NJE Bridge: End User Command Reference
Version 2 Release 1.0

Contents

If You Have Trouble 47

Appendices 49

Figures

Tables

About This Document

This document describes the end user commands and functions of Sine Nomine's full NJE Bridge product. The document provides an introduction to NJE, discussion of common tasks and functions, and the reference description of the end user commands provided by the package.

Administrative commands and configuration statements are located in the Planning and Configuration Reference.

About the NJE Bridge

The NJE/IP Bridge is a product designed to enable TCPNJE connections between traditional IBM NJE hosts and open/discrete systems. It supports communication between any 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 and unattended file transfers between any combination of NJE-connected systems. 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 distributed processing system.

The NJE/IP Bridge provides:

- The ability to define TCP (version 4 and version 6) links point-to-point connection with or without SSL protection

- A store-and-forward file transfer service

- Interactive messaging

- The ability to take arbitrary action via user-defined exits on receipt of a message or file; this can include mapping of "pseudo-users" to particular services.

- If the system is equipped with NQS, the ability to accept remotely-submitted jobs, execute them, and return output to arbitrary NJE destinations.

A product activation key (PAK) is required for the product to function. Please contact Sine Nomine Associates before installation to obtain a PAK for your system.

Common NJE Tasks and Processes

Before You Start

What Is NJE?

Network Job Entry (NJE) is a set of protocols and software implementations defined by IBM that provide the ability to transfer files and interactive messages between hosts implementing NJE applications. NJE implementations can run on all of IBM's major operating systems, and implementations of varying degrees of completeness exist for non-IBM systems, such as Sine Nomine's NJE Bridge (which provides a full implementation). NJE allows the user to perform unattended file transfers to remote systems where files and messages can be delivered directly to users, or be processed by a batch job monitor if the system supports one, or processed by an application that responds to incoming files and requests with data requested by a user. Finally, NJE provides the ability to send commands related to the operation of the network or system to be executed at a remote node, returning the results of those commands to another node in the NJE network.

NJE relies on some basic concepts and ideas to operate. The next few sections discuss these basic concepts.

Units of Work

An NJE unit of work (sometimes referred to in the IBM documentation as a "transfer unit") is a discrete unit of information that is transmitted across the network. An NJE transfer unit can be either an NJE job or a nodal message record (NMR).

An NJE job is a transfer unit that contains data to be processed at another node in the NJE network. It begins with a job header, is followed by data, and ends with a job trailer. The type of data contained in the NJE job further defines the type of NJE job. The data between the job header and job trailer can be either SYSIN or SYSOUT data. An NJE SYSIN job is an NJE job that contains JCL for a job and may have one or more SYSIN data sets. An NJE SYSOUT job is an NJE job that contains one or more SYSOUT data sets. Each SYSOUT data set is preceded by a data set header.

A nodal message record (NMR) is a unit of work that begins with an NMR header and is followed by message text. The message text can be either a message or a system command.

Nodes

A node is a system or complex of systems that supports NJE connectivity. A node in a NJE network can be another complex or system within a single location or it can be a complex that resides in a remote location. Each node that a complex can access must be identified to other complexes by a unique NJE node name.

The NJE node name appears in job headers, data set headers, and NMRs. Do not confuse link or line names with the node name, they are two separate entities.

Each node in the network can do the following with an NJE transfer unit:

- Transmit

  The node packages the NJE transfer unit and transmits it to another node.

- Receive

  The node recognizes the NJE transfer unit, receives, and stores it.

- Store-and-forward

  The node accepts the NJE transfer unit, stores it, and schedules it to be forwarded to another node.

The IBM documentation for NJE uses the following terminology for the nodes that comprise an NJE network.

- Originating Node

  An originating node is the node where the user submitted the request to transmit the data to another complex.

- Intermediate Node

  An intermediate node is a node that lies in the path of either the:

  - Originating node and execution node

  - Execution node and the destination node

  An intermediate node both receives and transmits the NJE transfer unit to the next node in the path of the target node.

- Target Node A target node is the node where a NJE job or NMR is received and will either be executed or be processed. The target node can be either a:

  - Destination Node

    A destination node is a node that receives and processes:

    — An NJE SYSOUT job. A node processes an NJE SYSOUT job by printing or punching the SYSOUT data set.

    — A message contained in an NMR.

    When an NJE transfer unit reaches its destination, it may or may not be processed as the user intended, depending on the facilities available at that node. NJE protocols allow the destination node to reject files that it cannot process or perform other system-dependent actions.

  - Execution Node

    An execution node is the node where:

    — JCL contained in an NJE SYSIN job executes. The node packages the SYSOUT data sets created by the SYSIN in an NJE SYSOUT job and sends the NJE job to the destination node.

    — A command contained in an NMR is processed. The node packages the messages that are a result of the command in an NMR and sends the NMR to the destination node.

    The execution node may not necessarily be the destination node. If, for example, a user submits a job specifying that the job execute at one complex and job's output be printed at a different complex then the complex where the job runs is the execution node and the complex where the SYSOUT prints is the destination node. If no SYSOUT destination is specified, then the origin node and destination node are the same by default.
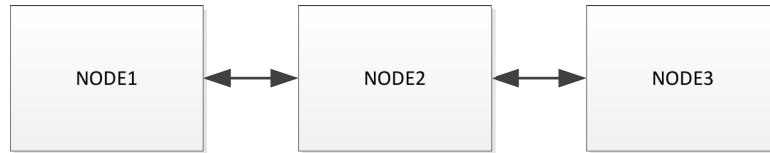
Figure 1. Sample NJE Network

Figure 1 on page 6 illustrates the different types of nodes in an NJE network. The network is composed of 3 nodes. If a user submits a job at node 1 to be executed at node 3:

- Node 1 is the originating node because that is where the user submitted the request.

- Node 2 is the intermediate node, because it is in the path of the destination node. Node 2 receives the data, stores it, then forwards it to the next node in the path of the target node.

- Node 3 is both the execution and destination node because it is the node that the user specified as the target.

To transmit an NJE transfer unit to a complex other than the user's installation (a remote node), the user issues a command or submits a job specifying a destination node name. The destination node can be either directly- or indirectly-connected to the originating node. In the network depicted in Figure 1, if NODE1 is the originating node, NODE2 is a directly-connected node to NODE1, and NODE3 is an indirectly-connected node to NODE1.

Users

The IBM documentation refers to several types of user.

- Originating User

  An originating user is the user that submits an NJE transfer unit at the originating node. The originating user submits the NJE transfer unit at an operator console, terminal, or an RJE workstation. An NJE transfer unit may originate from another NJE transfer unit.

- Destination User

  A destination user is a user or device (printer or punch) that is the target of an NJE SYSOUT job. Printers at a node are destination users.

- Notification User

  A notification user is the user who receives messages that notify the user of the status of the NJE transfer unit.

- Accounting User

  An accounting user is the user that receives the notification of the amount or cost of system resources used in processing an NJE transfer unit.

NJE Addressing

NJE protocols route an NJE job to the specified destination based on addressing information supplied by the originating system. The format of specifying that information depends on the NJE implementation in use; some systems use the common user@node convention present in modern electronic mail addresses, but some implementations use variant forms that integrate with that node's JCL. Figure 2 on page 8 shows some alternative variations implemented by z/OS and JES2 or JES2. The form using only the nodename is not recommended for new installations; if the destination is a NJE Bridge node, the desination is assumed to be Node.SYSTEM and is processed according to the file processing exit entry in `file-exit.cf`. The destination node routes the job to the remote or user at the destination node.

