

# 33491

Rishabh Garg

April 26, 2013

## Abstract

This document talks about the following error: Compiler hangs when program ends on `//`. This bug is not limited to the use of `//` in the input. Other unexpected inputs like `///` or the like will hang the compiler. Compiler hangs on such an input and enters into an infinite loop.

## 1 Introduction

The parser is quite fragile and some unexpected errors occur with a class of inputs. In this document, the bug fix of such unexpected errors is explained. For the simple program, for example, `//` the compiler hangs and enters into an infinite loop.

## 2 Finding the error

To find a detailed report of the error the following program was written:

```
package prolog;
import gnu.prolog.term.AtomTerm;
import gnu.prolog.term.CompoundTerm;
import gnu.prolog.term.Term;
import gnu.prolog.term.VariableTerm;
import gnu.prolog.vm.Environment;
import gnu.prolog.vm.Interpreter;
import gnu.prolog.vm.PrologException;

/**
 *
 * @author Rishabh
 */
public class Prolog {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) throws PrologException {
//TODO code application logic here

//Create an environment environment
```

```

gnu.prolog.vm.Environment environment = new Environment();
//A file is created which has the facts and code in Prolog
environment.ensureLoaded(AtomTerm.get("/Users/Rishabh/Desktop/test.pl"));

//environment.ensureLoaded(AtomTerm.get("a(1)")); For testing.

//Interpreter
gnu.prolog.vm.Interpreter interp = environment.createInterpreter();
environment.runInitialization(interp);

//Setting a goal b(X)
VariableTerm term = new VariableTerm("X");
    Term[] ter = new Term[1];
    ter[0] = term;
    CompoundTerm comp = new CompoundTerm("b",ter);

//Printing for output and debugging
System.out.println(comp.args[0]);
System.out.println(interp.runOnce(comp));
System.out.println(term.dereference());

    }

```

If the file test.pl contains the following code: a(7). b(X) :- a(X).

```

X
1
7
BUILD SUCCESSFUL (total time: 0 seconds)

```

The above output is displayed.

Now, to test the compiler the file test.pl was made to contain the following text: //

On running the compiler now, the compiler hangs and enters into an infinite loop.

The debugger was run and a breakpoint was inserted in the environment.ensureLoaded(..) and it was realised that the compiler hanged because of this line. The next step was to step-into the method and tracked to prologTextLoaderState.ensureLoaded(term). From here, the tracking took to newPrologTextLoader(this, term). This finally led to the following fragment of code:

```

protected void processFile()
{
prologTextLoaderState.beforeProcessFile(this);
while (currentReader != null)
{
Term term;
try
{
term = currentReader.readTerm(operatorSet);
}
catch (ParseException ex)
{
// ex.printStackTrace();
logError(ex);
}
}
}

```

```
continue;
}
...(code continues)
```

The error was in `logError(ex)`. On encountering `continue` the program calls itself and runs from the start again where `currentReader` does not change in state in case `ParseException` is caught. This makes the loop run all over again and again and an infinite loop occurs.

### 3 Bug-Fix

```
protected void processFile()
{
  prologTextLoaderState.beforeProcessFile(this);
  while (currentReader != null)
  {
    Term term;
    try
    {
      term = currentReader.readTerm(operatorSet);
    }
    catch (ParseException ex)
    {
      try
      {
        currentReader.readTermEof();
      } catch (ParseException e) {
        e.printStackTrace();
        break;
      }
      logError(ex);
      continue;
    }
    ...(code continues)
```

Now the output for `//` would be:

```
gnu.prolog.io.ParseException: Encountered "/" at line 1, column 1.
Was expecting one of:
    <EOF>
    <END_TOKEN> ...
    <NAME_TOKEN> ...
    <VARIABLE_TOKEN> ...
    <INTEGER_TOKEN> ...
    <FLOAT_NUMBER_TOKEN> ...
    <OPEN_LIST_TOKEN> ...
    <OPEN_CURLY_TOKEN> ...
    <CHAR_CODE_LIST_TOKEN> ...
    <OPEN_TOKEN> ...
```

```
at gnu.prolog.io.TermReader.readTermEof(TermReader.java:87)
at gnu.prolog.io.TermReader.readTermEof(TermReader.java:104)
at gnu.prolog.io.TermReader.readTermEof(TermReader.java:114)
```

```
at gnu.prolog.database.PrologTextLoader.processFile(PrologTextLoader.java:208)
at gnu.prolog.database.PrologTextLoader.<init>(PrologTextLoader.java:103)
at gnu.prolog.database.PrologTextLoaderState.ensureLoaded(PrologTextLoaderState.java:387)
at gnu.prolog.vm.Environment.ensureLoaded(Environment.java:441)
at Prolog.main(Prolog.java:29)
```

X

```
Exception in thread "main" java.lang.IllegalArgumentException: The goal is not currently active
at gnu.prolog.vm.Interpreter.stop(Interpreter.java:564)
at gnu.prolog.vm.Interpreter.runOnce(Interpreter.java:629)
at Prolog.main(Prolog.java:38)
```

## 4 Conclusion

The source code has been made a jar file again and submitted on <https://savannah.gnu.org/bugs/?33491>.