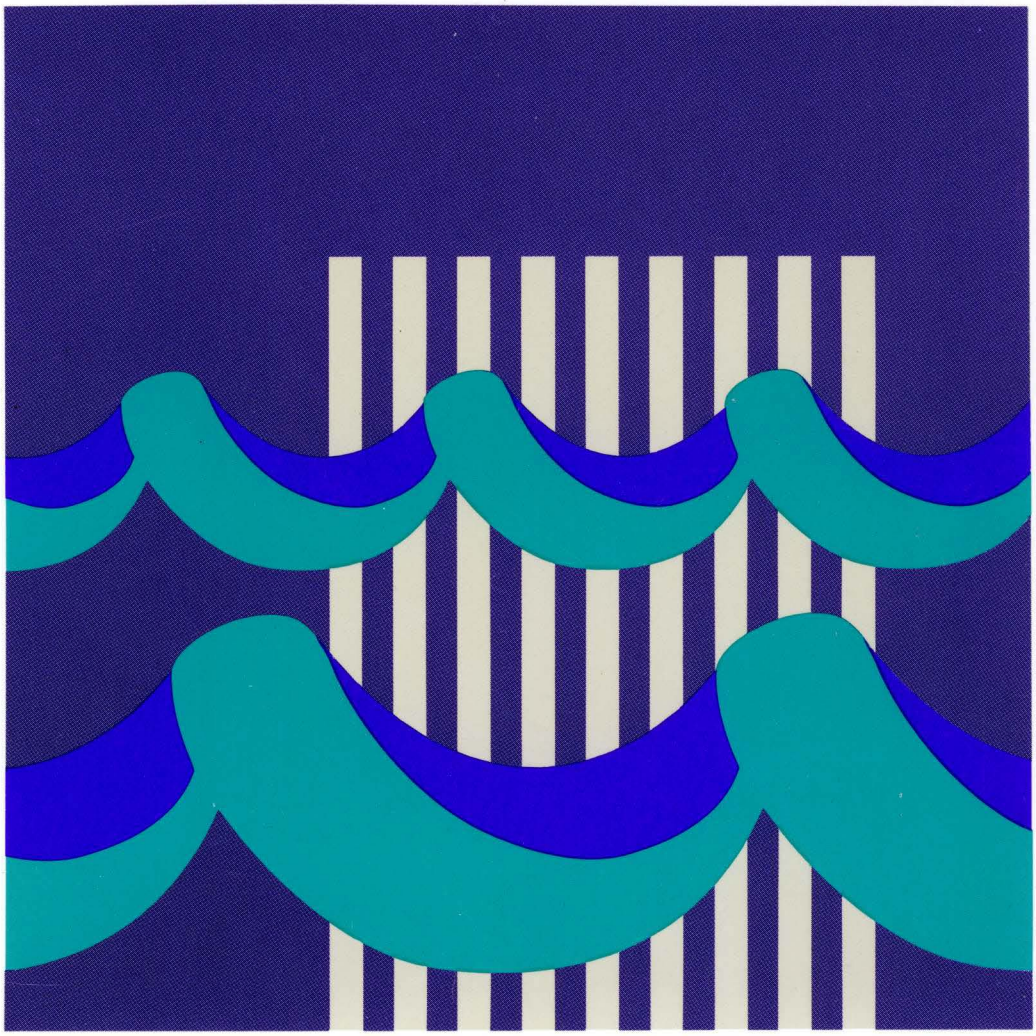
Network Control Program
System Support Programs
Emulation Program

# Generation and Loading Guide

SC30-3348-4



Network Control Program
System Support Programs
Emulation Program

SC30-3348-4

# Generation and Loading Guide

**Fifth Edition (June 1991)**

This major revision replaces SC30-3348-3. This edition applies to the following IBM licensed programs:

- Advanced Communications Function for Network Control Program

  Version 3 (program number 5667-124)
  Version 4 (program number 5668-854) Releases 1, 2, 2 Feature, 3, and 3.1
  Version 4 Subset (program number 5668-754)
  Version 5 (program number 5668-738) Releases 1, 2, 2.1, 3, 3.1, and 4.

- Advanced Communications Function for System Support Programs

  Version 2 (program number 5735-XXA) Release 2
  Version 3 for MVS (program number 5665-338), for VM (program number 5664-289), and for VSE (program number 5666-322): Releases 1, 2, 3 (MVS only), 4, 4.1, 5, 5.1 (MVS only), and 6 (MVS and VM only).

- Emulation Program for IBM Communication Controllers (program number 5735-XXB) Releases 5, 6, 6.1, 7, and 9.

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Department E15
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

# Special Notices

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement. Changes are made periodically to the information herein; before you use this document in connection with the operation of IBM systems, consult the latest *IBM System/370, 30XX, 4300, and 9370 Processors Bibliography*, GC20-0001, for the editions that are applicable and current.

Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's program may be used.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send inquiries, in writing, to the IBM Director of Commercial Relations, International Business Machines Corporation, Purchase, New York, 10577.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

Some trademarks of IBM Corporation in the United States and/or other countries appear in this document. These IBM trademarks are:

| | | | | |
|---|---|---|---|---|
| IBM | MVS/XA | VM/ESA | VM/XA | VTAM |

# Contents

# Figures

# Tables

# About This Book

This book helps you generate and load Network Control Program (NCP) and Emulation Program (EP). It contains information for generating and loading under MVS, VM, and VSE operating systems.

## Who Should Use This Book

This book is for system programmers who generate and load NCP.

Before using this book, you must be familiar with:

- Systems Network Architecture (SNA) and the functions NCP provides in an SNA network
- Your communication controller
- Your access method
- Your operating system.

## How to Use This Book

This book helps you understand the generation and loading procedures and helps you determine the control statements you need to generate and load your NCP.

Before you generate and load NCP, you must define the resources in your network to NCP. See *NCP, SSP, and EP Resource Definition Guide* for detailed explanations of definition statements and their keywords, and use *NCP, SSP, and EP Resource Definition Reference* to find out how to code them.

When you are ready to generate and load your NCP, locate the part of this book that covers the operating system under which you are generating and loading. The part of the book covering your operating system gives you information on the generation and loading procedures, tells you which control statements you need, and provides examples of control statements needed to generate and load your NCP.

## Terms Used in This Book

### New Terms

This book uses several new terms to replace terms used in previous versions of NCP. The following is a list of these new terms and the terms they replace.

| This term | Replaces this term |
|---|---|
| definition statement | macro |
| keyword | operand |

## How "Network" Is Used

The term *network* has at least two meanings. A *public network* is a network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet-switched, and leased-circuit services to the public. A *user-application network* is a configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. *Network*, as used in this book, refers to a user-application network.

## How "BER" Is Used

The abbreviation *BER* stands for *box event record* and *box error record.*

## How "VM" Is Used

In general, the term *VM* means the VM/SP, VM/XA,* and VM/ESA* systems in the CMS environment. However, if the book discusses information applicable to only one system, the specific system name (VM/SP, VM/XA, or VM/ESA) is used.

## How "ACF" Is Used

The following abbreviations refer to the associated Advanced Communications Function (ACF) licensed programs. An abbreviation without an *ACF/* preceding it should be assumed to be the ACF version. However, where it is necessary to distinguish between an ACF and a non-ACF version of one of these products, the book will make this distinction clear.

| Licensed Program | Abbreviation |
|---|---|
| Advanced Communications Function for Network Control Program | NCP |
| Advanced Communications Function for System Support Programs | SSP |
| Advanced Communications Function for Virtual Telecommunications Access Method | VTAM* |
| Advanced Communications Function for Telecommunications Access Method | TCAM |

## How "IBM Special Products or User-Written Code" Is Used

This book sometimes refers to *IBM special products or user-written code.* This phrase means IBM* special products such as Network Terminal Option (NTO), Network Routing Facility (NRF), and X.25 NCP Packet Switching Interface (NPSI), or user-written code.

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

## How IBM 3745 Communication Controller Model Numbers Are Used

In this book, the term *IBM 3745 Communication Controller* refers to all IBM 3745 models (IBM 3745-130, 3745-150, 3745-170, 3745-210, and 410), unless the text specifically notes a difference. Where the IBM 3745-130, 3745-150, and 3745-170 Communication Controllers are discussed but a briefer term is needed, such as in a table, they are referred to as the IBM 3745-1xx.

## How Numbers Are Written

This book shows numbers over 9999 in metric style, which means that a space is used instead of a comma to separate groups of three digits. For example, the number ten thousand five hundred fifty-two is written 10 552. However, if the number is a keyword value, for example, SALIMIT=65535, it does not include a blank.

## What Is New in This Book

In addition to editorial changes, technical corrections, and general clarifications, this edition contains new information about the following topics:

*Table 1. Location of new information*

| New information | Location |
|---|---|
| Storage available in MVS/XA* and VM/XA environments | "Defining Virtual Storage" on page 13 (for MVS); "Defining Virtual Storage" on page 70 (for VM) |
| Additional information on NEWDEFN and VTAMLST data sets and files | Chapter 1, "Generating the Program under MVS"; Chapter 4, "Generating the Program under VM"; and Chapter 7, "Generating the Program under VSE" |

## Where to Find More Information

The following list shows all of the books in the NCP V5 library, arranged according to related tasks. For more information on related publications, see "Bibliography" on page 199.

**Note:** These NCP manuals describe EP R5, R6, R6.1, R7, and R9.[1] For more information about earlier EP releases, refer to the appropriate EP manuals.

### The NCP V5 Library

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

1 EP R8 is a stand-alone release for the IBM 3745 and is documented in *Emulation Program Resource Definition and Diagnosis*.

**Education**

| | |
|---|---|
| GC31-6815 | *Bibliography and Master Index for NetView, NCP, and VTAM* |

**Planning**

| | |
|---|---|
| SC31-6811 | *Planning and Reference for NetView, NCP, and VTAM* |

**Installation and Resource Definition**

| | |
|---|---|
| SC30-3348 | *NCP, SSP, and EP Generation and Loading Guide* |
| SC30-3440 | *NCP Migration Guide*[2] |
| SC31-6204 | *NCP Migration Guide for Version 5 Release 4* |
| SC30-3447 | *NCP, SSP, and EP Resource Definition Guide* |
| SC30-3448 | *NCP, SSP, and EP Resource Definition Reference* |
| SD35-0251 | *NCP and SSP Library Supplement* |

**Customization**

| | |
|---|---|
| LY30-5606 | *NCP Customization Guide* |
| LY30-5607 | *NCP Customization Reference* |
| LY43-0021 | *SSP Customization* |

**Operation**

| | |
|---|---|
| SC30-3169 | *NCP, SSP, and EP Messages and Codes* |

**Diagnosis**

| | |
|---|---|
| LY30-5591 | *NCP, SSP, and EP Diagnosis Guide* |
| LY30-5605 | *NCP and EP Reference* |
| LY30-5603 | *NCP and EP Reference Summary and Data Areas* |

---

2 For migration from NCP V2 and later releases to NCP V5R1, V5R2, V5R2.1, or V5R3.

# Part 1. Generating and Loading under MVS

# Chapter 1. Generating the Program under MVS

After you install your Network Control Program (NCP) and System Support Programs (SSP) product from the tape and define NCP's configuration, the next step in producing an operating NCP is to generate the program.

This chapter contains information about generating NCP under the MVS operating system. It discusses the following topics:

- Understanding the generation procedure
- Controlling the generation procedure
- Performing different types of NCP generations
- Correlating NCP and resource resolution table (RRT) load modules
- Understanding listings and error messages.

SSP Version 3 includes the NCP/EP definition facility (NDF), a program used in generating an NCP, partitioned emulation program (PEP), or Emulation Program (EP) load module. NDF can be used to perform the following tasks:

- FASTRUN validation of an NCP, PEP, or EP generation definition
- Generation of an NCP, PEP, or EP load module
- Generation of an NCP or PEP V4 Subset load module
- Generation of an NCP or PEP load module with IBM* special products or user-written code
- Generation of a text data set for dynamic reconfiguration
- Migration of an existing generation definition to a different version and release or a different controller.

SSP V3R2 and later releases contain the NDF standard attachment facility, which allows user-written generation applications to interface with NDF during an NCP generation. The NDF standard attachment facility helps you define resources for user-written code. Use the NDF standard attachment facility to generate user-written code with NCP V4 Subset, NCP V4R2 and later releases, or NCP Version 5. For more information, see "Running an NCP or PEP Generation with IBM Special Products or User-Written Code" on page 14.

Before running NDF, you must supply job control language (JCL) to control the generation procedure. NDF does not create any JCL for you. This chapter discusses how to control the generation procedure, and Chapter 2 supplies examples of JCL for generation.

## Understanding the Generation Procedure

Generating an NCP with NDF under the MVS operating system is a two-step process. Figure 1 on page 5 shows the input NDF accepts and the output it produces under the MVS operating system.

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

## Generation Steps

**Step 1:** The first generation step, the NDF step, consists of four phases.

**Phase 1:** In the first phase, the generation validation phase, NDF does the following:

- Reads your GENDECK data set containing your generation definition
- Validates the definition statements and keywords coded in the generation definition
- Creates the NEWDEFN data set when you code the NEWDEFN keyword on the OPTIONS definition statement
- Generates assembler language source code for the resources coded in the generation definition
- Creates link-edit control statements; these statements will later link control-block object code with preassembled NCP object code to generate an NCP load module.

If you are using the NDF standard attachment facility to generate resources using IBM special products or user-written code, the first phase has two additional steps. During the validation phase, NDF does the following:

- Dynamically loads one or more user-written generation load modules
- Calls routines in the user-written generation load modules to perform generation processing and allows the routines to call NDF internal routines.

**Phases 2 and 3:** The second and third phases are the table 1 and table 2 assemblies. Each assembly reads the source code specification created in the generation validation phase and generates object code for the control blocks.

**Phase 4:** In the fourth phase, the return code summary, NDF does the following:

- Generates a composite return code that shows the success or failure of each phase
- Creates a compact listing that gives return codes for the generation validation and table assembly phases.

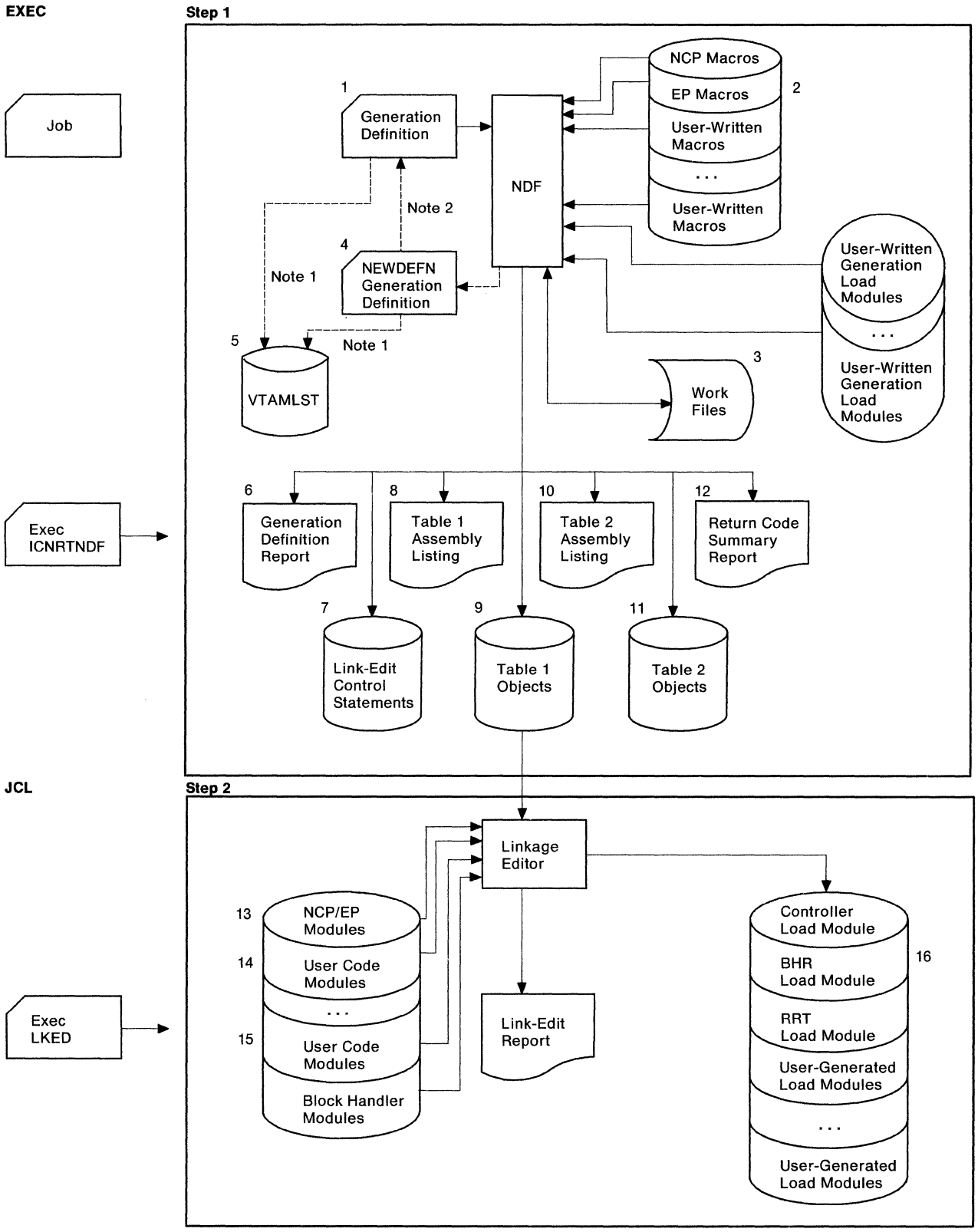



*Figure 1. The generation procedure under MVS. The numbers in this figure correspond to the data sets described in "Specifying Data Sets Used by NDF" on page 8. For an explanation of Note 1 and Note 2, see the following text.*

**Notes on Figure:**

1. Input to VTAMLST, from either the NCP definition statements or the NEWDEFN data set, is required. Refer to the description of NEWDEFN in Table 2 on page 8.
2. You can use definitions from NEWDEFN as input to the generation definition. For more information, refer to *NCP, SSP, and EP Resource Definition Reference* (the REUSE suboperand of the NEWDEFN keyword).

**Step 2:** The second generation step is the link-edit. During this step, the control-block object code produced in the first step is linked with preassembled object modules to generate a load module. If you included IBM special products or user-written code or block handlers in the generation definition, the appropriate object module libraries must be available during the link-edit. You should ignore a zero-length control section (CSECT) indication in the NCP link-edit.

With SSP V3R4 and later releases, when no errors occur, the link-edit return code is 0. When the NDF standard attachment facility generates a load module separate from the NCP load module, the return code is 4 if the load module is generated from table 1, and 0 if from table 2. You should investigate all return codes of 4 to determine if they are informational or indicate an error that must be resolved.

You cannot use LA, BAL, or BLG instructions to reference labels that reside above the 4MB (MB equals 1 048 576 bytes) boundary. For NCP V5R4, load modules can be larger than 4MB but not larger than 8MB. To determine how much storage is available for NCP buffers in your communication controller, perform the following calculation:

1. Locate the $BUFPOOL value in the link-edit portion of your generation listing ($BUFPOOL marks the end of the load module) and add it to the value from the ICN076I informational message issued in the GENEND definition statement. Both values are hexadecimal.
2. Subtract the value obtained in number 1 from the amount of storage available in your NCP.
3. From the value obtained in number 2, subtract the amount of storage allocated for the maintenance and operator subsystem (MOSS) Mailbox/TSS Workspace. You can find this amount in the control data set (CDS) control block at offset 46(2E); it is also entered as a number of 4KB (KB equals 1024 bytes) pages when the operator initializes NCP.
4. The number remaining from this subtraction is the amount of storage available for buffers.
5. To determine the number of buffers, add 12(C) to the value coded for BFRS on the BUILD definition statement; divide this number into the amount of storage available for buffers (obtained in number 4).

**Note:** If you want to run a FASTRUN generation to validate your generation definition without creating control blocks or if you want to do a generation for dynamic reconfiguration, do not specify the link-edit step to be run in your JCL.

## DASD Work Space Requirements

NDF uses a storage manager to organize its work space during NDF generation validation. Whenever possible, storage manager data is kept in virtual storage. However, if data overflows the virtual storage region available to the storage manager, this extra data is written into a storage manager work data set. Generally, you should be able to allocate enough virtual storage to hold all the work data. For very large generations, however, you may be required to define a work data set.

All NDF generations require about 200KB of work space and an additional 48 bytes for each resource specified in the generation definition. The storage manager usually requires less than 500KB of virtual storage.

If you need additional work space or if you are using the NDF standard attachment facility, you need to define a storage-manager work data set (DBWORKFL) in your JCL. You should ensure this space allocation is not excessive because the time required to initialize a large work data set adds a significant amount of running time to generation validation. For information about how to specify a storage-manager work data set, see "Specifying Data Sets Used by NDF" on page 8.

Generally, one cylinder of disk space allocated for a work data set should be adequate.

If you are generating NCP/Token-Ring interconnection resources, you need an additional 80 bytes for each NCP/Token-Ring interconnection physical line specified in the generation definition and an additional 160 bytes for each NCP/Token-Ring interconnection logical line generated using the AUTOGEN facility.

## Performance Considerations

NDF requires approximately 3.5MB of region size to achieve optimal performance, although as much as 8MB may be required to complete a generation. If the available region size drops below 3.5MB, paging during the generation validation phase significantly degrades performance.

In addition, a block size of at least 3630 bytes for the table 1 listing data set is recommended. Using even larger block sizes can noticeably improve performance. If you define an inadequate block size, the time required for the input and output operations to the data set can significantly affect elapsed time for NDF execution.

# Controlling the Generation Procedure

You control the generation procedure through the JCL that you code and through the definition statements and keywords that you coded in the generation definition.

This section explains the different types of generations you can run. It discusses some of the definition statements and keywords you can code in your generation definition. It also discusses the parameters you need to code and the data sets you need to specify in your JCL. For examples of JCL for generation, see Chapter 2.

## Specifying Data Sets Used by NDF

This section contains the data definition names (ddnames) of the data sets used by NDF. You specify the ddnames in your JCL. Table 2 lists the ddnames and describes the data sets they specify.

**Note:** The numbers following the ddnames correspond to the data sets shown in Figure 1 on page 5.

*Table 2 (Page 1 of 3). ddnames of data sets used by NDF (MVS)*

| ddname | Description |
|---|---|
| **STEPLIB** | Specifies the library containing NDF and IHR load modules. If you have any user-written generation applications that use the NDF standard attachment facility, STEPLIB must also specify the libraries containing the user-written generation load modules. |
| **GENDECK (1)** | Specifies the data set containing the definition statements for the NCP network definition. This data set must contain 80-byte fixed-format records. |
| **SYSLIB (2)** | Specifies the chain of definition statement libraries. The libraries included depend on the particular NDF generation. You need the IBM 3705, 3720, 3725, or 3745 NCP definition statements for the table assemblies. You may also need additional libraries for both the generation validation phase and the table assemblies if IBM special products or user-written code is in your NCP. |
| **DBWORKFL (3)** | Specifies the storage manager work data set. This temporary data set stores internal data in 4KB records during NDF generation validation. The BDAM accesses records in the data set. Use this data set if NDF cannot obtain enough virtual storage to hold all of the temporary data for the generation validation phase or, if you are using the NDF standard attachment facility, to generate IBM special products or user-written code. Ensure that this space allocation is not excessive because the time required to initialize a large work data set adds a significant amount of running time to generation validation. |
| **TBL1SRCE (3)** | Specifies the data set that contains the table 1 assembly source code. This data set contains the output from generation validation and serves as the input data set for the table 1 assembly. |
| **TBL2SRCE (3)** | Specifies the data set that contains the table 2 assembly source code. This data set contains output from the generation validation phase and serves as input for the table 2 assembly. |
| **SYSUT1 (3)** | Specifies the assembler work data set. This work data set temporarily stores internal NDF data for the assembly of table 1 and table 2 objects. |

*Table 2 (Page 2 of 3). ddnames of data sets used by NDF (MVS)*

| ddname | Description |
|---|---|
| **NEWDEFN (4)** | Specifies the output data set containing the new generation definition created by NDF. For more information, refer to *NCP, SSP, and EP Resource Definition Guide*.<br><br>The new generation definition consists of the input from the definitions from the NCP generation definition plus statements and keywords added during the generation process.<br><br>**Notes:**<br><br>1. If you specified NEWDEFN=YES on the OPTIONS definition statement in your generation definition, you must define the NEWDEFN data set in the JCL for your generation.<br>2. If VTAM users generate a NEWDEFN data set, they must include a NEWDEFN data set in the VTAMLST that VTAM accesses during the activation of this NCP. If they do not generate a NEWDEFN data set, they must include their generation definition (GENDECK) in VTAMLST.<br>3. All VTAM users must include the NEWDEFN source in the VTAMLST that VTAM accesses during the activation of NCP. |
| **VTAMLST (5)** | For information about VTAMLST, refer to *VTAM Network Implementation Guide*. |
| **SYSPRINT (6)** | Specifies the data set that contains the generation validation listing. |
| **LNKSTMT (7)** | Specifies the link-edit statement data set. This data set contains the link-edit control statements produced by NDF that are used to build the NCP load module. |
| **SYSLIN (7)** | Specifies the input data set that contains the link-edit control statements passed to the linkage editor from NDF. This is the same data set as LNKSTMT from phase 1 of the NDF job. |
| **TBL1LIST (8)** | Specifies the data set for the table 1 assembly listing. This data set can be large, and its data control-block parameters can have a significant impact on NDF performance. |
| **TBL1OBJ (9)** | Specifies the table 1 object data set. This data set must be a member of a partitioned data set that is passed to the link-edit step of an NDF generation. The member name for this data set is ICNTABL1. |
| **TBL2LIST (10)** | Specifies the table 2 assembly listing data set. |
| **TBL2OBJ (11)** | Specifies the output data set for the control-block objects generated by the table 2 assembly. This data set must be a member of the same partitioned data set as TBL1OBJ. The member name for this data set is ICNTABL2. |
| **SYSPUNCH (9, 11)** | Specifies the library where the table assemblies place the control-block objects. This data set contains the TBL1OBJ and TBL2OBJ data set members created in phases 2 and 3 of the NDF job. |
| **PRINTER (12)** | Specifies the data set for the return code summary report. |

*Table 2 (Page 3 of 3). ddnames of data sets used by NDF (MVS)*

| ddname | Description |
|---|---|
| **xxxxxxx (13)** | Specifies the library that contains the preassembled NCP object modules. The ddname and the specific library vary with the target of the generation. Specify ANCPMOD1 if you are generating NCP V4R3, V4R3.1, or V5R2 or a later release. Specify OBJ3705 if you are generating NCP V3 for the IBM 3705 Communications Controller. Specify OBJ3725 if you are generating any other versions of NCP, regardless of which communication controller you plan to load. |
| **____ (14)** | Specifies a library with a ddname determined by IBM special products or user-written code. This library contains preassembled object code for user-written code modules. |
| **ULIB (15)** | Specifies a library that contains block-handler object modules or preassembled user-written code modules. |
| **SYSLMOD (16)** | Specifies the library where the communication controller, block-handler set resolution table (BHR), and RRT load modules will be placed. For SSP V3R3 and later releases, SYSLMOD also specifies the library where user-generated load modules will be placed. Do not code BLKSIZE in the generation JCL or when preallocating data sets. |
| The following three data sets are defined only if you code ASSEMBLY=YES when invoking NDF; this causes the NDF controller assembler to assemble control block code outside the regular NDF process. | |
| **ASMSRCE** | Specifies the assembly source code used as input to the NDF controller assembler when the assembly option is specified. The data set must contain 80-byte fixed-format records. |
| **ASMLIST** | Specifies the data set for the ASMSRCE assembly listing. |
| **ASMOBJ** | Specifies the data set for the ASMSRCE object data set. |

# Specifying Parameters for NDF

This section describes the optional NDF parameters that you can specify in your JCL. When specifying more than one parameter in the parameter field, you must separate the parameters with a comma.

## LINECNT Parameter

Use the LINECNT parameter to specify the number of lines on each page of the generation validation listing and the table assembly listing. The valid range for this parameter is 10 to 99. The default value for the validation listing and for the assembly listing is 60. If you specify a value for LINECNT, this value is used in all listings.

The following is an example of LINECNT in the JCL:

```
//STEP EXEC PGM=ICNRTNDF,PARM='LINECNT=40'
```

## FASTRUN Parameter

For SSP V3R2 and later releases, you can use the FASTRUN parameter to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

Using the FASTRUN parameter is the same as coding FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition.

The following is an example of FASTRUN in the JCL:

```
//STEP EXEC PGM=ICNRTNDF,PARM='FASTRUN=ON'
```

## ASSEMBLY Parameter

For SSP V3R2 and later releases, you can use the ASSEMBLY parameter to invoke the NDF controller assembler to assemble table source code. A complete NCP generation does not need nor require the ASSEMBLY parameter.

The input and output data set names used by NDF when you specify ASSEMBLY are different from those used for the table assembly, except for the assembler work data set (SYSUT1). See "Specifying Data Sets Used by NDF" on page 8 for the names and descriptions of these data sets.

**Note:** You cannot specify both the FASTRUN parameter and the ASSEMBLY parameter for the same job step.

The following is an example of ASSEMBLY in the JCL:

```
//STEP EXEC PGM=ICNRTNDF,PARM='ASSEMBLY=YES'
```

## ASSMLIST Parameter

For SSP V3R5 and later releases, you can use the ASSMLIST parameter to generate the table assembly listing. Valid values for ASSMLIST are YES and NO. The default is ASSMLIST=YES. When ASSMLIST=NO, the table assembly listing is suppressed.

**Note:** You cannot specify both the FASTRUN parameter and the ASSMLIST parameter for the same job step.

The following is an example of ASSMLIST in the JCL:

```
//STEP EXEC PGM=ICNRTNDF,PARM='ASSMLIST=NO'
```

## Migration Aid Parameters

For SSP V3R5 and later releases, you can use the migration aid parameters to invoke the migration aid. The migration aid is an NDF function that automates much of the NCP migration task. For more information on these parameters, refer to *NCP, SSP, and EP Resource Definition Reference* and *NCP Migration Guide for Version 5 Release 4.*

The following is an example of the migration aid parameters in the JCL:

```
//NDF EXEC PGM=ICNRTNDF,REGION=6000K,PARM=('TMODEL=3745-410',
//            'TUSGTIER=5','TVERSION=V5R4'.)
```

## Naming Resources

Avoid using the prefixes shown in Table 3 and the labels shown in Table 4 when naming resources because they are used as control-block identifiers and can cause duplicate labels that result in an error message from the assembler.

*Table 3. Prefixes to avoid (MVS). Avoid names that are similar to control-block acronyms.*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| @ | BOQ | CPT | EPI | IX | LTV | NPB | QCB | SNP | UNA |
| $ | BPB | CRB | EQB | J* | LTX* | NPF | RAT | SOT | USC |
| AAB | BSB | CRP | ERB | LAA | LU | NQB | RCB | SPC | UXR* |
| ABN | BST | CTB | ERX | LAB | LX | NQE | RCQ | SST | U1 |
| ACB | BTT | CTP | FCT | LB* | L1B | NSQ | RCV | STE | VAT |
| ACT | BTU | CUB | FLB | LCB | L4B | NVT | RG | STQ | VIT |
| ACU | BUE | CY | FMT | LCC | MBF | NVX | RH* | SUT | VLB |
| AEB | CA* | CX | FVT | LCI | MBX | OLL | RN* | SVT | VR |
| ALE | CAB | DAE | GCB | LCP | MCT | OLT | RU* | SXB | VST |
| AST | CAI | DDB | GPT | LCS | MDR | PAB | R* | SYS | VTS |
| ATB | CAR | DIA | GRW | LCW | MIB | PAD | RMB | TCB | VVT |
| ATP | CAT | DPT | GVT | LDA* | MIC | PCB | ROSH* | TET | WCB |
| ATT | CB | DQB | HWE | LDI* | MIF | PIU | RST | TGB | WRP |
| AV* | CBB | DRS | HWX | LGT | MIH | PL* | RTR | TH* | WU |
| AXB | CDS | DRX | IB | LKB | MIM | PL2 | RVT | TIM | X |
| BC | CER | DSP | ICE | LKC | MLT | PMF | SCB | TND | XDA |
| BCU | CGP | DTG | ICI | LNB | MMV | PRB | SEB | TQB | XDB |
| BER | CHC | DVB | ICW | LNV | MSC | PSA | SGE | TRT | XDH |
| BGS | CHV | DVI | IDD | LPB | MTF | PSB | SGT | TVS | XID |
| BH | CIE | DVQ | IDE | LRB | NET | PSI | SHB | UAC | |
| BHD | CM | ECB | IDL | LRC | NIB | PSP | SID | UAD | |
| BHR | COE | ECD | IDB | LTC | NIX | PST | SIT | UIB | |
| BHS | CPI | ECL | IRN | LTR | NLB | PUV | SMB | UIC | |
| BLU | CPN | EML | IRQ | LTS | NLX | QAB | SMM | ULVSGN | |

**Note:** * Indicates that a number from 0 to 9 follows this prefix.

*Table 4. Labels to avoid (MVS). Avoid names that are similar to control-block acronyms.*

| | | | | | |
|---|---|---|---|---|---|
| ACITRAP | CSPQH2 | NCPHIST1 | SVCQUT | THLOB | TMRF |
| CAACER | CSPQOFF | NCPLVL | SWQTMQ1 | THLOM | TTCUR |
| CACCER | CSPQON | NEWLNE | SWQTMQ2 | THMID | TTEND |
| CADCER | DCTABND | OLDLNE | TABEND | THMPF | TTRECNTR |
| CAECER | DCTSAVEK | PEPQSCNB | TABSTAR | THODAIB | TTSKPCNT |
| CAFCER | D*RCB | PEPQSCNM | THAFIB | THODAIM | TTSTAR |
| CCPH1 | EPLVL | PSCA | THAFIM | THONLY | UIHRCCW |
| CCPSAVE | FILLB | ROSSVADDR | THBCUVVT | THPSIB | USTAGETR |
| CHANSNS1 | FILLC | ROSSVCCR | THFID | THPSIM | UTILSTSZ |
| CHANSNS2 | HDRNENT | ROSSVCCU | THFIRST | THTYPO | |
| CHSVBKSV | ICNTABL1 | ROSSWK1 | THFOB | THTYP1 | |
| CHSVH1 | LCDBSCB | SECNTRI | THFOM | THTYP2 | |
| CSPQH1 | LCDSSBIT | SVCO | THLAST | THTYP3 | |

**Note:** * Indicates that a number from 0 to 9 can appear as this character.

## Defining Virtual Storage

You can control virtual storage size by specifying the REGION parameter on the JOB statement or the EXEC statement for the NDF step in your JCL. A region of 4MB should be adequate for most NDF runs, although very large generation definitions may require more than twice this much storage.

The following is an example of the REGION specification in the JCL for 4MB of storage:

```
//NDF    EXEC  PGM=ICNRTNDF,REGION=4096K
```

For SSP V3R6, the NDF IHR Assembler supports 31-bit addressing, allowing you to use the addressable storage available above the 16MB line in an MVS/XA* environment. To submit larger generation decks, you must increase the region size on the EXEC statement for the NDF step in your JCL.

## Naming Load Modules

Besides creating an NCP load module, NDF also produces an RRT load module and, if you have coded any block-handling routines, a block-handler set resolution table (BHR) load module. These load modules contain information that the access method requires. For more information on the RRT load module, see "Correlating NCP and Resource Resolution Table Load Modules" on page 18.

Use the NEWNAME keyword on the BUILD definition statement to designate the names for the BHR, RRT, and NCP load modules. NDF appends a *B* to the NEWNAME value to name a BHR load module and appends an *R* to the NEWNAME value to name an RRT load module.

For information about the NEWNAME keyword on the BUILD definition statement, see *NCP, SSP, and EP Resource Definition Guide*. For information on how to code this keyword, see *NCP, SSP, and EP Resource Definition Reference*.

## Controlling Succeeding Generation Steps

NDF produces an overall return code for the NDF step. NDF prints this return code as part of the return code summary section of the NDF report and denotes the NDF phase where it encountered the error. NDF also passes this overall return code to the operating system as the NDF return code.

You can use this overall return code to determine whether to run succeeding job steps. If there are no errors in the NDF step, code a step in the JCL to run the link-edit. This technique is used in the sample JCL for an NCP/PEP generation with output written to tape on page 34. The following list shows the overall return code values and meanings; notice that leading zeros are suppressed:

| Value | Meaning |
|---|---|
| 1 | Input validation error |
| 10 | Table 1 error |
| 100 | Table 2 error |
| 1000 | Printer file error. |

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

# Performing Different Types of NCP Generations

This section discusses the different types of NCP generations and what you must do to run them.

## Running a FASTRUN Generation

Do a FASTRUN generation to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

To run a FASTRUN generation, code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for SSP Version 3. For SSP V3R2 and later releases, you may alternately code FASTRUN=ON as a parameter in your JCL when calling NDF. Ensure that your JCL does not call the linkage editor; if the link-edit step is present, an error will result. Also, do not define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

For an example of the JCL for a FASTRUN generation, see page 25.

## Running a Standard NCP or PEP Generation

To run a standard NCP or PEP generation, supply your generation definition as input and specify the various input and output data sets in your JCL. You can specify that input and output data sets be written to disk or that certain data sets be written to tape.

If you detect an error while generating or running your NCP, you can write to tape certain listing data sets, such as the table 1 assembly listing, the table 2 assembly listing, and the link-edit listing.

If you are including certain types of resources in your generation definition (such as those listed under NEWDEFN (4) in Table 2 on page 8), you must code NEWDEFN=YES on the OPTIONS definition statement as the first executable statement in your generation definition and define the NEWDEFN data set in your JCL. For more information on coding the NEWDEFN keyword, refer to *NCP, SSP, and EP Resource Definition Reference.*

For examples of JCL for running standard NCP or PEP generations, see "Example of an NCP or PEP Generation with Output Written to Disk" on page 27 and "Example of an NCP or PEP Generation with Output Written to Tape" on page 34.

## Running an NCP or PEP Generation with IBM Special Products or User-Written Code

If you included IBM special products or user-written code—such as Network Terminal Option (NTO), Network Routing Facility (NRF), or X.25 NCP Packet Switching Interface (NPSI)—in an NCP or PEP generation, you must modify the basic JCL.

If you are using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing. You are not required to use this method.

If you choose to generate NCP and user-written code *without* using the NDF standard attachment facility or if you are generating user-written code using SSP V3R1, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of the link-edit statements by coding keywords on the GENEND definition statement.

## Using the NDF Standard Attachment Facility

You can run this type of generation *only* if you are using SSP V3R2 or a later release to generate NCP V4 Subset, NCP V4R2 or a later release, or NCP Version 5. To use the NDF standard attachment facility, you must supply a user-written generation application. For information on writing user-written code and user-written generation applications, see *SSP Customization.* Figure 2 on page 16 shows how to include your user-written code and user-written generation load modules in the generation procedure.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation load modules to be loaded in the generation. Each application must have its own generation load module. You can specify up to 25 generation load modules.
- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation load modules.
- Modify the JCL for a standard NCP or PEP generation to include the ddnames for the NEWDEFN data set, the DBWORKFL data set, and the libraries for user-supplied modules.

For an example of the JCL for generating user-written code using the NDF standard attachment facility, see page 42.

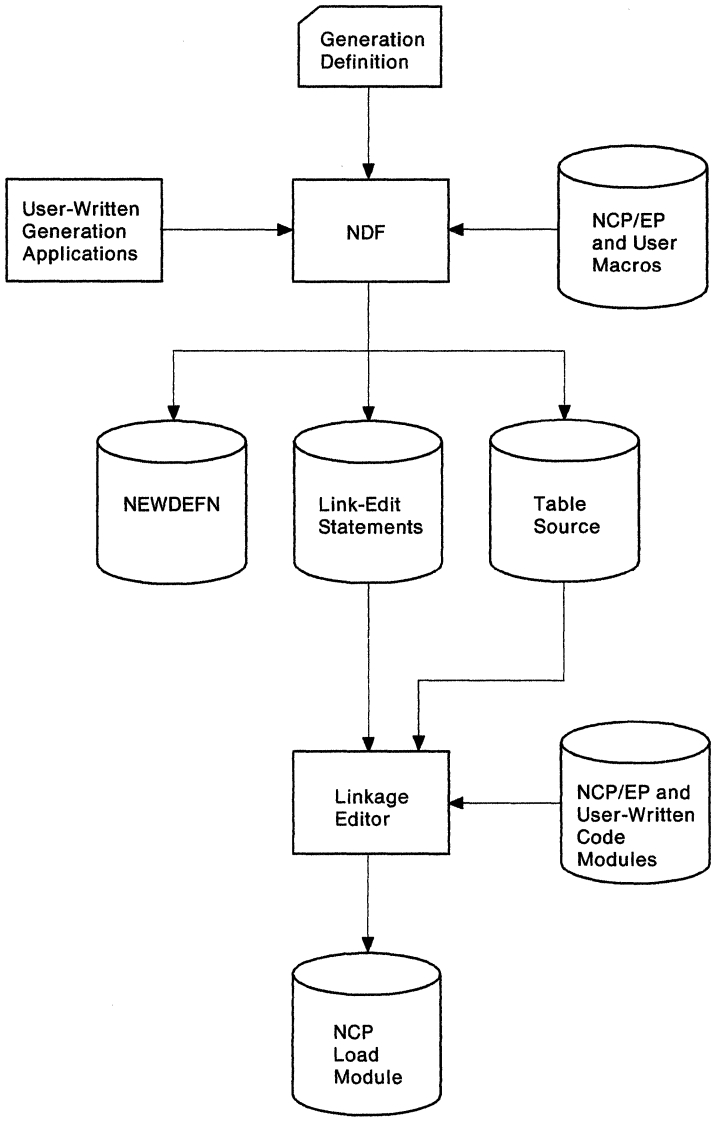



*Figure 2. Generating an NCP containing user-written code using the NDF standard attachment facility (MVS). This figure shows how to include user-written generation load modules in an NCP or PEP generation.*

## Using the GENEND Definition Statement

Besides being able to generate NCP V4 Subset, NCP V4R2 and later releases, or NCP Version 5 with IBM special products or user-written code using the NDF standard attachment facility, you can also generate them as described here.

You can use the GENEND definition statement to generate any version of NCP using SSP Version 3. Before generating NCP, code the link-edit statements for the routines and identify the location of these link-edit statements by coding certain keywords on the GENEND definition statement.

Figure 3 on page 17 shows how to include your IBM special products or user-written code in the generation procedure.

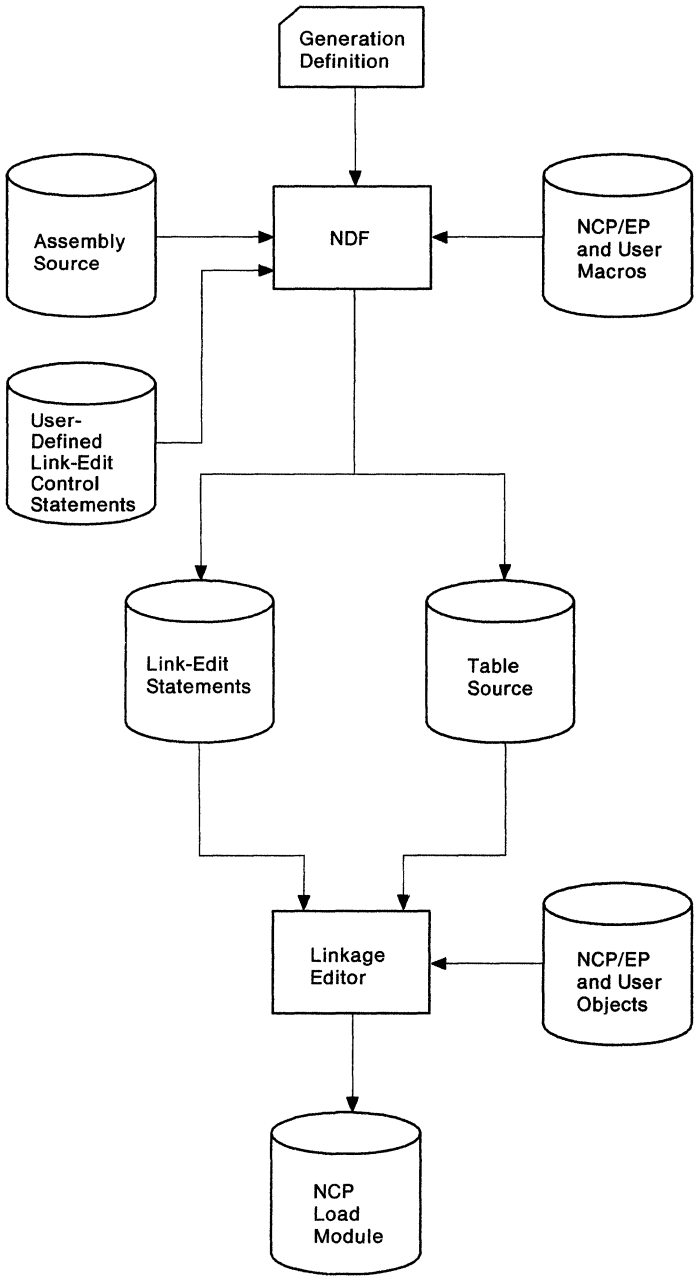



*Figure 3. Generating an NCP containing user-written code using the GENEND definition statement (MVS). This figure shows how to include user-written code in an NCP or PEP generation.*

Before you generate IBM special products or user-written code using the GENEND definition statement, ensure that you:

- Assemble the user-written routines and code the link-edit control statements for the routines
- Code the appropriate keywords on the GENEND definition statement for your user-written routines
- Place the members with SRCLO or SRCHI code in a definition statement library (SYSLIB) available to NDF

- Place all members that contain INCLUDE or ORDER link-edit control statements in a definition statement library in the NDF SYSLIB chain
- Place all definition statements in the NDF SYSLIB chain
- Modify the JCL to include the SYSLIB chain and the ULIB or user object code library ddname statement.

The generation validation phase of NDF reads the link-edit control statements and writes them into the same data set as the standard NCP link-edit control statements.

For an example of JCL for generating user-written code using the GENEND definition statement, see page 42.

## Running a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, use the text data set from a dynamic reconfiguration generation. Ensure that you coded the original NCP to allow dynamic reconfiguration. If you did, the dynamic reconfiguration generation produces a text data set that the access method can use to modify NCP.

**Note:** VTAM* has its own dynamic reconfiguration procedures that do not require you to use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Network Implementation Guide*.

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration data set consisting of ADD or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration data set is the input for the dynamic reconfiguration generation. This type of generation produces a text file that the access method uses to modify an NCP that is already running in a communication controller. For information on using ADD, DELETE, PU, and LU definition statements, see *NCP, SSP, and EP Resource Definition Guide.*

A dynamic reconfiguration generation requires one table assembly and no link-edit. For an example of JCL for a dynamic reconfiguration generation, see page 45.

# Correlating NCP and Resource Resolution Table Load Modules

For VTAM V3R2 and later releases, NCP V4R3 and later releases, or NCP Version 5, when VTAM activates NCP, the two programs must be in synchronization. VTAM and NCP must start in agreement with network addresses because both programs perform network address management.

To ensure this synchronization, NDF creates a file that contains the resource name and element address of each resource definition statement found during the processing of the generation definition. This file is called the *resource resolution table (RRT).* You must place the RRT and the NCP load modules in VTAM's NCP load library following the generation.

When VTAM attempts to contact NCP, part of the contact process involves sending an SNA active physical unit request unit to NCP. NCP responds by sending a correlation element that permits VTAM to verify that the RRT and the NCP load modules correspond.

---

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

You must specify the correlation element, stored in both the RRT and NCP load modules, in the NCP generation definition using the GENLEVEL keyword on the BUILD definition statement. If you do not specify it on the GENLEVEL keyword, the correlation element defaults to the date and time of generation.

VTAM compares the correlation element found in the RRT to the one returned by NCP to ensure that the two programs are synchronized. If the correlation elements differ, VTAM informs you of the mismatch if you specified VFYC=YES on the PCCU definition statement. If VFYC=IGNORE, VTAM automatically overrides the mismatch and continues with the NCP activation. If VFYC=YES, VTAM gives you the option of continuing with the activation. Choosing to continue could result in serious consequences, depending on your configuration. Consider the following before you decide to continue:

- If one host owns all of an NCP's resources and that host is the only one that will ever activate that NCP, a mismatch could indicate you are referencing an RRT that corresponds to a different NCP. It could also mean that you have generated an NCP at two different times or that either the NCP or the RRT is down-level. In either case, the mismatch implies a problem, and you should not take the VTAM option to continue.
- If one host owns all of an NCP's resources but other hosts have the ability to activate that NCP, the concerns covered in the preceding paragraph apply. However, for non-owning hosts, a mismatch is of no concern because these hosts will never contact any of the resources in that NCP. Therefore, if you are using a non-owning host, you can safely instruct VTAM to override the mismatch and continue the NCP activation.
- If two or more hosts divide ownership of an NCP's resources, it is essential that the RRT in each host reflect that NCP's resources. You should never instruct VTAM to override the mismatch and continue the NCP activation. The safest way to ensure that the RRTs in each host correspond to each other is for the host that generated the NCP to send copies of the RRT to the other hosts. IBM recommends this procedure.

For more information about the VFYC keyword, see *VTAM Resource Definition Reference*.

If you are working with a configuration in which two or more hosts divide ownership of an NCP, an alternative is for each host to generate its own RRT using NDF. Only the hosts that load NCP need to save the generated NCP load module.

You should use this alternative only after you have established procedures to verify that the NCP generation definition in each host is identical to those of other hosts and you have specified the GENLEVEL keyword identically on all the generation definitions. Following these procedures will ensure that you insert the identical correlation element into each of the RRTs. This extra care is necessary because using the GENLEVEL keyword negates the VTAM correlation check. If a generation definition change is made in one host and not propagated to the others and that host then generates and loads NCP, the RRTs in the other host immediately become down-level and addressing mismatches can occur. You may not discover a mismatch until long after its creation, and you will have difficulty diagnosing the problem without VTAM traces running continually.

# Understanding Listings and Error Messages

During a generation validation run, NDF creates a report that contains:

- The input statements interspersed with informational and error messages
- The keywords and statements passed to NDF from user-written generation load modules using the NDF standard attachment facility (SSP V3R2 and later releases)
- A resource name and network address cross-reference (only if the generation validation run is valid)
- An error message summary
- A return code for the generation validation phase.

Also, NDF creates a listing during the return code summary phase that gives return codes for the generation definition and for each of the table assemblies.

In NCP V4R3 and later releases or NCP Version 5, generation definition listings produced by SSP V3R4.1 and later releases include a message indicating how much storage NCP needs for initialization, in excess of the storage that the load module displaces. For information on calculating buffer storage, see "Generation Steps" on page 4.

If any errors occur in generation validation, NDF notes these errors through diagnostic messages in the report. Table 5 shows the NDF message severity levels and their meanings.

*Table 5. NDF message severity levels (MVS)*

| Severity Level | Meaning |
|---|---|
| Info | This is an informational message that either informs you of NDF calculations (such as message ICN0761) or indicates how NDF has changed, ignored, deleted, or added a keyword. NDF did not consider the message serious enough to stop the generation process; however, you should examine the message to determine whether you want to accept the NDF change or make your own to the generation definition. |
| Warning | An error has occurred for which NDF has taken corrective action by assuming a default keyword value or by ignoring the value supplied. The generation process is terminated after validation of the generation definition. The NDF migration aid also issues a warning message when it cannot determine a value to use. |
| Error | A user error has occurred for which NDF cannot assume a value or ignore the value supplied. The generation process is terminated after validation of the generation definition. |
| Ten | A fatal user error has been detected. The generation process is terminated. |
| Severe | A system error has occurred. NDF produces a procedure traceback. The generation process is terminated after validation of the generation definition. |
| Fatal | A fatal system error has occurred. A procedure traceback is printed and the generation process is terminated. |

For all but the informational messages, NDF ends output of control-block source and link-edit control statements but continues to validate the input definition statements.

In this case, you must correct the errors and run the generation validation again. If the return code from the generation validation and the table assemblies is 0, NDF runs to completion, runs the link-edit, and produces a load module.

Other programs, such as VTAM and configuration report program, require the same definition statements that you used to generate NCP, plus additional keywords and definition statements specific to each procedure. *NCP, SSP, and EP Resource Definition Reference* identifies these additional keywords and definition statements. Although you can place these keywords and definition statements in the NCP generation definition either before or after you generate NCP, it is recommended that you add them before you generate NCP because executing the different procedures with different inputs can create errors.

If you are using the NDF standard attachment facility (SSP V3R2 or a later release) to generate user-written code with your NCP, products that use the same generation definition may require as input the source statements or keywords passed to NDF from the user-written generation load modules. By specifying NEWDEFN=YES on the OPTIONS definition statement in your generation definition and by specifying the NEWDEFN data set in your JCL, you instruct NDF to create a new generation definition for use by these procedures. This new generation definition contains both the NCP source statements and the statements passed from the user-written generation load modules.

NDF checks only the accuracy of the values coded for NCP generation procedures that appear in NCP generation input. In the same way, these other procedures do not check the validity of the NCP definition statements and keywords. Therefore, NDF requires that NCP generation have errors less than a severity code of 4.

## Sample NDF Generation Report

Figure 4 on page 22 contains a sample NDF generation report. Reverse-coded numbers (for example, 3 ) indicate comments about the report that are not a part of the actual report. You can find the comments corresponding to these numbers following the report. Ellipses (. . .) indicate that parts of the report were deleted for this example.

```
1                    2                    3
ACF SSP V3R4.1       12/20/90  18:30:03     DEFINITION SPECIFICATION                                    PAGE 1
PAGE 1
4
LINE #                STATEMENT
    83  *         FULL LINE COMMENT 5
    84  PHY01     GROUP ECLTYPE=PHY           NTRI PHYSICAL GROUP  6
                  . . .
                     D               VIRTUAL=NO  7
                     D               TEXTTO=3.0
                     D               COMPACB=NO
    85  PHYLN1    LINE  ADDRESS=(80,FULL),UACB=WRONG,LOCADD=400001234567
                  . . .
                     D               MAXTSL=265
*WARNING  ICN021I   04  UACB=WRONG INVALID, INVALID NUMBER OF SUBOPERANDS, 2 EXPECTED WHEN ECLTYPE=PHYSICAL ON GROUP  8
                     STATEMENT, REPLACED FOR STATEMENT KEYWORD VALIDATION
9             DELETED          UACB
                     ADDED           UACB(1)-X$P1AX  10
                     ADDED           UACB(2)-X$P1AR
                     G               TYPE=NCP
                     G               MAXPU=1
                     D               CLOCKNG=EXT
11       . . .
GENERATED BY ECL
    86  X$P1SQ    SERVICE
    87  ECLUPU1   PU
                     ADDED           ADDR(1)=01
                     G               PUTYPE=1
                     D               AVGPB=532
                     D               ANS=STOP
                     D               BNNSUP=()
                     D               MAXDATA=78
                  . . .
    88  ECLLU1    LU
                     ADDED           LOCADDR(1)=0
                     D               PACING(1)=1
                     D               BATCH=NO
                     D               LUDR=NO
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30     LABEL CROSS REFERENCE                                       PAGE 6

                     LABEL CROSS REFERENCE -- SORTED BY LABEL NAME  12

LABEL         LINE     SA  ELEM       LABEL        LINE     SA  ELEM       LABEL        LINE     SA  ELEM

                  . . .

G14B1         178                     T14020BB     545      0E  001B       T14022G8     915      0E  004D
G14S1         362                     T14020BC     549      0E  001C       T14022G9     916      0E  004E
                  . . .
L14022        828      0E  0044       T14020B8     533      0E  0018       T14023A7     305      0E  0009
L14023        201      0E  0001       T14020B9     537      0E  0019       T14023A8     315      0E  000A
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30     LABEL CROSS REFERENCE                                            PAGE 7

                     LABEL CROSS REFERENCE -- SORTED BY NETWORK ADDRESS  13

SA  ELEM     LINE   LABEL              SA  ELEM    LINE    LABEL              SA  ELEM    LINE    LABEL

0E  0001     201    L14023             0E  0038    747     T14020F8           0E  006E    90      DRPOOLLU
0E  0002     259    B14023A            0E  0039    749     T14020F9           0E  006F    6       NCPBUILD
0E  0003     269    T14023A1           0E  003A    767     T14020G            0E  0070    6       NCPBUILD
0E  0004     275    T14023A2           0E  003B    785     T14020G1           0E  0071    6       NCPBUILD
                  . . .
ACF SSP V3R4.1       11/20/90  18:30:03   ERROR SUMMARY                                                  PAGE 8
                                                  14
TOTAL MESSAGES      INFO     WARNING     ERROR     SEVERE     FATAL
      1              0          0         1         0          0

RETURN CODE IS 8


MESSAGES APPEAR AFTER THE LINES NUMBERED:

85

REGENERATION REQUIRED
```

*Figure 4. Sample NDF generation report (MVS)*

## Comments

**1** **SSP version and release number.**

**2** **Date and time of the NDF run.** The date and time of the NDF run are the same as those recorded in the date and time generation control block in the NCP or PEP load module and printed in the formatted portion of the NCP or PEP load module and dump.

