Transaction Processing Facility

# ACF/SNA Network Generation

*Version 4 Release 1*

SH31-0131-14

Transaction Processing Facility

# ACF/SNA Network Generation

*Version 4 Release 1*

SH31-0131-14

**Note!**

Before using this information and the product it supports, be sure to read the general information under “Notices” on page ix.

**Fifteenth Edition (June 2002)**

This is a major revision of, and obsoletes, SH31-0131-13 and all associated technical newsletters.

This edition applies to Version 4 Release 1 Modification Level 0 of IBM Transaction Processing Facility, program number 5748-T14, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

IBM welcomes your comments. Address your comments to:

IBM Corporation
TPF Systems Information Development
Mail Station P923
2455 South Road
Poughkeepsie, NY 12601-5400
USA

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

# Contents

# Figures

# Tables



# Notices

References in this book to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service in this book is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Department 830A
Mail Drop P131
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this book to non-IBM Web sites are provided for convenience only and do not in any way serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this book or accessed through an IBM Web site that is mentioned in this book.

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Peer-to-Peer Networking
APPN
CICS
DB2
ESCON
IBM
MVS/XA
RISC System/6000
VTAM.

# About This Book

This book describes how to define a Systems Network Architecture (SNA) data communications network for use with the TPF system.

TPF Network Extension Facility (NEF) support depends on either:
- The Network Extension Facility (NEF2 PRPQ P85025), or
- Airlines Line Control Interface (ALCI) feature of ACF/NCP.

All subsequent occurrences of NEF in the text of this book are superseded by the previous information.

TPF Network Control Program (NCP) support depends on:
- Subarea support (T5), which requires SNA Network Interconnection (SNI)
- T2.1 low-entry networking (LEN) support, which requires NCP Version 4 Release 3, or later
- T2.1 Advanced Peer-to-Peer Networking (APPN) support, which requires NCP Version 6 Release 2, or later.

All subsequent occurrences of NCP in the text of this book are superseded by the previous information, unless otherwise specified.

In this book, abbreviations are often used instead of spelled-out terms. Every term is spelled out at first mention followed by the all-caps abbreviation enclosed in parentheses; for example, Systems Network Architecture (SNA). Abbreviations are defined again at various intervals throughout the book. In addition, the majority of abbreviations and their definitions are listed in the master glossary in the *TPF Library Guide*.

## Before You Begin

Before you use this book, review *TPF ACF/SNA Data Communications Reference*.

## Who Should Read This Book

This book is intended for system programmers who are responsible for SNA data communications support.

## Conventions Used in the TPF Library

The TPF library uses the following conventions:

| Conventions | Examples of Usage |
|---|---|
| *italic* | Used for important words and phrases. For example:<br>A *database* is a collection of data.<br>Used to represent variable information. For example:<br>Enter **ZFRST STATUS MODULE** *mod*, where *mod* is the module for which you want status. |

<table>
<tr><th>Conventions</th><th>Examples of Usage</th></tr>
<tr><td>bold</td><td>Used to represent text that you type. For example:<br>Enter ZNALS HELP to obtain help information for the ZNALS command.<br><br>Used to represent variable information in C language. For example:<br>level</td></tr>
<tr><td><code>monospaced</code></td><td>Used for messages and information that displays on a screen. For example:<br><code>PROCESSING COMPLETED</code><br><br>Used for C language functions. For example:<br><code>maskc</code><br><br>Used for examples. For example:<br><code>maskc(MASKC_ENABLE, MASKC_IO);</code></td></tr>
<tr><td>bold italic</td><td>Used for emphasis. For example:<br>You must type this command exactly as shown.</td></tr>
<tr><td><u>Bold underscore</u></td><td>Used to indicate the default in a list of options. For example:<br>Keyword=OPTION1 | <u>DEFAULT</u></td></tr>
<tr><td>Vertical bar |</td><td>Used to separate options in a list. (Also referred to as the OR symbol.) For example:<br>Keyword=Option1 | Option2<br><br>Note: Sometimes the vertical bar is used as a pipe (which allows you to pass the output of one process as input to another process). The library information will clearly explain whenever the vertical bar is used for this reason.</td></tr>
<tr><td>CAPital LETters</td><td>Used to indicate valid abbreviations for keywords. For example:<br>KEYWord=option</td></tr>
<tr><td>Scale</td><td>Used to indicate the column location of input. The scale begins at column position 1. The plus sign (+) represents increments of 5 and the numerals represent increments of 10 on the scale. The first plus sign (+) represents column position 5; numeral 1 shows column position 10; numeral 2 shows column position 20 and so on. The following example shows the required text and column position for the image clear card.<br><code>|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...</code><br><br><code>LOADER   IMAGE CLEAR</code><br>Notes:<br>1. The word LOADER must begin in column 1.<br>2. The word IMAGE must begin in column 10.<br>3. The word CLEAR must begin in column 16.</td></tr>
</table>

## How to Read the Syntax Diagrams

This section describes how to read the syntax diagrams (informally called *railroad tracks*) used in this book.

