

ooRexx – Version 4.0

GSE WG Rexx plus ISPF

Prof. Dr. Rony G. Flatscher

ooRexx 4.0 - Veröffentlichung

- Offiziell am 17. August 2009 veröffentlicht
- Erzeugte Installationspakete für
 - AIX 5.2/5.3/6.0 (32 und 64 bit)
 - Debian Linux 4.0 (i386 und x86_64)
 - Fedora 10/11 (i386)
 - RedHat Linux 5 (i386 und x86_64)
 - SUSE Linux 10/11 (i386 und x86_64)
 - Ubuntu Linux 6.04/7.04 (i386 und x86_64)
 - Windows 2000/XP/Vista (32 und 64)

- **ooRexx**

- <<http://www.ooRexx.org>>
 - Homepage "Open Object Rexx (ooRexx)"
- <<http://sourceforge.net/projects/oorexx/>>
 - Projektverwaltungs-Seite (Sourceforge)
- <<http://www.oorexx.org/download.html>>
 - Bessere Download-Seite als Sourceforge !
- <http://wi.wu-wien.ac.at/rgf/rexx/misc/ecoop06/ECOOP2006_RDL_Workshop_Flatscher_Paper.pdf>
 - Übersichtsartikel über die Geschichte und Konzepte von ooRexx

ooRexx 4.0 – Kernel, 1

- Kernel
 - Vollständig neu geschrieben
 - Arbeiten daran haben bereits vor fünf (!) Jahren begonnen!
 - Erstmals für 32-Bit *und* 64-Bit kompilierbar
 - Dateien können größer als vier GB sein
 - Neue APIs für C++ Programmierer
 - Mehrere Rexx-Interpreter-Instanzen parallel ausführbar
 - **.local** pro Rexx-Interpreter-Instanz geführt
 - Pro Rexx-Interpreter-Instanz beliebig viele Rexx-Programme ausführbar

ooRexx 4.0 – Kernel, 2

- Neue, objektorientierte APIs für die Interaktion mit Rexx-Objekten
 - JNI (Java Native Interface) ähnliche Architektur
 - Mehr als über 100 APIs verfügbar
 - C++ Programme können direkt mit ooRexx-Objekten interagieren
 - ooRexx-Objekte anlegen
 - ooRexx-Nachrichten schicken
 - ooRexx-Ausnahmen aufwerfen
 - Funktionen (Routinen) und Methoden in C++ implementierbar!

ooRexx 4.0 – Sprache, 1

- Neues Umgebungssymbol **.context**
 - Erlaubt Zugriff auf Kontextinformationen
 - args
 - condition
 - digits
 - **executable**
 - form
 - fuzz
 - line
 - **package**
 - rs
 - variables

ooRexx 4.0 – Sprache, 2

- Neue Klassen
 - Für die Nutzung von C++ aus
 - **.Buffer**, **.Pointer**, **.RexxContext**, **.WeakReference**
 - Für ooRexx-Programmierer
 - **.IdentityTable**
 - **.Orderable**
 - Implementiert alle Vergleichsoperationen
 - Als "MIXINCLASS" definiert, daher mit "INHERIT" einbindbar

- Neue Klassen (Fortsetzung)
 - **.Package**
 - Erlaubt das dynamische (zur Laufzeit) Erzeugen eines Programms
 - Repräsentiert Programmdatei, erlaubt Inspektion zur Laufzeit
 - Erlaubt das Erzeugen und Suchen von definierten Klassen und Routinen
 - Verwaltet, welche Klassen und Routinen sichtbar sind
 - **.Routine**
 - Erlaubt das dynamische (zur Laufzeit) Erzeugen einer Routine
 - Ermöglicht das Aufrufen mit `call()` bzw. `callWith()`

