twit88.com

New SMS Library at http://www.twit88.com/

Do a graceful shutdown of your Java Application when Ctr-C, Kill…

By admin on Sep 27, 2007 in Java, Programming

When I am developing Java application for backend processes to do ETL, I often need to ensure that resources are properly cleaned up when the application is shutdown. However, user behavior is quite unpredictable. In either Windows or Unix, process can be terminated by force by the user, by sending the kill signal.

In Java, in order to do graceful shutdown, I normally add a shutdown hook to the Java runtime, using the **Runtime.getRunTime().addShutdownHook** method.

As an example, I have an interface for my application which has the **start** and **shutdown** method

```
public interface IApp {

void start();

void shutDown();
}
```

I wrote a **ShutdownInterceptor** class which extends the **Thread** class.

```
public class ShutdownInterceptor extends Thread {

private IApp app;

public ShutdownInterceptor(IApp app) {
this.app = app;
}

public void run() {
     System.out.println("Call the shutdown routine");
     app.shutDown();
     }
}
```

My sample application will implement the **IApp** interface. Have a look at the **main** method.

```
public class SampleApp extends BaseApp implements IApp {

public void start() {
try {
     System.out.println("Sleeping for 5 seconds before shutting
     down");
     Thread.sleep(5000);
} catch (Exception e) {
     System.out.println(e.getMessage());
}
}


public void shutDown() {
// Do a graceful shutdown here
     System.out.println("Shutdown is called");
}

public static void main(String args[]) {
     IApp app = new SampleApp();
     ShutdownInterceptor shutdownInterceptor = new
     ShutdownInterceptor(app);
     Runtime.getRuntime().addShutdownHook(shutdownInterceptor);
     app.start();
}
}
```

At startup, I created the **SampleApp** class, and then the **ShutdownInterceptor** class with the application instance. Then I add the interceptor to the Java runtime.

The application sleeps for 5 seconds. When it exits, **ShutdownInterceptor** will call application **shutdown** method automatically.