# RPG Makes Friends with Open-Source Apps

## Open Your i

July 2021

MobiGoGo
*Work at the Speed of Mobility*

# Your Presenter

## Richard Schoen


richard@richardschoen.net
-or-
richard@mobigogo.net


- 30+ yr developer IBM i, Windows, Linux, Mac
- Started RJS Software Systems in 1990
- Created automated report distribution, doc management and app integration software
- Sold business in 2014 when partner retired
- Started MobiGoGo to build multi platform apps
- Provides training and consulting services
- Contributes open source to Github repositories
- Author of iForGit Native IBM i Git client for RDI, SEU and PDM developers

Github: http://github.com/richardschoen
Business site: http://www.mobigogo.net
Personal site: http://www.richardschoen.net
LinkedIn: https://www.linkedin.com/in/richardschoen
Twitter: @richardschoen
Blog: http://blog.richardschoen.net

iForgit IBM i Git Client: http://www.iforgit.com

# Richards Journey with IBM i System Integration

- Integrate S/36 to DOS Turbo Pascal screen -HLLAPI (**1988**)
- Windows 3.1, Microsoft Visual C/C++, Visual Basic 3 (**1993**)
- APPC programming with PC Support to access RPG (**1995**)
- TCP/IP, Email, FTP from Windows to IBM i (**1997**)
- Sockets programming with VB and RPG (**2002**)
- Web development and web services (**1997 –current**)
- IBM i Access ODBC/OLEDB/ADO.Net (**2000-current**)
- JT400 java API converted to .Net with IKVM (**2005-current**)
- HTTP URL API –(AKA REST Services) (**2004 –current**)
- XMLSERVICE –universal open source IBM i DB access (**2012-current**)
- Editor/IDE choices: RDI, Eclipse, Visual Studio, Visual Studio Code, Notepad++, etc…
- (**1984–today : Green screen to smart devices**)
  The greatest computing era !!

MobiGoGo
Work at the Speed of Mobility

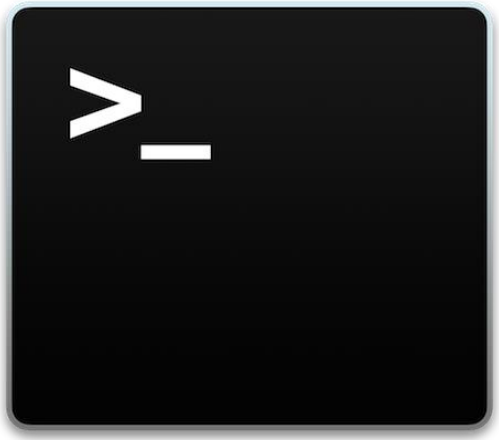
# Session Topics

- Why consider using RPG and Open Source together ?
- How to utilize open-source applications from RPG and CL
- Intro to QShell on i
- Review logic flow
- A few examples

MobiGoGo
Work at the Speed of Mobility

# How To Use Shells and SSH on IBM i

- Your command line for QShell, PASE and Bash
- A good intro to the various shells on IBM i
- Andy Youens – FormaServe (UK)
- https://www.youtube.com/watch?v=9rL9U8hfIHA
- We will use QShell, Pase and Bash

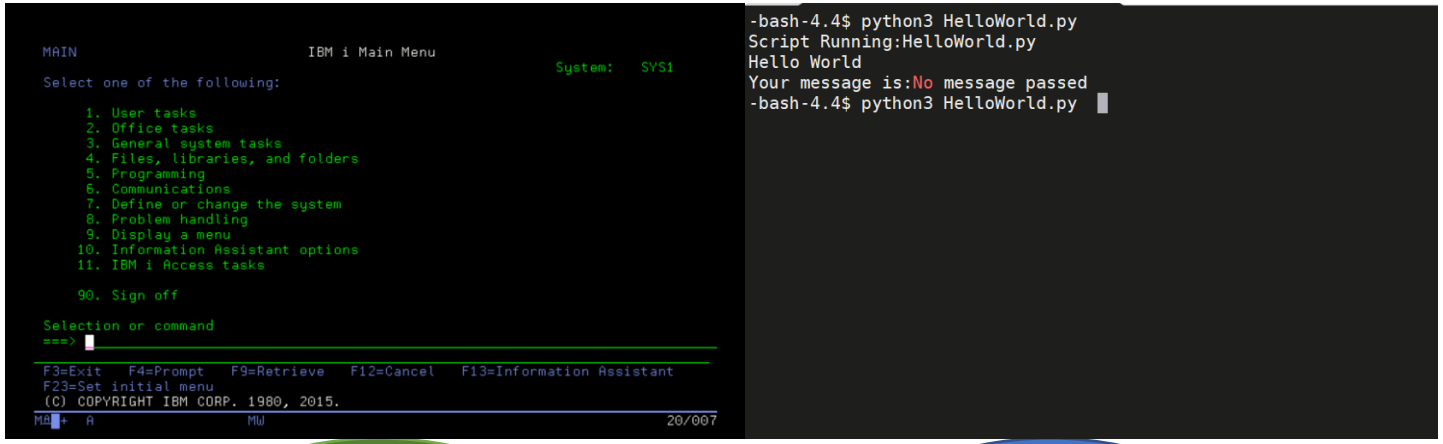


# Why consider using RPG and Open Source together ?

- Functionality not easily available in CL or RPG by themselves
- Tons of Python and other language example code, libraries and tutorials
- Access web services
- Send and receive email
- Crawl IFS directories
- Become productive quickly
- Integrates or extends existing IBM i apps across process boundaries
- QShell on I can be used for all kinds of IBM i utility programs
- Run any QShell or PASE code from traditional programs

