

# Network Job Entry (NJE)

## for the 21<sup>ST</sup> Century

BY NEALE FERGUSON

Network Job Entry (NJE) is a protocol for transmitting jobs, SYSOUT data sets, operator commands and messages, and job accounting information from one computing system to another.

IBM has provided implementations of this protocol on all its mainframe systems: RSCS for z/VM, JES2 and JES3 for z/OS, and POWER for z/VSE. There's also an implementation for iSeries OS. NJE is supported as an application over different transports, including native BSC and CTC communications, SNA networks and TCP/IP (VM, VSE, and JES2 at z/OS 1.7 only).

Although a relatively "old" protocol, there are several significant reasons to extend the facilities provided by NJE outside the mainframe and down to the distributed systems:

- Clean bi-directional integration of programmable workstations with mainframe data transfer
- Leverage Linux-based development tooling and skills in tandem with mainframe services (the right tool for the right job)
- Move data between IBM and non-IBM environments without complex automation requirements
- Eliminate SNA as a requirement to run NJE (required for pre-1.7 z/OS systems).

This article provides examples of several different scenarios for using NJE in a distributed environment, including:

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

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

receive 664
File PRINTLOG OUTPUT A1 created from PRINTLOG.OUTPUT received from NEALE at VM
```

Figure 1: SENDFILE Commands and Response Example

```
* $$ JOB JNM=PRINTLOG,CLASS=P,LDEST=(,NEALE),XDEST=VSE27,NTFY=YES
* $$ LST CLASS=0
// JOB TEST NOTIFY
// EXEC PRINTLOG,SIZE=PRINTLOG,PARM='F4'
/*
/&
* $$ EOJ

submit @vse27 test.jcl
VSE27: IQSDI 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

qrdr
QRDR: Spool dir: `~/var/spool/nje/NEALE/'
From: To: FName: FExt: Type Form: SpoolID
NEALE@NJE2 NEALE@NJE2 PRINTLOG OUTPUT PRT 0 0112
Found 1 file.

peek 1
// JOB TEST NOTIFY DATE 11/18/2005, CLOCK 16/28/53
// EXEC PRINTLOG,SIZE=PRINTLOG,PARM='F4'
1554I 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 2: Sample VSE JECL, Job Submission, and Output Return

- Unattended file transfer: Being able to deliver files without having to write elaborate scripts controlling File Transfer Protocol (FTP)
- Development workstation: Coding programs and jobs using the Integrated Development Environment (IDE) of choice, submitting the jobs, and receiving the job output
- Automated delivery of job output to people, programs and devices
- Remote job execution on the distributed system from jobs submitted from "classic" mainframe systems.

### An Introduction to NJE

NJE is primarily an application facilitating the transfer of commands, messages, programs, and jobs among different computing systems in a network. *The NJE Formats and Protocols manual (SC23-0070-03)* describes NJE as follows: An NJE network is a group of two or more complexes or systems that communicate with each other. An NJE network is comprised of nodes that can transmit or receive a unit of work. The nodes in an NJE network use protocols to communicate with each other. Protocols are rules a node uses to:

- Become part of an NJE network
- Receive a unit of work
- Send a unit of work
- Indicate it was removed from the network.

NJE is an extension of Remote Job Entry (RJE), which was also developed by IBM. Before the widespread use of the Internet, NJE networks such as BITNET formed the basis of the largest collection of computers yet known. During its lifetime, NJE has accrued a heritage of tools, infrastructure, and procedures that still have merit in today's environment.

Let's now consider how the NJE facility can be used to establish communications, controls, and remote execution facilities between nodes running on the classic mainframe operating systems and those on the distributed platforms. Throughout this article the examples reference one of the available implementations of NJE on distributed systems.

### Automated File Transfer

Automated file transfer between systems using FTP scripts is tedious and error-prone. When it works, it's fine, but as soon as something breaks, everything

crashes with no simple method for detecting the problem and recovering appropriately. Commands such as FTP were designed for interactive transfers and require things such as retry and recovery to be bolted on.

