QShell

and

OpenSSH

for

IBM System i

OMNI

15-May-2007

QShell

- In the beginning... (1970)

- Ken Thompson wrote the original Unix shell, and it was good.

- ..but not that good. It was replaced by Stephen Bourne’s shell, sh.

- This sh was better, but not good enough.

- David Korn wrote ksh, and it was much better.

- More recent developers wrote the Bourne-Again shell, bash.

- Qshell is a native i5/OS ported version of Korn shell
**Qshell**

What is it for?

What is it?

How do you use it?

Customization

Cool (really useful) stuff

Tricky stuff

More stuff

---

**Qshell - What is it for?**

- Interactive command environment
  - Familiar to Unix users
  - New utilities for i5/OS

- Run shell scripts
  - Typical batch routines
  - Text processing
  - Scripts ported from Unix
  - Capture output
  - Automate input
  - IFS files

- Run Java programs
- Run PERL scripts

---
Qshell - What is it?

- Native i5/OS ported version of Korn shell
- Compatible with Bourne & Korn shells
- Based on POSIX and X/Open standards
- Runs UNIX commands interactively or from a script (text file)
- No charge option (#30) of i5/OS
- Installed by default

QShell and Utilities

- Interpreter (qsh)
  - Reads commands from an input source
  - Interprets each command (line)
  - Directs input and output as requested
  - Many built-in functions

- Utilities (or commands)
  - External programs
  - Provide additional functions
  - Some quite simple, some very complex
Qshell - How do you use it?

- **QSH CL command**
  - Start interactive session
  - Run a single command or script

```
Selection or command
====> QSH
F3=Exit  F4=Prompt
```

```
QSH Command Entry
$
```

```
Qshell - How do you use it?

- **Built-in functions**
  - system
  - read
  - ...

```
```
QSH Command Entry
> echo hello world!
hello world!
$
```

```
Qshell - How do you use it?

- **External programs**
  - /qsys.lib/mylib.lib/mypgm.pgm
  - Searches PATH variable
  - Many utilities...

```
```
Qshell - How do you use it?

- **Variables**
  - name=value
  - arithmetic

```
```
Qshell - How do you use it?

- **Logic loops**
  - if
  - while
  - until
  - ...

```
```
> __echo hello again! ____________
```

```
F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear  F17=Top  F18=Bottom  F21=CL command entry
```

```
F3=Exit  F6=Print  F9=Retrieve  F12=Disconnect
F13=Clear  F17=Top  F18=Bottom  F21=CL command entry
```
Qshell - How do you use it?  

cd = change directory

- . (dot) Refers to current directory
  - Useless command: cd .

- .. (2 dots) Refers to parent of current directory
  - More useful: cd ..

- Regular expressions match file names
  - Examples:

    | Files beginning with “A” | A* |
    |--------------------------|----|
    | Files beginning with “A-M”| [A-M]* |
    | Files ending with numbers 3-9 | *[3-9] |
    | Files containing “qtr” | *qtr* |
    | Files with 4-letter names | ??? |

Qshell - Some things you can do with it

- Manage ownership with chown
  - Ex: assign entire directory and content to Bob
    chown -R Bob directory1

- Manage authority (including public) with chmod
  - Ex: remove all public authority from directory and contents
    chmod -R o-rwx directory1

- Manage files in IFS
  - Ex: make a local backup copy of an entire directory
    cp -R directory1 /temp/directory1backup
**Qshell Utilities**

- Program objects in library QSHELL
- Symbolic links in /usr/bin
- **Replicate for your program objects:**
  In -s /QSYS.LIB/MYLIB.LIB/MYPGM.PGM mypgm

```
head
sed
grep
```

```
/usr/bin/
/QSYS.LIB/QSHELL.LIB
```

**Qshell - How do you use it?**

- Running scripts
  - Text files in IFS
    - Enable execution bit with chmod command
      chmod a+x script-name.sh
  - Redirect input and output (not terminal I/O)
    - Automate input with <
    - Capture output with >
      ```
      echo hello there! > hello.txt
      ```
      One program to another:
      ```
      echo hello | awk ...
      ```
  
- Running Java
  ```
  java hello.class
  ```
Using Qshell within a CL Program

Qshell script /nightly/merge.sh

```
#!/usr/bin/sh

/qsys.lib/etl.lib/merge.pgm >> /logs/merge.out.txt
status=$?
if [[status = 0]] ; then
    echo Successfully ran merge1.sql on `date` > /logs/run.txt
    exit 0
else
    exit 1
fi
```

CL Program

```
PGM
ADDENVVAR VAR(QIBM_QSH_CMD_ESCAPE_MSG) VALUE(Y)
QSH CMD(`/nightly/merge.sh')
MONMSG MSG(QSH0005) EXEC(GOTO FAILED)
GOTO END

FAILED: SNDPGMMMSG MSG('Error running script')
END:
```

