FWD - Bug #3002

static code analysis issues

02/24/2016 10:04 AM - Paul E

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Igor Skornyakov	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	Cleanup and Stablization for Server Features		
billable:	No	**case_num:**	
vendor_id:	GCD	**version:**	

Description

Related issues:

Related to Bugs - Bug #2483: Fix issues identified by static code analysis	**Closed**	**01/09/2015**

History

#1 - 02/24/2016 10:06 AM - Paul E

I've been looking at doing some static analysis as one way of trying to identify potential performance problems.
Whilst doing things of this nature, I ran a few checks on the P2J code and thought some of the results were worth reporting.

I'm using P2J rev 10971

On a quick review it's clear that not all of these will matter. Some (e.g. the resource leaks in AppServerHelper and FileStream) seem like they probably will matter.

```
Null pointer access: The variable att can only be null at this location    DirectoryCopy.java    /p2j/src/com/goldencode/p2j/directory    line 1216

Resource leak: 'fk' is not closed at this location    KeyImport.java    /p2j/src/com/goldencode/p2j/security    line 123
Resource leak: 'in' is never closed    AstGenerator.java    /p2j/src/com/goldencode/p2j/uast    line 1592
Resource leak: 'in' is never closed    ConversionDriver.java    /p2j/src/com/goldencode/p2j/convert    line 1750
Resource leak: 'in' is never closed    ReportWorker.java    /p2j/src/com/goldencode/p2j/pattern    line 2117
Resource leak: 'lowSocket' is not closed at this location    SessionManager.java    /p2j/src/com/goldencode/p2j/net    line 1017
Resource leak: 'raf' is never closed    FileStream.java    /p2j/src/com/goldencode/p2j/util    line 131
Resource leak: 'rs' is never closed    AppServerHelper.java    /p2j/src/com/goldencode/p2j/util    line 1301

The assignment to variable opts has no effect    E4GLPreprocessor.java    /p2j/src/com/goldencode/p2j/e4gl    line 307
```

The above errors, along with about 150 others that are more stylistic and less important, were found with my favoured default settings for eclipse Java compiler errors and warnings. I like having these checks in eclipse, as well as running headless static code checkers on commit, as the checks in eclipse provide the earliest possible feedback that I've made a mistake.

It may make sense for you guys to review eclipse preferences Java;Compiler;Errors/Warnings and make a few changes. Personally I like rules to either show up as errors or ignore - else you end up with 10,000 warnings that you don't really care about and never address.

#2 - 02/24/2016 11:21 AM - Paul E

Some further analysis with findbugs indicates a few more potentially noteworthy problems:

```
Return value of java.math.BigDecimal.setScale(int) ignored in com.goldencode.p2j.persist.TableResultSet$DataHa
ndler.invoke(Object, Method, Object[]) Line 622

Using pointer equality to compare a Object[] with a Object in com.goldencode.p2j.util.SharedVariableManager.up
dateExtentVar(BaseDataType[], BaseDataType[]) Line 1473

java.util.Map$Entry<com.goldencode.p2j.ui.WidgetConfig,com.goldencode.p2j.ui.WidgetConfig> is incompatible wit
h expected argument type WidgetConfig in com.goldencode.p2j.ui.ConfigSyncManager.markScopeEnd() Line 223
```

Again, it may make sense for you guys to review these results yourselves?

Findbugs documentation is annoying. The stackoverflow answer to this question tells you all you really need to know:
http://stackoverflow.com/questions/24157777/findbugs-command-line-how-to-specify-the-project-to-be-analyzed

Whilst it's ugly, when initially running these checks you'll probably want to use their UI. Mainly because the View menu provides some useful filters.

#3 - 02/24/2016 12:56 PM - Paul E

I've also ran Google's errorprone against P2J and the converted code: nothing interesting to report from this (very few failures).

Happily, eclipse compiler checks against the converted code are similarly dull.

Findbugs against the converted code is reasonably interesting. You'll need to give it a substantial heap (I threw 8GB at it - you might get away with less).

The main errors of interest are probably known to you already - lots of cases of non-static inner classes that could be made static. A stunning number of unread fields (i.e. dead code). Lots of switch statement fall throughs. If you like I'm happy to do some more detailed analysis and open redmine issues for failure categories of interest?

#4 - 02/24/2016 01:29 PM - Greg Shah

Thank you for running this and posting some results. We do intend to run some form of this in the future, but have not gotten to it yet.

> If you like I'm happy to do some more detailed analysis and open redmine issues for failure categories of interest?

