

Introduction to Linux for QSS/OASIS System Administrators and System Operators

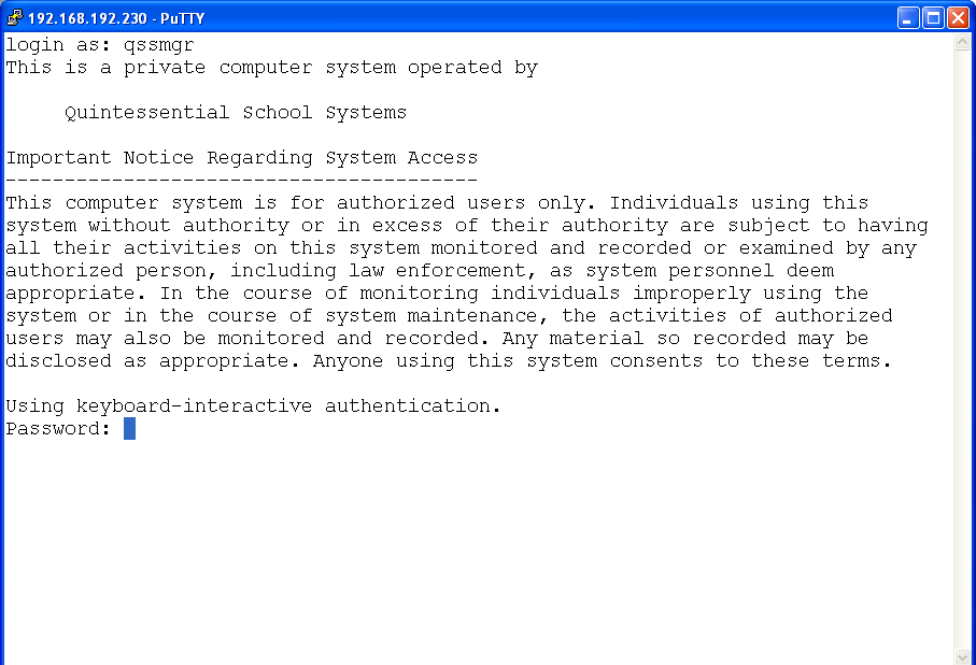
A **QSS** Webinar

September 25, 2012

10:00 AM – 12:00 PM

Logging on

- Type user and press enter after
logon as:
- Type password
and press enter
after
Password:



A screenshot of a PuTTY terminal window titled "192.168.192.230 - PuTTY". The terminal displays the following text:

```
login as: qssmgr
This is a private computer system operated by

    Quintessential School Systems

Important Notice Regarding System Access
-----
This computer system is for authorized users only. Individuals using this
system without authority or in excess of their authority are subject to having
all their activities on this system monitored and recorded or examined by any
authorized person, including law enforcement, as system personnel deem
appropriate. In the course of monitoring individuals improperly using the
system or in the course of system maintenance, the activities of authorized
users may also be monitored and recorded. Any material so recorded may be
disclosed as appropriate. Anyone using this system consents to these terms.

Using keyboard-interactive authentication.
Password: █
```

System Users

- root – System administrator
 - ‘Best Practice’ is to use this user only when needed to prevent making a mistake while learning
 - Most commands do not prompt for confirmation
 - There is no undo!
 - root can do anything!

QSS Users

- qssmgr – QSS System Administrator
 - Used to install software and configure the system
- qssopr – QSS System operator
 - Used for accounts payable and payroll processing
- qssdba – QSS Database administrator

Commands

- Case-sensitive
- Some commands are built into the shell
 - bash – Bourne Again Shell – Interactive use
 - sh – Bourne Shell – Scripts
- Found based on path
 - Omitted path uses \$PATH variable
 - Relative path uses ./ (current directory) or ../ (parent directory)
 - Absolute path is a fully qualified name

Information about a file

- file shows you the file's type
- ls -l shows you a file's attributes

```
qssmgr@smv-se9-ap:~> ls -l hello_world
```

```
-rwxr-xr-x  1 qssmgr qssmgr 27 2008-10-02 22:19 hello_world
```

- Permissions `-rwxr-xr-x`
- owner and group `qssmgr qssmgr`
- Size `27`
- Date and time file was modified `2008-10-02 22:19`

Permissions

- `-rwxr-x---`
 - 1st character is 'd' if a directory
 - Next 3 characters 'rwx' show read, write, and execute permission for owner of file
 - Next 3 characters 'r-x' show read, write and execute permission for group
 - Last 3 character '---' show read, write and execute permission for everyone else.
 - If a value is '-' instead of 'r', 'w', or 'x' then that permission is not granted.

