut. There are several join points where we could consider interleaving code. The pointcut selects specific join points. In this example, we will pick before and after call to the `printGrade` method. Write the following code:

```
public aspect Grade {
    pointcut gradeCall() :
        call(void printGrade());
    before(): gradeCall() {
        System.out.println("A call to printGrade has been invoked."); }
    after(): gradeCall() {
        System.out.println("A call to printGrade finished executing."); }
}
```

We have added one advice to print a text when the pointcut is invoked. Our code will print "A call to `printGrade` has been invoked." before every call to the `printGrade` method. Similarly, we have also added an advice to print something after the call to `printGrade()`.

Let us run the code as an AspectJ Application and see the results.



You should see the following output:

```
Venkatesh
A call to printGrade has been invoked.
9.0
A call to printGrade finished executing.
```

You may read more about pointcuts and advices here:  
<https://www.eclipse.org/aspectj/doc/next/progguide/starting.html>.

We shall now extend our tutorial to profile a method. We are interesting in knowing the time taken to execute the printName method.

Copy the following code as Profiling.aj to your project.  
import java.util.Date;

```
public aspect Profiling {
    long startTime;
    pointcut profiler() :
        call(void printName());
    before(): profiler() {
        startTime = System.nanoTime();
    }
    after(): profiler() {
        long endTime = System.nanoTime();
        System.out.println("Took " + (endTime - startTime) + " nano seconds.");
    }
}
```

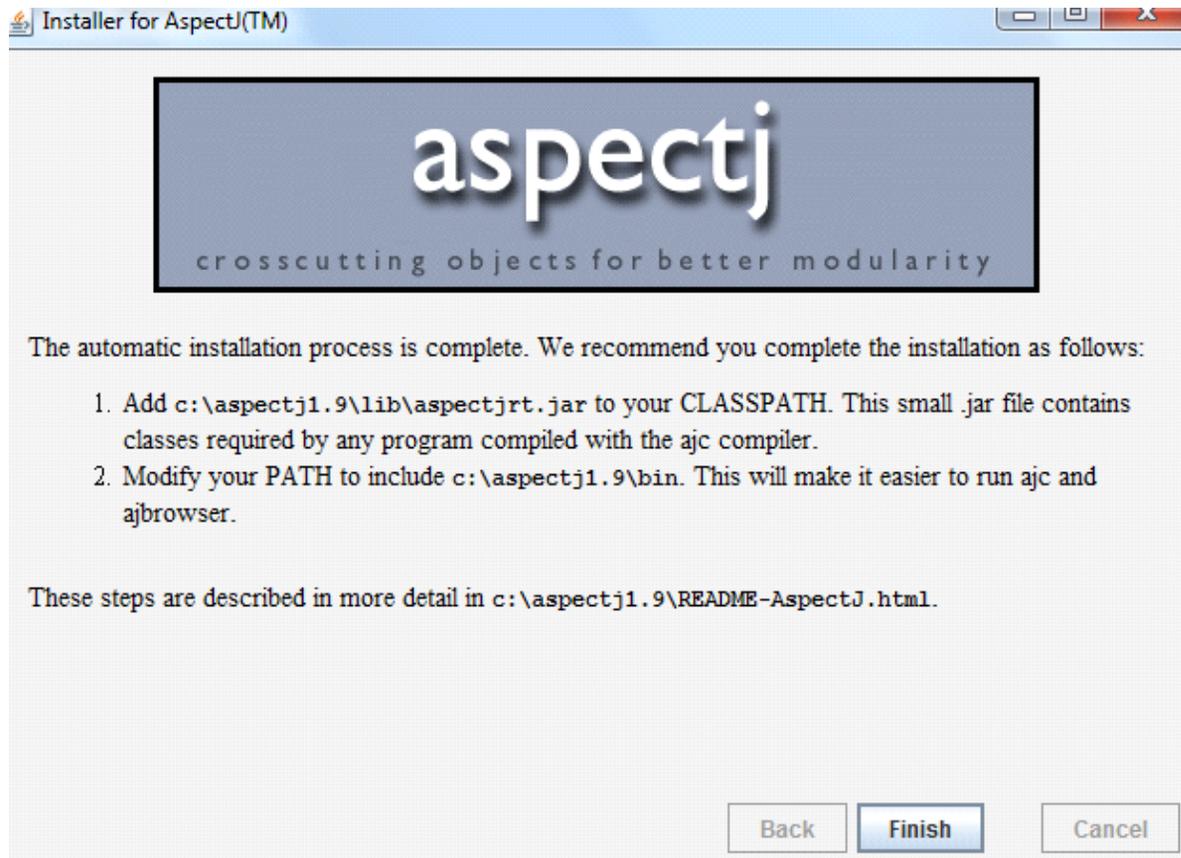
We use the same idea of defining a pointcut for the method of interest. Further, we use the before and after advice to compute the time taken by a method.

If both .aj files are available, the aspectj compiler will weave both aspects into the target code. So, if you run the program Student.java as an AspectJ application, you should see the following output:

```
Venkatesh
Took 168550 nano seconds.
A call to printGrade has been invoked.
9.0
A call to printGrade finished executing.
```

## For non-Eclipse users

Download and install AspectJ. Do not forget to follow the instructions to set the classpath and path variables.



Ensure JAVA\_HOME is pointing to the right place.

To run the examples, your classpath must include the AspectJ run-time Java archive (aspectjrt.jar). The following screenshot shows you few examples of using the command-line version of aspectj.

```
C:\code\aspectj>dir
Volume in drive C is Windows7_OS
Volume Serial Number is 5C03-BF0C

Directory of C:\code\aspectj

02/27/2019  06:55 PM    <DIR>          .
02/27/2019  06:55 PM    <DIR>          ..
02/27/2019  06:33 PM           17,835,991  aspectj-1.9.2.jar
02/27/2019  06:34 PM              347  MyClass.java
02/27/2019  06:36 PM              328  Profiling.aj
                3 File(s)      17,836,666 bytes
                2 Dir(s)      8,006,307,840 bytes free

C:\code\aspectj>ajc Profiling.aj MyClass.java

C:\code\aspectj>dir
Volume in drive C is Windows7_OS
Volume Serial Number is 5C03-BF0C

Directory of C:\code\aspectj

02/27/2019  06:55 PM    <DIR>          .
02/27/2019  06:55 PM    <DIR>          ..
02/27/2019  06:33 PM           17,835,991  aspectj-1.9.2.jar
02/27/2019  06:55 PM           1,587  MyClass.class
02/27/2019  06:34 PM              347  MyClass.java
02/27/2019  06:36 PM              328  Profiling.aj
02/27/2019  06:55 PM           2,980  Profiling.class
                5 File(s)      17,841,223 bytes
```

```
02/27/2019 06:55 PM          1,587 MyClass.class
02/27/2019 06:34 PM           347 MyClass.java
02/27/2019 06:36 PM           328 Profiling.aj
02/27/2019 06:55 PM          2,980 Profiling.class
          5 File(s)      17,841,233 bytes
          2 Dir(s)      8,006,234,112 bytes free
```

```
C:\code\aspectj>java -cp .;C:\aspectj1.9\lib\aspectjrt.jar MyClass
Inside Main
Inside Print1
Inside Print1
Inside Print2
Took 260915 nano seconds.
```