MobiGoGo
Work at the Speed of Mobility

# Get The Best of Both Worlds




Traditional 5250 Applications

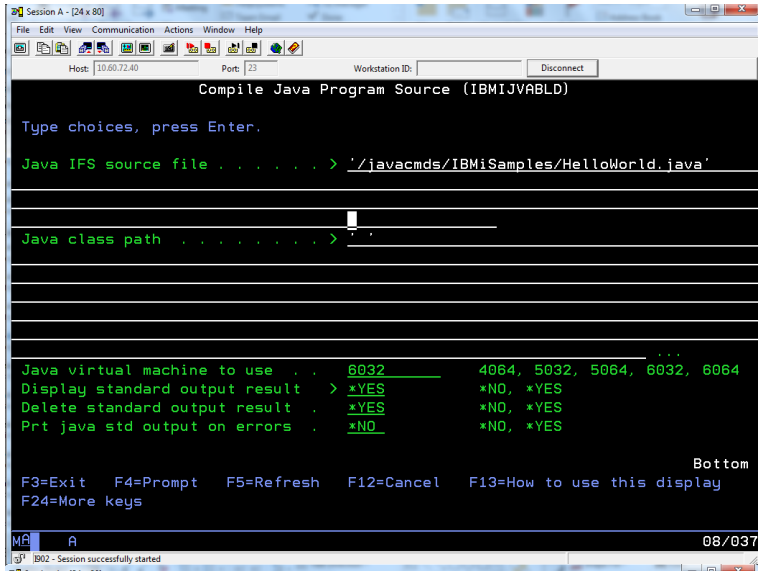
RPG, CL, COBOL Applications

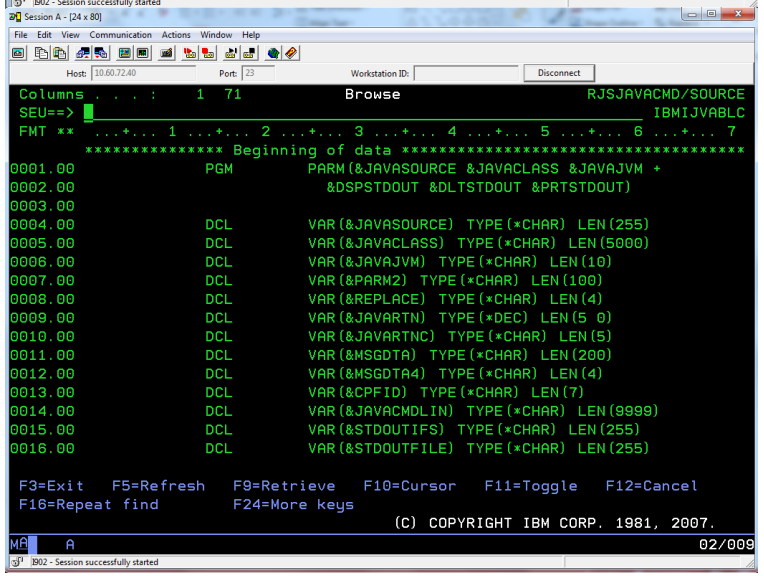
QShell, PASE, Bash Shell AIX Apps

Java, Python, PHP, Node, C, C++, Ruby, R, Rust, ODBC, Postgres, MariaDB, SQL Server

MobiGoGo
Work at the Speed of Mobility

# Traditional IBM i Job Program Call Flow




Type CL Command or
Embed in a CL Program

Run CL/RPG Processing Pgm
May pass parameters

Processing Program
CL, RPG, COBOL

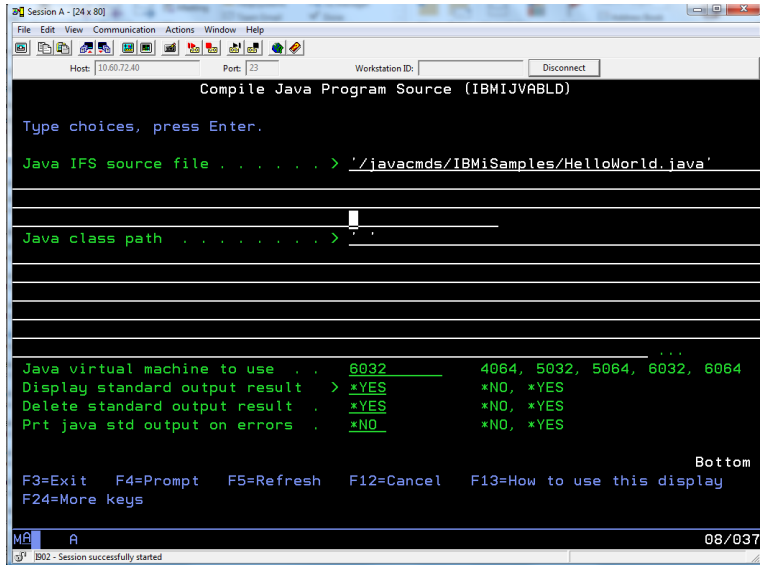
Parameters or Data Returned to
Calling CL Program or to
Command Line as CPF Messages

