

## Conversion Tools - Feature #1565

test the full conversion process on Windows and document all problems that are found

09/20/2012 08:35 PM - Greg Shah

<b>Status:</b>	Closed	<b>Start date:</b>	09/20/2012
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Eugenie Lyzenko	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	Conversion Support for Server Features	<b>version:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

### History

#### #1 - 09/20/2012 08:39 PM - Greg Shah

- Assignee set to Eugenie Lyzenko

Run the conversion process for primes.p on Windows. Put detailed documentation into this issue's history for all problems encountered. If there are no obvious problems, make sure to compare the output of the Linux conversion run with the Windows conversion run to detect any output differences.

If the process aborts before getting to the end, you may look into potential solutions to get the process going all the way through. In this case, create a new task (in this project) for the abend/failure and work on it.

Even if the process gets all the way through, if there are deviations, you can create tasks for each one can work on a resolution.

#### #2 - 09/21/2012 08:00 AM - Eugenie Lyzenko

The question:

Is it important to have specific Windows version to test the conversion? I can run it in Windows XP, is it enough level of the Operating System?

#### #3 - 09/21/2012 09:36 AM - Eugenie Lyzenko

Another question:

Where the primes.p file is located?

EVL: I have found the file under timco/p2j/testcases/uast. Is this the file we need to convert in this task?

#### #4 - 09/21/2012 09:45 AM - Eugenie Lyzenko

The file has been found. Is it correct file?

#### #5 - 09/21/2012 10:02 AM - Greg Shah

Windows XP is fine.

Yes, p2j/testcases/uast/primes.p from CVS is the correct file.

#### #6 - 09/28/2012 10:43 AM - Eugenie Lyzenko

The java P2J classes are compiled without problems with ANT 170 and Java 160\_u35. To be able to start the conversion for primes.p the following changes should be made to uast/cfg/p2j.cfg.xml:

1. Change the path separator value to the ";" from ":"
2. Change every entry that has embedded path separator like "patpath"

Otherwise the required files will not be found on Windows file system.

Also it is required to set P2J\_HOME system variable at least to "." by javac -D option.

To be sure Linux compatibility it is available to set UTF-8 file encoding by setting java file.encoding property.

After these changes the conversion starts with the following command file:

```
SET prog=java -DP2J_HOME=. -Dfile.encoding="UTF-8"
```

```
SET cpath=-classpath ../../build/lib/p2j.jar.;
```

```
CMD.EXE /C prog cpath com.goldencode.p2j.convert.ConversionDriver -D2 F2+M0+CB %1
```

However the conversion fails on the stage "P2O Generation (database schema files)" with the problem:

#### #7 - 09/28/2012 10:54 AM - Eugenie Lyzenko

The conversion log file:

-----  
P2J Conversion Driver  
-----

-----  
SchemaLoader  
-----

Importing 'standard.df' for schema 'standard'...  
Persisted schema 'standard' to 'standard.dict'  
Importing 'p2j\_test.df' for schema 'p2j\_test'...  
Persisted schema 'p2j\_test' to 'p2j\_test.dict'

-----  
Scanning Progress Source (preprocessor, lexer, parser, persist ASTs)  
-----

primes.p  
-----

Post-Parse Fixups  
-----

.\primes.p  
Elapsed job time: 00:00:03.265  
-----

Schema Fixups (data dictionary)  
-----

.\data\namespace\p2j\_test.dict  
Elapsed job time: 00:00:00.516  
-----

Schema Fixups (Progress source file schemas)  
-----

.\primes.p.dict

Elapsed job time: 00:00:00.641

---

Schema Annotations (scan Progress source code)

---

.\primes.p  
ALL NATURAL JOINS:

Elapsed job time: 00:00:00.157

---

P2O Generation (database schema files)

---

EXPRESSION EXECUTION ERROR:

-----  
pkgPath = sprintf('%s%s',  
package.replaceAll("\.", fileSep),  
fileSep)

-----  
ERROR:  
java.lang.RuntimeException: ERROR! Active Rule:

-----  
RULE REPORT

-----  
Rule Type : INIT  
Source AST: null  
Copy AST : null  
Condition : pkgPath = sprintf('%s%s',  
package.replaceAll("\.", fileSep),  
fileSep)  
Loop : false  
--- END RULE REPORT ---

```
at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:718)
    at com.goldencode.p2j.convert.ConversionDriver.processTrees (ConversionDriver.java:892)
    at com.goldencode.p2j.convert.ConversionDriver.middle (ConversionDriver.java:593)
    at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1655)
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @2:36
    at com.goldencode.expr.Expression.execute (Expression.java:430)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:401)
    at com.goldencode.p2j.pattern.PatternEngine.applyGlobal (PatternEngine.java:1153)
    at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:674)
    ... 3 more
Caused by: java.lang.StringIndexOutOfBoundsException: String index out of range: 1
    at java.lang.String.charAt (Unknown Source)
    at java.util.regex.Matcher.appendReplacement (Unknown Source)
    at java.util.regex.Matcher.replaceAll (Unknown Source)
    at java.lang.String.replaceAll (Unknown Source)
    at com.goldencode.expr.CE264.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:336)
    ... 6 more
```

#### #8 - 09/28/2012 11:43 AM - Eugenie Lyzenko

The conversion from \*.df to \*.dict is performing without the problems. The results are the same for Linux and Win32 except the file separator chars.

#### #9 - 09/28/2012 11:46 AM - Constantin Asofiei

For some reason, on a Windows machine, any escaped value in a regexp needs two backslashes, as in:  
`package.replaceAll("\\.", fileSep)`

There are not many cases with explicit backslash in a regexp, but there are others, like `common-progress.rules:729` :  
`<rule>rel = rel.replaceAll(fileSep, ".")</rule>`

which will fail if fileSep is backslash.

More investigation is need to find the not-so-obvious cases where the regexp in a `replaceAll` contains a single backslash.

#### #10 - 09/28/2012 11:50 AM - Eugenie Lyzenko

The schema conversion has the difference:  
Win32 version does not have any `peerid` entries like below:  
`<annotation datatype="java.lang.Long" key="peerid" value="30064771073"/>`

May be we have the `java.lang.Long` issue in Java on Windows XP 32-bit?

#### #11 - 09/28/2012 11:55 AM - Eugenie Lyzenko

The resulting files:

`primes.p.cache`, `primes.p.dict`, `primes.p.lexer`, `primes.p.parser`

are the same for both Linux and Win32 platforms

#### #12 - 09/28/2012 11:58 AM - Eugenie Lyzenko

The resulting files `primes.p.schema` have difference:

The Windows version does not have the entries like this:  
`<annotation datatype="java.lang.Long" key="peerid" value="34359738369"/>`

But Win32 has another Long type related entry, so looks like this is not handling of Long type on Windows 32

#### #13 - 09/28/2012 12:12 PM - Eugenie Lyzenko

- File `primes.p.ast.linux` added

- File `primes.p.ast.win32` added

The main difference at conversion stage is the `primes.p.ast` files. For example the following lines are missed in Win32 version. The full list of differences can be found by comparing the two files attached here.

-----  
`<annotation datatype="java.lang.String" key="package-base" value=""/>`

```

<annotation datatype="java.lang.String" key="pkgname" value="com.goldencode.testcases"/>
<annotation datatype="java.lang.String" key="basename" value="primes.p"/>
<annotation datatype="java.lang.String" key="classname" value="Primes"/>
<annotation datatype="java.lang.String" key="relative-name" value="primes.p"/>
<annotation datatype="java.lang.Long" key="numundolocal" value="3"/>
<annotation datatype="java.lang.Long" key="numundoglobal" value="0"/>
<annotation datatype="java.lang.Long" key="numparms" value="0"/>
<annotation datatype="java.lang.Boolean" key="new_shared_scope" value="false"/>
<annotation datatype="java.lang.Boolean" key="recordscoping" value="true"/>
<annotation datatype="java.lang.Boolean" key="loop" value="false"/>
<annotation datatype="java.lang.Boolean" key="loopnext" value="false"/>
<annotation datatype="java.lang.Boolean" key="toplevel" value="true"/>
<annotation datatype="java.lang.Long" key="translevel" value="133"/>
<annotation datatype="java.lang.Long" key="reasons" value="0"/>
<annotation datatype="java.lang.String" key="javaname" value="Primes"/>
<annotation datatype="java.lang.Long" key="peerid" value="150323855383"/>
<annotation datatype="java.lang.Long" key="controlid" value="150323855381"/>
<ast col="0" id="12884902367" line="0" text="list_list" type="BUFFER_SCOPE">
<annotation datatype="java.lang.String" key="bufname" value="list"/>
<annotation datatype="java.lang.String" key="schemaname" value="list"/>
<annotation datatype="java.lang.String" key="dbname" value=""/>
<annotation datatype="java.lang.Long" key="scopetype" value="23"/>
<annotation datatype="java.lang.Boolean" key="implicit" value="false"/>
<annotation datatype="java.lang.String" key="javaname" value="list"/>
<annotation datatype="java.lang.Long" key="recordtype" value="14"/>
<annotation datatype="java.lang.Boolean" key="first" value="true"/>
<annotation datatype="java.lang.String" key="classname" value="TempRecord1"/>
<annotation datatype="java.lang.String" key="pkgname" value="com.goldencode.testcases.dmo._temp"/>
<annotation datatype="java.lang.Long" key="peerid" value="150323855386"/>
</ast>

```

-----

It is not clear if the Linux file separator("/") hardly encoded in P2J java sources(for example ConversionDriver.java) affects the conversion because in Windows the symbol is different("\"). For now attempts to experiment with replacement to System.getProperty("file.separator") does not affect the conversion results.

#### #14 - 09/28/2012 01:41 PM - Eugenie Lyzenko

The backslash issue is the real problem. Now we have stopped in the other place on the generating DDL stage:  
 28.09.2012 20:55:42 org.hibernate.cfg.Environment <clinit>  
 INFO: Hibernate 3.0.5

```
28.09.2012 20:55:42 org.hibernate.cfg.Environment <clinit>
INFO: hibernate.properties not found
28.09.2012 20:55:42 org.hibernate.cfg.Environment <clinit>
INFO: using CGLIB reflection optimizer
28.09.2012 20:55:42 org.hibernate.cfg.Environment <clinit>
INFO: using JDK 1.4 java.sql.Timestamp handling
28.09.2012 20:55:42 org.hibernate.dialect.Dialect <init>
INFO: Using dialect: com.goldencode.p2j.persist.dialect.P2JH2Dialect
\data\namespace\p2j_test.p2o
EXPRESSION EXECUTION ERROR:
```

```
-----
pkgPath = pkgPath.replaceAll(fileSep, ".")
^ { Unexpected internal error near index 1
\
^ }
```

```
-----
ERROR:
java.lang.RuntimeException: ERROR! Active Rule:
```

#### RULE REPORT

```
-----
Rule Type : INIT
Source AST: [ p2j_test ] DATA_MODEL/ @0:0 {30064771073}
Copy AST : [ p2j_test ] DATA_MODEL/ @0:0 {30064771073}
Condition : pkgPath = pkgPath.replaceAll(fileSep, ".")
Loop      : false
--- END RULE REPORT ---
```

```
at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:718)
    at com.goldencode.p2j.convert.ConversionDriver.processTrees (ConversionDriver.java:937)
    at com.goldencode.p2j.convert.ConversionDriver.generateDDL (ConversionDriver.java:701)
    at com.goldencode.p2j.convert.ConversionDriver.middle (ConversionDriver.java:654)
    at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1655)
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:19
    at com.goldencode.expr.Expression.execute (Expression.java:430)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:401)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:488)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:50)
    at com.goldencode.p2j.pattern.PatternEngine.apply (PatternEngine.java:1084)
    at com.goldencode.p2j.pattern.PatternEngine.processAst (PatternEngine.java:1003)
    at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:690)
    ... 4 more
Caused by: java.util.regex.PatternSyntaxException: Unexpected internal error near index 1
\
^
    at java.util.regex.Pattern.error (Unknown Source)
    at java.util.regex.Pattern.compile (Unknown Source)
    at java.util.regex.Pattern.<init> (Unknown Source)
    at java.util.regex.Pattern.compile (Unknown Source)
    at java.lang.String.replaceAll (Unknown Source)
    at com.goldencode.expr.CE1942.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:336)
    ... 10 more
```

The rule is schema/generate\_ddl.xml

#### #15 - 09/28/2012 02:18 PM - Constantin Asofiei

I think real problem here is how windows JRE handles regexps... especially cases when special regexp characters need to be escaped (like how the dot char is). IMO the fact that the windows file sep is the same as the regexp escape char is just a coincidence.

