REXX Utilities for your Windows PC

How to get more out of your Windows system using Object REXX function packages!

Speaker: Christian Michel (cmichel@de.ibm.com)
REXX Development, IBM Germany

The 10th International Rexx Symposium
3-5 May1999, Jacksonville/Florida

Available Function Packages

- Included with Object REXX:
  - REXXUTIL: System utilities, file handling
  - RxSock: TCP/IP socket communication
  - RxFtp: TCP/IP file transfer protocol
- Included with other IBM products:
  - DB2 interface (since DB2 2.1.2)
  - EHLLAPI interface (since PCOM 4.2)
- Other function packages:
  - REXX/SQL by Mark Hessling (other databases, including ODBC)

Predefined Classes (1)

- .WindowsRegistry class
  - Read/write registry keys
- .WindowsProgramManager class
  - Create program groups, shortcuts
- .WindowsEventlog class
  - Read/write/clear NT event log records
- .WindowsClipboard class
  - Copy/Paste/Clear text data in clipboard

Predefined Classes (2)

- .WindowsManager class
  - Find/manipulate windows on the screen by title
- .WindowObject class
  - Object REXX representation for window objects
  - Move/size/hide/query individual windows
- .MenuObject class
  - Select menu items programatically

Date Conversion Utilities

- Object REXX supports new DATE() function with 5 arguments:
  - Convert dates between different formats
  - 2 digit years use sliding window technique
  - Specify separators used in input/output formats
- Calculate someone's age in years:

```
Age = (Date('B') - Date('B','19680107','S'))/365.25
```

- Convert yymmdd to yyyy-mm-dd:

```
NewFormat = Date("S", "990225", "O", "-", "")
```

REXXUTIL Library (1)

- File system functions:
  - list files, search files, delete files, get drive information, etc.
- System information functions:
  - Get boot drive, system directory, Windows version
- Semaphore functions for process synchronization
- Macrospace functions:
  - Load/save/reorder/drop REXX macros

REXXUTIL Library (2)

- Upcoming additions:
  - Get/set file dates (date of creation, last update, last access)
  - Functions to handle stem based arrays (copy, delete, insert, sort)
  - Dump all visible variables to console or file

REXXUTIL Samples (1)

- **Find all .JPG files on disk and sort them descending by date/size:**

```
/* Find all *.JPG files in the current path with 4 digit dates */
Call SysFileTree "*.JPG", "Files.", "SL"

/* sort the stem entries descending (date starts in first column) */
Call SysStemSort "Files.", "D"
Do i = 1 To Files.0
  Say i Files.i
End

/* sort the stem entries descending by file size (size is in column 21 to 31) */
Call SysStemSort "Files.", "D",,,, 21, 31
Do i = 1 To Files.0
  Say i Files.i
End
```

REXXUTIL Samples (2)

- **Get a list of all files on local drives:**

```
/* List all files on all local drives */
Numeric Digits 12                       /* we might encounter large numbers */
AllDrives = SysDriveMap(, "LOCAL")
AllFiles.0 = 0; OverallSize = 0         /* initialize list and size of all files */

/* Scan all drives and add file entries to AllFiles. stem */
Do i = 1 To Words(AllDrives)
  Call SysFileTree Word(AllDrives, i) || "\*.*", "Files.", "SL"
  Call SysStemCopy "Files.", "AllFiles.", 1, AllFiles.0 + 1
End

Do i = 1 To AllFiles.0
  Say i AllFiles.i
  OverallSize = OverallSize + SubStr(AllFiles.i, 21, 11)
End

Say "Size of all listed files:" OverallSize "bytes."
```

RxSock Library

- Wrapper for TCP/IP socket functions
- Compatible with OS/2 library
- Custom programs can connect to:
  - Web server
  - News server
  - Mail server
  - Custom applications
- Tutorial on REXX sockets:

  http://www2.hursley.ibm.com/rexxtut/socktut1.htm

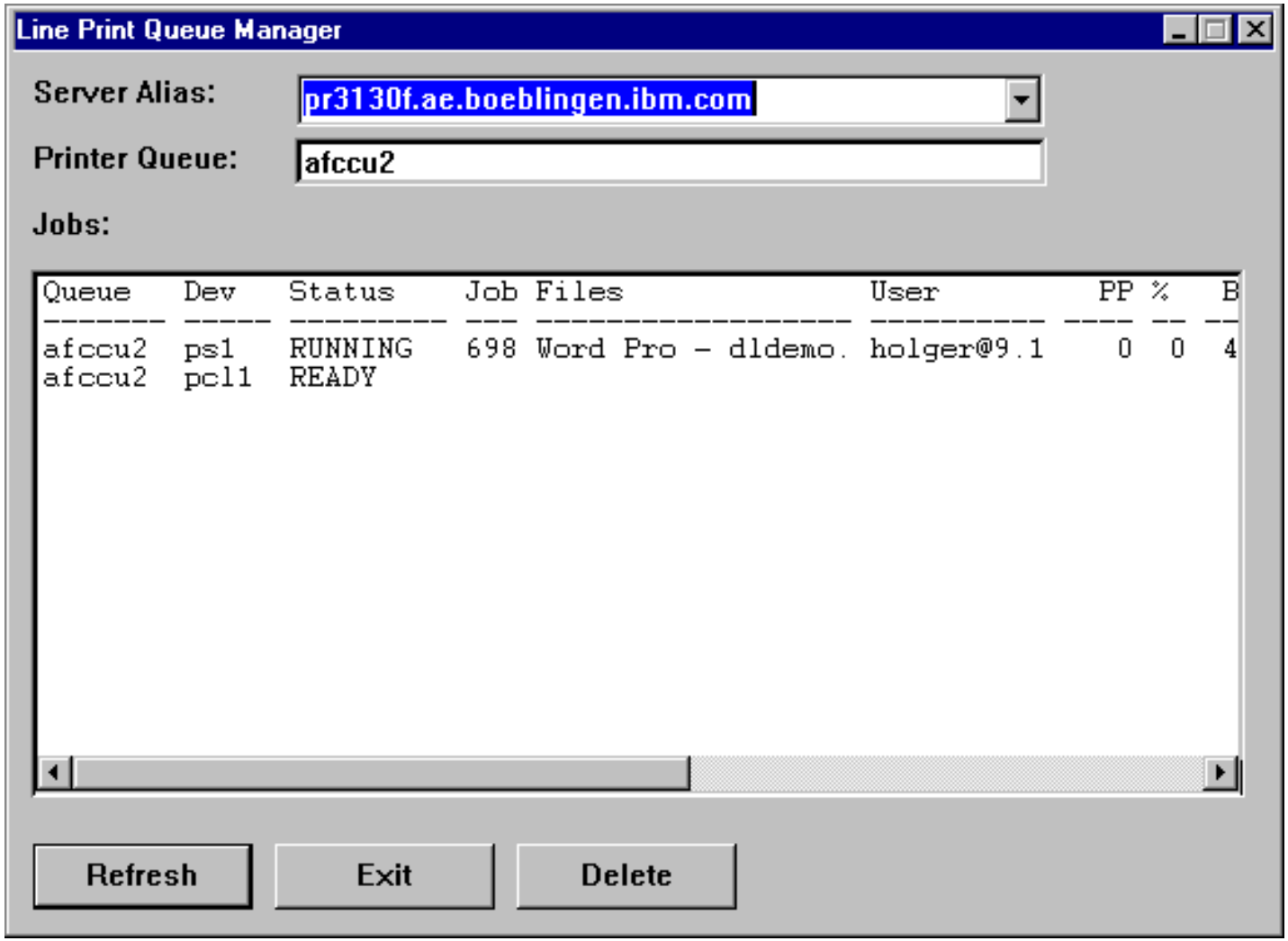
RxSock Sample - LPQMgr (1)

- Queue manager for LPR print queues
- Show jobs on LPR print queues in GUI
- Delete jobs from print queue
- Select from a list of print servers
- Customizable through profile (define print servers and queues)
- First freeware program of its kind for Windows platforms
- Only 250 lines of REXX code!

RxSock Sample - LPQMgr (2)

- Use DEFAULTSERVER to identify printer that will be queried on startup
- PRINTER defaults to "afccu2", AGENT defaults to a string built from USERNAME and COMPUTERNAME variables (or local IP address)

LPQMgr - Screenshot



RxSock Sample - SmtpMail

- A sample implementation for the SMTP protocol
- Sends internet/intranet mail from a REXX program
- Useful for problem notification to administrators of unattended systems

RxFtp Library

- Transfer files using FTP protocol
- Compatible with OS/2 library
- Recent Windows additions:
  - Added PASV support for transfer through firewalls
  - Reply string from QUOTE command can be retrieved

RxFtp Sample - Ping

- **Ping IP address:**

```
/* Ping an IP address to see if it is up and running */
If RxFuncQuery("FtpLoadFuncs") Then
 Do
   Call RxFuncAdd "FtpLoadFuncs","RxFtp","FtpLoadFuncs"
   Call FtpLoadFuncs
 End

Call FtpVersion "RxFTPVersion"
Say "RxFTP Version:" RxFTPVersion

Say "Ping www.ibm.com:" FtpPing("www.ibm.com", "500") "ms for 500 byte"
```

