



iSeries

OS/400 PASE Shells and Utilities





@server

iSeries

OS/400 PASE Shells and Utilities

Contents

OS/400 PASE Shells and Utilities	1
What's new in V5R2	2
Print this topic	4
OS/400 PASE Commands.	4
system - Run a CL Command for OS/400 PASE	12
qsh, qsh_inout, qsh_out - Run a QShell Command for OS/400 PASE	14

OS/400 PASE Shells and Utilities

OS/400^(R) Portable Application Solutions Environment (OS/400 PASE) includes three shells (Korn, Bourne, and C Shell) and over 200 utilities that run as OS/400 PASE programs. OS/400 PASE shells and utilities provide an extensible scripting environment that includes a large number of industry-standard and defacto-standard commands.

To become more familiar with OS/400 PASE shells and utilities, see the following topics. You can also find information about what's new in this release and how you can print this topic.

- OS/400 PASE Commands
- system - Run a CL command for OS/400 PASE
- qsh, qsh_inout, qsh_out - Run a QShell command for OS/400 PASE

Using the shells and utilities

A simple way to access OS/400 PASE shells and utilities is to call program Run an OS/400 PASE Terminal Session (QP2TERM), which presents an interactive display with a command line that lets you enter OS/400 PASE commands. You can also call program Run any OS/400 PASE program (QP2SHELL) to run any OS/400 PASE program, including a shell or utility.

Many OS/400 PASE utilities have the same name (with similar options and behavior) as QShell utilities in directory /usr/bin, so OS/400 PASE utilities are provided in directory /QOpenSys/usr/bin or /QOpenSys/usr/sbin. The OS/400 PASE **PATH** environment variable should generally include directories /QOpenSys/usr/bin, /QOpenSys/usr/bin/X11 and /QOpenSys/usr/sbin when running an OS/400 PASE shell. See Run any OS/400 PASE program (QP2SHELL) for information about setting initial values for OS/400 PASE environment variables.

Overview of OS/400 PASE utilities

See OS/400 PASE Commands for information about these OS/400 PASE utilities:

admin	dspcat	kill	qsh	time
aixterm	dspmsg	ksh	qsh_inout	tnameserv
ajar	du	ksh93	qsh_out	touch
alias	dump	ld	ranlib	tput
appletviewer	echo	lex	read	tr
apply	ed	line	red	trace
ar	edit	ln	regcmp	trbsd
as	egrep	locale	reset	trcoff
attr	env	locale64	resize	trcon
awk	ex	logger	rev	trcstop
banner	execerror	logname	Rfile	true
basename	expand	look	rgb	tset
bc	expr	lorder	rm	tsort
bdiff	extcheck	ls	rmdel	tty
bfs	false	m4	rmdir	type
bg	fc	make	rmic	ulimit
bsh	fg	makekey	rmid	umask
cat	fgrep	mkcatdefs	rmiregistry	unalias
cb	file	mkdir	rtl_enable	uname
cd	find	mkfifo	runcat	uncompress
cdc	fold	mkfontdir	sact	unexpand
chgrp	gencat	mknod	sccs	unget
chmod	get	more	sccsdiff	unifdef
chown	getconf	mv	sdiff	uniq
chroot	getjobid	mwm	sed	unpack
cksum	getopt	native2ascii	serialver	untab
clear	getopts	nawk	setccsid	val
clrtmp	grep	newform	setmaps	vc

cmp	hash	nice	sh	vedit
colrm	head	nl	size	vi
comb	hostname	nm	sleep	view
comm	iconv	nohup	sort	wait
command	id	od	split	wc
compress	indent	pack	strings	what
cp	install	pagesize	strip	which
cpio	ipcrm	paste	stty	X
csd	ipcs	patch	sum	xargs
csplit	jar	pax	syslogd	xauth
cut	jarsigner	pcat	system	xhost
date	java	pg	sysval	xlsfonts
dbx	javac	policytool	tab	xmodmap
dc	javadoc	pr	tabs	xset
dd	javah	printenv	tail	xterm
delta	javakey	printf	tar	xwd
diff	javap	prn	tee	xwud
diff3	jobs	ps	termdef	yacc
dircmp	join	psh	test	yes
dirname	keytool	pwd	tic	zcat

In addition to these utilities, each OS/400 PASE shell supports a number of built-in commands (such as **cd**, **exec**, **if**, etc.). See AIX^(R) documentation for information about the built-in commands supported by each OS/400 PASE shell.

