

# Calling Commandline Applications

## The Project Wonder way

Project Wonder, in the ERExtensions framework, have a utility class to help you run command-line applications, [ERXRuntimeUtilities](#). This utility class is used inside Wonder, for example ERAttachment use it to call ImageMagick when you want to resize images.

Check the Javadoc for the `execute` method to see examples.

## The Java way

Here, the "cat" and "outputCat" are debugging output categories:

```
Process process=null;
try {
    process = Runtime.getRuntime().exec(commandLine);
    OutputStream output = process.getOutputStream();
    output.write(inputString.getBytes());
    output.close();
    process.waitFor();
    DataInputStream dis=new DataInputStream(process.getInputStream());
    String output;
    do {
        output = dis.readLine();
        if (output != null)
            outputCat.debug(output);
    } while (output != null);
    dis.close();
} catch (IOException e) {
    cat.error("IOException: " + e.toString());
} catch (InterruptedException e) {
    cat.error("Interrupted process: " + e.toString());
} finally {
    if (process != null)
        process.destroy();
    outputCat.debug("Process finished.");
}
```

Note: A number of people have reported problems with `process.waitFor()` on Windows. The WebObjects Development List at Omnigroup has a number of people's workaround code for this problem.

**Note 2:** The procedure given here, of course, is to call anything executable from the command line from your Java program, not just Perl scripts.

## Mike Schrag

`process.waitFor()` is not just a problem on Windows. This code *will* cause problems. Process maintains buffers for stdout and stderr. If you do not consume those streams, you will run into deadlocks. The proper way to use `Runtime.exec` is to setup a thread for stderr and a thread for stdout and consume them yourself into your own buffer that does not have the same restrictions that the stock VM has.

There's a good example of this technique on [JavaWorld](#).