SNA NRJE User/Programmer Reference Manual

HP 3000 MPE/iX Computer Systems

Edition 5

HEWLETT®
PACKARD

Manufacturing Part Number: 30292-90006
E1092

U.S.A. October 1992

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

Acknowledgments

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Contents

1. Introducing SNA NRJE

2. Getting Started

3. User Commands

4. Job Output

5. User Intrinsics

6. NRJE Manager Intrinsics

A. Parsing Algoithms and User Exit Procedures

Glossary

Figures

Tables

Preface

This manual describes Hewlett-Packard Company's Systems Network Architecture Network Remote Job Entry (SNA NRJE) for both the MPE V (NRJE/V) and MPE XL (NRJE/XL) operating systems. Systems Network Architecture (SNA) is a comprehensive specification for distributed data processing networks developed by IBM.

NOTE In this manual, the term **NRJE** is used when the information being given is true for both NRJE/V and NRJE/XL. The terms **NRJE/V** and **NRJE/XL** are used when a distinction between the two subsystems is necessary.

NRJE allows HP 3000 users to submit batch jobs to a remote host system for processing. The host system uses one of the supported job entry subsystems in an SNA environment.

NRJE can be used in two ways: with commands or with programmatically callable intrinsics. Commands are processed by a user interface. Both user commands and NRJE manager commands are available. NRJE intrinsics are called from HP 3000 user-written programs. Both user intrinsics and NRJE manager intrinsics are available.

Audience

Two kinds of NRJE users exist:

- *NRJE managers*, those with Node Manager (NM) user capability. An NRJE manager is responsible for managing the NRJE workstation. The NRJE manager has the capability of controlling all files transmitted from the workstation and all data sets returned from the host to the workstation.
- *Users*, those without NM user capability. An NRJE user submits batch jobs to a host system for processing.

An NRJE manager can do everything that a user can do, with some extended capabilities. An NRJE manager can determine and alter the status of the job of any user, and can cancel the job of any user.

An NRJE manager plans and monitors the operation of the NRJE subsystem. NRJE manager responsibilities include maintenance and problem solving. Logging capabilities allow an NRJE manager to monitor network and NRJE events. Tracing capabilities permit an NRJE manager to examine internal NRJE processes and the use of NRJE commands and intrinsics. Logging and tracing capabilities are provided through the HP 3000 Multiprogramming Executive (MPE) operating system, and through the HP SNA link product.

NOTE

In this manual, the term **SNA link product** is used when the information being given is true for all of the SNA link products: SNA Link/V, SNA/SDLC Link/XL, and SNA/X.25 Link/XL. The terms **SNA Link/V**, **SNA/SDLC Link/XL**, and **SNA/X.25 Link/XL** are used when a distinction among the three products is necessary.

An NRJE user submits jobs to a remote host system. Users can determine the status of their jobs to be submitted to a host, alter the status of their jobs, and cancel their jobs. Users also can display information about the status of NRJE.

The NRJE manager is directly responsible for NRJE operation and usage, but another level of responsibility is assigned to the HP 3000 SNA node manager. The HP 3000 SNA node manager is responsible for overall HP-IBM data communications. The node manager builds the configuration file, which defines the following:

- The communications link
- The SNA node and logical units
- Logging and tracing characteristics
- The SNA workstation

The node manager also monitors operations of the node. Node management functions are described in the *SNA Link Services Reference Manual* and the *SNA Link/XL Node Manager's Guide*.

Organization

This manual is divided into the following chapters and appendix:

Chapter 1 , “Introducing SNA NRJE,” provides an overview of NRJE, its operating environment, what it can do, and hardware and software requirements.

Chapter 2 , “Getting Started,” describes, in general terms, how to get NRJE to submit a job to your host and how output is handled for your job.

Chapter 3 , “User Commands,” provides a description of every NRJE user command, its syntax, and its parameters. A discussion and example follow the command description. A text reference relates a command to NRJE intrinsics. Commands are in alphabetic order for easy reference.

Chapter 4 , “Job Output,” provides a description of how SNA NRJE handles job output from a host system to the HP 3000. NRJE job management and output management capabilities are described. Carriage control translation, special forms processing, and use of the HP2680A laser printer are discussed. The relationship between host output classes and logical writers is described.

Chapter 5 , “User Intrinsics,” has a description of every NRJE user intrinsic, its syntax, and its parameters. Parameter data types are identified in the syntax description. A discussion follows each intrinsic description. A text reference relates each intrinsic to a command. Intrinsics are in alphabetic order for easy reference.

Chapter 6 , “NRJE Manager Intrinsics,” is a description of every NRJE manager intrinsic, its syntax, and its parameters. Parameter data types are identified in the syntax description. A discussion follows each intrinsic description. A text reference relates each intrinsic to a command. Intrinsics are in alphabetic order for easy reference.

Chapter 7 , “MPE Support Utilities and Commands,” describes the MPE commands and subsystems that can be used with NRJE.

Appendix A , “Parsing Algoithms and User Exit Procedures,” describes two exits from SNA NRJE, and describes and lists the procedures called from the exits:

- Host “Job Received” message parsing procedure
- Host banner page decode procedure

These procedures are supplied with SNA NRJE and are required for the full job and output management capabilities of NRJE to function.

Related HP Publications

- *Installing and Troubleshooting SNA NRJE, Node Manager's Guide*
- *SNA NRJE Node Manager's Guide*
- *SNA Link Services Reference Manual*
- *SNA Link/XL Node Manager's Guide*
- *Getting Started With SNA Node Management*
- *HP30020A/B Intelligent Network Processor (INP) Installation and Service Manual*
- *MPE Commands Reference Manual*
- *Console Operator's Guide*
- *MPE System Utilities*
- *System Operation and Resource Management Reference Manual*
- *MPE Software Pocket Guide*
- *MPE Quick Reference Guide*
- *Data Communications Handbook, Fundamental Chapter*
- *Data Communications Handbook, Chapter H, SNA NRJE Network Remote Job Entry*
- *MPE XL Commands Reference Manual*

Series 9xx System Administrator manuals:

- *System Startup and Shutdown*
- *Managing Jobs and Sessions*
- *Managing Peripherals*
- *Storing and Restoring Files*
- *Backup and Recovery*
- *Account Structure and Security*
- *System Configuration*
- *Volume Management*
- *Localizing and Customizing System Information*

Related IBM Publications

- *Systems Network Architecture Reference Summary*
- *IBM Synchronous Data Link Control*
- *Marketing Publications KWIC Index*
- *Field Engineering Handbook, SNA*
- *System Network Architecture - Sessions between Logical Units*
- *IBM 3704*, *3705-I*, and *3705-II Communication Controllers Principles of Operation*

ACF/NCP/VS Publications:

- *Introduction to the IBM 3704 and 3705 Communications Controllers*
- *Introduction to the IBM 3725 Communications Controller*
- *Guide to Using the IBM 3705 Communications Controller Panel*
- *Network Control Program Logic Manual*
- *ACF/TAP User's Guide*
- *Advanced Communication Function for Network Control Program, for IBM 3705 Advanced Communication Function for System Support Programs, for IBM 3705 General Information Manua*l
- *ACF/NCP/VS Network Control Program/System Support Programs Installation Manua*l
- *ACF / Network Control Program / VS Handbook*
- *Network Control Program/VS Generation and Utilities Guide* and *Reference Manual (for OS/VS and DOS/VS VTAM Users),*
  IBM 3704 and *3705 Program Reference Handbook*

ACF/VTAM Publications:

- *ACF/VTAM Operator's Guide*
- *ACF/VTAM Messages and Codes*
- *ACF/VTAM Library Summary*
- *ACF/VTAM General Information: Introduction*
- *ACF/VTAM General Information: Concepts*
- *ACF/VTAM Planning and Installation Reference*
- *ACF/VTAM Reference Summary*

8100 DPPX Publications:

- *8100 DPPX Host Connect Cookbook for DSC, DSX, HCF*

JES2 Publications (version 1):

- *Operator's Library: JES2 Commands*
- *Systems Programming Library: JES2 Initialization and Tuning*
- *Operator's Library: OS/VS2 Remote Terminals (JES2)*
- *Operator's Library: JES2 Command Language Reference Summary*

JES2 Publications (version 2):

- *JES2 Operations: XA JES2 Commands*
- *JES2 System Programming Library: XA Initialization and Tuning*
- *JES2 Operations: XA JES2 Command Syntax*
- *MVS/XA JES2 Licensed Program Specifications*
- *Message Library: XA JES2 Messages*

JES2 Publications (version 3):

- *MVS/ESA Program Directory (JES2)*
- *MVS/SP Version 3—JES2 LPS*
- *MVS/ESA Operations: JES2 Commands*
- *MVS/ESA Operations: JES2 Command Syntax*
- *MVS/ESA System Programming Library: JES2 Initialization and Tuning*
- *MVS/ESA Message Library: JES2 Messages*
- *MVS/ESA JES2 Directory of Programming Interfaces for Customers*
- *MVS/ESA Remote Workstation Generation*
- *MVS/ESA JES2 Logic*
- *MVS/ESA System Programming Library: JES2 Customization*

JES3 Publications (version 1):

- *JES3 Introduction*
- *JES3 Operator's Library Reference Summary*
- *JES3 Programming Library: Initialization and Tuning*
- *JES3 System Programming Library: Diagnosis*
- *JES3 Messages*

JES3 Publications (version 2):

- *JES3 Command Syntax*
- *JES3 Introduction*

- *JES3 Initialization and Tuning*
- *JES3 Messages*
- *JES3 Commands*
- *MVS/SP JES3 Licensed Program Specifications*
- *JES3 Conversion Notebook, Version 2 Release 2.1*
- *JES3 Conversion Notebook, Version 2 Release 1.5*

VSE/POWER Publications (version 2):

- *Introduction to the VSE System*
- *VSE/POWER Remote Job Entry User’s Guide*
- *VSE/POWER Shared Spooling User's Guide*
- *VSE/POWER Messages*
- *VSE/POWER Installation and Operations Guide*
- *VSE/POWER Networking User’s Guide*
- *VSE/POWER Reference Summary: Local Processing*
- *VSE/POWER Reference Summary: Remote Processing*

VSE/POWER Publications (version 4):

- *VSE/SP General Information*
- *VES/SP Messages and Codes*
- *VSE/POWER Licensed Program Specifications*
- *VSE/POWER Installation and Operations Guide*
- *VSE/POWER Remote Job Entry User’s Guide*
- *VSE/POWER Networking User's Guide*
- *VSE/POWER Application Programming*

1 Introducing SNA NRJE

This chapter introduces you to SNA NRJE and describes the following:

- SNA NRJE workstation emulation
- SNA NRJE product features
- SNA NRJE hardware and software requirements

Introduction

Network Remote Job Entry (NRJE) is a data communications subsystem that provides HP-to-IBM batch communications in a Systems Network Architecture (SNA) environment. SNA is a comprehensive specification for distributed data processing networks, developed by IBM.

SNA NRJE users at remote locations submit jobs to a host system for processing. The host sends output to SNA NRJE users at the remote location. SNA NRJE emulates a remote batch workstation as described below.

SNA NRJE's relationship to the International Standards Organization (ISO) and SNA model communications networks is described in Table 1-1, “Relationship Between Network Models.”

A computer network is designed in structured layers. The reason for a layered approach is to simplify the design of the overall network. Each layer rests on the next lower layer as a foundation. Each layer provides a service to the next higher layer in its own node, and relates to its counterpart layer on another node at a peer level. Direct communication with another node takes place only at the lowest layer of a network.

Table 1-1 **Relationship Between Network Models**

Layer	ISO	SNA	HP Implementation
7	Application	End User*	NRJE User Interface and Intrinsics
6	Presentation	Network Addressable Unit (NAU) Services	NRJELU Program
5	Session	Data Flow Control	
4	Transport	Transmission Control	SNA Transport Intrinsics
3	Network	Path Control	
2	Data Link	Data Link Control	SLDC dRIVER
1	Physical	Physical	INP (for NRJE/V) PSI (for NRJE/XL)
*IBM describes SNA with only six layers.			

A Workstation

A batch workstation is known to the host as a combination of one or more of the following peripheral devices, or their equivalents:

- Card Readers
- Line Printers
- Card Punches
- Operator Console

It is identified by `RMT`*nnnn* which represents the remote workstation number.

Jobs are submitted from a remote workstation through card readers, or their equivalent, over a communication line to a host system.

Output is received from a host system over a communication line to card punches and line printers, or their equivalent, at a remote workstation.

An operator console at a workstation allows interaction with a host system by transmitting job entry subsystem commands and by receiving job entry subsystem messages. An operator console at a workstation cannot be used for an interactive session with a host system. Job entry subsystem commands can display information and can control the following:

- Remote workstation sessions at a host system
- Jobs operating at a host system
- Input and output queues

Primitive Workstations

A primitive workstation requires an operator. All jobs must be submitted by an operator, all output is received by an operator, and an operator executes all console commands and receives all console messages.

Advanced Workstations

Advanced workstations are designed to allow users other than the operator to perform the following tasks:

- Submit a job to a transmission queue
- Display information about jobs in a transmission queue
- Modify a transmission queue
- Receive job output data into output queues
- Modify output queues

- Use the workstation console

The user of an advanced workstation can work independently of the availability of the host system. Most actions relate to the domain of the workstation and do not enter into the host domain. An advanced workstation performs many functions without communicating with its host.

SNA NRJE Emulation of a Workstation

NRJE on an HP 3000 emulates an **advanced batch workstation**. The SNA NRJE characteristics resemble an IBM 8100 **Distributed Processing Program Executive (DPPX)/RJE** workstation. An 8100 DPPX/RJE workstation is suitable for many distributed processing applications in many industries. It communicates with IBM System/370, 43xx, and 30xx, or plug-compatible processors using a 37x5 type communications controller.

8100 DPPX/RJE Overview

When DPPX is used with an SDLC communications link, the 8100 is operating in an SNA environment. In this case both a workstation console facility and multiple LU-LU sessions between an 8100 and a host system are supported.

A DPPX/RJE operator controls each workstation. The operator can issue interactive DPPX commands to establish a communications link and to manage input/output sessions or devices. Also, the operator can enter host system commands and receive host system messages at the workstation console. The operating system also supports a Programmed Operator Facility, which allows user-written programs to carry out operator functions.

DPPX allows a user to input jobs from a card reader, disk files, a diskette, or magnetic tape. A user issues a DPPX/RJE command to submit the name of the file to a transmission queue. The queue itself is a disk file containing the names of files with jobs. When appropriate operator commands are issued, DPPX/RJE transmits the files named in the transmission queue to the host system. A user can submit file names to a transmission queue even when an 8100 is not attached to a host. Job file names can remain in the queue until an operator sends the files to the host.

When output returns to an 8100 over an SDLC link, DPPX/RJE handles the data set in one of two ways. The operator routes the data set either (1) directly to a printer or punch, or (2) to a disk for deferred printer or punch output.

SNA NRJE Emulation Overview

SNA NRJE workstations transmit jobs to the host via **logical readers**.You configure a **virtual reader**, which is an MPE spooled output device that holds submitted jobs until they are transmitted to the host. Workstations receive output data sets from the host via **logical writers**.

NRJE assembles job input into a spool file. When communications with the host exist, job input spool files are transmitted to a host system as logical card reader data streams. Job output data is received from the host system in logical printer and punch data streams at the workstation and is then routed to the appropriate destination. The remote host console sends commands from the workstation to the host and receives console messages at the workstation from the host.

An NRJE workstation communicates with a job entry subsystem at a host. The job entry subsystem is under the control of the host operating system.

Several NRJE workstations can operate on an HP 3000. You must always identify which NRJE workstation you want to access.

One spooled reader, a **virtual reader**, must be configured for each NRJE workstation. The spooled reader is used for queued transmission from an HP 3000 to the host system. A virtual reader on an HP 3000 has no corresponding definition at the host.

NRJE users can submit jobs whether or not their HP 3000 is communicating with the host system. An NRJE user submits a job by designating a set of MPE files. A spool file is created and receives the MPE job input files in the order they are designated. Each spool file is an element in the workstation transmission queue. When communication with the host starts and the host system is accepting job input, each job is transmitted according to its position in the queue. If communication with the host system is not active, or if the host system is not accepting job input, the job remains in the transmission queue. Direct transmission of a job to the host--transmitting a job without first entering job input data into the transmission queue--is not supported.

A workstation and host communicate with each other in an **LU-LU session**. In SNA, a workstation is a set of logical units (LUs). The job entry subsystem at the host is also an LU. In an LU-LU session between a workstation and a host, NRJE retrieves job input data, translates and compresses it when requested, prefixes SNA format headers, and passes it to the SNA link product on the HP 3000. The SNA link product transmits the data to the host. A spool file containing job input data is retained until it is successfully transmitted.

NOTE In this manual, the term **SNA link produc**t is used when the information being given is true for all of the SNA link products: SNA Link/V, SNA/SDLC Link/XL, and SNA/X.25 Link/XL. The terms **SNA Link/V**, **SNA/SDLC Link/XL**, and **SNA/X.25 Link/XL** are used when a distinction among the three products is necessary.

Multiple data sets can be created for each job submitted to a host. An output data set is always created for each job submitted to a host for Job Control Language (JCL) logging. Data sets are received at an HP 3000 using an SNA link product. Data is decompressed and translated if desired. Data sets are returned to a logical output device, a logical writer, with those from a specific job typically making up one spool file. A **logical writer** must exist for each printer or punch defined at the host for your remote.

NRJE uses the console command and message stream between a workstation and the host job entry subsystem. An NRJE user sends commands and receives messages through the console.

All job entry subsystem commands configured for a workstation are permitted. (This, however, depends on the capabilities assigned the user by the node manager.)

NRJE can be accessed two ways: interactively and programmatically. Interactive access is through the NRJE subsystem command interpreter. Programmatic access is through a user-written program calling NRJE intrinsics. Most NRJE commands correspond with intrinsics.

Operating Environment

These are the operating environments needed for SNA NRJE:

- HP 3000 to SNA host: SNA NRJE emulates 8100/DPPX/RJE. An SNA host must be configured for an IBM 8100 DPPX/RJE workstation when communicating with an HP 3000 running SNA NRJE. See the *SNA Link Services Reference Manual* or the *SNA NRJE Node Manager's Guide* for information on configuring SNA NRJE on your HP 3000 system.
- SNA Host: An 8100/DPPX can communicate with IBM System/370, 43xx, or 30xx, or plug compatible processors using a 37x5 type communications controller.

NRJE Features

The basic features available with SNA NRJE are as follows:

- The HP 3000 can emulate the major functions of an IBM 8100 DPPX/RJE workstation.
- Multiple HP 3000 users can have simultaneous batch access to the host.
- An HP 3000 user can submit jobs and receive output by NRJE subsystem commands, and programmatically by NRJE intrinsics.
- A maximum data communications line speed of 64Kbps is supported.
- Concurrent communication with multiple hosts or multiple data communication lines to a single host is allowed.
- An HP30020A/B/C Intelligent Network Processor (INP) is used as a hardware interface, for NRJE/V; this is included with SNA Link/V. For NRJE/XL, an HP30263A Programmable Serial Interface (PSI) is used; this is included with SNA/SDLC Link/XL and SNA/X.25 Link/XL.
- Host output can be sent to any supported output device on an HP 3000.
- Job management capabilities enable the user, at the time the job is submitted, to control the final destination of the output coming from the host.
- Output management capabilities enable the user, through the use of an NRJE manager developed **Lookup Table**, to control the final destination of the output coming from the host.
- Reverse NRJE can be accomplished. Output data sets from the host can be streamed on the HP 3000 through the use of a special forms identifier specified in the job JCL.
- SNA NRJE can run concurrently on the HP 3000 with other applications.
- SNA NRJE can operate on an HP 3000 that is part of an NS network of HP 3000 and HP 1000 systems and is accessible from all networked systems.
- Jobs can be submitted into the transmission queue independent of the communication line status.
- Customizable error and help message facilities are available.
- Transmission of console commands and reception of console messages are supported.

- Data compression and character code translation are performed (or disabled if desired).
- Multiple console users can be allowed, or access can be restricted to those with node manager (NM) capability.
- User exits are available to enable manipulation of data.

How to Use NRJE

Examples of how to use NRJE are in Chapter 2 , “Getting Started.” For descriptions of subsystem commands, see Chapter 3 , “User Commands.”

You can write routines to perform the same functions as NRJE commands. These are described in Chapter 5 , “User Intrinsics,” and Chapter 6 , “NRJE Manager Intrinsics.”

Output from the host may be received routinely at any supported HP 3000 output device. This includes laser and line printers, disk files, and tape files. See Chapter 4 , “Job Output,” for procedures to control job output destination.

HP 3000 Hardware and Software Requirements

Hardware Requirements

The HP 3000 hardware requirements for SNA NRJE are as follows:

- An HP 3000 Series 37, 39, 4x, 5x, 6x, or 7x that supports an HP 30020A/B/C Intelligent Network Processor (for NRJE/V).
- A HP 3000 900 Series that supports an HP30263A Programmable Serial Interface (for NRJE/XL).
- Communications line, leased or switched.
- These modems are supported:

HP 37230A	up to 9600 bps
Bell 201C	2400 bps
Bell 208A/B	4800 bps
Bell 209A	9600 bps
Bell 2024A	2400 bps

  See the *Data Communications Handbook, Fundamental Chapter* for communications configurations and strapping options.
- A block mode terminal in addition to the system console for Node Management Services configuration. This terminal must be supported by VPLUS. See the *SNA Link Services Reference Manual* or the *SNA Link/XL Node Manager's Guide*.

Software Requirements

HP 3000 software requirements to run SNA NRJE are as follows:

- MPE V Operating System: HP 32033G (for NRJE/V).
- MPE XL Operating System (for NRJE/XL).
- SNA Link/V: HP 30246A (for NRJE/V). This enables SNA NRJE to communicate with the host job entry subsystems in the SNA environment.

  or
- SNA/SDLC Link/XL: HP 30291 (for NRJE/XL).

  or
- SNA/X.25 Link/XL: HP 30298 (for NRJE/XL).

Host Hardware and Software Requirements

Hardware Requirements

Host hardware requirements for SNA NRJE are as follows:

- An IBM System/370-compatible mainframe, such as the Model 370, 43xx, or 303x.
- An IBM 37xx-compatible communications controller that supports an SNA line. An HP 3000 can also be connected to the communications controller via an IBM 3710 network controller.

Software Requirements

Host software requirements for SNA NRJE are as follows:

- MVS/SP or MVS/XA operating system with JES2 or JES3 job entry subsystem

  or

- VSE operating system with VSE/POWER

  and both

- ACF/VTAM telecommunications access method
- ACF/NCP network control program.

CAUTION

Hewlett-Packard has not verified the NRJE product with all possible combinations of host system software releases. We are continually certifying with new host system software releases. Check with your Hewlett-Packard representative for the most up-to-date list.

Hewlett-Packard does not require that the customer run on one of the verified versions. However, not doing so applies limits to Hewlett-Packard's liability in addition to the normal limits. Because of the certification problems imposed by the host system versions, Hewlett-Packard must restrict its support policy in the following ways. The resolution of all problems will fall into one of the categories listed below:

If a problem is caused by incorrect operation of the Hewlett-Packard product, Hewlett-Packard will repair the product.

If a problem is caused by incorrect operation at the host,

(a) Hewlett-Packard may require that the user change to a known working version of the software;

(b) Hewlett-Packard may elect to change the Hewlett-Packard product to conform to the situation.

As always, Hewlett-Packard will in good faith attempt to solve the problem with the customer.

2 Getting Started

This chapter describes how to perform the following tasks:

- Plan jobs to send to a host system.
- Use NRJE to send jobs to a host system.
- Start and stop NRJE.

Planning Jobs

When preparing jobs for a host, an NRJE user must consider host system characteristics, such as the type of job entry subsystem and programming languages supported, as well as the HP 3000 resources configured for SNA NRJE.

Job Input

Jobs sent to a host system using a physical card reader normally consist of Job Control Language (JCL) and data. These cards are concatenated to form a job stream.

Jobs sent to a host system using NRJE must be constructed as if they would be sent directly to the host using a card reader. However, input through NRJE is in **card images** instead of physical cards. Card record image sizes must be a minimum of 80 bytes for all supported host job entry subsystems, and can be a maximum size of the following:

- 248 bytes for JES2 or JES3 workstations that use a 256-byte RU.
- 252 bytes for JES2 or JES3 workstations that use a 512-byte RU.
- 252 bytes for JES2 or JES3 workstations that use a 768-byte RU.
- 128 bytes for VSE/POWER workstations.

You combine records into files. Then you identify to NRJE the files to be sent to the host. You can use the first and last files of the set for routine JCL statements.

When NRJE processes a `SUBMIT` command, it opens each card reader image file in sequence. Each file is opened, its records are read and then written to a spool file that is submitted to the host. Then the file is closed.

SNA NRJE accepts input from two types of **card image** files:

- Infiles
- FD files

Infiles

An *`Infile`* contains a complete job or a partial job. The file can contain ##FD cards that point to additional files to be included in your spool file. In fact, an *`Infile`* can consist of ##FD cards only. Also, an *`Infile`* can be transmitted without translation. Chapter 3 , “User Commands,” describes the `SUBMIT` command, which is used to identify *`Infiles`* to be transmitted.

FD Files

FD files are similar to *Infiles* in that they may contain the same kind of information, they are constructed in the same way, and they can be transmitted without translation. They are referenced directly by means of ##FD cards in the *Infile*. (SeeFigure 2-1).

FD files may reference other FD files. This nesting can be continued up to 20 levels deep. Figure 2-1 illustrates the relationships among several **card image** files that have been merged into a single job stream. The names of the *Infiles* were indicated by the user when submitting the job. First-level FD files, RED and BLUE, were specified indirectly by means of records—card images—contained in *Infile* A. Two records within RED point to the two level-two files.

When SNA NRJE processes a SUBMIT command, it opens each *Infile* named directly in the command, and each FD file as it is referenced. The files are opened with the MPE share and read-only file restrictions. Each file remains open until all of its records have been processed, at which time it is immediately closed.

NOTE

If an *Infile* or FD file is kept as a *numbered* file and the record width is less than 80 bytes, NRJE will transmit the line numbers as part of the data record.

Figure 2-1 Relationship Among Files in a Job Stream

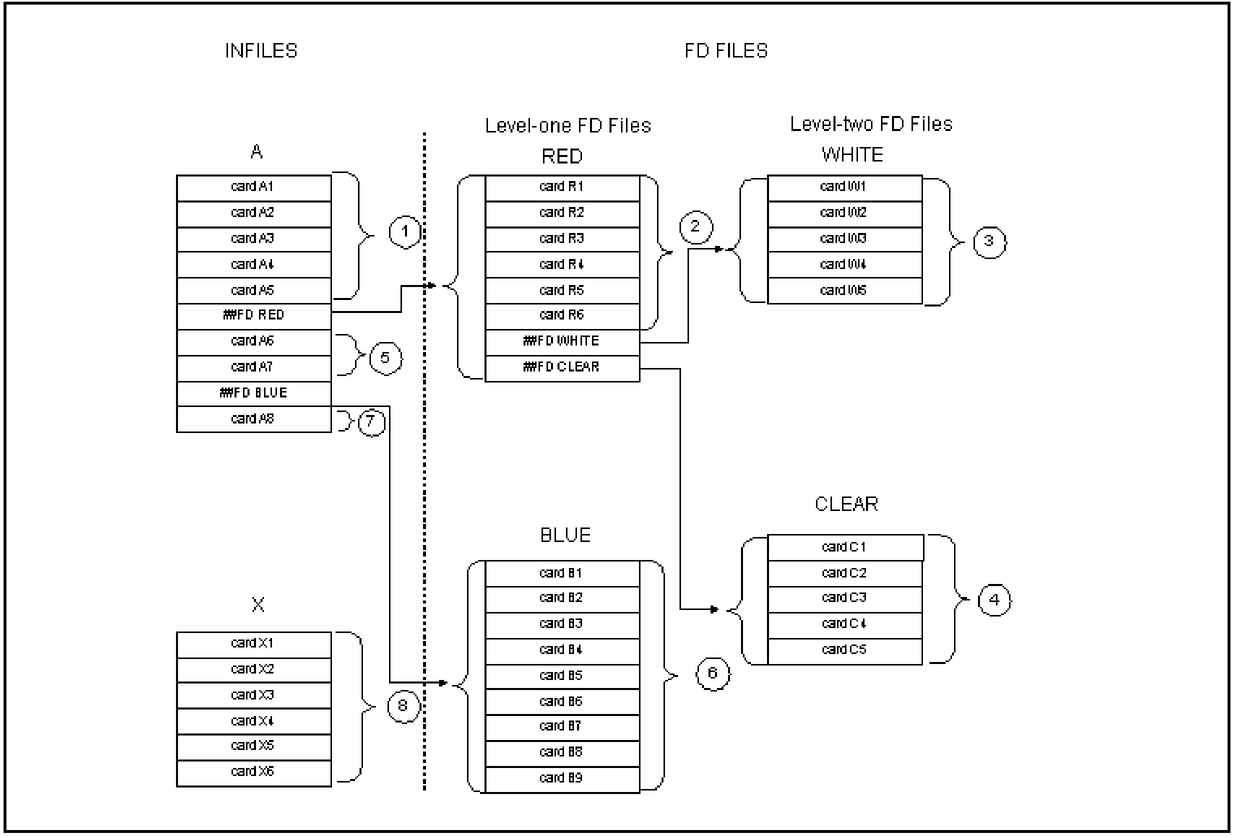
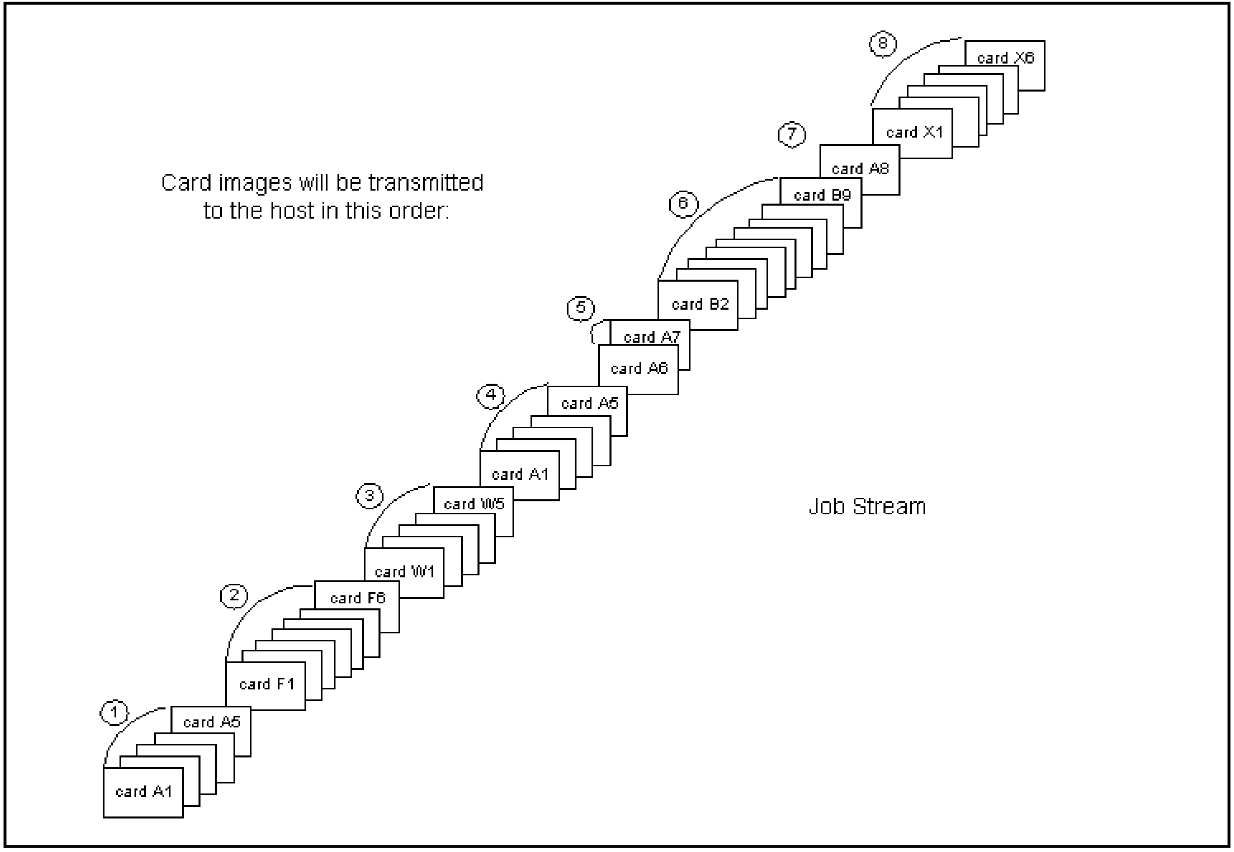


Figure 2-1 shows the relationship among `infiles` and FD files in a job stream. Figure 2-2 shows the order in which the card images from the files in Figure 2-1 are transmitted to the host.

