

# NJE/IP Bridge Installation and Configuration

Sine Nomine Associates

July 1, 2006

# Contents

<b>1</b>	<b>Introduction</b>	<b>6</b>
1.1	Purpose of the Product . . . . .	6
1.2	Audience . . . . .	6
1.3	Revision History . . . . .	6
1.4	Features . . . . .	7
<b>2</b>	<b>Preparing for Installation</b>	<b>8</b>
2.1	LPAR Appliance . . . . .	8
2.2	z/VM Guest Appliance . . . . .	9
2.3	Open Systems . . . . .	10
2.3.1	Supported Platforms . . . . .	10
2.3.2	Supported OS Distributions . . . . .	10
2.3.3	Prerequisite Packages . . . . .	11
2.4	Windows XP . . . . .	11
2.4.1	Pre-Requisite Package . . . . .	11
2.5	Product Enablement Keys . . . . .	11
<b>3</b>	<b>Installing the NJE/IP Bridge</b>	<b>12</b>
3.1	LPAR Appliance Installation . . . . .	12
3.2	Linux on S/390 z/VM Appliance Installation . . . . .	13
3.3	Open Systems Installation . . . . .	14
3.3.1	Linux on System/390 or zSeries . . . . .	14
3.3.1.1	For RPM-based (SuSE, RedHat, TurboLinux) systems . . . . .	14

<i>CONTENTS</i>	2
3.3.1.2 For Debian systems . . . . .	15
3.3.1.3 For Slackware systems or systems without a package manager . . . . .	15
3.3.1.4 CTC Configuration . . . . .	15
3.3.2 Linux on Intel and Other Supported Architectures . . . . .	15
3.3.2.1 For RPM-based (SuSE, RedHat, TurboLinux) systems	15
3.3.2.2 For Debian systems . . . . .	16
3.3.2.3 For Slackware systems or systems without a package manager . . . . .	16
3.3.3 Linux on other Architectures . . . . .	16
3.3.4 Solaris . . . . .	16
3.3.5 AIX . . . . .	17
3.3.6 HP/UX . . . . .	17
3.4 Windows XP Installation . . . . .	17
3.4.1 MKS Toolkit . . . . .	17
3.4.2 NJE Installation . . . . .	22
<b>4 Configuration of the NJE/IP Bridge</b>	<b>27</b>
4.1 Concepts . . . . .	27
4.1.1 Links . . . . .	27
4.1.2 Routing . . . . .	28
4.1.3 Userid Mapping . . . . .	28
4.1.4 Printing . . . . .	28
4.1.5 NJE-to-SMTP Gateway . . . . .	29
4.1.6 Redirecting Messages to Programs . . . . .	29
4.1.7 Unknown User Handling . . . . .	29
4.1.8 Topological Design . . . . .	29
4.2 Modifying NJE Configuration and Behavior . . . . .	30
4.2.1 nje.cf . . . . .	30
4.2.1.1 General Fields . . . . .	30
4.2.1.2 Line-specific statements . . . . .	32
4.2.2 file-exit.cf . . . . .	34

<i>CONTENTS</i>	3
4.2.3 msg-exit.cf . . . . .	35
4.3 Creating and Managing Links . . . . .	37
4.3.1 Connecting via CTC . . . . .	37
4.3.1.1 Necessary Information . . . . .	37
4.3.1.2 Defining a Link . . . . .	37
4.3.2 Connecting via TCPNJE . . . . .	38
4.3.2.1 Necessary Information . . . . .	38
4.3.2.2 Defining a Link . . . . .	39
4.4 Deleting Links . . . . .	39
4.5 Managing Link Traffic . . . . .	39
4.5.1 ucp . . . . .	39
4.5.2 Start/stop links . . . . .	41
4.5.3 Purge files on link . . . . .	41
<b>5 Verification of Installation for the NJE/IP Bridge</b>	<b>42</b>
5.1 Checking Default Configuration . . . . .	42
5.1.1 Location of Files on UNIX-like Systems . . . . .	42
5.1.2 Location of Files on Windows XP Systems . . . . .	42
5.1.3 Verification Procedure . . . . .	42
5.2 LPAR Appliance Verification . . . . .	43
5.2.1 Starting and Accessing the Appliance . . . . .	43
5.2.2 Starting the Bridge . . . . .	44
5.2.3 Testing Basic Functionality . . . . .	44
5.2.4 Stopping the Bridge . . . . .	45
5.2.5 Stopping the Appliance . . . . .	45
5.3 Linux on S/390 Guest Verification . . . . .	45
5.3.1 Starting the Bridge . . . . .	45
5.3.2 Testing Basic Functionality . . . . .	45
5.3.3 Stopping the Bridge . . . . .	46
5.4 Open Systems Verification . . . . .	46
5.4.1 Linux on Open Systems, Solaris, AIX, and HP/UX . . . . .	46
5.4.1.1 Starting the Bridge . . . . .	46

<i>CONTENTS</i>	4
5.4.1.2 Testing Basic Functionality . . . . .	47
5.4.1.3 Stopping the Bridge . . . . .	48
5.5 Windows Verification . . . . .	48
5.5.1 Starting the Bridge . . . . .	48
5.5.2 Start the NJE Console . . . . .	48
5.5.3 Testing Basic Functionality . . . . .	48
5.5.4 Stopping the Bridge . . . . .	49
5.5.5 Printing on Windows . . . . .	49
5.5.5.1 FORM Files . . . . .	49
5.5.5.2 Sample Forms . . . . .	50
5.5.5.3 FORM Schema . . . . .	50
5.5.6 file-exit Configuration . . . . .	50
<b>6 User Commands</b>	<b>52</b>
6.1 send . . . . .	52
6.2 sendfile . . . . .	53
6.3 njeprint . . . . .	54
6.4 submit . . . . .	54
6.5 qrdr . . . . .	54
6.6 receive . . . . .	54
6.7 peek . . . . .	55
6.8 purge . . . . .	55
6.9 ygone . . . . .	56
<b>7 Problem Determination</b>	<b>57</b>
7.1 Logging and Log Files . . . . .	57
7.2 Setting Debugging Level . . . . .	57
7.3 What to Capture . . . . .	57
7.4 Turning Debugging Off . . . . .	57
7.5 Reporting Problems . . . . .	58
7.5.1 If you have support . . . . .	58
7.5.2 If you do not have support . . . . .	58

<i>CONTENTS</i>	5
<b>8 References</b>	<b>59</b>
8.1 NJE References . . . . .	59
8.2 BSD License Text . . . . .	60
8.3 Support Information . . . . .	61

# Chapter 1

## Introduction

### 1.1 Purpose of the Product

The NJE/IP Bridge is a product designed to enable TCPNJE connections between traditional IBM NJE hosts and Open Systems. It supports communication between any of NJE's supported platforms (broadly, almost any system with a POSIX.1 interface) and RSCS, JES2, JES3, VSE/POWER, or other NJE/IP Bridge systems.

The NJE/IP Bridge is designed to allow file transfer and remote job submission to traditional mainframe hosts from Open Systems platforms, as well as to enable submission to Open Systems platforms from mainframe hosts. It is particularly useful for enabling printing from mainframe hosts to printers in an Open Systems environment. The NJE/IP Bridge provides a mechanism for batch job control, and a general mechanism for messaging between users/services/virtual machines on either end of the link, which makes it well-suited for the design of a

In addition to supporting NJE-over-TCP, if the NJE/IP Bridge is running on Linux on the System/390, it can use real or virtual CTCs to enable direct CTC communication with any of the above systems, thus extending TCPNJE functionality to systems that do not directly support it.

### 1.2 Audience

This document is intended for system programmers/system administrators who will be tasked with installing and maintaining the NJE/IP Bridge.

### 1.3 Revision History

- September 23, 2005: version 1.0 (initial version)

- October 1, 2005: version 1.0.1 (additional LPAR and VM appliance installation information)
- November 10, 2005: version 1.0.2 (finalized CTC device documentation)

## 1.4 Features

The NJE/IP Bridge provides:

- The ability to define links either via TCP or via a CTC (on System/390 or zSeries only) point-to-point connection
- A store-and-forward file transfer service
- Interactive messaging
- The ability to take arbitrary action via user-defined exits on receipt of a message; this can include mapping of “pseudo-users” to particular services.
- An NJE-to-SMTP mail gateway
- If the POSIX system is equipped with NQS, the ability to accept remotely-submitted jobs, execute them, and return output to arbitrary NJE destinations.

## Chapter 2

# Preparing for Installation

### 2.1 LPAR Appliance

The LPAR must be defined as follows:

- 1 CPU (Standard or IFL, and the CPU may be shared)
- 256 MB memory
- A 3270 console at address 001F
- An OSA device triple in QDIO Ethernet mode at 005A-C. This OSA must have a microcode level of at least Driver 3G-EC stream J11204, MCL032 (OSA level 3.33). If it does not, please upgrade the microcode level prior to installation. OSA in LCS mode is not supported.
- A CTC device at 0060
- A 3390-3 at address 0150
- A 3390-3 at address 0151
- A 3490 tape drive at address 0181 (temporary: only used for installation)

**The LPAR should not have any other disks or devices visible to it. Specifically, additional OSA addresses or tape drives may cause the installer to malfunction.**

The initial password for the `root` user in the NJE Bridge Appliance is `rootpass`. This may be changed with the `passwd` command.

For information on configuring an LPAR, please refer to zSeries Processor Resources/System Management Planning Guide, IBM Document Number SB10-7033-03, which is freely available online at :

<http://publibz.boulder.ibm.com/zoslib/pdf/b1070333.pdf> <http://publibz.boulder.ibm.com/zoslib/pdf/b1070333.pdf>

## 2.2 z/VM Guest Appliance

The z/VM Guest should be defined with the following characteristics:

- 2 virtual CPUs
- 128 MB memory
- A 3215 console at address 001F
- An OSA device triple in QDIO mode (either dedicated or virtual) at address 005A-C
- A virtual CTC device at address 0060
- A 500-cylinder minidisk at address 0150
- A 3338-cylinder minidisk at address 0151
- A 200,000 block VDISK at address 0152
- A 50-cylinder minidisk at address 0191

(All cylinders are 3390 CKD.) The base VM system must have access to a real 3490 tape device it can use for installation of the NJE Bridge appliance from installation media. This device must be attached to the guest for installation, but is not necessary after the product is installed.

If virtual OSA devices are used, they should be coupled to an appropriate VSWITCH or Guest LAN; the virtual CTC should be coupled to an appropriate target guest (such as OS/390 or z/OS running within a virtual machine).

If a real CTC is supplied, then the other end of the CTC must be connected as well, and the operating system running on the host at the other end must be able to use an NJE CTC link.

The initial password for the `root` user in the NJE Bridge Appliance is `rootpass`. This may be changed with the `passwd` command.

A sample directory entry is as follows:

```
USER TCPNJE password 128M 512M G
* Only use one if you have only one real CPU
  CPU 0
  CPU 1
  IPL CMS PARM AUTOCR
  MACHINE XA 2
  CONSOLE 001F 3215 T *
* OSA For networking
```

```

SPECIAL 005A QDIO
* If it's on SYSTEM guest lan GLAN01, something like
* SPECIAL 005A QDIO 3 SYSTEM GLAN01
* If it's a real OSA,
* DEDICATE 005A 005A
* Virtual CTC: don't forget to COUPLE it
SPECIAL 0060 CTCA
* Or use a real CTC (we assume 0060 is real CTC addr):
*   DEDICATE 0060 0060
* Root filesystem
MDISK 0150 3390 1 500 VOL001 MR
* NJE spool space
MDISK 0151 3390 1 3338 VOL002 MR
* V-DISK for swap
MDISK 0152 FB-512 V-DISK 200000
* 191-disk for CMS files
MDISK 0191 3390 1 50 VOL001 MR
* System minidisk links
LINK MAINT 190 190 RR
LINK MAINT 19D 19D RR
LINK MAINT 19E 19E RR

```

## 2.3 Open Systems

### 2.3.1 Supported Platforms

The NJE/IP bridge is supported on:

**Linux:** zSeries (s390x), System/390 (s390), PowerPC (ppc), Intel x86 (x86) , AMD Opteron 64 (x86\_64), and PA-RISC (parisc).