**3** **Report section identification.** This identification has one of the following values: DEFINITION SPECIFICATION, LABEL CROSS REFERENCE, or ERROR SUMMARY.

**4** **Line number column.** This column contains the line numbers of the generation definition listing.

**5** **Full-line comments from the generation deck.**

**6** **Partial-line comment from the generation deck.**

**7** **Information describing defaulted or inherited keywords.** This 1-letter prefix to the message shows keywords that are defaulted or keywords that use values from previous definition statements. These prefixes are:

- G Keyword inherited from GROUP
- L Keyword inherited from LINE
- T Keyword inherited from TERMINAL
- C Keyword inherited from CLUSTER
- P Keyword inherited from PU
- D Keyword that has been defaulted

**8** **Error message.** This error message has an appropriate error number followed by a severity code and error message text. A severity code of 4 or more requires correction to the generation definition before you can generate a load module. A severity code of less than 4 informs you that NDF has taken corrective action and does not require regeneration. You should, however, verify that the correction made by NDF will satisfy your generation requirements.

**9** **DELETE message.** This DELETE message indicates keywords replaced by a user-written generation application or replaced by NDF for NCP/Token-Ring interconnection resources.

**10** **ADDED or APPENDED message.** This ADDED or APPENDED message indicates keywords passed to NDF by a user-written generation application or added to the generation definition by NDF for NCP/Token-Ring interconnection resources. A keyword is added if you did not code the keyword or if you coded then replaced the keyword; a keyword is appended if you coded the keyword.

**11** **GENERATED BY ECL or GENERATED BY** *usergen name*. This GENERATED BY ECL or GENERATED BY statement precedes statements passed to NDF by user-written generation applications using the NDF standard attachment facility or precedes statements added to the generation definition by NDF for NCP/Token-Ring interconnection resources or automatic line replication.

**12** **First label cross-reference.** This list contains all user-coded labels, sorted by label name. If the label has an associated network address, it is printed. Resources defined by the keywords LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist. It is included in this sample as an illustration only.

13 **Second label cross-reference.** This list contains all user-coded labels, sorted by network address. Labels without associated network addresses are omitted. Resources defined by the keywords LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist.

14 **Error summary section.** This summary contains an error count and a list of the line numbers immediately preceding error messages. If more than one error message immediately follows a given line, the line number is printed only once. If only informational messages follow a given line, an asterisk is printed next to the line number.

# Chapter 2. Examples of JCL for Generation under MVS

This chapter contains examples of JCL for generating your NCP under the MVS operating system. Although you can find most of these examples on the SSP tape sent from IBM Software Distribution, they are also supplied here for easy reference. Before using these examples, ensure that they reflect your operating environment.

This chapter includes examples of JCL for the following types of generations:

- A FASTRUN generation
- An NCP or PEP generation with output written to disk
- An NCP or PEP generation with output written to tape
- An NCP or PEP generation with user-written code using the NDF standard attachment facility
- An NCP or PEP generation with user-written code using the GENEND definition statement
- A dynamic reconfiguration generation.

## Example of a FASTRUN Generation

Before running a complete generation, you can run a FASTRUN generation to check your generation definition for syntax and definition errors without creating control blocks or link-edit control statements. Figure 5 on page 26 shows the JCL that generates an NCP load module using FASTRUN generation.

To run a FASTRUN generation:

- Code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for any release of SSP Version 3. You can also code FASTRUN=ON as a parameter in your JCL when calling NDF using SSP V3R2 and later releases. This example uses the FASTRUN parameter coded in the JCL.
- Ensure your JCL *does not* call the linkage editor. If the link-edit step is present, an error results.
- *Do not* define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

This example assumes you have *not* included any user-written code using keywords on the GENEND definition statement. If you did, however, you must include a SYSLIB DD statement in your JCL containing the user-written code table assembly and link-edit statements.

The following is an example of a FASTRUN generation.

```
//MVSFAST JOB   (ACCOUNT INFO),'NAME',MSGLEVEL=(1,1)
//*
//*********************************************************************
//*
//*  COPYRIGHT=NONE
//*
//*  EXAMPLE OF A FASTRUN GENERATION.
//*
//*  FASTRUN IS SET ON BY SPECIFYING IT ON THE FIRST
//*  EXECUTABLE STATEMENT IN THE GENERATION DEFINITION AND/OR BY
//*  CODING IT AS A PARAMETER IN THE JCL AS SHOWN IN THE
//*  FOLLOWING EXAMPLE.
//*
//*********************************************************************
//*
//NDF    EXEC  PGM=ICNRTNDF,REGION=6000K,PARM='FASTRUN=ON'
//*
//*          * THE LIBRARY WITH NDF AND IHR LOAD MODULES
//STEPLIB  DD  DSN=SYS1.SSPLIB,DISP=SHR
//*          * THE GENERATION DEFINITION DATA SET - CARD IMAGE DATA
//GENDECK  DD  DSN=NCPSRC(SAMPLE),DISP=SHR
//*          * THE REPORT DATA SET
//SYSPRINT DD  DSN=SAMPLE.LISTINGS(NDF),UNIT=SYSDA,
//             DISP=(NEW,CATLG),
//             SPACE=(CYL,(17,3,1)),DCB=BLKSIZE=3630
//*          * THE SUMMARY DATA SET
//PRINTER  DD  SYSOUT=A
//*
//*********************************************************************
//*
//*          * THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH
//*          * VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
//*          * USERGEN IS SPECIFIED.
//* //DBWORKFL DD  DSN=&&WORKF,DISP=(,DELETE),UNIT=SYSDA,
//* //            SPACE=(CYL,(1,1))
//*          * IF NEWDEFN=YES OR NEWDEFN=(YES,ECHO) IS SPECIFIED IN
//*          * THE GENERATION DEFINITION, A "DD" STATEMENT SIMILAR
//*          * TO THE FOLLOWING IS NEEDED.
//* //NEWDEFN  DD  DSN=SAMPLE.NEWDEFN,UNIT=SYSDA,
//* //            DISP=(NEW,CATLG),SPACE=(TRK,(4,2)),
//* //            DCB=BLKSIZE=3200
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN VALIDATION, PRINT REPORT DATA SET
//*********************************************************************
//*
//ERRV   EXEC  PGM=IEBGENER,COND=(1,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.SYSPRINT,VOL=REF=*.NDF.SYSPRINT,
//             UNIT=SYSDA,DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//
```

*Figure 5. Example of a FASTRUN generation (MVS)*

# Example of an NCP or PEP Generation with Output Written to Disk

When running a standard NCP or PEP generation, you supply your generation definition as input and specify the various input and output data sets in your JCL. You can specify that input and output data sets be written to disk or tape. Figure 6 on page 28 shows the JCL that generates an NCP load module for an IBM 3745 Communication Controller with output data sets written to disk.

When reading this example, remember the following differences among the communication controllers:

- The JCL is slightly different.
- The NCP definition statement library used for the NDF job step may be different.
- The ddname for the library of preassembled NCP object modules in the link-edit step may be different.

**IBM 3705, 3720, or 3725 Communication Controller:** When running the link-edit step for a standard NCP or PEP generation on the IBM 3705, 3720, or 3725 Communication Controller, specify the ALIGN2 option in the JCL for the link-edit step. ALIGN2 ensures that certain control sections within the load module are aligned on 2KB page boundaries. If you do not specify ALIGN2, the default is alignment on 4KB page boundaries, which may use excessive communication controller storage.

**IBM 3745 Communication Controller:** *Do not specify* ALIGN2. The default is alignment on 4KB page boundaries, the correct alignment for the IBM 3745 Communication Controller.

The following list shows when to specify ALIGN2 as the link-edit step:

| Communication Controller | ALIGN2 |
|---|---|
| 3705 | Yes |
| 3720 | Yes |
| 3725 | Yes |
| 3745 | No. |

The following is an example of an NCP or PEP generation with output written to disk.

```
//MVSNCP JOB (ACCOUNT INFO),'NAME',MSGLEVEL=(1,1)
//*
//*********************************************************************
//*
//*  COPYRIGHT=NONE
//*
//*  EXAMPLE OF AN NCP GENERATION WITH ALL LISTINGS WRITTEN
//*  TO DISK.  (THIS EXAMPLE ASSUMES 3380 DISK DRIVES.)
//*
//*  THIS JOB CAN ALSO BE USED FOR A GENERATION FOR IBM SPECIAL
//*  PRODUCTS WHEN THE MACROS AND OBJECT MODULES FOR THOSE PRODUCTS
//*  HAVE BEEN MERGED INTO THE APPROPRIATE NCP LIBRARIES.
//*
//*********************************************************************
//*
//*  THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT
//*  LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOU
//*  CHANGED YOUR LIBRARY NAMES WHEN THE PRODUCT WAS INSTALLED, YOU
//*  MAY WANT TO UPDATE THIS TABLE.  BE SURE TO CHECK THE SYSLIB "DD"
//*  STATEMENTS AND THE OBJ37XX OR SNCPMOD1 "DD" STATEMENTS THAT
//*  DEFINE THESE LIBRARIES.
//*
//*  NOTE:  THE HIGH LEVEL QUALIFIER FOR ALL THE LIBRARIES IS "SYS1".
//*
```

*Figure 6 (Part 1 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

```
//*********************************************************************
//*
//*                                 M O D E L
//*
//*                3705          3725          3720          3745
//*             ------------------------------------------------
//*     V3/3705| MAC3705    |  NOT        |  NOT        |  NOT        |
//*            | OBJ3705    |  SUPPORTED  |  SUPPORTED  |  SUPPORTED  |
//*            |------------------------------------------------------|
//*     V3/3725| NOT        |  MAC3725    |  NOT        |  NOT        |
//*  V          | SUPPORTED  |  OBJ3725    |  SUPPORTED  |  SUPPORTED  |
//*  E          |------------------------------------------------------|
//*  R  V4R1    | NOT        |  MAC3725    |  NOT        |  NOT        |
//*  S          | SUPPORTED  |  OBJ3725    |  SUPPORTED  |  SUPPORTED  |
//*  I          |------------------------------------------------------|
//*  O  V4R2    | NOT        |  MAC3725    |  MAC3725    |  NOT        |
//*  N          | SUPPORTED  |  OBJ3725    |  OBJ3725    |  SUPPORTED  |
//*             |------------------------------------------------------|
//*     V4      | NOT        |  NOT        |  MAC3725    |  NOT        |
//*     SUBSET  | SUPPORTED  |  SUPPORTED  |  OBJ3725    |  SUPPORTED  |
//*             |------------------------------------------------------|
//*     V5R1    | NOT        |  NOT        |  MAC3725    |  MAC3725    |
//*             | SUPPORTED  |  SUPPORTED  |  OBJ3725    |  OBJ3725    |
//*             |------------------------------------------------------|
//*     V4R3 &  | NOT        |  SNCPMAC1   |  NOT        |  NOT        |
//*     LATER   | SUPPORTED  |  SNCPMOD1   |  SUPPORTED  |  SUPPORTED  |
//*             |------------------------------------------------------|
//*     V5R2 &  | NOT        |  NOT        |  SNCPMAC1   |  SNCPMAC1   |
//*     LATER   | SUPPORTED  |  SUPPORTED  |  SNCPMOD1   |  SNCPMOD1   |
//*              ------------------------------------------------------
//*
//*
//*                                 M O D E L
//*
//*           3745-130, 3745-150,
//*           3745-170, 3745-210,
//*           3745-410
//*           --------------
//*     V3    |  NOT       |
//*           |  SUPPORTED |
//*           |------------|
//*     V4    |  NOT       |
//*  V        |  SUPPORTED |
//*  E        |------------|
//*  R  V5R2 & |  NOT       |
//*  S  BEFORE |  SUPPORTED |
//*  I        |------------|
//*  O  V5R2.1 |  SNCPMAC1  |
//*  N  & LATER|  SNCPMOD1  |
//*           --------------
//*
//*
```

*Figure 6 (Part 2 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

```
//*********************************************************************
//*            RUN THE GENERATION STEP WITH NDF EXEC
//*********************************************************************
//*
//NDF    EXEC  PGM=ICNRTNDF,REGION=6000K
//*
//*          * THE LIBRARY WITH NDF AND IHR LOAD MODULES
//STEPLIB  DD  DSN=SYS1.SSPLIB,DISP=SHR
//*          * THE GENERATION DEFINITION DATA SET - CARD IMAGE DATA
//GENDECK  DD  DSN=NCPSRC(SAMPLE),DISP=SHR
//*          * THE REPORT DATA SET
//SYSPRINT DD  DSN=SAMPLE.LISTINGS(NDF),UNIT=SYSDA,
//             DISP=(NEW,CATLG),
//             SPACE=(CYL,(17,3,1)),DCB=BLKSIZE=3630
//*          * THE SUMMARY DATA SET
//PRINTER  DD  SYSOUT=A
//*
//*********************************************************************
//*
//*          * THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH
//*          * VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
//*          * USERGEN IS SPECIFIED.
//* //DBWORKFL DD  DSN=&&WORKF,DISP=(,DELETE),UNIT=SYSDA,
//* //            SPACE=(CYL,(1,1))
//*          * IF NEWDEFN=YES OR NEWDEFN=(YES,ECHO) IS SPECIFIED IN
//*          * THE GENERATION DEFINITION, A "DD" STATEMENT SIMILAR
//*          * TO THE FOLLOWING IS NEEDED.
//* //NEWDEFN  DD  DSN=SAMPLE.NEWDEFN,UNIT=SYSDA,
//* //            DISP=(NEW,CATLG),SPACE=(TRK,(4,2)),
//* //            DCB=BLKSIZE=3200
//*
//*********************************************************************
//*
//*          * THE TABLE 1 SOURCE DATA SET
//TBL1SRCE DD  DSN=&&SRCE1,DISP=(,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(3,2)),DCB=BLKSIZE=3200
//*          * THE TABLE 1 LISTING DATA SET
//TBL1LIST DD  DSN=SAMPLE.LISTINGS(TABLE1),UNIT=SYSDA,
//             DISP=(OLD,KEEP),
//             DCB=BLKSIZE=3630,VOL=REF=*.SYSPRINT
//*          * THE TABLE 1 OBJECT DATA SET
//TBL1OBJ  DD  DSN=SAMPLE.TBLOBJ(ICNTABL1),DISP=(NEW,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(3,1,1)),DCB=BLKSIZE=3200
//*          * THE TABLE 2 SOURCE DATA SET
//TBL2SRCE DD  DSN=&&SRCE2,DISP=(,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(1,1)),DCB=BLKSIZE=3200
//*          * THE TABLE 2 LISTING DATA SET
//TBL2LIST DD  DSN=SAMPLE.LISTINGS(TABLE2),UNIT=SYSDA,
//             DISP=(OLD,KEEP),
//             DCB=BLKSIZE=3630,VOL=REF=*.SYSPRINT
//*          * THE TABLE 2 OBJECT DATA SET
//TBL2OBJ  DD  DSN=SAMPLE.TBLOBJ(ICNTABL2),DISP=(OLD,KEEP),
//             DCB=BLKSIZE=3200,VOL=REF=*.TBL1OBJ
//*
```

*Figure 6 (Part 3 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

```
//*********************************************************************
//*
//*          * WORK DATA SET FOR THE TABLE ASSEMBLIES
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(10,2)),DISP=(,DELETE)
//*          * THE NCP MACRO LIBRARY
//SYSLIB   DD  DSN=SYS1.SNCPMAC1,DISP=SHR
//*          * THE LINK EDIT STATEMENTS DATA SET
//LNKSTMT  DD  DSN=&&LNKFL,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=BLKSIZE=3200
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN VALIDATION, PRINT NDF LISTING
//*********************************************************************
//*
//ERRV   EXEC  PGM=IEBGENER,COND=(1,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.SYSPRINT,VOL=REF=*.NDF.SYSPRINT,
//             UNIT=SYSDA,DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN TABLE 1 ASSEMBLY, PRINT TABLE 1
//*********************************************************************
//*
//ERR1   EXEC  PGM=IEBGENER,COND=(10,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.TBL1LIST,VOL=REF=*.NDF.TBL1LIST,
//             UNIT=SYSDA,DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN TABLE 2 ASSEMBLY, PRINT TABLE 2
//*********************************************************************
//*
//ERR2   EXEC  PGM=IEBGENER,COND=(100,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.TBL2LIST,VOL=REF=*.NDF.TBL2LIST,
//             UNIT=SYSDA,DISP=(OLD,KEEP)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
```

*Figure 6 (Part 4 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

```
//*********************************************************************
//*        IF NO NDF ERROR OCCURRED, THEN RUN LINK EDIT EXEC
//*********************************************************************
//*
//*          * NOTE: FOR IBM 3705/3720/3725 YOU MUST ADD ALIGN2
//*          * TO THE PARM LIST ON THE FOLLOWING CALL.
//*          * DO NOT SPECIFY ALIGN2 FOR THE IBM 3745.
//LINK   EXEC  PGM=HEWL,COND=(0,LT,NDF),REGION=6000K,
//             PARM='LIST,NCAL,MAP,LET,SIZE=(7800K,512K)'
//*          * THE DATA SET OF LINK EDIT CONTROL STATEMENTS FROM STEP 1
//SYSLIN   DD  DSN=*.NDF.LNKSTMT,VOL=REF=*.NDF.LNKSTMT,
//             DISP=(OLD,PASS)
//*          * THE LIBRARY OF TABLE OBJECTS PASSED FROM STEP 1
//SYSPUNCH DD  DSN=SAMPLE.TBLOBJ,DISP=(OLD,KEEP)
//*********************************************************************
//* NOTE THAT THE DDNAME ON THE NEXT JCL STATEMENT IS ANCPMOD1, AND THE
//* DATASET NAME INDICATES SNCPMOD1.  THIS IS REQUIRED TO FACILITATE
//* MAINTENANCE OF NCP LOAD MODULES VIA NORMAL SMP APPLY PROCESSING.
//* BY VARYING THE DATASET NAME, YOU CAN SPECIFY WHETHER YOU WANT
//* THE LINKEDIT TO BE DONE WITH THE NCP TARGET LIBRARY (SNCPMOD1) OR
//* NCP'S DISTRIBUTION LIBRARY (ANCPMOD1).
//*********************************************************************
//ANCPMOD1 DD  DSN=SYS1.SNCPMOD1,DISP=SHR
//*          * THE LIBRARY OF NCP LOAD MODULES
//SYSLMOD  DD  DSN=NCPZZZZZ.LOADNCP,UNIT=SYSDA,DISP=(NEW,CATLG),
//             SPACE=(CYL,(6,2,1))
//*          * THE LINK EDIT WORK DATA SET
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(200,20))
//SYSPRINT DD  DSN=SAMPLE.LISTINGS(LKEDLIST),UNIT=SYSDA,
//             DISP=(OLD,KEEP),
//             SPACE=(TRK,(5,5)),DCB=BLKSIZE=3630
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN LINK EDIT, PRINT LINK EDIT DATA SET
//*********************************************************************
//*
//ERRL   EXEC  PGM=IEBGENER,COND=((4,GE,LINK),(0,LT,NDF))
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.LINK.SYSPRINT,VOL=REF=*.LINK.SYSPRINT,
//             UNIT=SYSDA,DISP=(OLD,KEEP)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        PREPARE LINK EDIT JCL STATEMENTS FOR SMP UTILITY
//*********************************************************************
//*
//SMP1   EXEC  PGM=IEBGENER,COND=((0,LT,NDF),(4,LT,LINK))
//SYSPRINT DD  DUMMY
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=SAMPLE.JCLIN,DISP=(NEW,CATLG),
//             UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUT1   DD  DATA,DLM=ZZ
```

*Figure 6 (Part 5 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

```
//*********************************************************************
//*          * NOTE: FOR IBM 3705/3720/3725 YOU MUST ADD ALIGN2
//*          * TO THE PARM LIST ON THE FOLLOWING CALL.
//*          * DO NOT SPECIFY ALIGN2 FOR THE IBM 3745.
//LINK   EXEC  PGM=HEWL,COND=(0,LT,NDF),REGION=6000K,
//             PARM='LIST,NCAL,MAP,LET,SIZE=(7800K,512K)'
//*********************************************************************
//*THE FOLLOWING DD STATEMENT HAS DDNAME OF ANCPMOD1 AND DSN INDICATING
//*SNCPMOD1.  THIS IS REQUIRED BY SMP TO FACILITATE NCP LOAD MODULE
//*MAINTENANCE. THE DSN SHOULD REFLECT THE ACTUAL DATASET USED BY THE
//*LINKEDIT STEP OF THE GENERATION PROCESS.
//*********************************************************************
//ANCPMOD1 DD  DSN=SYS1.SNCPMOD1,DISP=SHR
//*        DD  STATEMENTS FOR NRF AND NTO OBJECT LIBRARIES
//*            AT VARIOUS RELEASES OR MAINTENANCE LEVELS
//SYSLMOD  DD  DSN=NCPZZZZZ.LOADNCP,DISP=SHR
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(800,100))
//SYSPRINT DD  SYSOUT=A
//SYSPUNCH DD  DSN=SAMPLE.TBLOBJ,DISP=(OLD,KEEP)
//SYSLIN   DD  *
ZZ
//*********************************************************************
//*        PREPARE LINK EDIT STATEMENTS DATA SET FOR SMP UTILITY
//*********************************************************************
//*
//SMP2   EXEC  PGM=IEBGENER,REGION=2048K,
//             COND=((0,LT,NDF),(4,LT,LINK))
//SYSPRINT DD  DUMMY
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=SAMPLE.JCLIN,DISP=(MOD,KEEP),UNIT=SYSDA
//SYSUT1   DD  DSN=*.NDF.LNKSTMT,VOL=REF=*.NDF.LNKSTMT,
//             DISP=(OLD,PASS)
//*********************************************************************
//*        RUN THE SMP UTILITY EXEC
//*********************************************************************
//*          * RUN SMP3 WHEN YOU ARE PREPARED TO ACCEPT THE NEW NCP
//*          * CHANGE THIS EXEC STEP FOR OTHER VERSIONS OF NCP
//*          * RUN SMP PROGRAM FOR NCP V5, WHERE XXXX IS MODEL NUMBER
//* //SMP3   EXEC  NCPZZZZZ
//* //             COND=((0,LT,NDF),(4,LT,LINK))
//* //SMPJCLIN DD  DSN=SAMPLE.JCLIN,DISP=(OLD,KEEP),UNIT=SYSDA
//*          * IF YOU ARE USING SMP4 THEN DELETE LINE WITH
//*          * "SET BDY(NCPTGT)"
//* //SYSIN    DD  *
//*   SET BDY(NCPTGT) .
//*   JCLIN .
//
```

*Figure 6 (Part 6 of 6). Example of an NCP or PEP generation with output written to disk (MVS)*

# Example of an NCP or PEP Generation with Output Written to Tape

When running a standard NCP or PEP generation, you supply your generation definition as input and specify the various input and output data sets in your JCL. You can specify that input and output data sets be written to disk or tape. If you detect an error while generating or running your NCP, you can write certain listing data sets to tape. Figure 7 on page 35 shows the JCL for generating an NCP load module for an IBM 3725 Communication Controller with listing data sets from the table 1 assembly, the table 2 assembly, and the link-edit written to tape.

When reading this example, remember the following differences among the communication controllers:

- The JCL is slightly different.
- The NCP definition statement library used for the NDF job step may be different.
- The ddname for the library of preassembled NCP object modules in the link-edit step may be different.

**IBM 3705, 3720, or 3725 Communication Controller:** When running the link-edit step for a standard NCP or PEP generation on the IBM 3705, 3720, or 3725 Communication Controller, specify the ALIGN2 option in the JCL for the link-edit step. ALIGN2 ensures that certain control sections within the load module are aligned on 2KB page boundaries. If you do not specify ALIGN2, the default is alignment on 4KB page boundaries, which may use excessive controller storage.

**IBM 3745 Communication Controller:** *Do not specify* ALIGN2. The default is alignment on 4KB page boundaries, the correct alignment for the IBM 3745 Communication Controller.

The following list shows when to specify ALIGN2 as the link-edit step:

| Communication Controller | ALIGN2 |
|---|---|
| 3705 | Yes |
| 3720 | Yes |
| 3725 | Yes |
| 3745 | No. |

The following is an example of an NCP or PEP generation with output written to tape.

```
//MVSTAPE JOB   (ACCOUNT INFO),'NAME',MSGLEVEL=(1,1)
//*
//*  COPYRIGHT=NONE
//*
//*  EXAMPLE OF AN NCP GENERATION WITH SOME LISTINGS WRITTEN
//*  TO TAPE.
//*
//*  THE TABLE 1, TABLE 2 AND LINK EDIT LISTINGS ARE WRITTEN TO
//*  TAPE AND PRINTED ONLY WHEN SEVERE ERRORS OCCUR.
//*
//*********************************************************************
//*
//*  THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT
//*  LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOU
//*  CHANGED YOUR LIBRARY NAMES WHEN THE PRODUCT WAS INSTALLED, YOU
//*  MAY WANT TO UPDATE THIS TABLE.  BE SURE TO CHECK THE SYSLIB "DD"
//*  STATEMENTS AND THE OBJ37XX OR SNCPMOD1 "DD" STATEMENTS THAT
//*  DEFINE THESE LIBRARIES.
//*
//*  NOTE:  THE HIGH LEVEL QUALIFIER FOR ALL THE LIBRARIES IS "SYS1".
//*
```

*Figure 7 (Part 1 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*
//*                                  M O D E L
//*
//*                 3705          3725          3720          3745
//*               ------------------------------------------------
//*     V3/3705  |  MAC3705    |  NOT        |  NOT        |  NOT        |
//*              |  OBJ3705    |  SUPPORTED  |  SUPPORTED  |  SUPPORTED  |
//*              |-------------------------------------------------|
//*     V3/3725  |  NOT        |  MAC3725    |  NOT        |  NOT        |
//*  V           |  SUPPORTED  |  OBJ3725    |  SUPPORTED  |  SUPPORTED  |
//*  E           |-------------------------------------------------|
//*  R  V4R1     |  NOT        |  MAC3725    |  NOT        |  NOT        |
//*  S           |  SUPPORTED  |  OBJ3725    |  SUPPORTED  |  SUPPORTED  |
//*  I           |-------------------------------------------------|
//*  O  V4R2     |  NOT        |  MAC3725    |  MAC3725    |  NOT        |
//*  N           |  SUPPORTED  |  OBJ3725    |  OBJ3725    |  SUPPORTED  |
//*              |-------------------------------------------------|
//*     V4       |  NOT        |  NOT        |  MAC3725    |  NOT        |
//*     SUBSET   |  SUPPORTED  |  SUPPORTED  |  OBJ3725    |  SUPPORTED  |
//*              |-------------------------------------------------|
//*     V5R1     |  NOT        |  NOT        |  MAC3725    |  MAC3725    |
//*              |  SUPPORTED  |  SUPPORTED  |  OBJ3725    |  OBJ3725    |
//*              |-------------------------------------------------|
//*     V4R3 &   |  NOT        |  SNCPMAC1   |  NOT        |  NOT        |
//*     LATER    |  SUPPORTED  |  SNCPMOD1   |  SUPPORTED  |  SUPPORTED  |
//*              |-------------------------------------------------|
//*     V5R2 &   |  NOT        |  NOT        |  SNCPMAC1   |  SNCPMAC1   |
//*     LATER    |  SUPPORTED  |  SUPPORTED  |  SNCPMOD1   |  SNCPMOD1   |
//*               -------------------------------------------------
//*
//*
//*                                  M O D E L
//*
//*            3745-130, 3745-150,
//*            3745-170, 3745-210,
//*            3745-410
//*              -------------
//*     V3       |  NOT        |
//*              |  SUPPORTED  |
//*              |-------------
//*     V4       |  NOT        |
//*  V           |  SUPPORTED  |
//*  E           |-------------
//*  R  V5R2 &   |  NOT        |
//*  S  BEFORE   |  SUPPORTED  |
//*  I           |-------------
//*  O  V5R2.1   |  SNCPMAC1   |
//*  N  & LATER  |  SNCPMOD1   |
//*              -------------
//*
//*
```

*Figure 7 (Part 2 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*            INITIALIZE THE TAPE
//*********************************************************************
//*
//*          * RUN STEP0 IF YOU NEED TO INITIALIZE YOUR TAPE
//*//STEP0  EXEC  PGM=IEHINITT
//*//SYSPRINT DD  SYSOUT=A
//*//TAPE     DD  UNIT=(TP6250,1,DEFER),DCB=DEN=4,VOL=(PRIVATE,RETAIN),
//*//             DISP=(NEW,PASS)
//*//SYSIN    DD  *
//*TAPE  INITT  SER=XXXXXX
//*
//*********************************************************************
//*            RUN THE GENERATION STEP WITH NDF EXEC
//*********************************************************************
//*
//NDF    EXEC  PGM=ICNRTNDF,REGION=6000K
//*
//*          * THE LIBRARY WITH NDF AND IHR LOAD MODULES
//STEPLIB  DD  DSN=SYS1.SSPLIB,DISP=SHR
//*          * THE GENERATION DEFINITION DATA SET - CARD IMAGE DATA
//GENDECK  DD  DSN=NCPSRC(SAMPLE),DISP=SHR
//*          * THE REPORT DATA SET
//SYSPRINT DD  DSN=SAMPLE.LISTINGS.NDF,UNIT=(TP6250,,DEFER),
//             LABEL=(1,SL),DISP=(OLD,KEEP),
//             DCB=(DEN=4,BLKSIZE=3630,LRECL=121,RECFM=FBM),
//             VOL=(,RETAIN,,SER=XXXXXX)
//*          * THE SUMMARY DATA SET
//PRINTER  DD  SYSOUT=A
//*
//*********************************************************************
//*
//*          * THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH
//*          * VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
//*          * USERGEN IS SPECIFIED.
//* //DBWORKFL DD  DSN=&&WORKF,DISP=(,DELETE),UNIT=SYSDA,
//* //             SPACE=(CYL,(1,1))
//*          * IF NEWDEFN=YES OR NEWDEFN=(YES,ECHO) IS SPECIFIED IN
//*          * THE GENERATION DEFINITION, A "DD" STATEMENT SIMILAR
//*          * TO THE FOLLOWING IS NEEDED.
//* //NEWDEFN  DD  DSN=SAMPLE.NEWDEFN,UNIT=SYSDA,
//* //             DISP=(NEW,CATLG),SPACE=(TRK,(4,2)),
//* //             DCB=BLKSIZE=3200
//*
```

*Figure 7 (Part 3 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*
//*          * THE TABLE 1 SOURCE DATA SET
//TBL1SRCE DD  DSN=&&SRCE1,DISP=(,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(3,2)),DCB=BLKSIZE=3200
//*          * THE TABLE 1 LISTING DATA SET
//TBL1LIST DD  DSN=SAMPLE.LISTINGS.TABLE1,UNIT=(TP6250,,DEFER),
//             LABEL=(2,SL),DISP=(OLD,KEEP),
//             DCB=(DEN=4,BLKSIZE=3630,LRECL=121,RECFM=FBM),
//             VOL=(,RETAIN,,SER=XXXXXX)
//*          * THE TABLE 1 OBJECT DATA SET
//TBL1OBJ  DD  DSN=SAMPLE.TBLOBJ(ICNTABL1),DISP=(NEW,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(3,1,1)),DCB=BLKSIZE=3200
//*          * THE TABLE 2 SOURCE DATA SET
//TBL2SRCE DD  DSN=&&SRCE2,DISP=(,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(1,1)),DCB=BLKSIZE=3200
//*          * THE TABLE 2 LISTING DATA SET
//TBL2LIST DD  DSN=SAMPLE.LISTINGS.TABLE2,UNIT=(TP6250,,DEFER),
//             LABEL=(3,SL),DISP=(OLD,KEEP),
//             DCB=(DEN=4,BLKSIZE=3630,LRECL=121,RECFM=FBM),
//             VOL=(,RETAIN,,SER=XXXXXX)
//*          * THE TABLE 2 OBJECT DATA SET
//TBL2OBJ  DD  DSN=SAMPLE.TBLOBJ(ICNTABL2),DISP=(OLD,KEEP),
//             DCB=BLKSIZE=3200,VOL=REF=*.TBL1OBJ
//*
//*********************************************************************
//*
//*          * WORK DATA SET FOR THE TABLE ASSEMBLIES
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(10,2)),DISP=(,DELETE)
//*          * THE NCP MACRO LIBRARY
//SYSLIB   DD  DSN=SYS1.SNCPMAC1,DISP=SHR
//*          * THE LINK EDIT STATEMENTS DATA SET
//LNKSTMT  DD  DSN=&&LNKFL,DISP=(,PASS),
//             UNIT=SYSDA,SPACE=(CYL,(1,1)),DCB=BLKSIZE=3200
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN VALIDATION, PRINT REPORT DATA SET
//*********************************************************************
//*
//ERRV   EXEC  PGM=IEBGENER,COND=(1,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.SYSPRINT,VOL=REF=*.NDF.SYSPRINT,
//             UNIT=TAPE,LABEL=(1,SL),DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
```

*Figure 7 (Part 4 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*        IF ERROR OCCURRED IN TABLE 1 ASSEMBLY, PRINT TABLE 1
//*********************************************************************
//*
//ERR1   EXEC  PGM=IEBGENER,COND=(10,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.TBL1LIST,VOL=REF=*.NDF.TBL1LIST,
//             UNIT=TAPE,LABEL=(2,SL),DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN TABLE 2 ASSEMBLY, PRINT TABLE 2
//*********************************************************************
//*
//ERR2   EXEC  PGM=IEBGENER,COND=(100,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.TBL2LIST,VOL=REF=*.NDF.TBL2LIST,
//             UNIT=TAPE,LABEL=(3,SL),DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        IF NO NDF ERROR OCCURRED, THEN RUN LINK EDIT EXEC
//*********************************************************************
//*
//*          * NOTE: FOR IBM 3705/3720/3725 YOU MUST ADD THE
//*          * ALIGN2 PARAMETER TO THE FOLLOWING PARM LIST.
//*          * DO NOT SPECIFY ALIGN2 FOR THE IBM 3745.
//LINK   EXEC  PGM=HEWL,COND=(0,LT,NDF),REGION=6000K,
//             PARM='LIST,NCAL,MAP,LET,SIZE=(7800K,512K)'
//*          * THE DATA SET OF LINK EDIT CONTROL STATEMENTS FROM STEP 1
//SYSLIN   DD  DSN=*.NDF.LNKSTMT,VOL=REF=*.NDF.LNKSTMT,
//             DISP=(OLD,PASS)
//*          * THE LIBRARY OF TABLE OBJECTS PASSED FROM STEP 1
//SYSPUNCH DD  DSN=SAMPLE.TBLOBJ,DISP=(OLD,KEEP)
//*********************************************************************
//* NOTE THAT THE DDNAME ON THE NEXT JCL STATEMENT IS ANCPMOD1, AND THE
//* DATASET NAME INDICATES SNCPMOD1.  THIS IS REQUIRED TO FACILITATE
//* MAINTENANCE OF NCP LOAD MODULES VIA NORMAL SMP APPLY PROCESSING.
//* BY VARYING THE DATASET NAME, YOU CAN SPECIFY WHETHER YOU WANT
//* THE LINKEDIT TO BE DONE WITH THE NCP TARGET LIBRARY (SNCPMOD1) OR
//* NCP'S DISTRIBUTION LIBRARY (ANCPMOD1).
//*********************************************************************
//ANCPMOD1 DD  DSN=SYS1.SNCPMOD1,DISP=SHR
//*          * THE LIBRARY OF NCP LOAD MODULES
//SYSLMOD  DD  DSN=NCPZZZZZ.LOADNCP,UNIT=SYSDA,DISP=(NEW,CATLG),
//             SPACE=(CYL,(6,2,1))
//*          * THE LINK EDIT WORK DATA SET
//SYSUT1   DD  UNIT=SYSDA,SPACE=(1024,(200,20))
//*          * THE LINK EDIT LISTING DATA SET
//SYSPRINT DD  DSN=SAMPLE.LISTINGS.LKEDLIST,UNIT=(TP6250,,DEFER),
//             LABEL=(5,SL),DISP=(OLD,KEEP),
//             DCB=(DEN=4,BLKSIZE=3630,LRECL=121,RECFM=FBM),
//             VOL=(,RETAIN,,SER=XXXXXX)
//*
```

*Figure 7 (Part 5 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*        IF ERROR OCCURRED IN LINK EDIT, PRINT LINK EDIT DATA SET
//*********************************************************************
//*
//ERRL   EXEC  PGM=IEBGENER,COND=((4,GE,LINK),(0,LT,NDF))
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.LINK.SYSPRINT,VOL=REF=*.LINK.SYSPRINT,
//             UNIT=TAPE,LABEL=(5,SL),DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        PREPARE LINK EDIT JCL STATEMENTS FOR SMP UTILITY
//*********************************************************************
//*
//SMP1   EXEC  PGM=IEBGENER,COND=((0,LT,NDF),(4,LT,LINK))
//SYSPRINT DD  DUMMY
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=SAMPLE.JCLIN,DISP=(NEW,CATLG),
//             UNIT=SYSDA,SPACE=(TRK,(1,1)),
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB)
//SYSUT1   DD  DATA,DLM=ZZ
//*********************************************************************
//*          * NOTE: FOR IBM 3705/3720/3725 YOU MUST ADD THE
//*          * ALIGN2 PARAMETER TO THE FOLLOWING PARM LIST.
//*          * DO NOT SPECIFY ALIGN2 FOR THE IBM 3745.
//LINK   EXEC  PGM=HEWL,COND=(0,LT,NDF),REGION=6000K,
//             PARM='LIST,NCAL,MAP,LET,SIZE=(7800K,512K)'
//*********************************************************************
//*THE FOLLOWING DD STATEMENT HAS DDNAME OF ANCPMOD1 AND DSN INDICATING
//*SNCPMOD1.  THIS IS REQUIRED BY SMP TO FACILITATE NCP LOAD MODULE
//*MAINTENANCE.  THE DSN SHOULD REFLECT THE ACTUAL DATASET USED BY THE
//*LINKEDIT STEP OF THE GENERATION PROCESS.
//*********************************************************************
//ANCPMOD1 DD  DSN=SYS1.SNCPMOD1,DISP=SHR
//*        DD  STATEMENTS FOR NRF AND NTO OBJECT LIBRARIES
//*            AT VARIOUS RELEASES OR MAINTENANCE LEVELS
//SYSLMOD  DD  DSN=NCPZZZZZ.LOADNCP,DISP=SHR
//SYSUT1   DD  UNIT=SYSDA,SPACE=(2048,(800,100))
//SYSPRINT DD  SYSOUT=A
//SYSPUNCH DD  DSN=SAMPLE.TBLOBJ,DISP=(OLD,KEEP)
//SYSLIN   DD  *
ZZ
```

*Figure 7 (Part 6 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

```
//*********************************************************************
//*        PREPARE LINK EDIT STATEMENTS DATA SET FOR SMP UTILITY
//*********************************************************************
//*
//SMP2   EXEC  PGM=IEBGENER,REGION=2048K,
//             COND=((0,LT,NDF),(4,LT,LINK))
//SYSPRINT DD  DUMMY
//SYSIN    DD  DUMMY
//SYSUT2   DD  DSN=SAMPLE.JCLIN,DISP=(MOD,KEEP),UNIT=SYSDA
//SYSUT1   DD  DSN=*.NDF.LNKSTMT,VOL=REF=*.NDF.LNKSTMT,
//             DISP=(OLD,PASS)
//*********************************************************************
//*        RUN THE SMP UTILITY EXEC
//*********************************************************************
//*          * RUN SMP3 WHEN YOU ARE PREPARED TO ACCEPT THE NEW GEN
//*          * CHANGE THIS EXEC STEP FOR OTHER VERSIONS OF NCP
//*          * RUN SMP PROGRAM FOR NCP V5, WHERE XXXX IS MODEL NUMBER
//* //SMP3   EXEC  NCPZZZZZ
//* //             COND=((0,LT,NDF),(4,LT,LINK))
//* //SMPJCLIN DD  DSN=SAMPLE.JCLIN,DISP=(OLD,KEEP),UNIT=SYSDA
//*          * IF YOU ARE USING SMP4 THEN DELETE LINE WITH
//*          * "SET BDY(NCPTGT)"
//* //SYSIN    DD  *
//*   SET BDY(NCPTGT) .
//*   JCLIN .
//
```

*Figure 7 (Part 7 of 7). Example of an NCP or PEP generation with output written to tape (MVS)*

## Example of an NCP or PEP Generation with User-Written Code Using the NDF Standard Attachment Facility

To run an NCP or PEP generation with IBM special products or user-written code using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing.

Figure 8 on page 42 shows the JCL for generating user-written code and NCP using the NDF standard attachment facility. For more information about running this type of generation, see page 14. You can run this type of user-written code generation *only* if you are using SSP V3R2 or a later release to generate NCP V4 Subset, NCP V4R2 or a later release, or NCP Version 5.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation load modules to be loaded in the generation. Each application must have its own generation load module. You can specify up to 25 generation load modules.

- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation load modules.
- Modify the JCL for a standard NCP or PEP generation to include the ddnames for the NEWDEFN data set, the DBWORKFL data set, and the libraries for user-supplied modules.

The following examples show how to code the NEWDEFN data set in the NDF step of the JCL and how to code the ULIB data sets or the libraries for user-written code modules in the link-edit step of the JCL.

```
//* EXAMPLE FOR INCLUDING NEWDEFN IN THE NDF STEP IN THE JCL
//*
//NEWDEFN  DD  DSN=SAMPLE.NEWDEFN,UNIT=SYSDA
               DISP=(NEW,CATLG),SPACE=(CYL,(1,1)),
               DCB=BLKSIZE=3200
//*
//*


//* EXAMPLE FOR INCLUDING STEPLIB
//STEPLIB DD DSN=SYS1.SSPLIB,DISP=SHR
//        DD DSN=USER.LIB,DISP=SHR
//*
//*


//*
//* EXAMPLE OF LIBRARIES CONCATENATED ON THE SYSLIB CHAIN
//* IN THE NDF STEP OF A NCP GENERATION WITH USER CODE
//*
//SYSLIB   DD  DSN=SYS1.SNCPMAC1,DISP=SHR
//         DD  DSN=USER.MACLIB



//* EXAMPLE FOR INCLUDING LIBRARIES FOR USER-WRITTEN CODE
//* MODULES IN THE LINK EDIT STEP
//*
//*
//* LIBRARIES FOR USER-WRITTEN CODE OBJECT MODULES
//USER1    DD  DSN=USER.OBJ1,DISP=SHR
  .        .           .
  .        .           .
  .        .           .
//USERN    DD  DSN=USER.OBJN,DISP=SHR
```

*Figure 8. Example of an NCP or PEP generation with user-written code using the NDF standard attachment facility (MVS)*

# Example of an NCP or PEP Generation with User-Written Code Using the GENEND Definition Statement

To run an NCP or PEP generation with IBM special products or user-written code without using the NDF standard attachment facility, or to generate user-written code using SSP Version 3, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of the link-edit statements by coding keywords on the GENEND definition statement.

Figure 9 on page 44 shows the JCL for generating IBM special products or user-written code using the GENEND definition statement. For more information about running this type of generation, see page 15. You can run this type of user-written code generation if you are using SSP Version 3 to generate any version of NCP.

Before you generate IBM special products or user-written code using the GENEND definition statement, ensure that you:

- Assemble the user-written routines and code the link-edit statements for the routines
- Code the appropriate keywords on the GENEND definition statement for your user-written routines
- Place the members with SRCLO or SRCHI code in a definition statement library (SYSLIB) available to NDF
- Place all members that contain INCLUDE or ORDER link-edit control statements in a definition statement library in the NDF SYSLIB chain
- Place all definition statements in the NDF SYSLIB chain
- Modify the JCL to include the SYSLIB chain and the ULIB or user object code library ddname statement.

The generation validation phase of NDF reads the link-edit control statements and writes them into the same data set as the standard NCP link-edit control statements.

The following are examples of how to specify the SYSLIB chain and the link-edit ULIB statement. Code the SYSLIB chain in the NDF step in the JCL for a standard NCP or PEP generation, and code the ULIB statement in the link-edit step.

**Note:** In these examples, library names such as SNCPMAC1 and SNCPMOD1 are release dependent.

```
//*
//* EXAMPLE OF LIBRARIES CONCATENATED ON THE SYSLIB CHAIN
//* IN THE NDF STEP OF A NCP GENERATION WITH USER CODE
//*
//SYSLIB   DD  DSN=SYS1.SNCPMAC1,DISP=SHR
//         DD  DSN=USER.MACLB

//* EXAMPLE OF THE ULIB DATA DEFINITION FROM THE LINK EDIT
//* STEP OF A NCP GENERATION WITH USER CODE
//*
//* THIS EXAMPLE DEFINES A LIBRARY OF NPSI PREASSEMBLED MODULES
//* THE DDNAME WILL VARY FOR OTHER IBM SPECIAL PRODUCTS
//*
//ULIB     DD  DSN=NPSI,UNIT=SYSDA,DISP=SHR
//*
//*
```

*Figure 9. Example of an NCP or PEP generation with user-written code using the GENEND definition statement (MVS)*

# Example of a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, you can use the text data set from a dynamic reconfiguration generation. Figure 10 on page 45 shows the JCL for dynamic reconfiguration generation.

To use this type of generation, ensure that you coded the original NCP to allow dynamic reconfiguration. The dynamic reconfiguration generation produces a text data set that the access method can use to modify NCP.

**Note:** VTAM has its own dynamic reconfiguration procedures that do not require you to use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Network Implementation Guide*.

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration data set consisting of ADD or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration data set is the input for the dynamic reconfiguration generation. This type of generation produces a text data set that the access method uses to modify an NCP already running in a communication controller. For information on using ADD, DELETE, PU, or LU definition statements, see *NCP, SSP, and EP Resource Definition Guide.*

A dynamic reconfiguration generation requires one table assembly and no link-edit. The following is an example of JCL for a dynamic reconfiguration generation.

```
//MVSDR JOB (ACCOUNT INFO),'NAME',MSGLEVEL=(1,1)
//*
//*********************************************************************
//*
//*  COPYRIGHT=NONE
//*
//*  EXAMPLE OF A DYNAMIC RECONFIGURATION GENERATION.
//*
//*********************************************************************
//*
//*  THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT
//*  LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOU
//*  CHANGED YOUR LIBRARY NAMES WHEN THE PRODUCT WAS INSTALLED, YOU
//*  MAY WANT TO UPDATE THIS TABLE.  BE SURE TO CHECK THE SYSLIB "DD"
//*  STATEMENTS AND THE OBJ37XX OR OBJNCP "DD" STATEMENTS THAT
//*  DEFINE THESE LIBRARIES.
//*
//*  NOTE:  THE HIGH LEVEL QUALIFIER FOR ALL THE LIBRARIES IS "SYS1".
//*
```

*Figure 10 (Part 1 of 4). Example of a dynamic reconfiguration generation (MVS)*

```
//*********************************************************************
//*
//*                                M O D E L
//*
//*                3705         3725         3720         3745
//*             ------------------------------------------------
//*     V3/3705| MAC3705    |  NOT       |  NOT       |  NOT       |
//*            |            |  SUPPORTED |  SUPPORTED |  SUPPORTED |
//*            |------------------------------------------------|
//*     V3/3725|  NOT       |  MAC3725   |  NOT       |  NOT       |
//*  V          |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |
//*  E          |------------------------------------------------|
//*  R  V4R1    |  NOT       |  MAC3725   |  NOT       |  NOT       |
//*  S          |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |
//*  I          |------------------------------------------------|
//*  O  V4R2    |  NOT       |  MAC3725   |  MAC3725   |  NOT       |
//*  N          |  SUPPORTED |            |            |  SUPPORTED |
//*             |------------------------------------------------|
//*     V4      |  NOT       |  NOT       |  MAC3725   |  NOT       |
//*     SUBSET  |  SUPPORTED |  SUPPORTED |            |  SUPPORTED |
//*             |------------------------------------------------|
//*     V5R1    |  NOT       |  NOT       |  MAC3725   |  MAC3725   |
//*             |  SUPPORTED |  SUPPORTED |            |            |
//*             |------------------------------------------------|
//*     V4R3 &  |  NOT       |  SNCPMAC1  |  NOT       |  NOT       |
//*     LATER   |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |
//*             |------------------------------------------------|
//*     V5R2 &  |  NOT       |  NOT       |  SNCPMAC1  |  SNCPMAC1  |
//*     LATER   |  SUPPORTED |  SUPPORTED |            |            |
//*              ------------------------------------------------
//*
//*
//*                                M O D E L
//*
//*           3745-130, 3745-150,
//*           3745-170, 3745-210,
//*           3745-410
//*           -------------
//*     V3    |  NOT      |
//*           |  SUPPORTED |
//*           |-------------
//*     V4    |  NOT      |
//*  V        |  SUPPORTED |
//*  E        |-------------
//*  R  V5R2 & |  NOT      |
//*  S  BEFORE |  SUPPORTED |
//*  I        |-------------
//*  O  V5R2.1 |  SNCPMAC1  |
//*  N  & LATER|  SNCPMOD1  |
//*           -------------
//*
//*
```

*Figure 10 (Part 2 of 4). Example of a dynamic reconfiguration generation (MVS)*

```
//*********************************************************************
//*           RUN THE GENERATION STEP WITH NDF EXEC
//*********************************************************************
//*
//NDF    EXEC  PGM=ICNRTNDF,REGION=6000K
//*
//*          * THE LIBRARY WITH NDF AND IHR LOAD MODULES
//STEPLIB  DD  DSN=SYS1.SSPLIB,DISP=SHR
//*          * THE GENERATION DEFINITION DATA SET - CARD IMAGE DATA
//GENDECK  DD  DSN=NCPSRC(SAMPLE),DISP=SHR
//*          * THE REPORT DATA SET
//SYSPRINT DD  DSN=SAMPLE.LISTINGS(NDF),UNIT=SYSDA,
//             DISP=(NEW,CATLG),
//             SPACE=(CYL,(17,3,1)),DCB=BLKSIZE=3630
//*          * THE SUMMARY DATA SET
//PRINTER  DD  SYSOUT=A
//*
//*********************************************************************
//*
//*          * THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH
//*          * VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
//*          * USERGEN IS SPECIFIED.
//* //DBWORKFL DD  DSN=&&WORKF,DISP=(,DELETE),UNIT=SYSDA,
//* //             SPACE=(CYL,(1,1))
//*
//*********************************************************************
//*
//*          * THE TABLE 1 SOURCE DATA SET
//TBL1SRCE DD  DSN=&&SRCE1,DISP=(,DELETE),UNIT=SYSDA,
//             SPACE=(CYL,(3,2)),DCB=BLKSIZE=3200
//*          * THE TABLE 1 LISTING DATA SET
//TBL1LIST DD  DSN=SAMPLE.LISTINGS(TABLE1),UNIT=SYSDA,
//             DISP=(OLD,KEEP),
//             DCB=BLKSIZE=3630,VOL=REF=*.SYSPRINT
//*          * THE TABLE 1 OBJECT DATA SET
//TBL1OBJ  DD  DSN=SAMPLE.TBLOBJ(ICNTABL1),DISP=(NEW,CATLG),
//             UNIT=SYSDA,SPACE=(CYL,(3,1,1)),DCB=BLKSIZE=3200
//*
//*********************************************************************
//*
//*          * WORK DATA SET FOR THE TABLE ASSEMBLIES
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(10,2)),DISP=(,DELETE)
//*          * THE NCP MACRO LIBRARY
//SYSLIB   DD  DSN=SYS1.SNCPMAC1,DISP=SHR
//*
```

*Figure 10 (Part 3 of 4). Example of a dynamic reconfiguration generation (MVS)*