- Read the diagrams from left-to-right, top-to-bottom, following the main path line. Each diagram begins on the left with double arrowheads and ends on the right with 2 arrowheads facing each other.

```
►►─┤ Syntax Diagram ├──────────────────────────────────────────────────►◄
```

- If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.

```
►►─┤ The first line is long and extends the width of the diagram ├────────►

►─┤ Second Line ├──────────────────────────────────────────────────────►◄
```

- A word in all uppercase is a parameter that you must spell ***exactly*** as shown.

```
►►──PARAMETER──────────────────────────────────────────────────────────►◄
```

- If you can abbreviate a parameter, the optional part of the parameter is shown in lowercase. (You must type the text that is shown in uppercase. You can type none, one, or more of the letters that are shown in lowercase.)

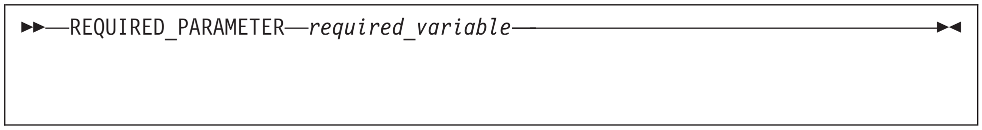
  **Note:** Some TPF commands are case-sensitive and contain parameters that must be entered exactly as shown. This information is noted in the description of the appropriate commands.

```
►►──PARAMeter──────────────────────────────────────────────────────────►◄
```

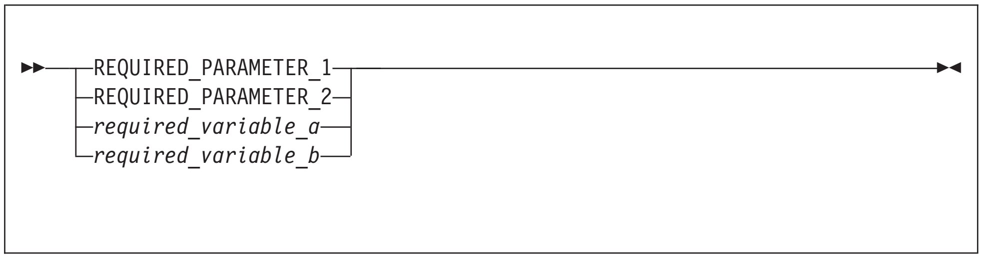
- A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

```
►►──variable───────────────────────────────────────────────────────────►◄
```

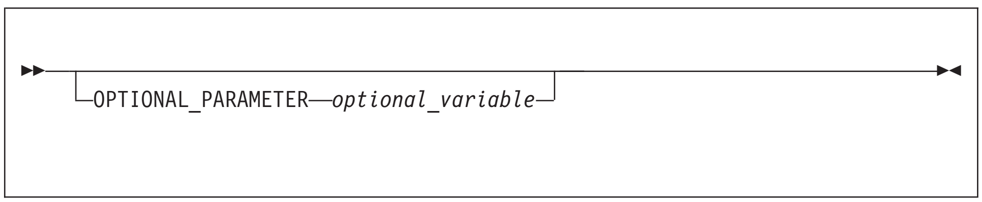
- Required parameters and variables are shown on the main path line. You must code required parameters and variables.



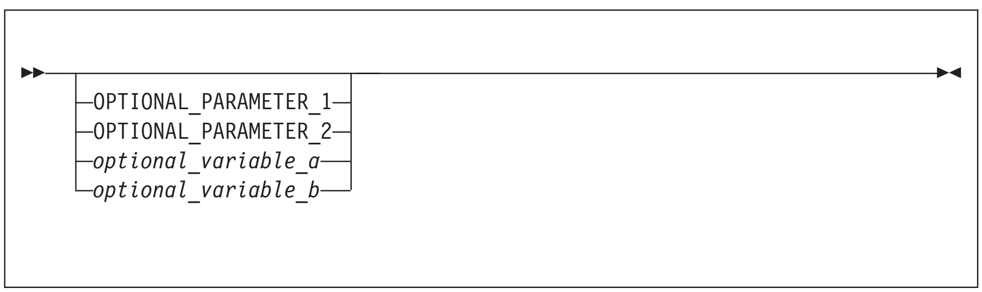

- If there is more than one mutually exclusive required parameter or variable to choose from, they are stacked vertically.