**Sparc32:** Solaris

**Sparc64:** Solaris

**AIX:** POWER4 and POWER5

**HP\_UX:** PA-RISC

### 2.3.2 Supported OS Distributions

- Linux: Red Hat and Red Hat-derived systems, Novell/SuSE and UnitedLinux derivatives, Debian, Slackware.
- Solaris: Solaris 8, 9, 10

- AIX: 4.3, 5.x
- HP\_UX: 11i

### 2.3.3 Prerequisite Packages

- TCP/IP networking must be installed and configured.
- The GNU Privacy Guard (GPG) must be installed on the platform prior to the NJE Bridge's installation. It is either package `gnupg` or `gpg` on Linux distributions. Its source is available from <http://www.gnupg.org>.

## 2.4 Windows XP

The Windows version of the product runs under Windows XP Service Pack 2 and above. It is installed using an Install Wizard.

### 2.4.1 Pre-Requisite Package

The MKS Toolkit is required for the NJE/IP Bridge to work. This is provided on the distribution medium and should be run prior to running the NJE installation wizard.

## 2.5 Product Enablement Keys

After installation, please contact Barnard Software via email at [bsiopti@bsiopti.com](mailto:bsiopti@bsiopti.com) or via telephone at (407) 323-4773 to obtain a license key. The product will not function without a current license key.

## Chapter 3

# Installing the NJE/IP Bridge

### 3.1 LPAR Appliance Installation

1. Mount the installation tape on the 3490 tape drive at device address 0181.
2. IPL from the 181 tape device.
3. Allow the process to format the disks and copy files from the tape to disk. This process will take considerable time.
4. The following prompt will eventually appear:

```
*****  
*                Network Configuration                *  
*****  
IP Address ( 10.0.0.2 ) >
```

Enter the network parameters appropriate to your network. You will need to know the following items:

- IP Address
  - Netmask
  - Address of the router (gateway between local network and the wider world)
  - Address of the DNS server used for name resolution.
  - Fully-qualified domain name of the NJE IP Bridge.
5. The LPAR will halt.
  6. Unload the tape drive. Remove the 181 tape device from the LPAR configuration if desired.



Enter the network parameters appropriate to your network. You will need to know the following items:

- IP Address
  - Netmask
  - Address of the router (gateway between local network and the wider world)
  - Address of the DNS server used for name resolution.
  - Fully-qualified domain name of the NJE IP Bridge.
4. The system will halt. After it halts, unload the tape.
  5. Execute the following series of commands:

```
#CP DET 181
#CP IPL CMS
```

You may instead `IPL 150` if you prefer (and as the on-screen text directs), as the V-DISK has already been prepared for swapping as part of the tape installation; however, the directory is set up to `IPL CMS` so that the guest can initialize its swap space within CMS. The default root password is `rootpass`, and this may be changed with the `passwd` command.

### 3.3 Open Systems Installation

For any Linux or Unix-based system, it is presumed that the GNU Privacy Guard has been installed prior to NJE IP Bridge installation. If you have not yet installed the GNU Privacy Guard, do so now. For most Linux systems, it will be part of your distribution, and its package name is `gnupg` or `gpg`.

#### 3.3.1 Linux on System/390 or zSeries

##### 3.3.1.1 For RPM-based (SuSE, RedHat, TurboLinux) systems

1. Acquire the `nje_linux_s390.rpm` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the command:

```
rpm -Uvh nje_linux_s390.rpm
```

You should see output like:

```
nje #####
```

### 3.3.1.2 For Debian systems

1. Acquire the `nje_linux_s390.deb` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the command:

```
dpkg -i nje_linux_s390.deb
```

### 3.3.1.3 For Slackware systems or systems without a package manager

1. Acquire the `nje_linux_s390.tar.gz` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the root directory (`/`).
3. As root, execute the following sequence of commands:

```
tar xvpfz /path/to/nje_linux_s390.tar.gz  
. /etc/nje/postinstall.sh
```

### 3.3.1.4 CTC Configuration

The CTC device in the default configuration will be automatically detected, and will appear as `/dev/njctc0`. Additional CTC lines added will appear as `/dev/njectcX`, with one device per CTC line.

## 3.3.2 Linux on Intel and Other Supported Architectures

### 3.3.2.1 For RPM-based (SuSE, RedHat, TurboLinux) systems

1. Acquire the `nje_linux_arch.rpm` file, where `arch` is the abbreviation for your system architecture as described in section 2.3.1. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the command:

```
rpm -Uvh nje_linux_arch.rpm
```

### 3.3.2.2 For Debian systems

1. Acquire the `nje_linux_arch.deb` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the command:

```
dpkg -i nje_linux_arch.deb
```

### 3.3.2.3 For Slackware systems or systems without a package manager

1. Acquire the `nje_linux_arch.tar.gz` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the root directory (`/`).
3. As root, execute the following sequence of commands:

```
tar xvpfz /path/to/nje_linux_arch.tar.gz  
. /etc/nje/postinstall.sh
```

## 3.3.3 Linux on other Architectures

If you require the NJE IP Bridge for a different architecture, please contact BSI for a custom build.

### 3.3.4 Solaris

1. Acquire the `nje_solaris_sparc.pkg` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the following sequence of commands:

```
pkgadd nje_solaris_sparc.pkg  
. /etc/nje/postinstall.sh
```

### 3.3.5 AIX

1. Acquire the `nje_AIX_ppc.installp` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the following command:

```
installp -ac nje_AIX_ppc.installp
```

### 3.3.6 HP/UX

1. Acquire the `nje_HPUX_parisc.rpm` file from BSI. Contact BSI for details about obtaining installation media.
2. Change to the directory in which it resides.
3. As root, execute the following command:

```
rpm -Uvh nje_HPUX_parisc.rpm
```

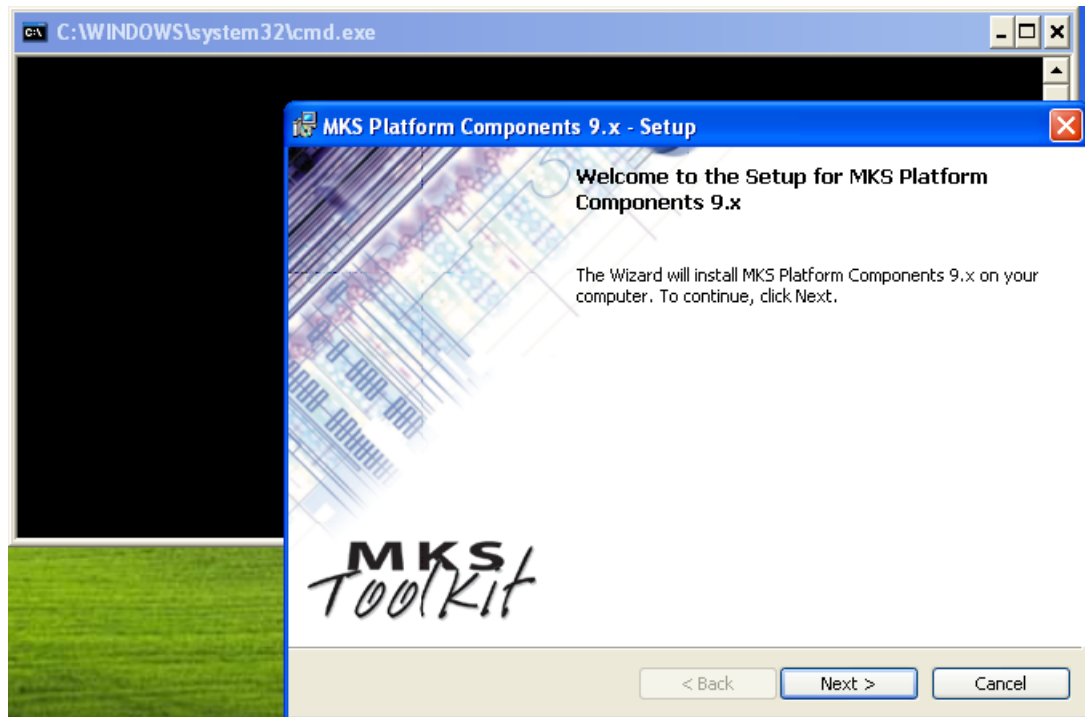
## 3.4 Windows XP Installation

1. Acquire the NJE product CD from BSI. Contact BSI for details about obtaining installation media.
2. Insert the CD.
3. If Autoplay is enabled then the install processes for the MKS Toolkit and NJE will be started.

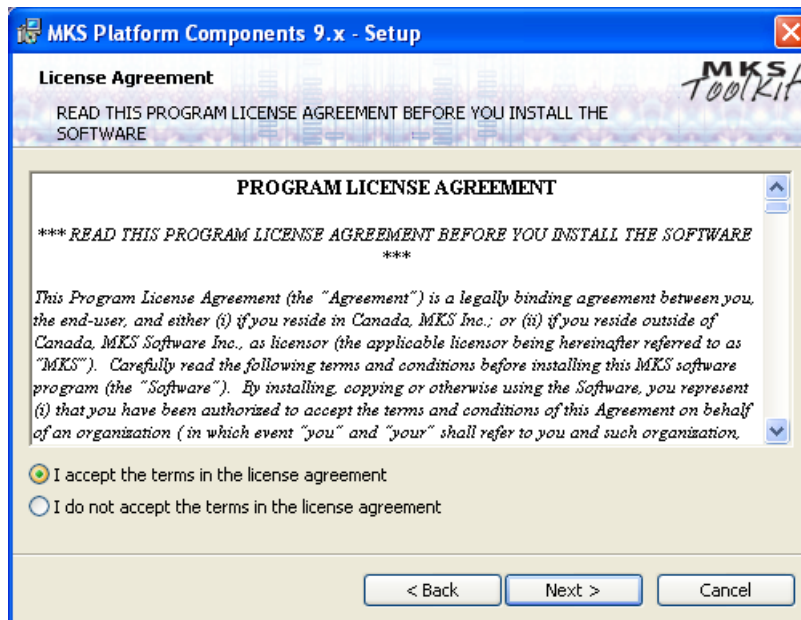
### 3.4.1 MKS Toolkit

The MKS Toolkit (also known as the “Nutcracker Workstation Client”) provides the UNIX API support for NJE.

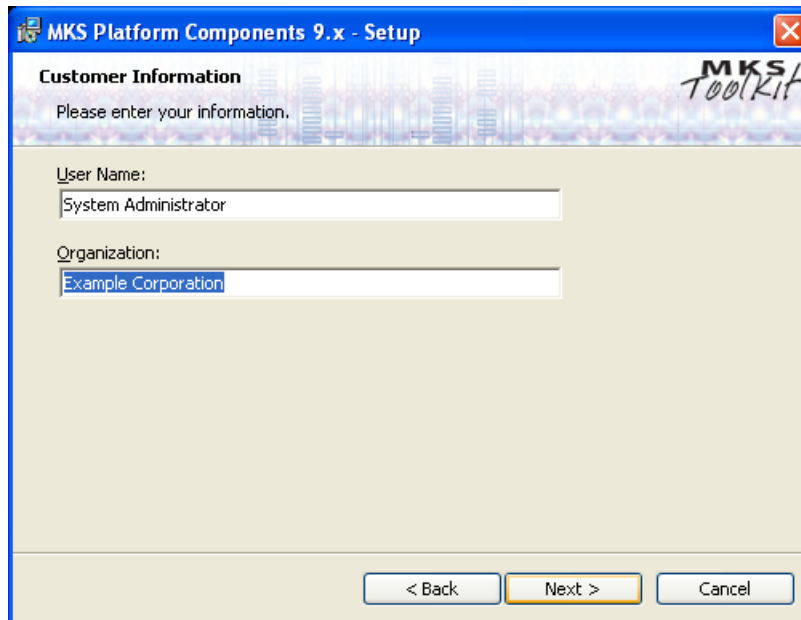
- A DOS box appears as the install process commences. The first screen you will see is the Welcome dialog for the MKS Toolkit. Just click on “Next” to get things going.