```
//*********************************************************************
//*        IF ERROR OCCURRED IN VALIDATION, PRINT REPORT DATA SET
//*********************************************************************
//*
//ERRV   EXEC  PGM=IEBGENER,COND=(1,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.SYSPRINT,VOL=REF=*.NDF.SYSPRINT,
//             UNIT=SYSDA,DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//*********************************************************************
//*        IF ERROR OCCURRED IN TABLE 1 ASSEMBLY, PRINT TABLE 1
//*********************************************************************
//*
//ERR1   EXEC  PGM=IEBGENER,COND=(10,NE,NDF)
//SYSPRINT DD  DUMMY
//SYSUT1   DD  DSN=*.NDF.TBL1LIST,VOL=REF=*.NDF.TBL1LIST,
//             UNIT=SYSDA,DISP=(OLD,PASS)
//SYSUT2   DD  SYSOUT=A
//SYSIN    DD  DUMMY
//*
//
$$
```

*Figure 10 (Part 4 of 4). Example of a dynamic reconfiguration generation (MVS)*

# Chapter 3. Loading the Program under MVS

The last step in producing an operating NCP is to load the load module into the communication controller where it will reside. You can load your NCP into a channel-attached communication controller in two ways. You can use the loader utility provided by SSP, or you can use a loader facility provided by an access method. This chapter tells you how to use the SSP loader utility. For information on how to use the access-method loader facility to load both channel-attached and link-attached communication controllers, see the *VTAM Network Implementation Guide* or the *TCAM Installation, Resource Definition, and Customization Guide*.

The SSP loader utility is run as an MVS job or job step. A communication controller module disables all channel adapters except the one over which the load operation takes place. When NCP completes its initialization phase, it enables any additional channel adapters specified as ACTIVE in the NCPCA keyword. EP enables any additional channel adapters with the keywords HICHAN and LOCHAN coded in a PEP load module.

You must manually disable any channel adapter connected to a nonoperational host before starting the load process. Messages sent to the message data set indicate syntax errors or permanent I/O errors occurring during loading.

You can load the NCP load module from the host and save it on the MOSS disk if you are loading your NCP into the IBM 3720 Communication Controller using SSP V3R2 or a later release or into the IBM 3745 Communication Controller using SSP V3R3 or a later release. You can then later reload the NCP load module from the MOSS disk.

If you are loading your NCP into the IBM 3705 Communications Controller, you can load an optional diagnostic routine called the *initial test routine* before the loader utility loads NCP into the communication controller. If the initial test routine detects no malfunctions, the loader utility loads the NCP into the communication controller. However, if the routine detects trouble, it stops, and the loader utility issues an error message (IFL004I). The loader utility then loads any remaining communication controllers specified in the loader job. Loading and running the initial test routine are optional but recommended steps because this routine can detect conditions that can later cause NCP failure. The initial test routine is run unless you specify its omission in the LOAD control statement.

**Note:** If the communication controller's power has been turned off since the last time you ran the initial test routine, run the routine again before reloading the communication controller. This ensures that correct parity is set in the communication controller's storage.

## Loader Utility

This section discusses the following about the SSP loader utility in an MVS environment:

- Host processor and communication controller requirements
- Input to the loader utility
- Output from the loader utility.

## Host Processor and Communication Controller Requirements

You can run the loader utility's communication controller module in any channel-attached communication controller. The loader utility requires a minimum virtual region for MVS operation, and you do not need work data sets to run it. Before you can load the loader utility, you must ensure the communication controller:

- Has its power on
- Is identified to the MVS system where you plan to run the loader utility
- Is free so you can allocate it to the loader job step
- Is not in a program-stop condition
- Has the channel online that attaches it to the operating system
- Has enabled the channel adapter where the load is to occur.

**Note:** After you start the loader utility, do not cancel the load job.

The loader utility consists of the load modules IFLOADRN, IFLLD1P1, IFLLD1P2, IFLLD2P1, and IFLLD2P2. For SSP V3R2 and later releases, it also contains the load module IFWLEVEL. You must ensure these modules are in the SYS1.LINKLIB data set or in a partitioned data set pointed to by a STEPLIB or JOBLIB statement.

## Input to the Loader Utility

The input to the loader utility consists of two data sets. One is a DASD partitioned data set, an input data set that contains the NCP load module to be loaded into the communication controller. The other contains a LOAD statement specifying the NCP load module to be loaded from the host or the MOSS disk and the communication controller where it will be loaded.

**Note:** If you move the load module to another data set or system prior to loading, you must ensure that the load module retains its original characteristics (for example, block size).

If you are loading your NCP into the IBM 3705 Communications Controller, you can include an optional data set as input to the loader utility. This additional, partitioned data set contains the initial test routine and consists of modules IFL3705A, IFL3705B, IFL3705D, and IFL3705E. If you do not want to run the initial test routine, you can omit this data set.

## Output from the Loader Utility

The loader utility produces one output data set, the message data set SYSPRINT, which contains completion or error messages produced by the loader utility. See *NCP, SSP, and EP Messages and Codes* for a description of the messages issued by the loader utility.

# Controlling the Loader Utility

This section discusses examples of the job control statements and the utility control statement that you supply to the loader utility.

## Job Control Statements

The job control statements you need for calling the loader utility are shown in Table 6.

*Table 6. Job control statements for loader utility (MVS)*

| | |
|---|---|
| //jobname JOB | Starts the job. |
| // EXEC | Specifies the program name, IFLOADRN, or the name of a procedure containing the job control statements. |
| //STEPLIB DD | Specifies the data set containing the loader utility. |
| //SYSPRINT DD | Specifies a sequential data set. This data set can be sent to the SYSOUT device, magnetic tape volume, or direct-access volume. |
| //SYSUT1 DD | Specifies the DASD input data set containing the NCP load module. |
| //SYSUT3 DD | Only for the IBM 3705 Communications Controller. Specifies the DASD input data set containing the initial test routine load module; it is not required if you specify DIAG=NO in the LOAD statement. |
| //ccname DD | Specifies the unit address of the communication controller to be loaded. |
| //SYSIN DD | Specifies the data set (input stream) containing the LOAD control statement. |
| /* | |

## Utility Control Statement

The loader utility requires only one utility control statement, the LOAD statement. It specifies:

- The member of the input data set that contains the NCP load module
- The name of the load module to be loaded from the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller
- The communication controller to be loaded
- Whether or not you want to run the initial test routine, if loading NCP into an IBM 3705 Communications Controller
- Whether or not you want to save the load module on the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller
- Whether or not you want to request automatic initial program load (IPL), using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller.

The following conventions are used to describe the LOAD statement:

- Capital letters represent values you code directly without change.
- Lowercase letters represent parameters for which you must supply a value.
- Braces { } indicate you must choose from the enclosed items.
- *Or* signs | indicate you can choose between various keywords.
- An underlined value represents the default value of the keyword (that is, the loader utility uses that value if you omit the keyword).
- Brackets [ ] enclose keywords or symbols that are either optional or conditional.

The format of the LOAD statement is:

```
LOAD        LOADMOD={member name}
                    {(D)name    }
              [,UNIT=ccname|3725=ccname|3705=ccname]
              [,AUTOIPL={NO }]
                        {YES}
              [,DIAG={Y6}]
                     {Y8}
                     {NO}
              [,SAVE={NO }]
                     {YES}
```

**LOADMOD**={*member name*}
{**(D)***name* }

Identifies the load module.

*member name*
Specifies which member of the input data set indicated by SYSUT1 contains the desired NCP load module. The member must be in standard MVS load module form, without the overlay (OVLY) or scatter (SCTR) parameters.

**(D)***name*
Specifies the name of the load module to be loaded from a MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller.

**UNIT**=*ccname*|**3725**=*ccname*|**3705**=*ccname*

Specifies the ccname given to the data definition (DD) statement identifying the communication controller to be loaded. Use Table 7 on page 53 to determine which keyword to code.

*Table 7. Keywords for the UNIT control statement (MVS)*

| Communication Controller | SSP Release | Keyword |
|---|---|---|
| 3745 | SSP V3R3 and later | UNIT=ccname |
| 3720 | SSP V3R2 and later | UNIT=ccname |
| 3725 | SSP V3R2 and later | UNIT=ccname or 3725=ccname |
| 3705 | SSP V3R2 and later | UNIT=ccname or 3705=ccname |
| 3725 | SSP V3R1 | 3725=ccname |
| 3705 | SSP V3R1 | 3705=ccname |

**[,AUTOIPL={NO }]**
**{YES}**

Specifies whether or not you want to request automatic IPL from the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller. If you specify AUTOIPL=YES, automatic dump is also assumed. When an abend occurs, the dump in communication controller storage is automatically stored on the MOSS disk and an automatic IPL is initiated from the MOSS disk.

**[,DIAG={Y6}]**
**{Y8}**
**{NO}**

For the IBM 3705 Communications Controller, specifies whether or not the loader utility is to load the initial test routine.

**Y6**
Is the default and specifies that the routine be loaded into a communication controller without extended addressing. An IBM 3705-II Communications Controller having 64KB or less of storage uses 16-bit storage addresses and, therefore, does not require extended addressing.

**Y8**
Specifies that the routine be loaded into a communication controller with extended addressing. An IBM 3705-II Communications Controller having more than 64KB of storage requires extended addressing.

**NO**
Specifies that the initial test routine not be loaded.

**[,SAVE={NO }]**
**{YES}**

Specifies whether or not you want to save the load module from communication controller storage on the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R3 or a later release for the IBM 3745 Communication Controller. Specifying SAVE=YES is not valid with LOADMOD={(D)name}.

# Examples of Job and Utility Control Statements

The following are examples of statements that load NCP into different communication controllers. Since these are only examples, you must modify them to fit your particular system. To load into the IBM 3720 Communication Controller, you must use SSP V3R2 or a later release; to load into the IBM 3745 Communication Controller, you must use SSP V3R3 or a later release.

## Example 1. Loading into the IBM 3720 or 3745 Communication Controller with Disk Support

Assume you want to load an NCP load module named NCP1, residing in a data set named ONENCP, into an IBM 3720 or 3745 Communication Controller with a unit address of 030. You also want to save the load module from communication controller storage onto the MOSS disk, and you want automatic IPL from the MOSS disk. To load NCP1, use the following job and utility statements:

```
//CCLOAD  JOB  123456,SMITH,MSGLEVEL=1
//       EXEC  PGM=IFLOADRN
//STEPLIB  DD  DSN=SSP.SYS1LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=ONENCP,UNIT=3380,
//             VOL=SER=111111,DISP=SHR
//CC030    DD  UNIT=030
//SYSIN    DD  *
         LOAD  LOADMOD=NCP1,UNIT=CC030,SAVE=YES,AUTOIPL=YES
/*
```

When you want to load the saved NCP load module from the MOSS disk, issue the following load statement:

```
    LOAD  LOADMOD=(D)NCP1,UNIT=CC030
```

Because you did not specify AUTOIPL=YES or AUTOIPL=NO, the default setting was taken and the value of AUTOIPL was reset to NO.

## Example 2. Loading into the IBM 3720, 3725, or 3745 Communication Controller

Assume you want to load an NCP load module named NCP1, residing in a data set named ONENCP, into an IBM 3720, 3725, or 3745 Communication Controller with a unit address of 030. To load NCP1, use the following job and utility statements:

```
//CCLOAD  JOB  123456,SMITH,MSGLEVEL=1
//       EXEC  PGM=IFLOADRN
//STEPLIB  DD  DSN=SSP.SYS1LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=ONENCP,UNIT=3380,
//             VOL=SER=111111,DISP=SHR
//CC030    DD  UNIT=030
//SYSIN    DD  *
         LOAD  LOADMOD=NCP1,UNIT=CC030
/*
```

The LOAD statement in this example works only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3725 Communication Controller using SSP V3R1, you must specify the LOAD statement this way:

```
LOAD  LOADMOD=NCP1,3725=CC030
```

## Example 3. Loading into the IBM 3705 Communications Controller

Assume you want to load an NCP load module named NCP1, residing in a data set named ONENCP, into an IBM 3705 Communications Controller. This communication controller has a unit address of 030 and does not have extended addressing. To load NCP1, use the following job and utility statements:

```
//CCLOAD  JOB  123456,SMITH,MSGLEVEL=1
//       EXEC  PGM=IFLOADRN
//STEPLIB  DD  DSN=SSP.SYS1LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=ONENCP,UNIT=3380,
//             VOL=SER=111111,DISP=SHR
//SYSUT3   DD  DSN=SYS1.LINKLIB,DISP=SHR
//CC030    DD  UNIT=030
//SYSIN    DD  *
         LOAD  LOADMOD=NCP1,UNIT=CC030,DIAG=Y8
/*
```

The LOAD statement in this example works only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3705 Communications Controller using SSP V3R1, you must specify the LOAD statement this way:

```
LOAD  LOADMOD=NCP1,3705=CC030,DIAG=Y8
```

In this example, you should load and run the initial test routine before you load the NCP load module. If you do not want to load and run the initial test routine, you would include DIAG=NO on the LOAD statement and omit the SYSUT3 DD statement.

## Example 4. Loading More Than One NCP into IBM 3720, 3725, or 3745 Communication Controller

You can load more than one NCP into more than one IBM 3720, 3725, or 3745 Communication Controller at the same time by including a separate DD statement and LOAD statement for each communication controller. For example, assume you want to load an NCP load module named NCP2 into an additional communication controller with a unit address of 040. Both NCP1 and NCP2 reside in a data set named TWONCPS. To load NCP1 and NCP2, use the following job and utility statements:

```
//CCLOAD  JOB  123456,SMITH,MSGLEVEL=1
//       EXEC  PGM=IFLOADRN
//STEPLIB  DD  DSN=SSP.SYS1LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=TWONCPS,UNIT=3380,
//             VOL=SER=111111,DISP=SHR
//CC030    DD  UNIT=030
//CC040    DD  UNIT=040
//SYSIN    DD  *
     LOAD  LOADMOD=NCP1,UNIT=CC030
     LOAD  LOADMOD=NCP2,UNIT=CC040
/*
```

The LOAD statements in this example work only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3725 Communication Controller using SSP V3R1, you must specify the LOAD statements this way:

```
LOAD  LOADMOD=NCP1,3725=CC030
LOAD  LOADMOD=NCP2,3725=CC040
```

## Example 5. Loading More Than One NCP into More Than One IBM 3705 Communications Controller

You can load more than one NCP into more than one IBM 3705 Communications Controller at the same time by including a separate DD statement and LOAD statement for each communication controller. For example, assume you want to load an additional NCP load module named NCP2 into a communication controller with a unit address of 040. Both NCP1 and NCP2 reside in a data set named TWONCPS. To load NCP1 and NCP2, use the following job and utility statements:

```
//CCLOAD  JOB  123456,SMITH,MSGLEVEL=1
//       EXEC  PGM=IFLOADRN
//STEPLIB  DD  DSN=SSP.SYS1LIB,DISP=SHR
//SYSPRINT DD  SYSOUT=A
//SYSUT1   DD  DSNAME=TWONCPS,UNIT=3380,
//             VOL=SER=111111,DISP=SHR
//CC030    DD  UNIT=030
//CC040    DD  UNIT=040
//SYSIN    DD  *
         LOAD  LOADMOD=NCP1,UNIT=CC030,DIAG=NO
         LOAD  LOADMOD=NCP2,UNIT=CC040,DIAG=NO
/*
```

The LOAD statements in this example work only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3705 Communications Controller using SSP V3R1, you must specify the LOAD statements this way:

```
LOAD  LOADMOD=NCP1,3705=CC030,DIAG=NO
LOAD  LOADMOD=NCP2,3705=CC040,DIAG=NO
```

This example does not include loading and running of the initial test routine.

# Generating and Loading a Remote Communication Controller with a Floppy Disk

This section describes how to create a floppy disk and load it into a remote communication controller. It includes information on how to prepare a load module, download the module to a floppy disk, and upload the module from a floppy disk to a MOSS disk. For more information, see *3745 Models 130/150/170 Advanced Operations* and *3720/3721 Operator's Guide*.

## Load Module Preparation

You must first generate a load module. It must be able to fit on a single floppy disk and therefore cannot exceed 1MB. After you create the load module, use the VTAM MODIFY command to transfer the module from the VTAM host to a local communication controller.

## Downloading a Load Module to a Floppy Disk

Use the Display IPL Information (DII) option to download the load module from the local communication controller to a floppy disk. From the DII option, select FLOPPY MANAGEMENT. Then, use the PF key for COPY LM TO FLOPPY. You will be prompted for the CCU (only if two CCUs are available) and the name of the load module to be downloaded.

**Note:** You must format the disk before downloading the load module. A MOSS function is provided for this purpose.

## Uploading a Load Module from a Floppy Disk to a MOSS Disk

You must perform the uploading task on a remote communication controller.

Use the DII option to upload the load module from a floppy disk to a MOSS disk on a remote communication controller. From the DII option, select FLOPPY MANAGEMENT. You will be prompted for the CCU (only if two CCUs are available) and the name of the load module to be uploaded. If a load module with the same name exists on the MOSS disk, it is replaced. If a load module with the same name does not exist and space is available on the MOSS disk, the module from the floppy disk is added to the MOSS disk. If no space is available on the MOSS disk, the load module from the floppy disk replaces the oldest load module on the MOSS disk.

You must then perform an IPL of the load module from the MOSS console.

# Part 2. Generating and Loading under VM

# Chapter 4. Generating the Program under VM

After you install your Network Control Program (NCP) and System Support Programs (SSP) product from the tape and after you define the NCP configuration, the next step in producing an operating NCP is to generate the program.

This chapter contains information about generating NCP under the VM operating system. It discusses the following topics:

- Understanding the generation procedure
- Controlling the generation procedure
- Performing different types of NCP generations
- Understanding listings and error messages.

SSP Version 3 includes the NCP/EP definition facility (NDF), a program used in generating an NCP, partitioned emulation program (PEP), or Emulation Program (EP) load module. NDF can be used to perform the following tasks:

- FASTRUN validation of an NCP, PEP, or EP generation definition
- Generation of an NCP, PEP, or EP load module
- Generation of an NCP or PEP V4 Subset load module
- Generation of an NCP or PEP load module with IBM special products or user-written code
- Generation of a text file for dynamic reconfiguration
- Migration of an existing generation definition to a different version and release of a different controller.

SSP V3R2 and later releases contain the NDF standard attachment facility, which allows user-written generation applications to interface with NDF during an NCP generation. The NDF standard attachment facility helps you define resources for user-written code. Use the NDF standard attachment facility to generate user-written code with NCP V4 Subset, NCP V4R2 and later releases, or NCP V5R2 and later releases. For more information, see "Running an NCP or PEP Generation with IBM Special Products or User-Written Code" on page 71.

This chapter discusses how to control the generation procedure, and Chapter 5 supplies examples of EXECs for generation.

## Understanding the Generation Procedure

Generating an NCP with NDF under the VM operating system is a two-step process. Figure 11 on page 62 shows the input NDF accepts and the output it produces under the VM operating system.

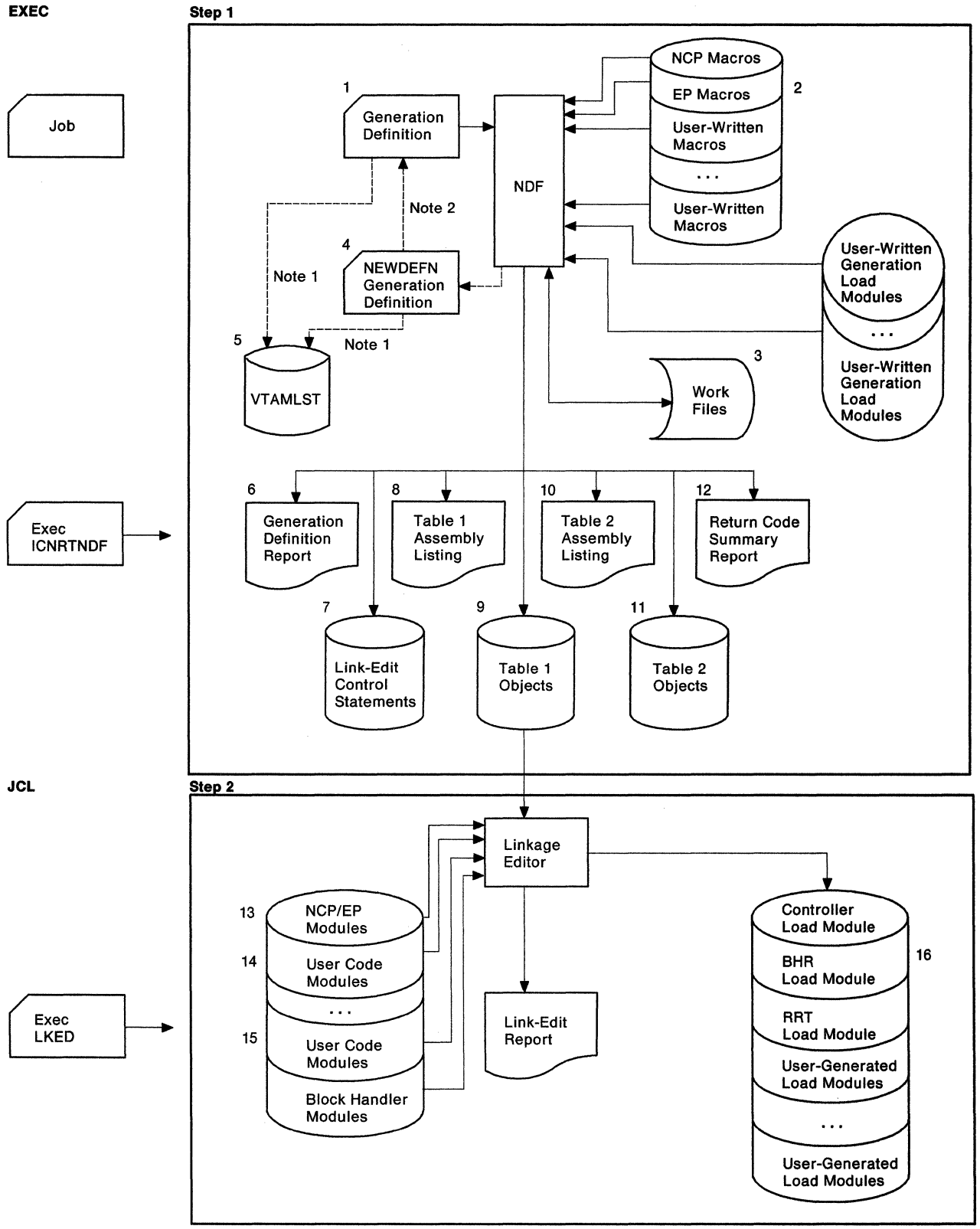



*Figure 11. The generation procedure under VM. The numbers in this figure correspond to the files described in "Specifying Files Used by NDF" on page 65. For an explanation of Note 1 and Note 2, see the following text.*

**Notes on Figure:**

1. Input to VTAMLST, from either the NCP definition statements or the NEWDEFN file, is required. Refer to the description of NEWDEFN in Table 8 on page 65.
2. You can use definitions from NEWDEFN as input to the generation definition. For more information, refer to *NCP, SSP, and EP Resource Definition Reference* (the REUSE suboperand of the NEWDEFN keyword).

## Generation Steps

**Step 1:** The first generation step, the NDF step, consists of four phases.

**Phase 1:** In the first phase, the generation validation phase, NDF does the following:

- Reads your GENDECK file containing your generation definition
- Validates the definition statements and keywords coded in the generation definition
- Creates the NEWDEFN file when you code the NEWDEFN keyword on the OPTIONS definition statement
- Generates assembler language source code for the resources coded in the generation definition
- Creates link-edit control statements; these statements will later link control-block object code with preassembled NCP code objects to generate an NCP load module.

If you are using the NDF standard attachment facility to generate resources using IBM special products or user-written code, the first phase has two additional steps. During the validation phase, NDF does the following:

- Dynamically loads one or more user-written generation load modules
- Calls routines in the user-written generation load modules to perform generation processing and allows the routines to call NDF internal routines.

**Phases 2 and 3:** The second and third phases are the table 1 and table 2 assemblies. Each assembly reads the source code specification created in the generation validation phase and generates object code for the control blocks.

**Phase 4:** In the fourth phase, the return code summary, NDF does the following:

- Generates a composite return code that shows the success or failure of each phase
- Creates a compact listing that gives return codes for the generation validation and table assembly phases.

**Step 2:** The second generation step is the link-edit. During this step, the control-block object code produced in the first step is linked with preassembled object modules to generate a load module. If you included IBM special products or user-written code or block handlers in the generation definition, the appropriate object module libraries must be available during the link-edit. You should ignore a zero-length control section (CSECT) indication in the NCP link-edit.

With SSP V3R4 and later releases, when no errors occur, the link-edit return code is 0. When the NDF standard attachment facility generates a load module separate from the NCP load module, the return code is 4 if the load module is generated from

table 1, and 0 if from table 2. You should investigate all return codes of 4 to determine if they are informational or indicate an error that must be resolved.

You cannot use LA, BAL, or BLG instructions to reference labels that reside above the 4MB (MB equals 1 048 576 bytes) boundary. For NCP V5R4, load modules can be larger than 4MB but not larger than 8MB. To determine how much storage is available for NCP buffers in your communication controller, perform the following calculation:

1. Locate the $BUFPOOL value in the link-edit portion of your generation listing ($BUFPOOL marks the end of the load module) and add it to the value from the ICN076I informational message issued in the GENEND definition statement. Both values are hexadecimal.
2. Subtract the value obtained in number 1 from the amount of storage available in your NCP.
3. From the value obtained in number 2, subtract the amount of storage allocated for the maintenance and operator subsystem (MOSS) Mailbox/TSS Workspace. You can find this amount in the control data set (CDS) control block at offset 46(2E); it is also entered as a number of 4KB (KB equals 1024 bytes) pages when the operator initializes NCP.
4. The number remaining from this subtraction is the amount of storage available for buffers.
5. To determine the number of buffers, add 12(C) to the value coded for BFRS on the BUILD definition statement; divide this number into the amount of storage available for buffers (obtained in number 4).

**Note:** If you want to run a FASTRUN generation to validate your generation definition without creating control blocks or if you want to do a generation for dynamic reconfiguration, do not specify the link-edit step to be run in your EXEC.

## Virtual Storage Requirements

NDF uses a storage manager to organize its work space during NDF generation validation. Whenever possible, storage manager data is kept in virtual storage. However, if data overflows the virtual storage region available to the storage manager, this extra data is written into a storage manager work file. Generally, you should be able to allocate enough virtual storage to hold all the work data. For very large generations, however, you may be required to define a work file.

All NDF generations require about 200KB of work space and an additional 48 bytes for each resource specified in the generation definition. The storage manager usually requires less than 500KB of virtual storage.

If you need additional work space or if you are using the NDF standard attachment facility, you need to define a storage manager work file (DBWORKFL) in your EXEC. You should ensure that this space allocation is not excessive because the time required to initialize a large work file adds a significant amount of running time to generation validation. For information about how to specify a storage manager work file, see "Specifying Files Used by NDF" on page 65.

Generally, one cylinder of disk space allocated for a work file should be adequate.

If you are generating NCP/Token-Ring interconnection resources, you need an additional 80 bytes for each NCP/Token-Ring interconnection physical line specified in

the generation definition and an additional 160 bytes for each NCP/Token-Ring interconnection logical line generated using the AUTOGEN facility.

## Performance Considerations

NDF requires approximately 3.5MB of virtual storage to achieve optimal performance, although as much as 8MB of virtual storage may be required to complete generation. If the available virtual storage drops below 3.5MB, paging during the generation validation phase significantly degrades performance.

In addition, a block size of at least 3630 bytes for the table 1 listing file is recommended. Using even larger block sizes can noticeably improve performance. If you define an inadequate block size, the time required for the input or output operations to the file can significantly affect elapsed time for NDF execution.

# Controlling the Generation Procedure

You control the generation procedure through the EXEC that you code and through the definition statements and keywords that you coded in the generation definition.

This section explains the different types of generations you can run. It discusses some of the definition statements and keywords you can code in your generation definition. It also discusses the parameters you need to code and the files you need to specify in your EXEC. For examples of EXECs for generating, see Chapter 5.

## Specifying Files Used by NDF

This section contains the FILEDEFs of the files used by NDF. You specify the FILEDEFs in the files in your EXEC. Table 8 lists the FILEDEFs and describes the files they specify.

**Note:** The numbers following the FILEDEFs correspond to the files shown in Figure 11 on page 62.

*Table 8 (Page 1 of 3). FILEDEFs of files used by NDF (VM)*

| FILEDEFs | Description |
|---|---|
| **GENDECK (1)** | Specifies the file containing the definition statements for the NCP network definition. This file must contain 80-byte fixed-format records. |
| **SYSLIB (2)** | Specifies the chain of definition statement libraries. The libraries included depend on the particular NDF generation. You need the IBM 3705, 3720, 3725, or 3745 NCP definition statements for the table assemblies. You may also need additional libraries for both the generation validation phase and the table assemblies if user-written code is in NCP. |
| **DBWORKFL (3)** | Specifies the storage manager work file. This temporary file stores internal data in 4KB records during NDF generation validation. The BDAM accesses the records in the file. Use this file if NDF cannot obtain enough virtual storage to hold all of the temporary data for the generation validation phase or, if you are using the NDF standard attachment facility, to generate user-written code. Ensure that this space allocation is not excessive because the time required to initialize a large work file adds a significant amount of running time to generation validation. |

*Table 8 (Page 2 of 3). FILEDEFs of files used by NDF (VM)*

| FILEDEFs | Description |
|---|---|
| **TBL1SRCE (3)** | Specifies the file that contains the table 1 assembly source code. This file contains the output from generation validation and serves as the input file for the table 1 assembly. |
| **TBL2SRCE (3)** | Specifies the file that contains the table 2 assembly source code. This file contains output from the generation validation phase and serves as input for the table 2 assembly. |
| **SYSUT1 (3)** | Specifies the assembler work file. This work file is used to temporarily store internal NDF data for the assembly of table 1 and table 2 objects. |
| **NEWDEFN (4)** | Specifies the output file containing the new generation definition created by NDF. For more information, refer to *NCP, SSP, and EP Resource Definition Guide*.<br><br>The new generation definition consists of the input from the definitions from the NCP generation definition plus statements and keywords added during the generation process.<br><br>**Notes:**<br><br>1. If you specified NEWDEFN=YES on the OPTIONS definition statement in your generation definition, you must define the NEWDEFN file in the EXEC for your generation.<br>2. If VTAM users generate a NEWDEFN file, they must include a NEWDEFN file in the VTAMLST that VTAM accesses during the activation of this NCP. If they do not generate a NEWDEFN file, they must include their generation definition (GENDECK) in VTAMLST.<br>3. All VTAM users must include the NEWDEFN source in the VTAMLST that VTAM accesses during the activation of NCP. |
| **VTAMLST (5)** | For information about VTAMLST, refer to *VTAM Network Implementation Guide*. |
| **SYSPRINT (6)** | Specifies the file that contains the generation validation listing. |
| **LNKSTMT (7)** | Specifies the link-edit statement file. This file contains the link-edit control statements produced by NDF and used to build the NCP load module. |
| **SYSLIN (7)** | Specifies the input file that contains the link-edit control statements passed to the linkage editor from NDF. This is the same file as LNKSTMT from phase 1 of the NDF job. |
| **TBL1LIST (8)** | Specifies the file for the table 1 assembly listing. This file can be large, and its data control-block parameters defined for it can have a significant impact on NDF performance. |
| **TBL1OBJ (9)** | Specifies the table 1 object file. This file must be a member of a partitioned data set that is passed to the link-edit step of an NDF generation. The member name for this file is ICNTABL1. |
| **TBL2LIST (10)** | Specifies the table 2 assembly listing file. |
| **TBL2OBJ (11)** | Specifies the output file for the control-block objects generated by the table 2 assembly. This file must be a member of the same partitioned data set as TBL1OBJ. The member name for this file is ICNTABL2. |

*Table 8 (Page 3 of 3). FILEDEFs of files used by NDF (VM)*

| FILEDEFs | Description |
|---|---|
| **SYSPUNCH (9, 11)** | Specifies the library where the table assemblies place the control-block objects. This contains the TBL1OBJ and TBL2OBJ file members created in phase 1 of the NDF job. |
| **PRINTER (12)** | Specifies the file for the return code summary report. |
| **xxxxxxx (13)** | Specifies the library that contains the preassembled NCP object modules. The FILEDEF and the specific library vary with the target of the generation. Specify ANCPMOD1 if you are generating NCP V4R3, V4R3.1, or V5R2 or a later release. Specify OBJ3705 if you are generating NCP V3 for the IBM 3705 Communications Controller. Specify OBJ3725 if you are generating any other versions of NCP, regardless of which communication controller you plan to load. |
| **_____ (14)** | Specifies a library with a FILEDEF determined by IBM special products or user-written code. This library contains preassembled object code for user-written code modules. |
| **ULIB (15)** | Specifies a library that contains block-handler object modules or preassembled user-written code modules. |
| **SYSLMOD (16)** | Specifies the library where the communication controller, block-handler set resolution table (BHR), and resource resolution table (RRT) load modules will be placed. For SSP V3R4 and later releases, SYSLMOD also specifies the library where user-generated load modules will be placed. Do not code BLKSIZE in the generation EXEC or when preallocating data sets. |

The following three files are defined only if you code ASSEMBLY=YES when invoking NDF; this causes the NDF controller assembler to assemble control block code outside the regular NDF process.

| | |
|---|---|
| **ASMSRCE** | Specifies the assembly source code used as input to the NDF controller assembler when the assembly option is specified. The file must contain 80-byte fixed-format records. |
| **ASMLIST** | Specifies the file for the ASMSRCE assembly listing. |
| **ASMOBJ** | Specifies the file for the ASMSRCE object file. |

## Specifying Parameters for NDF

This section describes the optional NDF parameters that you can specify in your EXEC. When specifying more than one parameter in the parameter field, you must separate the parameters with one or more spaces. The right parenthesis closing the parameter list is not required.

### LINECNT Parameter

Use the LINECNT parameter to specify the number of lines on each page of the generation validation listing and the table assembly listing. The valid range for this parameter is 10 to 99. The default value for the validation listing and for the assembly listing is 60. If you specify a value for LINECNT, this value is used in all listings.

The following is an example of LINECNT in the EXEC:

```
ICNRTNDF (LINECNT(40))
```

## FASTRUN Parameter

For SSP V3R2 and later releases, you can use the FASTRUN parameter to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

Using the FASTRUN parameter is the same as coding FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition.

The following is an example of FASTRUN in the EXEC:

```
ICNRTNDF (FASTRUN(ON))
```

## ASSEMBLY Parameter

For SSP V3R2 and later releases, you can use the ASSEMBLY parameter to invoke the NDF controller assembler to assemble table source code. A complete NCP generation does not need nor require the ASSEMBLY parameter.

The input and output file names used by NDF when you specify ASSEMBLY are different from those used for the table assembly, except for the assembler work file (SYSUT1). See "Specifying Files Used by NDF" on page 65 for the names and descriptions of these files.

**Note:** You cannot specify both the FASTRUN parameter and the ASSEMBLY parameter for the same step.

The following is an example of ASSEMBLY in the EXEC:

```
ICNRTNDF (ASSEMBLY(YES))
```

## ASSMLIST Parameter

For SSP V3R5 and later releases, you can use the ASSMLIST parameter to generate the table assembly listing. Valid values for ASSMLIST are YES and NO. The default is ASSMLIST=YES. When ASSMLIST=NO, the table assembly listing is suppressed.

**Note:** You cannot specify both the FASTRUN parameter and the ASSMLIST parameter for the same job step.

The following is an example of ASSMLIST in the EXEC:

```
ICNRTNDF (ASSMLIST(YES))
```

## Migration Aid Parameters

For SSP V3R5 and later releases, you can use the migration aid parameters to invoke the migration aid. The migration aid is an NDF function that automates much of the NCP migration task.

The VM generation EXECs VMNCP, VMTAPE, and VMFAST have been modified for the migration aid. If you want to code its parameters in your VM generation EXEC, do not use previous versions of these EXECs. The updated versions are supplied with the migration aid. For more information on these parameters, refer to *NCP, SSP, and EP Resource Definition Reference* and *NCP Migration Guide for Version 5 Release 4*.

The following is an example of the migration aid parameters in the EXEC:

```
VMFAST FN=GEN_FN,FT=GEN_FT,V=V5R3,M=3745-410,...,TMODEL=3745-410,
       TUSGTIER=5,TVERSION=V5R4
```

## Naming Resources

Avoid using the prefixes shown in Table 9 and the labels shown in Table 10 when naming resources because they are used as control block identifiers and can cause duplicate labels that result in an error message from the assembler.

*Table 9. Prefixes to avoid (VM). Avoid names that are similar to control-block acronyms.*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| @ | BOQ | CPT | EPI | IX | LTV | NPB | QCB | SNP | UNA |
| $ | BPB | CRB | EQB | J* | LTX* | NPF | RAT | SOT | USC |
| AAB | BSB | CRP | ERB | LAA | LU | NQB | RCB | SPC | UXR* |
| ABN | BST | CTB | ERX | LAB | LX | NQE | RCQ | SST | U1 |
| ACB | BTT | CTP | FCT | LB* | L1B | NSQ | RCV | STE | VAT |
| ACT | BTU | CUB | FLB | LCB | L4B | NVT | RG | STQ | VIT |
| ACU | BUE | CY | FMT | LCC | MBF | NVX | RH* | SUT | VLB |
| AEB | CA* | CX | FVT | LCI | MBX | OLL | RN* | SVT | VR |
| ALE | CAB | DAE | GCB | LCP | MCT | OLT | RU* | SXB | VST |
| AST | CAI | DDB | GPT | LCS | MDR | PAB | R* | SYS | VTS |
| ATB | CAR | DIA | GRW | LCW | MIB | PAD | RMB | TCB | VVT |
| ATP | CAT | DPT | GVT | LDA* | MIC | PCB | ROSH* | TET | WCB |
| ATT | CB | DQB | HWE | LDI* | MIF | PIU | RST | TGB | WRP |
| AV* | CBB | DRS | HWX | LGT | MIH | PL* | RTR | TH* | WU |
| AXB | CDS | DRX | IB | LKB | MIM | PL2 | RVT | TIM | X |
| BC | CER | DSP | ICE | LKC | MLT | PMF | SCB | TND | XDA |
| BCU | CGP | DTG | ICI | LNB | MMV | PRB | SEB | TQB | XDB |
| BER | CHC | DVB | ICW | LNV | MSC | PSA | SGE | TRT | XDH |
| BGS | CHV | DVI | IDD | LPB | MTF | PSB | SGT | TVS | XID |
| BH | CIE | DVQ | IDE | LRB | NET | PSI | SHB | UAC | |
| BHD | CM | ECB | IDL | LRC | NIB | PSP | SID | UAD | |
| BHR | COE | ECD | IDB | LTC | NIX | PST | SIT | UIB | |
| BHS | CPI | ECL | IRN | LTR | NLB | PUV | SMB | UIC | |
| BLU | CPN | EML | IRQ | LTS | NLX | QAB | SMM | ULVSGN | |

**Note:** * Indicates that a number from 0 to 9 follows this prefix.

*Table 10. Labels to avoid (VM). Avoid names that are similar to control-block acronyms.*

| | | | | | |
|---|---|---|---|---|---|
| ACITRAP | CSPQH2 | NCPHIST1 | SVCQUT | THLOB | TMRF |
| CAACER | CSPQOFF | NCPLVL | SWQTMQ1 | THLOM | TTCUR |
| CACCER | CSPQON | NEWLNE | SWQTMQ2 | THMID | TTEND |
| CADCER | DCTABND | OLDLNE | TABEND | THMPF | TTRECNTR |
| CAECER | DCTSAVEK | PEPQSCNB | TABSTAR | THODAIB | TTSKPCNT |
| CAFCER | D*RCB | PEPQSCNM | THAFIB | THODAIM | TTSTAR |
| CCPH1 | EPLVL | PSCA | THAFIM | THONLY | UIHRCCW |
| CCPSAVE | FILLB | ROSSVADDR | THBCUVVT | THPSIB | USTAGETR |
| CHANSNS1 | FILLC | ROSSVCCR | THFID | THPSIM | UTILSTSZ |
| CHANSNS2 | HDRNENT | ROSSVCCU | THFIRST | THTYPO | |
| CHSVBKSV | ICNTABL1 | ROSSWK1 | THFOB | THTYP1 | |
| CHSVH1 | LCDBSCB | SECNTRI | THFOM | THTYP2 | |
| CSPQH1 | LCDSSBIT | SVCO | THLAST | THTYP3 | |

**Note:** * Indicates that a number from 0 to 9 can appear as this character.

## Defining Virtual Storage

You can control virtual storage size by using the CP DEFINE STORAGE command in your EXEC. The storage specified must include space for CMS and that space required by NDF. For most NDF runs, 4MB should be adequate, although very large generation definitions may require more than twice this much storage.

The following is an example of the CP DEFINE STORAGE command for 4MB of storage:

```
CP DEFINE STORAGE AS 4096K
```

For SSP V3R6, the NDF IHR Assembler supports 31-bit addressing, allowing you to use the addressable storage available above the 16MB line in a VM/XA* environment. To submit larger generation decks, you must increase the virtual storage size on the CP DEFINE STORAGE command.

## Naming Load Modules

Besides creating an NCP load module, NDF also produces a resource resolution table (RRT) load module and, if you have coded any block-handling routines, a block-handler set resolution table (BHR) load module. These load modules contain information that the access method requires.

Use the NEWNAME keyword on the BUILD definition statement to designate the names for the BHR, RRT, and NCP load modules. NDF appends a *B* to the NEWNAME value to name a BHR load module and appends an *R* to the NEWNAME value to name an RRT load module.

For information about the NEWNAME keyword on the BUILD definition statement, see *NCP, SSP, and EP Resource Definition Guide*. For information on how to code this keyword, see *NCP, SSP, and EP Resource Definition Reference*.

## Controlling Succeeding Generation Steps

NDF produces an overall return code for the NDF step. NDF prints this return code as part of the return code summary section of the NDF report and denotes the NDF phase where it encountered the error. NDF also passes this overall return code to the operating system as the NDF return code.

You can use this overall return code to determine whether to run succeeding steps. This technique is used in the sample EXEC for an NCP or PEP generation with output written to tape on page 100. The following list shows the overall return code values and meanings; notice that leading zeros are suppressed:

| Value | Meaning |
|---|---|
| 1 | Input validation error |
| 10 | Table 1 error |
| 100 | Table 2 error |
| 1000 | Printer file error |
| 10000 | Error freeing auxiliary directory (CMS). |

* Trademark of IBM Corporation—see "Special Notices" on page iii for a list of all IBM trademarks used in this book.

# Performing Different Types of NCP Generations

This section discusses the different types of NCP generations and how to run them.

## Running a FASTRUN Generation

Do a FASTRUN generation to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

To run a FASTRUN generation, code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for SSP Version 3. For SSP V3R2 and later releases, code FASTRUN=ON as a parameter in your EXEC when calling NDF. Ensure that your EXEC does not call the linkage editor; if the link-edit step is present, an error will result. Also, do not define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

For an example of the EXEC for a FASTRUN generation, see page 81.

## Running a Standard NCP or PEP Generation

To run a standard NCP or PEP generation, supply your generation definition as input and specify the various input and output files in your EXEC. You can specify that input and output files be written to disk or that certain files be written to tape.

If you detect an error while generating or running your NCP, you can write to tape certain listings files, such as the table 1 assembly listing, the table 2 assembly listing, and the link-edit listing.

If you are including certain types of resources in your generation definition (such as those listed under NEWDEFN (4) in Table 8 on page 65), you must code NEWDEFN=YES on the OPTIONS definition statement as the first executable statement in your generation definition and define the NEWDEFN file in your EXEC. For more information on coding the NEWDEFN keyword, refer to *NCP, SSP, and EP Resource Definition Reference*.

For examples of EXECs for running standard NCP or PEP generations, see "Example of an NCP or PEP Generation with Output Written to Disk" on page 86 and "Example of an NCP or PEP Generation with Output Written to Tape" on page 100.

## Running an NCP or PEP Generation with IBM Special Products or User-Written Code

If you included IBM special products or user-written code—such as Network Terminal Option (NTO), Network Routing Facility (NRF), or X.25 NCP Packet Switching Interface (NPSI)—in an NCP or PEP generation, you must modify the basic EXEC.

If you are using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing. You are not required to use this method.

If you choose to generate NCP and user-written code *without* using the NDF standard attachment facility or if you are generating user-written code using SSP V3R1, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of the link-edit statements by coding keywords on the GENEND definition statement.

## Using the NDF Standard Attachment Facility

You can run this type of generation *only* if you are using SSP V3R2 or a later release to generate NCP V4 Subset, NCP V4R2 or a later release, or NCP V5R2 or a later release. To use the NDF standard attachment facility, you must supply a user-written generation application. For information on writing user-written code and user-written generation applications, see *SSP Customization*. Figure 12 on page 73 shows how your user-written code and user-written generation load modules are included in the generation procedure.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation load modules to be loaded in the generation. Each application must have its own generation load module. You can name up to 25 generation load modules.
- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation load modules.
- Modify the EXEC for a standard NCP or PEP generation to include the FILEDEFs for the NEWDEFN file, the DBWORKFL file, and the libraries for user-supplied modules.

For an example of the EXEC for generating user-written code using the NDF standard attachment facility, see page 115.

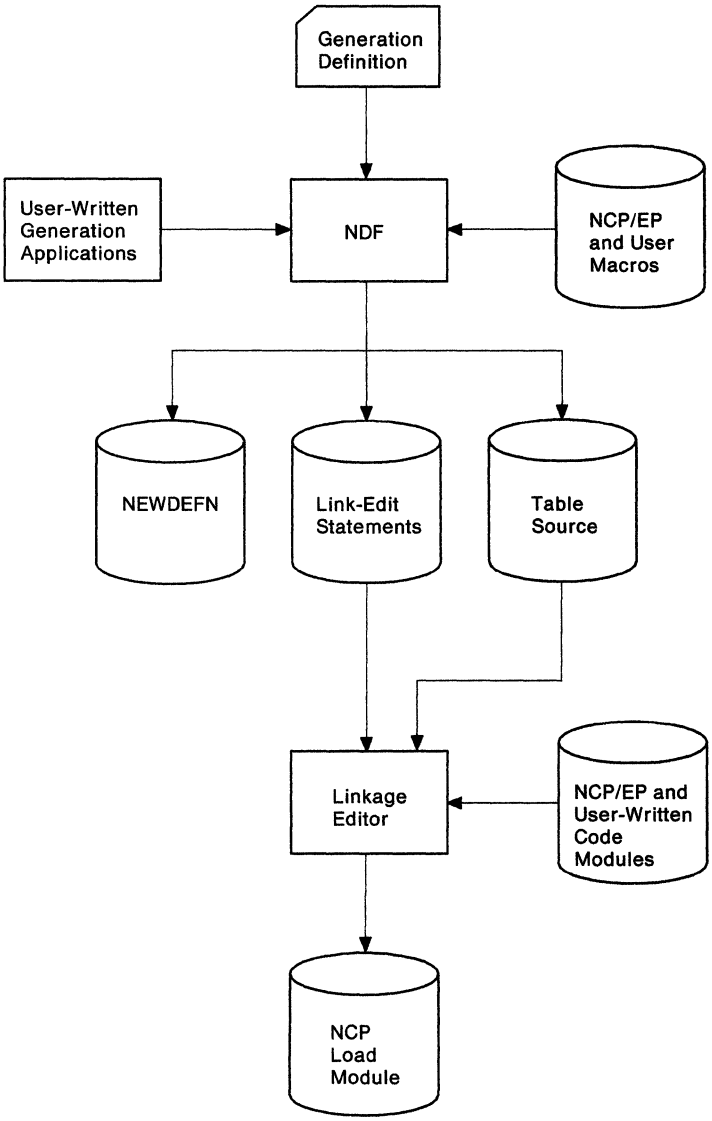



*Figure 12. Generating an NCP containing user-written code using the NDF standard attachment facility (VM). This figure shows how to include user-written generation load modules in an NCP or PEP generation.*

## Using the GENEND Definition Statement

Besides being able to generate NCP V4 Subset, NCP V4R2 and later releases, or NCP V5R2 and later releases with user-written code using the NDF standard attachment facility as described previously, you can also generate them as described here.

You can use the GENEND definition statement to generate any version of NCP using SSP Version 3. Before generating NCP, code the link-edit statements for the routines and identify the location of these link-edit statements by coding certain keywords on the GENEND definition statement.

Figure 13 on page 74 shows how to include your IBM special products or user-written code in the generation procedure.

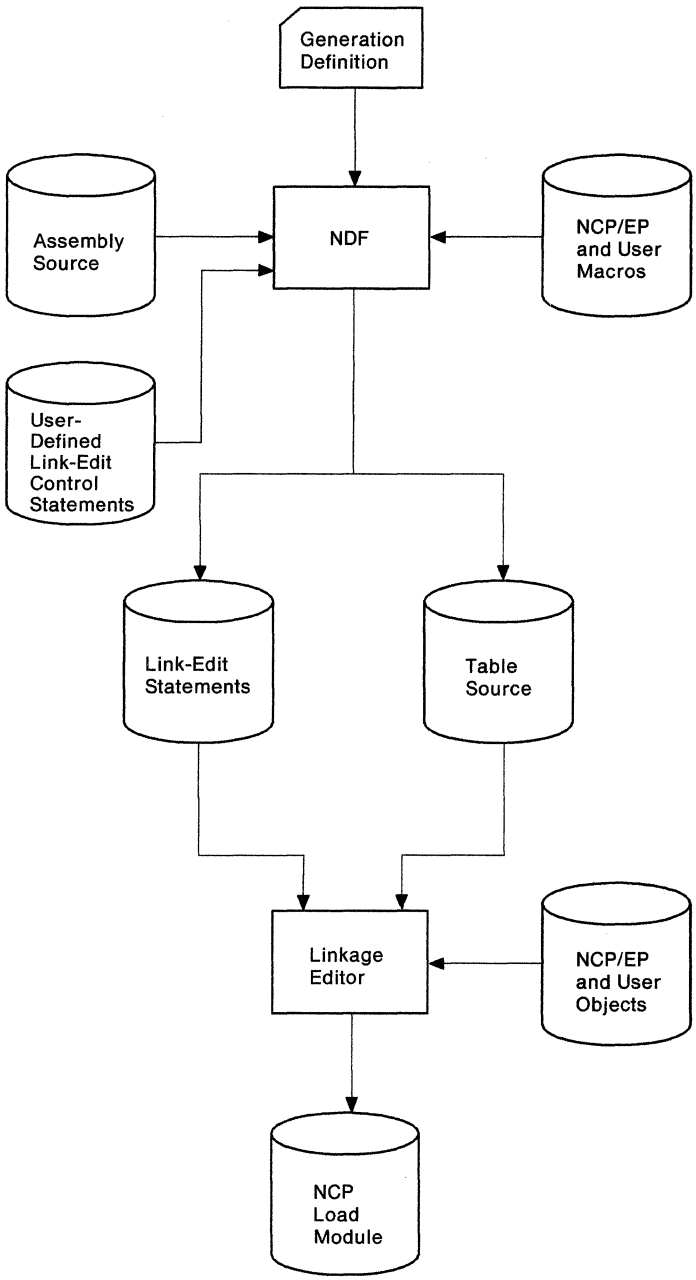



*Figure 13. Generating an NCP containing user-written code using the GENEND definition statement (VM). This figure shows how to include user-written code in an NCP or PEP generation.*

Before you generate IBM special products or user-written code using the GENEND definition statement, ensure that you:

- Assemble the user-written routines and code the link-edit control statements for the routines
- Code the appropriate keywords on the GENEND definition statement for your user-written routines
- Place the members with SRCLO or SRCHI code in a definition statement library (SYSLIB) available to NDF

- Place all members that contain INCLUDE or ORDER link-edit control statements in a definition statement library in the NDF SYSLIB chain
- Place all definition statements in the NDF SYSLIB chain
- Modify the EXEC to include the SYSLIB chain and the ULIB or user object code library FILEDEF statement.

The generation validation phase of NDF reads the link-edit control statements and writes them into the same file as the standard NCP link-edit control statements.

For an example of an EXEC for generating user-written code and the NCP using the GENEND definition statement, see page 116.

## Running a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, use the text file from a dynamic reconfiguration generation. Ensure that you coded the original NCP to allow dynamic reconfiguration. If you did, the dynamic reconfiguration generation produces a text file that the access method can use to modify the NCP.

**Note:** VTAM has its own dynamic reconfiguration procedures that do not require you use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Network Implementation Guide*.

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration file consisting of ADD or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration file is the input for the dynamic reconfiguration generation. This type of generation produces a text file that the access method uses to modify an NCP already running in a communication controller. For information on using ADD, DELETE, PU, and LU definition statements, see *NCP, SSP, and EP Resource Definition Guide.*

A dynamic reconfiguration generation requires one table assembly and no link-edit. For an example of an EXEC for a dynamic reconfiguration generation, see page 117.

# Understanding Listings and Error Messages

During generation validation, NDF creates a report that contains:

- The input statements interspersed with informational and error messages
- The keywords and statements passed to NDF from user-written generation load modules using the NDF standard attachment facility (SSP V3R2 and later releases)
- A resource name and network address cross-reference (only if the generation validation run is valid)
- An error message summary
- A return code for the generation validation phase.

Also, NDF creates a listing during the return code summary phase that gives return codes for the generation definition and for each of the table assemblies.

In NCP V4R3 and later releases or NCP Version 5, generation definition listings produced by SSP V3R4.1 and later releases include a message indicating how much storage NCP needs for initialization in excess of the storage that the load module

displaces. For information on calculating buffer storage, see "Generation Steps" on page 63.

If any errors occur in generation validation, NDF notes these errors through diagnostic messages in the report. Table 11 shows the NDF message severity levels and their meanings.

*Table 11. NDF message severity levels (VM)*

| Severity Levels | Meaning |
|---|---|
| Info | This is an informational message that either informs you of NDF calculations (such as message ICN0761) or indicates how NDF has changed, ignored, deleted, or added a keyword. NDF did not consider the message serious enough to stop the generation process; however, you should examine the message to determine whether you want to accept the NDF change or make your own to the generation definition. |
| Warning | An error has occurred for which NDF has taken corrective action by assuming a default keyword value or by ignoring the value supplied. The generation process is terminated after validation of the generation definition. The NDF migration aid also issues a warning message when it cannot determine a value to use. |
| Error | A user error has occurred for which NDF cannot assume a value or ignore the value supplied. The generation process is terminated after validation of the generation definition. |
| Ten | A fatal user error has been detected. The generation process is terminated. |
| Severe | A system error has occurred. NDF produces a procedure traceback. The generation process is terminated after validation of the generation definition. |
| Fatal | A fatal system error has occurred. A procedure traceback is printed and the generation process is terminated. |

For all but the informational messages, NDF ends output of control-block source and link-edit control statements but continues to validate the input definition statements. In this case, you must correct the errors and run the generation validation again. If the return code from the generation validation and the table assemblies is 0, NDF runs to completion, runs the link-edit, and produces a load module.

Other programs, such as VTAM and the configuration report program, require the same definition statements that you used to generate NCP as input, plus additional keywords and definition statements specific to each procedure. *NCP, SSP, and EP Resource Definition Reference* identifies these additional keywords and definition statements. Although you can place these keywords and definition statements in the NCP generation definition either before or after you generate NCP, it is recommended that you add them before you generate NCP because executing the different procedures with different inputs can create errors.

If you are using the NDF standard attachment facility (SSP V3R2 or a later release) to generate user-written code with your NCP, products that use the same generation definition may require as input the source statements or keywords passed to NDF from the user-written generation load modules. By specifying NEWDEFN=YES on the OPTIONS definition statement in your generation definition and by specifying the NEWDEFN file in your EXEC, you instruct NDF to create a new generation definition for use by these procedures. This new generation definition contains both the NCP

source statements and the statements passed from the user-written generation load modules.

NDF only checks the accuracy of the values coded for NCP generation procedures that appear in the NCP generation input. In the same way, these other procedures do not check the validity of the NCP definition statements and keywords. Therefore, NDF requires that the NCP generation have errors less than a severity code of 4.

## Sample NDF Generation Report

Figure 14 on page 78 shows an example of an NDF generation report. Reverse-coded numbers (for example, 3 ) indicate comments about the report that are not a part of the actual report. You can find the comments corresponding to these numbers following the report. Ellipses (. . .) indicate that parts of the report were deleted for this example.

```
1                  2                  3
ACF SSP V3R4.1       01/20/90  18:30:03      DEFINITION SPECIFICATION                                    PAGE 1
4
LINE #               STATEMENT
    83  *         FULL LINE COMMENT 5
    84  PHY01     GROUP ECLTYPE=PHY            NTRI PHYSICAL GROUP  6
                  . . .
                    D               VIRTUAL=NO  7
                    D               TEXTTO=3.0
                    D               COMPACB=NO
    85  PHYLN1    LINE  ADDRESS=(80,FULL),UACB=WRONG,LOCADD=400001234567
                  . . .
                    D               MAXTSL=265
*WARNING  ICN021I   04  UACB=WRONG INVALID, INVALID NUMBER OF SUBOPERANDS, 2 EXPECTED WHEN ECLTYPE=PHYSICAL ON GROUP  8
                    STATEMENT, REPLACED FOR STATEMENT KEYWORD VALIDATION
9            DELETED          UACB
                  ADDED             UACB(1)-X$P1AX  10
                  ADDED             UACB(2)-X$P1AR
                  G                 TYPE=NCP
                  G                 MAXPU=1
                  D                 CLOCKNG=EXT
                  D                 ATTACH=MODEM
11       . . .
GENERATED BY ECL
    86  X$P1SQ    SERVICE
    87  ECLUPU1   PU
                  ADDED             ADDR(1)=01
                  G                 PUTYPE=1
                  D                 AVGPB=532
                  D                 ANS=STOP
                  D                 BNNSUP=(
                  D                 MAXDATA=78
                  . . .
    88  ECLLU1    LU
                  ADDED             LOCADDR(1)=0
                  D                 PACING(1)=1
                  D                 BATCH=NO
                  D                 LUDR=NO
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30      LABEL CROSS REFERENCE                                       PAGE 6

                     LABEL CROSS REFERENCE -- SORTED BY LABEL NAME  12

LABEL         LINE     SA  ELEM       LABEL        LINE     SA  ELEM       LABEL        LINE     SA  ELEM

                  . . .

