iSeries

OS/400 PASE shells and utilities

Version 5 Release 3

iSeries

OS/400 PASE shells and utilities

Version 5 Release 3

Note

Before using this information and the product it supports, be sure to read the information in “Notices,” on page 15.

Fourth Edition (August 2005)

This edition applies to version 5, release 3, modification 0 of Operating System/400 (product number 5722-SS1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CISC models.

Contents

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 V5R3” on page 2 and how you can “Print this topic” on page 4.

- “OS/400 PASE commands” on page 4
- “system - Run a CL command for OS/400 PASE” on page 11
- “qsh, qsh_inout, qsh_out - Run a QShell command for OS/400 PASE” on page 13

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” on page 4 for information about these OS/400 PASE utilities:

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

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

What's new in V5R3

Here are some of the significant enhancements and changes for OS/400(R) PASE in V5R3M0:

- OS/400 PASE for V5R3M0 is derived from AIX(R) 5.2 (versus AIX 5.1 for OS/400 PASE V5R1M0)
- The following compiler products announced support to run on OS/400 PASE (at either V5R2M0 or V5R3M0) after OS/400 V5R2 was announced:
  - IBM(R) VisualAge(R) C++ Professional for AIX, Version 6.0
  - IBM C for AIX, Version 6.0
  - IBM XL Fortran for AIX, Version 8.1.1
- OS/400 option 13 (System Openness Includes) also added support (in V5R2 and V5R3) for header files needed to compile OS/400 PASE C and C++ programs.
- The following utilities are new or changed:
  - df (Reports information about space on file systems)
  - idlj (Run the QShell `idlj` command, to run the IDL-to-Java(TM) compiler)
  - orbd (Run the QShell `orbd` command, to run the Java Object Request Broker Daemon)
  - servertool (Run the QShell `servertool` command, to run the Java IDL Server Tool)
  - Changes to the OS/400 QP0ZCALL API and CL CALL command increase the number of arguments that can be passed to an ILE or OPM utility program invoked by OS/400 PASE shell scripts qsh, qsh_inout, or qsh_out
- The following libraries were added:

libnsl.a	Transport Independent Remote Procedure Call (TI-RPC)
libtli.a	Transport Library Interface
libxti.a	X/OPEN(TM) Transport Library Interface

- More specific error messages are sent from the Qp2RunPase API and the QP2SHELL and QP2TERM programs when the arguments are not valid (for example, path not found, not authorized, not a valid executable object)
- The OS/400 PASE loader implementation is derived from the 64-bit AIX kernel, providing improved performance and adding support for functions such as unloading 64-bit shared libraries. Runtime interfaces (such as sysconf and _system_configuration) are updated to reflect a 64-bit kernel
- New or changed OS/400 PASE runtime functions:
  - _GETTS64 (Get 64-bit Teraspace Address for OS/400 PASE address)
  - _GETTS64_SPP (Get 64-bit Teraspace Address from Space Pointer)
  - _GETTS64M (Get Multiple 64-bit Teraspace Addressses)
  - _ILECALLX add ARG_MEMTS64 and ARG_TS64PTR support
  - _ILELOADX (64-bit ILE activation mark support)
  - _ILESYMX (64-bit ILE activation mark support)
  - _PGMCALL support for more arguments and ASCII-to-EBCDIC string conversion support
  - _SETSPP_TS64 (Set Space Pointer to 64-bit Teraspace Address)
  - _SETSPPM (Set Multiple Space Pointers)
  - fork and f_fork support for PASE_FORK_JOBNAME environment variable
  - fork400 and f_fork400 (specify job name and resource identifier)
  - getgrent (get group entry)
  - getrpid (get real process identifier)
  - getpwent (get password/user entry)
  - mntctl (to retrieve attributes of mounted file system)
  - Qp2setenv_ile (to set ILE environment variables)
- New or changed (ILE) APIs for OS/400 PASE:
  - QP2SHELL, QP2SHELL2, and QP2TERM now default the PASE_TZ environment variable to match the OS/400 job TIMZON attribute (time zone support)
  - QP2SHELL, QP2SHELL2, and Qp2RunPase now check ILE environment variable QIBM_PASE_FLUSH_STDIO to determine whether to flush standard output (stdout and stderr) when *not* using integrated file system descriptors (when QIBM_USE_DESCRIPTOR_STDIO is unset)
  - Support on the Qp2RunPase API to fill a specific named symbol with arbitrary data) was dropped. The second argument for Qp2RunPase (symbolName) now **must** be a null pointer. OS/400 PASE programs can call ILE and OPM code (using _ILECALL and _PGMCALL) with by-address arguments to retrieve inputs that cannot be expressed as null-terminated arguments or environment variable character strings.
- No job start messages are written in the job log or QHST for fork jobs, and a job completion message is produced only if a job ends abnormally
- OS/400 PASE runtime now relies on the system clock to keep track of coordinated universal time (UTC), so time zone changes are reflected immediately (instead of being delayed until the Qp2RunPase API first runs after the change).
- OS/400 PASE now supports UTF-16 encoding for internal system support interfaces (for example, file systems) that were previously restricted to UCS-2.
- Support for *zombie* processes was added (per the UNIX(R) standard).
- New locales were added.