RxFtp Sample - UpLd

- **Upload a file to a ftp server:**

```
/* Send a local file to a ftp server */
If RxFuncQuery("FtpLoadFuncs") Then Do
  Call RxFuncAdd "FtpLoadFuncs","RxFtp","FtpLoadFuncs"
  Call FtpLoadFuncs
End

If FtpSetUser("my.ftp.server.com", "myuserid", "mypassword") = 0 Then
  Exit

Say "Server operating system:" FtpSys("OperSys")

Call FtpChDir "upload" /* change to upload directory */
If FtpPut("localfile.dat", "remotefile.dat") \= 0 Then
  Say "File PUT failed."

Call FtpLogoff
```

DB2 Access Library

- Included since DB2 Version 2.1.2
- Create/modify databases, tables
- Insert/modify/delete data from tables
- List data in tables/views
- Handle BLOBs (Binary Large OBject)

DB2 Samples

- **CREATEDB.REX:**
  - Create new database (needs to be run from DB2CMD)
- **LOADTBL.REX:**
  - Load table from ASCII file
  - ASCII file contains used columns in first line
- **LISTTBL.REX:**
  - List all entries with all columns from DB

EHLLAPI Library

- Send commands to host
- Read screen contents
- Query session parameters/status
- Query/set cursor position
- Query/set field contents, attributes
- Transfer files
- Included with Personal Communications for Windows 4.2

EHLLAPI Sample - HLLAPI.CLS

- Defines .HLLAPISession class:
  - Wrapper for existing EHLLAPI functions
  - Translates encoded information into usable formats (.HLLAPISessionStatus class)
  - New functions added:
    - Send Command
    - Read reply from a command (line oriented output)

EHLLAPI Sample - MoveFls

- Move files from host to PC
- Append data to files on PC if they exist
- Set PC file timestamp to the host file timestamp after transfer
- Erase files on host after successful transfer
- Syntax:

  rexx movefls [*filemask* [*targetdir* [*hostsession*]]]

  (*filemask* uses dot notation, e.g. *.NOTEBOOK.A)

REXX/SQL Library

- Send SQL commands to a large number of databases
- Supported Windows databases:
  - DB2, Generic ODBC, Openlink UDBC, Oracle 7.x, Solid Server, Sybase SQL Anywhere, mSQL 2.0
  - Under construction: Informix, Ingres II, MySQL, Sybase System 10/11, mSQL 1.0.16
- More details at author's homepage:

  http://www.lightlink.com/hessling/#REXXSQL

.WindowsRegistry Class

- Create/delete/query/set registry keys and values
- List subkeys and values
- Save/load keys to/from files
- See sample in attached code

.WindowsProgramManager Class

- Create/delete program groups
- Create/delete program items
- Open program groups
- Create shortcuts on the Desktop or in any other location

.WindowsProgramManager Sample (1)

```
/* show the .WindowsProgramManager class */
pm = .WindowsProgramManager~New
If pm~InitCode \= 0 Then Exit

pm~AddGroup("My own group")
pm~AddItem("My own notepad", "notepad.exe", "NOTEPAD.EXE", 0)
pm~AddItem("My own calculator", "calc.exe",,,, 1)
pm~AddItem("My own write", "write.exe",,, "c:\")
pm~ShowGroup("My own group", "MAX")

Call SysSleep 15

pm~DeleteItem("My own write")
pm~ShowGroup("My own group")

Call SysSleep 15

pm~DeleteGroup("My own group")
pm~Deinstall

::REQUIRES WINSYSTM.CLS
```

.WindowsProgramManager Sample (2)

```
/* show the .WindowsProgramManager class */
pm = .WindowsProgramManager~New
If pm~InitCode \= 0 Then Exit

/* Create a shortcut to notepad editor with shortcut key CONTROL+ALT+N */
/* The shortcut should be personal for the current user. */
rc = pm~AddDesktopIcon("My Notepad 1",,
                       "%SystemRoot%\system32\notepad.exe",,,,
                       "%HOMEDRIVE%%HOMEPATH%",,, "N")
If rc = .False Then
  Say "Error creating sortcut: My Notepad 1"

/* Create a shortcut to run REXXTRY, with the REXX.ICO icon.  */
/* The shortcut should be common for all users, working directory */
/* is %TEMP%, shortcut key is CTRL+ALT+T.                    */
rc = pm~AddDesktopIcon("RexxTry", "rexx.exe", "c:\objrexx\rexx.ico", 0,,
                       "%TEMP%", "COMMON", "rexxtry", "T", "MAXIMIZED")
If rc = .False Then
  Say "Error creating sortcut: RexxTry"

::REQUIRES WINSYSTM.CLS
```

.WindowsEventLog Class

- Available on NT systems
- Get number of event log records
- Read/Write event log records
- Delete event log with optional backup