The JES2 and JES3 NJE implementations support defining an abstract identifier (referred to as a DESTID) that indicates a node/userid combination. When possible, we recommend use of DESTIDs to indicate specific destinations, particularly when used to indicate printers or applications defined using the file exits. Use of DESTIDs allows the actual node and userid combination representing a function to change in the future without updating JCL or processes.

Links/Lines

Links and/or lines (the terms are used interchangably) connect nodes together. An NJE network is a fully-specified tree. Each node must know the next link to use to reach 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.

The Spool Area

As mentioned in the NJE overview, files sent and received with NJE are handled as store-and-forward units of work. This means that a file sent from the node is copied into a temporary area (usually considered volatile and not guaranteed) in it's entirety, then examined for where it is to be delivered, and scheduled to be processed accordingly. This temporary area is referred to as the *system spool area*. Files received from other nodes are stored in the system spool until the user indicates what should be done with the file by receiving them into his or her home directory or other location on the NJE node.

In the IBM implmentations, the system spool is an architected part of the operating system and is generally not directly accessible to user applications except via special operations not available to ordinary users. Spooling operations such as printing and file transmission are executed as part of the operating system environment.

In the NJE Bridge implementation, the system spool area is a secured directory containing data files and metadata indicating the ownership and status of spooled data. Userspace commands (heavily modeled on the z/VM implementation of NJE) are provided to manip-

```
Destination = Node

Destination = (Node.Remote-id)

Destination = (Node.Userid)

Destination = (DESTID)
```

Figure 2. Variant Addressing Forms Used with NJE on JES2 and JES3

ulate spooled files, and individual entries are identified by a unique integer, referred to as the file *spoolid*. The system spool directory is shared between all the users of the system, and the size of the filesystem supporting this directory is the maximum size of files that can be transmitted using the NJE Bridge. The tasks in the following sections rely on files arriving and leaving the system via the spool directory. Most user commands refer to the unique spoolid assigned each file as it enters or leaves the spool area.

Users are provided with "fake" devices that access the spool area, corresponding to the z/VM virtual card reader, punch and printer that are simulated by the z/VM Control Program (CP) for each CMS user. Incoming files from other users are marked as "reader" files, outgoing files can be tagged as print or punch files (the convention is that card images (80 column text files) are treated as punch files, other output is tagged as printed output).

The commands provided to manipulate spool files copy data in and out of the spool area, or change metadata related to the spool file in the spooling area. No direct access to the spool area is permitted (except to privileged users, and the Unix superuser). To process data in spool, it must be copied to another location using the "receive" command, or transmitted using one of the file sending commands.

Common NJE Tasks

There are a number of common tasks that users may perform when interacting with a NJE network. The following sections divide those tasks into two categories:

- File-related Tasks

  File-related tasks involve sending or receiving datasets and/or files to NJE destinations. This includes sending and receiving data files, printing output on printers, and processing files from other NJE nodes.

- Message- and Command-related Tasks

  Message and command-related tasks involve sending and receiving interactive messages or sending commands to remote nodes and receiving responses via interactive messages. Commands may also be sent from remote nodes to this node to inquire about processing status or query operational parameters.

File-Related Tasks

The most common NJE user tasks involve sending and receiving data files and printout to other NJE users. The following sections describe the methods to perform these tasks.

Sending Files to Other Users

To send a file to another NJE user, the `sendfile` command is used. For some specialized tasks (like sending print files to a printer, or batch jobs to a remote node for execution), the NJE Bridge provides some convenience commands to set the necessary options to perform such a task with the correct options. Table 1 on page 11 shows the commands available for sending files and the recommended situation for using each command.

Table 1. Commands for Sending Files		
Command	**Description**	**For More Information**
sendfile	sendfile is the basic file transmission tool (most of the other commands invoke sendfile with specialized option sets). If you are not sure which command to use, sendfile is usually the best choice. Files of any size or type can be sent with sendfile.	"sendfile (Send File to NJE Destination)" on page 34
sf	Alias for sendfile.	"sf (Abbreviation for sendfile Command)" on page 36
punch	Send files as card decks (fixed format, 80 columns)	"punch (Send Card Images To NJE Destination)" on page 28
submit	Send files as JOBs to remote batch system (fixed format, 80 columns)	"submit (Send Card Deck to Batch System)" on page 37

A common example of sending a file to another NJE user might appear like this:

```
sendfile joe@snafu data.txt
```

The NJE Bridge uses the common user@node syntax for destination NJE addresses. The designated file is translated to EBCDIC and sent to the remote user.

The punch and submit commands are intended to make particular tasks easier by auto-mating supplying the correct options to sendfile to accomplish the task. For example, the punch command is often used in z/VM environments to send configuration files to virtual machines, similar to:

```
punch tcpip@snavm4 intdown.obeyfile
```

submit allows job streams and JCL to be edited and managed on a discrete system and sent to remote systems for execution, similar to:

```
submit zos112@snavm4 unload.jcl
```

See the notes on the individual commands for more examples of use.

Querying and Manipulating Spool Files

As noted elsewhere, incoming files are delivered to a holding area until processed by the user. The commands in this section allow you to display what files are waiting, get the spool file id for use by commands, and optionally change and/or delete files from the spool area without processing them.

Command	Description	For More Information
Table 2. Commands for Querying and Manipulating Spool Files		
qrdr	Display spool files waiting to be processed.	"qrdr (List Incoming Files in Spool Area)" on page 31
purge	Discard the file without further processing.	"purge (Discard File from Spool Area)" on page 30
peek	peek displays the first few lines of the spoole file, but leaves the file in the spool area.	"peek (Display Lines from Spool File)" on page 26
transfer	Transfer file to another user on this node.	"transfer (Transfer Spool File to Different Userid on This Node)" on page 38

The spool file commands allow you to work with received files without receiving them to local disk space. These commands are modeled on the CMS implementation of NJE, as it is the most interactive implementation.

The qrdr command displays files waiting to be processed in the spool area (the name of the command comes from the z/VM QUERY READER command used to display files in the user's virtual card reader). An example of the command might look like:

```
[root@vsrv053 nje]# qrdr
SPID From      To       FName      FExt       Type C Form ...
0001 DBOYES    ROOT     PROFILE    EXEC       PUN  A STANDARD ...
[root@vsrv053 nje]#
```

The first token (SPID) is the spool identifier. Commands that reference the spool area use this value to identify which file in the spool area to operate on.

The purge command deletes a file from the spool area, identified by the spoolid, and peek displays the first few lines of the file. The transfer command gives the ownership of the file over to another user or pseudouser at the same node as if the file had just been received (e.g., any arrival processing configured by your system administrator will be performed as if the file had just arrived from another site).

Receiving Files From Other Users

Table 3. Commands for Receiving Files		
Command	**Description**	**For More Information**
`receive`	`recieve` is the basic file receiving tool. If you are not sure which command to use, `receive` is usually the best choice. Files of any size or type can be received with `receive`.	"receive (Receive File from Spool Area to Your ID)" on page 32

`receive` has many options, but the most common ones are `-k -o`, and `-b`.