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^R PASE Shells and Utilities information (about 243 KB).

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^R 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^R 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 Managment 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 utilties do *no* automatic Coded Character Set Identifier (CCSID) conversion of stream file data. The OS/400 PASE utilities "system - Run a CL command for OS/400 PASE" on page 11, 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, and so on.). 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 B C D E F G H I J K L M N O P Q R S T U V W X Y Z

	A
admin	Create and control Source Code Control System (SCCS) files
aixterm	Initialize an Enhanced X Window System^{R} terminal emulator
alias	Define or display aliases
appletviewer	Run the QShell **appletviewer** command, to run Java applets without a web browser
apply	Apply a command to a set of parameters
ar	Maintain the indexed libraries used by the linkage editor
as	Run the assembler
attr	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
	B
banner	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
	C
cat	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
csh	Invoke the C shell
csplit	Split files by context
cut	Write out selected bytes, characters, or fields from each line of a file
	D
date	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
df	Reports information about space on file systems
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
	F
false	Return a nonzero exit value (false)
fc	Process the command history list
fg	Run jobs in the foreground
fgrep	Generate the figure list in a format supported by the build process
file	Determine file type
find	Find files with a matching expression
fold	Fold long lines for finite-width output device
	G
gencat	Create and modify a message catalog
get	Create a specified version of a SCCS file
getconf	Write system configuration varible values to standard output
getjobid	Run the QShell **getjobid** command to determine the OS/400 job name for a process identifier
getopt	Parse command line flags and parameters

getopts	Process command-line arguments and check for valid options
grep	Search a file for a pattern
	H
hash	Remember or report command path names
head	Display the first few lines or bytes of a file or files
hostname	Set or display the name of the current host system
	I
iconv	Convert the encoding of characters from one code page encoding scheme to another
id	Display the system identifications of a specified user
idlj	Run the QShell idlj command, to run the IDL-to-Java compiler
indent	Reformat a C language program
install	Install a command
ipcs	Run the QShell **ipcs** command to display interprocess communications objects
ipcrm	Run the QShell **ipcrm** command to remove interprocess communications objects
	J
jar	Run the QShell **jar** command, to archive Java files
jarsigner	Run the QShell **jarsigner** command, to sign or verify the signature of a Java archive
java	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
orbd	Run the QShell orbd command, to run the Java Object Request Broker Daemon
	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, qsh_inout, qsh_out - Run a QShell command for OS/400 PASE" on page 13	Run a QShell command
"qsh, qsh_inout, qsh_out - Run a QShell command for OS/400 PASE" on page 13	Run a QShell command
"qsh, qsh_inout, qsh_out - Run a QShell command for OS/400 PASE" on page 13	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	Pipe output data from the **mkcatdefs** command to the **gencat** command
	S
sact	Display current SCCS file-editing status
serialver	Run the QShell **serialver** command, to return the version number for Java classes
sccs	Administration program for SCCS commands
sccsdiff	Compare two versions of a SCCS file
sdiff	Compare two files and display the differences in a side-by-side format
sed	Provide a stream editor
servertool	Run the QShell servertool command, to run the Java IDL Server Tool
setmaps	Set terminal maps or code set maps
setccsid	Run the QShell **setccsid** command to set the CCSID for an Integrated File System object
sh	Invoke the default (Korn) shell
size	Display the section sizes of the Extended Common Object File Format (XCOFF) object files
sleep	Suspend execution for an interval
sort	Sort files, merge files that are already sorted, and check files to determine if they have been sorted
split	Split a file into pieces
strings	Find the printable strings in an object or binary file
strip	Reduce the size of an Extended Common Object File Format (XCOFF) object file by removing information used by the binder and symbolic debug program
stty	Set, reset, and report workstation operating parameters
sum	Display the checksum and block count of a file
syslogd	Log system messages
"system - Run a CL command for OS/400 PASE" on page 11	Run a CL command
sysval	Run the QShell **sysval** command to display an OS/400 system value or network attribute
	T
tab	Change spaces into tabs
tabs	Set tab stops on a terminal
tail	Write a file to standard output, beginning at a specified point