Figure 2-2 Order of Card Images Transmitted



Host System Job Control

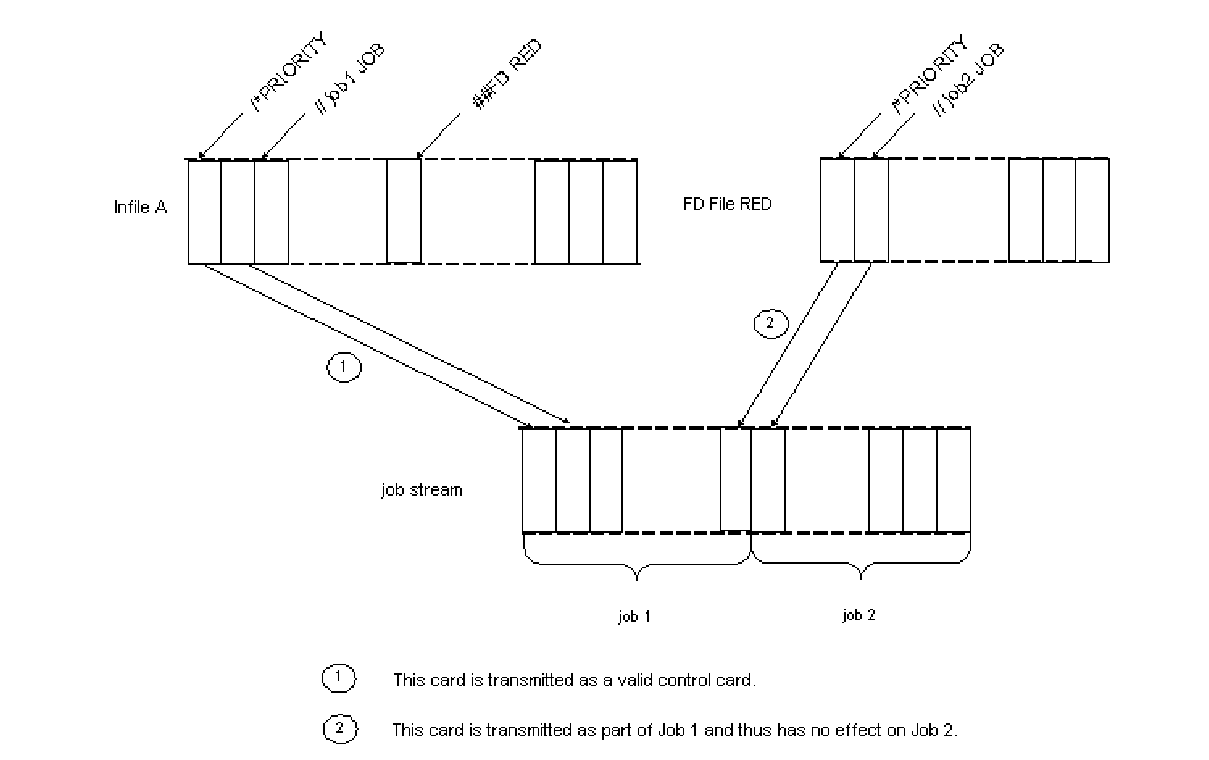
SNA NRJE checks for valid JOB cards, SIGNOFF cards, and host console commands before transmission. If NRJE detects a SIGNOFF card, it prints a warning message. The card is not transmitted to the host.

SNA NRJE requires at least one JOB card whenever a `SUBMIT` command is issued. If the *TRANSPARENT* parameter is omitted from an FD card image, SNA NRJE recognizes all JOB cards contained in the referenced file as job delimiters, not data.

Valid JOB cards are used to break an input job stream into separate spool files, which are placed in the reader transmission queue. If any host command card images are between host jobs in an input job stream, they are sent with the first of the jobs. For example, host system PRIORITY cards have no effect on the host when associated with the second or subsequent job in a job stream. Figure 2-3 shows two jobs that are submitted to SNA NRJE together. SNA NRJE merges `Infile` A with FD file RED. The first priority card has the desired effect on JOB1. The second priority card is transmitted to the host as part of JOB1 and therefore does not influence JOB2.

When priorities are important, each job should be submitted to SNA NRJE separately.

Figure 2-3 Priority Card to Job Stream Relationship


/*PRIORITY
// job1 JOB
##FD RED
/*PRIORITY
//job2 JOB
Infile A
FD File RED
2
1
job stream
job 1
job 2
1 This card is transmitted as a valid control card.
2 This card is transmitted as part of Job 1 and thus has no effect on Job 2.

Input File Libraries

The job streaming feature of SNA NRJE allows you to construct jobs from job control cards, programs, and data that already exist in disk files. You can create libraries of job modules that can be assembled into a variety of different jobs.

The SNA NRJE user must ensure that `Infiles` and FD files submitted together constitute a valid job according to host system requirements.

Job Output

Output from a job can be received from a host whenever communication has been established with a workstation. The default output destination for the workstation must be a spooled device as configured in the NMMGR Workstation Data screen.

You can control job output. Chapter 4 , “Job Output,” describes what to do within a job to route the returning data sets to specific destinations.

Sending Jobs to a Host

To operate NRJE you must do the following:

- Log on to the HP 3000.
- Optionally define any formal designators you will use.
- Invoke NRJE.
- Issue a `SUBMIT` command.

Defining Formal Designators

Define formal file designators with a `FILE` command. You can do this before you invoke NRJE, or within NRJE. When a formal designator is used, it must be preceded by an asterisk (*). The asterisk indicates a reference to a previous `FILE` command. See the *MPE Commands Reference Manual* or the *MPE Software Pocket Guide*.

CAUTION NRJE uses the files INPUT and OUTPUT for processing internal messages. Therefore, these files are reserved for use by NRJE and should not be specified by the user. Use of these files could result in corrupted user data.

NRJE Command

Invoke NRJE from MPE by issuing the command

```
:NRJE Wsid
```

where *`Wsid`* is a workstation identifier. See `NRJE` in Chapter 3 , “User Commands.”

When you use `NRJE` as an MPE command you can omit the workstation identifier; however, you must use a *`Wsid`* parameter in every command that refers to a workstation.

Kinds of NRJE Users

NRJE users can perform the following tasks:

- Transmit and receive batch jobs between an HP 3000 serving as a workstation and a host system. No special HP 3000 user capabilities are required.
- Control an HP 3000 workstation. NM user capability is required. Users configured with NM capability can be NRJE managers.

An NRJE manager can do everything a user can do. Some user commands have NRJE manager extensions.

Sample NRJE Session on an HP 3000

See Chapter 3 , “User Commands,” for a description of each of the commands in the following sample session. In this sample, user input is shown in the first column, and instructions and explanations are in the second.

Log onto the HP 3000

```
:HELLO USER.ACCT,GROUP
WELCOME TO THE HP 3000
```

Invoke the NRJE subsystem by entering the NRJE command and specifying RMT11 as the default workstation identifier.

```
:NRJE RMT11
```

A banner line identifies NRJE and its version.

product number *version* number NJRE (C) HEWLETT-PACKARD CO. *copyright* year

An NRJE welcome message follows the banner line. The text of this message is configured by an NRJE manager.

```
WELCOME TO NRJE
```

A configured prompt string, RMT11>, is sent by NRJE. This identifies your default workstation.

```
RMT11>...
```

Now you can issue an NRJE subsystem command.

A two-line SUBMIT command is entered next. An ampersand (&) continues the command to a second line. File identifiers follow the command. All files are from your logon group and account.

```
RMT11>SUBMIT FRONTJCL, DATA1, DATA2,&
RMT11>BACKJCL
```

A message is displayed indicating that your job stream has been constructed and is ready to be sent to the host system.

Note the number in the message (1042) for future reference, or use the job naming capabilities of the SUBMIT command.

```
JOB SUCCESSFULLY SUBMITTED. THE SPOOL FILE IS:
#O1042
```

Now you can issue other NRJE commands.

```
RMT11>...
```

Issue an EXIT command which returns control to MPE.

```
RMT11>EXIT
```

Your MPE prompt is returned and you can issue MPE commands.

```
:...
```

You may want to find out about the jobs you have submitted, cancel your jobs, or change their transmission priority. The commands available for these tasks are described in Chapter 3 , “User Commands,” and summarizes all of the routine commands. (Manager commands required for day-to-day workstation operations are described in the *SNA NRJE Node Manager's Guide*.)

NRJE Startup

Later in this chapter, an example illustrates how to start and stop NRJE. To start NRJE, you must do the following:

- Activate the data communications line and host PU and LUs.
- Start the SNA node using an MPE `SNACONTROL START` command.
- Start the NRJE workstation using an MPE `NRJECONTROL START` command or an NRJE `STARTWS` command.

NOTE A job stream file must exist before you start a workstation.

An MPE job stream file is required to start each workstation. The job stream starts NRJEMON, which is the father communication process for an NRJE workstation. The structure of the job stream file and the `STARTWS` command are described in the *SNA NRJE Node Manager's Guide*. `SNACONTROL START` is described in the *SNA Link Services Reference Manual* or the *SNA Link/XL Node Manager's Guide*. The MPE command `NRJECONTROL START` is also described in the *SNA NRJE Node Manager's Guide*.

NRJE Shutdown

Two ways to shut down NRJE using MPE commands exist:

- Stop NRJE using an `NRJECONTROL STOP` or `NRJECONTROL HALT` command from MPE.
- Issue an `SNACONTROL STOP` command from MPE. This shuts down the SNA link and NRJE.

A `STOPWS` issued from within NRJE has the same effect as an `NRJECONTROL STOP`. A `HALT` issued from within NRJE has the same effect as an `NRJECONTROL HALT`. `NRJECONTROL HALT` and `NRJECONTROL STOP` as well as the and `HALT` commands are described in the *SNA NRJE Node Manager's Guide*. `SNACONTROL STOP` is described in the *SNA Link Services Reference Manual* or the *SNA Link/XL Node Manager's Guide*.

Upon terminating communication with the host, NRJE will set NRJEJCW in the job stream to indicate the reason for the termination. This JCW can be used to determine whether to attempt to restart NRJE in the job stream.

Possible values for NRJEJCW:	**Reason**
2	NRJE `STOPWS` workstation command
4	NRJE `HALT` workstation command
17	The LU is not active on the host.
1224	An unbind was received.
3000	`SNACONTROL STOP;type=Q` command
3001	`SNACONTROL STOP;type=P` command
3003	Link failure was detected.
3005	Hierarchical shutdown received.

These JCW values may vary depending on the version of the SNA link product installed on the system.

In general, all other values set upon termination are a result of a potential error condition encountered.

Use of NRJEJCW

The following example can be used to restart NRJE after the host system disconnects:

```
!JOB NRJE,MANAGER>>SYS/password,NRJE;OUTCLASS=,1
!RUN NRJEMON;INFO="wsid"
!IF NRJEJCW = 1224 OR NRJEJCW = 3005 THEN
!  CONTINUE     << Received unbind or hierarchical shutdown >>
!   SNACONTROL START;NODE=nodename
!   NRJECONTROL START;WSID=wsid
!ENDIF
!EOJ
```

The example below includes SNACONTROL commands, which are the responsibility of the SNA node manager and are described in the *SNA Link Services Reference Manual* or the *SNA Link/XL Node Manager's Guide*.

In this example, the input entered by the NRJE manager is shown in the first column, and instructions and explanations are in the other columns.

Example

```
                    Issue a status command before the SNA node
                    is started.
:snacontrol status;node=ibmnode
NODE SPECIFIED NOT ACTIVE.  (SNACIERR 9)

                    Start the SNA node.

:snacontrol start;node=ibmnode
SNA TRANSPORT VERSION: A.XX.XX

                    Enter the NRJE subsystem.

:nrje r20
product number version number NRJE (C) HEWLETT-PACKARD CO copyright year

WELCOME TO NRJE

                    Issue DISPLAY INFO command.
R20>display;info

Workstation: R20
GENERAL INFORMATION
               State:      INACTIVE       The workstation is INACTIVE
               Jes:        JES2           even though the SNA node has
               Remote:     RMT20          been started.

Host command prefix char:   $
Allowed console commands:
     Formid lookup table:   NRJETABL.NRJE
 Default print form name:   STD
 Default punch form name:   STD

WELCOME MESSAGE:
WELCOME TO NRJE
                    Issue STARTWS command. A job number is returned
                    identifying the file that has been streamed.

R20>startws

 #J745
                   Immediately issue another DISPLAY INFO command.
R20>display;info

Workstation: R20
GENERAL INFORMATION
               State:      Mon Pending     The state is now Monitor
               Jes:        JES2            Pending, a transient state.
               Remote:     RMT20           It will soon become ACTIVE.
Host command prefix char:  $
Allowed console commands:
     Formid lookup table:   NRJETABL.NRJE
 Default print form name:   STD
 Default punch form name:   STD

WELCOME MESSAGE:
WELCOME TO NRJE


                         Issue another DISPLAY INFO command.

R20>display;info

Workstation: R20
GENERAL INFORMATION
               State:      ACTIVE     The state is now ACTIVE.
               Jes:        JES2
               Remote:     RMT20
Host command prefix char:  $
Allowed console commands:
     Formid lookup table:   NRJETABL.NRJE
 Default print form name:   STD
 Default punch form name:   STD

WELCOME MESSAGE:
WELCOME TO NRJE

R20>display lutable

WORKSTATION LU LIST
LU Name   Chain Size  LU Address   Auto Start

LU01     *     99            1          YES
LU02     *     99            2          YES
```

```
LU03     *     99           3          YES
LU04     *     99           4          YES
LU05           99           0           NO
LU06           99           0           NO
LU07           99           0           NO
LU08           99           0           NO

          *  = The LU is active.
```

Now jobs that have been submitted are transmitted. Data sets can be received. Console commands can be sent. Console messages can be received.

Identify the status of all sessions, PU-SSCP, LU-SSCP, LU-LU.<F100P12D>

```
R20>:snacontrol status;node=ibmnode
```

SNACONTROL commands can be entered from your configured NRJE prompt by preceding them with colon (:).

For MPE V systems, the following displays:

```
   PU     - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   1 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   1 - LU  1   STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   2 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   2 - LU  1   STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   3 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   3 - LU  1   STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   4 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   4 - LU  1   STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   5 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   6 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   7 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
   LU   8 - SSCP    STATE: ACTIVE         TRACE OFF   USER: MANAGER.SYS
```

For MPE XL systems, the following displays:

```
   SESSION ID        STATE       TRACE        USER
   ----------------------------------------------------
   PU    - SSCP    SESS_ACTIVE     OFF      MANAGER.SYS
   LU 1 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 2 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 3 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 4 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 5 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 6 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 7 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
   LU 8 - SSCP     SESS_ACTIVE     OFF      MANAGER.SYS
```

```
                       Stop the SNA node, and so all workstations,
                       in a quiescent manner.

R20>:snacontrol stop;node=ibmnode;type=q

                       Issue an SNACONTROL STATUS command after
                       the node is closed<F100P12D>.

R20>:snacontrol status;node=ibmnode
NODE SPECIFIED IS NOT ACTIVE.  (SNAERR 11)

                       Issue another DISPLAY INFO command.

R20>display;info

Workstation: R20
GENERAL INFORMATION
               State: INACTIVE Workstation is INACTIVE.
               Jes:        JES2
               Remote      RMT20
Host command prefix char:  $
Allowed console commands:
     Formid lookup table:   NRJETABL.NRJE
 Default print form name:   STD
 Default punch form name:   STD

WELCOME MESSAGE:
WELCOME TO NRJE

R20>exit
```

3 User Commands

SNA NRJE is used to transmit and receive batch jobs between a workstation and an IBM or IBM plug-compatible host system, and to control the workstation. This chapter describes how to use NRJE to submit jobs and receive data between a host system and an HP 3000 serving as a workstation. It also describes how NRJE is used to perform the following tasks:

- Display information about your jobs.
- Display information about the workstation.
- Alter the status of your jobs.
- Cancel your jobs.

Introduction

Some user commands have NRJE manager extensions. You must be configured with NM capability to use them.

Job Output

Output from a job is automatically received from a host whenever communication is established between a workstation and a host. You do not need to issue an NRJE subsystem command or related intrinsic to receive your job output.

Job output can be directed to any output device that is supported by your HP 3000. These include laser printers, line printers, and disk and tape files. These devices can be spooled or non-spooled, but the default output device for your workstation, as configured in the NMMGR Workstation Data Page 2 screen, must be spooled. Control of the final destination can be determined either at the time your job is submitted for transmission to the host, or after the output has been returned from the host. Chapter 4 , “Job Output,” describes job output, including the following:

- How to route job output.
- The relationship between SNA control codes and HP 3000 carriage controls.
- The JCL to use for special forms processing.
- Directing output to an HP 268XX laser printer.
- How to relate host output classes and message classes (*msgclass*) with logical writers by using JCL and host job entry subsystem commands.

Workstation Identification

Whenever you enter NRJE, you can identify a workstation to serve as the default identifier for all subsequent subsystem commands. The following is an example:

```
:NRJE RMT11

                 This MPE command invokes NRJE for workstation RMT11.

product number version number NJRE (C) HEWLETT-PACKARD CO. copyright year

NRJE WELCOME MESSAGE

                 The NRJE welcome message is configured by the
                 SNA node manager or provided by an NRJE manager.

RMT11>SUBMIT infile

                 The NRJE subsystem SUBMIT command prepares
                 infile for the default workstation, RMT11.

JOB SUCCESSFULLY SUBMITTED. THE SPOOL FILE IS:
#O ...

                 NRJE sends you a message once the SUBMIT
                 command has completed.

RMT11> ...

                 Other NRJE commands follow.

RMT11>SUBMIT infile;WSID=RMT99

                 This NRJE subsystem SUBMIT command prepares
                 infile for workstation RMT99 even though the
                 default workstation is RMT11.

JOB SUCCESSFULLY SUBMITTED.  THE SPOOL FILE IS:
#O ...

                 NRJE sends you a message once the SUBMIT command
                 has completed.

RMT11> ...

                 Other NRJE commands follow.
```

Both “Command Structure” and command descriptions in this chapter describe how some NRJE subsystem commands can override the default workstation identifier.

Access Methods

You can access NRJE two ways: interactively and programmatically.

Interactive access is through the NRJE subsystem command interpreter. Use NRJE this way from your terminal or from a streamed job. Interactive commands have a simple format and provide the most common parameter values as defaults. As a convenience, abbreviations and a flexible range of parameter values are supported.

Programmatic access allows your program to call NRJE intrinsics. Most NRJE commands have a corresponding intrinsic so that, for example, the submission of a large number of jobs can be done programmatically. This access method can also be used to build an alternate subsystem command interface for a workstation, or a background operator interface to a workstation.

MPE Commands

When running NRJE interactively you can execute many MPE commands in the middle of NRJE operations. Enter an MPE command preceded by a colon (:), and the command will be passed to the MPE command interpreter. Many MPE commands can be entered without preceding them with a colon, if they are not members of the NRJE command set. The following is an example of how to enter an MPE command:

```
RMT11>...             Some NRJE command.

RMT11>:LISTF         List the file names in your logon group and account.

RMT11>...            Another NRJE command.
```

You could have entered LISTF without preceding it with a colon, because it is not an NRJE subsystem command.

Subsystem Command Limits

Each subsystem command is limited to 256 characters. If a command requires more than one line, the command can be continued by ending the line with an ampersand (&).

Command Structure

The structure of most NRJE commands is as follows:

Syntax

```
COMMAND [parameters][;W[SID]=Wsid]
```

Parameters

parameters	Each command may have its own parameters.
Wsid	Some commands also have an optional parameter to override the default workstation identifier.

Description

Each NRJE subsystem command starts with a command name, which may be abbreviated and which may be followed by parameters. Some commands have positional parameters and some may also have keyword parameters. An optional parameter common to many commands is `WSID=`*`Wsid`*, which is used to override a default workstation identifier.

NOTE Some user commands have extended capabilities that allow NRJE managers to affect files owned by other NRJE users. An NRJE manager is an HP 3000 user with NM capability.

Example

Each command description can have an illustrative example.

Text Reference

Some user commands are related to intrinsics. A reference to them is made here.

User Commands

This subsection describes each NRJE subsystem command available to a user. Some user command parameters are available only to NRJE managers. Those are described in this section, also. NRJE manager commands are described in the *SNA NRJE Node Manager's Guide*. Users who do not have NM capability can access the remote console facility, provided one or more commands have been “allowed” by their node manager. Once in the console facility, all users remain there indefinitely until either the workstation terminates or the user issues the RELEASE command to exit the console. If the workstation terminates, the RELEASE command is executed automatically for the user.

The user commands discussed in this section are shown in Table 3-1.

Table 3-1 NRJE User Commands

Command	Description
ALTER *	Changes the priority of a submitted job or jobs.
CANCEL *	Purges jobs awaiting transmission. Deletes returning output data sets for jobs already transmitted to the host.
	If you are connected to a VSE/POWER host system, only jobs awaiting transmission are cancelled.
CONSOLE	Obtains access to the host remote console of a workstation.
DISPLAY	Provides display of workstation and workstation-related information.
EXIT	Causes the user to exit from the NRJE command interpreter (CI) and returns control to the process from which the CI was run.
##FD	Used within a job to include the contents of a file.
HELP	Provides you with information about subsystem commands.
NRJE	Invokes the subsystem command interpreter. It is also an NRJE subsystem command and allows you to switch to a different default workstation.
REDO	Allows you to modify and/or execute the previous NRJE subsystem command. Use this like the MPE REDO command.
RELEASE	Relinquishes access to the host remote console of a workstation.
REMARK	Provides a way to do in-line comments when you use NRJE subsystem commands in an HP 3000 job.
RUN	Executes a prepared program.

Table 3-1 NRJE User Commands

Command	Description
`SHOW *`	Provides information about jobs you have submitted.
	If you are connected to a VSE/POWER host system, only those jobs awaiting transmission are shown.
`SUBMIT *`	Prepares a batch input job stream for transmission to a host system.
`VERSION`	Identifies the software version of SNA NRJE.
`[]`	A host job entry system console command preceded by the host command prefix character configured for the workstation.
*This command has NRJE Manager extensions.	

ALTER

Changes the transmission queue priority of a submitted job or group of jobs.

Syntax

```
          { SpoolfileID       }
A[LTER]   {    Name           },Newpri [;W[SID]=Wsid]
          {UserName.AcctName  }
          {     @             }
```

Parameters

SpoolfileID

Use to alter the priority of a specific job to *Newpri*.

The *SpoolfileID* parameter is assigned by the spooler after execution of a SUBMIT command. Its form is #O*nnnnn*, where *n* is a digit. You enter just the *nnnnn* portion of the ID.

Name

Use to alter all your jobs with this *Name* to the *Newpri* value

You can create a job *Name* in the SUBMIT command. You can submit several jobs with the same *Name*.

A job *Name* begins with a letter, and may be up to eight alphanumeric characters long.

UserName.AcctName

Use to alter all jobs from a *UserName.AcctName* to the *Newpri* value.

If your logon *UserName.AcctName* is different from the one you specified in this command, you must be configured with NM capability to use this parameter.

@

Use to alter the priority of all jobs you submitted to the *Newpri* value when your logon *UserName.AcctName* is the same as the jobs you want to change. If you have NM capability, this will affect all jobs.

Newpri

Specifies the new priority value you want for jobs. The value of *Newpri* must be an integer from 0 through 14, where 14 is the highest priority.

You can defer submitting your job by setting *Newpri* less than or equal to the value of RDRFENCE, which is described in the *SNA NRJE Node Manager's Guide*.

The initial priority of a job is established when you submit it via a parameter of the SUBMIT command.

Wsid

Overrides the default *Wsid* named in the NRJE command, for this command only. When you specify WSID=*Wsid* in this command, the job files you alter are in workstation *Wsid*.

Unless you are configured with NM capability, you must be the submitter of the jobs you want to alter.

Description

The ALTER command enables you to control the order of jobs within the transmission queue. Each job in the queue has a particular priority. You can determine the priority of a job or a group of jobs by using the SHOW command. Then, if you would like to change the priority of one or more jobs, you can use the ALTER command.

You can specify a particular transmission queue associated with a particular workstation identification by entering the WSID=*Wsid* parameter in this command. If you omit the *Wsid* parameter, this command refers to the transmission queue of the default workstation. You establish a default workstation by using the *Wsid* parameter with an NRJE command.

Restrictions

- You can alter the priority of only those jobs that are in a READY state. You can use the SHOW command to determine the state of jobs.
- If you are configured with an NM capability, you can alter the priority of jobs submitted by any *UserName.AcctName*. However, if you do not have NM capability, you can alter the priority of only those jobs submitted under your logon *UserName.AcctName*.

Example

```
RMT53>...

                      You have entered NRJE for RMT53.<F255P255D>

RMT53>DISPLAY;QUEUE

                      Use this command to obtain the value of Reader fence.

Workstation: R53

READER QUEUE
      Reader ldev:  12
       Chain size:  99
Number of readers:  1
      Compression:  YES
 Translation code:  0: Native-3000
    Reader status:  UP
     Reader fence:  6
     System fence:  6

RMT53>SHOW @

                      It is useful to see the priorities of your jobs.
                      If you are logged on as KEN.NRJE, you would see
                      this display.

   #O Name Pri State Rank Size User.Account   Time

 1069      13  READY  3    32   KEN.NRJE      10:48

RMT53>ALTER 1069,4

                      You changed the priority of SpoolfileID 1069 to 4.
                      You deferred transmission of this file because
                      Newpri is below the reader fence value.

RMT53>...

                      Other commands follow
```

In the example, you changed the priority of one file to a *Newpri* value. You can change the priority of groups of files to a *Newpri* value when you use the *Name, UserName.AcctName*, or *@* parameter. This is useful when you advance a priority value above the deferred value set by RDRFENCE.

Text Reference

The intrinsic related to this command is NRJEAlter, which is described in Chapter 5 , “User Intrinsics.”.

CANCEL

Purges jobs awaiting transmission. Deletes returning output data sets for jobs already transmitted to the host.

If you are connected to a VSE/POWER host system, only jobs awaiting transmission are cancelled.

Syntax

```
          {         SpoolfileID           }
          {            Name               }
CA[NCEL]  {      UserName.AcctName        } [;W[SID]=Wsid]
          {              @                }
          { ;HJ=HostJob [,HostJob [,...]]  }
```

Parameters

SpoolfileID

Cancels one of your jobs.

The *SpoolfileID* is assigned by the spooler after a job is submitted. Jobs can be cancelled by *SpoolfileID* only before they have been transmitted to the host.

Enter a value of the form *nnnnn*, where *n* is any digit.

Name

Job name assigned by the user in the SUBMIT command. You can submit several jobs with the same name.

Use of this parameter cancels all of your jobs with this *Name*.

A job name begins with a letter, and may be up to eight alphanumeric characters long.

UserName.AcctName

Cancels jobs for a specific *UserName.AcctName*. A period (.) must separate *UserName* from *AcctName*.

Jobs awaiting transmission to the host under this *UserName.AcctName* are purged. Data sets returning from the host for jobs submitted under this *UserName.AcctName* prior to this command are purged as they return.

If your logon *UserName.AcctName* is different from the jobs you want to cancel, you must be a user configured with NM capability to use this parameter.

@

Cancels all jobs from the current logon *UserName.AcctName*. Jobs awaiting transmission to the host under the current *UserName.AcctName* are purged. Data sets returning from the host for jobs submitted under this *UserName.AcctName* prior to this command are purged as they return.

HostJob

Either the job name from the JCL JOB card or the job number assigned by the host. All entries in the NRJE Job Log matching the job name and the logon *UserName.AcctName* are cancelled.

Multiple job numbers or names, to a maximum of five jobs, are allowed and must be delimited by commas.

Jobs can be cancelled using the *HostJob* parameter only after they have been entered in the Job Log (that is, transmitted to the host). The Job Log is marked so that data sets returning for the specified jobs are purged.

If you are connected to a VSE/POWER host system, this parameter is not allowed.

Wsid

Overrides the default *Wsid* named in the NRJE command, for this command only.

When you specify WSID=*Wsid* in the CANCEL command, you cancel jobs you have submitted at workstation *Wsid*.

Description

The CANCEL command enables you to remove previously submitted jobs from the transmission queue and, for JES2 and JES3 users, to cause NRJE to automatically purge output data sets for jobs that have already been transmitted to the host. Use the SHOW command to get information about jobs you want to cancel.

You can specify a particular workstation transmission queue by entering the WSID=*Wsid* parameter in this command. If you omit the *Wsid* parameter, this command refers to the transmission queue of the default workstation. You establish a default workstation by using the

`Wsid` parameter when you start an NRJE workstation with an `NRJE` command.

You must confirm each interactively issued `CANCEL` command (see example).

Restrictions

- Jobs you want to cancel before they are transmitted must be in the `READY` state. You can enter a `SHOW` command to determine the state of a job.
- If you are configured with NM capability, you can cancel any job. However, if you do not have NM capability, you can cancel only those jobs submitted under your logon *`UserName.AcctName`*.

Example

```
R19>CANCEL;HJ=6461
CONFIRM (Y/N)Y
1 HOST JOB(S) CANCELLED.

R19>SHOW;HJ=6461

--JOB#= 6461  JOBNAME= $SP#CI@L  USER=USER.NRJETEST      **CANCELLED**
  2 DATA SET(S) RECEIVED FROM HOST AS OF WED, AUG 5, 1987, 10:50 AM
  PR= (NOT SPECIFIED)  PU= (NOT SPECIFIED)  FO=FORMSOUT.NRJELU.NRJETEST

1 JOB(S) SHOWN TRANSMITTED TO HOST.

R19> ...
```

Text Reference

The `NRJECancel` intrinsic removes untransmitted jobs from the MPE Spooler. The `NRJEHJCancel` intrinsic marks the Job Log to dispose of returning data sets. See Chapter 5 , “User Intrinsics.”

CONSOLE

Obtains access to the host remote console of a workstation.

Syntax

```
CONSOLE
```

Description

This command obtains access to the host console of a workstation. While you have access to the console, the console mode prompt that you have configured for the workstation is displayed on the screen in place of the usual NRJE prompt. (The console mode prompt is configured in the NRJE Configuration: Workstation Data screen.)

The following conditions must be met before you can issue the `CONSOLE` command successfully:

- You must have NM capability or, if you do not have NM capability, this command must have been “allowed” by your node manager.
- The communications line for your workstation must be open; the workstation must be connected to the host job entry subsystem.
- The console facility must be free or available. Up to 28 users can access the console simultaneously.

Note that since you cannot specify a *wsid* in this command, you can obtain access to the console of the default workstation only.

Once you can access the console, you can enter host console commands. Each of these commands must begin with the configured host command prefix character, configured on the NRJE Configuration: Workstation Data screen. Here is an example:

```
CONS>$DA
```

CONS> is the console mode prompt configured for the workstation, and $DA is a JES2 host command. (The $ is the host command prefix character.)