The `-k` option leaves the file in the spool area for later processing, similar to the KEEP option to the z/VM CP SPOOL RDR KEEP command. Receiving a file with this option might look like this:

```
receive -k 123
```

The `-o` option allows specifying what Unix filename and path should be used for the output file. If this option is not used, the file is received with the name found in the NJE dataset headers. An example might look like this (for spool file 3369 with a dataset name of INPUT.DSN.FUBAR):

```
receive -o fubar.data 3369
```

The file data is stored in file `fubar.data`. If the `-o` option is omitted, file INPUT.DSN.FUBAR is created.

The `-b` option indicates that the file should be treated as binary data (and not translated from EBCDIC to ASCII). This option is especially useful for executables and binary datasets. An example of receiving a binary file might appear like:

```
receive -b 123
```

Printing Files on Remote Printers

Table 4. Commands for Printing Files		
Command	**Description**	**For More Information**
`njeprint`	`njeprint` marks the file as printed output and sends the file to the specified NJE destination.	"njeprint (Print File on NJE Printer)" on page 24

NJE printing is a special case of file transfer. In most cases, print sent to remote NJE destinations will be treated as line printer output, but some sites have specialty printers. Contact the system manager of the remote site for details of what printers and options are available.

Interactive Message-Related Tasks

Sending A Message

Table 5. Commands for Sending Interactive Messages		
Command	**Description**	**For More Information**
send	send supports both interactive messages and commands. If you are not sure which command to use, send -m is usually the best choice.	"send (Send Interactive Message or Command)" on page 40

send can transmit both textual messages and commands, depending on the options supplied. In most cases, send -m is the desired option in that it transmits textual data that can be parsed by humans and programs. For example, to send a message to TYRONE@WKST15, the command might look like this:

```
send -m tyrone@wkst15 hi!
TYRONE@WKST15: Hi back!
```

Receiving A Message

There are no special commands to receive NJE messages. The message is written directly on all terminals where your userid is logged in. If you have multiple login sessions on that machin, the message is written on all login sessions.

Executing Commands on Remote Nodes

Some NJE nodes provide the ability to execute commands on the remote NJE system via NJE commands. The send -c command tags a message as a command and returns the response from the remote system via messages.

Sending Commands

To send a command to a remote NJE node, use the following command:

```
send -c @DEVHUB CPQ TIME
SYSTEM@DEVHUB: CPQ: TIME IS 14:09:23 EDT Monday 2015-05-11
```

Note that when sending commands, no userid part is supplied in the NJE address. Commands are always processed by the "SYSTEM" userid, and do not need userid information.

The validity and content of the command response is determined by the destination node. Consult the documentation for the remote node to determine what commands are valid. Note also that some NJE implementations are case-sensitive with regard to commands (eg $D xxx may work, but $d xxx may not).

Interpreting Command Responses

As noted above, command responses and formats are determined by the remote system. Consult the documentation for the remote NJE implementation to determine what commands can be used.

Commands Available In the NJE Bridge

Consult the documentation for your system for information how to invoke these commands from your platform.

Table 6. Commands Available from Remote Nodes

Command	Description
CPQ TIME	Displays system clock time at remote node (with time zone and offset).
CPQ USERS	Displays number of logged in users on the NJE node itself.
CPQ NAMES	Displays usernames of logged in users and associated terminal ids.
CPQ CPU	Displays system level identifier (as shown by uname -a).

End User Command Reference

Reading Syntax Diagrams

To read a syntax diagram for entering a command, follow the path of the line. Read from left to right and from top to bottom.

Syntax Items

Syntax items, such as a keyword or a variable, can be:

- On the line (required element)

- Above the line (default element)

- Below the line (optional element)

Symbols

Enter these symbols exactly as they appear in the syntax diagram.

Symbol(s)	Description
Table 7. Symbols in Syntax Diagrams	
*	Asterisk
{ }	Braces
:	Colon
,	Comma
=	Equal Sign
-	Hyphen
( )	Parentheses
.	Period
	Space
"	Quotation mark
'	Single quotation mark

Variables

Italicized lowercase items such as *<var_name>* indicate variables. In this example, you can specify a *<var_name>* when you enter the *cmd_name* command.

```
┌── Example Syntax Diagram for cmd_name ────────────────────────┐
│                                                               │
│  ►►──cmd_name──<var_name>─────────────────────────────────►◄  │
│                                                               │
└───────────────────────────────────────────────────────────────┘
```
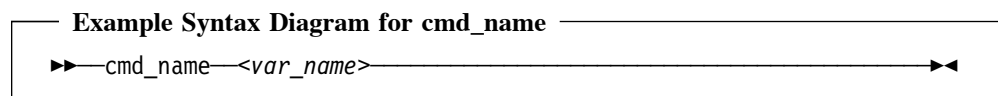
Repetition

An arrow returning to the left means that the item can be repeated. A character within the arrow means that you must separate repeated items with that character.

A footnote (1) by the arrow refers to a limit that tells how many times the item can be repeated.

Required Choices

When two or more items are in a stack and one of them is on the line, you must specify one item.

In this example, you must choose A, B, or C.

```
┌─ Example for Required Choices ─────────────────────────────┐
│                                                            │
│  ►►──cmd_name──┬─A─┬──────────────────────────────────►◄   │
│               ├─B─┤                                        │
│               └─C─┘                                        │
│                                                            │
└────────────────────────────────────────────────────────────┘
```
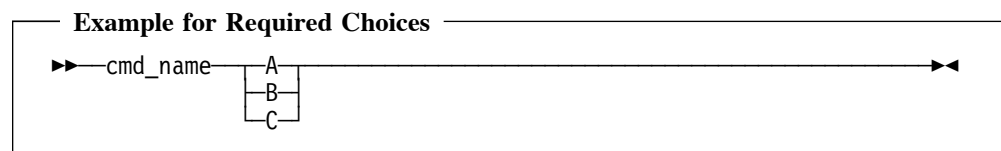
Optional Choices

When an item is below the line, that item is optional. In the first example, you can select A or nothing at all.

```
┌─ Example for Optional Choices ─────────────────────────────┐
│                                                            │
│  ►►──cmd_name──┬───┬──────────────────────────────────►◄   │
│               └─A─┘                                        │
│                                                            │
└────────────────────────────────────────────────────────────┘
```
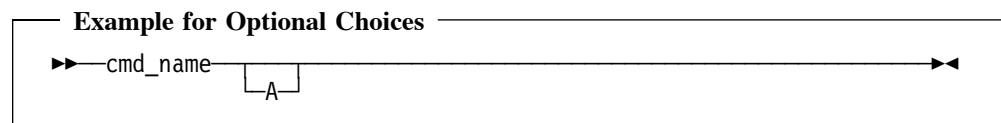
When two or more items are in a stack below the line, all of them are optional. In the second example, you can choose A, B, C, or nothing at all.

```
┌────────────────────────────────────────────────────────────┐
│                                                            │
│  ►►──cmd_name──┬───┬──────────────────────────────────►◄   │
│               ├─A─┤                                        │
│               ├─B─┤                                        │
│               └─C─┘                                        │
│                                                            │
└────────────────────────────────────────────────────────────┘
```
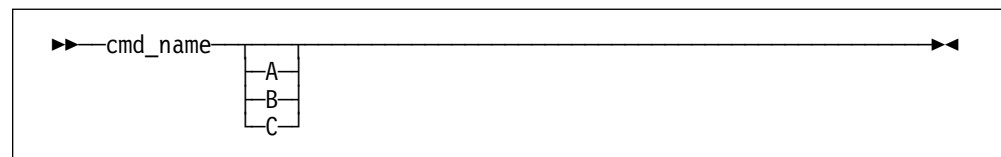
A stack of items followed by an arrow returning to the left indicates that you can select more than one item, or in some cases, repeat a single item.