tar	Manipulate archives
tee	Display the output of a program and copy it into a file
test	Evaluate conditional expressions
tic	Translate the terminfo description files from source to compiled format
time	Print the time of the execution of a command
tnameserv	Run the QShell **tnameserv** command, to provide access to the Java naming service
touch	Update the access and modification times of a file
tput	Query the **terminfo** database for terminal-dependent information
tr	Translate characters
trace	Record selected system events
trbsd	Translate characters (BSD version)
trcoff	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
	U
ulimit	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
	V
val	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
	W
wait	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 Window System on the current primary system
xlsfonts	Display the font list for X Window System

xmodmap	Modify keymaps in the X Server
xset	Set options for your X Window System environment
xterm	Provide a terminal emulator for the X Window System
xwd	Dump the image of an Enhanced X Window System window
xwud	Retrieve and display the dumped image of an Enhanced X Window System window
	Y
yacc	Generate an LALR(1) parsing program from input consisting of a context-free grammar specification
yes	Output an affirmative response repetitively
	Z
zcat	Expand a compressed file to standard output

system - Run a CL command for OS/400 PASE

Syntax

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

Description

The OS/400^{R} 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 unpredicatable 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 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 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.

-O 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.

» **-p** This option is ignored. The OS/400 PASE **system** utility always handles only messages sent to the program that runs the CL command (the way the QShell **system** utility works with option **-p**). «

-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

» If any exception message is sent by the CL command analyzer or the command processing program, the **system** utility returns an exit status of 255. Error messages always appear in the job log of the OS/400 job that ran the command, and may also be sent to standard output or standard error (unless option **-q** is specified).

If CL command processing did not send an exception message, the **system** utility returns the exit status set by whatever program the the CL comand called, or zero if that program did not set exit status. «

Examples

This example shows three ways to run the **CRTDTAARA** CL command with the same parameter values. Options **-bOE** 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 -bOE "crtdtaara mydata *char text('Output queue text')"
or
      system -bOE crtdtaara mydata "*char text('Output queue text')"
or
      system -BOE crtdtaara 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')"
```

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^{R} 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” on page 11

- **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 unpredicatable 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 $$"
```

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

```
IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY  10594-1785
U.S.A.
```

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

```
IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan
```

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:
e (logo)
IBM
iSeries
Operating System/400
OS/400

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for downloading and printing publications

Permissions for the use of the publications you have selected for download are granted subject to the following terms and conditions and your indication of acceptance thereof.

Personal Use: You may reproduce these Publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these Publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these Publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these Publications, or reproduce, distribute or display these Publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the Publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the Publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations. IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

All material copyrighted by IBM Corporation.

By downloading or printing a publication from this site, you have indicated your agreement with these terms and conditions.

Printed in USA