

Using the NJE/IP Bridge: Sample Configuration and Examples

By:
Sine Nomine Associates
43596 Blacksmith Square
Ashburn, VA 20147

01 December 2005

Copyright © 2005 Sine Nomine Associates

All rights reserved. No part of the contents of this material may be reproduced or transmitted in any form or by any means without the written permission of Sine Nomine Associates.

Document History

<i>Date</i>	<i>Revision/ Reissue</i>	<i>Nature of Change</i>
Dec 1, 2005	Version 01	New Release

Table of Contents

1	USING THE NJE/IP BRIDGE.....	9
1.1	Definitions.....	9
1.2	Topology.....	10
2	PRODUCT INSTALLATION AND CUSTOMIZATION.....	12
2.1	Appliance Installation	12
2.1.1	Appliance Virtual Machine Definition	12
2.1.2	Appliance Installation Synopsis	12
2.2	Pre-Existing Linux Installation	21
2.3	Solaris Installation	22
2.3.1	GNU Support Packages.....	23
2.3.2	Installing the NJE/IP Bridge Application on Solaris	23
2.3.3	Updating User Profiles to Allow Users to Access NJE Services	24
2.4	Apply License Key.....	24
2.5	Node Configuration	25
2.5.1	NJE2	25
2.5.2	RSCS2	26
2.5.3	SNAVM5.....	27
2.5.4	TEST1	28
2.5.5	VA1DEV10.....	29
2.5.6	VSE27	29
3	USING NJE/IP BRIDGE SERVICES.....	31
3.1	Logging.....	31
3.2	Using the UCP Control Interface.....	31
3.2.1	Help Information	31
3.2.2	Querying Link/Line State.....	32
3.3	Obtaining Status Information from Other Nodes	32
3.3.1	Querying a Remote RSCS Node.....	33
3.3.2	Querying Other NJE/IP Bridge Nodes.....	33

3.4 Submitting Jobs from Linux34

3.5 Sending Commands to VSE.....35

3.6 Sending Commands to z/VM.....36

3.7 Sending Files via NJE.....37

3.8 Using the file-exit Configuration File.....38

3.9 Using NQS to Execute Jobs39

 3.9.1 Sample Job..... 39

 3.9.2 Submitting a Script For Execution 39

4 SUMMARY42

4.1 IBM Documentation42

List of Figures

Figure 1: Sample NJE Network Topology 10

Figure 2: Sample z/VM Directory Entry for Appliance..... 12

Figure 3: Annotated Appliance Install Log 13

Figure 4: Installing NJE/IP Bridge RPMs on Pre-Existing Linux Instance 21

Figure 5: Installing GNU Prerequisites on Solaris 23

Figure 6: Installing the NJE/IP Bridge on Solaris 23

Figure 7: Example User Profile Updates to Enable NJE Service Access 24

Figure 8: Command to Add Users to Group nje 24

Figure 9: Configuration for Node NJE2..... 26

Figure 10: z/VM Directory Entry for Node RSCS2 27

Figure 11: RSCS2 Configuration File 27

Figure 12: z/VM Directory Entry for Node SNAVM5..... 28

Figure 13: RSCS Configuration Statements for Node SNAVM5..... 28

Figure 14: Configuration File for Node TEST1..... 29

Figure 15: Configuration for Node VA1DEV10..... 29

Figure 16: POWER PNODE Macro Definitions..... 30

Figure 17: POWER Generation Macros..... 30

Figure 18: Example Output from UCP Help Command 32

Figure 19: Sample Output from UCP SHOW LINE..... 32

Figure 20: Sample Remote RSCS QUERY SYSTEM Response..... 33

Figure 21: NJE/IP Bridge Default Response To The Remote HELP Command..... 33

Figure 22: NJE/IP Bridge Response to CPQUERY CPU..... 33

Figure 23: Sample Output from CPQUERY USER..... 34

Figure 24: Sample Output from CPQUERY CPLEVEL..... 34

Figure 25: Sample VSE JECL To Submit to Node VSE27 34

Figure 26: Example Job Submission and Messages 34

Figure 27: Sample QRDR Output for Example Job 35

Figure 28: Output of PEEK for NJE Spool File 1 35

Figure 29: Using RECEIVE to Copy A NJE Spool File To The Current Directory 35

Figure 30: Response from Remote POWER ‘D A’ Command 36

Figure 31: Response to Remote RSCS Command 36

Figure 32: Interactive Message Example 37

Figure 33: SENDFILE Example 37

Figure 34: Displaying Incoming Virtual Reader Spool Files on VM 37

Figure 35: Displaying File Tags on VM 37

Figure 36: Previewing Spool Files on VM 38

Figure 37: file-exit.cf Entry to Direct a NJE Address to a Local Printer Queue 38

Figure 38: Example POWER JECL to Directly Print Output on NJE/IP Bridge Controlled Printer
..... 38

Figure 39: Example Shell Script For Remote NQS Execution 39

Figure 40: Example Job Submission from VM to Linux Node Running NJE/IP Bridge and NQS
..... 40

Figure 41: Results of a Remotely Executed NQS Job 40

Figure 42: Preview Output from Remotely Executed Job 41

List of Tables

Table 1: Terms Used in This Document	9
Table 2: Sample Network Node Description.....	11

1 Using the NJE/IP Bridge

This document contains descriptions of a sample installation and configuration of the NJE/IP Bridge to integrate VSE, z/VM, Linux, and Solaris systems. The configuration example allows:

1. Submission of jobs from Linux to a VSE system
2. Transfer of files from one system to another
3. Printing of documents from z/VM and VSE on a Linux system
4. Execution of simple scripts on a Linux system instigated by job submission on z/VM

The configuration includes the use of the “NJE Appliance” which provides channel-to-channel (CTC) access to a system not supporting NJE over IP. Standard TCPNJE connections are also described.

z/OS systems may also participate in such an NJE environment. In fact, one of the main reasons for developing this product was to allow “older” JES systems to participate in a TCPNJE configuration. However, at the time this document was written no access to such a system was available and thus discussion of this configuration is not included here. Such a system would operate very similarly to the use of the appliance mentioned above.

1.1 Definitions

The following definitions are used throughout this document.

Table 1: Terms Used in This Document

Term	Definition
Appliance	A pre-configured Linux-based “black box” implementing a specific function (for example, NJE) that is installed in a virtual machine or LPAR.
NJE	Network Job Entry: A facility for transmitting jobs (JCL and in-stream data sets), SYSOUT data sets, (job-oriented) operator commands/messages, and job accounting information from one computing system to another.

NJE/IP Bridge	Software that implements the NJE protocol and provides a CTC device driver that allows a bridge from a non-IP NJE system to an IP network.
NQS	The NJE/IP Bridge includes support for an optional batch submission and execution application for UNIX and UNIX-like systems. It is based on NQS (Network Queuing System). It is automatically installed with the NJE/IP Appliance and can be installed on non-appliance systems using normal installation tools like RPM.

1.2 Topology

The following diagram illustrates the systems and their inter-connections for the NJE network defined in this document. .

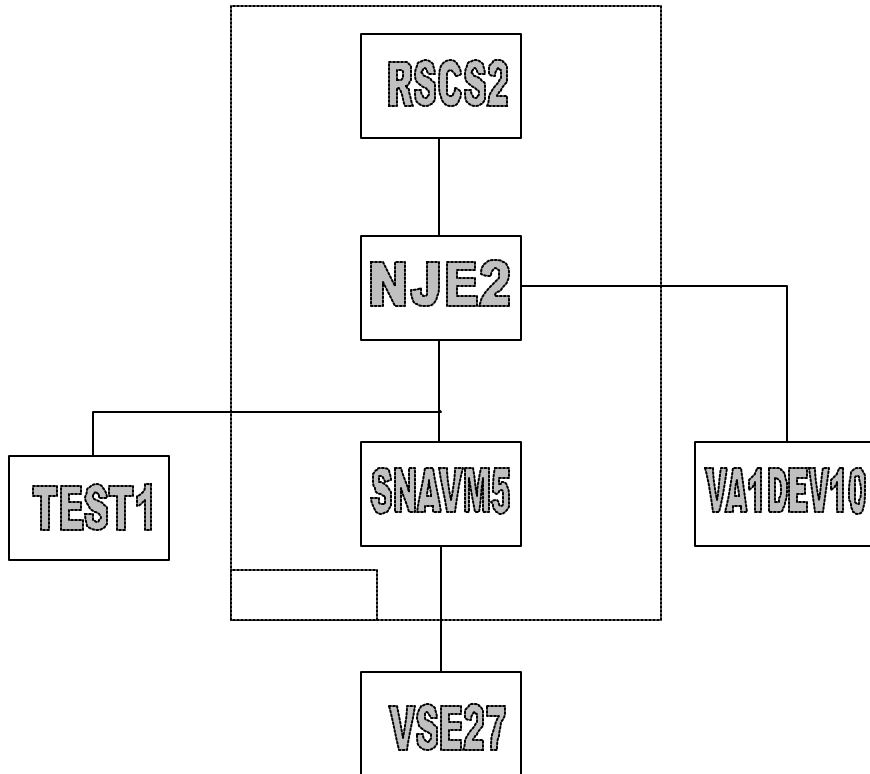


Figure 1: Sample NJE Network Topology

The attributes of each of these nodes is show in the following table:

Table 2: Sample Network Node Description

Node	Platform	Operating System	IP Name
RSCS2	zSeries	z/VM 4.4 - RSCS	N/A
NJE2	zSeries	NJE/IP Appliance	nje2.vm.example.com
SNAVM5	zSeries	z/VM 4.4 – RSCS	vm.example.com
TEST1	x86	Fedora Core 3	test1.example.com
VA1DEV10	Sparc	Solaris 8	va1dev10.example.com
VSE27	zSeries	VSE/ESA 2.7	vse27.external.net

2 Product Installation and Customization

In this section we describe the installation of the NJE/IP Bridge as an appliance and as an application on Linux and Solaris systems.