- Following the welcome the typical License Agreement panel is shown which requires you to accept the terms before the install will continue:

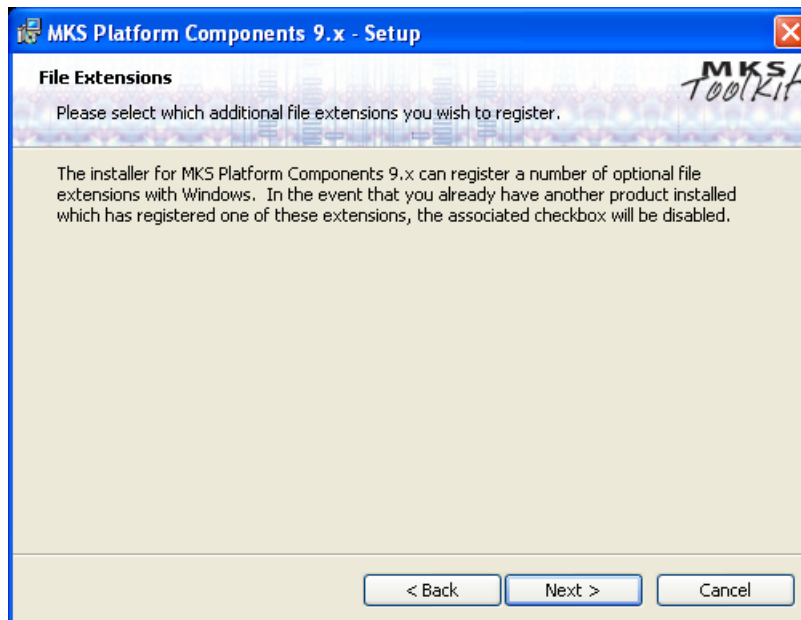


- The installation dialog will then prompt you for identification information:



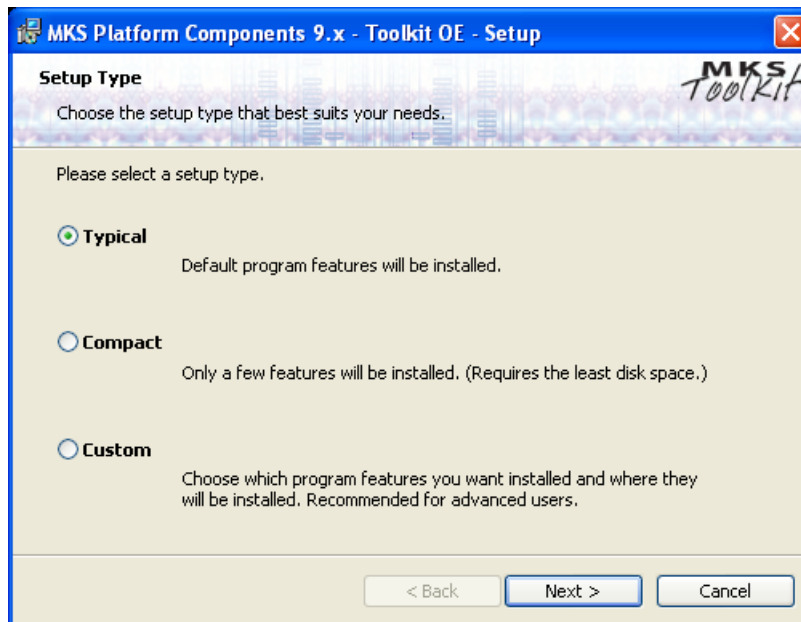
The screenshot shows a Windows-style dialog box titled "MKS Platform Components 9.x - Setup". The window has a blue title bar with a close button (X) in the top right corner. The main content area has a light beige background with a decorative header bar featuring the "MKS Toolkit" logo. Below the header, the text "Customer Information" is displayed, followed by the instruction "Please enter your information." There are two text input fields: the first is labeled "User Name:" and contains the text "System Administrator"; the second is labeled "Organization:" and contains the text "Example Corporation". At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel".

- Next there is an information panel shown that gives guidance for the next section of the install process:

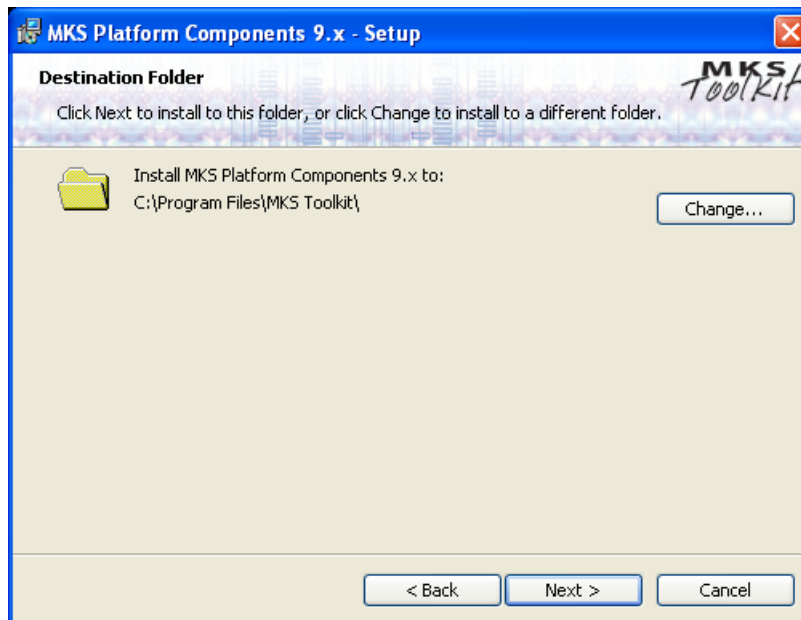


The screenshot shows the same "MKS Platform Components 9.x - Setup" dialog box, but now on the "File Extensions" step. The title bar and logo remain the same. The text "File Extensions" is displayed, followed by the instruction "Please select which additional file extensions you wish to register." Below this, there is a paragraph of text: "The installer for MKS Platform Components 9.x can register a number of optional file extensions with Windows. In the event that you already have another product installed which has registered one of these extensions, the associated checkbox will be disabled." At the bottom of the dialog, there are three buttons: "< Back", "Next >", and "Cancel".

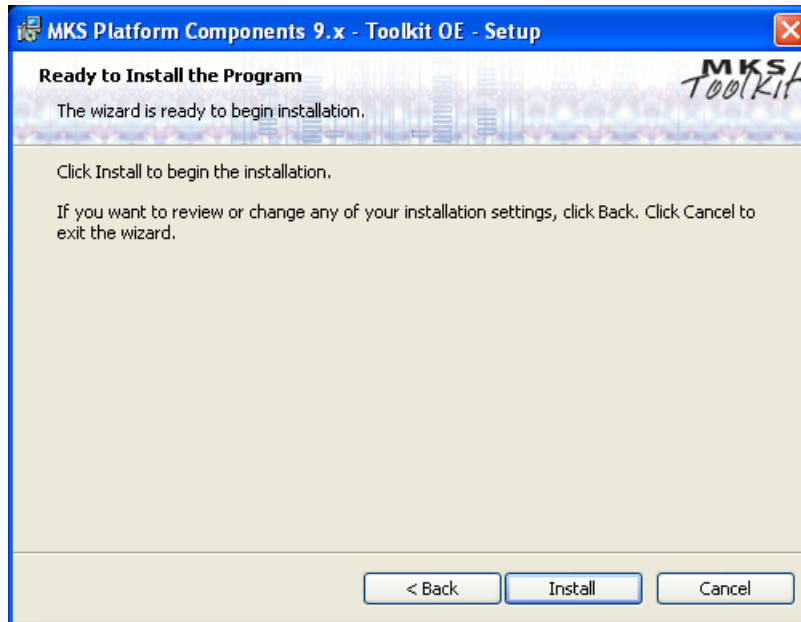
- There are three different setup types that you can choose from. We recommend the “Typical” setup type:



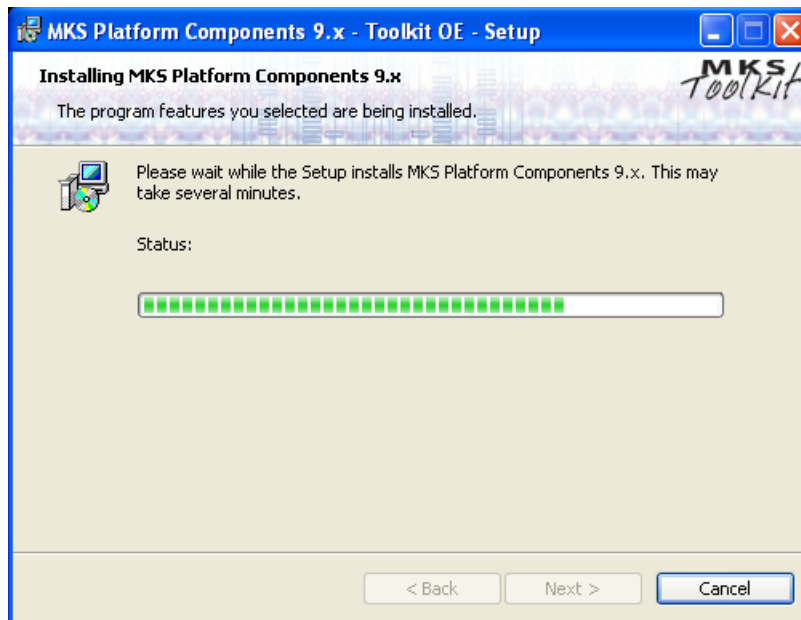
- At this point you have the option of changing where the MKS Toolkit will be installed:



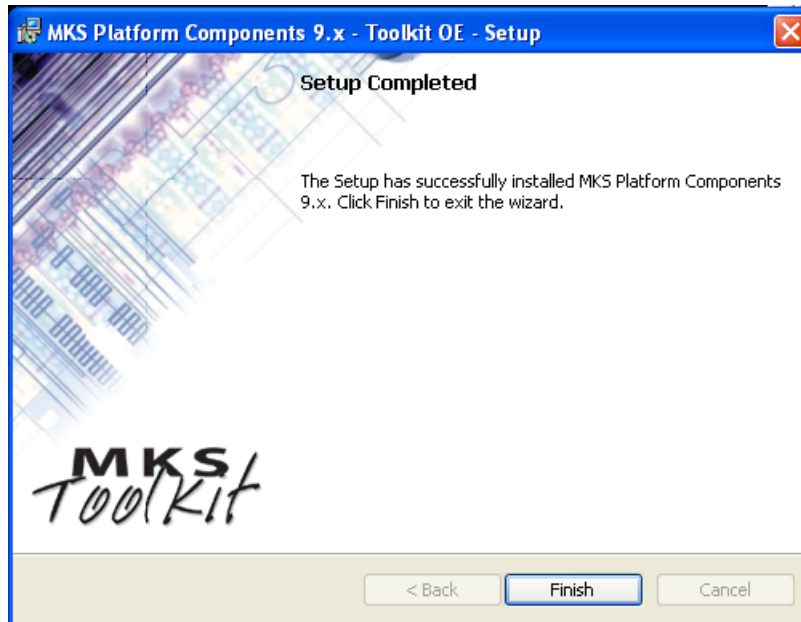
- Having gathered all the information it needs for the installation you are presented with the option of proceeding:



- The install process takes a few minutes to complete. You will be given a visual indicator that the installation is proceeding:

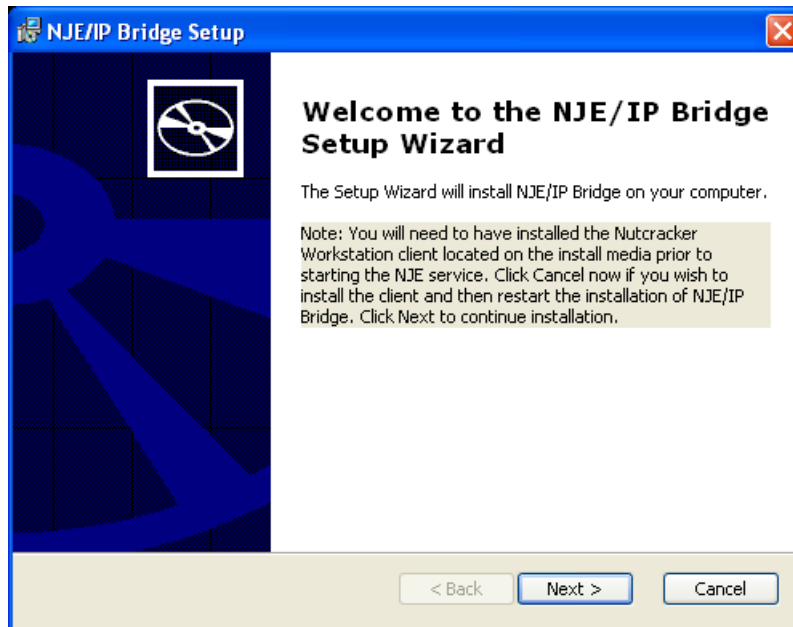


- When the installation process completes the following panel will be displayed. Just click “Finish” to terminate the procedure:

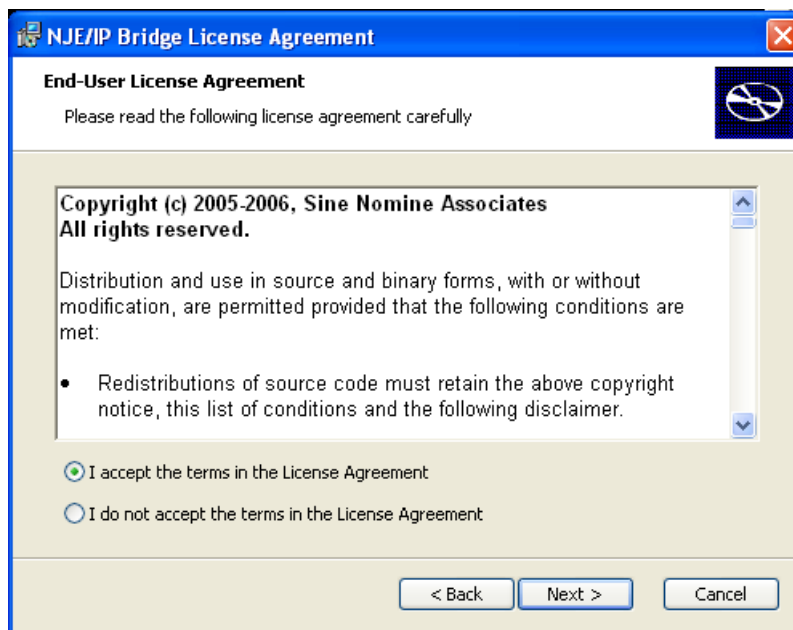


### 3.4.2 NJE Installation

- When the MKS Toolkit installation completes, the NJE install will commence. If you've completed step one of the Windows Install then the MKS (Nutcracker) Toolkit has been installed and you may click on the “Next” button to start the NJE installation.

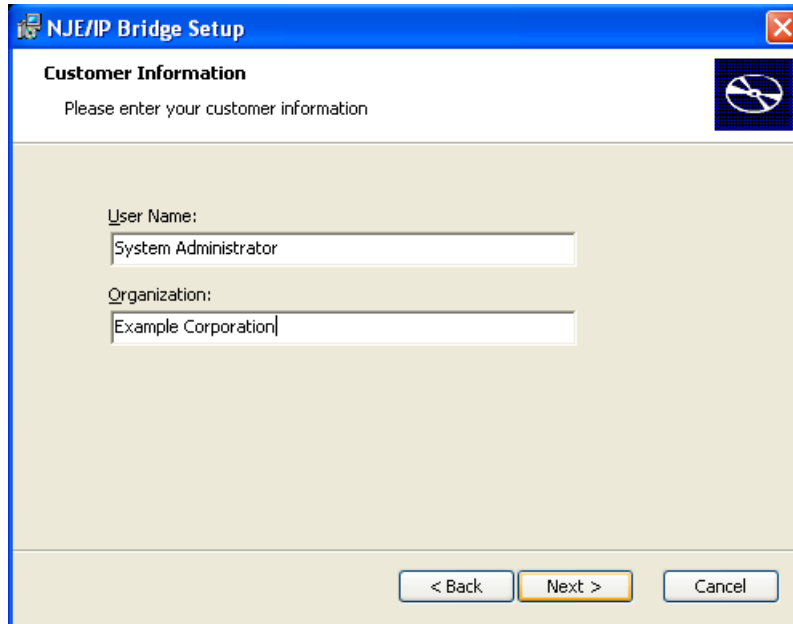


- The NJE product consists of several components each of which operates under a different license. This dialog describes each of the licenses involved:



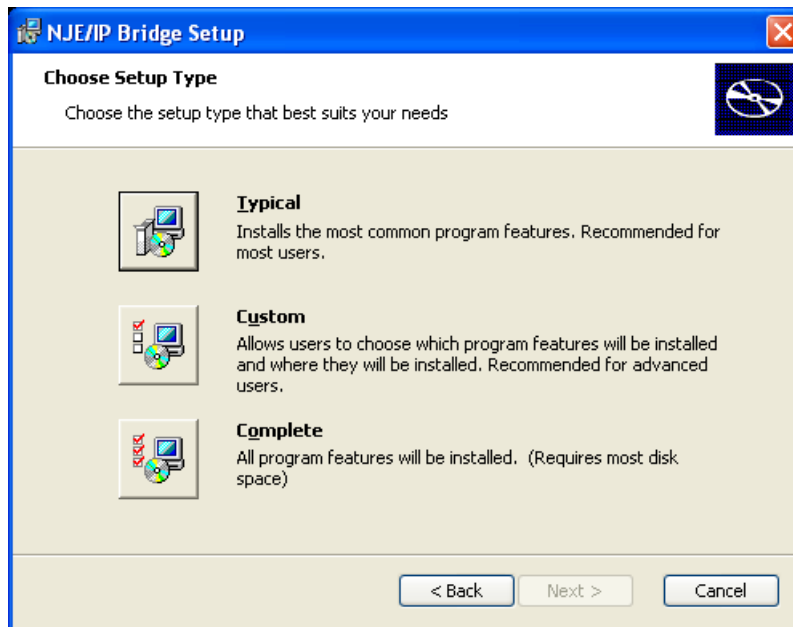
- If you accept the terms by clicking "Next" you will be prompted to enter your

customer identification information:



The screenshot shows a Windows-style dialog box titled "NJE/IP Bridge Setup". The main heading is "Customer Information" with a sub-instruction "Please enter your customer information". There are two text input fields: "User Name:" containing "System Administrator" and "Organization:" containing "Example Corporation". At the bottom, there are three buttons: "< Back", "Next >" (highlighted with a yellow border), and "Cancel".

- Next you will be solicited to choose the type of setup required. We recommend "Typical":

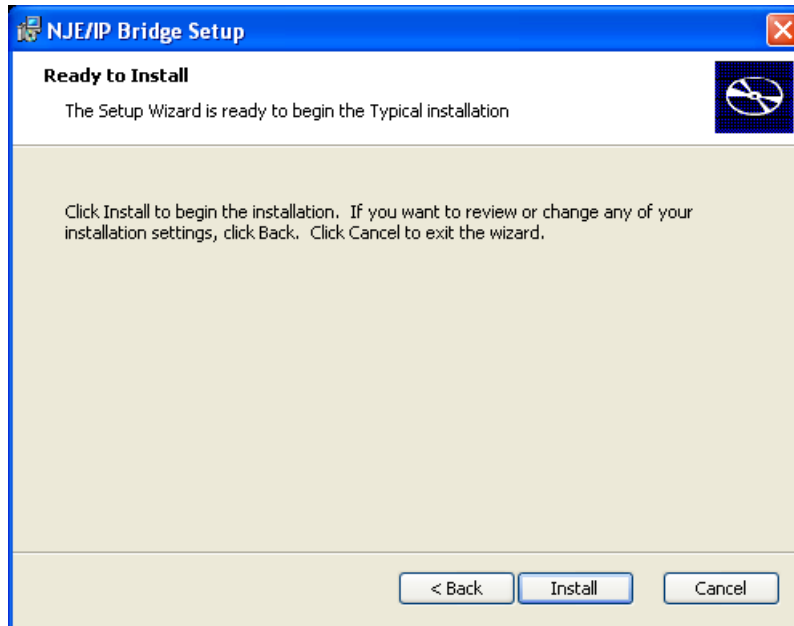


The screenshot shows a Windows-style dialog box titled "NJE/IP Bridge Setup". The main heading is "Choose Setup Type" with a sub-instruction "Choose the setup type that best suits your needs". There are three radio button options, each with an icon and a description:

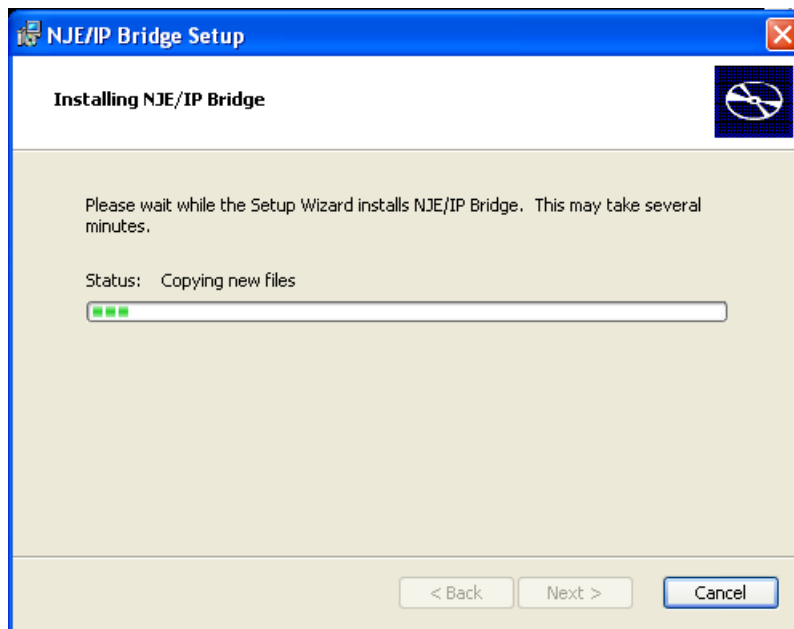
- Typical**: Installs the most common program features. Recommended for most users.
- Custom**: Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.
- Complete**: All program features will be installed. (Requires most disk space)

At the bottom, there are three buttons: "< Back", "Next >" (highlighted with a yellow border), and "Cancel".

