"ooRexxDoc
A Tool for Generating Javadoc-Like Documentation for ooRexx Programs"

International 2014 Rexx Symposium
Memphis, Tennessee

© 2014 Rony G. Flatscher (Rony.Flatscher@wu.ac.at)
WU Vienna, Austria (http://www.wu.ac.at)
Overview

• Documentation of programs

• A ski seminar project
  – Alexander Seik
  – oRexxDoc
    • Principles
    • GUI
    • Configuration file

• Demonstration
Documentation of Programs, 1

• „Real“ programmers do not like to document!
• „Rexx programs can be read like English, hence no need to document!“
• Programs are usually written under time pressure
  – No time planned for extensive documentation
  – While programming everything seems to be clear, why waste time to document?
• How about
  – Needing to change Rexx programs years later?
  – Other programmers trying to extend (or fix) your program?
• The Java folks came up with a nice idea
  – Some documentation has to be done anyway, so some documentation is always present
  – Make it easy for „real“ programmers to write documentation
    • Write a tool („javadoc“) that scans the Java source code and creates automatically a documentation
    • Allow Java programmers to easily mark comments that should be added to the documentation
    • The resulting documentation should be usable via a normal web browser
      – Documentation can be placed on a server or locally
A Ski-seminar Project

• Find a student who
  – Knows Java and JavaDoc
  – Knows ooRexx

• Task to tackle
  – Study JavaDoc and come up with a documentation utility that does the same for ooRexx programs

• Found student
  – Alexander Seik
    • Works as a Java programmer
    • Enrolled to both of my „Business Programming“ classes at WU Vienna
Principles, 1

• Rexx files get analyzed
  – Routines get identified
    • Arguments
    • Return value
  – Directives get identified
    • Subdirectives get extracted
  – Dependencies among Rexx programs („packages“)
  – „ooRexxDoc-comments“, if available
• Create JavaDoc-like HTML-based documentation
Principles, 2

- Ad „ooRexxDoc-comments“
  - Start with /** and ends with */
  - May contain annotations led in with @
    - @author
    - @condition
    - @deprecated
    - @param
    - @return
    - @see
    - @since
    - @version
GUI, 1

![ooRexxDoc](image)

- **Title**
- **Input (required)**
- **Output (required)**
  (Directory will be created if it does not exist. Content will be appended to an existing directory)
- **Overview Text**
  (Text will be displayed on the start page)
- **Info-Properties**
  (E.g. Author: Rexxler)

Buttons:
- Add
- Remove
- Browse
- Create Doc
GUI, 2

![GUI Interface](image)

- **Title**
- **Input (required)**
- **Output (required)**
  - Directory will be created if it does not exist. Content will be appended to an existing directory.
- **Overview Text**
  - Text will be displayed on the start page.
- **Info-Properties**
  - E.g. Author: Rexxler

Buttons:
- **Add**
- **Remove**
- **Create Doc**
<?xml version="1.0" encoding="UTF-8"?>
<oorexxdoc>
  <parser>
    <input></input>
    <output></output>
  </parser>
  <title></title>
  <header>
    <property>
      <key></key>
      <value></value>
    </property>
    <property>
      <key></key>
      <value></value>
    </property>
  </header>
  <overview>Sample Text</overview>
</oorexxdoc>
Demonstration

• Prerequisite
  – Java
  – http://sourceforge.net/projects/oorexxdoc

• Program name: ooRexxDoc.jar
  – „jar“: Java archive, a zip archive

• GUI
  – Double click on jar file

• Command line
  – java -jar ooRexxDoc.jar
    • Uses default configuration file named ooRexxDoc.xml
  – java -jar ooRexxDoc.jar myConfigurationFile.xml