G14B1         178                     T14020BB     545      0E  001B       T14022G8     915      0E  004D
G14S1         362                     T14020BC     549      0E  001C       T14022G9     916      0E  004E
                  . . .
L14022        828      0E  0044       T14020B8     533      0E  0018       T14023A7     305      0E  0009
L14023        201      0E  0001       T14020B9     537      0E  0019       T14023A8     315      0E  000A
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30      LABEL CROSS REFERENCE                                       PAGE 7

                     LABEL CROSS REFERENCE -- SORTED BY NETWORK ADDRESS  13

SA  ELEM     LINE   LABEL              SA  ELEM    LINE    LABEL              SA  ELEM    LINE    LABEL

0E  0001     201    L14023             0E  0038    747     T14020F8           0E  006E    90      DRPOOLLU
0E  0002     259    B14023A            0E  0039    749     T14020F9           0E  006F    6       NCPBUILD
0E  0003     269    T14023A1           0E  003A    767     T14020G            0E  0070    6       NCPBUILD
0E  0004     275    T14023A2           0E  003B    785     T14020G1           0E  0071    6       NCPBUILD
                  . . .
ACF SSP V3R4.1       11/20/90  18:30:03   ERROR SUMMARY                                                  PAGE 8
                                                    14
TOTAL MESSAGES      INFO     WARNING     ERROR     SEVERE     FATAL
       1             0         0          1         0          0

RETURN CODE IS 8


MESSAGES APPEAR AFTER THE LINES NUMBERED:

85

REGENERATION REQUIRED
```

*Figure 14. Sample NDF generation report (VM)*

## Comments

**1** **SSP version and release number.**

**2** **Date and time of the NDF run.** The date and time of the NDF run are the same as those recorded in the date and time generation control block in the NCP or PEP load module and printed in the formatted portion of the NCP or PEP load module and dump.

3 **Report section identification.** This identification has one of the following values: DEFINITION SPECIFICATION, LABEL CROSS REFERENCE, or ERROR SUMMARY.

4 **Line number column.** This column contains the line numbers of the generation definition listing.

5 **Full-line comments from the generation deck.**

6 **Partial-line comment from the generation deck.**

7 **Information describing defaulted or inherited keywords.** This 1-letter prefix to the message shows keywords that are defaulted or that use values from previous definition statements. These prefixes are:

- G Keyword inherited from GROUP
- L Keyword inherited from LINE
- T Keyword inherited from TERMINAL
- C Keyword inherited from CLUSTER
- P Keyword inherited from PU
- D Keyword that has been defaulted.

8 **Error message.** This error message has an appropriate error number followed by a severity code and error message text. A severity code of 4 or more requires correction to the generation definition before you can generate a load module. A severity code of less than 4 informs you that NDF has taken corrective action and does not require regeneration. You should, however, verify that the correction made by NDF will satisfy your generation requirements.

9 **DELETE message.** This DELETE message indicates keywords replaced by a user-written generation application or replaced by NDF for NCP/Token-Ring interconnection resources.

10 **ADDED or APPENDED message.** This ADDED or APPENDED message indicates keywords passed to NDF by a user-written generation application or added to the generation definition by NDF for NCP/Token-Ring interconnection resources. A keyword is added if you did not code the keyword or if you coded then replaced the keyword; a keyword is appended if you coded the keyword.

11 **GENERATED BY ECL or GENERATED BY** *usergen name*. This GENERATED BY ECL or GENERATED BY statement precedes statements passed to NDF by user-written generation applications using the NDF standard attachment facility or precedes statements added to the generation definition by NDF for NCP/Token-Ring interconnection resources or automatic line replication.

12 **First label cross-reference.** This list contains all user-coded labels, sorted by label name. If the label has an associated network address, it is printed. Not all labels have associated network addresses. Resources defined by LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU keywords appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist. It is included in this sample as an illustration only.

13 **Second label cross reference.** This list contains all user-coded labels, sorted by network address. Labels without associated network addresses are omitted. Resources defined by LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU keywords appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist.

14 **Error summary section.** This summary contains an error count and a list of the line numbers immediately preceding error messages. If more than one error message immediately follows a given line, the line number is printed only once. If only informational messages follow a given line, an asterisk is printed next to the line number.

# Chapter 5. Examples of EXECs for Generation under VM

This chapter contains examples of EXECs written in the REXX language for generating your NCP under the VM operating system. Most of these EXECs are shipped on the SSP tape sent from IBM Software Distribution. They are installed on your SSP BASE disk and have a file type of SMPLEXEC. Before using these examples, ensure that they reflect your operating environment.

This chapter includes examples of EXECs for the following types of generations:

- A FASTRUN generation
- An NCP or PEP generation with output written to disk
- An NCP or PEP generation with output written to tape
- An NCP or PEP generation with user-written code using the NDF standard attachment facility
- An NCP or PEP generation with user-written code using the GENEND definition statement
- A dynamic reconfiguration generation.

## Example of a FASTRUN Generation

Before running a complete generation, you can run a FASTRUN generation to check your generation definition for syntax and definition errors without creating control blocks or link-edit control statements. This example shows the EXEC that generates an NCP load module using FASTRUN generation.

To run a FASTRUN generation:

- Code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for SSP Version 3. You can also code FASTRUN=ON as a parameter in your EXEC when calling NDF using SSP V3R2 and later releases. Figure 15 on page 82 uses the FASTRUN parameter coded in the EXEC.
- Ensure your EXEC *does not* call the linkage editor. If the link-edit step is present, an error results.
- *Do not* define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

This example assumes you have *not* included any user-written code using keywords on the GENEND definition statement. If you did, however, you must include a SYSLIB FILEDEF statement in your EXEC containing the user-written code table assembly and link-edit statements.

The following is an example of a FASTRUN generation.

```
/*********************************************************************/
;
/* EXAMPLE OF A REXX EXEC TO RUN A FASTRUN GENERATION                */
;
/* COPYRIGHT=NONE                                                    */
;
/* YOU MUST SUPPLY THE FILENAME AND FILETYPE OF THE INPUT GENERATION*/
/* DEFINITION ON THE COMMAND LINE WHEN INVOKING THE EXEC.            */
/*                                                                   */
/* THE FOLLOWING PARAMETERS ARE REQUIRED:                            */
/*   1. FN - FILENAME OF YOUR INPUT GENERATION.                      */
/*   2. FT - FILETYPE OF YOUR INPUT GENERATION.                      */
/*                                                                   */
/* THE FOLLOWING PARAMETERS ARE OPTIONAL (NOTE TO EP USERS:          */
/*   DO NOT SPECIFY TUSGTIER, CHANNELS, OR DPU):                     */
/*   1. FM - FILEMODE OF YOUR INPUT GENERATION (DEFAULTS TO '*').    */
/*   2. LINECNT - SPECIFIES NUMBER OF LINES PER PAGE OF THE          */
/*          GENERATION VALIDATION LISTING AND THE TABLE ASSEMBLY     */
/*          LISTING (DEFAULTS TO 60).                                */
/*   3. TVERSION - SPECIFIES A TARGET VERSION FOR THE MIGRATION AID.*/
/*          IF TVERSION IS SPECIFIED, TMODEL AND TUSGTIER MUST       */
/*          ALSO BE SPECIFIED.                                       */
/*   4. TMODEL - SPECIFIES A TARGET MODEL FOR THE MIGRATION AID.     */
/*          IF TMODEL IS SPECIFIED, TVERSION AND TUSGTIER MUST       */
/*          ALSO BE SPECIFIED.                                       */
/*   5. TUSGTIER - SPECIFIES A TARGET USAGE TIER FOR THE MIGRATION   */
/*          AID.  IF TUSGTIER IS SPECIFIED, TVERSION AND TMODEL      */
/*          MUST ALSO BE SPECIFIED.                                  */
/*   6. CHANNELS (3720 USERS ONLY) - SPECIFIES A VALUE FOR THE       */
/*          MIGRATION AID "CHANNELS" PARAMETER (DEFAULTS TO          */
/*          LOCATION OF CHANNEL DEFINITIONS IN GENERATION            */
/*          DEFINITION FROM WHICH YOU ARE MIGRATING).                */
/*   7. DPU- SPECIFIES A VALUE FOR THE MIGRATION AID "DPU"           */
/*          PARAMETER (DEFAULTS TO 'YES').                           */
/*   8. SAVEADDR - SPECIFIES A VALUE FOR THE MIGRATION AID           */
/*          "SAVEADDR" PARAMETER (DEFAULTS TO "YES" WHEN TMODEL      */
/*          AND MODEL ON BUILD STATEMENT ARE THE SAME; DEFAULTS TO */
/*          "NO" WHEN THEY DIFFER).                                  */
/*   9. REMOVCOM - SPECIFIES A VALUE FOR THE "REMOVCOM" PARAMETER    */
/*          (DEFAULTS TO 'NO').                                      */
;
/* CORRECT FORM FOR INVOKING THE EXEC:                               */
/*     VMFAST FN=GEN_FN,FT=GEN_FT,FM=GEN_FM,                         */
/*     LINECNT=NUM_LINES,TVERSION=TARGET_VERSION,                    */
/*     TMODEL=TARGET_MODEL,TUSGTIER=TARGET_USAGE_TIER,               */
/*     CHANNELS=BUILD|GROUP,DPU=YES|NO,SAVEADDR=YES|NO,              */
/*     REMOVCOM=YES|NO                                               */
/*                                                                   */
/*       -ALL THE POSSIBLE PARAMETERS HAVE BEEN SPECIFIED IN THE     */
/*        ABOVE EXAMPLE, FOR THE SAKE OF ILLUSTRATION.  IN           */
/*        PRACTICE, A SUBSET OF THE PARAMETERS WOULD BE CODED.       */
/*       -GEN_FN, GEN_FT, AND GEN_FM ARE VARIABLES THAT YOU          */
/*        SUPPLY ACCORDING TO YOUR GENERATION                        */
/*       -ORDER OF PARAMETERS IS NOT IMPORTANT                       */
/*       -SPACES MAY BE USED INSTEAD OF COMMAS                       */
```

*Figure 15 (Part 1 of 5). Example of a FASTRUN generation (VM)*

```
;
;
;
  ADDRESS COMMAND                      /* ENSURE CP/CMS ENVIRONMENT  */
  TRACE N
  GEN_FN=""
  GEN_FT=""                            /* INITIALIZE STRING VARIABLES*/
  GEN_FM=""
  ARG REST                             /* GET PARAMETERS FROM COMMAND*/
                                       /* LINE                       */
  REST=TRANSLATE(REST,' ',',')         /* GET RID OF COMMAS          */
  COUNT=WORDS(REST)
  LPCNT=1
  OPTIONS="(FASTRUN(ON)"
  NUMOPTS=1
  DO WHILE LPCNT<=COUNT                /* LOOP THROUGH ONCE FOR EACH */
    TEMP=WORD(REST,LPCNT)              /* WORD IN THE STRING         */
    PARSE VALUE TEMP WITH FRONT '=' BACK
    SELECT
      WHEN (ABBREV(TEMP,'FN')) THEN
        GEN_FN=BACK
      WHEN (ABBREV(TEMP,'FT')) THEN    /* SET APPROPRIATE VARIABLE   */
        GEN_FT=BACK                    /* ACCORDING TO THE ASSIGNMENT*/
      WHEN (ABBREV(TEMP,'FM')) THEN    /* MADE ON THE COMMAND LINE   */
        GEN_FM=BACK
      WHEN (ABBREV(TEMP,'LINECNT')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'TVERSION')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'TMODEL')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'TUSGTIER')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
```

*Figure 15 (Part 2 of 5). Example of a FASTRUN generation (VM)*

```
      WHEN (ABBREV(TEMP,'CHANNELS')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'DPU')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'SAVEADDR')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'REMOVCOM')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'FASTRUN')) THEN
        NOP
      OTHERWISE
        SAY TEMP" IS NOT VALID, IGNORED"
    END                                 /* END SELECT                */
    LPCNT=LPCNT+1
  END                                   /* END DO                    */
;
  IF NUMOPTS > 8 THEN
    EXIT
;
  IF GEN_FM="" THEN                     /* DEFAULT FILETYPE TO "*" IF */
    GEN_FM="*"                          /* NOT CODED                 */
;
;
/* SEE IF GEN_FN AND GEN_FT WERE PASSED ON COMMAND LINE.              */
/* IF GEN NAME WAS NOT SPECIFIED, GIVE CORRECT FORM AND EXIT.         */
;
```

*Figure 15 (Part 3 of 5). Example of a FASTRUN generation (VM)*

```
  IF (GEN_FN="") THEN
    DO
      SAY "FN PARAMETER MISSING; SPECIFY AS FN=GEN_FN"
      EXIT
    END
  IF (GEN_FT="") THEN
    DO
      SAY "FT PARAMETER MISSING; SPECIFY AS FT=GEN_FT"
      EXIT
    END
;
  'ESTATE' GEN_FN GEN_FT GEN_FM       /* SEE IF GEN EXISTS ON DISK  */
  IF RC ¬= 0 THEN
    DO
      SAY GEN_FN GEN_FT GEN_FM "DOES NOT EXIST"
      EXIT RC                         /* EXIT IF GEN DOESN'T EXIST  */
    END
;
;
 /* CLEAR OLD FILE DEFINITIONS                                      */
   'FILEDEF * CLEAR'
 /* WORKING SPILL FILE                                              */
 /* THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH VIRTUAL    */
 /* MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF USERGEN IS SPECIFIED.*/
 /*'FILEDEF DBWORKFL DISK  DBWORKFL FILE A ( XTENT 40'             */
;
 /* IF NEWDEFN=YES IS SPECIFIED OR DEFAULTED IN THE                 */
 /* GENERATION DEFINITION, A FILE DEFINITION SIMILAR TO THE         */
 /* FOLLOWING IS NEEDED.                                            */
 /*'FILEDEF NEWDEFN DISK NEWFAST FILE A'                            */
;
 /* INPUT FILE WITH NCP/EP GENERATION STATEMENTS                    */
   'FILEDEF GENDECK DISK' GEN_FN GEN_FT GEN_FM
 /* GENERATION VALIDATION STEP OUTPUT                               */
   'FILEDEF SYSPRINT DISK' GEN_FN 'LISTING A'
 /* NDF SUMMARY LISTING                                             */
   'FILEDEF PRINTER TERM'
 /* RUN THE NDF STEP                                                */
 /* FASTRUN IS SET ON BY SPECIFYING IT ON THE FIRST EXECUTABLE      */
 /* STATEMENT ON THE GENERATION DECK AND/OR BY INCLUDING IT IN THE  */
 /* PARAMETER LIST WHEN INVOKING ICNRTNDF (IT IS IMBEDDED IN        */
 /* "OPTIONS" BELOW).                                               */
   'ICNRTNDF' OPTIONS
   EXIT RC
CHKSUM:
```

*Figure 15 (Part 4 of 5). Example of a FASTRUN generation (VM)*

```
/*****************************************************************/
/* CHECK SUM OF ICNRTNDF PARAMETERS AGAINST LIMIT                */
/*****************************************************************/
NUMOPTS=NUMOPTS+1                    /* INCREMENT COUNTER        */
IF NUMOPTS > 8 THEN
  DO
    SAY " "
    SAY TEMP" INVALID, THE MAXIMUM NUMBER OF ICNRTNDF OPTIONS"
    SAY "(8) HAVE ALREADY BEEN SPECIFIED;  MOVE EXTRA OPTIONS"
    SAY "TO OPTIONS STATEMENT IN NCP GENERATION DEFINITION."
  END
RETURN
```

*Figure 15 (Part 5 of 5). Example of a FASTRUN generation (VM)*

# Example of an NCP or PEP Generation with Output Written to Disk

When running a standard NCP or PEP generation, you supply your generation definition as input and specify the various input and output files in your EXEC. You can specify that input and output files be written to disk or tape. Figure 16 on page 87 shows the EXEC that generates an NCP load module for an IBM 3745 Communication Controller with output files written to disk.

When reading this example, remember the following differences among the communication controllers:

- The EXEC is slightly different.
- The NCP definition statement library used for NDF may be different.
- The FILEDEF for the library of preassembled NCP object modules in the link-edit step may be different.

**IBM 3705, 3720, or 3725 Communication Controller:** When running the link-edit step for a standard NCP or PEP generation on the IBM 3705, 3725, or 3720 Communication Controller, specify the ALIGN2 option in the EXEC for the link-edit step. ALIGN2 ensures that certain control sections within the load module are aligned on 2KB page boundaries. If you do not specify ALIGN2, the default is alignment on 4KB page boundaries, which may use excessive communication controller storage.

**IBM 3745 Communication Controller:** *Do not specify* ALIGN2. The default is alignment on 4KB page boundaries, the correct alignment for the IBM 3745 Communication Controller.

The following list shows when to specify ALIGN2 as the link-edit step:

| Communication Controller | ALIGN2 |
|---|---|
| 3705 | Yes |
| 3720 | Yes |
| 3725 | Yes |
| 3745 | No. |

The following is an example of an NCP or PEP generation with output written to disk.

```
/*****************************************************************/
;
/* EXAMPLE OF AN EXEC TO RUN A NCP/PEP GENERATION WITH ALL OUTPUT   */
/* FILES WRITTEN TO DISK                                           */
;
/* COPYRIGHT=NONE                                                  */
;
/* THIS SAMPLE CAN BE USED TO GENERATE AN NCP WITH IBM SPECIAL      */
/* PRODUCTS IF THE MACROS AND OBJECT MODULES FOR THESE PRODUCTS HAVE*/
/* BEEN MERGED INTO THE APPROPRIATE NCP LIBRARIES.                 */
;
;
/* YOU MUST SUPPLY THE FILENAME AND FILETYPE OF THE INPUT GENERATION*/
/* DEFINITION ON THE COMMAND LINE WHEN INVOKING THE EXEC.  YOU MUST */
/* SUPPLY THE MODEL NUMBER OF THE CONTROLLER.                      */
/*                                                                 */
/* THE FOLLOWING PARAMETERS ARE REQUIRED:                          */
/*   1. FN - FILENAME OF YOUR INPUT GENERATION.                    */
/*   2. FT - FILETYPE OF YOUR INPUT GENERATION.                    */
/*   3. M  - MODEL NUMBER OF THE CONTROLLER.                       */
/*                                                                 */
/* THE FOLLOWING PARAMETERS ARE OPTIONAL:                          */
/*   1. FM - FILEMODE OF YOUR INPUT GENERATION (DEFAULTS TO '*').  */
/*   2. V - VERSION OF THE GENERATION (DEFAULTS TO 'V3', OR        */
/*         TO TVERSION IF SPECIFIED).                              */
/*   3. T - SPECIFIES WHETHER TEST NCP LIBRARIES ARE IN USE        */
/*         (DEFAULTS TO 'NO').                                     */
/*   4. LINECNT - SPECIFIES NUMBER OF LINES PER PAGE OF THE        */
/*          GENERATION VALIDATION LISTING AND THE TABLE ASSEMBLY   */
/*          LISTING (DEFAULTS TO 60).                              */
/*   5. FASTRUN - SPECIFY "FASTRUN=ON" TO MAKE NDF DO KEYWORD      */
/*          VALIDATION ONLY.                                       */
/*   6. ASSEMBLY - SPECIFY "ASSEMBLY=YES" IF YOU ARE INVOKING NDF  */
/*          FOR THE SOLE PURPOSE OF ACCESSING THE NDF CONTROLLER   */
/*          ASSEMBLER TO ASSEMBLE TABLE SOURCE CODE.               */
/*   7. TVERSION - SPECIFIES A TARGET VERSION FOR THE MIGRATION AID.*/
/*          IF TVERSION IS SPECIFIED, TMODEL AND TUSGTIER MUST     */
/*          ALSO BE SPECIFIED.                                     */
/*   8. TMODEL - SPECIFIES A TARGET MODEL FOR THE MIGRATION AID.   */
/*          IF TMODEL IS SPECIFIED, TVERSION AND TUSGTIER MUST     */
/*          ALSO BE SPECIFIED.                                     */
/*   9. TUSGTIER - SPECIFIES A TARGET USAGE TIER FOR THE MIGRATION  */
/*          AID.  IF TUSGTIER IS SPECIFIED, TVERSION AND TMODEL    */
/*          MUST ALSO BE SPECIFIED.                                */
```

*Figure 16 (Part 1 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/*  10. CHANNELS - SPECIFIES A VALUE FOR THE MIGRATION AID          */
/*           "CHANNELS" PARAMETER (DEFAULTS TO LOCATION OF CHANNEL  */
/*           DEFINITIONS IN GENERATION DEFINITION FROM WHICH YOU    */
/*           ARE MIGRATING).                                        */
/*  11. DPU- SPECIFIES A VALUE FOR THE MIGRATION AID "DPU"          */
/*           PARAMETER (DEFAULTS TO 'YES').                         */
/*  12. SAVEADDR - SPECIFIES A VALUE FOR THE MIGRATION AID          */
/*           "SAVEADDR" PARAMETER (DEFAULTS TO "YES" WHEN TMODEL    */
/*           AND MODEL ON BUILD STATEMENT ARE THE SAME; DEFAULTS TO */
/*           "NO" WHEN THEY DIFFER).                                */
/*  13. REMOVCOM - SPECIFIES A VALUE FOR THE "REMOVCOM" PARAMETER   */
/*           (DEFAULTS TO 'NO').                                    */
;
/* CORRECT FORM FOR INVOKING THE EXEC:                              */
/*   VMNCP FN=GEN_FN,FT=GEN_FT,FM=GEN_FM,V=VERSION,M=MODEL,         */
/*     T=YES|NO,LINECNT=NUM_LINES,FASTRUN=ON,ASSEMBLY=YES,          */
/*     TVERSION=TARGET_VERSION,TMODEL=TARGET_MODEL,                 */
/*     TUSGTIER=TARGET_USAGE_TIER,CHANNELS=BUILD|GROUP,DPU=YES|NO,  */
/*     SAVEADDR=YES|NO,REMOVCOM=YES|NO                              */
/*                                                                  */
/*       -ALL THE POSSIBLE PARAMETERS HAVE BEEN SPECIFIED IN THE    */
/*        ABOVE EXAMPLE, FOR THE SAKE OF ILLUSTRATION.  IN          */
/*        PRACTICE, A SUBSET OF THE PARAMETERS WOULD BE CODED.      */
/*       -GEN_FN, GEN_FT, GEN_FM, VERSION, AND MODEL ARE VARIABLES  */
/*        THAT YOU SUPPLY ACCORDING TO YOUR GENERATION              */
/*       -YOU MAY CODE 'T=YES' FOR A RUN ON TEST NCP LIBRARIES      */
/*       -ORDER OF PARAMETERS IS NOT IMPORTANT                      */
/*       -SPACES MAY BE USED INSTEAD OF COMMAS                      */
/*       -FOR NCP SUBSET, CODE V=V4S                                */
;
;
/* THE ASSIGNMENT OF THE MACRO AND OBJECT LIBRARY NAMES IS DERIVED  */
/* FROM THE PARAMETERS PASSED ON THE COMMAND LINE; THEY MAY RESIDE  */
/* ON ANY ACCESSED DISK.                                            */
/*                                                                  */
/* VARIABLES ARE DEFINED AS FOLLOWS:                                */
/*      MACRO = MACLIB NAME                                         */
/*      OBJECT = OBJLIB NAME                                        */
;
```

*Figure 16 (Part 2 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/*****************************************************************/
/* THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT              */
/* LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOUR */
/* LIBRARY NAMES DO NOT AGREE WITH THIS TABLE, YOU WILL NEED TO  */
/* SUBSTITUTE YOUR LIBRARY NAME FOR THE STANDARD LIBRARY NAME WHERE */
/* APPROPRIATE.  YOU CAN USE THE "ALL" COMMAND ON THE FOLLOWING  */
/* STRINGS TO FIND LINES TO CHANGE FOR A PARTICULAR VERSION & MODEL.*/
/*                                                               */
/*     ALLTAG1 - TEST                                            */
/*     ALLTAG2 - V3 & 3705                                       */
/*     ALLTAG3 - V3 & 3725                                       */
/*     ALLTAG4 - V4R2 & 3725 OR 3720                             */
/*     ALLTAG5 - V4 SUBSET & 3720                                */
/*     ALLTAG6 - V4R3 & 3725                                     */
/*     ALLTAG7 - V5R2 & 3720 OR 3745                             */
/*     ALLTAG8 - V4R3.1 & 3725                                   */
/*     ALLTAG9 - V5R2.1 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR */
/*                       3745-170 OR 3745-210 OR 3745-410        */
/*     ALLTAGA - V5R3 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR  */
/*                       3745-170 OR 3745-210 OR 3745-410        */
/*     ALLTAGB - V5R4 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR  */
/*                       3745-170 OR 3745-210 OR 3745-410        */
/*                                                               */
/* 'T=YES' IS ALLOWED FOR TESTING TO FACILITATE LIBRARY MAINTENANCE.*/
/* (YOU MUST GO TO THE APPROPRIATE PLACE AND KEY IN YOUR TEST-LIB */
/*  NAMES TO USE T=YES.)                                         */
/* T=YES WILL RUN WITH ANY MODEL AND VERSION (YOU MUST CODE MODEL). */
/*                                                               */
```

*Figure 16 (Part 3 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/*****************************************************************/
/*                                                               */
/*                              M O D E L                        */
/*                                                               */
/*              3705         3725         3720         3745     */
/*            ---------------------------------------------      */
/*    V3/3705| MAC3705   |  NOT       |  NOT       |  NOT       | */
/*           | OBJ3705   |  SUPPORTED |  SUPPORTED |  SUPPORTED | */
/*           |-------------------------------------------------| */
/*    V3/3725| NOT       |  MAC3725   |  NOT       |  NOT       | */
/*  V        | SUPPORTED |  OBJ3725   |  SUPPORTED |  SUPPORTED | */
/*  E        |-------------------------------------------------| */
/*  R V4R1   | NOT       |  NOT       |  NOT       |  NOT       | */
/*  S        | SUPPORTED |  SUPPORTED |  SUPPORTED |  SUPPORTED | */
/*  I        |-------------------------------------------------| */
/*  O V4R2   | NOT       |  MAC3725   |  MAC3725   |  NOT       | */
/*  N        | SUPPORTED |  OBJ3725   |  OBJ3725   |  SUPPORTED | */
/*           |-------------------------------------------------| */
/*    V4     | NOT       |  NOT       |  MAC3725   |  NOT       | */
/*    SUBSET | SUPPORTED |  SUPPORTED |  OBJ3725   |  SUPPORTED | */
/*           |-------------------------------------------------| */
/*    V4R3 & | NOT       |  SNCPMAC1  |  NOT       |  NOT       | */
/*    LATER  | SUPPORTED |  SNCPMOD1  |  SUPPORTED |  SUPPORTED | */
/*           |-------------------------------------------------| */
/*    V5R1   | NOT       |  NOT       |  MAC3725   |  MAC3725   | */
/*           | SUPPORTED |  SUPPORTED |  OBJ3725   |  OBJ3725   | */
/*           |-------------------------------------------------| */
/*    V5R2 & | NOT       |  NOT       |  SNCPMAC1  |  SNCPMAC1  | */
/*    LATER  | SUPPORTED |  SUPPORTED |  SNCPMOD1  |  SNCPMOD1  | */
/*            ---------------------------------------------      */
/*                                                               */
/*                              M O D E L                        */
/*                                                               */
/*          3745-130, 3745-150,                                  */
/*          3745-170, 3745-210,                                  */
/*          3745-410                                             */
/*          -------------                                        */
/*    V3    | NOT       |                                        */
/*          | SUPPORTED |                                        */
/*          |-------------                                       */
/*    V4    | NOT       |                                        */
/*  V       | SUPPORTED |                                        */
/*  E       |-------------                                       */
/*  R V5R2 & | NOT       |                                       */
/*  S BEFORE | SUPPORTED |                                       */
/*  I       |-------------                                       */
/*  O V5R2.1 | SNCPMAC1  |                                       */
/*  N & LATER| SNCPMOD1  |                                       */
/*          -------------                                        */
/*                                                               */
/*                                                               */
/*****************************************************************/
```

*Figure 16 (Part 4 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/*********************************************************************/
/* CHANGE ACTIVITY                                                   */
/*    - BUILD CMS LOADLIB FROM TABLE1 & 2 TEXT FILES IF     @A550438*/
/*      TXTLIB COMMAND FAILS                                @A550438*/
/*                                                                   */
/*********************************************************************/
;
  ADDRESS COMMAND                    /* ENSURE CP/CMS ENVIRONMENT  */
  TRACE N
  GEN_FN=""
  GEN_FT=""                          /* INITIALIZE STRING VARIABLES*/
  GEN_FM=""
  VERSION=""
  TVERSION=""
  TMODEL=""
  MODEL=""
  T="NO"
  ARG REST                           /* GET PARAMETERS FROM COMMAND*/
                                     /* LINE                       */
  REST=TRANSLATE(REST,' ',',')       /* GET RID OF COMMAS          */
  COUNT=WORDS(REST)
  LPCNT=1
  OPTIONS="("
  FSTRUN="FALSE"
  NUMOPTS=0
  DO WHILE LPCNT<=COUNT              /* LOOP THROUGH ONCE FOR EACH */
    TEMP=WORD(REST,LPCNT)            /* WORD IN THE STRING         */
    PARSE VALUE TEMP WITH FRONT '=' BACK
    SELECT
      WHEN (ABBREV(TEMP,'FN')) THEN
        GEN_FN=BACK
      WHEN (ABBREV(TEMP,'FT')) THEN  /* SET APPROPRIATE VARIABLE   */
        GEN_FT=BACK                  /* ACCORDING TO THE ASSIGNMENT*/
      WHEN (ABBREV(TEMP,'FM')) THEN  /* MADE ON THE COMMAND LINE   */
        GEN_FM=BACK
      WHEN (ABBREV(TEMP,'V')) THEN
        VERSION=BACK
      WHEN (ABBREV(TEMP,'M')) THEN
        MODEL=BACK
      WHEN (ABBREV(TEMP,'LINECNT')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'FASTRUN')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          FSTRUN="TRUE"
          CALL CHKSUM
        END
```

*Figure 16 (Part 5 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
      WHEN (ABBREV(TEMP,'ASSEMBLY')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'TVERSION')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
          TVERSION=BACK
        END
      WHEN (ABBREV(TEMP,'TMODEL')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          TMODEL=BACK
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'TUSGTIER')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'CHANNELS')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'DPU')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'SAVEADDR')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'REMOVCOM')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'T')) THEN
        T=BACK
      OTHERWISE
        SAY TEMP" IS NOT VALID, IGNORED"
    END                              /* END SELECT                */
    LPCNT=LPCNT+1
  END                                /* END DO                    */
;
```

*Figure 16 (Part 6 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
  IF NUMOPTS > 8 THEN
    EXIT
;
  IF GEN_FM="" THEN                    /* DEFAULT FILETYPE TO "*" IF */
    GEN_FM="*"                         /* NOT CODED                  */
;
;
/* SEE IF GEN_FN, GEN_FT, AND MODEL WERE PASSED ON COMMAND LINE.    */
/* IF GEN NAME OR MODEL WERE NOT SPECIFIED GIVE CORRECT FORM & EXIT.*/
;
  IF (GEN_FN="") THEN
    DO
      SAY "FN PARAMETER MISSING; SPECIFY AS FN=GEN_FN"
      EXIT
    END
  IF (GEN_FT="") THEN
    DO
      SAY "FT PARAMETER MISSING; SPECIFY AS FT=GEN_FT"
      EXIT
    END
  IF (MODEL="" & FSTRUN="FALSE")
  THEN
    IF TMODEL="" THEN
      DO
        SAY "M PARAMETER MISSING; SPECIFY AS M=MODEL"
        EXIT
      END
    ELSE
      DO
        MODEL=TMODEL
        SAY "DEFAULTING TO M="MODEL
      END
;
;
  'ESTATE' GEN_FN GEN_FT GEN_FM        /* SEE IF GEN EXISTS ON DISK  */
  IF RC ¬= 0 THEN
    DO
      SAY GEN_FN GEN_FT GEN_FM "DOES NOT EXIST"
      EXIT RC                          /* EXIT IF GEN DOESN'T EXIST  */
    END
;
;
```

*Figure 16 (Part 7 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/* THIS STRUCTURE VALIDATES VERSION AND MODEL AS TAKEN FROM THE     */
/* COMMAND LINE AND SETS "MACRO" AND "OBJECT" ACCORDINGLY.          */
/*                                                                  */
/* NOTE:  IF YOU HAVE CHANGED A LIBRARY NAME, OR YOUR LIBRARY NAMES */
/* DO NOT AGREE WITH THE TABLE ABOVE, CHANGE THE ASSIGNMENT         */
/* STATEMENTS OF "MACRO" AND "OBJECT" TO REFLECT YOUR LIBRARY NAMES */
/* IN THE APPROPRIATE PLACE IN THIS STRUCTURE (ACCORDING TO YOUR    */
/* VERSION AND MODEL).                                              */
;
;
  IF FSTRUN="FALSE"                  /* FASTRUN NOT SPECIFIED      */
  THEN
  DO
   IF TVERSION¬=""                   /* TVERSION SPECIFIED         */
   THEN
    IF VERSION="" THEN               /* VERSION NOT SPECIFIED      */
      DO
        SAY "DEFAULTING TO V="TVERSION
        VERSION=TVERSION
      END
    ELSE                             /* VERSION SPECIFIED          */
      IF VERSION ¬= TVERSION
      THEN
        DO
          SAY "V="VERSION "INVALID - WHEN TVERSION IS SPECIFIED,"
          SAY "V AND TVERSION MUST BE EQUAL"
          EXIT
        END
  IF (VERSION="") THEN               /* VERSION DEFAULTS TO V3     */
    DO                               /* IF NOT CODED               */
      SAY "DEFAULTING TO V=V3"
      VERSION="V3"
    END
  SELECT
    WHEN (T="YES") THEN
      DO
        MACRO=MACXXXX                /* FOR TEST           ALLTAG1*/
        OBJECT=OBJXXXX               /*                    ALLTAG1*/
        DDNAME=OBJXXXX               /*                    ALLTAG1*/
      END
```

*Figure 16 (Part 8 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
WHEN (VERSION="V3")&(MODEL="3705") THEN
  DO
    MACRO=MAC3705                 /* FOR V3 & 3705       ALLTAG2*/
    OBJECT=OBJ3705                /*                     ALLTAG2*/
    DDNAME=OBJ3705                /*                     ALLTAG2*/
  END
WHEN (VERSION="V3")&(MODEL="3725") THEN
  DO
    MACRO=MAC3725                 /* FOR V3 & 3725       ALLTAG3*/
    OBJECT=OBJ3725                /*                     ALLTAG3*/
    DDNAME=OBJ3725                /*                     ALLTAG3*/
  END
WHEN (VERSION="V4R2")&((MODEL="3725")|(MODEL="3720")) THEN
  DO
    MACRO=MAC3725                 /* FOR V4R2&3725|3720  ALLTAG4*/
    OBJECT=OBJ3725                /*                     ALLTAG4*/
    DDNAME=OBJ3725                /*                     ALLTAG4*/
  END
WHEN (VERSION="V4S")&(MODEL="3720") THEN
  DO
    MACRO=MAC3725                 /* FOR V4 SUBSET       ALLTAG5*/
    OBJECT=OBJ3725                /* & 3720              ALLTAG5*/
    DDNAME=OBJ3725                /*                     ALLTAG5*/
  END
WHEN (VERSION="V4R3")&(MODEL="3725") THEN
  DO
    MACRO=SNCPMAC1                /* FOR V4R3            ALLTAG6*/
    OBJECT=SNCPMOD1               /* & 3725              ALLTAG6*/
    DDNAME=ANCPMOD1               /*                     ALLTAG6*/
  END
WHEN (VERSION="V5R2")&((MODEL="3720")|(MODEL="3745")) THEN
  DO
    MACRO=SNCPMAC1                /* FOR V5R2            ALLTAG7*/
    OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAG7*/
    DDNAME=ANCPMOD1               /*                     ALLTAG7*/
  END
WHEN (VERSION="V4R3.1")&(MODEL="3725") THEN
  DO
    MACRO=SNCPMAC1                /* FOR V4R3.1          ALLTAG8*/
    OBJECT=SNCPMOD1               /* & 3725              ALLTAG8*/
    DDNAME=ANCPMOD1               /*                     ALLTAG8*/
  END
```

*Figure 16 (Part 9 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
    WHEN (VERSION="V5R2.1")&((MODEL="3720")|(MODEL="3745")|,
      (MODEL="3745-130")|,
      (MODEL="3745-150")|,
      (MODEL="3745-170")|,
      (MODEL="3745-210")|,
      (MODEL="3745-410")) THEN
      DO
        MACRO=SNCPMAC1                /* FOR V5R2.1          ALLTAG9*/
        OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAG9*/
        DDNAME=ANCPMOD1               /*                     ALLTAG9*/
      END
    WHEN (VERSION="V5R3")&((MODEL="3720")|(MODEL="3745")|,
      (MODEL="3745-130")|,
      (MODEL="3745-150")|,
      (MODEL="3745-170")|,
      (MODEL="3745-210")|,
      (MODEL="3745-410")) THEN
      DO
        MACRO=SNCPMAC1                /* FOR V5R3            ALLTAGA*/
        OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAGA*/
        DDNAME=ANCPMOD1               /*                     ALLTAGA*/
      END
    WHEN (VERSION="V5R4")&((MODEL="3720")|(MODEL="3745")|,
      (MODEL="3745-130")|,
      (MODEL="3745-150")|,
      (MODEL="3745-170")|,
      (MODEL="3745-210")|,
      (MODEL="3745-410")) THEN
      DO
        MACRO=SNCPMAC1                /* FOR V5R4            ALLTAGB*/
        OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAGB*/
        DDNAME=ANCPMOD1               /*                     ALLTAGB*/
      END
    OTHERWISE
      DO
        SAY "VERSION = "VERSION" NOT VALID WITH MODEL = "MODEL
        SAY "WHEN RUNNING UNDER VM"
        EXIT
      END
  END                                 /* END SELECT                */
  END                                 /* FASTRUN NOT SPECIFIED     */
;
;
  IF FSTRUN="FALSE" THEN
    DO
      'ESTATE' MACRO 'MACLIB *'       /* SEE IF MACLIB EXISTS      */
      IF RC ¬= 0 THEN
        SAY "ERROR IN ACCESSING" MACRO "MACLIB"
    END
;
;
```

*Figure 16 (Part 10 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
 /* CLEAR OLD FILE DEFINITIONS                                          */
   'FILEDEF * CLEAR'
 /* WORKING SPILL FILE                                                  */
 /* THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH VIRTUAL        */
 /* MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF USERGEN IS SPECIFIED.*/
 /*'FILEDEF DBWORKFL DISK DBWORKFL FILE A ( XTENT 40'                  */
;
 /* IF NEWDEFN=YES IS SPECIFIED OR DEFAULTED IN THE                     */
 /* GENERATION DEFINITION, A FILE DEFINITION SIMILAR TO THE             */
 /* FOLLOWING IS NEEDED.                                                */
 /*'FILEDEF NEWDEFN DISK NEWDEFN FILE A'                               */
;
 /* MACRO LIBRARIES USED IN THE TABLE ASSEMBLY PHASE OF NDF             */
   IF FSTRUN="FALSE" THEN
     DO
       'FILEDEF SYSLIB DISK' MACRO 'MACLIB *'
       'GLOBAL MACLIB' MACRO
     END
 /* INPUT FILE WITH NCP/EP GENERATION STATEMENTS                        */
   'FILEDEF GENDECK DISK' GEN_FN GEN_FT GEN_FM
 /* GENERATION VALIDATION STEP OUTPUT                                   */
   'FILEDEF SYSPRINT DISK' GEN_FN 'LISTING A'
 /* NDF SUMMARY LISTING                                                 */
   'FILEDEF PRINTER TERM'
 /* SOURCE FOR TABLE 1 ASSEMBLY - OUTPUT FROM GENERATION VALIDATION     */
   'FILEDEF TBL1SRCE DISK TABLE1 SOURCE A'
 /* LISTING FROM THE TABLE 1 ASSEMBLY                                   */
   'FILEDEF TBL1LIST DISK TABLE1 LISTING A'
 /* TEXT OUTPUT FROM THE TABLE 1 ASSEMBLY                               */
   'FILEDEF TBL1OBJ DISK TABLE1 TEXT A'
 /* SOURCE FOR TABLE 2 ASSEMBLY - OUTPUT FROM GENERATION VALIDATION     */
   'FILEDEF TBL2SRCE DISK TABLE2 SOURCE A'
 /* LISTING FROM THE TABLE 2 ASSEMBLY                                   */
   'FILEDEF TBL2LIST DISK TABLE2 LISTING A'
 /* TEXT OUTPUT FROM THE TABLE 2 ASSEMBLY                               */
   'FILEDEF TBL2OBJ DISK TABLE2 TEXT A'
 /* LINK EDIT STATEMENTS OUTPUT FROM THE GENERATION VALIDATION STEP     */
   'FILEDEF LNKSTMT DISK NCPINCL TEXT A'
 /* TEMPORARY WORK FILE USED BY THE TABLE ASSEMBLIES                    */
   'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4'
 /* RUN THE NDF STEP                                                    */
   'ICNRTNDF' OPTIONS
 /* EXIT BECAUSE OF AN ERROR DURING GENERATION VALIDATION               */
   IF RC ¬= 0 THEN
     DO
       SAY "***ERROR IN EXECUTING NDF***"
       EXIT RC
     END
   IF FSTRUN="TRUE" THEN
     EXIT RC
```

*Figure 16 (Part 11 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/* PUT TEXT OUTPUT FROM TABLE ASSEMBLIES INTO A SIMULATED PDS       */
  SET CMSTYPE HT                                                  /*@A550438*/
      'FILEDEF SYSLIB CLEAR'                                      /*@A550438*/
      /*************************************************************/
      /* BUILD THE TABLE 1 LOADLIB                                 */
      /*************************************************************/
      'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4 (BLKSIZE 8192'     /* JM */
      'FILEDEF SYSLMOD DISK ICNTABL1 LOADLIB A (BLKSIZE 8192'  /*JM*/
      'FILEDEF TABLE1 DISK TABLE1 TEXT A'                         /*@A550438*/
      LINE=' INCLUDE TABLE1'                                      /*@A550438*/
      'EXECIO 1 DISKW' ICNTABL1 TEXT A 1 F '(VAR LINE'            /*@A550438*/
      FINIS ICNTABL1 TEXT A                                       /*@A550438*/
      'LKED ICNTABL1 (NCAL LET NOTERM SIZE 2300K'              /* JM */
      /*************************************************************/
      /* BUILD THE TABLE 2 LOADLIB                                 */
      /*************************************************************/
      'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4 (BLKSIZE 8192'     /* JM */
      'FILEDEF SYSLMOD DISK ICNTABL2 LOADLIB A (BLKSIZE 8192'  /*JM*/
      'FILEDEF TABLE2 DISK TABLE2 TEXT A'                         /*@A550438*/
      LINE=' INCLUDE TABLE2'                                      /*@A550438*/
      'EXECIO 1 DISKW' ICNTABL2 TEXT A 1 F '(VAR LINE'            /*@A550438*/
      FINIS ICNTABL2 TEXT A                                       /*@A550438*/
      'LKED ICNTABL2 (NCAL LET NOTERM SIZE 2300K'              /* JM */
      /*************************************************************/
      /* BUILD THE LINKAGE EDITOR SYSIN CONTROL STATEMENT          */
      /*************************************************************/
      LINE=' '                                                    /*@A550438*/
      'EXECIO 1 DISKW' NDFSYSIN FILE A 1 F '(VAR LINE'            /*@A550438*/
      FINIS NDFSYSIN FILE A                                       /*@A550438*/
      /*********************************************************@A550438*/
      /* COPY THE TABLE 2 LOADLIB INTO THE TABLE 1 LOADLIB  @A550438*/
      /*********************************************************@A550438*/
      'LOADLIB COPY ICNTABL2 LOADLIB A ICNTABL1 LOADLIB A
         NDFSYSIN FILE A (MODIFY'                                 /*@A550438*/
      SET CMSTYPE RT                                              /*@A550438*/
      'COPY ICNTABL1 LOADLIB A OBJ LOADLIB A (REP'                /*@A270000*/
      'ERASE ICNTABL1 TEXT A'                                     /*@A550438*/
      'ERASE ICNTABL1 LKEDIT A'                                   /*@A550438*/
      'ERASE ICNTABL2 TEXT A'                                     /*@A550438*/
      'ERASE ICNTABL2 LKEDIT A'                                   /*@A550438*/
      'ERASE ICNTABL1 LOADLIB A'                                  /*@A550438*/
      'ERASE ICNTABL2 LOADLIB A'                                  /*@A550438*/
      'ERASE NDFSYSIN FILE A'                                     /*@A550438*/
      'FILEDEF SYSPUNCH DISK OBJ LOADLIB A (RECFM U'              /*@A550438*/
```

*Figure 16 (Part 12 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
  /*************************************************************@A550438*/
  /* ERASE TEMPORARY FILES                                               */
  /*************************************************************@A550438*/
  'ERASE SYSUT1 TEMP A'
  'ERASE TABLE1 SOURCE'
  'ERASE TABLE2 SOURCE'
  'ERASE TABLE1 TEXT'
  'ERASE TABLE2 TEXT'
;
  'ESTATE' OBJECT 'TXTLIB *'      /* SEE IF SNCPMOD1/OBJ3725 EXISTS */
  IF RC ¬= 0 THEN
    DO                                                        /*@A550438*/
      SAY "ERROR IN ACCESSING" OBJECT "TXTLIB"
      EXIT(RC)                                                /*@A550438*/
    END                                                       /*@A550438*/
;
  /*********************************************************************/
  /* FILDEFS FOR THE LINK EDIT STEP                                    */
  /*********************************************************************/
  'FILEDEF SYSUT1 CLEAR'
  'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4 (BLKSIZE 8192'          /* JM */
  'FILEDEF' DDNAME 'DISK' OBJECT 'TXTLIB *'
;
  /*********************************************************************/
  /* NAME OF OUTPUT LIBRARY FOR THE LOAD MODULE                        */
  /*********************************************************************/
  'FILEDEF SYSLMOD DISK' GEN_FN 'LOADLIB A (BLKSIZE 8192'      /* JM */
;
  /*********************************************************************/
  /* RUN LINKAGE EDITOR                                                */
  /* NOTE: THE ALIGN2 PARAMETER IS CODED ONLY FOR THE IBM              */
  /*       3705/3720/3725, SINCE IT RESULTS IN 2K PAGE BOUNDARIES.     */
  /*       THE IBM 3745 USES 4K PAGE BOUNDARIES, WHICH ARE             */
  /*       ACHIEVED BY NOT CODING ALIGN2.                              */
  /*********************************************************************/
  IF MODEL = "3705" | MODEL = "3720" | MODEL = "3725" THEN
    'LKED NCPINCL (MAP NCAL NOTERM LET LIST ALIGN2 SIZE 2300K'
  ELSE
    'LKED NCPINCL (MAP NCAL NOTERM LET LIST SIZE 2300K'
  RCODE=RC                                                    /*@A550438*/
  'ESTATE OBJ LOADLIB A'      /* IF TXTLIB/LOADLIB EXISTS,  @A270000*/
  IF RC = 0 THEN              /* ERASE IT                    @A270000*/
    'ERASE OBJ LOADLIB A'                                     /*@A270000*/
  EXIT (RCODE)                                                /*@A550438*/
CHKSUM:
```

*Figure 16 (Part 13 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

```
/*****************************************************************/
/* CHECK SUM OF ICNRTNDF PARAMETERS AGAINST LIMIT                */
/*****************************************************************/
NUMOPTS=NUMOPTS+1                    /* INCREMENT COUNTER        */
IF NUMOPTS > 8 THEN
  DO
    SAY " "
    SAY TEMP" INVALID, THE MAXIMUM NUMBER OF ICNRTNDF OPTIONS"
    SAY "(8) HAVE ALREADY BEEN SPECIFIED;  MOVE EXTRA OPTIONS"
    SAY "TO OPTIONS STATEMENT IN NCP GENERATION DEFINITION."
  END
RETURN
```

*Figure 16 (Part 14 of 14). Example of an NCP or PEP generation with output written to disk (VM)*

# Example of an NCP or PEP Generation with Output Written to Tape

When running a standard NCP or PEP generation, you supply your generation definition as input and specify the various input and output files in your EXEC. You can specify that input and output files be written to disk or tape. If you detect an error while generating or running your NCP, you can write certain listing files to tape. Figure 17 on page 101 shows the EXEC for generating an NCP load module for an IBM 3725 Communication Controller with listing files from the table 1 assembly, the table 2 assembly, and the link-edit written to tape.

When reading this example, remember the following differences among the communication controllers:

- The EXEC is slightly different.
- The NCP definition statement library used for NDF may be different.
- The FILEDEF for the library of preassembled NCP object modules in the link-edit step may be different.

**IBM 3705, 3720, or 3725 Communication Controller:** When running the link-edit step for a standard NCP or PEP generation on the IBM 3705, 3720, or 3725 Communication Controller, specify the ALIGN2 option in the EXEC for the link-edit step. ALIGN2 ensures that certain control sections within the load module are aligned on 2KB page boundaries. If you do not specify ALIGN2, the default is alignment on 4KB page boundaries, which may use excessive controller storage.

**IBM 3745 Communication Controller:** *Do not specify* ALIGN2. The default is alignment on 4KB page boundaries, the correct alignment for the IBM 3745 Communication Controller.

The following list shows when to specify ALIGN2 as the link-edit step:

| Communication Controller | ALIGN2 |
|---|---|
| 3705 | Yes |
| 3720 | Yes |
| 3725 | Yes |
| 3745 | No. |

The following is an example of an NCP or PEP generation with output written to tape.

