Getting Started with ANTLR v4

Introduction

Hi and welcome to the Honey Badger 4.2 release of ANTLR! It's named after the fearless hero of the Crazy Nasty-Ass Honey Badger since ANTLR v4 takes whatever you give it--it just doesn't give a crap! See Why do we need ANTLR v4? and the preface of the ANTLR v4 book.

Installation

ANTLR is really two things: a tool that translates your grammar to a parser/lexer in Java (or other target language) and the runtime needed by the generated parsers/lexers. Even if you are using ANTLRWorks to run the ANTLR tool, the generated code will still need the runtime library.

The first thing you should do is probably download and install ANTLRWorks, built by Sam Harwell co-author of ANTLR v4. Even if you only use it for editing, it's great. Then, follow the instructions below to get the runtime environment available to your system to run generated parsers/lexers. In what follows, I talk about antlr-4.2-complete.jar, which has the tool and the runtime and any other support libraries (e.g., ANTLR v4 is written in v3).

If you are going to integrate ANTLR into your existing build system using mvn, ant, or want to get ANTLR into your IDE such as eclipse or intellij, see Integrating ANTLR into Development Systems.

UNIX

0. **Install Java** (version 1.6 or higher)

1. **Download**

```
$ cd /usr/local/lib
$ curl -O http://www.antlr.org/download/antlr-4.2-complete.jar
```

Or just download in browser from website:

http://www.antlr.org/download.html

and put it somewhere rational like `/usr/local/lib`.

2. **Add antlr-4.2-complete.jar to your CLASSPATH:**

```
$ export CLASSPATH=".:/usr/local/lib/antlr-4.2-complete.jar:$CLASSPATH"
```

It's also a good idea to put this in your `.bash_profile` or whatever your startup script is.

3. **Create aliases for the ANTLR Tool, and TestRig.**

```
$ alias antlr4='java -jar /usr/local/lib/antlr-4.2-complete.jar'
$ alias grun='java org.antlr.v4.runtime.misc.TestRig'
```

WINDOWS

(Thanks to Graham Wideman)

0. **Install Java** (version 1.6 or higher)

1. **Download** http://www.antlr.org/download/antlr-4.2-complete.jar
Save to your directory for 3rd party Java libraries, say `C:\Javalib`

2. **Add antlr-4.2-complete.jar to CLASSPATH**, either:

- Permanently: Using System Properties dialog > Environment variables > Create or append to CLASSPATH variable

- Temporarily, at command line:

  ```
  SET CLASSPATH=.;C:\Javalib\antlr-4.2-complete.jar;%CLASSPATH%
  ```

3. Create short convenient commands for the ANTLR Tool, and TestRig, using batch files or doskey commands:

Batch files (in directory in system PATH)

antlr4.bat

```
java org.antlr.v4.Tool %*
```

grun.bat

```
java org.antlr.v4.runtime.misc.TestRig %*
```

Or, use doskey commands:

```
doskey antlr4=java org.antlr.v4.Tool $*
doskey grun =java org.antlr.v4.runtime.misc.TestRig $*
```

Testing the installation

Either launch `org.antlr.v4.Tool` directly:

```
$ java org.antlr.v4.Tool
ANTLR Parser Generator Version 4.2
-o ___ specify output directory where all output is generated
-lib ___ specify location of .tokens files
...
```

or use `-jar` option on java:

```
$ java -jar /usr/local/lib/antlr-4.2-complete.jar
ANTLR Parser Generator Version 4.2
-o ___ specify output directory where all output is generated
-lib ___ specify location of .tokens files
...
```

A First Example

In a temporary directory, put the following grammar inside file Hello.g4:

Hello.g4

```
// Define a grammar called Hello
grammar Hello;
r  : 'hello' ID ;         // match keyword hello followed by an identifier
ID : [a-z]+ ;             // match lower-case identifiers
WS : [ \t\r\n]+ -> skip ; // skip spaces, tabs, newlines
```

Then run ANTLR the tool on it:

```
$ cd /tmp
$ antlr4 Hello.g4
$ javac Hello*.java
```

Now test it:

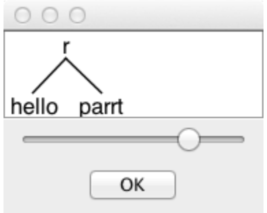
```
$ grun Hello r -tree
hello parrt
^D
(r hello parrt)
```

(That ^D means EOF on unix; it's ^Z in Windows.) The -tree option prints the parse tree in LISP notation.

It's nicer to look at parse trees visually.

```
$ grun Hello r -gui
hello parrt
^D
```

That pops up a dialog box showing that rule `r` matched keyword `hello` followed by identifier `parrt`.



Book source code

The book has lots and lots of examples that should be useful to. You can download them here for free:

http://pragprog.com/titles/tpantlr2/source_code

Also, we are starting a collection of grammars for v4 and tracking it at github:

https://github.com/antlr/grammars-v4