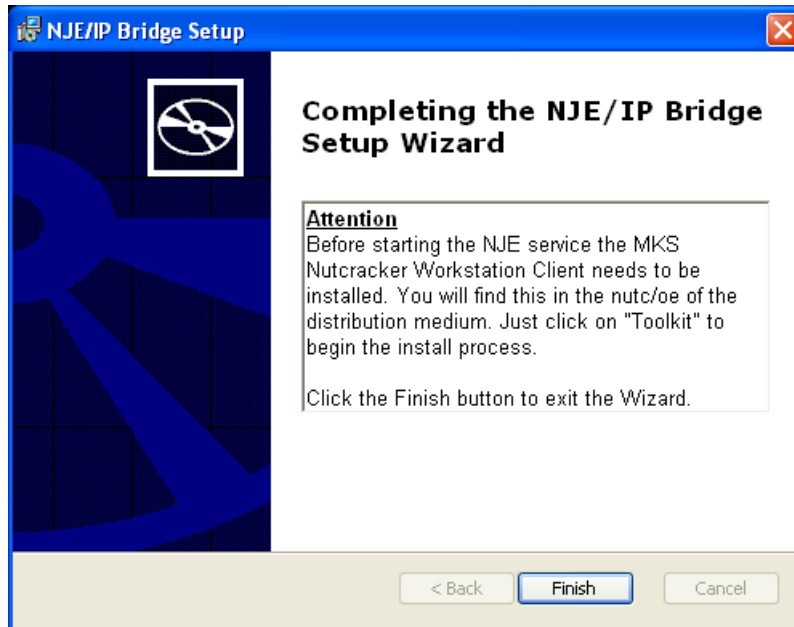
- The Installation Wizard will collect all necessary information and then ask if you wish to proceed:



- Assuming you clicked on the "Install" button the wizard will now copy the product files from the installation medium to the workstation:



- Upon successful installation the final dialog is presented. Just click “Finish” to complete the process:



## Chapter 4

# Configuration of the NJE/IP Bridge

Configuration of the product consists of editing text files that are used by the various components to perform their work.

### 4.1 Concepts

Prior to introducing the configuration files and their directives there are some concepts that are worth explaining which will make it easier to build these files.

#### 4.1.1 Links

An NJE network is a fully-specified tree. Each node must know the next hop hop for each and every other node; unlike TCP/IP networking, an unknown destination address is an error, rather than a problem for the next node in the chain.

In contrast to Ethernet, which can be thought of as a bus topology, NJE is a connected set of point-to-point links. The definition of these links and their logical topology provides the heart of NJE routing and message transmission. Each link provides a point-to-point connection between two nodes.

Each node name is 1-8 characters, and each node name must be unique. This does restrict the maximum number of nodes compared to an arbitrarily-nested scheme such as the DNS. In practice, each site will configure the NJE IP Bridge for its own internal use, and use other TCP/IP protocols to talk to other sites, so this does not present a significant limitation.

### 4.1.2 Routing

For every node known to NJE (which is every node in the entire network) NJE must know which link traffic for that node should flow through. Although NJE is a fully-specified tree, it is not necessary for each node to know the complete routing path for every other node; instead, only the next hop for traffic for a given node must be specified.

Each directly-connected node has an implicit route: the link that connects the two nodes. All other valid nodes need route statements instructing NJE how to route traffic to those nodes. NJE has the concept of a default route, down which all traffic not explicitly specified for a particular link will flow.

It is presumed that the recipient node will have additional routing information, should it not be the ultimate destination, so that it can choose the appropriate link for the messages it relays for further transmission. It is advisable, if at all possible, to make the nodename (or an alias for the real nodename) either the DNS hostname, or a truncated version thereof.

### 4.1.3 Userid Mapping

NJE transmissions are directed to a particular user at a specified node. However, the NJE IP Bridge supports the concept of a “pseudouser,” which allows particular (false) users to be mapped to specific services without requiring entries in the password file for those usernames. The “ECHO” service specified in `msg-exit.cf` is one such.

The bridge also allows for mappings between real userids; for instance, it is common to define messages to `MAINT` to be redirected to `root`, as `root` is the Unix world’s rough equivalent of the VM `MAINT` superuser.

### 4.1.4 Printing

By convention, a file sent to destination user `SYSTEM` or `LPR` is assumed to be a print file. NJE will, in the absence of other information, send the file to the local printing system to print on the default queue. Use the `distribution` parameter to label the file on the sending end, and the `DIST8` parameter in `file-exit.cf` (see section 4.2.2) to categorize the file based on distribution in order to send it to a different printer, print it with a different form, or in some manner handle it in some way different from the default.

It is also possible to differentiate a print file from a punch file in `file-exit.cf`, although no historical NJE printing systems used this as a selection criterion.

When a print file arrives, it is handed off to the `rprint` command. `rprint`, in turn, uses the destination user name and the form name, concatenated with a period (e.g. `SYSTEM.STANDARD`), to choose a form definition file from `/etc/nje/forms`.

Each file in `/etc/nje/forms` should have two components, each in uppercase, and the contents of each file are simply the options that should be passed to the `lpr` program to produce output appropriate to that form. The NJE IP Bridge ships with `SYSTEM.STANDARD` as a blank file, and `SYSTEM.STANDARD` functions as a catchall default for unknown forms: if a form is unknown, it will be fed to the local printing system's default print queue with default options.

#### 4.1.5 NJE-to-SMTP Gateway

The included program `mailify` is used to convert NJE mail messages to RFC-822 compliant email, and then feed that email to the system mailer. To use `mailify`, simply run it with the name of the file it is supposed to turn into an SMTP email: `/usr/local/bin/mailify /var/spool/nje/NJEFILE`. This is usually configured for class M files sent to user MAILER within `file-exit.cf`.

#### 4.1.6 Redirecting Messages to Programs

Incoming messages or commands can be redirected to specific programs using the `RUN` directive in `msg-exit.cf`. This is explained more fully below in section 4.2.

#### 4.1.7 Unknown User Handling

By using a wildcard to match user names, a default action is defined in `file-exit.cf` and `msg-exit.cf`. The default action is to send files for unknown users to the `POSTMAST` pseudo-user.

#### 4.1.8 Topological Design

For enterprises whose entire NJE infrastructure is within a single site, a hub-and-spoke design is recommended. One host can act as the hub, with multiple links defined to each of its leaf nodes, and each leaf node requires only a default route over a line to the host.

This hub-and-spoke model can easily be extended to an arbitrarily deep tree for enterprises with multiple locations: at each site, there should be a single hub, or branch point, and that hub should have lines defined for each other node at its site, and a default route going off-site. A second-level branch point would then communicate with all the hubs at remote sites. This design can be extended trivially to three or more levels. However, you must remember that each branch point must not only know how to route to its immediate neighbors, but ultimately to each leaf connected to any of its neighbors.

More complicated designs provide redundant links for inter-hub communications; however, since rerouting over alternate links still needs to be done manually, and since the

public Internet is a fairly robust communications network, these designs are no longer recommended.

## 4.2 Modifying NJE Configuration and Behavior

On Open Systems, the configuration files will be located in `/etc/nje` on Windows they are located in `\Program Files\NJE\etc`.

### 4.2.1 nje.cf

Using a text editor is the easiest way to manage initial configuration of the NJE IP Bridge and to make permanent changes in the Bridge's behavior. The primary file to edit will be `nje.cf`. A line in the file beginning with either an asterisk or an octothorpe is treated as a comment. `nje.cf` is an ASCII text file modifiable with any text editor, and it contains the following configurable fields:

#### 4.2.1.1 General Fields

These fields are not specific to a particular `LINE`, but instead apply to the installation as a whole.

**NAME** This specifies the primary NJE name of your node. It must be 1-8 characters long; usually the hostname (or a truncated version thereof) is appropriate. The node-name should be in your site's DNS database. If it is necessary to truncate the hostname to 8 characters, it is highly recommended that you add a `CNAME` record using the shortened hostname to your site's DNS database.

`NAME` *must* be customized for your site. When doing so, you should also change the value of the `TONODE` field in the `file-exit.cf` entries (discussed below) to reflect your primary nodename.

**ALIAS** Each alias line specifies an additional node name that identifies the host. As with `NAME`, this must be 1-8 characters. If you have additional DNS names for your host, these (truncated to 8 characters if necessary) should be specified as aliases. Additionally, the truncated names should be entered into the DNS as `CNAME` records if at all possible.

**IPADDRESS** This is used on link signon messages. `VM` and other NJE IP Bridges ignore this parameter, but `VSE/POWER` respects it, which can make tunneling NJE through NAT difficult. This may also be a name resolvable in the DNS, in which case the IP address in the signon message will be the first `A` record found for that name.

`IPADDRESS` *must* be customized for your site.

**QUEUE** This defines the directory for queued files and messages. Typically you should not change this from the default, `/var/spool/nje`.

**CMDMAILBOX** `CMDMAILBOX` can take three different parameters as its first argument: `FIFO`, `SOCK`, or `UDP`. In general, you should not change this from its default of `UDP 127.0.0.1 175`. The remaining arguments specify either the name of the Named Pipe or Unix Domain Socket (for `FIFO` or `SOCK`), or the IP address and port to bind to for the `UDP` socket.

**LOG** This specifies the log file destination. The default, `/var/log/nje/nje.log`, should generally be left untouched.

**LLEVEL** This specifies the default log level at startup; smaller numbers mean less logging, and the `LLEVEL` must be an integer. 6 is the useful maximum, with everything above 6 being treated as equivalent to 6. In general, you should leave `LLEVEL` at 1 and use the `LOGLEVEL` command from within the `ucp` control panel to manipulate the log level.

**RSCSACCT** If defined, this specifies the location of an IBM RSCS-like accounting file. This is defined by default as `/var/log/nje/nje.acct`. It is disabled by default.

**TABLE** This binary routing table is built from the routing information given in the `LINE` definitions below. The default should not be changed from `/etc/nje/nje.route`.

**EBCDICTBL** `EBCDICTBL` is an optional configuration resource. Leaving `EBCDICTBL` undefined is generally fine, as the Bridge's default behavior is based on reverse-engineering common card deck translations. If you want to specify your own mapping, however, you may. If you define it, it probably should go in the `/etc/nje` directory, and therefore setting it to a value such as `/etc/nje/e2a.tbl` is wise.

**INFORM** There may be multiple `INFORM` statements; each one defines a user to be notified (via an NJE message) if one of the defined lines changes states. This does not have a default value, but `POSTMAST@local-nodename` would be a good value (provided that `POSTMAST` itself was mapped to `root` or some other administrative user). With clever use of message exits, one could define a `PROP`-like facility to automatically take action, such as dynamic node rerouting, on line state changes.

**FILEEXITS** FILEEXITS specifies the file that contains mappings providing special destinations and handling for arriving or departing files. Although the location of the file, given here, should not be changed from its default of `/etc/nje/file-exit.cf`, you very likely will want to edit the contents of the file as described below in section 4.2.2.

**MSGEXITS** MSGEXITS performs the same function as FILEEXITS (that is, special destinations and handling for incoming or outgoing messages), but for messages rather than transferred files. Again, the default location of `/etc/nje/msg-exit.cf`, is probably fine, but the file itself will likely need customization for your own system as described in section 4.2.3.

**DEFFORM** This is the default NJE “FORM”. If an incoming file has no print form specified, this is the form given to it. By using this tag as an input discriminator to the CUPS printing subsystem, it is possible to emulate sophisticated form handling in the NJE IP Bridge (the “`rprint`” command does this already, although it is left to the printing system to pick apart the options to use the form tag). The default value is simply “DEFAULT”; change this as necessary to reflect the expectations of your users.

**DEFAULT-ROUTE** If the DEFAULT-ROUTE is defined, then you may have an empty file for your routing table (defined in TABLE), and every non-local message or file will simply be sent down the DEFAULT-ROUTE line. However, the TABLE must still exist, even if zero length.

This setting is most useful for sites which are “leaf nodes,” which speak only to a single remote host. This configuration is commonly seen in servers providing printing services for NJE users, or in developer workstations.

However, DEFAULT-ROUTE may also be specified for nodes not at the root of a hierarchical tree: all links to hosts at the local site would have specific routes, but all non-local destination would be pointed to the single route to a remote site via the DEFAULT-ROUTE setting.

The value of DEFAULT-ROUTE is simply a node name: all otherwise unrouted traffic will be sent over that LINE, with the assumption that the receiving host will know how to deliver it. There must exist a LINE statement corresponding to the node name for default delivery.

#### 4.2.1.2 Line-specific statements

**LINE** Each LINE statement defines a single link to another host. The NJE IP Bridge will not start unless at least one LINE is defined. There may be an arbitrary number of LINE statements. The format of the LINE statement is

```
LINE x dest
```

Where  $x$  is a nonnegative integer, and *dest* is the name of the node at the other end of the line. Each `LINE` statement is then followed by the following parameters.