NOTE NRJE does not check the host console command syntax. NRJE simply sends all the text you entered, including the host console command prefix, to your host as a console command.

If you are connected to a VSE/POWER host, the host console command prefix is required before each command, but by host system requirement, it is stripped off before transmission.

While you have access to the console, you receive all console messages issued by the host. So, you can receive console messages in response to your own commands, as well as messages in response to commands by other users. You also receive all other output sent by the host to the remote console. This includes host “Job Received” messages sent by some host systems when a job is received from a remote workstation. NRJE writes this output to your job or session output device, $STDLIST.

Node Management Services can also log messages (CLAS0013 in SUB0002) sent to the host remote console of your workstation. If no user has access to the console, you can retrieve a lost message by using the Node Management Services utility program, NMDUMP. See the *SNA Link Services Reference Manual* (for MPE V) or *Using the Node Management Services (NMS) Utilities* (for MPE XL) for more information about this utility.

If you currently have access to the console and no longer need it, issue a RELEASE command. That command terminates your access to the console and frees that facility for use by other users.

Example

```
:NRJE RMT11             Enter the NRJE subsystem with RMT11
                        as the default workstation.

HP30245A.60.00 NRJE (C) HEWLETT-PACKARD CO. 1987

WELCOME TO NRJE

RMT11> ...              Execute subsystem commands.

RMT11>CONSOLE           Acquire the console.

CON11> ...              Execute other subsystem commands from the
                        configured console mode prompt.

CON11>$DA               Issue a host remote console command, with
                        the configured host command prefix character.
                        output messages related to this command
                        might not return immediately.

CON11> ...              Execute subsystem commands.

CON11>RELEASE           Relinquish ownership of the console.

RMT11> ...              Remote workstation prompt is returned.
```

Text Reference

Programmatic access to the host remote console is provided by these manager intrinsics:

- `NRJEConsole`: obtains access to the console.
- `NRJESendCmd`: transmits a console command.
- `NRJEConsCheck`: determines whether any outstanding host remote console messages are waiting to be received.
- `NRJERcvMsg`: receives a console message.
- `NRJERelease`: relinquishes access to the console.

These intrinsics are described in Chapter 6 , “NRJE Manager Intrinsics.”

DISPLAY

Provides display of workstation and workstation-related information.

Syntax

```
D[ISPLAY] [I[NFO]] [;Q[UEUE]] [;WR[ITER]] [;AC[TIVEWS]] [;LUT[ABLE]]
         [;W[SID]=Wsid]
```

Parameters

INFO Provides this information:

- Whether your workstation is active or inactive.
- Which Job Entry System the host uses.
- The name of the remote workstation.
- The host command prefix character.
- The host console commands allowed for users without NM capability.
- The Lookup Table file name in effect for the workstation.
- The standard print form in effect for the workstation.
- The standard punch form in effect for the workstation.
- The text of the NRJE welcome message.

QUEUE Provides information about transmission from a workstation:

- The logical device number (*ldev*) of the spooled reader for this workstation.
- The chain size, in RUs, configured for the reader of this workstation.
- The number of readers.
- The compression indicator configured for this workstation. You might need to know this when you submit a job.
- The translation indicator configured for this workstation. You might need to know this when you submit a job.
- The status of the workstation reader.

- The value of the reader outfence. You might need to know this when you use the *PRIORITY* parameter of the SUBMIT command.
- The system *OutFence* value.

WRITER Displays information about each logical writer configured for this workstation:

- Name of the writer.
- Logical device number (*ldev*), logical device class names, or “formid” entry in the Lookup Table.
- The FORCED flag value: forces the formid to use the default output destination when output is routed to the writer and alternative routing is not specified in SUBMIT command. Will be Y or N.
- Output priority.
- Translation code.

ACTIVEWS Provides a list of all *active* NRJE workstations on your HP 3000.

LUTABLE Displays information about each logical unit:

- Logical unit name. An asterisk (*) beside a logical unit name indicates that it is active.
- Chain size: the number of RUs sent to the host before requesting a positive response.
- Logical unit address.
- Auto-start flag.

Wsid Overrides the default workstation named in your NRJE command. All information received is for workstation *Wsid*. Subsequent commands are not affected.

Description

The DISPLAY command enables you to obtain a broad range of information about NRJE workstations. You can obtain information about transmission queues, logical writers, and active workstations.

If you enter this command without any parameters, NRJE displays all the information possible about the default workstation. If you enter a DISPLAY command with only a *Wsid* parameter, NRJE displays all the information possible about the workstation specified in that parameter. The *Wsid* parameter overrides a default workstation established earlier in an NRJE command. As the following examples show, you can vary the output from the DISPLAY command, depending on the parameters that you enter.

When you enter DISPLAY INFO, or DISPLAY without any parameters, the State of a workstation is one of those described in Table 3-2.

The information that DISPLAY provides is presented one screen at a time if the command is issued from an interactive session.

Table 3-2 Workstation States

State:	**Meaning:**
ACTIVE	The workstation is connected to the host job entry subsystem.
STOPPEND *	One of these has been issued and is pending: • An NRJE subsystem STOPWS command. • An NRJE subsystem NRJEStopWS intrinsic call in stop mode. • One of these MPE commands: NRJECONTROL STOP ... SNACONTROL STOP ... ; TYPE=Q
SHUTPEND *	One of these has been issued and is pending: • An NRJE subsystem HALT command. • An NRJE subsystem NRJEStopWS intrinsic call in halt mode. • The MPE command SNACONTROL STOP...; TYPE=P
MONPEND	The NRJE monitor process has been activated, but the LUs are not yet active.
PENDING *	The NRJE monitor process has not yet been activated.
NACTIVE	The NRJE monitor process has not been created, and no STARTWS NRJE subsystem command, NRJEStartWS intrinsic call, or NRJECONTROL START MPE command has been issued.

* This is a highly transient state.

Example

```
R19>DISPLAY

Workstation: R19
GENERAL INFORMATION
                   State:  Pending
                     Jes:  JES2
                  Remote:  RMT19
Host command prefix char:  $
Allowed console commands:  $D
     Formid lookup table:  NRJETABL.NRJE
 Default print form name:  STD
 Default punch form name:  STD

WELCOME MESSAGE:
*********************************
*     Welcome to NRJE           *
*********************************
READER QUEUE
      Reader ldev:  12
       Chain size:  20
 Number of Readers:  1
      Compression:  YES
 Translation code:  0: NATIVE-3000
    Reader status:  UP
     Reader fence:  1
     System fence:  1

WORKSTATION LU LIST
LU Name    Chain Size   LU Address   Auto Start

LU01          20              0          YES
LU02          20              0          YES
LU03          20              0          YES
LU04          20              0           NO
LU05          20              0           NO
LU06          20              0           NO
LU07          20              0           NO
LU08          20              0           NO

LOGICAL WRITER
Writer Name   Output Destination   Forced   Priority     Translation Code

   PR1             LP                NO         8         0: NATIVE-3000
   PR2             EPOC              NO         8         0: NATIVE-3000
   PR3             'FORM1'           NO         8         0: NATIVE-3000
   PR4             LP                NO         8         0: NATIVE-3000
   PU1             'FORM2'           YES        8         0: NATIVE-3000
   PU2             EPOC              NO         8         0: NATIVE-3000

NO WORKSTATION IS ACTIVE. (CI=806)
R19>
```

Text Reference

The intrinsics related to this command are described in Chapter 5 , “User Intrinsics.” These are as follows:

- `NRJEQueue`: obtains information about the workstation transmission queue.
- `NRJEWSInfo`: obtains general information about the workstation.
- `NRJEWriter`: obtains information about configured host logical writers for a given workstation.
- `NRJEActiveWS`: finds out how many and which NRJE workstations are active.
- `NRJEWelcome`: receives the NRJE welcome message.
- `NRJELUList`: provides a list of all LUs and related characteristics for a workstation.

EXIT

Causes the user to exit from the NRJE command interpreter (CI) and returns control to the process from which the CI was run.

Syntax

```
E[XIT]
```

Description

The EXIT command exits you from the NRJE command interpreter (CI) and returns control to the process that called the CI. If you have specified a default workstation (or no workstation) and then changed it by using an NRJE command, use the NRJE command to return to the previous workstation.

Example

Enter NRJE from MPE (in this case, without a *Wsid* parameter).

```
:NRJE
```

product number version number NJRE (C) HEWLETT-PACKARD CO. *copyright year*

```
WELCOME TO NRJE
```

Enter NRJE from within itself using RMT19 as the default workstation.

```
>NRJE RMT19
```

Do some commands affecting RMT19.

```
RMT19> ...
```

Return control to MPE by using an EXIT command.

You do not return to an outer NRJE. If you want to do so, you should use another NRJE command.

```
RMT19>EXIT

: ...
```

You have returned to MPE.

##FD

Used within a job to include the contents of a file.

Syntax

```
##FD Infile [(Options)] [Comment]
```

Parameters

Infile Specifies the file whose contents are to be included. This is any valid MPE file name. For both $STDIN and $STDINX you must close with :EOD.

(*Options*) Any combination of these:

```
N[OTRANSLATE]
T[RANSPARENT]
```

- NOTRANSLATE causes the FD file to be transmitted to the host without translation.
- TRANSPARENT causes NRJE to ignore all JOB cards in the ##FD and pass them to the host as data. Otherwise, JOB cards in the ##FD file are treated as job delimiters by NRJE. If this option is included, the proper JCL must be included so that the host also treats the JOB card as data. To use this option you must be configured with NM capability.

If both *Options* are specified, they must be separated by a comma (,). If no options are specified, the default will be file translation to EBCDIC and no transparency.

Comment Any character string following the closing parenthesis. The *Comment* becomes part of the prompting string when the Infile is $STDIN or $STDINX.

Description

Use the ##FD command within a job to include the contents of the *Infile*. ##FD files can be nested up to 20 levels deep. When a job file is submitted, NRJE scans each record for occurrences of ##FD in columns 1–4. Whenever it finds one, it opens the *Infile* and adds the contents to the job being spooled for transmission to the host. Any options specified must be compatible with the SUBMIT options and the workstation configuration.

If a job card is included in the *Infile*, then its format must adhere to the same syntax as that specified for the SUBMIT command. Refer to the “Description” section of the SUBMIT command discussed later in this chapter.

If a file is built that requests input from $STDIN or $STDINX, the optional comment field on the ##FD card should be used to identify the type of input that is expected. Then, when the job input stream is constructed, the comment is printed at the user's terminal ahead of the standard NRJE message:

```
ENTER INPUT ENDING WITH ":EOD"
```

For example, if an *Infile* DATA1 contains the ##FD record shown in example three below, then the following will appear at the user's terminal:

```
THIS INPUT IS FOR FILE DATA1.
ENTER INPUT ENDING WITH ":EOD"
```

The ##FD card, in this case, prompts the user for interactive input into the DATA1 file. Note that the comment has been capitalized by NRJE.

Example

```
##FD TATK100J (N)                         TATK100J is already translated.

##FD DS1 (T)                              Job cards are ignored.

##FD $STDIN This input is for file DATA1. Prompts user for interactive input.
```

HELP

Provides you with information about subsystem commands.

Syntax

```
H[ELP] [NRJECommand [,Keyword]]
       [MSGnumber]
```

Parameters

NRJECommand A full or abbreviated version of any NRJE command. If you omit this parameter you receive a list of all NRJE commands.

NRJE displays the command name, the command syntax, and a list of keywords for the command.

MSG Number An integer representing any valid NRJE message. Information on most frequently referenced error messages is available.

Keyword Keywords for all commands are as follows:

```
S[YNTAX]
PA[RMS]
O[PERATION]
A[LL]
```

- `SYNTAX` lists the syntax of the specified command (this is the default if no *Keyword* is specified).
- `PARMS` lists all parameters of the specified command.
- `OPERATION` discusses the use of the specified command.
- `ALL` lists all help information for the specified command.

The *Keyword* parameter can be entered with a command. For example, `HELP SUBMIT,PARMS`. In this case, `PARMS` is a keyword and must be separated from the command with a comma.

If the keyword `ALL` is used, `SYNTAX`, `PARMS`, and `OPERATION` information is displayed, and you are prompted for a carriage return before each new set of information.

Description

The HELP command enables you to obtain online information about NRJE commands and error messages. If you enter HELP without any parameters, NRJE displays a list of commands. If you specify any NRJE command in the *NRJECommand* parameter and omit the *Keyword* parameter, you receive a one-line description of that particular command and its syntax. Output is displayed one screen at a time, and you are prompted for a carriage return before each new screen.

The function of this help facility is closely patterned after MPE's help facility.

Example

```
R20>HELP NRJE
NRJE

Switch to a new workstation.

SYNTAX:  N[RJE] wsid

KEYWORDS: SYNTAX, PARMS, OPERATION, ALL


R20>HELP NRJE,OPERATION
OPERATION

The NRJE command switches the NRJE command interpreter to a new workstation.
The next prompt displayed is that configured for the new workstation.

KEYWORDS: SYNTAX, PARMS, OPERATION, ALL
R20>
```

NRJE

Invokes the subsystem command interpreter. It is also an NRJE subsystem command and allows you to switch to a different default workstation.

Syntax

```
N[RJE] Wsid
```

Parameter

Wsid Identifies a specific workstation you wish to use as a default. The *Wsid* parameter is an alphanumeric string that must begin with a letter and can be up to eight characters long.

You must use *Wsid* in an NRJE subsystem command; it is optional in an MPE command.

To specify another workstation within the subsystem you can

- Use a different *Wsid* as a parameter in every command that affects a workstation.
- Execute another NRJE command with a different *Wsid* to establish another default workstation.

Although the *Wsid* parameter is optional when you enter the NRJE subsystem from MPE, using the parameter is usually convenient, because it establishes a default *Wsid* and allows you to omit the *Wsid* parameter from all subsystem commands for that workstation.

You can always specify another *Wsid* value in other subsystem commands, or you can change the default *Wsid* by issuing another NRJE command.

Description

The NRJE command enables you to enter the NRJE subsystem from MPE or, when used as an SNA NRJE subsystem command, to change default workstations. When you issue the command from MPE, the SNA NRJE subsystem prints a banner line identifying the version of NRJE installed on your computer. You then receive the NRJE welcome message and a workstation prompt string. When you enter an NRJE command from the subsystem, you receive only the workstation prompt string.

If you enter an NRJE command from MPE, the *Wsid* parameter is optional; however, it is required if you issue the command within the NRJE subsystem. In either case, when you specify a *Wsid* in an NRJE command, you designate a default workstation for your NRJE session. A workstation specified in an NRJE command remains the default workstation for your NRJE session until you enter another NRJE command with its own *Wsid*.

When you establish a default workstation, you can utilize NRJE user and manager commands more efficiently. For example, the SUBMIT command contains an optional WSID=*Wsid* parameter. Once you have established a default workstation, you can issue a SUBMIT command for a workstation without appending a WSID=*Wsid* parameter. The subsystem defaults to the workstation specified in your last NRJE command.

NOTE If you do not set up a default workstation, you must include the WSID=*Wsid* parameter in every command that includes that parameter.

After entering NRJE from MPE, you can set up a new default workstation by issuing an NRJE command with a *Wsid*. If you did not identify a *Wsid* in your initial NRJE command, your new NRJE command with the *Wsid* establishes the first default workstation for your session. If you did identify a *Wsid* in your initial NRJE command, your new NRJE command enables you to switch to a different default workstation.

You can designate a default workstation and then enter certain user or manager commands with a WSID=*Wsid* parameter. In this case, the subsystem executes the command for the workstation in the WSID=*Wsid* parameter. However, use of a WSID=*Wsid* parameter does not change the default workstation. If you then issue another command without the WSID=*Wsid* parameter, the subsystem executes that command for the default workstation. The following example illustrates switching from RMT11 as a default workstation to RMT99 as a default, and then back to RMT11. Remember that an EXIT command always returns control to the process that called the NRJE CI.

Example

```
: ...              You are logged on to MPE.

:NRJE RMT11        Enter NRJE with RMT11 as the default workstation.

product number version number NJRE (C) HEWLETT-PACKARD CO. copyright year

NRJE WELCOME       The NRJE WELCOME text is configured by an
                   NRJE manager.

RMT11> ...         Execute NRJE commands with RMT11 as the default
                   workstation. The prompting string is configured
                   in the NMMGR Workstation Data screen.

RMT11>NRJE RMT99   Switch to RMT99 as the default workstation.

RMT99> ...         Execute NRJE commands with RMT99 as the default.

RMT99>NRJE RMT11   Switch to RMT11 as the default workstation.

RMT11> ...         Execute some NRJE commands with RMT11 as the default
                   workstation.

RMT11>EXIT         Return control to MPE.

:  ...             Execute some MPE commands.
```

In the example above, you enter NRJE specifying RMT11 as the default workstation. After executing some commands, you switch the default workstation from RMT11 to RMT99, and then back to RMT11. Eventually, you return control to MPE.

Text Reference

See the *SNA Link Services Reference Manual* or the *SNA NRJE Node Manager's Guide* for workstation configuration information. See the *SNA NRJE Node Manager's Guide* for a description of NRJE manager commands.

REDO

Allows you to modify and/or execute the previous NRJE subsystem command. Use this like the MPE REDO command.

Syntax

```
RED[O]
```

Subcommands

D

Deletes the character above the D. If the D is repeated, each character above each D is deleted.

I

Inserts one or more characters ahead of the character above the I. The D and I subcommands can be used together to replace strings of unequal length.

R

Replaces the characters above the R with the characters that immediately follow it.

U

Cancels the preceding subcommand, unless it was a U, in which case the original NRJE command line is restored.

Any character but D, I, R, or U.

Replaces one-for-one the characters in the string with the characters you enter.

Description

The REDO command allows you to correct certain kinds of errors in an incorrect command entry or to change a correct command entry, eliminating the need for re-entering the command in its entirety. The REDO command applies only to the latest command entered. When a REDO command is entered, NRJE enters a mode similar to the Editor and displays the command to be modified.

To modify the command output by NRJE, use the space bar on the terminal to position the cursor under the characters to be modified, then enter one of the subcommands described above. If you space over too far, use the backspace key to reposition the cursor.

The REDO command operates exactly like its MPE counterpart.

Example

```
>SHEW MINE      An incorrect command was entered.

NO SUCH COMMAND. (CI=721)

>REDO

SHEW MINE       The incorrect command is displayed.

  RO            Enter the R subcommand below the character “E”
                you are replacing, followed by the correct character “O”.

                You could have omitted the R subcommand.

SHOW MINE       The correct command is displayed.

[RETURN]        Enter another carriage return to execute this command.
```

REMARK

Provides a way to do in-line comments when you use NRJE subsystem commands in an HP 3000 job.

Syntax

```
REM[ARK] [String]
```

Parameters

String	Contains any printing character. If you omit the *String* in an HP 3000 job, only the REMARK command itself is printed.

Description

The REMARK command enables you to include comments or notes within a sequence of subsystem commands in job listings produced on hard copy devices, to create headings or explain the purpose of commands. After a REMARK command is entered, it can be followed by a *String* made up of any ASCII characters. This subsystem command is used primarily in batch jobs but also can be used in sessions.

Example

```
>REMARK                   No comment.
>REMARK Comment           Comment.
>REM                      No comment.
```

RELEASE

Relinquishes access to the host remote console of a workstation.

Syntax

```
REL[EASE]
```

Description

Use this command to relinquish access to the host remote console of a workstation. You must have NM capability or be an “allowed” console user to use this command. You remain in the console of a workstation until a RELEASE command is issued or until you exit from the NJRE CI. Note that, since you cannot specify a *Wsid* in this command, you can release access to the default workstation only.

Text Reference

See CONSOLE earlier in this chapter for a general description of how to use the host remote console. NRJERelease is the manager intrinsic counterpart of this command; it is described in Chapter 6 , “NRJE Manager Intrinsics.”

RUN

Executes a prepared program.

Syntax

```
RU[N] ProgramFile [;PARM=n]
```

Parameter

ProgramFile	Identifies a program file. A *ProgramFile* is an actual designator of a program file containing a prepared program.
PARM	Contains the run time parameter. Set to zero if not specified.

Description

The RUN command enables you to execute a prepared *ProgramFile* from within the subsystem. *This command is different from an MPE* RUN *command*.

Restrictions

- You cannot specify an *entrypoint* to the *ProgramFile*.
- Only the run parameter *PARM* may be specified. Other parameters such as INFO and LIB may not be specified.

SHOW

Provides information about jobs you have submitted.

If you are connected to a VSE/POWER host system, only those jobs awaiting transmission are shown.

Syntax

```
          {        SpoolfileID          }
          {           Name              }
          {      UserName.AcctName      }[;W[SID]=Wsid]
SH[OW]    {             @               }
          {;HJ=HostJob [,HostJob[,...]] }
          {;HJ=FirstJob/LastJob         }
          {;HJ=#O SpoolfileID           }
```

Parameters

SpoolfileID

Use when you want to show information about a specific job. The *SpoolfileID* parameter is assigned by the spooler after you submit a job. Its form is nnnnnnn, where n is a digit. Jobs can be shown by *SpoolfileID* only before they are transmitted to the host. To view job status by *SpoolfileID* after the job is transmitted to the host, you would indicate the output spool ID in the *HostJob* parameter.

Name

The spool file name associated with your job in the NRJE reader queue and the name associated with your job in the Job Log. You can submit several jobs with the same name, and then use the SHOW *Name* command to receive information on all of them.

UserName.AcctName

Use for information about jobs submitted by *UserName.AcctName*.

A period (.) must separate *UserName* from *AcctName*.

If your logon *UserName.AcctName* is different from what you specified, you must be configured with NM user capability to use this parameter.

@

Use to show information about all jobs you submitted, when your logon *UserName.AcctName* is the same as the job *UserName.AcctName*. If you have NM capability, all jobs submitted by other users are shown, also.

HostJob

Either the jobname from the job card or the job number assigned by the host. Jobs can be shown by *HostJob* only after they have been entered in the Job Log (that is, transmitted to the host). Multiple job numbers and names are allowed; a maximum of five single job numbers or jobnames are allowed, delimited by commas (,). There is no limit when a range is specified.

You can also view job status after the job is transmitted to the host without getting into the console (;HJ=*#0 SpoolfileID*).

Wsid

Overrides the default *Wsid* named in the NRJE command, for this command only.

Unless you are configured with NM capability, you must be the submitter of the jobs you want to SHOW.

Description

The SHOW command enables you to display a list of specified jobs submitted to the host. You specify the list of jobs to show by entering a *SpoolfileID*, a *Name*, a *UserName.AcctName*, an @ symbol, or *HostJob* name or number. You can specify a particular workstation transmission queue by entering the WSID=*Wsid* parameter in this command. If you omit the *Wsid* parameter, this command refers to the transmission queue of the default workstation. You establish a default workstation by using the *Wsid* parameter when you start an NRJE workstation with an NRJE command.

The display from the SHOW command is split into two sections. The first section provides information on jobs that have not yet been transmitted to the host. The second section provides information on jobs that have already been transmitted. The second section of information comes from the Job Log and therefore is not provided if you are connected to a VSE/POWER host system.

For MPE XL, release 2.1 or later, only the first 100 untransmitted jobs are shown at a time.

Untransmitted Files

The list you receive identifies each specified job by *SpoolfileID*, *Name*, and *UserName.AcctName*. The priority, state, rank size, and time when each specified job was submitted are also displayed. Each column of output provided by the SHOW command for untransmitted files is described below.

Column Heading	Contents
#O	A SpoolfileID
Name	The name of the file associated with your job.
Pri	The spool file transmission queue priority number. The highest priority is 14, the lowest 0.
State	The spooling state of the file:

MPE V and MPE XL prior to release 2.1	MPE XL release 2.1 and later
ACTIVE	ACTIVE
READY	READY
OPEN	OPEN
LOCKED	DEFER
	CREATE
	TRANSFER
	PRINT
	PROBLEM
	DEL_PENDING
	SPSAVE

Refer to the *MPE V/E System Operation and Resource Management Reference Manual* or the *MPE XL Native Mode Spooler Reference Manual* for an additional discussion of spooling states.

Column Heading	Contents
Rank	The order of a file within its transmission queue.
Size	The number of spool file sectors of the file.
User.Account	The *UserName.AcctName* of the submitter of this job.
Time	When this job was submitted.

Transmitted Files

For files that have already been transmitted to the host, display of the PRINT, PUNCH, and FORMS dispositions is in a free format. NRJE attempts to display these fields on a single line if possible, but will use multiple lines as necessary. If you took the default for any of these options, NRJE displays the string (NOT SPECIFIED) for each applicable option. In such cases, output routing depends on the configured defaults for the remote logical writers and any routing designations using form names and Lookup Table entries. The keyword FILE indicates that the user entered a backreferenced formal file designator. See the SUBMIT command for information on the PRINT, PUNCH and FORMS options.

Restriction

You must be configured with NM capability to use this command for jobs outside of your logon *UserName.AcctName*.

Example

```
RMT32>REMARK  THIS SHOW @ IS BY A USER WITH NM CAPABILITY

                Provide information about all jobs.

RMT32>SHOW @

  #O Name     Pri     State     Rank     Size     User.Account     Time
1069          13      READY      1        32       KEN.NRJE        10:48
1075          12      READY      2        20       ANNE.NRJE       10:55

2 JOB(S) AWAITING TRANSMISSION TO HOST.

--JOB#=1030 JOBNAME=DS1    USER=ANNE.NRJE
  TRANSMITTED TO HOST WED, AUG 5, 1987, 4:05 PM HOST#=3022
  PR= (NOT SPECIFIED)  PU= (NOT SPECIFIED)  FO= (NOT SPECIFIED)

--JOB#=1022 JOBNAME=I      USER=KEN.NRJE
  5 DATA SETS RECEIVED FROM HOST AS OF WED, AUG 5, 1987, 4:20 PM
  PR= (NOT SPECIFIED)  PU= (NOT SPECIFIED)  FO= (NOT SPECIFIED)

2 JOB(S) TRANSMITTED TO HOST.

RMT32>REMARK  THIS SHOW @ IS BY A USER WITHOUT NM CAPABILITY

                  Provide information about all jobs owned by user who
                  entered SHOW command.

RMT32>SHOW @

  #O Name     Pri     State     Rank     Size     User.Account     Time
1069          13      READY      3        32       KEN.NRJE        10:48

1 JOB(S) AWAITING TRANSMISSION TO HOST.

--JOB#=1022 JOBNAME=I      USER=KEN.NRJE
  5 DATA SETS RECEIVED FROM HOST AS OF WED, AUG 5, 1987, 4:20 PM
  PR= (NOT SPECIFIED)  PU= (NOT SPECIFIED)  FO= (NOT SPECIFIED)

1 JOB(S) TRANSMITTED TO HOST.

RMT32>REMARK  THIS USER HAS NM CAPABILITY
RMT32>SHOW ;HJ=1022,1030

--JOB#=1022 JOBNAME=I      USER=KEN.NRJE
  5 DATA SETS RECEIVED FROM HOST AS OF WED, AUG 5, 1987, 4:20 PM
  PR= FILE OUTPUT;DEV=PP;ENV=LP602.HPENV.SYS  PU= (NOT SPECIFIED)
  FO=discfile.mygroup.myacct.

--JOB#=1030 JOBNAME=DS1    USER=ANNE.NRJE
  TRANSMITTED TO HOST WED, AUG 5, 1987, 4:05 PM HOST#=3022
  PR= (NOT SPECIFIED)  PU= (NOT SPECIFIED)  FO= (NOT SPECIFIED)

2 JOB(S) TRANSMITTED TO HOST.
```

Text Reference

The following intrinsics are related to this command:

- `NRJEJobInfo`: returns information about jobs awaiting transmission.
- `NRJEHJInfo`: returns information about transmitted jobs from the Job Log.

These intrinsics are described in Chapter 5 , “User Intrinsics.”

SUBMIT

Prepares a batch input job stream for transmission to a host system.

Syntax

```
                    (N)
S[UBMIT] Infile [ (NOTRANSLATE) ], ...

         [ ;D[IRECT] ]
         [ ;PRIO[RITY] = n ]
         [ ;J[OBNAME] = Name ]
         [ ;W[SID] = Wsid ]
         [ ;PR[INT] = PrintFile ]
         [ ;PU[NCH] = PunchFile ]
         [ ;F[ORMS] = FormFile ]
         [ ;M[AXREC] = MaxRecSize ]
```

Parameters

Infile

Identifies a file to be submitted. An *Infile* can be a qualified or unqualified file designator, or a formal file designator which has been equated with an actual file designator. See the discussion below for additional information on *Infile* designation.

System files $NULL, $STDLIST, and $NEWPASS are not allowed as *Infiles*.

For both $STDIN and $STDINX, you must close with :EOD.

You must specify at least one *Infile*; you can specify up to five.

(NOTRANSLATE)

Indicates that the contents of the *Infile* should not be translated into the character code configured for the host system. You must include the ( ) as part of this parameter. The abbreviated form of (NOTRANSLATE) is (N).

If you indicate NOTRANSLATE) for some *Infiles*, the other *Infiles* will be translated while being stored in the transmission queue. You cannot use (NOTRANSLATE) and DIRECT together.

DIRECT

When you use this parameter, translation and compression will be done during transmission to the host.

You cannot use (NOTRANSLATE) and DIRECT together.

n

Identifies a spool file transmission queue priority number. The highest priority is 14, and the lowest is 1.

Job transmission to the host can be deferred by specifying a priority value less than or equal to the reader fence value, RDRFENCE, for a workstation. If RDRFENCE=0, then OUTFENCE, an MPE parameter, is used to select job spool files for transmission to the host. The configuration of RDRFENCE is described in the *SNA NRJE Node Manager's Guide*.

If you omit this parameter, the MPE-configured default is used.

Name

Identifies a job when it is in a transmission queue. You can use this *Name* in subsequent NRJE commands.

A *Name* must begin with a letter and may contain up to eight alphanumeric characters.

If you give several jobs the same *Name*, then other NRJE commands, such as SHOW, can refer to all job files with that *Name*.

If you omit this parameter, the job you submit and output from the job are assigned the name specified in the JCL JOB card.

Wsid

Identifies an alternate NRJE workstation on your node to which this command applies. The job will be submitted to the reader in the specified alternate workstation.

You normally omit this parameter and submit jobs to the workstation you identified in the NRJE command. This parameter does not alter the default workstation for subsequent commands.

PrintFile

Used to specify routing for output data sets that are sent to an NRJE logical printer on the standard form.

PunchFile

Used to specify routing for output data sets that are sent to an NRJE logical punch on the standard form.

FormFile

Refers to print or punch output routed to any form other than the standard form.

PR=, PU=, and FO= specifications require the use of the Job Log. Use of these parameters is not supported for workstations connected to a VSE/POWER host.

One of each type of file (PRINT, PUNCH, and FORMS) can be specified by any of the following:

- A formal file designator (**fileid*) which backreferences an actual file designator.
- The logical device number (*ldev*) of an MPE output device (numeric entry).
- An MPE device class name, enclosed in double quotation marks (*"DevClass"*).
- An actual file designator (alphanumeric). The file must be built before the job is submitted. Both the user and MANAGER.SYS must have READ/WRITE access to the destination file.
- An alternate form name (**formid**) used as a key into the Lookup Table. This designation is made with single quotation marks (*'formid'*). This designation references a predefined file equation in the NRJE output management Lookup Table. See below under "Default FOPEN Parameters Used," and under "Job and Output Management" in Chapter 4 , "Job Output," for additional information.

Limitations:

- The total length of *PrintFile*, *PunchFile*, and *FormFile* entries cannot exceed 197 bytes. Each item cannot exceed 86 bytes in length.
- SNA NRJE does not support NOWAIT I/O, MULTIRECORD ACCESS, or any combination of file equation attributes which require Privileged Mode to open the file.
- The maximum record width for output files is 255 bytes and the minimum is 40 bytes. Output directed to files which are not wide enough to accommodate the received data records may be

wrapped by NRJE. If a received record exceeds the width of the destination file, output is segmented and any extra characters are written to the line immediately following the truncated record.