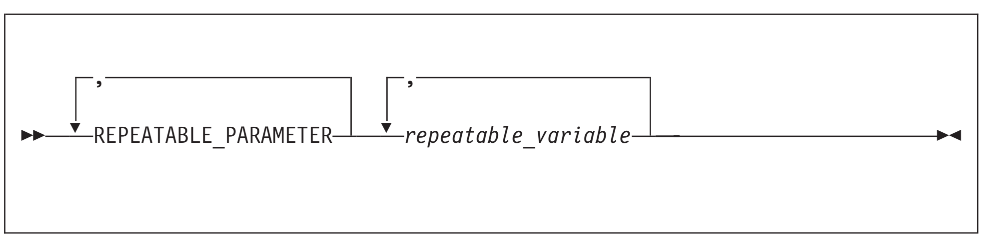

- Optional parameters and variables are shown below the main path line. You can choose not to code optional parameters and variables.




- If there is more than one mutually exclusive optional parameter or variable to choose from, they are stacked vertically below the main path line.




- An arrow returning to the left above a parameter or variable on the main path line means that the parameter or variable can be repeated. The comma (,) means that each parameter or variable must be separated from the next parameter or variable by a comma.




- An arrow returning to the left above a group of parameters or variables means that more than one can be selected, or a single one can be repeated.

```
     ┌─,──────────────────────────┐
     ▼                            │
►►──┬─REPEATABLE_PARAMETER_1─┬────┴───────────────────────────────────►◄
    ├─REPEATABLE_PARAMETER_2─┤
    └─repeatable_variable────┘
```

- If a diagram shows a blank space, you must code the blank space as part of the syntax. In the following example, you must code **PARAMETER** *variable*.

```
►►──PARAMETER── ──variable────────────────────────────────────────────►◄
```

- If a diagram shows a character that is not alphanumeric (such as commas, parentheses, periods, and equal signs), you must code the character as part of the syntax. In the following example, you must code **PARAMETER=(***begin***.***end***)**.

```
►►──PARAMETER=(begin.end)─────────────────────────────────────────────►◄
```

- Default parameters and values are shown above the main path line. The TPF system uses the default if you omit the parameter or value entirely.

```
     ┌─DEFAULT───┐   ┌─0────────┐
►►───┴───────────┴───┴──────────┴─────────────────────────────────────►◄
     └─PARAMETER─┘   └─variable─┘
```

- References to syntax notes are shown as numbers enclosed in parentheses above the line. Do not code the parentheses or the number.

```
                (1)
►►──PARAMETER─────────────────────────────────────────────────────────►◄
```

**Notes:**

1 An example of a syntax note.

- Some diagrams contain *syntax fragments*, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.

```
►►──┤ Reference to Syntax Fragment ├─────────────────────────────────────────►◄
```

**Syntax Fragment:**

```
├──1ST_PARAMETER,2ND_PARAMETER,3RD_PARAMETER─────────────────────────────────┤
```

## Related Information

A list of related information follows. For information on how to order or access any of this information, call your IBM representative.

## IBM Transaction Processing Facility (TPF) 4.1 Books

- *TPF ACF/SNA Data Communications Reference*, SH31-0168
- *TPF Application Requester User's Guide*, SH31-0133
- *TPF General Macros*, SH31-0152
- *TPF Migration Guide: Program Update Tapes*, GH31-0187
- *TPF Operations*, SH31-0162
- *TPF Programming Standards*, SH31-0165
- *TPF System Installation Support Reference*, SH31-0149.

## IBM Transmission Control Protocol/Internet Protocol (TCP/IP) Books

- *TPF Transmission Control Protocol/Internet Protocol*, SH31-0120.

## Miscellaneous IBM Books

- *NCP/SSP/EP Resource Definition Reference* (order the correct version and release for your installation).

## Online Information

- *Messages (Online)*
- *Messages (System Error and Offline)*.

## How to Send Your Comments

Your feedback is important in helping to provide the most accurate and highest quality information. If you have any comments about this book or any other TPF information, use one of the methods that follow. Make sure you include the title and number of the book, the version of your product and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

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.

- If you prefer to send your comments electronically, do either of the following:
  – Go to http://www.ibm.com/tpf/pubs/tpfpubs.htm.

There you will find a link to a feedback page where you can enter and submit comments.