That would be great! I guess it would be best to group related problems into a single task to reduce effort. It probably makes sense to make all of

these reported issues sub-tasks of this task. Use the "Add" link in the "Subtasks" section of the header above to do that.

#5 - 02/24/2016 01:57 PM - Eric Faulhaber

Paul wrote:

> Lots of switch statement fall throughs.

Ovidiu just fixed an issue with switch statement fall throughs in P2J rev. 10973, so some (hopefully all) of these should be addressed.

#6 - 03/11/2016 12:06 PM - Eric Faulhaber

- Target version set to Milestone 11

#7 - 03/18/2016 06:13 AM - Paul E

> Thank you for running this and posting some results. We do intend to run some form of this in the future, but have not gotten to it yet.

Given the difficulty of achieving good test coverage for P2J and the converted code, I think static code checking is of vital importance. It takes 10 minutes to run findbugs against the P2J jars and that time rewards you with a good list of problems, many of which will be the cause of application bugs. I've had a look at a few of these and I'm convinced that allocating someone to look at this for a couple of days will reward you richly.

#8 - 03/18/2016 06:14 AM - Paul E

As for findbugs against the converted code, this is much more time consuming and less rewarding (there are fewer scary-looking issues). I ran it overnight on the latest converted code last night (converted using P2J rev 10982) and I will list a few areas of most interest. The first of these is concerning and will likely be the cause of application bugs. The rest are less important in my opinion.

Unfortunately the findbugs XML output is pretty verbose and, even when tarred and gzipped, it is greater than the 5MB attachment limit so if you want me to share this I'll have to email it.

Switch statement problems (692 fall throughs, 2112 missing default cases), e.g:

```
Switch statement found in com.something.server.app.A1$23.body() where one case falls through to the next case
Switch statement found in com.something.server.app.A2$9.body() where default case is missing
```

Various GenericExpression inner classes could be static inner classes, e.g:

```
com.something.server.abc.Abc$HeaderExpr3
```

Some dead code could be removed, e.g:

```
Private method com.something.server.common.Abcfn0.newWhereClause(character, character, character, character) i
s never called
```

Lots of unread fields could be removed, e.g:

```
Unread field: com.something.server.app.Ghi.gcApiResult
```

There are some duplicated code branches and also some duplicated code in switch clauses, e.g:

```
com.something.server.common.Rdsf0$4.body() uses the same code for two branches
com.something.server.common.Gfh$37.body() uses the same code for two switch clauses
```

There are some useless control flows (e.g. empty if statements), e.g:

```
Useless control flow in com.something.server.app.Ignan0$9.body()
```

LE: GES modified this record to remove customer code references in the package and class names.

#9 - 03/23/2016 11:13 PM - Eric Faulhaber

- Status changed from New to WIP

- Assignee set to Igor Skornyakov

Igor, please create a 2483a task branch off the latest P2J trunk and investigate/address the P2J-specific items in notes 1 and 2 of this issue, some of which overlap with your findings from #2483. Since you don't have a converted server environment yet, please leave the converted code items for now.

#10 - 03/30/2016 06:36 PM - Igor Skornyakov

I have a converted server environment now, and can run FindBugs against it. Am I supposed to fix the conversion to eliminate the reposted issues in the converted code? This can be a time-consuming tasks simple because of the time the conversion takes.

#11 - 03/30/2016 06:41 PM - Eric Faulhaber

Please fix the items in notes 1 and 2 of this issue for now, which affect P2J code itself. We will come around for another look at the converted code issues later, after addressing some more pressing problems.

#12 - 03/30/2016 06:42 PM - Igor Skornyakov

Igor Skornyakov wrote:

> I have a converted server environment now, and can run FindBugs against it. Am I supposed to fix the conversion to eliminate the reposted issues in the converted code? This can be a time-consuming tasks simple because of the time the conversion takes.

I see. Thank you.

#13 - 03/31/2016 12:33 PM - Igor Skornyakov

The issues mentioned in notes 1 and 2 where reviewed and those one which are really potential bugs have been fixed.

Committed to the task branch 2483a revision 10991.

#14 - 07/22/2016 10:54 AM - Greg Shah

- % Done changed from 0 to 100

- Status changed from WIP to Closed

- Start date deleted (02/24/2016)

We expect future efforts to run static code analysis and review the results. Whomever does that should encode the results of this task to ensure that the things previously found as not issues won't be reported by future runs as issues.

#15 - 11/16/2016 12:06 PM - Greg Shah

- Target version changed from Milestone 11 to Cleanup and Stablization for Server Features