Defaults

Defaults are above the line. The default is selected unless you override it, or you can select the default explicitly. To override the default, include an option from the stack below the line.

In this example, A is the default. Select either B or C to override A.

```
┌──  Example for Default Values ─────────────────────────────────────┐
│                          ┌─A─┐                                     │
│  ►►──cmd_name───────────┤ B ├──────────────────────────────►◄     │
│                          └─C─┘                                     │
└────────────────────────────────────────────────────────────────────┘
```
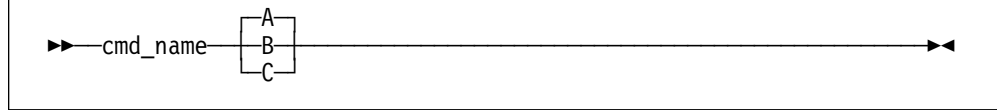
If you have trouble interpreting the syntax diagrams, please contact SNA support or submit a readers comment form.

File-Related Commands

njeMeta
(Display NJE Metadata)

Purpose

The njeMeta command displays a JSON-formatted version of the NJE file metadata headers and trailers.

Format
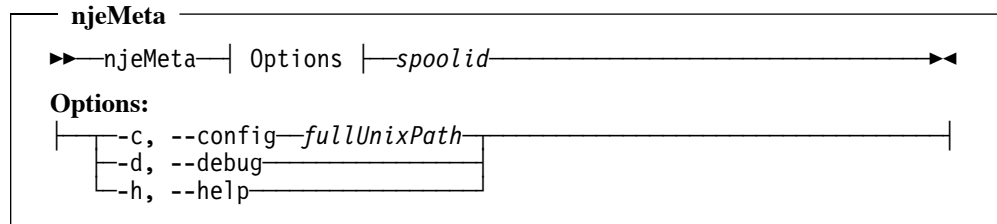
```
         ┌── njeMeta ──────────────────────────────────────────────────────┐
  ►►──njeMeta──┤ Options ├───spoolid────────────────────────────────────────►◄

  Options:
  ├───┬── -c, --config──fullUnixPath──┬──────────────────────────────────┤
      ├── -d, --debug─────────────────┤
      └── -h, --help──────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

config
    The location of the configuration file.

fullUnixPath
    Full Unix path and filename for alternate `nje.cf` configuration file. Refer to the Planning and Configuration Reference manual for information about the `nje.cf` file.

debug
    Print processing details for SNA support.

help
    Displays a help summary.

spoolid
    NJE spool id for this file. Refer to the output of `qrdr` to obtain spool file information.

Usage

The `config` option allows supplying an alternate `nje.cf` file to change system defaults temporarily. If this option is not specified, the file `/etc/nje/nje.cf` is used.

The `debug` option is used for diagnostics purposes. The output is meaningful only to the application developer, and should be used only when requested by SNA support.

The `help` displays an abbreviated help summary.

Examples

An example of the output of `njeMeta` might appear similar to:

```
[root@vsrv053 ~]# qrdr
SPID From      To        FName      FExt          Type C Form ...
0001 DBOYES    ROOT      PROFILE    EXEC          PUN  A STANDARD...

[root@vsrv053 ~]# njeMeta 0001

{
  "JOB_HEADER" : {
     "NJHLEN" : 204,
     "NJHFLAGS" : 0,
     "NJHSEQ" : 0,
     "GENERAL_SECTION" : {
     "NJHGLEN" : 200,
     "NJHGMAJ" : 0,
        "NJHGMMOD": 0,
            .
     (many lines of output)
            .
            .
            .
     }
  }

}
[root@vsrv053 ~]#
```

Comments

1. Normally, njeMeta is only used for diagnostic purposes. The descriptions of field names and values are drawn from the IBM NJE Protocol and Data Formats reference manual.

njeprint
(Print File on NJE Printer)

Purpose

The `njeprint` command prints output files on NJE printers.

Format
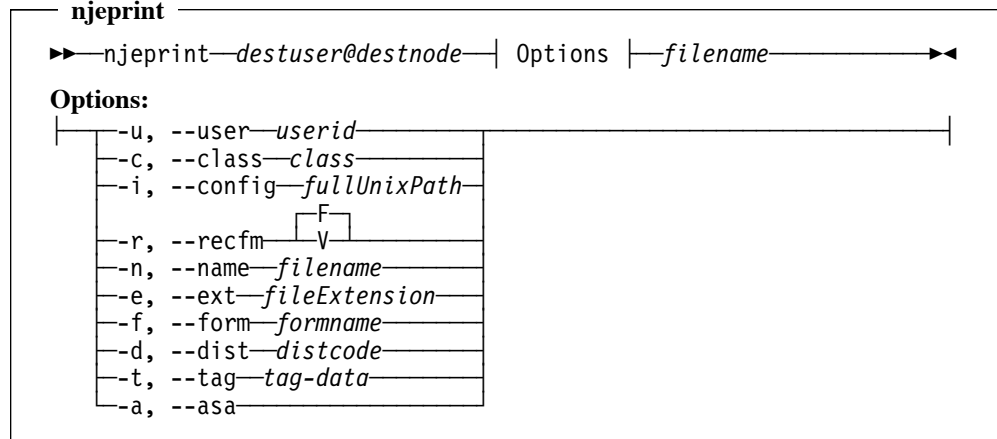
```
          ┌─ njeprint ──────────────────────────────────────────────────┐
          │                                                              │
  ►►──njeprint──destuser@destnode──┤ Options ├──filename──────────────►◄

  Options:
  ├──┬──-u, --user──userid────────────┬───────────────────────────────────┤
     ├──-c, --class──class────────────┤
     ├──-i, --config──fullUnixPath────┤
     │              ┌─F─┐             │
     ├──-r, --recfm──┴─V─┴────────────┤
     ├──-n, --name──filename──────────┤
     ├──-e, --ext──fileExtension──────┤
     ├──-f, --form──formname──────────┤
     ├──-d, --dist──distcode──────────┤
     ├──-t, --tag──tag-data───────────┤
     └──-a, --asa─────────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

destuser
> The NJE userid of the printer.

destnode
> The NJE node of the printer.

filename
> The file name to be printed.

user
> Print the file as another user. (this command requires superuser authority)

userid
> The userid to print the file as. This user will be placed in the USER field of the file to be printed and will be used as the remote userid.

class
> The output class of the file. Output classes may be A-Z and 0-9

config
> Location of alternate configuration file.

recfm
> Fixed or variable (F/V)

name
> The name used for the CMS filename. Files are printed using this name; if not specified, the first 8 characters of the filename are used.

filename
> Short 8 character name for the file.

ext

Filetype of the CMS filename.

form

Use a form code for this file. This may specify special processing for your files; see your system administrator if specifying this parameter for valid values.

formname

Form code to use for this file.

dist

Distribution code to use for this file. This may specify special processing for your files; see your system administrator if specifying this parameter for valid values.

distcode

Distribution code to use for this file.

tag

Special tag information for this file. This may specify special processing for your files; see your system administrator if specifying this parameter for valid values.

tag-data

Tag information for this file.

asa

This file contains print data (valid only for print files).