Customize: ~/.profile

- Special character for user’s starting directory: (tilde) ~
  - From user profile object HOMEDIR
- File is processed when starting Qshell
- Define your environment
  - Command aliases
    - alias ll='ls -l'
  - Environment variables
    - PATH=$PATH:/QOpenSys/usr/bin
- Starting program(s)
- Other script(s)
**PASE Shells & Utilities**

- AIX binary executables
- Additional utilities and shell interpreters (sh, csh, ksh)
- Add `/QOpenSys/usr/bin` to PATH in `~/.profile` for use in Qshell

/`QOpenSys/usr/bin/`

- ksh
- csh
- sh
- tar
- egrep
- compress
- awk

**Cool Stuff**

- Unzip files with `jar` utility
  ```
  jar -xvf filename.zip
  ```

- Analyze disk usage with `du` utility
  ```
  /QOpenSys/usr/bin/du -sk *
  ```

- Search for files by name with `find` utility
  ```
  find . -name filespec
  ```
Cool Stuff

- **awk** utility uses a field-oriented pattern processing language
  - Calculations from numbers in text files
  - Manipulating text reports
  - Translating files from one format to another
  - Re-ordering columns of data
  - Much more

- Examples
  
  ```
  echo A B C | awk '{print $2 $1 $3}'
  BAC
  ```

  ```
  echo A 2 4 | awk '{print $2+$3 $1}'
  6A
  ```

Cool Stuff

- Analyze disk usage with **du** utility

```bash
export PATH=$PATH:/QOpensys/usr/bin/
du -sk /* | awk '{printf "%010d %s\n",$1,$2}' | sort
```

```
0000000002 /QNTC
0000000004 /QFileSvr.400
0000000008 /bin
0000000008 /lib
0000000008 /QOPT
0000000024 /QSR
0000000032 /QTCPTMM
0000000040 /var
0000000120 /etc
0000000135 /QDLS
0000000240 /examples
0000001024 /home
000001513112 /tmp
000009197864 /usr
0001333952 /www
0002003880 /QOpenSys
0005470224 /VSRPTF
0011336096 /QIBM
0029244900 /QSYS.LIB
```
Cool Stuff

- Analyze library space consumed with `du` utility

```bash
export PATH=$PATH:/QOpensys/usr/bin/

du -sk /QSYS.LIB/*.LIB | awk '{printf "%010d %s\n",$1,$2}' | sort
```

Cool Stuff

- **db2**
  - Process SQL statement(s) by DB2 engine in i5/OS

  ```sql
db2 "select * from mytable where ...."
  ```

- **system**
  - Process i5/OS CL command

  ```bash
  system "CHGUSRPRF BOB STATUS(*ENABLED)"
  ```

- **catsplf**
  - Concatenate and print spool files

- **native2ascii**
  - Convert stream to ASCII
Useful Stuff - db2 utility

- SQL naming: MYLIB.MYTABLE

- 3 modes
  - Interactive
  - Command-line SQL statement
  - Read SQL from file

- Unique job = unique QTEMP

- Examples:
  
  db2 "update employees set salary=salary+10000
          where name='Bob' "
  
  db2 -t -f weekly_merge.sql

Typically returns 0, even on SQL error

Tricky Stuff

Common i5/OS Shell Script Migration Issues

- Additional carriage-return characters in script
  - PASE shell interpreters dislike this
  - Fix: tr –d “\r” < test.sh > testnew.sh

- Hard-coded path names

- UNIX-specific administration utilities
  - useradd, mkuser, installp

- Expect to run as user named “root” or UID=0

- UNIX file system dependency
  - /proc (used for kernel information)
  - /etc (system configuration files)
Tricky Stuff - ASCII vs EBCDIC

- CCSID: Coded Character Set Identifier
  A single number that defines a set of available characters, a scheme for encoding them (such as single- or double-byte) and the assignment of those characters to numbers for binary representation

- AS/400 legacy is based on EBCDIC encoding
- PASE shells only use ASCII
- Many programs only read one encoding

- Qshell does well with both
  - Opens files in text mode
  - i5/OS converts files as Qshell reads and writes
  - Creates EBCDIC output files by default

Environment variable controls QShell output encoding
example: QIBM_CCSID = 819

- Every file is tagged with a CCSID
  See these tags:
  ls -S

- i5/OS file I/O: either **text** or **binary**

- Text operations → Automatic conversion
- Keeping files in ASCII is often ideal
More Stuff - PERL on i5/OS