- Neue Direktiven
 - **::CONSTANT**
 - Erlaubt das Definieren von Konstanten für Klassen
 - Sind verfügbar sowohl über das Klassenobjekt als auch über die Instanzen der Klasse
 - **::OPTIONS**
 - Erlaubt das Setzen von Optionen, die für die gesamte Programmdatei gelten!
 - DIGITS
 - FORM
 - FUZZ
 - TRACE

ooRexx 4.0 – Sprache, 5

- Neue Methoden
 - .Object
 - Send
 - SendWith
 - StartWith
 - .DateTimeClass
 - offset
 - toTimeZone
 - utcDate
 - utcISOdate

ooRexx 4.0 – Sprache, 6

- Neue Methoden (Fortsetzung)
 - .Method
 - package
 - .MutableBuffer
 - Hat mittlerweile fast alle Methoden von .String
 - .RexxQueue
 - empty
 - makeArray
 - exists
 - open

ooRexx 4.0 – Sprache, 7

- Neue Methoden (Fortsetzung)
 - .Stem
 - toDirectory
 - .String
 - replaceAt

ooRexx 4.0 – Windows-Unterstützung, 1

- `.OLEObject`
 - Windows-Ereignisbehandlung stark verbessert
 - `addEventMethod`, `removeEventMethod`
 - `connectEvents`, `disconnectEvents`
 - `isConnectable`, `isConnected`
 - `removeEventHandler`
- `ooDialog`
 - Auf den neuesten Stand gebracht
 - Verwendet die neuesten Windows-Renderings
 - Erweitert um neue Klassen (Dialoge)
 - Neue Dokumentationen (`ooDialog` Reference, insb. Kapitel 1.2, 2.4)

ooRexx 4.0 – Windows-Unterstützung, 2

- Windows Script Host (WSH, ActiveX)
 - Wegen des neuen Kernels **nicht mehr enthalten** !
 - Daher ist folgendes nicht mehr möglich
 - WSH/ActiveX (z.B. DHTML)
 - WSC
 - WSF
 - Versprechen, dass diese Funktionalität nach dem Erscheinen von ooRexx 4.0 nachprogrammiert wird
 - Derzeit *leider* keine Anzeichen dafür!
 - Muss von Rexx-Programmierern wahrscheinlich laufend aktiv nachgefragt und eingefordert werden

ooRexx 4.0 – Weitere Neuerungen, 1

- BIFs ("built-in functions")
 - Neues ANSI-BIF `QUALIFY(dateiname)`
 - `FILESPEC()`, neue Argumente
 - "L[ocation]", "E[xtension]"
 - `TIME()`, neues Argument
 - "O[ffset]"
 - `LASTPOS()`, `POS()`, neues Argument
 - Länge
 - `TRANSLATE()`, neue Argumente
 - Startposition, Länge

ooRexx 4.0 – Weitere Neuerungen, 2

- Neue Schlüsselwörter bei Direktiven
 - **EXTERNAL** für **::ATTRIBUTE**, **::METHOD**, **::ROUTINE**
 - Implementierung liegt in Maschinencode vor (z.B. in C++)
 - **LIBRARY** für **::REQUIRES**
 - Externe Funktionsbibliothek nach den neuen ooRexx 4.0 API-Konventionen erstellt

ooRexx 4.0 – Weitere Neuerungen, 3

- Neues bzw. geändertes Verhalten
 - **::REQUIRES**-Direktive
 - Wird nur mehr einmal pro Rexx-Interpreter-Instanz initialisiert
 - Ansonsten nur mehr wiederverwendet
 - Private Methoden
 - Können nunmehr auch von Objekten, die aus derselben Klasse stammen (und dem Klassenobjekt selbst), angesprochen werden
 - **::ATTRIBUTE**-Direktive
 - Fehler behoben, nunmehr von Haus aus **GUARDED**

ooRexx 4.0 – Weitere Neuerungen, 4

- Neue ooRexx-Klassenbibliothek "[socket.cls](#)"
- Zahlreiche neue Beispiele im Verzeichnis "[samples](#)"
 - ooDialog
 - Socket-Programme
- Für Windows gibt es nunmehr auch ein "GUI-rexxtry"
 - [ooRexxTry.rex](#)

Zusammenfassung

- ooRexx 4.0 seit August 2009 verfügbar
- Neuer Kernel
 - Stabiler
 - Plattformunabhängig
 - Neues, mächtiges OO-API für C++ Programmierer
- Interessante neue Klassen
- Hilfreiche, neue Methoden
- Karten neu gemischt, da eine stabile und zukunftssichere Quellcode-Basis, die auch von Dritten verstanden werden kann
 - Das 1. Mal, dass ich ooRexx aus betriebswirtschaftlicher Sicht uneingeschränkt empfehlen kann!

