



Your Presenter

30+ yr developer



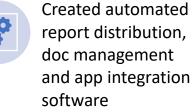


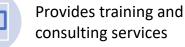
모티 Software Systems IBM i, Windows, Linux, Mac in 1990 Sold business in Started MobiGoGo 2014 when partner to build multi platform apps retired Contributes open Author of iForGit Git source to Github client for RDI, SEU and ¥Ξ repositories PDM developers http://github.com/richardschoen http://www.iforgit.com Business web site: http://www.mobigogo.net Personal web site: http://www.richardschoen.net

LinkedIn: https://www.linkedin.com/in/richardschoen

Twitter: @richardschoen

Started RJS





consulting services

Richard Schoen richard@richardschoen.net

-orrichard@mobigogo.net

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) (2004 –current)
- XMLSERVICE –universal open source IBM i DB access (2012current)
- Editor/IDE choices: RDI, Eclipse, Visual Studio, Visual Studio Code, Notepad++, etc...
- (1984–today : Green screen to smartphones) The greatest computing era !!







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





How to use shells on the IBM

- A good intro to various shells on IBM i
- Andy Youens FormaServe
- <u>https://www.youtube.com/watch?v=9rL9U8hfIHA</u>
- 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 without using HTTPGETCLOB etc.

Send and receive email

Crawl IFS directories

Become productive quickly

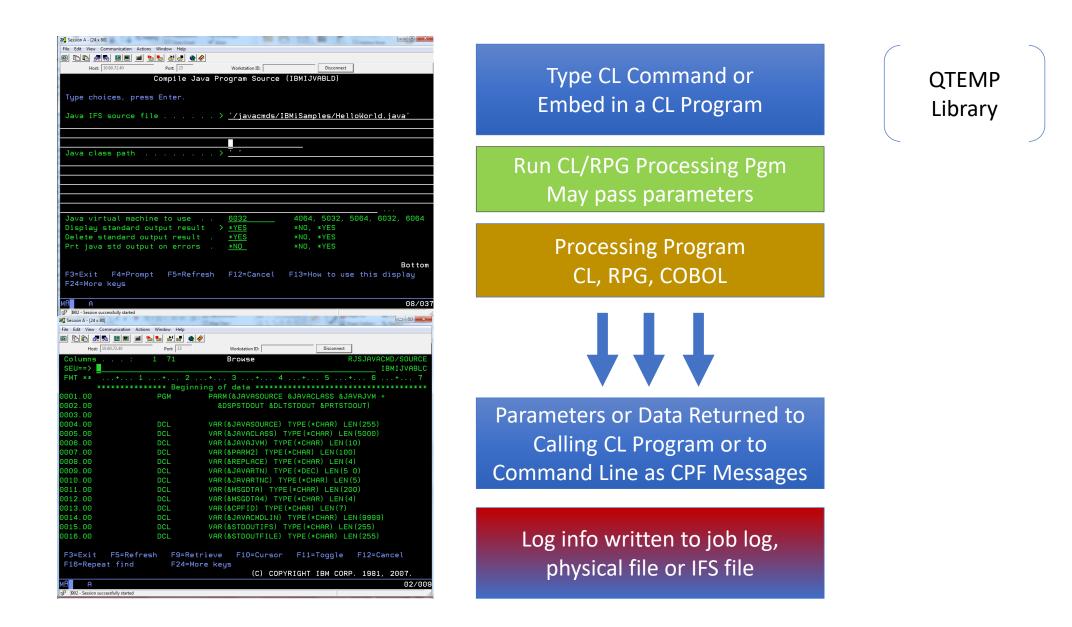
Integrates or extends existing IBM i apps across boundaries