2.1 Appliance Installation

The NJE IP Appliance is a pre-configured Linux system providing a fully functional virtual machine or LPAR system that supports both TCPNJE and CTC connections to other NJE systems such as z/VM, VSE, or z/OS. It is particularly aimed at z/OS systems that do not have the level of JES2 that supports TCPNJE connections.

2.1.1 Appliance Virtual Machine Definition

The appliance was installed on the VM system using the following z/VM Directory entry.

Note: The appliance system assumes the provision of a portname for OSA devices is optional. You must have the necessary microcode or z/VM PTF applied to enable this.

```
USER NJE2 XXXXXXXX 128M 512M G
CPU 0
CPU 1
IPL CMS PARM AUTOCR
MACHINE XA 2
CONSOLE 001F 3215 T *
SPECIAL 005A QDIO 3 SYSTEM LNXLAN3
SPECIAL 0060 CTCA
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
LINK MAINT 019F 019F RR
MDISK 0150 3390 2501 500 VM100D
MDISK 0151 3390 1043 100 310W01
MDISK 0152 FB-512 V-DISK 200000
MDISK 0191 3390 2397 50 310W01
```

2 virtual CPUs were defined just to verify SMP operation of CTC driver

Figure 2: Sample z/VM Directory Entry for Appliance

2.1.2 Appliance Installation Synopsis

Installation of the appliance is performed by booting from a tape that contains a Linux installer system; the appliance operating system; and application code. The appliance expects the following:

1. Tape drive
2. 2 minidisks:

- 0x151 – 500 cylinders
 - 0x152 – 100 cylinders (may be bigger in case of larger spool area requirements)
3. Main storage: 128MB
 4. IP addresses: node, gateway, DNS
 5. IP netmask

The following annotated log is derived from an install under z/VM using a tape drive attached as 181 (user responses/commands are in bold):

Figure 3: Annotated Appliance Install Log

```
#CP REW 181
Rewind complete
#CP IPL 181 CLEAR
Linux version 2.4.27-2-s390 (waldi@raptor) (gcc version 3.3.5 (Debian 1:3.3.5-6)) #1 SMP Wed Jan 26 14:44:31 CET 2005
We are running under VM (31 bit mode)
This machine has no PFI support
This machine has an IEEE fpu
On node 0 totalpages: 32768
zone(0): 32768 pages.
zone(1): 0 pages.
zone(2): 0 pages.
Kernel command line: ro debian-installer/locale=C

Highest subchannel number detected (hex) : 000E
Calibrating delay loop... 412.05 BogoMIPS
Memory: 123392k/131072k available (2235k kernel code, 0k reserved, 380k data, 60k init)
Dentry cache hash table entries: 16384 (order: 5, 131072 bytes)
Inode cache hash table entries: 8192 (order: 4, 65536 bytes)
Mount cache hash table entries: 512 (order: 0, 4096 bytes)
Buffer cache hash table entries: 8192 (order: 3, 32768 bytes)
Page-cache hash table entries: 32768 (order: 5, 131072 bytes)
debug: Initialization complete
POSIX conformance testing by UNIFIX
Detected 2 CPU's
Boot cpu address 0
cpu 0 phys_idx=0 vers=FF ident=01131C machine=7060 unused=0000
Waiting on wait_init_idle (map = 0x2)
cpu 1 phys_idx=1 vers=FF ident=01131C machine=7060 unused=0000
All processors have done init_idle
init_mach : starting machine check handler
mach_handler : ready
mach_handler : waiting for wakeup
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Initializing RT netlink socket
Starting kswapd
VFS: Disk quotas vdquot_6.5.1
```

The installer is based on a 2.4 kernel, but the appliance itself is a 2.6 kernel.

```

Journalled Block Device driver loaded
devfs: vl.12c (20020818) Richard Gooch (rgooch@atnf.csiro.au)
devfs: boot_options: 0x0
pty: 256 Unix98 ptys configured
devfs_register(ttyS): could not append to parent, err: -17
dasd: initializing...
dasd: Registered successfully to major no 94
dasd(eckd): ECKD discipline initializing
dasd(eckd): We are interested in: CU 3880/00
dasd(eckd): We are interested in: CU 3990/00
dasd(eckd): We are interested in: CU 2105/00
dasd(eckd): We are interested in: CU 9343/00
dasd: Registered ECKD discipline successfully
dasd(fba): FBA discipline initializing
dasd(fba): We are interested in: Dev 9336/00 @ CU 6310/00
dasd(fba): We are interested in: Dev 3370/00 @ CU 3880/00
dasd: Registered FBA discipline successfully
dasd: initialization finished
RAMDISK driver initialized: 16 RAM disks of 24576K size 1024 blocksize
Initializing Cryptographic API
qdio: loading QDIO base support version 2 ($Revision: 1.145.4.11 $/$Revision:
1.66.4.5 $)
IPv6 v0.8 for NET4.0
IPv6 over IPv4 tunneling driver
NET4: Linux TCP/IP 1.0 for NET4.0
IP: routing cache hash table of 512 buckets, 8Kbytes
TCP: Hash tables configured (established 4096 bind 8192)
NET4: Unix domain sockets 1.0/SMP for Linux NET4.0.
RAMDISK: Compressed image found at block 0
Freeing initrd memory: 2631k freed
VFS: Mounted root (ext2 filesystem) readonly.
Freeing unused kernel memory: 15k freed
Setting up filesystem, please wait ...
Using /lib/modules/2.4.27-2-s390/kernel/drivers/s390/char/tape390.o
Warning: loading tape390 will taint the kernel: no license
tape_devmap:No parameters supplied, enabling auto detect mode for all
supported devices.
TCHAR:Tape gets major 152 for char device
TBLOCK:tape gets major 4 for block device
T390:(0181): Tape has been mounted
Running in VM mode....
dasd: /proc/dasd/devices: 'add device range=150-151'
dasd(eckd): /dev/dasda ( 94: 0),0150@09: 3390/0A(CU:3990/02) Cyl:500 Head:15
Sec:224
dasd(eckd): /dev/dasda ( 94: 0),0150@09: 3390/0A(CU:3990/02): Configuration
data read
dasd(eckd): /dev/dasdb ( 94: 4),0151@0a: 3390/0A(CU:3990/02) Cyl:100 Head:15
Sec:224
dasd(eckd): /dev/dasdb ( 94: 4),0151@0a: 3390/0A(CU:3990/02): Configuration
data read
dasd: waiting for responses...
dasd(eckd): /dev/dasda ( 94: 0),0150@09: (4kB blks): 360000kB at 48kB/trk
compatible disk layout
dasd(eckd): /dev/dasdb ( 94: 4),0151@0a: (4kB blks): 72000kB at 48kB/trk
compatible disk layout
Partition check:
/dev/dasd/0150:VOL1/ 0X0150: p1

```

```
/dev/dasd/0151:VOL1/ 0X0151: p1
Preparing disks....
Printing hashmark every 100 cylinders.
```

Disk Devices are automatically partitioned and formatted by the installer

```
#####
```

```
dasd: waiting for responses...
dasd(eckd): /dev/dasda ( 94: 0),0150@09: (4kB blks): 360000kB at 48kB/trk
compatible disk layout
/dev/dasd/0150:(nonl)/ : p1
Finished formatting the device.
/dev/dasd/0150:VOL1/ 0X0150:
Rereading the partition table... ok
Printing hashmark every 100 cylinders.
```

```
#
```

```
dasd: waiting for responses...
dasd(eckd): /dev/dasdb ( 94: 4),0151@0a: (4kB blks): 72000kB at 48kB/trk
compatible disk layout
/dev/dasd/0151:(nonl)/ : p1
Finished formatting the device.
/dev/dasd/0151:VOL1/ 0X0151:
Rereading the partition table... ok
auto-creating one partition for the whole disk...
```

```
writing volume label...
writing VTOC...
rereading partition table...
/dev/dasd/0150:VOL1/ 0X0150: p1
auto-creating one partition for the whole disk...
writing volume label...
writing VTOC...
rereading partition table...
```

```
/dev/dasd/0151:VOL1/ 0X0151: p1
mke2fs 1.37 (21-Mar-2005)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
90048 inodes, 89976 blocks
899 blocks (1.00%) reserved for the super user
First data block=0
3 block groups
32768 blocks per group, 32768 fragments per group
30016 inodes per group
Superblock backups stored on blocks:
    32768
```

```
Writing inode tables: 0/3 1/3 2/3 done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 31 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
mke2fs 1.37 (21-Mar-2005)
Filesystem label=
OS type: Linux
```

```

Block size=4096 (log=2)
Fragment size=4096 (log=2)
17984 inodes, 17976 blocks
179 blocks (1.00%) reserved for the super user
First data block=0
1 block group
32768 blocks per group, 32768 fragments per group
17984 inodes per group

Writing inode tables: 0/1          done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information:
done