**TYPE** `TYPE` is always `UNIX_TCP` on any host system but Linux on System/390 or zSeries. However, for Linux on System/390 or zSeries (whether as an LPAR appliance, a VM guest, or a standalone system), the type `UNIX_CTC` is also supported; in that case, `UNIX_CTC` indicates a link across a Channel-to-Channel connection to another NJE node.

**BUFSIZE** This is the TTB (Total Transfer Buffer) size for NJE transmissions. This specifies the maximum amount of data that NJE will transmit in a single block. This needs to match the value configured on the other end of the link; the default is 4096. The maximum value is determined by the communicating operating systems. Refer to the NJE support information for your operating system for further details.

**IPPORT** `IPPORT` is usually 175, the standard port used for NJE-over-IP. However, it may be set to any value depending on the other end's requirements. Port 40175 is recommended for SSL-wrapped transmissions (there is no standard port number assigned). It is ignored for `CTC` links.

**TCPNAME** This defines the other end of the link as either a DNS-resolvable name, or (if necessary) an IP address. This too is ignored for `CTC` links.

**DEVICE** This defines the name of the `CTC` device over which a `UNIX_CTC` link will be routed. From a Linux perspective, this will be a device name, such as `/dev/njectc0`. It is ignored for `UNIX_TCP` links.

**MAX-STREAMS** This defines the number of parallel streams used for this line. This is the maximum number of transactions that can be simultaneously in progress. This must match the other end's definition. Most operating systems, unless this parameter is specified, default to 1 stream.

**TCP-SIZE** The `TCP-SIZE` parameter indicates the maximum block size before a write is forced. It is really more of a transmission buffer size than anything directly TCP-related. This parameter is specific to the NJE IP Bridge.

**TIMEOUT** When `TCP-SIZE` has not been reached, but `TIMEOUT` seconds have elapsed, the partial block will be written regardless. This parameter is specific to the NJE IP Bridge.

## 4.2.2 file-exit.cf

This file specifies how files to or from particular destinations or users are to be handled; effectively, it routes special cases for file transfers. The `file-exit.cf` file will look familiar to users of PROP. A line with an octothorpe or semicolon in column 1 is treated as a comment.

**Spool-Dir:** `Spool-Dir:` should be left unchanged at `/var/spool/nje/`.

**Postmast-Dir:** `Postmast-Dir:` should not be changed from `/var/spool/nje/POSTMAST`.

**Exit-Table:** The `Exit-Table:` tag introduces the table itself. Each line in the table is a single (long) line, containing the following fields. Each field provides a set of matchable characteristics, from which the IP Bridge can determine how to appropriately route the file.

**TOUSER** This is the 1-8 character user name of the file recipient. This may or may not correspond to an actual user on your system; `file-exit.cf` allows you to create pseudo-users such that a user name may, rather than triggering a delivery attempt, trigger some other action.

**TONODE** This is the 1-8 character node name of the file recipient. Although intended for routing messages to other nodes, this too can be trapped and trigger arbitrary actions. Note that, like the node name in `nje.cf`, the default of `NJENODE1` here should be globally changed to your own nodename.

**FNAME** This is the 1-8 character file name of the received file.

**FTYPE** This is the 1-8 character file type of the received file.

**PUN?** If this is set to `PUN` it matches punch files. If it is set to `PRT`, it matches print files. If it is set to `JOB`, it matches incoming jobs, and if set to `*`, all files.

**CLASS** This is a classification mechanism for incoming files. For example, `PUN` files of class `M` and destined for user `MAILER` are taken (by convention) to be mail files and will be turned into SMTP mail by the program “`mailify`.”

**FRUSER** This is the 1-8 character userid of the originating user for the file.

**FRNODE** This is the 1-8 character node name of the node the file came from.

**DIST** This is the 1-8 character version of the NJE distribution code tag. This is an arbitrary tag that the sender sets to allow classification of transmitted files.

**FORM** This is the 1-8 character form name.

**SPOOLDIR** This is usually left at “default” (and thereby specified by the SpoolDir: directive); however, it can be modified here to provide relocation to an alternate location based on any of the other fields.

**ACTION** The ACTION field must be one of DISCARD, KEEP, NOTIFY, or RUN.

- DISCARD takes no arguments, and simply throws away the message.
- KEEP takes no arguments: it places the message either into the default Spool Directory, or into one specified in the action line.
- NOTIFY sends a message to an NJE user at the local or at a remote node.
- RUN takes two arguments: the name of an arbitrary program to run, and then a string which is passed as the program’s argument string. RUN executes the run program as the recipient user if that is a real user, or as root otherwise. Thus, be careful when specifying arguments to RUN, as these can quite easily run as the superuser. It is especially unwise to uncritically submit JOB files for execution.

**EXTRAARGUMENTS** These will specify either the notification recipient, for a NOTIFY action, or the program to run and its arguments for a RUN action.

### 4.2.3 msg-exit.cf

While `file-exit.cf` handles special-case disposition of incoming or outgoing files, `msg-exit.cf` does the same for messages and commands.

**CmdHelpFile:** This, by default, points to `cmd-help.txt` which is located in the same directory as the configuration files. Change the location of this file and replace it with your own help text if you want to change it.

### Pattern Matching

**Token patterns** Patterns are handled as follows:

A single asterisk ( \* ) surrounded by spaces matches any token, but not a blank.

A pair of asterisks ( \*\* ) surrounded by spaces match any number of tokens, including zero tokens.

An asterisk in the middle of a word specifies a mandatory prefix and then an optional suffix. For example, TO\*KEN will match any of TO, TOK, TOKE, or TOKEN, and no other string.

**Remap patterns** \$*n* substitutes the *n*th text token into its place; any other string is treated as a literal.

### Action line

**TOUSER** This represents the 1-8 character destination user name on the incoming message.

**TONODE** This is the 1-8 character destination node name on the incoming message. The default of NJENODE1 should be globally changed to reflect your own node name.

**FRUSER** This is the 1-8 character user name of the user that originated the message.

**FRNODE** This is the 1-8 character node name of the node that originated the message.

### C pattern ACTION arguments

**MACTION arguments** For Commands “C,” the ACTION may be one of BUILTIN or RUN. TOUSER is not set for commands.

For Messages “M,” the ACTION may be one of BRCST, DISCARD, or RUN.

Argument tokens known to RUN are: TOUSER, TONODE, FRUSER, FRNODE, and TEXT: these can be substituted in the command string by prepending their name with a dollar sign. TEXT is simply the text of the received message. Thus the ECHO pseudo-user is implemented as (this is all actually one long line, with no backslash: continuation characters are not allowed in the exit tables):

```
ECHO NODENAME * * M RUN /usr/local/bin/send -u echo \
$FRUSER@$FRNODE *Got Message: '$TEXT'
```

Argument tokens for **BUILTIN** are: **HELP**, **HARDCODED**, **ERROR**, and **ALIAS**.

**HELP** and **ERROR** may have message strings appended to them. This message string will be displayed to the user.

**ALIAS** takes a remap-pattern as its argument; this changes one command into another for the consumption of the message exit. For example, **CP TIME** is understood as **CPQ TIME** thanks to:

```
. NODENAME * * C "CP*QUERY T*IME" BUILTIN ALIAS CPQ $2
```

**HARDCODED** entries are required and should not be changed except by consultation with NJE IP Bridge support staff.

There is no error checking on **RUN** commands: from the NJE IP Bridge's perspective, it is purely fire-and-forget, so the `msg-exit.cf` maintainer should be aware of this fact and handle error conditions appropriately in the executed program. Additionally, commands will be run as the superuser, and therefore the `msg-exit.cf` maintainer should be very careful indeed about what exit commands do with the parameters passed to them.

## 4.3 Creating and Managing Links

### 4.3.1 Connecting via CTC

To set up the remote side of a CTC link, please consult the appropriate IBM documentation; full bibliographical references are available in section 8.1.

#### 4.3.1.1 Necessary Information

The necessary information from the NJE IP Bridge for a **TYPE CTC** link is simply the **DEVICE** parameter, specifying the device (as known to **Linux**) that the CTC uses. For any link, **BUFSIZE** and **MAX-STREAMS** must be identical at each end, although as long as they match, any value is acceptable.

#### 4.3.1.2 Defining a Link

**RSCS** Please consult VM RSCS Planning And Configuration, IBM Document Number SH24-5219, for information on RSCS CTC links.

**VSE/POWER** Please consult VSE/Power Networking and VSE/Power Administration and Configuration, IBM Document Numbers SC33-6735 and SC33-6733, for information on VSE/POWER CTC links

**JES2** Please consult the JES2 documentation for version 2 (5740-XC6):

- MVS/SP JES2 Initialization and Tuning, IBM Document Number SC23-0065
- MVS/SP JES2 Commands, IBM Document Number SC23-0064
- MVS/SP JES2 Logic, IBM Document Number LY24-6008

**JES3** JES3 documentation for version 4 (5695-048) is available:

- JES3 V4 Initialization and Tuning Guide, IBM Document Number SC23-0088
- JES3 V4 Initialization and Tuning Reference, IBM Document Number SC23-0089
- JES3 V4 Commands, IBM Document Number GC23-0090
- JES3 V4 Customization, IBM Document Number LY28-1026
- Diagnosis Reference, IBM Document Number LY28-1032

**Another NJE/IP Bridge** If the `LINE` type is `CTC` and the `CTC` is actually a virtual `CTC` emulated by `z/VM`, then it is necessary to ensure that the virtual `CTCs` of the two bridges are coupled together with the `CP COUPLE` command. Typically this should be done in `CMS PROFILE EXEC` prior to `IPL` of the Linux system.

Each end of the link must have identical `MAX_STREAM` and `MAX_BUF` parameters.

## 4.3.2 Connecting via TCPNJE

### 4.3.2.1 Necessary Information

In order to define a link via `TCPNJE`, the three absolutely critical pieces of information are:

- The destination's NJE nodename.
- Its hostname in the DNS, or its IP address.
- The TCP port on which it listens.

Additionally, `BUFSIZE` and `MAX-STREAMS` must agree at each end, although as long as they do, any values can be used.

### 4.3.2.2 Defining a Link

**RSCS** For an RSCS link, `IPPORT` is usually 175. Assuming a relatively fast and reliable network link (which almost all are in the early 21st century, at least in comparison to the networks of the IP Bridge's nascence), `BUFSIZE 4096` is appropriate (raise from the default 576). `MAX-STREAMS 7` should be used rather than the default 1.

**VSE/POWER** Please consult the VSE/POWER Networking and Administration and Configuration guides for information on configuring NJE links under VSE/POWER.

**JES2** Please consult the JES2 documentation for information on configuring NJE links under JES2.

**JES3** Please consult the JES3 documentation for information on configuring NJE links under JES3.

**Another NJE/IP Bridge** Use the same parameters as for RSCS: `IPPORT 175`, `BUFSIZE 4096`, `TCP-SIZE 8192`, `TIMEOUT 3`, and `MAX-STREAMS 7`. As with all other link partners, `BUFSIZE` and `MAX-STREAMS` must be identical at each end of the link. If `IPPORT` is not set correctly, the link will be unable to be established. Mismatched `TCP-SIZE` and `TIMEOUT` values will still permit operation, but only with a degradation in performance.

## 4.4 Deleting Links

To permanently delete a link from the NJE IP Bridge, remove its corresponding `LINE` definition from `nje.cf`, and restart the bridge.

## 4.5 Managing Link Traffic

### 4.5.1 ucp

The user using the `ucp` control panel must be the superuser or a member of the "nje" group in order to successfully run `ucp`. Commands available within `ucp` are as follows:

**SHOW LINES** Displays defined lines and their status.

**SHOW QUEUE** Shows the status of the file transmission and receiving queue

**START LINE *x*** Starts inoperative line *x*.

**STOP LINE *x*** Stops line *x* and puts it into an inoperative state; this attempts a graceful shutdown of the line, waiting for a clean link signoff.