#### #16 - 09/28/2012 02:37 PM - Eugenie Lyzenko

There is still issue with p2o generation I guess. Consider the following:

In Linux we have the following line:

```
<annotation datatype="java.lang.String" key="package" value="com.goldencode.testcases.dmo"/>
```

```
<annotation datatype="java.lang.String" key="path" value="com/goldencode/testcases/dmo"/>
```

While in Win32:

```
<annotation datatype="java.lang.String" key="package" value="com.goldencode.testcases.dmo"/>
```

```
<annotation datatype="java.lang.String" key="path" value="com.goldencode.testcases.dmo\"/>
```

What is the expected line in Win32? The following:

```
<annotation datatype="java.lang.String" key="path" value="com\goldencode\testcases\dmo\"/>
```

The rule to make path from package is(Linux):

```
<rule>package = sprintf('%s.dmo', getConfigParameter('pkgroot'))</rule>
```

```
<rule>
```

```
pkgPath = sprintf('%s%s',
```

```
package.replaceAll('\.', fileSep),
```

```
fileSep)
```

```
</rule>
```

Windows:

```
<rule>
```

```
pkgPath = sprintf('%s%s',
```

```
package.replaceAll('\.', fileSep),
```

```
fileSep)
```

```
</rule>
```

And because the "." is not replaces with "\" in resulting p2o files can we tell this rule does not work?

#### #17 - 09/28/2012 02:55 PM - Constantin Asofiei

Eugenie, a question. Are you sure that the following:

```
<annotation datatype="java.lang.String" key="path" value="com.goldencode.testcases.dmo\"/>
```

was generated with this rule on windows?

```
<rule>
```

```
pkgPath = sprintf('%s%s', package.replaceAll('\\.', fileSep), fileSep)
```

```
</rule>
```

If escaped properly, the dot should have been replaced with the backslash. But maybe this is not the rule which generates that path.

If conversion abends on some `pkgPath = pkgPath.replaceAll(fileSep, ".")` expression (where first param - the regexp - is `fileSep`), when you run conversion on windows just replace `fileSep` with `fileSep.concat(fileSep)`, as in `pkgPath = pkgPath.replaceAll(fileSep.concat(fileSep), ".")`. You should be able to convert `primes.p` properly on Windows with these changes.

Idea is, the real problem is the first parameter of the `replaceAll` call. This is a regexp, in which the dot (and others) are special chars which need to be escaped, to be able to use the actual char. We need to ensure that the special chars in the first parameter are always escaped properly on windows and linux... but I can't tell you at this time what is the proper way to do this. I think we can choose from:

1. if we need to replace just one char with another char (and we are sure of it), use `String.replace(char old, char new)` instead of `String.replaceAll(String regexp, String repl)`. But this doesn't solve cases which may actually need of `String.replaceAll`
2. build our own `replaceAll` API in `StringHelper` or somewhere else, which will double each backslash in the regexp if the OS is Windows

#### #18 - 09/28/2012 04:35 PM - Eugenie Lyzenko

Eugenie, a question. Are you sure that the following:

```
<annotation datatype="java.lang.String" key="path" value="com.goldencode.testcases.dmo\"/>
```

was generated with this rule on windows?

```
<rule>
pkgPath = sprintf("%s%s", package.replaceAll("\\.", fileSep), fileSep)
</rule>
```

If escaped properly, the dot should have been replaced with the backslash.

Yes, this is the rule from `p2o.xml`. Having this rule in Linux style does not give the `p2o` files. But fixing it to `\\` gives the `com.goldencode.testcases.dmo\` entries. And moreover this creates such directories (`com.goldencode.testcases.dmo`) instead of the correct `com\goldencode\testcases\dmo` on filesystem. What about the `\\\\` alternative?

#### #19 - 09/30/2012 03:01 PM - Eugenie Lyzenko

Now we have the `primes.p` converted. The main issue is the Windows file separator char: `"\"` and some calls that uses this char as special one. The call `replaceAll("\\", "...)` does not work at all even in case `replaceAll("\\\\", "...)` as recommended in Windows. The result is the path names is not transforming correctly like `com.goldencode.p2j...` The only way it to replace `replaceAll()` with another call - `replace(char, char)`. However the `split("\\")` works fine. If the file separator is used as variable in: `replaceAll(fileSep, ...)` it should be doubled to work: `replaceAll(fileSep.concat(fileSep), ...)`

Files to be modified to have the `primes.p` converted in the rules directory:

```
annotations\annotations.xml:
Line 494:
outfile = sprintf("%s%s%s%s%s",
outroot,
fileSep,
pkgroot.replaceAll("\\.", fileSep),
fileSep,
```



```
xmlfile)
replaced with:
outfile = sprintf("%s%s%s%s%s",
outroot,
fileSep,
pkgroot.replace('.', '\'),
fileSep,
xmlfile)
```

callgraph\dynamic\_code\_generation.xml

Line 57:

```
<rule>type == string and parent != arguments and matches(text, "\"def.*")
```

replace with:

```
<rule>type == string and parent != arguments and matches(text, "\"def.*")
```

convert\brew.xml

Line 156

```
<action>pkgname = pkgname.replaceAll("\.", fileSep)</action>
```

replace with:

```
<action>pkgname = pkgname.replace('.', '\')</action>
```

dbref\collect\_refs.xml

Line 91

```
res = createListFromArray(tbl.toLowerCase().split("\."))
```

replace with:

```
res = createListFromArray(tbl.toLowerCase().split("\."))
```

Line 127

```
res = createListFromArray(fld.toLowerCase().split("\."))
```

replace with:

```
res = createListFromArray(fld.toLowerCase().split("\."))
```

Line 174

```
res = createListFromArray(var.toLowerCase().split("\."))
```

replace with:

```
res = createListFromArray(var.toLowerCase().split("\."))
```

Line 207

```
res = createListFromArray(tabName.toLowerCase().split("\."))
```

replace with:

```
res = createListFromArray(tabName.toLowerCase().split("\."))
```

include\common-progress.rules

Line 738

```
<rule>rel = rel.replaceAll(fileSep, ".")</rule>
```

replace with:

```
<rule>rel = rel.replaceAll(fileSep.concat(fileSep), ".")</rule>
```

schema\p2o.xml

Line 218

```
<rule>
```

```
pkgPath = sprintf("%s%s",
```

```
package.replaceAll("\.", fileSep),
```

```
fileSep)
```

```
</rule>
```

replace with:

```
<rule>
```

```
pkgPath = package.replace('.', '\').concat(fileSep)
```

```
</rule>
```

schema\hibernate.xml

Line 647

```
packagePath = getNoteString('package').replaceAll("\.", fileSep)
```

replace with:

```
packagePath = getNoteString('package').replace('.', '\')
```

schema\generate\_ddl.xml

Line 111

```
hbmFile = mergeDMORoot.
```

```
concat(fileSep).
```

```
concat(dmoClass.replaceAll("\.", fileSep)).
```

```
concat(".hbm.xml")
```

replace with

```
hbmFile = mergeDMORoot.
```

```
concat(fileSep).
```

```
concat(dmoClass.replace('.', '\')).
```

```
concat(".hbm.xml")
```

Line 228

```

pkgPath = pkgPath.replaceAll(fileSep, ".")
replace with:
pkgPath = pkgPath.replaceAll(fileSep.concat(fileSep), ".")
Line 355
pkgPath = pkgPath.replaceAll(fileSep, ".")
replace with:
pkgPath = pkgPath.replaceAll(fileSep.concat(fileSep), ".")
Line 390
hbmFile = getConfigParameter("output-root").
concat(fileSep).
concat(dmoClass.replaceAll("\\.", fileSep)).
concat(".hbm.xml")
replace with:
hbmFile = getConfigParameter("output-root").
concat(fileSep).
concat(dmoClass.replace('.', '\\')).
concat(".hbm.xml")
Line 460
pkgPath = pkgPath.replaceAll(fileSep, ".")
replace with:
pkgPath = pkgPath.replaceAll(fileSep.concat(fileSep), ".")

It is not yet clear why we do not need to change these cases:
convert\\frame_generator.xml
Line 2620:
res = sprintf("%s\\n%s", res, progressToJavaString(ltext))

schema\\fixups.xml
Line 273:
<action>safeDescr = text.replaceAll("\\*", "&#47;&#42;")</action>
<action>safeDescr = safeDescr.replaceAll("\\*", "&#42;&#47;")</action>

```

## #20 - 10/03/2012 07:32 PM - Eugenie Lyzenko

Further investigations and testing shows the only real issue we have converting on Windows is the replaceAll call with Windows file separator as the part of the first parameter. We can substitute "\\" with fileSep in Windows and get the conversion. So the call replace('.', fileSep) become OS neutral and can work on both Windows and Linux systems(with properly configured config file). The calls matches() and split() can be left untouched as we have for Linux version. So the only we need to change for rules is:

1. Replace replaceAll("\\.", fileSep) with replace('.', fileSep)
2. Replace replaceAll(fileSep, ".") with replace(fileSep, '.')
3. Adjust p2j.cfg.xml file modifying patpath,path-separator,file-separator and propath

This allows to use the same rules directory contents for both Windows and Linux and to modify only one file(point 3 above) further.

The question:

Is it mandatory to use `replaceAll()` in rules where the expression pattern string has only one parameter? I guess in this case we need to change the call to `replace()` to avoid possible issues and OS dependency.

#### #21 - 10/04/2012 08:01 AM - Greg Shah

1. Replace `replaceAll("\.", fileSep)` with `replace('.', fileSep)`
2. Replace `replaceAll(fileSep, ".")` with `replace(fileSep, '.')`

I assume you are referring to `String.replace()` and `String.replaceAll()`. The above approach will work as long as `fileSep` is a single character in size.

Unless the rule-set has a local variable called `fileSep`, any reference would normally be resolved to `com.goldencode/p2j/pattern/FileOperationsWorker.FileOps.getFileSep()` which returns a `String` that is the `File.separator` for the current system ON WHICH the conversion is running.

It is possible for this to be more than 1 character, although for Linux and Windows it is not.

To make it safe in all possible conditions, we would need to write a helper method that edits the regular expression (1st parameter) to `replaceAll()` to be appropriate for the platform on which the conversion is running. This code would have to properly escape the `fileSep` `String` as needed to make it work. This could be added as `FileOperationsWorker.FileOps.getFileSepRegex()`. Then we could just change the `fileSep` references to `fileSepRegex`. Please look at `character.convertToRegex()` as an example. The version for `fileSepRegex` would be simpler and it would NOT have all the Progress stuff. But you can get the idea.

I think this would be a more appropriate solution.

Is it mandatory to use `replaceAll()` in rules where the expression pattern string has only one parameter?

I don't understand what you mean by "where the expression pattern string has only one parameter". Please provide some examples of what you are talking about.

#### #22 - 10/04/2012 11:45 AM - Eugenie Lyzenko

I think this would be a more appropriate solution.

OK. I'll look into this.

Is it mandatory to use `replaceAll()` in rules where the expression pattern string has only one parameter?

I don't understand what you mean by "where the expression pattern string has only one parameter". Please provide some examples of what you are talking about.

I mean exactly what you discussed in the first part of the message: when we use `replaceAll()` with only one char in first parameter, like `replaceAll("\\.", ...)` or `replaceAll(fileSep, ...)`. Anyway I've got the answer from your reply - we need to have the special helper to handle this. Correct?

Another point is the case when we use `String.replaceAll("\\.", someString)`. This is not file separator case (but looks like this). This call does not work in Windows with our xml rule files. As the result we do not have proper conversion from class name to path string.

#### #23 - 10/04/2012 12:01 PM - Greg Shah

After thinking more about this, I realize that my idea about adding a `getFileSepRegex()` is not a full solution. The problem is not just with `fileSep`. In fact, your example already shows that case [#1](#) would not work because the `fileSep` is the 2nd parameter in that example.

Constantin previously suggested this:

```
build our own replaceAll API in StringHelper or somewhere else, which will double each backslash in the regexp
if the OS is Windows
```

I think the solution is:

1. Add a static `StringHelper.fixupRegex()` which doubles the backslashes if the current platform is Windows is the correct approach.
2. Add a `CommonAstSupport.Library.fixupRegex()` which just calls the `StringHelper` version.
3. Everywhere we call `replaceAll()` where there are problem regular expressions as the first param, we must fixup the regex before passing it. For example:

1. Replace `replaceAll("\\.", fileSep)` with `replaceAll(fixupRegex("\\."), fileSep)`
2. Replace `replaceAll(fileSep, ".")` with `replaceAll(fixupRegex(fileSep), ".")`

See if this works.