Usage

This command has many variations; see the examples for details.

Examples

To print a file on the default form:

```
njeprint va1p8@print foo.txt
```

To print a file using a special form:

```
njeprint va1p8@print --form PORTRAIT foo.txt
```

Comments

1. The `--user` option is restricted to users who are members of the NJE administrative group.

peek
(Display Lines from Spool File)

Purpose

Format
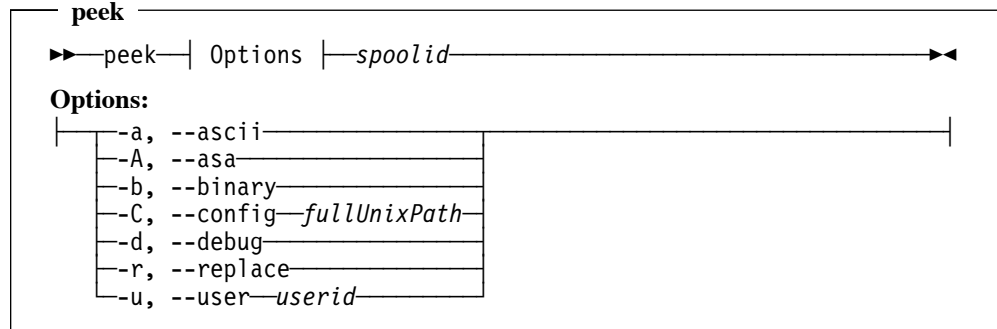
```
        ┌─ peek ──────────────────────────────────────────────────┐
 ►►──peek─┤ Options ├──spoolid──────────────────────────────────►◄
 Options:
 ├──┬──-a, --ascii──────────────────┬────────────────────────────┤
    ├──-A, --asa───────────────────┤
    ├──-b, --binary────────────────┤
    ├──-C, --config──fullUnixPath──┤
    ├──-d, --debug─────────────────┤
    ├──-r, --replace───────────────┤
    └──-u, --user──userid──────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

ascii
Display output in ASCII.

asa
Convert file formatting codes to equivalent print file commands and display output.

binary
Display output in untranslated form.

config
Location of configuration file.

fullUnixPath
Full Unix path to configuration file.

debug
Display debugging output. Use this option when instructed to do so by SNA support.

replace
Replace the file specified.

user
Act on behalf of the user specified.

userid
User to masquerade as when executing the command.

spoolid
Operate on "spoolid" when executing the command. If not specified, the first file in your virtual reader is used.

Usage

peek displays the first few lines of the file.

Examples

To display the first few lines of the file, execute:

```
peek --ascii 80
```

Comments

1. The `user` option is restricted to users who are members of the NJE administration group.

punch
(Send Card Images To NJE Destination)

Purpose

This command sends card images (fixed format, 80-column) to a NJE user.

Format
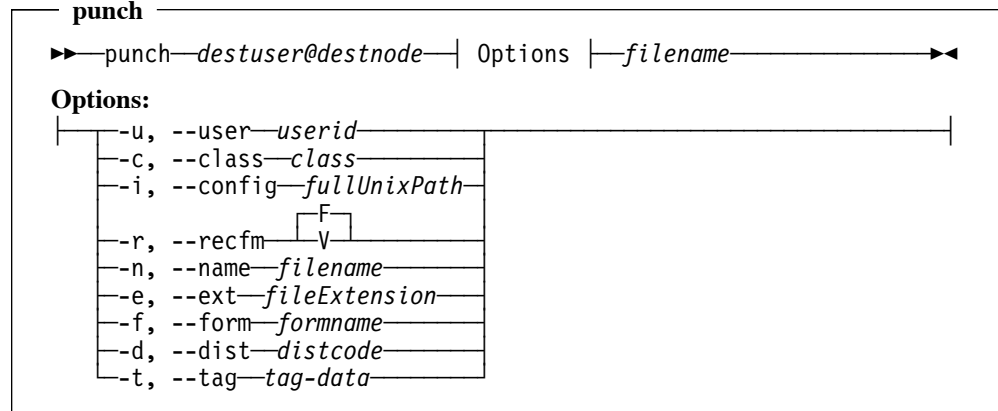
```
      ┌─── punch ──────────────────────────────────────────────────┐
      │                                                             │
►►──── punch ── destuser@destnode ─┤ Options ├── filename ──────────────►◄

Options:
├──┬─────────────────────────────────────────────┬──────────────────────┤
   ├── -u, --user ── userid ────────────────────┤
   ├── -c, --class ── class ────────────────────┤
   ├── -i, --config ── fullUnixPath ────────────┤
   │                        ┌─ F ─┐              │
   ├── -r, --recfm ─────────┴─ V ─┴──────────────┤
   ├── -n, --name ── filename ──────────────────┤
   ├── -e, --ext ── fileExtension ──────────────┤
   ├── -f, --form ── formname ──────────────────┤
   ├── -d, --dist ── distcode ──────────────────┤
   └── -t, --tag ── tag-data ───────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

destuser
> Destination NJE userid.

destnode
> Destination NJE node name

filename
> Filename to be transmitted. File must be 80 columns or less.

user
> Act on behalf of specified user.

userid
> User to act on behalf of.

class
> Use specified class for file. Valid values are A-Z and 0-9.

config
> Full Unix pathname for alternate configuration file.

recfm
> Fixed or variable (F/V)

name
> CMS filename to be used in transit. If not specified, the first 8 characters of the filename is used.

ext
> CMS filetype to be used in transit. If not specified, the extension FILE is used.

form
>   Use formname for processing.

dist
>   Distribution code for processing.

tag
>   Use tag data for further processing.

Usage

This command is used as a shortcut for sendfile. It sends card decks to batch users or other users where cards are input.

Examples

To send a card deck to z/OS:

```
punch vse@node1 test.jcl
```

Comments

1. The user option is restricted to users with ADMGROUP authority.

purge
(Discard File from Spool Area)

Purpose

This command purges a entry from the spool area.

Format
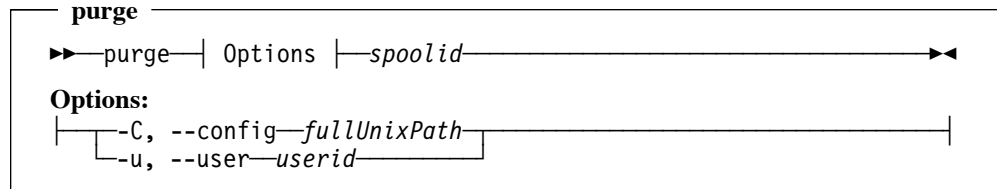
```
          ┌─ purge ────────────────────────────────────────────┐
  ►►──purge──┤ Options ├───spoolid──────────────────────────────►◄

  Options:
  ├──┬─────-C, --config──fullUnixPath──┬──────────────────────────┤
     └─────-u, --user──userid──────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

config
Specifies full pathname to alternate configuration.

user
Act on behalf of another user. The userid specified here is the alternate userid.

spoolid
Spool id to act on.

Usage

This command purges a file from the spool area. If no spoolid is specified, the first file in the spool area is purged.

Examples

To remove entry 78 from the spool area, type:

```
purge 78
```

Comments

1. The user entry is restricted to those with ADMGROUP authority.

qrdr
(List Incoming Files in Spool Area)

Purpose

This command is used to get information about files in the spool area for this user.

Format
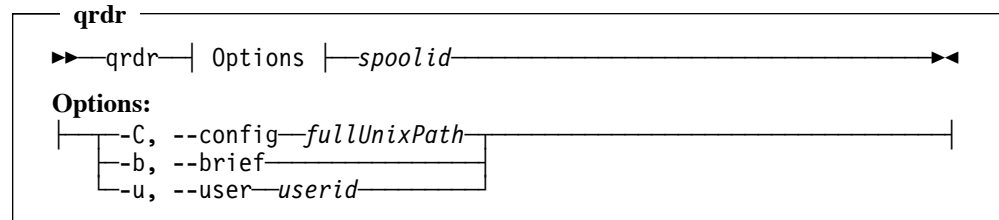
```
                  ┌─ qrdr ─────────────────────────────────────────────────┐
      ►►──qrdr──┤ Options ├──spoolid──────────────────────────────────►◄
      Options:
      ├──┬─────-C, --config──fullUnixPath─┬────────────────────────────┤
         ├──-b, --brief────────────────┤
         └──-u, --user──userid──────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

config
>   Full Unix path name to alternate config file.

brief
>   Display brief information about each spool file.

user
>   Act on behalf of userid.

spoolid
>   Specify which spoolid to operate on. If no spoolid is supplied, the first spoolid in the queue is used.

Usage

This command is primarily used to query information about transmitted files for use by the receive command (or any command referring to files in the spool area).

Examples

To list the files waiting in the spool area:

```
qrdr
```

Comments

1. The user option is restricted to users with ADMGROUP authority.

receive
(Receive File from Spool Area to Your ID)

Purpose

This command receives a file from the spool are to your local disk space.

Format
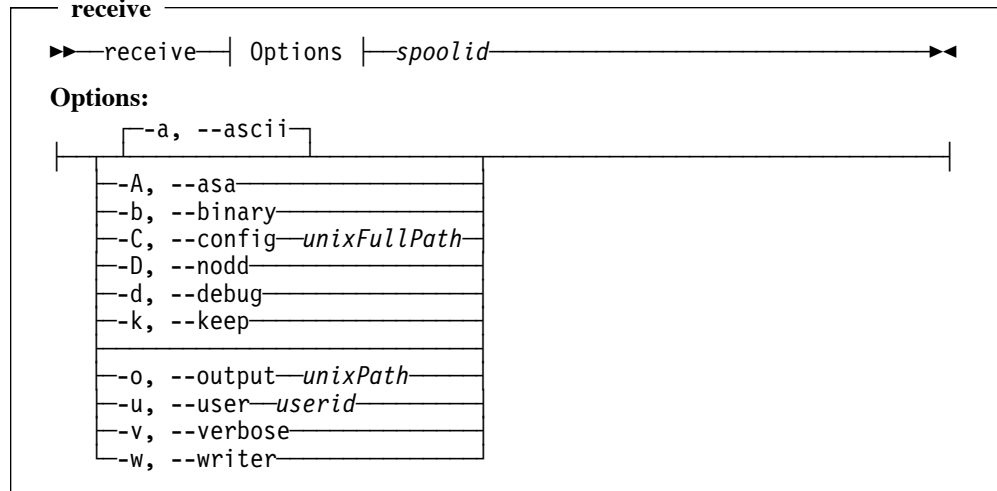
```
         ┌─ receive ──────────────────────────────────────────────────┐
►►──receive─┤ Options ├──spoolid───────────────────────────────────────►◄

Options:
              ┌──-a, --ascii─┐
├─────────────┴──────────────┴──────────────────────────────────────────┤
    ├──-A, --asa──────────────────────┤
    ├──-b, --binary───────────────────┤
    ├──-C, --config──unixFullPath─────┤
    ├──-D, --nodd─────────────────────┤
    ├──-d, --debug────────────────────┤
    ├──-k, --keep─────────────────────┤

    ├──-o, --output──unixPath─────────┤
    ├──-u, --user──userid─────────────┤
    ├──-v, --verbose──────────────────┤
    └──-w, --writer───────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

ascii

File contents are translated from EBCDIC to ASCII.

asa

File contents are translated from machine-dependent carriage control to ASA-format (aka "Fortran") carriage controls. Applies to print files only.

binary

Receive file as-is (no translation). Files containing binary data are used with this command.

config

Specify full Unix path to alternate configuration file.

nodd

Ignore DD/STEP value from dataset header when building output name.

debug

Print debugging information. Use only when requested by SNA support.

keep

Don't delete the spool file on successful processing. The file is kept for future processing.

replace

Overwrite file if it already exists.

output

Output to named file. If the file exists, the command aborts.

user

    Act on behalf of user.

verbose

    Output details about each dataset processed. This is especially useful when receiving multiple output streams in a single file.

writer

    Use the writer name from dataset header for filename. This parameter is useful when referring to output from a specific writer stream.

spoolid

    Spoolid to operate on. If not specified, the first file in the reader is used.

Usage

This command specifies how to receive a file.

Examples

To receive file 79 to disk:

```
receive 79
```

To receive file 79 to disk without translation to ASCII:

```
receive --binary 79
```

Comments

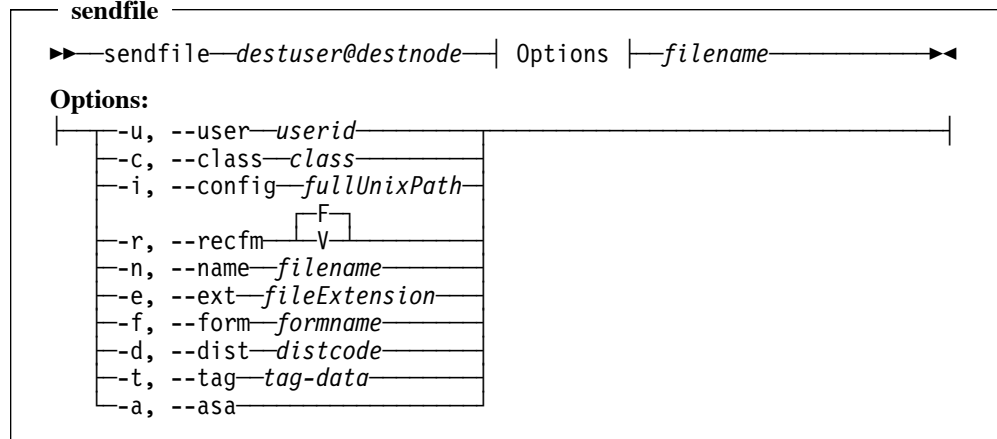1. The user option is restricted to users with ADMGROUP authority.

sendfile
(Send File to NJE Destination)

Purpose

This command sends an arbitrary file to another user.

Format

```
┌─── sendfile ──────────────────────────────────────────────────┐
│ ►►──sendfile──destuser@destnode──┤ Options ├──filename──────►◄ │
│                                                                │
│ Options:                                                       │
├─┬──-u, --user──userid──────────────────┬──────────────────────┤
  ├──-c, --class──class──────────────────┤
  ├──-i, --config──fullUnixPath──────────┤
  │                     ┌─F─┐            │
  ├──-r, --recfm────────┴─V─┴────────────┤
  ├──-n, --name──filename────────────────┤
  ├──-e, --ext──fileExtension────────────┤
  ├──-f, --form──formname────────────────┤
  ├──-d, --dist──distcode────────────────┤
  ├──-t, --tag──tag-data─────────────────┤
  └──-a, --asa───────────────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

destuser
> The NJE userid of the destination user.