The OS/400 PASE default shell (/QOpenSys/usr/bin/sh) is the Korn shell.

Top | APIs by category

What's new in V5R2

Here are some of the significant enhancements and changes for OS/400 PASE in V5R2:

- Pseudo terminal (PTY) support and UNIX-style job control. See Pseudo terminal (PTY) for details.
- Over 100 new utilities. See the OS/400 PASE Shells and Utilities topic for a complete listing.
- The following new libraries were added. See the OS/400 PASE Run-time Libraries topic for a complete listing.

libcur.a	AIX legacy Curses library
libg.a	Debug support
libgair4.a	Internal X Windows support
libl.a	lex support
libld.a	Object File Access Routine library
libm.a	IEEE Math library
libPW.a	Programmers Workbench library
libxcurses.a	Curses library
libXi.a	X Windows input processing
libXtst.a	X Windows testing support
liby.a	yacc support

- A new message (MCH3204) appears in the job log for any unsupported system call used by an OS/400 PASE program. This message text includes the name of the system call and the instruction address that caused the error.
- New and changed OS/400 PASE run-time functions:
 - **_CVTERRNO** (convert OS/400 PASE errno to ILE errno)
 - **_ILECALLX** (enhanced ILE procedure call)
 - **_PMGCALL** (call an OS/400 program)
 - **_RETURN** (return without exiting OS/400 PASE)



- _RSLOBJ, _RSLOBJ2 (resolve to an OS/400 object)
- _STRLEN_SPP, _STRCPY_SPP (string manipulation using 16-byte ILE pointers)
- Qp2paseCCSID (retrieve OS/400 PASE CCSID)
- Qp2jobCCSID (retrieve job default CCSID from last time OS/400 PASE CCSID set)
- faccessx
- fchdir
- fclear
- fclear
- getaddrinfo, getnameinfo
- getcontext, setcontext
- getpri, getpriority, setpriority
- getprocs64, getthrs64
- gettimer, settimer
- msem_init, msem_lock, msleep, msem_unlock, msem_remove
- pread, pwrite
- setgroups
- sigstack, sigaltstack (alternate signal stack)
- statpriv
- statvfs, fstatvfs
- sync
- ustat
- New and changed (ILE) APIs for OS/400 PASE:
 - QP2SHELL2 (like QP2SHELL, but runs in caller's activation group)
 - Qp2ptrsize (retrieve OS/400 PASE pointer size)
 - Qp2paseCCSID (retrieve OS/400 PASE CCSID)
 - Qp2jobCCSID (retrieve job default CCSID when OS/400 PASE CCSID last set)
 - Qp2errnop (locate OS/400 PASE errno for current thread)
 - Qp2malloc (allocate OS/400 PASE heap memory)
 - Qp2free (free OS/400 PASE heap memory)
 - Qp2dlopen (dynamically load an OS/400 PASE module)
 - Qp2dlsym (find a symbol in a module opened by OS/400 PASE dlopen)
 - Qp2dlclose (close and unload a module loaded by OS/400 PASE dlopen)
 - Qp2dlerror (retrieve error information for the last dynamic load operation)
 - Qp2CallPase (and Qp2CallPase2) enhancements for by-address arguments and results, and to call an OS/400 PASE procedure in a thread that was not started by OS/400 PASE
- OS/400 PASE locales (and other files for globalization support) are now packaged with OS/400 language feature codes; see Globalization for details. In addition, over 200 new files for X Windows processing of different keyboards and character sets, plus the following 65 new locales. See the OS/400 PASE Locales topic for a complete listing.

AR_AE.UTF-8	ES_CO.UTF-8	de_AT.8859-15
AR_BH.UTF-8	ES_MX.UTF-8	de_AT.8859-15@euro
AR_EG.UTF-8	ES_PE.UTF-8	de_LU.8859-15
AR_J0.UTF-8	ES_PR.UTF-8	de_LU.8859-15@euro
AR_KW.UTF-8	ES_UY.UTF-8	en_CA.8859-15
AR_LB.UTF-8	ES_VE.UTF-8	en_IE.8859-15
AR_OM.UTF-8	FR_LU.UTF-8	en_IE.8859-15@euro
AR_QA.UTF-8	FR_LU.UTF-8@euro	en_IN.8859-15