**#24 - 10/04/2012 12:45 PM - Constantin Asofiei**

Greg,

I've done a little more testing in why it abends in the first place and it looks like calls like this:

```
System.out.println("com.goldencode.p2j.test".replaceAll("\\.", "\\\"));
```

fail with the same stacktrace as the replaceAll from TRPL rules.

For some reason, if we double the backslash in the first argument, as in:

```
System.out.println("com.goldencode.p2j.test".replaceAll("\\\\.", "\\\"));
```

it no longer abends, but it makes no replacements (the result is "com.goldencode.p2j.test").

Looking in Javadoc for replaceAll, I finally understood that the problem is not with the regex but with the second parameter, the replacement. If we do this instead:

```
System.out.println("com.goldencode.p2j.test".replaceAll("\\.", java.util.regex.Matcher.quoteReplacement("\\\""))
);
```

the result is OK, "com\goldencode\p2j\test".

What threw me off was the fact that doubling the backslash in the regex solved the abend... anyway, the correct solution is the above, we need to process the second parameter for String.replaceAll.

**#25 - 10/04/2012 12:48 PM - Greg Shah**

Please prepare the minimum set of changes needed to resolve all the known problems. Add a helper as needed to CommonAstSupport and fixup the rules.

**#26 - 10/04/2012 01:10 PM - Eugenie Lyzenko**

Constantin,

There are no such elements in our rule files as second parameter for replaceAll(): "\\" as the file separator. And we can not use it if we want to be OS independent. The following code does work without any problems on Windows:

```

public static void main(String[] args) {
String strToTest = "one.two.three.four";
String fileSep = System.getProperty("file.separator");

System.out.println("Initial string: "+strToTest+
    " File separator: "+fileSep);
String pathLike = strToTest.replaceAll("\\\\.",
    fileSep.concat(fileSep)).concat(fileSep);
System.out.println( "Full path simulation: "+pathLike );
strToTest = strToTest.replaceAll( "\\.", fileSep.concat(fileSep) );
System.out.println( "Replacing dots with file separator: "+strToTest );
strToTest = strToTest.replaceAll( fileSep.concat(fileSep), "." );
System.out.println( "Replacing file separator with dots: "+strToTest );

System.out.println( "Alternative replacement, fileSep as char: "+
    fileSep.charAt(0) );
String pathAlt = strToTest.replace('.', fileSep.charAt(0));
System.out.println( "Replacing dots with file separator: "+pathAlt );
}
Try and see.

```

Greg,

I'm preparing the fixes with fixupRegex() and test. Correct?

**#27 - 10/04/2012 01:25 PM - Greg Shah**

I'm preparing the fixes with fixupRegex() and test. Correct?

I thought the problem is with the replacement string, requiring use of the `Matcher.quoteReplacement()`? If so, then the added helper should probably be called `quoteReplacement()` and it can be placed in `CommonAstSupport.Library`.

Yes, go ahead with preparing and testing the fixes.

## #28 - 10/04/2012 09:42 PM - Eugenie Lyzenko

Preliminary results:

1. Substitution `replaceAll(fileSep, ".")` with `replaceAll(fixupRegex(fileSep), ".")` works fine.

2. Substitution `replaceAll("\\.", fileSep)` with `replaceAll(fixupRegex("\\."), fileSep)` does not work. We have neither exception nor expected result. The dots are not replacing with path separator. Moreover using `java.util.regex.Matcher.quoteReplacement()` for the second parameter does not help too. I've used the following combination:

- `replaceAll(fixupRegex("\\."), fileSep).`
- `replaceAll("\\.", fileSep)`
- `replaceAll("\\.", quoteReplacement(fileSep))`
- `replaceAll("\\\\\\\\.", fileSep)`
- `replaceAll("\\\\\\\\.", quoteReplacement(fileSep))` and even
- `replaceAll("\\.", "/"`
- `replaceAll("\\\\\\\\.", "/"`

All cases leave the dots untouched. Note this is the Windows result and this is not pure Java code. This is handling XML from Java code. XML and Java string constraints intersection? Continue investigation.

## #29 - 10/05/2012 06:26 AM - Constantin Asofiei

Eugenie, please try the following call in the conversion rules:

```
replaceAll("\\.", quoteReplacement(fileSep))
```

I suspect this should work.

Also, what does `fixupRegex` do? I think you can use `quoteReplacement` instead with the same result.

## #30 - 10/05/2012 10:21 AM - Eugenie Lyzenko

Constantin,

Yes, the call

```
replaceAll("\\.", quoteReplacement(fileSep))
```

does work. So I applied `fixupRegex()` and `quoteReplacement()` and get conversion OK in Windows. I think we need to use both:

1. `replaceAll("\\.", quoteReplacement(fileSep))`
2. `replaceAll(fixupRegex(fileSep), ".")`

The `StringHelper` calls(will work only in Windows):

```
// EVL***<<
```

```
public static String fixupRegex(String text) {  
    StringBuffer sb = new StringBuffer();
```

```
    if (System.getProperty("os.name").toLowerCase().indexOf("win") >= 0)  
        for (int i = 0; i < text.length(); i++)  
        {  
            sb.append(text.charAt(i));  
            if (text.charAt(i) == '\\')  
                sb.append(text.charAt(i));  
        }  
    else  
        sb.append(text);
```

```

return sb.toString();
}

public static String quoteReplacement(String text)
{
    StringBuffer sb = new StringBuffer();

    if (System.getProperty("os.name").toLowerCase().indexOf("win") >= 0)
        sb.append(java.util.regex.Matcher.quoteReplacement(text));
    else
        sb.append(text);

    return sb.toString();
}
// EVL***<<

```

### #31 - 10/05/2012 04:42 PM - Eugenie Lyzenko

Greg, Constantin,

I've performed a bit more testing and found we can use both `fixupRegex()` and `quoteReplacement()` for substitutions:

1. `replaceAll("\\.", fixupRegex(fileSep))` and
2. `replaceAll("\\.", quoteReplacement(fileSep))`

with the same result because in this particular case these calls do the same work and get the conversion for `primes.p` (for Windows and Linux). My idea is to use `fixupRegex()` because:

- this is our own API call and we have the full control of what it does
- `quoteReplacement()` can produce more effect than just doubling the backslashes. For example the string `"$"` is converting to `"\$"`.

So I'm offering to leave only `fixupRegex()` in `CommonAstSupport.Library` and use it in both substitutions:

1. `replaceAll("\\.", fixupRegex(fileSep))`
2. `replaceAll(fixupRegex(fileSep), ".")`

May be we need to change the name (to something like `fixupSpecialString`) of the method because in [#1](#) above `fileSep` is not a regular expression according to official Java documentation and I'm still confused why we need to double backslash here. What do you think about this offering?



**#32 - 10/08/2012 09:20 AM - Greg Shah**

Please do use the quoteReplacement() for all of the 2nd parameter cases. That is exactly what that code is meant to do and we get it "for free" with the J2SE platform. So they can handle all the cases and we don't have to maintain the code.

For all usage on the 1st parameter, please use our own fixupRegex(). We may find other needs for this over time and can tune it exactly to our own needs.

SO: please use both, but use them consistently, depending on which parameter is being processed.

**#33 - 10/08/2012 03:49 PM - Eugenie Lyzenko**

The changes are tested and prepared. Do we need them to be applied to the current P2J build? Note we still need to manually configure p2j.cfg.xml file when changing the Operating System to run conversion. Or we need to add the OS detection code to this file to be able to choose the correct set of initial parameters for conversion "on the fly".

**#34 - 10/08/2012 04:04 PM - Greg Shah**

Don't apply the changes to staging or CVS yet. Please do the following:

1. Send the changes to Constantin (and copy me). He will review the changes and let you know if any additional issues exist.
2. Please create 2 identical Majic conversion projects on your local Linux system (if you don't already have it). Run conversion with the current (unmodified version of P2J) in one of them. Run conversion with the modified rules (as the only difference on top of the current P2J) in the other. Then compare (using dirdiff or similar) the Java output for these two builds. There should be no difference.
3. Create a Majic conversion project on your Windows system. Use the current P2J + your changes to run conversion there. Zip up the results and copy them to your Linux box. Unzip them and compare them to the other results that were run there (using dirdiff or similar). Again: there should be no difference.

Resolve any problems you find from the above 3 items.

**#35 - 10/08/2012 05:41 PM - Eugenie Lyzenko**

OK.

The files to be changed to run conversion on Windows.

Rules:

p2j/rules/annotations/annotations.xml  
p2j/rules/convert/brew.xml  
p2j/rules/schema/p2o.xml  
p2j/rules/schema/hibernate.xml  
p2j/rules/schema/generate\_ddl.xml  
p2j/rules/include/common-progress.rules

Java source files:

p2j/src/com/goldencode/util/StringHelper.java  
p2j/src/com/goldencode/p2j/pattern/CommonAstSupport.java

Added command file:

p2j/testcases/uast/conv2.cmd

The project config changes:

<parameter name="patpath" value=".,\${P2J\_HOME}/pattern;\${P2J\_RULES}/include;\${P2J\_RULES};" />

```
<parameter name="path-separator" value=";" />
<parameter name="file-separator" value="\\" />
<parameter name="propath" value="." />
```

One note more. It is better to use NTFS file system to the project conversion to avoid the filename truncation. The FAT32 works as well with primes.p but there is potential risk of having file name issues in P2J project conversion.

#### #36 - 10/09/2012 05:57 AM - Constantin Asofiei

Eugenie,

Please address the following:

1. the p2j.cfg.xml you sent is for Windows OS. Please leave the p2j.cfg.xml as is, and:

- put the Windows configuration file in a p2j.cfg.xml.win
  - add a copy of p2j.cfg.xml to the update, named p2j.cfg.xml.linux
2. Change the copyright date in the header for all the files in the update, if needed.

3. I think the following String.replace\* calls need to be fixed too:

annotations/naming.rules

- line 91: <action>relfile = relfile.replaceAll(fileSep, projectsep)</action>

annotations/comments.rules

- line 837: content = content.replaceFirst(basename, classname)

schema/dmo\_common.rules

- line 150: concat(dmoClass.replaceAll("\.", fileSep))

#### #37 - 10/09/2012 10:06 AM - Eugenie Lyzenko

Constantin,

1. OK.

2. I do not understand. Why this special update need this modification? Do we need every file of the project be updated in copyright area?

3. Consider the case convert/comments.rules:

content = content.replaceFirst(basename, classname)

What parameter you think we need to fix? First, second or both? And we do not have any info about replaceFirst() wrong behavior in Windows. Why do not we try to leave this case as is and see what will happen?

#### #38 - 10/09/2012 11:45 AM - Constantin Asofiei

Eugenie,

For 2 - as you change a file, if the current year is different than the one in the copyright area, you need to change that too.

i.e.

Copyright (c) 2009-2010, Golden Code Development Corporation.

now must be:

Copyright (c) 2009-2012, Golden Code Development Corporation.

For 3 - java.lang.String.replaceFirst has the same parameters as String.replaceAll, except that it replaces only the first occurrence. But, in our case, basename and classname do not contain the package or path, so it is safe to leave it as is.

**#39 - 10/10/2012 03:36 AM - Constantin Asofiei**

Eugenie,

Please remove the cfg/ folder from the root of your update. Except this, the update looks OK.

**#40 - 10/10/2012 05:21 AM - Eugenie Lyzenko**

Constantin,

This was not mistake. I have added the config files for TIMCO conversion under Linux/Windows. This is just addition. Does not break the current main p2j.cfg.xml file. The previous config files are there too but they are applying to the uast testcases only.

**#41 - 10/10/2012 06:44 AM - Constantin Asofiei**

Understood. Then you need to put the cfg files for MAJIC in a separate update.

**#42 - 10/10/2012 05:16 PM - Greg Shah**

Please report on the status of the 2 testing activities:

1. Old P2J majic conversion on Linux vs new P2J majic conversion on Linux
2. Old P2J majic conversion on Linux vs new P2J majic conversion on Windows

(Old current CVS P2J and New current P2J + your changes)

I would also like to see the 3 zip files of the timco/src/ directory where all the Java code exists so I can review the results myself.

**#43 - 10/10/2012 06:27 PM - Eugenie Lyzenko**

Greg,

OK. I'm running the conversions. The difference reports will be here. But what is the place to upload results? I guess it is too big to send via e-mail.

**#44 - 10/10/2012 06:49 PM - Greg Shah**

Since the comparisons are just about the differences in src/aero/timco/majic/, the amount of code is actually not so big that it can't be zipped up easily. Try that and see if it can be sent via email. If not, create a temp directory in shared/clients/timco/ and let me know where to look.

**#45 - 10/10/2012 07:06 PM - Eugenie Lyzenko**

OK.

