



COMMON EUROPE CONGRESS 2026
14 - 17 June
Lyon, France

The largest conference in Europe
for solutions around IBM Power (IBM i, AIX, Linux) & IBM Storage


common EUROPE www.comeur.org **common** FRANCE

LYON EVENTS | CENTRE DE CONGRÈS DE LYON

**Welcome to Lyon, France
and the 2026 Common Europe Congress**

**Bienvenue à Lyon, en France,
et au Congrès de Common Europe 2026**

1




Power Education

CEC 2026 Workshop

How to use IBM i bash scripting for PowerVS, COS
and a lot of open source applications

Version: 13.06.2026

2



Bash scripts are the **perfect complement to CL programming**. However, many connections are much easier to implement if you do this directly in a Bash script.


We use examples and procedures to show how **IBM PowerVS and COS** can be controlled using Bash.

Here, too, you can connect it in a few simple steps. As a third scenario, we show participants how to use various **open source tools** under Bash and IBM i. For example, **db2util, gzip, and many more**. An exciting presentation that highlights the further development of the system into an open source platform.

Excerpts from this presentation have already been successfully shown in German workshops.

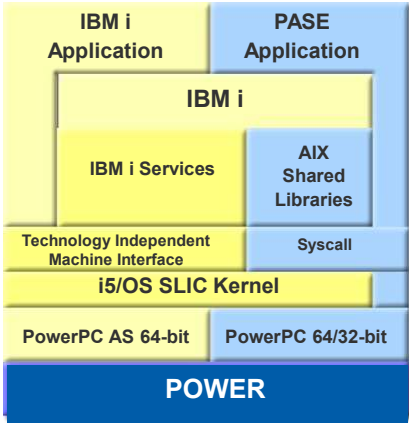
IBM i Automation with Bash Programming

3



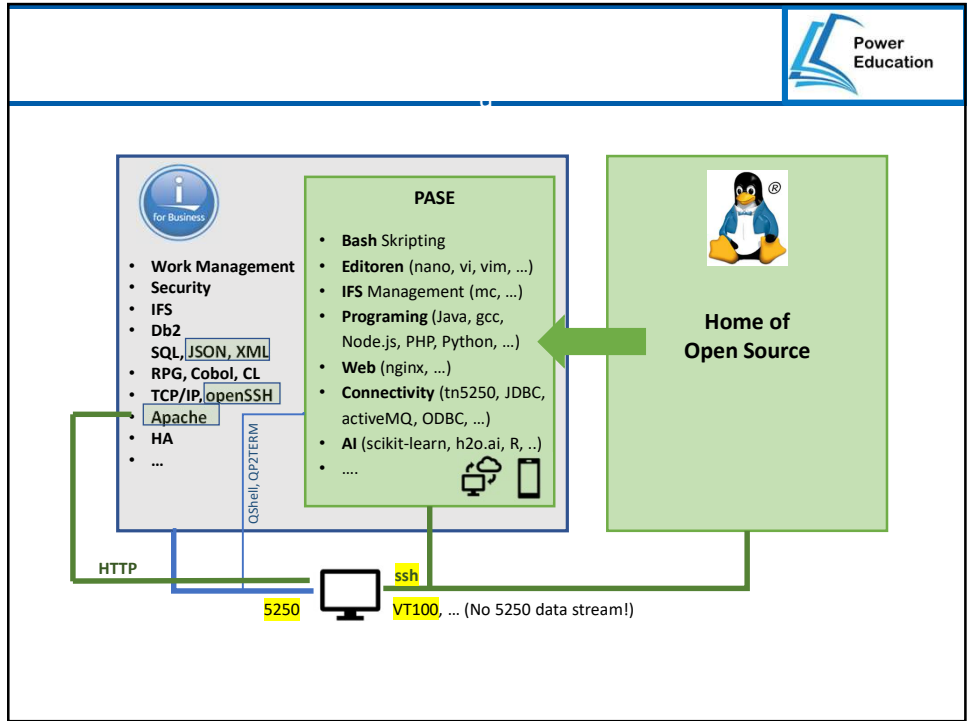
Portable Application Solution Environment (PASE)

- An integrated IBM i runtime for porting selected UNIX applications
 - Supports AIX 32/64-bit application model
 - Exploits PowerPC's ability to switch runtime modes
- Applications
 - Integrated with iSeries file systems and work management
 - Can call Db2, Java and ILE programs
 - Exploit all aspects of IBM i operations environment



IBM i Automation with Bash Programming

4



5


Where we should start today?

- PASE Shells in a 5250 terminal session: qshell, qp2term
- ssh with IBM i (& security)
- Useful bash commands
- CL command execution in PASE: system, cl
- Open Source tools: db2util, jq, sed, curl, ...
- IBM Cloud
- IBM Cloud SDK
- IBM Cloud Webservices

IBM i Automation with Bash Programming

6

IBM i Automation with Bash Programming




Examples of useful Open Source Software

© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

7

Installation von OpenSSH



1. Installation
 - RSTLICPGM LICPGM(5733SC1) DEV(OPTxx) OPTION(*BASE) RSTOBJ(*ALL) LNG(2924)
 - RSTLICPGM LICPGM(5733SC1) DEV(OPTxx) OPTION(1) RSTOBJ(*PGM)

 - Create an DSA or RSA Key-Pair
 - ssh-keygen -t dsa -N "" or ssh-keygen -t rsa -N ""

2. STRTCPSVR *SSHD

IBM i Automation with Bash Programming

8

8

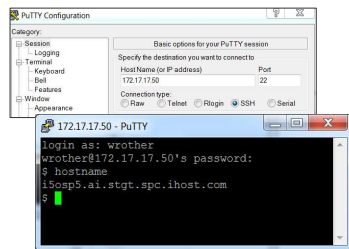


```
export PATH=$PATH$:/opt/freeware/bin
X11 Forwarding

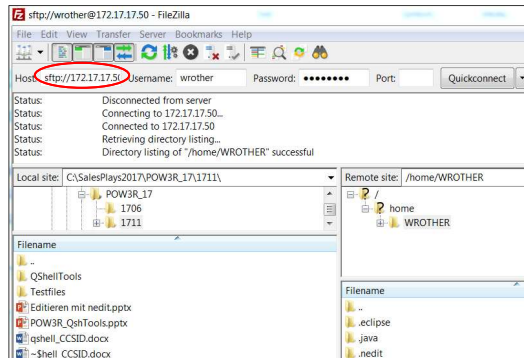
NO EBCDIC support -> Converting to CCSID 819
```

9

OpenSSH – Usage of ssh terminal und sftp



Putty



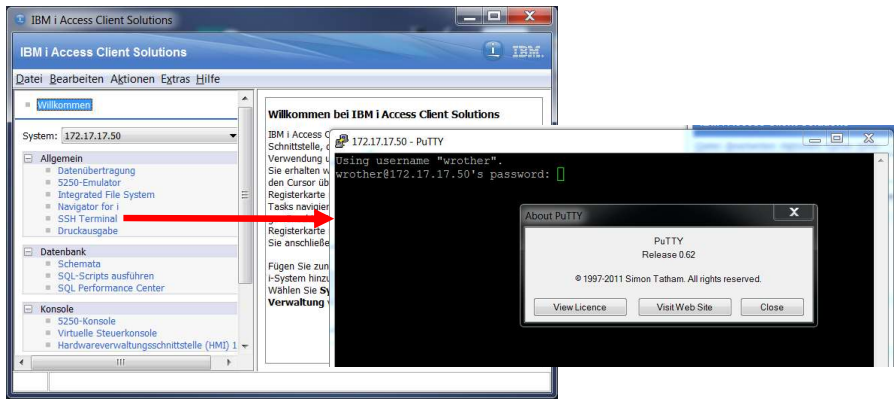
Filezilla

10

Open SSH Terminal with ACS

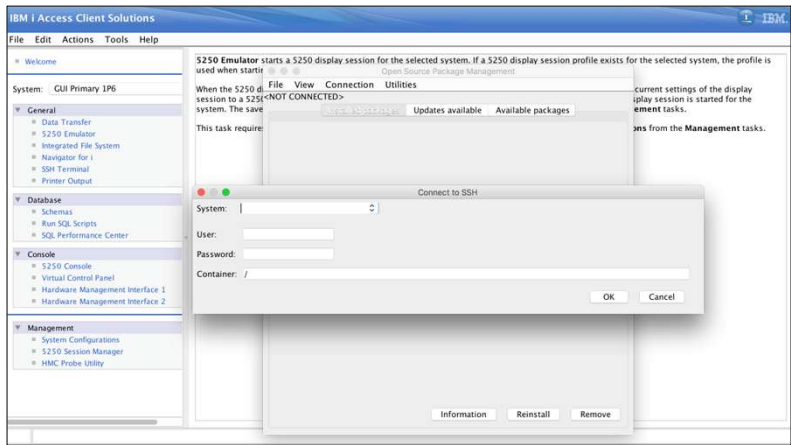


Launches an SSH client if it detects an **already-installed** Putty client




11

Open Source Package Management



12

Default SSH Shell


Set the current user's shell to BASH:

CALL QSYS2.SET_PASE_SHELL_INFO('*CURRENT', '/QOpenSys/QIBM/ProdData/OPS/tools/bin/bash'); →

```
172.17.17.26 - PuTTY
Using username "wrother".
wrother@172.17.17.26's password:
$ echo $0
-ksh
$ cd t*
ksh: cd: bad substitution
$
```

Set the default shell to be ksh for any users that do not have an explicit shell set.

CALL QSYS2.SET_PASE_SHELL_INFO('*DEFAULT', '/QOpenSys/usr/bin/ksh');


```
1 CALL QSYS2.SET_PASE_SHELL_INFO('*CURRENT', '/QOpenSys/QIBM/ProdData/OPS/tools/bin/bash');
2
3 select AUTHORIZATION_NAME, PASE_SHELL_PATH from QSYS2.USER_INFO where AUTHORIZATION_NAME='WROTHERR';
```

AUTHORIZATION_NAME	PASE_SHELL_PATH
WROTHERR	/QOpenSys/QIBM/ProdData/OPS/tools/bin/bash


```
172.17.17.26 - PuTTY
Using username "wrother".
wrother@172.17.17.26's password:
-bash-4.3$ cd t*
-bash-4.3$ pwd
/home/WROTHERR/test
-bash-4.3$ echo $0
-bash
-bash-4.3$
```

13

IBM i Automation with Bash Programming




Konfigurieren von ssh/sftp und bash



© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

14



starten, stoppen – autostart

Configure your ssh environment:
https://ibmi-oss-docs.readthedocs.io/en/latest/user_setup/README.html


ssh security
 Exercise: usage of password-less signon (key based)

sftp (Example: PTF Download)

ssh user@system or install MobaXterm: <https://mobaxterm.mobatek.net/>

IBM i Automation with Bash Programming

15



SSH password-less signon (key based)

```
mkdir -p $HOME
cd $HOME
pwd

/QOpenSys/pkgs/bin/chsh -s /QOpenSys/pkgs/bin/bash


touch $HOME/.profile
setccsid 1208 $HOME/.profile
echo 'PATH=/QOpenSys/pkgs/bin:$PATH' >> $HOME/.profile
echo 'export PATH' >> $HOME/.profile

ssh-keygen
cat $HOME/.ssh/id_rsa.pub >> $HOME/.ssh/authorized_keys
rm $HOME/.ssh/id_rsa.pub
chown "$(/QOpenSys/usr/bin/id -u -n)" $HOME
chown -R "$(/QOpenSys/usr/bin/id -u -n)" $HOME/.ssh
chmod 0755 $HOME
chmod 0700 $HOME/.ssh
chmod 0644 $HOME/.ssh/authorized_keys
```

Now try: `ssh user@system 'cl "crtlib cec"'`

IBM i Automation with Bash Programming

16

sshd_config – Overriding default settings


DSPUSRPRF USRPRF(SSHGRP) TYPE(*GRPMBR)

```

Display Group Members
User profile . . . . . : SSHGRP
BARLEN
THOMAS
WROTHER
                    
```

```


-bash-5.2$ id wrother
uid=1200(wrother) gid=109(sshgrp)
                    
```

sshd_config

```

AllowGroups sshgrp
# Example of overriding settings on a per-user basis
#Match User anoncvs
#   X11Forwarding no
#   AllowTcpForwarding no
#   ForceCommand cvs server
Match User tomtest1
ChrootDirectory /home
AllowTCPForwarding no
X11Forwarding no
ForceCommand internal-sftp
                    
```

17

bash


Technique #1: chsh

1. Use yum to install the `chsh` package (for instance, `yum install chsh`)
2. From a shell, use the `chsh` command to set your shell (for instance, `chsh -s /QOpenSys/pkgsg/bin/bash`). You can set the shell for another user via the `-u` option (for instance, `chsh -s /QOpenSys/pkgsg/bin/bash -u otherusr`).