NJE was designed to handle asynchronous file transfers. It copes well with queuing and retrying. It will alert operations when anything goes wrong, so your existing automation tools can intervene.

The NJE service running on the

non-classic systems also has a sendfile command that lets you send files to an NJE-capable system. NJE interactions can be scripted simply because of its single command interface, store-and-forward mechanism, extensive logging support and use of return codes to indicate success or failure.

Files can be delivered to individual users, to devices, or to a program that can act on the contents of the file. Figure 1 shows a trivial file transfer from Linux to z/VM.

## Do you know who F\*\*'d your data?

FTP is a dirty word in some circles. Without you knowing, personnel, patient, payroll or other confidential corporate information can be retrieved from your company and sent *anywhere!*

This can all be achieved with the universal File Transfer Protocol (FTP), often used, easily abused but rarely managed.

## FREE FTP ACTIVITY CHECKER:

Download your free copy of the z/OS FTP activity checker from [www.willdata.com/zj.htm](http://www.willdata.com/zj.htm)

*ftpalert* from William Data Systems lets you control who can ftp files to and from your zSeries eServer. *ftpalert* installs in minutes, secures all FTP activity and, to satisfy regulatory compliance, incorporates comprehensive, real time audit and logging facilities for all FTP usage.

Evaluate *ftpalert* today. You have nothing to lose but your data!

*ftpalert*


Call (877) 723-0008 or (703) 674-2200

or visit [www.willdata.com/zj.htm](http://www.willdata.com/zj.htm)

IBM and ServerProven are trademarks of International Business Machines Corporation in the United States, other countries, or both.

NJE FACILITATES ROBUST SYSTEM MANAGEMENT AND PROVIDES A SET OF FUNCTIONS THAT CAN BE EXPLOITED TO HELP STANDARDIZE THE AUTOMATION OF OPERATIONS THAT ARE TYPICALLY INSTALLATION-UNIQUE. THESE FACILITIES CAN BE EXTENDED BEYOND THE MAINFRAME AND ACT AS A "GLUE" BETWEEN THE CLASSIC MAINFRAME SYSTEMS AND THE DISTRIBUTED SYSTEMS.

### Submitting Jobs From Linux

Job streams can be created on the non-classic platform and submitted using the submit command.

An NJE job contains either SYSIN or SYSOUT data and the control records used to identify the data being transmitted. A node uses these control records to transmit an NJE job:

- A job header containing information that affects scheduling, security and accounting
- A data set header if the data is SYSOUT or optionally SYSIN data
- A job trailer containing information that affects output formatting and accounting.

All NJE jobs contain both a job header and job trailer.

Under VSE, the LDEST operand of the JOB JECL statement can be used to cause the output to be returned to the submitting node (see Figure 2). NJE

wraps the JCL with the required control records. The job is transmitted to VSE, where POWER will schedule the job for execution. The job is executed and output is returned via the XMIT queue. The output from the job will be placed in the submitter's "reader queue," a temporary spool area used to hold NJE input and output files. You can interrogate the contents of your NJE spool queues using the:

- *Qrdr* command to query what files are available to you
- *Peek* command to preview files in your reader
- *Receive* command to transfer a file from the NJE spool area to your current working directory
- *Purge* command to discard an NJE spool file without receiving it.

### Redirecting Files to Programs or Devices

Besides simply routing or returning files to the "reader" of an NJE user, you can have the NJE system invoke a script or a program that acts on the file. For example, to print the output file directly on a local or network-attached printer under the control of Common Unified Printing System (CUPS), the NJE system can be configured to intercept files routed to a pseudo-user such as "LPR." For example, under VSE, the LDEST operand can be set to "LPR" rather than a user ID. When NJE receives a file with this destination, it could invoke the *lpr* command to send the file to CUPS for post-processing in such ways as:

- Directing it to a real printer
- Sending it as a fax
- Converting it to a PDF and storing it for later viewing (or staging it to CD or DVD)
- Sending it as an attachment to a prepared electronic mailing list.