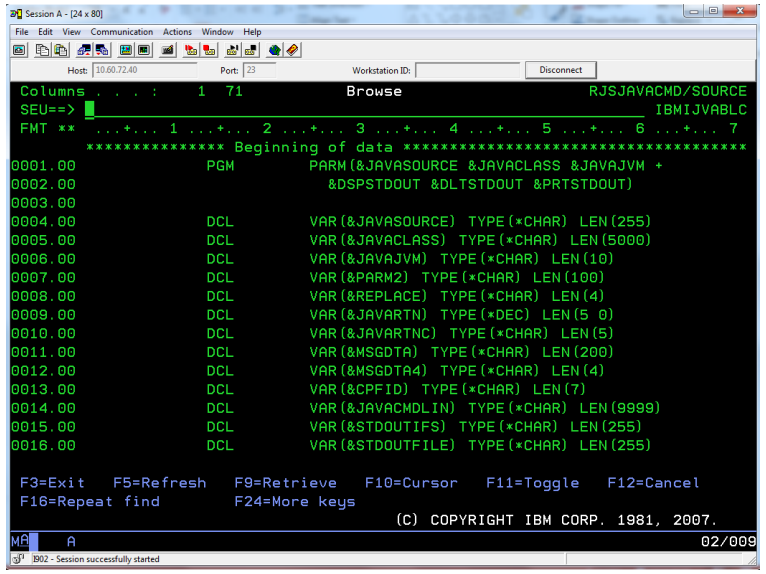
Log info written to job log,
physical file or IFS file

QTEMP
Library

MobiGoGo
Work at the Speed of Mobility

# Traditional Program Call Flow with QShell/PASE Apps




Type CL Command or
Embed in a CL Program

Run CL/RPG Processing Pgm
May pass parameters

Processing Program
CL, RPG, COBOL

Run QShell/PASE
Program or Command

QTEMP
Library

TMP
Library

QShell/PASE app runs in a
thread job

Parameters Returned in STDOUT
IFS file from QShell/Pase call

STDOUT
log info can be
Processed by RPG

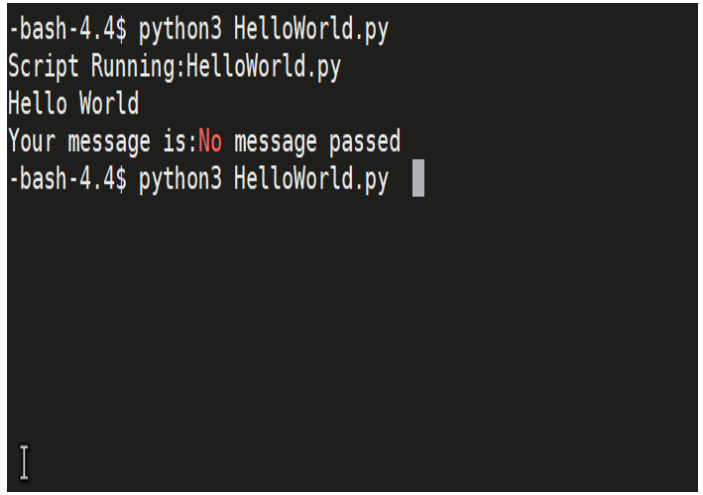
Parameters or Data Returned to
Calling CL Program or to
Command Line as CPF Messages

STDOUT log info written to job
log, physical file or IFS file

# What is Standard Output (STDOUT) ?

**STDOUT – Standard Output**

- Console output from a PASE/QShell application
- Similar to IBM i joblog
- Allows PASE/QShell apps to pass back data and messages
- All QShell/PASE languages can generate STDOUT console messages
- STDOUT data can be used from CL/RPG/COBOL apps
- Great way to communicate between PASE/QShell and traditional IBM i apps such as CL/RPG/COBOL



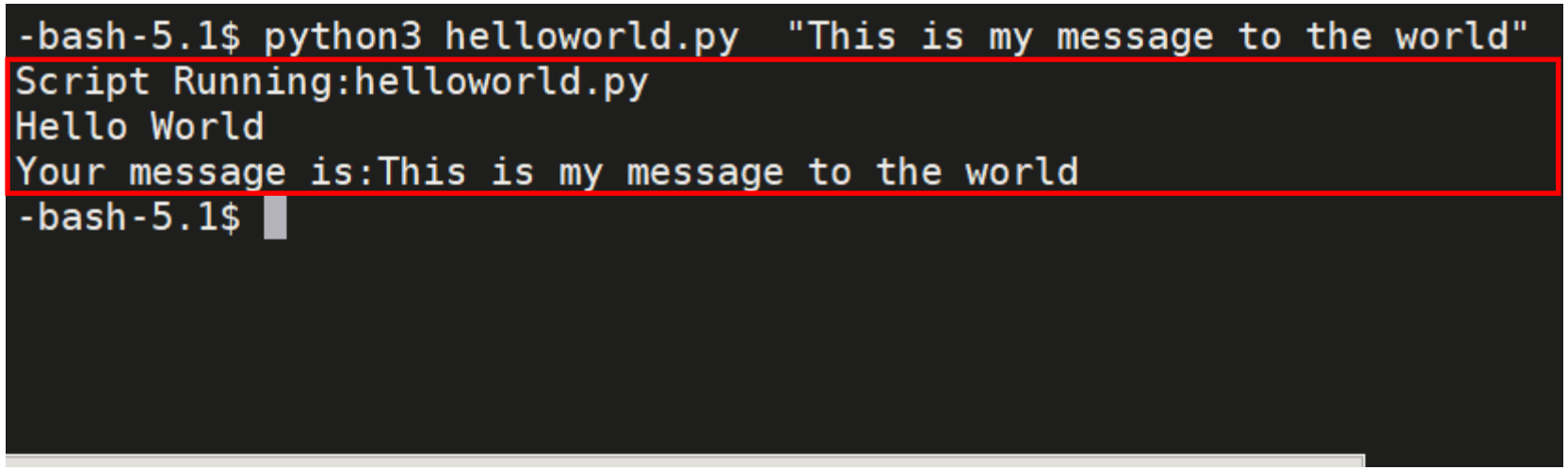
MobiGoGo
Work at the Speed of Mobility

# Return Code from QShell/PASE Calls

- Tells if the QShell/PASE app ran succsessfully
- Return code is 0 if call end normally
- Any non-zero return code indicates a call error or failure
- Stdout can be reviewed for other meaningful messages/data