QSS Version

- Replacement for QSS/OASIS Version H getver command or MPE's version commands
- Use -h switch for list of options
 - qv -h
- Examples
 - qv \$QSS_ROOT/lib/libqapp.so
 - Similar to 'version qsslib.library'
 - qv \$QSS_ROOT/\$QSS_BINSCH/gl0100
 - Similar to 'version gl0100.!ppsch.!pact'

Commands to view a file

- cat – display a file
- more – page forward thru file
- less – page forward and backward thru file.
- head – show the 1st n lines (default is 10)
- tail – show the last n lines (default is 10)
- grep – search a file for a string

Commands to manage files

- touch – create a empty file or change timestamp on a file
- cp – copy a file
- mv – rename a file
- rm – remove a file
- ls – list files
- find – find files
- vi – edit a file

Using vi to edit files

- To edit a file with vi
 - vi filename
- vi's modes
 - Normal mode
 - One or two keystroke commands to navigate and make changes
 - Keystrokes are not echoed in most cases
 - Insert mode
 - Press 'i' to enter insert mode
 - Can use arrow keys, page up, page down, to move around, delete key to delete characters, what you type is inserted, etc. Much like Notepad.
 - Press Esc to return to normal mode
- To exit vi (must be in Normal mode)
 - :wq to save file and exit
 - :q to exit if you didn't make any change
 - :q! to exit without saving
- To learn vi use the command: vimtutor

QSS Directories

- /etc/opt/qss – QSS configuration files
- /opt/qss – QSS software
- /var/opt/qss – QSS data files

QSS Configuration files in `/etc/opt/qss`

- `jobsched/` – configuration directory for QSS Job System
 - `jobsched/datemask` – contains accepted date masks for submitting jobs with the `-at` option
 - `jobsched/jobid` – Job number odometer. Modify to reset your job numbers.
 - `jobsched/JobSec.cfg` – lists users allowed to submit jobs. Maintain with `JobSecCfg` application
 - `Jobsched/JobSvr.cfg` – configures port number, queues, limits, file locations, etc.

QSS Configuration files in `/etc/opt/qss`

- `login/` – login directory
 - `login/copyright` – QSS copyright statement
 - `login/message` – login message
 - `login/netcobolrt.sh` – script to setup path to NetCOBOL runtime.
 - `login/PATH.qss` – Added to `$PATH` of QSS user. Modify this to change path for all QSS users as directed by QSS.
 - `login/qss_check` – this is ‘hooked’ into system login file.
 - `login/qssenv.sh` – this sets the default environment variables.
 - `login/qsslocal.sh` – this sets customer specific variables and commands. Modify this file to set your values.
 - `login/qss_login` – this is the main login script.
 - `qssrun.sh` – this is the list of commands run when user logs in.
 - `user/` - directory for user specific settings.

QSS Configuration files in /etc/opt/qss

- qat.d/ – QSS automated tasks directory
 - logfiles - settings to restrict size of log files
 - releases - settings to remove older release / archive files
 - stdlists - settings for how to archive and delete stdlists.
- qcc_ctl.cfg – configuration file for QCC servers
- qdbi/ – database configuration.

QSS Software directories in /opt/qss

- archive – archived versions of files
- bin – essential QSS binaries (utilities)
- DataDirect – DataDirect libraries for accessing database
- finance – Directory for QSS/OASIS software
- install – skeleton and initial version of installation files.
- jobsched/bin – binaries for QSS Job System
- lib – QSS libraries
- qat – QSS Automated Tasks modules
- qcc/bin – binaries for QCC servers
- sbin – binaries for system tasks done by qssmgr

QSS Data Directories in `/var/opt/qss`

- `cmd/` - command files, scripts, for any qss user
- `cmd/mgr/` - command files, scripts, for qssmgr
- `cmd/opr/` - command files, scripts, for qssopr
- `data/` - data files
- `data/qcc` - data files for QCC servers
- `dist/####/` - district specific files
- `fmt/` - printer format files
- `history/subdir/` - history files separated into sub directories such as 1099 - 1099 files, apy - accounts payable, pay - payroll, ret - retirement, w2 - w2 files, etc
- `jcl/` - site specific jcl files
- `log/` - log files

QSS Data Directories in `/var/opt/qss`

- `parms/` - parameter files
- `pdl/` - personnel downloader files and district specific sub directories
- `pfiles/` - print file directory
- `pox/` - purchase order print driver files
- `qcc/` - execution directory for QCC servers
- `release/` - release directory
- `tmp/` - QSS temporary files
- `w21099/year/` - w2 and 1099 files in year specific sub directories
- `web/` - web template files for QCC servers
- `xfr/` - xfr directory used by QSSXFR