### Executing Batch Jobs on Linux

NJE provides interfaces to an open source batch execution facility known as NQS. This allows execution of jobs on the Unix or Linux-based platforms the NJE/IP Bridge supports. 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's no file staging capability (yet). Figure 3 illustrates remote task execution. This file can be created on any

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

sendfile test sh job at nje2
File TEST SH A1 sent to JOB at NJE2 on 11/21/05 12:40:32
Ready;
  From NJE2(NJEANON): NQS request: R0188 (0.NJE2.vm.example.com) beginning 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

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

q r all *
ORIGINID  FILE CLASS  RECORDS  CPY  HOLD DATE  TIME      NAME      TYPE      DIST
RSCS      0677  A  PRT  00000000  001  NONE  11/21  12:40:44  NJE_ERR  00000000  ROOT
RSCS      0678  A  PRT  00000062  001  NONE  11/21  12:40:44  NJE_OUT  00000000  ROOT

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
```

Figure 3: Job Submission on Linux and Return of Results

NJE node. In this example, we've used our VM system to prepare and submit the job.

Submitting a job for execution is a matter of sending the script to the user "JOB" on the remote NJE node. When the job is complete, the *stdout* and *stderr* files from the job are returned to the originating user. The figure shows two files returned to the originating user, with the file type set to the NQS job number and using the *peek* command to display the contents of *stdout*. The

platform tools can then manipulate the returned files.

### Sending Commands and Messages Between Nodes

NJE provides the facility to send commands and messages between nodes. For example, you can display POWER queues on a VSE system from your Linux system. Figure 4 shows the "D A" command initiated on Linux and executed on VSE. Similarly, users residing on NJE nodes may exchange mes-

sages with each other. Unlike file transmissions, there's no store-and-forward mechanism. Messages that can't be delivered immediately are discarded. Tools such as NetView, NetMaster, and Tivoli's management suite can be used to centralize management of all NJE nodes.

### Bringing It All Together

NJE continues to fill an important role in the modern IT infrastructure and its benefits can be extended to distributed environments. NJE facilitates robust system management and provides a set of functions that can be exploited to help standardize the automation of operations that are typically installation-unique. **Z**

### About the Author

Neale Ferguson has more than 25 years of experience with IBM mainframe systems. He has been involved with Linux on the mainframe since the beginning. Originally from Sydney, Australia, he now works with Sine Nomine Associates in Northern Virginia.  
 e-Mail: [neale@sinenomine.net](mailto:neale@sinenomine.net)  
 Website: [www.sinenomine.net](http://www.sinenomine.net)

```

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,A0I,, INACTIVE,
VSE27: 1R48I F2,FEC,L2,, CICSICCF.00573,2
VSE27: 1R48I F3,FEC,K3,, VTAMSTR,00025,3
VSE27: 1R48I F4,FEC,J4,, INACTIVE,
:
:
VSE27: 1R48I RDR,00C,A,,
VSE27: 1R48I PUN,00D,0,VM, INACTIVE,
VSE27: 1R48I LST,00E,0,1,VM, INACTIVE,
  
```

Figure 4: Response From Remote POWER 'D A' Command

**THINKING ABOUT DATA INTEGRATION ?**

**THINK**





**Connecting more than 40 Data Sources**

- ✓ for SOA and Web integration
- ✓ for SQL access
- ✓ for simplified reporting
- ✓ for Mainframe, Linux OpenVMS, UNIX and Windows databases

Take a 2 minute test drive to see how easy data integration can be

[www.connx.com/testdrive](http://www.connx.com/testdrive)  
**888.88CONNX**

CONNX Solutions, Inc.  
 2039 152nd Avenue NE  
 Redmond, WA 98052