BTW, I have faced the strange issue. I did run two conversions in Linux. The both was OK. Then I have compared the results. The generated java sources were identical. But some \*.lexer files and id="digit" in \*.jast values was not. So I have restarted the conversion again in Old P2J majic project and it fails on java sources compilation stage. So I have restarted it again to have recreateable results. Is it normal the id value has different digital value from run to run?

**#46 - 10/10/2012 07:39 PM - Greg Shah**

But some \*.lexer files and id="digit" in \*.jast values was not.

Please show examples of both.

I am worried about the lexer file differences. It is possible that the difference is in how the preprocessor leaves things, but I need to see details.

The id= differences are probably not a problem.

So I have restarted the conversion again in Old P2J majic project and it fails on java sources compilation stage.

Hmmm. That is not good. How are you running conversion (what command line)?

Is it normal the id value has different digital value from run to run?

It used to be. I don't remember if we ever changed that.

**#47 - 10/11/2012 09:02 AM - Eugenie Lyzenko**

Please show examples of both.

Here is the src/syman/adp/adpout0.p.lexer

The Old MAJIC conversion with "classic" rules set:

```
[01382:046] <PLUS>          +
[01382:048] <KW_CHR>         chr
[01382:051] <LPARENS>       (
[01382:052] <NUM_LITERAL>   10
[01382:054] <RPARENS>      )
[01382:055] <DOT>           .
[01384:003] <KW_END>        end
[01384:006] <DOT>           .
[01386:003] <KW_OUTPUT>
```

The New MAJIC conversion with modified rules set:

```
[01382:046] <PLUS>          +
[01382:048] <KW_CHR>         chr
[01382:051] <LPARENS>       (
[01382:052] <NUM_LITERAL>   10
```

```

[01382:054] <RPARENS>      )
[01382:055] <DOT>          .
[01384:003] <KW_END>        end
[01384:006] <DOT>          .
[01386:003] <KW_OUTPUT>     output
[01386:010] <KW_STREAM>    stream
[01386:017] <SYMBOL>       adpout
[01386:024] <KW_CLOSE>     close
[01386:029] <DOT>          .
[01387:003] <KW_OUTPUT>     output
[01387:010] <KW_STREAM>    stream
[01387:017] <SYMBOL>       adpform
[01387:025] <KW_CLOSE>     close
[01387:030] <DOT>          .
[01390:003] <KW_IF>         if
[01390:006] <SYMBOL>       got-data
[01390:015] <EQUALS>        =
[01390:017] <BOOL_TRUE>     yes
[01391:003] <KW_THEN>       then
[01391:008] <KW_DO>         do
[01391:010] <COLON>         :
[01392:005] <KW_ASSIGN>     assign
[01392:012] <SYMBOL>       hardcopy
[01392:021] <EQUALS>        =
[01392:023] <STRING>       "n"
[01392:026] <DOT>          .
[01393:005] <KW_MSG>        message
[01393:013] <STRING>       " Print ADP Checklist ? <y/n>: "
[01394:005] <KW_UPDATE>    update
[01394:012] <SYMBOL>       hardcopy
[01394:021] <KW_AUTO_RET>  auto-ret
[01394:029] <DOT>          .
[01395:005] <KW_IF>         if
[01395:008] <SYMBOL>       hardcopy
[01395:017] <EQUALS>        =
[01395:019] <STRING>       "y"
[01396:005] <KW_THEN>       then
[01396:010] <KW_DO>         do
[01396:012] <COLON>         :
[01397:007] <SYMBOL>       outfile
[01397:015] <EQUALS>        =
[01397:017] <STRING>       "lp -s -dhpjet -ol"
[01397:036] <DOT>          .
[01398:007] <KW_UPDATE>    update
[01398:014] <SYMBOL>       outfile
[01398:022] <KW_LABEL>     label
[01398:028] <STRING>       "Printer"
[01399:014] <KW_WITH>      with
[01399:019] <KW_ROW>       row
[01399:023] <NUM_LITERAL>  3
[01399:025] <KW_CENTER>   centered
[01399:034] <KW_SIDE_L>   side-labels
[01399:046] <KW_FRAME>    frame
[01399:052] <SYMBOL>       f-printer
[01399:061] <DOT>          .
[01400:007] <KW_OUTPUT>    output
[01400:014] <KW_THROUGH>  thru
[01400:019] <KW_VALUE>    value
[01400:024] <LPARENS>      (
[01400:025] <SYMBOL>       outfile
[01400:032] <RPARENS>     )
[01400:033] <DOT>          .
[01401:007] <KW_RUN>       run
[01401:011] <FILENAME>    adp/adpchklis.p
[01401:025] <DOT>          .
[01402:007] <KW_ASSIGN>   assign
[01402:014] <SYMBOL>       hardcopy
[01402:023] <EQUALS>        =
[01402:025] <STRING>       "n"
[01402:028] <DOT>          .
[01403:007] <KW_OUTPUT>    output
[01403:014] <KW_CLOSE>    close
[01403:019] <DOT>          .
[01404:005] <KW_END>       end
[01404:008] <DOT>          .

```

```

[01406:005] <KW_MSG>          message
[01406:013] <STRING>          " Download "
[01406:026] <SYMBOL>          outfile
[01406:034] <STRING>          " file at this time ? <y/n>: "
[01407:005] <KW_UPDATE>       update
[01407:012] <SYMBOL>          hardcopy
[01407:021] <KW_AUTO_RET>     auto-ret
[01407:029] <DOT>             .
[01408:005] <KW_IF>           if
[01408:008] <SYMBOL>          hardcopy
[01408:017] <EQUALS>          =
[01408:019] <STRING>          "y"
[01409:005] <KW_THEN>         then
[01409:010] <KW_DO>           do
[01409:012] <COLON>           :
[01411:007] <KW_ASSIGN>       assign
[01411:014] <SYMBOL>          hardcopy
[01411:023] <EQUALS>          =
[01411:025] <STRING>          "n"
[01411:028] <DOT>             .
[01412:007] <KW_PAUSE>       pause
[01412:013] <NUM_LITERAL>     0
[01412:015] <KW_B4_HIDE>     before-hide
[01412:026] <DOT>             .
[01413:007] <KW_IF>           if
[01413:010] <KW_OPSYS>       opsys
[01413:016] <EQUALS>          =
[01413:018] <STRING>          "UNIX"
[01414:007] <KW_THEN>         then
[01415:009] <KW_UNIX>         unix
[01415:014] <KW_VALUE>       value
[01415:019] <LPARENS>        (
[01415:020] <SYMBOL>          t-lbin
[01415:027] <PLUS>           +
[01415:029] <STRING>          "/ksend"
[01415:037] <RPARENS>        )
[01415:038] <DOT>             .
[01416:007] <KW_PAUSE>       pause
[01416:013] <KW_B4_HIDE>     before-hide
[01416:024] <DOT>             .
[01417:005] <KW_END>          end
[01417:008] <DOT>             .
[01418:003] <KW_END>          end
[01418:006] <DOT>             .
[01420:003] <KW_HIDE>         hide
[01420:008] <KW_ALL>          all
[01420:012] <KW_NO_PAUSE>    no-pause
[01420:020] <DOT>             .
[01421:001] <KW_END>          end
[01421:004] <DOT>             .
[01422:001] <DOT>             .
[01422:002] <EOF>            null

```

The example of id mismatch.

src/syman/adp/ewrexcp.p.ast Line 2394,

Old MAJIC:

```

...
<annotation datatype="java.lang.String" key="uniquename" value="majic.crew_majic.crew"/>
<annotation datatype="java.lang.Long" key="refid" value="**25769807683**"/>
<annotation datatype="java.lang.Long" key="peerid" value="19026705121810"/>
...

```

New MAJIC:

```

...
<annotation datatype="java.lang.String" key="uniquename" value="majic.crew_majic.crew"/>
<annotation datatype="java.lang.Long" key="refid" value="**25769807683**"/>
<annotation datatype="java.lang.Long" key="peerid" value="19026705121810"/>
...

```

src/syman/adp/ewrexcp.p.jast Line 1177,

Old MAJIC:

```

...
<ast col="0" id="19026705121822" line="0" text="RecordBuffer.openScope" type="STATIC_METHOD_CALL">
<annotation datatype="java.lang.Long" key="peerid" value="**25769807683**"/>
<ast col="0" id="19026705121823" line="0" text="crew" type="REFERENCE"/>
</ast>

```

```
...
New MAJIC:
...
<ast col="0" id="19026705121822" line="0" text="RecordBuffer.openScope" type="STATIC_METHOD_CALL">
<annotation datatype="java.lang.Long" key="peerid" value="**25769807682**"/>
<ast col="0" id="19026705121823" line="0" text="crew" type="REFERENCE"/>
</ast>
...
```

How are you running conversion (what command line)?

Always ./run\_full.sh. And last conversion was OK. May be some machine issue. Do you use ECC memory modules in your conversion working machines?

Shortly I'll send the two src conversion directories for you to compare. And preparing to run conversion in Windows.

#### #48 - 10/11/2012 09:10 AM - Eugenie Lyzenko

One more note.

Since the comparisons are just about the differences in src/aero/timco/majic/

These directories are identical.

#### #49 - 10/11/2012 10:59 AM - Greg Shah

OK, these are some strange results.

Here is the src/syman/adp/adpout0.p.lexer

The Old MAJIC conversion with "classic" rules set:

```
[01382:046] <PLUS> +
[01382:048] <KW_CHR> chr
[01382:051] <LPARENS> (
[01382:052] <NUM_LITERAL> 10
[01382:054] <RPARENS> )
[01382:055] <DOT> .
[01384:003] <KW_END> end
[01384:006] <DOT> .
[01386:003] <KW_OUTPUT>
```

The .lexer file is truncated at [01386:003] and there is nothing after that?

If so, there is a problem with your environment. I have checked at file in other places and it is not normally truncated there.

The difference in peerids is also unexpected. They would be the same unless there is 1 extra node in the NEW jast. But there should not be any extra nodes at all.

#### #50 - 10/11/2012 11:15 AM - Eugenie Lyzenko

The .lexer file is truncated at [01386:003] and there is nothing after that?

Yes, this file finished at [01386:003] line.

The only change I did is commenting out the export below(because my JVM has different location):

```
#export JAVA_HOME=/opt/java
$JAVA_HOME/bin/java -server -classpath $SCRIPT_PATH/p2j/build/lib/p2j.jar.: -Xmx768m -DP2J_HOME=$SCRIPT_PATH
com.goldencode.p2j.convert.ConversionDriver -sd1 F2+M0+CB $SCRIPT_PATH/src/syman "[pPwWJ]" 2>&1 | tee
$SCRIPT_PATH/build/err_f2m0cb.log
```

The other system-wide environment:

```
export ANT_HOME=/usr/ant-1.7.0
```

```
export CHARVA_HOME=/home/eugenie/timco/charva
```

```
export PATH=${PATH}:${ANT_HOME}/bin
```

```
export CLASSPATH=${HOME}/timco/p2j/build/lib/p2j.jar.:
```

But I guess these values are meaningless for ./run\_full.sh. What more I need to check?

And the both src archives have been uploaded into /clients/timco/temp\_conversion\_results for you to review.

#### #51 - 10/11/2012 02:34 PM - Greg Shah

I didn't mean that the command line environment itself was a problem (although it is possible). I meant that in general, your overall installation and its configuration and how you have it setup is somehow a problem or is different.

Do you have 2 completely independent project installations? With 2 completely separate P2J installations? How did you set it all up?

Did the converted result fully build in both installations?



Were there any errors or differences in the conversion driver's output? (you can check the logfiles)

**#52 - 10/11/2012 04:21 PM - Eugenie Lyzenko**

Do you have 2 completely independent project installations? With 2 completely separate P2J installations?

Yes, the both are completely separate

How did you set it all up?

1. I have created separate timco\_cvs dir.
2. Download timco and p2j projects into timco\_cvs
3. Unzip majic\_src.zip, syman\_src.zip and syman\_4gl\_src.zip into src subdir of the timco\_cvs
4. Copy the customer\_libs directory from my current timco dir into timco\_cvs
5. Use this "master" timco\_cvs directory to make two new projects, "Old" and "New"
6. For "New" one apply the rule fixes.
7. Build p2j and run conversion for "Old"
8. Build p2j and run conversion for "New"

The question: what the "project\_root"/srcnew directory should contain? Only java and jrxml?

Did the converted result fully build in both installations?

Yes, I have majic\*.jar files for both.

Were there any errors or differences in the conversion driver's output? (you can check the logfiles)

The err\_f2m0cb.log files are identical(except time and root directory names)

**#53 - 10/11/2012 04:30 PM - Eugenie Lyzenko**

The previous message is updated with customer\_libs directory instruction. And I have checked all created log files and they are identical(except elapsed time and directory name).