MaxRecSize

Identifies the maximum record size (in bytes) that will be encountered in the *Infiles* or ##FD files. The minimum usable value is 80. The maximum usable values are as follows:

- 248 for JES2 or JES3 workstations that use a 256-byte RU.
- 252 for JES2 or JES3 workstations that use a 512-byte RU.
- 252 for JES2 or JES3 workstations that use a 768-byte RU.
- 128 for VSE/POWER workstations.

The default *MaxRecSize* is 80.

No intrinsic (programmatic) support is provided for this parameter.

Description

The SUBMIT command enables you to prepare a set of *Infiles* for transmission to a host system. You can specify up to five *Infiles* in a SUBMIT command. Taken together, these files can contain one or more host system batch jobs.

The host jobs contained in the submitted *Infiles* (and any included ##FD files) are broken into separate spool files. A list of the *SpoolfileIDs* of the jobs is returned to you. If the submitted files comprise more than 30 host jobs, all jobs are submitted to the host, but only the first 30 *SpoolfileIDs* are returned.

Infile Designation

An *Infile* can be a file reference, or a formal file designator that has been equated with an actual file designator by a backreference indication.

A file reference is in the following form:

Infile[/*LockWord*][.*GrpName*[.*AcctName*]][:*Envid*]

GrpName and *AcctName* refer to the group name and account name, respectively, where the file is located. If you do not specify an *:Envid*, the default values for *GrpName* and *AcctName* are those for your logon on your HP 3000.

SNA NRJE provides the capability of submitting files that are stored at remote locations. The *:Envid* parameter is used to identify the remote location from which the file is to be submitted. This capability is possible if you have either DS/3000 services in a point-to-point connection to the remote, or NS/3000 services if you are connected in a Local Area Network (LAN) with the remote.

If you have DS/3000 services, *:Envid* is a character string representing a specific logon session on the remote. If you have NS/3000 services on your HP 3000, *:Envid* is the name assigned to the remote node as it is configured into the LAN. Refer to the *NS/3000 User/Programmer Reference Manual* for additional information on designating *:Envid*.

If you specify an *:Envid*, the default *GrpName* and *AcctName* are those of your logon on the remote machine.

Formal file designators backreference actual file designators and must be prefixed with an asterisk (*). Actual file designators are specified in FILE equations created prior to the backreference.

An example of a formal file designator is IN, where a backreference is made to an MPE FILE statement that identifies card reader images of 80-byte records:

```
:FILE IN;DEV=TAPE;REC=-80,16,F,ASCII
:NRJE RMT9
...
RMT9>SUBMIT *IN
...
```

At least one valid job card must be contained within an Infile specified with SUBMIT. This job card may be contained within the Infile or any included ##FD file. The job card must adhere to the following format:

Column 1,2	//
Column 3/10	job name beginning with a letter
Column 11,12,13	JOB

Translation

While preparing your *Infiles* for transmission, NRJE can perform all necessary data translation. You can vary this process by entering various combinations of parameters. You can utilize these parameters to inhibit translation for a variety of reasons, for example, if a file contains non-printable characters such as packed decimal numerical data.

Files submitted already translated must contain only one host record per HP 3000 file record. SNA NRJE expects the following in each record:

- An SNA String Control Byte (SCB) in byte 0.

- A TRN SNA Character String (SCS) control word in bytes 1 and 2.
- An IRS SCS, which is used to delimit records, as the last byte.

If these restrictions are not met, NRJE features such as job management and output management and support of ##FD files may not function correctly for these jobs.

Default FOPEN Parameters

Table 3-3 lists the default FOPEN parameters that NRJE uses when opening the output file for PRINT output. (Differences for PUNCH and FORMS output are noted.)

Table 3-3 Default FOPEN Parameters

NRJE SUBMIT Option	foptions	aoptions	Comments
None, no Lookup Table in use.	%2404	%4	ASCII, CCTL, read/write, no file equation
None, Lookup Table in use, standard form.	%2404	%4	ASCII, CCTL, read/write, no file equation
None, Lookup Table in use. JCL specifies special form.	%605	%204	OLD permanent, ASCII, CCTL, undefined record format, semi-exclusive, read/write[1]
PR=**filereference*	%605	%204	OLD permanent, ASCII, CCTL, undefined record format, semi-exclusive, read/write[1]
PR='*formid*'	%605	%204	OLD permanent, ASCII, CCTL, undefined record format, semi-exclusive, read/write[1]

PR= *"devclass"* or
PR=*ldev*
	%2604	%204	ASCII, CCTL, no file equation, undefined record format, semi-exclusive, read/write

PR=*discfilename*	%2001	%344	Old permanent, no file equation, read/write, FLOCK, shared[1]

[1] If a job generates multiple datasets to the same form, and output is directed to a disk file, then **read/write** access is used for the first data set only. All subsequent data sets are written with **append** access.

PUNCH output differences: For PUNCH output, the parameters are the same as described in Table 3-3, except that NRJE specifies NOCCTL and BINARY.

FORMS output differences: If you are using the FORMS= option, NRJE uses the options specified in Table 3-3, except for using NOCCTL and BINARY if punch output is routed to a special form. Also, if FORMS output is routed without a Lookup Table, or if FORMS output is routed to a device class or *ldev*, the name of the form is passed in the forms field of the FOPEN intrinsic.

This causes a hard copy forms mount request for the form specified to be sent to the HP 3000 console when the spoolfile goes active.

If the FCB field is specified in your JCL, and FORMS output is routed without the Lookup Table, and you do not specify any alternate PR=, PU=, or FO= destinations, the FCB field is passed to FOPEN as an environment file name in the group HPENV.SYS. Note that in all other cases, the FCB field of the PDIR is ignored.

NOTE You can override the default FOPEN options by specifying a backreferenced formal file designator; for example,

```
FILE out=myfile;acc=append
```

Then, in the SUBMIT command include the following:

```
;PU=*out
```

You may track the progress of your job using the host console facility. NRJE users may access the console provided that one or more console commands have been allowed to users by the NRJE manager. Refer to the *SNA NRJE Node Manager's Guide* for additional information on console commands.

Other Parameters

Unless you are routing with PR=*discfilename*, NRJE specifies a record size of 256 bytes and uses the number of copies specified in the PDIR. This value is set by the host in response to the number of copies specified in your JCL. NRJE uses the priority configured in the NMMGR Writer Data Screen for the logical writer on which the output was received.

Console Commands

Console commands can be embedded in a submitted file. If included, these commands must precede the JCL JOB card. If multiple jobs are included in an input file, the host commands must precede the first JCL JOB card. Embedded console commands between jobs in a single input stream are not supported.

Note that “single input stream” means either a single input file with multiple JOB cards or multiple input files entered with a single SUBMIT command. If you have NM capability, or the embedded command is one of those allowed to all users (specified in the NMMGR Workstation Data screen), the console command is passed on as part of the job. Otherwise, the command is stripped out and an error message is displayed. The remainder of the job is submitted. SIGNOFF cards are stripped from the input stream and an error message is displayed.

Embedded console commands must be preceded by a host console command prefix so that they are recognized as commands. The host console command prefix you use depends on the host system to which you are connected:

- JES2: commands must be preceded by `/*[]`, where `[]` is the host command prefix configured for your workstation in the NMMGR Workstation Data screen.
- JES3: commands must be preceded by `//*[]`, where `[]` is the host command prefix configured for your workstation in the NMMGR Workstation Data screen.
- VSE/POWER: commands must be preceded by * Δ. . Δ, where Δ is a required space.

NOTE The asterisk (*) is required by NRJE to parse console commands and is not required by the host.

Additional Notes

The communications line to a host must be opened before the subsystem can transmit files in the corresponding transmission queue of the workstation. The communications line is opened as part of the workstation startup.

In executing a `SUBMIT` command, NRJE prepares a spool file, but it does not send the file to a host. If the communications line is down, your files remain in the transmission queue. Once the communications line is open, NRJE can transmit the spool files that have a priority greater than the *`ReaderFence`* value, or if that value is zero, the MPE *`OutFence`*. The subsystem defers files that have a priority less than or equal to the operative fence value.

NRJE transmits undeferred files that have the highest priority first. You can influence the initial position of a file in a transmission queue by using the *`PRIORITY`* parameter in the `SUBMIT` command.

You can specify a particular workstation transmission queue by entering the `WSID=`*`Wsid`* parameter in this command. If you omit the *`Wsid`* parameter, this command refers to the transmission queue of the default workstation. You establish a default workstation by using the *`Wsid`* parameter when you start an NRJE workstation with an `NRJE` command.

Output from job streams is received when the communications line with the host system is active. See Chapter 4 , “Job Output,” for a discussion of how job output is received. Also see the *Installing and Troubleshooting SNA NRJE Manual* or the *SNA NRJE Node Manager’s Guide* for information on what to do when your output appears to go to the wrong destination.

CAUTION

Do not direct multiple output data sets of different types (that is, PRINT, PUNCH, or FORMS) to the same disk file; for example,

```
SUBMIT myjob;PR=myfile;FO=myfile
```

This causes data to be overwritten in the file myfile.

Example

```
:FILE OUTLIST;DEV=PP;ENV=LP2.HPENV.SYS
:NRJE R20

product number version number NRJE (C) HEWLETT-PACKARD CO. copyright year
WELCOME TO NRJE
R20>SUBMIT myjob;pr=*outlist;fo='long';prio=9

JOB SUCCESSFULLY SUBMITTED. THE SPOOL FILE IS:
#O86
R20>...
```

This example causes a copy of the file myjob to be placed in the NRJE reader queue with a transmission priority of 9. If the link to the host system is up, myjob is transmitted when it is the first in the queue. Successful submittal is indicated by a message also identifying the spool file.

Output data sets directed by your JCL to the host **standard form** on a logical printer device are routed according to the attributes in the file equation FILE outlist ... (determined by the *pr=*outlist* parameter).

Output data sets directed by your JCL to any **special form** on a printer or punch logical device are routed according to the attributes in the Lookup Table under the entry long.

Example

```
R20>SUBMIT FRONTJCL,BINDATA(NOTRANSLATE),BACKJCL

JOB SUCCESSFULLY SUBMITTED.  THE SPOOL FILE IS:
#O95
R20>SUBMIT NEWJOB(N)

JOB SUCCESSFULLY SUBMITTED.  THE SPOOL FILE IS:
#O96
R20>...
```

The examples demonstrate how to submit binary data to a host. In the first example, three *Infiles* are concatenated to one job file. The file FRONTJCL begins the job stream, followed by file BINDATA and then BACKJCL.

The (NOTRANSLATE) parameter after the BINDATA file name indicates that this file has already been translated into the character code configured for the host system.

In the second example, the file NEWJOB is already in EBCDIC and does not need to be translated by NRJE. Note that NEWJOB must still contain a valid job card.

Multiple files can be included by submitting a single file that contains ##FD statements referencing the JCL and/or data files. See “Job Input” in Chapter 2 , “Getting Started,” for suggestions and examples.

Text Reference

The intrinsic for this command is NRJE2Submit. It is described in Chapter 5 , “User Intrinsics.”

VERSION

Identifies the software version of SNA NRJE.

Syntax

```
V[ERSION]
```

Description

The VERSION command enables you to identify the versions of SNA NRJE program files, segments, the message file, and the overall version of the subsystem. Output from this command is sent to $STDLIST.

Example

```
R20>version

***************   NRJE VERSION CHECK    ***************

PROGRAM      NRJE.NRJE.SYS                HAS VERSION:A6000000
PROGRAM      NRJECONS.NRJE.SYS            HAS VERSION:A6000000
PROGRAM      NRJEMON.NRJE.SYS             HAS VERSION:A6000000
PROGRAM      NRJELU.NRJE.SYS              HAS VERSION:A6000000
PROGRAM      NSHUT.NRJE.SYS               HAS VERSION:A6000000
SL SEGMENT   NRJEMISCVERS                 HAS VERSION:A6000000
SL SEGMENT   NRJEMISC2VERS                HAS VERSION:A6000000
SL SEGMENT   NRJEMISC3VERS                HAS VERSION:A6000000
SL SEGMENT   NRJEINTVERS                  HAS VERSION:A6000000
SL SEGMENT   SUBSYS2FMTVERS               HAS VERSION:A6000000
MESSAGE FILE NRJECAT.NRJE.SYS             HAS VERSION:A6000000

OVERALL VERSION: A.60.00
```

Different versions of NRJE may have different modules installed. The first five digits of each module's version number should match and be reflected in the overall version level.

Text Reference

This command has the same effect as the MPE command NRJECONTROL VERSION.

[] ... HOST CONSOLE COMMAND

A host job entry system console command preceded by the host command prefix character configured for the workstation.

Syntax

```
[]command
```

Parameters

[] The host job entry subsystem command prefix character you have configured for the workstation. Begin every remote console command to the host with this character.

command Any legitimate command recognized by the host remote console command processor. If you do not have NM capability, this command must be allowed by your node manager.

Description

The host command prefix character and a console mode prompt are configured options in NRJE. They are entered in the NRJE Configuration: Workstation Data screen. The configured host command prefix character must be consistent with what is configured on the host system. See the *SNA Link Services Reference Manual* (for MPE V) or the *SNA NRJE Node Manager's Guide* (for MPE XL) for configuration information.

NOTE If you are connected to a VSE/POWER host system, the host expects no command prefix. NRJE does require a prefix with each command, however, to recognize that the string you enter is a VSE/POWER command. NRJE strips off the command prefix before sending it to the VSE/POWER host.

Entering a command string that begins with the host command character enables you to issue a host job entry system command. If your workstation is communicating with the host system and you have access to the console, your command is transmitted. If you have not obtained the console (see the CONSOLE command earlier in this chapter) before issuing a host command, NRJE obtains it for you. While in console mode, the configurable console mode prompt is displayed instead of the standard workstation prompt. To relinquish access to the host remote console, use the RELEASE command

NRJE does not check the syntax of the host console command. Check the IBM documentation that applies to your host installation for the commands available and the syntax you must use. A list of IBM documents is in the preface of this manual.

Example

```
R22>$DA              Display all active jobs for a workstation.
                     Two NRJE subsystem messages follow.

OPENING CONSOLE IN PROGRESS ... (CI=780)
HOST CONSOLE IS OPEN.  (CI=858)


10.01.14 JOB  753 $HASP608 XPRTRUN5 EXECUTING I PRIO  6 IPO1
10.01.14 JOB  754 $HASP608 XPRTRUN6 EXECUTING I PRIO  6 IPO1
```

This example shows that you can issue a host job entry subsystem command without issuing a CONSOLE command.

Text Reference

See CONSOLE earlier in this chapter for the use in context of host console commands. NRJESendCmd is the manager intrinsic counterpart of this command; it is described in Chapter 6 , “NRJE Manager Intrinsics.”

4 Job Output

This chapter details the procedures you follow to use SNA NRJE to receive output from jobs you have submitted to your host system.

The topics discussed are as follows:

- Job output handling, including a number of general considerations that apply to all output received from the host.
- Job management, by which final destination of output can be controlled when a job is submitted for transmission to the host.
- Output management, by which the final destination of output can be controlled through the use of a Lookup Table.
- Handling spooled output.
- Handling unsolicited output.

Output from a job can be received from a host whenever communication is established between an NRJE workstation and a host.

NOTE

If you are connected to a VSE/POWER host system, all output must be requested. That is, a VSE/POWER command, `START LST`*`n,class`* must be issued either from the console (if it is owned) or in your transmitted file before the first VSE/POWER JECL or VSE JCL JOB card. Submitting this command allows output data sets to be transmitted via the VSE/POWER list task, `LST`*`n`*. The list tasks for VSE/POWER control printer output entering the list queue. Similarly, for punch output, a punch task must be started (`START PUN`*`n,class`*) to get the output to be returned to your NRJE workstation. *`class`* is an alphabetic character indicating output class to be handled by the started device. Multiple classes can be specified in a single command. See the *IBM publication VSE/POWER Remote Entry User's Guide* for additional information.

Note that if the command is issued from the console mode, you must precede it with the host console command prefix as configured in the NMMGR Workstation Data screen. If the command is embedded in a submitted job stream, you must precede it with the character string * Δ . . Δ where Δ is a required space.

Job output can be directed to any output device that is supported on your HP 3000. These include laser printers, line printers, and disk and tape files. These devices can be spooled or non-spooled.

Job management is the ability to specify the final destination of the output at the time the job is submitted to NRJE to be transmitted. These procedures are described under “Job Management with the SUBMIT Command” later in this chapter.

Output management is the ability to route output by specifying special forms for output data sets using file equation records in a Lookup Table. These procedures are described under “Output Management with the Lookup Table” later in this chapter.

Job Output Handling

SNA NRJE's control hierarchy for resolving any potentially conflicting output routing specifications is as follows:

1. Use the SUBMIT command specifications PR, PU, or FO, if supplied.
2. If no SUBMIT command routing specifications are supplied, use the Lookup Table.
3. If no specifications are in the SUBMIT command and the Lookup Table is not in use, route the data sets to the default destination specified in the NMMGR Writer Data Screen.
4. When configuring your NRJE workstation, you have the option of configuring a *'formid'* in the Output Destination field of the NMMGR Writer Data screen. (This screen is described in the *SNA NRJE Node Manager's Guide* and the *SNA Link Reference Manual*.) This *'formid'* must be a valid entry in the Lookup Table for the workstation.

   You can configure your system to use this *'formid'* to route output received from the host. On the NMMGR Writer Data screen, set the field identified as "FORCED" as follows:

   - To route all output to *'formid'*, set FORCED=Y
   - To route only standard form output to *'formid'*, set FORCED=N.

NOTE Note that FORCED=Y option is valid only when a *'formid'* is configured in the Output Destination field.

The following table illustrates how NRJE resolves output routing specifications for each data set. This resolution depends on whether the user selects a special form (under the Selected Form in JCL heading) and how the writer is configured (under the Writer Data Screen Specifications heading).

In Table 4-1 FORM "X" represents any JCL-selected form. CLASS "A" represents any valid device class, and LDEV "N" represents any valid ldev. For example, if you routed output to a JCL-specified form (FORM "X"), configured a *'formid'* in the Output Destination field, and specified FORCED=Y, output for both the standard form ('STD') and the JCL-specified form (FORM "X") would be routed to the *'formid'*.

Table 4-1 JES2 Example for Default Output Destination

	Writer Data Screen Specifications			
Selected Form in JCL	‘FORMID’		DEVICE CLASS “A”	LDEV “N”
	FORCED=Y	FORCED=N	FORCED=N	FORCED=N
STANDARD FORM ‘STD’	‘FORMID’	‘FORMID’	“A”	“N”
FORM “X”	‘FORMID’	“X”	“X”	“X”

General Considerations

Regardless of the method you use to control the output returning from the host, a number of general considerations must be taken into account:

- Matching data set output class to device output class.
- Handling carriage control characters.
- Controlling output directed to an HP 268xx laser printer.
- The structure of the Peripheral Device Information Record (PDIR).
- Routing standard forms output.
- Routing special forms output.
- Specifying multiple copies.
- Routing output to disk files.
- Routing output to tape files.

This subsection describes those features.

Host Output Classes and Logical Writers

Output data sets are routed to logical writers configured for your NRJE workstation. The routing is based on a match between the class of the output and the classes assigned to the logical output devices. This match can be determined by combining the following:

- The workstation configuration at the host; specifically, the classes that are defined for each of the logical output devices configured for your remote,

  and
- The JCL in your job, which defines the class of the output data sets,

  and

- The use of host job entry subsystem console commands, if you are an NRJE manager.

Host Configuration

The configuration of logical writers and output classes is described in the *SNA Link Services Reference Manual* or the *SNA NRJE Node Manager's Guide*. Also, see the Preface of this manual for a list of IBM publications.

JCL Class Specification

You associate data sets with output classes by using JCL in your jobs. The specific statements you use depend on the operating system (either MVS or VSE) and the job entry subsystem of the host to which you are connected.

MVS JCL Class Specification If you are connected to an MVS host system (using either JES2 or JES3), associate the output data sets with a specific output class in the following ways:

- The *class* subparameter of the SYSOUT parameter as part of a DD statement; for example,

  ```
  //       DD ... SYSOUT=x
  ```

  where x is a class letter

and/or

- The *MSGCLASS* parameter of the JOB statement; for example,

  ```
  //       JOB ... MSGCLASS=x
  ```

  where x is a class letter

MVS JCL Class Example Use JCL statements as shown here.

```
// DD ... SYSOUT=(A,,P2)
```

The first subparameter of SYSOUT is the class letter A. The third subparameter indicates special form names, in this case P2.

Output would be received on a logical device that is configured on the host to be associated with output class=A, or on a logical device that has been modified by console command to be associated with class=A. The required console commands are described in “Host Job Entry Subsystem Console Commands” later in this chapter.

```
// JOB ... MSGCLASS=D
```

This statement causes JCL message data sets to be sent to a logical output device that is configured or modified by console command to be associated with class=D.

VSE/POWER JECL Class Specification If you are connected to a VSE host system (using VSE/POWER), associate the output data sets with a specific output class in the following way:

- The `CLASS=`*`class`* parameter of either the `* $$ LST` statement or the `* $$ PUN` statement; for example,

  ```
  * $$ LST,CLASS=x
  ```

  where `x` is a class letter.

Host Job Entry Subsystem Console Commands

Each supported host job entry subsystem has interactive methods by which you can use console commands to match the class of the output to the class of a logical output device.

JES2 Commands If you are in the Host Console Facility, you can use the JES2 `$T` (Set) command to relate a class to a logical writer. See the *IBM publication Operator's Library: JES2 Commands*.

Examples

`$T R11.PR1,Q=A`	Job output for class A (`Q=A`) goes to printer 1 (`R11.PR1`).
`$T R11.PR3,Q=A`	If you have done the previous `$T` command, job output for class A goes to PR1 or PR3 (`R11.PR1, R11.PR3`).
`$T R11.PR2,Q=DE`	Job output for classes D and E (`Q=DE`) goes to PR2 (`R11.PR2`).

JES3 Commands If you are connected to a JES3 host system, the console operator command that you use to modify the class of a logical output device is the `*START` (`*S`) command: `*S,`*dev*`,WC=`*`cls,cls,`*`....` See the *IBM publication Operator's Library: OS/VS2 MVS JES3 Commands*.

Example

`*S,PR1,WC=A,P`	This command modifies the device PR1 so that it is associated with output classes A and P.

VSE/POWER Commands If your NRJE workstation is connected to a VSE/POWER host system, you can modify the class of the output to match the class of a logical output device configured on your host as your remote. The command that you use is the ALTER (A) command:

[]A,*queue,jobname*,CLASS=*class*

where [] is the host command prefix that you have configured in the NMMGR Workstation Data screen for your workstation; *queue* is either LST for list queue output, or PUN for punch queue output; *jobname* is the name you used on your VSE/POWER JECL or VSE JCL JOB statement; and *class* is an alphabetic (A–Z) character.

See the IBM publication *VSE/POWER Remote Job Entry User's Guide*.

Example

```
*A,PUN,MYJOB,CLASS=A
```

This command alters the punch output for MYJOB to class A. In this example, the host console command prefix character has been configured to be an asterisk (*).

Summary

You can associate each output data set in a job with a logical writer by using appropriate job control statements in your job stream or, for VSE/POWER, by issuing an []ALTER console command.

Your host system must relate an output class to a logical writer. This occurs when your host is configured, or for JES2 and JES3, when appropriate console commands are issued.

Carriage Control Translation

The host system inserts SNA character strings into printer, console, and punch data. SNA character strings contain carriage control and record separation information.

Carriage control characters are written only to printer devices. For punch devices, carriage control characters are not written. For console output, the carriage control character %201 (print then single space) is written with each record to the $STDLIST device of the console user. For console output, multiple carriage control characters sent in sequence are treated as a single carriage control character by NRJE. The carriage control characters (except No operation) delimit records.

Table 4-2 describes the relationship between the control codes and the operation performed on printer devices by each code.

NOTE HP line printers interpret channel skipping control codes (%300–%313) differently from one another. You can redefine channel skipping control codes using environment files containing VFCs.

See the *SNA Link Services Reference Manual*, the *SNA NRJE Node Manager's Guide*, and the *HP 2608S Line Printer Reference*.

Table 4-2 SNA and HP 3000 Printer Carriage Control Codes

SNA Character String	HP 3000 Control Code and Function
Set Vertical Format (SVF) `! 2B ...`	No operation. Multiple `2B` characters may be sent in the string.
Vertical Channel Skip (VCS) `! 04 ...`	%3*nn* *nn*=the number of `04` characters received contiguously (0<=*nn*<=13). Causes *nn* vertical channel skips.
New Line (NL) `! 15`	%201 Space one line.
Required New Line (RNL) `! 06`	%201 Space one line.
Carriage Return (CR) `! 0D`	%53 No space; return to 1.
Carriage Return/New Line (CR NL) `0D15`	%201 Space one line.
Carriage Return/Form Feed (CR FF `0D0C`	%61 Form feed.
Carriage Return/Vertical Channel Skip (CR VCS) `0D04`	%3*nn* nn=the number of `04` characters received contiguously (0<=*nn*<=13). Causes *nn* vertical channel skips.
Form Feed (FF) `! 0C`	%61 Form feed.
Backspace (BS) `! 16`	No operation.

Table 4-2 SNA and HP 3000 Printer Carriage Control Codes

SNA Character String	HP 3000 Control Code and Function
Inter-record separator (IRS) `! 1E`	%201 Space one line.
Transparent Data (TR) `! 35`	No code; indicates transparent data follows.
All SNA character strings are hexadecimal numbers depicted by *!abcd*.	All HP 3000 control codes are octal numbers depicted by *%abc*.

HP 268xx Laser Printer Output

Output from jobs you submit using NRJE can be printed on an HP 268xx laser printer. You must configure your workstation to include the logical device number (*ldev*) or logical device class (for example, PP) of the HP 268xx in the NMMGR Writer Data screen. Output can be directed to the laser printer by one of these methods:

- Creating file equations in your Lookup Table that specify the use of the laser printer. In this case, you can specify an environment file or you can omit that specification and use the default environment file.
- JCL specifications that assign the output class of your data sets to the same output class being handled by the remote writer that is your laser printer. The link between the logical device (for example, R11.PR3) that is configured to handle the given output data sets, and the physical device (for example, the laser printer) is established in the NMMGR Writer Data screen. Note that the relationship defined by JCL specifications can be overridden by the `SUBMIT` command and Lookup Table specifications as described above.

For example,

- The *writername* of your laser printer is R11.PR3, as configured under NRJE and at the host.
- By the host configuration, R11.PR3 is assigned to handle output class=C.
- In your JCL you assign class C to the output data sets. These data sets are then directed by default to the laser printer.

Data sets can be printed using environment files you specify. You can specify an environment file by any of these methods:

- In your JCL, by routing the output data sets to a special form that corresponds to a Lookup Table entry specifying an environment file.
- In the NMMGR Writer Data screen configuration for a logical writer.

- In a file equation that you backreference directly as a SUBMIT command output destination.
- In a Lookup Table file equation that you reference in a SUBMIT command by its form name.

If you do not specify an environment file by one of these methods, the default environment file configured for the HP 268xx is used.

The recommended procedure is to use the Lookup Table and map form names to environment files that can be used when output is routed to the laser printer. This process is described under “Output Management with the Lookup Table” later in this chapter. In this case, FCB records returned from the host are ignored.

It is possible to name an environment file in an FCB, although this procedure is more difficult than the procedures listed above. If you do not use the Lookup Table, the FCB information is used as an environment file name that is appended with the accounting information HPENV.SYS. An example of an MVS JCL DD card with an environment file is:

```
      //    DD.... FCB=KIM
```

Environment file KIM.HPENV.SYS is used for this data set, unless the FORCED option is Y for your workstation logical writer configuration.

When your output is received from the host system, NRJE uses the environment file named in the FCB and residing in HPENV.SYS to produce your output on an HP 268xx. The environment file is always copied to your spool file.

Peripheral Device Information Record

Whether you are controlling the output received by NRJE by specifications made in the SUBMIT command, through the use of the Lookup Table, or by allowing the MPE Spooler to handle the output, much of the information used by NRJE to perform job and output management is transferred in a header data structure called the Peripheral Device Information Record (**PDIR**). SNA NRJE can be configured to log PDIR information. The contents of a PDIR are described in Table 4-3.

NOTE NRJE requires that the IBM host job entry subsystem (JES2, JES3, or VSE/POWER) send a PDIR after selecting the output destination with an FMH1 BDS (Begin Destination Selection) before sending job output. NRJE will reject output not preceded by a PDIR.

Table 4-3 Peripheral Device Information Record Structure

Byte	Contents and Meaning
0	The length of this record, including this byte
1	Function Management Header (FMH) concatenation. An FMH is SNA-specific information which may be present to perform the following functions: • Select a destination and control the way data is handled there. • Change the characteristics of data or its destination. • Transmit status information about a destination, such as for a program or a device Bit 0 indicates the following: • 0 = No FMH follows. • 1 = Another FMH follows. Bits 1–7 are set to binary 0000010 to indicate an FMH-2 identifier.
2	Indicates that this record is a PDIR. Its value is !01.
3	A PDIR request identifier. Values are as follows: • 0 = Standard PDIR. • 1 = Job separator PDIR. • 2 = System message PDIR. All other values are undefined
	NOTE: Bytes 4–99 are transmitted by the host in EBCDIC. NRJE translates each of the fields in this range to ASCII. If NRJE is configured to log the PDIRs, those fields are logged in ASCII.
4–11	Data set creation date, in ASCII characters. The format is
	MM/DD/YY
	The default value is 00/00/00.
12–19	Data set creation time, in ASCII characters. The format is
	The default is 00.00.00
	HH.MM.SS
20–27	Form name as specified in the job JCL, in ASCII, left-justified, with trailing blanks. JES2 typically uses STD, JES3 typically uses 1PRT, and VSE/POWER uses (eight blank characters) to indicate standard form. Special form names are transferred in this field.
28–35	Forms Control Buffer (FCB), in ASCII, left-justified, with trailing blanks. This field is ignored by NRJE if your workstation is configured to route output using the Lookup Table. If the Lookup Table is not in use, the FCB is used to identify an HP 2680A environment file name.

Table 4-3 Peripheral Device Information Record Structure

Byte	Contents and Meaning
	The default FCB value is all blank characters.
36–43	Printer train name to be mounted, in ASCII, left-justified, with trailing blank characters. JES2 uses `0` as a default train.
44–51	Number of *additional* copies of the data set to be reproduced, in ASCII characters, right-justified, with leading blank characters. The default value is `0`.
52–59	The volume of output, in ASCII characters, right-justified, with leading blank characters. This specifies the *approximate* number of printer lines, or the *approximate* number of cards in the data set. The default value is all blank characters.
	The volume of output can be used for scheduling and estimating processing time.
60–67	Jobname field, in ASCII, left-justified, with trailing blank characters. A jobname can be entered in a JCL JOB card image in columns 3–10, to identify a batch job input stream submitted to a host system. The default value is all blank characters
	NOTE: If you are connected to a VSE host system, bytes 60–99 are used as a Jobname field, left-justified, with trailing blank characters. The following fields are defined for an MVS host system.
68–75	Step name field, in ASCII, left-justified, with trailing blank characters. A step name can be entered in a JCL EXEC card image in columns 3–10, to identify an action in a batch job input stream submitted to a host system. The default value is all blank characters
76–83	Procedure step name field, in ASCII, left-justified, with trailing blank characters. A procedure step is a unit of work associated with one processing program and its related data, within a cataloged or in-stream procedure, as in columns 3–10 in a JCL PROC statement. The default value is all blank characters.
84–91	Data definition (DD) name field, in ASCII, left justified, with trailing blank characters. A DD name field can be entered in a JCL DD card image in columns 3–10, to identify the name of a data set within a job stream. The default value is all blank characters.
92–99	Spin number, in ASCII, left-justified, with trailing blank characters. The spin number is a unique identifier to distinguish among data sets with the same jobname (bytes 60–67). The default value is all blank characters.
	NOTE: The spin number is used by JES3 to transfer the host job number to the workstation.