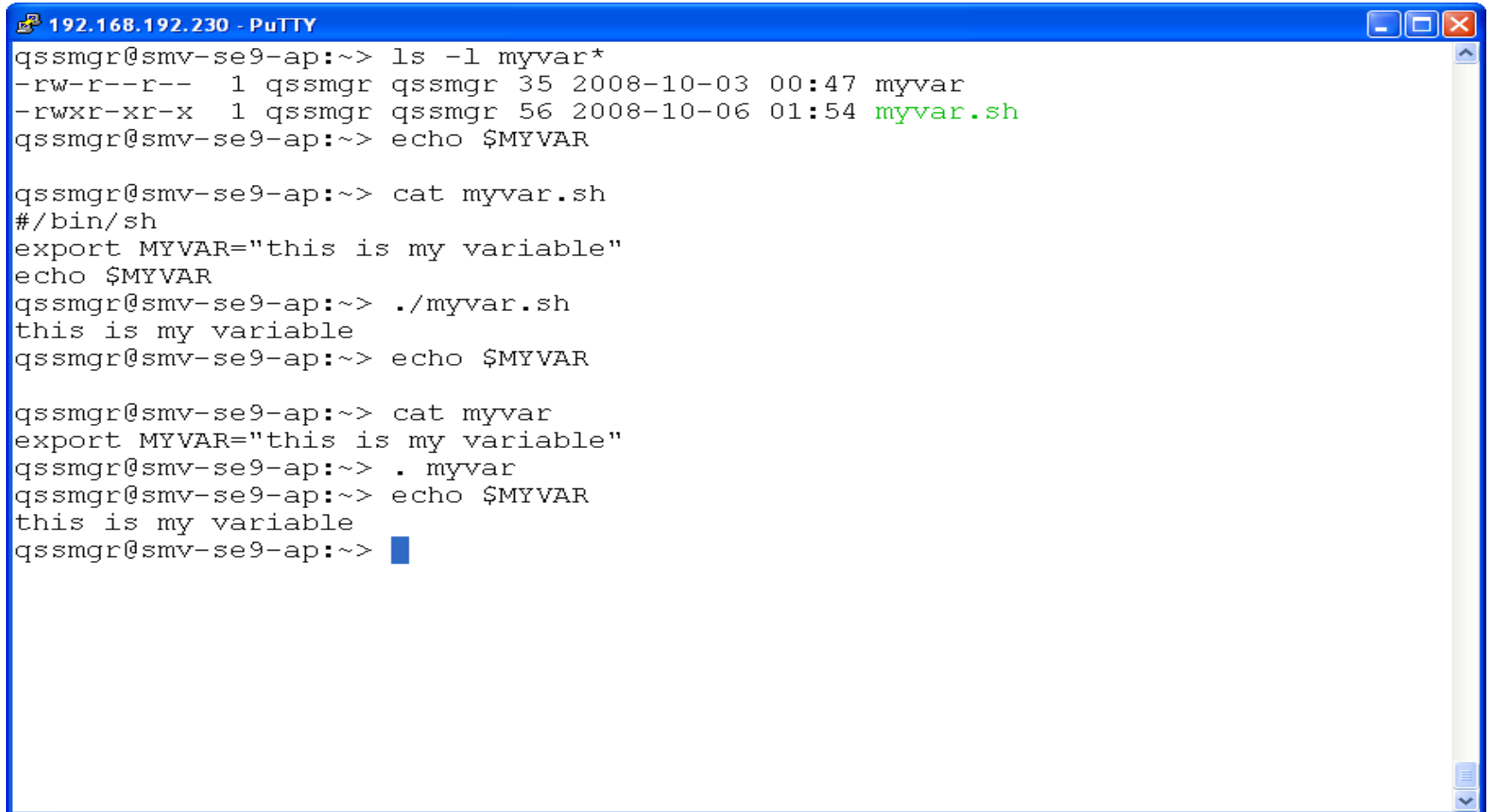
QSS Variables

- `$QSS_ROOT` – set to main software directory, usually `/opt/qss`
- `$QSS_CFG` – set to configuration directory, usually `/etc/opt/qss`
- `$QSS_DATA` – set to data directory, usually `/var/opt/qss`
- Most other QSS variables can be found by
 - `printenv | sort | grep Q`
- Set by QSS login scripts
 - `qss_login`, `qssenv.sh`, `qsslocal.sh`, and user specific login script.

Sourcing a file

- Sourcing a file executes the commands in that file in the current shell.
 - Syntax
 - `source filename`
 - `. filename`
 - Different than executing a script which cannot pass variables back to parent.
 - Only method to export variables from a file.