Technique #2: sql

You can set bash to be your default shell by running the following command from anywhere you have an SQL context, such as the Run SQL Scripts tool:

```
CALL QSYS2.SET_PASE_SHELL_INFO('*CURRENT', '/QOpenSys/pkgsg/bin/bash')
```

You can also set bash to be the default shell for all users, by running:

```
CALL QSYS2.SET_PASE_SHELL_INFO('*DEFAULT', '/QOpenSys/pkgsg/bin/bash')
```


Or, for a specific user:

```
CALL QSYS2.SET_PASE_SHELL_INFO('OTHERUSR', '/QOpenSys/pkgsg/bin/bash')
```

IBM i Automation with Bash Programming
18

18

Bash Configuration Files




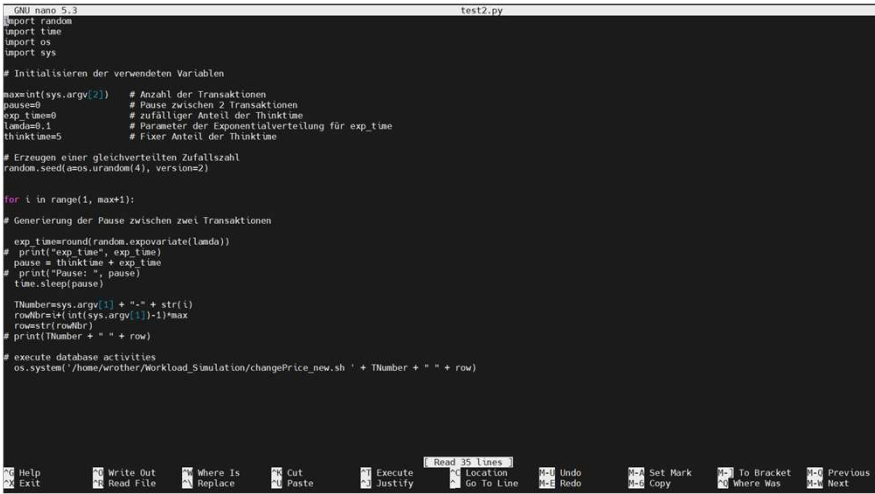
<div style="text-align: center;"> ↓ ↓ ↓ time </div>	<p>/etc/profile</p> <p>.profile</p> <p>.bashrc</p>	<p>Sourced by all user processes</p> <p>Sourced by login shells</p> <p>Login shells source these files in the user's home directory</p>
--	---	---

IBM i Automation with Bash Programming
19

19

nano – a terminal-based editor (ssh-terminal)





```

GNU nano 5.3 test2.py
import random
import time
import os
import sys

# Initialisieren der verwendeten Variablen
maxInt(sys.argv[2]) # Anzahl der Transaktionen
pause# # Pause zwischen 2 Transaktionen
exp_time# # zufälliger Anteil der Thinktime
lamda=#.1 # Parameter der Exponentialverteilung für exp_time
thinktime# # Fixer Anteil der Thinktime

# Erzeugen einer gleichverteilten Zufallszahl
random.seed(os.urandom(4), version=2)


for i in range(1, max+1):
# Generierung der Pause zwischen zwei Transaktionen
exp_time=round(random.expovariate(lamda))
print("exp_time", exp_time)
pause = thinktime * exp_time
print("Pause: ", pause)
time.sleep(pause)

Number=sys.argv[1] + "-" + str(i)
rowNr=(int(sys.argv[3])-1)*max
rowestr(rowNr)
print(Number + " " + row)

# execute database activities
os.system('/home/wrother/Workload_Simulation/changePrice_new.sh ' + Number + " " + row)
    
```

IBM i Automation with Bash Programming
20

20

vim


Beliebter Terminaleditor

```


SELECT
  REMOTE_ADDRESS as RemoteAddr, REMOTE_PORT as RmtPort, REMOTE_PORT_NAME as RmtPortName,
  LOCAL_ADDRESS, LOCAL_PORT as Port, LOCAL_PORT_NAME as PortName,
  CONNECTION_TYPE as TYPE,
  TRIM(AUTHORIZATION_NAME) AS AUTH_NAME, JOB_NAME, SLIC_TASK_NAME
FROM QSYS2.NETSTAT_JOB_INFO
{0} -- WHERE CLAUSE
...
ORDER BY LOCAL_PORT, LOCAL_ADDRESS, REMOTE_PORT, REMOTE_ADDRESS

sql = sql.format("WHERE LOCAL_PORT = ?") if args.port is not None else sql.format("")
params = (args.port,) if args.port is not None else None
if args.limit is not None:
    sql += "\n LIMIT {0}".format(args.limit)
if args.offset is not None:
    sql += "\n OFFSET {0}".format(args.offset)

cur.execute(sql, params)
# One advantage to using a spec-compliant driver is that it works well with many Python modules, li
tabulate(),
# which understands what to do with a standard cursor object. https://pypi.python.org/pypi/tabulate
print(tabulate(cur, 'keys'))
cur.close()

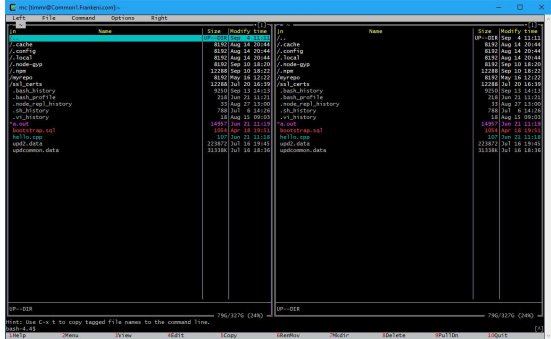
except:
    
```

21

Midnight Commander (MC)


File Manager – a Norton Commander Clone
Alternative zu ls, cp, rm, mv, chmod, ...


- FTP Access
- Compare Files
- Compare Directories
- Edit Files
- Move, Copy, Delete, usw.
- Subshell
- Scroll Wheel
- Mouse
- Directory Side-by-Side und Tree Views
- See and modify file info
- ...



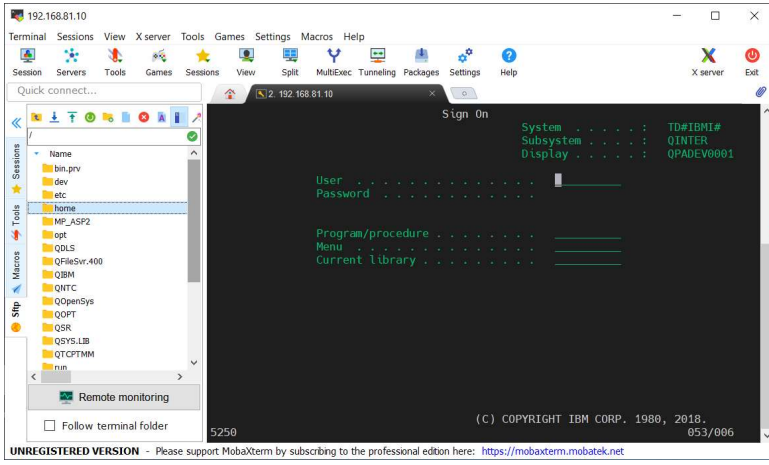
IBM i Automation with Bash Programming
22

22

tn5250 / xt5250




Running using ssh in IBM i



IBM i Automation with Bash Programming
23

23

tn5250 - US keyboard layouts only



~		@	#	\$	%	^	&	*	()	-	+	←	Backspace
Tab	Q	W	E	R	T	Y	U	I	O	P	{	}		\
Caps Lock	A	S	D	F	G	H	J	K	L	:	"	'	Enter	
Shift	Z	X	C	V	B	N	M	<	>	?	/	'	Shift	
Ctrl	Win Key	Alt									Alt	Win Key	Menu	Ctrl

Keyboard Mapping <http://plinux.perso.libertysurf.fr/tn5250-faq.html#FAQ-1999-12-07-03>

The following table lists the 5250 functions implemented by tn5250, and the corresponding key presses.

Keys are represented as Emacs does:
 C-a means hold Ctrl and press A,
 M-a means press Esc or C-g followed by A, and
 C-M-a means press Esc or C-g followed by C-a.

[1] Which keys generate f11-f24, backtab, backspace and delete is very dependent on the configuration of the terminal.
 [2] + and - only work as Field+ and Field- in signed numeric fields.

Function	Keypress
F1 - F10	f1 to f10, M-1 to M-
F11	f11 [1], M=-
F12	f12 [1], M=-
F13 - F24	f13 to f24 [1], M-1 to M-+
Enter	return, enter, C-j, C-m
Left	left
Right	right
Up	up
Down	down
Roll Up	next, pagedown, C-d, C-f
Roll Down	prev, pageup, C-b, C-u
Backspace	backspace [1]
Home	home, C-o
End	end
Insert	insert, M-i, M-delete
Delete	delete [1]
Reset	C-r, M-r
Print	C-p, M-p
Help	M-h
SysReq	C-c, M-s
Clear	M-c
FieldExit	C-k, C-x, M-x
TestReq	C-t
Toggle	M-t
Erase	C-e
Attn	C-a, M-a
Dup	M-d
Field+	+ [2], same as FieldExit
Field-	M-m, - [2]
NewLine	C-M-j
Next Field	tab, C-l
Prev Field	backtab [1]
Refresh	C-l, M-l
Quit	C-q

IBM i Automation with Bash Programming
24

24

wget – Download from a list



Example: Software
Download from ESS

Produkt/Release	Downloadpaket	Größe (MB)
5770-551	Details IBM i	21275
07.06.00	Pakete 5861:17.6 B_GROUP1 v07.06.00.ENLJ2V0	15219
	B_GROUP1_01_IBM_i_RS_760-00	
	↓ B_GROUP1_01_IBM_i_RS_760-00_LCD8_2952_00.usdf	3258
	B_GROUP1_02_IBM_i_RS_760-00	
	↓ B_GROUP1_02_IBM_i_RS_760-00_LCD8_2953_00.usdf	3219
	B_GROUP1_03_IBM_i_RS_760-00	
	↓ B_GROUP1_03_IBM_i_RS_760-00_LCD8_2954_00.usdf	922
	B_GROUP1_04_IBM_i_RS_760-00	
	↓ B_GROUP1_04_IBM_i_RS_760-00_LCD8_2955_00.usdf	2800
	B_GROUP1_05_IBM_i_RS_760-00	---

```
wget -i ESS_selected_HTTP_download_links\{3\}.txt &
```

IBM i Automation with Bash Programming

25

25

cURL



curl is a tool for transferring data from or to a server using URLs. It supports these protocols: DICT, FILE, FTP, FTPS, GOPHER, GOPHERS, HTTP, HTTPS, IMAP, IMAPS, LDAP, LDAPS, MQTT, POP3, POP3S, RTMP, RTMPS, RTSP, SCP, SFTP, SMB, SMBS, SMTP, SMTPS, TELNET, TFTP, WS and WSS

Examples:

Get a README file from an FTP server: `curl ftp://ftp.example.com/README`

Get a directory listing of an FTP site: `curl ftp://ftp.example.com/`


Get a webpage and store in a local file with a specific name:
`curl -o thatpage.html http://www.example.com/`

Get a file from an HTTP server that requires user and password, using a proxy:
`curl -u user:passwd -x my-proxy:8888 http://www.example.com/`


IBM i Automation with Bash Programming

26


26

man – Search for Manual Pages	 Power Education
Install package: man-db (man command and related utilities)	
IBM i Automation with Bash Programming	
27	

27

IBM i Automation with Bash Programming	 Power Education
PASE Shells	
<small>© Copyright IBM Corporation 2001, 2016, 2020, 2023 Course materials may not be reproduced in whole or in part without the prior written permission of IBM.</small>	


28

PASE


- QSH, QP2TERM
- QShell Utilities
 - <https://www.ibm.com/docs/en/i/7.5.0?topic=languages-i-pase-shells-utilities>

IBM i Automation with Bash Programming

29

Konfigurieren der QShell


STRQSH oder QSH
PATH

Redirection of the output „>“ bzw. „>>“

Set and Export of Variables

Watch out!!!

```


> set PARM100=10;echo $PARM100
$
> export PARM100=10;echo $PARM100
10
$
            
```

```


> export PARM100=100
$
> echo $PARM100
100
$
> qsh
$
> export PARM100=10
$
> echo $PARM100
10
$
> exit
$
> echo $PARM100
100
$
            
```

IBM i Automation with Bash Programming
30

30

QP2TERM	 Power Education
call qp2term	
If no parameters are specified, QP2TERM runs the default IBM PASE for i shell as an interactive login shell.	
The default IBM PASE for i shell is an implementation of the Korn shell , with path name /QOpenSys/usr/bin/sh.	
IBM i Automation with Bash Programming	
31	

31

Executing CL Commands	 Power Education
<ul style="list-style-type: none">• system<ul style="list-style-type: none">• System Utility • cl<ul style="list-style-type: none">• Built-in command	

32

Ausführung von CL Befehlen - Examples



```
cl dsplibl
```

```
cl "ENDTCPSVR *DDM"
```

```
cl "STRTCPSVR *DDM"
```

```
LPAR Info: cl call QLZARCAPI
```

33

What Is a Shell Script?



A readable text file which can be edited with a text editor

- `vi shell_prog`
- `nano shell_prog`
- `EDTF shell_prog`

Anything that you can do from the shell prompt

A program, containing:

- System commands
- Variable assignments
- Flow control syntax
- Shell commands


And comments

- Comment line starts with #
- `#!/bin/bash` is not a comment if `#!` is in the first position (shebang)

IBM i Automation with Bash Programming

34

IFS CCSID Issues



Default: QShell CCSID = Job CCSID of the job, often 273 or 37

-> problem when transferring to e.g. Windows workstations, because EBCDIC character sets are not supported and usually no automatic conversion takes place

Solution: **export QIBM_CCSID=819**

- i.e. conversion to ASCII
- alternatively ANSI CCSID 1252 or Unicode UTF-8 codepage 1208 (Not all tools support Unicode!)

Example:

cat ccsid273.txt > ccsid273cat819.txt

ccsid273.txt - Notepad

@, f,,@ñòóó%@À@j@Ð%@cücÜ@a%


ccsid273cat819.txt - Notepad

abcd 1234 ä ö ü []{} /

IBM i Automation with Bash Programming
35

35

CCSID Konvertierung mit CL:



Copy Object (CPY)

Type choices, press Enter.

```
Object . . . . . > '/home/WROTHOR/ccsid273.txt'
To directory . . . . . '/'
To object . . . . . ccsid273cpy819.txt
Symbolic link . . . . . *NO *NO, *YES
From CCSID . . . . . *OBJ 1-85533, *OBJ, *PCASCI...
To CCSID . . . . . 819 1-85533, *OBJ, *CALC...
Data Format . . . . . *text *BINARY, *TEXT
Directory subtree . . . . . *NODIR *NODIR, *NONE, *ALL
Replace object . . . . . *NO *NO, *YES
Owner . . . . . *NEW *NEW, *KEEP
Authority . . . . . *OBJ
```

ccsid273cpy819.txt - Notepad

abcd 1234
ä ö ü

IBM i Automation with Bash Programming
36

36

Code Page Demo

```

##### show codepage issues by connecting via ssh

cat curl_test_1.sh          ### codepage 273

##### copy curl_test_1.sh to codepage 819 or 1208 within a 5250
session

CPY OBJ('/home/WROTHER/curl_test_1.sh')
TOOBJ('/home/WROTHER/curl_test_1_4CE') TOCCSID(819)
DTAFMT(*TEXT)

##### show the new file

cat curl_test_1_4CE.sh      ### codepage 819
                    
```

IBM i Automation with Bash Programming
37

37

export QIBM_CCSID=819

Example: `system dsplbl > libl.txt`

Display Attributes

```

Object ..... /home/WROTHER/libl.txt
Type ..... STMF
Owner ..... WROTHER
System object is on ..... Local
Auxiliary storage pool ..... 1
Object overflowed ..... No
Coded character set ID ..... 819
                    
```

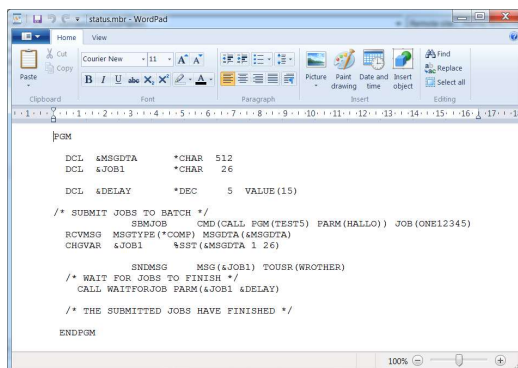
IBM i Automation with Bash Programming
38