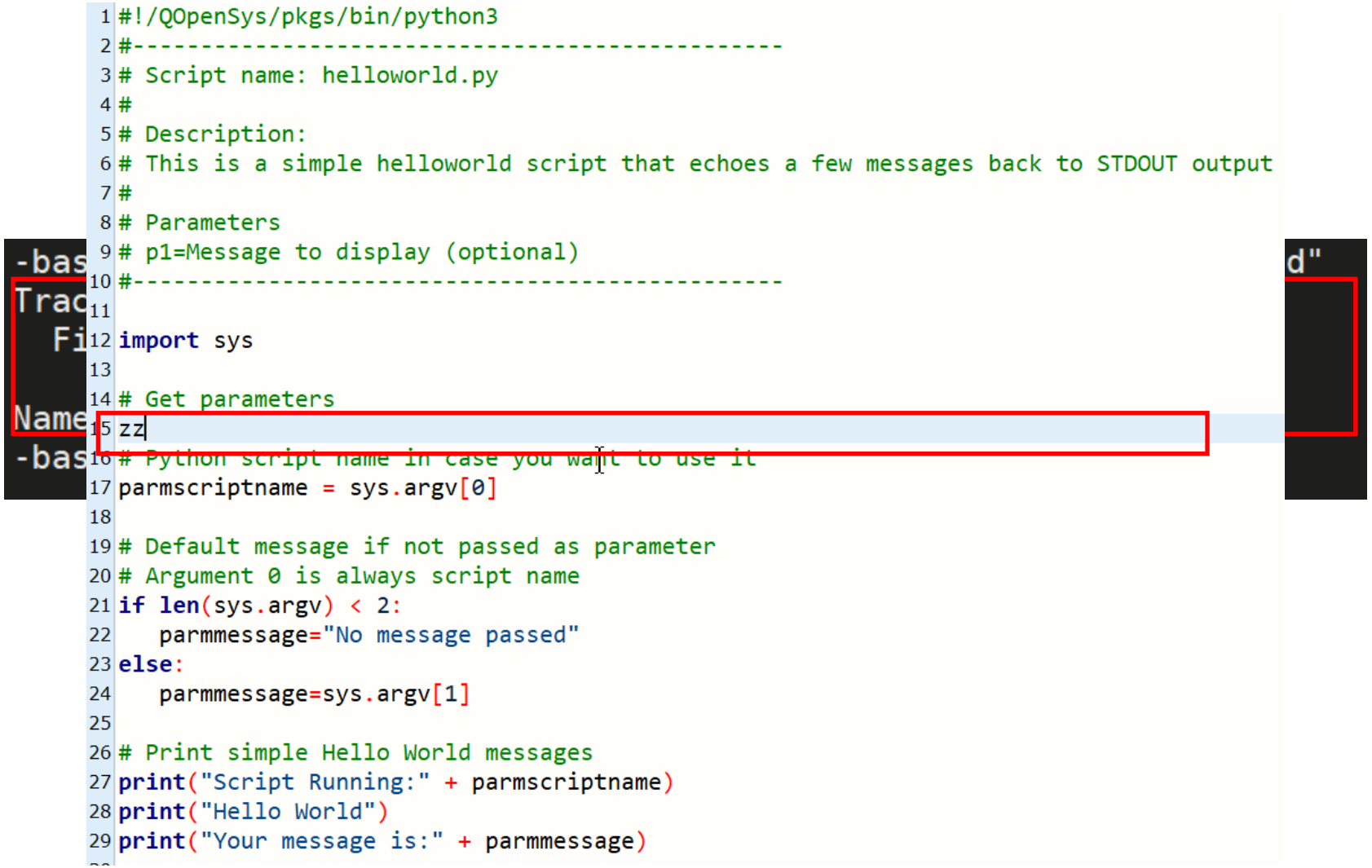
Return Code 0  Non-Zero Return Code <> 0

# Hello World Console Output - STDOUT



MobiGoGo
Work at the Speed of Mobility

# Hello World Console Output Error - STDOUT



```
1  #!/QOpenSys/pkgs/bin/python3
2  #------------------------------------------------
3  # Script name: helloworld.py
4  #
5  # Description:
6  # This is a simple helloworld script that echoes a few messages back to STDOUT output
7  #
8  # Parameters
9  # p1=Message to display (optional)
10 #------------------------------------------------
11
12 import sys
13
14 # Get parameters
15 zz
16 # Python script name in case you want to use it
17 parmscriptname = sys.argv[0]
18
19 # Default message if not passed as parameter
20 # Argument 0 is always script name
21 if len(sys.argv) < 2:
22    parmmessage="No message passed"
23 else:
24    parmmessage=sys.argv[1]
25
26 # Print simple Hello World messages
27 print("Script Running:" + parmscriptname)
28 print("Hello World")
29 print("Your message is:" + parmmessage)
```