.WindowsEventLog Sample

```
/* show the .WindowsEventlog class */
evl = .WindowsEventLog~New
If evl~InitCode \= 0 Then Exit

If evl~Open(,"Application") \= 0 Then Exit

Say "The Application Eventlog has" evl~GetNumber "records."
Events = evl~Read
If Events \= .nil Then
  Do Record Over Events
    Parse Var Record Type Date Time """ SourceName""" ID UserId Computer """ String """ """ Data """
    Say "============================================================================="
    Say 'Type     :' Type
    Say 'Date     :' Date
    Say 'Time     :' Time
    Say 'Source   :' SourceName
    Say 'ID       :' ID
    Say 'UserId   :' UserId
    Say 'Computer :' Computer
    Say 'Detail   :' String
    Say 'Data     :' Data
  End
evl~Close
evl~Deinstall

::REQUIRES WINSYSTM.CLS
```

.WindowsClipboard Class

- Exchange text data with other applications through the Windows clipboard
- Copy/Paste/Empty data to/from clipboard
- Query availibility of data in clipboard

.WindowsClipboard Sample

```
/* show the .WindowsClipboard class */
cb = .WindowsClipboard~New

Call ShowClipboard

cb~Empty /* empty clipboard */
Call ShowClipboard

cb~Copy("Hello from Object REXX.") /* copy new text into clipboard */
Call ShowClipboard
Exit

ShowClipboard:
  If cb~IsDataAvailable Then
    Say "Text data is available in the clipboard:" cb~Paste
  Else
    Say "No text data is available in the clipboard."
  Return

::REQUIRES WINSYSTM.CLS
```

.WindowsManager Class

- Locate Windows by name, position, Z-order (foreground)
- Query/set console window titles
- Send a set of keystrokes to a window
- Push buttons in a window
- Select menu items in a window

.WindowsManager Sample (1)

```
/* Show features of the .WindowsManager class */
NewWindowTitle = "REXX Command Prompt"
'start "" || NewWindowTitle || " %COMSPEC%' /* start a new command prompt */
Call SysSleep 1 /* wait for window to appear */

wMgr = .WindowsManager~New
wMgr~SendTextToWindow(NewWindowTitle, "echo Hello from REXX" || "0D"x)
Call SysSleep 5

wMgr~SendTextToWindow(NewWindowTitle, "exit" || "0D"x)
wMgr~Deinstall  /* Deinstall the WindowsManager and clean up */

::REQUIRES WINSYSTM.CLS
```

.WindowsManager Sample (2)

```
/* Show features of the .WindowsManager class */
wMgr = .WindowsManager~New
Say "The current console title is:" wMgr~ConsoleTitle
Say "Watch the title for a change..."
Call SysSleep 3

wMgr~ConsoleTitle = "Object REXX .WindowsManager class demonstration"
Say "See it?"
Call SysSleep 3

wMgr~Deinstall  /* Deinstall the WindowsManager and clean up */

::REQUIRES WINSYSTM.CLS
```

.WindowObject Class (1)

- Proxy object for real windows on screen
- Set window properties: enable, disable, hide, minimize, maximize, restore, position, size, foreground, title
- Query window information: window class, id, child windows
- Send commands, key strokes, messages to window

.WindowObject Class (2)

- Simulate selection of PushButton
- Send menu commands
- Query menu objects belonging to window

.WindowObject Sample - WinMgr3

- Use Notepad as controlled window
- Resize, move window
- Minimize, maximize, restore window
- Send keystrokes to edit window
- React on additional dialog popping up (confirmation dialog on close)

.MenuObject Class

- Proxy object for menu controls in windows
- Query ids/text of menu items
- Locate sub menus
- Process menu items

.MenuObject Sample

```
/* Show features of the .MenuObject class */
wMgr = .WindowsManager~New
"start notepad"
Call SysSleep 1
np = wMgr~Find("Untitled - Notepad")
Say "Notepad has the following menu structure:"
Call ShowMenu np~Menu, 0
Say "The System Menu has the following structure:"
Call ShowMenu np~SystemMenu, 0
np~SendSysCommand("CLOSE")
wMgr~Deinstall  /* Deinstall the WindowsManager and clean up */

::REQUIRES WINSYSTM.CLS
::ROUTINE ShowMenu
  Use Arg Menu, Indent
  Do i = 0 To Menu~Items - 1
    Say Copies(" ", Indent) || Menu~TextOf(i)
    subMenu = Menu~SubMenu(i)
    If subMenu \= .Nil Then
      Call ShowMenu subMenu, Indent + 2
  End
```

Further Information

Object REXX Homepage:

http://www.software.ibm.com/ad/obj-rexx/

REXX Sockets Tutorial:

http://www2.hursley.ibm.com/rexxtut/socktut1.htm