destnode
> The NJE node name of the destination user.

filename
> The Unix filename of the file to send.

user
> Act on behalf of the specified user.

class
> Specify class of file when transmitted. Valid values are A-Z and 0-9.

config
> Specify full Unix path to file containing alternate configuration.

recfm
> Fixed or variable (F/V)

name
> CMS filename to use when transmitting file. If not specified, the first 8 characters of the filename are used.

ext
> CMS filetype to use when transmitting file. If not specified, the first 8 characters of the file extension are used. If no file extension is available, a extension of FILE is used.

form

Use form name when sending file. If no form name is specified, a form name of STANDARD is used.

dist

Distribution code to be used when transmitting file. See your local system administrator for valid values for this field.

tag

Tag data to use when transmitting file. This data is not checked.

asa

Indicates this file contains print data that needs conversion. The file is sent as a print file and marked as such.

Usage

This command processes and sends a file to another user.

Examples

To send a file to another user:

```
sendfile joe@user1 foo.c
```

To send a file with a special form:

```
sendfile foo@print --form PORTRAIT file.ext
```

To send a file using a different name:

```
sendfile joe@node1 --name FILE22 --ext WORKFILE filethatistoolong.example
```

Comments

1. The user option is restricted to users with ADMGROUP authority.

sf
(Abbrevation for sendfile Command)

Purpose

sf is a convenient abbreviation for sendfile. The function and and syntax of the command are identical to the sendfile command shown on "sendfile (Send File to NJE Destination)" on page 34.

Format

The syntax for sf is identical to sendfile(see "sendfile (Send File to NJE Destination)" on page 34).

Parameters

The parameters for sf are identical to sendfile(see "sendfile (Send File to NJE Destination)" on page 34).

Usage

See the sendfile command for more details.

Examples

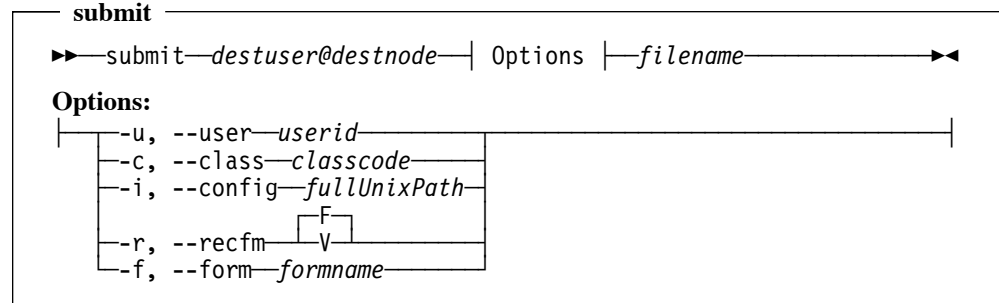See the sendfile command for more details.

Comments

1. See the sendfile command for more details.

submit
(Send Card Deck to Batch System)

Purpose

This command submits a batch job to a remote NJE system.

Format

```
          ┌─ submit ──────────────────────────────────────────────────┐
►►──submit──destuser@destnode─┤ Options ├───filename───────────────────►◄

Options:
├──┬──-u, --user──userid──────────────┬───────────────────────────────┤
   ├──-c, --class──classcode──────────┤
   ├──-i, --config──fullUnixPath──────┤
   │                ┌─F─┐             │
   ├──-r, --recfm───┴─V─┴─────────────┤
   └──-f, --form──formname────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

user
Act on behalf of the specified user.

class
Send file with class value of x. Valid values are A-Z and 0-9.

config
Full Unix path to alternate configuration file.

recfm
Fixed or variable (F/V)

form
Specify form name for transmitted file. See your local administrator for valid values.

Usage

This is a shortcut for "sendfile user@node --recfm F filename". Files must be 80 columns or less.

Examples

To send a job to a remote z/OS system:

```
submit zos112@snavm4 foo.jcl
```

Comments

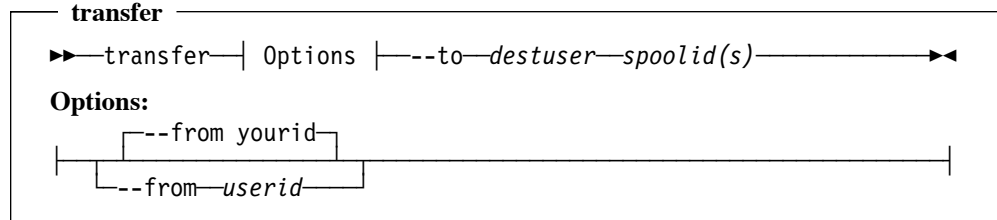1. Files will be truncated at 80 columns.

transfer
(Transfer Spool File to Different Userid on This Node)

Purpose

This command transferrs a spool file to another userid on the same node.

Format

```
        ┌─────── transfer ───────────────────────────────────────┐
        ►►──transfer──┤ Options ├──--to─destuser─spoolid(s)──────────►◄

        Options:
                   ┌───--from yourid───┐
        ├──────────┼───────────────────┼─────────────────────────────┤
                   └───--from──userid───┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

from
  Transfer files from "yourid"

userid
  Transfer files from arbitrary userid. This option requires administrator privileges.

--to
  Transfer files to user. Files are treated as newly arrived, and all processing is performed as if the file was new.

destuser
  Destination userid for this file.

spoolid
  Spool id to operate on.

Usage

This command provides a way to transfer ownership of a NJE file in the spool area to another user for processing. It is commonly used to transfer files to service machines for further processing.

Examples

To transfer files to another user:

```
transfer --to printers 79
```

Comments
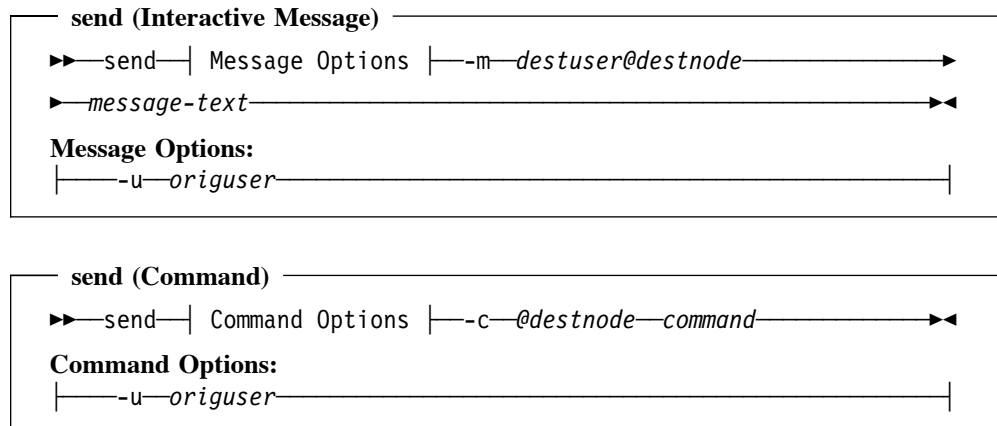
1. transfer is limited to userids on the same NJE node.