38

Copying (non-DB2) members to IFS



```
qsh_out -c 'cat /QSYS.LIB/ROTHER.LIB/QCLSRC.FILE/status.mbr' > status.mbr
```



Open Source Tool in "TOBI"(aka BOB): <https://ibm.github.io/ibmi-bob/#/>

39

db2 in ssh session



```

-bash: /usr/bin/db2: Permission denied
-bash-5.2$ qsh_out -c "db2 'select * from qivs.qcustcdt'"
CUSNUM  LSTNAM  INIT  STREET          CITY  STATE  ZIPCOD  CDTLMT  CHGCOD  BALDUE  CDTDUE
-----
938472  Henning  G K   4859 Elm Ave    Dallas TX  75217  5000    3        37.00   0.00
839283  Jones   B D   21B NW 135 St   Clay NY   13041  400     1        100.00  0.00
392859  Vine    S S   P0 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   1.50
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

12 RECORD(S) SELECTED.
    
```

40

iconv (install iconv-gnu)

There are 2 versions available!
 PASE and an OpenSource package

```
-bash-5.1$ iconv -f CP01140 -t UTF-8 /QIBM/USERDATA/OS400/TCPIP/NTP/QTOT20251217
NTP Client started.
NTP Client Activity Log /QIBM/USERDATA/OS400/TCPIP/NTP/QTOT20251217 12/17/25 5:18:51.947 AM
TCP9184 Using Client address 10.3.29.11 to connect server 10.3.29.10 mode 3 version 4 minpoll 6 maxpoll 10 keyid 00000000
TCP9146 Using time server 10.3.29.10.
TCP9185 12/17/25 5:23:47.677 AM Synchronized client to server 10.3.29.10 stratum 1.
TCP9187 12/17/25 5:23:47.678 AM Client clock was slewed for offset -0.000 s.
TCP9146 Using time server 10.3.29.10.
TCP9187 12/17/25 5:27:30.803 AM The combine offset of servers is 0.000 S.
TCP9197 12/17/25 5:27:30.804 AM Client clock was slewed for offset 0.000 s.
TCP9146 Using time server 10.3.29.10.
TCP9187 12/17/25 5:37:25.700 AM The combine offset of servers is 0.000 S.
TCP9197 12/17/25 5:37:25.700 AM Client clock was slewed for offset 0.000 s.
-bash-5.1$
```

iconv [OPTION...] [-f encoding] [-t encoding] [inputfile ...]

To get a list of supported encodings: `iconv -l`

IBM i Automation with Bash Programming
41

41

iconv -l


```
-bash-5.2$ iconv -l
BIG5-HKSCS
CNS11643.1986-1
CNS11643.1986-2
GB18030
GBK
IBM-1006
IBM-1046
IBM-1124
IBM-1129
IBM-1251
IBM-1252
IBM-850
IBM-856
IBM-921
IBM-922
IBM-932
IBM-943
IBM-eucCN
IBM-eucJP
IBM-eucKR
IBM-eucTW
IBM-sbdTW
IBM-udcJP
IBM-udcTW
ISCII.1991
ISO8859-1
ISO8859-15
ISO8859-2
ISO8859-3
ISO8859-4
ISO8859-5
ISO8859-6
ISO8859-7
ISO8859-8
ISO8859-9
```

```
-bash-5.1$ cat while_test.sh
%-bash-5.1$ @[@[@[@[@[@[@[@[@[@[@@^c
-bash-5.1$
-bash-5.1$ iconv -f CP01140 -t UTF-8 while_test.sh
max=100
index=0
while AA $index -lt $max ÜÜ ; do
echo Index is $index
let index+=1
done
-bash-5.1$
-bash-5.1$ iconv -f CP01141 -t UTF-8 while_test.sh
max=100
index=0
while [[ $index -lt $max ]] ; do
echo Index is $index
let index+=1
done
-bash-5.1$
```

IBM i Automation with Bash Programming
42

42

iconv Use Cases




- - Import OF exported Certificates in pem files
 - Import only works with CCSID 1208
- Copy with cp of files in ssh
 - change CCSID to 819

IBM i Automation with Bash Programming 43

43

IBM i Automation with Bash Programming



Flow Control

© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

44

The Simple *if - then - else* Construct



```

if expression1
then
commands to be executed if expression1 is true
else
commands to be executed if all expressions are false
fi

```

```

if [[ "$UID" -lt "100" ]]
then
export PS1="# "
else
export PS1="$ "
fi

```

45

Boolean Expressions Examples



```

-bash-5.2$ if [[ 1 == 0 ]] ; then echo wahr; else echo falsch; fi
falsch
-bash-5.2$ if [[ 1 == 1 ]] ; then echo wahr; else echo falsch; fi
wahr
-bash-5.2$ if [[ 1 -eq 0 ]] ; then echo wahr; else echo falsch; fi
falsch
-bash-5.2$ if [[ 1 -eq 1 ]] ; then echo wahr; else echo falsch; fi
wahr
-bash-5.2$ if [[ $HOSTNAME = IAXEL.TDLAB.LOCAL ]] ; then echo wahr; else echo falsch; fi
Wahr
-bash-5.2$ if [[ $HOSTNAME = IAXEL.TDLAB.LO ]] ; then echo wahr; else echo falsch; fi
Falsch
-bash-5.2$ if [[ $HOSTNAME = IAXEL.TDLAB.LO* ]] ; then echo wahr; else echo falsch; fi
Wahr
-bash-5.2$ if [ $HOSTNAME = IAXEL.TDLAB.LO* ] ; then echo wahr; else echo falsch; fi
falsch

```

46

if Example



```
#!/usr/bin/bash
# Usage: goodbye username
#
if [[ $# -ne 1 ]]
then
    printf "Usage is: goodbye username"
    printf "Please try again."
    exit 1
fi
rmuser $1
printf "O.K., $1 is removed."
```

When we run "goodbye", this is what we get ...

```
$ goodbye
Usage is: goodbye username
Please try again.
$ goodbye pete
O.K., pete is removed.
```

47

Conditional Loop Syntax



```
until cc prog.c
do
    vi prog.c
done
```

This program sets up a continuous code-edit-compile loop to help you fix coding errors. It attempts to compile your C program (`cc prog.c`). If it fails, it opens the file in a text editor (`vi prog.c`) so you can fix the bugs, and repeats this cycle until the code compiles perfectly.

```
while [ "$x" -lt 3 ]
do
    lsps -a >> ./statfile
    df >> ./statfile
    let x=x+1
done # should be "< ./statfile" here
```

48

while true Example



The Script "forever" is a tough cookie!

```
#!/usr/bin/bash
# An endless loop with a trap for INT QUIT TSTP
trap 'printf "hasta la vista - baby!"' 2 3 18
while true
do
    printf "I'll be back."
    sleep 10
done

$ forever
I'll be back.           every ten seconds
I'll be back.           the script speaks!
I'll be back.

<Ctrl-c>               an attempt to stop it...

hasta la vista - baby!  invokes the trap, and
I'll be back.           it carries on.
I'll be back.
```

49

for Loop Syntax



```
for identifier in word1 word2 ...
do
  commands using $identifier
  more commands
done
```

```
for identifier # equivalent to: for identifier i"$@"
do
  commands using $identifier which takes values from the
  positional parameters
done # optional < filename
```

50

for - in Loop Example



Here we have a quick tidy-up to delete files:

```
$ for varfile in *.tmp
> do
>   rm -f $varfile
> done
$ _
```

Why use the option `-f` ?
What else could be `tested`?

51

for Loop Example



The sample Script "getprice.sh" will look up the price list:

```
#!/usr/bin/bash
# getprice.sh - select price from "pricelist" file
# for each item entered on the command line
# Usage: getprice item1 item2 ...

for item
do
    grep -i "$item" /home/cashier/pricelist
done

$ getprice.sh "Shock Absorbers" "Air Filter"
Front Shock Absorbers      49.99
Rear Shock Absorbers      59.99
Air Filter                  10.99
$ _
```

52

Arithmetic for Loop



The arithmetic for loop is available in ksh93 and bash.

```
for (( initialize; test; increment ))
do
  commands
done
```

Example:

```
for (( num=0; num <5; num++ ))
do
  mv file${num} file${num}.bkup
done
```

53

The case Statement



```
case word in
  ( pattern1 | pattern2 | ... )
    action      ;;
  (*) default  ;;
esac
```

```
case $identifier in
  (pattern1)                command1
                             more_commands ;;
  (pattern2 | pattern3)    commands ;;
  (*)                       commands ;;
esac
```

54

case Code Example



A guessing game of sorts:

```
#!/usr/bin/bash
# Usage:  match string
# To see how lucky you are feeling today

case "$1" in
  Ace )      print "You are really close."      ;;
  King )     print "Missed it by that much."   ;;
  Queen )    print "Finally!"                  ;;
  Jack )     print "Maybe next time."         ;;
  Ten|10 )   print "Getting closer."           ;;
  * )       print "Guess again."              ;;
esac
```

55

case Code Example Output



A casino dealer in the making?

```
$ match Three
Guess again.

$ match Jack
Maybe next time.

$ match Ace
You are really close.

$ match King
Missed it by that much.

$ match Queen
Finally!
```

56

The Shell select Syntax



```
select identifier in word1 word2 ...
do
commands using $identifier usually containing a case statement
done
```

```
select identifier # equivalent to: select identifier in "$@"
do
commands using $identifier from positional parameters usually
containing a case statement
done
```

57

select Code Example




To help identify animals we have a "barn.sh" Shell Script:

```
#!/usr/bin/bash
# usage: barn.sh
PS3="Pick an animal: "
select animal in cow pig dog quit
do
    case $animal in
        (cow) printf "Moo"           ;;
        (pig) printf "Oink"          ;;
        (dog) printf "Woof"         ;;
        (quit) exit                  ;;
        (') printf "Not in the barn" ;;
    esac
done
```

58

break the Loop



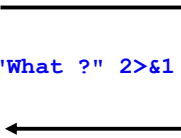
The `break` command jumps out of `do . . . done` loops:

- Exits from the smallest enclosing loop
- Jumps out a specified *number* of layers/loops

```

break number


select choice in Backup Restore Quit
do
  case $choice in
    (Backup) find . -print|backup -iqf /dev/rfd0
    ;;
    (Restore) restore -xqf /dev/rfd0
    ;;
    (Quit)   break
    ;;
    ('')     print "What ?" 2>&1
    ;;
  esac
done
    
```



IBM i Automation with Bash Programming

59

continue the Loop




The `continue` command begins the next iteration of a `do . . . done` loop:

- Starts at the top of the smallest enclosing loop
- Begins again a specified *number* of layers/loops out

```

continue number


$ for File in *
> do
> if [[ -d $File ]]
> then
>   continue
> fi
> file $File
> done
$ _
    
```



IBM i Automation with Bash Programming

60

Program Logic Constructs Example



Here's a Script to delete empty files:


```
#!/usr/bin/bash
# Usage: delfile file1 file2 ...
while [[ $# -gt 0 ]]
do
    if [[ -f "$1" ]]
    then
        if [[ ! -s "$1" ]]
        then
            rm "$1" && print "$1" deleted
        else
            print "$1" not deleted 2>&1
        fi
    elif [[ -d "$1" ]]
    then
        print "$1" is a directory
    else
        print "$1" is a special file
    fi
    shift
done
```

File test operators:
<https://tldp.org/LDP/abs/html/fto.html>

IBM i Automation with Bash Programming

61

IBM i Automation with Bash Programming




More Examples

Qshell Utility Examples

© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

62

"Calculator"


expr examples

IBM Power Systems

```
$ var1=6
$ var2=3
$ expr $var1 / $var2
2

$ expr $var1 - $var2
3
```

=> Use `\(\)` to group expressions:

```
$ expr \( $var1 + $var2 \) \* 5
45
```


=> Use `command substitution` to store the result in a variable:

```
$ var3=$(expr $var1 / $var2)
$ echo $var3
2
```

© Copyright IBM Corporation 2011, 2013. All Rights Reserved.
 US Government Users Restricted Rights - Use duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Figure 8-5. expr examples
AN732.0

63

For Loops


```
> cat Datei_1
Das ist Datei 1
$
```

```
> cat Datei_2
Das ist Datei 2
$
```


```
> cat Datei_3
Das ist Datei 3
$
```

```
> for i in 1 2 3
> do
> echo "Das ist Datei $i" >> Datei_all
> done
$
> cat Datei_all
Das ist Datei 1
Das ist Datei 2
Das ist Datei 3
$
```

IBM i Automation with Bash Programming
64

64

While



Erstellen einer Testdatei:

```

> cat crtstdata.sh

max=10000
index=0
rm testdaten 2> /dev/null
while [[ $index -lt $max ]] ; do
echo $RANDOM >> testdaten
let index+=1
done

$
                
```

Unterdrückung von Fehlermeldungen: 2 > /dev/null

Erzeugung von Zufallszahlen


```

> echo $RANDOM
53064
$
> echo $RANDOM
23016
$
> echo $RANDOM
53514
$
                
```

IBM i Automation with Bash Programming
65

65

Head and Tail



head and tail commands

The head command can be used to view the *first few lines* of a file or files.

```
head [-number_of_lines] file(s)
```

```
$ head -5 myfile
$ ls -l | head -12
```

The tail command writes a file to standard output, *beginning at a specified point*.

```
tail [-number_of_lines | +starting_line_number] file(s)
```

```
$ tail -20 file
$ tail +20 file
```

Erstellen einer kleineren Testdatei:

```


> head -3 testdaten > tstdata
$
> tail -3 testdaten >> tstdata
$
> cat tstdata
0: 6319
1: 20859
2: 50525
9997: 2113
9998: 39844
9999: 48145
$
                
```

© Copyright IBM Corporation 2011, 2013. All Rights Reserved.
 US Government Users Restricted Rights - Use duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Figure 5-5. head and tail commands AN732.0

IBM i Automation with Bash Programming
66

66


WC


Anzahl Zeilen mit Word Count **wc**


```

> wc testdaten
10000 20000 117172 testdaten
$
> wc -l testdaten
10000 testdaten
$
        
```

67

sed


stream editor



There are several ways of running **sed**:

```

sed 'edit-instructions' filename
command | sed 'edit-instructions'
sed -f command.file filename
        
```

Note: The input file is not changed or overwritten by **sed**!