AR_SA.UTF-8	HI_IN.UTF-8	en_NZ.8859-15
AR_SY.UTF-8	SH_YU.UTF-8	es_AR.8859-15
AR_TN.UTF-8	SR_YU.UTF-8	es_CL.8859-15
DE_AT.UTF-8	ar_AE.ISO8859-6	es_CO.8859-15
DE_AT.UTF-8@euro	ar_BH.ISO8859-6	es_MX.8859-15
DE_LU.UTF-8	ar_EG.ISO8859-6	es_PE.8859-15
DE_LU.UTF-8@euro	ar_JO.ISO8859-6	es_PR.8859-15
EN_CA.UTF-8	ar_KW.ISO8859-6	es_UY.8859-15
EN_IE.UTF-8	ar_LB.ISO8859-6	es_VE.8859-15
EN_IE.UTF-8@euro	ar_OM.ISO8859-6	fr_LU.8859-15
EN_IN.UTF-8	ar_QA.ISO8859-6	fr_LU.8859-15@euro
EN_NZ.UTF-8	ar_SA.ISO8859-6	sh_YU.ISO8859-2
ES_AR.UTF-8	ar_SY.ISO8859-6	sr_YU.ISO8859-5
ES_CL.UTF-8	ar_TN.ISO8859-6	

How to see what's new or changed

To help you see where technical changes have been made, this information uses:

- The  image to mark where new or changed information begins.
- The  image to mark where new or changed information ends.

To find other information about what's new or changed this release, see the Memo to Users  .

Print this topic


To view or download the PDF version, select OS/400 PASE Shells and Utilities information (about 119 KB or 24 pages).

Saving PDF files

To save a PDF on your workstation for viewing or printing:



1. Right-click the PDF in your browser (right-click the link above).
2. Click **Save Target As...**
3. Navigate to the directory in which you would like to save the PDF.
4. Click **Save**.

Downloading Adobe Acrobat Reader

If you need Adobe Acrobat Reader to view or print these PDFs, you can download a copy from the Adobe Web site (www.adobe.com/products/acrobat/readstep.html)  .

OS/400 PASE Commands

Most OS/400 PASE commands support the same options and provide the same behavior as AIX commands, except in these areas:

-  Many OS/400 PASE commands for display operations and UNIX(TM) job control only work in a TTY session such as a session started by the **aixterm** or **xterm** command. These functions do not work on 5250 workstation devices (including the display presented by program **QP2TERM**). 
- OS/400 PASE generally does not support interfaces provided on AIX for system management. For example, OS/400 PASE provides no support for the AIX System Management Interface Tool (SMIT) and does not support functions that require a SMIT database.

- OS/400 is fundamentally an EBCDIC system. OS/400 PASE shells and utilities run in ASCII and generally do no automatic conversion of stream data. You may need to use tools such as the **iconv** utility to convert between ASCII and EBCDIC encoding.

Unlike the QShell interpreter and utilities, most OS/400 PASE shells and utilities do *no* automatic Coded Character Set Identifier (CCSID) conversion of stream file data. The OS/400 PASE utilities **system**, and any OS/400 PASE utility that runs a QShell command are exceptions because they provide CCSID conversion support for data the CL command or QShell command reads from standard input or writes to standard output or standard error.

» OS/400 PASE utilities that run QShell Java^(TM) utilities (such as the **java** command) set the Java `file.encoding` property to match the OS/400 PASE CCSID, so that stream data read and written by the Java program is converted from and to the OS/400 PASE CCSID. You can force a specific `file.encoding` value by setting OS/400 PASE environment variable `PASE_JAVA_ENCODING` before running the utility.◀

- OS/400 uses case-insensitive names for many system resources that have case-sensitive names in AIX (user and group names, object names in the root file system, etc.). Some OS/400 PASE shell and utility functions require matching case for resources that have case-insensitive names in OS/400, and others may return names in uppercase that would normally be lowercase on AIX. For example, file name expansion in OS/400 PASE shells is case-sensitive, so you must specify uppercase to match generic names in the /QSYS.LIB file system:

```
ls /qsys.lib/qgpl.lib/GEN*.PGM
rather than
ls /qsys.lib/qgpl.lib/gen*.pgm
```

- To provide case-sensitivity and avoid name collisions with directories and files used for ILE support, most OS/400 PASE directories and files (including shells and utilities) are stored in the /QOpenSys file system. In particular, OS/400 PASE shells and utilities are in /QOpenSys/usr/bin and /QOpenSys/usr/sbin (rather than /usr/bin and /usr/sbin on AIX).