**FORCE LINE *x*** Stops line *x* immediately; this does not attempt any graceful shutdown, and the remote end will have to cope as best it can with the sudden severing of the line.

**GONE *x y*** Adds the user *x*, whose home directory is *y* to the “gone list.” This allows NJE administrators to handle message disposition for arbitrary users without having themselves to log in as those users. If a user is on the “gone list” and the user does not have a connected terminal, incoming messages for that user will be stored in the file *y/.ygone.messages*.

**UNGONE *x*** Removes user *x* from the “gone list.”

**QUEUE *x [SIZE y]*** Queues file *x* for sending. Optionally, specify its size (in bytes) with **SIZE *y***.

**SHUT** Shut down the NJE IP Bridge, attempting a graceful shutdown, by signing off all lines first.

**SHUT ABORT** Abort the NJE IP Bridge.

**DEBUG DUMP** Dumps all line buffers to the defined logfile.

**RESCAN QUEUE** Rescans the queue and requeues files.

**RESCAN EXITS** Rescans the file and message exits.

**RESCAN ROUTE** Reopen the routing database.

**LOGLEVEL *x*** Sets the loglevel to *x*. Useful values are integers between 1 and 6 inclusive. This is the preferred method of varying the log level.

**ROUTE x TO y** Route node *x* via line *y*. Setting *y* to “OFF” will delete the route entry. These changes persist across restarts of the NJE IP Bridge, and this is the recommended way to update the routing table. For example, to instruct the Bridge to route NODE3 via LINK2, you would issue

```
ucp ROUTE NODE3 TO LINK2
```

To disable NODE3 entirely,

```
ucp ROUTE NODE3 TO OFF
```

**EXIT** Quit `ucp`.

**HELP** Displays help text.

### 4.5.2 Start/stop links

Use `ucp` and the `START LINE` and `STOP LINE` commands to bring links up and down.

### 4.5.3 Purge files on link

Use the external `purge` command, described in section 6.8.

## Chapter 5

# Verification of Installation for the NJE/IP Bridge

### 5.1 Checking Default Configuration

The procedure for verification under UNIX-like systems and Windows is fairly similar. There are only a couple of differences relating to the location of files.

#### 5.1.1 Location of Files on UNIX-like Systems

- Configuration files live in the `/etc/nje` directory.
- Spool files live in the `/var/spool/nje` subdirectory.
- Log files live in the `/var/log/nje` subdirectory.

#### 5.1.2 Location of Files on Windows XP Systems

- Configuration files live in the `\Program Files\NJE\etc` directory.
- Spool files live in the `\Program Files\NJE\var\spool` subdirectory.
- Log files live in the `\Program Files\NJE\var\log` subdirectory.

#### 5.1.3 Verification Procedure

- Ensure that the directory configuration directory exists. Change your working directory to it with:

## CHAPTER 5. VERIFICATION OF INSTALLATION FOR THE NJE/IP BRIDGE 43

```
cd <configuration directory>
```

- Ensure that the files `file-exit.cf`, `msg-exit.cf`, `cmd-help.txt`, `nje.cf`, and `nje.key` are all present. Execute the following command:

```
ls -l
```

You should see output like this on a UNIX-like system:

```
-rw-r--r-- 1 root root 2942 Sep 16 17:00 cmd-help.txt
-rw-r--r-- 1 root root 2620 Sep 16 17:00 file-exit.cf
-rw-r--r-- 1 root root 3704 Sep 16 17:00 msg-exit.cf
-rw-r--r-- 1 root root 1582 Sep 16 17:00 nje.cf
-rw-r--r-- 1 root root 840 Sep 16 17:02 nje.key
```

Do not worry if the file sizes are somewhat different, or if the dates are different.

- Edit `nje.cf` with any text editor.
  - Set the variables `NAME` and `IPADDRESS` correctly for your system (see section 4.2 for details).
  - Define at least one line to communicate with another NJE node. The NJE IP Bridge will not start unless at least one line is defined (see section 4.2 for details). The example configuration files assume that the local node name is `NJENODE` and that the single defined line is `DUMMY`. These will need to be changed.
- On UNIX-like systems, ensure that spool directory exists and is owned by (and is writeable by) group `nje`.

`ls -ld /var/spool/nje` should produce output that resembles:

```
drwxrwxr-x 8 root nje 4096 Sep 22 19:02 /var/spool/nje
```

## 5.2 LPAR Appliance Verification

### 5.2.1 Starting and Accessing the Appliance

Using the HMC, IPL the LPAR from device 150.

## 5.2.2 Starting the Bridge

The bridge will automatically start at appliance startup time; it will not start on its first IPL because it lacks a product activation key until you install one. If you need to manually start it, use the command:

```
/etc/init.d/njed start
```

## 5.2.3 Testing Basic Functionality

1. As root, run the ucp program.

```
/usr/local/bin/ucp
```

ucp should start and connect to the running NJE IP Bridge process. If you get an error message rather than the NJE-CP> prompt, stop here and verify that the NAME and IPADDRESS in /etc/nje/nje.cf are correct. If they are, and stopping and restarting the Bridge does not solve the problem, contact support immediately.

2. Display the active lines.

```
SHOW LINES
```

This should display the LINE you created when you modified your configuration file. The output should look something like:

```
NJENODE1: openNJE version 1.0(sna)/200509151300, Lines status:  
NJENODE1: Line.1 NJENODE2 0 (Q=0000) ACTIVE TCP  
NJENODE1: 7 streams in service.  
WrSum: 0fil/2404B RdSum: 0fil/2530B  
NJENODE1: End of Q SYS display
```

3. Display the contents of the queue.

```
SHOW QUEUE
```

There should be nothing in the queue. so you should see something like:

```
NJENODE1: Files waiting:  
NJENODE1: End of list: (0 files, 30 shown)
```

4. Exit the ucp control panel.

```
QUIT
```

## 5.2.4 Stopping the Bridge

To stop the bridge, as root, execute the command

```
/etc/init.d/njed stop
```

## 5.2.5 Stopping the Appliance

As root, execute the command

```
shutdown -h now
```

## 5.3 Linux on S/390 Guest Verification

### 5.3.1 Starting the Bridge

The NJE IP Bridge should start automatically when the guest starts; it will not start on its first IPL because it lacks a product activation key until you install one. If you need to start it manually, execute the command (as root):

```
/etc/init.d/njed start
```

### 5.3.2 Testing Basic Functionality

1. As root, run the ucp program.

```
/usr/local/bin/ucp
```

ucp should start and connect to the running NJE IP Bridge process. If you get an error message rather than the NJE-CP> prompt, stop here and verify that the NAME and IPADDRESS in /etc/nje/nje.cf are correct. If they are, and stopping and restarting the Bridge does not solve the problem, contact support immediately.

2. Display the active lines.

```
SHOW LINES
```

This should display the LINE you created when you modified your configuration file. The output should look something like:

## CHAPTER 5. VERIFICATION OF INSTALLATION FOR THE NJE/IP BRIDGE46

```
NJENODE1: openNJE version 1.0(sna)/200509151300, Lines status:
NJENODE1: Line.1 NJENODE2 0 (Q=0000) ACTIVE TCP
NJENODE1: 7 streams in service.
WrSum: 0fil/2404B RdSum: 0fil/2530B
NJENODE1: End of Q SYS display
```

3. Display the contents of the queue.

```
SHOW QUEUE
```

There should be nothing in the queue. so you should see something like:

```
NJENODE1: Files waiting:
NJENODE1: End of list: (0 files, 30 shown)
```

4. Exit the ucp control panel.

```
QUIT
```

### 5.3.3 Stopping the Bridge

On UNIX-like systems, to stop the bridge, as root, execute the command

```
/etc/init.d/njed stop
```

On Windows XP use the `ucp shut` command to terminate the Bridge. The NJE Console may be closed via CTRL-C.

## 5.4 Open Systems Verification

### 5.4.1 Linux on Open Systems, Solaris, AIX, and HP/UX

#### 5.4.1.1 Starting the Bridge

The NJE IP Bridge should start automatically when the system starts. If you need to start it manually, execute the command (as root):

```
/etc/init.d/njed start
```

### 5.4.1.2 Testing Basic Functionality

If you are running Solaris, before you begin, it is necessary to set your LD\_LIBRARY\_PATH environment variable to find the libraries necessary to run the NJE IP Bridge:

```
export \  
LD_LIBRARY_PATH=/usr/local/lib:/usr/local/ssl/lib:\  
$LD_LIBRARY_PATH
```

or, under csh:

```
setenv \  
LD_LIBRARY_PATH=/usr/local/lib:/usr/local/ssl/lib:\br/>$LD_LIBRARY_PATH
```

Operating systems other than Solaris do not require this step.

1. As root, run the ucp program.

```
/usr/local/bin/ucp
```

ucp should start and connect to the running NJE IP Bridge process. If you get an error message rather than the ucp> prompt, stop here and verify that the NAME and IPADDRESS in /etc/nje/nje.cf are correct. If they are, and stopping and restarting the Bridge does not solve the problem, contact support immediately.

2. Display the active lines.

```
SHOW LINES
```

This should display the LINE you created when you modified your configuration file.

3. Display the contents of the queue.

```
SHOW QUEUE
```

There should be nothing in the queue. so you should see something like:

```
NJENODE1: Files waiting:  
NJENODE1: End of list: (0 files, 30 shown)
```

4. Exit the ucp control panel.

```
QUIT
```

### 5.4.1.3 Stopping the Bridge

To stop the bridge, as root, execute the command

```
/etc/init.d/njed stop
```

## 5.5 Windows Verification

As part of the installation process, the %PATH% settings for the Windows system was updated so that the system will look in the NJE folders for executable commands.

### 5.5.1 Starting the Bridge

To start the NJE/IP Bridge automatically place a shortcut to NJE.EXE in the startup folder. This file lives in the \Program Files\NJE\sbin directory. To start it manually double-click on the NJE.EXE icon or from “Start->Run” enter NJE.EXE and click “Run”.

### 5.5.2 Start the NJE Console

Status messages and messages from other nodes are delivered to a console application that must be started either by placing a shortcut to NJECONS.EXE in the Startup folder or being executed manually. This file lives in the \Program Files\NJE\sbin directory.

### 5.5.3 Testing Basic Functionality

1. From a DOS window, run the ucp program.

```
ucp
```

ucp should start and connect to the running NJE/IP Bridge process. If you get an error message rather than the ucp> prompt, stop here and verify that the NAME and IPADDRESS in nje.cf are correct. If they are, and stopping and restarting the Bridge does not solve the problem, contact support immediately.

2. Display the active lines.

```
SHOW LINES
```

This should display the LINE you created when you modified your configuration file.

3. Display the contents of the queue.

```
SHOW QUEUE
```

There should be nothing in the queue. so you should see something like:

```
NJENODE1: Files waiting:
NJENODE1: End of list: (0 files, 30 shown)
```

4. Exit the ucp control panel.

```
QUIT
```

### 5.5.4 Stopping the Bridge

To stop the bridge execute the command

```
ucp shut
```

### 5.5.5 Printing on Windows

The default LP command on Windows is insufficient to do useful work. In fact it's pretty brain-dead. Therefore to provide a slightly more intelligent method of doing remote printing using NJE we have provided a script (WINPRINT.CMD) and executable (WINPRT.EXE).

#### 5.5.5.1 FORM Files

To specify the attributes of a FORM a user may create a file "<formname>.WINFORM" in the etc folder. This file is a simple XML document that allows you to specify:

1. COLLATE - Whether pages are to be collated (default is true)
2. COLOR - Whether colors are to be enabled (default is true)
3. COPIES - How many copies of the document to print (default is 1)
4. DUPLEX - Horizontal, Vertical, or none (default is none)
5. FONT - the face and size to be used (default is arial 12)
6. ORIENTATION - portrait or landscape (default is portrait)
7. MARGINS - the top, bottom, left, and right margin settings (default is [25,25,25,25])
8. PAPERSIZE - a value that is matched to the capabilities of the printer describing the size of the paper (e.g. Letter)
9. PAPERSOURCE - a value that is matched to the capabilities of the printer describing where the paper is taken from for the print job

## CHAPTER 5. VERIFICATION OF INSTALLATION FOR THE NJE/IP BRIDGE 50

### 5.5.5.2 Sample Forms

NJE supplies two WINFORM files: STANDARD and LCN08LT.