There are two versions: OSS sed (GNU) and IBM i utility sed with different behavior !

68

sed



There are two versions: OSS sed (GNU) and IBM i utility sed with different behavior !

- /QOpensys/pkgs/bin/sed
- /QOpensys/usr/bin/sed
 - Reprogrammed by IBM for IBM i

```
-bash-5.2$ find /QOpensys -name sed -exec ls -l {} \;
-rwxr-xr-x 1 qsys 0 36525 Dec 12 2024 /QOpensys/QIBM/ProdData/OS400/PASE/bin/sed
lrwxrwxrwx 1 qsys 0 76 Apr 29 2025 /QOpensys/usr/bin/sed -> ../QIBM/ProdData/OS400/PASE/bin/sed
-rwxr-xr-x 1 qsys 0 1696485 Nov 25 2019 /QOpensys/pkgs/bin/sed
-bash-5.2$ which sed
/QOpenSys/pkgs/bin/sed
-bash-5.2$
```

Example: lparinfo:

```
#!/QOpensys/pkgs/bin/bash
system "call QLZARCAPI" | sed 's/\. /\n/g' | sed 's/CPF9897: //g'
```

69

sed and printf Examples



This instruction changes data:

Syntax: `s/old string/new string/`


Some examples

1. To replace the first occurrence of "Smith" on each line with "Smythe"
`sed 's/Smith/Smythe/' phone.list`
2. To replace the same as above using a different delimiter
`sed 's!Smith!Smythe!g' phone.list`
3. To replace every match in a line, add the "global"
`printf xxx | sed 's/x/y/' # responds with yxx`
`printf xxx | sed 's/x/y/g' # responds with yyy`
`printf xxx | sed 's/x/y/2' # responds with yxy`

LPAR Info: `cl call QLZARCAPI | sed "s/\. /\n/g"`

70

printf for Formatted Printing




- Better printing formats required
 - Use `printf`
 - **printf syntax:** `printf (fmt [, args])`
 Parentheses are optional
 - `fmt` is usually a string constant with format specifications
- Specifiers are like the C language `printf`
- Format specification: `%<char>`
 - `%s` string
 - `%d` decimal integer
 - `%f`, `%e` floating point (fixed or exponent notation)
 - `%o` unsigned octal
 - `%%` literal percent

IBM i Automation with Bash Programming

71

printf Formats




- Format specification strings can use modifiers
 - `%-width.precision`
 - If width used, contents are right justified
 - Use `-` (minus/hyphen) after `%` to left justify
 - Precision controls
 - Number of digits to right of decimal point for numeric values
 - Maximum number of characters to print for string values

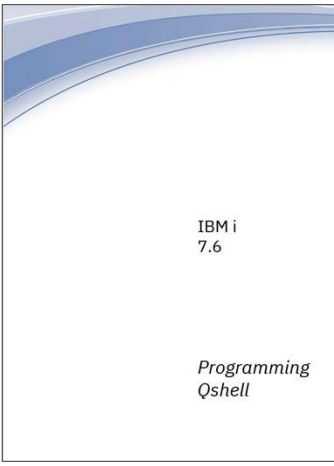
```
printf "%#10s#\n" "Hello"

printf "#\n%30s# \n%d\n" "Hello" 1234567890
```

IBM i Automation with Bash Programming

72





IBM i
7.6


Programming
Qshell

National language support (NLS) considerations.....	29
Performance considerations.....	34
Developing your own utilities.....	34
Editing files with Qshell Interpreter.....	35
Differences with other interpreters.....	35
Utilities.....	35
Define aliases.....	41
alias.....	41
unalias.....	42
Run commands.....	42
builtin.....	42
command.....	43
dot.....	44
env.....	44
eval.....	45
iii.....	
exec.....	45
exit.....	46
help.....	46
nchup.....	47
qsh.....	48
rexec.....	49
rxx.....	50
source.....	51
system.....	51
type.....	53
whence.....	53
xargs.....	54
Manage data.....	55
rmm.....	55

https://www.ibm.com/docs/en/ssw_ibm_i_76/pdf/rzahzpdf.pdf

IBM i Automation with Bash Programming

73



"Portscan" mit ACS

```

#!/usr/bin/bash

portlist=$(echo $2 | sed "s/ //g")
java -jar /home/WROTHER/ACS/acsbundle.jar /plugin=ping /system=$1 /port=$portlist
          
```

```

-bash-5.2$ cwbping.sh 172.29.65.20 "21, 22, 80, 992"
Verifying connection to port 21 service ... Success!      using port number 21
Verifying connection to port 22 service ... Success!      using port number 22
Verifying connection to port 80 service ... Failed **** MSGSOCK007 - Could not connect.
(A remote host refused an attempted connect operation. (connect failed))      us
ing port number 80
Verifying connection to port 992 service .. Success!      using port number 992
Done
          
```

IBM i Automation with Bash Programming

74

Task: Onboarding User



Create a new user mit username=user, a new password, that he becomes a member of a group and create a library owned by him. If no library name was given, name it as the user.

```
newuser.sh name password group library
```

IBM i Automation with Bash Programming

75


Lab: How many users are there on the system?



```
db2util "select AUTHORIZATION_NAME from QSYS2.USER_INFO_BASIC" | wc -l
```

76

Example from day 1



```

#!/bin/bash
#
# task: creation date for each user profile on the system
#

userlist=$(db2util "select AUTHORIZATION_NAME from QSYS2.USER_INFO_BASIC")

for name in $userlist
do
    CREATED=$(cl dspusrprf $name | grep "Creation ")
    USRPRF=$(echo $name | sed "s/\\/\\/g")
    printf "$USRPRF : $CREATED\n"
done
    
```


```

-bash-5.2$ dspuser.sh
ADSYNC : Creation date/time . . . . . : 30.10.19 15:11:47
APPTEAM : Creation date/time . . . . . : 22.11.17 13:11:50
BARLEN : Creation date/time . . . . . : 19.05.17 17:02:58
DB_USER : Creation date/time . . . . . : 05.09.22 17:00:34
DBATEAM : Creation date/time . . . . . : 22.11.17 13:11:51
DBUSR4TST : Creation date/time . . . . . : 14.09.22 05:55:58
DB2USER : Creation date/time . . . . . : 22.09.22 22:17:11
    
```

IBM i Automation with Bash Programming

77

cut



Optionen von cut

Option	Beschreibung
-b oder --bytes	Angabe der Bytepositionen
-c oder --characters	Angabe der Zeichenpositionen (entspricht derzeit -b)
-d oder --delimiter	Angabe des Trennzeichens (Delimiter) anstelle des Tabulators (Standard)
-f oder --fields	Angabe der zu extrahierenden Felder
-s oder --only-delimited	Keine Ausgabe von Zeilen ohne Trennzeichen (Delimiter)

```

echo "123456789" | cut -c 5-
56789
        
```

```

-bash-5.2$ echo 'one_two_three_four_five' | cut -d '_' -f4
four
-bash-5.2$ echo 'one_two_three_four_five' | cut -d '_' -f2,4
two_four
-bash-5.2$ echo 'one_two_three_four_five' | cut -d '_' -f2-4
two three four
        
```

```

-bash-5.2$ db2util -o csv "select * from qiws.qcustcdt fetch first 2 rows only"
"938472","Henning ","G K","4859 Elm Ave ","Dallas","TX","75217","5000","3","37.00","0.00"
"839283","Jones ","B D","21B NW 135 St","Clay ","NY","13041","400","1","100.00","0.00"
-bash-5.2$
-bash-5.2$ db2util -o csv "select * from qiws.qcustcdt fetch first 2 rows only" | cut -d, -f2
"Henning "
"Jones "
    
```

IBM i Automation with Bash Programming

78

Open from yesterday (1/2)



```
#!/bin/bash
#
# task: creation date for each user profile on the system
#
userlist=$(db2util "select AUTHORIZATION_NAME, CREATION_TIMESTAMP from QSYS2.USER_INFO" | sed "s/\\/\\/g")
for name in $userlist
do
  USRPRF=$(echo $name | cut -d',' -f1)
  CREATED=$(echo $name | cut -d',' -f2)
  # printf "$USRPRF" wurde am $CREATED erstellt \n"
  printf "%-11s wurde erstellt am %s\n" "$USRPRF" "$CREATED"
done
```

```
-bash-5.2$ dspuser2.sh
ADSYNC wurde am 2019-10-30-15.11.47.000000 erstellt
APPTTEAM wurde am 2017-11-22-13.11.50.000000 erstellt
BARLEN wurde am 2017-05-19-17.02.58.000000 erstellt
DB_USER wurde am 2022-09-05-17.00.34.000000 erstellt
DBATEAM wurde am 2017-11-22-13.11.51.000000 erstellt
DBUSR4TST wurde am 2022-09-14-05.55.58.000000 erstellt
DB2USER wurde am 2022-09-22-22.17.11.000000 erstellt
```

79

Open from yesterday (2/2)



- Formating:

- `printf "%-5s: %f\n" "Abe" 3.5 "John" 5 "It" 8.221`

```
[wrother@centos79 ~]$ printf "%-5s: %f\n" "Abe" 3.5 "John" 5 "It" 8.221
Abe : 3.500000
John : 5.000000
It : 8.221000
```

- String Editing: https://openbook.rheinwerk-verlag.de/shell_programmierung/shell_004_002.htm#RxxKap00400204004DAE1F021174

80

Open from yesterday (2/2) - Solution



```
#!/bin/bash
#
# task: creation date for each user profile on the system
#
userlist=$(db2util "select AUTHORIZATION_NAME from QSYS2.USER_INFO_BASIC")
for name in $userlist
do
  CREATED=$(cl dspusrprf $name | grep "Creation " | cut -c50-)
  USRPRF=$(echo $name | sed "s/\\/\\/g")
  printf "%-11s wurde erstellt am %s\n" "$USRPRF" "$CREATED"
done
```

```
-bash-5.2$ dspuser2a.sh
ADSYNC   wurde erstellt am 2019-10-30-15.11.47.000000
APPTEAM  wurde erstellt am 2017-11-22-13.11.50.000000
BARLEN   wurde erstellt am 2017-05-19-17.02.58.000000
DB_USER  wurde erstellt am 2022-09-05-17.00.34.000000
DBTEAM   wurde erstellt am 2017-11-22-13.11.51.000000
```

81

split and csplit

**Example: check_sshd.sh**

```
#Delete all comment lines
cat sshd_config | sed '/^[[:space:]]*#/d' > temp.txt

#Now split temp.txt into "match" sections
csplit -f sshd_config_temp.txt '/[[:space:]]*Match* /' {*}

i=$(ls sshd_config_* | wc -l); i=$((i-1))

for (( c=0; c<$i; c++ )) do echo $(printf '%02d' "$c") ; done
```

```
===== different from default =====
AuthorizedKeysFile      .ssh/authorized_keys
Subsystem sftp          /QOpenSys/QIBM/ProdData/SC1/OpenSSH/libexec/sftp-server
Match User wrother
  PasswordAuthentication yes
  X11Forwarding yes
=====
Default
-----
Match User wrother
PasswordAuthentication yes
X11Forwarding yes
```

82

cron

Package list: cronie cronie-anacron

```

SHELL=/QOpenSys/pkgs/bin/bash
PATH=/QOpenSys/pkgs/bin:/QOpenSys/usr/sbin:/QOpenSys/usr/bin
MAILTO=qsecofr

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * ibmi-user-name command to be executed
# Run a sample IBMi CL command every minute to send message to QSYSOPR
*/1 * * * * QSECOFR system "SNDMSG MSG(CRONJOB) TOUSR(QSYSOPR)"
# Run a sample IBMi CL command every minute to send message to QSYSOPR and log any stdout/stderr messages
*/1 * * * * QSECOFR system "SNDMSG MSG(CRONJOB) TOUSR(QSYSOPR)" >> /tmp/ibmicommand.log 2>&1
    
```

<https://github.com/cronie-crond/cronie>
<https://github.com/richardschoen/howtostuff/blob/master/installcronie.md>

83

crontab

```

-bash-5.2$ cat /QOpenSys/etc/crontab
SHELL=/QOpenSys/pkgs/bin/bash
PATH=/QOpenSys/pkgs/bin:/QOpenSys/usr/sbin:/QOpenSys/usr/bin
MAILTO=qsecofr

# For details see man 4 crontabs


# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name command to be executed
4 6 * * * wrother cl "SNDMSG MSG(CRONJOB) TOUSR(QSYSOPR)"
    
```

```

-bash-5.2$ cl "dspmsg qsysopr" | tail -n 10
CPC1126 50 COMPLETION Job 676962/WROTHER/CROND was ended by user WROTHER.
          QSQRVR QUSER 676651 QWTCCNJ 0000 02/28/26 04:12:40.648921 WROTHER
          Journal receivers AUDRCV0564 and *N detached.
CPF7020 00 INFO QDBSRV02 QSYS 638020 QJORETRY 0000 02/28/26 04:34:58.094243 QSYS
          80 INFO CRONJOB
          QP0ZSPWT QSECOFR_NC 676745 QCMDXC 0000 02/28/26 04:51:07.242255 WROTHER
CPC1126 50 COMPLETION Job 677034/WROTHER/CROND was ended by user WROTHER.
          QSQRVR QUSER 676907 QWTCCNJ 0000 02/28/26 05:25:09.280067 WROTHER
          80 INFO CRONJOB
          CROND WROTHER 677246 QCMDXC 0000 02/28/26 06:04:01.266580 WROTHER
    
```

<https://github.com/cronie-crond/cronie>
<https://github.com/richardschoen/howtostuff/blob/master/installcronie.md>

84

sc – Service Commander


```

-bash-5.2$ sc start crond
Performing operation 'START' on service 'crond'
Service 'Cron daemon' successfully started


-bash-5.2$ sc status group:all
  RUNNING | crond (Cron daemon)

-bash-5.2$ sc stop crond
Performing operation 'STOP' on service 'crond'
Service 'Cron daemon' successfully stopped

-bash-5.2$ sc list group:all
crond (Cron daemon)

-bash-5.2$ sc status group:all
  NOT RUNNING | crond (Cron daemon)
        
```