MobiGoGo
Work at the Speed of Mobility

# db2util Console Output - STDOUT

```
-bash-5.1$ db2util  "select * from qiws.qcustcdt"
"938472","Henning ","RJS","4859 Elm Ave ","Dallas","TX","75217","5000","3","37.00","1234.56"
"839283","Jones   ","B D","21B NW 135 St","Clay   ","NY","13041","400","1","100.00","0.00"
"392859","Vine    ","S S","PO Box 79    ","Broton","VT","5046","700","1","439.00","0.00"
"938485","Johnson ","J A","3 Alpine Way ","Helen ","GA","30545","9999","2","3987.50","33.50"
"397267","Tyron   ","W E","13 Myrtle Dr ","Hector","NY","14841","1000","1","0.00","0.00"
"389572","Stevens ","K L","208 Snow Pass","Denver","CO","80226","400","1","58.75","2.50"
"846283","Alison  ","J S","787 Lake Dr  ","Isle  ","MN","56342","5000","3","10.00","0.00"
"475938","Doe     ","J W","59 Archer Rd ","Sutter","CA","95685","700","2","250.00","100.00"
"693829","Thomas  ","A N","3 Dove Circle","Casper","WY","82609","9999","2","0.00","0.00"
"593029","Williams","E D","485 SE 2 Ave ","Dallas","TX","75218","200","1","25.00","0.00"
"192837","Lee     ","F L","5963 Oak St  ","Hector","NY","14841","700","2","489.50","0.50"
"583990","Abraham ","M T","392 Mill St  ","Isle  ","MN","56342","9999","3","500.00","0.00"
-bash-5.1$
```

# QShell on i – What is it ?

- Easy to use wrapper around QShell and PASE commands
- Simplifies running QShell or PASE commands including bash
- Sets up multithreading environment
- Handles command output logging – STDOUT capture
- Allows QShell/PASE commands to be submitted
- Don’t need an SSH session or STRQSH to run QShell/PASE commands
- Can be called directly from RPG or CL

MobiGoGo
Work at the Speed of Mobility

# QShell on i – Use Cases

- Call QShell, Pase or Bash commands directly from RPG or CL
- Run any Python, PHP, Node, shell scripts, etc
- Submit nginx, gunicorn or other background server/web service jobs
- Run interactively, submit or job schedule QShell/Pase commands

  (For batch jobs, use subsystems/job queues that allow multiple active threads QUSRNOMAX/QSYSNOMAX)
- Consume web services without HTTPGETCLOB/HTTPPOSTCLOB (Java)
- Compose, send and receive emails
- Talk to other databases (PostgreSQL, MariaDB, SQL Server, SQLite)

