

# Development-Common Pitfalls and Troubleshooting

## Debugging Classpath Problems

### Chuck Hill

I recently found this useful and thought others might too:

```
System.out.println("\nClassPath Follows...\n");
ClassLoader classLoader = ClassLoader.getSystemClassLoader();
URL[] urls = ((URLClassLoader)classLoader).getURLs();
for(int i=0; i<urls.length; i++) {
    System.out.println(urls[i].toString() + "\n");
}
```

Add this to public static void main(String argv[]) before the call to WOApplication.main(argv, Application.class) and add these imports:

```
import java.io.*;
import java.net.*;
```

## Backtracked Too Far Errors

There are several things that can cause this:

- having a page cache size that is too small, set the cache size on Application
- the session missing on the URL. If WO creates a new session and then tries to use a component action URL you can get this error
- bad HTML on the page can cause a new session to get generated. If you are storing the session ID in cookies, this can replace the "real" session ID
- something (java script, bad HTML) creating a flurry of bad requests to the app and thus knocking the page out of the cache. You can override and add logging to savePage() in Session to check for this
- other things...

## Not Using PageWithName to Create Pages

Creating pages directly like this

```
public EditOrder editBooking() {
    EditOrder nextPage = new EditOrder(context()); // BAD BAD BAD
    return nextPage;
}
```

instead of,

```
public EditOrder editBooking() {  
    EditOrder nextPage = (EditOrder)pageWithName(EditOrder.class.getName());  
    return nextPage;  
}
```

can result in the page not getting saved in the page cache (and probably other bad things too) which will result in this back tracking error.