Exporting a variable in script vs. source example



```
192.168.192.230 - PuTTY
qssmgr@smv-se9-ap:~> ls -l myvar*
-rw-r--r--  1 qssmgr qssmgr 35 2008-10-03 00:47 myvar
-rwxr-xr-x  1 qssmgr qssmgr 56 2008-10-06 01:54 myvar.sh
qssmgr@smv-se9-ap:~> echo $MYVAR

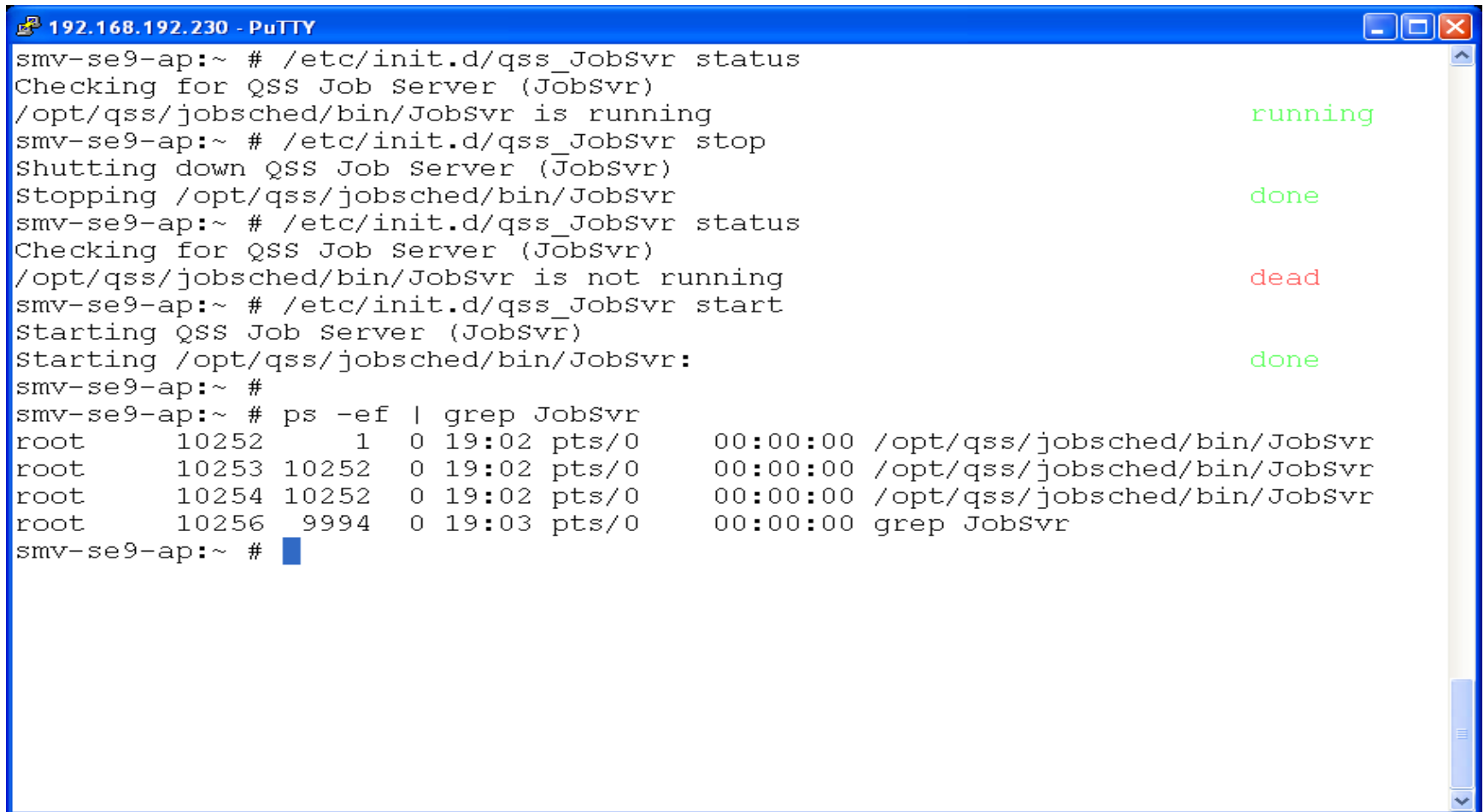
qssmgr@smv-se9-ap:~> cat myvar.sh
#!/bin/sh
export MYVAR="this is my variable"
echo $MYVAR
qssmgr@smv-se9-ap:~> ./myvar.sh
this is my variable
qssmgr@smv-se9-ap:~> echo $MYVAR

qssmgr@smv-se9-ap:~> cat myvar
export MYVAR="this is my variable"
qssmgr@smv-se9-ap:~> . myvar
qssmgr@smv-se9-ap:~> echo $MYVAR
this is my variable
qssmgr@smv-se9-ap:~> 
```