**#54 - 10/11/2012 04:48 PM - Greg Shah**

The question: what the "project\_root"/srcnew directory should contain? Only java and jrxml?

Yes.

```
majic_src.zip, syman_src.zip and syman_4gl_src.zip
```

Where did you get these?

**#55 - 10/11/2012 04:52 PM - Eugenie Lyzenko**

Where did you get these?

In my current timco/src subdirectory:

```
eugenie@cyborg64:~/timco/src> ls |*.*.zip
-rw-rw-r- 1 eugenie users 46788922 Sep 15 2009 majic_src.zip
-rw-rw-r- 1 eugenie users 7407009 Dec 20 2010 syman_4gl_src.zip
-rw-rw-r- 1 eugenie users 9771238 Sep 15 2009 syman_src.zip
```

**#56 - 10/11/2012 04:58 PM - Eugenie Lyzenko**

Another question. Trying to run conversion in Windows I have the following message:

```
D:\timco_win>CMD.EXE /C ""C:\Program Files\Java\jdk1.6.0_35\bin\java" -server -Xmx768m -Dfile.encoding="UTF-8"
-DP2J_HOME=D:\timco_win -classpath D:\timco_win\p2j\build\lib\p2j.jar;. com.goldencode.p2j.convert.ConversionDriver -SD1 F2+M0+CB
D:\timco_win\src\syman "".[pPwW]" 2>&1 D:\timco_win\build\err_f2m0cb.log"
Picked up JAVA_TOOL_OPTIONS: -Dfile.encoding="UTF-8"
Missing/incomplete filespec parameter.
```

-----P2J Conversion Driver-----

Syntax:

```

java -DP2J_HOME=&lt;home&gt; ConversionDriver [options] &lt;mode&gt; &lt;filelist&gt;
    java -DP2J_HOME=&lt;home&gt; ConversionDriver -F[options] &lt;mode&gt; &lt;filename&gt;
    java -DP2J_HOME=&lt;home&gt; ConversionDriver -S[options] &lt;mode&gt; &lt;directory&gt; "&lt;filespec&gt;"

```

## Where

Mode (1 or more of the following values, use '+' to delimit if there are 2 or more values; don't insert spaces!)

```

-----Front End-----
F0 = preprocessor (in honor cache mode), lexer, parser,
...

```

The script file to run conversion:

```

-----
SET SCRIPT_PATH=%~dp0
IF SCRIPT_PATH:~-1 == \ (SET SCRIPT_PATH=%SCRIPT_PATH:~,-1%)
SET ANT_OPTS=-Xmx1300m
SET CONV_OPTS=-server -Xmx768m -Dfile.encoding="UTF-8" -DP2J_HOME=%SCRIPT_PATH%
SET FILE_SPEC="*.pPwWJ"

CMD.EXE /C ant clean clean_generated
CMD.EXE /C ant prepare

CMD.EXE /C ""%JAVA_HOME%\bin\java" CONV_OPTS -classpath SCRIPT_PATH\p2j\build\lib\p2j.jar;;
com.goldencode.p2j.convert.ConversionDriver -SD1 F2+M0+CB SCRIPT_PATH\src\syman FILE_SPEC 2>&1 SCRIPT_PATH\build\err_f2m0cb.log"

CMD.EXE /C ant jar 2>&1 SCRIPT_PATH\build\err_ant_jar.log
-----

```

The error meaning the filespec parameter("\*.pPwWJ") can not be found? Another Windows regular expression issue?

The ConversionDriver tells(idx == 1):

```

...
// there must be exactly 2 args left
if ((args.length - idx) != 2) {
    syntax("Missing/incomplete filespec parameter.", -8);
}
...

```

Does it mean two arguments are SCRIPT\_PATH\src\syman FILE\_SPEC? Is the parameter <Mode>(F2+M0+CB) the another argument?

**#57 - 10/11/2012 05:48 PM - Greg Shah**

Those versions of the code (from 2009 and 2010) are much too old to be useful. Please rebuild your local environment using the same version of all TIMCO code as in staging. Get help from Constantin as needed to ensure you can duplicate the staging environment locally. Run conversion there with the same unmodified P2J as in use in staging. Then compare the results with staging and ensure it is identical.

Then copy your local environment, modify the P2J with your changes, run conversion and compare the results.

In regards to the filespec issue, I suspect that your inclusion of the 2->1 filename stuff inside the "" string is causing the problem. Use the Java debugger to find out.

**#58 - 10/11/2012 07:15 PM - Eugenie Lyzenko**

OK.

**#59 - 10/12/2012 04:58 AM - Eugenie Lyzenko**

Constantin, Greg,

Strange thing I have got while converting staging sources:  
Conversion has been finished but then...

Buildfile: build.xml

prepare:

[copy] Copying 646 files to /home/eugenie/timco\_old/build/classes

[copy] Copying 110 files to /home/eugenie/timco\_old/src

compile\_jasper:

[jrc] Compiling 1 report design files.

File : /home/eugenie/timco\_old/srcnew/jrxml/SA-M122P\_20110126\_20110126.jrxml ...

BUILD FAILED

/home/eugenie/timco\_old/build.xml:654: java.lang.ExceptionInInitializerError

This is the exact copy of staging directory under conversion. Have somebody faced this issue? Do I miss some file in customer\_libs directory? Or some config file may be?

**#60 - 10/12/2012 05:32 AM - Eugenie Lyzenko**

Another very interesting finding. Take a look at staging build on lightning. The files

src/syman/adp/adpchks.p.lexer

src/syman/tools/aeroref.p.lexer

are inexpectedly truncated too. So we have some issues in staging too. And I guess this is not my local config problem. And moreover I think because the converted \*.java code is the same in my local "Old" build and in staging on lightning and taking into account my yesterday experiments with "Old" and "New" rules set we can conclude the "New" conversion rules are OK for Linux conversion.

What do you think about the assumption above? Can we go to the Windows conversion run?

**#61 - 10/12/2012 07:00 AM - Eugenie Lyzenko**

Guys,

The problem with

```
compile_jasper:  
[jrc] Compiling 1 report design files.  
File : /home/eugenie/timco_old/srcnew/jrxml/SA-M122P_20110126_20110126.jrxml ...
```

has been solved after removing  
export CLASSPATH=\${HOME}/timco/p2j/build/lib/p2j.jar.:  
from my system profile initial(this is meaningless I guess) and reboot. I think the memory was in some not a good state on my machine(need to check my hardware/OS). Now the java sources compilation is OK.

**#62 - 10/12/2012 03:32 PM - Eugenie Lyzenko**

The news.

I have converted another MAJIC build. The source is from staging + "New" rule set and helper update. The results is:

1. The generated java code is identical for all 3 builds - we have the same results
2. There are differences inside src/syman. But looks like the issue is happening accidentally because
  - we have New == staging but New != Old
  - we have Old == staging nul Old != New
  - we have New == Old but both New,Old != staging
  - we have New Old staging
  - we have New != Old != staging (3 different content in files)

staging - build downloaded from lightning  
Old - build reconverted locally using staging sources  
New - build reconverted locally using staging sources + new rule set and helper update

Taking into account the staging sources also have the truncated files I'm inclining to conclude the "new" rule set and helper update does not affect the conversion results and safe to use. So I'm planning to start the conversion in Windows.

**#63 - 10/12/2012 04:25 PM - Greg Shah**

I agree, the truncation problem does appear to happen on staging too, so that is just a bug that already exists.

Go ahead with the Windows conversion of staging + new changes.

#### #64 - 10/12/2012 05:13 PM - Eugenie Lyzenko

OK. The Windows conversion is in progress.

For now the known issues are:

1. The maximum heap size I can request on my Windows XP 32-bit system is 1300Mb. Will it be enough for ant working - we will see soon. I guess 64-bit XP should work and be a solution.

2. Currently I have not yet solved the conversion command issue with duplicating the output with "2>&1 SCRIPT\_PATH\build\err\_f2m0cb.log". So for now I just removed this part of command and see the process by eyes. BTW the command "1> SCRIPT\_PATH\build\err\_f2m0cb.log 2>&1" is seems to be working but I do not see on the screen what is happening. So for this run there will be no log file. Working on solution.

#### #65 - 10/15/2012 08:49 AM - Eugenie Lyzenko

The new set of issues is related to the file separator difference between Linux(/) and Windows(\) inside the conversion code.

1. We need to change in p2j.cfg.xml file

```
from <parameter name="basepath" value=".src/syman" />
to <parameter name="basepath" value=".src\syman" />
```

2. The embedded commands like RUN menu/fhdp.p generates exception and can not be converted. We need to either change source to menu/fhdp.p or has the helper function like fixupFileName()(The approach I have applied). This helper substitutes the "/" char with "\" in filenames under Windows and does nothing in this case under Linux. After this fix we need to change the rule file convert/base\_structure.xml Line 172 from

```
<action>targetname = nmap.resolveJavaClass(text, pkgname)</action>
```

```
<action>typelist = nmap.getSignature(text)</action>
```

to

```
<action>targetname = nmap.resolveJavaClass(fixupFileName(text), pkgname)</action>
```

```
<action>typelist = nmap.getSignature(fixupFileName(text))</action>
```

This solves the issue.

3. The next point is handling of the manually supported cfg/name\_map\_merge.xml file. The entries have / as file separator and transferred to the project name\_map.xml as is. This causes the following exception:

Rule Type : WALK

Source AST: [ so/oper-p-ref.p ]

BLOCK/INNER\_BLOCK/BLOCK/INNER\_BLOCK/BLOCK/INNER\_BLOCK/BLOCK/STATEMENT/KW\_IF/KW\_THEN/INNER\_BLOCK/BLOCK/STATEMENT/KW\_RUN/FILENAME

/ @1114:14 {7451768267925}

Copy AST : [ so/oper-p-ref.p ]

BLOCK/INNER\_BLOCK/BLOCK/INNER\_BLOCK/BLOCK/INNER\_BLOCK/BLOCK/STATEMENT/KW\_IF/KW\_THEN/INNER\_BLOCK/BLOCK/STATEMENT/KW\_RUN/FILENAME

/ @1114:14 {7451768267925}

Condition : targetname = nmap.resolveJavaClass(fixupFileName(text), pkgname)

Loop : false

--- END RULE REPORT ---

```
at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:718)
    at com.goldencode.p2j.convert.ConversionDriver.processTrees (ConversionDriver.java:892)
    at com.goldencode.p2j.convert.ConversionDriver.back (ConversionDriver.java:789)
    at com.goldencode.p2j.convert.ConversionDriver.main (ConversionDriver.java:1661)
Caused by: java.lang.IllegalStateException: No mapping available for so\oper-p-ref.p
    at com.goldencode.p2j.convert.NameMappingWorker$Library.resolveJavaClass (NameMappingWorker.java:301)
    at com.goldencode.expr.CE10139.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:318)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions (Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing (Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply (Rule.java:440)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:488)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:50)
    at com.goldencode.p2j.pattern.AstWalker.walk (AstWalker.java:213)
    at com.goldencode.p2j.pattern.AstWalker.walk (AstWalker.java:160)
    at com.goldencode.p2j.pattern.PatternEngine.apply (PatternEngine.java:1105)
    at com.goldencode.p2j.pattern.PatternEngine.processAst (PatternEngine.java:1003)
    at com.goldencode.p2j.pattern.PatternEngine.run (PatternEngine.java:690)
    ... 3 more
```

```
To resolve this issue we need to modify annotations/annotations.xml rule Lines 421 and 444 from
pname = xml.getAttributeValue(mergeIndex, "pname")
to
pname = fixupFileName(xml.getAttributeValue(mergeIndex, "pname"))
```

This creates the consistent name\_map.xml file without requirement to have two different cfg/name\_map\_merge.xml for both Linux and Windows. And conversion continues from this point.

#### #66 - 10/15/2012 09:21 AM - Eugenie Lyzenko

The following line of code causes the exception generated, file train2/rst01.p.cache Line 1409:9

```
...
brws:query in frame fRoster = h.
```

The query is not considered as attribute? Problems in 4GL language syntax interpretation? Any ideas what to check?