```

```

This filesystem will be automatically checked every 22 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
kjournald starting. Commit interval 5 seconds
EXT3 FS 2.4-0.9.19, 19 August 2002 on dasd(94,1), internal journal
EXT3-fs: mounted filesystem with ordered data mode.
kjournald starting. Commit interval 5 seconds
EXT3 FS 2.4-0.9.19, 19 August 2002 on dasd(94,5), internal journal
EXT3-fs: mounted filesystem with ordered data mode.
Restoring file system from tape...
dasd: /proc/dasd/devices: 'add device range=152'
dasd(fba): /dev/dasdc ( 94: 8),0152@0b: 9336/10(CU:6310/80) 97MB at(512
B/blk)
/dev/dasd/0152:CMS1/ LXSWAP(MDSK): pl
Adding Swap: 99156k swap-space (priority -1)
Setting up environment....

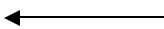
```

```

*****
*                               *
*           Network Configuration           *
*                               *
*****
IP Address ( 10.0.0.0.2 ) >
192.168.130.92
Netmask ( 255.255.255.0 ) >
255.255.255.192
IP Address of router ( 192.168.130.65 ) >
192.168.130.68
IP Address of DNS server ( 192.168.130.68 ) >
192.168.101.26
Fully-qualified domain name ( tcpnje.example.com ) >
nje2.vm.example.com
IP:          192.168.130.92
Netmask:    255.255.255.192
Router:     192.168.130.68
DNS Server: 192.168.101.26
FQDN:      nje2.vm.example.com
Parameters OK ( Y ) >
y
Using config file '/etc/zipl.conf'
Building bootmap '/boot/bootmap'
Building menu 'menu'
Adding #1: IPL section 'Linux-2.6' (default)
Adding #2: IPL section 'Linux-2.4'
Preparing boot device: 0150.

```

The user specifies the network configuration information here



```
Done.
Preparing to halt....
IPL from device 150 after system halts.
The system is going down NOW !!
Sending SIGTERM to all processes.
Sending SIGKILL to all processes.
The system is halted. You may reboot now.
System halted.
01: HCPGSP2629I The virtual machine is placed in CP mode due to a SIGP stop
from CPU 00.
00: HCPGSP2630I The virtual machine is placed in CP mode due to a SIGP stop
and store status from CPU 00.
```

IPL 150 CLEAR

zIPL v1.3.1 interactive boot menu

- 0. default (Linux-2.6)
- 1. Linux-2.6
- 2. Linux-2.4

Note: VM users please use '#cp vi vmsg <input>'

Please choose (default will boot in 10 seconds):

```
Booting default (Linux-2.6)...
Linux version 2.6.8 (root@d26test) (gcc version 3.3.5 (Debian 1:3.3.5-13)) #1
SMP Fri Jun 3 16:22:34 EDT 2005
We are running under VM (31 bit mode)
This machine has an IEEE fpu
Built 1 zonelists
Kernel command line: noinitrd ro vmpoff="LOGOFF" devfs=nomount
dasd=0.0.0150,0.0.0151,0.0.0152 root=/dev/dasdal BOOT_IMAGE=0
PID hash table entries: 1024 (order 10: 8192 bytes)
Dentry cache hash table entries: 32768 (order: 5, 131072 bytes)
Inode-cache hash table entries: 16384 (order: 4, 65536 bytes)
Memory: 125152k/131072k available (1907k kernel code, 0k reserved, 679k data,
84k init)
Calibrating delay loop...
412.05 BogoMIPS
Security Scaffold v1.0.0 initialized
Capability LSM initialized
Mount-cache hash table entries: 512 (order: 0, 4096 bytes)
Detected 2 CPU's
Boot cpu address 0
cpu 0 phys_idx=0 vers=FF ident=01131C machine=7060 unused=0000
cpu 1 phys_idx=1 vers=FF ident=01131C machine=7060 unused=0000
Brought up 2 CPUs
debug: Initialization complete
NET: Registered protocol family 16
VFS: Disk quotas dquot_6.5.1
Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
Initializing Cryptographic API
RAMDISK driver initialized: 16 RAM disks of 24576K size 1024 blocksize
Channel measurement facility using basic format (autodetected)
qdio: loading QDIO base support version 2 ($Revision: 1.84 $/$Revision: 1.57
$/$Revision: 1.26 $)
dasd(eckd): 0.0.0150: 3390/0A(CU:3990/02) Cyl:500 Head:15 Sec:224
```

The appliance may run in an SMP configuration but it is not required

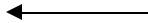


```

Using anticipatory io scheduler
dasd(eckd): 0.0.0150: (4kB blks): 360000kB at 48kB/trk compatible disk layout
  dasda:VOL1/ 0X0150: dasdal
dasd(eckd): 0.0.0151: 3390/0A(CU:3990/02) Cyl:100 Head:15 Sec:224
dasd(eckd): 0.0.0151: (4kB blks): 72000kB at 48kB/trk compatible disk layout
  dasdb:VOL1/ 0X0151: dasdbl
dasd(fba): 0.0.0152: 9336/10(CU:6310/80) 97MB at(512 B/blk)
  dasdc:CMS1/ LXSWAP(MDSK): dasdcl
NET: Registered protocol family 2
IP: routing cache hash table of 1024 buckets, 8Kbytes
TCP: Hash tables configured (established 4096 bind 8192)
NET: Registered protocol family 1
NET: Registered protocol family 10
IPv6 over IPv4 tunneling driver
NET: Registered protocol family 17
kjournald starting. Commit interval 5 seconds
EXT3-fs: mounted filesystem with ordered data mode.
VFS: Mounted root (ext3 filesystem) readonly.
Freeing unused kernel memory: 84k freed
INIT: version 2.86 booting
devfsd: No devfs on /dev, not starting.
/etc/rcS.d/S02mountvirtfs: line 40: /dev/null: Read-only file system
Mounting a tmpfs over /dev...done.
Creating initial device nodes...done.
Activating swap.
Adding 99156k swap on /dev/dasdcl. Priority:-1 extents:1
Checking root file system...
fsck 1.37 (21-Mar-2005)
/dev/dasdal: clean, 16520/90048 files, 77936/89976 blocks
EXT3 FS on dasdal, internal journal
Cleaning up ifupdown...done.
Calculating module dependencies...
done.
Loading modules...
  qeth
qeth: loading qeth S/390 OSA-Express driver ($Revision: 1.130 $/$Revision:
1.113 $/$Revision: 1.36 $/$Revision: 1.11 $/$Revision: 1.
9 $/$Revision: 1.10 $/$Revision: 1.33 $ :VLAN)
  njectc_mod
njectc: Driver loaded for 5 connections
All modules loaded.
Checking all file systems...
fsck 1.37 (21-Mar-2005)
/dev/dasdbl: clean, 12/17984 files, 1598/17976 blocks
Setting kernel variables ...
... done.
Mounting local filesystems...
kjournald starting. Commit interval 5 seconds
EXT3 FS on dasdbl, internal journal
EXT3-fs: mounted filesystem with ordered data mode.
/dev/dasdbl on /var/spool/nje type ext3 (rw)
Cleaning /tmp /var/run /var/lock.
Initializing ccwgroup devices...qeth: Device 0.0.005a/0.0.005b/0.0.005c is a
Guest LAN QDIO card (level: V44G)
with link type GuestLAN QDIO (portname: )
qeth: IP fragmentation not supported on eth0
qeth: VLAN enabled

```

The CTC line driver is loaded and detects devices. By default up to 5 CTC devices can be supported

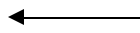


```

qeth: Multicast enabled
qeth: Broadcast enabled
qeth: Could not set up broadcast echo filtering on eth0: 0x2
qeth: Using SW checksumming on eth0.
done.
Setting up networking...done.
Starting hotplug subsystem:
  pci
  pci      [success]
  usb
  usb      [success]
  isapnp
  isapnp   [success]
  ide
  ide      [success]
  input
  input    [failed]
  scsi
  scsi     [success]
done.
Setting up IP spoofing protection: rp_filter.
Configuring network interfaces...
done.
Initializing random number generator...done.
Recovering nvi editor sessions... done.
INIT: Entering runlevel: 2
devfsd: No devfs on /dev, not starting.
Starting system log daemon: syslogd
.
Starting kernel log daemon: klogd.
Starting MTA:
exim4.
Starting internet superserver: inetd.
Starting printer spooler: lpd
.
Starting OpenBSD Secure Shell server: sshd
.
/etc/nqs/machines not found: initializing NQS environment
NMAP_SUCCESS: Successful completion.
NMAP_SUCCESS: Successful completion.
  MID      PRINCIPAL NAME  ALIASES
  ---      -
3232268892   nje2.vm.example.com   nje2
Utility [nqsmkdirs]: Creating: control/+++++.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: control/+++++0.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: control/+++++1.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: control/+++++2.
Utility [nqsmkdirs]: mkdir exited (0).
:
:
:
Utility [nqsmkdirs]: Creating: transact/+++++R.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: transact/+++++S.