In addition to the OS/400 PASE commands listed below, each OS/400 PASE shell supports a number of built-in commands (such as **cd**, **exec**, **if**, etc.). See AIX documentation for information about the built-in commands supported by each OS/400 PASE shell and for detailed information about most of these OS/400 PASE commands:

A (see page 5) B (see page 6) C (see page 6) D (see page 6) E (see page 6) F (see page 7) G (see page 7) H (see page 7) I (see page 7) J (see page 7) K (see page 8) L (see page 8) M (see page 8) N (see page 8) O (see page 9) P (see page 9) Q (see page 9) R (see page 9) S (see page 10) T (see page 10) U (see page 11) V (see page 11) W (see page 11) X (see page 11) Y (see page 11) Z (see page 12)

» **admin**

» **aixterm**

» **ajar**

alias

» **appletviewer**

apply

ar

» **as**

» **attr**

A

Create and control Source Code Control System (SCCS) files ◀

Initialize an Enhanced X-Windows terminal emulator ◀

Run the QShell **ajar** command, which is the alternative Java archive tool ◀

Define or display aliases

Run the QShell **appletviewer** command, to run Java applets without a web browser ◀

Apply a command to a set of parameters

Maintain the indexed libraries used by the linkage editor

Run the assembler ◀

Run the QShell **attr** command to display or change Integrated File System object attributes ◀

awk	Find lines in files matching patterns and then perform specified actions on them
banner	B Write ASCII character strings in large letters to standard output
basename	Return the base filename of a string parameter
bc	Provide an interpreter for arbitrary-precision arithmetic language
bdiff	Use the diff command to find differences in very large files
bfs	Scan files
bg	Run a job in the background
bsh	Invoke the Bourne shell
cat	C Concatenate or display files
cd	Change the current directory
» cdc	Change the comments in a SCCS delta «
chgrp	Change the group ownership of a file or directory
chmod	Change permission modes
chown	Change the user associated with a file
chroot	Change the root directory of a command
cksum	Display the checksum and byte count of a file
» clear	Clear the terminal screen «
» clrtmp	Run the QShell clrtmp command to clear directory /tmp «
cmp	Compare two files
colrm	Extract columns from a file
» comb	Combine SCCS deltas «
comm	Select or reject lines common to two sorted files
command	Execute a simple command
compress	Compress data
cp	Copy files
cpio	Copy files into and out of archive storage and directories
csch	Invoke the C shell
csplit	Split files by context
cut	Write out selected bytes, characters, or fields from each line of a file
date	D Display or set the date or time
dbx	Provide an environment to debug and run OS/400 PASE programs
dc	Provide an interactive desk calculator for arbitrary-precision integer arithmetic
dd	Convert and copy a file
diff	Compare text files
diff3	Compare three files
dircmp	Compare two directories and the contents of their common files
dirname	Write to standard output all but the last part of a specified path
dspcat	Display all or part of a message catalog
dspmsg	Display a selected message from a message catalog
du	Summarize disk usage
dump	Dump selected parts of an object file
	E

echo	Write character strings to standard output
» ed	Edit text by line «
» edit	Provide a simple line editor for the new user «
egrep	Search a file for a pattern
env	Display the current environment or set the environment for the execution of a command
» ex	Edit lines interactively, with a screen display «
execerror	Write error messages to standard error
expand	Write to standard output with tabs changed to spaces
expr	Evaluate arguments as expressions
» extcheck	Run the QShell extcheck command, to detect Java archive conflicts «
false	F
fc	Return a nonzero exit value (false)
fg	Process the command history list
fgrep	Run jobs in the foreground
file	Generate the figure list in a format supported by the build process
find	Determine file type
fold	Find files with a matching expression
» gencat	Fold long lines for finite-width output device
» get	G
getconf	Create and modify a message catalog «
» getjobid	Create a specified version of a SCCS file «
getopt	Write system configuration variable values to standard output
getopts	Run the QShell getjobid command to determine the OS/400 job name for a process identifier «
grep	Parse command line flags and parameters
hash	Process command-line arguments and check for valid options
head	Search a file for a pattern
hostname	H
iconv	Remember or report command path names
id	Display the first few lines or bytes of a file or files
» indent	Set or display the name of the current host system
install	I
» ipcs	Convert the encoding of characters from one code page encoding scheme to another
» ipcrm	Display the system identifications of a specified user
» jar	Reformat a C language program «
» jarsigner	Install a command
» java	Run the QShell ipcs command to display interprocess communications objects «
	Run the QShell ipcrm command to remove interprocess communications objects «
	J
	Run the QShell jar command, to archive Java files «
	Run the QShell jarsigner command, to sign or verify the signature of a Java archive «
	Run the QShell java command, to run the Java interpreter «