  – Send your comments by e-mail to tpfid@us.ibm.com

- If you prefer to send your comments by mail, address your comments to:

  IBM Corporation
  TPF Systems Information Development
  Mail Station P923
  2455 South Road
  Poughkeepsie, NY 12601-5400
  USA

- If you prefer to send your comments by FAX, use this number:
  – United States and Canada: 1 + 845 + 432 + 9788
  – Other countries: (international code) + 845 + 432 +9788

# SNA Network Definition

This publication describes how you define the SNA network to which the TPF system will connect. To define the SNA network, do the following:

1. Update and assemble the SNA keypoint (CTK2), which uses the SNAKEY macro to describe the SNA network.
2. Define the appropriate SNA resources to the TPF system by preparing the input definition statements and running the offline ACF/SNA table generation (OSTG) program to create the SNA pilot tape or general data set (GDS).

The process for each of these steps is described in this publication.

In general, the SNA keypoint (CTK2, data macro CK2SN) contains global information that is used by the TPF system to set up the operating environment. This information is defined using the SNAKEY macro, and includes such things as:

- The maximum number of resources allowed in the TPF system
- Whether or not to include optional SNA facilities
- Other global information such as the shutdown and restart levels for SNA.

For more information about the SNAKEY macro, see “Updating and Assembling the SNA Keypoint” on page 3.

You can use the OSTG program to define the SNA resources to the TPF system. These resources include:

- TPF applications
- Remote terminals and applications
- Cross-domain resource manager (CDRM) resources
- Network control program (NCP) resources
- Adjacent link station (ALS) resources
- Channel-to-channel (CTC) resources.

**Note:** You can also use dynamic LU support and the ZNDYN ADD command to define ALS, CDRM, CTC, NCP, and remote LU resources to the TPF system. See *TPF ACF/SNA Data Communications Reference* for more information about dynamic LU support and defining resources to the TPF system.

The SNA network is typically owned and managed by VTAM, which is an IBM program product. The TPF system, therefore, has no need to know about the network topology. The TPF system does, however, need to know (through the OSTG program, ZNDYN ADD command, or dynamic LU support) about channel-attached NCP resources and the LU resources that will be in session with TPF applications. This means that you do not need to define the physical network (LINEs or PUs) to the TPF system.

Process selection vectors (PSVs) can be associated for some LUs using the OSTG program. Additional information about PSVs can be found in the *TPF ACF/SNA Data Communications Reference*. PSVs are defined in segment COBU using the IPSVE and IPSVT macros. For more information about the IPSVE and IPSVT macros, see *TPF General Macros*.

If you are running TPF Advanced Program-to-Program Communications (TPF/APPC) support, you must define the local TPF transaction programs that are activated by either the remote transaction programs, the TPF/APPC ACTIVATE_ON_CONFIRMATION verb, or the TPF/APPC ACTIVATE_ON_RECEIPT verb to the TPF system in the transaction program name table (TPNT). The TPNT resides in a user-replaceable program segment (CHQ0), and contains the following:

1. The transaction program name carried in the ATTACH message received from a remote LU 6.2 node, or issued on the TPF/APPC ACTIVATE_ON_CONFIRMATION verb or the TPF/APPC ACTIVATE_ON_RECEIPT verb.
2. The associated TPF E-type program segment that provides the application function.

| To define the TPNT tables, use the ITPNT macro as described in *TPF General Macros*.

| **Note:** To use TPF/APPC in a loosely coupled complex, you may need to add 2 entries to the TPNT for the service transaction program. See the progam considerations for the ITPNT macro for more information about these entries.

Any TPF program defined in the TPNT that is activated by a remote transaction program must be designed and implemented to accept the ATTACH interface. For more information about this interface, see *TPF ACF/SNA Data Communications Reference*.

# Updating and Assembling the SNA Keypoint

The SNA table sizes and network characteristics are defined by the SNAKEY macro and assembled offline into the SNA keypoint (CTK2). This process eliminates the need to regenerate the TPF system when the network configuration changes, simplifying installation and tuning. The value of certain selected parameters in the SNA keypoint can also be changed using the ZNKEY command. For more information about the ZNKEY command, see *TPF Operations*.

The SNA keypoint (CTK2) is a 4-KB record residing only in the basic subsystem (BSS). It is used to maintain the SNA network configuration and status. As a standard TPF keypoint, CTK2 has the following characteristics:

- The address of the SNA keypoint for each processor is contained in the master system keypoint (CTKX).
- It is demand keypointed.
- Fallback keypointing copies it to other DASD modules.