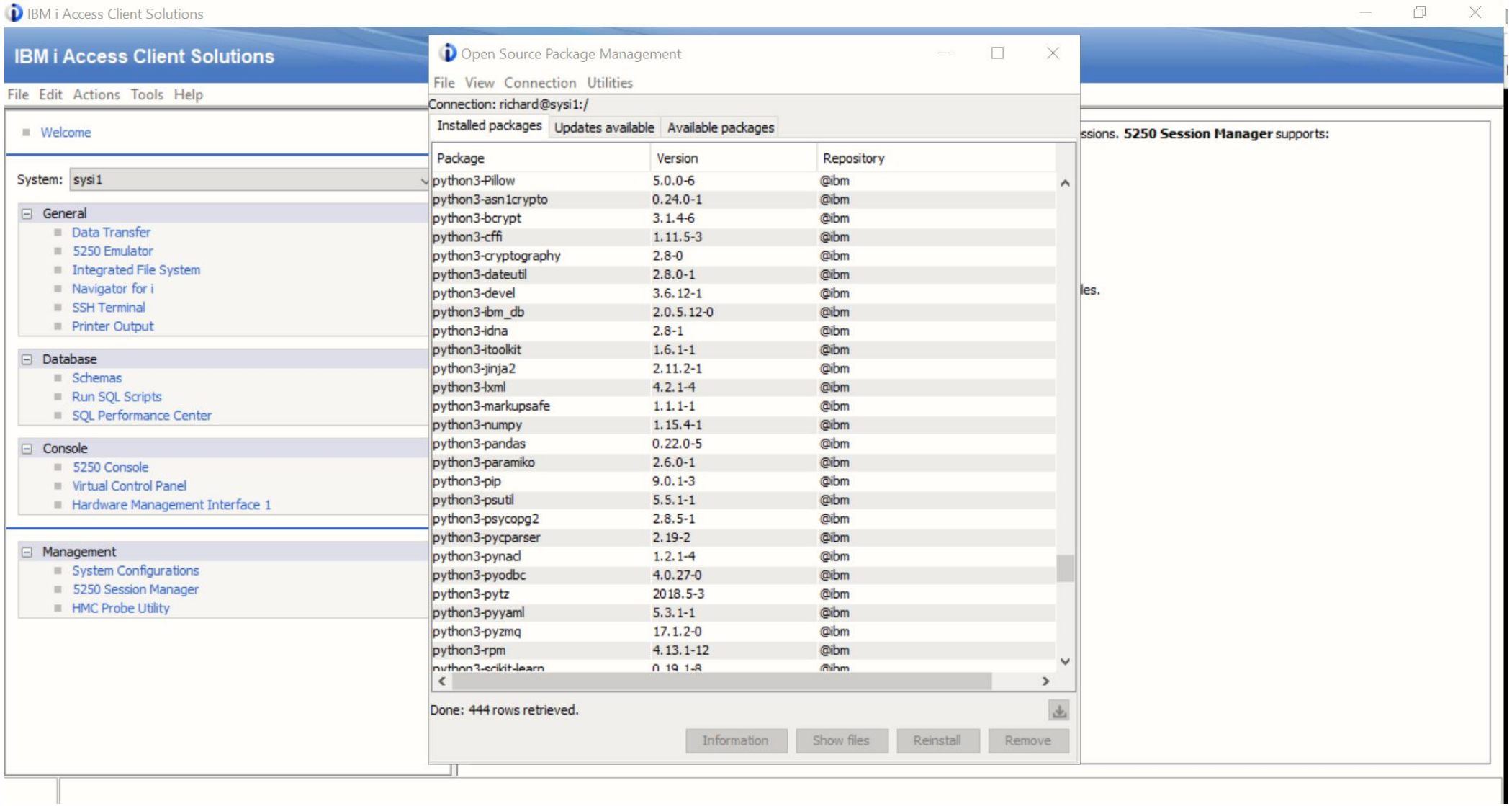
MobiGoGo
Work at the Speed of Mobility

# IBM i System Prerequisites – V7R2 and Above

- IBM Open Source Package Management
- Install all Python 3 packages (if using Python)
- Install unixODBC and unixODBC-dev
- Install IBM i Access ODBC Driver
- Install freetds if you want to talk to SQL Server
- Install QShell on i (https://github.com/richardschoen/qshoni)
- Uninstall 5733OPS if you can. Yum packages rule

MobiGoGo
*Work at the Speed of Mobility*

# Install All Python Packages



# Installing QShell on i



- Visit the Github site, download and build

  http://www.github.com/richardschoen/qshoni

- Command documentation in main **readme.md** page
- Library name: **QSHONI**
- Installing and Building QSHONI via getrepo-qshoni.sh
- Installing and Building QSHONI via Git Clone
- Installing QSHONI via Save File

# QShell on i Commands

- **QSHEXEC** - Run QShell Command Line
- **QSHBASH** - Run Bash Command via QShell
- **QSHPYRUN** - Run Python Script via QShell
- **QSHLOGSCAN** - Scan QShell Log File for Values
- **QSHPATH** - Set Open Source Package Path Environment Variable
- **QSHIFSCHK** – Check for IFS file existence

MobiGoGo
Work at the Speed of Mobility

# QSHEXEC - Run QShell Command Line

- Easy to use wrapper around QShell and PASE commands
- Simplifies running QShell or PASE commands including bash
- Sets up multithreading environment
- Handles command output logging - STDOUT
- Allows QShell/PASE commands to be submitted
- **Don’t need an SSH session or direct STRQSH call**
- Can be called directly from RPG or CL

MobiGoGo
Work at the Speed of Mobility

# QSHBASH - Run Bash Command via QShell

- Simplifies bash script calls
- **Convenience wrapper over bash calls**
- No need to type: **bash –c** to run your bash command.
- Don’t need an SSH session or call to STRQSH
- All the same benefits of QSHEXEC because it **uses QSHEXEC**
- Can be called directly from RPG or CL

# QSHPYRUN - Run Python Script via QShell



- **Convenience wrapper over Python calls**
- Simplifies Python 2 or Python 3 script calls
- No need to type **python2** or **python3** command
- Script path and script name passed individually
- Up to 40 parameters can be passed to script
- All the same benefits of QSHEXEC because it **uses QSHEXEC**
- Can be called directly from RPG or CL

MobiGoGo
Work at the Speed of Mobility

# QSHLOGSCAN – Scan QShell Log for Value

- Scans the QTEMP/STDOUTQSH outfile for an anticipated value
- Great way to check your results file for information
- Scans log line by line via RPG command processing program
- Looks for a specific value or a partial match if desired
- Returns CPF9898 Escape message if no value found
- Completes normally if value was found

# QSHIFSCHK – Check for IFS file existence

- Check if IFS file or directory exists
- Send CPF9898 escape message if no file/dir
- Send CPF9897 escape message if file/dir found
- You should monitor for both CPF9897 and CPF9898 when using

MobiGoGo
Work at the Speed of Mobility

# QSHPATH – Set Open-Source Package Path

- Adds **/QOpenSys/pkgs/bin** to QShell **PATH** environment variable
- Used by QSHEXEC, QSHBASH and QSHPYRUN  if SETPKGPATH = *YES

MobiGoGo
Work at the Speed of Mobility

# QSHEXEC - Run QShell Command Line

```
QSHEXEC CMDLINE('ls /tmp')
        SETPKGPATH(*YES)
        DSPSTDOUT(*YES)
        LOGSTDOUT(*NO)
        PRTSTDOUT(*NO)
        DLTSTDOUT(*YES)
        IFSSTDOUT(*NO)
        IFSFILE(' ')
        IFSOPT(*REPLACE)
        CCSID(*SAME)
        PRTSPLF(QSHEXECLOG)
        PRTUSRDTA(*NONE)
        PRTTXT(*NONE)
```

# QSHEXEC Command Temporary Objects



- **/tmp/qsh** (Temporary log file IFS location)
- Use **ERASE '/tmp/qsh/*'** CL command to clear IFS dir periodically
- **QSHONITMP library**
- Can use QSHONITMP library for temporary files or objects
- QSHONITMP auto-created by QSHell on i commands
- **CLRLIB QSHONITMP** periodically or after system IPL to keep clean

# CMDLINE and SETPKGPATH parameters



- **CMDLINE** - Takes any QShell or PASE command line
- **SETPKGPTH** adds **/QOpenSys/pkgs/bin** to search path
- Ensures IBM Open-Source Package Management Yum packages found

# Logging Command Output

- Outfile **QTEMP/STDOUTQSH** is always created with STDOUT data
- There are several additional options for handling STDOUT logs
- Or you can CPYF QTEMP/STDOUTQSH and append to your own log files

# Displaying Standard Output - *YES

**DSPSTDOUT - *YES – Display standard output result**

- Set this setting to *YES if you want to display console results
- After running, the stdout log output from /tmp/qsh IFS file for job is always captured to outfile QTEMP/STDOUTQSH so it can be used.
- After running QShell/PASE program or command, the STDOUT log data is displayed interactively from the outfile.
- **You would Only use this mode for testing and debugging when you need to see QShell/PASE console output logs.**