```

Upon 1st boot, the NQS subsystem will initialize itself. NQS is a batch processing facility for Linux.



```
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: NQS database subdirectory construction is complete.
Utility [nqsmkdirs]: Exiting.
Utility [nqsmktrans]: Beginning transaction descriptor construction:
```

```

  1    2    3    4    5    6    7    8    9   10   11   12   13
14   15   16   17   18   19   20   21   22   23   24   25   26
27   28   29   30   31   32   33   34   35   36   37   38   39
40   41   42   43   44   45   46   47   48   49   50   51   52
53   54   55   56   57   58   59   60   61   62   63   64   65
66   67   68   69   70   71   72   73   74   75   76   77   78
79   80   81   82   83   84   85   86   87   88   89   90   91
92   93   94   95   96   97   98   99  100  101  102  103  104
105  106  107  108  109  110  111  112  113  114  115  116  117
118  119  120  121  122  123  124  125  126  127  128  129  130
131  132  133  134  135  136  137  138  139  140  141  142  143
144  145  146  147  148  149  150  151  152  153  154  155  156
157  158  159  160  161  162  163  164  165  166  167  168  169
170  171  172  173  174  175  176  177  178  179  180  181  182
183  184  185  186  187  188  189  190  191  192  193  194  195
196  197  198  199  200  201  202  203  204  205  206  207  208
209  210  211  212  213  214  215  216  217  218  219  220  221
222  223  224  225  226  227  228  229  230  231  232  233  234
235  236  237  238  239  240  241  242  243  244  245  246  247
248  249  250  251  252  253  254  255  256  257  258  259  260
261  262  263  264  265  266  267  268  269  270  271  272  273
274  275  276  277  278  279  280  281  282  283  284  285  286
287  288  289  290  291  292  293  294  295  296  297  298  299
300  301  302  303  304  305  306  307  308  309  310  311  312
313  314  315  316  317  318  319  320  321  322  323  324  325
326  327  328  329  330  331  332  333  334  335  336  337  338
339  340  341  342  343  344  345  346  347  348  349  350  351
352  353  354  355  356  357  358  359  360  361  362  363  364
365  366  367  368  369  370  371  372  373  374  375  376  377
378  379  380  381  382  383  384  385  386  387  388  389  390
391  392  393  394  395  396  397  398  399
```

```
Utility [nqsmktrans]: Waiting for permanent storage update.
Utility [nqsmktrans]: Wait time = 10 seconds.
Utility [nqsmktrans]: Completing construction of final transaction
descriptor.
400
Utility [nqsmktrans]: NQS transaction descriptor construction complete.
Utility [nqsmktrans]: Exiting.
NQS manager[TCML_COMPLETE ]: Transaction complete at local host
NQS manager[TCML_COMPLETE ]: Transaction complete at local host
Queue standard created.
NQS manager[TCML_COMPLETE ]: Transaction complete at local host
NQS manager[TCML_COMPLETE ]: Transaction complete at local host
NQS manager[TCML_COMPLETE ]: Transaction complete at local host
Starting NQS daemon: NQS manager[TCML_COMPLETE ]: Transaction complete at
local host
done.
```

```
*****
* NO NQE LICENSE KEY FOUND IN /etc/nje/nje.key! *
* Please install key in order to run NQE Bridge. *
*****
```

To enable NQE you will have to acquire a license key

```
Starting deferred execution scheduler: atd.
```

```
Starting periodic command scheduler: cron
```

```
.
```

```
Debian GNU/Linux 3.1 nje2 console
```

```
nje2 login:
```

```
root
```

```
Password:
```

```
rootpass
```

Make sure you change root's password!

```
Last login: Thu Oct 27 09:31:54 2005 on console
```

```
Linux nje2 2.6.8 #1 SMP Fri Jun 3 16:22:34 EDT 2005 s390 GNU/Linux
```

```
Version 1.0
```

```
nje2:~#
```

2.2 Pre-Existing Linux Installation

If your site already has Linux guests running, the NJE/IP Bridge can be installed as an application on a pre-existing Linux system. This section describes that process in our example network.

Installation of NJE and NQS is performed using the rpm command:

Figure 4: Installing NJE/IP Bridge RPMs on Pre-Existing Linux Instance

```

rpm -Uhv nje-1.0-5.i386.rpm
Preparing...                               ##### [100%]
 1:nje                                       ##### [100%]
rpm -Uhv nqs-4.0-1.i386.rpm
Preparing...                               ##### [100%]
 1:nqs                                       ##### [100%]
NMAP_SUCCESS: Successful completion.
NMAP_SUCCESS: Successful completion.
MID      PRINCIPAL NAME  ALIASES
---      -
3232261406    test1.example.com    test1
Utility [nqsmkdirs]: Creating: control/+++++.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: control/+++++0.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: control/+++++1.
Utility [nqsmkdirs]: mkdir exited (0).
:
:
:
Utility [nqsmkdirs]: Creating: transact/+++++Q.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: transact/+++++R.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: Creating: transact/+++++S.
Utility [nqsmkdirs]: mkdir exited (0).
Utility [nqsmkdirs]: NQS database subdirectory construction is complete.
Utility [nqsmkdirs]: Exiting.
Utility [nqsmktrans]: Beginning transaction descriptor construction:

```

```

1      2      3      4      5      6      7      8      9      10     11     12     13
14     15     16     17     18     19     20     21     22     23     24     25     26
27     28     29     30     31     32     33     34     35     36     37     38     39
40     41     42     43     44     45     46     47     48     49     50     51     52
53     54     55     56     57     58     59     60     61     62     63     64     65
66     67     68     69     70     71     72     73     74     75     76     77     78
79     80     81     82     83     84     85     86     87     88     89     90     91
92     93     94     95     96     97     98     99     100    101    102    103    104
105    106    107    108    109    110    111    112    113    114    115    116    117
118    119    120    121    122    123    124    125    126    127    128    129    130
131    132    133    134    135    136    137    138    139    140    141    142    143
144    145    146    147    148    149    150    151    152    153    154    155    156
157    158    159    160    161    162    163    164    165    166    167    168    169
170    171    172    173    174    175    176    177    178    179    180    181    182
183    184    185    186    187    188    189    190    191    192    193    194    195
196    197    198    199    200    201    202    203    204    205    206    207    208
209    210    211    212    213    214    215    216    217    218    219    220    221
222    223    224    225    226    227    228    229    230    231    232    233    234
235    236    237    238    239    240    241    242    243    244    245    246    247
248    249    250    251    252    253    254    255    256    257    258    259    260
261    262    263    264    265    266    267    268    269    270    271    272    273
274    275    276    277    278    279    280    281    282    283    284    285    286
287    288    289    290    291    292    293    294    295    296    297    298    299
300    301    302    303    304    305    306    307    308    309    310    311    312
313    314    315    316    317    318    319    320    321    322    323    324    325
326    327    328    329    330    331    332    333    334    335    336    337    338
339    340    341    342    343    344    345    346    347    348    349    350    351
352    353    354    355    356    357    358    359    360    361    362    363    364
365    366    367    368    369    370    371    372    373    374    375    376    377
378    379    380    381    382    383    384    385    386    387    388    389    390
391    392    393    394    395    396    397    398    399

```

```

Utility [nqsmktrans]:  Waiting for permanent storage update.
Utility [nqsmktrans]:  Wait time = 10 seconds.
Utility [nqsmktrans]:  Completing construction of final transaction
descriptor.
400
Utility [nqsmktrans]:  NQS transaction descriptor construction complete.
Utility [nqsmktrans]:  Exiting.
NQS manager[TCML_COMPLETE ]:  Transaction complete at local host
NQS manager[TCML_COMPLETE ]:  Transaction complete at local host
Queue standard created.
NQS manager[TCML_COMPLETE ]:  Transaction complete at local host
NQS manager[TCML_COMPLETE ]:  Transaction complete at local host
NQS manager[TCML_COMPLETE ]:  Transaction complete at local host

```

Users who wish to use NJE must belong to the nje group.

2.3 Solaris Installation

Installation of the NJE application on Solaris is a little more complicated than on Linux due to the pre-requisite of several application packages that are not part of Solaris, but are required by the NJE/IP Bridge package.