Routing to the Standard Form

Standard form output refers to any output data sets for which you choose the host default form and do not specify an alternate form, using job control statements. SNA NRJE must be configured for the name of the standard form in the NMMGR Workstation Data Page 2 screen. This configured value must agree with the value configured on the host. The standard form identifier is usually `STD` for JES2, `1PRT` for JES3, and a blank string for VSE/POWER.

Unless you have specified PR=*'formid'* or PU=*'formid'* options in your SUBMIT command (not available if NRJE is connected to a VSE/POWER host system), output routed to the standard form is not controlled by the Lookup Table even if it is in use. For all host job entry subsystems, when using the Lookup Table, if a data set is to be routed to the standard or host default form, the output file characteristics defined in the NMMGR Writer Data screen are used to open the destination file. These characteristics include logical device number or name of a spooled device and output priority. So, you do not have to put an entry for the standard form in the table. *Any entry in the Lookup Table under the standard form name is ignored for standard form output.*

PR= and PU= options to the SUBMIT command are used to control standard forms output in printer and punch streams, respectively. If you have used PR=*'formid'* or PU=*'formid'* options in a SUBMIT command, the standard forms output is routed according to the file equation attributes for the entry *formid* in the Lookup Table.

Data sets routed to forms other than the standard form are referred to as **special forms output**.

Special Forms Processing

For SNA NRJE, a special form is any form name other than the default or standard form name configured for your host system. Special forms are supported for both print and punch output. NRJE processes special forms through one of the following:

- The SUBMIT command FO= parameter specifications, or
- The Lookup Table, which maps the form name to an output destination.

If you are not using job management or output management, NRJE processes special forms requests as follows:

1. Whenever special forms output destined for printers is received from the host system and becomes active in the MPE Spooler, the HP 3000 console operator is prompted to mount the required special form on the printer.
2. After the form has been mounted, MPE writes the output file to the printer.

Remember, if you submit several jobs to a host with one SUBMIT command, any alternate output destinations specified in the *PRINT* or *PUNCH* parameters (for standard forms output) or *FORMS* parameters (for special forms output) override JCL specifications. The SUBMIT command is explained in Chapter 3 , “User Commands.”

Specifying Multiple Copies

You can direct NRJE to print multiple copies of a data set when output is routed to an NRJE print device. You select multiple copies by using the *COPIES* parameter in your JCL. For example, for MVS JCL,

```
//ddname DD SYSOUT=C,COPIES=27
```

To select multiple copies of a single data set, you should specify a unique class for each data set. To generate multiple data sets with multiple copies, you should select a separate class for each data set or generate the data sets in separate jobs. Also, each workstation writer should be configured to receive a unique class of output.

NOTE The *COPIES* parameter is not supported on punch output.

Routing Output to Disk Files

When writing to a fixed-width disk file, MPE pads with ASCII blanks if the destination file is ASCII and with binary zeros if the destination file is binary. No padding is done for variable- or undefined-width files.

For print output, the Carriage Control character is the first byte of each record. No carriage control information is in punch output.

Routing Output to Tape Files

When writing to a tape device, variable record widths should be specified in the file equation. Note that the default record width used by NRJE is 256 bytes. This value can be overridden in a file equation if desired. For print output, include the `cctl` designation in the file equation.

Examples

`file tape;dev=tape;rec=-132,,v,ascii;cctl` (for print output)

or

`file tape;dev=tape;rec=128,,v,binary;nocctl` (for punch output)

then, issue this command:

```
infile;FO=*tape
```

Use of the SUBMIT command is described in Chapter 3 , “User Commands,” and under “Job Management with the SUBMIT Command” later in this chapter. Alternately, similar file equation records could be referenced in the output management Lookup Table. Use of the Lookup Table is described in “Output Management with the Lookup Table” later in this chapter.

Job and Output Management

Job management is the ability to specify the final destination of the output at the time the job is submitted to NRJE to be transmitted to the host. NRJE accomplishes this through parameters included in the SUBMIT command which can directly or indirectly identify a final destination.

Output management is the ability to route output by specifying special forms for output data sets using file equation records in a lookup table. The Lookup Table is configured as part of SNA NRJE and keyed into through special form names specified in the job. The Lookup Table maps these special form names to file equations defining final destinations for the output.

The flow of data through these procedures is described next.

Job Output Data Flow Description

You submit a job within the SNA NRJE subsystem by using the SUBMIT command. As part of that command you may optionally specify PRINT, PUNCH, or FORMS output designations. PRINT and PUNCH designations provide control of output that is routed from the host to printer or punch devices, using the default or standard form name. These designations override JCL specifications for standard output.

FORMS designations provide control of output that is routed from the host to printer or punch devices, using any form name other than the default.

This SUBMIT command information is placed into a specially formatted Job Log record and is written to the NRJE reader along with your file. These parameters are defined in the SUBMIT command discussion in Chapter 3 , “User Commands.”.

When your workstation is active, the NRJE Logical Unit scans the NRJE reader for ready spool files. When the LU finds one, it copies the spool file to the host and waits for information containing the job’s unique job number from the host. With JES2 and JES3 host job entry subsystems, the unique job number is supplied as part of a “Job Received” message that is returned when the host receives a job. The job number, along with other information in the formatted Job Log record, is written to a Job Log file that you configure for your NRJE workstation.

When an output data set is received from a JES2 host, the NRJE LU attempts to obtain the job number by scanning the output banner. For JES3, the job number is returned in the PDIR. When NRJE finds the job number, it checks the Job Log for routing information for the job, and updates a count of the number of data sets received for the job. If necessary, NRJE also checks the Lookup Table configured for NRJE for routing information. The special form names used to access the information from the Lookup Table are specified in the job stream or as part of the SUBMIT command. The special form name is sent back to NRJE in a PDIR generated by the host.

Job Management with the SUBMIT Command

Details for use of the SUBMIT command are in Chapter 3 , "User Commands."

For job management to work, the NRJE subsystem must be able to obtain host job numbers from both "Job Received" messages and host output banner pages. For the JES2 host job entry subsystem, if standard messages and banners are generated, internal parsing algorithms (available with SNA NRJE) are used to obtain this information. For JES3, a parsing algorithm is used to obtain the job number from the "Job Received" message, but the output banner need not be scanned. The job number is transferred to NRJE from the host in the spin number field of the PDIR.

User exits are provided if you need to generate parsing programs to handle non-standard information coming from your host system. These exits and the algorithms for the NRJE-supplied parsing routines are described in Appendix A , "Parsing Algoithms and User Exit Procedures."

NOTE If you are connected to a VSE/POWER host, the job number that is generated cannot be used to track the output data sets. As a result, use of the Job Log is not in effect for NRJE connected to VSE/POWER. Job management with the SUBMIT command is therefore not possible. The SUBMIT command is still used to make jobs available for transmission to the host.

With the SUBMIT command you can direct standard output with PR= or PU=, and special forms output with FO= options. In each field you can specify the following:

- A formal file designator (**filename*).
- A logical device class in double quotes (*"DEVICE"*).
- A logical device number (*ldev*), which is a numeric entry.
- An actual file designator (alphanumeric, first character is alphabetic).
- An alternate *formid* listed in the Lookup Table, surrounded by single quotes (*'formid'*).

The routing information you specify is stored in the workstation Job Log. The Job Log consists of a KSAM data file and key file. The default names for these files are as follows:

`NRJEJOB` KSAM data file containing a maximum of 10,000 records, with information about each job submitted.

`NRJEKEY` KSAM key file used to access the data file. The elements are the job numbers for JES2 and JES3 that are returned from the host. Alternate keys are the *`UserName.AcctName`* of the user that submitted the job, and the host job name.

These names can be configured in the NMMGR Workstation Data Page 2 screen as described in the *SNA Link Services Reference Manual* or the *SNA NRJE Node Manager's Guide*. If multiple NRJE workstations are configured on your HP 3000, each workstation must be directed to access separate job and key files.

If you specified no alternate output destination parameters in the `SUBMIT` command, then output routing will depend on whether the optional output management Lookup Table is in use.

Output Management with the Lookup Table

This section describes an output routing feature of SNA NRJE that enables you to designate a final destination of an output data set by directing output to a special form name (*`formid`*). The special form name you specify is used to key into a Lookup Table which you generate. The Lookup Table contains file equation attributes to identify the output file.

Use of the Lookup Table is an optional feature of SNA NRJE. Note that output routed to the standard form is not controlled by the Lookup Table even if it is in use. Any entry in the Lookup Table under the standard form name is ignored for standard form output.

Description

To route output using special forms identifiers, you must specify special forms in your output data sets by using the appropriate job control language. A *`formid`* is any form name that identifies an entry in the Lookup Table. For example, if your NRJE installation is connected to a JES2 host, you could use the following:

`/*OUTPUT EPOC F=PYCK` (JCL JES2 control statement)

with JCL DD statements like:

`// DD SYSOUT (A,,EPOC)` (JCL data definition statement referring to a previous control statement)

or

`// DD SYSOUT (A,,PYCK)` (JCL data definition statement)

With either of the above DD statements, the applicable data set will be routed to special form PYCK. The form name is placed into the PDIR by the host system. When it receives the PDIR for the data set, NRJE will use the name of the special form PYCK to key into the Lookup Table. This table must be built by the NRJE manager and must contain all special *formids* supported at your remote, along with file equation attributes for each *formid*. SNA NRJE will open the output file based on the file equation attributes found in the table. An example Lookup Table entry is shown below under "Example Lookup Table Entry." Note that a sample Lookup Table file is provided in the SAMPTABL sample file in NRJE.SYS.

Creating the Lookup Table

The Lookup Table is a file created in the SYS account, and it contains the *formid* names and associated file equation attributes. To enable the Lookup Table feature, the NRJE manager must perform the following tasks:

1. Build the Lookup Table in the SYS account using an editor such as EDIT/3000 or TDP/3000.
2. Specify the Lookup Table file name in the field of the NMMGR Workstation Data Page 2 screen. (The default value is NRJETABL.NRJE.SYS.)

Format of Lookup Table Entries

The format you must use for the entries when creating or adding to the Lookup Table with the editor is as follows:

Columns 1–8: *formid* names. Any combination of alphanumeric or IBM National Characters (#, $, or @), left-justified, padded with blanks if necessary. *formid* CMD is reserved.

NOTE If you have a '*formid*' in the Output Destination field of the writer screens, the Lookup Table entry can be only 6 characters. Although the screen field length is 8 characters, 2 of those characters are apostrophes (‘ ’).

Columns 9–EOR: File equation attributes as allowed by the MPE FILE command. See *MPE Commands Reference Manual* for details on the FILE command.

(EOR = End of Record; upper limit is 260 characters.)

In addition, the following rules apply to the Lookup Table entries:

- The Lookup Table can be of any record width supported by the editor, to a limit of 260 bytes. 80 bytes are recommended to facilitate viewing the file on a terminal screen.

- Continuation records are allowed, using the ampersand (&) as the last non-blank character.
- Concatenated records cannot exceed 260 non-blank characters.
- The file can be kept either numbered or unnumbered.
- No distinction is made between upper and lower case characters.
- An entry for each *formid* to be supported by the workstation must appear in the table. Each entry is limited to 6 characters if configured in the writer screen.
- NRJE performs a linear search when accessing the table. If duplicate *formids* reside in the table, only the first one is used.

Example Lookup Table Entry

```
Column:

1            9

.

.

.

form?        ;dev=epoc;env=lp602.hpenv.sys
```

NRJE will access the above entry whenever an output data set is to be printed on the form `form7`. That output will be routed to device `EPOC` using environment file `lp602.hpenv.sys`, unless you indicate different routing specifications with the `SUBMIT` command routing options.

Using the Lookup Table

At startup, SNA NRJE references the Lookup Table identified in the `Formid Lookup Table File` field in the NMMGR Workstation Data Page 2 screen. If this field is blank, NRJE will operate without a Lookup Table. If NRJE finds a Lookup Table configured in this field, but the file does not exist, NRJE terminates and the following message is displayed:

```
NRJE UNABLE TO OPEN LOOKUP TABLE.  (MON=544)
```

Operation With the Lookup Table

Each workstation copies the Lookup Table into a temporary internal file. A linear search through the temporary file is performed whenever a data set requesting a special form is received. If a match is found, NRJE uses the corresponding file equation record to route the data set. If the *formid* is not found in the temporary file, NRJE opens the permanent file and checks to determine whether it has increased in size since the last access. If it has increased in size, NRJE will reload the internal temporary file and search again for the *formid* before going into error recovery.

In error recovery, NRJE issues an error message indicating the failure to locate the *formid* specified. If this *formid* cannot be found in the Lookup Table, NRJE will attempt to route output to the default destination with a priority of 2. This default destination is specified in the Output Destination field of the NMMGR Writer Data screen. See the *SNA Link Services Reference Manual* or the *SNA NRJE Node Manager's Guide* for the configuration procedures.

When using the Lookup Table, NRJE uses the default file system options for printer and punch streams as described in Table 3-3 and the punch output differences following it. The output priority used is that configured for the applicable writer in the NMMGR Writer Data screen.

NOTE MPE has established a hierarchy to resolve potentially conflicting file system options:

File equation attributes always override FOPEN specifications.

Lookup Table Examples

Each entry in the Lookup Table is identified by a *formid* in positions 1–8. The parameters following a *formid* function like a FILE command from MPE. Example Lookup Table entries are shown below in shaded text, with explanations after each.

```
I2      ;dev=EPOC,4,2;env=lplong.HPENV
```

This Lookup Table entry causes all output for form I2 to be routed to device class EPOC. The output priority is 4. Two copies are to be printed. The environment file is lplong.hpenv.sys. Note that no distinction is made between upper and lower case characters.

```
5927    ;DEV=LPS,,5;ENV=LP60LONG.HPENV
```

This causes all output for form 5927 to be routed to output device class LPS. The environment file is lp60long.hpenv.sys. Five copies are requested. The output priority used is the value configured in the NMMGR Writer Data screen for the writer receiving the output.

```
payroll ;DEV=EPOC,5,3;ENV=CHECKENV.HPENV;forms=checks.
```

This entry routes output for the special form PAYROLL to device class EPOC. The output priority is 5. Three copies are requested. The environment file is checkenv.hpenv.sys. When the spool file becomes active on the HP 3000, a forms mount message for the special form checks. is displayed on the system console. The period (.) in the *forms* parameter is required to designate a form.

NOTE The *forms* parameter should always be placed after the *env* parameter.

The *formid* PAYROLL has more than four characters. Some host systems support four characters or less for a special form name in their job control language. However, a user can direct forms output to this entry by using the SUBMIT command with solicited output:

```
SUBMIT filename;FORMS='PAYROLL'
```

```
disc =mydata.nrje,new
```

Output is routed to a disk file with the above entry. Output destined for the *formid* disc is sent to the disk file mydata.nrje.sys. The new parameter overrides the default value of old. NRJE builds the file for you.

CAUTION If the file already exists as a permanent file, NRJE cannot close the file successfully. In such cases the file is renamed as NRJEnnnn, where nnnn is a 24 hour time stamp. If the file cannot be renamed, the output is lost.

Files cannot be built across accounts. Use the *new* parameter to build files in the SYS account only.

```
disc2   =discS.nrje.myacct
```

This entry causes all output for the *formid* disc2 to be routed to disk file discs.nrje.myacct. This file must be built in advance. NRJE assumes that the file already exists unless the *new* parameter is specified. A new option could not be used here because NRJE cannot build files across accounts.

```
$@@long#;dev=pp;env=lp604.hpenv
```

A *formid* can begin with numeric or IBM national characters (as well as alphabetic characters) and can be up to eight characters long. Output routed to *formid* $@@long# is sent to device class PP using environment file lp604.hpenv.sys.

```
test    =special;dev=pp,3;forms=special.
```

This entry causes the output file to be named SPECIAL, rather than the host jobname. The device class is PP. The specified output priority is 3. The period (.) in the *forms* parameter is required to designate a form. The form name in this example matches the file name assigned to the spool file. This enables HP 3000 operators to more readily identify the names of special forms required by spool files, since the SHOWOUT output indicates the name of the form required.

```
frm1    =newfile ,new
```

This causes NRJE to build a new file in the NRJE logon group and account (NRJE.SYS) using the host jobname as the filename. This entry provides a mechanism for routing either solicited or unsolicited output to disk, and building a new file using the name of the job as the filename.

CAUTION

You can do this only in NRJE.SYS. File equations of this type apply only to the logon group and account of the NRJE monitor stream job.

Always use a unique host jobname with this feature. Use of non-unique host jobnames may cause data to be lost in an attempt to create duplicate permanent files.

System Security Considerations

If the Lookup Table is used to route output to disk files, the node manager should ensure that users are not able to direct output to files for which they do not have legal access. Note that output is routed by NRJE running as a batch job under *UserName.SYS*. By obtaining the form name of a record in the Lookup Table which directs output to a disk file, a user can direct his output to this file, provided *UserName.SYS* has write access to the file (regardless of whether the user has access). To minimize the potential for abuse of system security, the node manager should use the MPE ALTSEC or ALTGROUP commands to ensure that only authorized users have read access to the Lookup Table.

Starting Jobs on the HP 3000 (Reverse NRJE)

The special *formid* CMD is reserved by NRJE. It is used to cause NRJE to stream a received data set rather than route the data set using file equation attributes. To cause job output to be streamed, you must do the following:

1. Route the output to a PUNCH device.
2. Prefix all MPE commands with an exclamation mark (!). The first data record must be a valid MPE JOB card.

It is not necessary to add an entry for CMD in the Lookup Table to use the reserved form. Also, the reserved form name CMD is used by all logical punches to stream output, regardless of whether the Lookup Table is in use.

Uses For Reverse NRJE

The following is a list of suggested uses of the Reverse NRJE feature:

- Scheduling HP 3000 jobs from the host.
- Querying the status of the HP 3000 from the host.
- Submitting jobs to remote or local HP 3000s to access data and/or software available on those machines.
- Gaining access to HP 3000 communications facilities (NS/3000, X.25, etc.).

- Gaining access to HP 3000 peripherals (HP 268xx laser printer, reading STORE tapes, etc.).
- Uploading and downloading files. (Due to limitations of the host subsystem, deblocking may be necessary.)
- Performing postprocessing of data on the HP 3000 before printing.
- Deferring printing.
- Providing a method for routing unsolicited output to disk files without compromising system security. See “System Security Considerations” earlier in this chapter.

  With Reverse NRJE, you do not have to route to a specially created *formid* in the Lookup Table. The output need only be routed to CMD on a punch device job stream. You do need to supply the appropriate logon passwords in your job.
- Performing miscellaneous system management functions without having to physically go to the NRJE site. You could run FREE5, LISTLOG5, issue a SHOWJOB command and send the output back to the host, or start an HP 3000 system backup, etc.

Reverse NRJE Example

Below is an example of the use of Reverse NRJE which informs the HP 3000 operator that the host is about to disconnect.

```
//WARNALL JOB ,FORELL,MSGCLASS=A
/*ROUTE  PUNCH RMT1
/*ROUTE  PRINT RMT1
// EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN    DD DUMMY
//SYSUT2   DD SYSOUT=(B,,CMD)
//SYSUT1   DD *
!JOB IBMWARN,MANAGER.SYS;HIPRI;OUTCLASS=LP,1
!TELLOP ; #### IBM IS GOING DOWN IN 10 MINUTES
!TELLOP ; #### SIGNOFF MRJE AND NRJE NOW
!EOJ
/*
//
```

Unsolicited Output

Unsolicited output is defined as data sets received from a host or another remote that are not recognized by SNA NRJE as coming from jobs that were submitted on your workstation.

Causes of Unsolicited Output

Output can be considered unsolicited because of any of these circumstances:

- The output was generated by a user on the host or another remote, not by a job submitted on the workstation.
- NRJE was not able to interpret the print or punch banners because they were not standard, the standard format changed, or the host did not send banners with the output.
- NRJE was not able to read the host “Job Received” message because the message was not standard, the standard format changed, or the host does not send “Job Received” messages upon receipt of jobs. (VSE/POWER does not send “Job Received” messages.)
- You have suppressed printing or punching of banners. This took place in your JCL, or by means of a host console command, or the host may not be configured to send banners.
- There was a significant lag between the time the host received a job and the time the host returned output to NRJE. During this lag the contents of the Job Log, including output routing information for your job, were purged by the NRJE manager with the NRJE `PURGE` command.

Controlling Unsolicited Output

If banners are off, turn them on (for example, `JES2 $T PR1,S=Y`). If the host is waiting to transmit output to NRJE and you know that the output will be treated as unsolicited, you still can direct it to a specific destination by any of these methods:

- **Output management**

  Make sure the Lookup Table file name is configured for your workstation and that *`formids`* that correspond to entries in the Lookup Table are specified in your JCL.

- **SPOOK or MPE spooler**

  Raise the Outfence value for the MPE spooler so that it is above the priority value of your output data sets (defer the output).

Make sure in the job JCL that the output is directed to a spooled output device. See “Host Output Classes and Logical Writers” earlier in this chapter.

Use COPY, FCOPY or (for MPE V and MPE XL prior to release 2.1) the MPE utility SPOOK to copy the output data set to another disk file, or output the data set to tape or to an output device.

- **Banner decode exit procedure**

  You can configure your workstation to use an exit procedure that you supply to parse the host banner page. As part of this procedure you can execute an FOPEN and redirect the output data sets. See Appendix A , “Parsing Algoithms and User Exit Procedures,” for details.

- **“Job Received” message exit procedure**

  You can configure your workstation to use an exit procedure that you supply to parse “Job Received” messages, if your host sends them. (VSE/POWER does not.) See Appendix A , “Parsing Algoithms and User Exit Procedures,” for details.

5 User Intrinsics

This chapter describes NRJE user intrinsics. Some of these intrinsics have NRJE manager extensions. You must be configured with NM capability to use the extensions to the intrinsics. Any MPE user can call these intrinsics without the manager extensions.

Introduction

All of these intrinsics are related to NRJE commands described in Chapter 3 , “User Commands.” The manager extensions to the commands are also indicated in this chapter.

NRJE consists of a subsystem command interface and a set of user-callable intrinsics. An NRJE intrinsic, or a set of intrinsics, corresponds with each subsystem command that affects job stream input or output.

NRJE intrinsics can be called from:

- SPL
- PASCAL
- FORTRAN
- COBOL II

If you omit any parameters, you must retain the commas that follow them. No variable length calling sequences exist.

Byte array parameters with the left-most character blank indicate that a default value will be used.

Split-stack mode intrinsic calls are prohibited.

See Table 5-1, “NRJE User Intrinsics” for a summary. Some user intrinsics have capabilities that can be used by NRJE managers only.

NRJE intrinsics that can be used only by NRJE managers are described in Chapter 6 , “NRJE Manager Intrinsics.”

Table 5-1 NRJE User Intrinsics

Name	Description
NRJEActiveWS	Provides the number and identification of active NRJE workstations.
NRJEAlter *	Revises the priority of a submitted job or jobs.
NRJECancel *	Cancels previously submitted jobs not yet transmitted.
NRJEErrMsg	Obtains and writes an NRJE error message string.
NRJEHJCancel *	Cancels jobs already transmitted to the host, by marking the Job Log to dispose of returning data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.
NRJEHJInfo *	Obtains a list of jobs transmitted to the host, the number of data sets received against those jobs, and the special destinations of those data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.
NRJEJobInfo *	Obtains a list of jobs submitted but not transmitted to the host, and job spool file characteristics.
NRJELUList	Provides a list of all LUs and related characteristics for a workstation.
NRJEQueue	Obtains information about a workstation transmission queue (the spooled reader).
NRJESubmit *	Prepares a batch input job stream for transmission to a host system.
NRJE2Submit *	Creates a job input stream on the transmission queue to send to a host system, and allows the user to specify output destinations for data sets returning from the job submitted to the host. If you are connected to a VSE/POWER host system, the output routing capabilities are not available.
NRJEWelcome	Obtains the contents of the NRJE welcome message.
NRJEWriter	Obtains information about the logical writers configured for a workstation.
NRJEWSInfo	Obtains general information about a workstation.
* This intrinsic has NRJE manager extensions.	

Parameters Common to NRJE Intrinsics

Some parameters are common to most NRJE intrinsics; each parameter is described below.

Workstation Identification

The workstation identification (*Wsid*) parameter is common to all NRJE intrinsics.

It is an eight-character input byte array.

It must always begin with a letter followed by alphanumeric characters and be left-justified with trailing blank characters. If the workstation identifier is less than eight characters long, it must be followed by blank characters.

Intrinsic Result

A *Result* parameter is common to all intrinsics. It is an eight-element integer array. When your intrinsic call is successful, the first element of the *Result* array is set to zero.

See Table 5-2. and also see “Result Codes and Messages,” in the *Installing and Troubleshooting SNA NRJE* manual or the *SNA NRJE Node Manager's Guide*.

The elements of the Result array are as follows:

Table 5-2 Result Array Structure

COBOL II FORTRAN, and PASCAL Subscripting:	SPL Subscripting:	Contents
RESULT(1)	RESULT(0)	Intrinsic error code.
RESULT(2)	RESULT(1)	File System error code.
RESULT(3)	RESULT(2)	NRJE management error code.
RESULT(4)	RESULT(3)	NRJE configuration error code.
RESULT(5)	RESULT(4)	NRJE spooling error code.
RESULT(6)	RESULT(5)	Parameter 1 associated with Intrinsic Error Code.
RESULT(7)	RESULT(6)	Parameter 2 associated with Intrinsic Error Code.
RESULT(8)	RESULT(7)	Parameter 3 associated with Intrinsic Error Code.

Stack Size

The NRJE user and manager intrinsics require 8500 words for local variables and working space. If your program requires 9000 words for its data stack, you should prepare your program file with a *Segsize* value of 17500 words. Your PREP command would look like this:

```
:PREP USLFILE,PROGFILE;MAXDATA=17500
```

You also can specify *Segsize* in the PREPRUN and RUN commands. You cannot specify *Segsize* with any compile-and-prepare or compile-and-go commands such as SPLPREP or SPLGO.

Intrinsic Data Types

These data types are used in intrinsic descriptions:

- BA Byte array
- D Double-word integer variable
- DA Double-word integer array
- I Integer variable (single word)
- IA Integer array
- L Logical variable (single word)
- LA Logical array
- V Value

NOTE Even numbers in logical variables and elements of logical arrays are FALSE.

Odd numbers in logical variables and elements of logical arrays are TRUE.

NRJEActiveWS

Provides the number and identification of active NRJE workstations.

Syntax

```
                 I             BA            I       IA
NRJEActiveWS ( InCount, WorkStations, OutCount, Result )
```

Parameters

InCount **An integer input variable by reference**. The *InCount* parameter specifies the maximum number of entries that can be returned in the *WorkStations* array, from 1 through 8.

If *InCount* is less than the actual number of active *WorkStations*, an error code will be returned in *Result*.

WorkStations **An output byte array**. The *WorkStations* array is a list of active NRJE workstations. Each entry is eight characters long, left-justified, and blank filled when necessary. A three-entry list will look like this:

```
STA1
RICHARD
RMT99
```

OutCount **An output integer variable by reference**. The *OutCount* parameter is the number of active workstations and entries in *WorkStations*. Using the example for *WorkStations* above, *OutCount*=3.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

The NRJEActiveWS intrinsic provides a list of active workstations. The maximum list length is entered in the *InCount* parameter. The list is in the *Workstations* array. The number of active workstations is returned in the *OutCount* parameter. The *Result* array contains any intrinsic execution error codes.

Text Reference

This intrinsic is used to provide information for the DISPLAY command. See Chapter 3 , “User Commands.”

NRJEAlter

Revises the priority of a submitted job or jobs.

Syntax

```
               BA    BA     I      I      IA
NRJEAlter ( Wsid, Name, NameLen, NewPri, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

Name **An input byte array**. The *Name* parameter identifies the job or set of jobs whose priority will be revised. The length of *Name* depends on the way it is used (see the *NameLen* parameter).

Identify the job or job set one of these ways:

- By job name. Begin the *Name* string with a letter. Use up to eight alphanumeric characters.

  All jobs in your logon account. Use one @ character in *Name* to indicate that all jobs in your logon account will be revised.

- For a specific job stream file. Use a spool file identification in the form *nnnnn*, where *n* is a digit.

- For all files of a user. Follow MPE standards. Use the form *UserName.AcctName* with up to seventeen alphanumeric characters. Separate *UserName* from *AcctName* with a period (.).

NOTE If you are referring to a job that is different from your logon, you must have NM user capability.

See Table 5-3 in the NRJEJobInfo intrinsic description.

NameLen **An input integer variable.** The *NameLen* parameter is the number of characters entered in the *Name* parameter, described above.

NewPri **An input integer variable.** All jobs that are qualified by *Name* have their transmission priority revised to *NewPri*, which is from 0 through 14. The highest priority is 14.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use NRJEAlter to revise the priority of jobs submitted from a *Wsid* but not transmitted to the host. Specify the job set in *Name* and the desired priority in *NewPri*. The *NameLen* parameter is the length of the string in *Name*. The consequences of NRJEAlter execution are returned in *Result*.

Text Reference

See the ALTER command in Chapter 3 , “User Commands.”

NRJECancel

Cancels previously submitted jobs not yet transmitted.

Syntax

```
              BA     BA       I        IA
NRJECancel ( Wsid, Name, NameLen, Result )
```

Parameters

Wsid — **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

Name — **An input byte array**. The *Name* array identifies the job stream file to be cancelled.

Identify the job or job set one of these ways:

- By job name. Begin the *Name* string with a letter. Use up to eight alphanumeric characters.
- All jobs in your logon account. Use one @ character in *Name* to indicate that all jobs in your logon account will be revised. If you have NM capability, all jobs will be affected.
- For a specific job stream file. Use a spool file identification in the form *nnnnn*, where *n* is a digit.
- For all files of a user. Follow MPE standards. Use the form *UserName.AcctName* with up to seventeen alphanumeric characters. Separate *UserName* from *AcctName* with a period (.).

NOTE If you are referring to a job that is different from your logon, you must have NM user capability.

See Table 5-3 in the NRJEJobInfo instrinsic description.

NameLen — **An input integer variable**. The *NameLen* parameter contains the number of characters in *Name*.

Result — **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *`Result`* array is set to zero if no errors took place. The structure of the *`Result`* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *`Result`* immediately after you call this intrinsic. If the first element of *`Result`* is not zero, test the other elements of *`Result`* to determine the nature of the problem that has occurred.

Description

Use `NRJECancel` to remove job stream spool files from the transmission queue. Specify a spool file set one of these ways:

- Individually, by spool file identifier, or by job name, if it is unique.
- Collectively, by job name (if it is generic), for all files of the logon user, or for all files of another user (if you have NM capability).

After `NRJECancel` has returned control to your program, test the first element of *`Result`*.

Text Reference

See `CANCEL` in Chapter 3 , “User Commands.”

NRJEErrMsg

Obtains and writes an NRJE error message string.

Syntax

```
                IA      I       BA       I       I
NRJEErrMsg ( Error, FileNum, Message, OutLen, Result )
```

Parameters

Error **An input integer array**. The *Error* array is the result of another NRJE intrinsic call for which ASCII text is desired.

The structure of the *Error* array is the same as the *Result* array. See “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

FileNum **An input integer variable**. The *FileNum* parameter identifies a file to which you want the contents of the *Message* parameter written.

Obtain the value for the *FileNum* parameter by calling the FOPEN intrinsic.

If the value of *FileNum* is zero, no output takes place; however, information is returned in the *Message* parameter.

Message **An output byte array**. The *Message* array contains one or more messages associated with the *Error* array.

The *Message* array must be 370 bytes long. Multiple line error messages are separated by ASCII carriage return and line feed characters.

Messages are listed and described in the *Installing and Troubleshooting SNA NRJE* manual or the *SNA NRJE Node Manager's Guide*.

OutLen **An output integer**. The *OutLen* parameter contains the total number of all characters in the *Message* array.

Result **An output integer variable**. The *Result* parameter contains error codes that occurred during execution of this intrinsic.

The *Result* parameter is set to zero if no errors took place. Always test *Result* immediately after you call this intrinsic. If *Result* is not zero, a problem has occurred. See “Result Codes and Messages,” in the *Installing and Troubleshooting SNA NRJE* manual or the *SNA NRJE Node Manager’s Guide*.

Description

Use NRJEErrMsg to obtain and write messages that correspond with *Result* arrays of calls to other NRJE intrinsics. Information associated with *Error* is returned in *Message*. Find out the actual size of *Message* from *OutLen*. Set *FileNum*=0 if you want to prevent NRJEErrMsg from writing the contents of *Message*. After NRJEErrMsg has returned control to your program, test the first element of *Result*.

Text Reference

See the *Installing and Troubleshooting SNA NRJE* manual or the *SNA NRJE Node Manager’s Guide*, for the messages you receive in *Message*.

NRJEHJCancel

Cancels jobs already transmitted to the host, by marking the Job Log to dispose of returning data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.

Syntax

```
               BA      BA          I          BA        I
NRJEHJCancel (Wsid, UserName, UserNameLen, JobList, JobListLen,
                    I         IA
              NumCancelled, Result)
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

UserName **An input byte array** (value required if *JobList* and *JobListLen* are not supplied. If both *UserName* and *JobList* are supplied, only *JobList* is used to match host jobs). The *UserName* parameter specifies the kind of job information you want. A name can be one of the following:

- An @ character. Use of this character cancels all your own jobs. If you have NM capability, all jobs will be cancelled.
- A *UserName.AcctName*. (If you are referring to jobs submitted by another user with this parameter, you must have NM capability).