```
/*********************************************************************/
;
/* EXAMPLE OF AN EXEC TO RUN A NCP/PEP GENERATION WITH SOME OUTPUT  */
/* FILES WRITTEN TO TAPE                                            */
;
/* COPYRIGHT=NONE                                                   */
;
/* THIS SAMPLE CAN BE USED TO GENERATE AN NCP WITH IBM SPECIAL      */
/* PRODUCTS IF THE MACROS AND OBJECT MODULES FOR THOSE PRODUCTS     */
/* HAVE BEEN MERGED INTO THE APPROPRIATE NCP LIBRARIES.             */
;
/* THE TABLE 1 LISTING AND THE TABLE 2 LISTING ARE WRITTEN TO TAPE  */
/* AND ONLY COPIED TO DISK WHEN AN ERROR IS DETECTED DURING THE     */
/* TABLE ASSEMBLIES.                                                */
;
;
/* YOU MUST SUPPLY THE FILENAME AND FILETYPE OF THE INPUT GENERATION*/
/* DEFINITION ON THE COMMAND LINE WHEN INVOKING THE EXEC.           */
/* YOU MUST ALSO SUPPLY THE MODEL NUMBER OF THE CONTROLLER.         */
/*                                                                  */
/* THE FOLLOWING PARAMETERS ARE REQUIRED:                           */
/*   1. FN - FILENAME OF YOUR INPUT GENERATION.                     */
/*   2. FT - FILETYPE OF YOUR INPUT GENERATION.                     */
/*   3. M  - MODEL NUMBER OF THE CONTROLLER.                        */
/*                                                                  */
/* THE FOLLOWING PARAMETERS ARE OPTIONAL:                           */
/*   1. FM - FILEMODE OF YOUR INPUT GENERATION (DEFAULTS TO '*').   */
/*   2. V - VERSION OF THE GENERATION (DEFAULTS TO 'V3', OR         */
/*         TO TVERSION IF SPECIFIED).                               */
/*   3. T - SPECIFIES WHETHER TEST NCP LIBRARIES ARE IN USE         */
/*         (DEFAULTS TO 'NO').                                      */
/*   4. LINECNT - SPECIFIES NUMBER OF LINES PER PAGE OF THE         */
/*          GENERATION VALIDATION LISTING AND THE TABLE ASSEMBLY    */
/*          LISTING (DEFAULTS TO 60).                               */
/*   5. FASTRUN - SPECIFY "FASTRUN=ON" TO MAKE NDF DO KEYWORD       */
/*          VALIDATION ONLY.                                        */
/*   6. ASSEMBLY - SPECIFY "ASSEMBLY=YES" IF YOU ARE INVOKING NDF   */
/*          FOR THE SOLE PURPOSE OF ACCESSING THE NDF CONTROLLER    */
/*          ASSEMBLER TO ASSEMBLE TABLE SOURCE CODE.                */
/*   7. TVERSION - SPECIFIES A TARGET VERSION FOR THE MIGRATION AID.*/
/*          IF TVERSION IS SPECIFIED, TMODEL AND TUSGTIER MUST      */
/*          ALSO BE SPECIFIED.                                      */
/*   8. TMODEL - SPECIFIES A TARGET MODEL FOR THE MIGRATION AID.    */
/*          IF TMODEL IS SPECIFIED, TVERSION AND TUSGTIER MUST      */
/*          ALSO BE SPECIFIED.                                      */
/*   9. TUSGTIER - SPECIFIES A TARGET USAGE TIER FOR THE MIGRATION  */
/*          AID.  IF TUSGTIER IS SPECIFIED, TVERSION AND TMODEL     */
/*          MUST ALSO BE SPECIFIED.                                 */
```

*Figure 17 (Part 1 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/*  10. CHANNELS - SPECIFIES A VALUE FOR THE MIGRATION AID          */
/*          "CHANNELS" PARAMETER (DEFAULTS TO LOCATION OF CHANNEL  */
/*          DEFINITIONS IN GENERATION DEFINITION FROM WHICH YOU    */
/*          ARE MIGRATING).                                        */
/*  11. DPU- SPECIFIES A VALUE FOR THE MIGRATION AID "DPU"          */
/*          PARAMETER (DEFAULTS TO 'YES').                         */
/*  12. SAVEADDR - SPECIFIES A VALUE FOR THE MIGRATION AID          */
/*          "SAVEADDR" PARAMETER (DEFAULTS TO "YES" WHEN TMODEL    */
/*          AND MODEL ON BUILD STATEMENT ARE THE SAME; DEFAULTS TO */
/*          "NO" WHEN THEY DIFFER).                                */
/*  13. REMOVCOM - SPECIFIES A VALUE FOR THE "REMOVCOM" PARAMETER   */
/*          (DEFAULTS TO 'NO').                                    */
;
/* CORRECT FORM FOR INVOKING THE EXEC:                             */
/*   VMNCP FN=GEN_FN,FT=GEN_FT,FM=GEN_FM,V=VERSION,M=MODEL,        */
/*     T=YES|NO,LINECNT=NUM_LINES,FASTRUN=ON,ASSEMBLY=YES,         */
/*     TVERSION=TARGET_VERSION,TMODEL=TARGET_MODEL,                */
/*     TUSGTIER=TARGET_USAGE_TIER,CHANNELS=BUILD|GROUP,DPU=YES|NO, */
/*     SAVEADDR=YES|NO,REMOVCOM=YES|NO                             */
/*                                                                 */
/*       -ALL THE POSSIBLE PARAMETERS HAVE BEEN SPECIFIED IN THE   */
/*        ABOVE EXAMPLE, FOR THE SAKE OF ILLUSTRATION.  IN         */
/*        PRACTICE, A SUBSET OF THE PARAMETERS WOULD BE CODED.     */
/*       -GEN_FN, GEN_FT, GEN_FM, VERSION, AND MODEL ARE VARIABLES */
/*        THAT YOU SUPPLY ACCORDING TO YOUR GENERATION             */
/*       -YOU MAY CODE 'T=YES' FOR A RUN ON TEST NCP LIBRARIES     */
/*       -ORDER OF PARAMETERS IS NOT IMPORTANT                     */
/*       -SPACES MAY BE USED INSTEAD OF COMMAS                     */
/*       -FOR NCP SUBSET, CODE V=V4S                               */
;
;
/* THE ASSIGNMENT OF THE MACRO AND OBJECT LIBRARY NAMES IS DERIVED */
/* FROM THE PARAMETERS PASSED ON THE COMMAND LINE; THEY MAY RESIDE */
/* ON ANY ACCESSED DISK.                                           */
/*                                                                 */
/* VARIABLES ARE DEFINED AS FOLLOWS:                               */
/*      MACRO = MACLIB NAME                                        */
/*      OBJECT = OBJLIB NAME                                       */
;
```

*Figure 17 (Part 2 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/*********************************************************************/
/* THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT                  */
/* LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOUR  */
/* LIBRARY NAMES DO NOT AGREE WITH THIS TABLE, YOU WILL NEED TO      */
/* SUBSTITUTE YOUR LIBRARY NAME FOR THE STANDARD LIBRARY NAME WHERE  */
/* APPROPRIATE.  YOU CAN USE THE "ALL" COMMAND ON THE FOLLOWING      */
/* STRINGS TO FIND LINES TO CHANGE FOR A PARTICULAR VERSION & MODEL.*/
/*                                                                   */
/*     ALLTAG1 - TEST                                                */
/*     ALLTAG2 - V3 & 3705                                           */
/*     ALLTAG3 - V3 & 3725                                           */
/*     ALLTAG4 - V4R2 & 3725 OR 3720                                 */
/*     ALLTAG5 - V4 SUBSET & 3720                                    */
/*     ALLTAG6 - V4R3 & 3725                                         */
/*     ALLTAG7 - V5R2 & 3720 OR 3745                                 */
/*     ALLTAG8 - V4R3.1 & 3725                                       */
/*     ALLTAG9 - V5R2.1 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR    */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*     ALLTAGA - V5R3 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR      */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*     ALLTAGB - V5R4 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR      */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*                                                                   */
/* 'T=YES' IS ALLOWED FOR TESTING TO FACILITATE LIBRARY MAINTENANCE.*/
/* (YOU MUST GO TO THE APPROPRIATE PLACE AND KEY IN YOUR TEST-LIB    */
/*  NAMES TO USE T=YES.)                                             */
/* T=YES WILL RUN WITH ANY MODEL AND VERSION (YOU MUST CODE MODEL). */
/*                                                                   */
```

*Figure 17 (Part 3 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/*****************************************************************/
/*                                                               */
/*                              M O D E L                        */
/*                                                               */
/*              3705        3725        3720        3745         */
/*            ------------------------------------------------   */
/*    V3/3705| MAC3705   | NOT       | NOT       | NOT       |  */
/*           | OBJ3705   | SUPPORTED | SUPPORTED | SUPPORTED |  */
/*           |-------------------------------------------------|  */
/*    V3/3725| NOT       | MAC3725   | NOT       | NOT       |  */
/*  V        | SUPPORTED | OBJ3725   | SUPPORTED | SUPPORTED |  */
/*  E        |-------------------------------------------------|  */
/*  R V4R1   | NOT       | NOT       | NOT       | NOT       |  */
/*  S        | SUPPORTED | SUPPORTED | SUPPORTED | SUPPORTED |  */
/*  I        |-------------------------------------------------|  */
/*  O V4R2   | NOT       | MAC3725   | MAC3725   | NOT       |  */
/*  N        | SUPPORTED | OBJ3725   | OBJ3725   | SUPPORTED |  */
/*           |-------------------------------------------------|  */
/*    V4     | NOT       | NOT       | MAC3725   | NOT       |  */
/*    SUBSET | SUPPORTED | SUPPORTED | OBJ3725   | SUPPORTED |  */
/*           |-------------------------------------------------|  */
/*    V4R3 & | NOT       | SNCPMAC1  | NOT       | NOT       |  */
/*    LATER  | SUPPORTED | SNCPMOD1  | SUPPORTED | SUPPORTED |  */
/*           |-------------------------------------------------|  */
/*    V5R1   | NOT       | NOT       | MAC3725   | MAC3725   |  */
/*           | SUPPORTED | SUPPORTED | OBJ3725   | OBJ3725   |  */
/*           |-------------------------------------------------|  */
/*    V5R2 & | NOT       | NOT       | SNCPMAC1  | SNCPMAC1  |  */
/*    LATER  | SUPPORTED | SUPPORTED | SNCPMOD1  | SNCPMOD1  |  */
/*            ------------------------------------------------   */
/*                                                               */
/*                              M O D E L                        */
/*                                                               */
/*           3745-130, 3745-150,                                 */
/*           3745-170, 3745-210,                                 */
/*           3745-410                                            */
/*           -------------                                       */
/*    V3     | NOT       |                                      */
/*           | SUPPORTED |                                      */
/*           |-------------                                      */
/*    V4     | NOT       |                                      */
/*  V        | SUPPORTED |                                      */
/*  E        |-------------                                      */
/*  R V5R2 & | NOT       |                                      */
/*  S BEFORE | SUPPORTED |                                      */
/*  I        |-------------                                      */
/*  O V5R2.1 | SNCPMAC1  |                                      */
/*  N & LATER| SNCPMOD1  |                                      */
/*           -------------                                       */
/*                                                               */
/*                                                               */
```

*Figure 17 (Part 4 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/*********************************************************************/
/* CHANGE ACTIVITY                                                   */
/*    - BUILD CMS LOADLIB FROM TABLE1 & 2 TEXT FILES IF     @A550438*/
/*      TXTLIB COMMAND FAILS                                @A550438*/
/*                                                                   */
/*********************************************************************/
;
  ADDRESS COMMAND                     /* ENSURE CP/CMS ENVIRONMENT  */
  TRACE N
  GEN_FN=""
  GEN_FT=""                           /* INITIALIZE STRING VARIABLES*/
  GEN_FM=""
  VERSION=""
  TVERSION=""
  TMODEL=""
  MODEL=""
  T="NO"
  ARG REST                            /* GET PARAMETERS FROM COMMAND*/
                                      /* LINE                       */
  REST=TRANSLATE(REST,' ',',')        /* GET RID OF COMMAS          */
  COUNT=WORDS(REST)
  LPCNT=1
  OPTIONS="("
  FSTRUN="FALSE"
  NUMOPTS=0
  DO WHILE LPCNT<=COUNT               /* LOOP THROUGH ONCE FOR EACH */
    TEMP=WORD(REST,LPCNT)             /* WORD IN THE STRING         */
    PARSE VALUE TEMP WITH FRONT '=' BACK
    SELECT
      WHEN (ABBREV(TEMP,'FN')) THEN
        GEN_FN=BACK
      WHEN (ABBREV(TEMP,'FT')) THEN   /* SET APPROPRIATE VARIABLE   */
        GEN_FT=BACK                   /* ACCORDING TO THE ASSIGNMENT*/
      WHEN (ABBREV(TEMP,'FM')) THEN   /* MADE ON THE COMMAND LINE   */
        GEN_FM=BACK
      WHEN (ABBREV(TEMP,'V')) THEN
        VERSION=BACK
      WHEN (ABBREV(TEMP,'M')) THEN
        MODEL=BACK
      WHEN (ABBREV(TEMP,'LINECNT')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
```

*Figure 17 (Part 5 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
WHEN (ABBREV(TEMP,'FASTRUN')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    FSTRUN="TRUE"
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'ASSEMBLY')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'TVERSION')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
    TVERSION=BACK
  END
WHEN (ABBREV(TEMP,'TMODEL')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    TMODEL=BACK
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'TUSGTIER')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'CHANNELS')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'DPU')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
  END
WHEN (ABBREV(TEMP,'SAVEADDR')) THEN
  DO
    OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
    CALL CHKSUM
  END
```

*Figure 17 (Part 6 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
      WHEN (ABBREV(TEMP,'REMOVCOM')) THEN
        DO
          OPTIONS=OPTIONS  || " " || FRONT || "(" || BACK || ")"
          CALL CHKSUM
        END
      WHEN (ABBREV(TEMP,'T')) THEN
        T=BACK
      OTHERWISE
        SAY TEMP" IS NOT VALID, IGNORED"
    END                                 /* END SELECT                */
    LPCNT=LPCNT+1
  END                                   /* END DO                    */
  IF GEN_FM="" THEN                     /* DEFAULT FILETYPE TO "*" IF */
    GEN_FM="*"                          /* NOT CODED                 */
;
  IF NUMOPTS > 8 THEN
    EXIT
;
;
/* SEE IF GEN_FN, GEN_FT, AND MODEL WERE PASSED ON COMMAND LINE.    */
/* IF GEN NAME OR MODEL WERE NOT SPECIFIED GIVE CORRECT FORM & EXIT.*/
;
  IF (GEN_FN="") THEN
    DO
      SAY "FN PARAMETER MISSING; SPECIFY AS FN=GEN_FN"
     EXIT
    END
  IF (GEN_FT="") THEN
    DO
      SAY "FT PARAMETER MISSING; SPECIFY AS FT=GEN_FT"
      EXIT
    END
  IF (MODEL="" & FSTRUN="FALSE")
  THEN
    IF TMODEL="" THEN
      DO
        SAY "M PARAMETER MISSING; SPECIFY AS M=MODEL"
        EXIT
      END
    ELSE
      DO
        MODEL=TMODEL
        SAY "DEFAULTING TO M="MODEL
      END
;
;
  'ESTATE' GEN_FN GEN_FT GEN_FM       /* SEE IF GEN EXISTS ON DISK  */
  IF RC ¬= 0 THEN
    DO
      SAY GEN_FN GEN_FT GEN_FM "DOES NOT EXIST"
      EXIT RC                         /* EXIT IF GEN DOESN'T EXIST  */
    END
;
;
```

*Figure 17 (Part 7 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/* THIS STRUCTURE VALIDATES VERSION AND MODEL AS TAKEN FROM THE     */
/* COMMAND LINE AND SETS "MACRO" AND "OBJECT" ACCORDINGLY.          */
/*                                                                  */
/* NOTE:  IF YOU HAVE CHANGED A LIBRARY NAME, OR YOUR LIBRARY NAMES */
/* DO NOT AGREE WITH THE TABLE ABOVE, CHANGE THE ASSIGNMENT         */
/* STATEMENTS OF "MACRO" AND "OBJECT" TO REFLECT YOUR LIBRARY NAMES */
/* IN THE APPROPRIATE PLACE IN THIS STRUCTURE (ACCORDING TO YOUR    */
/* VERSION AND MODEL).                                              */
;
;
  IF FSTRUN="FALSE"                  /* FASTRUN NOT SPECIFIED      */
  THEN
  DO
  IF TVERSION¬=""                    /* TVERSION SPECIFIED         */
  THEN
    IF VERSION="" THEN               /* VERSION NOT SPECIFIED      */
      DO
        SAY "DEFAULTING TO V="TVERSION
        VERSION=TVERSION
      END
    ELSE                             /* VERSION SPECIFIED          */
      IF VERSION ¬= TVERSION
      THEN
        DO
          SAY "V="VERSION "INVALID - WHEN TVERSION IS SPECIFIED,"
          SAY "V AND TVERSION MUST BE EQUAL"
          EXIT
        END
  IF (VERSION="") THEN               /* VERSION DEFAULTS TO V3     */
    DO                               /* IF NOT CODED               */
      SAY "DEFAULTING TO V=V3"
      VERSION="V3"
    END
  SELECT
    WHEN (T="YES") THEN
      DO
        MACRO=MACXXXX                /* FOR TEST           ALLTAG1*/
        OBJECT=OBJXXXX               /*                    ALLTAG1*/
        DDNAME=OBJXXXX               /*                    ALLTAG1*/
      END
```

*Figure 17 (Part 8 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
WHEN (VERSION="V3")&(MODEL="3705") THEN
  DO
    MACRO=MAC3705                 /* FOR V3 & 3705       ALLTAG2*/
    OBJECT=OBJ3705                /*                     ALLTAG2*/
    DDNAME=OBJ3705                /*                     ALLTAG2*/
  END
WHEN (VERSION="V3")&(MODEL="3725") THEN
  DO
    MACRO=MAC3725                 /* FOR V3 & 3725       ALLTAG3*/
    OBJECT=OBJ3725                /*                     ALLTAG3*/
    DDNAME=OBJ3725                /*                     ALLTAG3*/
  END
WHEN (VERSION="V4R2")&((MODEL="3725")|(MODEL="3720")) THEN
  DO
    MACRO=MAC3725                 /* FOR V4R2&3725|3720  ALLTAG4*/
    OBJECT=OBJ3725                /*                     ALLTAG4*/
    DDNAME=OBJ3725                /*                     ALLTAG4*/
  END
WHEN (VERSION="V4S")&(MODEL="3720") THEN
  DO
    MACRO=MAC3725                 /* FOR V4 SUBSET       ALLTAG5*/
    OBJECT=OBJ3725                /* & 3720              ALLTAG5*/
    DDNAME=OBJ3725                /* & 3720              ALLTAG5*/
  END
WHEN (VERSION="V4R3")&(MODEL="3725") THEN
  DO
    MACRO=SNCPMAC1                /* FOR V4R3            ALLTAG6*/
    OBJECT=SNCPMOD1               /* & 3725              ALLTAG6*/
    DDNAME=ANCPMOD1               /*                     ALLTAG6*/
  END
WHEN (VERSION="V5R2")&((MODEL="3720")|(MODEL="3745")) THEN
  DO
    MACRO=SNCPMAC1                /* FOR V5R2            ALLTAG7*/
    OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAG7*/
    DDNAME=ANCPMOD1               /*                     ALLTAG7*/
  END
WHEN (VERSION="V4R3.1")&(MODEL="3725") THEN
  DO
    MACRO=SNCPMAC1                /* FOR V4R3.1          ALLTAG8*/
    OBJECT=SNCPMOD1               /* & 3725              ALLTAG8*/
    DDNAME=ANCPMOD1               /*                     ALLTAG8*/
  END
WHEN (VERSION="V5R2.1")&((MODEL="3720")|(MODEL="3745")|,
  (MODEL="3745-130")|,
  (MODEL="3745-150")|,
  (MODEL="3745-170")|,
  (MODEL="3745-210")|,
  (MODEL="3745-410")) THEN
  DO
    MACRO=SNCPMAC1                /* FOR V5R2.1          ALLTAG9*/
    OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAG9*/
    DDNAME=ANCPMOD1               /*                     ALLTAG9*/
  END
```

*Figure 17 (Part 9 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
    WHEN (VERSION="V5R3")&((MODEL="3720")|(MODEL="3745")|,
      (MODEL="3745-130")|,
      (MODEL="3745-150")|,
      (MODEL="3745-170")|,
      (MODEL="3745-210")|,
      (MODEL="3745-410")) THEN
      DO
        MACRO=SNCPMAC1                /* FOR V5R3            ALLTAGA*/
        OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAGA*/
        DDNAME=ANCPMOD1               /*                     ALLTAGA*/
      END
    WHEN (VERSION="V5R4")&((MODEL="3720")|(MODEL="3745")|,
      (MODEL="3745-130")|,
      (MODEL="3745-150")|,
      (MODEL="3745-170")|,
      (MODEL="3745-210")|,
      (MODEL="3745-410")) THEN
      DO
        MACRO=SNCPMAC1                /* FOR V5R4            ALLTAGB*/
        OBJECT=SNCPMOD1               /* & 3720 OR 3745      ALLTAGB*/
        DDNAME=ANCPMOD1               /*                     ALLTAGB*/
      END
    OTHERWISE
      DO
        SAY "VERSION = "VERSION" NOT VALID WITH MODEL = "MODEL
        SAY "WHEN RUNNING UNDER VM"
        EXIT
      END
  END                                 /* END SELECT                */
  END                                 /* FASTRUN NOT SPECIFIED     */
;
;
  IF FSTRUN="FALSE" THEN
    DO
      'ESTATE' MACRO 'MACLIB *'       /* SEE IF MACLIB EXISTS      */
      IF RC ¬= 0 THEN
        SAY "ERROR IN ACCESSING" MACRO "MACLIB"
;
;
      'TAPE REW'
      IF RC ¬= 0 THEN
        DO
          SAY "ERROR IN ACCESSING TAPE"
          EXIT RC
        END
      'TAPE WVOL1 TAPE1'
    END
```

*Figure 17 (Part 10 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/* CLEAR OLD FILE DEFINITIONS                                          */
  'FILEDEF * CLEAR'
/* WORKING SPILL FILE                                                  */
/* THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH VIRTUAL        */
/* MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF USERGEN IS SPECIFIED.*/
/*'FILEDEF DBWORKFL DISK  DBWORKFL FILE A ( XTENT 40'                  */
;
/* IF NEWDEFN=YES IS SPECIFIED OR DEFAULTED IN THE                     */
/* GENERATION DEFINITION, A FILE DEFINITION SIMILAR TO THE             */
/* FOLLOWING IS NEEDED.                                                */
/*'FILEDEF NEWDEFN DISK NEWDEFN FILE A'                                */
;
/* MACRO LIBRARIES USED IN THE TABLE ASSEMBLY PHASE OF NDF             */
  IF FSTRUN="FALSE" THEN
    DO
      'FILEDEF SYSLIB DISK' MACRO 'MACLIB *'
      'GLOBAL MACLIB' MACRO
    END
/* INPUT FILE WITH NCP/EP GENERATION STATEMENTS                        */
  'FILEDEF GENDECK DISK' GEN_FN GEN_FT GEN_FM
/* GENERATION VALIDATION STEP OUTPUT                                   */
  'FILEDEF SYSPRINT DISK' GEN_FN 'LISTING A'
/* NDF SUMMARY LISTING                                                 */
  'FILEDEF PRINTER TERM'
/* SOURCE FOR TABLE 1 ASSEMBLY - OUTPUT FROM GENERATION VALIDATION     */
  'FILEDEF TBL1SRCE DISK TABLE1 SOURCE A'
/* LISTING FROM THE TABLE 1 ASSEMBLY                                   */
  IF FSTRUN="FALSE" THEN
    'FILEDEF TBL1LIST TAP1 SL  ( BLKSIZE 7260 LRECL 121 RECFM FB'
/* TEXT OUTPUT FROM THE TABLE 1 ASSEMBLY                               */
  'FILEDEF TBL1OBJ DISK TABLE1 TEXT A'
/* SOURCE FOR TABLE 2 ASSEMBLY - OUTPUT FROM GENERATION VALIDATION     */
  'FILEDEF TBL2SRCE DISK TABLE2 SOURCE A'
/* LISTING FROM THE TABLE 2 ASSEMBLY                                   */
  IF FSTRUN="FALSE" THEN
    'FILEDEF TBL2LIST TAP1 SL (LEAVE BLKSIZE 7260 LRECL 121 RECFM FB'
/* TEXT OUTPUT FROM THE TABLE 2 ASSEMBLY                               */
  'FILEDEF TBL2OBJ DISK TABLE2 TEXT A'
/* LINK EDIT STATEMENTS OUTPUT FROM THE GENERATION VALIDATION STEP     */
  'FILEDEF LNKSTMT DISK NCPINCL TEXT A'
/* TEMPORARY WORK FILE USED BY THE TABLE ASSEMBLIES                    */
  'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4'
/* RUN THE NDF STEP                                                    */
  'ICNRTNDF' OPTIONS
/* EXIT BECAUSE OF AN ERROR DURING GENERATION VALIDATION               */
```

*Figure 17 (Part 11 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
IF RC ¬= 0 & RC¬=10 & RC¬=100
THEN
  DO
    SAY "***ERROR IN EXECUTING NDF***"
    EXIT RC
  END
IF FSTRUN="TRUE" THEN
  EXIT RC
SELECT
  WHEN (RC = 0) THEN
    DO
      SET CMSTYPE HT                                    /*@A550438*/
          'FILEDEF SYSLIB CLEAR'                        /*@A550438*/
          /*********************************************************/
          /* BUILD THE TABLE 1 LOADLIB                             */
          /*********************************************************/
          'FILEDEF SYSUT1 DISK SYSUT1 TEMP A (BLKSIZE 8192'
          'FILEDEF SYSLMOD DISK ICNTABL1 LOADLIB A (BLKSIZE 8192'
          'FILEDEF TABLE1 DISK TABLE1 TEXT A'           /*@A550438*/
          LINE=' INCLUDE TABLE1'                        /*@A550438*/
          'EXECIO 1 DISKW' ICNTABL1 TEXT A 1 F '(VAR LINE'
                                                        /*@A550438*/
          FINIS ICNTABL1 TEXT A                         /*@A550438*/
          'LKED ICNTABL1 (NCAL LET NOTERM SIZE 2300K'
          /*********************************************************/
          /* BUILD THE TABLE 2 LOADLIB                             */
          /*********************************************************/
          'FILEDEF SYSUT1 DISK SYSUT1 TEMP A (BLKSIZE 8192'
          'FILEDEF SYSLMOD DISK ICNTABL2 LOADLIB A (BLKSIZE 8192'
          'FILEDEF TABLE2 DISK TABLE2 TEXT A'           /*@A550438*/
          LINE=' INCLUDE TABLE2'                        /*@A550438*/
          'EXECIO 1 DISKW' ICNTABL2 TEXT A 1 F '(VAR LINE'
                                                        /*@A550438*/
          FINIS ICNTABL2 TEXT A                         /*@A550438*/
          'LKED ICNTABL2 (NCAL LET NOTERM SIZE 2300K'
          /*********************************************************/
          /* BUILD THE LINKAGE EDITOR SYSIN CONTROL STATEMENT      */
          /*********************************************************/
          LINE=' '                                      /*@A550438*/
          'EXECIO 1 DISKW' NDFSYSIN FILE A 1 F '(VAR LINE'
                                                        /*@A550438*/
          FINIS NDFSYSIN FILE A                         /*@A550438*/
```

*Figure 17 (Part 12 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
        /*********************************************************/
        /*COPY THE TABLE 2 LOADLIB INTO THE TABLE 1 LOADLIB      */
        /*********************************************************/
        'LOADLIB COPY ICNTABL2 LOADLIB A ICNTABL1 LOADLIB A
          NDFSYSIN FILE A (MODIFY'                      /*@A550438*/
        SET CMSTYPE RT                                  /*@A550438*/
        'COPY ICNTABL1 LOADLIB A OBJ LOADLIB A (REP'    /*@A270000*/
        'ERASE ICNTABL1 TEXT A'                         /*@A550438*/
        'ERASE ICNTABL1 LKEDIT A'                       /*@A550438*/
        'ERASE ICNTABL2 TEXT A'                         /*@A550438*/
        'ERASE ICNTABL2 LKEDIT A'                       /*@A550438*/
        'ERASE ICNTABL1 LOADLIB A'                      /*@A270000*/
        'ERASE ICNTABL2 LOADLIB A'                      /*@A550438*/
        'ERASE NDFSYSIN FILE A'                         /*@A550438*/
        'FILEDEF SYSPUNCH DISK OBJ LOADLIB A (RECFM U'/*@A550438*/
      /*********************************************************/
      /* ERASE TEMPORARY FILES                                 */
      /*********************************************************/
      'ERASE SYSUT1 TEMP A'
      'ERASE TABLE1 SOURCE'
      'ERASE TABLE2 SOURCE'
      'ERASE TABLE1 TEXT'
      'ERASE TABLE2 TEXT'
;
      'ESTATE' OBJECT 'TXTLIB *'     /* SEE IF OBJLIB EXISTS     */
      IF RC ¬= 0 THEN
        DO                                              /*@A550438*/
          SAY "ERROR IN ACCESSING" OBJECT "TXTLIB"
          EXIT(RC)                                      /*@A550438*/
        END                                             /*@A550438*/
;
      /*********************************************************/
      /* FILEDEFS FOR THE LINK EDIT STEP                       */
      /*********************************************************/
      'FILEDEF SYSUT1 CLEAR'
      'FILEDEF SYSUT1 DISK SYSUT1 TEMP A (BLKSIZE 8192'
      'FILEDEF' DDNAME 'DISK' OBJECT 'TXTLIB *'
;
      /*********************************************************/
      /* NAME OF OUTPUT LIBRARY FOR THE LOAD MODULE            */
      /*********************************************************/
      'FILEDEF SYSLMOD DISK' GEN_FN 'LOADLIB A (BLKSIZE 8192'
;
```

*Figure 17 (Part 13 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
        /*************************************************************/
        /* RUN LINKAGE EDITOR                                        */
        /* NOTE:                                                     */
        /*   THE ALIGN2 PARAMETER IS CODED ONLY FOR THE IBM          */
        /*   3705/3720/3725, SINCE IT RESULTS IN 2K PAGE             */
        /*   BOUNDARIES.  THE IBM 3745 USES 4K PAGE BOUNDARIES,      */
        /*   WHICH ARE ACHIEVED BY NOT CODING ALIGN2                 */
        /*************************************************************/
        IF MODEL = "3705" | MODEL = "3720" | MODEL = "3725" THEN
         'LKED NCPINCL (MAP NCAL NOTERM LET LIST ALIGN2
            SIZE 2300K'
        ELSE
         'LKED NCPINCL (MAP NCAL NOTERM LET LIST SIZE 2300K'
        RCODE=RC                                          /*@A550438*/
        'ESTATE OBJ LOADLIB A'
        IF RC = 0 THEN                                    /*@A270000*/
          'ERASE OBJ LOADLIB A'
        EXIT (RCODE)                                      /*@A550438*/
      END
;
/*********************************************************************/
/* FOR TABLE 1 ERROR COPY TABLE 1 LISTING FROM TAPE                  */
/*********************************************************************/
    WHEN (RC = 10) THEN
      DO
        'FILEDEF TBL1LIST CLEAR'
        'FILEDEF TBL1LIST TAP1 SL 1 ( BLKSIZE 7260 LRECL 121 RECFM FB'
        'FILEDEF OUTFILE DISK TABLE1 LISTING A (LRECL 121'
        'MOVEFILE TBL1LIST OUTFILE'
        EXIT 10
      END
;
/*********************************************************************/
/* FOR TABLE 2 ERROR COPY TABLE 2 LISTING FROM TAPE                  */
/*********************************************************************/
    WHEN (RC = 100) THEN
      DO
        'FILEDEF TBL2LIST CLEAR'
        'FILEDEF TBL2LIST TAP1 SL 2 ( BLKSIZE 7260 LRECL 121 RECFM FB'
        'FILEDEF OUTFILE DISK TABLE2 LISTING A  (LRECL 121'
        'MOVEFILE TBL2LIST OUTFILE'
        EXIT 100
      END
;
    OTHERWISE;
  END                                 /* END SELECT              */
CHKSUM:
```

*Figure 17 (Part 14 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

```
/*****************************************************************/
/* CHECK SUM OF ICNRTNDF PARAMETERS AGAINST LIMIT                */
/*****************************************************************/
NUMOPTS=NUMOPTS+1                    /* INCREMENT COUNTER        */
IF NUMOPTS > 8 THEN
  DO
    SAY " "
    SAY TEMP" INVALID, THE MAXIMUM NUMBER OF ICNRTNDF OPTIONS"
    SAY "(8) HAVE ALREADY BEEN SPECIFIED;  MOVE EXTRA OPTIONS"
    SAY "TO OPTIONS STATEMENT IN NCP GENERATION DEFINITION."
  END
RETURN
```

*Figure 17 (Part 15 of 15). Example of an NCP or PEP generation with output written to tape (VM)*

# Example of an NCP or PEP Generation with User-Written Code Using the NDF Standard Attachment Facility

To run an NCP or PEP generation with IBM special products or user-written code using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing.

Figure 18 on page 116 shows the EXEC for generating user-written code and NCP using the NDF standard attachment facility. For more information about running this type of generation, see page 71. You can run this type of user-written code generation *only* if you are using SSP V3R2 or a later release to generate NCP V4 Subset, NCP V4R2 or a later release, or NCP V5R2 or a later release.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation load modules to be loaded in the generation. Each application must have its own generation load module. You can specify up to 25 generation load modules.
- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation load modules.
- Modify the EXEC for a standard NCP or PEP generation to include the FILEDEFs for the NEWDEFN file, the DBWORKFL file, and the libraries for user-supplied modules.

The following are examples of the GLOBAL and FILEDEF commands used to include user-written generation load modules in a standard NCP or PEP generation using the NDF standard attachment facility.

```
/* EXAMPLE OF GLOBAL COMMAND TO IDENTIFY THE LIBRARY CONTAINING     */
/* A USER WRITTEN GENERATION LOAD MODULE                            */
  'GLOBAL LOADLIB USERLIB'

/* EXAMPLE OF THE FILE DEFINITION FOR NEWDEFN                       */
  'FILEDEF NEWDEFN DISK' GEN_FN 'NEWDEFN' GEN_FM

/* EXAMPLE OF THE FILE DEFINITIONS FOR THE SYSLIB CHAIN FOR A       */
/* NCP or PEP GENERATION WITH USER CODE                             */
  'FILEDEF SYSLIB DISK' MACRO 'MACLIB *'
  'FILEDEF SYSLIB DISK USER1 MACLIB * (CONCAT'
  'FILEDEF SYSLIB DISK USER2 MACLIB * (CONCAT'
  'GLOBAL MACLIB' MACRO 'USER1 USER2'

/* EXAMPLE OF THE FILE DEFINITIONS FOR USER OBJECT LIBRARIES FOR    */
/* THE LINK EDIT OF AN NCP or PEP LOAD MODULE WITH USER CODE        */
  'FILEDEF USER1 DISK USER1 TXTLIB *'
       .         .         .
       .         .         .
       .         .         .
  'FILEDEF USER1 DISK USERN TXTLIB *'
```

*Figure 18. Example of an NCP or PEP generation with user-written code using the NDF standard attachment facility (VM)*

# Example of an NCP or PEP Generation with User-Written Code Using the GENEND Definition Statement

To run an NCP or PEP generation with IBM special products or user-written code without using the NDF standard attachment facility, or to generate user-written code using SSP Version 3, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of the link-edit statements by coding keywords on the GENEND definition statement.

Figure 19 on page 117 shows the EXEC for generating IBM special products or user-written code using the GENEND definition statement. For more information about running this type of generation, see page 72. You can run this type of user-written code generation if you are using SSP Version 3 to generate any version of NCP.

Before you generate IBM special products or user-written code using the GENEND definition statement, ensure that you:

- Assemble the user-written routines and code the link-edit statements for the routines
- Code the appropriate keywords on the GENEND definition statement for your user-written routines
- Place the members with SRCLO or SRCHI code in a definition statement library (SYSLIB) available to NDF
- Place all members that contain INCLUDE or ORDER link-edit control statements in a definition statement library in the NDF SYSLIB chain

- Place all definition statements in the NDF SYSLIB chain
- Modify the EXEC to include the SYSLIB chain and the ULIB or user object code library FILEDEF statement.

The generation validation phase of NDF reads the link-edit control statements and writes them into the same file as the standard NCP link-edit control statements.

The following are examples of GLOBAL and FILEDEF commands used to include the SYSLIB chain and the link-edit ULIB statement in a standard NCP or PEP generation.

**Note:** In these examples, library names such as SNCPMAC1 and SNCPMOD1 are release dependent.

```
/* EXAMPLE OF THE FILE DEFINITIONS FOR THE SYSLIB CHAIN FOR A       */
/* NCP or PEP GENERATION WITH USER CODE                             */
  'FILEDEF SYSLIB DISK' MACRO 'MACLIB *'
  'FILEDEF SYSLIB DISK USER1 MACLIB * (CONCAT'
  'FILEDEF SYSLIB DISK USER2 MACLIB * (CONCAT'
  'GLOBAL MACLIB' MACRO 'USER1 USER2'

/* EXAMPLE OF THE FILE DEFINITIONS FOR USER OBJECT LIBRARIES FOR    */
/* THE LINK EDIT OF AN NCP or PEP LOAD MODULE WITH USER CODE        */
/*                                                                  */
/* LIBRARY FOR BLOCK HANDLER AND USER-WRITTEN CODE MODULES          */
  'FILEDEF ULIB DISK USER1 TXTLIB *'
/* LIBRARIES FOR USER-WRITTEN CODE MODULES                          */
  'FILEDEF USER1 DISK USER1 TXTLIB *'
       .         .         .
       .         .         .
       .         .         .
  'FILEDEF USER1 DISK USERN TXTLIB *'
```

*Figure 19. Example of an NCP or PEP generation with user-written code using the GENEND definition statement (VM)*

# Example of a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, you can use the text file from a dynamic reconfiguration generation. Figure 20 on page 118 shows the EXEC for dynamic reconfiguration generation.

To use this type of generation, ensure that you coded the original NCP to allow dynamic reconfiguration. The dynamic reconfiguration generation produces a text file that the access method can use to modify NCP.

**Note:** VTAM has its own dynamic reconfiguration procedures that do not require you to use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Network Implementation Guide*.

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration file consisting of ADD statements or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration file is the input for the dynamic reconfiguration generation. This type of generation produces a text file that the access method uses to modify an NCP already running in a com-

munication controller. For information on using ADD, DELETE, PU, or LU definition statements, see *NCP, SSP, and EP Resource Definition Guide*.

A dynamic reconfiguration generation requires one table assembly and no link-edit. The following is an example of an EXEC for a dynamic reconfiguration generation.

```
/*********************************************************************/
;
/* EXAMPLE OF AN EXEC TO RUN A DYNAMIC RECONFIGURATION GENERATION   */
;
/* COPYRIGHT=NONE                                                    */
;
/* YOU MUST SUPPLY THE FILENAME AND FILETYPE OF THE INPUT GENERATION*/
/* DEFINITION ON THE COMMAND LINE WHEN INVOKING THE EXEC.           */
/* YOU MUST ALSO SUPPLY THE MODEL NUMBER OF THE CONTROLLER.         */
/*                                                                  */
/* YOU CAN SPECIFY THE FILEMODE OF YOUR INPUT GENERATION            */
/* (WHICH DEFAULTS TO '*'), VERSION OF YOUR GENERATION (WHICH       */
/* DEFAULTS TO 'V3'), AND TEST (WHICH DEFAULTS TO 'T=NO').          */
/* THESE PARAMETERS ARE OPTIONAL.*/
;
/* CORRECT FORM FOR INVOKING THE EXEC:                              */
/*     VMDR  FN=GEN_FN,FT=GEN_FT,FM=GEN_FM,V=VERSION,M=MODEL,T=NO   */
/*       -GEN_FN, GEN_FT, GEN_FM, VERSION, AND MODEL ARE VARIABLES  */
/*        THAT YOU SUPPLY ACCORDING TO YOUR GENERATION              */
/*       -YOU MAY ALSO CODE 'T=YES' FOR A RUN ON TEST NCP LIBRARIES */
/*       -ORDER OF PARAMETERS IS NOT IMPORTANT                      */
/*       -SPACES MAY BE USED INSTEAD OF COMMAS                      */
/*       -FOR NCP SUBSET, CODE V=V4S                                */
;
;
/* THE ASSIGNMENT OF THE MACRO LIBRARY NAME IS DERIVED FROM THE     */
/* PARAMETERS PASSED ON THE COMMAND LINE; IT MAY RESIDE ON ANY      */
/* ACCESSED DISK.                                                   */
/*                                                                  */
/* THE MACRO LIBRARY NAME IS REPRESENTED BY THE VARIABLE "MACRO".   */
;
```

*Figure 20 (Part 1 of 7). Example of a dynamic reconfiguration generation (VM)*

```
/*********************************************************************/
/* THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT                  */
/* LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOUR */
/* LIBRARY NAMES DO NOT AGREE WITH THIS TABLE, YOU WILL NEED TO     */
/* SUBSTITUTE YOUR LIBRARY NAME FOR THE STANDARD LIBRARY NAME WHERE */
/* APPROPRIATE.  YOU CAN USE THE "ALL" COMMAND ON THE FOLLOWING     */
/* STRINGS TO FIND LINES TO CHANGE FOR A PARTICULAR VERSION & MODEL.*/
/*                                                                   */
/*     ALLTAG1 - TEST                                                */
/*     ALLTAG2 - V3 & 3705                                           */
/*     ALLTAG3 - V3 & 3725                                           */
/*     ALLTAG4 - V4R2 & 3725 OR 3720                                 */
/*     ALLTAG5 - V4 SUBSET & 3720                                    */
/*     ALLTAG6 - V4R3 & 3725                                         */
/*     ALLTAG7 - V5R2 & 3720 OR 3745                                 */
/*     ALLTAG8 - V4R3.1 & 3725                                       */
/*     ALLTAG9 - V5R2.1 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR    */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*     ALLTAGA - V5R3 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR      */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*     ALLTAGB - V5R4 & 3720 OR 3745 OR 3745-130 OR 3745-150 OR      */
/*                        3745-170 OR 3745-210 OR 3745-410           */
/*                                                                   */
/* 'T=YES' IS ALLOWED FOR TESTING TO FACILITATE LIBRARY MAINTENANCE.*/
/* (YOU MUST GO TO THE APPROPRIATE PLACE AND KEY IN YOUR TEST-LIB    */
/*  NAMES TO USE T=YES.)                                             */
/* T=YES WILL RUN WITH ANY MODEL AND VERSION (YOU MUST CODE MODEL). */
/*                                                                   */
```

*Figure 20 (Part 2 of 7). Example of a dynamic reconfiguration generation (VM)*

```
/*********************************************************************/
/*                                                                   */
/*                                 M O D E L                         */
/*                                                                   */
/*              3705          3725          3720          3745       */
/*            ------------------------------------------------       */
/*    V3/3705| MAC3705   |  NOT       |  NOT       |  NOT       |    */
/*           |            |  SUPPORTED |  SUPPORTED |  SUPPORTED |    */
/*           |------------------------------------------------|       */
/*    V3/3725| NOT        |  MAC3725   |  NOT       |  NOT       |    */
/*  V        |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |    */
/*  E        |------------------------------------------------|       */
/*  R V4R1   |  NOT       |  NOT       |  NOT       |  NOT       |    */
/*  S        |  SUPPORTED |  SUPPORTED |  SUPPORTED |  SUPPORTED |    */
/*  I        |------------------------------------------------|       */
/*  O V4R2   |  NOT       |  MAC3725   |  MAC3725   |  NOT       |    */
/*  N        |  SUPPORTED |            |            |  SUPPORTED |    */
/*           |------------------------------------------------|       */
/*    V4     |  NOT       |  NOT       |  MAC3725   |  NOT       |    */
/*    SUBSET |  SUPPORTED |  SUPPORTED |            |  SUPPORTED |    */
/*           |------------------------------------------------|       */
/*    V4R3 & |  NOT       |  SNCPMAC1  |  NOT       |  NOT       |    */
/*    LATER  |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |    */
/*           |------------------------------------------------|       */
/*    V5R1   |  NOT       |  NOT       |  MAC3725   |  MAC3725   |    */
/*           |  SUPPORTED |  SUPPORTED |  OBJ3725   |  OBJ3725   |    */
/*           |------------------------------------------------|       */
/*    V5R2 & |  NOT       |  NOT       |  SNCPMAC1  |  SNCPMAC1  |    */
/*    LATER  |  SUPPORTED |  SUPPORTED |            |            |    */
/*            ------------------------------------------------       */
/*                                                                   */
/*                                 M O D E L                         */
/*                                                                   */
/*          3745-130, 3745-150,                                      */
/*          3745-170, 3745-210,                                      */
/*          3745-410                                                 */
/*          -------------                                            */
/*    V3    |  NOT       |                                           */
/*          |  SUPPORTED |                                           */
/*          |------------                                            */
/*    V4    |  NOT       |                                           */
/*  V       |  SUPPORTED |                                           */
/*  E       |------------                                            */
/*  R V5R2 & |  NOT       |                                          */
/*  S BEFORE |  SUPPORTED |                                          */
/*  I        |------------                                          */
/*  O V5R2.1 |  SNCPMAC1  |                                          */
/*  N & LATER|  SNCPMOD1  |                                          */
/*           -------------                                           */
/*                                                                   */
/*                                                                   */
```

*Figure 20 (Part 3 of 7). Example of a dynamic reconfiguration generation (VM)*

```
/*********************************************************************/
;
  ADDRESS COMMAND                     /* ENSURE CP/CMS ENVIRONMENT  */
  TRACE N
  GEN_FN=""
  GEN_FT=""                           /* INITIALIZE STRING VARIABLES*/
  GEN_FM=""
  VERSION=""
  MODEL=""
  T="NO"
  ARG REST                            /* GET PARAMETERS FROM COMMAND*/
                                      /* LINE                       */
  REST=TRANSLATE(REST,' ',',')        /* GET RID OF COMMAS          */
  COUNT=WORDS(REST)
  LPCNT=1
  DO WHILE LPCNT<=COUNT               /* LOOP THROUGH ONCE FOR EACH */
    TEMP=WORD(REST,LPCNT)             /* WORD IN THE STRING         */
    PARSE VALUE TEMP WITH FRONT '=' BACK
    SELECT
      WHEN (ABBREV(TEMP,'FN')) THEN
        GEN_FN=BACK
      WHEN (ABBREV(TEMP,'FT')) THEN   /* SET APPROPRIATE VARIABLE   */
        GEN_FT=BACK                   /* ACCORDING TO THE ASSIGNMENT*/
      WHEN (ABBREV(TEMP,'FM')) THEN   /* MADE ON THE COMMAND LINE   */
        GEN_FM=BACK
      WHEN (ABBREV(TEMP,'V')) THEN
        VERSION=BACK
      WHEN (ABBREV(TEMP,'M')) THEN
        MODEL=BACK
      WHEN (ABBREV(TEMP,'T')) THEN
        T=BACK
      OTHERWISE
        SAY TEMP" IS NOT VALID, IGNORED"
    END                               /* END SELECT                 */
    LPCNT=LPCNT+1
  END                                 /* END DO                     */
  IF GEN_FM="" THEN                   /* DEFAULT FILETYPE TO "*" IF */
    GEN_FM="*"                        /* NOT CODED                  */
;
;
/* SEE IF GEN_FN, GEN_FT, AND MODEL WERE PASSED ON COMMAND LINE. IF */
/* GEN NAME OR MODEL WERE NOT SPECIFIED, GIVE CORRECT FORM & EXIT.  */
;
```

*Figure 20 (Part 4 of 7). Example of a dynamic reconfiguration generation (VM)*

```
  IF (GEN_FN="")|(GEN_FT="")|(MODEL="") THEN
    DO
      SAY "CORRECT FORM:"
      SAY ""
      SAY "VMDR  FN=GEN_FN,FT=GEN_FT,FM=GEN_FM,V=VERSION,M=MODEL,T=NO"
      SAY ""
      SAY "     -GEN_FN, GEN_FT, GEN_FM, VERSION, MODEL ARE VARIABLES"
      SAY "      THAT YOU SUPPLY ACCORDING TO YOUR GENERATION"
      SAY "     -YOU MAY CODE 'T=YES' FOR A RUN ON TEST NCP LIBRARIES"
      SAY "     -ORDER OF PARAMETERS IS NOT IMPORTANT"
      SAY "     -SPACES MAY BE USED INSTEAD OF COMMAS"
      SAY "     -FOR NCP SUBSET, CODE V=V4S"
      SAY "     -IF OMITTED, DEFAULTS ARE:"
      SAY "        FM=*"
      SAY "        V=V3"
      SAY "        T=NO"
      SAY "     -GEN_FN, GEN_FT, AND MODEL ARE REQUIRED"
      EXIT
    END
;
;
  'STATE' GEN_FN GEN_FT GEN_FM        /* SEE IF GEN EXISTS ON DISK  */
  IF RC ¬= 0 THEN
    DO
      SAY GEN_FN GEN_FT GEN_FM "DOES NOT EXIST"
      EXIT RC                         /* EXIT IF GEN DOESN'T EXIST  */
    END
;
;
/* THIS STRUCTURE VALIDATES VERSION AND MODEL AS TAKEN FROM THE     */
/* COMMAND LINE AND SETS "MACRO" ACCORDINGLY.                       */
/*                                                                  */
/* NOTE:  IF YOU HAVE CHANGED A LIBRARY NAME, OR YOUR LIBRARY NAME  */
/* DOES NOT AGREE WITH THE TABLE ABOVE, CHANGE THE ASSIGNMENT       */
/* STATEMENT OF "MACRO" TO REFLECT YOUR LIBRARY NAMES IN THE        */
/* APPROPRIATE PLACE IN THIS STRUCTURE (ACCORDING TO YOUR VERSION   */
/* AND MODEL).                                                      */
;
;
  IF (VERSION="") THEN                /* VERSION DEFAULTS TO V3     */
    DO                                /* IF NOT CODED               */
      SAY "DEFAULTING TO VERSION = V3"
      VERSION="V3"
    END
```

*Figure 20 (Part 5 of 7). Example of a dynamic reconfiguration generation (VM)*

```
SELECT
  WHEN (T="YES") THEN
      MACRO=MACXXXX                 /* FOR TEST            ALLTAG1*/
  WHEN (VERSION="V3")&(MODEL="3705") THEN
      MACRO=MAC3705                 /* FOR V3 & 3705       ALLTAG2*/
  WHEN (VERSION="V3")&(MODEL="3725") THEN
      MACRO=MAC3725                 /* FOR V3 & 3725       ALLTAG3*/
  WHEN (VERSION="V4R2")&((MODEL="3725")|(MODEL="3720")) THEN
      MACRO=MAC3725                 /* FOR V4R2&3725|3720  ALLTAG4*/
  WHEN (VERSION="V4S")&(MODEL="3720") THEN
      MACRO=MAC3725                 /* FOR V4 SUBSET       ALLTAG5*/
  WHEN (VERSION="V4R3")&(MODEL="3725") THEN
      MACRO=SNCPMAC1                /* FOR V4R3            ALLTAG6*/
  WHEN (VERSION="V5R2")&((MODEL="3720")|(MODEL="3745")) THEN
      MACRO=SNCPMAC1                /* FOR V5R2            ALLTAG7*/
  WHEN (VERSION="V4R3.1")&(MODEL="3725") THEN
      MACRO=SNCPMAC1                /* FOR V4R3.1          ALLTAG8*/
  WHEN (VERSION="V5R2.1")&((MODEL="3720")|(MODEL="3745")|,
    (MODEL="3745-130")|,
    (MODEL="3745-150")|,
    (MODEL="3745-170")|,
    (MODEL="3745-210")|,
    (MODEL="3745-410")) THEN
      MACRO=SNCPMAC1                /* FOR V5R2.1          ALLTAG9*/
  WHEN (VERSION="V5R3")&((MODEL="3720")|(MODEL="3745")|,
    (MODEL="3745-130")|,
    (MODEL="3745-150")|,
    (MODEL="3745-170")|,
    (MODEL="3745-210")|,
    (MODEL="3745-410")) THEN
      MACRO=SNCPMAC1                /* FOR V5R3            ALLTAGA*/
  WHEN (VERSION="V5R4")&((MODEL="3720")|(MODEL="3745")|,
    (MODEL="3745-130")|,
    (MODEL="3745-150")|,
    (MODEL="3745-170")|,
    (MODEL="3745-210")|,
    (MODEL="3745-410")) THEN
      MACRO=SNCPMAC1                /* FOR V5R4            ALLTAGB*/
  OTHERWISE
    DO
      SAY "VERSION = "VERSION" NOT VALID WITH MODEL = "MODEL
      SAY "WHEN RUNNING UNDER VM"
      EXIT
    END
END                                 /* END SELECT                 */
```

*Figure 20 (Part 6 of 7). Example of a dynamic reconfiguration generation (VM)*

```
;
;
  'ESTATE' MACRO 'MACLIB *'          /* SEE IF MACLIB EXISTS       */
  IF RC ¬= 0 THEN
      SAY "ERROR IN ACCESSING" MACRO "MACLIB"
;
;
/* CLEAR OLD FILE DEFINITIONS                                      */
  'FILEDEF * CLEAR'
/* WORKING SPILL FILE                                              */
/* THE DBWORKFL IS NEEDED ONLY WHEN THERE IS NOT ENOUGH VIRTUAL    */
/* MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF USERGEN IS SPECIFIED.*/
/*'FILEDEF DBWORKFL DISK  DBWORKFL FILE A ( XTENT 40'             */
/* MACRO LIBRARIES USED IN THE TABLE ASSEMBLY PHASE OF NDF         */
  'FILEDEF SYSLIB DISK' MACRO 'MACLIB *'
  'GLOBAL MACLIB' MACRO
/* INPUT FILE WITH NCP/EP GENERATION STATEMENTS                    */
  'FILEDEF GENDECK DISK' GEN_FN GEN_FT GEN_FM
/* GENERATION VALIDATION STEP OUTPUT                               */
  'FILEDEF SYSPRINT DISK' GEN_FN 'LISTING A'
/* NDF SUMMARY LISTING                                             */
  'FILEDEF PRINTER TERM'
/* SOURCE FOR TABLE 1 ASSEMBLY - OUTPUT FROM GENERATION VALIDATION */
  'FILEDEF TBL1SRCE DISK TABLE1 SOURCE A'
/* LISTING FROM THE TABLE 1 ASSEMBLY                               */
  'FILEDEF TBL1LIST DISK TABLE1 LISTING A'
/* TEXT OUTPUT FROM THE TABLE 1 ASSEMBLY                           */
  'FILEDEF TBL1OBJ DISK TABLE1 TEXT A'
/* TEMPORARY WORK FILE USED BY THE TABLE ASSEMBLY                  */
  'FILEDEF SYSUT1 DISK SYSUT1 TEMP A4 (BLOCK 4000'
/* RUN THE NDF STEP                                                */
  'ICNRTNDF'
  EXIT RC
```

*Figure 20 (Part 7 of 7). Example of a dynamic reconfiguration generation (VM)*

# Chapter 6. Loading the Program under VM

The last step in producing an operating NCP is to load the load module into the communication controller where it is to reside. You can load your NCP into a channel-attached communication controller in two ways. You can use the loader utility provided by SSP, or you can use a loader facility provided by an access method. This chapter tells you how to use the SSP loader utility. For information on how to use the access-method loader facility to load both a channel-attached and link-attached communication controller, see the *VTAM Network Implementation Guide* or the *TCAM Installation, Resource Definition, and Customization Guide*.

A communication controller module disables all channel adapters except the one over which the load operation takes place. When NCP completes its initialization phase, it enables any additional channel adapters specified as ACTIVE in the NCPCA keyword. EP enables any additional channel adapters with the keywords HICHAN and LOCHAN coded in a PEP load module.

You must manually disable any channel adapter connected to a nonoperational host before starting the load process. Messages sent to the message file indicate syntax or permanent I/O errors occurring during loading.

A virtual machine can load any IBM 3705, 3720, 3725, or 3745 Communication Controller for which you have generated a real device block. The load operation requires only that the communication controller's power be on and that the communication controller be attached to the virtual machine. If you are loading your NCP into an IBM 3720, 3725, or 3745 Communication Controller, the load controller's MOSS must also be active.

You can load the NCP load module from the host and save it on the MOSS disk if you are loading your NCP into the IBM 3720 Communication Controller using SSP V3R2 or a later release or into the IBM 3745 Communication Controller using SSP V3R4 or a later release. You can then later reload the NCP load module from the MOSS disk.

If you are loading your NCP into the IBM 3705 Communications Controller, you can load an optional diagnostic routine called the *initial test routine* before the loader utility loads NCP into the communication controller. If the initial test routine detects no malfunctions, the loader utility loads NCP into the communication controller. However, if the routine detects trouble, it stops, and the loader utility issues an error message (IFL004I). The loader utility then loads any remaining communication controllers specified in the loader step. Loading and running the initial test routine are optional but recommended steps because this routine can detect conditions that can later cause NCP failure. The initial test routine is run unless you specify its omission in the LOAD control statement.

**Note:** If the communication controller's power has been turned off since the last time you ran the initial test routine, run the routine again before reloading the communication controller. This ensures that correct parity is set in the communication controller's storage.

# Loader Utility

This section discusses the following about the SSP loader utility in a VM environment:

- Host processor and communication controller requirements
- Input to the loader utility
- Output from the loader utility.

## Host Processor and Communication Controller Requirements

The load module requires a 16KB section of user virtual storage. No work files are required to run the loader utility.

You can run the loader utility's communication controller module in any channel-attached communication controller. Before you can load the loader utility, you must ensure the communication controller:

- Has its power on
- Is identified to the VM system where you plan to run the loader utility
- Is attached to the user ID where the load is to occur
- Is not in a program-stop condition
- Has the channel online that attaches it to the operating system
- Has enabled the channel adapter where the load is to occur.

**Note:** After you start the loader utility, do not cancel the load job.

The loader utility consists of the load module IFLOADRN and the text files IFLLD1P1, IFLLD1P2, IFLLD2P1, and IFLLD2P2. For SSP V3R2 or later releases, it also contains IFWLEVEL.

For the IBM 3705 Communications Controller, the loader utility also contains one load library, SSPLIB, for the diagnostic routine.

## Input to the Loader Utility

The input to the loader utility consists of two files. One is the CMS input file that contains the NCP load module to be loaded into the communication controller. The other contains a LOAD statement specifying the NCP load module to be loaded from the host or the MOSS disk and the communication controller where it will be loaded.

**Note:** If you move the load module to another file or system prior to loading, you must ensure that the load module retains its original characteristics (for example, block size).

If you are loading your NCP into the IBM 3705 Communications Controller, you can include an optional file as input to the loader utility. This additional file contains the initial test routine and consists of modules IFL3705A, IFL3705B, IFL3705D, and IFL3705E. If you do not want to run the initial test routine, you can omit this file.

## Output from the Loader Utility

The loader utility produces one output listing, SYSPRINT. This listing contains completion or error messages produced by the loader utility. See *NCP, SSP, and EP Messages and Codes* for a description of the messages issued by the loader utility.

# Controlling the Loader Utility

This section discusses examples of the VM commands and the utility control statement that you supply to the loader utility.

## VM Commands

To run the loader utility, you must supply a number of file definitions (FILEDEFs) and then issue the command IFLOADRN to call the nonrelocatable module generated for the loader utility. The commands you need for calling the loader utility are shown in Table 12.

*Table 12. Commands for loader utility (VM)*

| | |
|---|---|
| FILEDEF SYSPRINT | Specifies the output listing file. |
| FILEDEF SYSUT1 | Specifies the input file containing the NCP load module. |
| FILEDEF SYSUT3 | Only for the IBM 3705 Communications Controller. Specifies the input file containing the initial test routine load module; it is not required if you specify DIAG=NO in the LOAD statement. |
| FILEDEF SYSIN | Specifies the file (input stream) containing the LOAD control statement. |
| GLOBAL LOADLIB | Only for the IBM 3705 Communications Controller. Specifies the LOAD library containing the initial test routine. |
| IFLOADRN | Specifies the name of the nonrelocatable module generated for the loader utility. |

**Note:** To ensure that the previously defined FILEDEFs are not in effect, clear all file definitions by issuing the command FILEDEF * CLEAR *before* issuing the file definitions for the loader utility.

## Utility Control Statement

The loader utility requires only one utility control statement, the LOAD statement. It specifies:

- The member of the input file that contains the NCP load module
- The name of the load module to be loaded from the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller
- The communication controller to be loaded
- Whether or not you want to run the initial test routine, if loading into an IBM 3705 Communications Controller
- Whether or not you want to save the load module on the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller
- Whether or not you want to request the initial program load (IPL), using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller.

The following conventions are used to describe the LOAD statement:

- Capital letters represent values you code directly without change.
- Lowercase letters represent parameters for which you must supply a value.
- Braces { } indicate you must choose from the enclosed items.
- *Or* signs | indicate you can choose between various keywords.
- An underlined value represents the default value of the keyword (that is, the loader utility uses that value if you omit the keyword).
- Brackets [ ] enclose keywords or symbols that are either optional or conditional.

The format of the LOAD statement is:

```
LOAD        LOADMOD={member name}
                    {(D)name    }
              UNIT=cuu|3725=cuu|3705=cuu
              [,AUTOIPL={NO }]
                        {YES}
              [,DIAG={Y6}]
                     {Y8}
                     {NO}
              [,SAVE={NO }]
                     {YES}
```

**LOADMOD**={*member name*}
{**(D)***name* }

Identifies the load module.

*member name*
Specifies which member of the load library indicated by SYSUT1 contains the desired NCP load module. The member must be in standard VM load module form.

**(D)***name*
Specifies the name of the load module to be loaded from a MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller.

**UNIT**=*cuu*|**3725**=*cuu*|**3705**=*cuu*

Specifies the cuu address, the virtual subchannel address where the communication controller is defined. Use Table 13 on page 129 to determine which keyword to code.