# Displaying Standard Output - *NO

**DSPSTDOUT - *NO – DO NOT Display standard output result**

- Set this setting to *NO if you don't want to display console results
- After running, the stdout log output from /tmp/qsh IFS file for job is always captured to outfile QTEMP/STDOUTQSH so it can be used.
- After running the command, the STDOUT log outfile can be used to selectively process log information returned from QShell/PASE program call via STDOUT.
- **You would use this mode for production where you might want to selectively process only certain messages in the log file.**
- **We utilize this OUTFILE to write the STDOUT messages to joblog as well if that option is enabled.**

# Log Standard Output - *YES

**LOGSTDOUT – Log standard output to job log**

- Set this setting to *YES if you want to write STDOUT to the main job log
- After running the command, the STDOUT log data is written to the calling jobs job log.
- Each console message will have a CPF message id of: **QSS9898**
- Job log data can be used for debugging
- Job log data can be captured for use in subsequent job steps
- **You would normally use this option in production perhaps when you want to capture QShell/PASE output to your job log.**
- **If you log LOTS of messages to STDOUT, you probably want to leave this setting to *NO or you could overload your joblog with messages.**

# Print Standard Output - *YES

**PRTSTDOUT – Print standard output result**

- Set this setting to *YES if you want to print your command log
- After running the command, the STDOUT log data is written to a spool file
- **Spool file name, user data and print text** can be specified
- **You would normally use this option in production perhaps when you want to capture STDOUT log output to a spool file for auditing rather than an OUTFILE or the job log.**

# Delete Standard Output - *YES

**DLTSTDOUT – Delete standard output result**

- Set this setting to *YES to delete the IFS log file after processing
- After running, the stdout log output from /tmp/qsh IFS file for job is always captured to an outfile so it can be used.
- STDOUT data is optionally written to the joblog or printed to a spool
- Finally the temporary STDOUT log file in /tmp/qsh is deleted if this setting is set to *YES which is the default
- **This parameter should normally ALWAYS be set to *YES unless you're debugging an unknown problem. Normally you always want to clean up these files since the data gets captured to an OUTFILE automatically before cleanup anyway**

# Copy STDOUT to IFS File - *YES

**IFSSTDOUT – Copy standard output result to IFS file**

- Set this setting to *YES to copy or append STDOUT to IFS file
- Allows /tmp/qsh temporary IFS log file to be copied or appended to IFS
- Finally the temporary STDOUT log file in /tmp/qsh is deleted if this setting is set to *YES which is the default
- Useful to aggregate log info to a single IFS log file.
- **This parameter should normally ALWAYS be set to *NO unless you want to copy or aggregate STDOUT data to a single IFS file location or directory before deleting the temporary IFS stdout log.**

MobiGoGo
Work at the Speed of Mobility

# Integrating QShell/PASE Calls with IBM i Jobs

- Use the **QSHEXEC, QSHBASH or QSHPYRUN** command in **QSHONI** lib
- Allows QShell/PASE calls to be embedded in CL, RPG and COBOL
- Called via standard QCMDEXC mechanism
- Pass complete commands with parameters in to QShell/Pase calls
- Receive return parameters from calls via Console/STDOUT log
- Pipeline STDOUT directly back to job so RPG/CL/COBOL can process any response information and check for errors in the logs.
- Send STDOUT to IFS file, outfile, job log or print file

# Qshell on i Demo



Demo

# Wrap Simple QShell/Pase Commands

- List files in IFS directory with ls command

  **QSHEXEC CMDLINE('cd /rpgopensource; ls -l') DSPSTDOUT(*YES)**

- Locate files starting with **Send***

  **QSHEXEC CMDLINE('find /rpgopensource -name Send*')
  DSPSTDOUT(*YES)**

- Call shell script to ping an IP address – **PASE/Qshell has no PING cmd**

  **QSHEXEC CMDLINE('cd /rpgopensource; pingi.sh "8.8.8.8"')
  DSPSTDOUT(*YES)**

# Scan the STDOUT Return Outfile Log for Value

- List contents of directory
  **QSHEXEC CMDLINE('cd /rpgopensource; ls -l') DSPSTDOUT(*NO)**
- View outfile
  **DSPPFM FILE(QTEMP/STDOUTQSH)**
- Look for a file named: **pingi.sh** on any line in file
  **QSHLOGSCAN SCANFOR(pingi.sh) EXACTMATCH(*NO)**

MobiGoGo
Work at the Speed of Mobility

# Hello World Python Sample

- Simple Hello world example
- Illustrates the plumbing and how it works a couple ways
- Python script – **helloworld.py**
- Calling via **QSHEXEC** or **QSHPYRUN**
- Calling via convenience wrapper command: **HELLOWORLD**

MobiGoGo
Work at the Speed of Mobility

# Use Python 3 Qsh/Pase/Bash Command Line

- Python 3 command line.
- Call from pase or bash command line or shell script

**HelloWorld.py “This is a test”**

**python3 HelloWorld.py “This is a test”**

# Use QSHPYRUN Command

- Call Python via **QSHPYRUN** or **QSHEXEC** Command
- Call interactively from RPG/CL, schedule or submit to batch

```
QSHONI/QSHEXEC CMDLINE('python3 /rpgopensource/helloworld.py
"This is a test from IBM i"') DSPSTDOUT(*YES)

QSHONI/QSHPYRUN SCRIPTDIR('/rpgopensource')
    SCRIPTFILE(helloworld.py)
    ARGS('This is a test from IBM i') DSPSTDOUT(*YES)
```

