Got extra dev drop in office, 3158 #2 = IP 10.252.175.54

Downloaded Getting Started, Product Notes, Installation Guide, Service Manual.

Installed additional hard drives (2) and CD/DVD drive. See instructions in service manual.

Got cable for remote admin.

Got server in office.

Connected serial console cable with DB9 adapter to PC and serial mgt port on server.

Set up hypertrm.exe as console: 9600baud, 8bit, parity=None, Stop Bits=1, Flow Control None

Connected WUCON patch cable to 0 slot

Connected server to power. This brought up ALOM prompt.

Set sc password as 7098R2^!

sc>**poweron** This powers on the server and boots it.

sc>**console** This switches you over to the server console (#.) will take you back to ALOM.

agreement appeared.. entered q to skip, accept to accept

Select a language: **0** English

Select a locale: **0** English

What kind of terminal are you using? **6** PC Console (server works for a while; blue/green screen comes up) Note: For selection boxes, use the arrow key to move to the box you want, then the enter key to set or clear the X. For select one boxes, selecting the one you want will clear the previously selected one.

Networked? **X YES, F2**

Networked interfaces **X bge0 only, F2**

Used DHCP for bge0 **X NO, F2**

Host name for bge0 **relay1, F2**

IP address for bge0 **10.39.186.233, F2**

System part of a subnet **X YES, F2**

Netmask for bge0 **255.255.255.0, F2**

Enable IPv6 for bge0 **X NO, F2** (server works for a while. At first it has a “seconds to go” message, but then this disappears. Wait until new screen displays)

Default route selection for bge0 **X Specify one, F2**

Router IP Address for bge0 **10.39.186.254, F2**

Verify bge0 configuration:

 Networked: Yes

 Use DHCP: No

 Host name: relay2

 IP address: 10.39.186.233

 System part of a subnet: Yes

 Netmask: 255.255.255.0

 Enable IPv6: No

 Default Route: Specify one

 Router IP Address: 10.39.186.254

**F2** (server will work for a while)

Configure Kerberos **X NO, F2**

Confirm

 Configure Kerberos Security: No

**F2**

Name service **X DNS, F2**

Domain name **wucon.wustl.edu, F2**

DNS server IP addresses **10.39.232.239 10.39.46.239, F2**

Search domains **carenet.org, F2**

Confirm:

 Name service: DNS

 Domain name: erl.wustl.edu

 Server address(es): 10.252.180.50

 10. 252.180.51

 Search domain(s): wustl.edu ccirdev.mir carenet.org

**F2**

System works for a while, can’t find “relay2”, asks if you want to enter new name service information. **X NO, F2**

Time zone **X Americas, F2 X United States, F2 X Central Time, F2**

Accept default date and time **F2**

Confirm time zone information **F2**

Enter root password **7098R2^!, F2**

Do you need to override the systems default NFS version 4 domain name? **no Enter**

system reboots, complains for a while that it can’t resolve bond for sendmail.

Log in as root

**Patch system**

**Got Sun Update Registration:** **ID=moultonr@mir.wustl.edu** **PW=7066MS##**

**cd /tmp**

**cp** /usr/lib/breg/data/RegistrationProfile.properties /tmp/myreg.profile

Edited myreg.profile:

username=moultonr@mir.wustl.edu

password=7066MS##

.

.

hostname=relay2

This is a bit of a trick if your hyperterm does not handle cursor addressing, but can be done by searching for the keys, using “A” to go into insert mode at the end of the line, typing the addition, and hitting ESC.

**sconadm proxy -r /tmp/myreg.profile**

**sconadm register -a -r /tmp/myreg.profile**

**smpatch set patchpro.patchset=sfv2x5**

**smpatch analyze**

**smpatch update**

When done, **reboot** to finish last installations.

Make new groups:

**groupadd erl**

**groupadd spectrum**

**groupadd mysql**

**groupadd qc**

Added spectrum account:

**useradd –c Spectrum –d /export/home/spectrum –g erl –s /bin/csh spectrum**

Added spectrum home directory:

**cd /export**

**mkdir home**

**cd home**

**mkdir spectrum**

**chown spectrum:erl spectrum**

**passwd spectrum** set password to spectrum standard pw.

**MySQL installation @@@@@@@@@@@@@@**

Installed 5.0.41, latest production release available for Solaris 10

Added mysql account:

**useradd -c MySQL -d /export/home/mysql -g mysql mysql**

**cd /export/home**

**mkdir mysql**

**chown mysql:mysql mysql**

Downloaded mysql and its dependencies from sun freeware as root. Note that most of the server side dialog has been omitted, and that the longer transfers do take time:

**cd /tmp**

**ftp** **ftp.sunfreeware.com**

Name: **anonymous**

Password: **moultonr@mir.wustl.edu**

ftp> **bin**

ftp> **cd pub/freeware/sparc/10**

ftp> **get mysql-5.0.41-sol10-sparc-local.gz**

ftp> **get gcc-3.4.6-sol10-sparc-local.gz**

ftp> **get openssl-0.9.8e-sol10-sparc-local.gz**

ftp> **get ncurses-5.6-sol10-sparc-local.gz**

ftp> **get zlib-1.2.3-sol10-sparc-local.gz**

ftp> **get libiconv-1.11-sol10-sparc-local.gz**

ftp> **get make-3.81-sol10-sparc-local.gz** (Only needed if server used to compile software)

ftp> **quit**

Installed the packages from the tmp directory as root using pkgadd. Each package requires the following commands, with the specific package name and version substituted from the above list:

# **gunzip foo-version-sol10-sparc-local.gz**

# **pkgadd –d foo-version-sol10-sparc-local**

pkgadd will ask if you want to install all of the package; answer all, which is the default. If the /usr/local directory does not exist (the first package) pkgadd will prompt if you want to create it, answer y. In some cases, pkgadd may ask if it can change some attributes, answer y to this also. I installed the packages in this order:

**ncurses-5.6-sol10-sparc-local.gz**

**libiconv-1.11-sol10-sparc-local.gz**

**expat-1.95.5-sol10-sparc-local.gz**

**zlib-1.2.3-sol10-sparc-local.gz**

**get make-3.81-sol10-sparc-local.gz**

**gcc-3.4.6-sol10-sparc-local.gz**

**openssl-0.9.8e-sol10-sparc-local.gz**

**gdb-6.6-sol10-sparc-local.gz**

Then **rm /tmp/\*sol10-sparc-local** to delete the install files

Put the mysql install under the mysql user:

**cd /usr/local**

**chown –R mysql:mysql mysql**

Created the mysql data directory on the data disc, which is mounted to /space:

**cd /space**

**mkdir mysql**

**mkdir mysql/data**

**chown –R mysql:root mysql**

**chmod –R 700 mysql/data**

Ran the install script:

**cd /usr/local/mysql**

**bin/mysql\_install\_db --user=mysql --ldata=/space/mysql/data**

Which warned that a hostname lookup was not working so GRANT commands would have to use IP addresses. This is not a problem because the GRANT commands used in our system are never host specific.

Added /usr/local/mysql/bin to the path variable in spectrum .cshrc

Added this link, so mysql server could find hostname.

**ln -s /usr/bin/hostname /usr/local/bin/hostname**

Started mysql service for testing (Note: --datadir parameter)

**cd /usr/local/mysql**

**bin/mysqld\_safe --user=mysql --datadir=/space/mysql/data &**

Added a root password for localhost and host:

**/usr/local/mysql/bin/mysqladmin -u root password 'new pw'**

**/usr/local/mysql/bin/mysqladmin -u root -h relay3 password 'new pw'**

Solaris 10 uses Service Management Facility (SMF) to handle services. While scripts in the startup directories still work, they are considered a legacy method, and do not have all the features available with SMF. As a result, MySQL was configured to use with SMC as follows:

Created /var/svc/manifest/network/mysql.xml using vi with contents shown below:

<?xml version='1.0'?>

<!DOCTYPE service\_bundle SYSTEM '/usr/share/lib/xml/dtd/service\_bundle.dtd.1'>

<!-- Service manifest for MySQL -->

<service\_bundle type='manifest' name='mysql:mysql'>

 <service

 name='network/mysql'

 type='service'

 version='1'>

 <create\_default\_instance enabled='false' />

 <single\_instance />

 <dependency name='fs'

 grouping='require\_all'

 restart\_on='none'

 type='service'>

 <service\_fmri value='svc:/system/filesystem/local' />

 </dependency>

 <dependency name='net'

 grouping='require\_all'

 restart\_on='none'

 type='service'>

 <service\_fmri value='svc:/network/loopback' />

 </dependency>

 <exec\_method

 type='method'

 name='start'

 exec='/lib/svc/method/svc-mysql start'

 timeout\_seconds='-1'>

 <method\_context>

 <method\_credential user='mysql' group='mysql' />

 </method\_context>

 </exec\_method>

 <exec\_method

 type='method'

 name='stop'

 exec=':kill'

 timeout\_seconds='-1'>

 </exec\_method>

 <exec\_method

 type='method'

 name='restart'

 exec='/lib/svc/method/svc-mysql restart'

 timeout\_seconds='-1'>

 </exec\_method>

 </service>

</service\_bundle>



Created /lib/svc/method called svc-mysql using vi with contents shown below:

#!/usr/bin/sh

# SMF Method file for MySQL

.. /lib/svc/share/smf\_include.sh

DB\_DIR=/space/mysql/data

PIDFILE=${DB\_DIR}/`/usr/bin/uname -n`.pid

case "$1" in

 start)