UserNameLen **An input integer variable** (value required if *UserName* is supplied). The *UserNameLen* parameter specifies the number of bytes in *UserName*, from 0 through 17.

JobList **An input byte array** (value required if *UserName* and *UserNameLen* are not supplied; if *JobList* is supplied, *UserName* and *UserNameLen* are not required, and will be ignored).

The *JobList* array is a list of host job names and/or numbers. Names are those on the host JCL JOB card. When cancelling jobs by job names, only jobs submitted by the logon *UserName.AcctName* are cancelled. The exception to this is a logon with NM capability. When cancelling by job number, you must have NM capability to cancel jobs submitted by another user. Numbers are those assigned by the host system. Each entry is eight characters long, left-justified, and blank-filled. An example of *JobList* with four entries is:

```
DS1
1099
2122
FIRSTONE
```

JobListLen **An input integer variable** (value required if *JobList* is supplied). The *JobListLen* parameter specifies the number of entries in the *JobList*.

NumCancelled **An output integer (required)**. *NumCancelled* is returned to reflect the number of host jobs that were successfully cancelled.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use NRJEHJCancel to cancel jobs already transmitted to the host. This intrinsic functions by marking the Job Log to purge returning output data sets for the cancelled jobs. Specify a job set one of these ways:

- All files of the logon user, or all files of another user if you have NM capability.
- Individually, by host job number, or host job name. Note that if more than one job matches the job name, only those matching the logon *UserName.AcctName* are cancelled. If you have NM capability, all jobs that match the job name are cancelled.

After NRJEHJCancel has returned control to your program, test the first element of *Result*. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Text Reference

See the `CANCEL` command in Chapter 3 , “User Commands.”

NRJEHJInfo

Obtains a list of jobs transmitted to the host, the number of data sets received against those jobs, and the special destinations of those data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.

Syntax

```
             BA      I         I        BA           I          BA
NRJEHJInfo (Wsid, StartPos, Incount, UserName, UserNameLen, JobList,
                    I        I       IA       BA     BA     IA
            JobListLen, Outcount, JobNum, JobName, User, XmitRcv,
             LA   BA  BA  BA    IA
            Time, PR, PU, FO, Result)
```

Parameters

Wsid — **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

StartPos — **An input integer variable (required)**. This parameter allows the user to fill the output arrays beginning with the *StartPos* — the matched item from the Job Log. In this way, smaller output arrays can be used with repeated calls to `NRJEHJInfo` to show all the jobs transmitted to the host. Set *StartPos* to `1` (not 0) to return the first *Incount* or *Outcount* number of matched jobs, whichever is less.

Incount — **An input integer variable (required)**. The *Incount* parameter specifies the maximum number of elements to be returned in each *output* array parameter except *Result*.

UserName — **An input byte array** (required if *JobList* and *JobListLen* are not supplied. If both *UserName* and *JobList* are supplied, only *JobList* is used to match host jobs). The *UserName* parameter specifies the kind of job information you want. A *Username* can be one of the following:

- An @ character. If you have NM capability, this means all jobs of all users. Otherwise, it means all jobs of the logon *UserName.AcctName*.

- A *UserName.AcctName*. You must have NM capability to specify jobs submitted by other users.

UserNameLen **An input integer variable** (required if *UserName* is supplied). The *UserNameLen* parameter specifies the number of bytes in *UserName*, from 0 through 17.

JobList **An input byte array** (required if *UserName* and *UserNameLen* are not supplied. If *JobList* and *JobListLen* are supplied, then *UserName* and *UserNameLen* are not required parameters and are ignored).

The *JobList* array is a list of host job names, host job numbers, and/or SpoolfileIds.

- Names are those on the host JCL JOB card. By using job names, you obtain information for all jobs with this name under your own logon *UserName.AcctName*, or for all users if you have NM capability.
- Numbers are those assigned by the host system. When using job numbers, you must have NM capability to show jobs submitted by another user.
- The SpoolfileId is assigned by the spooler after a job is successfully submitted. When used in the Joblist array, the SpoolfileIds must be entered starting with the character “^” followed by up to seven digits.

Each entry is eight characters long, left-justified, and blank-filled. Following is an example of a *JobList* with four entries:

```
DS1
1099
^59422
FIRSTONE
```

JobListLen **An input integer variable** (required if *JobList* is supplied). The *JobListLen* parameter specifies the number of entries in the *JobList*.

Outcount **An output integer variable (required)**. The *Outcount* parameter contains the actual number of elements in each of the output arrays, except for *Result*.

JobNum **An output integer array (required)**. The *JobNum* array is a list of host job numbers of all submitted jobs that meet the specification in *UserName* and *JobList*.

JobName **An output byte array (required)**. The *JobName* array is a list of host job names of all submitted jobs that met the specification in *UserName* and *JobList*.

Each entry in *JobName* is eight characters long, left-justified, and blank-filled. Each entry in *JobName* corresponds to an entry in *JobNum*.

User **An output byte array (required)**. The *User* array is a list of users of all submitted jobs that meet the specifications in *UserName* and *JobList*.

Each entry in *User* corresponds to an entry in *JobNum*.

Each entry is eighteen characters long, left-justified, and blank-filled, in the form *UserName.AcctName*. An example of *User* with three entries is:

```
tyrone.shoelace
phillip.mgastank
hardlei.able
```

Each entry has at least one trailing blank.

XmitRcv **An output integer array (required)**. The *XmitRcv* array is a list of the number of data sets received against the transmitted jobs that met the specifications in *UserName* and *JobList*.

Each entry corresponds to an entry in *JobNum*.

Time **An output logical array (required)**. The *Time* array is a list of three-word entries in a format that is compatible with intrinsics for the native language in use. The first word contains the date information, and the remaining two words contain the time information.

If the corresponding *XmitRcv* entry is zero, the *Time* entry is the time that the job was transmitted to the host. If the corresponding *XmitRcv* entry is greater than zero, the *Time* entry is the time that the most recent data set was received.

Each entry in *Time* corresponds to an entry in *JobNum*.

PR **An output byte array (required)**. The *PR* array is a list of destinations for output routed to NRJE printers on the standard form. The entries are from transmitted jobs that met the specifications in *UserName* and *JobList*.

Each entry in *PR* corresponds to an entry in *JobNum*.

Each entry is eighty-eight characters long, left-justified, and blank-filled, containing at least one trailing blank. No embedded blanks are in an entry except in the case of a file equation. In this case, one blank follows the initial `:FILE`.

An entry is one of these:

- file equation (:FILE *filename*, etc.).
- *ldev* of an MPE output device (numeric entry).
- an MPE device class name, enclosed in double quotes.
- an actual file reference.
- an alternate *formid* in the Lookup Table, enclosed in single quotes.

PU

An output byte array (required). The *PU* array is a list of destinations for output routed to NRJE punches on the standard form. The entries are from transmitted jobs that met the specifications in *UserName* and *JobList*.

Each entry in *PU* corresponds to an entry in *JobNum*.

Each entry is eighty-eight characters long, left-justified, and blank-filled. The same form is used as for *PR* entries.

FO

An output byte array (required). The *FO* array is a list of destinations for output routed to NRJE printers and punches on forms other than the standard form. Entries are from transmitted jobs that met the specifications in *UserName* and *JobList*.

Each entry in *FO* corresponds to an entry in *JobNum*.

Each entry is eighty-eight characters long, left-justified, and blank-filled. The same form is used as for *PR* entries.

Result

An eight-element integer output array (required). The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use NRJEHJInfo to obtain information about jobs that have already been transmitted to the host. Input information to this intrinsic is as follows:

- Workstation identification: *Wsid*.
- Logical starting position within the Job Log: *StartPos*.
- Maximum number of elements to be returned in all *output* arrays except *Result*: *Incount*.
- Job set desired: *UserName*, *UserNameLen*, *JobList*, and *JobListLen*. The structure and contents of *UserName* and *JobList* determine the job set desired.

You can specify either a *UserName* and *UserNameLen*, or a *JobList* and *JobListLen*. If both pairs are specified, only *JobList* and *JobListLen* are used.

NRJEHJInfo zeros or blanks all output arrays on entry.

After NRJEHJInfo has returned control to your program, test the first element of *Result*.

Text Reference

This intrinsic provides information for the SHOW command. See Chapter 3 , “User Commands.”

NRJEJobInfo

Obtains a list of jobs submitted, but not transmitted to the host, and job spool file characteristics.

Syntax

```
               BA      I      BA      I        I          IA  BA
NRJEJobInfo ( Wsid, InCount, Name, NameLen, OutCount, FileId, User,

              BA        DA       IA      IA    LA     IA     IA
            JobName, JobSize, Priority, Rank, Time, State, Result
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

InCount

An input integer variable. The *InCount* parameter specifies the maximum number of elements (from 1 through 53) to be returned in all *output* array parameters except *Result*.

Name

An input byte array. The *Name* parameter specifies the kind of job information you want.

A *Name* can be any of the following:

- A file ID
- A job name
- An @ character
- A *UserName.AcctName*

Table 5-3, following this intrinsic's description, describes the various ways Name can be used.

NameLen

An input integer variable. The *NameLen* parameter specifies the number of bytes in *Name*, from 1 through 17.

OutCount

An output integer variable. The *OutCount* parameter contains the actual number of elements in all of the output arrays except *Result*.

The maximum value for *OutCount* is 100.

FileId

An output integer array. The *FileId* array is a list of file identifications of all submitted jobs that met the specification in *Name*.

Each entry is an integer reference to a spool file identification.

User

An output byte array. The *User* array is a list of users of all submitted jobs that met the specification in *Name*.

Each entry in *User* corresponds to an entry in *FileId*.

Each entry is eighteen characters long, left-justified, and blank-filled, in the form *UserName.AcctName*. Following is an example of *User* with three entries:

```
TYRONE.SHOELACE
PHILLIP.MGASTANK
HARDLEI.ABLE
```

At least one trailing blank is in an entry.

JobName

An output byte array. The *JobName* array is a list of job names of all submitted jobs that met the specification in *Name*.

Each entry in *JobName* corresponds to an entry in *FileId*.

Each entry is eight characters long, left-justified, and blank-filled. An example of *JobName* with three entries is as follows:

```
FIRSTONE
PERRY
ENTE
```

JobSize

An output double-word integer array. The *JobSize* array is a list of job sizes, in sectors, of all submitted jobs that met the specification in *Name*.

Each entry in *JobSize* corresponds to an entry in *FileId*.

Priority

An output integer array. The *Priority* array is a list of job priorities of all submitted jobs that met the specification in *Name*.

Each entry in *Priority* corresponds to an entry in *FileId*, where *Priority[i]* is from 0 through 14; 14 is the highest priority.

Rank

An output integer array. The *Rank* array is a list of job ranks in the transmission queue of all submitted jobs that met the specification in *Name*.

Each entry in *Rank* corresponds to an entry in *FileId*.

Time **An output logical array**. The *Time* array is a list of four-word entries describing when a spool file was created.

Each entry in *Time* corresponds to an entry in *FileId*.

The structure of each four-word entry is as follows:

year
day of the year
hour
minute

where *day of the year* is from 1 through 366.

State **An output integer array**. The *State* array is a list of the spooling states of each file.

Each entry in *State* corresponds to an entry in *FileId*.

The values returned in *State[i]* are as follows:

MPE V and MPE XL prior to release 2.1

0 = *FileId[i]* is *active*.
1 = *FileId[i]* is *ready*.
2 = *FileId[i]* is *opened*.
3 = *FileId[i]* is *locked*.

MPE XL release 2.1 and later

0 = *FileId[i]* is *active*.
1 = *FileId[i]* is *ready*.
2 = *FileId[i]* is *opened*.
3 = *FileId[i]* is *deferred*.
4 = *FileId[i]* is *created*.
5 = *FileId[i]* is being *transferred*.
6 = *FileId[i]* is being *printed*.
7 = *FileId[i]* has a *problem*.
8 = *FileId[i]* is *delete pending*.
9 = *FileId[i]* has *SPSAVE* specified.

Refer to *MPE V/E System Operation and Resource Management Reference Manual* or the *MPE XL Native Mode Spooler Reference Manual* for an additional discussion of spooling states.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use NRJEJobInfo to obtain information about jobs that have been submitted, but are not yet transmitted. Input information to this intrinsic is as follows:

- Workstation identification: *Wsid*.
- Maximum number of elements to be returned in all output arrays except *Result*: *InCount*.
- Job set desired: *Name* and *NameLen*. The structure and contents of *Name* determines the job set desired. Refer to Table 5-3.

  Output from NRJEJobInfo consists of the following:

- The number of jobs qualified by *Name*: *OutCount*.
- An entry in each of the output arrays, except *Result*, describing spool file characteristics

  After NRJEJobInfo has returned control to your program, test the first element of *Result*.

Text Reference

This intrinsic provides information for the SHOW command. See Chapter 3 , “User Commands.”

Table 5-3 Use of Name

Structure	Form	Effect
`File identification`	A five-character numeric string in the form *nnnnn*.	Information is wanted for a submitted job with a spool file identifier of *nnnnn*.
`Job name`	An eight-character alphanumeric string beginning with a letter.	Information is wanted for all submitted jobs that have this job name.
`@`	A single @ character.	Information is wanted for all jobs submitted by the logon user.
`Username.Acctname`	A seventeen-character string meeting MPE standards for *`UserName.AcctName`*. A period (.) must be part of the string. If the *`UserName.AcctName`* is different from the logon *`UserName.AcctName`*, the logon user must be configured with NM capability.	Information is wanted for all submitted jobs that have this `UserName.AcctName`.

NRJELUList

Provides a list of all LUs and related characteristics for a workstation.

Syntax

```
                      BA      I             I          BA        IA
NRJELUList          (Wsid, MaxEntries, ReturnEntries, LUList, ChainSizeList,
                          DA             DA                IA
                    MinJobSizeList, MaxJobSizeList, ActiveLuNameList
                          IA          IA
                    AutoStartList, Result
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

MaxEntries

An input integer. The *MaxEntries* parameter contains the maximum number of entries expected in all *output* array parameters except *Result*. The *MaxEntries* parameter value should be the number of LUs configured for the workstation, from 1 through 16.

Make sure that the available length of each of the *output* array parameters (*LUList*, *ChainSizeList*, *MinJobSizeList*, *MaxJobSizeList*, *ActiveLUNumList*, and *AutoStartList*) is sufficient for the expected number of entries.

ReturnEntries

An output integer. The *ReturnEntries* parameter normally contains the actual number of entries in all *output* array parameters except *Result*.

After calling NRJELUList, *ReturnEntries* can contain the required number of entries for *LUList*, *ChainSizeList*, *MinJobSizeList*, *MaxJobSizeList*, *ActiveLUNumList*, and *AutoStartList*. This occurs only when the first word of *Result* is 137, which means that *MaxEntries* was too small.

LUList

An output byte array. The *LUList* array contains a list of entries of all LU names for the workstation. Each entry is eight characters, left-justified, with trailing blanks. Following is an example of a three-entry *LUList*:

```
ERST
NEXTONE
LAST
```

ChainSizeList

An output integer array. The *ChainSizeList* array contains a list of chain sizes for each LU in a workstation.

A chain size is the number of RUs sent from the workstation to the host before requesting a positive response.

MinJobSizeList

An output double-word integer array. This parameter is hardcoded to return 0.

MaxJobSizeList

An output double-word integer array. This parameter is hardcoded to return a maximum sector size of 2147483647.

ActiveLUNumList

An output integer array. The *ActiveLUNumList* array contains a list of active LU numbers. A zero entry indicates an LU is inactive. An active entry is a positive integer, which is the LU number.

Following is a three-entry *ActiveLUNumList* corresponding to *LUList*, above:

`0` ERST is inactive.

`34` NEXTONE is active, and its LU number is 34.

`22` LAST is active, and its LU number is 22.

AutoStartList

An output integer array. The *AutoStartList* array contains a list of flags of LUs that are automatically initiated.

Automatic starting is a configurable item.

A zero element is returned if an LU is not initially started when a workstation is started.

A -1 element is returned if an LU is initially started.

following is a three-entry *AutoStartList* corresponding to *LUList*, above:

0 ERST is not initially started.

-1 NEXTONE is initially started.

-1 LAST is initially started.

Result

An eight-element integer output array (required). The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in "Parameters Common to NRJE Intrinsics" in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Call NRJELUList to obtain a list of all LUs, and lists of LU characteristics, for a workstation. A positional correspondence exists among the entries in *LUList* and the characteristics listed in *ChainSizeList*, *MinJobSizeList*, *MaxJobSizeList*, *ActiveLUNumList*, and *AutoStartList*. The length of all lists must be large enough so that all configured LUs can be entered; otherwise, an error will occur. After NRJELUList has returned control to your program, test the first element of *Result*.

Text Reference

DISPLAY shows LUs and related characteristics. This is the same information that displays when you enter the DISPLAY command with the *LUT* parameter. See DISPLAY in Chapter 3 , "User Commands."

NRJEQueue

Obtains information about a workstation transmission queue (the spooled reader).

Syntax

```
                 BA        I           I        BA          I          I
NRJEQueue (Wsid, Reserved, QueueLdev, Reserved, Reserved, ChainSize,
              I             I             I             I
           CompFlag, TransIndicator, ReaderStatus, ReaderFence,
              I          IA
           SystemFence, Result )
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

Reserved

This field is not in use at this time.

QueueLdev

An output integer variable. The *QueueLdev* variable contains the MPE logical device number for the transmission queue. This is always a spooled device.

Reserved

This field is not in use at this time.

Reserved

This field is not in use at this time.

ChainSize

An output integer. Use the *ChainSize* value to find the default chain size configured for transmission to the host.

The size of a chain is the number of request units (RUs) to be transmitted to a host before NRJE requires a response.

CompFlag

An output integer. Use the value of *CompFlag* to learn the state of compression configured for a workstation transmission queue. The values of *CompFlag* are as follows:

- Any odd number: Data is transmitted to the host in compressed form.
- Any even number: Data transmitted to the host is not in compressed form.

TransIndicator

An output integer. The value of *TransIndicator* identifies the Native Language ID used by the NRJE reader to obtain the ASCII–EBCDIC translation table. Language IDs range from 0 through 255.

ReaderStatus

An output integer. The *ReaderStatus* value shows the spooling status of the reader identified by *QueuedLdev*. The values returned for *ReaderStatus* are as follows:

- Any odd number: The reader is spooled at this time.
- Any even number: The reader configured in *QueuedLdev* is not spooled. You are unable to submit jobs. The NRJE operator must issue a reader up command (RDRUP) or the system operator must issue the MPE command OPENQ.

ReaderFence

An output integer. The *ReaderFence* parameter functions like an MPE OUTFENCE for the *QueuedLdev*. When *ReaderFence* > 0, and communication to the host is initiated, each spool file in the transmission queue whose priority is less than or equal to the *ReaderFence* value is not transmitted. When *ReaderFence* = 0, the MPE system output fence value (*outfence*) is used.

Call NRJEAlter to reset the priority of job stream files in the workstation transmission queue.

SystemFence

An output integer. The *SystemFence* is the MPE *outfence* parameter.

When *ReaderFence* = 0, each output spool file whose priority is less than or equal to *SystemFence* is not transmitted.

Result

An eight-element integer output array (required). The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use NRJEQueue to obtain information about the spooled reader for your workstation. Identify your workstation in *Wsid* as an input parameter. The NRJEQueue intrinsic provides you with the following information:

- The logical device number of the queued reader for your workstation.
- The following set of parameters associated with the queued logical reader:
  - — The *ChainSize*, the number of RUs per job stream.
  - — A flag indicating whether data compression is performed.
  - — A data translation indicator.
  - — A spooling indicator showing the current status of the queued reader.
  - — An output fence value for the queued reader.
- The system-wide output fence parameter.

After NRJEQueue has returned from your call, test the *Result* array.

Text Reference

The NRJEQueue intrinsic provides information for the DISPLAY command. See Chapter 3 , “User Commands.”

NRJESubmit

Prepares a batch input job stream for transmission to a host system.

NOTE The NRJESubmit intrinsic is documented to provide backward compatibility with releases of the SNA NRJE product prior to version A.52.00. Use of the NRJE2Submit intrinsic, available on version A.52.00 or later, is the preferred method because of the enhanced capabilities available with that intrinsic.

Syntax

```
              BA      I         BA        IA        IA       BA    I
NRJESubmit ( Wsid, NumFiles, FileList, FileLens, SubCode, Name, Pri,
                I         D          IA
             Direct, SpoolNumber, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

NumFiles **An integer input variable, from 1 through 40 (required)**. The *NumFiles* parameter indicates the following:

- The number of file names that are in *FileList*.
- The number of elements in *FileLens*.
- The number of elements in *SubCode*.

FileList **An input byte array (required)**. The *FileList* array contains actual or formal file designators for each file, in sequence, that goes into your job stream. The contents of the files named in *FileList* are broken into spool files, with one host job per spool file.

The format of the *FileList* is a continuous string of actual or formal designators without any delimiters between entries. Instead of using delimiters between substrings, the length of each substring is in the *FileLens* array. The file designators $NULL, $STDLIST, and $NEWPASS are not allowed in the *FileList*.

You must provide sufficient elements for this array. The maximum length for this array is 3440 bytes (40 files *

86-character maximum per file reference). The format of a file reference is as follows:

```
infile[/lockword][.GrpName[.AcctName[:envid]]]
```

The meanings of the parameters used in this reference are described in Chapter 3 , “User Commands,” in the SUBMIT command description.

FileLens **An input integer array (required)**. The *FileLens* array contains the length in bytes of each file identifier in the *FileList* array.

Each element of *FileLens* specifies the length, in bytes, of its corresponding file identifier in *FileList*, where *FileLens[i]* is from 0 through 86.

The maximum length of the *FileLens* array is 40 elements, to accommodate the MPE stacksize for the NRJE intrinsics.

SubCode **An input integer array**. Each element of the *SubCode* array corresponds with a file name in the *FileList* array. Each *SubCode* element must contain one of these values:

- 0 = NRJE should translate the file from ASCII to EBCDIC.
- 2 = NRJE should not attempt to translate the file from ASCII to EBCDIC.

You must use *SubCode[i]*=0 with *Direct*=1. Table 5-4, in this intrinsic’s description, shows how the *SubCode* and *Direct* parameters relate to each other.

Name **An eight-character input byte array**. The *Name* array provides an identification for each spool file created (one per host job).

The identification in *Name* must begin with a letter, contain alphanumeric characters, and be left-justified with trailing blanks.

See Table 5-3, in the *NRJEJobInfo* intrinsic description.

Pri **An integer input parameter**. The *Pri* parameter specifies your job input priority, from 0 through 14; 14 is the highest priority.

If you set *Pri*=0, the MPE-configured default is used.

Direct **An input integer**. The *Direct* parameter, together with the *SubCode* entry for each *FileList* entry, indicates translation and compression.

The settings for *Direct* are as follows:

- An odd number means your job file is written directly. Any translation and compression will be done during transmission of the job file to the host.
- An even number means that any required translation or compression is done while copying the files you specified in *FileList* to the spool file of your job.

Table 5-4, in this intrinsic's description, shows how the Direct and SubCode parameters relate to each other.

SpoolNumber **An output double integer variabl**e. The *SpoolNumber* variable contains an MPE spool file identification number of the first host job successfully submitted with a given NRJESubmit.

When NRJESubmit is not successful in submitting any host job, the contents of *SpoolNumber* are undefined.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in "Parameters Common to NRJE Intrinsics" in the introduction to this chapter.

Always test the first element of immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

The NRJESubmit intrinsic calls the NRJE2Submit intrinsic to accomplish the submit process. The call is made with the *PR*, *PU*, and *FO* parameters blank because NRJESubmit cannot handle output routing. NRJE2Submit breaks the input stream into spool files on the basis of host jobs (one spool file per host job), and returns a list of the MPE spool file IDs to NRJESubmit. NRJESubmit then returns the first spool file ID in that list.

Console commands may be embedded in the submitted input files. If the submitter has NM capability, or an embedded command is one of those allowed to all users (specified in the NMMGR Workstation Data screen), the console command is passed on as part of the job. Otherwise, the command is stripped out and a warning message is written to $STDLIST. (The remainder of the job is submitted.)

This intrinsic does not transmit your job to the host. Job transmission takes place after communication is established with the host. See NRJECONTROL START in the *SNA NRJE Node Manager's Guide*.

Use the *Wsid* parameter to identify your workstation. The *NumFiles* parameter must specify the number of files that comprise your job input stream. The *FileList*, *FileLens*, and *SubCode* parameter arrays contain an entry for each file that goes into the job stream.

You can affect a job stream file you create with *NRJESubmit* parameters in these ways:

- Identify your job stream with the *Name* parameter. Use *Name* later to retrieve information about your job stream. If you don't specify a *Name*, the host job name as detected by NRJE2Submit is used.
- Specify your job priority in the *Pri* parameter. The higher the *Pri*, the more urgent your job stream. When you use *Pri*=0 you are specifying the NRJE configuration default for the workstation.
- Use *Direct* and the *SubCode* array together to specify (1) when translation and compression are to take place: either (a) during transmission of the job stream spool file to the host, or (b) during job stream creation, when each of your files is copied to the spool file; and (2) what type of translation and compression are to take place.

For example, you would omit translating a file that contains packed decimal data. Specify that the file has already been translated by setting its *SubCode[i]*=2, and setting *Direct*=0.

Table 5-4 SubCode — Direct Relationship

	Direct=0	**Direct=1**
SubCode[i]=0	Translation and compression take place when *FileList[i]* is submitted.	No translation and compression take place when *FileList[i]* is submitted. Translation and compression take place when a job stream is transmitted.
SubCode[i]=2	Compression takes place when *FileList[i]* is submitted.	This combination is illegal for any *FileList* entry.

Test *Result* to learn the effects of your call to this intrinsic.

Retain the contents of *SpoolNumber* for use as an input parameter in other intrinsics, such as NRJEJobInfo.

Text Reference

See the SUBMIT command in Chapter 3 , "User Commands."

NRJE2Submit

Creates a job input stream on the transmission queue to send to a host system, and allows the user to specify output destinations for data sets returning from the job submitted to the host. If you are connected to a VSE/POWER host system, the output routing capabilities are not available.

Syntax

```
               BA      I        BA        IA         IA        BA
NRJE2Submit (Wsid, NumFiles, FileList, FileLens, SubCode, Name,
              I     I     BA  BA  BA      I         IA         I
             Pri, Direct, PR, PU, FO, MaxJobs, SpoolList, NumJobs,
               IA
             Result )
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

NumFiles

An integer input variable, from 1 through 40 (required). The *NumFiles* parameter indicates the following:

- The number of file names that are in *FileList*.
- The number of elements in *FileLens*.
- The number of elements in *SubCode*.

FileList

An input byte array (required). The *FileList* array contains actual file designators, formal file designators, or file references for each file, in sequence, that goes into your job stream. The contents of the files named in *FileList* are broken into spool files, with one host job per spool file.

The format of the *FileList* is a continuous string of actual or formal file designators without any delimiters between entries; the length of each substring is in the *FileLens* array.

The file designators $NULL, $STDLIST, and $NEWPASS are not allowed in the *FileList*.

You must provide sufficient elements for this array. The maximum length for this array is 3440 bytes (40 files * 86 character maximum per file reference). The format of a file reference is as follows:

Infile[/LockWord][.GrpName[.AcctName[:Envid]]]

The meanings of the parameters used in this reference are described in Chapter 3 , "User Commands," in the SUBMIT command description.

FileLens **An input integer array (required)**. Each element of the *FileLens* array specifies the length in bytes of its corresponding file identifier in the *FileList* array; *FileLens[i]* is from 0 through 86.

The maximum length of the *FileLens* array is 40 elements, to accommodate the MPE stacksize for the NRJE intrinsics.

SubCode **An input integer array**. Each element of the *SubCode* array corresponds to a file name in the *FileList* array. Each *SubCode* element must contain one of these values:

- 0 = NRJE should translate the file from ASCII to EBCDIC.
- 2 = NRJE should not translate the file to EBCDIC.

  You must use *SubCode(i)*=0 with *Direct*=1. Table 5-4, in the NRJESubmit intrinsic description, shows how the *SubCode* and *Direct* parameters relate to each other.

Name **An eight-character input byte array**. The *Name* array provides an identification for each spool file created (one per host job in your submitted stream).

The identification in *Name* must begin with a letter, contain alphanumeric characters, and be left-justified and blank-filled.

If a blank name is supplied, the spool file is named with the jobname from your JCL JOB card.

Pri **An integer input parameter**. The *Pri* parameter specifies your job input priority, from 0 through 14; 14 is the highest priority.

If you set *Pri*=0, the MPE-configured default is used.

Direct **An input integer**. The *Direct* parameter, together with the *SubCode* element for each *FileList* entry, indicates translation and compression.

The settings for *Direct* are as follows:

- An odd number: your job file is written directly to the MPE Spooler. Any translation or compression is done during transmission of the job file to the host.
- An even number: any required translation or compression is done while copying the files you specified in *FileList* to the spool file of your job.

Table 5-4, in the NRJESubmit intrinsic description, shows how the *Direct* and *SubCode* parameters relate to each other.

The three parameters described below (*PR*, *PU*, and *FO*) cannot be used if you are connected to a VSE/POWER host system.

PR **An input byte array** of up to eighty-eight characters, left-justified and blank-filled, with no embedded blanks.

The *PR* array is used to control standard forms output to the printer and contains one of these:

- A backreferenced formal file designator preceded by an asterisk (**fileid*).
- The *ldev* of an MPE output device (a numeric entry).
- An MPE device class name, enclosed in double quotation marks ("DevClass").
- An actual file designator. The file must be built before the job is submitted.
- An alternate form name used as a key into the Lookup Table. This designation can be up to eight characters long and must be delimited by single quotes (*'formid'*).

PU **An input byte array** of up to eighty-eight characters, left-justified and blank-filled, with no embedded blanks. The *PU* array is used to control standard forms output to the punch stream. It is the same form as the *PR* array.

FO **An input byte array** of up to eighty-eight characters, left-justified and blank-filled, with no embedded blanks. The *FO* array is used to control special forms output for both the printer and punch stream. It is the same form as the *PR* array.

MaxJobs **An input integer**. The *MaxJobs* parameter specifies the maximum number of elements to be returned in the *SpoolList* array.

SpoolList **An output integer array**. The first *NumJobs* or *MaxJobs* (whichever is less) elements of the *SpoolList* array contain the MPE spool file identification numbers of the host jobs (one per spool file) successfully submitted.

When NRJE2Submit is not successful in submitting a job, the corresponding element of *SpoolList* is set to 0 and control is returned to the caller. The elements before that element in *SpoolList* are valid spool file IDs, and elements following it are undefined. The contents of the *Result* array more fully describe the error that occurred.

NumJobs **An output integer**. The *NumJobs* parameter specifies the total number of jobs submitted to the host.

If insufficient room exists in *SpoolList* to return job numbers of jobs submitted to the host, then *NumJobs* is greater than *MaxJobs*.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use *PR*, *PU*, and *FO* to specify destinations for data sets returning from the jobs submitted to the host.

The host jobs contained in the submitted *Infiles* (and any included ##FD files) are broken into separate spool files. A list of the spool file IDs of the jobs is returned to the user.

Console commands can be embedded in the submitted input files, provided the console commands precede the first JCL JOB card in an input stream. If the submitter has NM capability, or an embedded command is one of those allowed to all users (specified in the NMMGR Workstation Data screen), the console command is passed on as part of the job. Otherwise, the command is stripped out and an error message is written to $STDLIST (the remainder of the job is submitted). Console commands embedded between jobs supplied on a single input stream are not supported.

SIGNOFF card images are stripped from the input stream and a warning message is written to $STDLIST.

See the SUBMIT command explanation in Chapter 3 , “User Commands,” for additional information.

NRJEWelcome

Obtains the contents of the NRJE welcome message.

Syntax

```
                I         I        BA     IA
NRJEWelcome ( MaxLen, ActLen, Msg, Result )
```

Parameters

MaxLen **An input integer variable**. The *MaxLen* parameter specifies the maximum text length that can be returned in *Msg*.

The maximum length of the NRJE welcome message is 1912 characters. (This includes the carriage return and the line feed.)

ActLen **An output integer variable**. The *ActLen* parameter contains the actual number of characters of the NRJE welcome message returned to you in the *Msg* array.

Msg **An output byte array**. The *Msg* array contains the text of the NRJE welcome message.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

NRJEWelcome is used to obtain the text of an NRJE welcome message. You must provide an array large enough to hold all of the welcome message text. Specify the message array size when you call NRJEWelcome. NRJE can accept lines up to 80 characters long and adds a carriage return and a line feed at the end of each line. The maximum total number of characters that can be included in the welcome file is 1912. The actual message length is returned by the intrinsic. When NRJEWelcome returns control to your program, you should test *Result*.

Text Reference

NRJE welcome messages are output immediately after the product identification banner when the subsystem command interface is used.

NRJEWriter

Obtains information about the logical writers configured for a workstation.

Syntax

```
                BA      I        I          BA         BA        IA
NRJEWriter ( Wsid, InCount, OutCount, WriterList, LdevList, PrList,
                   IA        BA        BA       IA       IA
             TransList, Reserved, Reserved, Forced, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

InCount **An integer input variable**. The *InCount* parameter specifies the maximum number of elements to be returned in all of this intrinsic's *output* arrays except *Result*.