Starting/Stopping QSS Job System

- Status (from 'qssmgr' or 'root')
 - /etc/init.d/qss_JobSvr status
- Starting (from 'root' user)
 - /etc/init.d/qss_JobSvr start
 - Starts every 'instance' defined in /etc/sysconfig/qss_JobSvr
 - Can be set to start automatically at boot.
- Stopping (from 'root' user)
 - /etc/init.d/qss_JobSvr stop
 - Stops every 'instance' defined in /etc/sysconfig/qss_JobSvr
 - Can be set to stop automatically at shutdown
- Stopping (from 'qssmgr' user)
 - /opt/qss/jobsched/bin/JobCli
 - Use 'connect' and 'stop' commands.

qss_JobSvr examples

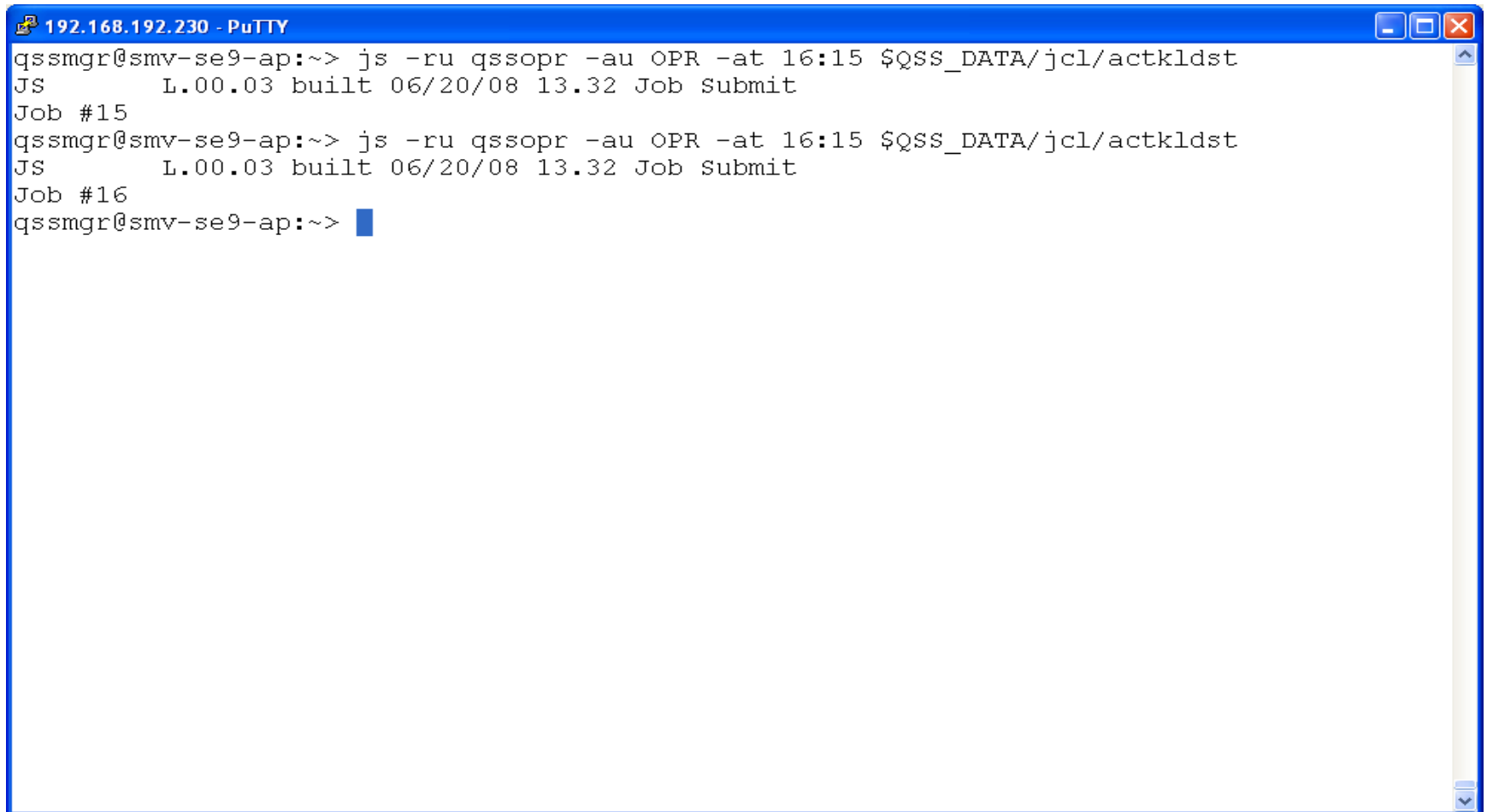


```
192.168.192.230 - PuTTY
smv-se9-ap:~ # /etc/init.d/qss_JobSvr status
Checking for QSS Job Server (JobSvr)
/opt/qss/jobsched/bin/JobSvr is running running
smv-se9-ap:~ # /etc/init.d/qss_JobSvr stop
Shutting down QSS Job Server (JobSvr)
Stopping /opt/qss/jobsched/bin/JobSvr done
smv-se9-ap:~ # /etc/init.d/qss_JobSvr status
Checking for QSS Job Server (JobSvr)
/opt/qss/jobsched/bin/JobSvr is not running dead
smv-se9-ap:~ # /etc/init.d/qss_JobSvr start
Starting QSS Job Server (JobSvr)
Starting /opt/qss/jobsched/bin/JobSvr: done
smv-se9-ap:~ #
smv-se9-ap:~ # ps -ef | grep JobSvr
root      10252      1    0 19:02 pts/0    00:00:00 /opt/qss/jobsched/bin/JobSvr
root      10253 10252    0 19:02 pts/0    00:00:00 /opt/qss/jobsched/bin/JobSvr
root      10254 10252    0 19:02 pts/0    00:00:00 /opt/qss/jobsched/bin/JobSvr
root      10256  9994    0 19:03 pts/0    00:00:00 grep JobSvr
smv-se9-ap:~ #
```

