HISTORY

- John Green (Joanju)
- Initially written in C / C++
- Java version since 2008
- Now on GitHub
Main repository is oehive/proparse
  - Almost no activity here

Most active fork is jakejustus/proparse
  - Version used in the SonarQube plugin
WHAT IS PROPARSE?

- A tool to analyse and understand ABL source code

- Different levels:
  - Character stream
  - Token stream
  - Abstract syntax tree
  - Symbols
ANTLR

- Another Tool for Language Recognition

- Why program by hand in five days what you can spend 25 years automating?
PREPROCESSOR & LEXER

- Convert a stream of characters to a stream of tokens
- A token is the most simple unit to represent a set of characters:
  - < is token LEFTANGLE
  - <= is token LTOREQUAL
  - variable is token VARIABLE
**PREPROCESSOR & LEXER**

<table>
<thead>
<tr>
<th>Line</th>
<th>Text</th>
<th>Tokens</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>DEFINE VARIABLE foo AS ¶</code></td>
<td><code>DEFINE VARIABLE</code> no-undo</td>
</tr>
<tr>
<td>2</td>
<td><code>· · · CHARACTER ¶</code></td>
<td><code>· · · CHARACTER</code></td>
</tr>
<tr>
<td>3</td>
<td><code>· · · ¶</code></td>
<td><code>· · ·</code></td>
</tr>
<tr>
<td>4</td>
<td><code>· NO-UNDO ¶</code></td>
<td><code>· NO-UNDO</code></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td><code>WS</code></td>
</tr>
</tbody>
</table>
&SCOPED-DEFINE DEBUG TRUE
&IF (&DEBUG) AND OPSYS EQ 'WIN32' &THEN
MESSAGE "Debug message".
&ENDIF
**PREPROCESSOR & LEXER**

```
MESSAGE "{param1}" VIEW-AS ALERT-BOX BUTTONS {param2}.
```

```
{ myinc.i &param1='abc' &param2=YES-NO-CANCEL }
```
Convert a stream of tokens into a syntax tree
**TREE PARSER**

- Actions being triggered on the abstract syntax tree
- Where most detailed information can be found
- Recursive structure of SymbolScope
  - Top level element is SymbolScopeRoot
SYMBOL SCOPE

- Each symbol scope contains:
  - Table buffers, unnamed or not,
  - Variables
  - Widgets
  - Call list
  - Child scopes
DIFFERENCES BETWEEN VERSIONS

- Using Maven build system instead of Ant
- A lot of code cleanup
  - But you may not like it
- No ABL bridge anymore
  - Performance were horrible
- Might be back, if somebody’s willing to maintain it!
NEXT DEVELOPMENTS

- Deprecate ANTLR 2
  - ANTLR 4 is the current version
  - Easier said than done
- Allow listeners to be implemented
- Full access to r-code structure in the propath
- ETA: end of Q4
INTEGRATION FROM CUSTOM SONAR RULE

- Custom rule has access to the ParseUnit
  - `public void lint(ParseUnit unit)`

- ParseUnit gives access to:
  - AST: `public ProgramRootNode getTopNode()`
  - SymbolScope: `public SymbolScopeRoot getRootScope()`
REPORT ISSUES

- From a node:
  - `public void reportIssue(JPNode node, String msg)`

- From a line number:
  - `public void reportIssue(int lineNumber, String msg)`

- On any include file:
  - `public void reportIssue(String fileName, int lineNumber, String msg)`