Qshell on I can be used for all kinds of IBM i utility programs

Run any Qshell or PASE code from traditional programs



Traditional CL Command/Program Call Flow





Command/Program Call Flow using QShell/PASE



	tion Actions Window Help		
	🔳 📓 🐭 🐭 🔳	٠	
Host: 10.60.72.40	Port: 23	Workstation ID:	Disconnect
	Compile	Java Program Source	≥ (IBMIJVABLD)
Type choices,	press Enter.		
Tous TES cour	co filo	\ //ipupemde	/IBMiSamples/HelloWorld.java'
Java IFS Soul	ce file	· · · / / Javaciiius/	
Java class pa	ath	\dots \longrightarrow $\sum_{i=1}^{n-1}$	
Java virtual	machine to use	e <u>6032</u>	4064, 5032, 5064, 6032, 6064
	dard output res		*NO, *YES
	ard output resu		*NO, *YES
Prt java std	output on erro	ors . <u>*NO</u>	*NO, *YES
			Botto
F3=Exit F4=	Promot F5=Re	fresh F12=Cancel	F13=How to use this display
F24=More keys			
A A			08/0
1902 - Session successfully s	tarted		
Session A - [24 x 80]		P. Smith Gill	
	tion Actions Window Help		
	🔳 🖬 🏡 🌄 🕷 🔳		
Host: 10.60.72.40	Port: 23	Workstation ID:	Disconnect
Host: 10.60.72.40	Port: 23		RJSJAVACMD/SOURCE
Host: 10.60.72.40 Columns SEU==>	Port 23	Workstation ID: Browse	RJSJAVACMD/SOURCE
Host: 10.60.72.40 Columns SEU==>	Port 23 : 1 71 1+ 2	Workstation ID: Browse +3	RJSJAVACMD/SOURCE IBMIJVABLC 4+ 5+ 6+ 7
Host 10.60.72.40 Columns SEU==> FMT **+. ******	Port 23 : 1 71 1+ 2	Workstation ID: Browse + 3+ nning of data *****	RJSJAVACMD/SOURCE IBMIJVABLO 4+ 5+ 6+ 7
Host 10.60.72.40 Columns SEU==> FMT **+. ******	Port 23 : 1 71 1+2 ********** Begi	Workstation ID: Browse + 3+ nning of data ***** PARM (&JAVASOURC	RJSJAVACMD/SOURCE IBMIJVABLC 4+ 5+ 6+ 7
Hot 10.00.7240 Columns SEU=> _ FMT **+. ****** 0001.00 0002.00 0003.00	Pont 23 : 1 71 1+2 ********** Begi PGM	Workstation ID: Browse + 3+ nning of data ***** PARM (&JAVASOURC &DSPSTDOUT &E	RJSJAVACMD/SOURCE IBMIJVABLC 4+ 5+ 6+ 7 **********************************
Hot 10.60.7240 Columns SEU=>> FMT **+. ****** 0001.00 0002.00 0003.00 0004.00	Port 23 : 1 71 1+2 **********************************	Workstation ID: Browse +3+ nning of data ***** PARM (&JAVASOURC &DSPSTDOUT &E VAR (&JAVASOURCE	RJSJAVACMD/SOURCE RIBMIJVABLC 4+ 5+ 7 ************************************
Hot: 10507240	Port 23 : 1 71 1+2 PGM DCL DCL	Workstation ID: Browse + 3+ prark (&JavAsSURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACLASS)	RJSJAVACMD/SOURCE IBMIJVABLC 4+ 5+ 6+ 7 E &JAVACLASS &JAVAJVM + LTSTDOUT &PRTSTDOUT) I) TYPE (*CHAR) LEN (255) TYPE (*CHAR) LEN (5000)