2.3.1 GNU Support Packages

The NJE/IP Bridge application is built using GNU tools and libraries, which require the supporting libraries and applications to be installed on the target system. The NJE/IP Bridge distribution includes these components in a single tar ball. Extract the files from the tar ball and install using the following command sequence:

Figure 5: Installing GNU Prerequisites on Solaris

```
mkdir NJEPrereq
cd NJEPrereq
tar -xf /cdrom/NJEprereq.tar
ls
pkgadd -d ./<package name>
```

2.3.2 Installing the NJE/IP Bridge Application on Solaris

Use the pkgadd command to install the NJE/IP Bridge:

Figure 6: Installing the NJE/IP Bridge on Solaris

```
pkgadd -d /cdrom/SNAnje.pkg
Processing package instance <SNAnje> from </var/spool/pkg>

NJE - A Network Job Entry Bridge
(sparc) 1.0.5
(C) 2005 Sine Nomine Associates
## Processing package information.
## Processing system information.
    22 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <SNAnje> [y,n,?] y

Installing NJE - A Network Job Entry Bridge as <SNAnje>

## Executing preinstall script.
## Installing part 1 of 1.
/etc/init.d/njed
/etc/nje/file-exit.cf
/etc/nje/forms/SYSTEM.STANDARD
/etc/nje/msg-exit.cf
/etc/nje/nje.cf
/usr/local/bin/acctcat
/usr/local/bin/bitSEND
/usr/local/bin/bmail
/usr/local/bin/mailify
```

```

/usr/local/bin/peek
/usr/local/bin/purge
/usr/local/bin/qrdr
/usr/local/bin/receive
/usr/local/bin/rprint
/usr/local/bin/send
/usr/local/bin/sendfile
/usr/local/bin/transfer
/usr/local/bin/ygone
/usr/local/sbin/namesfilter
/usr/local/sbin/nje
/usr/local/sbin/njengs
/usr/local/sbin/njeroutes
/usr/local/sbin/ucp
/usr/local/share/doc/nje-1.0/AUTHORS
/usr/local/share/doc/nje-1.0/COPYING
/usr/local/share/doc/nje-1.0/ChangeLog
/usr/local/share/doc/nje-1.0/NEWS
/usr/local/share/doc/nje-1.0/README
/usr/local/share/info/nje-1.0/nje.info
[ verifying class <nje> ]
## Executing postinstall script.

```

Installation of <SNAnje> was successful.

2.3.3 Updating User Profiles to Allow Users to Access NJE Services

To allow users to access NJE services, add the following statements to the login scripts (for example, `.profile`) of users who will use NJE services (note that these users will also have to belong to the group `nje`, typically performed using the command shown in Figure 8):

Figure 7: Example User Profile Updates to Enable NJE Service Access

```

export PATH=/usr/local/bin:/usr/local/sbin:$PATH
export LD_LIBRARY_PATH=/usr/local/lib:/usr/local/ssl/lib:$LD_LIBRARY_PATH

```

Figure 8: Command to Add Users to Group nje

```

usermod -g nje neale

```

2.4 Apply License Key

To run the NJE/IP Bridge a license file (`nje.key`) must be acquired and placed in `/etc/nje/`. This file is supplied by your reseller and will include installation instructions at the top of the supplied file.

2.5 Node Configuration

Once the software has been installed, it must be customized to define the example network and how the nodes are connected. In this section each node configuration is presented as a table describing the overall connectivity requirements, followed by the actual configuration statements required for each of the NJE connections illustrated.

2.5.1 NJE2

Note in the table below the use of various buffer sizes. These were chosen just to exercise the line driver with different sizes. Generally, the bigger the better, but your mileage will vary according to the location of the node and the attributes of the medium used to connect to it.

Partner	IP Name	TCP Port	CTC Address	Buffer Size
RSCS2	N/A	N/A	600	31000
VA1DEV10	va1dev10.example.com	175	N/A	4096
TEST1	test1.example.com	175	N/A	4096
VSE27	vse27.external.net	175	N/A	24000
SNAVM5	vm.example.com	175	N/A	4096

```

NAME                NJE2
IPADDRESS           nje2.vm.example.com
QUEUE              /var/spool/nje
CMDMAILBOX         UDP 127.0.0.1 175
LOG                /var/log/nje/nje.log
TABLE              /etc/nje/nje.route
INFORM             ROOT@VMSL932
USEREXITS          /etc/nje/file-exit.cf
MSGEXITS           /etc/nje/msg-exit.cf
LLEVEL             1
DEFAULT-ROUTE      SNAVM5

```

```

LINE 0 VSE27
  TYPE              UNIX_TCP
  BUFSIZE           24000
  TIMEOUT           3
  IPPORT            175
  TCPNAME           vse27.bsinet.org
  TCP-SIZE          24000
  MAX-STREAMS       1

```

```

LINE 1 SNAVM5
  TYPE              UNIX_TCP
  BUFSIZE           4096
  TIMEOUT           3
  IPPORT            175

```

```

TCPNAME          VM.EXAMPLE.COM
TCP-SIZE         8192
MAX-STREAMS     7

LINE 2 RSCS2
TYPE            UNIX_CTC
BUFSIZE        24000
TIMEOUT        3
DEVICE         /dev/njectc0
MAX-STREAMS    2

LINE 3 VALDEV10
TYPE            UNIX_TCP
BUFSIZE        4096
TIMEOUT        3
IPPORT         175
TCPNAME        valdev10.example.com
TCP-SIZE       8192
MAX-STREAMS    7

LINE 4 TEST1
TYPE            UNIX_TCP
BUFSIZE        4096
TIMEOUT        3
IPPORT         175
TCPNAME        test1.example.com
TCP-SIZE       8192
MAX-STREAMS    7

```

Figure 9: Configuration for Node NJE2

2.5.2 RSCS2

Although RSCS directly supports TCPNJE, the example network uses a CTC connection to demonstrate use of the NJE/IP Bridge over a CTC connection. This example is directly analogous to the use of a CTC to connect to a z/OS or other system that does not natively support TCPNJE.

Partner	IP Name	TCP Port	CTC Address	Buffer Size
NJE2	N/A	N/A	600	31000

2.5.2.1 RSCS2 Directory Entry

```

USER RSCS2 XXXXXXXXX 16M 32M BG
  ACCOUNT 1 RSCS2
  ACIGROUP DVM
  IPL GCS PARM AUTOLOG
  IUCV ANY
  MACHINE ESA
  NAMESAVE GCS
  OPTION ACCT MAXCONN 1024 SVMSTAT LANG AMENG SETORIG

```

```

XAUTOLOG GCS
CONSOLE 001F 3215 T OPERATOR
SPECIAL 0600 CTCA NJE2
SPOOL 000C 2540 READER A
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK TCPMAINT 0198 0198 RR
LINK MAINT 0190 0190 RR
LINK MAINT 0193 0193 RR
LINK MAINT 019E 019E RR
MDISK 0191 3390 2077 0005 310RES RR

```

Figure 10: z/VM Directory Entry for Node RSCS2

2.5.2.2 RSCS CONFIG

```

LINKDEFINE NJE2 TYPE NJE QUEUE PRI LINE 600 NODE NJE2
PARAM NJE2 LIST=NO TA=1 TAPARM='TH=100' M=S B=31000
ROUTE NODE NJE2 TO LINK NJE2

```

Figure 11: RSCS2 Configuration File

2.5.3 SNAVM5

This example demonstrates the use of the standard RSCS TCPNJE support using the RSCS TCPNJE line driver.

Partner	IP Name	TCP Port	CTC Address	Buffer Size
NJE2	nje2.vm.example.com	175	N/A	4096
TEST1	test1.example.com	175	N/A	4096

2.5.3.1 SNAVM5 Directory Entry

```

USER RSCS XXXXXXXXX 16M 32M BG
ACCOUNT 1 RSCS
ACIGROUP DVM
IPL GCS PARM AUTOLOG
IUCV ANY
MACHINE ESA
NAMESAVE GCS
OPTION ACCT MAXCONN 1024 SVMSTAT LANG AMENG SETORIG
XAUTOLOG GCS
CONSOLE 001F 3215 T OPERATOR
SPOOL 000C 2540 READER A
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK TCPMAINT 0198 0198 RR
LINK MAINT 0190 0190 RR
LINK MAINT 0193 0193 RR

```

```
LINK MAINT 019E 019E RR
MDISK 0191 3390 2077 0005 310RES RR
```

Figure 12: z/VM Directory Entry for Node SNAVM5

2.5.3.2 RSCS CONFIG

```
LINKDEFINE NJE2 TYPE TCPNJE QUEUE SIZE REPLY NODE NJE2
PARAM NJE2 TCPID=TCPIP HOSTN=NJE2.VM.EXAMPLE.COM KEEPALIV=YES
ROUTE NODE NJE2 TO LINK NJE2
```

```
LINKDEFINE TEST1 TYPE TCPNJE QUEUE SIZE REPLY NODE NJE2
PARAM TEST1 TCPID=TCPIP HOSTN=TEST1.EXAMPLE.COM KEEPALIV=YES
ROUTE NODE TEST1 TO LINK TEST1
```