#### Core Code Conversion

```
.\src\syman\train2\rst01.p
EXPRESSION EXECUTION ERROR:
```

```
throwException(errmsg)
^ { Unsupported method or attribute KW_QUERY. [COLON id <8065948591420> 1409:9] }
```

```
ERROR:
java.lang.RuntimeException: ERROR! Active Rule:
```

#### RULE REPORT

```
Rule Type : WALK
Source AST: [ : ] BLOCK/FUNCTION/BLOCK/ASSIGNMENT/ASSIGN/COLON/ @1409:9 {8065948591420}
Copy AST : [ : ] BLOCK/FUNCTION/BLOCK/ASSIGNMENT/ASSIGN/COLON/ @1409:9 {8065948591420}
Condition : throwException(errmsg)
Loop : false
--- END RULE REPORT ---
```

```
at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:718)
    at com.goldencode.p2j.convert.ConversionDriver.processTrees(ConversionDriver.java:892)
    at com.goldencode.p2j.convert.ConversionDriver.back(ConversionDriver.java:796)
    at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1661)
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:1 [COLON id=8065948591420]
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:226)
    at com.goldencode.p2j.pattern.AstWalker.walk(AstWalker.java:160)
    at com.goldencode.p2j.pattern.PatternEngine.apply(PatternEngine.java:1105)
    at com.goldencode.p2j.pattern.PatternEngine.processAst(PatternEngine.java:1003)
    at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:690)
    ... 3 more
Caused by: com.goldencode.expr.ExpressionException: Expression execution error @1:1
    at com.goldencode.expr.Expression.execute(Expression.java:430)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:401)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.Rule.executeActions(Rule.java:640)
    at com.goldencode.p2j.pattern.Rule.coreProcessing(Rule.java:609)
    at com.goldencode.p2j.pattern.Rule.apply(Rule.java:440)
    at com.goldencode.p2j.pattern.RuleContainer.apply(RuleContainer.java:488)
```

```

    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:50)
    at com.goldencode.p2j.pattern.RuleContainer.apply (RuleContainer.java:488)
    at com.goldencode.p2j.pattern.RuleSet.apply (RuleSet.java:50)
    at com.goldencode.p2j.pattern.AstWalker.walk (AstWalker.java:213)
    ... 7 more
Caused by: com.goldencode.p2j.pattern.CommonAstSupport$UserGeneratedException: Unsupported method or attribute
KW_QUERY. [COLON id <8065948591420> 1409:9]
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.throwException (CommonAstSupport.java:2084)
    at com.goldencode.p2j.pattern.CommonAstSupport$Library.throwException (CommonAstSupport.java:2069)
    at com.goldencode.expr.CE9706.execute (Unknown Source)
    at com.goldencode.expr.Expression.execute (Expression.java:336)
    ... 25 more

```

## #67 - 10/16/2012 12:17 PM - Eugenie Lyzenko

The news/more findings from Windows conversion process.

I have searched the results for syman/train2/rst01.p conversion in Linux. And found there is no code in converted Java sources for the line: brws:query in frame fRoster = h.

Probably the conversion driver considers this code as "unreferenced"? Yes, it is reachable but does not generates any actions in our Linux java results. So I have experimented with this line commenting it out. And the conversion has been finished in Windows. And moreover the conversion result for file syman/train2/rst01.p are binary identical for both Windows and Linux conversion.

The interesting results. In Windows we have the less amount of the \*.jast files. The reason highlights the next cross-OS issue. The filenames in Windows are case insensitive. Two examples:

In Linux we have the following generated files:

syman/item/Item34RF132Desc.jast

syman/item/Item34RF132desc.jast

These are two different files with different content. While in Windows we have only:

syman\item\Item34RF132desc.jast - we can not have two files with same name on file system

In Linux we have the following generated files:

syman/job/Job95RcaHead132.jast

syman/job/Job95Rcahead132.jast

syman/job/Job95RcaHead80.jast

syman/job/Job95Rcahead80.jast

While in Windows we have only

syman\job\Job95RcaHead132.jast

syman\job\Job95RcaHead80.jast

I think the newer file overwrites older one in Windows. The result - we have incomplete generated source file set in Windows and compilation causes the error. The question here: can we modify the name generation to avoid such duplications? Why do we need to generate the almost the same file names with only difference in uppercase letters?

Also the directory handling order is a bit different in Windows. The train subdirectory is handled after train2 while in Linux - train2 after train. Not sure if this is important or not.

The other small thing is the line.separator difference. In Windows we have 2 chars 0x0D 0x0A, while in Linux - 0x0A. This can be important if we have the code depending on the size or value of the line.separator.

And as I have expected the compilation of the Java converted source files takes ~2500 MB of memory. It is not possible to allocate this amount in 32-bit Windows. So to finalize the conversion creating majic\*.jar files it is mandatory to have 64-bit Windows XP system.



In regards to the exception, this 4GL code:

```
h = query q:handle.
  brws:query in frame fRoster = h.
  /*h:get-last().*/
  GET LAST q.
  cur-con-emp = num-results("q").
```

is converted to this:

```
h.assign(query0);
  query0.last();
  curConEmp.assign(query0.size());
```

Indeed, there is no corresponding output for that line. However, that is not in itself a problem. I think something in the Linux conversion process is hiding or deleting that portion of the tree, so that this error is not encountered. Here is the AST:

```
<ast col="0" id="8108898264359" line="0" text="assignment" type="ASSIGNMENT">
  <ast col="7" id="8108898264360" line="1408" text="=" type="ASSIGN">
    <annotation datatype="java.lang.Long" key="peerid" value="68015102104065"/>
    <ast col="5" id="8108898264363" line="1408" text="h" type="VAR_HANDLE">
      <annotation datatype="java.lang.Long" key="oldtype" value="2332"/>
      <annotation datatype="java.lang.Long" key="refid" value="8108898255219"/>
      <annotation datatype="java.lang.Long" key="peerid" value="68015102104066"/>
    </ast>
    <ast col="0" id="8108898264368" line="0" text="expression" type="EXPRESSION">
      <ast col="16" id="8108898264369" line="1408" text=":" type="COLON">
        <ast col="9" id="8108898264370" line="1408" text="query" type="KW_QUERY">
          <ast col="15" id="8108898264373" line="1408" text="query0" type="QUERY">
            <annotation datatype="java.lang.Long" key="query_ref" value="8108898256000"/>
            <annotation datatype="java.lang.String" key="oldtext" value="q"/>
            <annotation datatype="java.lang.Long" key="peerid" value="68015102104067"/>
          </ast>
        </ast>
        <ast col="17" id="8108898264375" line="1408" text="handle" type="ATTR_HANDLE">
          <annotation datatype="java.lang.Long" key="oldtype" value="1431"/>
        </ast>
      </ast>
    </ast>
  </ast>
  <ast col="0" id="8108898264396" line="0" text="statement" type="STATEMENT">
    <ast col="5" id="8108898264397" line="1411" text="GET" type="KW_GET">
      <annotation datatype="java.lang.Long" key="peerid" value="68015102104068"/>
    <ast col="9" id="8108898264402" line="1411" text="LAST" type="KW_LAST">
      <ast col="14" id="8108898264405" line="1411" text="query0" type="QUERY">
        <annotation datatype="java.lang.Long" key="query_ref" value="8108898256000"/>
        <annotation datatype="java.lang.String" key="oldtext" value="q"/>
        <annotation datatype="java.lang.Long" key="peerid" value="68015102104069"/>
      </ast>
    </ast>
  </ast>
  <ast col="0" id="8108898264406" line="0" text="assignment" type="ASSIGNMENT">
    <ast col="17" id="8108898264407" line="1412" text="=" type="ASSIGN">
      <annotation datatype="java.lang.Long" key="peerid" value="68015102104070"/>
    <ast col="5" id="8108898264410" line="1412" text="cur-con-emp" type="VAR_DEC">
      <annotation datatype="java.lang.Long" key="oldtype" value="2332"/>
      <annotation datatype="java.lang.Long" key="refid" value="8108898255495"/>
      <annotation datatype="java.lang.Long" key="peerid" value="68015102104071"/>
    </ast>
    <ast col="0" id="8108898264413" line="0" text="expression" type="EXPRESSION">
      <ast col="19" id="8108898264414" line="1412" text="num-results" type="FUNC_INT">
        <annotation datatype="java.lang.Long" key="oldtype" value="1773"/>
        <annotation datatype="java.lang.Boolean" key="builtin" value="true"/>
        <annotation datatype="java.lang.Boolean" key="returnsunknown" value="true"/>
        <annotation datatype="java.lang.Long" key="peerid" value="68015102104072"/>
      <ast col="31" id="8108898264416" line="1412" text="query0" type="QUERY">
        <annotation datatype="java.lang.Long" key="query_ref" value="8108898256000"/>
        <annotation datatype="java.lang.String" key="oldtext" value="q"/>
      </ast>
    </ast>
  </ast>
```

```

        <annotation datatype="java.lang.Long" key="peerid" value="68015102104073"/>
    </ast>
</ast>
</ast>
</ast>
</ast>
</ast>

```

So the code in question doesn't exist in the AST.

Eugenie: your job is to track back to the code that hides or removes this part of the tree and find out why it doesn't run on Windows properly.

Eric/Constantin: I assume this removal is OK? Do you know what is responsible or why we do it?

#### #69 - 10/16/2012 05:27 PM - Greg Shah

In regard to the case insensitive filename problem in Windows:

In Windows we have the less amount of the \*.jast files. The reason highlights the next cross-OS issue. The filenames in Windows are case insensitive. Two examples:

In Linux we have the following generated files:

syman/item/Item34RF132Desc.jast

syman/item/Item34RF132desc.jast

These are two different files with different content. While in Windows we have only:

syman\item\Item34RF132desc.jast - we can not have two files with same name on file system

Actually, it looks to me like this is an application BUG. Look at the 4GL:

```

display stream rpt
    item.description at 1 when avail item
    space(4)
    item.item-desc      when avail item
    space(4)
    "Product Code:"    when avail prodcode
    prodcode.product-code when avail prodcode
    skip
    with width 132 no-box no-labels frame f-132desc.
down stream rpt with frame f-132-desc.

```

The display statement references f-132desc.

The down statement references f-132-desc.

The extra hyphen in the down statement is probably an error since this frame name is probably supposed to be the same name. So we should let the customer know that this is broken code.

HOWEVER, this still highlights a problem in the windows conversion. Progress is case-insensitive. But our conversion takes 4GL names and attempts to safely convert them to good Java names. But good Java names CANNOT contain hyphens. But we get away with camel-casing the resulting character after a hyphen, which leads to 2 different names in this case where there is an extra hyphen.

As you note, the conversion MUST generate 2 different files here. So we probably need to add some detection logic for when we have name collisions that are caused by our conversion but were in fact different names in the 4GL. This will affect all names that convert to file names (classes).

**#70 - 10/16/2012 05:28 PM - Greg Shah**

Also the directory handling order is a bit different in Windows. The train subdirectory is handled after train 2 while in Linux - train2 after train. Not sure if this is important or not.

This is OK.

**#71 - 10/16/2012 05:31 PM - Greg Shah**

The other small thing is the line.separator difference. In Windows we have 2 chars 0x0D 0x0A, while in Linux - 0x0A. This can be important if we have the code depending on the size or value of the line.separator.

I understand. I can see that some people would want the converted code to have the proper Windows line termination characters. But then again, just because you run the conversion on Windows, doesn't mean that you want the results to have the Windows line termination characters.

How do we emit the result today? I suspect we should provide an optional global configuration parameter in cfg/p2j.cfg.xml to allow a non-default line termination to be specified.

**#72 - 10/16/2012 05:31 PM - Greg Shah**

And as I have expected the compilation of the Java converted source files takes ~2500 MB of memory. It is not possible to allocate this amount in 32-bit Windows. So to finalize the conversion creating majic\*.jar files it is mandatory to have 64-bit Windows XP system.

This is OK. Of course, we will need to document this.

**#73 - 10/16/2012 07:07 PM - Eugenie Lyzenko**

I understand. I can see that some people would want the converted code to have the proper Windows line termination characters. But then again, just because you run the conversion on Windows, doesn't mean that you want the results to have the Windows line termination characters.

May be this difference is not a problem at all if we have all files(including \*.p and generated ones) in the same format, Windows or Linux. But consider the situation when we have some part of the files with 0x0A and some part with 0x0A0D. The conversion code asks the current System value for it. Then scanning the code the conversion driver or another handlers will wants to see the 0x0D char to mark the end of the line. But if the sources are taken from Linux - there will be no such mark and the new line will not be recognized properly. Is it the possible scenario? Having the single line separator for every system may simplify this workm what do you think? Say if we have only one 0x0A char for all situations(because this char exists in both OS as part of the separator).

However, again, if we have no code strongly dependent on exact line separator value - there is no problem here.

How do we emit the result today? I suspect we should provide an optional global configuration parameter in cfg/p2j.cfg.xml to allow a non-default line termination to be specified.

Yes, I think something like line-separator-hex in p2j.cfg.xml file to specify the value in hexadecimal integer.