85

sc – Service Commander



```

-bash-5.2$ ls -la /00pensys/etc/sc/services/system/
total 292
drwxr-sr-x 2 qsys 0 28672 Feb 28 03:54 .
drwxr-sr-x 4 qsys 0 8192 Aug 16 2023 ..
-rw-r--r-- 1 qsys 0 213 Aug 16 2023 system_admin1.yaml
-rw-r--r-- 1 qsys 0 215 Aug 16 2023 system_admin2.yaml
-rw-r--r-- 1 qsys 0 229 Aug 16 2023 system_admin3.yaml
-rw-r--r-- 1 qsys 0 202 Aug 16 2023 system_admin4.yaml
-rw-r--r-- 1 qsys 0 199 Aug 16 2023 system_admin5.yaml
-rw-r--r-- 1 qsys 0 166 Aug 16 2023 system_ajs.yaml
-rw-r--r-- 1 qsys 0 178 Aug 16 2023 system_as-central.yaml
-rw-r--r-- 1 qsys 0 186 Aug 16 2023 system_as-database.yaml
-rw-r--r-- 1 qsys 0 169 Aug 16 2023 system_as-dtaq.yaml
-rw-r--r-- 1 qsys 0 186 Aug 16 2023 system_as-file.yaml
-rw-r--r-- 1 qsys 0 189 Aug 16 2023 system_as-netprt.yaml
-rw-r--r-- 1 qsys 0 174 Aug 16 2023 system_as-rnrcmd.yaml
-rw-r--r-- 1 qsys 0 174 Aug 16 2023 system_as-signon.yaml
-rw-r--r-- 1 qsys 0 104 Aug 16 2023 system_as-sts.yaml
-rw-r--r-- 1 qsys 0 168 Aug 16 2023 system_as-svrmapi.yaml
-rw-r--r-- 1 qsys 0 186 Aug 16 2023 system_c_inom.yaml
-rw-r--r-- 1 qsys 0 135 Aug 16 2023 system_dbgsvr.yaml
-rw-r--r-- 1 qsys 0 176 Aug 16 2023 system_ddm.yaml
-rw-r--r-- 1 qsys 0 183 Aug 16 2023 system_directoryserver.yaml
-rw-r--r-- 1 qsys 0 149 Aug 16 2023 system_ftp.yaml
-rw-r--r-- 1 qsys 0 208 Aug 16 2023 system_http_admin.yaml
-rw-r--r-- 1 qsys 0 182 Aug 16 2023 system_lpd.yaml
-rw-r--r-- 1 qsys 0 179 Aug 16 2023 system_netserver.yaml
-rw-r--r-- 1 qsys 0 167 Aug 16 2023 system_pop.yaml
-rw-r--r-- 1 qsys 0 316 Aug 16 2023 system_rmc.yaml
-rw-r--r-- 1 qsys 0 172 Aug 16 2023 system_slp.yaml
-rw-r--r-- 1 qsys 0 168 Aug 16 2023 system_smtp.yaml
-rw-r--r-- 1 qsys 0 195 Aug 16 2023 system_sptprx.yaml
-rw-r--r-- 1 qsys 0 176 Aug 16 2023 system_sshd.yaml
-rw-r--r-- 1 qsys 0 178 Aug 16 2023 system_telnet.yaml
-rw-r--r-- 1 qsys 0 186 Aug 16 2023 system_webquery.yaml
-rw-r--r-- 1 qsys 0 178 Aug 16 2023 system_ypsjsvr.yaml

-bash-5.2$ ls -la /00pensys/etc/sc/services/oss_common/
total 48
drwxr-sr-x 2 qsys 0 8192 Feb 28 03:54 .
drwxr-sr-x 4 qsys 0 8192 Aug 16 2023 ..
-rw-r--r-- 1 qsys 0 219 Aug 16 2023 crond.yaml
-rw-r--r-- 1 qsys 0 172 Aug 16 2023 mariadb.yaml
-rw-r--r-- 1 qsys 0 755 Aug 16 2023 postgres.yaml
-rw-r--r-- 1 qsys 0 186 Aug 16 2023 zenddbt.yaml

-bash-5.2$ sc status system_admin1
  RUNNING | system_admin1 (Navigator for i (latest generation))
        
```

86


sc - groups


```

-bash-5.2$ sc groups
autostart
system

-bash-5.2$ sc status group:system
RUNNING system_admin1 (Navigator for i (latest generation))
PARTIAL (3/6) system_admin2 (IBM Navigator for i) [not running at →PORT:2005, PORT:16324, PORT:5471]
RUNNING system_admin3 (management (Digital Certificate Manager, Db2 Mirror))
RUNNING system_admin4 (HTTP Administration for i)
RUNNING system_admin5 (HTTP Administration UI)
RUNNING system_ajs (Advanced Job Scheduler)
RUNNING system_as-central (System *CENTRAL Host Server)
RUNNING system_as-database (System *DATABASE Host Server)
RUNNING system_as-dtaq (System *DTAQ Host Server)
RUNNING system_as-file (System *FILE Host Server)
RUNNING system_as-netprt (System *NETPRT Host Server)
RUNNING system_as-rmtcmd (System *RMTCMD Host Server)
RUNNING system_as-signon (System *SIGNON Host Server)
RUNNING system_as-sts (Service Tools Server)
RUNNING system_as-svrmap (System *SVRMAP Host Server)
NOT RUNNING system_c_inom (System CIM Server)
NOT RUNNING system_dbgsvr (System Debug Server)
RUNNING system_ddm (System DDM server)
PARTIAL (1/2) system_directoryserver (System Directory Server) [not running at →PORT:3539]
RUNNING system_ftp (System FTP server)
RUNNING system_http_admin (HTTP Admin Server)
NOT RUNNING system_lpd (System Line Printer Daemon Server)
RUNNING system_netserver (System NetServer)
NOT RUNNING system_pop (System POP3 server)
RUNNING system_rmc (Resource Monitoring and Control [RMC])
RUNNING system_slp (System SLP server)
RUNNING system_smtp (System SMTP server)
NOT RUNNING system_sptprx (IBM Support Communication Service)
RUNNING system_sshd (System Secure Shell server)
RUNNING system_telnet (System Telnet server)
NOT RUNNING system_webquery (Db2 WebQuery)
NOT RUNNING system_ypsjsvr (IBM Management Central Server)
    
```

87


scopenports


```

-bash-5.2$ scopenports
Address Port 'sc' service name (and friendly name)
-----
0.0.0.0 21 system_ftp (System FTP server)
:: 21 system_ftp (System FTP server)
0.0.0.0 22 system_sshd (System Secure Shell server)
:: 22 system_sshd (System Secure Shell server)
0.0.0.0 23 system_telnet (System Telnet server)
:: 23 system_telnet (System Telnet server)
0.0.0.0 25 system_smtp (System SMTP server)
:: 25 system_smtp (System SMTP server)
0.0.0.0 137 system_netserver (System NetServer)
0.0.0.0 139 system_netserver (System NetServer)
0.0.0.0 389 system_directoryserver (System Directory Server)
:: 389 system_directoryserver (System Directory Server)
127.0.0.1 427 system_slp (System SLP server)
:: 427 system_slp (System SLP server)
0.0.0.0 445 system_netserver (System NetServer)
:: 445 system_netserver (System NetServer)
0.0.0.0 446 system_ddm (System DDM server)
:: 446 system_ddm (System DDM server)
0.0.0.0 447 system_ddm (System DDM server)
:: 447 system_ddm (System DDM server)
0.0.0.0 448 system_ddm (System DDM server)
:: 448 system_ddm (System DDM server)
0.0.0.0 449 system_as-svrmap (System *SVRMAP Host Server)
:: 449 system_as-svrmap (System *SVRMAP Host Server)
0.0.0.0 512 port:512
:: 512 port:512
0.0.0.0 636 port:636
:: 636 port:636
0.0.0.0 657 system_rmc (Resource Monitoring and Control [RMC])
:: 657 system_rmc (Resource Monitoring and Control [RMC])
0.0.0.0 992 system_telnet (System Telnet server)
:: 992 system_telnet (System Telnet server)
0.0.0.0 2001 system_http_admin (HTTP Admin Server)
:: 2001 system_http_admin (HTTP Admin Server)
0.0.0.0 2002 system_admin1 (Navigator for i (latest generation))
0.0.0.0 2003 system_admin1 (Navigator for i (latest generation))
0.0.0.0 2004 system_admin2 (IBM Navigator for i)
0.0.0.0 2006 system_admin3 (management (Digital Certificate Manager, Db2 Mirror))
0.0.0.0 2007 port:2007
0.0.0.0 2008 system_admin4 (HTTP Administration for i)
0.0.0.0 2010 system_http_admin (HTTP Admin Server)
:: 2010 system_http_admin (HTTP Admin Server)
0.0.0.0 2011 system_admin5 (HTTP Administration UI)
0.0.0.0 2012 port:2012
127.0.0.1 2015 port:2015
127.0.0.1 2016 system_admin2 (IBM Navigator for i)
    
```

88

IBM i Automation with Bash Programming




IBM i Command Execution

© Copyright IBM Corporation 2001, 2016, 2020, 2023
 Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

89

Es muss nicht immer CL sein



```

-bash-5.1$ system dspLibl
5770SS1 V7RAM0 196621
                                Library List
Library  Type  ASP  Text Description
QSYS     SYS   Device  System Library
QSYS2    SYS           System Library for CPI's
QHLP5YS  SYS           System Library for Users
QUSRSYS  SYS           System Library for Users
OSHELL   PRD           General Purpose Library
UNIMASTER CUR
QGPL     USR
QTEMP    USR
***** END OF LISTING *****
                    
```

Auslesen von Informationen aus SBSDB →

Erstellen von Testusern ↓

```

#!/bin/bash
export UNI=IWS
export YEAR=21
mkdir lab5
for value in {000..010}
do
    echo -----
    system -v "dspbsdb iwsrks$value/iwssbs$value" > lab5/SBS_$(SUNI$YEAR$value).txt
    grep JOB0 lab5/SBS_$(SUNI$YEAR$value).txt
    grep "Work station name" lab5/SBS_$(SUNI$YEAR$value).txt
    grep "Sign-on display file" lab5/SBS_$(SUNI$YEAR$value).txt
    echo -----
done
echo All done
                    
```

```

#!/bin/bash
export UNIGRP=IWS21
for value in {000..010}
do
    echo Create $(SUNI$GRP$value)
    system -v "unimaster/CRTSTUUSR USRPRF($(SUNI$GRP$value) PASSWORD(abcd1234) FNAME(as$value) EMAIL(b) STUNBR($value) UNIACCT('Intensiv Workshop') UNIID(IWS)"
done
echo All done
                    
```

IBM i Automation with Bash Programming

90

90

Neu: getjobid, cl, and liblist



Unix process model causing BASH to start a new process in which the command starts up

```
$ /QOpenSys/usr/bin/getjobid
Process identifier 1575015 is 073904/QSECOFR/QP0ZSPWT
$ /QOpenSys/usr/bin/getjobid
Process identifier 1575016 is 073985/QSECOFR/QP0ZSPWT
$ /QOpenSys/usr/bin/getjobid
Process identifier 1575017 is 073986/QSECOFR/QP0ZSPWT
```

New functions are builtins and not commands, they execute in BASH itself, ie. the current job

```
$ getjobid
Process identifier 1574877 is 073846/QSECOFR/QP0ZSPWT
$ getjobid
Process identifier 1574877 is 073846/QSECOFR/QP0ZSPWT
$ getjobid $$
Process identifier 1574877 is 073846/QSECOFR/QP0ZSPWT
```

```
$ liblist -a kadler
CPC2196: Library KADLER added to library list.
$ liblist -d kadler
CPC2197: Library KADLER removed from library list.
$ liblist -c kadler
CPC2198: Current library changed to KADLER.
$ liblist
QSYS      SYS
QSYS2     SYS
QHLPSSYS  SYS
QUSRSYS   SYS
QSHELL    PRD
KADLER    CUR
QGPL      USR
QTEMP     USR
QDEVELOP  USR
QBLDSYS   USR
```

In addition, the **cl** function behaves almost the same as system, except it defaults to running in the current job, so no **-i** is needed.

<https://kadler.io/2020/12/23/bash-builtins.html#>

91

Example - nslookup



cl_nslookup:

```
#!/bin/bash
cl "NSLOOKUP HOSTNAME($1)"
```

nslookup:

```
#!/bin/bash
server=$1
db2util -o json "SELECT * FROM TABLE(QSYS2.DNS_LOOKUP('$server'))" | jq
.records[.].IP_ADDRESS | sed "s/\\//g"

db2util -o json "SELECT * FROM TABLE(QSYS2.DNS_LOOKUP('$server'))" | jq .records[.]
```

92

Ordering PTFs via IBM Fix Central

B7135394 Update Received, Customer#: FCPROD1
ccssbldr to: wolfgang.rother

ccssbldr B7135394 Update Received, Customer#: FCPROD1

additional correspondence related to this transaction.

 Note: The fix package will be deleted in 72 hours.

INFORMATION YOU WILL NEED TO RETRIEVE YOUR ORDER

Your UserID	--> B433T9
FTP Server	--> delivery01-bld.dhe.ibm.com
Transfer type	--> ascii/binary
Directory on server	--> 24449480/C
Files to get	--> ftpSF99737.txt
	--> 11stSF99737.txt
	--> sha256.txt
	--> SF99737_1.bin
	--> SF99737_2.bin
	--> SF99737_3.bin
	--> SF99737_4.bin
	--> SF99737_5.bin

B7135394 Update Received, Customer#: FC
ccssbldr to: wolfgang.rother

ccssbldr B7135394 Update Received, Customer#:

Your password: 068HoH5M3b2G958
 Server: delivery01-bld.dhe.ibm.com

Note: For security reasons, your userid was sent in email. Please, check your inbox or contact CCSS Sup you didn't receive it.