The STANDARD WINFORM describes a non-duplexing, collating, black-and-white form:

```
<FORM>
<COLOR val="N"/>
<COLLATE val="Y"/>
<DUPLEX val="S"/>
</FORM>
```

LCN08LT describes a Landscape orientation, using an 8 point Courier New font, non-duplex, collating, black-and-white form:

```
<FORM>
<COLLATE val="Y"/>
<COLOR val="N"/>
<DUPLEX val="S"/>
<FONT face="Courier New" size="8"/>
<MARGINS left="10" right="10" top="10" bottom="10"/>
<ORIENTATION val="L"/>
</FORM>
```

### 5.5.5.3 FORM Schema

The XML document conforms to the schema WINPRINT.XSD that may be found in the etc folder.

## 5.5.6 file-exit Configuration

This section assumes you have read the information contained within section 4.2.2 on page 34. Using the following entry in file-ext.cf as an example (which actually spans a single line), we'll examine the salient entries:

```
T610PS WIN32 * * * * * default
RUN \WINDOWS\system32\cmd.exe /D /E:ON /C
\PROGRA~1\NJE\bin\winprint.cmd $TOUSER $FID \\VA1PDC\dbt610ps
$SPOOL $FRUSER $FRNODE $FORM $FNAME $FTYPE
```

**T610PS** The alias of the printer. When jobs are printed they are directed to this user id.

**WIN32** The name of the node from where the printing will be performed.

*CHAPTER 5. VERIFICATION OF INSTALLATION FOR THE NJE/IP BRIDGE 51*

**VA1PDC** The domain to which the printer belongs

**dbt610ps** The name of the printer on the domain

When a file is directed to this printer NJE will execute the `winprint.cmd` script which takes the parameters (such as `$FORM`) and passes them to `winprt.exe` which drives the printer. The `FORM` information is used to select the correct `WINFORM` file (if a corresponding file cannot be found it will default to `STANDARD`).

## Chapter 6

# User Commands

This chapter describes some of the useful user commands for the NJE IP Bridge. Only the most common and useful options to these programs will be discussed; there is quite a lot of functionality present that is only, or mostly, of historical interest (for example, the various dump formats that receive can produce, many of which produce dumps for VMS tools that no longer exist).

All of these commands are restricted to users in group “nje”: for a user to use any NJE IP Bridge functionality, that user must be part of the group nje. Group membership may be managed in Linux and other POSIX systems with the `groups` command.

### 6.1 send

`send` is the basic command you use to send interactive messages or commands to other NJE nodes. To send a message to a user at another node:

```
send -m user@node Message Text ....
```

To send a command to another node:

```
send -c node Command Text ...
```

For both of these commands, you will need to escape any shell metacharacters you send in the text string. Thus to send “Asterisk (\*)” to the other end, you would need to execute the command:

```
send -m user@node Asterisk \(\*\)
```

## 6.2 sendfile

The basic form of sendfile is:

```
sendfile user@node file
```

However, unlike most other commands, sendfile has quite a number of useful options. The full command syntax is:

```
sendfile ToUser@ToNode [-u origuser]
      [-class SpoolClass] [-fix|-var]
      [-bin] [-fn fname ftype]
      [-form FormName] [-dist Distribution]
      [-tag RSCS-TAG] [-asa] [filepath]
```

The options work as follows:

- `-u origuser`: Set the originating user to `origuser`. This option is restricted to privileged users.
- `-class SpoolClass`: Set the spool file class; for instance, class “M” is, by convention, a mail file.
- `-fix`: define the file as fixed record format
- `-var`: define the file as variable record format (the default)
- `-bin`: the file is binary, rather than ASCII. This also sets it to class “N” when it transmits.
- `-fn fname ftype`: set the filename of the received file to `fname`, and the filetype to `ftype`, rather than taking them from the source file name and type. If this is omitted, and there is no period-followed-by-extension in the filename, filetype “DATA” is set.
- `-form FormName`: use print form `FormName`. This is useful for varying the parameters when submitting a file to a print queue.
- `-dist Distribution`: add an arbitrary (up-to-8-character) Distribution tag. This can be useful for automatic classification of files on arrival.
- `-tag RSCS-TAG`: add an arbitrary (up-to-28-character) NJE tag. Again, this is useful for automated file classification.
- `-asa`: file contains ASA carriage control
- `filepath`: path to file; if omitted, the standard input is used.

sendfile will not send a zero-length file, although it produces no error if you attempt to.

### 6.3 njeprint

`njeprint` is just the same as `sendfile`, except that it sets the type of file to `PRT` rather than `PUN`. On other NJE IP Bridge nodes, incoming print files are handled by the `rprint` command and sent to the system print queue; it is typical to do something similar on other NJE nodes.

### 6.4 submit

`submit` is just the same as `sendfile` or `njeprint`, except that the type of file is `JOB`, rather than `PRT` or `PUN`.

### 6.5 qrdr

Each NJE user has a virtual card reader, in which files sent to that user accumulate; it behaves essentially as a mailbox. To see files that have accumulated in your reader, use the `qrdr` command. It takes no arguments:

```
qrdr
```

This will show you a list of files in your reader; the last column is the file number, which you will need in order to receive or peek at the file. Privileged users may peek at other users' file queues by using the `-u` option:

```
qrdr -u user
```

### 6.6 receive

The `receive` command moves files from your reader to the current working directory. It takes a huge variety of options, but the most important ones are:

- `receive filename`
- `receive -n filename`
- `receive -b filename`
- `receive -u user filename`

An additional option, available if you have sufficient Unix file permissions (which usually means being a privileged user) is:

- `receive filename`

These options work as follows:

- The `-n` option leaves the file in your reader, rather than deleting it; the default behavior is to remove the file from your reader queue once you receive it.
- The `-b` option receives it as binary rather than translating it to ASCII; the default behavior is to presume the file was transmitted in EBCDIC and translate it to ASCII on receipt, but this obviously does not work well for files that are intended to be binary, such as executables.
- The `-u user` option allows you to look at other users' spool files; however, unless your Unix user privilege is sufficient to allow you to view their spool files, you will not actually be able to display or receive files from them. Flags may be combined, which is especially useful with `-u`.
- Many of the remaining options create output in dump formats for VMS tools that are no longer useful (e.g. "bitcat" style). One that may be still be of interest is `-asa`, useful if the file contains ASA carriage control directives, as this will attempt to translate those control directives into something more friendly to a Unix or Linux terminal.
- The `receive filename` option is mostly useful for removing files from outbound queues, when you know that they will never be successfully transmitted; however, `purge` can equally well be used for this purpose.

## 6.7 peek

The `peek` command displays the contents of the reader to the standard output, rather than copying it to a file. It is equivalent to `receive -n -o - filename`. Its syntax is:

```
peek filename
```

The `-u` flag works for `peek`, too, so privileged users can see what is in other users' queues.

## 6.8 purge

`Purge` deletes a file from the incoming queue, rather than copying it to a file. It is equivalent to `receive -o /dev/null filename`. Its syntax is:

```
purge filename
```

As with `peek`, the `-u` flag is applicable to `purge`. As with `receive`, it is possible to purge a file given its pathname; this makes it an excellent way to purge outbound files that you really don't want to send:

```
purge filename
```

## 6.9 `ygone`

`ygone` tells the NJE IP Bridge that you are gone and that messages arriving for you should be saved. The `-d` option disables the “gone” setting and displays any messages that have arrived in the interim. Command syntax is just:

- `ygone`
- `ygone -d`

Note that even if you are “gone” according to `ygone`, as long as your `userid` has a terminal device, messages will be displayed rather than stored. For `ygone` to function and queue messages you must both be entered in the “gone list” and be disconnected from the host.

## Chapter 7

# Problem Determination

### 7.1 Logging and Log Files

The primary log file is found in `/var/log/nje/nje.log`.

### 7.2 Setting Debugging Level

From the `ucp` control panel, use the command `LOGLEVEL 6`. This is the highest setting and will log the most debug data; levels between 1 and 6 log intermediate amounts.

### 7.3 What to Capture

From the `ucp` control panel, execute the `DEBUG DUMP` command to display the contents of the queues, and then submit the log file with your problem report.

### 7.4 Turning Debugging Off

From `ucp`, use the command `LOGLEVEL 1` to turn debugging messages back down to normal operational levels.

## **7.5 Reporting Problems**

### **7.5.1 If you have support**

If you have a support contract, please open a trouble ticket per the instructions in your support contract. If requested by the support team, please send the log file demonstrating the problem at that point.

### **7.5.2 If you do not have support**

If you do not have support, you may send email to [nje-support@bsiopti.com](mailto:nje-support@bsiopti.com). Attach the log file to your email message. All non-contract support is provided strictly on an as-available basis. For this reason, production users of the NJE IP Bridge are strongly advised to purchase a support contract.

# Chapter 8

## References

### 8.1 NJE References

In this section, all URL references, even if split across multiple lines, are to be entered as a single long string into the browser's location field.

The best reference to NJE as a whole is unquestionably:

- [z/OS NJE Formats And Protocols \(IBM Document Number SA22-7539\)](#)

An earlier version (lacking TCPNJE), IBM Document Number SC23-0070-03, is publicly available at:

- [http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/BOOKS/IEA1M503/CCONTENTS?SHELF=IEA1MB00&DN=SC23-0070-03&DT=19980414152716](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA1M503/CCONTENTS?SHELF=IEA1MB00&DN=SC23-0070-03&DT=19980414152716)

For the initial definition of NJE transmission over TCP/IP, consult the Peter Olenick (at Princeton University) document "Definition of the BITNETII Protocol: A Technical Overview of VMNET". It is available from

- <http://www.funet.fi/pub/netinfo/BITNET/brfc0002.text>.

For RSCS (including TCPNJE) configuration, use VM RSCS Planning And Configuration, IBM Document Number SH24-5219, available at:

- [http://publibz.boulder.ibm.com/cgi-bin/bookmgr\\_OS390/download/DMTA2A03.pdf?DT=19990511213526](http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/download/DMTA2A03.pdf?DT=19990511213526)

There are two primary VSE/POWER reference books. These are VSE/POWER Networking, IBM Document Number SC33-6735, and VSE/POWER Administration and Configuration, IBM Document Number SC33-6733. These are respectively available at

- <http://publibfp.boulder.ibm.com/epubs/pdf/iesnpe20.pdf>

and

- <http://publibfp.boulder.ibm.com/epubs/pdf/iespme30.pdf>
- JES2 documentation for version 2 (5740-XC6) is available:
  - MVS/SP JES2 Initialization and Tuning, IBM Document Number SC23-0065
  - MVS/SP JES2 Commands, IBM Document Number SC23-0064
  - MVS/SP JES2 Logic, IBM Document Number LY24-6008
- JES3 documentation for version 4 (5695-048) is available:
  - JES3 V4 Initialization and Tuning Guide, IBM Document Number SC23-0088
  - JES3 V4 Initialization and Tuning Reference, IBM Document Number SC23-0089
  - JES3 V4 Commands, IBM Document Number GC23-0090
  - JES3 V4 Customization, IBM Document Number LY28-1026
  - Diagnosis Reference, IBM Document Number LY28-1032

## 8.2 BSD License Text

Copyright (c) 2005, Sine Nomine Associates.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sine Nomine Associates nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### **8.3 Support Information**

For support, please contact Barnard Software, Inc.

- Telephone: (407) 323-4773
- Email: [bsiopti@bsiopti.com](mailto:bsiopti@bsiopti.com)