Het 10007240 Columns SEU==> FMT ** + . ******* 0001.00 0002.00 0003.00 0003.00 0005.00 0005.00	Port [2] : 1 71 1 + 2	Workstation ID: Browse *	RJSJAVACMD/SOURCE IBMIJVABLC 4 * 5 * 6 * 7 ************************************
Hott. 10507240 Columns SEU==> FMT **	Port [23 : 1 71 1 + 2 ********* Begi PGM DCL DCL DCL DCL DCL DCL	Worktation ID: Browse + 3+ nning of data ***** PARM (&JAVASOURC &DSPSTDOUT &C VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVAJVM) T VAR (&PARM2) TYF	RJSJAVACMD/SOURCE RJSJAVACMD/SOURCE IBMIJVABLC 4 + 5 + 6 + 7 ************************************
Het: 10:072.40 Columns SEU==> FMT **+. 70001.00 2002.00 2003.00 2004.00 2005.00 2005.00 2005.00 2007.00 2007.00 2008.00	Port [23 : 1 71 1 + 2 2 PGM DCL DCL DCL DCL DCL DCL DCL	Workstation ID: Browse Browse Anning of data ****** PARM (&JAVASOURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&APARM2) TYF VAR (&REPLACE) TY	RJSJAVACMD/SOURCE IBMIJVABLC 4+ 5+ 6+ 7 ************************************
Het: 10:072.40 Columns SEU==> FMT ** + . ******* 0001.00 0002.00 0003.00 0004.00 0005.00 0005.00 0005.00 0005.00 0005.00 0005.00 0005.00 0008.00	Port [2] : 1 71 1 + 2	Workstation ID: Browse + 3+ nning of data ***** PARM (&JAVASOURCE &DSPSTDOUT &C VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&ARAM2) TYF VAR (&AAVARTN) T VAR (&AAVARTN) T	RJSJAVACMD/SOURCE IBMIJVABLC 4+5+6+7 ************************************
Het: 10:072.40 Columns SEU==> FMT **+. 70001.00 2002.00 2003.00 2004.00 2005.00 2005.00 2005.00 2007.00 2007.00 2008.00	Port [23 : 1 71 1 + 2 2 PGM DCL DCL DCL DCL DCL DCL DCL	Workstation ID: Browse Drowse DRRM (&JAVASOURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVARTN) T VAR (&JAVARTN) T VAR (&JAVARTN)	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC SUPERATION #*
Het: 10:072.40 Columns SEU==> FMT **+. 2001.00 2002.00 2003.00 2004.00 2004.00 2005.00 2005.00 2005.00 2007.00 2007.00 2008.00 2008.00 2008.00 2010.00 2011.00	Port [23 : 1 71 1 + 2	Workstation ID: Browse Browse Anning of data ***** PARM (&JAVASOURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVARCE) VAR (&JAVARTNC) VAR (&JAVARTNC) VAR (&MSGDTA) TY	RJSJAVACMD/SOURCE IBMIJVABLC 4+5+6+7 ************************************
Het 10007240 Columns SEU==> FMT ** + . 0001.00 0002.00 0003.00 0005.00 0006.00 0006.00 0006.00 0006.00 0006.00 0006.00 0006.00 0000.00 0008.00 00	Port [23 : 1 71 1 + 2 PGM DCL DCL DCL DCL DCL DCL DCL DCL	Workstation ID: Browse Browse ARM (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&APARM2) TYV VAR (&AREPLACE) T VAR (&JAVARTNC) VAR (&MSGDTA) T VAR (&MSGDTA) T	RJSJAVACMD/SOURCE RJSJAVACMD/SOURCE IBMIJVABLC 4+5+6+7 ************************************