Figure 13: RSCS Configuration Statements for Node SNAVM5

2.5.4 TEST1

Partner	IP Name	TCP Port	CTC Address	Buffer Size
NJE2	nje2.vm.example.com	175	N/A	4096
SNAVM5	vm.example.com	175	N/A	4096

```
NAME TEST1
IPADDRESS test1.example.com
QUEUE /var/spool/nje
CMDMAILBOX UDP 127.0.0.1 175
LOG /var/log/nje/nje.log
TABLE /etc/nje/nje.route
INFORM ROOT@TEST1
USEREXITS /etc/nje/file-exit.cf
MSGEXITS /etc/nje/msg-exit.cf
LLEVEL 1
DEFFORM STANDARD
DEFAULT-ROUTE NJE2
```

```
LINE 0 NJE2
TYPE UNIX_TCP
IPPORT 175
TCPNAME nje2.vm.example.com
BUFSIZE 4096
TIMEOUT 3
MAX-STREAMS 7
TCP-SIZE 8192
```

```
LINE 1 SNAVM5
TYPE UNIX_TCP
IPPORT 175
TCPNAME vm.example.com
```

```

BUFSIZE      4096
TIMEOUT      3
MAX-STREAMS  7
TCP-SIZE     8192

```

Figure 14: Configuration File for Node TEST1

2.5.5 VA1DEV10

Partner	IP Name	TCP Port	CTC Address	Buffer Size
NJE2	nje2.vm.example.com	175	N/A	4096

```

NAME          VA1DEV10
IPADDRESS     valdev10.example.com
QUEUE         /var/spool/nje
CMDMAILBOX    UDP 127.0.0.1 175
LOG           /var/log/nje/nje.log
TABLE         /etc/nje/nje.route
INFORM        ROOT@VA1DEV10
USEREXITS     /etc/nje/file-exit.cf
MSGEXITS      /etc/nje/msg-exit.cf
LLEVEL       1
DEFFORM       STANDARD
DEFAULT-ROUTE NJE2

LINE 0 NJE2
TYPE          UNIX_TCP
IPPORT        175
TCPNAME       nje2.vm.example.com
BUFSIZE       4096
TIMEOUT       3
MAX-STREAMS   7
TCP-SIZE      8192

```

Figure 15: Configuration for Node VA1DEV10

2.5.6 VSE27

Partner	IP Address	TCP Port	CTC Address	Buffer Size
NJE2	10.33.204.20	175	N/A	4096

2.5.6.1 Recommended PTFs

The following VSE PTFs are recommended.

- DY46418 for 6.7 (VSE/ESA 2.7)
- DY46426 for 6.6 (VSE/ESA 2.6)

- DY46427 for 7.1 (z/VSE 3.1).

2.5.6.2 PNODE Statements

The z/VSE node is connected over the Internet and uses NAT to allow connection between the two fire-walled systems.

```
V27NBT  PNODE  NODE=VSE27 ,                                -
          LOCAL=YES ,                                       -
          PORT=175
*
          PNODE  NODE=NJE2 , LOCAL=NO , IPHOSTAD=192.168.203.218 , -
          IPEXTRAD=010.033.204.020 , PORT=0175 ,           -
          AUTH=NET , BUFSIZE=32000
*
          END
```

Figure 16: POWER PNODE Macro Definitions

2.5.6.3 POWER Generation

The following macro parameters are inserted into the POWER generation deck.

```
NETPOWER POWER                                           -
...
          PNET=V27NBT ,                                     -
...

```

Figure 17: POWER Generation Macros

3 Using NJE/IP Bridge Services

Once installed and configured you can start the daemon using the “init” script:

```
/etc/init.d/njed start
Starting NJE daemon...NJE/IP Bridge licensed to: Example Company
Expires: 2010/12/31
Platform: s390
done.
```

3.1 Logging

By default, the NJE/IP Bridge daemon logs events and messages to `/var/log/nje/nje.log`. The logging level may be altered by the `ucp loglevel <n>` command. Level 5 is the most intensive tracing level; Level 1, the default, is the least detailed level.

3.2 Using the UCP Control Interface

The NJE/IP Bridge includes a command interface to query the status of the NJE services on the system, make changes to link states, and terminate the application. The `ucp` command is installed in `/usr/local/sbin` as it is intended as a system administrator command, not a general user interface.

Note: One aspect of this commands operation may cause confusion: the `ucp` command operates by sending a request command block to the main NJE protocol management daemon. The daemon is responsible for processing the command and reporting the results to the caller. The daemon is not aware from which terminal device the command was issued but it does know the id of the user. When it replies it will look up this user id in the table of logged on users and directs the results to all the terminal devices that this user is logged in to. If you had issued the “`su`” command to switch from another user in order to issue the command then you will be surprised to find no output appearing on the terminal. The `su` command does not update the table of logged on users in the way the NJE protocol daemon expects, so NJE does not know that your user id is operating from the terminal where the command was issued, and thus no output is displayed. The workaround for this is to suspend any non-login shells and issue the command from a shell associated with a login user.

3.2.1 Help Information

To obtain information about the operands to the `ucp` command use the help argument:

```
ucp help
HELP - Show this message
SHOW LINE/QUEUE - Show lines or queue status
START LINE n - Start a closed line
* START STREAM n LINE m - Start specific stream in active line
SHUT [ABORT] - Shutdown or abort the whole program
STOP LINE - Stop a line
FORCE LINE - Stop a line immediately
* STOP STREAM n LINE m - Stop a single stream in active line
```

```

* FORCE STREAM n LINE m - Stop immediately
QUEUE file-name [SIZE size] - To queue a file to send
RESCAN EXITS - Rescan file and message exits.
RESCAN QUEUE - Rescan queue and requeue files.
RESCAN ROUTE - Reopen route database.
DEBUG DUMP - Dump all lines buffers to logfile
% DEBUG RESCAN - Rescan queue and requeue files.
LOGLEVEL n - Set the loglevel to N
ROUTE xxx TO yyy - Change the routing table.
    Route to OFF will delete the route entry.
GONE username LoginDirectory - Add username to gone list
UNGONE username - Remove a user from the Gone list.
EXIT/QUIT - Exit this utility.
* - Not yet implemented, % - obsoleted

```

Figure 18: Example Output from UCP Help Command

3.2.2 Querying Link/Line State

To display the state of each of the lines defined on the node use the “show line” arguments:

```

ucp show line
NJE2: NJE/IP Bridge version 1.0(sna)/200511022000, Lines status:
NJE2: Line.0 VSE27 (0x439a48)  0 (Q=0000) ACTIVE      TCP
NJE2:  Bufinfo: InAge=23.319s, RecvSize=0, XmitAge=24.851s, XmitSize=0
NJE2:  Flags: AUTO_RESTART|DEFERRED_OK|RELIABLE|RESET_BCB|VMNET_FORMAT|XMIT_QUEUE
NJE2:  1 streams in service.  WrSum: 1fil/1659699B  RdSum: 1fil/2366615B
NJE2: Line.1 SNAVM5 (0x460348)  0 (Q=0000) ACTIVE      TCP
NJE2:  Bufinfo: InAge=18.120s, RecvSize=0, XmitAge=18.120s, XmitSize=0
NJE2:  Flags: AUTO_RESTART|DEFERRED_OK|RELIABLE|RESET_BCB|VMNET_FORMAT|XMIT_QUEUE
NJE2:  7 streams in service.  WrSum: 0fil/4399346B  RdSum: 0fil/4384771B
NJE2: Line.2 RSCS2  (0x486c48)  0 (Q=0000) ACTIVE      CTC
NJE2:  Bufinfo: InAge=1.494s, RecvSize=6, XmitAge=515347.294s, XmitSize=0
NJE2:  Flags: AUTO_RESTART|DEFERRED_OK|RELIABLE|RESET_BCB|SLOW_ILEASE|XMIT_QUEUE
NJE2:  2 streams in service.  WrSum: 0fil/310B  RdSum: 2fil/2793544B
NJE2: Line.3 VALDEV10 (0x44cec8)  0 (Q=0000) ACTIVE      TCP
NJE2:  Bufinfo: InAge=24.848s, RecvSize=0, XmitAge=24.851s, XmitSize=0
NJE2:  Flags:
AUTO_RESTART|DEFERRED_OK|RELIABLE|RESET_BCB|SLOW_ILEASE|VMNET_FORMAT|XMIT_QUEUE
NJE2:  7 streams in service.  WrSum: 1fil/1743081B  RdSum: 3fil/5222288B
NJE2: Line.4 TEST1 (0x4737c8)  2 (Q=0000) Retry      TCP
NJE2:  Bufinfo: InAge=1209479.239s, RecvSize=0, XmitAge=1209180.335s, XmitSize=24
NJE2:  Flags:
AUTO_RESTART|CALL_ACK|DEFERRED_OK|RELIABLE|VMNET_FORMAT|WAIT_V_A_BIT|XMIT_QUEUE
NJE2: End of Q SYS display

```

Figure 19: Sample Output from UCP SHOW LINE

3.3 Obtaining Status Information from Other Nodes

It is possible to query the state of other nodes in the collection by using the `send -c` command.