#### #74 - 10/17/2012 09:53 AM - Eugenie Lyzenko

The root cause for src\syman\train2\rst01.p issue has been found. Take a look at pattern/customer\_specific\_annotations\_prep.rules Line 1455:

<!-- remove uses of the query attribute; this appears in two places,  
both of which are embedded within an assignment; the entire  
assignment is removed -->

```
<rule>
type prog.colon                and
parent.type prog.assign        and
parent.parent.type == prog.assignment    and
this.numImmediateChildren >= 2        and
this.getChildAt(0).type prog.wid_browse    and
this.getChildAt(1).type prog.attr_handle    and
#(int) (#(long) this.getChildAt(1).getAnnotation("oldtype")) ==
prog.kw_query                and
file.endsWith("train2\rst01.p")
```

```
<action>copy.parent.parent.remove()</action>
</rule>
```

We actually must remove this line of code but the file name is hardcoded with Linux file separator. Replacing the last condition with

```
file.endsWith(fixupFileName("train2\rst01.p"))
solves the issue - we have same .ast and .jast files.
```

There are many other places in this file where we have the same file separator explicit specification. I think all of them should be fixed here before rerun conversion in Windows. I'll make the change and test the results.

#### #75 - 10/18/2012 12:10 PM - Eugenie Lyzenko

Greg, some considerations about names

... But good Java names CANNOT contain hyphens. But we get away with camel-casing the resulting character after a hyphen, which leads to 2 different names in this case where there is an extra hyphen.

What if we substitute the hyphen with another char? May be '(' or ')' or another "exotic" char from UTF-8 charset? The '@' and '#' chars are invalid because of using in Javadoc. Or may be we can use some defined string like "HP"(meaning hyphen)? What do you think?

#### #76 - 10/18/2012 05:11 PM - Eugenie Lyzenko

The conversion has been completed in Windows. But the compilation can not be performed due to the following name generation issues:

1. FHdrp

There are two source files:

syman/menu/fhdrp.p and syman/menu/f-hdrp.p, in Windows we have only menu\Fhdrp.java. The javac can not find the FHdrp class

2. Job95RRcaHead80 and Job95RRcaHead132.

The source syman/job/job95-r.p refers to the frames:

WITH NO-BOX NO-LABELS WIDTH 80 FRAME rcahead80.

WITH NO-BOX NO-LABELS WIDTH 80 FRAME rca-head80.

and

WITH NO-BOX NO-LABELS WIDTH 132 FRAME rcahead132.

WITH NO-BOX NO-LABELS WIDTH 132 FRAME rca-head132.

In Windows we have only Job95RRcahead80 and Job95RRcahead132 classes in majic\ui\job. This causes javac to generate class missing error.

3. Item34RF132Desc.

The source syman/item/item34-r.p refers to the both frames:

f-132desc and f-132-desc

In Windows we have only Item34RF132desc class in majic\ui\item. This causes javac to generate class missing error. But looks like this is just a bug in Progress application source and can be eliminated without additional efforts.

As you can see we have 2 real problems here with Java name generation. The point [#2](#) can easily be solved by manually modification of the syman/job/job95-r.p file, to change the rca-head132 to something other without hyphen inside the \*.p source. The point [#1](#) is more complex because we need to track all references to f-hdrp.p(or fhdrp.p), rename them to other name and rename file itself. Of course we can implement a bit different name generation approach and then do nothing in Progress source file.

#### #77 - 10/19/2012 08:42 AM - Eugenie Lyzenko

Another viewpoint to the naming issue. Theoretically the NTFS is case-sensitive. In XP we can turn this feature on according to MS knowledge base by setting

HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\ObCaseInsensitive to 0 in registry. The problem is the Win32 programs are still case-insensitive and I have got very strange behaved system turning on case-sensitivity. On the other hand to run conversion in Windows we will have to use 64-bit version to compile the target and probably in this OS the case-sensitivity will also work.

So if we can handle the case-sensitivity issue by Windows OS tools - we can leave our P2J code untouched. What do you think? Does it make sense to dig into this direction?

#### #78 - 10/22/2012 08:47 AM - Eugenie Lyzenko

Further investigation shows the following:

1. We can set the NTFS in Windows XP 64-bit to be case sensitive. And it works stable. The programs that support this mode will work correctly.
2. Java 6 SE SDK does not support this mode in Windows. The class File refers to the native implementation methods of the FileSystem class. The results is depending on native library implementation for Windows for which the source code is not public. Currently the files with letter name case only difference are considered to be the same on Windows.

So I think in this case we can not be dependent from this level of support even we are able to hack the native library. So for Windows we need to implement the approach to provide alternative name construction if the name candidate(ignoring case state) is already occupied by some class or variable before. We can apply this approach only when the conversion is running under Windows OS.

#### #79 - 10/22/2012 09:23 AM - Greg Shah

However, again, if we have no code strongly dependent on exact line separator value - there is no problem here .

Our parsing code doesn't care what the line termination characters are. We are fine both ways, so there is no problem here.

How do we emit the result today?

I still need the answer to this question. What line termination characters are placed in the output files today?

#### #80 - 10/22/2012 09:27 AM - Greg Shah

Making the Windows file system case-sensitive is not an option.

It is clear that we have to rework our name conversion to disambiguate any names that would only be differentiated by case. We can do that on both Linux and Windows, but it will take some work. DON'T work on that problem any more, I will create a separate task and we will assign someone to work it later.

Please prepare the full set of changes needed to get the results you have so far. Make sure that the final result is re-tested on Linux such that it makes no change to the Majic conversion. Then submit it for review. Constantin will do the review.

**#81 - 10/22/2012 10:51 AM - Eugenie Lyzenko**

I still need the answer to this question. What line termination characters are placed in the output files today?

Windows conversion:

1. Resulting \*.java converted classes - 0x0A
2. \*.cache file - 0x0A
3. \*.ast, \*.dict, \*.jast, \*.lexer, \*.p2o, \*.parser, \*.schema - (0x0D+0x0A)

Linux conversion:

1. Everywhere - 0x0A

**#82 - 10/22/2012 10:52 AM - Eugenie Lyzenko**

Please prepare the full set of changes needed to get the results you have so far. Make sure that the final result is re-tested on Linux such that it makes no change to the Majic conversion. Then submit it for review. Constantin will do the review.

OK.

**#83 - 10/22/2012 02:43 PM - Greg Shah**

Don't make any changes to the line termination processing. I'll create a separate task for that and we will work it later.

**#84 - 10/22/2012 02:47 PM - Eugenie Lyzenko**

Don't make any changes to the line termination processing. I'll create a separate task for that and we will work it later.

I have no changes regarding this point. Just because for now I have not found any real issue caused by the line termination difference.

**#85 - 10/24/2012 11:11 AM - Eugenie Lyzenko**

- File *evl\_upd20121022a.zip* added
- File *evl\_upd20121022b.zip* added

Updates to make Windows conversion have been uploaded.

**#86 - 10/31/2012 01:23 PM - Greg Shah**

- Target version set to 24

**#87 - 11/01/2012 01:37 PM - Greg Shah**

- File *evlupd\_20121010b.zip* added
- File *evlupd\_20121010a.zip* added

Please merge your changes to `pattern/customer_specific_annotations.rules` with the latest version in the TIMCO CVS project. Constantin just checked in a new version and yours must be based on that.

**#88 - 11/01/2012 03:22 PM - Eugenie Lyzenko**

- File *evl\_upd20121101b.zip* added
- File *evl\_upd20121101a.zip* added

The files to run Windows conversion has been updated merging with the recent code base. The full testing cycle will be performed to check if this update will work fine on both Linux and Windows systems.

**#89 - 11/02/2012 03:53 PM - Greg Shah**

Some feedback:

1. `annotations.xml` should have only 1 history entry (merge them together).
2. `p2o.xml` needs its copyright end date updated to 2012.
3. `StringHelper` should have only 1 history entry (merge them together).
4. `StringHelper.quoteReplacement()` is not needed.
5. `CommonAstHelper` should have only 1 history entry (merge them together).
6. `CommonAstHelper.Library.quoteReplacment` should just call the `J2SE` method directly (the `StringHelper` version is not needed). It also can just be simplified to this:

```
return java.util.regex.Matcher.quoteReplacement(text);
```

I don't want to limit it to just Windows. It is a useful tool generally. Is there a problem if it is enabled on Linux?



**#90 - 11/02/2012 04:03 PM - Eugenie Lyzenko**

OK. I'll prepare the modified updates. And the conversion is identical for these updates turned on and off in Linux.

I don't want to limit it to just Windows. It is a useful tool generally. Is there a problem if it is enabled on Linux?

I can not say it is 100% safe in Linux. But I'll test to see what is happening.

**#91 - 11/02/2012 05:19 PM - Eugenie Lyzenko**

- File *evl\_upd20121102a.zip* added

The P2J part has been updated:

1. Fixed copyright date stamps
2. The history entries have been merged to have 1 per file.
3. The method `StringHelper.quoteReplacement()` has been removed. The J2SE call `java.util.regex.Matcher.quoteReplacement(text)` is now invoked in `CommonAstHelper.Library` unconditionally. The safety of usage this in Linux is now under testing.

**#92 - 11/05/2012 10:21 AM - Greg Shah**

The code looks good. If it passes testing, please work with Constantin to get it applied to staging. At that time you can also check it into CVS (both the P2J and the TIMCO projects). Don't forget to tag the specific files with the Hxxx history entries.

Once it is both on staging and in CVS, you can send out the updates.

**#93 - 11/12/2012 10:48 AM - Greg Shah**

----- Original Message -----

Subject: Re: The Windows conversion related update

Date: Wed, 07 Nov 2012 18:35:55 +0400

From: Eugenie V. Lyzenko <[evl@goldencode.com](mailto:evl@goldencode.com)>

To: Eric Faulhaber <[ecf@goldencode.com](mailto:ecf@goldencode.com)>

CC: Constantin Asofiei <[ca@goldencode.com](mailto:ca@goldencode.com)>, Greg Shah <[ges@goldencode.com](mailto:ges@goldencode.com)>

Eric,

The regression testing completed. There are no failures for both parts. But I had to restart the CTRL-C part several times due to lightning loading I guess. I'm stopping the staging server.

When the P2J repository will be available again I'll check the changes in and send the update to everyone.

Regards,  
Eugenie.

Eugenie,

I don't recall what this is, but it looks familiar. Constantin, do you remember what this *hedwig.log* is about?

In any event, I don't expect it will get in the way of regression testing. I think this is custom code added by TIMCO, which is not part of the regression baselines.

Let me know if the harness has any issues.

Thanks,  
Eric

Eric,

There is still permission issue while compiling the converted results. The conversion itself seems to be OK. Is it severe issue? I'll try to start the regression testing in a 30 min if there will be no answer from you.

Regards,  
Eugenie.

-----  
Buildfile: build.xml

prepare:

[copy] Copying 646 files to /mnt/san/sata/gc/20121029/build/classes

[copy] Copying 110 files to /mnt/san/sata/gc/20121029/src

compile\_jasper:

[jrc] log4j:ERROR setFile(null,true) call failed.

