

# REXX EXTENSIONS FOR OS/2

CHARLES DANNEY  
Quercus

## A General-Purpose REXX Extension Package

REXX Symposium  
May, 1993

Charles Daney  
Quercus Systems  
P. O. Box 2157  
Saratoga, CA 95070

(408) 867-7399  
Fax: (408) 867-7489  
BBS: (408) 867-7488  
CompuServe: 75300,2450  
Internet: 75300.2450@compuserve.com

## Functions for compound variable and array handling

- Array handling
  - copying from one array to another (overlay)
  - copying from one array to another (insert)
  - deleting portions of an array
  - sorting portions of an array
- Compound variable handling
  - copy all elements
  - save all elements to disk
  - restore all elements from disk
  - find all tails (with optional pattern match)
  - find all values (with pattern match)
- Groups of variables
  - write group to disk
  - read group from disk
  - dump group for debugging

## Motivation

- Make handling of various kinds of aggregates easier
  - lists
  - sets
  - collections
  - ordered pairs
  - ordered triples
- Allow easier simulation of data structures
- Treat compound variables or groups of variables as a "database"
  - defaults & user program configuration
  - representation of program state
  - externalize setup of data tables
  - "true" databases (e. g. user directory)
- Facilitate exchange of data between programs
- Faster loading of data

## "Quasistem.:"

- REXX language says that stems contain only final period
- REXX users use compound names hierarchically
  - pet.katie.rabbit = 'Flopsy'
  - pet.katie.cat = 'Fluffy'
  - pet.katie.dog = 'Fido'
- Users expect "drop pet.katie." to behave like a stem (without affecting pet.lisa, pet.susy, etc.)
- Similarly for other operations like copy, read, write
- When used carefully, this seems useful and internally consistent

## Conventions of library functions

- Quasistems allowed in context where stem is expected
- Final period is assumed in a stem context
- Substitution is not performed on quasistem components
- Case is significant in all but first component of quasistem

## Array conventions

- Common convention is that arrays are integrally subscripted
- First array element is 1
- Zeroth element is number of array elements
- The array stem may be a quasistem

## Array operations

- Copy - ARRAYCOPY
  - from-position, to-position, count are options
  - elements of target array are overlaid
- Insert - ARRAYINSERT
  - from-position, to-position, count are options
  - elements of target after insertion point move
- Delete - ARRAYDELETE
  - from-position, count are options
  - remaining elements shift position to eliminate gap
- Sort - ARRAYSORT
  - from-position, count are options
  - start, length, order, type specified for each field

## Compound variable operations

- Copy - CVCOPY
  - Makes exact copy of a compound variable with different stem
  - Target is dropped first
- Write to file - CVWRITE
  - Existing file is erased
  - File contains tails only, not "stem"
- Read from file - CVREAD
  - Target is dropped first
  - File contains tails only, not "stem"
- List compound variable tails - CVTAILS
  - Creates an array with all tails
  - Enables iteration on all tails
  - Regular expression pattern match is optional
  - Case sensitivity is optional in pattern match
- Search compound variable values - CVSEARCH
  - Creates an array with tails of matches
  - Matching by regular expression
  - Case sensitivity is optional

## Regular expressions

- \ - escapes special characters
- ^ - matches beginning of string
- \$ - matches end of string
- . - matches anything but newline
- :a - matches alphabetic character
- :d - matches digits 0 - 9
- :n - matches alphabetic & digits
- \* - matches 0 or more of expression
- + - matches 1 or more of expression
- ? - matches exactly 0 or 1 or expression
- [ ] - list of matching characters

## Regular expression examples

- fido - matches string "fido" anywhere
- ^fido\$ - matches only "fido" by itself
- [01234567] - matches only valid octal characters
- [01234567]\* - matches string of valid octal characters
- [^ABC] - matches anything but A, B, C

## ∞ Variable group operations

- Write to file - VARWRITE
  - Existing file is appended
  - All variables in program may be written
  - Optional list of variables & stems to include or exclude
  - Stems are kept in the file
- Read from file - VARREAD
  - Existing stems aren't dropped - data is merged
  - All variables in file may be read
  - Optional list of variables & stems to include or exclude
- Dump variables - VARDUMP
  - Like VARWRITE except output is formatted for visual inspection