» javac	Run the QShell javac command, to compile a Java program <<
» javadoc	Run the QShell javadoc command, to generate Java documentation <<
» javah	Run the QShell javah command, to generate C header or stub files for Java classes <<
» javakey	Run the QShell javakey command, to manage Java security keys <<
» javap	Run the QShell javap command, to disassemble a compiled Java program <<
jobs	Display status of jobs in the current session
join	Join the data fields of two files
K	
» keytool	Run the QShell keytool command, to manage keys and certificates for Java <<
kill	Send a signal to running processes
ksh	Invoke the Korn shell
» ksh93	Invoke the enhanced Korn shell <<
L	
» ld	Link object files <<
» lex	Generate a C or C++ language program that matches patterns for simple lexical analysis of an input stream <<
line	Read one line from the standard input
ln	Link files
locale	Write information about current locale or all public locales
logger	Make entries in the system log
logname	Display login name
» look	Find lines in a sorted file <<
» lorder	Find the best order for member files in an object library <<
ls	Display the contents of a directory
M	
» m4	Preprocess files, expanding macro definitions
» make	Maintain, update, and regenerate groups of programs <<
» makekey	Generate an encryption key <<
» mkcatdefs	Preprocess a message source file <<
mkdir	Create one or more new directories
mkfifo	Make first-in-first-out (FIFO) special files
» mkfontdir	Create a fonts.dir file from a directory of font files <<
mknod	Create a special file
» more	Display the contents of files one screen at a time <<
mv	Move files
» mwm	Run the AIXwindows ^(R) Window Manager (MWM) <<
N	
» native2ascii	Run the QShell native2ascii command, to convert characters encoded in the OS/400 PASE CCSID to Unicode encoding <<
nawk	Invoke the new version of awk
newform	Change the format of a text file
» nice	Run a command at a lower or higher priority <<
nl	Number lines in a file

nm	Display the symbol table of an object file
nohup	Run a command without hangups
	O
od	Display files in a specified format
	P
pack	Compress files
pagesize	Display the system page size
paste	Merge the lines of several files or subsequent lines in one file
patch	Apply changes to files
pax	Extract, write, and list members of archive files; copy files and directory hierarchies
pcat	Unpack files and write them to standard output
» pg	Format files to the display <<
» policytool	Run the QShell policytool command, to create and manage Java policy files <<
pr	Write a file to standard output
printenv	Display the values of environment variables
printf	Write formatted output
» prs	Display a Source Code Control System (SCCS) file <<
ps	Show current status of processes
psh	Invoke the POSIX (Korn) shell
pwd	Display the pathname of the working directory
	Q
» qsh	Run a QShell command <<
» qsh_inout	Run a QShell command <<
» qsh_out	Run a QShell command <<
	R
» ranlib	Convert archive libraries to random libraries <<
read	Read one line from standard input
» red	Edit text by line <<
» regcmp	Compile patterns into C language char declarations <<
» reset	Initialize a terminal <<
» resize	Set the TERMCAP environment variable and terminal settings to the current window size <<
rev	Reverse characters in each line of a file
» Rfile	Run the QShell Rfile command to read or write OS/400 record files <<
» rgb	Create the database used by the X-Window system server for colors <<
rm	Remove (unlink) files or directories
» rmdel	Remove a delta from a SCCS file <<
rmdir	Remove a directory
» rmic	Run the QShell rmic command, to compile Java RMI stubs <<
» rmid	Run the QShell rmid command, to run the Java RMI activation system <<
» rmiregistry	Run the QShell rmiregistry command, to start a Java remote object registry <<
» rtl_enable	Relink shared objects to enable the runtime linker to use them <<

» **runcat**

» **sact**

» **serialver**

» **sccs**

» **sccsdiff**

» **sdiff**

sed

» **setmaps**

» **setccsid**

sh

» **size**

sleep

sort

split

strings

strip

» **stty**

sum

syslogd

system

» **sysval**

tab

» **tabs**

tail

tar

tee

test

» **tic**

time

» **tnameserv**

touch

» **tput**

tr

» **trace**

» **trbsd**

» **trcoff**

Pipe output data from the **mksccs** command to the **gencat** command <<

S

Display current SCCS file-editing status <<

Run the QShell **serialver** command, to return the version number for Java classes <<