[jrc] java.io.FileNotFoundException: /tmp/hedwig.log (Permission denied)

[jrc] at java.io.FileOutputStream.openAppend(Native Method)

[jrc] at java.io.FileOutputStream.<init>(FileOutputStream.java:177)

[jrc] at java.io.FileOutputStream.<init>(FileOutputStream.java:102)

[jrc] at org.apache.log4j.FileAppender.setFile(FileAppender.java:290)

[jrc] at org.apache.log4j.RollingFileAppender.setFile(RollingFileAppender.java:194)

[jrc] at org.apache.log4j.FileAppender.activateOptions(FileAppender.java:164)

[jrc] at org.apache.log4j.config.PropertySetter.activate(PropertySetter.java:257)

[jrc] at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:133)

[jrc] at org.apache.log4j.config.PropertySetter.setProperties(PropertySetter.java:97)

[jrc] at org.apache.log4j.PropertyConfigurator.parseAppender(PropertyConfigurator.java:689)

[jrc] at org.apache.log4j.PropertyConfigurator.parseCategory(PropertyConfigurator.java:647)

[jrc] at org.apache.log4j.PropertyConfigurator.configureRootCategory(PropertyConfigurator.java:544)

[jrc] at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:440)

[jrc] at org.apache.log4j.PropertyConfigurator.doConfigure(PropertyConfigurator.java:476)

[jrc] at org.apache.log4j.helpers.OptionConverter.selectAndConfigure(OptionConverter.java:471)

[jrc] at org.apache.log4j.LogManager.<clinit>(LogManager.java:125)

[jrc] at org.apache.log4j.Logger.getLogger(Logger.java:105)

[jrc] at org.apache.commons.logging.impl.Log4JLogger.getLogger(Log4JLogger.java:289)

[jrc] at org.apache.commons.logging.impl.Log4JLogger.<init>(Log4JLogger.java:109)

[jrc] at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)

[jrc] at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:39)

[jrc] at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:27)

[jrc] at java.lang.reflect.Constructor.newInstance(Constructor.java:513)

[jrc] at org.apache.commons.logging.impl.LogFactoryImpl.createLogFromClass(LogFactoryImpl.java:1116)

[jrc] at org.apache.commons.logging.impl.LogFactoryImpl.discoverLogImplementation(LogFactoryImpl.java:914)

[jrc] at org.apache.commons.logging.impl.LogFactoryImpl.newInstance(LogFactoryImpl.java:604)

[jrc] at org.apache.commons.logging.impl.LogFactoryImpl.getInstance(LogFactoryImpl.java:336)

[jrc] at org.apache.commons.logging.impl.LogFactoryImpl.getInstance(LogFactoryImpl.java:310)

[jrc] at org.apache.commons.logging.LogFactory.getLog(LogFactory.java:685)

[jrc] at net.sf.jasperreports.engine.util.JRLoader.<clinit>(JRLoader.java:60)

[jrc] at net.sf.jasperreports.engine.util.JRProperties.loadProperties(JRProperties.java:291)

[jrc] at net.sf.jasperreports.engine.util.JRProperties.initProperties(JRProperties.java:164)

[jrc] at net.sf.jasperreports.engine.util.JRProperties.<clinit>(JRProperties.java:150)

[jrc] at net.sf.jasperreports.ant.JRAntCompileTask.execute(JRAntCompileTask.java:224)

[jrc] at org.apache.tools.ant.UnknownElement.execute(UnknownElement.java:288)

[jrc] at sun.reflect.GeneratedMethodAccessor2.invoke(Unknown Source)

[jrc] at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)

[jrc] at java.lang.reflect.Method.invoke(Method.java:597)

[jrc] at org.apache.tools.ant.dispatch.DispatchUtils.execute(DispatchUtils.java:105)

[jrc] at org.apache.tools.ant.Task.perform(Task.java:348)

[jrc] at org.apache.tools.ant.Target.execute(Target.java:357)

[jrc] at org.apache.tools.ant.Target.performTasks(Target.java:385)

[jrc] at org.apache.tools.ant.Project.executeSortedTargets(Project.java:1329)

[jrc] at org.apache.tools.ant.Project.executeTarget(Project.java:1298)

[jrc] at org.apache.tools.ant.helper.DefaultExecutor.executeTargets(DefaultExecutor.java:41)

[jrc] at org.apache.tools.ant.Project.executeTargets(Project.java:1181)

[jrc] at org.apache.tools.ant.Main.runBuild(Main.java:698)

[jrc] at org.apache.tools.ant.Main.startAnt(Main.java:199)

[jrc] at org.apache.tools.ant.launch.Launcher.run(Launcher.java:257)

[jrc] at org.apache.tools.ant.launch.Launcher.main(Launcher.java:104)

[jrc] Compiling 1 report design files.

[jrc] File : /mnt/san/sata/gc/20121029/srcnew/jrxml/SA-M122P\_20110126\_20110126.jrxml ... OK.

compile:

[javac] Compiling 14607 source files to /mnt/san/sata/gc/20121029/build/classes

[javac] Note: Some input files use unchecked or unsafe operations.

[javac] Note: Recompile with -Xlint:unchecked for details.  
[echo] Compiling Java Sources Done.

admin\_compile:  
[echo] Compiling Java Sources Done.

admin\_jar:  
[jar] Building jar: /mnt/san/sata/gc/20121029/build/lib/majic\_admin\_server.jar  
[jar] Building jar: /mnt/san/sata/gc/20121029/build/lib/majic\_admin\_client.jar  
[signjar] Signing JAR: /mnt/san/sata/gc/20121029/build/lib/majic\_admin\_client.jar to  
/mnt/san/sata/gc/20121029/build/lib/majic\_admin\_client.jar as majic\_admin\_client  
[signjar] Enter Passphrase for keystore: Enter key password for majic\_admin\_client:

jar:  
[jar] Building jar: /mnt/san/sata/gc/20121029/build/lib/majic.jar  
[jar] Building jar: /mnt/san/sata/gc/20121029/build/lib/majic\_ui.jar

BUILD SUCCESSFUL  
Total time: 23 minutes 8 seconds  
-----

Eugenie,

Actually, I ended up chown-ing the entire staging directory to myself, because I had other permissions problems when relaunching conversion. If you have any issues running regression testing when conversion is done, please let me know.

Thanks,  
Eric

On 11/06/2012 01:45 PM, Eugenie V. Lyzenko wrote:

Eric,

We now use the H2 database to store statistics when running reports. This produces a database file under the rptdb subdirectory (apparently even when we are not running reports). There was a version of this file with the group write bit turned off. I have turned it on. It should work now.

Constantin is done for the day. I will restart conversion in staging shortly.

OK. Thanks.

Thanks,  
Eric

Regards,  
Eugenie.

On 11/06/2012 12:58 PM, Eugenie V. Lyzenko wrote:

Constantin,

Conversion has started. You can check "tail -f /gc/convert/staging/nohup.out" to find the progress and in ~4 hours should be done.

Please start the conversion once again. It is failed due to strange reason. Any ideas?:

-----  
Generate Hibernate Mappings (database P2Os)  
-----

ERROR:  
java.lang.RuntimeException: ERROR! Active rule could not be determined  
at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:718)  
at com.goldencode.p2j.convert.ConversionDriver.processTrees(ConversionDriver.java:892)  
at com.goldencode.p2j.convert.ConversionDriver.middle(ConversionDriver.java:628)  
at com.goldencode.p2j.convert.ConversionDriver.main(ConversionDriver.java:1655)  
Caused by: com.goldencode.p2j.cfg.ConfigurationException: Error loading configuration: hibernate.xml; XML path:

```

/cfg/worker
at com.goldencode.p2j.pattern.ConfigLoader.loadImpl(ConfigLoader.java:469)
at com.goldencode.p2j.pattern.ConfigLoader.load(ConfigLoader.java:405)
at com.goldencode.p2j.pattern.PatternEngine.initialize(PatternEngine.java:797)
at com.goldencode.p2j.pattern.PatternEngine.run(PatternEngine.java:671)
... 3 more
Caused by: com.goldencode.p2j.cfg.ConfigurationException: Unable to register worker:
com.goldencode.p2j.util.StatsWorker
at com.goldencode.p2j.pattern.ConfigLoader.worker(ConfigLoader.java:641)
at com.goldencode.p2j.pattern.ConfigLoader.processChildElements(ConfigLoader.java:592)
at com.goldencode.p2j.pattern.ConfigLoader.loadImpl(ConfigLoader.java:439)
... 6 more
Caused by: java.lang.RuntimeException: Error preparing database
at com.goldencode.p2j.util.StatsWorker.prepareDatabase(StatsWorker.java:411)
at com.goldencode.p2j.util.StatsWorker.<init>(StatsWorker.java:268)
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:39)
at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:27)
at java.lang.reflect.Constructor.newInstance(Constructor.java:513)
at java.lang.Class.newInstance0(Class.java:355)
at java.lang.Class.newInstance(Class.java:308)
at com.goldencode.p2j.pattern.RuleContainer.registerWorker(RuleContainer.java:574)
at com.goldencode.p2j.pattern.ConfigLoader.worker(ConfigLoader.java:636)
... 8 more
Caused by: org.h2.jdbc.JdbcSQLException: The database is read only; SQL statement:
create cached local temporary table match (id identity, stat int, file int, line int, col int, text varchar) [90097-169]
at org.h2.message.DbException.getJdbcSQLException(DbException.java:329)
at org.h2.message.DbException.get(DbException.java:169)
at org.h2.message.DbException.get(DbException.java:146)
at org.h2.message.DbException.get(DbException.java:135)
at org.h2.engine.Database.checkWritingAllowed(Database.java:1702)
at org.h2.store.PageStore.update(PageStore.java:1049)
at org.h2.store.PageFreeList.allocate(PageFreeList.java:98)
at org.h2.store.PageStore.allocatePage(PageStore.java:1154)
at org.h2.store.PageStore.allocatePage(PageStore.java:1141)
at org.h2.index.PageDataIndex.<init>(PageDataIndex.java:76)
at org.h2.table.RegularTable.<init>(RegularTable.java:83)
at org.h2.schema.Schema.createTable(Schema.java:552)
at org.h2.command.ddl.CreateTable.update(CreateTable.java:134)
at org.h2.command.CommandContainer.update(CommandContainer.java:75)
at org.h2.command.Command.executeUpdate(Command.java:230)
at org.h2.jdbc.JdbcStatement.executeInternal(JdbcStatement.java:177)
at org.h2.jdbc.JdbcStatement.execute(JdbcStatement.java:152)
at com.goldencode.p2j.util.StatsWorker.prepareDatabase(StatsWorker.java:380)
... 17 more

```

Regards,  
Eugenie.

On 11/6/2012 5:22 PM, Eugenie V. Lyzenko wrote:

Constantin,

I'm rebuilding P2J now.

Please execute ./run\_full.sh in staging directory. I have no permissions to do this.

And let me know when the conversion is done.

Regards,  
Eugenie.

On 11/06/2012 01:36 PM, Eugenie V. Lyzenko wrote:

Guys,

I can not build staging:

...

/mnt/san/sata/gc/20121029/p2j/build.xml:469: Unable to determine generated class

Don't I have enough permissions? Can you execute ant all in staging/p2j?

Regards,  
Eugenie.

Eugenie,

You can use it at any time.

On 11/05/2012 11:11 PM, Eugenie V. Lyzenko wrote:

Constantin,

I'm going to check the Windows conversion related update into CVS. Please let me know when I can apply it to staging and run the regression testing.

Regards,  
Eugenie.

**#94 - 11/12/2012 10:50 AM - Greg Shah**

Please check in your update to Bazaar. Remember that in Bazaar, we do NOT do any tagging of history entries.

Do make sure that your check in ONLY includes the files in this update AND it includes ALL the files in a single "bzd commit" command. If those are the only changes in your p2j tree, then it will be easy. If you have other changes there as well, then you must explicitly list all the files to be committed in the "bzd commit" command.

**#95 - 11/12/2012 11:31 AM - Eugenie Lyzenko**

Please check in your update to Bazaar. Remember that in Bazaar, we do NOT do any tagging of history entries.

Do make sure that your check in ONLY includes the files in this update AND it includes ALL the files in a single "bzz commit" command.

OK. The questions.

1. The first thing I need is to set up the Golden Code history entry tools. After this step all history entries will be handled automatically(no need to tag each files). Correct?

2. I have several files to commit in P2J. The key point here is to do SINGLE commit command with ALL files. Something like:  
bzz commit -m "Windows conversion ability update" file1 file2 flie3 ... fileN. Correct?

**#96 - 11/12/2012 11:33 AM - Greg Shah**

1. The first thing I need is to set up the Golden Code history entry tools. After this step all history entries will be handled automatically(no need to tag each files). Correct?

No. The history tools are only there for your convenience. They have no impact on the check in process. You can check in without them installed.

2. I have several files to commit in P2J. The key point here is to do SINGLE commit command with ALL files. So something like:

bzz commit -m "Windows conversion ability update" file1 file2 flie3 ... fileN. Correct?

Yes. Exactly right.

**#97 - 11/12/2012 12:46 PM - Eugenie Lyzenko**

The changes are in bzz.

**#98 - 11/13/2012 08:04 AM - Greg Shah**

- % Done changed from 0 to 100

- Status changed from WIP to Closed

**#99 - 11/16/2016 01:17 PM - Greg Shah**

- Target version changed from 24 to Conversion Support for Server Features

**Files**

primes.p.ast.win32	61.8 KB	09/28/2012	Eugenie Lyzenko
primes.p.ast.linux	114 KB	09/28/2012	Eugenie Lyzenko
evl_upd20121022a.zip	124 KB	10/24/2012	Eugenie Lyzenko
evl_upd20121022b.zip	15 KB	10/24/2012	Eugenie Lyzenko
evlupd_20121010a.zip	121 KB	11/01/2012	Greg Shah
evlupd_20121010b.zip	1.75 KB	11/01/2012	Greg Shah
evl_upd20121101a.zip	126 KB	11/01/2012	Eugenie Lyzenko
evl_upd20121101b.zip	14.7 KB	11/01/2012	Eugenie Lyzenko
evl_upd20121102a.zip	125 KB	11/02/2012	Eugenie Lyzenko