The SNA keypoint is defined by the CK2SN DSECT and is divided into three major parts:

- Part 1
  - Time stamp of the last build or modification.
  - SNA parameter list area. The contents of the SNA parameter list is defined in the SNAEQ macro and are initialized by SNA restart segments.
  - TPFAR storage allocation area table.
  - SNAKEY parameters.
  - Other system-wide communication configuration and status information.
- Part 2
  - SNA main storage allocation tables (MSAT1 and MSAT2). MSAT1 and MSAT2 contain a parameter list for main storage SNA control block requirements.

    These tables allow the initializer, CTIN, to allocate main storage and fill in the dump label table without needing to know the main storage calculations for SNA support.

    **Notes:**

    1. The subarea address table (SAT) is not user specified; it is always defined with 256 entries.
    2. The process selection vector (PSV) table area is not user specified; it is defined using the values from the SNAEQ macro.
    3. All areas except the PSV table and the sense (SNS) table can be in high storage above 16 MB.
  - Other communication-related main storage allocation tables (MSAT3). MSAT3 contains a parameter list for main storage control block requirements that are ***not*** related to SNA communications. For example, MSAT3 contains a parameter list for the main storage Transmission Control Protocol/Internet Protocol (TCP/IP) offload support and TPF Application Requester (TPFAR) control block structures.

    MSAT3 allows the initializer, CTIN, to allocate main storage and fill in the dump label table without needing to know the main storage calculations for the communication-related tables.

  Each parameter list entry in the MSAT contains the following:

– The 4-character dump label name of the control block; for example, CL4'SAT '.
– The size of an entry in the control block.
– The number of entries in the table to be allocated by CTIN.
– A 4-byte field indicating the total main storage allocated for the control block.
– A 4-byte field filled in by CTIN with the starting address of the allocated control block.
– A table validity indicator to support fast recovery processing.

• Part 3
  – The symbolic device address table (SDAT). The symbolic device address table is created by the system initialization package (SIP) with IODEV macros.

The SNA keypoint is shipped as source code with default values supplied by the SNAKEY macro. The SIP process generates the JCL required to assemble the keypoint. Your system programmer must modify the SNAKEY macro before this assembly step to reflect your configuration. This keypoint can also be changed and reloaded at other times without rerunning the entire SIP process.

Your system programmer can change online the value of certain selected parameters in the SNA keypoint with the ZNKEY command. Changes take effect either immediately, at cycle up, or after a restart depending on the parameters that were specified for the ZNKEY command. For more information about the ZNKEY command, see *TPF Operations*.

Figure 1 on page 5 shows how the required information is defined and passed through the various stages from SIP to the TPF online routines.