 /usr/local/mysql/bin/mysqld\_safe --user=mysql --datadir=${DB\_DIR} --pid-file=${PIDFILE} > /dev/null &

 ;;

 stop)

 if [ -f ${PIDFILE} ]; then

 /usr/bin/pkill mysqld\_safe >/dev/null 2>&1

 /usr/bin/kill `cat ${PIDFILE}` > /dev/null 2>&1 && echo -n ' mysqld'

 fi

 ;;

 'restart')

 stop

 while pgrep mysqld > /dev/null

 do

 sleep 1

 done

 start

 ;;

 \*)

 echo ""

 echo "Usage: `basename $0` { start | stop | restart }"

 echo ""

 exit 64

 ;;

esac

#---EOF



Set the ownership and permissions of these two files:

**chown root:bin /lib/svc/method/svc-mysql**

**chmod 555 /lib/svc/method/svc-mysql**

**chown root:sys /var/svc/manifest/network/mysql.xml**

**chmod 444 /var/svc/manifest/network/mysql.xml**

Shut down the mysql server if it is running:

/usr/local/mysql/bin/mysqladmin –u root –pgloria shutdown

Import the service into SMC and enable it:

**svccfg import /var/svc/manifest/network/mysql.xml**

**svcadm -v enable mysql**

Instatiated databases on mysql:

**bin**

**mysql –u root –pgloria**

**\. create\_imagelog.sql**

Note: I added grant privileges for this server for testing purposes; a one time operation.

**Compile software**

Because this is the first install of Solaris 10, it is necessary to install and compile the software. Future Solaris 10 installs can simply copy the compiled software from this server. NOTE: This software is compiled for dynamic loading of common libraries, rather than static loading; The crle command noted below is needed to insure that ld can find all the needed libraries.

Obtained ctn directory tree from relay1, which was the compile server for Solaris 9.

Created pkg-config directory.

**cd**

**mkdir pkgconfig**

*This next part is done as spectrum on relay1.*

**tar cf ctn.tar ctn/**

**gzip ctn.tar**

**tar cf masters.tar masters/**

**gzip masters.tar**

**ftp 10.39.186.233**

log in as spectrum with spectrum password

**ftp> put ctn.tar.gz**

**ftp> put masters.tar.gz**

**ftp> lcd /usr/local/lib/pkgconfig**

**ftp> cd pkgconfig**

**ftp> mput \*.pc**

**ftp> quit**

*Now back to new server*

**cd**

**gunzip ctn.tar.gz**

**tar xvf ctn.tar**

**rm ctn.tar**

**gunzip masters.tar.gz**

**tar xvf masters.tar**

**rm masters.tar**

Added /usr/ccs/bin to the path in the .cshrc

Added setenv PKG\_CONFIG\_PATH "$HOME/pkgconfig" to .cshrc

Created two new files in the ./ctn/environments/solaris directory:

**make.solaris.10.mysql.noopt**



**solaris.10.mysql.noopt.env**



These had minor changes; The previous line has been retained commented out.

Then did a normal build process for the ctn software:

**cd /export/home/spectrum/ctn**

**source environments/solaris/ solaris.10.mysql.noopt.env**

The .pc files in /export/home/spectrum/pkgconfig required some modifications to properly locate the libraries, especially those in gtk, gdk, mysql, and glib. These can be examined in place.

The programs in masters/programs and masters/qc could then be compiled using the gcc commands documented in the corresponding .c file.

End of Compile Software ==========================================

set up images directory

As root…

**ln –s /space /distributor**

**mkdir /distributor/imgs**

**chown spectrum:rel /distributor/imgs**

as spectrum…

**cd /distributor/imgs**

**mkdir todelete**

**mkdir problems**

**cd**

**mkdir production**

**cd production**

**mkdir qc**

**mkdir bond**

**cd bond**

**mkdir tmp**

**mkdir bin**

**mkdir log**

using ftp, mput over home directory from relay2, set up preliminary .cshrc. Worked through scripts for any needed modifications. Mostly was deleting junk which had built up on relay2. No actual changes to scripts were needed. Make scripts executable:

**cd**

**chmod 754 \***

**chmod 744 stat state PACS**

using ftp, mput over bin directory from relay2. Deleted compiled programs, cfg, .hdr, and .msg files. Worked through scripts for any needed modifications.

loaded programs from ctn target directory to bin directory:

**cd $HOME/tctn/bin**

**cp MIRCS\_dicom\_broadcast $BIN**

**cp MIRCS\_simple\_storage $BIN**

**cp dcm\_modify\_object $BIN**

**cp dcm\_dump\_file $BIN**

**cp dicom\_echo $BIN**

set access and SUID on MIRCS\_simple\_storage (as su)

