

ooRexxPlugin for IntelliJ IDEA

Alexander Seik

Agenda

- Motivation
- Structure of the Plugin
- Demo
- Standalone
- Outlook

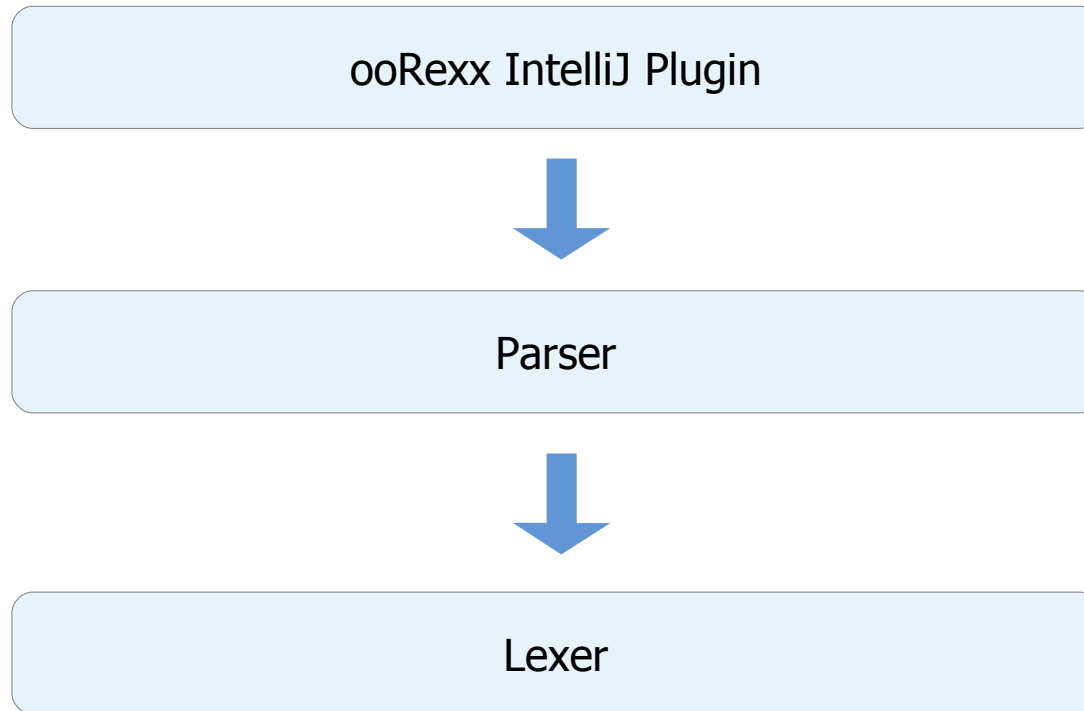
Motivation

- Help beginners
- Support programmers
- Create a reusable parser
- Personal: gain knowledge of ooRexx and IntelliJ

Prerequisites

- Get and install the free community edition of IntelliJ IDEA
 - <https://www.jetbrains.com/idea/download/>
- Get and install the latest beta version of the ooRexx plugin
 - File name: [ooRexxPlugin4IntelliJ.jar](#)
 - <https://sourceforge.net/projects/bsf4oorex/ files/Sandbox/aseik/ooRexxIDEA/beta/>
 - Follow the [README.txt](#) instructions
- Get and install the standalone parser with nutshell examples
 - File name: [OoRexxStandaloneExample.zip](#)
 - <https://sourceforge.net/projects/bsf4oorex/ files/Sandbox/aseik/ooRexxIDEA/beta/>
 - Unzip this archive file

Structure of the Plugin, 1



JFlex, 1

- Tokenization of a text input
- Rules are defined in JFlex Syntax
- File: [ooRexx.flex](#)
- Built with JFlex (<http://www.jflex.de/>)
- JFlex generates an optimized Java Lexer class