The maximum value you can specify for *InCount* is 14. This is because no more than seven printer writers and seven punch writers can be configured for a workstation. This is a host restriction.

OutCount **An integer output variable**. The *OutCount* parameter identifies the actual number of elements returned in all of this intrinsic's *output* arrays except *Result*.

WriterList **An output byte array**. The *WriterList* array is a list of logical writer names configured for your workstation, such as *PRn* and *PUn*, where *n* is from 1 through 7.

Each entry is eight characters long, left-justified, and filled with trailing blanks. Following is a typical set of three *WriterList* entries:

```
PR1
PR2
PU1
```

LdevList **An output byte array**. The *LdevList* array is a list of output designators that correspond to the names of logical writers in *WriterList*. These may be an *ldev*, a logical device class, or a *'formid.'* Each entry is eight characters long, left-justified, and filled with trailing blanks.

PriList — **An output integer array**. The *PriList* array is a list of output priorities that correspond to logical device numbers in *LdevList*.

Each entry is an output priority, where *PriList[i]* is from 0 through 14; 14 is the highest priority. When *PriList[i]*=0, the system-configured default value is used.

TransList — **An output integer array**. The *TransList* array is a list of Native Language identifiers used to designate an EBCDIC–ASCII translation table.

Each entry indicates the kind of translation done for a corresponding logical writer in *WriterList*. Each entry in *TransList* contains one of these values:

- 0–255 = A language ID.
- -1 = No translation will be done.

Reserved — This field is not in use at this time.

Reserved — This field is not in use at this time.

Forced — **An output integer array**. The *Forced* array is a list of flags for each writer of a workstation. Each entry indicates whether the configured *formids* are "forced" specifications.

Values for *Forced[i]* are as follows:

- 0 = The configured *'formid'* is "forced". This supersedes JCL-specified form references.
- -1 = User-specified special form names are used.

Result — **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in "Parameters Common to NRJE Intrinsics" in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

`NRJEWriter` is used to obtain information about the logical writers configured for your workstation. Enter your workstation identification (*`Wsid`*) and the maximum number of writers you expect (*`InCount`*). You receive the following output from *`NRJEWriter`*:

- The number of writers (the *`OutCount`* parameter value). *`OutCount`* is also the length of the lists shown below.
- Writer characteristics are output in separate lists:

  — Name

  — Logical device number

  — Output priority value

  — Native Language ID

  — A flag indicating whether the *`formids`* are "forced"

  When `NRJEWriter` returns control to your routine, you should test *`Result`*.

Text Reference

The `NRJEWriter` intrinsic provides information for the `DISPLAY` command. See Chapter 3 , "User Commands."

NRJEWSInfo

Obtains general information about a workstation.

Syntax

```
              BA       I        BA    BA       BA       IA
NRJEWSInfo ( Wsid, ActiveFlag, JES, Remote, Reserved, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

ActiveFlag **An output integer variable**. The *ActiveFlag* parameter is used to learn whether your workstation is communicating with a host. Possible values for this parameter are as follows:

- `1` = Your workstation is active.
- `2` = A stop is pending.
- `4` = A protocol shutdown is pending.
- `%10` = A monitor request is pending.
- `%20` = LU activation is pending.
- `%40` = Your workstation is inactive.

JES **An eight-character output byte array**. The *JES* parameter is used to identify the job entry system configured at your host. Information is returned left-justified, with trailing blanks if needed.

Remote **An eight-character output byte array**. Use the *Remote* parameter to learn how the host system identifies your workstation.

For example, RMT11 is returned as

`RMT11`

Reserved This field is not in use at this time.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to this chapter.

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

NRJEWSInfo is used to obtain general information about a workstation.

Input to NRJEWSInfo is the workstation identification (*Wsid*).

Output from this intrinsic is as follows:

- Whether the workstation is actively communicating with the host.
- The host job entry system in use.
- The remote identification used for the workstation at the host.

Once NRJEWSInfo has returned control to your routine, you should test *Result*.

Text Reference

Output from the NRJEWSInfo is supplied to the DISPLAY command. See Chapter 3 , “User Commands.”

6 NRJE Manager Intrinsics

This chapter describes intrinsics that can be used only by an NRJE manager (a user with NM capability).

Introduction

All user intrinsics, including those with manager extensions, are, described in Chapter 5 , “User Intrinsics.” An NRJE manager also can call user intrinsics that do not have manager extensions.

All NRJE commands relate to intrinsics. The commands directly related to the intrinsics described in this chapter are described in the *SNA NRJE Node Manager's Guide*. User commands, including those with manager extensions, are described in Chapter 3 , “User Commands.”

An NRJE manager is responsible for day-to-day operations of NRJE workstations. These are discussed in the *SNA NRJE Node Manager's Guide*.

Common Parameters and Languages

The introduction to Chapter 5 , “User Intrinsics,” describes the following:

- Supported languages
- Programming constraints
- Common parameters
- Stack size required

These apply to NRJE manager intrinsics, also.

Intrinsic Data Types

These data types are used in intrinsic descriptions:

- BA Byte Array
- D Double-word integer variable
- DA Double-word integer array
- I Integer variable (single word)
- IA Integer array
- L Logical variable (single word)
- LA Logical array

NOTE Even numbers in logical variables and elements of logical arrays are FALSE.

Odd numbers in logical variables and elements of logical arrays are TRUE.

Manager Intrinsics

Table 6-1 lists the NRJE manager intrinsics. These intrinsics require NM capability.

NOTE

If you are allowed access to the host console facility, you can use the console intrinsics: NRJEConsCheck, NRJEConsole, NRJERcvMsg, NRJERelease, and NRJESendCmd.

Table 6-1 NRJE Manager Intrinsics

Name	Description
NRJEConsCheck	Indicates whether any outstanding host remote console messages are waiting to be received.
NRJEConsole	Obtains the host remote console of a workstation.
NRJELUStatus	Reports the status and state of an LU session of a given LU name.
NRJERcvMsg	Reads one recent host remote console message.
NRJERdrAccess	Enables or disables NRJE user access to the NRJE reader.
NRJERdrFence	Updates the outfence value of the NRJE workstation reader.
NRJERelease	Relinquishes ownership of the host remote console of a workstation.
NRJESendCmd	Sends a remote host console command.
NRJEStartWS	Initiates a workstation or LUs.
NRJEStopWS	Stops or halts LU-LU session(s) or workstation.
NRJETrace	Initiates or terminates NRJE tracing.
NRJEUpdWel	Updates the NRJE subsystem welcome message.

User Intrinsics with Manager Extensions

Table 6-2 lists the NRJE user intrinsics with manager extensions. NRJE managers (MPE users configured with NM capability) use these intrinsics with extended features. These intrinsics are also available to all NRJE users, but with limited features. See Chapter 5 , “User Intrinsics.”

Table 6-2 NRJE User Intrinsics with Manager Extensions

Name	Description
NRJEAlter	Revises the priority of a submitted job or jobs.
NRJECancel	Cancels previously submitted jobs not yet transmitted.
NRJEHJCancel	Cancels jobs already transmitted to the host by marking the Job Log to dispose of returning data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.
NRJEHJInfo	Obtains a list of jobs transmitted to the host, the number of data sets received against those jobs, and the special destinations of those data sets. This intrinsic does not apply if you are connected to a VSE/POWER host system.
NRJEJobInfo	Obtains a list of jobs submitted but not transmitted to the host, and job spool file characteristics.
NRJESubmit	Prepares a batch input job stream for transmission to a host system.
NRJE2Submit	Creates a job input stream on the transmission queue to send to a host system, and allows the user to specify output destinations for data sets returning from the job submitted to the host. If you are connected to a VSE/POWER host system, the output routing capabilities are not available.

User Intrinsics

NRJE managers and users specify these intrinsics in the same way:

- NRJEActiveWS
- NRJEErrMsg
- NRJELUList
- NRJEQueue
- NRJEWelcome
- NRJEWriter
- NRJEWSInfo

See Chapter 5 , “User Intrinsics.”

NRJEConsCheck

Indicates whether any outstanding host remote console messages are waiting to be received.

Syntax

```
                  D          IA
NRJEConsCheck ( ConsoleId, Result )
```

Parameters

ConsoleId **An input double-word integer variable**. The *ConsoleId* parameter is the remote host console identification. Obtain the value for this parameter by calling `NRJEConsole` before you call this intrinsic.

Result **An eight-element integer output array**. The *Reply* array contains error codes which occurred during execution of this intrinsic.

If no error took place, and a message is waiting, the first element of the *Reply* array is set to zero. If the first element of *Reply*=97, no message was received.

The structure of the *Reply* array is the same as the *Result* array described in “Parameters Common to NRJE Intrinsics,” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first word of *Reply* to determine whether any outstanding host remote console messages are waiting to be received. The first word of *Reply* is set to zero if no error took place.

Description

Call `NRJEConsCheck` to indicate whether any outstanding host remote console messages are waiting to be received. Obtain the *ConsoleId* value by calling `NRJEConsole` first. Test *Reply* to determine if messages are waiting to be received.

NOTE The `NRJEConsCheck` intrinsic uses *Reply* as a message-waiting indicator. Other NRJE intrinsics use the last formal parameter for error indication only.

While you own the host remote console, you receive all recent host remote console messages directed to your workstation. These messages may be in response to commands other than those you issued.

NRJEConsCheck is used with the following intrinsics:

- NRJERcvMsg: to read one recent remote host console message.
- NRJERelease: to relinquish ownership of the remote host console.

Text Reference

See the *SNA NRJE Node Manager's Guide* for the description of "Host Console Commands."

NRJEConsole

Obtains the host remote console of a workstation.

Syntax

```
                BA       D        IA
NRJEConsole ( Wsid, ConsoleId, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

ConsoleId **An output double-word integer variable**. The *ConsoleId* parameter contains the console identification used with the NRJEConsCheck, NRJERcvMsg, NRJERelease, and NRJESendCmd intrinsics.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Call NRJEConsole before NRJESendCmd, NRJERcvMsg, and NRJERelease. NRJEConsole obtains access to the host remote console for a workstation and returns *ConsoleId*.

ConsoleId is used to identify the host remote console in these intrinsics:

- NRJEConsCheck: to determine if any host remote console messages are waiting to be received.
- NRJESendCmd: to transmit a host remote console command.

- `NRJERcvMsg`: to receive a host remote console message.
- `NRJERelease`: to relinquish control of the host remote console.

After `NRJEConsole` has returned control to your program, test the first element of *`Result`*.

Text Reference

CONSOLE also obtains access to the host remote console. See the *SNA NRJE Node Manager's Guide* for more information.

NRJELUStatus

Reports the status and state of an LU session of a given LU name.

Syntax

```
                 BA     BA        IA        IA
NRJELUStatus ( Wsid, LUName, InfoArray, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

LUName **An input byte array**. In MPE V, this refers to an LU class name configured in the MPE V Configuration: Classes screen. For MPE XL, this is the host name (not the NAU) of an LU. An *LUName* begins with a letter and may be up to eight alphanumeric characters long. If an *LUName* is less than eight characters, it must be filled with trailing blanks.

InfoArray **An output integer array**. For each LU session, the *InfoArray* contains status and state information such as termination reason, session status, session state, and LU state. The structure of this array is given in the description below.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

The `NRJELUStatus` intrinsic reports the status and states of an LU session of a given *`LUName`* in *`InfoArray`* (a description of *`InfoArray`* is given below). Transmission chain size is also reported. Obtain the *`LUName`* of a *`Wsid`* by calling `NRJELUList`. When `NRJELUStatus` returns control to your routine, test the first element of *`Result`*.

Table 6-3 InfoArray Structure

Word	Contents and Meaning
0–3	LU name
4	LU Entry State. Values are as follows: • 1 = active • 2 = stop pending • 4 = protocol shut down pending • %10 = monitor request pending • %20 = LU activation pending • %40 = inactive
5	Chain size, in number of RUs.
6	LU number.
7	Reserved.
8	Data Flow Control state. Upper byte values are as follows: • `%000000, (0)` = Normal • %000400, (1) = <F255P255>SHUTD received • %001000, (2) = <F255P255>SHUTC received • %001400, (3) = <F255P255>Quiescing • %002000, (4) = <F255P255>Quiescent Data Flow Control Line State. Lower byte values are as follows: • 0 = Inbound flow • 1 = Outbound flow • 2 = Contention mode
9	Reserved.

Table 6-3 InfoArray Structure

Word	Contents and Meaning
10	Session Control Status. Values are as follows: • 0 = reset • 1 = waiting for Start Data Traffic • 2 = normal flow • 3 = UNBIND received • 4 = terminate state
11	Monitor Request Status. Values are as follows: • 0 = rest • 1 = *init-self* requested • 2 = *term-self* requested
12	Reserved.
13	Network Service Status. Values are as follows: • 0 = reset • 1 = init-self sent • 2 = active • 3 = quiesce pending • 4 = term-self sent
14	Secondary LU status. Values are as follows: • 0 = reset • 1 = initiation state • 2 = normal flow • 3 = quiesce shut down • 4 = terminate state
15	LU process identification number (pin)
16–49	Reserved.

Text Reference

The STATUS command also reports the status and state of an LU session. See the *SNA NRJE Node Manager's Guide* for more information on this command.

NRJERcvMsg

Reads one recent host remote console message.

Syntax

```
                 D            LA           I              I            IA
NRJERcvMsg ( ConsoleId, OutBuffer, MaxOutBuf, OutputLength, Result )
```

Parameters

ConsoleId **An input double-word integer variable**. The *ConsoleId* parameter is the remote host console identification. Obtain the value for this parameter by calling NRJEConsole before you call this intrinsic.

OutBuffer **An output logical array**. The *OutBuffer* array contains the text of a host remote console message.

OutBuffer is a byte array, and the *MaxOutBuf* and *OutputLength* parameters are byte counts.

NOTE The size of *OutBuffer* must be large enough to contain a host remote console message. If *OutBuffer* is not large enough, the message will be truncated.

MaxOutBuf **An input integer variable**. The *MaxOutBuf* parameter contains the maximum number of bytes that can be put into *OutBuffer*. 132 bytes is recommended.

OutputLength **An output integer variable**. The *OutputLength* parameter contains the actual number of bytes of the host remote console message contained in *OutBuffer*.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Call `NRJERcvMsg` to read one recent host console message. Obtain the *`ConsoleId`* by calling `NRJEConsole` first. Indicate the maximum byte length of the *`OutBuffer`* array in the *`MaxOutBuf`* parameter. Obtain the actual message length from the *`OutputLength`* parameter.

While you own the host remote console, you receive all recent host remote console messages that are directed to your workstation. These messages may be in response to commands other than those you issued.

Messages will return in the order they are received from the time that the user opens the console.

Use `NRJERcvMsg` with the following intrinsics:

- `NRJEConsCheck`: to determine whether any outstanding console messages are waiting to be received.
- `NRJERelease`: to relinquish ownership of the host remote console.

After `NRJERcvMsg` returns control to your program, test the first element of *`Result`*.

Text Reference

See the *SNA NRJE Node Manager's Guide* for a description of "Host Console Commands."

NRJERdrAccess

Enables or disables NRJE user access to the NRJE reader.

Syntax

```
                   BA      I        IA
NRJERdrAccess ( Wsid, OpCode, Result )
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

OpCode

An input integer. The *OpCode* parameter indicates the function to be performed. Its values are as follows:

- 1 = Enable NRJE user access to the NRJE reader.
- 0 = Disable NRJE user access to the NRJE reader.

Result

An eight-element integer output array (required). The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Call NRJERdrAccess to enable or disable NRJE user access to the NRJE reader. After NRJERdrAccess returns control to your program, test the first element of *Result*.

Text Reference

Two commands correspond to this intrinsic: RDRUP and RDRDOWN. See the *SNA NRJE Node Manager's Guide* for more information on these commands.

NRJERdrFence

Updates the outfence value of the NRJE workstation reader.

Syntax

```
                   BA        I           IA
NRJERdrFence ( Wsid, NewOutFence, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

NewOutFence **An input integer**. The *NewOutFence* parameter is the output fence for the workstation reader, from 0 through 14.

Any spooled file whose priority is equal to or greater than the value of *NewOutFence* is transmitted when the workstation is in communication with the host.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use the NRJERdrFence intrinsic to update the output fence value of the reader transmission queue by using the *NewOutFence* parameter. Call NRJEQueue to obtain the current output fence value.

When control is returned to your program, test the first element of *Result*.

Text Reference

The RDRFENCE command also updates the *outfence* value of the NRJE workstation reader. See the *SNA NRJE Node Manager's Guid*e for a discussion of the command, spool files, and output priorities.

NRJERelease

Relinquishes access to the host remote console of a workstation.

Syntax

```
             D         IA
NRJERelease ( ConsoleId, Result )
```

Parameters

ConsoleId **An input double-word integer variable**. The *ConsoleId* parameter is the remote host console identification. Obtain the value for this parameter by calling NRJEConsole before you call this intrinsic.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Call NRJERelease to relinquish access to the host remote console a workstation. Obtain the *ConsoleId* value by calling NRJEConsole first.

Use NRJERelease with the following intrinsics:

- NRJEConsole: to obtain access to the host remote console.
- NRJESendCmd: to send host remote console commands.
- NRJEConsCheck: to determine whether any outstanding host remote console messages are waiting to be received.
- NRJERcvMsg: to read one recent host console message.

These intrinsics are described in this chapter. After NRJERelease returns control to your program, test the first element of *Result*.

Text Reference

RELEASE also relinquishes access to the host remote console of a workstation. See the *SNA NRJE Node Manager's Guide* for more information on this command.

NRJESendCmd

Sends a remote host console command.

Syntax

```
                D          LA           I         IA
NRJESendCmd ( ConsoleId, CmdBuffer, CmdLength, Result )
```

Parameters

ConsoleId **An input double-word integer variable.** The *ConsoleId* parameter is the remote host console identification. Obtain the value for this parameter by calling NRJEConsole before you call this intrinsic.

CmdBuffer **An input logical array**. The *CmdBuffer* array contains the remote host console command text to be transmitted. The syntax of the command is not checked by NRJE. The text of the command must meet the standards of the host.

A remote host console command can be sent successfully but be in error.

CmdLength **An input integer variable**. The *CmdLength* parameter contains the number of characters in *CmdBuffer*. The maximum host console command length is 80 characters for all supported host job entry subsystems.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Obtain the *ConsoleId* value by calling NRJEConsole first. The *CmdBuffer* parameter must contain a valid remote host console command for your workstation. The *CmdLength* parameter must contain the command length in bytes. Use NRJESendCmd with the following intrinsics:

- NRJERcvMsg: to read one recent remote host console message.
- NRJEConsCheck: to determine whether any outstanding console messages are waiting to be received.
- NRJERelease: to relinquish access to the remote host console.

After NRJESendCmd returns control to your program, test the first element of *Result*.

Text Reference

See the *SNA NRJE Node Manager's Guide* for a description of "Host Console Commands."

NRJEStartWS

Initiates a workstation or LUs.

Syntax

```
                BA        I         BA         I          IA
NRJEStartWS ( Wsid, ChainSize, LUNames, LUNamesLen, Traces,
                  I              I            I         BA
              TraceFileLen, TraceMedium, TraceSize, TraceFile,
                 BA         IA
              DefaultFile, Result )
```

Parameters

Wsid **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

ChainSize **An input integer variable**. The *ChainSize* parameter contains the number of request units (RUs) to send in one SNA chain, from 1 through 99. See “Glossary” for a definition of RU. If the value of *ChainSize* is 0, the configured default value applies.

LUNames **An input byte array**. The *LUNames* array is a list of eight-character logical unit (LU) names. In MPE V, these are LU class names as configured in the SNA Configuration: Classes screen. In MPE XL, these are host names (not NAUs) of LUs that you wish to start. See “Glossary of Terms” for the definition of an LU.

Each entry is left-justified, and blank-filled. The number of entries is specified in *LUNamesLen*.

Following is an example of an *LUNames* array of four entries:

```
FIRSTLU
NEXTUNIT
NTHLU
LASTUNIT
```

You cannot start an *LUName* that is already started.

LUNamesLen **An input integer**. The *LUNamesLen* parameter specifies the number of entries in *LUNames*, from 0 through 16. The value of *LUNamesLen* for the *LUNames* example above is 4.

If the value of *LUNamesLen* is 0 and the workstation is inactive, the configured auto-start LU names are initiated.

If *LUNamesLen*=0 and the workstation is active, a *Result* will indicate an error.

The maximum *LUNamesLen* value is 16.

Traces **A two-element input integer array.** Each element of *Traces* contains an integer that represents a type of tracing to turn on or off.

The use of two elements allows you to enable multiple trace types simultaneously. The order of the elements is not significant. *Traces* element values are as follows:

- 0 = No tracing.
- 1 = NRJE LU process tracing.

NOTE Use this option only when recommended by your HP representative. This type of tracing can have a performance impact.

If LU tracing is initiated, it must be done when the workstation is started. Thereafter, it can be stopped and restarted.

- 2 = Intrinsic tracing.

TraceFileLen **An input integer**. The *TraceFileLen* parameter is the length in characters of the trace file designator, *TraceFile*.

The number of characters may be between one and thirty-five. If this parameter is zero, the trace is recorded in the *DefaultFile*.

TraceMedium **An input integer**. *TraceMedium* indicates the *TraceFile* device type. This parameter is called by reference. The only valid value for *TraceMedium* is 0, which indicates that the trace information will be written to a disk file.

TraceSize **An input integer**. *TraceSize* contains the maximum number of logical records *TraceFile* is to contain, from 0 through 32767.

A logical record is 128 words long. If *TraceSize* is zero, the default file size of 1024 records is used.

TraceFile **An input byte array**. The *TraceFile* byte array contains an actual file designator used for trace output. This parameter is used only when *TraceFileLen* is greater than zero. This parameter identifies a disk trace output file. *TraceFile* can contain a fully qualified 35-character disk file name, with lockword, in this form:

filename[lockword][groupname[acctname]]

The contents of *TraceFile* must be left-justified, with trailing blanks, and meet MPE standards for an actual file designator.

If you specify a disk file that does not exist (a NEW file), then it will be created as the destination for trace output.

If you specify an OLD file (it already exists), and it already is in use for trace output, then trace output records for this activity are interleaved with other trace output records. You minimize the use of disk space by using an OLD file appropriately; however, trace output can be overwritten. If you specify an OLD file that is not already open, the file will be overwritten with new trace data. No warning is issued. Trace file records are created after successful completion of the intrinsic. If an error is encountered during the execution of the intrinsic, only the status array will be traced.

DefaultFile **An output byte array**. The *DefaultFile* parameter is the file designator of the default trace file. A default file name is created whenever *TraceFileLen* is zero.

Default file names are in the form NMTC*nnnn*.PUB.SYS, where *nnnn* is a four-digit number that is incremented each time a default trace is created. Default file names are up to 27 bytes long. The last character is always a blank. You can always retain trace data by using a *DefaultFile*; however, considerable disk space might be used.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

The NRJEStartWS intrinsic initiates the NRJE workstation. It causes the NRJE monitor to be streamed and run as an MPE job. See the *SNA NRJE Node Manager's Guide* for more information.

When you call NRJEStartWS, you can start all LUs of a workstation that are configured for automatic starting (by setting *LUNamesLen* to zero), or you can supply a list of LUs in *LUNames* and set *LUNamesLen* to the length of the list. Call NRJELUList and NRJELUStatus to determine which LUs have started. You can initiate intrinsic or LU (internal or user) tracing with this intrinsic.

If you set *TraceFileLen* to zero, then a default trace file, NMTC*nnnn*.PUB.SYS, is created and identified in *DefaultFile*. You can always retain trace data with this method; however, considerable disk space might be used.

NOTE If an error occurs because of a bad trace parameter, the workstation or LUs will not start.

After you have called this intrinsic, test the first element of *Result* to determine its operation.

Text Reference

STARTWS is the command counterpart to this intrinsic. See the *SNA Node Manager's Guide* for additional information. Also see NRJECONTROL START in the same manual.

NRJEStopWS

Stops or halts LU-LU session or workstation.

Syntax

```
              BA     I        BA       I        IA
NRJEStopWS ( Wsid, NumLUs, LUNames, StopCode, Result )
```

Parameters

Wsid

An eight-character input byte array. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

NumLUs

An input integer. The *NumLUs* parameter specifies the number of entries in *LUNames*, from 0 through 16. The value of *NumLUs* for the *LUNames* example below is 4.

If the value of *NumLUs* is 0, the action indicated by *StopCode* applies to all active LUs.

LUNames

An input byte array. The *LUNames* array is a list of eight-character logical unit (LU) names. In MPE V, these are LU class names as configured in the SNA Configuration: Classes screen. In MPE XL, these are host names (not NAUs) of LUs that you wish to stop. See “Glossary” for the definition of an LU.

Each entry is eight alphanumeric characters long, begins with a letter, and is left-justified and blank-filled. The number of entries is specified in *NumLUs*.

Following is an example of an *LUNames* array of four entries:

```
FIRSTLU
NEXTUNIT
NTHLU
LASTUNIT
```

StopCode

An input integer. The *StopCode* indicates how the LU-LU sessions indicated in *LUNames* are to stop. Value can be one of the following:

- 0 = Stop the sessions in an orderly manner.
- 1 = Halt the sessions immediately.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

This intrinsic provides an orderly or an immediate way to stop selected LU-LU sessions or all active LU-LU sessions. Specify the number of LUs to be stopped in *NumLUs*. Specify each LU name to be stopped as an entry in *LUNames*. After *NRJEStopWS* returns control to your program, test the first element of *Result*.

Text Reference

The NRJE manager commands STOPWS and HALT stop a workstation. See the *SNA NRJE Node Manager’s Guide* for more information on these commands.

The MPE commands to stop a workstation are NRJECONTROL HALT< and NRJECONTROL STOP. See the *SNA NRJE Node Manager’s Guide* and the *SNA Link Services Reference Manual* or the *SNA Link/XL Node Manager’s Guide*.

NRJETrace

Initiates or terminates NRJE tracing.

Syntax

```
             BA       I         IA          I            I
NRJETrace ( Wsid, TraceOnOff, Traces, TraceFileLen, TraceMedium,
                I          BA         BA         IA
            TraceSize, TraceFile, DefaultFile, Result )
```

Parameter

Wsid — **An eight-character input byte array**. The *Wsid* parameter identifies your workstation. It must contain a left-justified alphanumeric name beginning with a letter. If the workstation identifier is less than eight characters long, it must be followed by blanks.

TraceOnOff — **An input integer**. The *TraceOnOff* parameter indicates whether to initiate or terminate the tracing specified in *Traces*. Values are as follows:

- 0 = Terminate the tracing.
- 1 = Initiate the tracing.

Traces — **A two-element input integer array**. Each element of *Traces* contains an integer that represents a type of tracing to turn on or off.

The use of two elements allows you to enable multiple trace types simultaneously. The order of entry of the elements is not significant. *Traces* element values are as follows:

- 0 = No trace.
- 1 = NRJE LU process tracing.

NOTE Use this option only when recommended by your HP representative. This type of tracing can have a performance impact.

If LU tracing is initiated, it must be done when the workstation is started. Thereafter, it can be stopped and restarted.

- 2 = Intrinsic trace. Intrinsic tracing records the execution of NRJE intrinsics. All intrinsics are traced.

Following is an example of *Traces* for LU tracing only:

```
1
0
```

TraceFileLen **An input integer**. The *TraceFileLen* parameter is the length in characters of the trace file designator, *TraceFile*.

The number of characters may be between one and thirty-five. If this parameter is zero, the trace is recorded in the *DefaultFile*.

TraceMedium **An input integer**. *TraceMedium* indicates the *TraceFile* device type. This parameter is called by reference. The only valid value for *TraceMedium* is 0, which indicates that the trace information will be written to a disk file.

TraceSize **An input single-word integer**. *TraceSize* contains the maximum number of logical records *TraceFile* is to contain, from 0 through 32767.

A logical record is 128 words long. If *TraceSize* is zero, the default size of 1024 records is used.

TraceFile **An input byte array**. The *TraceFile* byte array contains an actual file designator used for trace output. This parameter is used only when *TraceFileLen* is greater than zero. This parameter identifies a disk trace output file. *TraceFile* can contain a fully qualified 35-character disk file name, with lockword, in this form:

filename[lockword][groupname[acctname]]

The contents of *TraceFile* must be left-justified, with trailing blanks, and meet MPE standards for an actual file designator.

If you specify a disk file that does not exist (a NEW file), then it will be created as the destination for trace output.

If you specify an OLD file (it already exists), and it already is in use for trace output, then trace output records for this activity are interleaved with other trace output records. You minimize the use of disk space by using an OLD file appropriately; however, trace output can be overwritten. If you specify an OLD file that is not already open, the file will be overwritten with new trace data. No warning is issued. Trace file records are created after successful completion of the intrinsic. If an error is encountered during the execution of the intrinsic, only the status array will be traced.

DefaultFile **An output byte array**. The *DefaultFile* parameter is the file designator of the default trace file. A default file name is created whenever *TraceFileLen* is zero.

Default file names are in the form NMTC*nnnn*.PUB.SYS, where *nnnn* is a four-digit number that is incremented each time a default trace is created. Default file names are up to 27 bytes long. The last character is always a blank. You can always retain trace data by using a *DefaultFile*; however, considerable disk space might be used.

Result **An eight-element integer output array (required**). The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

NRJETrace starts and stops NRJE tracing. Specify trace types and trace file characteristics in the input parameters. Tracing can also be initiated when a STARTWS or TRACEON command is issued, or when an NRJEStartWS or another NRJETrace intrinsic is called. If you set *TraceFileLen* to zero, then a default trace file, NMTC*nnnn*.PUB.SYS, is created and identified in *DefaultFile*. You can always retain trace data this way; however, considerable disk space might be used.

This intrinsic examines all input parameters when *TraceOnOff*=1, but it examines *Traces* only when *TraceOnOff*=0, which stops tracing. When NRJETrace returns control to your program, test the first element of *Result*.

Text Reference

TRACEON and TRACEOFF are NRJE subsystem commands that turn trace on and off. See the *SNA NRJE Node Manager's Guide* for more information on these commands.

NRJECONTROL TRACEON and NRJECONTROL TRACEOFF are MPE commands that turn trace on and off. Also see the *SNA NRJE Node Manager's Guide* for more information.

NRJEUpdWel

Updates the NRJE subsystem welcome message.

Syntax

```
                BA        I         IA
NRJEUpdWel ( Message, MessageLen, Result )
```

Parameters

Message **An input byte array**. The *Message* parameter contains the message to be broadcast to any NRJE subsystem command interpreter user. The maximum *Message* length is 1920 characters.

Separate each line of text by a carriage return character and a line feed character. A carriage return is %15, or !0D; a line feed is %12, or !0A.

MessageLen **An input integer**. The *MessageLen* parameter contains the length of the text in the *Message* array.

The maximum *MessageLen* value is 1920 characters.

Result **An eight-element integer output array (required)**. The *Result* array contains error codes that occurred during execution of this intrinsic.

The first element of the *Result* array is set to zero if no errors took place. The structure of the *Result* array is described in “Parameters Common to NRJE Intrinsics” in the introduction to Chapter 5 , “User Intrinsics.”

Always test the first element of *Result* immediately after you call this intrinsic. If the first element of *Result* is not zero, test the other elements of *Result* to determine the nature of the problem that has occurred.

Description

Use the NRJEUpdWel intrinsic to revise the whole NRJE subsystem welcome message. The welcome message is broadcast every time any user or manager enters the subsystem; however, it is not broadcast when any user invokes the NRJE subsystem command to change from one default workstation to another.

The text of the welcome message is input in *Message*. The length of the welcome message text is input in *MessageLen*. After NRJEUpdWel returns control to your program, test the first element of *Result*.

Text Reference

The `WELCOME` command also updates the NRJE subsystem welcome message. See the *SNA NRJE Node Manager's Guide* for more information on this command.

The `NRJEWelcome` intrinsic retrieves the text of the welcome message. See Chapter 5 , “User Intrinsics.”

7 MPE Support Utilities and Commands

This chapter describes certain MPE support utilities and commands that can be used with NRJE:

- SPOOK (MPE V, and MPE XL releases prior to 2.1, only)
- MPE spooling commands (MPE XL release 2.1 and later).

Several other MPE commands are used in day-to-day NRJE operations:

- `LINKCONTROL` (MPE V only)
- `NRJECONTROL`
- `RESUMENMLOG`
- `SHOWNMLOG`
- `SNACONTROL`
- `SWITCHNMLOG`

The `NRJECONTROL` commands are described in the *SNA NRJE Node Manager's Guide*. The other commands are described in the S*NA Link Services Reference Manual* or the *SNA Link/XL Node Manager's Guide*.

SPOOK

If, after submitting jobs to NRJE readers, spooled files have not been transmitted to the host system, an HP 3000 user can use the SPOOK utility to alter the spooled files' priority or to purge the files. You also can manipulate output spool files. This utility is described in *MPE System Utilitie*s.

NOTE SPOOK exists only on MPE V and releases 1.1, 1.2, and 2.0 of MPE XL.

MPE Spooling Commands

Table 7-1 lists MPE commands that can be used to affect spooling.

Table 7-1 MPE Spooling Commands

MPE V, and MPE XL prior to release 2.1	MPE XL release 2.1 and later	
ALTSPOOLFILE	SPOOLF ;ALTER	Alters the characteristics of a spool file.
DELETESPOOLFILE	SPOOLF ;DELETE	Deletes a spool file.
OPENQ	OPENQ	Opens the spool queue for a specified logical device or device class.
OUTFENCE	OUTFENCE	Selects the priority level above which spool files can become active.
RESUMESPOOL	SPOOLER ;RESUME	Resumes suspended spooler output to a spooled device.
SHUTQ	SHUTQ	Closes the spool queue for a specified logical device or device class.
STARTSPOOL	SPOOLER ;START	Initiates spooling of a device.
STOPSPOOL	SPOOLER ;STOP	Terminates spooling to a specified device.
SUSPENDSPOOL	SPOOLER ;SUSPEND	Suspends output to a spooled device.

These commands are executed from the HP 3000 system console. They are described in the *Console Operator's Guide*.

A Parsing Algoithms and User Exit Procedures

These procedures are supplied with SNA NRJE and are required for the full job and output management capabilities of NRJE to function.

Parsing Algorithms

To route output properly, SNA NRJE must parse both “Job Received” messages (such as the $HASP100 message from JES2) and host system output banner pages of the received data sets. This capability is necessary for NRJE to be able to interpret the host system job number. This chapter describes the algorithms used to perform the parsing. The algorithms work for all host messages and banners which have not been modified by host system programmers. If nonstandard “Job Received” messages or host banners are sent to your workstation, an exit procedure might be needed to parse for the necessary information. The user exits and exit procedure specifications are described in this chapter.

NOTE

If you are connected to a JES3 host system, NRJE does not use an internal banner decode routine. For both print and punch output, the job number is returned from the host in the spin number field of the PDIR.

VSE/POWER does not send “Job Received” messages.

If a user-written banner decode exit procedure is not in use, these rules apply for host banner translation:

- **Print Output**

  If the configured Language ID is 0 to 255, NRJE translates the banner by using the EBCDIC-to-ASCII translation table for the indicated language. If set to BINARY (-1 translation code), NRJE assumes it is an EBCDIC banner and will attempt to parse the banner accordingly. Printer banners are always written to the output file, translated according to the configuration for the writer. (If “no translation,” the banner is written without translating.)

- **Punch Output**

  NRJE attempts to parse only those records which are preceded by a *Job Separator PDIR* (PDIR byte 3 = 1). For additional information, see “Peripheral Device Information Record” in Chapter 4 , “Job Output.” The records are assumed to be in Hollerith regardless of the language ID configuration field for the logical writer. You can configure NRJE to assume either Hollerith or EBCDIC banners on the NMMGR Workstation Data Page 2 screen. If NRJE fails to obtain a host job number from the banner, it translates the banner card according to the language ID configuration for the writer. Punch banners are always written to the output file, unless the output was received on reserved form CMD. NRJE will not print the punch banner for output directed to the *formid* CMD, even if the output was redirected by the user with the SUBMIT command.

“Job Received Message” Parsing Algorithm

The algorithms used by NRJE to parse JES2 or JES3 “Job Received” console messages are shown below. Output variables are *JobNumber*, *ReaderNumber*, and *JobName*.

JES2 Algorithm

```
scan buffer for host'command'char;
if found and host'command'char followed by "HASP100" then begin
      scan from buffer(0) for keyword "JOB"
      skip trailing blanks;
      move to JobNumber variable while numeric;
      skip all blanks trailing "HASP100" string;
      move 8 characters to JobName variable;
      scan from jobname'position for period;
      skip two characters and assign ReaderNumber variable;
    end;
```

JES3 Algorithm

```
begin
  scan for "IAT6101";
  if found scan for "JOB";
  if found skip trailing blanks;
  move to JobNumber while numeric;
  scan for string " IS ";
  skip trailing blanks;
  move to JobName for 8 characters;
  ReaderNumber := 0;        ! They are not in JES3 messages.
end;
```

NOTE If a “Job Received” message algorithm detects an IAT6101 message, NRJE checks each active reader for a match with *JobName* to associate the *JobNumber* with the correct job. If two jobs are active on two readers at the same time with the same *JobName*, correct output routing cannot be guaranteed.

VSE/Power

Power does not send “Job Received” messages.

Banner Decode Parsing Algorithms

The algorithms used by NRJE to parse JES2 output banner pages are shown below. JES3 obtains host job numbers from the spin number field of the PDIR. No attempt is made by NRJE to decode output banners from VSE/POWER.

JES2 Algorithm

```
scan buffer for one of three keywords: "JOB" or "STC" or "TSU";
if found then skip trailing blanks;
if "JOB" found then
   move while numeric to JobNumber variable
else
   if "STC" or "TSU" found then
     JobNumber := "-1";
```

NOTE

A *JobNumber* value of -1 indicates that the output is unsolicited and that the Job Log should not be checked for routing information.

Punch Banner Decode Algorithms

The algorithm used by NRJE to parse punch output banners for data sets returning from JES2 is shown below.

```
Translate first line of output using Hollerith table:
     %160 (hex 70):               to ASCII "0";
     %221 - %231 (hex 91 to 99):  to ASCII digits "1" through "9"
     all other characters:        to ASCII blank " ";
begin
  translate from Hollerith to ASCII;
  skip leading blanks;
  for digit'count 0 to 7 do
    begin
      outnum(digit'count) := this'char;
      scan while this'char;
      skip trailing blanks;
      digit'count := digit'count + 1;
    end;         { For digits 0 to 7}
  if digit'count =4
    then offset := digit'count - 4
  move jobnumber  := outnum(offset) , (4 bytes);
end;             { Parse punch banner }
```

User Exits

Two exits have been added to SNA NRJE to facilitate job management and output management operation. One exit calls a customer's procedure that decodes the host banner (separator page) and informs NRJE of the host job number for the data set. A separate exit procedure can be specified for each logical writer.

A second exit passes all console data to a customer's procedure that parses the host "Job Received" messages sent by the host to the remote console when a job is submitted.

NOTE If an exit procedure is not specified, NRJE uses its own internal parsing algorithms to decode banners and "Job Received" messages. The exit procedures are not necessary unless the host has been modified to send nonstandard banners or console messages.

Configuration

These two steps must be performed to configure SNA NRJE to access an exit procedure:

1. Install the procedure segment from the USL file into `SL.NRJE.SYS`.
2. Identify the name of the console exit procedure in the NMMGR Workstation Data Page 2 screen and the banner decode procedure(s) in the NMMGR Writer Data Screen. These exit procedures are configured differently. Configuring a banner exit as a console exit can cause a system failure.

CAUTION NRJE exits grant privileged mode access to system resources. Improper installation and configuration of user exits can cause a system failure.

Banner Decode Exit Procedure Specification

A description of the exit procedure to be used to parse host output banner pages is given below.

NOTE In NRJE/XL, the Banner Decode Exit Procedure is called the Writer Decode Exit Procedure Specification.

Syntax

```
                           LA          BA         I        I
Procedure NRJEBANNER(UserWorkArea, DataLine, LineLength, Cctl,
                           I         BA       I       I
                       RecordType, Jobnum, Filenum, Action)
```

NOTE The procedure name is identified in the NMMGR Writer Data screen for the applicable writer and need not necessarily be called NRJEBANNER.

Parameters

UserWorkArea **A logical array for input/output** (of 100 words). *UserWorkArea* can be used for global storage; its size is from 0 through 99. This parameter is initialized to zeros by NRJE, but it is not subsequently modified by NRJE.

DataLine **An input byte array** containing one logical record from the data set. This array will be decompressed and translated according to the configuration of the logical writer receiving the data set. For logical printers, the carriage control character is passed separately; it is not prefixed to the data line.

LineLength **A positive input integer variable** (passed by value) indicating the number of bytes in *DataLine*. This does not include the carriage control character for print data sets, which is passed separately. A `0` indicates a blank line, and a `-1` indicates that the end of the data set has been received.

Cctl **Input integer variable passed by value**. This is the carriage control character for print data sets; it is set to zero for punch data sets.

RecordType **Input integer variable passed by value**. This parameter is set to zero if *DataLine* contains a line of text from the output data set. If *DataLine* contains a PDIR, *RecordType* is set to one.

Jobnum **An output byte array** (of 4 bytes). This array's size is from zero through three ASCII characters. *Jobnum* is initialized to ASCII blanks prior to each call of the exit procedure. If successful in decoding the host banner, the exit procedure is to return the job number in ASCII, left-justified, padded with blanks if necessary. If the exit procedure detects a banner indicating the output is from a "Started Task" (denoted by "STC" in the banner) or a "Timesharing User" (denoted by "TSU"), the procedure should indicate a job number consisting of these ASCII characters: minus, one, blank, and blank. This indicates that the banner decoding was successful, but NRJE should not check the Job Log for output routing information. This is because all "TSU" and "STC" jobs are unsolicited, and valid output routing information cannot be found in the Job Log for this job.

Filenum **Output integer parameter** (passed by reference). Used only if the procedure will open the output file for NRJE rather than return a job number. (If *Jobnum* is returned as non-blank and *Filenum* is returned as non-zero, *Jobnum* is ignored and output is routed to *Filenum*.) *Filenum* is initialized to zero by NRJE prior to each call to the exit procedure.

Action **Output integer**. *Action* is initialized to zero by NRJE prior to each call to the exit procedure. It is read only if the exit procedure sets a value for *Jobnum* or *Filenum*. If zero is returned, NRJE writes the saved banner to the output file. If *Action* is set to any other value, NRJE will not write the saved banner.

Description

The exit procedure name is specified in the NMMGR Writer Data screen. A different procedure can be accessed by each logical writer.

Note that the banner decode exit procedure is optional, and need not be specified unless NRJE's internal banner decode algorithm and output routing mechanism are not sufficient.

Refer to "Banner Decode Parsing Algorithms," earlier in this chapter, for additional information.

The exit procedure can implement any one of three functions:

1. Provide the host job number by decoding the data set banner. This enables NRJE to check the Job Log for output disposition information for the job.

2. Open the output file for NRJE by returning *Filenum* rather than *Jobnum*. NRJE then writes the data set to the file.
3. Open and write the entire data set to the output file. This is accomplished by not setting either of the output parms (*Filenum* or *Jobnum*), which causes NRJE to pass the entire data set to the procedure. FWRITE error handling is then the responsibility of the procedure.

The exit procedure will be passed each record of a received data set, beginning with the PDIR, until the procedure returns a non-blank value for *Jobnum*, a non-zero *Jobnum*, or a non-zero value for *Filenum*. If either value is returned by NRJE, NRJE will not call the exit procedure again until the next data set is received, except for this condition:

If a PDIR is received for which:

- FORM name has changed, or
- FCB name has changed, or
- number of copies has changed, or
- number of copies is not 0,

then the exit procedure will be called again. If the procedure returns a job number or file number to NRJE, the original file will be closed and the new output file will be used.

If the *DataLine* passed is a data record, the data line will be decompressed and translated as specified by the logical writer receiving the data set. If the received *DataLine* is a PDIR, the first four bytes will be passed without translation, and the remaining bytes in the record will be translated using the Native-3000 EBCDIC-to-ASCII translation table.

NRJE will save each banner line in an internal data structure, up to a maximum of 1024 lines.

If *Jobnum* is not blank, nor is it the “minus, one, blank, blank” ASCII character string, NRJE accesses the Job Log to determine the proper output destination. The saved banner lines are printed if *Action* is returned with a value of zero; otherwise, the saved banner lines are not printed. Allowing for the one exception described above, no subsequent calls are made to the exit procedure until the next data set is received.

If *Filenum* is not zero, NRJE writes the data set to the file opened by the exit procedure. If *Action* is returned with a value of zero, the saved banner lines are printed; otherwise, the saved banner lines are not printed.

If both *Jobnum* and *Filenum* are returned by the exit procedure, NRJE uses *Filenum* to route the data set and ignores *Jobnum*. *Action* is interpreted as described above.

If neither *Filenum* nor *Jobnum* is returned, NRJE passes the entire data set to the procedure and logs a warning message. NRJE assumes that the exit procedure is handling all output routing and writing operations.

If an exit procedure opens an output file by using the COMMAND intrinsic to execute a file equation, and then uses an FOPEN that back-references the file equation, it should also issue a RESET (using the COMMAND intrinsic) to remove the file equation before returning to NRJE. This is required to minimize the possibility of NRJE misrouting output on other writers.

For “End of Data Set” calls to the exit procedure (line length of -1), the returned values for *Filenum* and *Jobnum* are ignored.

This hierarchy is used to resolve any routing specification conflicts:

> Banner Decode *Filenum* return
> overrides
> Job Log routing information (SUBMIT command parms)
> overrides
> Lookup Table form-to-file mapping (user’s JCL “FORM” specification)
> overrides
> Defaults.

The procedure will be called by the SNA NRJE Logical Unit, which will be running with traps on, in user mode. The NRJELU is “prepped” with PM, PH, DS, MR, IA, and BA capabilities.

An SPL example of a banner decode exit procedure for use with JES2 banners is shown below. This example is also in the SAMPEXIT sample file in NRJE.SYS.

Example

```
procedure get'jobnumber(work'area,record,length,cctl,type,jobnumber,
                        file'num,action);
value length,cctl,type;

logical array work'area;        ! In, scratch area.

byte array record;              ! In, a line of data (or a PDIR) from
                                ! the host.

integer  length,                ! In, num bytes in record.

         cctl,                  ! In, carriage control value.

         type;                  ! In, if 0 record is data, if 1 record
                                ! is a PDIR.

byte array job number;          ! Out, the host job number.

integer    file'num,            ! Out, we won't set this.

           action;              ! Out, also not set.
###########################################################
!
! This procedure will decode a JES2 host banner and return the job
! number.  We do not open the output file here but let the NRJE LU
! handle the fopen and fwrites to the output file.
! If we received an "STC" or "TSU" job we return a job number of "-1"
! to cause NRJE to suppress its search of the Job Log (such data sets
! are unsolicited and will not have an entry in the job log).
!
! The banner we are checking looks something like this:
!
! *A START JOB nnnn jobname   . . . misc. other information
!          |
!          |--------- JOB'STRING'POS
!
! The first byte is byte "0", JOB'STRING'POS is a byte offset of
! the string "JOB" or "STC" or "TSU" which is followed by one or
! more blanks and then the job number.
!
############################################################
end of comment;

begin             ! Local declarations.

equate JOB'STRING'POS = 9,
       DATA'RECORD    = 0;

byte pointer bptr;
```

```
< ********************* Begin Get'Jobnumber ******************* >

if type = DATA'RECORD then
  begin
    if record(JOB'STRING'POS) = "JOB" then
      begin                        ! A hit.
        scan record(JOB'STRING'POS + 3) while "  ",1;
        @bptr := tos;              ! Skip any leading blanks.
        move job number := bptr while N;    ! Move while numeric.
      end                          ! If "JOB" found.
    else
      if record(JOB'STRING'POS) = "STC" or   ! Check for Started Task
         record(JOB'STRING'POS) = "TSU" then ! and Time Sharing User.
        move job number := "-1  ";
  end;                             ! If data record received.

! Else if PDIR record we'll just ignore it.

end;              ! Get'Jobnumber
```

"Job Received" Message Exit Procedure

A description of the exit procedure to parse the console stream for a "Job Received" message is shown below; the message is `HASP100` for JES2 and `IAT6101` for JES3.

Syntax

```
                         LA             BA         I          I
Integer Procedure CON(UserWorkArea, ConsoleRecord, Length, ReaderNum,
                          BA      BA
                        Jobnum, JobName)
```

Function Return:

Integer (-32768 through 32767). Procedure should return 1 if a "Job Received" message is detected; otherwise, the procedure should return 0.

Parameters

UserWorkArea

Logical array for input/output (of 100 words). This array is initialized to zeros by NRJE, but it is not subsequently modified by NRJE. *UserWorkArea* can be used for global storage.

ConsoleRecord

An input byte array containing a console message from the host. *ConsoleRecord* is decompressed and translated into ASCII, using the Native-3000 EBCDIC-to-ASCII translation table.

Length

Input (positive) integer (passed by value). This contains the number of bytes in *ConsoleRecord*.

ReaderNum

An output integer value (passed by reference). The number of the logical reader (1 through 7) for which the "Job Received" message was received. The exit procedure is responsible for translating the ASCII value of the reader number into an integer value, using the `BINARY` intrinsic. If the host subsystem is JES3, this value should be set to 0, because JES3 "Job Received" messages do not contain a reader number.

Jobnum

An output byte array (of 4 bytes). *Jobnum* is initialized to blanks by NRJE prior to each call to the exit procedure. If *ConsoleRecord* contains the “Job Received” message from the host, *Jobnum* is to be returned here, left-justified (four bytes maximum).

JobName

Output byte array (of 8 bytes). *JobName* is initialized to blanks prior to each call to the exit procedure. If the procedure detects a “Job Received” message, it is to return the name of the job to *JobName*, left-justified.

Description

The procedure is identified in the NMMGR Workstation Data screen. If the procedure is not specified, an internal algorithm is used to parse the message. This algorithm will work unless the host job entry subsystem has been modified.

If the procedure detects a “Job Received” message from the host, it is to return the integer value of 1 and the corresponding job number and job name. Any other function return value (≠ 1) is ignored by NRJE. The procedure is called by the NRJELU in user mode, with traps on. The NRJELU is prepped with IA, BA, DS, MR, PH, and PM capabilities. An example of a console exit procedure in SPL to parse a JES2 HASP100 message is shown below.

Example

```
integer procedure parse'jes2'console(work'area,record,length,
                                     rdr'num,job number,jobname);
value length;

logical array work'area;        ! In, scratch area.

byte array record;              ! In, the console record.

integer length,                 ! In, positive byte count of record.

      rdr'num;            ! Out, range 1 to 7 if HASP100
! detected.

byte array job number,          ! Out, set if HASP100 detected.

           jobname;             ! Out, set if HASP100 detected.

comment
```

```
############################################################
!
! Scans the console record from the host and checks for whether a
! HASP100 (Job Received message) was received from the host.  Returns
! reader number, job number and jobname if HASP100 detected.
!
! The format of the console message is as follows:
!
! hh.mm.ss JOB nnnn $HASP100 jobname   ON Rxx.RDy  username
!   JNUM'POS --|    |        |                  |
!   HASP'POS -------|        |                  |
!   JNAME'POS ---------------|                  |
!   RDR'POS ------------------------------------|
!
! All "POS" offsets are byte offset equates.  The first byte is
! byte "0".
!
############################################################
end of comment;


begin             ! Local declarations

equate  CR          = %15,    ! Carriage return; scan terminator
        BLANK       = %6440,  ! Blank and CR, for scanning.
        ON'RDR'MSG = 1,       ! Function return if HASP100 detected.
        HASP'POS    = 18,     ! Location of $HASP100
        JNAME'POS   = 27,     ! Where jobname should be located.
        JNUM'POS    = 13,     ! Location of job number.
        RDR'POS     = 45;     ! Location of the reader number.

byte pointer bptr;

byte array hold(*) = work'area;

define hold'byte = hold(0)#; ! Temp storage for last char of
                             ! record(length).
intrinsic binary;

< **************** Begin Parse'JES2'Console ******************** >

parse'jes2'console := 0;       ! Initialize

if length < RDR'POS then       ! Don't bother to check if it's not
  return;                      ! long enough.

hold'byte := record(length);   ! We don't know what's here, so we'll
record(length) := CR;          ! save it and replace it with a known
                               ! character for scanning.

! Scan for the host command char and string "HASP100":

if record(HASP'POS) = "$HASP100" then
```

```
  begin                                   ! It's a hit!

    ! Need to skip leading blanks, if any, so that what we return is
    ! left justified.

    scan record(JNAME'POS) while BLANK,1;
    @bptr := tos;
    move jobname := bptr,(scan bptr until "  "); ! Move until blank.
    scan record(JNUM'POS) while BLANK,1;
    @bptr := tos;
    move job number := bptr while N;       ! Move while numeric.
    rdr'num := binary(record(RDR'POS),1);
    parse'jes2'console := ON'RDR'MSG;
  end;

! Now put that "hold" byte back:

record(length) := hold'byte;

end;              ! Parse'JES2'Console.
```

Glossary

A

ASCII American Standards Committee on Information Interchange. A standard used by computers for interpreting binary numbers as characters.

B

Batch communications: A form of data communications in which a facility collects data over a period of time and then submits collected information to a host computer. Typical batch communication consists of large amounts of information transmitted at infrequent intervals.

BIND: An SNA command sent by the host to NRJE. This command specifies the detailed protocol that NRJE accepts before initiating an LU-LU session.

C

Chain: A sequence of RUs constituting a recoverable entity.

Communications controller: A front-end processor which provides an interface between the communications facilities and a computer. IBM provides programmable and non-programmable communications controllers. An INP is a communications controller.

Console: A logical device through which commands are transmitted to the host job entry system, and responses (messages generated by the host) are received by NRJE.

D

Data set: In IBM terminology, any file.

DD: An IBM JCL command defining a data set.

F

FM header: An SNA Function Management header. These headers select destinations or sources for subsequent data transmission and supply information for other data management tasks.

Formid: A special form identifier for output data sets, which is used as a key into a Lookup Table of file equations. A formid is a string of eight characters or less. It is composed of alphanumeric characters, IBM National characters (#, $, or @), or a period (.). A formid reference, as in a `SUBMIT` command, must be delimited by single quote marks; for example, *'formid'*.

H

Half-session: In SNA, a component that provides FMD services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU).

Host: A host system is a central computer which provides services for other computers and terminals attached to it.

I

INIT-SELF: An SNA request to the SSCP to initiate an LU-LU session.

Intrinsic: A subprogram provided by Hewlett-Packard systems. These subprograms perform common functions such as opening files, opening communications lines, or transmitting data over a communications line.

J

JCL: IBM job control language. JCL is used to tell the operating system the names and locations of programs and the locations of data for those programs.

JECL: IBM job entry control language. JECL statements are used for job control under the VSE/POWER subsystem. These statements define a job to VSE/POWER and specify how job input and output are to be processed.

JES2: A common IBM job entry subsystem used under the MVS operating system.

JES3: A common IBM job entry subsystem used under the MVS operating system.

Job: A file to be transmitted to a host computer, to be executed as a job.

Job entry subsystem: An IBM subsystem which provides a gateway for the execution of jobs in a batch environment; for example, JES2.

Job management: The ability to specify the final destination of output data sets at the time a job is submitted to NRJE to be transmitted. For NRJE, this capability is enabled by the `PRINT`, `PUNCH`, and `FORMS` options of the `SUBMIT` command.

“Job Received” message: A message returned from the host system acknowledging that a job submitted through NRJE is being received at the host. For JES2 hosts it is the `$HASP100` “ON READER” message. For JES3 it is the `IAT6101` message. VSE/POWER does not acknowledge job reception to remote sites.

L

Ldev: A logical device; a disk file, a set of disk files, or a physical device. An HP 3000 ldev is identified by a number or a device class name.

Ldn: Logical device number. *See* **ldev**.

Logical device: *See* **ldev**.

Logical printer: A logical printer receives line printer images from a host system.

Logical punch: A logical punch receives card punch images from a host system.

Logical reader: A logical reader transmits card punch images to a host system. (*See Also* **Virtual reader**)

Logical unit: *See* **LU**.

Logical writer: A logical writer is a logical printer or a logical punch.

LU: An SNA logical unit. An LU is the SNA entity through which application data is transmitted within an SNA network. An LU provides access to an SNA network for an end user. It formats message units, displays information, and handles error recovery. Multiple LUs can reside in an SNA node.

LU class: A set of LUs on an HP 3000. NRJE/V will identify individual LUs only if they have separate class names.

M

Mainframe: A mainframe is a computer that generally has a large amount of memory and operates at a high speed.

N

NAU: Network addressable unit. The NAU represents the source and destination of data in a network. A communications session must exist before data can be transferred. Three types of NAUs exist: SSCPs, PUs, and LUs. Sessions are established by the SSCP,

NCP: Network control program. This program supports SNA communications on the IBM 3705 communications controller.

NM: Node management. This is an MPE user capability required for NRJE manager tasks.

NSPE: Network services procedure error. An SNA request from the SSCP to the LU indicating an error during session startup or shutdown. On receipt of this message, NRJE logs the content and terminates the LU session.

O

OP: MPE system supervisor capability.

Operating system: An operating system is a master control program for a computer.

Outfence: The priority value at which, or below which, jobs or line printer output will be deferred. Also, the MPE console command which sets the outfence.

Output management: The ability to route output by using special form identifiers (formids) in the job JCL. The formids are used to key into a Lookup Table that maps the formid to a file equation identifying the final destination of the output.

P

Physical unit: *See* **PU**.

Programmatic access

Programmatic access is a design feature of some systems which allows you to use the system's features from programs you write. SNA NRJE has both command access and programmatic access.

PU: A component of an SNA node. A PU controls the physical resources of a node and reports errors and physical failures to the SSCP. One PU resides in each node. A PU is the access method in a host node, the NCP in the communications controller node, and the hardware and software in peripheral nodes.

R

Remote: A workstation that is not local to the host system, and is configured under the host job entry subsystem. A remote can be connected to its host by communications facilities or by direct attachment.

RH: Request or response header. These are attached to SNA RUs to control data flow and support other network services.

RU: Request or response unit.

S

SDLC: Synchronous Data Link Control. SDLC is the data link level communications protocol used in SNA.

SIGNAL: An SNA request primarily used to interrupt an LU, to send higher priority data to that LU.

SNA: Systems Network Architecture, a comprehensive specification for distributed data processing developed by IBM. SNA defines a layered protocol for communicating and controlling a communications network within the IBM environment.

SNA NRJE: HP’s Systems Network Architecture Network Remote Job Entry. This HP product emulates the RJE product on an IBM 8100 minicomputer running the DPPX (Distributed Processing Programming Executive) Operating System. This emulation enables users on the HP 3000 to submit batch jobs, through an SNA network, to an IBM host or compatible mainframe for processing. The host can then send the output back to the HP 3000 for printing or storing on disk.

Spooled reader: The MPE ldev for holding jobs to be sent to the host for a given NRJE workstation.

SSCP: System services control point. An SSCP manages nodes within its domain. An SSCP stores status information for personnel to maintain a network. It exists only in the host node and is implemented by the communications access method

T

TERM-SELF: An SNA request to the SSCP to terminate an LU-LU session.

TH: Transmission header. A transmission header is used by the Transmission Control and Path Control layers for routing and sequencing data sent through an SNA network.

Transmission queue: A queue of jobs to be transmitted to the host; spooled reader.

V

Virtual reader: A logical device configured into the operating system that is used to queue files waiting for transmission to the host on a logical reader. (*See Also* **Logical reader**).

VSE/POWER: A common IBM subsystem used to control the input and output of jobs to the VSE operating system environment.

VTAM: Virtual Telecommunications Access Method. An IBM communications access method which implements SNA on the host.

W

Workstation: An intelligent terminal, or a collection of input/output devices managed by a computer or a control unit. An SNA NRJE workstation is a collection of LUs communicating with a host job entry subsystem, transmitting from a common spooled reader.

Index

Index

Index