REXX for CICS/ESA

Bob Vogel
IBM Dallas

Pages 252-272

REXX Symposium
REXX for CICS

Bob Vogel

May 3, 1995

Contents

Introduction

IBM

- Copyright

(C) Copyright IBM Corporation 1993, 1995

- Trademarks

The following terms used in this paper are trademarks or service marks of IBM Corporation in the United States or other countries:

CICS/ESA, IBM, MVS/ESA, OfficeVision, OS/2

What is "REXX for CICS/ESA"

IBM

- Two products (GA 7/29/94)
  - REXX Development System for CICS/ESA (5655-086)
  - REXX Runtime Facility for CICS/ESA (5655-087)
- REXX language support for CICS/ESA
- EXEC CICS Command support from REXX
- CEDA and CEMT REXX interfaces
- REXX-DB2 Interface
- Native CICS application environment
  - REXX Panel Facility
  - High-level file system & filelist utility
  - Text Editor
  - Interactive shell
  - Open Application Integration facilities
- High-level Client/Server support
- And More

The REXX Language

IBM

- Created by Mike Cowlishaw, at IBM Hursley

- In ANSI X3J18 committee since 1991, target for standard is 1995

- Strengths of REXX
  - Natural / high-level
  - Avoids unnecessary detail
  - Typeless
  - Strong parsing
  - Command and function support
  - Source level interactive tracing
  - Complete set of modern programming constructs
  - Fairly small language, easy to learn
  - Rich set of functions
  - Can be interpreted or compiled

Trends toward REXX popularity

IBM

- Widespread use of REXX under OS/2
  - Now in PC DOS 7.0
- REXX moving aggressively to new platforms
- Shift to very high level languages / devp systems
- Macro support taking off industry wide
- ANSI REXX effort progressing well
- REXX compilers
- Dramatic increases in computing power (improves REXX performance)
- Shift to new system architectures, where REXX is a natural
  - Client/Server computing
  - Workstation GUI to Enterprise data/appls (Visual REXX)
  - Object Oriented (OO REXX)
  - Messaging and Queueing (Workflow Scripts)

Shift to very high level languages

IBM

- Highly competitive times demand higher productivity
- Large numbers of non-DP pros coming on board
- Alignment of programming with business organization
- More complex systems difficult to develop & maintain
- Prototyping Development Methodology has come of age
- Building block approach and code reuse popular
- REXX and BASIC beefed up for serious programming

What were design goals for REXX/CICS

- Deliver a strong productivity tool
- Create a serious REXX-based application environment
- Make REXX work with CICS languages and facilities
- Provide a native prototyping, development and customization environment
- Common REXX support across CICS platforms
- Provide high-level Client/Server interfaces
- Utilize the power of REXX in an open application integration platform

Project history

IBM

- REXX prototype to IBM Program Product
  - From Assembler to PL/X for portability
  - FROM TSO/E REXX base to direct use of REXX kernel
  - From 1 person research project to formal development team

Why Now

- Growing exposure to REXX and its power
- Growing emphasis on productivity
- Product requirements for REXX under CICS
- Opportunity to improve a very important environment
- Enhance customers' large mainframe investment
  - REXX for CICS actually introduces some of the concepts of personal computing into the MVS/CICS environment.

Highlights

- Full REXX 3.48 language support under CICS
- Dynamic EXEC CICS command level support
- REXX interface to CEDA, CEMT
- DB2 Interface (SQL statements & DB2 commands)
- CICS native text editor for REXX execs and data
- High-level VSAM-based REXX file system (RFS)
- Execs may also be run from MVS Partitioned Datasets
- High-level Panel I/O facility
  - Also supports BMS

Function Overview (continued)

IBM

- Support for REXX Subcommands (themselves written in REXX)
- Pseudo-conversational support (conventional and auto)
- System and user profile exec support
- Shared execs in storage (via EXECLOAD & EXECDROP)
- High-level Client/Server interfaces
- Online help and softcopy REXX/CICS manuals
- Improved run-away REXX task management
- Concurrent international language support (English + 6)
  - German, Spanish, French, Canadian French
  - Japanese Kanji, Simplified Chinese

Need for REXX/CICS

- As a tool to streamline support staff activities
  - CICS Systems Programmers and Administrators
  - DB2 Analysts
  - CICS and DB2 testers, other support staff
- More productive CICS application development
  - Native CICS development (simpler)
  - Enjoy the strengths of REXX under CICS
- More flexible, powerful product customization & extension (macros)
- Quick prototyping and procedural language functions
- Preserve REXX investments in migrations
- Needed for products with REXX requirements
- As a script language to automate/streamline development sequences
- Help enable enterprise-wide Client/Server computing
- Better enable CICS end-user computing
- CICS Application Integration
  - Glue language to tie the pieces together
  - Building block support

REXX File System (RFS) Features

- Hierarchical Directory structure (like OS/2, AIX)
- VSAM based
- No need to register most new users
- No need to register individual EXECs
- Import/Export from/to MVS Partitioned Datasets
- Management functions for members (COPY, DELETE, RENAME)
- FLST file directory interface utility
- An EXECIO-like I/O utility (RFS)
- VSAM datasets can be added to a Filepool dynamically
- Number of filepools only limited by DASD

Editor features

- Two personalities
  - XEDIT
  - ISPF
- RFS and PDS file support
- Terminal models 2, 3, 4 & 5 supported
- Customizable
- REXX macro support
- Execs can be run without leaving editor

Security features

- CICS security facilities (via ESM) to control access
- REXX/CICS Authorized Command support
- REXX/CICS Authorized Library support
- REXX/CICS Authorized User support
- Security exits
- RFS AUTH command for directory sharing

Performance

IBM

- REXX/CICS interpreter uses sophisticated performance techniques
- Majority of execution time usually not in language processing
- Shared and Reentrant code / execs
- Performance numbers, courtesy of Steve Ware, University of Florida on WWW (see last page for Web address)
- REXX/CICS run-time support for compiled REXX/CICS execs a possibility

EXEC CICS commands not supported

IBM

- HANDLE ABEND
- HANDLE AID
- HANDLE CONDITION
- IGNORE CONDITION
- PUSH HANDLE
- POP HANDLE

REXX/CICS Summary

- REXX Development System for CICS/ESA much more than another language
- REXX/CICS introduces significant new capability
- REXX/CICS provides new approaches to CICS computing
- REXX/CICS opens CICS to a broader range of uses
- REXX/CICS is a strong productivity tool for devp and support
- REXX/CICS is a good application integration platform
- REXX/CICS is useful for serious programming
- REXX/CICS is natural for Client/Server computing
- REXX/CICS is in step with industry trends (application server)
- CICS and REXX are very synergistic
  - REXX = ease of use, high productivity, native devp env.
  - CICS = production computing and common support

Questions and Wrapup

- Future direction
  - Runtime Lite
  - Compiler Support
  - TCP/IP Sockets
- How to get more information on REXX or REXX/CICS
  - http://rexx.hursley.ibm.com/rexx/
  - http://sfware.nerdc.ufl.edu/rexxcics/rxkixhom.html
  - dshriver@vnet.ibm.com
- Questions