- Defining a macro

IDENTIFIER = `[a-zA-Z_\\?\\!\\.]+[0-9a-zA-Z_\\?\\!\\.]*`

- Applying a macro

`{IDENTIFIER} / "(" {return FUNC_IDENTIFIER;}`

`/ "("` defines a predicate

if immediately followed by an open parenthesis "(" a `FUNC_IDENTIFIER` token gets returned

BNF, 1

- Uses tokens as input
- ooRexx Syntax defined in Backus–Naur form (BNF)
- File: [ooRexx.bnf](#)
- Parser-Generator:
 - Grammar-Kit (<https://github.com/JetBrains/Grammar-Kit>)
- Output: Abstract Syntax Tree (AST)

BNF, 2

- `file ::= shebang? (new_line | instruction)* (directive)*`
 - `file`-rule is the root of an ooRexx AST
 - An ooRexx file can have a leading `shebang` line, e.g.
`#!/usr/bin/rexx`
 - `0..*` new lines or instructions
 - `0..*` directives

"plugin.xml", 1

- Configures the ooRexx support
 - plugin.xml
 - Configuration file
 - Maps Java classes that carry out a predefined (editor) function
- Defines different tools for the plugin
 - e.g. for ooRexx keyword highlighting

```
<annotator language="ooRexx"
implementationClass="org.oorexx.ide.intellij.annotator.KeywordHighlighter"/>
```
 - or for folding of ooRexx directives

```
<lang.foldingBuilder language="ooRexx"
implementationClass="org.oorexx.ide.intellij.folding.FoldingBuilder"/>
```

"plugin.xml", 2

- Current state: beta
- Focus of beta
 - Correct ooRexx 5.0 parsing
 - Syntax highlighting
 - Code folding
 - Run via IDE
 - Toggle comments for multiple lines
 - Option for special characters

Demo

Standalone, 1

- Use the Java parser without an IDE
- Get an AST from an Rexx/ooRexx input file or text
- Makes it easy to build new applications based on the AST with BSF4ooRexx and ooRexxPlugin4IntelliJ, e.g.

```
file = .bsf~new("java.io.File", rexxFile)
psiFile = bsf.loadClass("org.ooRexx.ide.standalone.OoRexxStandaloneParser")~parse(file)
root = psiFile~getNode~getPsi -- variable "root" is the AST root
... code to process the ooRexx parse tree using its root now possible ...
```

- Drawback: it is not completely independent of IntelliJ
- Next goal: make the library smaller

Standalone Nutshell Examples, 1

- After unzipping `OoRexxStandaloneExample.zip`
 - Change into the directory `OoRexxStandaloneExample`
 - Java archives ("`jar`" files) containing the necessary code
 - ooRexx 5.0 parser: `lib\ooRexxPlugin4IntelliJ.jar`
 - Java archives needed: `lib\ooRexxStandaloneAdapter.jar`
 - Add to `CLASSPATH` environment variable, so Java can find them
 - After changing into the `lib` directory on Windows:

```
set CLASSPATH=%CLASSPATH%;%CD%\ooRexxPlugin4IntelliJ.jar
set CLASSPATH=%CLASSPATH%;%CD%\ooRexxStandaloneAdapter.jar
```
 - After changing into `lib` directory on Unix:

```
export CLASSPATH=$CLASSPATH:`pwd`/ooRexxPlugin4IntelliJ.jar
export CLASSPATH=$CLASSPATH:`pwd`/ooRexxStandaloneAdapter.jar
```

Standalone Nutshell Examples, 2

- Change into the directory `OoRexxStandaloneExample` (continued)
 - Change into the directory `resources\examples`
 - Contains Rexx nutshell examples that use the ooRexx 5.0 parser
 - The directory `resources\examples\filesToParse` contains small Rexx programs that can be parsed using the nutshell examples
 - Example
 - Windows

```
rexxj CountKeywords.rxj filesToParse\example1.rexx
```
 - Unix

```
rexxj.sh CountKeywords.rxj filesToParse/example1.rexx
```

Demo Standalone

Demo

Outlook

- Bug fixes → Release first version
- Next version
 - ooRexxDoc integration
 - Refactoring
- In the future
 - ooRexx-file management
 - Code completion