Geschichte "Open Object Rexx (ooRexx)", 1

- Ende der 80er Entwicklungsarbeiten und Experimente für einen objektorientierten Nachfolger von Rexx
- Leitung: Simon Nash, IBM England, Arbeitskollege von und gut befreundet mit Mike F. Cowlshaw, dem "Vater" von Rexx
- OO-Sprache Smalltalk eines der Vorbilder
- In C++ zu einer Zeit implementiert, als C++ noch nicht standardisiert war !
 - Jede nennenswerte Firma entwickelte eigene Versionen von C++-Compilern

Geschichte "Open Object Rexx (ooRexx)", 2

- IBM verlagert die Entwicklung des geplanten Produktes "Object REXX" in die USA unter der Leitung von Rick McGuire
 - Code wurde praktisch von Grund auf neu geschrieben
 - Rick McGuire als "Vater" dieser Version ist der einzige Mensch, der die Architektur und Zusammenhänge des Interpreters kennt
 - Version für OS/2 (1997 als Bestandteil von OS/2 Warp 4 veröffentlicht)
 - Kaufversionen für AIX und Windows

Geschichte "Open Object Rexx (ooRexx)", 3

- Herbst 2004: Verhandlungen der "Rexx Language Association (RexxLA)" mit IBM führte zur Übertragung des Quellcodes von Object REXX an die RexxLA
 - RexxLA-Verhandler: Pam Taylor (SW-Industrie, USA), Mark Hessling (Regina-Maintainer, Australien), Rony G. Flatscher (Universität, Europa)
 - IBM: Manfred Schweizer (Böblingen), Manager REXX
 - Konnte sicherstellen, dass Rick McGuire im Opensource-Format (nicht als Angestellter von IBM, sondern ausschließlich in seiner Freizeit) die Federführung bei der Weiterentwicklung von Object Rexx übernimmt

Geschichte "Open Object Rexx (ooRexx)", 4

2004-12-14, e-Mail von Manfred Schweizer (IBM) an Chip Davis (Präsident, RexxLA):

To President of RexxLA Chip Davis

Dear Chip,

I like to inform you officially, that IBM REXX Development have handed over the source code for:

Object REXX Interpreter Edition for Windows,

Object REXX for AIX,

Object REXX for SUN/Solaris, and

Object REXX for Linux

to your organization. It is now the responsibility of RexxLA to publish and maintain this code.

I am convinced, that the REXX language and the REXX users will benefit from this transfer. Please let me say thank you to you and to all the people in your organization and within IBM who worked on the realization of this transfer.

Mit freundlichem Gruß

Manfred Schweizer

Mgr. REXX and Office Solutions, WSS Boeblingen, Germany

Geschichte "Open Object Rexx (ooRexx)", 5

- April 2005, erstmals Veröffentlichung der opensource Version "Open Object Rexx (ooRexx) 3.0.0"
 - Bereinigt um bekannte Fehler von IBM's Object REXX
 - Aufgrund von Lizenzproblemen enthält ooRexx nicht
 - Die externen Rexx-Funktionen von "rexftp", stattdessen eine (funktionsreichere, multithreadable) ooRexx-Klassenbibliothek
 - Die Windows-GUI-Entwicklungsumgebung
- Weitere Veröffentlichungen
 - ooRexx 3.0.1: August/September 2005
 - ooRexx 3.1.0: August 2006
 - ooRexx 3.1.1: November 2006
 - ooRexx 3.1.2: April 2007
 - ooRexx 3.2.0: November 2007