93

PTF Download with curl

User: B433T9
Paßwort: 068HoH5M3b2G958

Server: delivery01-bld.dhe.ibm.com
Directory: 24449480/C
File: SF99737_1.bin

```
curl -sS -u B433T9:068HoH5M3b2G958 --insecure
sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_1.bin -o SF99737_1.bin &
```

-u Benutzer:Paßwort
 -o Outputfile
 -sS silent

--insecure nur falls notwendig, z.B. die Zertifikate „selbst-signiert“ sind


Paralleler Download:

```
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_1.bin -o SF99737_1.bin &
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_2.bin -o SF99737_2.bin &
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_3.bin -o SF99737_3.bin &
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_4.bin -o SF99737_4.bin &
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_5.bin -o SF99737_5.bin &
curl -sS -u B433T9:068HoH5M3b2G958 --insecure sftp://delivery01-bld.dhe.ibm.com/24449480/C/SF99737_6.bin -o SF99737_6.bin &
```

IBM i Automation with Bash Programming
94

94

Group PTF Download – Bash script




```
-bash-5.2$ cat PTFdownload.sh
#!/bin/bash
USER="DX64T7"
PWD="HNo5G498qLb263I"
DIR="myptf/C"
SERVER="delivery01-mul.dhe.ibm.com"
PREFIX="grpprf"
n=1
i=1
while [ $i -le $n ]
do
    echo "curl -sS -u $USER:$PWD --insecure sftp://$SERVER/$DIR/${PREFIX}_$i.bin -o ${PREFIX}_$i.bin > t$i.txt &"
    curl -sS -u $USER:$PWD --insecure sftp://$SERVER/$DIR/${PREFIX}_$i.bin -o ${PREFIX}_$i.bin > t$i.txt &
    i=$(( $i+1 ))
done
```

IBM i Automation with Bash Programming 95

95


IBM i Automation with Bash Programming



IBM i (SQL) Services

© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

96


IBM i Services


IBM i Service	Type of Service	IBM i 7.6	IBM i 7.5	IBM i 7.4	IBM i 7.3
Application Services					
QSYS2.ACTIVATION_GROUP_INFO()	Table function	Base	Base	SF99704 Level 20	SF99703 Level 28
QSYS2.ADD_USER_INDEX_ENTRY()	Procedure	Base	Base	SF99704 Level 20	Not Supported
QSYS2.ADD_USER_INDEX_ENTRY_BINARY()	Procedure	Base	Base	SF99704 Level 20	Not Supported
QSYS2.BINDING_DIRECTORY_INFO	View	Base	Base	SF99704 Level 20	SF99703 Level 28
QSYS2.BOUND_MODULE_INFO	View	Base	Base	SF99704 Level 4	SF99703 Level 16

<https://www.ibm.com/support/pages/ibm-i-services-sql>

IBM i Automation with Bash Programming
97

97

Examples


```

SNR=$(db2util -o json "select * from qsys2.system_status_info" | jq -r .records[].MACHINE_SERIAL_NUMBER)
echo $SNR
7857E21

db2util -o json "select * from QSYS2.NETSTAT_INTERFACE_INFO" | jq .records[]

db2util -o json "SELECT * FROM TABLE(SYSTOOLS.PING('isec1'))" | jq .records[].RESULT

db2util -o json "SELECT * FROM TABLE(SYSTOOLS.PING(REMOTE_IP_ADDRESS => '172.29.65.180'));" | jq .records[].RESULT

db2util -o json "SELECT * FROM TABLE(SYSTOOLS.PING(REMOTE_IP_ADDRESS => '172.29.65.180'));" | jq ".records | map({RESULT , REMOTE_IP_ADDRESS})"
[
  {
    "RESULT": "SUCCESS",
    "REMOTE_IP_ADDRESS": "172.29.65.180"
  }
]
```

98

98

Example: Queries on other systems



```
db2util -o json "select * from ISEC1.QSYS2.NETSTAT_INTERFACE_INFO" | jq .records[]
```

Remote Systeme müssen zuvor mit ADDRDBDIR oder WRKRDBDIRE dem Relational Database Directory der IBM i LPAR hinzugefügt werden:

```
Work with Relational Database Directory Entries

Position to . . . . .

Type options, press Enter.
  1=Add  2=Change  4=Remove  5=Display details  6=Print details

Option  Entry                Remote Location      Text
-----  -----
  1      I_AXEL                    172.29.65.20
  2      IAXEL                     *LOCAL
  4      ISEC1                     172.29.65.180
```

99

Solution



- Ping all IP4-Adresses of a system

```
#!/QOpenSys/pkg/bin/bash

remote_system=""
length=$(echo $1 | wc -m )

if [ $length -gt 1 ]
then remote_system=$1."
fi

IPs=$(db2util -o json "SELECT INTERNET_ADDRESS FROM "$remote_system"QSYS2.NETSTAT_INTERFACE_INFO where CONNECTION_TYPE='IPV4'" | jq -r ".records[]|.INTERNET_ADDRESS")

printf "\n----- ping mit SQL service ----- \n\n"


for IP in $IPs
do
result=$(db2util -o json "SELECT * FROM TABLE(SYSTOOLS.PING(REMOTE_IP_ADDRESS => '$IP'))" | jq -r .records[]|.RESULT )
printf "%-15s : %s\n" "$IP" "$result"
done

printf "\n----- ping mit CL command ----- \n\n"

for IP in $IPs
do
result=$(cl "ping '$IP'" | grep TCP3210 | sed "s/TCP3210: //g" )
printf "%-15s : %s\n" "$IP" "$result"
done
```

100

Oder PINGing with SSSYTOOLS




```

select *
  from table (
    SSSYTOOLS.PING(REMOTE_SYSTEM => 'IBM.COM', /* VARCHAR(255) Default: *INTERNETADR */
      REMOTE_IP_ADDRESS => NULL, /* VARCHAR(45) - Default: NULL */
      ADDRESS_VERSION_FORMAT => *CALC', /* VARCHAR(5) - *CALC, *IP4, *IP6 - Default: *CALC */
    )
  )
);

• select *
  from table (
    SSSYTOOLS.PING(REMOTE_SYSTEM => *INTERNETADR', /* VARCHAR(255) Default: *INTERNETADR */
      REMOTE_IP_ADDRESS => '192.168.1.2', /* VARCHAR(45) - Default: NULL */
      ADDRESS_VERSION_FORMAT => *IP4', /* VARCHAR(5) - *CALC, *IP4, *IP6 - Default: *CALC */
      NUMBER_PACKETS_TO_SEND => 5, /* INTEGER - Default: 5 */
      PACKET_LENGTH_TO_SEND => 256, /* INTEGER - Default: 256 */
      WAIT_TIME => 1, /* INTEGER - Default: 1 */
      LOCAL_IP_ADDRESS => *ANY' /* VARCHAR(45) - *ANY - Default: *ANY */
    )
  );
    
```

101

IBM i Services



<https://www.ibm.com/docs/en/i/7.6.0?topic=is-application-services>

<https://www.ibm.com/docs/en/i/7.6.0?topic=services-active-job-info-table-function>

Example

– **Example 1:** Looking at only QZDASOINIT jobs, find the top 10 consumers of Elapsed I/O.

```

SELECT JOB_NAME, AUTHORIZATION_NAME, ELAPSED_TOTAL_DISK_IO_COUNT, ELAPSED_CPU_PERCENTAGE
FROM TABLE(QSYS2.ACTIVE_JOB_INFO(
  JOB_NAME_FILTER => 'QZDASOINIT',
  SUBSYSTEM_LIST_FILTER => 'QUSRWRK')) X
ORDER BY ELAPSED_TOTAL_DISK_IO_COUNT DESC
FETCH FIRST 10 ROWS ONLY;
    
```

Note: The data in the ELAPSED_xxx columns is updated upon each re-execution of the query. Elapsed data will not get returned the first time a query is run for ACTIVE_JOB_INFO for a connection. See the *reset-statistics* parameter for details.


– **Example 2:** Find the active jobs using the most temporary storage. Include the most recently executed SQL statement for each target job.

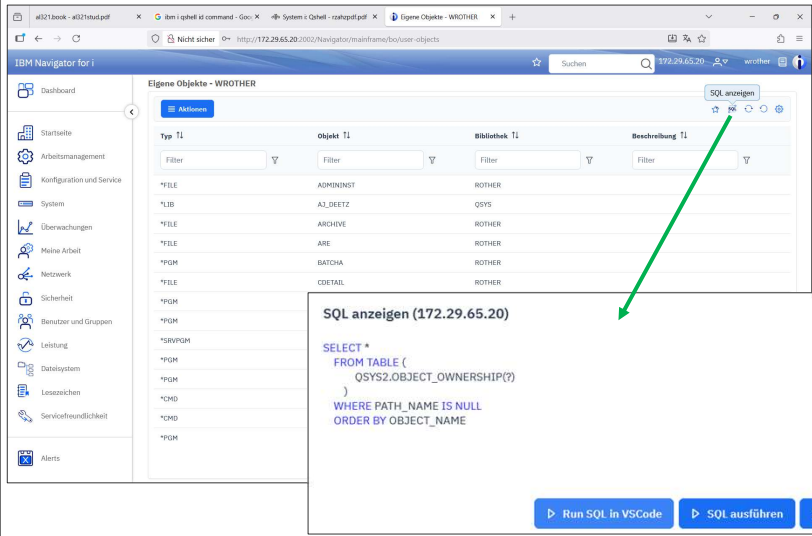
```

SELECT JOB_NAME, AUTHORIZATION_NAME, TEMPORARY_STORAGE, SQL_STATEMENT_TEXT
FROM TABLE (QSYS2.ACTIVE_JOB_INFO(DETAILED_INFO=>'ALL')) X
WHERE JOB_TYPE <> 'SYS'
ORDER BY TEMPORARY_STORAGE DESC;
    
```

102

Finding an appropriate IBM i Service






The screenshot shows the IBM Navigator for i interface. A table lists objects with columns for Type, Object, Library, and Description. A modal window titled "SQL anzeigen (172.29.65.20)" is open, displaying a SQL query: `SELECT * FROM TABLE (QSYS2.OBJECT_OWNERSHIP(?)) WHERE PATH_NAME IS NULL ORDER BY OBJECT_NAME`. The modal includes buttons for "Run SQL in VSCode", "SQL ausführen", and "Schließen".

IBM i Automation with Bash Programming 103

103

Using QShell db2 utility in bash



CCSID in User Profile must be different from 65535 (for example 37)

```
# SQL script test.sql
SELECT * FROM TABLE(QSYS2.OBJECT_STATISTICS('ROTHER','*FILE'))
```

```
qsh -c 'db2 -t -f /home/wrother/test.sql' | sed 1d | sed 2d | sed '$d' | sed '$d' > temp.txt
```

```
# conv2csv.sh temp.txt
if [ -f temp.csv ]; then rm temp.csv ; fi
while IFS= read -r line; do
  echo $line | sed 's/ /,/g' | sed 's/$/r/g' >> temp.csv
done < $1
```


```
cat temp.csv | python -c 'import csv, json, sys; print(json.dumps([dict(r) for r in csv.DictReader(sys.stdin)]))' | jq .
```

All together "in one line":

```
qsh -c 'db2 -t -f /home/wrother/test.sql' | sed 1d | sed 2d | sed '$d' | sed '$d' > temp.txt ;
conv2csv.sh temp.txt ; cat temp.csv | python -c 'import csv, json, sys;
print(json.dumps([dict(r) for r in csv.DictReader(sys.stdin)]))' | jq . > result.json ; rm temp.*
```

IBM i Automation with Bash Programming 104

104



Creating the List of Column Names with db2

```

-bash-5.2$ COL=$(qsh -c "db2 \select * from Table(QSYS2.OBJECT_STATISTICS('ROTHER','FILE')) fetch first row only\" | head -n 2 | sed '1d')
-bash-5.2$ echo $COL
OBJNAME OBJTYPE OBJOWNER OBJDEFINER OBJCREATED OBJSIZE OBJTEXT OBJLONGNAME LAST_USED_TIMESTAMP LAST_USED_OBJECT DAYS_USED_COUNT LAST_RESET_TIMESTAMP IASP_NUMBER
IASP_NAME OBJATTRIBUTE OBJLONGSCHEMA TEXT SQL_OBJECT_TYPE OBJLIB CHANGE_TIMESTAMP USER_CHANGED_SOURCE_FILE SOURCE_LIBRARY SOURCE_MEMBER SOURCE_TIMESTAMP CREATE
D_SYSTEM CREATED_SYSTEM VERSION LICENSED_PROGRAM LICENSED_PROGRAM_VERSION COMPILER_COMPILER_VERSION OBJECT_CONTROL_LEVEL BUILD_ID PTF_NUMBER APAR_ID USER_DEFINED
ATTRIBUTE ALLOW_CHANGE_BY_PROGRAM CHANGED_BY_PROGRAM COMPRESSED_PRIMARY_GROUP STORAGE_FREED_ASSOCIATED_SPACE_SIZE OPTIMUM_SPACE_ALIGNMENT OVERFLOW_STORAGE_OBJ
ECT_DOMAIN OBJECT_AUDIT OBJECT_SIGNED_SYSTEM_TRUSTED_SOURCE_MULTIPLE_SIGNATURES SAVE_TIMESTAMP RESTORE_TIMESTAMP SAVE_WHILE_ACTIVE_TIMESTAMP SAVE_COMMAND_SAVE_D
EVICE_SAVE_FILE_NAME SAVE_FILE_LIBRARY SAVE_VOLUME_SAVE_LABEL_SAVE_SEQUENCE_NUMBER LAST_SAVE_SIZE JOURNALED_JOURNAL_NAME JOURNAL_LIBRARY JOURNAL_IMAGES OMIT_JOU
RNAL_ENTRY REMOTE_JOURNAL_FILTER JOURNAL_START_TIMESTAMP APPLY_STARTING_RECEIVER APPLY_STARTING_RECEIVER_LIBRARY AUTHORITY_COLLECTION_VALUE
-bash-5.2$
    
```