**chmod 755 MIRCS\_simple\_storage**

**chown root:erl MIRCS\_simple\_storage**

**chmod +s MIRCS\_simple\_storage**

Afterwards, should look like this:

**ls –l MIRCS\_simple\_storage**

-rwsr-sr-x 1 root erl 1018752 Jun 29 10:42 MIRCS\_simple\_storage

**(as spectrum) loaded programs from masters/programs directory to bin directory:**

**cd $HOME/masters/programs**

**cp age $BIN**

**cp CheckStudy $BIN**

**cp configure $BIN**

**cp Itime $BIN**

**cp PACSO\_Query $BIN**

**cp PFix $BIN**

**cp SendAlert $BIN**

**cp SentTest $BIN**

**bin**

**chmod 755 \*.csh**

**cp $HOME/masters/qc**

**cp Run $BIN**

**using ftp mput over qc directory from relay2. Deleted compiled programs.**

Copied over programs from masters/qc

**cd $HOME/masters/qc**

**cp log\_purge $HOME/production/qc**

**cp qc\_signal\_server $HOME/production/qc**

**cp resend\_qc $HOME/production/qc**

**Set up process lock fifo’s:**

**cd $HOME/production/qc**

**mkfifo qc\_signal\_server.prs**

**mkfifo resend\_qc.prs**

**Copied most recent Distributor.cfg to bin directory and ran load command.**

**This install is being modified to run simple\_storage out of the bin directory, so I Modified the Distributor.cfg “SetFile” lines for MIRCS\_simple\_storage.ini and MIRCS\_simple\_storage.map to load those files into bin for bond, but not for the other servers. When future installs are done, this modification will have to be expanded to include them also. Also modified configure.c to handle server entries on “SetFile” lines. This modification has been propagated to all distributors.**

As root, copied the REBOOT\_SPECTRUM file from relay2 to /etc/init.d, set its permissions and ownership, and linked it into startup directory. REBOOT\_SPECTRUM is a script that clears distributor temporary files on reboot.

**cd /etc/init.d**

**cp /export/home/spectrum/REBOOT\_SPECTRUM .**

**chmod 755 REBOOT\_SPECTRUM**

**chown root:other REBOOT\_SPECTRUM**

**cd /etc/rc3.d**

**ln –s /etc/init.d/REBOOT\_SPECTRUM /etc/rc3.d/S90REBOOT\_SPECTRUM**

Did basically the same thing for dicomstorage. This is the script which starts up the simple\_storage DICOM receiver. This is a process, and could be reworked to run under the SMF, as MySQL was above, but I decided not to do that at this time. dicomstorage had to be modified slightly to run MIRCS\_simple\_storage out of the bin directory, and to change the AE title to BOND. This later change would also need to be made if this script is copied to other distributors.

**cd /etc/init.d**

**cp /export/home/spectrum/dicomstorage .**

**chmod 755 dicomstorage**

**chown root:other dicomstorage**

**ln -s /etc/init.d/dicomstorage /etc/rc3.d/S80dicomstorage**

This command is needed for dynamic library loading

**crle -l /usr/lib:/usr/ucblib:/usr/local/lib:/usr/local/mysql/lib/mysql**

Copied spectrum cron table from relay2 into crontab –e and edited it to be reasonable for this distributor, to change specific references to relay2 to bond, and to comment out all active lines.

Logged into rentgn as radman and set up my temporary host for testing. This would only be necessary if your temporary IP is different from the temporary IP I was given (10.39.186.233), but you would also need to do it for the permanent host name and IP if that was different from those previously assigned.

**UCX**

**TCPIP> SET HOST /ADDRESS=10.39.186.233 TEST\_BOND**

**TCPIP> EXIT**

**Test…**

**Set Host name and IP address to desired server IP and name if not already that address. Need to edit these files:**

**/etc/hosts**

**/etc/inet/ipnodes**

**/etc/defaultrouter**

**/etc/nodename**

**/etc/hostname.bge0 (and or other interfaces)**

**/etc/resolv.conf**

**/etc/netmasks**

**The PACSO merge process requires relay1 (which hosts the PACSO\_Query server) to be able to log in to the mysql db on all the distributors as spectrum;  However, I had only included privileges on the new bond to log in as spectrum if you were logging in from the localhost.  As a result, the queries to bond failed, and the PACSO\_Query server is set up to send the e-mail if any of the queries fail to be on the safe side.**

**Find your create\_imagelog.sql file on the new bigben and add this line:**

**GRANT SELECT ON studylog TO** **spectrum@’10.33.162.%****’;**

**after the other GRANT commands (near the end of the file.**

**Then rerun the step in the install which uses create\_imagelog.sql to define the database.**