Administration program for SCCS commands <<

Compare two versions of a SCCS file <<

Compare two files and display the differences in a side-by-side format <<

Provide a stream editor

Set terminal maps or code set maps <<

Run the QShell **setccsid** command to set the CCSID for an Integrated File System object <<

Invoke the default (Korn) shell

Display the section sizes of the Extended Common Object File Format (XCOFF) object files <<

Suspend execution for an interval

Sort files, merge files that are already sorted, and check files to determine if they have been sorted

Split a file into pieces

Find the printable strings in an object or binary file

Reduce the size of an Extended Common Object File Format (XCOFF) object file by removing information used by the binder and symbolic debug program

Set, reset, and report workstation operating parameters <<

Display the checksum and block count of a file

Log system messages

Run a CL command

Run the QShell **sysval** command to display an OS/400 system value or network attribute <<

T

Change spaces into tabs

Set tab stops on a terminal <<

Write a file to standard output, beginning at a specified point

Manipulate archives

Display the output of a program and copy it into a file

Evaluate conditional expressions

Translate the terminfo description files from source to compiled format <<

Print the time of the execution of a command

Run the QShell **tnameserv** command, to provide access to the Java naming service <<

Update the access and modification times of a file

Query the **terminfo** database for terminal-dependent information <<

Translate characters

Record selected system events <<

Translate characters (BSD version) <<

Stop the collection of trace data <<

» trcon	Start the collection of trace data <<
» trcstop	Stop the trace function <<
true	Return an exit value of zero (true)
» tset	Initialize a terminal <<
» tsort	Sort an unordered list of ordered pairs (a topological sort) <<
» tty	Write to standard output the full path name of your terminal <<
type	Write a description of the command type
ulimit	U Set or report user resource limits
umask	Display or set the file mode creation mask
unalias	Remove alias definitions
uname	Display the name of the current operating system
uncompress	Restore compressed files
unexpand	Write to standard output with tabs restored
» unget	Cancel a previous SCCS get command <<
» unifdef	Remove ifdef lines from a file <<
uniq	Delete repeated lines in a file
unpack	Expand files
untab	Change tabs into spaces
» val	V Validate SCCS files <<
» vc	Substitute assigned values for identification keywords <<
» vedit	Edit files with a full-screen display <<
» vi	Edit files with a full-screen display <<
» view	Start the vi editor in read-only mode <<
wait	W Wait until the termination of a process ID
wc	Count the number of lines, words, and bytes in a file
what	Display identifying information in files
which	Locate a program file, including aliases and paths (the csh (C shell) command only)
» x	X Run the X server. OS/400 PASE only supports virtual frame buffer processing <<
xargs	Construct a parameter list and run a command
» xauth	Edit and display the authorization information used in connecting to the X server <<
» xhost	Control who accesses Enhanced X-Windows on the current host machine <<
» xlsfonts	Display the font list for X-Windows <<
» xmodmap	Modify keymaps in the X Server <<
» xset	Set options for your X-Windows environment <<
» xterm	Provide a terminal emulator for the X Window System <<
» xwd	Dump the image of an Enhanced X-Windows window <<
» xwud	Retrieve and display the dumped image of an Enhanced X-Windows window <<
» yacc	Y Generate an LALR(1) parsing program from input consisting of a context-free grammar specification <<

yes	Output an affirmative response repetitively
zcat	Expand a compressed file to standard output

Top | OS/400 PASE Shells and Utilities

system - Run a CL Command for OS/400 PASE

Syntax

```
system [-beEhiIkKn0qsv] CL-command [ CL-parameters ... ]
```

Description

The OS/400 PASE **system** utility runs a CL command. By default, any spooled output produced by the command is written to standard output, and any messages sent by the command are written to standard output or standard error (depending on whether the CL command sent an exception message).

You need to set ILE environment variable **QIBM_USE_DESCRIPTOR_STDIO** to Y or I (so that OS/400 PASE runtime and ILE C runtime use descriptor standard I/O) to avoid unpredictable results. This is done by default in the OS/400 jobs that program **QP2TERM** uses to run OS/400 PASE shells and utilities.