Preparation and execution

```

-bash-5.2$ COL=$(echo $(qsh -c "db2 \select * from Table(QSYS2.OBJECT_STATISTICS('ROTHER','FILE')) fetch first row only\" | head -n 2 | sed '1d') | sed 's
/ /, /g' | sed -s/BUILD_ID/hex(BUILD_ID) as BUILD_ID/g')
-bash-5.2$ echo $COL
OBJNAME OBJTYPE OBJOWNER OBJDEFINER OBJCREATED OBJSIZE OBJTEXT OBJLONGNAME LAST_USED_TIMESTAMP LAST_USED_OBJECT DAYS_USED_COUNT LAST_RESET_TIMESTAMP
IASP_NUMBER IASP_NAME OBJATTRIBUTE OBJLONGSCHEMA TEXT SQL_OBJECT_TYPE OBJLIB CHANGE_TIMESTAMP USER_CHANGED_SOURCE_FILE SOURCE_LIBRARY SOURCE_MEMBER SOURCE_TIMESTAMP
SOURCE_TIMESTAMP_CREATED_SYSTEM_CREATED_SYSTEM_VERSION_LICENSED_PROGRAM_LICENSED_PROGRAM_VERSION_COMPILER_COMPILER_VERSION_OBJECT_CONTROL_LEVEL_hex(BUI
LD_ID) as BUILD_ID_PTF_NUMBER APAR_ID_USER_DEFINED_ATTRIBUTE_ALLOW_CHANGE_BY_PROGRAM_CHANGED_BY_PROGRAM_COMPRESSED_PRIMARY_GROUP_STORAGE_FREED_ASSOCIAT
ED_SPACE_SIZE_OPTIMUM_SPACE_ALIGNMENT_OVERFLOW_STORAGE_OBJECT_DOMAIN_OBJECT_AUDIT_OBJECT_SIGNED_SYSTEM_TRUSTED_SOURCE_MULTIPLE_SIGNATURES_SAVE_TIMESTAMP
RESTORE_TIMESTAMP_SAVE_WHILE_ACTIVE_TIMESTAMP_SAVE_COMMAND_SAVE_DEVICE_SAVE_FILE_NAME_SAVE_FILE_LIBRARY_SAVE_VOLUME_SAVE_LABEL_SAVE_SEQUENCE_NUMBER_L
AST_SAVE_SIZE_JOURNALED_JOURNAL_NAME_JOURNAL_LIBRARY_JOURNAL_IMAGES_OMIT_JOURNAL_ENTRY_REMOTE_JOURNAL_FILTER_JOURNAL_START_TIMESTAMP_APPLY_STARTING_RECE
IVER_APPLY_STARTING_RECEIVER_LIBRARY_AUTHORITY_COLLECTION_VALUE
-bash-5.2$ db2util -o json "select $COL from Table(QSYS2.OBJECT_STATISTICS('ROTHER','FILE')) fetch first row only"
{"records": [
  {
    "OBJNAME": "EVEVENT", "OBJTYPE": "FILE", "OBJOWNER": "WROTHER", "OBJDEFINER": "WROTHER", "OBJCREATED": "2022-09-15-12.18.00.000000", "OBJSIZE": "155648", "OBJTEXT": "null",
    "OBJLONGNAME": "EVEVENT", "LAST_USED_TIMESTAMP": "null", "LAST_USED_OBJECT": "YES", "DAYS_USED_COUNT": "0", "LAST_RESET_TIMESTAMP": "null", "IASP_NUMBER": "0", "IASP_NAME": "S
YSBAS", "OBJATTRIBUTE": "PF", "OBJLONGSCHEMA": "ROTHER", "TEXT": "null", "SQL_OBJECT_TYPE": "null", "OBJLIB": "ROTHER", "CHANGE_TIMESTAMP": "2025-11-26-14.41.24.000000", "US
ER_CHANGED": "YES", "SOURCE_FILE": "null", "SOURCE_LIBRARY": "null", "SOURCE_MEMBER": "null", "SOURCE_TIMESTAMP": "null", "CREATED_SYSTEM": "TDI001M", "CREATED_SYSTEM_VERS
ION": "V7R5M0", "LICENSED_PROGRAM": "null", "LICENSED_PROGRAM_VERSION": "null", "COMPILER": "null", "COMPILER_VERSION": "null", "OBJECT_CONTROL_LEVEL": "null", "BUILD_ID": "
null", "PTF_NUMBER": "null", "APAR_ID": "null", "USER_DEFINED_ATTRIBUTE": "null", "ALLOW_CHANGE_BY_PROGRAM": "YES", "CHANGED_BY_PROGRAM": "NO", "COMPRESSED": "null", "PRIMAR
Y_GROUP": "null", "STORAGE_FREED": "NO", "ASSOCIATED_SPACE_SIZE": "4096", "OPTIMUM_SPACE_ALIGNMENT": "YES", "OVERFLOW_STORAGE": "NO", "OBJECT_DOMAIN": "SYSTEM", "OBJECT_AUDIT
": "null", "OBJECT_SIGNED": "NO", "SYSTEM_TRUSTED_SOURCE": "NO", "MULTIPLE_SIGNATURES": "NO", "SAVE_TIMESTAMP": "2025-11-26-15.39.09.000000", "RESTORE_TIMESTAMP": "202
5-11-26-14.39.09.000000", "SAVE_WHILE_ACTIVE_TIMESTAMP": "null", "SAVE_COMMAND": "SAVLB", "SAVE_DEVICE": "SAV", "SAVE_FILE_NAME": "QSVRSF04F9", "SAVE_FILE_LIBRARY": "Q
TEMP", "SAVE_VOLUME": "null", "SAVE_LABEL": "null", "SAVE_SEQUENCE_NUMBER": "null", "LAST_SAVE_SIZE": "147456", "JOURNALED": "NO", "JOURNAL_NAME": "OBJJRN", "JOURNAL_LIBRARY": "Q
JRNLIB", "JOURNAL_IMAGES": "AFTER", "OMIT_JOURNAL_ENTRY": "MPCNCLD", "REMOTE_JOURNAL_FILTER": "NO", "JOURNAL_START_TIMESTAMP": "2022-09-15-12.18.00.000000", "APPLY_STAR
TING_RECEIVER": "QBRJCV0952", "APPLY_STARTING_RECEIVER_LIBRARY": "QJRNLIB", "AUTHORITY_COLLECTION_VALUE": "NONE"
  ]
}
    
```

IBM i Automation with Bash Programming

107



Using QShell db2 utility in bash (alternativ)

CCSID in User Profile must be different from 65535 (for example 37)

```


# SQL script json_test.sql
select json_object(
  key 'EMPNO' value empno, key 'FIRSTNME' value firstnme,
  key 'MIDINIT' value midinit, key 'LASTNAME' value lastname,
  key 'WORKDEPT' value workdept, key 'PHONENO' value phoneno,
  key 'HIREDATE' value hiredate, key 'JOB' value job,
  key 'EDLEVEL' value edlevel, key 'SEX' value sex,
  key 'BIRTHDATE' value birthdate, key 'SALARY' value salary,
  key 'BONUS' value bonus, key 'COMM' value comm
  absent on null returning clob(2g) ccsid 1208 format json
)
from testdb.employee;
    
```

```

qsh -c 'db2 -t -f /home/wrotther/json_test.sql' | sed '1d' | sed '2d' | sed '$d' | sed '$d' | jq .
    
```

```

-bash-5.2$ qsh -c "db2 -t -f /home/wrotther/json_test.sql" | sed '1d' | sed '2d' | sed '$d' | sed '$d' | jq .
{
  "EMPNO": "000010",
  "FIRSTNME": "CHRISTINE",
  "MIDINIT": "F",
  "LASTNAME": "HAAS",
  "WORKDEPT": "AB0",
  "PHONENO": "3978",
  "HIREDATE": "1965-01-01",
  "JOB": "PRES",
  "EDLEVEL": "18",
  "SEX": "F",
  "BIRTHDATE": "1933-08-24",
  "SALARY": "52750",
  "BONUS": "1800",
  "COMM": "4220"
}
    
```



108

Installing License Keys.



1. Download the Word document from ESS
2. Convert to txt file (LF only)
3. Transfer to the system

```
cat IBMi_keys.txt | (while read foo; do echo "$foo"; done)
```

IBM i Automation with Bash Programming

109

Deleting Joblogs within a FOR loop



```
db2util -o csv "SELECT SPOOLED_FILE_NAME, JOB_NAME FROM
QSYS2.OUTPUT_QUEUE_ENTRIES_BASIC where
((OUTPUT_QUEUE_NAME='QEZJOBLOG') AND (CREATE_TIMESTAMP > '2025-12-05
10:00:00.000000') AND (CREATE_TIMESTAMP < '2025-12-05 18:00:00.000000'))"
> Liste.txt
```

```
JOBNAME=$(cut -f2 -d, Liste.txt | sed 's/"/\&quot;')
for line in $JOBNAME;
do
    echo $line
    echo cl "dltspfl file(QPJOBLOG) JOB($line) SPLNBR(*any) SELECT(*ALL)"
    cl "dltspfl file(QPJOBLOG) JOB($line) SPLNBR(*ANY) SELECT(*ALL)"
    echo -----
done
```

IBM i Automation with Bash Programming

110

What's wrong?

```

-bash-5.1$ cat dltsplf_while_do.sh
#!/usr/bin/bash

echo =====

db2util -o csv "SELECT SPOOLED FILE NAME, JOB NAME FROM QSYS2.OUTPUT QUEUE ENTRIES BASIC where (
(OUTPUT_QUEUE_NAME='QEZJOBLOG') AND (CREATE_TIMESTAMP > '2025-12-08 10:00:00.000000') AND (CREATE
_TIMESTAMP < '2025-12-10 18:00:00.000000')) fetch first 5 rows only" > Liste.txt

echo =====

while read -r zeile; do

    SPLFNAME=$(echo $zeile | cut -f1 -d, | sed 's/"//g')
    JOBNAME=$(echo $zeile | cut -f2 -d, | sed 's/"//g')
    echo "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"
    cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)"

done < Liste.txt

echo Fertig!
-bash-5.1$ dltsplf_while_do.sh

=====

dltsplf file(QPJOBLOG) JOB(456895/IWS25004/BATCH1) SPLNBR(*any) SELECT(*ALL)
CPD000D: Command *LIBL/DLTSPLF not safe for a multithreaded job.
Fertig!
-bash-5.1$
    
```

IBM i Automation with Bash Programming

111

```

1  dltsplf_while_do.sh
2  #!/usr/bin/bash
3  # set -x
4
5  # clear
6
7  echo =====
8
9  db2util -o csv "SELECT SPOOLED FILE NAME, JOB NAME FROM QSYS2.OUTPUT QUEUE ENTRIES
10 =====
11
12 while read -r zeile; do
13
14     echo $zeile
15     SPLFNAME=$(echo $zeile | cut -f1 -d, | sed 's/"//g')
16     JOBNAME=$(echo $zeile | cut -f2 -d, | sed 's/"//g')
17     echo "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"
18     cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)"
19
20 done < Liste.txt
21
22 echo Fertig!
    
```

Hi Bob, why does the while loop break after the first line with the `cl` command in line 18?

It help you understand why the while loop breaks after the first line with the `cl` command, let me first read the file to see the code.

I want to read this file:

```
dltsplf_while_do.sh
```

Now I can see the issue: the while loop breaks after the first line because the `cl` command on line 18 is consuming the remaining input from `Liste.txt`.

The Problem

When you execute `cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"` inside the while loop, the `cl` command line starts to reading from `stdin` instead of `read`. Since your while loop is also reading from `stdin` via `cl`, the `cl` command consumes all the remaining lines from `Liste.txt`, leaving nothing for the next iteration of the while loop.

The Solution

You need to redirect the `cl` command to read from a different file descriptor or `/dev/null` to prevent it from consuming your input file. Here are two solutions:

Solution 1: Redirect stdin for the `cl` command

```
cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)" < /dev/null
```

Solution 2: Use a different file descriptor for the while loop

```
while read -r zeile <2; do
    SPLFNAME=$(echo $zeile | cut -f1 -d, | sed 's/"//g')
    JOBNAME=$(echo $zeile | cut -f2 -d, | sed 's/"//g')
    echo "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"
    cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)"
done < 2 Liste.txt
```

Recommended Fix

Auto approval disabled


What's on your mind?

Ⓜ to add content, / for commands, hold shift to drag in files/images

7 Ask ...

IBM i Automation with Bash Programming

112

BOB's Solutions


Recommended Fix

The simplest fix is **Solution 1** - modify line 18 to:

```
c1 "dltspif file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)" < /dev/null
```

This ensures that the `c1` command reads from `/dev/null` instead of consuming the input from `Liste.txt`, allowing your while loop to continue processing all lines in the file.

Task Completed

The while loop breaks after the first line because the `c1` command on line 18 is consuming the remaining input from `Liste.txt`.

Root Cause:
When `c1 "dltspif file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)"` executes inside the while loop, it reads from `stdin`. Since the while loop is also reading from `stdin` (via `< Liste.txt`), the `c1` command consumes all remaining lines from the file, leaving nothing for subsequent loop iterations.

Solution:
Redirect the `c1` command to read from `/dev/null` instead:

```
c1 "dltspif file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)" < /dev/null
```

This prevents `c1` from consuming the input file, allowing the while loop to process all lines in `Liste.txt`.

Mode Settings

Modes

- Plan
- Code
- Advanced
- Ask ✔


Start New Task

Fit to drag in files/images

Ln 22, Col 10 (662 selected) Spaces: 4 UTF-8 LF () Shell Script Bob - Settings

IBM i Automation with Bash Programming
113

113

BOB's Solution 1 – Works!


```
-bash-5.1$ cat dltspif_while_do.sh
#!/usr/bin/bash
echo =====
db2util -o csv "SELECT SPOOLED_FILE_NAME, JOB_NAME FROM QSYS2.OUTPUT_QUEUE_ENTRIES_BASIC where (
(OUTPUT_QUEUE_NAME='QEZJOBLOG') AND (CREATE_TIMESTAMP > '2025-12-08 10:00:00.000000') AND (CREATE_TIMESTAMP < '2025-12-10 18:00:00.000000')) fetch first 5 rows only" > Liste.txt
echo =====
while read -r zeile; do
    SPLFNAME=$(echo $zeile | cut -f1 -d, | sed 's"/"/g')
    JOBNAME=$(echo $zeile | cut -f2 -d, | sed 's"/"/g')
    echo "dltspif file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"
    c1 "dltspif file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)" < /dev/null
done < Liste.txt
echo Fertig!
-bash-5.1$ dltspif_while_do.sh
=====
dltspif file(QPJOBLOG) JOB(456896/IWS25004/BATCH2) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltspif file(QPJOBLOG) JOB(456935/IWS25013/DSPSBS0013) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltspif file(QPJOBLOG) JOB(456937/IWS25013/DSPSBS0013) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltspif file(QPJOBLOG) JOB(456940/IWS25013/DSPSBS0013) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltspif file(QPJOBLOG) JOB(456942/IWS25013/DSPSBS0013) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
Fertig!
-bash-5.1$
```

IBM i Automation with Bash Programming
114

114

BOB's Solution 2 – Works!



```
-bash-5.1$ cat dltsplf_bob.sh
#!/usr/bin/bash

echo =====
db2util -o csv "SELECT SPOOLED_FILE_NAME, JOB_NAME FROM QSYS2.OUTPUT_QUEUE_ENTRIES_BASIC where (
(OUTPUT_QUEUE_NAME='QEZJOBLOG') AND (CREATE_TIMESTAMP > '2025-12-08 10:00:00.000000') AND (CREATE
_TIMESTAMP < '2025-12-10 18:00:00.000000')) fetch first 5 rows only" > Liste.txt

echo =====

while read -r zeile <&3; do
    SPLFNAME=$(echo $zeile | cut -f1 -d, | sed 's/"/''g')
    JOBNAME=$(echo $zeile | cut -f2 -d, | sed 's/"/''g')
    echo "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*any) SELECT(*ALL)"
    cl "dltsplf file($SPLFNAME) JOB($JOBNAME) SPLNBR(*ANY) SELECT(*ALL)"
done 3< Liste.txt

echo Fertig!
-bash-5.1$ dltsplf_bob.sh
=====
dltsplf file(QPJOBLOG) JOB(456946/IWS25011/QPADEV0010) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltsplf file(QPJOBLOG) JOB(454611/IWS25013/QPADEV0000) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltsplf file(QPJOBLOG) JOB(457074/IWS25013/QPADEV0000) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltsplf file(QPJOBLOG) JOB(456633/IWS25004/QPADEV000L) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
dltsplf file(QPJOBLOG) JOB(456270/IWS25010/QPADEV000C) SPLNBR(*any) SELECT(*ALL)
CPD0000: Command *LIBL/DLTSPLF not safe for a multithreaded job.
Fertig!
-bash-5.1$
```