QSS Job System add-on Commands

- sj – show job
 - Use sj -h to see syntax
- dj <jobnum> – delete job
- js – job submit
 - Use js -h to see syntax
- lsl [<user>]– list stdlists
 - use lsl -l for long version
- prjob <jobnum> - print stdlist
- dljob <jobnum> - delete stdlist (in development)

js example to schedule jobs



A screenshot of a PuTTY terminal window titled "192.168.192.230 - PuTTY". The terminal shows a user named "qssmgr" at a host named "smv-se9-ap" with a tilde (~) as the current directory. The user enters the command "js -ru qssopr -au OPR -at 16:15 \$QSS_DATA/jcl/actkldst". The output shows "JS L.00.03 built 06/20/08 13.32 Job Submit" followed by "Job #15". The user then enters the same command again, and the output shows "JS L.00.03 built 06/20/08 13.32 Job Submit" followed by "Job #16". The prompt "qssmgr@smv-se9-ap:~>" is followed by a blue cursor bar.

```
qssmgr@smv-se9-ap:~> js -ru qssopr -au OPR -at 16:15 $QSS_DATA/jcl/actkldst
JS      L.00.03 built 06/20/08 13.32 Job Submit
Job #15
qssmgr@smv-se9-ap:~> js -ru qssopr -au OPR -at 16:15 $QSS_DATA/jcl/actkldst
JS      L.00.03 built 06/20/08 13.32 Job Submit
Job #16
qssmgr@smv-se9-ap:~> █
```

sj example

```
192.168.192.230 - PuTTY
qssmgr@smv-se9-ap:~> sj

-----
QSS Job System 0.0.0 - Client      Mon Oct  6 19:36:26 2008
(c) 2003 QSS, RAC Consulting.      All Rights Reserved
-----

connecting to Job Server on 127.0.0.1:20001
connected...
===== 0 running jobs =====
===== 2 waiting jobs =====
id      name          queue      user          appuser      pri state  host
      intro-time/title      start-at-time
15      ST011000.tmp    default    qssopr        OPR          5   wait   localhost
      Mon Oct  6 19:34:47 2008  Tue Oct  7 16:15:00 2008
      actkldst
16      ST011003.tmp    default    qssopr        OPR          5   wait   localhost
      Mon Oct  6 19:34:58 2008  Tue Oct  7 16:15:00 2008
      actkldst

qssmgr@smv-se9-ap:~> █
```