Het: 10:07240 Columns SEU==> FMT ** 0001.00 0002.00 0003.00 0004.00 0005.00 0005.00 0005.00 0007.00 0006.00 0007.00 0007.00 0008.00 00010.00 0011.00 0011.00 0013.00 0014.00	Port [2] : 1 71 1 + 2	Worksteinen ID: Browse Browse Anning of data ***** PARM (&JAVASOURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVARCE) VAR (&JAVARTN) VAR (&JAVARTNC) VAR (&MSGDTA) TY VAR (&MSGDTA) TY VAR (&MSGDTA) TY VAR (&JAVACMDLIN	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC IBMIJVABLC Standbord Standbord E &JAVACLASS &JAVAJVM + LITSTDOUT &PRTSTDOUT) COMPE (*CHAR) LEN (255) TYPE (*CHAR) LEN (5000) YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (50) 'YPE (*CHAR) LEN (50) 'YPE (*CHAR) LEN (5) 'PE (*CHAR) LEN (5) 'PE (*CHAR) LEN (5) 'PE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (7) 'D 'E (*CHAR) LEN (7) 'D 'E (*CHAR) LEN (7)
Het 10007240 SEU=>> FMT **+. 0001.00 0002.00 0004.00 0005.00 0006.00 0005.00 0006.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00	Port [2] : 1 71 1 + 2	Worksteinen ID: Browse + 3+ PARM (&JAVASOURCE &DSPSTDDUT &C VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACHSI) VAR (&JAVACHSI) VAR (&JAVACHDIN VAR (&JAVACMDLIN VAR (&STDOUTIFS)	RJSJAVACMD/SOURCE IBMIJVABLC 4 * 5 * 6 * 7 ************************************
Het 10007240 SEU=>> FMT **+. 0001.00 0002.00 0004.00 0005.00 0006.00 0005.00 0006.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00005.00 00	Port [23 : 1 71 1 + 2 PGM DCL DCL DCL DCL DCL DCL DCL DCL	Worksteinen ID: Browse + 3+ PARM (&JAVASOURCE &DSPSTDDUT &C VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACHSI) VAR (&JAVACHSI) VAR (&JAVACHDIN VAR (&JAVACMDLIN VAR (&STDOUTIFS)	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC IBMIJVABLC Standbord Standbord E &JAVACLASS &JAVAJVM + LITSTDOUT &PRTSTDOUT) COMPE (*CHAR) LEN (255) TYPE (*CHAR) LEN (5000) YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (50) 'YPE (*CHAR) LEN (50) 'YPE (*CHAR) LEN (5) 'PE (*CHAR) LEN (5) 'PE (*CHAR) LEN (5) 'PE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (7) 'D 'E (*CHAR) LEN (7) 'D 'E (*CHAR) LEN (7)
Het: 10:072.40 Columns SEU==> FMT ** 0001.00 0002.00 0004.00 0005.00 0006.00 0007.00 0007.00 0007.00 0007.00 00010.00 0010.00 0011.00 0011.00 0013.00 0014.00 0015.00 0015.00	Port [23 : 1 71 1 + 2 PGM DCL DCL DCL DCL DCL DCL DCL DCL	Worksteinen ID: Browse Browse Anning of data ****** PARM (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&JAVACTNC) VAR (&STDOUTIFS) VAR (&STDOUTIFS) VAR (&STDOUTIFLE	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC IBMIJVABLC IBMIJVABLC Standbord ISMIJVACLASS &JAVAJVM + LITSTDOUT &PRTSTDOUT) ITYPE (*CHAR) LEN (5000) TYPE (*CHAR) LEN (5000) 'YPE (*CHAR) LEN (100) 'E *CHAR) LEN (100) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (255) 'YPE (*CHAR) LEN (255)