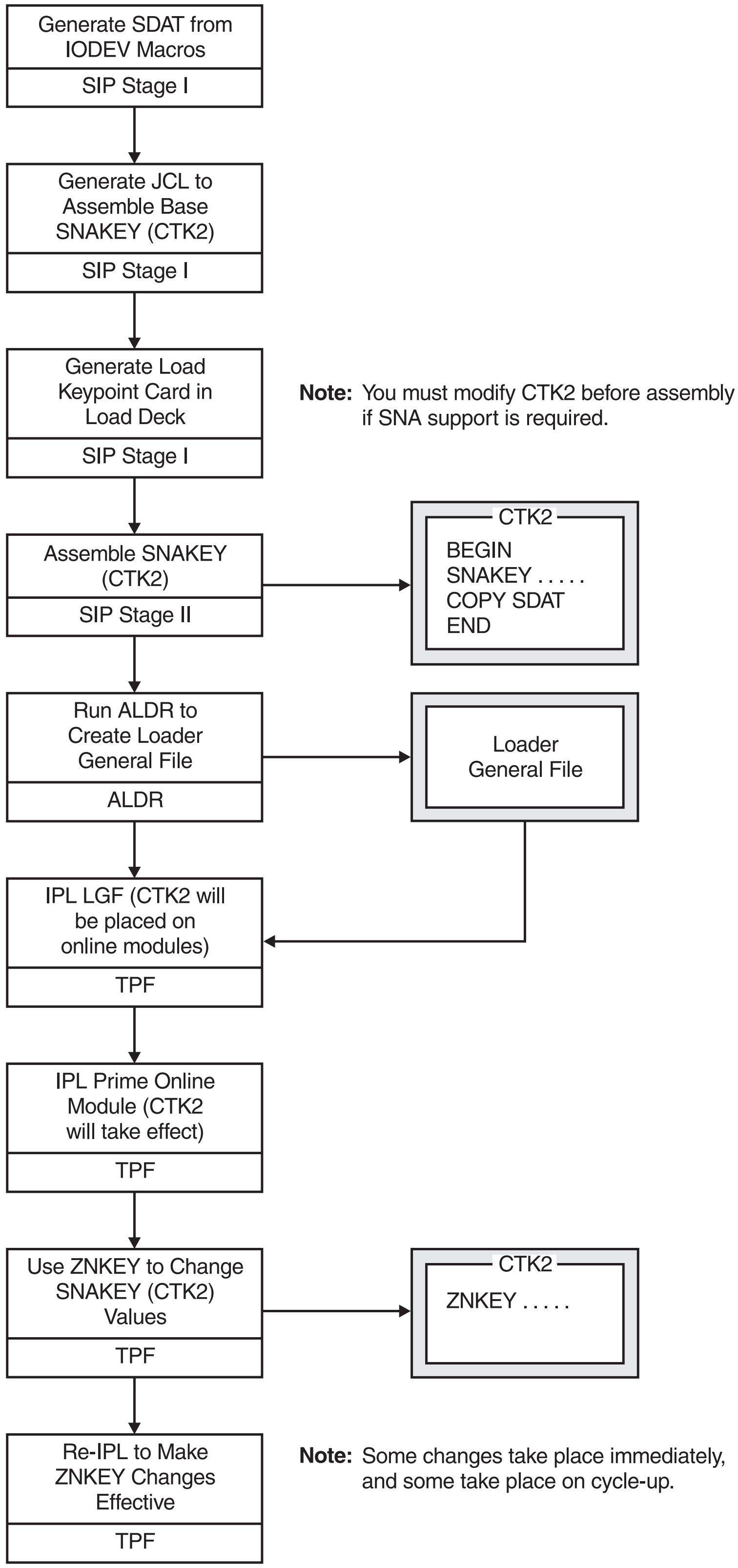



*Figure 1. SNA Communications Keypoint Generation*

## SNA Keypoint Definition

The TPF system ships a sample SNA keypoint source module. If you do not want to generate SNA support, use the default values for the MAXALS and MAXRVT parameters on the SNAKEY macro, which are zero. In order to reflect your actual environment, you must define the configuration of the system by doing one or both of the following:

1. Coding the SNAKEY macro to describe the size and characteristics of the SNA network and assembling a new SNA keypoint.
2. Modifying the SNA keypoint using the ZNKEY command to reflect the SNA network configuration.

## SNAKEY Macro

The SNAKEY macro defines the TPF SNA configuration. It also contains parameters for defining Transmission Control Protocol/Internet Protocol (TCP/IP) support.

This macro contains the information that is initializer dependent as well as the status and definition of the network. This allows the removal of the network configuration dependencies assembled in CTIN and provides an effective procedure for implementing configuration changes.

In a loosely coupled environment, one version of the SNA keypoint is loaded to each TPF processor. The keypoint actually becomes processor unique only after it is updated by the TPF processor.

All SNAKEY parameters are optional.

Table 1 on page 6 shows the minimum, maximum, and default values that are valid for the keypoint parameters.

Figure 2 on page 22 shows a sample SNAKEY definition.

*Table 1. SNAKEY Macro – Minimum, Maximum, and Default Values*

| **Parameter** | **Default** | **-----RANGE-----** | |
|---|---|---|---|
| | | **Minimum** | **Maximum** |
| *Table Allocation* | | | |
| MAXALS | 0 | 0 | 255 (See note 1.) |
| MAXCCB | 0 | 3 | 16777215 (See note 2.) |
| MAXCDRSC | 0 | 0 | 8388607 |
| MAXCTC | 0 | 0 | 255 (See note 1.) |
| MAXHCT | 0 | 0 | 65535 |
| MAXPCID | 127 | 127 | 32767 |
| MAXPRIM | MAXRVT/10 | 0 | MAXRVT (See note 8.) |
| MAXRVT | 0 | 0 | 8388607 |
| MAXSCB | 0 | 0 | 8388607 |
| MAXSDD | 0 | 0 | 80 |
| MAXSID | 0 | 0 | 65022 |
| MAXSNF | 1024 | 1024 | 16384 |