*Table 13. Keywords for the UNIT control statement (VM)*

| Communication Controller | SSP Release | Keyword |
|---|---|---|
| 3745 | SSP V3R4 and later | UNIT=cuu |
| 3720 | SSP V3R2 and later | UNIT=cuu |
| 3725 | SSP V3R2 and later | UNIT=cuu or 3725=cuu |
| 3705 | SSP V3R2 and later | UNIT=cuu or 3705=cuu |
| 3725 | SSP V3R1 | 3725=cuu |
| 3705 | SSP V3R1 | 3705=cuu |

**[,AUTOIPL={NO }]**
**{YES}**

Specifies whether or not you want to request automatic IPL from the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller. If you specify AUTOIPL=YES, automatic dump is also assumed. When an abend occurs, the dump in the communication controller storage is automatically stored on the MOSS disk and an automatic IPL is initiated from the MOSS disk.

**[,DIAG={Y6}]**
**{Y8}**
**{NO}**

For the IBM 3705 Communications Controller, specifies whether or not the loader utility is to load the initial test routine.

**Y6**
Is the default and specifies that the routine be loaded into a communication controller without extended addressing. An IBM 3705-II Communication Controller having 64KB or less of storage uses 16-bit storage addresses and, therefore, does not require extended addressing.

**Y8**
Specifies that the routine be loaded into a communication controller with extended addressing. An IBM 3705-II Communication Controller having more than 64KB of storage requires extended storage addressing.

**NO**
Specifies that the initial routine not be loaded.

**[,SAVE={NO }]**
**{YES}**

Specifies whether or not you want to save the load module from the communication controller storage on the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller. Specifying SAVE=YES is not valid with LOADMOD={(D)name}.

# Examples of VM Commands and Utility Control Statements

Since the following are examples of statements that load NCP into different communication controllers, you must modify them to fit your particular system. To load into the IBM 3720 Communication Controller, you must use SSP V3R2 or a later release; to load into the IBM 3745 Communication Controller, you must use SSP V3R4 or a later release.

## Example 1. Loading into the IBM 3720 or 3745 Communication Controller with Disk Support

Assume you want to load an NCP load module named NCP2, residing on a CMS disk, into an IBM 3720 or 3745 Communication Controller with a unit address of 030. You also want to save the load module from communication controller storage onto the MOSS disk, and you want automatic IPL from the MOSS disk. To load NCP2, use the following VM commands and utility statements:

```
/* Exec to call the loader for an IBM 3745 */
/* or 3720 Communication                  */
/* Controller with SAVE and AUTOIPL        */

address command

'filedef * clear'
'filedef sysut1 disk ncp2 loadlib a'
'filedef sysprint terminal'
'filedef sysin disk ncp2 card a'
'ifloadrn'

exit
```

where disk file NCP2 CARD A contains:

```
LOAD LOADMOD=NCP2,UNIT=030,SAVE=YES,AUTOIPL=YES
```

When you want to load the saved NCP load module from the MOSS disk, issue the following load statement:

```
LOAD LOADMOD=(D)NCP2,UNIT=030
```

Because you did not specify AUTOIPL=YES of AUTOIPL=NO, the default setting was taken and the value of AUTOIPL was reset to NO.

## Example 2. Loading into the IBM 3720, 3725, or 3745 Communication Controller

Assume you want to load an NCP load module named NCP2, residing on a CMS disk, into an IBM 3720, 3725, or 3745 Communication Controller with a unit address of 030. To load NCP2, use the following VM commands and utility statements:

```
/* Exec to call the loader for an IBM 3725,   */
/* 3720, or 3745 Communication              */
/* Controller                                 */

address command

'filedef * clear'
'filedef sysut1 disk ncp2 loadlib a'
'filedef sysprint terminal'
'filedef sysin disk ncp2 card a'
'ifloadrn'

exit
```

where disk file NCP2 CARD A contains:

```
LOAD LOADMOD=NCP2,UNIT=030
```

The preceding LOAD statement works only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3725 Communication Controller using SSP V3R1, you must specify the LOAD statement this way:

```
LOAD LOADMOD=NCP2,3725=030
```

## Example 3. Loading into the IBM 3705 Communications Controller

Assume you want to load an NCP load module named NCP1, residing on a CMS disk, into an IBM 3705 Communications Controller. This communication controller has a unit address of 030 and does not have extended addressing. To load NCP1, use the following VM commands and utility statements:

```
/* Exec to call the loader for an IBM 3705 */
/* Communications Controller              */

address command

'filedef * clear'
'filedef sysut1 disk ncp1 loadlib a'
'filedef sysprint terminal'
'filedef sysut3 disk ssplib loadlib a'
'filedef sysin disk ncp1 card a'
'global loadlib ssplib'
'ifloadrn'

exit
```

where disk file NCP1 CARD A contains:

```
LOAD LOADMOD=NCP1,UNIT=030,DIAG=Y8
```

The preceding LOAD statement works only if you are using SSP V3R2 or a later release.

If you are loading into an IBM 3705 Communications Controller using SSP V3R1, you must specify the LOAD statement this way:

```
LOAD LOADMOD=NCP1,3705=030,DIAG=Y8
```

In this example, you should load and run the initial test routine before loading the NCP load module. If you do not want to load and run the initial test routine, you would include DIAG=NO on the LOAD statement and omit the SYSUT3 FILEDEF and GLOBAL LOADLIB statements.

# Generating and Loading a Remote Communication Controller with a Floppy Disk

This section describes how to create a floppy disk and load it into a remote communication controller. It includes information on how to prepare a load module, download the module to a floppy disk, and upload the module from a floppy disk to a MOSS disk. For more information, see *3745 Models 130/150/170 Advanced Operations* and *3720/3721 Operator's Guide*.

## Load Module Preparation

You must first generate a load module. It must be able to fit on a single floppy disk and therefore cannot exceed 1MB. After you create the load module, use the VTAM MODIFY command to transfer the module from the VTAM host to a local communication controller.

## Downloading a Load Module to a Floppy Disk

Use the Display IPL Information (DII) option to download the load module from the local communication controller to a floppy disk. From the DII option, select FLOPPY MANAGEMENT. Then, use the PF key for COPY LM TO FLOPPY. You will be prompted for the CCU (only if two CCUs are available) and the name of the load module to be downloaded.

**Note:** You must format the disk before downloading the load module. A MOSS function is provided for this purpose.

## Uploading a Load Module from a Floppy Disk to a MOSS Disk

You must perform the uploading task on a remote communication controller.

Use the DII option to upload the load module from a floppy disk to a MOSS disk on a remote communication controller. From the DII option, select FLOPPY MANAGEMENT. You will be prompted for the CCU (only if two CCUs are available) and the name of the load module to be uploaded. If a load module with the same name exists on the MOSS disk, it is replaced. If a load module with the same name does not exist and space is available on the MOSS disk, the module from the floppy disk is added to the MOSS disk. If no space is available on the MOSS disk, the load module from the floppy disk replaces the oldest load module on the MOSS disk.

You must then perform an IPL of the load module from the MOSS console.

# Part 3. Generating and Loading under VSE

# Chapter 7. Generating the Program under VSE

After you install your Network Control Program (NCP) and System Support Programs (SSP) product from the tape and define NCP's configuration, the next step in producing an operating NCP is to generate the program.

This chapter contains information about generating NCP under the VSE operating system. It discusses the following topics:

- Understanding the generation procedure
- Controlling the generation procedure
- Performing different types of NCP generations
- Understanding listings and error messages.

SSP Version 3 includes the NCP/EP definition facility (NDF), a program used in generating an NCP, partitioned emulation program (PEP), or Emulation Program (EP) load module. NDF can be used to perform the following tasks:

- FASTRUN validation of an NCP, PEP, or EP generation definition
- Generation of an NCP, PEP, or EP load module
- Generation of an NCP or PEP V4 Subset load module
- Generation of an NCP or PEP phase with IBM special products or user-written code
- Generation of a text file for dynamic reconfiguration
- Migration of an existing generation definition to a different version and release or a different controller.

SSP V3R4 and later releases contain the NDF standard attachment facility, which allows user-written generation applications to interface with NDF during an NCP generation. The NDF standard attachment facility helps you define resources for user-written code. Use the NDF standard attachment facility to generate user-written code with NCP Version 4 or NCP V5R2 and later releases. For more information, see "Running an NCP or PEP Generation with IBM Special Products or User-Written Code" on page 145.

Before running NDF, you must supply job control language (JCL) to control the generation procedure. NDF does not create any JCL for you. This chapter discusses how to control the generation procedure, and Chapter 8 supplies examples of JCL for generation.

## Understanding the Generation Procedure

Generating an NCP with NDF under the VSE operating system is a six-step process. For a diagram of the input NDF accepts and the output it produces under the VSE operating system, see Figure 21 on page 136.

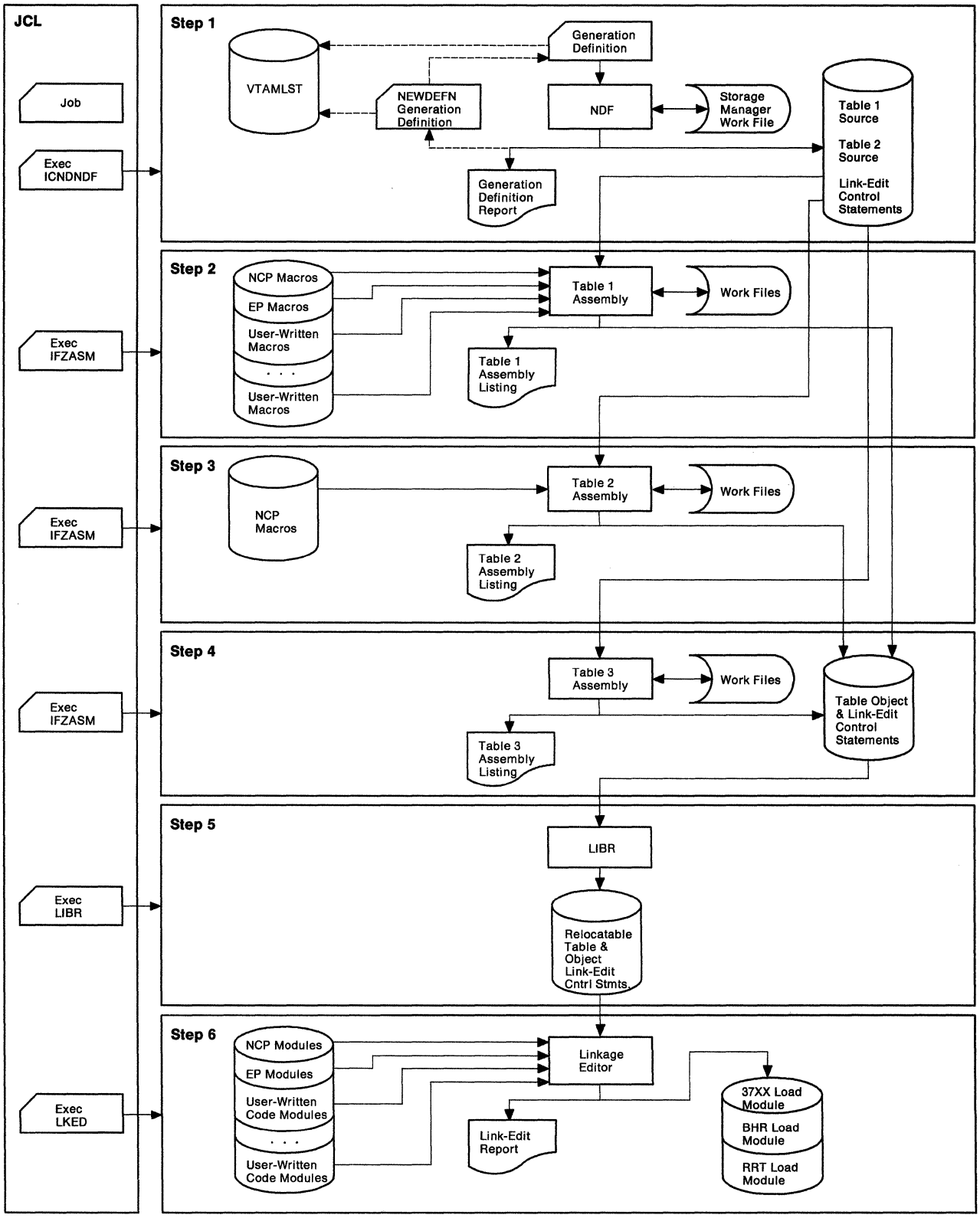



*Figure 21. The generation procedure under VSE*

## Generation Steps

**Step 1:** In the first step, the generation validation step, NDF does the following:

- Reads your generation definition file
- Validates the definition statements and keywords coded in the generation definition
- Creates the NEWDEFN file when you code the NEWDEFN keyword on the OPTIONS definition statement
- Generates assembler language source code for the resources coded in the generation definition
- Creates link-edit control statements; these statements will later link control-block objects with preassembled NCP code objects to generate NCP phases.

Step 1 produces three outputs: a listing, a file with input for the following table assembly steps, and if you are using the NDF standard attachment facility, a file containing a new generation definition.

If you are using the NDF standard attachment facility to generate resources using IBM special products or user-written code, the first step has two additional steps. During the validation step, NDF does the following:

- Dynamically loads one or more user-written generation routines
- Calls routines in the user-written generation phases to perform generation processing and allows the routines to call NDF internal routines.

**Steps 2, 3, and 4:** The second, third, and fourth steps are the table 1, table 2, and table 3 assemblies. Each assembly reads the source code specification for NCP, EP, and user control blocks into object code. All three assemblies use a single output file for all three assemblies so that only one file is needed as input for the LIBRARIAN.

Table 1 and table 2 assemblies require either NCP or EP definition statements (or both) because most of the control blocks are specified by definition statement calls. Table 3 assembly requires no definition statements.

**Step 5:** The fifth step calls the LIBRARIAN to catalog the output object modules and link-edit statements from the three table assemblies into the appropriate sublibrary.

**Step 6:** The last step, the link-edit step, links the control-block objects with the appropriate preassembled NCP code objects and generates the NCP phases. You should ignore a zero-length control section (CSECT) indication in the NCP link-edit. An NCP phase cannot exceed the 4MB (MB equals 1 048 576 bytes) boundary due to addressability constraints of the IBM 3745 Communication Controller's instruction set.

To determine how much storage is available for NCP buffers in your communication controller, perform the following calculation:

1. Locate the $BUFPOOL value in the link-edit portion of your generation listing ($BUFPOOL marks the end of the phase) and add it to the value from the ICN076I informational message issued in the GENEND definition statement. Both values are hexadecimal.
2. Subtract the value obtained in number 1 from the amount of storage available in your NCP.
3. From the value obtained in number 2, subtract the amount of storage allocated for the maintenance and operator subsystem (MOSS) Mailbox/TSS Workspace. You can find this amount in the control data set (CDS) control block at offset 46(2E); it is also entered as a number of 4KB (KB equals 1024 bytes) pages when the operator initializes NCP.
4. The number remaining from this subtraction is the amount of storage available for buffers.
5. To determine the number of buffers, add 12(C) to the value coded for BFRS on the BUILD definition statement; divide this number into the amount of storage available for buffers (obtained in number 4).

**Note:** If you want to run a FASTRUN generation to validate your generation definition without creating control blocks, do not specify the table assemblies and the link-edit to be run in your JCL. If you want to do a generation for a dynamic reconfiguration, specify in your JCL that you want only one table assembly to be run and that you do not want the link-edit to be run.

## DASD Work Space Requirements

NDF uses a storage manager to organize its work space during NDF generation validation. Whenever possible, storage manager data is kept in virtual storage. The storage manager buffers, VSAM space, and buffers for the non-VSAM files make up the GETVIS region. A fixed 64KB of virtual memory is reserved for VSAM, and the storage manager takes most of the remaining space for buffers. If data overflows these buffers, this extra data is written into a work file. Generally, enough GETVIS space is available to hold all the storage manager data. For very large generations, however, you may be required to define a work file.

NDF jobs require a fixed quantity of about 200KB for work space and an additional 48 bytes for each resource specified in the generation definition.

If you need additional work space or if you are using the NDF standard attachment facility, you need to define a work file (DBWRKFL) in your JCL. To define DBWRKFL, use the Access Method Services to specify a cluster for a relative record file of 4096 byte records. Establish this cluster before the NDF job that uses DBWRKFL.

The following is an example of an IDCAMS job to define such a cluster:

```
// JOB DEFCLUST
*
*  COPYRIGHT=NONE
*
*  EXAMPLE OF A JOB TO DEFINE A VSAM CLUSTER FOR THE DBWRKFL
*
// EXEC IDCAMS,SIZE=AUTO
   DEFINE CLUSTER                  -
        (NAME(VSAM.WORK)           -
         VOL(DT9354)               -
         CISZ(4608)                -
         NUMBERED                  -
         RECORDSIZE(4096 4096)     -
         TRACKS(10 5))             -
       DATA                        -
        (NAME(VSAM.WORK.DATA)      -
         FILE(DBWRKFL))
/&
```

Generally, one cylinder of disk space allocated for a work file defined for VSAM space should be adequate.

If you are generating NCP/Token-Ring interconnection resources, you need an additional 80 bytes for each NCP/Token-Ring interconnection physical line specified in the generation definition and an additional 160 bytes for each NCP/Token-Ring interconnection logical line generated using the AUTOGEN facility.

### Performance Considerations

NDF requires 3.5MB of virtual storage to achieve optimal performance. If the available virtual storage drops below 3.5MB, paging during the generation validation step significantly degrades performance.

During the generation procedure, intermediate files, written as SYSIPT, transfer NDF's output to IFZASM and IFZASM's output to the LIBRARIAN. Since these two files are blocked with fixed block architecture devices, you can achieve some reduction in generation time by using FBA devices for them.

## Controlling the Generation Procedure

You control the generation procedure through the JCL that you code and through the definition statements and keywords that you coded in the generation definition.

This section explains the different types of generations you can run. It discusses some of the definition statements and keywords you can code in your generation definition. It also discusses the parameters you need to code and the files you need to specify in your JCL. For examples of JCL for generation, see Chapter 8.

### Specifying Files Used by NDF

This section contains the names of the files (dtfnames) used by NDF. You specify the dtfnames in your JCL. Table 14 on page 140 lists the dtfnames and descriptions of these files.

*Table 14. dtfnames of files used by NDF (VSE)*

| dtfname | Description |
|---|---|
| ____ | Specifies the library with a dtfname determined by the user. A LIBDEF statement indicates the sublibrary where NDF and IFZASM reside. If you have any user-written generation applications that use the NDF standard attachment facility, LIBDEF must also specify the sublibrary containing the user-written generation phase. |
| **IJSYSIN** | Specifies the NDF input file. This file contains the NCP or PEP generation definition. IJSYSIN is also the dtfname for the input files for the IFZASM assembler and for the LIBRARIAN step that catalogs the NCP table objects into relocatable members. |
| **DBWRKFL** | Specifies the NDF work file. This temporary file stores internal data in 4KB records. This file is a VSAM relative record file. If you need this file, define it with IDCAMS before you start NDF. This file is required if you are using the NDF standard attachment facility. |
| **IJSYSPH** | Specifies the file into which NDF writes the input for all three assemblies. A "/*" statement separates inputs for different assemblies. The assembler reads this file as input for each of the assemblies. IJSYSPH also identifies the file used by the assembler for text output. The outputs from all three assemblies are written sequentially to one file. This file is then referred to as IJSYSIN by the LIBRARIAN. |
| **IJSYSNW** | Specifies the output file containing the new generation definition created by NDF. For more information, refer to *NCP, SSP, and EP Resource Definition Guide*.<br><br>The new generation definition consists of the input from the definitions from the NCP generation definition plus statements and keywords added during the generation process.<br><br>**Notes:**<br>1. If you specified NEWDEFN=YES on the OPTIONS definition statement in your generation definition, you must define the IJSYSNW file in the JCL for your generation.<br>2. If VTAM users generate an IJSYSNW file, they must include an IJSYSNW file in the VTAMLST that VTAM accesses during the activation of this NCP. If they do not generate an IJSYSNW file, they must include their generation definition (GENDECK) in VTAMLST.<br>3. All VTAM users must include the IJSYSNW source in the VTAMLST that VTAM accesses during the activation of NCP.<br><br>For information about NEWDEFN, refer to *VTAM Network Implementation Guide*. |
| **VTAMLST (5)** | |

## Specifying Parameters for NDF

This section describes the optional NDF parameters that you can specify in your JCL. When specifying more than one parameter in the parameter field, you must separate the parameters with a comma.

### LINECNT Parameter

Use the LINECNT parameter to specify the number of lines on each page of the generation validation listing and the table assembly listing. The valid range for this parameter is 10 to 99. The default value for the validation listing and for the assembly listing is 60. If you specify a value for LINECNT, this value is used in all listings.

The following is an example of LINECNT in the JCL:

```
//EXEC PGM=ICNDNDF,SIZE=AUTO,PARM='LINECNT=40'
```

### FASTRUN Parameter

For SSP V3R2 and later releases, you can use the FASTRUN parameter to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

Using the FASTRUN parameter is the same as coding FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition.

The following is an example of FASTRUN in the JCL:

```
//EXEC PGM=ICNDNDF,SIZE=AUTO,PARM='FASTRUN=ON'
```

### Migration Aid Parameters

For SSP V3R5 and later releases, you can use the migration aid parameters to invoke the migration aid. The migration aid is an NDF function that automates much of the NCP migration task. For more information on these parameters, refer to *NCP, SSP, and EP Resource Definition Reference* and *NCP Migration Guide for Version 5 Release 4*.

The following is an example of the migration aid parameters in the JCL:

```
// EXEC ICNDNDF,...,PARM='TMODEL=3745-410,TUSGTIER=5,TVERSION=V5R4'
```

## Naming Resources

Avoid using the prefixes shown in Table 15 on page 142 and the labels shown in Table 16 on page 142 when naming resources because they are used as control-block identifiers and can cause duplicate labels that result in an error message from the assembler.

*Table 15. Prefixes to avoid (VSE). Avoid names that are similar to control-block acronyms.*

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| @ | BOQ | CPT | EPI | IX | LTV | NPB | QCB | SNP | UNA |
| $ | BPB | CRB | EQB | J* | LTX* | NPF | RAT | SOT | USC |
| AAB | BSB | CRP | ERB | LAA | LU | NQB | RCB | SPC | UXR* |
| ABN | BST | CTB | ERX | LAB | LX | NQE | RCQ | SST | U1 |
| ACB | BTT | CTP | FCT | LB* | L1B | NSQ | RCV | STE | VAT |
| ACT | BTU | CUB | FLB | LCB | L4B | NVT | RG | STQ | VIT |
| ACU | BUE | CY | FMT | LCC | MBF | NVX | RH* | SUT | VLB |
| AEB | CA* | CX | FVT | LCI | MBX | OLL | RN* | SVT | VR |
| ALE | CAB | DAE | GCB | LCP | MCT | OLT | RU* | SXB | VST |
| AST | CAI | DDB | GPT | LCS | MDR | PAB | R* | SYS | VTS |
| ATB | CAR | DIA | GRW | LCW | MIB | PAD | RMB | TCB | VVT |
| ATP | CAT | DPT | GVT | LDA* | MIC | PCB | ROSH* | TET | WCB |
| ATT | CB | DQB | HWE | LDI* | MIF | PIU | RST | TGB | WRP |
| AV* | CBB | DRS | HWX | LGT | MIH | PL* | RTR | TH* | WU |
| AXB | CDS | DRX | IB | LKB | MIM | PL2 | RVT | TIM | X |
| BC | CER | DSP | ICE | LKC | MLT | PMF | SCB | TND | XDA |
| BCU | CGP | DTG | ICI | LNB | MMV | PRB | SEB | TQB | XDB |
| BER | CHC | DVB | ICW | LNV | MSC | PSA | SGE | TRT | XDH |
| BGS | CHV | DVI | IDD | LPB | MTF | PSB | SGT | TVS | XID |
| BH | CIE | DVQ | IDE | LRB | NET | PSI | SHB | UAC | |
| BHD | CM | ECB | IDL | LRC | NIB | PSP | SID | UAD | |
| BHR | COE | ECD | IDB | LTC | NiX | PST | SIT | UIB | |
| BHS | CPI | ECL | IRN | LTR | NLB | PUV | SMB | UIC | |
| BLU | CPN | EML | IRQ | LTS | NLX | QAB | SMM | ULVSGN | |

**Note:** * Indicates that a number from 0 to 9 follows this prefix.

*Table 16. Labels to avoid (VSE). Avoid names that are similar to control-block acronyms.*

| | | | | | |
|---|---|---|---|---|---|
| ACITRAP | CSPQH2 | NCPHIST1 | SVCQUT | THLOB | TMRF |
| CAACER | CSPQOFF | NCPLVL | SWQTMQ1 | THLOM | TTCUR |
| CACCER | CSPQON | NEWLNE | SWQTMQ2 | THMID | TTEND |
| CADCER | DCTABND | OLDLNE | TABEND | THMPF | TTRECNTR |
| CAECER | DCTSAVEK | PEPQSCNB | TABSTAR | THODAIB | TTSKPCNT |
| CAFCER | D*RCB | PEPQSCNM | THAFIB | THODAIM | TTSTAR |
| CCPH1 | EPLVL | PSCA | THAFIM | THONLY | UIHRCCW |
| CCPSAVE | FILLB | ROSSVADDR | THBCUVVT | THPSIB | USTAGETR |
| CHANSNS1 | FILLC | ROSSVCCR | THFID | THPSIM | UTILSTSZ |
| CHANSNS2 | HDRNENT | ROSSVCCU | THFIRST | THTYPO | |
| CHSVBKSV | ICNTABL1 | ROSSWK1 | THFOB | THTYP1 | |
| CHSVH1 | LCDBSCB | SECNTRI | THFOM | THTYP2 | |
| CSPQH1 | LCDSSBIT | SVCO | THLAST | THTYP3 | |

**Note:** * Indicates that a number from 0 to 9 can appear as this character.

## Defining Virtual Storage

You can control virtual storage available to NDF for work space by specifying SIZE=AUTO on the EXEC statement in the JCL. Specifying SIZE=AUTO allows the system to determine the amount of virtual storage available. A region of 4MB should be adequate for most NDF runs, although very large generation definitions may require more than twice this much storage. If storage is exceeded, increase virtual storage or define the DBWRKFL file.

The following is an example of SIZE=AUTO in the JCL:

```
// EXEC ICNDNDF,SIZE=AUTO
```

## Naming Phases

Besides creating NCP phases, NDF also produces a resource resolution table (RRT) phase and, if you have coded any block-handling routines, a block-handler set resolution table (BHR) phase. The RRT and BHR phases contain information that the access method requires. The NCP, RRT, and BHR phases are placed in a sublibrary, determined by the ACCESS librarian command.

Use the NEWNAME keyword on the BUILD definition statement to designate the names for the BHR, RRT, and NCP phases. NDF appends a *B* to the NEWNAME value to name a BHR phase, and NDF appends an *R* to the NEWNAME value to name an RRT phase.

For information about the NEWNAME keyword on the BUILD definition statement, see *NCP, SSP, and EP Resource Definition Guide*. For information on how to code this keyword, see *NCP, SSP, and EP Resource Definition Reference*.

If you are generating your NCP for the IBM 3745 Communication Controller, the link-edit produces seven phases. If you are generating your NCP for the IBM 3720 or 3725 Communication Controller, the link-edit produces six phases. If you are generating your NCP for the IBM 3705 Communications Controller, the number of phases depends on whether you included user-written code or block-handlers and whether the storage of the user code is high or low.

Table 17 illustrates how the number of phases for the IBM 3705 Communications Controller depends on the relationship between storage of user-written code, keywords specified, and block handlers coded.

*Table 17. Determining the number of phases for an IBM 3705 Communications Controller (VSE)*

| Block Handlers and User Code Included | Keywords Specified | Storage of User Code | Number of Phases |
|---|---|---|---|
| No user code | | | 1 |
| User code | SRCLO and/or INCLO | Low | 2 |
| User code | SRCHI and/or INCHI | High | 2 |
| User block handlers | | High | 2 |
| User code | SRCLO and/or INCLO and SRCHI and/or INCHI | Low and high | 3 |
| User code and user block handlers | SRCLO, INCHI, and/or INCLO | Low and high | 3 |

For all communication controllers with more than one phase present, the first phase is named with the value specified for the NEWNAME keyword. The other phase names are derived from this value. All phase names, except the first, are 8 bytes long and made up of the NEWNAME value with zeros concatenated to 7 bytes. The eighth byte contains a suffix starting at 2 and incremented by 1 for each phase, as shown in the following:

- If NEWNAME=NCPA for an NCP generated for the IBM 3720 or 3725 Communication Controller, the phase names are NCPA, NCPA0002, NCPA0003, NCPA0004, NCPA0005, and NCPA0006.

- If NEWNAME=NCPA for an NCP generated for the IBM 3745 Communication Controller, the phase names are NCPA, NCPA0002, NCPA0003, NCPA0004, NCPA0005, NCPA0006, and NCPA0007.
- If NEWNAME=NCPA for an NCP generated for the IBM 3705 Communications Controller and you specified INCLO and SRCHI on the GENEND definition statement, the phase names are NCPA, NCPA0002, and NCPA0003.

### Controlling Succeeding Generation Steps

You can use a system return code to determine whether to run succeeding job steps. The examples of JCL in Chapter 8 check the condition code before initiating succeeding job steps.

## Performing Different Types of NCP Generations

This section discusses the different types of NCP generations and what you must do to run them.

### Running a FASTRUN Generation

Do a FASTRUN generation to check for errors before running a complete generation. A FASTRUN generation checks your generation definition for syntax and definition errors without creating control blocks or link-edit control statements.

To run a FASTRUN generation, code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for SSP Version 3. For SSP V3R2 and later releases, you may alternately code FASTRUN=ON as a parameter in your JCL when calling NDF. Ensure that your JCL does not call the linkage editor; if the link-edit step is present, an error will result. Also, do not define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

For an example of the JCL for a FASTRUN generation, see page 153.

### Running a Standard NCP or PEP Generation

To run a standard NCP or PEP generation, supply your generation definition as input and specify the various input and output files in your JCL.

Ensure that you allow the LENAME keyword on the BUILD definition statement to default to the INLINKED suboperand or, if you code a value for this keyword, ensure that you use the same value name on the INCLUDE statement in the link-edit step of your JCL.

If you are including certain types of resources in your generation definition (such as those listed under IJSYSNW in Table 14 on page 140), you must code NEWDEFN=YES on the OPTIONS definition statement as the first executable statement in your generation definition and define the IJSYSNW file in your JCL. For more information on coding the NEWDEFN keyword, refer to *NCP, SSP, and EP Resource Definition Reference.*

For an example of JCL for running a standard NCP or PEP generation, see page 156.

## Running an NCP or PEP Generation with IBM Special Products or User-Written Code

If you included IBM special products or user-written code—such as Network Terminal Option (NTO), Network Routing Facility (NRF), or X.25 NCP Packet Switching Interface (NPSI)—in an NCP or PEP generation, you must modify the basic JCL.

If you are using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing. You are not required to use this method.

If you choose to generate your user-written code and NCP *without* using the NDF standard attachment facility or if you are generating user-written code using SSP V3R1, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of link-edit statements by coding certain keywords on the GENEND definition statement.

### Using the NDF Standard Attachment Facility

You can run this type of generation *only* if you are using SSP V3R4 or a later release to generate NCP Version 4 or NCP V5R2 or a later release. To use the NDF standard attachment facility, you must supply a user-written generation application. For information on writing user-written code and user-written generation applications, see *SSP Customization*. Figure 22 on page 146 shows how to include your user-written code and user-written generation phases in the generation procedure.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation phases to be loaded in the generation. Each application must have its own generation phase. You can name up to 25 generation phases.
- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation phases.
- Modify the JCL for a standard NCP or PEP generation to include the dtfnames for the IJSYSNW file, the DBWRKFL file, and the libraries for user-supplied modules.

For an example of the JCL for generating user-written code using the NDF standard attachment facility, see page 161.

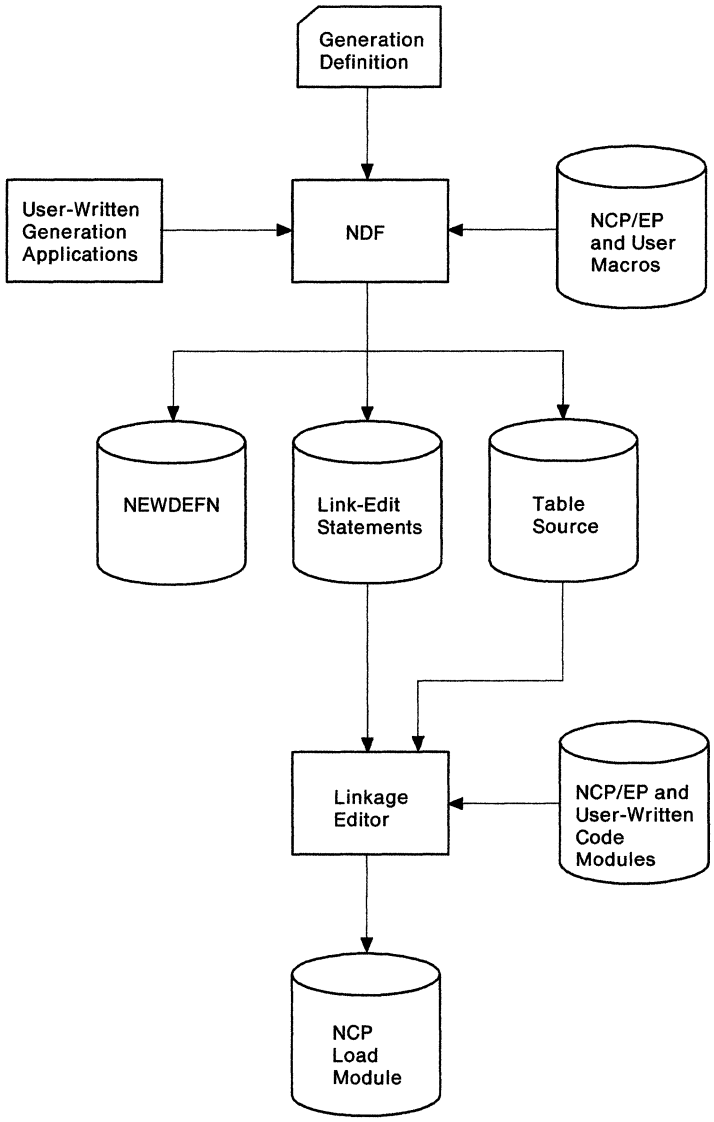



*Figure 22. Generating an NCP containing user-written code using the NDF standard attachment facility (VSE). This figure shows how to include user-written generation phases in an NCP or PEP generation.*

## Using the GENEND Definition Statement

Besides being able to generate NCP Version 4 or NCP V5R2 and later releases with IBM special products or user-written code using the NDF standard attachment facility, you can also generate them as described here.

You can use the GENEND definition statement to generate any version of NCP using SSP Version 3. Before generating NCP, code the link-edit statements for the routines and identify the location of these link-edit statements by coding certain keywords on the GENEND definition statement. Before you generate, ensure that you:

- Run any job required to generate SRCLO or SRCHI code
- Catalog the members with SRCLO or SRCHI code as members of type *A*
- Edit and catalog any definition statements called by the SRCLO or SRCHI code as members of type *F*

- Catalog link-edit control statements and preassembled object code as members of type *OBJ*
- Add sublibraries that contain source code or edited definition statements to the LIBDEF chain for SOURCE (establish this LIBDEF chain before IFZASM is called for the table assemblies)
- Add sublibraries that contain link-edit control statements to the LIBDEF chain for OBJ (establish this LIBDEF before you call the linkage editor).

For an example of the JCL for generating user-written code and the NCP using the GENEND definition statement, see page 162.

Figure 23 on page 148 shows how your user-written code is included in the generation procedure.

## Running a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, use the text file from a dynamic reconfiguration generation. Ensure that you coded the original NCP to allow dynamic reconfiguration. The dynamic reconfiguration generation produces a text file that the access method can use to modify NCP.

**Note:** VTAM has its own dynamic reconfiguration procedures that do not require you to use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Network Implementation Guide.*

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration file consisting of ADD or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration file is the input for the dynamic reconfiguration generation. This type of generation produces a text file that the access method uses to modify an NCP that is already running in a communication controller. For information on using ADD, DELETE, PU, and LU definition statements, see *NCP, SSP, and EP Resource Definition Guide.*

A dynamic reconfiguration generation requires one table assembly and no link-edit. For an example of JCL for a dynamic reconfiguration generation, see page 163.

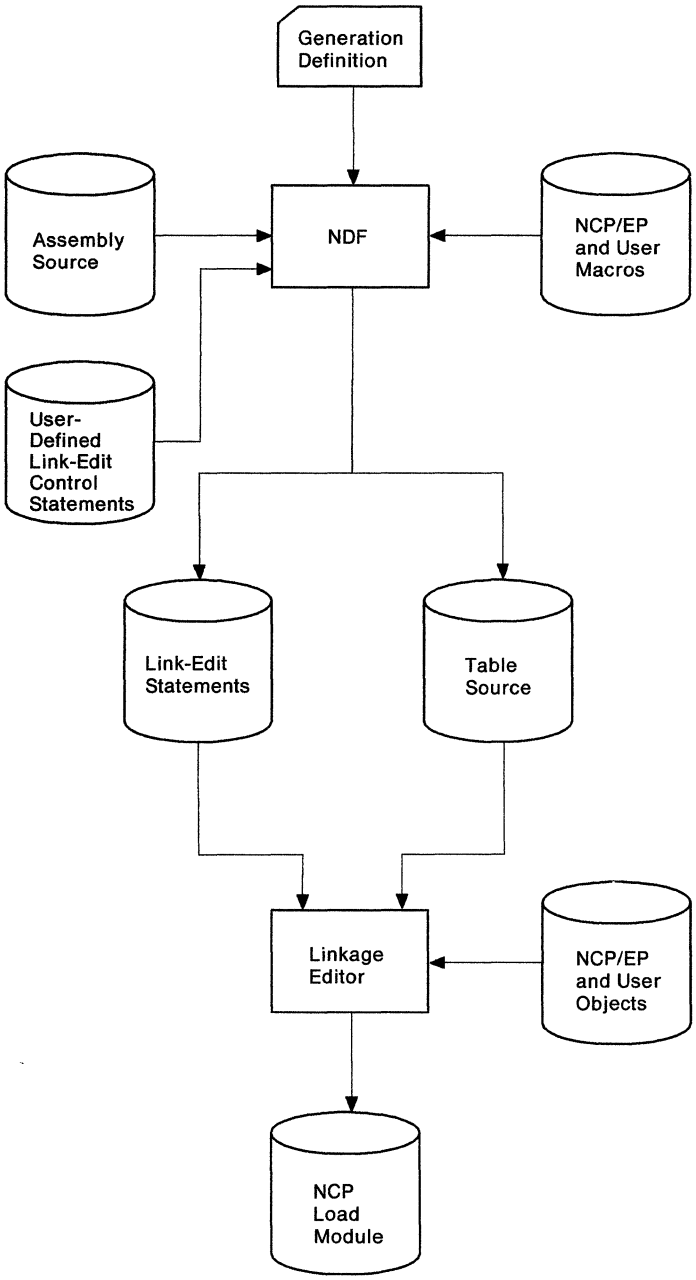



*Figure 23. Generating an NCP containing user-written code using the GENEND definition statement (VSE). This figure shows how to include user-written code in an NCP or PEP generation.*

# Understanding Listings and Error Messages

During generation validation, NDF creates a report that contains:

- The input statements interspersed with informational and error messages
- The keywords and statements passed to NDF from generation application phases using the NDF standard attachment facility (SSP V3R4 and later releases)
- A resource name and network address cross-reference (only if the generation validation run is valid)
- An error message summary
- A return code for the generation validation step.

The input for all three table assembly steps is contained in one output file. This file is the SYSPCH file during the NDF generation validation step and the SYSIPT file during the table assemblies.

In NCP V4R3 and later releases or NCP Version 5, generation definition listings produced by SSP V3R4.1 and later releases include a message indicating how much storage NCP needs for initialization, in excess of the storage that the phase displaces.

If any errors occur in generation validation, NDF notes these errors through diagnostic messages in the report. Table 18 shows the NDF message severity levels and their meanings.

*Table 18. NDF message severity levels (VSE)*

| Severity Levels | Meaning |
|---|---|
| Info | This is an informational message that either informs you of NDF calculations (such as message ICN0761) or indicates how NDF has changed, ignored, deleted, or added a keyword. NDF did not consider the message serious enough to stop the generation process; however, you should examine the message to determine whether you want to accept the NDF change or make your own to the generation definition. |
| Warning | An error has occurred for which NDF has taken corrective action by assuming a default keyword value or by ignoring the value supplied. The generation process is terminated after validation of the generation definition. The NDF migration aid also issues a warning message when it cannot determine a value to use. |
| Error | A user error has occurred for which NDF cannot assume a value or ignore the value supplied. The generation process is terminated after validation of the generation definition. |
| Ten | A fatal user error has been detected. The generation process is terminated. |
| Severe | A system error has occurred. NDF produces a procedure traceback. The generation process is terminated after validation of the generation definition. |
| Fatal | A fatal system error has occurred. A procedure traceback is printed and the generation process is terminated. |

For all but the informational messages, NDF ends output of control-block source and link-edit control statements, but continues to validate the input definition statements.

In this case, you must correct the errors and run the generation validation again. If the return code from the generation validation and the table assemblies is 0, NDF runs to completion, runs the link-edit, and produces load modules.

Other programs, such as VTAM and the configuration report program, require as input the same definition statements coded to generate NCP, plus several additional keywords and definition statements specific to each procedure. *NCP, SSP, and EP Resource Definition Reference* identifies these additional keywords and definition statements. Although you can place these keywords and definition statements in NCP generation definition either before or after you generate NCP, it is recommended that you add them before you generate NCP because executing the different procedures with different inputs can create errors.

If you are using the NDF standard attachment facility (SSP V3R4 or a later release) to generate user-written code with your NCP, products that use the same generation definition may require as input the source statements or keywords passed to NDF from the generation application phases. By specifying NEWDEFN=YES on the OPTIONS definition statement in your generation definition and by specifying the IJSYSNW file in your JCL, you instruct NDF to create a new generation definition for use by these procedures. This new generation definition contains both the NCP source statements and the statements passed from the user-written generation phases.

NDF checks only the accuracy of the values coded for NCP generation procedures that appear in the NCP generation input. In the same way, these other procedures do not check the validity of the NCP definition statements and keywords. Therefore, NDF requires that NCP generation have errors less than a severity code of 4.

## Sample NDF Generation Report

Figure 24 on page 151 contains an example NDF generation report. Reverse-coded numbers (for example, 3 ) indicate comments about the report that are not a part of the actual report. You can find the comments corresponding to these numbers following the report. Ellipses (. . .) in the report show that parts of the report were deleted for this example.

```
1                  2                  3
ACF SSP V3R4.1       05/09/90  09:52:13      DEFINITION SPECIFICATION                                  PAGE 5
4
LINE #                STATEMENT
   177  *         FULL LINE COMMENT  5
   178  G14B1     GROUP CUTOFF=1,              ONE SUB-BLOCK ACCEPTED           . X     6
   179                 DIAL=NO,               NON-SWITCHED LINES                 X
   180                 LCNTL=BSC,             BSC LINE CONTROL                   X
   181                 NPACOLL=YES,           COLLECT NPA ON THIS GROUP          X
   182                 PU=YES                 (V) VTAM.  FOR CROSS DOMAIN        X
                  . . .
                     D               WAKDLAY=2,2
                     D               SYNDLAY=1,0
                     D               TTDCNT=15
                     D               WACKCNT=15
   201  L14023    LINE  ADDRESS=(G23),         LINE ADDRESS ON 3705               X
   202                 CODE=EBCDIC,           EBCDIC 3720'S ONLY                 X
   203                 CUTYPE=3271,           3271'S DEFINED                     X
   204                 DUPLEX=FULL,           FULL DUPLEX FACILITY               X
   205                 ETRATIO=30,            ERROR-TO-TRANSMISSION RATIO-3% NPDA X
   206                 INTPRI=1,              INTERRUPT PRIORITY IS 1            X
   207                 LDATS=YES,             LINE PROBLEM DETERMINATION SUPPORTEDX
   208                 NEGPOLP=.2,            PAUSE .2 SEC AFTER NEGATIVE RESPONSEX
   209                 NEWSYNC=YES,           CONTROLLER SUPPLIES NEW-SYNC SIGNAL X
                  . . .
                     G               TYPE=NCP
*ERROR*     ICN001I  08  ADDRESS=(G23)   INVALID, REQUIRED  8
                     G               CLOCKNG=EXT     7
                     D               DATRATE=LOW
                     D               SPDSEL=NO
                     D               PROMPT=YES
                     D               DIALALT=NONE
                     G               TRANSFR=3
                  . . .
   259  B14023A  CLUSTER   CRITSIT=YES,        SEND CLOSE-DOWN MESSAGE            X00000400
   260                 NPACOLL=YES,           DATA CAN BE COLLECTED FOR NPA      X00000500
   261                 GPOLL=40407F7F,        GENERAL POLL ADDRESS               X00000600
   262                 XMITLIM=1,             SEND OR RECEIVE ONE TRANSMISSION   X00000700
   263                 ISTATUS=ACTIVE         (V) VTAM                            00000800
                     L               CUTYPE=3271
                     D               CDATA=NO
                     D               INHIBIT=NONE
                     D               LGRAPHS(1)=REJECT
                     D               LGRAPHS(2)=REJECT
                     D               FEATURE=NOGPLKUP
                     D               ITBMODE(1)=NO
                     D               ITBMODE(2)=NO
                     D               BHSET=NONE
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30      LABEL CROSS REFERENCE                                     PAGE 41

                        LABEL CROSS REFERENCE -- SORTED BY LABEL NAME  9

LABEL         LINE     SA  ELEM       LABEL        LINE     SA  ELEM       LABEL        LINE     SA  ELEM

                  . . .

G14B1         178                     T14020BB     545      0E  001B       T14022G8     915      0E  004D
G14S1         362                     T14020BC     549      0E  001C       T14022G9     916      0E  004E
                  . . .
L14022        828      0E  0044       T14020B8     533      0E  0018       T14023A7     305      0E  0009
L14023        201      0E  0001       T14020B9     537      0E  0019       T14023A8     315      0E  000A
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30      LABEL CROSS REFERENCE                                     PAGE 42

                        LABEL CROSS REFERENCE -- SORTED BY NETWORK ADDRESS  10

SA  ELEM     LINE   LABEL              SA  ELEM    LINE    LABEL              SA  ELEM    LINE    LABEL

0E  0001     201    L14023             0E  0038    747     T14020F8           0E  006E    90      DRPOOLLU
0E  0002     259    B14023A            0E  0039    749     T14020F9           0E  006F    6       NCPBUILD
0E  0003     269    T14023A1           0E  003A    767     P14020G            0E  0070    6       NCPBUILD
0E  0004     275    T14023A2           0E  003B    785     T14020G1           0E  0071    6       NCPBUILD
                  . . .
ACF SSP V3R4.1       05/09/90  13:51:30   ERROR SUMMARY                                                PAGE 41

TOTAL MESSAGES      INFO     WARNING     ERROR     SEVERE     FATAL
       1             0         0          1         0          0

RETURN CODE IS 8

MESSAGES APPEAR AFTER THE LINES NUMBERED:  11

209

REGENERATION REQUIRED
ACF SSP V3R4.1       05/09/90  09:52:13   NDP RETURN CODE SUMMARY

GEN DECK PROCESSOR         8
TABLE ONE BUILD       NOT RUN
TABLE TWO BUILD       NOT RUN
    NDF OVERALL            1
```

*Figure 24. Sample NDF generation report (VSE)*

## Comments

1. **SSP version and release number.**

2. **Date and time of the NDF run.** The date and time of the NDF run are the same as those recorded in the date and time generation control block in the NCP or PEP phase and printed in the formatted portion of the NCP or PEP phase and dump.

3. **Report section identification.** This identification has one of the following values: DEFINITION SPECIFICATION, LABEL CROSS REFERENCE, or ERROR SUMMARY.

4. **Line number column.** This column contains the line numbers of the generation definition listing.

5. **Full-line comments from the generation deck.**

6. **Partial-line comment from the generation deck.**

7. **Information describing defaulted or inherited keywords.** This 1-letter prefix to the message shows keywords that are defaulted or keywords that use values from previous definition statements. These prefixes are:

   G Keyword inherited from GROUP
   L Keyword inherited from LINE
   T Keyword inherited from TERMINAL
   C Keyword inherited from CLUSTER
   P Keyword inherited from PU
   D Keyword that has been defaulted

8. **Error message.** This error message has an appropriate error number followed by a severity code and error message text. A severity code of 4 or more requires correction to the generation definition before you can generate a phase. A severity code of less than 4 informs you that NDF has taken corrective action and does not require regeneration. You should, however, verify that the correction made by NDF will satisfy your generation requirements.

9. **First label cross-reference.** This list contains all user-coded labels, sorted by label name. If the label has an associated network address, it is printed. Resources defined by the keywords LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist. It is included in this sample as an illustration only.

10. **Second label cross-reference.** This list contains all user-coded labels, sorted by network address. Labels without associated network addresses are omitted. Resources defined by the keywords LUDRPOOL, PUDRPOOL, LUPOOL, and GWNAU appear in this list only if they are specified with a user-coded label. This section is not printed when severity codes of 4 or more exist.

11. **Error summary section.** This summary contains an error count and a list of the line numbers immediately preceding error messages. If more than one error message immediately follows a given line, the line number is printed only once. If only informational messages follow a given line, an asterisk is printed next to the line number.

# Chapter 8. Examples of JCL for Generation under VSE

This chapter contains examples of JCL for generating your NCP under the VSE operating system. Although you can find most of these examples on the SSP tape sent from IBM Software Distribution, they are also supplied here for easy reference. Before using these examples, ensure that they reflect your operating environment.

This chapter includes examples of JCL for the following types of generations:

- A FASTRUN generation
- An NCP or PEP generation
- An NCP or PEP generation with user-written code using the NDF standard attachment facility
- An NCP or PEP generation with user-written code using the GENEND definition statement
- A dynamic reconfiguration generation.

## Example of a FASTRUN Generation

Before running a complete generation, you can run a FASTRUN generation to check your generation definition for syntax and definition errors without creating control blocks or link-edit control statements. Figure 25 on page 154 shows the JCL that generates an NCP phase using FASTRUN generation. NDNAME in Figure 25 on page 154 is valid for SSP V3R5.

To run a FASTRUN generation:

- Code FASTRUN=ON on the OPTIONS definition statement as the first executable statement in your generation definition for SSP Version 3. You can also code FASTRUN=ON as a parameter in your JCL when calling NDF using SSP V3R2 and later releases. This example uses the FASTRUN parameter coded in the JCL.
- Ensure your JCL *does not* call the table assemblies or the linkage editor. If the link-edit step is present, an error results.
- *Do not* define the NCP definition statement library because NDF does not run table assemblies for a FASTRUN generation. However, if you include user-written code in the generation definition, define the definition statement library that contains user-written link-edit control statements.

This example assumes you have *not* included any user-written code using keywords on the GENEND definition statement. If you did, however, you must include a LIBDEF statement in your JCL containing the user-written code table assembly and link-edit statements.

The following is an example of JCL of a FASTRUN generation.

```
// JOB NDF
*
*  COPYRIGHT=NONE
*
*  EXAMPLE OF A FASTRUN GENERATION.
*
*  THE SSP MEMBERS REQUIRED TO CARRY OUT AN NCP GENERATION ARE
*  ASSUMED TO RESIDE IN THE SSPLIB SUBLIBRARY.
*
*  A LIBDEF STATEMENT IS NEEDED TO INDICATE THE SUBLIBRARY
*  WHERE NDF RESIDES.  THE SUBLIBRARY WHERE VSAM PHASES
*  RESIDE MUST ALSO BE PART OF THE SEARCH CHAIN IF THE
*  WORK FILE (DBWRKFL) IS TO BE USED.
// LIBDEF PHASE,SEARCH=(NCPLIB.SSPLIB,NCPLIB.PR$AM2)
*
*  THIS EXAMPLE ASSUMES THAT THE GENERATION DEFINITION IS INCLUDED IN
*  LINE.  IF IT IS ON DISK, A SERIES OF JOB CONTROL STATEMENTS SIMILAR
*  TO THE FOLLOWING ARE NEEDED.
*  // DLBL IJSYSIN,'SAMPLE GENERATION'
*  // EXTENT SYSIPT
*  // ASSGN SYSIPT,DISK,PERM,VOL=DSKID,SHR
*
*  IF NEWDEFN=YES OR NEWDEFN=(YES,ECHO) IS SPECIFIED IN THE
*  GENERATION DEFINITION, JOB CONTROL STATEMENTS SIMILAR TO
*  THE FOLLOWING ARE NEEDED.
*  // DLBL IJSYSNW,'NEWDEFN',0001,SD
*  // EXTENT SYS001,YYYYYY,,,3201,40
*  // ASSGN SYS001,DISK,PERM,VOL=YYYYYY,SHR
*
*  THE LISTING IS SENT TO A PRINTER
// ASSGN SYSLST,00E,PERM
*
*  THE STORAGE MANAGER WORK FILE IS NEEDED ONLY WHEN THERE IS NOT
*  ENOUGH VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
*  USERGEN IS SPECIFIED.
*  // DLBL DBWRKFL,'VSAM.WORK',0,VSAM
*
*  NDF IS EXECUTED WITH SIZE=AUTO SO THAT ANY EXTRA VIRTUAL
*  MEMORY WILL BE AVAILABLE IN THE GETVIS AREA
*  NOTE:  FASTRUN IS SET ON BY SPECIFYING IT ON THE FIRST
*  EXECUTABLE STATEMENT IN THE GENERATION DEFINITION AND/OR
*  BY CODING IT AS A PARAMETER IN THE JCL AS SHOWN BELOW.
// EXEC ICNDNDF,SIZE=AUTO,PARM='LINECNT=55,FASTRUN=ON'
*
*  PLACE YOUR GENERATION INPUT FILE HERE
*
/*
```

*Figure 25 (Part 1 of 2). Example of a FASTRUN generation (VSE)*

```
*
*  IF NDNAME IS SPECIFIED ON THE OPTIONS STATEMENT IN THE GENERATION
*  DEFINITION, STATEMENTS SIMILAR TO THE FOLLOWING ARE NEEDED.  THIS
*  WILL ALLOW YOU TO COPY YOUR NEWDEFN FILE DIRECTLY INTO A SPECIFIED
*  SUBLIBRARY.  THE CLOSE COMMAND IS NEEDED BEFORE REASSIGNMENT OF
*  SYSTEM LOGICAL UNITS.  IF THE NDF INPUT IS ON DISK, A CLOSE IS NEEDED
*  FOR SYSIPT.
*
// IF $RC GE 4 THEN
// GOTO FINISH
// DLBL IJSYSIN,'NEWDEFN',0001,SD
// EXTENT SYSIPT
// ASSGN SYSIPT,DISK,PERM,VOL=YYYYY,SHR
// EXEC LIBR,PARM='ACCESS SUBLIB=NCPLIB.NCPLOAD'
/*
CLOSE SYSIPT,00C
/. FINISH
/&
```

*Figure 25 (Part 2 of 2). Example of a FASTRUN generation (VSE)*

## Example of an NCP or PEP Generation

When running a standard NCP or PEP generation, you supply your generation definition as input and specify the various input and output files in your JCL. Figure 26 on page 156 shows the JCL used to generate NCP phases for the IBM 3725 Communication Controller. This JCL assumes that the LENAME keyword on the BUILD definition statement defaulted with the INLINKED operand. If you coded a value for this keyword, ensure that you use the same value name on the INCLUDE statement in the link-edit step in your JCL. This example includes conditional JCL to stop the job when one of the steps fails.

When reading this example, remember the following differences among the communication controllers:

- The JCL is slightly different.
- The NCP definition statement library used for the NDF job step may be different.
- The dtfname for the library of preassembled NCP object modules in the link-edit step may be different.

**IBM 3705, 3720, or 3725 Communication Controller:** When running the link-edit step for a standard NCP or PEP generation on the IBM 3705, 3725, or 3720 Communication Controller, specify the ALIGN2 option in the EXEC for the link-edit step. ALIGN2 ensures that certain control sections within the load module are aligned on 2KB page boundaries. If you do not specify ALIGN2, the default is alignment on 4KB page boundaries, which may use excessive communication controller storage.