3.3.1 Querying a Remote RSCS Node

To interrogate a remote RSCS system, use the commands valid for that subsystem. For example, to query the system use the `q sys` command

```
send -c rscs2 q sys
RSCS2: Link                               Line
RSCS2: Name      Status      Type      Addr LU Name  Logmode  Queueing
RSCS2: NJE2      active      NJE       0800 ...      ...      priority
RSCS2: 1 link found
```

Figure 20: Sample Remote RSCS QUERY SYSTEM Response

3.3.2 Querying Other NJE/IP Bridge Nodes

Other NJE/IP Bridge nodes respond to the following commands when executed from remote nodes. Sending the `HELP` command produces the following output:

```
send -c test1 help
TEST1: Commands available for the NJE IP Bridge:
TEST1:   Help / HILfe / SOS - Three commands to ask for this info
TEST1:   Query SYStem      - Show line status summary report, and activity
TEST1:   Query SYStem S    - Same as "Q SYS", but with extra activity data
TEST1:   Query STATs      - Show line statistics
TEST1:   Query Nodename    - Show the route entry to that node
TEST1:   Query linkname A/F - Available via Query SYStem
TEST1:   Query linkname Q  - Show 30 first files in queue on the link
TEST1:   CPQuery Names     - List all users logged on
TEST1:   CPQuery User      - Tell how many users are logged on
TEST1:   CPQuery User username - Look for a specific username
TEST1:   CPQuery LOG       - Send the WELCOME message
TEST1:   CPQuery CPU, CPLEVEL, IND, T - Machine type and time
TEST1:   MSG node user text.. - A way to send a NMR to some node via msg relay.
TEST1:   CMD node text..    - A way to send a NMR to some node via cmd relay.
TEST1: This system has also an NMR responding server:
TEST1:   ECHO@NJENODE1     -- throws the NMRs back to you.
TEST1: If you send files to the LPR user, they will be printed:
TEST1:   LPR@NJENODE1      -- files sent to LPR go to the system default print
queue.
```

Figure 21: NJE/IP Bridge Default Response To The Remote HELP Command

Many NJE/IP Bridge commands mirror those defined by RSCS. A useful example is the `CPQuery` command (`CPQ`). `CPQUERY CPU` displays the platform that is executing the NJE/IP Bridge application. For example:

```
send -c test1 cpq cpu
TEST1: CPU: i686
```

Figure 22: NJE/IP Bridge Response to CPQUERY CPU

CPQUERY USERS displays how many users are logged on to the remote system: Example output might appear similar to this:

```
send -c test1 cpq user
TEST1: CPQ:      0 USERS LOGGED ON
```

Figure 23: Sample Output from CPQUERY USER

To determine what operating system is running on the remote node, the CPQUERY CPLEVEL command can be used. For example, here is the response from a Linux system running the NJE/IP Bridge:

```
send -c test1 cpq cplevel
TEST1: Version: Linux-2.6.8-2-686-smp
```

Figure 24: Sample Output from CPQUERY CPLEVEL

3.4 Submitting Jobs from Linux

One of the most interesting and important capabilities of the NJE/IP Bridge is the ability to submit jobs from the UNIX/Linux/Windows world to the “classic” zSeries operating systems. In our example, we submit the following example JCL to a VSE system running on node VSE27:

```
* $$ JOB JNM=PRINTLOG,CLASS=P,LDEST=(,NEALE),XDEST=VSE27,NTFY=YES
* $$ LST CLASS=O
// JOB TEST NOTIFY
// EXEC PRINTLOG,SIZE=PRINTLOG,PARM='F4'
/*
/&
* $$ EOJ
```

Figure 25: Sample VSE JECL To Submit to Node VSE27

The NJE/IP Bridge submit command is used to submit this job to the target system:

```
submit @vse27 test.jcl
VSE27: 1Q5DI EXECUTION COMPLETED FOR PRINTLOG 00575 ON VSE27, RC=****,
TIME=16:28:56
NJE2: FILE (0575) spooled to NEALE -- origin NJE2(NEALE) 11/18/05 16:16:06 EST
NJE2: FILE (0575) spooled to NEALE -- origin NJE2(NEALE) 11/18/05 16:16:06 EST
VSE27: 1RA0I OUTPUT PRINTLOG 00575(00133) TRANSMITTED TO NJE2 FOR NJE2 O-TR1
```

Figure 26: Example Job Submission and Messages

The job is submitted to the POWER reader, and executed normally. When the job has completed, the output and results of the jobs are automatically transmitted back to the NJE destination indicated in the POWER JOB card, in this example, user NEALE at node NJE2. The output from the job will be placed in your “reader queue”; a temporary spool area used to hold NJE input and output files. Use the qrdcr command to interrogate the contents of your NJE spool queues. For example:

```

qrdr
QRDR: Spool dir:  `/var/spool/nje/NEALE/'
From:           To:           FName:   FExt:   Type  Form:   SpoolID
NEALE@NJE2     NEALE@NJE2     PRINTLOG OUTPUT  PRT  O     0112
Found 1 file.

```

Figure 27: Sample QRDR Output for Example Job

The peek command allows you to preview files in your reader. Peek displays a NJE spool file on your terminal. :

```

peek 1
// JOB TEST NOTIFY                                DATE 11/18/2005, CLOCK 16/28/53
// EXEC PRINTLOG,SIZE=PRINTLOG,PARM='F4'
1S54I PHASE PRINTLOG IS TO BE FETCHED FROM IJSYSRS.SYSLIB
OPTIONS SPECIFIED:  F4                                11/18/2005 PAGE      1

COUNT      MESSAGE TEXT                                CLOCK   DATE      CONSOLE

1 ***** LOGGING RESUMED AFTER VSE SYSTEM RE-IPL ***** 10:56:40 09/21/2005
2 F4 0004 // JOB STARTF4                                10:57:24 09/21/2005
      DATE 09/21/2005, CLOCK 10/57/24                  10:57:24 09/21/2005
3 F4 0004 1N90I EOP WAS FORCED BY EOJ                   10:58:10 09/21/2005
4 F4 0004 EOJ STARTF4                                    10:58:10 09/21/2005
      DATE 09/21/2005, CLOCK 10/58/10, DURATION 00/00/46 10:58:10 09/21/2005
5 F4 0001 1Q34I F4 WAITING FOR WORK                    10:58:10 09/21/2005

```

Figure 28: Output of PEEK for NJE Spool File 1

To transfer a file from the NJE spool area to your current working directory (to “receive” the file in NJE parlance) use the receive command:

```

receive 112
ls -l printlog.output
-rw-r--r-- 1 neale users 10976 2005-11-18 16:28 printlog.output

```

Figure 29: Using RECEIVE to Copy A NJE Spool File To The Current Directory

To discard a NJE spool file without receiving it, use the purge command.

3.5 Sending Commands to VSE

In our example network, it may be useful to send commands to VSE. As shown earlier, the send command is useful for this purpose. For example:

```

send -c @VSE27 d a
VSE27: 1R48I C-RV ,TCP, AWAITING NODE=BSIVM43
VSE27: 1R48I C-RV ,TCP, AWAITING NODE=BSIVSE26
VSE27: 1R48I C-RV ,TCP, AWAITING NODE=BSIVSE31
VSE27: 1R48I C-RV ,TCP, PROCESSING NODE=NJE2
VSE27: 1R48I C-RV ,TCP, AWAITING NODE=BSIJCB
VSE27: 1R48I C-RV ,TCP, AWAITING NODE=TEIVM43
VSE27: 1R48I BG,FEC,AOI,, INACTIVE,

```

```

VSE27: 1R48I F2,FEC,L2,, CICSICCF,00573,2
VSE27: 1R48I F3,FEC,K3,, VTAMSTRT,00025,3
VSE27: 1R48I F4,FEC,J4,, INACTIVE,
VSE27: 1R48I F5,FEC,H5,, INACTIVE,
VSE27: 1R48I F6,FEC,M6,, INACTIVE,
VSE27: 1R48I F7,FEC,N7,, INACTIVE,
VSE27: 1R48I F8,FEC,P8,, INETLCS ,00111,8
VSE27: 1R48I F9,FEC,Q9,, INACTIVE,
VSE27: 1R48I FA,FEC,TV,, INACTIVE,
VSE27: 1R48I S1,FEC,S,, INETOSAX,00286,S
VSE27: 1R48I S2,FEC,S,, INETHIPR,00288,S
VSE27: 1R48I Z1,FEC,Z,, BSTTFTPS,00112,Z
VSE27: 1R48I Z2,FEC,Z,, FTSP30 ,00287,Z
VSE27: 1R48I Z3,FEC,Z,, FTSP20 ,00289,Z
VSE27: 1R48I F3,FEE,,, VTAMSTRT,00025,A 21 LINES SPOOLED,QNUM=01886
VSE27: 1R48I F8,FEE,,, INETLCS ,00111,O 79 LINES SPOOLED,QNUM=01884
VSE27: 1R48I S2,FEE,,, INETHIPR,00288,O 53 LINES SPOOLED,QNUM=01883
VSE27: 1R48I S1,FEE,,, INETOSAX,00286,O 53 LINES SPOOLED,QNUM=01882
VSE27: 1R48I Z3,FEE,,, FTSP20 ,00289,O 25 LINES SPOOLED,QNUM=01879
VSE27: 1R48I Z3,FED,,, FTSP20 ,00289,O 0 CARDS SPOOLED,QNUM=01878
VSE27: 1R48I Z2,FEE,,, FTSP30 ,00287,O 25 LINES SPOOLED,QNUM=01877
VSE27: 1R48I Z2,FED,,, FTSP30 ,00287,O 0 CARDS SPOOLED,QNUM=01876
VSE27: 1R48I Z1,FEE,,, BSTTFTPS,00112,O 18 LINES SPOOLED,QNUM=01872
VSE27: 1R48I Z1,FED,,, BSTTFTPS,00112,O 0 CARDS SPOOLED,QNUM=01871
VSE27: 1R48I R1,FEC,R,, ITAMTEST,00572,R
VSE27: 1R48I R1,FEE,,, ITAMTEST,00572,O 170 LINES SPOOLED,QNUM=01881
VSE27: 1R48I F2,FEE,,, CICSICCF,00573,A 1864 LINES SPOOLED,QNUM=01870
VSE27: 1R48I RDR,00C,A,,
VSE27: 1R48I PUN,00D,O,VM, INACTIVE,
VSE27: 1R48I LST,00E,O,1,VM, INACTIVE,