Het 10007240 SEU==> FMT **+. ****** 0001.00 0002.00 0004.00 0005.00 0006.00 0005.00 0005.00 0009.00 0001.00 0011.00 0011.00 0011.00 0011.00 0015.00 0014.00 F3=Exit F5=	Port [2] : 1 71 1 + 2	Worksteinen ID: Browse + 3+ nning of data ****** PARM (&JAVASOURCE &DSPSTDOUT &E VAR (&JAVASOURCE VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACLASS) VAR (&JAVACHAS) VAR (&JAVACHAS) VAR (&SEPLACE) T VAR (&JAVACHAS) VAR (&SEPLACE) VAR (&MSGDTA4) T VAR (&MSGDTA4) T VAR (&JAVACHASI) VAR (&JAVACHASI) VAR (&STDOUTFILE Setrieve F10=Curso	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC IBMIJVABLC IBMIJVABLC Standbord WALLASS &JAVAJVM + LITSTDOUT &PRTSTDOUT) DYPE (*CHAR) LEN (5000) YPE (*CHAR) LEN (5000) YPE (*CHAR) LEN (100) E &CHAR) LEN (100) YPE (*CHAR) LEN (5) PE (*CHAR) LEN (5) YPE (*CHAR) LEN (100) YPE (*CHAR) LEN (200) TYPE (*CHAR) LEN (200) TYPE (*CHAR) LEN (255) YPE (*CHAR) LEN (255)
Het: 10:07240 Columns SEU==> FMT ** 0001.00 0002.00 0004.00 0004.00 0006.00 0006.00 0007.00 0007.00 0007.00 00010.00 0010.00 0011.00 0011.00 0013.00 0014.00 0014.00 0015.00 0016.00	Port [2] : 1 71 1 + 2	Workstation ID: Browse ARM (&JAVASOURCE ADSPSTDUJT &C VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACUST) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&SIDUTT) VAR (&SIDUTTIS) VAR (&SIDUTTILE etrieve F10=Curso More keys	RJSJAVACMD/SOURCE IBMIJVABLC 4 * 5 * 6 * 7 ************************************
Her 10007240 Columns SEU==> FMT **+. 2001.00 2002.00 2003.00 2004.00 2005.00 2006.00 2006.00 2006.00 2008.00 2008.00 2008.00 2008.00 2008.00 2011.0	Port [2] : 1 71 1 + 2	Workstation ID: Browse ARM (&JAVASOURCE ADSPSTDUJT &C VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVASOURCE VAR (&JAVACUST) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&JAVARTN) VAR (&SIDUTT) VAR (&SIDUTTIS) VAR (&SIDUTTILE etrieve F10=Curso More keys	RJSJAVACMD/SOURCE IBMIJVABLC IBMIJVABLC IBMIJVABLC IBMIJVABLC Standbord ISMIJVACLASS &JAVAJVM + LITSTDOUT &PRTSTDOUT) ITYPE (*CHAR) LEN (5000) TYPE (*CHAR) LEN (5000) 'YPE (*CHAR) LEN (100) 'E *CHAR) LEN (100) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (100) 'YPE (*CHAR) LEN (200) 'YPE (*CHAR) LEN (255) 'YPE (*CHAR) LEN (255)