**IBM 3745 Communication Controller:** *Do not specify* ALIGN2. The default is alignment on 4KB page boundaries, the correct alignment for the IBM 3745 Communication Controller.

The following list shows when to specify ALIGN2 as the link-edit step:

| Communication Controller | ALIGN2 |
|---|---|
| 3705 | Yes |
| 3720 | Yes |
| 3725 | Yes |
| 3745 | No. |

The following is an example of an NCP or PEP generation.

```
// JOB NDF
* ********************************************************************
*
* COPYRIGHT=NONE
*
* EXAMPLE OF AN NCP GENERATION WITH ALL LISTINGS WRITTEN TO DISK.
*
* ********************************************************************
* THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT                   *
* LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOU    *
* CHANGED YOUR LIBRARY NAMES WHEN THE PRODUCT WAS INSTALLED, YOU     *
* MAY WANT TO UPDATE THIS TABLE.  BE SURE TO CHECK THE "LIBDEF"      *
* STATEMENTS THAT DEFINE THESE LIBRARIES.                            *
*                                                                    *
* ********************************************************************
```

*Figure 26 (Part 1 of 5). Example of an NCP or PEP generation (VSE)*

```
//*                                  M O D E L
//*
//*                    3705          3725          3720          3745
//*                  ------------------------------------------------
//*       V3/3705   | MAC3705    |  NOT       |  NOT       |  NOT       |
//*                 |            |  SUPPORTED |  SUPPORTED |  SUPPORTED |
//*                 |------------------------------------------------|
//*       V3/3725   |  NOT       |  MAC3725   |  NOT       |  NOT       |
//*   V             |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |
//*   E             |------------------------------------------------|
//*   R   V4R1      |  NOT       |  MAC3725   |  NOT       |  NOT       |
//*   S             |  SUPPORTED |            |  SUPPORTED |  SUPPORTED |
//*   I             |------------------------------------------------|
//*   O   V4R2      |  NOT       |  MAC3725   |  MAC3725   |  NOT       |
//*   N             |  SUPPORTED |            |            |  SUPPORTED |
//*                 |------------------------------------------------|
//*       V4        |  NOT       |  NOT       |  MAC3725   |  NOT       |
//*       SUBSET    |  SUPPORTED |  SUPPORTED |            |  SUPPORTED |
//*                 |------------------------------------------------|
//*       V4R3 &    |  NOT       |  SNCPMAC1  |  NOT       |  NOT       |
//*       LATER     |  SUPPORTED |  SNCPMOD1  |  SUPPORTED |  SUPPORTED |
//*                 |------------------------------------------------|
//*       V5R1      |  NOT       |  NOT       |  MAC3725   |  MAC3725   |
//*                 |  SUPPORTED |  SUPPORTED |  OBJ3725   |  OBJ3725   |
//*                 |------------------------------------------------|
//*       V5R2 &    |  NOT       |  NOT       |  SNCPMAC1  |  SNCPMAC1  |
//*       LATER     |  SUPPORTED |  SUPPORTED |  SNCPMOD1  |  SNCPMOD1  |
//*                  ------------------------------------------------*
//*
//*                                  M O D E L
//*              3745-130, 3745-150,
//*              3745-170, 3745-210,
//*              3745-410
//*               ------------
//*       V3     |  NOT       |
//*              |  SUPPORTED |
//*              |------------|
//*   V   V4     |  NOT       |
//*   E          |  SUPPORTED |
//*   R          |------------|
//*   S   V5R2 & |  NOT       |
//*   I   BEFORE |  SUPPORTED |
//*   O          |------------|
//*   N   V5R2.1 |  SNCPMAC1  |
//*       & LATER|  SNCPMOD1  |
//*              ------------
//*
//*
//*
```

*Figure 26 (Part 2 of 5). Example of an NCP or PEP generation (VSE)*

```
* ****************************************************************
*
*  THIS EXAMPLE ASSUMES THAT ALL PROGRAMS RELATED TO THE NCP ARE
*  KEPT IN A SINGLE LIBRARY, NCPLIB.  THE SSP MEMBERS REQUIRED TO
*  CARRY OUT AN NCP GENERATION ARE ASSUMED TO RESIDE IN THE
*  SSPLIB SUBLIBRARY.  THE NCP PHASES ARE PLACED IN THE
*  NCPLOAD SUBLIBRARY.
*
*  A LIBDEF STATEMENT IS NEEDED TO INDICATE THE SUBLIBRARY
*  WHERE NDF AND IFZASM RESIDE.
// LIBDEF PHASE,SEARCH=(NCPLIB.SSPLIB,NCPLIB.MAC)
*
*  THIS EXAMPLE ASSUMES THAT THE GENERATION DEFINITION IS INCLUDED IN
*  LINE.  IF IT IS ON DISK, A SERIES OF JOB CONTROL STATEMENTS SIMILAR
*  TO THE FOLLOWING ARE NEEDED.
*  // DLBL IJSYSIN,'SAMPLE GENERATION'
*  // EXTENT SYSIPT
*  // ASSGN SYSIPT,DISK,PERM,VOL=DSKID,SHR
*
*  IF NEWDEFN=YES OR NEWDEFN=(YES,ECHO) IS SPECIFIED IN THE
*  GENERATION DEFINITION, JOB CONTROL STATEMENTS SIMILAR TO
*  THE FOLLOWING ARE NEEDED.
*  // DLBL IJSYSNW,'NEWDEFN',0001,SD
*  // EXTENT SYS001,YYYYYY,,,3201,40
*  // ASSGN SYS001,DISK,PERM,VOL=YYYYYY,SHR
*
*  THE SYSPCH FILE IS USED AS THE PUNCH CODE OUTPUT FILE BY NDF.
*  THIS FILE WILL BE USED AS THE INPUT FILE FOR THE TABLE ASSEMBLIES.
// DLBL IJSYSPH,'TMP FILE',0001,SD
// EXTENT SYSPCH,YYYYYY,,,1501,1200
// ASSGN SYSPCH,DISK,PERM,VOL=YYYYYY,SHR
*
*  THE LISTING IS SENT TO A PRINTER
// ASSGN SYSLST,00E,PERM
*
*  THE STORAGE MANAGER WORK FILE IS NEEDED ONLY WHEN THERE IS NOT
*  ENOUGH VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
*  USERGEN IS SPECIFIED.
*  // DLBL DBWRKFL,'VSAM.WORK',0,VSAM
*
*  NDF IS EXECUTED WITH SIZE=AUTO SO THAT ANY EXTRA VIRTUAL
*  MEMORY WILL BE AVAILABLE IN THE GETVIS AREA
// EXEC ICNDNDF,SIZE=AUTO,PARM='LINECNT=55'
*
```

*Figure 26 (Part 3 of 5). Example of an NCP or PEP generation (VSE)*

```
*  PLACE YOUR GENERATION INPUT FILE HERE
*
/*
// IF $RC GE 4 THEN
// GOTO CLSPCH
*
*  THE CLOSE COMMAND IS NEEDED BEFORE REASSIGNMENT OF SYSTEM
*  LOGICAL UNITS.  IF THE NDF INPUT FILE IS ON DISK, A CLOSE IS
*  ALSO NEEDED FOR SYSIPT.
CLOSE SYSPCH,UA
*
*  IF NDNAME IS SPECIFIED ON THE OPTIONS STATEMENT IN THE GENERATION
*  DEFINITION, STATEMENTS SIMILAR TO THE FOLLOWING SEVEN STATEMENTS ARE
*  NEEDED.  THIS WILL ALLOW YOU TO COPY YOUR NEWDEFN FILE DIRECTLY
*  INTO A SPECIFIED SUBLIBRARY.  THE CLOSE COMMAND IS NEEDED BEFORE
*  REASSIGNMENT OF SYSTEM LOGICAL UNITS.  IF THE NDF INPUT IS ON DISK, A
*  CLOSE IS NEEDED FOR SYSIPT.
*
// DLBL IJSYSIN,'NEWDEFN',0001,SD
// EXTENT SYSIPT
// ASSGN SYSIPT,DISK,PERM,VOL=YYYYY,SHR
// EXEC LIBR,PARM='ACCESS SUBLIB=NCPLIB.NCPLOAD'
/*
// IF $RC GT 0 THEN
// GOTO CLSIPT
CLOSE SYSIPT,UA
*
*  DEFINE THE INPUT FILE FOR ALL THREE ASSEMBLIES
// DLBL IJSYSIN,'TMP FILE'
// EXTENT SYSIPT
// ASSGN SYSIPT,DISK,PERM,VOL=YYYYYY,SHR
*
*  DEFINE THE PUNCH CODE OUTPUT FILE FOR ALL THREE ASSEMBLIES
// DLBL IJSYSPH,'TABLE TEXT',0001,SD
// EXTENT SYSPCH,YYYYYY,,,2701,500
// ASSGN SYSPCH,DISK,PERM,VOL=YYYYYY,SHR
*
*  DEFINE THE NCP MACRO LIBRARY
// LIBDEF SOURCE,SEARCH=(NCPLIB.SNCPMAC1)
*
*  THE DECK OPTION IS REQUIRED TO GET THE PROPER OUTPUT FROM THE
*  TABLE ASSEMBLIES.  THE SXREF OPTION WILL PROVIDE AN ADEQUATE
*  CROSS REFERENCE
// OPTION DECK,SXREF
// EXEC IFZASM
// IF $RC GT 4 THEN
// GOTO CLSBOTH
// EXEC IFZASM
// IF $RC GT 4 THEN
// GOTO CLSBOTH
// EXEC IFZASM
// IF $RC GT 0 THEN
// GOTO CLSBOTH
CLOSE SYSIPT,UA
CLOSE SYSPCH,00D
```

*Figure 26 (Part 4 of 5). Example of an NCP or PEP generation (VSE)*

```
*  THE LIBRARIAN MUST BE USED TO CATALOG THE TABLE TEXT FILES
*  INTO OBJ MEMBERS OF THE LIBRARY
*
*  THE PUNCH CODE OUTPUT FILE FROM THE TABLE ASSEMBLIES SERVES
*  AS THE INPUT FILE FOR THE LIBRARIAN JOB
// DLBL IJSYSIN,'TABLE TEXT'
// EXTENT SYSIPT
// ASSGN SYSIPT,DISK,PERM,VOL=YYYYYY,SHR
// EXEC LIBR,PARM='ACCESS SUBLIB=NCPLIB.NCPLOAD'
// IF $RC GT 0 THEN
// GOTO CLSIPT
*
*  A LIBDEF STATEMENT MUST BE USED TO DEFINE THE SEARCH CHAIN
*  FOR THE INPUT TO THE LINK EDIT STEP.
*  DEFINE THE NCP OBJECT LIBRARY
// LIBDEF OBJ,SEARCH=(NCPLIB.SNCPMOD1,NCPLIB.NCPLOAD)
*
*  A LIBDEF STATEMENT MUST ALSO BE USED TO DEFINE THE SUBLIBRARY
*  WHERE THE LINK EDIT OUTPUT PHASES WILL BE CATALOGED
// LIBDEF PHASE,CATALOG=NCPLIB.NCPLOAD
*
// OPTION CATAL
   ACTION MAP,NOAUTO
   INCLUDE INLINKED
// EXEC LNKEDT
// GOTO CLSIPT
*  CLOSE DIFFERENT COMBINATIONS OF SYSIPT AND SYSPCH
*  DEPENDING ON WHERE THE JOB TERMINATED
/. CLSPCH
CLOSE SYSPCH,00D
// GOTO FINISH
/. CLSBOTH
CLOSE SYSPCH,00D
/. CLSIPT
CLOSE SYSIPT,00C
/. FINISH
/&
```

*Figure 26 (Part 5 of 5). Example of an NCP or PEP generation (VSE)*

# Example of an NCP or PEP Generation with User-Written Code Using the NDF Standard Attachment Facility

To run an NCP or PEP generation with IBM special products or user-written code using the NDF standard attachment facility, you can generate user-written code by providing user-written generation applications. These applications use the NDF standard attachment facility to process and pass statements and keywords to NDF during generation processing.

Figure 27 on page 161 shows the JCL for generating user-written code and NCP using the NDF standard attachment facility. For more information about running this type of generation, see page 145. You can run this type of user-written code generation *only* if you are using SSP V3R4 or a later release to generate NCP Version 4 or NCP V5R2 or a later release.

Before you generate user-written code using the NDF standard attachment facility, do the following:

- Code the USERGEN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. The USERGEN keyword specifies the names of the user-written generation phases to be loaded in the generation. Each application must have its own generation phase. You can specify up to 25 generation phases.
- Code the NEWDEFN keyword on the OPTIONS definition statement as the first executable statement in your generation definition. NEWDEFN enables NDF to create a new generation definition consisting of the input NCP generation definition and the NCP statements and keywords passed to NDF from any user-written generation phases.
- Modify the JCL for a standard NCP or PEP generation to include the dtfnames for the IJSYSNW file, the DBWRKFL file, and the libraries for user-supplied modules.

The following are examples of how to code the IJSYSNW file in the NDF step of the JCL and how to code the files for user-written code modules in the link-edit step of the JCL.

```
*EXAMPLE FOR INCLUDING NEWDEFN IN THE NDF STEP IN THE JCL
*IF NEWDFN IS SPECIFIED
*
//DLBL IJSYSNW,'NEWDEFN',0000,SD
//EXTENT SYS001,DOSRES,,,10020,10
//ASSGN SYS001,150,TEMP



*EXAMPLE LIBDEF STATEMENT FOR USER LOAD MODULES IF USERGEN
*IS SPECIFIED
*
//LIBDEF PHASE,SEARCH(NCPLIB,SSPLIB,USER.LIB,NCPLIB.PR$AM2)



*EXAMPLE LIBDEF STATEMENT FOR USER MACRO LIBRARY IN TABLE
*ASSEMBLY STEPS
*
//LIBDEF SOURCE,SEARCH=(NCPLIB.SNCPMAC1,USER.MACLIB)



*EXAMPLE LIBDEF STATEMENT FOR USER MODULES IN LINK EDIT STEP
*
//LIBDEF OBJ,SEARCH=(NCPLIB.SNCPMOD1,NCPLIB.NCPLOAD,USER.OBJLIB)
```

*Figure 27. Example of an NCP or PEP generation with user-written code using the NDF standard attachment facility (VSE)*

# Example of an NCP or PEP Generation with User-Written Code Using the GENEND Definition Statement

To run an NCP or PEP generation with IBM special products or user-written code without using the NDF standard attachment facility, or to generate user-written code using SSP Version 3, you must code link-edit statements and CSECTs for your user routine. You must also identify the location of the link-edit statements by coding keywords on the GENEND definition statement.

Figure 28 on page 163 shows the JCL for generating IBM special products or user-written code using the GENEND definition statement. For more information about running this type of generation, see page 145. You can run this type of user-written code generation if you are using SSP Version 3 to generate any version of NCP.

Before you generate IBM special products or user-written code using the GENEND definition statement, ensure that you:

- Run any job required to generate SRCLO or SRCHI code
- Catalog the members with SRCLO or SRCHI code members of type *A*
- Edit and catalog any definition statements called by the SRCLO or SRCHI code as members of type *F*
- Catalog link-edit control statements and preassembled object code as members of type *OBJ*
- Add sublibraries that contain source code or edited definition statements to the LIBDEF chain for SOURCE (establish this LIBDEF chain before the table assemblies call IFZASM)
- Add sublibraries that contain link-edit control statements to the LIBDEF chain for OBJ (establish this LIBDEF before you call the linkage editor).

The following is an example of how to specify the table assembly search chain and the link-edit search chain in the JCL for an NCP or PEP generation with user-written code using the GENEND definition statement.

**Note:** In this example, library names such as SNCPMAC1 and SNCPMOD1 are release dependent.

```
*
*  COPYRIGHT=NONE
*
*  LIBDEF FOR THE TABLE ASSEMBLY SEARCH CHAIN
*
*  USER SOURCE CODE HAS BEEN CATALOGED IN D TYPE MEMBERS OF
*  THE USERMAC SUBLIB
*
*  EDITED MACROS CALLED FROM THE USER SOURCE CODE HAVE BEEN
*  CATALOGED IN F TYPE MEMBERS OF THE USERMAC SUBLIBRARY
*
// LIBDEF SOURCE,SEARCH=(NCPLIB.SNCPMAC1,NCPLIB.USERMAC)
*
*
*  LIBDEF FOR THE LINK EDIT SEARCH CHAIN
*
*  MEMBERS WITH INCLUDE STATEMENTS FOR USER CODE HAVE BEEN CATALOGED
*  IN OBJ TYPE MEMBERS OF THE USEROBJ SUBLIBRARY.  THE NAMES OF
*  THESE MEMBERS WERE CODED FOR ONE OF THE "INCxxx" KEYWORDS ON
*  THE GENEND STATEMENT
*
*  THE INCLUDED PREASSEMBLED USER OBJECT MODULES HAVE ALSO BEEN
*  CATALOGED IN OBJ TYPE MEMBERS IN THE USEROBJ SUBLIBRARY
*
// LIBDEF OBJ,SEARCH=(NCPLIB.SNCPMOD1,NCPLIB.USEROBJ)
```

*Figure 28. Example of an NCP or PEP generation with user-written code using the GENEND definition statement (VSE)*

# Example of a Dynamic Reconfiguration Generation

To modify an NCP already running in a communication controller, you can use the text file from a dynamic reconfiguration generation. Figure 29 on page 164 shows the JCL for dynamic reconfiguration generation.

To use this type of generation, ensure that you coded the original NCP to allow dynamic reconfiguration. The dynamic reconfiguration generation produces a text file that the access method can use to modify NCP.

**Note:** VTAM has its own dynamic reconfiguration procedures that do not require you to use NDF and the dynamic reconfiguration generation. For more information on dynamic reconfiguration for VTAM, see *VTAM Installation and Resource Definition.*

To dynamically reconfigure your NCP, you must define a dynamic reconfiguration file consisting of ADD or DELETE definition statements, or both, and their associated PU and LU definition statements. The dynamic reconfiguration file is the input for the dynamic reconfiguration generation. This type of generation produces a text file that the access method uses to modify an NCP already running in a communication controller. For information on using ADD, DELETE, PU, or LU definition statements, see *NCP, SSP, and EP Resource Definition Guide.*

A dynamic reconfiguration generation requires one table assembly and no link-edit. The following is an example of JCL for a dynamic reconfiguration generation.

```
// JOB NDF
* ********************************************************************
*
* COPYRIGHT=NONE
*
* EXAMPLE OF A DYNAMIC RECONFIGURATION GENERATION.
*
* ********************************************************************
* THE FOLLOWING TABLE SHOWS WHICH MACRO AND OBJECT                   *
* LIBRARIES CORRESPOND TO A PARTICULAR MODEL AND VERSION.  IF YOU    *
* CHANGED YOUR LIBRARY NAMES WHEN THE PRODUCT WAS INSTALLED, YOU     *
* MAY WANT TO UPDATE THIS TABLE.  BE SURE TO CHECK THE "LIBDEF"      *
* STATEMENTS THAT DEFINE THESE LIBRARIES.                            *
*                                                                    *
```

*Figure 29 (Part 1 of 4). Example of a dynamic reconfiguration generation (VSE)*

```
* ********************************************************************
*                                                                    *
//*                                  M O D E L
//*
//*                 3705          3725          3720          3745
//*               ------------------------------------------------
//*      V3/3705| MAC3705    |  NOT       |  NOT       |  NOT       |
//*             |            |  SUPPORTED |  SUPPORTED |  SUPPORTED |
//*             |------------------------------------------------|
//*      V3/3725| NOT        |  MAC3725   |  NOT       |  NOT       |
//*   V         | SUPPORTED  |            |  SUPPORTED |  SUPPORTED |
//*   E         |------------------------------------------------|
//*   R  V4R1   | NOT        |  MAC3725   |  NOT       |  NOT       |
//*   S         | SUPPORTED  |            |  SUPPORTED |  SUPPORTED |
//*   I         |------------------------------------------------|
//*   O  V4R2   | NOT        |  MAC3725   |  MAC3725   |  NOT       |
//*   N         | SUPPORTED  |            |            |  SUPPORTED |
//*             |------------------------------------------------|
//*      V4     | NOT        |  NOT       |  MAC3725   |  NOT       |
//*      SUBSET | SUPPORTED  |  SUPPORTED |            |  SUPPORTED |
//*             |------------------------------------------------|
//*      V4R3 & | NOT        |  SNCPMAC1  |  NOT       |  NOT       |
//*      LATER  | SUPPORTED  |  SNCPMOD1  |  SUPPORTED |  SUPPORTED |
//*             |------------------------------------------------|
//*      V5R1   | NOT        |  NOT       |  MAC3725   |  MAC3725   |
//*             | SUPPORTED  |  SUPPORTED |  OBJ3725   |  OBJ3725   |
//*             |------------------------------------------------|
//*      V5R2 & | NOT        |  NOT       |  SNCPMAC1  |  SNCPMAC1  |
//*      LATER  | SUPPORTED  |  SUPPORTED |  SNCPMOD1  |  SNCPMOD1  |
//*             ------------------------------------------------*
//*
//*                                  M O D E L
//*            3745-130, 3745-150,
//*            3745-170, 3745-210,
//*            3745-410
//*             ------------
//*      V3     |  NOT       |
//*             |  SUPPORTED |
//*             |------------|
//*   V  V4     |  NOT       |
//*   E         |  SUPPORTED |
//*   R         |------------|
//*   S  V5R2 & |  NOT       |
//*   I  BEFORE |  SUPPORTED |
//*   O         |------------|
//*   N  V5R2.1 |  SNCPMAC1  |
//*      & LATER|  SNCPMOD1  |
//*             ------------
//*
//*
//*
*                                                                    *
* ********************************************************************
```

*Figure 29 (Part 2 of 4). Example of a dynamic reconfiguration generation (VSE)*

```
*
*  THIS EXAMPLE ASSUMES THAT ALL PROGRAMS RELATED TO THE NCP ARE
*  KEPT IN A SINGLE LIBRARY, NCPLIB.  THE SSP MEMBERS REQUIRED TO
*  CARRY OUT AN NCP GENERATION ARE ASSUMED TO RESIDE IN THE
*  SSPLIB SUBLIBRARY.  THE NCP PHASES ARE PLACED IN THE
*  NCPLOAD SUBLIBRARY.
*
*  A LIBDEF STATEMENT IS NEEDED TO INDICATE THE SUBLIBRARY
*  WHERE NDF AND IFZASM RESIDE.  THE SUBLIBRARY WHERE VSAM PHASES
*  RESIDE MUST ALSO BE PART OF THE SEARCH CHAIN IF THE
*  WORK FILE (DBWRKFL) IS TO BE USED.
// LIBDEF PHASE,SEARCH=(NCPLIB.SSPLIB,NCPLIB.PR$AM2)
*
*  THIS EXAMPLE ASSUMES THAT THE GENERATION DEFINITION IS INCLUDED IN
*  LINE.  IF IT IS ON DISK, A SERIES OF JOB CONTROL STATEMENTS SIMILAR
*  TO THE FOLLOWING ARE NEEDED.
*  // DLBL IJSYSIN,'SAMPLE GENERATION'
*  // EXTENT SYSIPT
*  // ASSGN SYSIPT,DISK,PERM,VOL=DSKID,SHR
*
*  THE SYSPCH FILE IS USED AS THE PUNCH CODE OUTPUT FILE BY NDF.
*  THIS FILE WILL BE USED AS THE INPUT FILE FOR THE TABLE ASSEMBLY.
// DLBL IJSYSPH,'TMP FILE',0001,SD
// EXTENT SYSPCH,YYYYYY,,,1501,1200
// ASSGN SYSPCH,DISK,PERM,VOL=YYYYYY,SHR
*
*  THE LISTING IS SENT TO A PRINTER
// ASSGN SYSLST,00E,PERM
*
*  THE STORAGE MANAGER WORK FILE IS NEEDED ONLY WHEN THERE IS NOT
*  ENOUGH VIRTUAL MEMORY TO HOLD ALL OF NDF'S WORK DATA OR IF
*  USERGEN IS SPECIFIED.
*  // DLBL DBWRKFL,'VSAM.WORK',0,VSAM
*
*  NDF IS EXECUTED WITH SIZE=AUTO SO THAT ANY EXTRA VIRTUAL
*  MEMORY WILL BE AVAILABLE IN THE GETVIS AREA
// EXEC ICNDNDF,SIZE=AUTO,PARM='LINECNT=55'
*
*  PLACE YOUR DYNAMIC RECONFIGURATION SOURCE HERE
*
/*
// IF $RC GE 4 THEN
// GOTO CLSPCH
*
*  THE CLOSE COMMAND IS NEEDED BEFORE REASSIGNMENT OF SYSTEM
*  LOGICAL UNITS.  IF THE NDF INPUT FILE IS ON DISK, A CLOSE IS
*  ALSO NEEDED FOR SYSIPT.  SYSPCH IS ASSIGNED TO A PUNCH SO
*  THAT THE DR TEXT WILL APPEAR AS PUNCHED OUTPUT.
CLOSE SYSPCH,00D
```

*Figure 29 (Part 3 of 4). Example of a dynamic reconfiguration generation (VSE)*

```
*
*  DEFINE THE INPUT FILE FOR THE ASSEMBLY.
// DLBL IJSYSIN,'TMP FILE'
// EXTENT SYSIPT
// ASSGN SYSIPT,DISK,PERM,VOL=YYYYYY,SHR
*
*
*  DEFINE THE NCP MACRO LIBRARY
// LIBDEF SOURCE,SEARCH=(NCPLIB.SNCPMAC1)
*
*  THE DECK OPTION IS REQUIRED TO GET THE PROPER OUTPUT FROM THE
*  TABLE ASSEMBLIES.  THE SXREF OPTION WILL PROVIDE AN ADEQUATE
*  CROSS REFERENCE.
// OPTION DECK,SXREF
// EXEC IFZASM
CLOSE SYSIPT,00C
// GOTO FINISH
/. CLSPCH
CLOSE SYSPCH,00D
/. FINISH
/&
```

*Figure 29 (Part 4 of 4). Example of a dynamic reconfiguration generation (VSE)*

# Chapter 9. Loading the Program under VSE

The last step in producing an operating NCP is to load the phases into the communication controller where they will reside. You can load your NCP into a channel-attached communication controller in two ways. You can use the loader utility provided by SSP, or you can use a loader facility provided by an access method. This chapter tells you how to use the SSP loader utility. For information on how to use the access-method loader facility to load both a channel-attached and link-attached communication controller, see the *VTAM Network Implementation Guide*.

The SSP loader utility is run as a VSE job or job step. A communication controller module disables all channel adapters except the one over which the load operation takes place. When NCP completes its initialization step, it enables any additional channel adapters specified as ACTIVE in the NCPCA keyword. EP enables any additional channel adapters with HICHAN and LOCHAN coded in a PEP phase.

You must manually disable any channel adapter connected to a nonoperational host before starting the load process. Messages sent to the message logical unit indicate syntax errors or permanent input or output errors occurring during loading.

If you are loading your NCP into the IBM 3720 using SSP V3R2 or a later release or into the IBM 3745 Communication Controller using SSP V3R4 or a later release, you can load the NCP phases from the host and save them on the MOSS disk. You can then later reload the NCP phases from this MOSS disk.

If you are loading your NCP into the IBM 3705 Communications Controller, you can load an optional diagnostic routine called the *initial test routine* before the loader utility loads NCP into the communication controller. If the initial test routine detects no malfunctions, the loader utility loads NCP into the communication controller. However, if the initial test routine detects trouble, it stops, and the loader utility issues an error message (IFL004I). The loader utility then loads any remaining communication controllers specified in the loader job. Loading and running the initial test routine are optional but recommended steps because this routine can detect conditions that can later cause NCP failure. The initial test routine is run unless you specify its omission in the LOAD control statement.

**Note:** If the communication controller's power has been turned off since the last time you ran the initial test routine, run the routine again before reloading the communication controller. This ensures that correct parity is set in the communication controller's storage.

## Loader Utility

This section discusses the following about the VSE loader utility:

- Host processor and communication controller requirements
- Input to the loader utility
- Output from the loader utility.

## Host Processor and Communication Controller Requirements

You can run the loader utility's communication controller module in any channel-attached communication controller. Before you can load the loader utility, you must ensure the communication controller:

- Has its power on
- Is identified to the VSE system where you plan to run the loader utility
- Is free so you can allocate it to the loader job step
- Is not in a program-stop condition
- Has the channel online that attaches it to the operating system
- Has enabled the channel adapter where the load is to occur.

The loader utility consists of the phase IFULOAD. For SSP V3R2 and later releases, it also contains the phase IFWLEVEL. No work files are required to run the loader utility.

SSP V3R4 and later releases use the GETVIS region when loading NCP phases. You will want to maximize the GETVIS area to accommodate the NCP phases.

## Input to the Loader Utility

For SSP V3R2 and earlier releases, the input to the loader utility consists of two files. One, a DASD input file, contains the NCP phases to be loaded into the communication controller. The other contains a LOAD statement specifying the NCP phase to be loaded from the host or the MOSS disk and the communication controller where it will be loaded.

For SSP V3R4 and later releases, the input consists of the generated NCP phases and the LOAD statement. You can use the optional DASD file as backup if not enough GETVIS area is available to accommodate the phases or if the phases do not exist in the search library.

If you are loading your NCP into the IBM 3705 Communications Controller, you can include an optional file as input to the loader utility. This additional file contains the initial test routine and consists of phases IFU3705D and IFU3705E. If you do not want to run the initial test routine, you can omit this file.

## Output from the Loader Utility

The loader utility produces one output logical unit, the message logical unit SYSLST. This logical unit contains completion or error messages produced by the loader utility. See *NCP, SSP, and EP Messages and Codes* for a description of the messages issued by the loader utility.

# Controlling the Loader Utility

This section discusses examples of the job control statements and the utility control statement that you supply to the loader utility. It also explains how to punch NCP phases onto the disk using the LIBRARIAN.

## Punching Phases Using the LIBRARIAN

Before you load using the SSP loader utility, you may need to punch the NCP phases in the correct order onto the disk using the LIBRARIAN. This procedure is required if you are using SSP V3R2 or an earlier release but is optional if you are using SSP V3R4 or a later release.

In SSP V3R4 and later releases, the NCP phases can be loaded into the GETVIS region. If the GETVIS region is not large enough to accommodate the phases, you can increase this area or punch the phases as described below.

The following is an example of a job that moves the NCP phases for an IBM 3720 or 3725 Communication Controller from the NCPLIB.NCPLOAD sublibrary onto the disk. As with all communication controllers, do not punch the resource resolution table (RRT) and block-handler resolution table (BHR) phases. See "Naming Phases" on page 143 for more information.

```
// JOB     INITTEST
*  DEFINE SYSTEM PUNCH TO DISK LOCATION WHERE PHASES WILL BE STORED
// DLBL    IJSYSPH, other parameters
// EXTENT  SYSPCH, other parameters
*  SPECIFY PUNCH UNIT ADDRESS
// ASSGN   SYSPCH,X'xxx'
*  INVOKE LIBRARIAN TO PUNCH PHASES
// EXEC    LIBR
   ACCESS  S=NCPLIB.NCPLOAD
   PUNCH   VSE8.PHASE,VSE80002.PHASE,VSE80003.PHASE,-
           VSE80004.PHASE,VSE80005.PHASE,VSE80006.PHASE
/*
   CLOSE   SYSPCH,X'xxx'
/&
```

**Note:** For an IBM 3745 Communication Controller, the PUNCH statement in the preceding example is different. Code the PUNCH statement as follows:

```
   PUNCH   VSE8.PHASE,VSE80002.PHASE,VSE80003.PHASE,-
           VSE80004.PHASE,VSE80005.PHASE,VSE80006.PHASE -
           VSE80007.PHASE
```

If you are loading NCP into the IBM 3705 Communications Controller, you must also move the phases that make up the initial test routine. Depending on the relationship between storage of user-written code, keywords specified, and block handlers coded, the number of phases will vary between one and three. See Table 17 on page 143 for more information.

The following is an example of a job that moves the phases IFU3705D and IFU3705E from the NCPLIB.NCPLOAD sublibrary onto the disk:

```
// JOB     INITTEST
*  DEFINE SYSTEM PUNCH TO DISK LOCATION WHERE PHASES WILL BE STORED
// DLBL    IJSYSPH, other parameters
// EXTENT  SYSPCH, other parameters
*  SPECIFY PUNCH UNIT ADDRESS
// ASSGN   SYSPCH,X'xxx'
*  INVOKE LIBRARIAN TO PUNCH PHASES
// EXEC    LIBR
   ACCESS  S=NCPLIB.NCPLOAD
   PUNCH   IFU3705D.PHASE,IFU3705E.PHASE
/*
   CLOSE   SYSPCH,X'xxx'
/&
```

## Job Control Statements

The job control statements you need for calling the loader utility are shown in Table 19.

*Table 19 (Page 1 of 2). Job control statements for loader utility (VSE)*

| | |
|---|---|
| // JOB | Starts the job. |
| // ASSGN | Specifies the unit address of the communication controller to be loaded. You can omit this statement if there is a permanent assignment for the communication controller. |
| // DLBL | Defines a sequential file that contains suitably formatted phases. Use this statement with SSP V3R2 and earlier releases. |
| // EXTENT | Use this statement with SSP V3R2 and earlier releases. |
| // ASSGN | Assigns the file defined in the previous DLBL and EXTENT statements. Use this statement with SSP V3R2 and earlier releases. |
| // LIBDEF | Specifies the locations of the loader utility for all versions and specifies the NCP phases for SSP V3R4 and later releases. |
| // DLBL | DIAGFLE,'file-id'<br>(IBM 3705 Communications Controller only.) Defines the sequential file that contains the initial test routine; it is not required if you specify DIAG=NO on the LOAD statement. |
| // EXTENT | SYS008,vol.id,1<br>(IBM 3705 Communications Controller only.) It is required only if you specify or imply DIAG=Y6 or Y8 on any LOAD statement. |

*Table 19 (Page 2 of 2). Job control statements for loader utility (VSE)*

| | |
|---|---|
| // ASSGN | SYS008,X'ccu'<br><br>(IBM 3705 Communications Controller only.) Assigns the file defined in the previous DLBL and EXTENT statements; it is required only if you specify or imply DIAG=Y6 or Y8 on any LOAD statement. |
| // EXEC | Specifies the program name, IFULOAD. |

## Utility Control Statement

The loader utility requires only one utility control statement, the LOAD statement. It specifies:

- The name of the input file that contains the NCP load module to be loaded or the name of the NCP phases
- The name of the load module to be loaded from the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller
- The communication controller to be loaded
- Whether or not you want to run the initial test routine, if loading your NCP into an IBM 3705 Communications Controller
- Whether or not you want to save the load module on the MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller
- Whether or not you want to request automatic initial program load (IPL), using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller.

The following conventions are used to describe the LOAD statement:

- Capital letters represent values you code directly without change.
- Lowercase letters represent parameters for which you must supply a value.
- Braces { } indicate you must choose from the enclosed items.
- *Or* signs | indicate you can choose between various keywords.
- An underlined value represents the default value of the keyword (that is, the loader utility uses that value if you omit the keyword).
- Brackets [ ] enclose keywords or symbols that are either optional or conditional.

The format of the LOAD statement is:

```
LOAD        LOADMOD={file name }
                    {(D)name   }
                    {phase name}
          UNIT=sysxxx|3725=sysxxx|3705=sysxxx
            [,AUTOIPL={NO }]
                      {YES}
            [,DIAG={Y6}]
                   {Y8}
                   {NO}
            [,SAVE={NO }]
                   {YES}
```

**LOADMOD**=*{file name }*
*{*(D)*name }*
*{phase name}*

Identifies the NCP load modules and phases.

*file name*
Specifies the name of the file that contains the NCP load module. For SSP V3R2 and earlier releases, this name must be the same as the file specified in the DLBL statement.

**(D)***name*
Specifies the name of the load module to be loaded from a MOSS disk, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller.

*phase name*
Specifies, for SSP V3R4 and later releases, the name given to the phases at generation time as specified on the NEWNAME keyword. These phases must be present in the phase library indicated by the LIBDEF phase.

**UNIT**=*sysxxx*|**3725**=*sysxxx*|**3705**=*sysxxx*

Specifies the symbolic name of the communication controller to be loaded. Use Table 20 to determine which keyword to code.

*Table 20. Keywords for the UNIT control statement (VSE)*

| Communication Controller | SSP Release | Keyword |
|---|---|---|
| 3745 | SSP V3R4 and later | UNIT=SYSxxx |
| 3720 | SSP V3R2 and later | UNIT=SYSxxx |
| 3725 | SSP V3R2 and later | UNIT=SYSxxx or 3725=SYSxxx |
| 3705 | SSP V3R2 and later | UNIT=SYSxxx or 3705=SYSxxx |
| 3725 | SSP V3R1 | 3725=SYSxxx |
| 3705 | SSP V3R1 | 3705=SYSxxx |

```
[,AUTOIPL={NO }]
          {YES}
```

Specifies whether or not you want to request automatic IPL from the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller. If you specify AUTOIPL=YES, automatic dump is also assumed. When an abend occurs, the dump in communication controller's storage is automatically stored on the MOSS disk and an automatic IPL is initiated from the MOSS disk.

**Note:** Only one dump can exist on the disk. If there is already a dump on the disk when NCP abends, the communication controller will neither dump nor automatically IPL.

```
[,DIAG={Y6}]
       {Y8}
       {NO}
```

For the IBM 3705 Communications Controller, specifies whether the loader utility is to load the initial test routine.

**Y6**
Is the default and specifies that the routine be loaded into a communication controller without extended addressing. An IBM 3705-II Communication Controller having 64KB or less of storage uses 16-bit storage addresses and, therefore, does not require extended addressing.

**Y8**
Specifies that the routine be loaded into a communication controller with extended addressing. An IBM 3705-II Communication Controller having more than 64KB of storage requires extended storage addressing.

**NO**
Specifies that the initial routine not be loaded.

```
[,SAVE={NO }]
       {YES}
```

Specifies whether or not you want to save the phases from communication controller storage on the MOSS disk when loading, using SSP V3R2 or a later release for the IBM 3720 Communication Controller or SSP V3R4 or a later release for the IBM 3745 Communication Controller. Specifying SAVE=YES is not valid with LOADMOD={(D)name}.

## Examples of Job and Utility Control Statements

The following are examples of statements that load NCP into different communication controllers. Since these are only examples, you must modify them to fit your particular system. To load into the IBM 3720 Communication Controller, you must use SSP V3R2 or a later release; to load into the IBM 3745 Communication Controller, you must use SSP V3R4 or a later release. To use disk support with any of these communication controllers, you must use SSP V3R4 or a later release.

## Example 1. Loading into the IBM 3720 or 3745 Communication Controller with Disk Support

Assume you want to load NCP phases named NCP3MOD into an IBM 3720 or 3745 Communication Controller with disk support and a unit address of 001. These phases are in the NCPLIB.NCPLOAD library, and you have punched them to a file also named NCP3MOD (an optional step). In addition, you want to save the phases onto the MOSS disk, and you want automatic IPL from this disk. To load NCP3MOD, use the following job and utility statements:

```
// JOB LOADUNIT
// ASSGN SYS007,X'001'
* THE NEXT 3 LINES DEFINE THE PUNCHED FILE THAT WILL BE LOADED
* IF THERE IS NOT ENOUGH GETVIS STORAGE FOR THE PHASES OR IF THE
* PHASES ARE NOT FOUND IN THE LIBRARIES SPECIFIED IN THE SEARCH LIBDEF
// DLBL NCP3MOD,'NCPFILE'
// EXTENT SYS005,111111
// ASSGN SYS005,X'131'
// LIBDEF PHASE,SEARCH=(SSPLIB.LOADER,NCPLIB.NCPLOAD)
// EXEC IFULOAD
LOAD LOADMOD=NCP3MOD,UNIT=SYS007,SAVE=YES,AUTOIPL=YES
/*
/&
```

In the preceding example, you can leave out AUTOIPL or specify AUTOIPL=NO and then later, if desired, specify AUTOIPL=YES on a separate LOAD statement.

When you want to load the saved NCP phases from the MOSS disk, issue the following load statement:

```
LOAD LOADMOD=(D)NCP3MOD,UNIT=SYS007
```

Because you did not specify AUTOIPL=YES, the default setting of NO caused the value of AUTOIPL to be reset.

## Example 2. Loading into the IBM 3720 or 3725 Communication Controller

For SSP V3R4 and later releases, assume you want to load NCP phases named NCP3MOD into an IBM 3720 or 3725 Communication Controller with a unit address of 001. These phases are in the NCPLIB.NCPLOAD library, and you have punched them to a file also named NCP3MOD (an optional step). To load NCP3MOD, use the following job and utility statements:

```
// JOB LOADUNIT
// ASSGN SYS007,X'001'
* THE NEXT 3 LINES DEFINE THE PUNCHED FILE THAT WILL BE LOADED
* IF THERE IS NOT ENOUGH GETVIS STORAGE FOR THE PHASES.
// DLBL NCP3MOD,'NCPFILE'
// EXTENT SYS005,111111
// ASSGN SYS005,X'131'
// LIBDEF PHASE,SEARCH=(SSPLIB.LOADER,NCPLIB.NCPLOAD)
// EXEC IFULOAD
LOAD LOADMOD=NCP3MOD,UNIT=SYS007
/*
/&
```

### Example 3. Loading into the IBM 3705 Communications Controller

For SSP V3R4 and later releases, assume you want to load NCP phases named NCP3MOD into an IBM 3705 Communications Controller. This communication controller has a unit address of 001 and is located in the NCPLIB.NCPLOAD library. You want to run an initial test routine named INITTEST (residing in a file named DIAGFLE), and you have punched these phases to a file also named NCP3MOD (an optional step). To load NCP3MOD, use the following job and utility statements:

```
// JOB LOADUNIT
// ASSGN SYS007,X'001'
// DLBL DIAGFLE,'INITTEST'
// EXTENT SYS008,111111
// ASSGN SYS008,X'131'
* THE NEXT 3 LINES DEFINE THE PUNCHED FILE THAT WILL BE LOADED
* IF THERE IS NOT ENOUGH GETVIS STORAGE FOR THE PHASES.
// DLBL NCP3MOD,'NCPFILE'
// EXTENT SYS005,111111
// ASSGN SYS005,X'131'
// LIBDEF PHASE,SEARCH=(SSPLIB.LOADER,NCPLIB.NCPLOAD)
// EXEC IFULOAD
LOAD LOADMOD=NCP3MOD,UNIT=SYS007, DIAG=Y8
/*
/&
```

### Example 4. Link-Editing Object Code into Phases

If the host processor phases of the loader utility are cataloged as object code in the sublibrary, you can use the following control statements to link-edit them into phases in the sublibrary:

```
// JOB LINKLOAD
// OPTION CATAL
// LIBDEF OBJ,SEARCH=(SSPLIB.LOADER)
// LIBDEF PHASE,CATALOG=(SSPLIB.LOADER),PERM
INCLUDE IFULINK
// EXEC LNKEDT
/*
/&
```

In addition to the above JCL, include LIBDEF statements to tell the program where to locate objects and into which file to place phases.

## Generating and Loading a Remote Communication Controller with a Floppy Disk

This section describes how to create a floppy disk and load it into a remote communication controller. It includes information on how to prepare a phase, download the phase to a floppy disk, and upload the phase from a floppy disk to a MOSS disk. For more information, see *3745 Models 130/150/170 Advanced Operations* and *3720/3721 Operator's Guide*.

## Phase Preparation

You must first generate a phase. It must be able to fit on a single floppy disk and therefore cannot exceed 1MB. After you create the phase, use the VTAM MODIFY command to transfer the phase from the VTAM host to a local communication controller.

## Downloading a Phase to a Floppy Disk

Use the Display IPL Information (DII) option to download the phase from the local communication controller to a floppy disk. From the DII option, select FLOPPY MANAGEMENT. Then, use the PF key for COPY LM TO FLOPPY. You will be prompted for the CCU (only if two CCUs are available) and the name of the phase to be downloaded.

**Note:** You must format the disk before downloading the phase. A MOSS function is provided for this purpose.

## Uploading a Phase from a Floppy Disk to a MOSS Disk

You must perform the uploading task on a remote communication controller.

Use the DII option to upload the phase from a floppy disk to a MOSS disk on a remote communication controller. From the DII option, select FLOPPY MANAGEMENT. You will be prompted for the CCU (only if two CCUs are available) and the name of the phase to be uploaded. If a phase with the same name exists on the MOSS disk, it is replaced. If a phase with the same name does not exist and space is available on the MOSS disk, the phase from the floppy disk is added to the MOSS disk. If no space is available on the MOSS disk, the phase from the floppy disk replaces the oldest phase on the MOSS disk.

You must then perform an IPL of the phase from the MOSS console.

# Glossary, Bibliography, and Index

# Glossary

This glossary defines important NCP, NetView, NetView/PC, SSP, and VTAM abbreviations and terms. It includes information from the *IBM Dictionary of Computing*, SC20-1699.

It also includes terms and definitions from:

- The *American National Dictionary for Information Processing Systems*, copyright 1982 by the Computer and Business Equipment Manufacturers Association (CBEMA). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by an asterisk (*).
- The *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Committee (ISO/IEC JTC1/SC1). Definitions from published segments of the vocabularies are identified by the symbol **(ISO)**; definitions from draft international standards, draft proposals, and working papers in development by the ISO/IEC JTC1/SC1 vocabulary subcommittee are identified by the symbol **(TC97)**, indicating final agreement has not yet been reached among participating members.
- The *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the Consultative Committee on International Telegraph and Telephone of the International Telecommunication Union, Geneva, 1980 are preceded by the symbol **(CCITT/ITU)**.

For abbreviations, the definition usually consists only of the words represented by the letters; for complete definitions, see the entries for the words.

### Reference Words Used in the Entries

The following reference words are used in this glossary:

*Deprecated term for*. Indicates that the term should not be used. It refers to a preferred term, which is defined.

*Synonymous with*. Appears in the commentary of a preferred term and identifies less desirable or less specific terms that have the same meaning.

*Synonym for*. Appears in the commentary of a less desirable or less specific term and identifies the preferred term that has the same meaning.

*Contrast with*. Refers to a term that has an opposed or substantively different meaning.

*See*. Refers to multiple-word terms that have the same last word.

*See also*. Refers to related terms that have similar (but not synonymous) meanings.

**abend**. Abnormal end of task.

**abnormal end of task (abend)**. Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is executing.

**ACB**. In NCP, adapter control block.

**ACB name**. The name of an ACB macroinstruction.

**accept**. In NPM, an SMP process that moves distributed code to the distribution libraries.

**access method**. A technique for moving data between main storage and input/output devices.

**access method services (AMS)**. In NPM, the facility used to define and reproduce NPM key-sequenced data sets (KSDSs).

**ACF**. Advanced Communications Function.

**ACF/NCP**. Advanced Communications Function for the Network Control Program. Synonym for *NCP*.

**ACF/SSP**. Advanced Communications Function for the System Support Programs. Synonym for *SSP*.

**ACF/TCAM**. Advanced Communications Function for the Telecommunications Access Method. Synonym for *TCAM*.

**ACF/VTAM**. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

**activate**. To make a resource of a node ready to perform the functions for which it was designed. Contrast with *deactivate*.

**adapter**. Hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device.

**adapter control block (ACB)**. In NCP, a control block that contains line control information and the states of I/O operations for BSC lines, SS lines, or SDLC links.

**adaptive session-level pacing**. A form of session-level pacing in which session components exchange pacing windows that may vary in size during the course of a session. This allows transmission to adapt dynamically to variations in availability and demand of buffers on a

session by session basis. Session pacing occurs within independent stages along the session path according to local congestion at the intermediate nodes. Synonymous with *adaptive session pacing*. See *pacing*, *session-level pacing*, and *virtual route pacing*.

**adaptive session pacing**. Synonym for *adaptive session-level pacing*.

**Advanced Communications Function (ACF)**. A group of IBM licensed programs (principally VTAM, TCAM, NCP, and SSP) that use the concepts of Systems Network Architecture (SNA), including distribution of function and resource sharing.

**allocate**. A logical unit (LU) 6.2 application program interface (API) verb used to assign a session to a conversation for the conversation's use. Contrast with *deallocate*.

**AMS**. Access method services.

**APAR**. Authorized program analysis report.

**apply**. In NPM, an SMP process that moves distributed code to the system libraries.

**attaching device**. Any device that is physically connected to a network and can communicate over the network.

**authorized program analysis report (APAR)**. A request for correction of a problem caused by a defect in a current unaltered release of a program.

**autotask**. An unattended NetView operator station task that does not require a terminal or a logged-on user. Autotasks can run independent of VTAM and are typically used for automated console operations. Contrast with *logged-on operator*.

**basic information unit (BIU)**. In SNA, the unit of data and control information that is passed between half-sessions. It consists of a request/response header (RH) followed by a request/response unit (RU).

**basic transmission unit (BTU)**. In SNA, the unit of data and control information passed between path control components. A BTU can consist of one or more path information units (PIUs). See also *blocking of PIUs*.

**BER**. Box event record.

**binary synchronous communication (BSC)**. (1) Communication using binary synchronous line discipline. (2) A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary-coded data between stations.

**BIU**. Basic information unit.

**BIU segment**. In SNA, the portion of a basic information unit (BIU) that is contained within a path information unit (PIU). It consists of either a request/response header (RH) followed by all or a portion of a request/response unit (RU), or only a portion of an RU.

**blocking of PIUs**. In SNA, an optional function of path control that combines multiple path information units (PIUs) into a single basic transmission unit (BTU).

**boundary function**. (1) A capability of a subarea node to provide protocol support for attached peripheral nodes, such as: (a) interconnecting subarea path control and peripheral path control elements, (b) performing session sequence numbering for low-function peripheral nodes, and (c) providing session-level pacing support. (2) The component that provides these capabilities. See also *boundary node*, *network addressable unit (NAU)*, *peripheral path control*, *subarea node*, and *subarea path control*.

**boundary node**. (1) A subarea node with boundary function. See *subarea node*. See also *boundary function*. (2) The programming component that performs FID2 (format identification type 2) conversion, channel data link control, pacing, and channel or device error recovery procedures for a locally attached station. These functions are similar to those performed by a network control program for an NCP-attached station.

**BSC**. Binary synchronous communication.

**BTU**. Basic transmission unit.

**buffer**. A portion of storage for temporarily holding input or output data.

**call**. (1) * (ISO) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) To transfer control to a procedure, program, routine, or subroutine. (3) The actions necessary to make a connection between two stations. (4) To attempt to contact a user, regardless of whether the attempt is successful.

**call-accepted packet**. * (ISO) A call supervision packet that a called data terminal equipment (DTE) transmits to indicate to the data circuit-terminating equipment (DCE) that it accepts the incoming call.

**call connected packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to indicate to a calling data terminal equipment (DTE) that the connection for the call has been completely established.

**calling**. * (ISO) The process of transmitting selection signals in order to establish a connection between data stations.

**CALLOUT**. The logical channel type on which the data terminal equipment (DTE) can send a call, but cannot receive one.

**call request packet**. A call supervision packet that a data terminal equipment (DTE) transmits to ask that a connection for a call be established throughout the network.

**CCU**. Central control unit.

**CDS**. Control data set.

**CEN**. Composite end node.

**chain**. (1) A group of logically linked records. (2) See *RU chain*.

**channel**. * A path along which signals can be sent, for example, data channel, output channel. See *data channel* and *input/output channel*. See also *link*.

**channel adapter**. A communication controller hardware unit used to attach the controller to a System/360 or a System/370 channel.

**channel-attached**. (1) Pertaining to the attachment of devices directly by input/output channels to a host processor. (2) Pertaining to devices attached to a controlling unit by cables, rather than by telecommunication lines. Contrast with *link-attached*. Synonymous with *local*.

**circuit**. See *data circuit*.

**circuit switching**. (1) * (ISO) A process that, on demand, connects two or more data terminal equipments (DTEs) and permits the exclusive use of a data circuit between them until the connection is released. (2) Synonymous with *line switching*. (3) See also *message switching* and *packet switching*.

**clear indication packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to inform a data terminal equipment (DTE) that a call has been cleared.

**clear request packet**. A call supervision packet transmitted by a data terminal equipment (DTE) to ask that a call be cleared.

**CMS**. Conversational monitor system.

**command**. (1) A request from a terminal for the performance of an operation or the execution of a particular program. (2) In SNA, any field set in the transmission header (TH), request header (RH), and sometimes portions of a request unit (RU), that initiates an action or that begins a protocol; for example: (a) Bind Session (session-control request unit), a command that activates an LU-LU session, (b) the change-direction indicator in the RH of the last RU of a chain, (c) the virtual route reset window indicator in a FID4 transmission header.

**communication controller**. A type of communication control unit whose operations are controlled by one or more programs stored and executed in the unit; for example, the IBM 3725 Communication Controller. It manages the details of line control and the routing of data through a network.

**communication control unit**. A communication device that controls the transmission of data over lines in a network. Communication control units include transmission control units (such as the 2702 Transmission Control Unit) and communication controllers (such as the 3720 or 3725).

**communication line**. Deprecated term for *telecommunication line* and *transmission line*.

**communication management configuration host node**. The type 5 host processor in a communication management configuration that does all network-control functions in the network except for the control of devices channel-attached to data hosts. Synonymous with *communication management host*. Contrast with *data host node*.

**communication management host**. Synonym for *communication management configuration host node*. Contrast with *data host*.

**composite end node (CEN)**. A group of nodes made up of a single type 5 node and its subordinate type 4 nodes that together support type 2.1 protocols. To a type 2.1 node, a CEN appears as one end node. For example, NCP and VTAM act as a composite end node.

**configuration**. (1) (TC97) The arrangement of a computer system or network as defined by the nature, number, and the chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network. (3) In CCP, the arrangement of controllers, lines, and terminals attached to an IBM 3710 Network Controller. Also, the collective set of item definitions that describe such a configuration.

**configuration report program (CRP)**. An SSP utility program that creates a configuration report listing network resources and resource attributes for networks with NCP, EP, PEP, or VTAM.

**control block**. (1) (ISO) A storage area used by a computer program to hold control information. (2) In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the Adapter Support Interface to request an operation.

**control data set (CDS)**. In NPM, an SMP data set used in the NPM installation process.

**controller**. A unit that controls input/output operations for one or more devices.

**control point (CP)**. (1) A system services control point (SSCP) that provides hierarchical control of a group of nodes in a network. (2) A control point (CP) local to a specific node that provides control of that node, either in the absence of SSCP control (for type 2.1 nodes engaged in peer to peer communication) or to supplement SSCP control.

**control program (CP)**. The VM operating system that manages the real processor's resources and is responsible for simulating System/370s for individual users.

**control statement**. A statement in a command list that controls the processing sequence of the command list or allows the command list to send messages to the operator and receive input from the operator.

**conversational monitor system (CMS)**. A VM application program for general interactive time sharing, problem solving, and program development.

**CP**. (1) Control program. (2) Control point.

**cross-domain**. In SNA, pertaining to control of resources involving more than one domain.

**CRP**. Configuration report program.

**DASD**. Direct access storage device.

**data channel**. Synonym for *input/output channel*. See *channel*.

**data circuit**. (1) (ISO) A pair of associated transmit and receive channels that provide a means of two-way data communication. (2) See also *physical circuit* and *virtual circuit*.

- Between data switching exchanges, the data circuit may include data circuit-terminating equipment (DCE), depending on the type of interface used at the data switching exchange.
- Between a data station and a data switching exchange or data concentrator, the data circuit includes the data circuit-terminating equipment at the data station end, and may include equipment similar to a DCE at the data switching exchange or data concentrator location.

**data host**. Synonym for *data host node*. Contrast with *communication management configuration host*.

**data host node**. In a communication management configuration, a type 5 host node that is dedicated to processing applications and does not control network resources, except for its channel-attached or communication adapter-attached devices. Synonymous with *data host*. Contrast with *communication management configuration host node*.