Options

- b** Force binary mode for standard streams used by the CL command. When this option is omitted, the **system** command converts any data the CL command reads from standard input from the OS/400 PASE CCSID to the job default CCSID, and converts data written to standard output or standard error from the job default CCSID to the OS/400 PASE CCSID. This option avoids CCSID conversion for all standard streams except those associated with any of the options **-E**, **-I**, and **-O**.
- e** Copy OS/400 PASE environment variables to ILE environment variables before running the CL command. When this option is omitted, no ILE environment variables are set, so the ILE environment may be missing variables or have different variable values than the OS/400 PASE environment.

For most variables, the copy has the same name as the original, but the system adds a prefix "**PASE_**" to the name of the ILE copy of some environment variables. You can control what variables add the name prefix by storing a colon-delimited list of variable names in OS/400 PASE environment variable **PASE_ENVIRON_CONFLICT**, or the system defaults to adding the prefix when copying OS/400 PASE environment variables **SHELL**, **PATH**, **NLSPATH**, and **LANG**.

Any OS/400 PASE environment variable name with a prefix "**ILE_**" is copied to the ILE environment twice. The first copy uses the same variable name, and the second copy uses the name without the prefix. For example, if the OS/400 PASE environment contains a variable named **ILE_PATH**, the value of this variable is used to set both **ILE_PATH** and **PATH** in the ILE environment.

- **-E** Force CCSID conversion for the standard error stream used by the CL command. When this option is specified, the **system** command converts any data the CL command writes to standard error from the job default CCSID to the OS/400 PASE CCSID. This option overrides option **-b** for the standard error stream. ⏪
- h** Write a brief description of allowable syntax for the **system** command to standard output.
- i** Run the CL command in the same process (OS/400 job) where the **system** utility runs. When option **-i** is omitted, the CL command is run in a separate process (created using the ILE **spawn**

API) that is not multithread-capable and is not running an OS/400 PASE program. Many CL commands are not supported in a multithreaded job.

- **-I** Force CCSID conversion for the standard input stream used by the CL command. When this option is specified, the **system** command converts any data the CL command reads from standard input from the OS/400 PASE CCSID to the job default CCSID. This option overrides option **-b** for the standard input stream. CCSID conversion should only be used for standard input if the CL command reads standard input because processing done by the **system** command attempts to read and convert all standard input data regardless of whether the CL command uses the data, so it may leave the standard input stream positioned beyond what the CL command read. ⏪
- k** Keep all spool files generated by the CL command. When this option is omitted, spooled output files are deleted after their contents is written as text lines to standard output. Option **-i** has no effect when option **-s** is used.
- K** Force a job log for the OS/400 job where the CL command runs. If this option is omitted, a job log may only be produced if an unexpected error occurs.
- n** Do not include OS/400 message identifiers in any text line written to standard output or standard error for a message sent by the CL command. When this option is omitted, the format of any text lines written for OS/400 pre-defined messages is "XXX1234: message text", where "XXX1234" is the OS/400 message identifier. **-n** suppresses the message identifier, so only "message text" is written to the stream. Option **-n** has no effect when option **-q** is used.
- **-E** Force CCSID conversion for the standard output stream used by the CL command. When this option is specified, the **system** command converts any data the CL command writes to standard output from the job default CCSID to the OS/400 PASE CCSID. This option overrides option **-b** for the standard output stream. ⏪
- q** Do not write any text lines to standard output or standard error for OS/400 messages sent by the CL command. If this option is omitted, messages sent by the CL command are received, converted from the job default CCSID to the OS/400 PASE CCSID, and written as a text lines to standard output or standard error, depending on whether the CL command sent an exception message.
- s** Do not process spooled output files produced by the CL command. When this option is omitted, spooled output generated by the CL command is converted from the job default CCSID to the OS/400 PASE CCSID and written to standard output, and then the spooled output files are deleted.
- v** Write the complete CL command string to standard output before running the CL command.

Operands

CL-command is concatenated with any *CL-parameters* operands (with a single space between them) to form the CL command string. You may need to enclose CL command and parameter values in quotes to prevent the OS/400 PASE shell from expanding special characters (such as parentheses and asterisks).

If a CL command parameter value requires quotes (such as a text parameter with lowercase characters or embedded blanks), you must specify those quotes inside a quoted string because OS/400 PASE shells remove the outer quotes from any argument passed to the OS/400 PASE **system** utility.