IBM i Automation with Bash Programming

115

115

Bob's Solution (mit menschlichem Support)




```
-bash-5.2$ dltsplf_bob_wr.sh 2025-12-18-00-00-00 2025-12-18-18-00-00
=====
IBM i PASE Spoolfile Deletion Tool
=====
[INFO] 2025-12-18 15:53:30 - Normalizing timestamps...
[INFO] 2025-12-18 15:53:30 - Start timestamp: 20251218000000
[INFO] 2025-12-18 15:53:30 - End timestamp: 20251218180000
[INFO] 2025-12-18 15:53:30 - Searching for spoolfiles between 20251218000000 and 20251218180000...
[INFO] 2025-12-18 15:53:30 - Querying spoolfiles from QSYS2.OUTPUT_QUEUE_ENTRIES...
[SUCCESS] 2025-12-18 15:53:31 - Deleted: QPJOBLOG (Job: 906474/SCANOSKI/QDFTJOB0, File#: 2)
[SUCCESS] 2025-12-18 15:53:31 - Deleted: QPJOBLOG (Job: 906475/SCANOSKI/QDFTJOB0, File#: 2)
[SUCCESS] 2025-12-18 15:53:31 - Deleted: QPJOBLOG (Job: 907369/WROTHER/QPADEV0006, File#: 1)
=====
[INFO] 2025-12-18 15:53:32 - Operation Summary:
=====
Spoolfiles deleted: 3
Errors encountered: 0
=====
```

IBM i Automation with Bash Programming

116

116

IBM i Automation with Bash Programming




Using Web Services

© Copyright IBM Corporation 2001, 2016, 2020, 2023
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

117

Example Skripts




Bash	rseapi.sh
SQL	UsingIntegratedWS.sql
Python	rse_login.py

IBM i Automation with Bash Programming

118

118

<https://openlibrary.org/dev/docs/api/search>



The URL format for API is simple. Take the search URL and add .json to the end. Eg:

- <https://openlibrary.org/search.json?q=the+lord+of+the+rings>
- <https://openlibrary.org/search.json?title=the+lord+of+the+rings>
- <https://openlibrary.org/search.json?author=tolkien&sort=new>
- <https://openlibrary.org/search.json?q=the+lord+of+the+rings&page=2>
- <https://openlibrary.org/search/authors.json?q=twain>

Bsp:

```
curl -s https://openlibrary.org/search/authors.json?q=rother | jq .docs[].name
```

IBM i Automation with Bash Programming 119

119




https://libretranslate.com/docs/#/translate/post_translate_file


IBM i Automation with Bash Programming 120

120

IBM i Automation with Bash Programming




PASE Job Management



© Copyright IBM Corporation 2001, 2016, 2020, 2023
 Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

121

bg, fg, &



<https://www.redhat.com/en/blog/jobs-bg-fg>


```

-bash-5.2$ sleep 500
^Z
[1]+  Stopped                sleep 500
-bash-5.2$ jobs
[1]+  Stopped                sleep 500
-bash-5.2$ bg
[1]+  sleep 500 &
-bash-5.2$ jobs
[1]+  Running                 sleep 500 &
-bash-5.2$ fg
sleep 500
^C
-bash-5.2$ jobs
-bash-5.2$

```

IBM i Automation with Bash Programming
122

122

ps & getjobid



\$ ps -A This will show all the processes.

\$ ps -ef This will give you more details about all processes.

```
-bash-5.2$ ps -ef
  UID      PID  PPID  C   STIME  TTY  TIME CMD
  qsecofr   13    0  0 Feb 02   -  0:04 /00penSys/Q1BM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 1738
  qsys      30    1  0 Feb 02   -  2:12 /Q1BM/ProdData/OS/SLP/bin/lslp-kernel -f/Q1BM/UserData/OS/SLP/conf/ibmslp.conf
  qsecoff_  54    1  0 Feb 02   -  0:00 /00penSys/Q1BM/ProdData/SC1/0penSSH/sbin/sshd
  qwebadml 104   103  0 Feb 02   -  1:17 /00penSys/Q1BM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 1805
  qlwisvr   347   1  0 Feb 03   -  9:06 /00penSys/Q1BM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 2273
  qwebadml 348   1  0 Feb 03   -  9:06 /00penSys/Q1BM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2564
  qwebadml 349   1  0 Feb 03   -  0:32 /00penSys/Q1BM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2563
  qwebadml 351   1  0 Feb 03   -  0:32 /00penSys/Q1BM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 2257
  qlwisvr   454   1  0 Feb 03   -  11:06 /00penSys/Q1BM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2569
  qsecofr 17685 16251 0 Feb 16   -  0:00 /00penSys/Q1BM/ProdData/TransforServices/bin/ictt_daemon 0 1
  bob      28499 1  0 Feb 20   -  0:00 python run.py
  bob      2891 28499 0 Feb 20   -  355:21 /home/bob/Phase-5-bob-unverified/venv/bin/python run.py
  qsecofr 38973 1  0 Mar 02   -  0:05 /00penSys/pkg/bin/crond -n
  wrother 39938 1  0 Mar 03   -  0:33 /home/wrother/builds/CVE_Scanner/venv/bin/python /home/wrother/builds/CVE_Scanner/scripts/run_api.py --host f
  qsecoff_ 43384 54  0 04:52:24 -  0:00 /00penSys/Q1BM/ProdData/SC1/0penSSH/sbin/sshd -R
  qsecoff_ 43386 54  0 04:52:34 -  0:00 /00penSys/Q1BM/ProdData/SC1/0penSSH/sbin/sshd -R
  wrother 43387 43384 0 04:52:34 pts/0 -bash
  wrother 43310 43306 0 04:52:35 -  0:00 /00penSys/Q1BM/ProdData/SC1/0penSSH/libexec/sftp-server
  wrother 43315 43307 0 04:58:13 pts/0 0:00 ps -ef
```

IBM i Automation with Bash Programming
123

123

kill command


kill -l list all signals

```
-bash-5.2$ kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGEMT     8) SIGFPE     9) SIGKILL    10) SIGBUS
11) SIGSEGV    12) SIGSYS    13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGURG     17) SIGSTOP   18) SIGTSTP   19) SIGCONT    20) SIGCHLD
21) SIGTTIN    22) SIGTTOU   23) SIGIO     24) SIGXCPU    25) SIGXFSZ
27) SIGMSG     28) SIGWINCH  29) SIGPWR     30) SIGUSR1    31) SIGUSR2
32) SIGPROF    33) SIGDANGER 34) SIGVTALRM 35) SIGMIGRATE 36) SIGPRE
37) SIGVIRT    38) SIGALRM1  39) SIGWAITING 50) SIGRTMIN   51) SIGRTMIN+1
52) SIGRTMIN+2 53) SIGRTMIN+3 54) SIGRTMAX-3 55) SIGRTMAX-2 56) SIGRTMAX-1
57) SIGRTMAX   59) SIGCPUFAIL 60) SIGKAP     61) SIGRETRACT 62) SIGSOUND
63) SIGSAK
```

IBM i Automation with Bash Programming
124

124

Example



```

-bash-5.2$ ps -ef | grep sleep
wrother 89080 89074 0 19:15:27 - 0:12 bash -c while true; do sleep 1;head -v -n 8 /proc/meminfo; head -v -n 8 /proc/loadavg /proc/sys/fs/file-nr /proc/sys/kernel/hostname; tail -v -n 32 /proc/net/dev;echo '=> /proc/<=<';who;echo '=> /proc/end <=<';echo '##Moba##' ; done
wrother 101607 89078 0 19:58:23 pts/0 0:00 sleep 500
wrother 102568 89078 0 20:01:38 pts/0 0:00 sleep 500
wrother 102781 89080 0 20:02:21 - 0:00 sleep 1
-bash-5.2$ kill -s term 101607
-bash-5.2$ ps -ef | grep sleep
wrother 89080 89074 0 19:15:27 - 0:12 bash -c while true; do sleep 1;head -v -n 8 /proc/meminfo; head -v -n 8 /proc/loadavg /proc/sys/fs/file-nr /proc/sys/kernel/hostname; tail -v -n 32 /proc/net/dev;echo '=> /proc/<=<';who;echo '=> /proc/end <=<';echo '##Moba##' ; done
wrother 101607 89078 0 19:58:23 pts/0 0:00 sleep 500
wrother 102568 89078 0 20:01:38 pts/0 0:00 sleep 500
wrother 102843 89080 0 20:02:34 - 0:00 sleep 1
-bash-5.2$ kill -s kill 101607
-bash-5.2$ ps -ef | grep sleep
wrother 89080 89074 0 19:15:27 - 0:12 bash -c while true; do sleep 1;head -v -n 8 /proc/meminfo; head -v -n 8 /proc/loadavg /proc/sys/fs/file-nr /proc/sys/kernel/hostname; tail -v -n 32 /proc/net/dev;echo '=> /proc/<=<';who;echo '=> /proc/end <=<';echo '##Moba##' ; done
wrother 102568 89078 0 20:01:38 pts/0 0:00 sleep 500
wrother 102911 89080 0 20:02:47 - 0:00 sleep 1
[2]+ Killed sleep 500
-bash-5.2$

```

125

Terminate a process



To kill a process in PASE on IBM i, first use a command like `ps` to find the Process ID (PID) of the process you want to end, and then use the kill command with the PID.

For example, to terminate a process with PID 4685, you would type

```

kill -s term 4685
kill -s kill 4685      oder kill -kill 4685 oder kill -9 4685

```


The difference between `SIGTERM` and `SIGKILL` is the way an application may act on the signal:

TERM: an application will be able to terminate, i.e. properly run a shutdown routine.

KILL: the applications is stopped and killed immediately (which could lead to data loss or raising appport to report a presumed crash in some cases).

126

getjobid, QPOZSPWT Jobs




```

-bash-5.25 ps -ef
  UID      PID     PPID    C   STIME   TTY   TIME CMD
qsecofr   13      8      0   Feb 02   -   0:04 /00penSys/QIBM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 1738
qsys      30      1      0   Feb 02   -   2:12 /QIBM/ProdData/OS/SLP/bin/lslp-kernel -f/QIBM/UserData/OS/SLP/conf/ubslp.conf
qsecofr   54      1      0   Feb 02   -   0:00 /00penSys/QIBM/ProdData/SC1/OpenSSH/sbin/ssh
qwebadm1 184     103    0   Feb 02   -   1:17 /00penSys/QIBM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 1885
qlvlsvr   347     1      0   Feb 03   -   9:06 /00penSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 2273
qwebadm1 348     1      0   Feb 03   -   9:06 /00penSys/QIBM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2564
qwebadm1 349     1      0   Feb 03   -   0:32 /00penSys/QIBM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2563
qwebadm1 351     1      0   Feb 03   -   0:32 /00penSys/QIBM/ProdData/JavaVM/jdk80/64bit/jre/lib/ppc64/jvmStartPase 2257
qlvlsvr   454     1      0   Feb 03   -   11:06 /00penSys/QIBM/ProdData/JavaVM/jdk17/64bit/lib/jvmStartPase 2569
qsecofr 17685 16251  0   Feb 16   -   0:00 /00penSys/QIBM/ProdData/TransformServices/bin/lcft_daemon 0 1
bob       28499   1      0   Feb 20   -   0:00 python run.py
bob       28501 28499  0   Feb 20   -   356:21 /home/bob/Phase-5-bob-unverified/venv/bin/python run.py
qsecofr 38973   1      0   Mar 02   -   0:05 /00penSys/pkgs/bin/crond -m
wrother 39938   1      0   Mar 03   -   0:33 /home/wrother/builds/CVE_Scanner/venv/bin/python /home/wrother/builds/CVE_Scanner/scripts/run_api.py --host 0
qsecofr 43304   54     0   04:52:24 -   0:00 /00penSys/QIBM/ProdData/SC1/OpenSSH/sbin/ssh -R
qsecofr 43386   54     0   04:52:34 -   0:00 /00penSys/QIBM/ProdData/SC1/OpenSSH/sbin/ssh -R
wrother 43307 43304  0   04:52:34 pts/0  0:00 -bash
wrother 43310 43306  0   04:52:35 -   0:00 /00penSys/QIBM/ProdData/SC1/OpenSSH/libexec/sftp-server
wrother 43315 43307  0   04:58:13 pts/0  0:00 ps -ef
-bash-5.25
-bash-5.25
-bash-5.25
-bash-5.25 getjobid 28501
Process identifier 28501 is 667669/QSEC0FR_NC/QPOZSPWT
-bash-5.25
    
```

IBM i Automation with Bash Programming
127

127

QPOZSPWT Jobs



Job Name ↑↓	Detailed Status ↑↓	Current User ↑↓	Enhanced Type ↑↓
QPOZSPWT <input type="text" value="Filter"/>	<input type="text" value="Filter"/>	<input type="text" value="Filter"/>	<input type="text" value="Filter"/>
<input checked="" type="checkbox"/> QPOZSPWT	SELW	QSEC0FR_NC	
<input checked="" type="checkbox"/> QPOZSPWT	THDW	BOB	
<input checked="" type="checkbox"/> QPOZSPWT	THDW	BOB	

- QPOZSPW Printer Output
- QPOZSPW Job Log
- QPOZSPW Details >
- QPOZSPW Reply
- QPOZSPW Hold
- QPOZSPW Release
- QPOZSPW End
- QPOZSPW Move
- QPOZSPW Disconnect
- QPOZSPW Performance >
- QPOZSPW Properties

WROTHER

667669/QSEC0FR_NC/QPOZSPWT File System Objects

Object ↑↓	File System Type ↑↓
/	ROOT
/dev/null	ROOT
/dev/urandom	ROOT
/home/bob/Phase-5-bob-unverified/logs/app.log	ROOT
/home/bob/Phase-5-bob-unverified/logs/app.out	ROOT


Total Rows: 60 << 1 >> 100

IBM i Automation with Bash Programming
128

128

CEC 2026

64

Ricardo Martins Cloud Scripts	
https://github.com/rqmartins73/bluexport_api/tree/main	
IBM i Automation with Bash Programming 129	

129