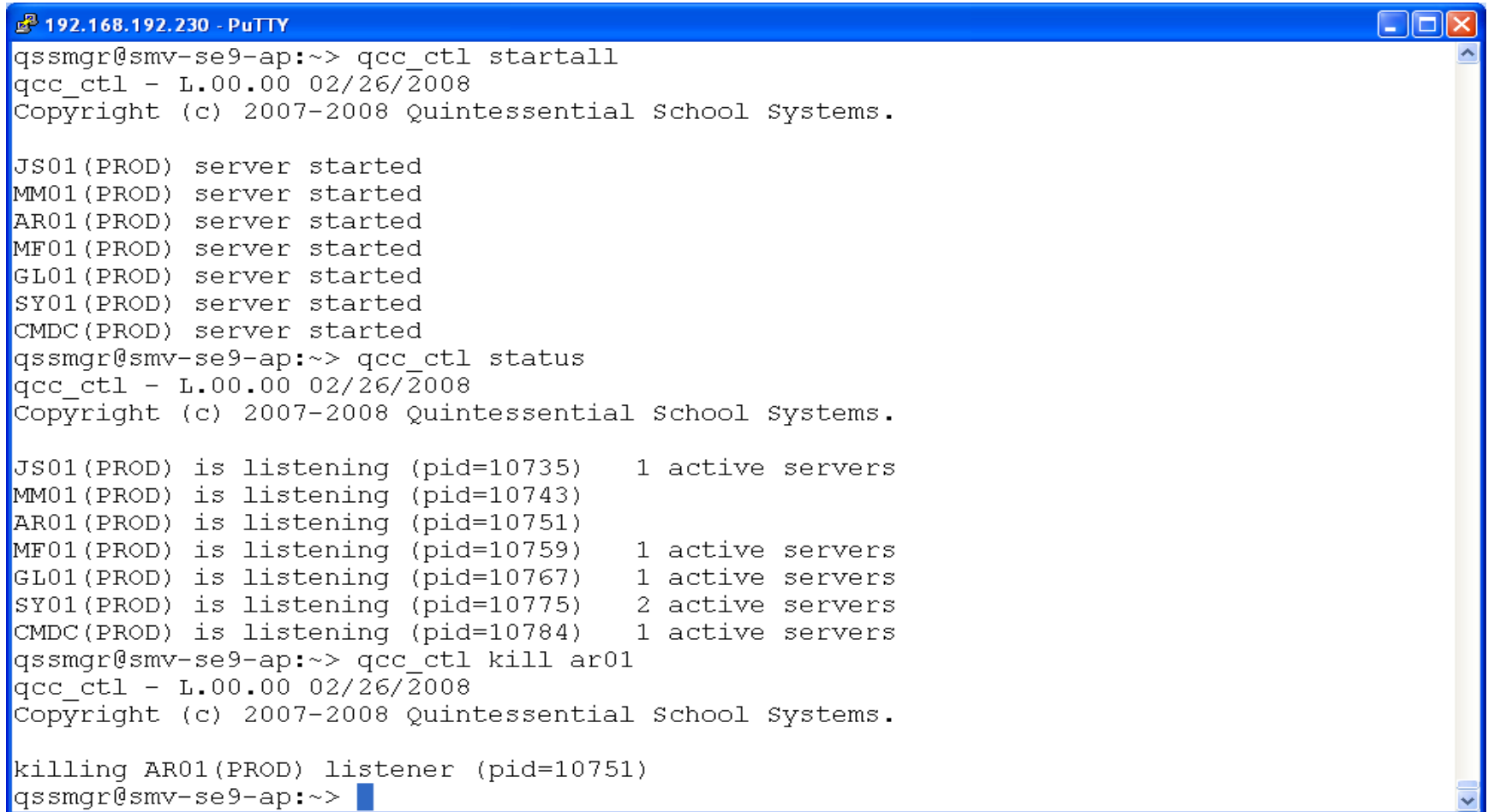
Starting/Stopping QCC Servers from 'root'

- QCC servers status
 - /etc/init.d/qss_qcc_ctl status
 - Status of servers for all 'instances'
- Starting QCC servers
 - /etc/init.d/qss_qcc_ctl start
 - Starts servers for all 'instances'
 - Can be set to start automatically at boot.
- Stopping QCC servers
 - /etc/init.d/qss_qcc_ctl stop
 - Stops servers for all 'instances'
 - Can be set to stop automatically at shutdown.

Starting/Stopping QCC Servers from 'qssmgr'

- qcc_ctl manages servers for the current 'instance'
 - Status for all servers
 - qcc_ctl status
 - Status for a single server
 - qcc_ctl status gl01
 - Start all servers
 - qcc_ctl startall
 - Start a single server
 - qcc_ctl start gl01
 - Stop all servers
 - qcc_ctl killall
 - Stop a single server
 - qcc_ctl kill gl01