MobiGoGo
Work at the Speed of Mobility

# Use HELLOWORLD Wrapper Command

- Make calling Python more user friendly
- Provides context to the actual function being performed
- Call interactively from RPG/CL, schedule or submit to batch

```
HELLOWORLD MESSAGE('This is my Hello World Message’)
  DSPSTDOUT(*YES)
```

MobiGoGo
Work at the Speed of Mobility

# Send Authenticated Email via Office 365

- Company has moved mail from On-Prem Exchange to Office 365
- Need new way to send authenticated emails via O365
- Python script with O365 functionality – **SendO365.py**
- **QSHPYRUN** to run Python script from standard job
- **SNDMAIL365** command to wrap call to Python/QSHPYRUN
- Wrapper command makes the Python much more user-friendly

MobiGoGo
Work at the Speed of Mobility

# Use Python 3 Qsh/Pase/Bash Command Line

- Python 3 command line.
- Call from pase or bash command line or shell script

**SendMail365.py "richard@richardschoen.net" "richard@richardschoen.net" "O365 Test from IBM i" "This is a test from IBM i" "/tmp/excel.xlsx" "richard@richardschoen.net" "“**

**python3 SendMail365.py "richard@richardschoen.net" "richard@richardschoen.net" "O365 Test from IBM i" "This is a test from IBM i" "/tmp/excel.xlsx" "richard@richardschoen.net" "“**

# Use QSHPYRUN Command

- Call Python via QSHPYRUN Command
- Call interactively from RPG/CL, schedule or submit to batch

```
QSHONI/QSHPYRUN SCRIPTDIR('/rpgopensource')
    SCRIPTFILE(SendMail365.py)
    ARGS('richard@richardschoen.net'
    'richard@richardschoen.net'
    'O365 Test from IBM i - QSHPYRUN'
    'This is a test from IBM i'
    '/rpgopensource/excel.xlsx'
    'richard@richardschoen.net' ' ')
```

MobiGoGo
Work at the Speed of Mobility

# Use SNDMAIL365 Wrapper Command

- Make calling Python more user friendly
- Provides context to the actual function being performed
- Call interactively from RPG/CL, schedule or submit to batch

```
SNDMAIL365 FROMEMAIL('richard@richardschoen.net')
           TOEMAIL('richard@richardschoen.net')
           SUBJECT('O365 Test from IBM i - SNDMAILO365')
           BODYTEXT('This is the body text')
           ATTACHFILE('/rpgopensource/excel.xlsx')
           O365USER(richard@richardschoen.net)
           O365PASS(' ')
           DSPSTDOUT(*YES)
```

MobiGoGo
Work at the Speed of Mobility

# RPG Read JSON from Web and Write to CUSTCDT

- Sample to Import JSON data from a web site to Physical File
- **All Work Called From RPG program**
- Simple JSON retrieval from web site
- Can get more sophisticated with REST calls
- Read JSON file to IFS file using https **(Python)**
- RPG program uses YAJL to read and parse the JSON file **(RPG/YAJL)**
- RPG program writes results to physical file
- Scott Klement's YAJL Site to install YAJL library
  http://www.scottklement.com/yajl

MobiGoGo
Work at the Speed of Mobility

# Python Directory Crawler Example

- Crawl IFS Directory Tree (Including IFS files and Libraries)
- Capture output to a tilde delimited IFS text file
- CPYFRMIMPF of data from IFS text file to PF – DIRCRAWL
- Use RPG program or SQL to read and process the PF

MobiGoGo
Work at the Speed of Mobility

# Python Directory Crawler Example

- Crawl selected dir tree

```
QSHONI/QSHEXEC CMDLINE(‘/rpgopensource/pydircrawl.py
"/python" "/tmp/dircrawl.txt" "true" "false" "~" "true"')
DSPSTDOUT(*YES)
```

- Copy contents of IFS text file to PF

```
CPYFRMIMPF FROMSTMF('/tmp/dircrawl.txt')
TOFILE(QSHONI/DIRCRAWL)
        MBROPT(*REPLACE)
        RCDDLM(*LF)
        FLDDLM('~')
        FROMRCD(2)
```

MobiGoGo
Work at the Speed of Mobility

# Python Directory Crawler Example

- Create SQL Table for Directory Crawler Data

```
create table qshoni.dircrawl
(IFSFULL VARCHAR(256),
IFSFILE VARCHAR(256),
IFSPREFIX VARCHAR(256),
IFSEXT VARCHAR(10),
IFSSIZE DECIMAL(15,2),
IFSTYPE VARCHAR(5),
IFSSYMLNK VARCHAR(5));
```

# Python Learning Links

- There are tons of free and paid learning resources
- Seiden Group offers paid Python training classes
  http://www.seidengroup.com
- Learn Python - Full Course for Beginners [Tutorial]
  https://www.youtube.com/watch?v=rfscVS0vtbw
- Freecodecamp
  https://www.freecodecamp.org/news/search/?query=python
- Google
  https://www.google.com/search?q=python+free+training

# Python Code Samples

- Read Web Pages Sample

  https://zetcode.com/python/requests/

MobiGoGo
Work at the Speed of Mobility

# Next Steps

- Install Python 3 and Other IBM i Open Source
- Install QShell on i Library

  https://github.com/richardschoen/Qshoni
- Try example Python scripts and IBM i Code Samples

  https://github.com/richardschoen/RpgOpenSource
- Start using it. Create your own Qshell/PASE scripts
- Give me feedback or examples of how you are using this technology

  richard@richardschoen.net