*Table 1. SNAKEY Macro – Minimum, Maximum, and Default Values (continued)*

| **Parameter** | **Default** | **-----RANGE-----** | |
|---|---|---|---|
| | | **Minimum** | **Maximum** |
| MAXSRT | 0 | 0 | 32767 |
| MAXTPI | 0 | 0 | 8388607 |
| MQITRC | 1 | 1 | 255 |
| NUMALS | 255 | 0 | MAXRVT (See note 9.) |
| SNSESZ | 0 | 0 | 255 |
| TRACSZ | 1 | 1 | 255 |
| *Network Parameters* | | | |
| CTCANS | 30 | 2 | 32767 |
| CTCRBFR | 1 | 1 | 16 (See note 3.) |
| CTCTGANY | YES | Yes | |
| CTCWBFRS | 1 | 1 | 4080 (See note 4.) |
| DYNTO | 60 | 0 | 65535 |
| FMHDR | 03,80,00 | 2–16 hex characters | |
| HARDREC | NO | Yes or No | |
| HPFMMR | NO | Yes or No | (See note 6.) |
| HPRALIVE | 30 | 1 | 65535 |
| HPRMTSIZ | 0 | 0 | 65535 |
| HPRPST | 60 | 1 | 65535 |
| ILWPC | 0 | 0 | 99 |
| ILWPE | 0 | 0 | 99 |
| ILWPF | 0 | 0 | 99 |
| ILWPI | 0 | 0 | 99 |
| ILWPS | 0 | 0 | 99 |
| INREC | NO | Yes or No | |
| LENNETID | blanks | 0–8 alphanumeric characters | |
| LMSCTI | 60 | 1 | 32767 |
| LUBLKT | 1 | 1 | 100 |
| MAXBFRU | 16 | 1 | 32 (See note 5.) |
| MAXHPRSA | 0 | 0 | 16777215 |
| MAXRTPCB | 0 | 0 | 2500000 |
| MAXSMTB | 0 | 0 | 5 |
| NBLKLU | 0 | 0 | 65535 |
| NETID | blanks | 0–8 alphanumeric characters | |
| OLDAPPL | NO | Yes or No | |
| PARACOS | | | (See note 7.) |
| PIUTAPEQ | 20 | 20 | 255 |
| RECIT | 2,30 | 1,1 | 255255 |

*Table 1. SNAKEY Macro – Minimum, Maximum, and Default Values (continued)*

| **Parameter** | **Default** | **-----RANGE-----** | |
|---|---|---|---|
| | | **Minimum** | **Maximum** |
| RECOP | 900 | 1 | 32767 |
| RECOT | 60 | 1 | 32767 |
| RSPTO | 30 | 1 | 32767 |
| RTPRSYNC | YES | Yes or No | |
| RVTCTRL | 20000 | 0 | 8388607 |
| SINGMODE | TPFLU62 | 0–8 alphanumeric characters | |
| SLOWTIME | 100 | 0 | 32767 |
| SNAPOLL | 5 | 1 | 5 |
| SNDWN | 11 | 11 | 32766 |
| SNKEY | 60 | 1 | 32767 |
| SNQDPT | 8 | 8 | 50 |
| SNRST | 12 | 12 | 32767 |
| SNSETO | 0 | 0 | 360 |
| TPALLOC | 180 | 1 | 32767 |
| TPRECV | 180 | 1 | 32767 |
| TPWAIT | 180 | 1 | 32767 |
| TRANA | NO | Yes or No | |
| UNITSZ | 256 | 104 | 4096 (See note 5.) |
| VRRTO | 10 | 0 | 120 |
| *TCP/IP Support Parameters* | | | |
| MAXASCU | 0 | 0 | 16777215 |
| MAXMATIP | 0 | 0 | 16777215 |
| SOCKSWP | 0 | 0 | 60 (See note 10.) |
| *TCP/IP Offload Support Parameters* | | | |
| CLAWADP | 0 | 0 | 84 |
| CLAWFD | 0 | 0 | 1680000 |
| CLAWIP | 0 | 0 | 6804 |
| *TCP/IP Native Stack Support Parameters* | | | |
| IPMTSIZE | 0 | 0 | 65535 (See note 11.) |
| IPRBUFFS | 16 | 8 | 32 |
| IPRBUFSZ | 1024 | 1024 | 4096 (See note 12.) |
| IPTOS | 0 | 0 | 255 (See note 15.) |
| IPTRCNUM | 0 | 0 | 10 (See notes 15 and 16.) |
| IPTRCSIZ | 0 | 0 | 100 (See notes 15 and 16.) |
| MAXIPCCW | 0 | 0 | 200 (See note 11.) |
| MAXOSA | 0 | 0 | 30 (See notes 14 and 15.) |
| MAXRTE | 0 | 0 | 2048 (See note 13.) |
| MAXSOCK | 0 | 0 | 1048576 (See note 11.) |

Table 1. SNAKEY Macro – Minimum, Maximum, and Default Values (continued)

| Parameter | Default | -----RANGE----- | |
|---|---|---|---|
| | | **Minimum** | **Maximum** |
| OSABUFF | 16 | 16 | 64 (See note 19.) |
| *Shared SSL Session Support Parameters* | | | |
| SSLPROC | 0 | 0 | 16 (See note 17.) |
| SSLTHRD | 0 | 0 | 32 (See note 18.) |