qcc_ctl examples



```
192.168.192.230 - PuTTY
qssmgr@smv-se9-ap:~> qcc_ctl startall
qcc_ctl - L.00.00 02/26/2008
Copyright (c) 2007-2008 Quintessential School Systems.

JS01(PROD) server started
MM01(PROD) server started
AR01(PROD) server started
MF01(PROD) server started
GL01(PROD) server started
SY01(PROD) server started
CMD01(PROD) server started
qssmgr@smv-se9-ap:~> qcc_ctl status
qcc_ctl - L.00.00 02/26/2008
Copyright (c) 2007-2008 Quintessential School Systems.

JS01(PROD) is listening (pid=10735)    1 active servers
MM01(PROD) is listening (pid=10743)
AR01(PROD) is listening (pid=10751)
MF01(PROD) is listening (pid=10759)    1 active servers
GL01(PROD) is listening (pid=10767)    1 active servers
SY01(PROD) is listening (pid=10775)    2 active servers
CMD01(PROD) is listening (pid=10784)    1 active servers
qssmgr@smv-se9-ap:~> qcc_ctl kill ar01
qcc_ctl - L.00.00 02/26/2008
Copyright (c) 2007-2008 Quintessential School Systems.

killing AR01(PROD) listener (pid=10751)
qssmgr@smv-se9-ap:~> 
```

Managing Nightly Processes

- Processes include ACTKLD, BALFIX, VENKLD
- JCL files are customizable, usually require end of year modification, and reside in \$QSS_DATA/jcl
- Schedule file to specify when
 - \$QSS_DATA/data/sched_actkld
- Command file to submit job
 - sub_actkld submit reoccurring scheduled job based on schedule file
 - sub_actkld once submit immediately
 - sub_actkld 18:00 submit reoccurring at 6:00 pm

sample setup of Accounts Payable Processing

- Logon as 'qssopr'
- Scripts are in \$QSS_DATA/cmd/opr
 - apyextract – extracts data to file in /home/qssopr
 - apywarr – prints warrants
 - apyreport – submits \$QSS_DATA/jcl/apyrptst job to print reports
 - apyupdate - submits \$QSS_DATA/jcl/apyrptst job to update database
 - apysave – saves production files from /home/qssopr to history directory \$QSS_DATA/history/apy
- Process is customized for each client

sample setup of Payroll Processing

- Logon as 'qssopr'
- Scripts are in \$QSS_DATA/cmd/opr
 - payextract – extracts data to file in /home/qssopr
 - paywarr – prints warrants
 - payreport – submits \$QSS_DATA/jcl/payrptst job to print reports
 - payupdate - submits \$QSS_DATA/jcl/payupdst job to update database
 - paysave – saves production files from /home/qssopr to history directory \$QSS_DATA/history/pay
- Process is customized for each client

QSS/OASIS Releases

- Software Releases - keeps program, jcl, data files current
 - GS36-0, GS36-1, ...
 - Current schedule includes new release every Thursday
 - Automated process that invokes install_qss script
 - Archive current files in \$QSS_ROOT/archive
- Environment Releases - keep the QSS directory structure, login files, etc current
 - Example EN34-0
 - Less common, but usually accompanies GSxx-0 release
 - Automated process that invokes build_qss script
- Sysctl data update - updates sysctl tables with latest menus, modules, tasks, etc.
 - sysctl_upate-1.36
 - Usually accompanies GSxx-0 release
 - Delivered via email, manual install process

autoupdate_qss

- Checks for new GS and EN releases
- Downloads and installs release automatically
- Releases are currently password protected
- Can be done either 'after hours', or while 'in use'
 - 'in use' method - safely replaces files being used allowing current process to finish using the previous file. May have 'after hours' step to restart QCC servers.

autoupdate_qss help example

```
> autoupdate_qss -h
```

```
autoupdate_qss (QSS Auto Update) - L.01.02 03/21/2012
```

```
Copyright (c) 2009-2012 Quintessential School Systems.
```

```
Syntax
```

```
autoupdate_qss [-l] [-r rel_id] [-v]
```

```
autoupdate_qss [-i | -n] [-r rel_id] [-v]
```

```
Options
```

```
-h          Help.
-i          Interactive install. Default option is an automated install.
           Cannot use this with -n.
-l          List the current releases and indicate status of release.
           Does not install any release or get the latest list of releases.
-n          Do not install release after downloading and extracting.
           Default option is an automated install. Cannot use this with -i
-r rel_id  Work with a specific release instead of just the all releases.
-v          Verbose mode, The script already shows what it is doing but this
           adds some additional detail.
```

```
Examples
```

```
To download, extract, and install all new releases
```

```
autoupdate_qss
```

```
To see what releases are currently known and check if they are installed or not
```

```
autoupdate_qss -l
```

```
To download and extract all new releases but not install. Install must be done
manually at a later time.
```

```
autoupdate_qss -n
```

```
To download, extract, and install a specific releases and have an interactive install.
```

```
autoupdate_qss -i -r GS22-0
```