Exit Status

- 0** The CL command completed successfully.
- >0** The CL command either could not be run or ended in error (sent an exception message to the caller). Error messages may have been sent to standard output, or may appear in the job log of the OS/400 job that ran the command.

Examples

This example shows three ways to run the **CRTDTAARA** CL command with the same parameter values. Options **-b0E** force CCSID conversion for standard output and standard error (but not standard input). The **"*char"** parameter value must be quoted to prevent the OS/400 PASE shell from expanding it as a set of file names, and the **TEXT** parameter requires two sets of enclosing quotes because it contains lowercase and embedded blanks:

```
system -b0E "crtdataara mydata *char text('Output queue text')"  
or  
system -b0E crtdataara mydata "*char text('Output queue text')"  
or  
system -B0E crtdataara mydata '*char' "text('Output queue text')"
```

This example shows how the **system** utility can run the **CALL** CL command to call a program that accepts two parameters. Option **-i** avoids the overhead of creating an additional process to run the CL command. Since no other options are specified, CCSID conversion is done for standard input, standard output, and standard error. The called program sees the first parameter converted to uppercase (**ARG1**) and the second parameter unchanged (**arg2**) because of the rules of CL:

```
system -i "call mypgm (arg1 'arg2')"
```

Top | OS/400 PASE Shells and Utilities

qsh, qsh_inout, qsh_out - Run a QShell Command for OS/400 PASE

Syntax

```
qsh [command-options]
```

```
qsh_inout [command-options]
```

```
qsh_out [command-options]
```

Description

➤ The OS/400 PASE **qsh**, **qsh_inout**, and **qsh_out** commands run a QShell command. These commands use the OS/400 PASE **system** command to copy OS/400 PASE environment variables to the ILE environment and then call the QShell command program through a link in directory `/usr/bin`.

The OS/400 PASE **qsh**, **qsh_inout**, and **qsh_out** commands all provide the syntax and behavior of the QShell **qsh** command, with additional support for ASCII/EBCDIC conversion of standard I/O provided by the OS/400 PASE **system** command. Any other command name that links to OS/400 PASE **qsh**, **qsh_inout**, or **qsh_out** (in directory `/QOpenSys/usr/bin`) provides the same syntax and behavior as the QShell command in directory `/usr/bin` with the same base name as the link. See the following topics for more information:

- **system** - Run a CL Command for OS/400 PASE
- **qsh** - QShell Command Language Interpreter (QShell version)

The **qsh** and **qsh_inout** commands do ASCII/EBCDIC conversion for standard input, standard output, and standard error. The **qsh_out** command only does ASCII/EBCDIC conversion for standard output and standard error. ⚡

You need to set ILE environment variable **QIBM_USE_DESCRIPTOR_STDIO** to Y or I (so that OS/400 PASE runtime and ILE C runtime use descriptor standard I/O) to avoid unpredictable results. This is done by default in the OS/400 jobs that program **QP2TERM** uses to run OS/400 PASE shells and utilities.

See the following topics for related information:

Examples

➤ You should use **qsh_out** (instead of **qsh** or **qsh_inout**) when the QShell command does not read from standard input to avoid unintended repositioning of the input stream. This example uses **qsh_out** to avoid repositioning the stream processed by the **read** command, and simply echos the contents of file "myinput" to standard output:

```
while read ; do
    qsh_out -c "echo $REPLY"
done < myinput
```

This example uses the QShell **cat** command to convert text in an OS/400 source database file to the (ASCII) OS/400 PASE CCSID and store the result in a stream file named `ascii_sqlcli.h`. This takes advantage of support in the QShell utility to insert linend characters in the stream that are not added if the OS/400 PASE **cat** command is used:

```
qsh_out -c 'cat /qsys.lib/qsysinc.lib/h.file/sqlcli.mbr' > ascii_sqlcli.h
```

The system provides an OS/400 PASE **getjobid** command using symbolic link `/QOpenSys/usr/bin/getjobid` -> `qsh_out` to run the QShell **getjobid** command. This example shows two ways to run the QShell utility to determine the name of the OS/400 job running the OS/400 PASE shell. The first example is more efficient because it avoids running QShell interpreter. Variable `$$` is expanded by the OS/400 PASE shell (to the process identifier of the shell), and the QShell **getjobid** command writes a line to standard output:

```
getjobid $$

qsh_out -c "/usr/bin/getjobid $$"
```

Top | OS/400 PASE Shells and Utilities



Printed in U.S.A.