Message-Related Commands

send
(Send Interactive Message or Command)

Purpose

This command sends a interactive message to a destination user, or a command to a remote NJE node.

Format

```
┌─ send (Interactive Message) ─────────────────────────────────┐
│                                                              │
│  ►►──send──┤ Message Options ├──-m─destuser@destnode────────► │
│  ►──message-text──────────────────────────────────────────►◄ │
│                                                              │
│  Message Options:                                            │
│  ├────-u─origuser─────────────────────────────────────────┤  │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

```
┌─ send (Command) ─────────────────────────────────────────────┐
│                                                              │
│  ►►──send──┤ Command Options ├──-c─@destnode─command───────►◄ │
│  Command Options:                                            │
│  ├────-u─origuser─────────────────────────────────────────┤  │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

Parameters

The parameter definitions for the long form of the options are referenced here. The short (one-character) version of the option is the minimum abbreviation.

destuser
Destination userid for NJE user. Commands do not have destination users, and are essentially omitted.

destnode
Destnation node for NJE user.

message-text
Text of interactive message. The message is limited to 140 characters.

command
Command to be executed at remote node. The results are returned by NJE messages.

origuser
Act on behalf of NJE user. This command is restricted to users with ADMGROUP privileges.

Usage

Interactive messages and commands are essentially the same thing. Command responses are returned via interactive messages and are system-dependent.

Examples

To send a interactive message to user JOE at node NODE1:

```
send -m joe@node1 Hi!

NODE1(joe): Hi back!
```

To send a command to a remote node:

```
send -c @node1 CPQ TIME

MSG FROM NODE1: TIME IS 00:32:14 EST...
```

Comments

1. The u option is limited to users with ADMGROUP privileges.

# Built-In Commands Executable By Remote Nodes

This section describes the built-in commands that other nodes can use to get information about this node. The commands available to you at other nodes are dependent on the remote system, and may be restricted by the remote system's policy on remote command execution.

CPQ TIME
(Query System Time At Remote Node)

Purpose

This remote command queries the time at this node remotely. The system time is returned in local format.

Format

```
►►──CPQ TIME──────────────────────────────────────────────────►◄
```
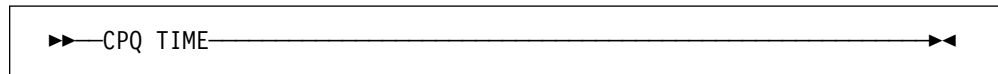
Parameters

There are no parameters for this command.

Usage

This command can be used as a keepalive to ensure the link is actually working.

Examples

To query the remote time:

```
send -c @node1 CPQ TIME

MSG FROM NODE1: TIME IS 00:03:14 EST ....
```

Comments

There are no comments.

CPQ USERS
(Query Number of Logged In Users at This Node)

Purpose

This command allows the number of users to be queried at the local node.

Format

```
►►──CPQ USERS──────────────────────────────────────────────►◄
```

Parameters

There are no parameters to this command.

Usage

The remote users can query the number of users at this node as a keepalive.

Examples

To query a remote node from RSCS:

SMSG RSCS CMD PRINT CPQ USERS

17:52:08 MSG FROM RSCS: From PRINT(SYSTEM): CPQ: 52 USERS, 0 DIALED...

Comments

There are no comments.

CPQ NAMES
(Display Logged In Usernames at This Node)

Purpose

Format

```
►►──CPQ NAMES──────────────────────────────────────────────►◄
```

Parameters

There are no parameters to this command.

Usage

The CPQ NAMES command shows a list of logged in users at the remote site.

Examples

To show a list of logged in users at a site:

```
SMGS RSCS CMD PRINT CPQ NAMES

* MSG CMD: From PRINT(SYSTEM):
* MSG CMD: JOE        (/dev/tty34)
* MSG CMD: FRED       (/dev/tty21)
```

Comments

There are no comments for this command.

CPQ CPU
(Display System Info for This Node)

Purpose

This command displays the information for this node from "uname * -a".

Format

```
►►──CPQ CPU──────────────────────────────────────────────────►◄
```

Parameters

There are no parameters for this command.

Usage

This command displays the local architecture of the NJE Bridge.

Examples

To remotely query the NJE Bridge architecture:

send -c @node1 CPQ CPU

MSG FROM PRINT(SYSTEM) : CPU x86_64 uname x86_64, NJE Build 1495

Comments

There are no comments for this command.

If You Have Trouble

If You Have Trouble

Gathering Information About The Problem

Logging and Log Files
The application logs information to the general Unix system log using syslog's local0 facility.

Setting An Elevated Debugging Level
From the ucp control panel tool, use the command LOGLEVEL 6. This is the highest setting and will log the most debug data.

What to Capture
To assist in debugging, capture the following files and submit them to support as email attachments when requested.

- The relevant portion of the log file from syslog.

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

Reporting A Problem

Use the instructions supplied to 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.

Helpful Hints While Working on the Problem
The following hints will help speed resolution of your issue.

- When reporting a problem, send problem reports to `support@sinenomine.net`. You will receive an automated response containing your case number.

- While working with support, please preserve the subject line of the email and ensure that "[sinenomine.net #nnnnn]" remains in the first 16 characters of the subject line of your replies. This allows the ticketing system to associate your replies with the problem report automatically, and keeps the flow of conversation connected.

- Please report only one issue per case. Reusing incident numbers confuses issues, and will delay resolution.

Appendices

Appendix A. Other Helpful References

The NJE protocol is well-documented. The following sections describe where to find more information on the protocol and on the IBM implementations of NJE.

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

NJE Protocol References

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 publically available at:

- 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 Peter Olenick's (of Princeton University) document "Definition of the BITNET II Protocol: A Technical Overview of VMNET". It is available from:

- http://www.funet.fi/pub/netinfo/BITNET/brf0002.text

Connecting to IBM Implementations

RSCS Connections

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

VSE/POWER Connections

For VSE, there are two primary reference books on NJE configuration. These books are:

- VSE/POWER Networking,IBM Document Number SC33-6735

  http://publibfp.boulder.ibm.com/epubs/pdf/iesnpe20.pdf

- VSE/POWER Administration and Configuration, IBM Document Number SC33-6733

  http://publibfp.boulder.ibm.com/epubs/pdf/iespme30.pdf

z/OS and JES2 Connections

NJE configuration documentation for JES version 2 (5740-XC6) is available in the following manuals:

- 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

z/OS and JES3 Connections

NJE documentation for JES3 version 4 (5695-048) is available in the following manuals:

- 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

iSeries Connections

NJE setup examples are best documented in a IBM redbook shown below. The manuals for your release of the iSeries operating system are likely quite dated.

To read more about iSeries NJE implementation, consult:

- VSE/POWER and OS/400 NJE Configuration Guide, IBM document number GG24-4259-00

  Note that this document includes only information about the IBM Systems Network Architecture (SNA) implementation of NJE; the utilities and system configurations used to transmit and receive information are the same when used with the NJE Bridge.

Appendix B. Your Comments are Welcome!

NJE Bridge: End User Command Reference

Use this form to tell us what you think about this manual, or send email to support AT sinenomine.net. If you have found errors in it, or if you want to express your opinion about how the information is organized or presented, or make suggestions for improvements, this is the form to use.

Be sure to include your name and email address (and optionally, a telephone number with country and area code) if you would like a reply.