```

Figure 30: Response from Remote POWER 'D A' Command

3.6 Sending Commands to z/VM

Similar to sending commands to POWER, RSCS remote commands may be sent to RSCS nodes:

```
send -c @snavm5 q sys
```

```

SNAVM5: Link                               Line
SNAVM5: Name      Status      Type      Addr LU Name  Logmode  Queueing
SNAVM5: PUCC      inactive  TCPNJE   0000 ...      ...      size
SNAVM5: MARIST    connect  TCPNJE   0000 ...      ...      size
SNAVM5: FSFZVM51  active   TCPNJE   0000 ...      ...      size
SNAVM5: FC3       inactive  TCPNJE   0000 ...      ...      size
SNAVM5: NJE2      connect  TCPNJE   0000 ...      ...      size
SNAVM5: D26TEST   active   TCPNJE   0000 ...      ...      size
SNAVM5: 6 links found

```

Figure 31: Response to Remote RSCS Command

Interactive messages can be sent to any NJE destination, similar to this example of sending a message to a user on a z/VM system:

```
send -m neale@snavm5 Hello VM!
SNAVM5: DMTRGX332E NEALE not receiving
```

Figure 32: Interactive Message Example

3.7 Sending Files via NJE

Modelled on the “sendfile” command of CMS, the NJE/IP Bridge `sendfile` command allows you to send files to an NJE-capable system. In this instance the file is sent to a user on a z/VM system.

```
sendfile neale@snavm5 printlog.output
SNAVM5: DMTAXM104I File (0134) spooled to NEALE -- origin NJE2(NEALE) 11/18/05
17:35:37 CDT
```

Figure 33: SENDFILE Example

Just as you used `qrdr` on the NJE/IP Bridge node, you use the `QUERY RDR ALL *` command on z/VM to check the contents of your reader queue.

```
query rdr all *
ORIGINID FILE CLASS RECORDS CPY HOLD DATE TIME NAME TYPE DIST
RSCS 0664 A PUN 00000142 001 NONE 11/18 16:33:48 PRINTLOG OUTPUT NEALE
```

Figure 34: Displaying Incoming Virtual Reader Spool Files on VM

The CP tag `query` command will show you where this file originated:

```
tag query file 664
FILE (0134) ORIGIN NJE2 NEALE 11/18/05 17:35:38 CDT JOBID 00134
```

Figure 35: Displaying File Tags on VM

Just like on the NJE/IP Bridge node, the `peek` command allows you to preview files in your reader queue without moving them from the queue or deleting them:

```
peek 664

0664 PEEK A0 V 132 Trunc=132 Size=109 Line=0 Col=1 Alt=0
File A.PRINTLOG.OUTPUT from NEALE at NJE2 Format is NETDATA.
* * * Top of File * * *

"// JOB TEST NOTIFY DATE 11/18/
2005, CLOCK 16/28/53
// EXEC PRINTLOG,SIZE=PRINTLOG,PARM='F4'
"1S54I PHASE PRINTLOG IS TO BE FETCHED FROM IJSYSRS.SYSLIB"OPTIONS SPECIFIED:
F4

11/18/2005 PAGE 1

COUNT MESSAGE TEXT
```


3.9 Using NQS to Execute Jobs

Just as it is possible to initiate jobs from NJE/IP Bridge nodes to the traditional NJE operating systems, NJE/IP Bridge provides interfaces to NQS, allowing the ability to execute jobs on the Unix or Linux-based platforms supported by the NJE/IP Bridge. Compared to other batch systems NQS is relatively primitive but can be used as the basis of a useful job execution environment. NQS expects scripts to be sent to it and for it to return the output of `stdout` and `stderr` to the user. There is no file staging capability (yet). All execution is performed under a user id `njeanon` that is created when NQS is installed.

NQS is licensed under the GPL and so its source is freely available for people to make enhancements. Some I think are important include:

- File staging
- Restoring the environment to the state before the last job was executed
- A “padded” cell environment for job execution

3.9.1 Sample Job

For our example, the following script was used to illustrate the ability to remotely execute tasks.

```
#!/bin/sh
id
hostname
hostname --domain
uname -a
date
env
cat /proc/cpuinfo
```

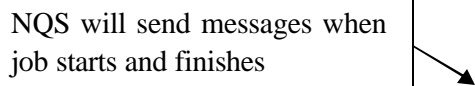
Figure 39: Example Shell Script For Remote NQS Execution

3.9.2 Submitting a Script For Execution

Submitting a job for execution is a matter of sending the script to the user “JOB” on the remote NJE/IP Bridge node.

```
sendfile test sh job at nje2
File TEST SH A1 sent to JOB at NJE2 on 11/21/05 12:40:32
Ready;
```

NQS will send messages when
job starts and finishes




```
From NJE2(NJEANON): NQS request: R0188 (0.NJE2.vm.example.com) beg  
ing at 12:40 on NJE2.vm.example.com
```

```

From NJE2(NJEANON): NQS request: R0188 (0.NJE2.vm.example.com) end
at 12:40 on NJE2.vm.example.com
From NJE2(NJEANON): NQS request: R0188 (0.NJE2.vm.example.com) end
at 12:40 on NJE2.vm.example.com

```

NQS will return the stdout
and stderr results



```

RDR FILE 0677 SENT FROM RSCS      PRT WAS 0189 RECS 0000 CPY  001 A NOHOLD NO
DMTAXM104I File (0036) spooled to NEALE -- origin NJE2(ROOT) 11/21/05 13
:08 CDT
RDR FILE 0678 SENT FROM RSCS      PRT WAS 0190 RECS 0062 CPY  001 A NOHOLD NO
DMTAXM104I File (0037) spooled to NEALE -- origin NJE2(ROOT) 11/21/05 13
:08 CDT

```

Figure 40: Example Job Submission from VM to Linux Node Running NJE/IP Bridge and NQS

When the job is complete, the ‘stdout’ and ‘stderr’ files from the job are returned to the originating user. As shown below, two files are returned to the originating user, with the file type set to the NQS job number.

```

q r all *
ORIGINID FILE CLASS RECORDS  CPY HOLD DATE  TIME      NAME      TYPE      DIST
RSCS      0677 A PRT 000000000 001 NONE 11/21 12:40:44 NJE_ERR   00000000 ROOT
RSCS      0678 A PRT 000000062 001 NONE 11/21 12:40:44 NJE_OUT   00000000 ROOT

```

Figure 41: Results of a Remotely Executed NQS Job

```

peek 678
0678      PEEK      A0  V 132  Trunc=132 Size=62 Line=0 Col=1 Alt=0
File NJE_OUT 00000000 from ROOT at NJE2 Format is PRINT.
* * * Top of File * * *
uid=1001(njeanon) gid=1003(njeanon) groups=1003(njeanon)
nje2
vm.example.com
Linux nje2 2.6.5-7.191-s390 #1 SMP Tue Jun 28 14:58:56 UTC 2005 s390 s390 s39
0 GNU/Linux
Mon Nov 21 12:40:42 EST 2005
LESSKEY=/etc/lesskey.bin
MANPATH=/usr/local/man:/usr/local/share/man:/usr/share/man:/usr/X11R6/man:/opt
/g
nome/share/man
INFODIR=/usr/local/info:/usr/share/info:/usr/info
NNTPSERVER=news
HOSTNAME=nje2
:
:
:
DEBUG=1
LESSCLOSE=lessclose.sh %s %s
COLORTERM=1
_=/usr/bin/env

```

```
vendor_id      : IBM/S390
# processors   : 1
bogomips per cpu: 412.05
processor 0: version = FF,  identification = 01131C,  machine = 7060
1= Help       2= Add line  3= Quit     4= Tab      5= Clocate   6= ?/Change
7= Backward   8= Forward   9= Receive 10= Rgtright 11= Spltjoin 12= Cursor

====>

X E D I T  1 File
```

Figure 42: Preview Output from Remotely Executed Job

4 Summary

In this document we have demonstrated setting up a basic network, configuring the links and routing between nodes, and some of the more useful features of the NJE/IP Bridge. For additional information consult the IBM NJE documentation listed below, or contact Sine Nomine Associates or your distributor for additional support.

4.1 IBM Documentation

- “z/OS Network Job Entry – Formats and Protocols”, SA22-7539-00.