Type CL Command or Embed in a CL Program	QTEMP Library		
Run CL/RPG Processing Pgm May pass parameters			
Processing Program CL, RPG, COBOL			
Run QShell/PASE Program or Command	Qshell/PASE app runs in a thread job		
$\checkmark \checkmark \checkmark \checkmark$			
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



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
- 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



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
- Consume web services without HTTPGETCLOB/HTTPPOSTCLOB (Java)
- Compose, send and receive emails
- Talk to other databases (PostgreSQL, MariaDB, SQLite, others)



System Prerequisites



- IBM Open Source Package Management
- Install all Python 3 packages (if using Python)
- Install unixODBC
- Install IBMi Access ODBC Driver
- Install Qshell on I
- Uninstall 5733OPS if you can. Yum packages rule



Install All Python Packages



BM i Access Client Solutions	Den Source Package N	Vlanagement	- 🗆 X		
	File View Connection Utilities				
e Edit Actions Tools Help	Connection: richard@sysi1:/				
Welcome	Installed packages Updates available Available packages				ssions. 5250 Session Manager supports:
	Package	Version	Repository		
System: sysi1	v python3-Pillow	5.0.0-6	@ibm	^	N
	python3-asn1crypto	0.24.0-1	@ibm		
General	python3-bcrypt	3.1.4-6	@ibm		
Data Transfer	python3-cffi	1.11.5-3	@ibm		
5250 Emulator	python3-cryptography	2.8-0	@ibm		
Integrated File System	python3-dateutil	2.8.0-1	@ibm		
Navigator for i	python3-devel	3.6.12-1	@ibm		les.
SSH Terminal	python3-ibm_db	2.0.5.12-0	@ibm		
Printer Output	python3-idna	2.8-1	@ibm		
	python3-itoolkit	1.6.1-1	@ibm		
Database	python3-jinja2	2.11.2-1	@ibm		
Schemas	python3-lxml	4.2.1-4	@ibm		
Run SQL Scripts SQL Performance Control	python3-markupsafe	1.1.1-1	@ibm		
SQL Performance Center	python3-numpy	1.15.4-1	@ibm		
Console	python3-pandas	0.22.0-5	@ibm		
 5250 Console 	python3-paramiko	2.6.0-1	@ibm		
Virtual Control Panel	python3-pip	9.0.1-3	@ibm		
 Hardware Management Interface 1 	python3-psutil	5.5.1-1	@ibm		
- The second consideration of the second second	python3-psycopg2	2.8.5-1	@ibm		
	python3-pycparser	2.19-2	@ibm		
Management	python3-pynad	1.2.1-4	@ibm		
System Configurations	python3-pyodbc	4.0.27-0	@ibm		
5250 Session Manager	python3-pytz	2018.5-3	@ibm		
HMC Probe Utility	python3-pyyaml	5.3.1-1	@ibm		
	python3-pyzmq	17.1.2-0	@ibm		
	python3-rpm	4.13.1-12	@ibm		
	nython3-scikit-learn	0 19 1-8	Mihm	~	
	<			>	
	Done: 444 rows retrieved.			de.	
		Inform	ation Show files Rein	stall Remove	







• 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







- 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 STRQSH
- Can be called directly from RPG or CL





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 STRQSH
- All the same benefits of 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
- Can be called directly from RPG or CL







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 line by line
- 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





- 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





QSHPATH – Set Open-Source Package Path

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



QSHEXEC - Run Qshell Command Line



QSHEXEC CMDLINE('ls /tmp')

SETPKGPATH(*YES)

DSPSTDOUT(*YES)

LOGSTDOUT(*NO)

PRTSTDOUT(*NO)

DLTSTDOUT(*YES)

IFSSTDOUT(*NO)

IFSFILE(' ')

IFSOPT(*REPLACE)

PRTSPLF(QSHEXECLOG)

PRTUSRDTA(*NONE)

CCSID(*SAME)

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
- There are several additional options for handling STDOUT 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.





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.





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.





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.





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





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.





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



Integrate QShell/Pase Application via QShell on i



QSHEXEC, QSHBASH, QSHPYRUN CL command executed from CL, RPG or COBOL application with parameters

QSHEXEC, QSHBASH or QSHPYRUN CL command calls Qshell or PASE command

New Qshell/PASE process starts to run command or program

PASE Pgm/Cmd Runs until Completion – Console logs captured to /tmp IFS log and outfile in QTEMP

If Qshell/PASE command succeeds, exit code = 0. If error, exit code will be <> 0.

Qshell/PASE process ends after call to run command or app

Control returned to QSHEXEC, QSHBASH or QSHPYRUN CL command which sends escape or completion message

Control returned to original CL, RPG or COBOL application



Console STDOUT results to IFS

Console results to OUTFILE physical file STDOUTQSH in QTEMP







Demo



Hello World Python Sample



- Simple Hello world example
- Illustrates the plumbing and how it works





Python Directory Crawler Example

- Crawl Directory Tree
- Capture output to a tilde delimited IFS text file
- CPYFRMIMPF of data from IFS text file to PF DIRCRAWL
- RPG program to read and process the PF







• Install QShell on i Library

https://github.com/richardschoen/Qshoni

- Start using it
- Try example Python scripts

https://github.com/richardschoen/RpgOpenSource

Give me feedback or example of how you are using

richard@richardschoen.net