**data link**. In SNA, synonym for *link*.

**data link control (DLC) layer**. In SNA, the layer that consists of the link stations that schedule data transfer over a transmission medium connecting two nodes and perform error control for the link connection. Examples of data link control are SDLC for serial-by-bit link connection and data link control for the System/370 channel.

**data link control protocol**. In SNA, a set of rules used by two nodes on a data link to accomplish an orderly exchange of information. Synonymous with *line control*.

**data packet**. At the interface between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE), a packet used to transmit user data over a virtual circuit.

**data set**. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

**data set members**. Members of partitioned data sets that are individually named elements of a larger file that can be retrieved by name.

**DCE clear confirmation packet**. A call supervision packet that a data circuit-terminating equipment (DCE) transmits to confirm that a call has been cleared.

**ddname**. Data definition name.

**deactivate**. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

**deallocate**. A logical unit (LU) 6.2 application program interface (API) verb that terminates a conversation, thereby freeing the session for a future conversation. Contrast with *allocate*.

**definite response (DR)**. In SNA, a value in the form-of-response-requested field of the request header. The value directs the receiver of the request to return a response unconditionally, whether positive or negative, to that request. Contrast with *exception response* and *no response*.

**definition statement**. In NCP, a type of instruction that defines a resource to the NCP. See Figure 30 on page 185 and Figure 31 on page 185. See also *macroinstruction*.

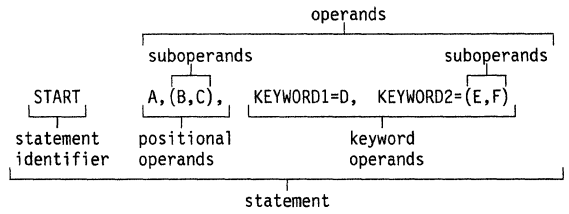



*Figure 30. Example of a language statement*

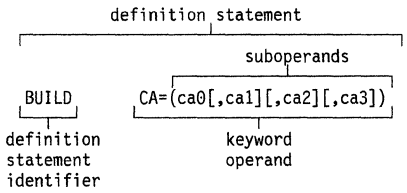



*Figure 31. NCP examples*

**device**. An input/output unit such as a terminal, display, or printer. See *attaching device*.

**direct access storage device (DASD)**. A device in which the access time is effectively independent of the location of the data. For example, a disk.

**directory**. In VM, a control program (CP) disk that defines each virtual machine's normal configuration.

**discarded packet**. A packet that is intentionally destroyed.

**display**. (1) To present information for viewing, usually on a terminal screen or a hard-copy device. (2) A device or medium on which information is presented, such as a terminal screen. (3) Deprecated term for *panel*.

**DLC layer**. Data link control layer.

**domain**. (1) An access method, its application programs, communication controllers, connecting lines, modems, and attached terminals. (2) In SNA, a system services control point (SSCP) and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests. See *system services control point domain* and *type 2.1 node control point domain*. See also *single-domain network* and *multiple-domain network*.

**domain operator**. In a multiple-domain network, the person or program that controls the operation of the resources controlled by one system services control point. Contrast with *network operator* (2).

**DR**. (1) In NCP and CCP, dynamic reconfiguration. (2) In SNA, definite response.

**DRDS**. Dynamic reconfiguration data set.

**dsname**. Data set name.

**dump**. (1) Computer printout of storage. (2) To write the contents of all or part of storage to an external medium as a safeguard against errors or in connection with debugging. (3) (ISO) Data that have been dumped.

**duplex**. * In data communication, pertaining to a simultaneous two-way independent transmission in both directions. Synonymous with *full duplex*. Contrast with *half duplex*.

**dynamic reconfiguration (DR)**. The process of changing the network configuration (peripheral PUs and LUs) without regenerating complete configuration tables.

**dynamic reconfiguration data set (DRDS)**. In VTAM, a data set used for storing definition data that can be applied to a generated communication controller configuration at the operator's request. A dynamic reconfiguration data set can be used to dynamically add PUs and LUs, delete PUs and LUs, and move PUs. It is activated with the VARY DRDS operator command. See also *dynamic reconfiguration*.

**EBCDIC**. * Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

**ECB**. Event control block.

**echo**. The return of characters to the originating SS device to verify that a message was sent correctly.

**ECL**. Electronic cabling link.

**element**. (1) A field in the network address. (2) The particular resource within a subarea identified by the element address. See also *subarea*.

**element address**. In SNA, a value in the element address field of the network address identifying a specific resource within a subarea. See *subarea address*.

**Emulation Program (EP)**. An IBM control program that allows a channel-attached 3705 or 3725 communication controller to emulate the functions of an IBM 2701 Data Adapter Unit, an IBM 2702 Transmission Control, or an IBM 2703 Transmission Control. See also *network control program*.

**end node**. A type 2.1 node that does not provide any intermediate routing or session services to any other

node. For example, APPC/PC is an end node. See *composite end node*, *node*, and *type 2.1 node*.

**EP**. Emulation Program.

**ER**. (1) Explicit route. (2) Exception response.

**event**. (1) In the NetView program, a record indicating irregularities of operation in physical elements of a network. (2) An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as an input/output operation.

**event control block (ECB)**. A control block used to represent the status of an event.

**exception response (ER)**. In SNA, a value in the form-of-response-requested field of a request header (RH). An exception response is sent only if a request is unacceptable as received or cannot be processed. Contrast with *definite response* and *no response*. See also *negative response*.

**exchange identification (XID)**. A data link control command and response passed between adjacent nodes that allows the two nodes to exchange identification and other information necessary for operation over the data link.

**EXEC**. In a VM operating system, a user-written command file that contains CMS commands, other user-written commands, and execution control statements, such as branches.

**explicit route (ER)**. In SNA, the path control network elements, including a specific set of one or more transmission groups, that connect two subarea nodes. An explicit route is identified by an origin subarea address, a destination subarea address, an explicit route number, and a reverse explicit route number. Contrast with *virtual route (VR)*. See also *path* and *route extension*.

**extended architecture (XA)**. An extension to System/370 architecture that takes advantage of continuing high performance enhancements to computer system hardware.

**FASTRUN**. One of several options available with the NCP/EP definition facility (NDF) that indicates only the syntax is to be checked in generation definition statements.

**FDX**. Full duplex.

**feature**. A particular part of an IBM product that a customer can order separately.

**flow control**. In SNA, the process of managing the rate at which data traffic passes between components of the network. The purpose of flow control is to optimize the rate of flow of message units, with minimum congestion in the network; that is, to neither overflow the buffers at the receiver or at intermediate routing nodes, nor leave the receiver waiting for more message units. See also *adaptive session-level pacing*, *pacing*, *session-level pacing*, and *virtual route pacing*.

**frame**. (1) The unit of transmission in some local area networks, including the IBM Token-Ring Network. It includes delimiters, control characters, information, and checking characters. (2) In SDLC, the vehicle for every command, every response, and all information that is transmitted using SDLC procedures.

**full duplex (FDX)**. Synonym for *duplex*.

**generalized path information unit trace (GPT)**. A record of the flow of path information units (PIUs) exchanged between the network control program and its attached resources. PIU trace records consist of up to 44 bytes of transmission header (TH), request/response header (RH), and request/response unit (RU) data.

**generation**. The process of assembling and link editing definition statements so that resources can be identified to all the necessary programs in a network.

**generation definition**. The definition statement of a resource used in generating a program.

**GPT**. Generalized path information unit trace.

**group**. In the NetView/PC program, to identify a set of application programs that are to run concurrently.

**half-duplex**. * In data communication, pertaining to an alternate, one way at a time, independent transmission. Contrast with *duplex*.

**half-session**. In SNA, a component that provides function management data (FMD) services, data flow control, and transmission control for one of the sessions of a network addressable unit (NAU). See also *primary half-session* and *secondary half-session*.

**help panel**. An online display that tells you how to use a command or another aspect of a product. See *task panel*.

**host node**. A node providing an application program interface (API) and a common application interface. See *boundary node*, *node*, *peripheral node*, *subarea host node*, and *subarea node*. See also *boundary function* and *node type*.

**host processor**. (1) (TC97) A processor that controls all or part of a user application network. (2) In a network, the processing unit in which the data communication access method resides.

**IBM software distribution (ISD)**. The IBM department responsible for software distribution.

**IMS**. Information Management System/Virtual Storage. Synonymous with *IMS/VS*.

**IMS/VS**. Information Management System/Virtual Storage. Synonym for *IMS*.

**incoming call packet**. A call supervision packet transmitted by a data circuit-terminating equipment (DCE) to inform a called data terminal equipment (DTE) that another DTE has requested a call.

**information (I) format**. A format used for information transfer.

**Information Management System (IMS)**. A general purpose system whose full name is Information Management System/Virtual Storage (IMS/VS). It enhances the capabilities of OS/VS for batch processing and telecommunication and allows users to access a computer-maintained data base through remote terminals.

**initial program load (IPL)**. (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of loading system programs and preparing a system to run jobs.

**input/output channel**. (1) (ISO) In a data processing system, a functional unit that handles the transfer of data between internal and peripheral equipment. (2) In a computing system, a functional unit, controlled by a processor, that handles the transfer of data between processor storage and local peripheral devices. Synonymous with *data channel*. See *channel*. See also *link*.

**interconnection**. See *SNA network interconnection*.

**interface**. * A shared boundary. An interface might be a hardware component to link two devices or it might be a portion of storage or registers accessed by two or more computer programs.

**intermediate routing node (IRN)**. In SNA, a subarea node with intermediate routing function.

**IPL**. (1) * Initial program loader. (2) Initial program load.

**IRN**. Intermediate routing node.

**ISD**. IBM software distribution.

**item**. In CCP, any of the components, such as communication controllers, lines, cluster controllers, and terminals, that comprise an IBM 3710 Network Controller configuration.

**JCL**. Job control language.

**job control language (JCL)**. * A problem-oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

**keyword**. (1) (TC97) A lexical unit that, in certain contexts, characterizes some language construction. (2) * One of the predefined words of an artificial language. (3) One of the significant and informative words in a title or document that describes the content of that document. (4) A name or symbol that identifies a parameter. (5) A part of a command operand that consists of a specific character string (such as DSNAME=). See also *definition statement* and *keyword operand*. Contrast with *positional operand*.

**keyword operand**. An operand that consists of a keyword followed by one or more values (such as DSNAME=HELLO). See also *definition statement*. Contrast with *positional operand*.

**keyword parameter**. A parameter that consists of a keyword followed by one or more values.

**LCC**. Link connection component.

**LCS**. Link connection subsystem.

**LCSM**. Link connection subsystem manager.

**line**. See *communication line*.

**line control**. Synonym for *data link control protocol*.

**line switching**. Synonym for *circuit switching*.

**link**. In SNA, the combination of the link connection and the link stations joining network nodes; for example: (1) a System/370 channel and its associated protocols, (2) a serial-by-bit connection under the control of Synchronous Data Link Control (SDLC). A link connection is the physical medium of transmission. A link, however, is both logical and physical. Synonymous with *data link*. See Figure 32 on page 188.

**link-attached**. Pertaining to devices that are physically connected by a telecommunication line. Contrast with *channel-attached*. Synonymous with *remote*.

**link connection component (LCC)**. Components of the link that perform functions for the physical layer of the link.

**link connection subsystem (LCS)**. The sequence of link connection components (LCCs) that belong to a link connection and are managed by one LCSM.

**link connection subsystem manager (LCSM)**. The transaction program that manages the sequence of link connection components (LCCs) that belong to a link connection.

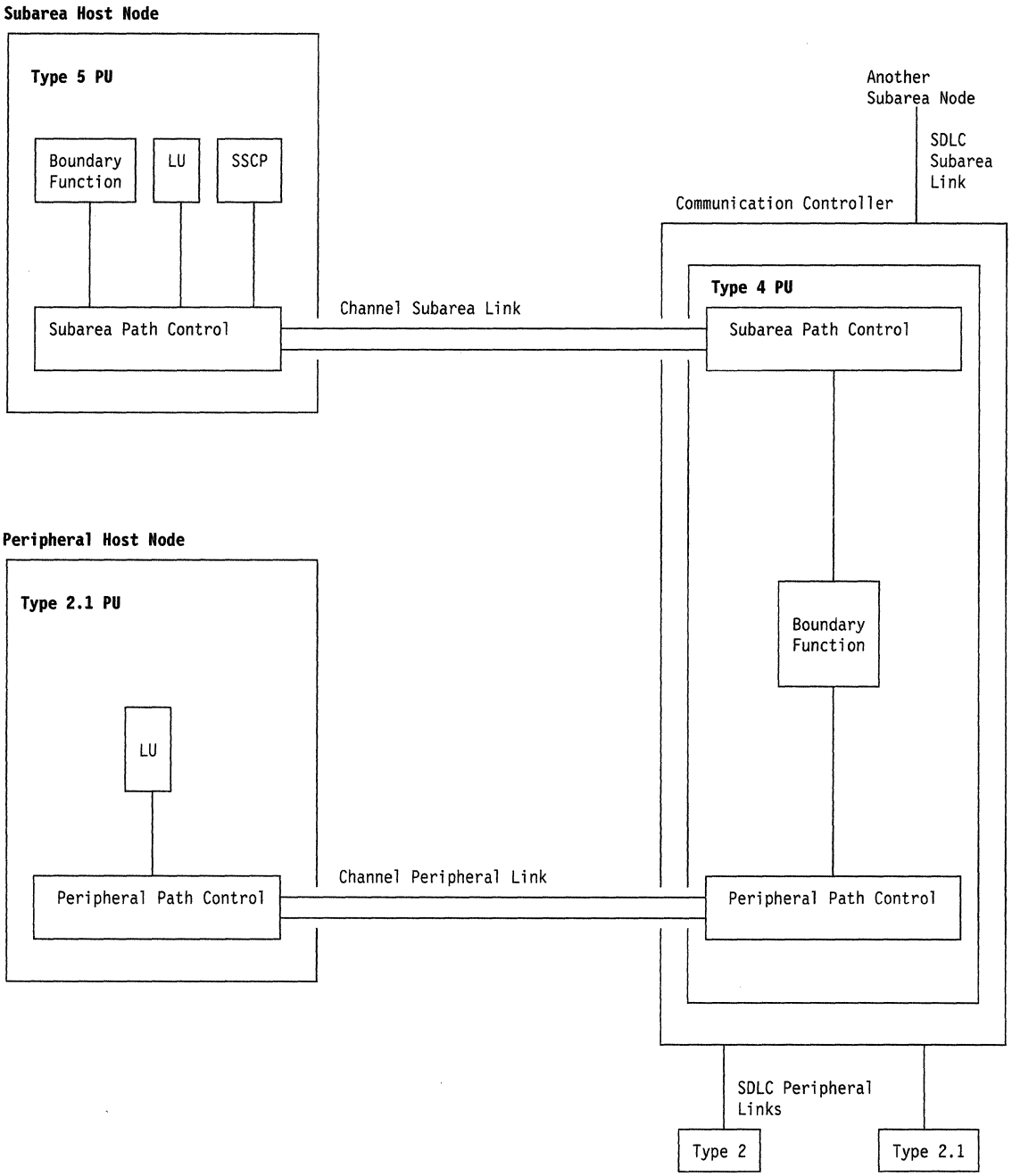



*Figure 32. Links and path controls*

**load module**. (ISO) A program unit that is suitable for loading into main storage for execution; it is usually the output of a linkage editor.

**local**. Pertaining to a device that is attached to a controlling unit by cables, rather than by a telecommunication line. Synonymous with *channel-attached*.

**local address**. In SNA, an address used in a peripheral node in place of an SNA network address and transformed to or from an SNA network address by the boundary function in a subarea node.

**logged-on operator**. A NetView operator station task that requires a terminal and a logged-on user. Contrast with *autotask*.

**logical unit (LU)**. In SNA, a port through which an end user accesses the SNA network and the functions provided by system services control points (SSCPs). An

LU can support at least two sessions—one with an SSCP and one with another LU—and may be capable of supporting many sessions with other LUs. See also *network addressable unit (NAU)*, *peripheral LU*, *physical unit (PU)*, *system services control point (SSCP)*, *primary logical unit (PLU)*, and *secondary logical unit (SLU)*.

**logical unit (LU) services**. In SNA, capabilities in a logical unit to: (1) receive requests from an end user and, in turn, issue requests to the system services control point (SSCP) in order to perform the requested functions, typically for session initiation; (2) receive requests from the SSCP, for example to activate LU-LU sessions via Bind Session requests; and (3) provide session presentation and other services for LU-LU sessions. See also *physical unit (PU) services*.

**logical unit (LU) 6.2**. A type of logical unit that supports general communication between programs in a distributed processing environment. LU 6.2 is characterized by (1) a peer relationship between session partners, (2) efficient utilization of a session for multiple transactions, (3) comprehensive end-to-end error processing, and (4) a generic application program interface (API) consisting of structured verbs that are mapped into a product implementation.

**LU**. Logical unit.

**LU 6.2**. Logical unit 6.2.

**MAC**. Medium Access Control.

**macroinstruction**. (1) An instruction that when executed causes the execution of a predefined sequence of instructions in the same source language. (2) In assembler programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macro definition. The statements normally produced from the macro definition replace the macroinstruction in the program. See also *definition statement*.

**maintenance and operator subsystem (MOSS)**. A subsystem of an IBM communication controller, such as the 3725 or the 3720, that contains a processor and operates independently of the rest of the controller. It loads and supervises the controller, runs problem determination procedures, and assists in maintaining both hardware and software.

**MDR**. Miscellaneous data record.

**Medium Access Control (MAC)**. The sublayer of DLC that supports medium-dependent functions and uses the services of the physical layer to provide services to Logical Link Control (LLC). The MAC sublayer includes the medium access port.

**medium access control (MAC) procedure**. (TC97) In a local area network, the part of the protocol that governs access to the transmission medium independently of the physical characteristics of the medium, but takes into account the topological aspects of the network, in order to enable the exchange of data between data stations.

**message**. (TC97) A group of characters and control bit sequences transferred as an entity.

**message switching**. (1) * (ISO) In a data network, the process of routing messages by receiving, storing, and forwarding complete messages. (2) The technique of receiving a complete message, storing, and then forwarding it unaltered to its destination.

**MIC**. Middle-in-chain.

**middle-in-chain (MIC)**. A request unit (RU) whose request header (RH) begin chain indicator and RH end chain indicator are both off. See also *RU chain*.

**migration**. Installing a new version or release of a program when an earlier version or release is already in place.

**miscellaneous data record (MDR)**. A record of a network hardware error recorded by the NCP and sent to the VTAM host that owns the failing component. Then VTAM writes the error on the operating system error data set.

**modem**. A device that modulates and demodulates signals transmitted over data communication facilities. The term is a contraction for modulator-demodulator.

**module**. * A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading; for example, the input to or output from an assembler, compiler, linkage editor, or executive routine.

**monitor**. In the IBM Token-Ring Network, the function required to initiate the transmission of a token on the ring and to provide soft-error recovery in case of lost tokens, circulating frames, or other difficulties. The capability is present in all ring stations.

**MOSS**. Maintenance and operator subsystem.

**multiple-domain network**. In SNA, a network with more than one system services control point (SSCP). Contrast with *single-domain network*.

**Multiple Virtual Storage (MVS)**. An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

**Multiple Virtual Storage for Extended Architecture (MVS/XA)**. An IBM licensed program whose full name

is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for Extended Architecture. Extended architecture allows 31-bit storage addressing. MVS/XA is a software operating system controlling the execution of programs.

**MVS**. Multiple Virtual Storage.

**MVS/ESA**. Multiple Virtual Storage/Enterprise Systems Architecture.

**MVS/XA**. Multiple Virtual Storage for Extended Architecture.

**NAU**. Network addressable unit.

**NC**. Network control.

**NCP**. (1) Network Control Program (IBM licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. (2) Network control program (general term).

**NCP/EP definition facility (NDF)**. A program that is part of System Support Programs (SSP) and is used to generate a partitioned emulation program (PEP) load module or a load module for a Network Control Program (NCP) or for an Emulation Program (EP).

**NCP/Token-Ring interconnection (NTRI)**. An NCP function that allows a communication controller to attach to the IBM Token-Ring Network and provides both subarea and peripheral node DLC services in the SNA network.

**NCP V4 Subset**. Advanced Communications Function for Network Control Program (NCP) V4 Subset. An IBM licensed program that is a subset of NCP. It operates only on IBM 3720 Communication Controllers with certain capacity limitations such as number of scanners, lines, and channel adapters supported.

**NDF**. NCP/EP definition facility.

**negative response (NR)**. In SNA, a response indicating that a request did not arrive successfully or was not processed successfully by the receiver. Contrast with *positive response*. See *exception response*.

**NetView**. A system 370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

**NetView-NetView task (NNT)**. The task under which a cross-domain NetView operator session runs. See *operator station task*.

**NetView-NetView task (NNT)**. The task under which a cross-domain NetView operator session runs. See *operator station task*.

**NetView Performance Monitor (NPM)**. An IBM licensed program that collects, monitors, analyzes, and displays data relevant to the performance of a VTAM telecommunication network. It runs as an online VTAM application program.

**network**. (1) (TC97) An interconnected group of nodes. (2) In data processing, a user application network. See *path control network, public network, SNA network*, and *user-application network*.

**network address**. In SNA, an address, consisting of subarea and element fields, that identifies a link, a link station, or a network addressable unit. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa. See *local address*. See also *network name*.

**network addressable unit (NAU)**. In SNA, a logical unit, a physical unit, or a system services control point. It is the origin or the destination of information transmitted by the path control network. Each NAU has a network address that represents it to the path control network. See also *network name*, *network address*, and *path control network*.

**network control (NC)**. In SNA, an RU category used for requests and responses exchanged for such purposes as activating and deactivating explicit and virtual routes and sending load modules to adjacent peripheral nodes. See also *session control*.

**network control program**. A program, generated by the user from a library of IBM-supplied modules, that controls the operation of a communication controller.

**Network Control Program (NCP)**. An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability. Its full name is Advanced Communications Function for the Network Control Program.

**network definition facility (NDF)**. The facility that defines the identities and characteristics of each node in the network and the arrangement of the nodes in that system.

**network name**. (1) In SNA, the symbolic identifier by which end users refer to a network addressable unit (NAU), a link, or a link station. See also *network address*. (2) In a multiple-domain network, the name of the APPL statement defining a VTAM application program is its network name and it must be unique across domains. Contrast with *ACB name*. See *uninterpreted name*.

**network operator**. (1) A person or program responsible for controlling the operation of all or part of a network. (2) The person or program that controls all

the domains in a multiple-domain network. Contrast with *domain operator*.

**network performance analyzer (NPA)**. A function of NCP that collects performance data about devices. The data is recorded by NPM.

**Network Problem Determination Application (NPDA)**. An IBM licensed program that helps you identify network problems, such as hardware, software, and microcode, from a central control point using interactive display techniques. It runs as an NCCF communication network management (CNM) application program. Its function is included and enhanced in NetView's hardware monitor.

**Network Routing Facility (NRF)**. An IBM licensed program that resides in the NCP, which provides a path for messages between terminals, and routes messages over this path without going through the host processor.

**Network Terminal Option (NTO)**. An IBM licensed program used in conjunction with NCP that allows certain non-SNA devices to participate in sessions with SNA application programs in the host processor. NTO converts non-SNA protocol to SNA protocol when data is sent to the host from a non-SNA device and reconverts SNA protocol to non-SNA protocol when data is sent back to the device.

**NNT**. NetView-NetView task.

**node**. In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. See *boundary node*, *host node*, *peripheral node*, and *subarea node*.

**node type**. In SNA, a designation of a node according to the protocols it supports and the network addressable units (NAUs) that it can contain. Five types are defined: 1, 2.0, 2.1, 4, and 5. Type 1, type 2.0, and type 2.1 nodes are peripheral nodes; type 4 and type 5 nodes are subarea nodes. See also *type 2.1 node*.

**no response**. In SNA, a value in the form-of-response-requested field of the request header (RH) indicating that no response is to be returned to the request, whether or not the request is received and processed successfully. Contrast with *definite response* and *exception response*.

**NPA**. Network performance analyzer.

**NPM**. NetView Performance Monitor.

**NPDA**. Network Problem Determination Application.

**NPSI**. X.25 NCP Packet Switching Interface.

**NR**. Negative response.

**NRF**. Network Routing Facility.

**NTO**. Network Terminal Option.

**NTRI**. NCP/Token-Ring interconnection.

**online**. Stored in a computer and accessible from a terminal.

**open**. (1) In the IBM Token-Ring Network, to make an adapter ready for use. (2) A break in an electrical circuit.

**operand**. (1) (ISO) An entity on which an operation is performed. (2) * That which is operated upon. An operand is usually identified by an address part of an instruction. (3) Information entered with a command name to define the data on which a command processor operates and to control the execution of the command processor. (4) An expression to whose value an operator is applied. See also *definition statement*, *keyword*, *keyword parameter*, and *parameter*.

**operator**. (1) In a language statement, the lexical entity that indicates the action to be performed on operands. See also *definition statement*. (2) A person who operates a machine. See *network operator*. (3) A person or program responsible for managing activities controlled by a given piece of software such as MVS, the NetView program, or IMS. See *logged-on operator* and *network operator* See also *autotask* and *operator station task*.

**operator station task (OST)**. The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program. See *NetView-NetView task*.

**OST**. Operator station task.

**pacing**. In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion. See *session-level pacing*, *send pacing*, and *virtual route (VR) pacing*. See also *flow control*.

**pacing response**. In SNA, an indicator that signifies a receiving component's readiness to accept another pacing group; the indicator is carried in a response header (RH) for session-level pacing, and in a transmission header (TH) for virtual route pacing.

**packet**. (ISO) A sequence of binary digits, including data and control signals, that is transmitted and switched as a composite whole. The data, control signals, and possibly error control information are arranged in a specific format. See *call-accepted packet*, *call-connected packet*, *call request packet*, *call*

*supervision packets, clear indication packet, clear request packet, data packet, DCE clear confirmation packet, discarded packet, incoming call packet, permit packet,* and *reset packet.*

**packet mode operation**. Synonym for *packet switching.*

**packet switching**. (1) (ISO) The process of routing and transferring data by means of addressed packets so that a channel is occupied only during the transmission of a packet. On completion of the transmission, the channel is made available for the transfer of other packets. (2) Synonymous with *packet mode operation.* See also *circuit switching.*

**page**. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also *scroll.* (3) (ISO) In a virtual storage system, a fixed-length block that has a virtual address and that can be transferred between real storage and auxiliary storage. (4) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

**panel**. (1) A formatted display of information that appears on a terminal screen. See *help panel* and *task panel.* Contrast with *screen.* (2) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

**parameter**. (1) (ISO) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure. See also *keyword, keyword parameter,* and *operand.*

**partitioned data set (PDS)**. A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partitioned emulation program (PEP) extension**. A function of a network control program that enables a communication controller to operate some telecommunication lines in network control mode while simultaneously operating others in emulation mode.

**path**. (1) In SNA, the series of path control network components (path control and data link control) that are traversed by the information exchanged between two network addressable units (NAUs). See also *explicit route (ER), route extension,* and *virtual route (VR).* (2) In the NetView/PC program, a complete line in a configuration that contains all of the resources in the service point command service (SPCS) query link configuration request list.

**path control (PC)**. The function that routes message units between network addressable units (NAUs) in the network and provides the paths between them. It converts the BIUs from transmission control (possibly segmenting them) into path information units (PIUs) and exchanges basic transmission units (BTUs) and one or more PIUs with data link control. Path control differs for peripheral nodes, which use local addresses for routing, and subarea nodes, which use network addresses for routing. See *peripheral path control* and *subarea path control.* See also *link, peripheral node,* and *subarea node.*

**path control (PC) layer**. In SNA, the layer that manages the sharing of link resources of the SNA network and routes basic information units (BIUs) through it. See also *BIU segment, blocking of PIUs, data link control layer,* and *transmission control layer.*

**path control (PC) network**. In SNA, the part of the SNA network that includes the data link control and path control layers. See *SNA network* and *user application network.* See also *boundary function.*

**path information unit (PIU)**. In SNA, a message unit consisting of a transmission header (TH) alone, or of a TH followed by a basic information unit (BIU) or a BIU segment. See also *transmission header.*

**PC**. Path control.

**PDS**. Partitioned data set.

**PEP**. Partitioned emulation program.

**peripheral host node**. A node that provides an application program interface (API) for running application programs but does not provide SSCP functions and is not aware of the network configuration. The peripheral host node does not provide subarea node services. It has boundary function provided by its adjacent subarea. See *boundary node, host node, node, peripheral node, subarea host node,* and *subarea node.* See also *boundary function* and *node type.*

**peripheral LU**. In SNA, a logical unit representing a peripheral node.

**peripheral node**. In SNA, a node that uses local addresses for routing and therefore is not affected by changes in network addresses. A peripheral node requires boundary-function assistance from an adjacent subarea node. A peripheral node is a physical unit (PU) type 1, 2.0, or 2.1 node connected to a subarea node with boundary function within a subarea. See *boundary node, host node, node, peripheral host node, subarea host node,* and *subarea node.* See also *boundary function* and *node type.*

**peripheral path control**. The function in a peripheral node that routes message units between units with

local addresses and provides the paths between them. See *path control* and *subarea path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

**peripheral PU**. In SNA, a physical unit representing a peripheral node.

**permit packet**. At the interface between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE), a packet used to transmit permits over a virtual circuit.

**physical circuit**. A circuit established without multiplexing. Contrast with *virtual circuit*. See also *data circuit*.

**physical unit (PU)**. In SNA, a type of network addressable unit (NAU). A physical unit (PU) manages and monitors the resources (such as attached links) of a node, as requested by a system services control point (SSCP) through an SSCP-PU session. An SSCP activates a session with the physical unit in order to indirectly manage, through the PU, resources of the node such as attached links. See also *peripheral PU* and *subarea PU*.

**physical unit (PU) services**. In SNA, the components within a physical unit (PU) that provide configuration services and maintenance services for SSCP-PU sessions. See also *logical unit (LU) services*.

**PIU**. Path information unit.

**PLU**. Primary logical unit.

**positional operand**. An operand in a language statement that has a fixed position. See also *definition statement*. Contrast with *keyword operand*.

**positive response**. A response indicating that a request was received and processed. Contrast with *negative response*.

**primary half-session**. In SNA, the half-session that sends the session activation request. See also *primary logical unit*. Contrast with *secondary half-session*.

**primary logical unit (PLU)**. In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Each session must have a PLU and secondary logical unit (SLU). The PLU is the unit responsible for the bind and is the controlling LU for the session. A particular LU may contain both primary and secondary half-sessions for different active LU-LU sessions. Contrast with *secondary logical unit (SLU)*.

**problem determination**. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environment failure such as a power loss, or a user error.

**PU**. Physical unit.

**public network**. A network established and operated by communication common carriers or telecommunication Administrations for the specific purpose of providing circuit-switched, packet switched, and leased-circuit services to the public. Contrast with *user-application network*.

**receive pacing**. In SNA, the pacing of message units that the component is receiving. See also *send pacing*.

**Recommendation X.21 (Geneva 1980)**. A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for a general purpose interface between data terminal equipment and data circuit equipment for synchronous operations on a public data network.

**Recommendation X.25 (Geneva 1980)**. A Consultative Committee on International Telegraph and Telephone (CCITT) recommendation for the interface between data terminal equipment and packet-switched data networks. See also *packet switching*.

**record**. (1) (ISO) In programming languages, an aggregate that consists of data objects, possibly with different attributes, that usually have identifiers attached to them. In some programming languages, records are called structures. (2) (TC97) A set of data treated as a unit. (3) A set of one or more related data items grouped for processing.

**remote**. Concerning the peripheral parts of a network not centrally linked to the host processor and generally using telecommunication lines with public right-of-way.

**remote modem self-test (RST)**. A check on hardware to identify a field-replaceable unit that is failing.

**request header (RH)**. In SNA, control information preceding a request unit (RU). See also *request/response header (RH)*.

**request/response header (RH)**. In SNA, control information, preceding a request/response unit (RU), that specifies the type of RU (request unit or response unit) and contains control information associated with that RU.

**request/response unit (RU)**. In SNA, a generic term for a request unit or a response unit. See also *request unit (RU)* and *response unit*.

**request unit (RU)**. In SNA, a message unit that contains control information, end-user data, or both.

**reset**. On a virtual circuit, reinitialization of data flow control. At reset, all data in transit are eliminated.

**reset packet**. A packet used to reset a virtual circuit at the interface between the data terminal equipment (DTE) and the data circuit-terminating equipment (DCE).

**resource**. (1) Any facility of the computing system or operating system required by a job or task, and including main storage, input/output devices, the processing unit, data sets, and control or processing programs. (2) In the NetView program, any hardware or software that provides function to the network.

**resource resolution table (RRT)**. In NPM, this table contains the names of network resources for which data is to be collected. The NPM RRT corresponds with an NCP and is built by NPMGEN from an NCP Stage I and an NCP RRT.

**response header (RH)**. In SNA, a header, optionally followed by a response unit (RU), that indicates whether the response is positive or negative and that may contain a pacing response. See also *negative response, pacing response*, and *positive response*.

**response unit (RU)**. In SNA, a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as session parameters in response to Bind Session), or if negative, contains sense data defining the exception condition.

**Restructured Extended Executor (REXX)**. An interpretive language used to write command lists.

**return code**. * A code [returned from a program] used to influence the execution of succeeding instructions.

**REX**. Route extension.

**REXX**. Restructured Extended Executor.

**RH**. Request/response header.

**ring**. A network configuration where a series of attaching devices are connected by unidirectional transmission links to form a closed path.

**route extension (REX)**. In SNA, the path control network components, including a peripheral link, that make up the portion of a path between a subarea node and a network addressable unit (NAU) in an adjacent peripheral node. See also *path, explicit route (ER)* and *virtual route (VR)*.

**routing**. The assignment of the path by which a message will reach its destination.

**RRT**. Resource resolution table.

**RST**. Remote modem self-test.

**RU**. Request/response unit.

**RU chain**. In SNA, a set of related request/response units (RUs) that are consecutively transmitted on a particular normal or expedited data flow. The request RU chain is the unit of recovery: if one of the RUs in the chain cannot be processed, the entire chain is discarded. Each RU belongs to only one chain, which has a beginning and an end indicated by means of control bits in request/response headers within the RU chain. Each RU can be designated as first-in-chain (FIC), last-in-chain (LIC), middle-in-chain (MIC), or only-in-chain (OIC). Response units and expedited-flow request units are always sent as only-in-chain.

**scanner interface trace (SIT)**. A record of the activity within the communication scanner processor (CSP) for a specified data link between a 3725 Communication Controller and a resource.

**SC**. Session control.

**SCB**. Session control block.

**screen**. An illuminated display surface; for example, the display surface of a CRT or plasma panel. Contrast with *panel*.

**scroll**. To move all or part of the display image vertically to display data that cannot be observed within a single display image. See also *page (2)*.

**secondary half-session**. In SNA, the half-session that receives the session-activation request. See also *secondary logical unit (SLU)*. Contrast with *primary half-session*.

**secondary logical unit (SLU)**. In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. An LU may contain secondary and primary half-sessions for different active LU-LU sessions. Contrast with *primary logical unit (PLU)*.

**secondary logical unit (SLU) key**. A key-encrypting key used to protect a session cryptography key during its transmission to the secondary half-session.

**send pacing**. In SNA, pacing of message units that a component is sending. See also *receive pacing*.

**service point (SP)**. An entry point that supports applications that provide network management for resources not under the direct control of itself as an entry point. Each resource is either under the direct control of another entry point or not under the direct control of any entry point. A service point accessing these resources is not required to use SNA sessions (unlike a focal point). A service point is needed when entry point support is not yet available for some network management function.

**session control (SC)**. In SNA, (1) One of the components of transmission control. Session control is used to purge data flowing in a session after an unrecoverable error occurs, to resynchronize the data flow after such an error, and to perform cryptographic verification. (2) A request unit (RU) category used for requests and responses exchanged between the session control components of a session and for session activation and deactivation requests and responses.

**session control block (SCB)**. In NPM, control blocks in common storage area for session collection.

**session-level pacing**. In SNA, a flow control technique that permits a receiver to control the data transfer rate (the rate at which it receives request units) on the normal flow. It is used to prevent overloading a receiver with unprocessed requests when the sender can generate requests faster than the receiver can process them. See also *pacing* and *virtual route pacing*.

**single-domain network**. In SNA, a network with one system services control point (SSCP). Contrast with *multiple-domain network*.

**SIT**. Scanner interface trace.

**SLU**. Secondary logical unit.

**SMP**. System Modification Program.

**SMP/E**. System Modification Program Extended.

**SNA**. Systems Network Architecture.

**SNA network**. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

**SNA network interconnection (SNI)**. The connection, by gateways, of two or more independent SNA networks to allow communication between logical units in those networks. The individual SNA networks retain their independence.

**SNI**. SNA network interconnection

**SP**. Service point.

**SSCP**. System services control point.

**SSP**. System Support Programs (IBM licensed program). Its full name is Advanced Communications Function for System Support Programs. Synonymous with *ACF/SSP*.

**statement**. A language syntactic unit consisting of an operator, or other statement identifier, followed by one or more operands. See *definition statement*.

**subarea**. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

**subarea address**. In SNA, a value in the subarea field of the network address that identifies a particular subarea. See also *element address*.

**subarea host node**. A host node that provides both subarea function and an application program interface (API) for running application programs. It provides system services control point (SSCP) functions, subarea node services, and is aware of the network configuration. See *boundary node*, *communication management configuration host node*, *data host node*, *host node*, *node*, *peripheral node*, and *subarea node*. See also *boundary function* and *node type*.

**subarea node**. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes. See *boundary node*, *host node*, *node*, *peripheral node*, and *subarea host node*. See also *boundary function* and *node type*.

**subarea path control**. The function in a subarea node that routes message units between network addressable units (NAUs) and provides the paths between them. See *path control* and *peripheral path control*. See also *boundary function*, *peripheral node*, and *subarea node*.

**subarea PU**. In SNA, a physical unit (PU) in a subarea node.

**subsystem**. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

**System Modification Program (SMP)**. An operating system component that facilitates the process of installing and servicing an MVS system. See also *System Modification Program Extended*.

**System Modification Program Extended (SMP/E)**. An IBM licensed program that facilitates the process of installing and servicing an MVS system. See also *System Modification Program*.

**system monitor**. The portion of the configuration image in a 3601 Finance Communication Controller that handles communications with control operators and records error statistics and other operational data.

**system services control point (SSCP)**. In SNA, a central location point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its domain.

**system services control point (SSCP) domain**. The system services control point and the physical units (PUs), logical units (LUs), links, link stations and all the resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

**Systems Network Architecture (SNA)**. The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

**System Support Programs (SSP)**. An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

**task**. A basic unit of work to be accomplished by a computer. The task is usually specified to a control program in a multiprogramming or multiprocessing environment.

**task panel**. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

**TCAM**. Telecommunications Access Method. Synonymous with *ACF/TCAM*.

**TC layer**. Transmission control layer.

**telecommunication line**. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

**Telecommunications Access Method (TCAM)**. (1) Synonymous with *ACF/TCAM*. (2) The IBM licensed program whose full name is Advanced Communications Function for TCAM and that provides queued message handling. TCAM Versions 1 and 2 are access methods, but TCAM Version 3 is a message handling subsystem.

**terminal**. A device that is capable of sending and receiving information over a link; it is usually equipped with a keyboard and some kind of display, such as a screen or a printer.

**TH**. Transmission header.

**token**. A sequence of bits passed from one device to another along the token ring. When the token has data appended to it, it becomes a frame.

**token ring**. A network with a ring topology that passes tokens from one attaching device to another. For example, the IBM Token-Ring Network.

**transmission control (TC) layer**. In SNA, the layer within a half-session that synchronizes and paces session-level data traffic, checks session sequence numbers of requests, and enciphers and deciphers end-user data. Transmission control has two components: the connection point manager and session control. See also *half-session*.

**transmission header (TH)**. In SNA, control information, optionally followed by a basic information unit (BIU) or a BIU segment, that is created and used by path control to route message units and to control their flow within the network. See also *path information unit*.

**transmission line**. Synonym for *telecommunication line*.

**type 2.1 node (T2.1 node)**. A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.0 nodes. Type 2.1 nodes can be directly attached to one another using peer-to-peer protocols. See *end node*, *node*, and *subarea node*. See also *node type*.

**type 2.1 node (T2.1 node) control point domain**. The CP, its logical units (LUs), links, link stations, and all resources that it activates and deactivates.

**uninterpreted name**. In SNA, a character string that a system services control point (SSCP) is able to convert into the network name of a logical unit (LU). Typically, an uninterpreted name is used in a logon or Initiate request from a secondary logical unit (SLU) to identify the primary logical unit (PLU) with which the session is requested.

**user**. Anyone who requires the services of a computing system.

**user-application network**. A configuration of data processing products, such as processors, controllers, and terminals, established and operated by users for the purpose of data processing or information exchange, which may use services offered by communication common carriers or telecommunication Administrations. Contrast with *public network*.

**user-written generation application**. A user-written program that runs with the NCP/EP definition facility (NDF) during NCP generation. It processes definition statements and operands.

**value**. (1) (TC97) A specific occurrence of an attribute, for example, "blue" for the attribute "color." (2) A quantity assigned to a constant, a variable, a parameter, or a symbol.

**variable**. In the NetView command list language, a character string beginning with & that is coded in a command list and is assigned a value during execution of the command list.

**verb**. In SNA, the general name for a transaction program's request for communication services.

**virtual circuit**. (TC97) In packet switching, those facilities provided by a network that give the appearance to the user of an actual connection. Contrast with *physical circuit*. See also *data circuit*.

**virtual machine**. A functional simulation of a computer and its associated devices.

**Virtual Machine (VM)**. A licensed program whose full name is the Virtual Machine/System Product (VM/SP). It is a software operating system that manages the resources of a real processor to provide virtual machines to end users. As a time-sharing system control program, it consists of the virtual machine control program (CP), the conversational monitor system (CMS), the group control system (GCS), and the interactive problem control system (IPCS).

**virtual route (VR)**. In SNA, a logical connection (1) between two subarea nodes that is physically realized as a particular explicit route, or (2) that is contained wholly within a subarea node for intranode sessions. A virtual route between distinct subarea nodes imposes a transmission priority on the underlying explicit route, provides flow control through virtual-route pacing, and provides data integrity through sequence numbering of path information units (PIUs). See also *explicit route (ER)*, *path*, and *route extension*.

**virtual route (VR) pacing**. In SNA, a flow control technique used by the virtual route control component of path control at each end of a virtual route to control the rate at which path information units (PIUs) flow over the virtual route. VR pacing can be adjusted according to traffic congestion in any of the nodes along the route. See also *pacing* and *session-level pacing*.

**virtual storage**. (ISO) The notion of storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations.

**Virtual Storage Access Method (VSAM)**. An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by relative-record number.

**Virtual Storage Extended (VSE)**. An IBM licensed program whose full name is the Virtual Storage Extended/Advanced Function. It is a software operating system controlling the execution of programs.

**Virtual Telecommunications Access Method (VTAM)**. An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

**VM**. Virtual Machine. Its full name is Virtual Machine/System Product. Synonymous with *VM/SP*.

**VM/SP**. Virtual Machine/System Product. Synonym for *VM*.

**VM/XA**. Virtual Machine/Extended Architecture.

**VR**. Virtual route.

**VSAM**. Virtual Storage Access Method.

**VSE**. Virtual Storage Extended. Synonymous with *VSE/Advanced Functions*.

**VSE/Advanced Functions**. The basic operating system support needed for a VSE-controlled installation. Synonym for *VSE*.

**VTAM**. Virtual Telecommunications Access Method (IBM licensed program). Its full name is Advanced Communications Function for the Virtual Telecommunications Access Method. Synonymous with *ACF/VTAM*.

**XA**. Extended architecture.

**XID.** Exchange identification.

**X.25.** See *Recommendation X.25 (Geneva 1980)*.

**X.25 NCP Packet Switching Interface (NPSI).** The X.25 Network Control Program Packet Switching Interface, which is an IBM licensed program that allows SNA users to communicate over packet-switched data networks that have interfaces complying with Recommendation X.25 (Geneva 1980) of the International Telegraph and Telephone Consultative Committee (CCITT). It allows SNA programs to communicate with SNA equipment or with non-SNA equipment over such networks. In addition, this product may be used to attach native X.25 equipment to SNA host systems without a packet network. See also *Recommendation X.25 (Geneva 1980)*.

# Bibliography

## NCP, SSP, and EP Publications

The following paragraphs briefly describe the library for NCP, SSP, and EP. The other books dealing with the network program products – VTAM, NPSI, the NetView program, and NPM – are listed without the accompanying descriptions.

*NCP, SSP, and EP Generation and Loading Guide* (SC30-3348)

This book contains information on generating and loading NCP and EP (in the PEP environment) using SSP.

*NCP Migration Guide for Version 5 Release 4* (SC31-6204)
*NCP and SSP Library Supplement* (SD35-0251 for NCP V5R3.1)
*NCP Migration Guide* (SC30-3440 for NCP V5R1, R2, R2.1, and R3)

These books are designed to help system analysts, system programmers, and system engineers who are responsible for migrating from their NCP to another level of NCP. These books present the updates that users must make to their NCP generation definition.

*NCP, SSP, and EP Resource Definition Guide* (SC30-3447)

This book describes the physical and operational characteristics of NCP and EP (in the PEP environment) using SSP. It also describes the definition statements and keywords associated with those characteristics.

*NCP, SSP, and EP Resource Definition Reference* (SC30-3448)

This book contains detailed descriptions of the definition statements and keywords used to define NCP and EP (in the PEP environment) using SSP.

*NCP and EP Reference Summary and Data Areas* (LY30-5603)

This two-volume book is for system programmers and IBM program support representatives. It provides quick access to often-used diagnostic and debugging information about NCP and EP in the PEP environment. If more comprehensive information is needed about NCP or PEP, refer to the other books in the library.

*NCP Customization Guide* (LY30-5606)

This book is designed to help system analysts, system programmers, and system engineers modify NCP.

*NCP Customization Reference* (LY30-5607)

This book supplements *NCP Customization Guide*. It describes the resources and macroinstructions provided by IBM for customizing NCP.

*SSP Customization* (LY43-0021)

This book is designed to help system analysts, system programmers, and system engineers modify SSP.

*NCP, SSP, and EP Messages and Codes* (SC30-3169)

This book is a reference book of abend codes issued by NCP and EP in the PEP environment, and messages issued by the system support programs associated with the NCP. It is intended to help people who operate, maintain, generate, or load an NCP. This edition includes all of the messages and codes for NCP, SSP, and EP.

*NCP, SSP, and EP Diagnosis Guide* (LY30-5591)

This book is designed to help customers and IBM program support representatives isolate and define problems in NCP and EP (in the PEP environment) using SSP. The primary purpose of the book is to help the user interact with the IBM Support Center to resolve a problem. In addition, it includes detailed descriptions of how to use the programming tools available with NCP and SSP.

*NCP and EP Reference* (LY30-5605)

This book contains reference material describing the internal organization and function of NCP and EP in the PEP environment. It provides information for customization and diagnosis.

## Other Network Program Products Publications

For more information about the publications listed in this bibliography, see *Bibliography and Master Index for NetView, NCP, and VTAM*.

## Network Program Products Publications

The following list shows the cross-product publications for NetView, NCP, and VTAM.

*Planning and Reference for NetView, NCP, and VTAM* (SC31-6092)

*Bibliography and Master Index for NetView, NCP, and VTAM* (GC31-6430)

## VTAM Publications

### VTAM V3R3 Publications for MVS, VM, and VSE.

The following list shows the publications for VTAM V3R3 running under MVS, VM, and VSE.

*VTAM Directory of Migration Information* (GC31-6429)

*VTAM Network Implementation Guide* (SC31-6404)

*VTAM Resource Definition Reference* (SC31-6412)

*VTAM Storage Estimates* (SK2T-2010, a diskette)

*VTAM Customization* (LY43-0046)

*VTAM Operation* (SC31-6408)

*VTAM Messages and Codes* (SC31-6405)

*VTAM Programming* (SC31-6409)

*VTAM Programming for LU 6.2* (SC31-6410)

*VTAM Diagnosis* (LY43-0042)

*VTAM Data Areas for MVS* (LY43-0043)

*VTAM Data Areas for VM* (LY43-0045)

*VTAM Data Areas for VSE* (LY43-0053)

*VTAM Reference Summary* (LY43-0047)

## NPSI Publications

The following list shows the publications for NPSI Version 3.

*X.25 NCP Packet Switching Interface General Information Version 3* (GC30-3469)

*X.25 NCP Packet Switching Interface Planning and Installation Version 3* (SC30-3470)

*X.25 NCP Packet Switching Interface Host Programming Version 3* (SC30-3502)

*X.25 NCP Packet Switching Interface Diagnosis, Customization, and Tuning Version 3* (LY30-5610)

## NetView V2R1 Publications

The following list shows the publications for NetView Version 2 Release 1, arranged according to related tasks.

**Evaluation and Education**

*Learning about NetView Operation* (SK2T-1995, 3.5-inch diskettes)

*Learning about NetView Graphic Monitor Facility* (SK2T-6005, 3.5-inch diskettes)

*NetView at a Glance* (GC31-6114)

**Planning**

*NetView Storage Estimates for MVS* (SK2T-1994, diskettes for a PC or PS/2)

*NetView Storage Estimates for MVS* (SK2T-1993, diskettes for a PS/55)

*Console Automation Using NetView: Planning* (SC31-6058)

**Installation and Administration**

*NetView Installation and Administration Guide* (MVS) (SC31-6051)

*NetView Administration Reference* (SC31-6044)

*NetView Samples* (MVS) (SC31-6047)

*NetView Tuning Guide* (SC31-6056)

*Console Automation Using NetView: Implementing* (LY43-0007)

*NetView Graphic Monitor Facility Installation and Administration* (SC31-6042)

*NetView Bridge Implementation* (SC31-6033)

**Customization**

*NetView Customization Guide* (SC31-6048)

*NetView Customization: Writing Command Lists* (SC31-6050)

*NetView Customization: Using PL/I and C* (SC31-6089)

*NetView Customization: Using Assembler* (SC31-6090)

**Operation**

*Learning about NetView Operation* (SK2T-1995, 3.5-inch diskettes)

*Learning about NetView Graphic Monitor Facility* (SK2T-6005, 3.5-inch diskettes)

*NetView Operation* (SC31-6053)

*NetView Graphic Monitor Facility Operation* (SC31-6099)

*NetView Command Summary* (SX75-0042)

*NetView Messages* (SC31-6097)

**Application Programming**

*NetView Application Programming: Program-to-Program Interface* (SC31-6093)

**Diagnosis**

*NetView Problem Determination and Diagnosis* (LY43-0005)

*NetView Resource Alerts Reference* (SC31-6055)

*NetView Graphic Monitor Facility Diagnosis* (LY43-0009)

## NPM Publications

The following list shows the publications for NPM.

*NetView Performance Monitor at a Glance* (GH20-6359)

*NetView Performance Monitor Operation* (SH20-6360)

*NetView Performance Monitor Messages and Codes* (SH20-6396)

*NetView Performance Monitor Graphic Subsystem* (SH20-6362)

*NetView Performance Monitor Installation and Customization* (SH20-6361)

*NetView Performance Monitor Diagnosis* (LY20-0381)

# Related Publications

The following publications, though not directly related to NCP, may be helpful in understanding your network.

*Emulation Program Resource Definition and Diagnosis* (SC31-6205)

*TCAM Installation, Resource Definition, and Customization Guide* (SC30-3237)

## Communication Controller Publications

### 3705 Publications

The following list shows selected publications for the IBM 3705 Communications Controller.

*3704, 3705-I, and 3705-II Communications Controller Principles of Operation* (GA27-3004)

*Introduction to the 3704 and 3705-II Communications Controller* (GA27-3051)

*Introduction to the 3705-80 Communications Controller* (GA27-3304)

*3705-80 Communications Controller Principles of Operation* (GA27-3074)

### 372x Publications

The following list shows selected publications for the IBM 372x Communication Controller.

*IBM 3720 Component Description* (GA27-2749)

*3720/3721 Communication Controllers Introduction* (GA33-0060)

*3720/3721 Communication Controllers Configuration Guide* (GA33-0063)

*3720/3725 Communication Controllers Principles of Operation* (GA33-0013)

*3720/3721 Operator's Guide* (GA33-0065)

### 3745 Publications

The following list shows selected publications for the IBM 3745 Communication Controller.

*IBM 3745 Communication Controller Introduction* (GA33-0092)

*IBM 3745 Communication Controller Configuration Program* (GA33-0093)

*IBM 3745 Principles of Operation* (SA33-0102)

*3745 Models 130/150/170 Advanced Operations* (SA33-0143)

## SNA Publications

The following publications contain information on SNA.

*Systems Network Architecture Concepts and Products* (GC30-3072)

*Systems Network Architecture Technical Overview* (GC30-3073)

*Systems Network Architecture Format and Protocol Reference Manual: Management Services* (SC30-3346)

*Systems Network Architecture Formats* (GA27-3136)

# Index

## E



## F

## G



## H



## I

## J



## L

# M



# N

## O



## P

## R



## S

## T

## U



## V



## W



## X



## Numerics

# Reader's Comments

**Network Control Program**
**System Support Programs**
**Emulation Program**
**Generation and Loading Guide**
**NCP Version 5 Release 4**
**SSP Version 3 Release 6**
**EP Release 9**
**Production Draft**

**Publication No. SC30-3348-4**

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply. If we have questions about your comment, may we call you? If so, please include your phone number.

Name

Address

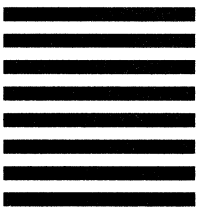
Company or Organization

Phone No.

Cut o
Alon

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



# BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-9990

Fold and Tape **Please do not staple** Fold and Tape



Cut c
Alon

# Reader's Comments

**Network Control Program**
**System Support Programs**
**Emulation Program**
**Generation and Loading Guide**
**NCP Version 5 Release 4**
**SSP Version 3 Release 6**
**EP Release 9**
**Production Draft**

**Publication No. SC30-3348-4**

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply. If we have questions about your comment, may we call you? If so, please include your phone number.

Name

Address

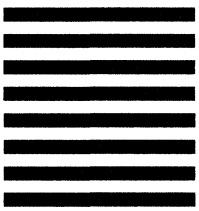
Company or Organization

Phone No.

Cut
Alor

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



# BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-9990

Fold and Tape **Please do not staple** Fold and Tape



Cut
Alor

# Reader's Comments

**Network Control Program**
**System Support Programs**
**Emulation Program**
**Generation and Loading Guide**
**NCP Version 5 Release 4**
**SSP Version 3 Release 6**
**EP Release 9**
**Production Draft**

**Publication No. SC30-3348-4**

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply. If we have questions about your comment, may we call you? If so, please include your phone number.

Name

Address

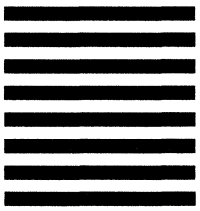
Company or Organization

Phone No.

Cut
Alo

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



# BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

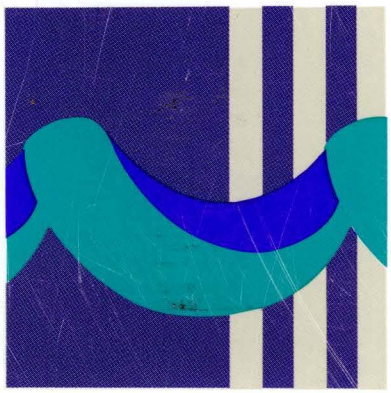
POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Information Development
Department E15
PO BOX 12195
RESEARCH TRIANGLE PARK, NORTH CAROLINA 27709-9990

Fold and Tape **Please do not staple** Fold and Tape



Cut
Alo



File Number: S370/30xx/4300/9370

Printed in USA

SC30-3348-04