- www.cpan.org/ports/index.html#os400
  - PASE Perl 5.8.0
  - ILE Perl 5.005_02

- iSeries Tools for Developers PRPQ 5799-PTL
  - PASE Perl 5.005_3

- "iSeries PERLs of Wisdom"
  - ibm.com/servers/enable/site/porting/iseries/overview/iperl/

OpenSSH on i5/OS

www.openssh.org

- Licensed Program Offering from IBM
  - No charge, 5733SC1 option 1
  - OpenSSH v. 3.5p1
  - ibm.com/servers/enable/site/porting/tools/openssh.html

- Secure interactive shell login
- Secure remote command
- Secure file transfer
- Secure tunnel

- Authentication using i5/OS or key exchange
**OpenSSH**

- Open source
- Free licensing
- Strong encryption on all communication
- X11 (Graphical) display forwarding
- TCP/IP Port forwarding (encrypt legacy protocols)
- Strong authentication (Public Key, Kerberos)
- SFTP client and server
- Data compression

---

**OpenSSH Getting Started**

- Install
  - RSTLICPGM 5733SC1 OPTION(*BASE)
  - RSTLICPGM 5733SC1 OPTION(1)

- Generate host (server-side) keys
  - 3 different cipher identity keys
  - Long commands, see website
  - [ibm.com](http://ibm.com)/servers/enable/site/porting/tools/openssh.html

- Start server by Qshell (user must have *ALLOBJ)
  - QSH CMD(‘/QOpenSys/usr/sbin/sshd’)
OpenSSH Login & Remote Command

- Login starting a PASE shell
- Clients
  - Windows: PuTTY by Simon Tatham
  - MacOS, Linux: ssh included
  - AIX: ssh client in Linux affinity tools package
  - PalmOS: pssh by Greg Parker

TuSSH

OpenSSH Bourne-Again Shell

- Get the **2.05b** version of bash for AIX
  - http://aixpdslib.seas.ucla.edu/packages/bash.html
- Significantly improved usability (arrow keys, etc)
- Copy to `/QopenSys/usr/bin`
- Do not get 3.x version
Helpful ~/.profile

PATH=$PATH:/QOpenSys/usr/bin
export PATH

if [ "$SHELL" = "/QOpenSys/usr/bin/bsh" ] ; then
exec bash
fi

OpenSSH
File Transfer

- IFS name format only
- Can use with *FILE objects, including save files
- sftp acts like FTP
- scp syntax:
  scp user@source:/dir/files user@target:/dir/files

- For example, for QSECOFR to retrieve a save file
  QGPL / MYSAVF

  scp qsecofr@my400:/qsys.lib/qgpl.lib/mysavf.file .
**OpenSSH Secure Tunnel**

- Transparently secures application traffic
- “Port forwarding”
- Example: encrypt i5/OS-to-i5/OS telnet

```
telnet localhost PORT(2300)
```
```
ssh -T -L2300:sysA:23 sysB
```

**OpenSSH Key Exchange**

- Generate one key pair per user: `ssh-keygen`
  - public key: on server, used to create lock
    `/home/user/.ssh/authorized_keys`
  - private key: used by client to unlock

* Private key never leaves client
OpenSSH Notes

- Protect private key(s)
- PuTTY settings:
  - Keyboard / Backspace: Ctrl-H
- Start sshd from Qshell:
- Use “ssh -T” from 5250-based interactive session
- User profile names must be 8 or less characters

My OpenSSH Use in an IBM Project

- Virtual Loaner Program
  - Fully automated i5/OS provisioning infrastructure
  - System image save/restore/initialize
  - TCP/IP management
  - LPAR operations to HMC
- Examples
  
  ```
  ssh qsecofr@target system "\’\’CHGUSRPRF QSECOFR PASSWORD(xxxx)\’\’"
  ```
  
  ```
  ssh hscroot@hmc "lssyscfg -r lpar -m 10ABCDE -F name,state"
  ```
  
  ```
  scp vlplib.savf qsecofr@target:/qsys.lib/qgpl.lib/vlplib.file
  ```
**Summary**

- **QShell**
  - Runs all i5/OS shell utilities
  - Interactive from 5250 session: QSH
  - Interacts in EBCDIC (default), but can do
  - PASE shells and utilities in /QOpenSys
  - Easy logic
  - Powerful text redirection
  - Variety of utilities

- **OpenSSH**
  - Shared open source code with AIX
  - Key exchange authentication secure, reliable
  - Interactive login or remote command
  - Secure file transfer (copy)

---

**What About Bob? (Bob Bittner)**

- 10 Years at IBM
- AS/400 -- System i Unix Porting Team
- System i architect: Virtual Loaner Program
- i5/OS Evangelist

- Private Pilot (ASEL, VFR)
- Airplane builder
- Cozy MkIV
- R/C Helicopter pilot
- Cheesehead