## Notes on Minimum, Maximum, and Default SNAKEY Values

1. The sum of MAXALS and MAXCTC must be less than 256. If MAXRVT=0, MAXALS and MAXCTC will also be set to 0.
2. If you specify a value for the MAXCCB parameter, the minimum value required is 3. The MAXCCB value includes a header and a trailer conversation control block (CCB) entry. Therefore, the actual number of CCB entries available is equal to the MAXCCB value minus 2. If you do not specify a value, the default is 0. This indicates that TPF/APPC support is not being used.
3. The number of 4-KB buffers allocated for channel-to-channel (CTC) input is sent as part of the exchange ID (XID) values when the CTC link station is activated. CTCRBFR specifies the number of input buffers per CTC link.
4. The number of 4-KB buffers allocated for CTC output is determined from the value of CTCWBFRS. The number of 4-KB buffers actually used for output to a CTC link is determined from the remote node as part of the XID activation sequence. If the number of output buffers requested in the XID is greater than the number available, the TPF system rejects the request for link activation. CTCWBFRS specifies the number of total output buffers for all CTC links.
5. The input area size used by the TPF system to read data from the 37*x*5 is determined from the UNITSZ and MAXBFRU parameters. This size is calculated by multiplying the UNITSZ value by the value of MAXBFRU. The minimum input area size is 1037 bytes, which is comprised of the following:
   a. 4 bytes for a link header
   b. 6 bytes for a transmission header
   c. 3 bytes for a request header
   d. 1024 bytes for the request unit.

   Therefore, if the value of MAXBFRU × UNITSZ is less than 1037 bytes, UNITSZ is set to 1037 / MAXBFRU and rounded up to the next integer.
6. The high-performance functional management message router (HPFMMR) affects a base-only system. HPO systems only use high-performance FMMR.
7. The class of service name used by TPF/APPC parallel sessions defaults to blank. When the TPF system acts as a PU 5 node and receives a CDINIT request, a class of service name is returned in the CDINIT response if it is a TPF/APPC parallel session. A class of service name can be specified with a string of as many as 8 characters (alphabetic, numeric, #, $, or @), starting with a nonnumeric character.
8. If the value specified for the MAXPRIM parameter is not a prime number, the value is rounded down to the largest prime number that is less than the value specified. The same is true when calculating the default value. If MAXRVT/10 is not a prime number, the default value is rounded down to the largest prime number that is less than MAXRVT/10.

9. The value of the NUMALS parameter cannot be greater than the value of the MAXRVT parameter. The default value for the NUMALS parameter is 255, unless 255 is less than MAXRVT/2. In this case, the default value is MAXRVT/2.
10. If the value specified for the SOCKSWP parameter is zero, the TCP/IP socket sweeper is disabled. See *TPF Transmission Control Protocol/Internet Protocol* for more information about the socket sweeper.
11. The values for IPMTSIZE, MAXIPCCW, and MAXSOCK must all be 0, or the values must all be greater than 0.
12. Valid values for the IPRBUFSZ parameter are 1024, 2048, or 4096.
13. If the value for IPMTSIZE, MAXIPCCW, or MAXSOCK is 0, MAXRTE will also be set to 0.
14. If the value specified for the MAXOSA parameter is zero, Open Systems Adapter (OSA)-Express support is disabled in the TPF system. See *TPF Transmission Control Protocol/Internet Protocol* for more information about OSA-Express support.
15. If the value specified for the MAXSOCK parameter is 0, the IPTOS, IPTRCNUM, IPTRCSIZ, and MAXOSA parameters will also be set to 0.
16. If the value specified for IPTRCNUM or IPTRCSIZ is 0, both parameters will be assigned a value of 0.
17. If the value for SSLPROC is 0, SSLTHRD must also be set to 0.
18. If the value for SSLTHRD is 0, SSLPROC must also be set to 0.
19. Valid values for the OSABUFF parameter are 16, 32, or 64.

## TPFAR Storage Allocation Area Table Parameters

**MAXHCT**

Specifies the maximum number of hotcon table (HCT) entries between relational databases and the TPF Application Requester (TPFAR) feature that the TPF system maintains. The HCT contains TPFAR communication parameters that are saved as an entry in the HCT for later use by another transaction.

*Hotcons* are described as follows depending on the communication protocol used:

- A hotcon for LU 6.2 is a TPF/APPC conversation that remains allocated and active past the completion of the transaction; that is, active until the SNA session is deactivated or the conversation is deallocated by the remote transaction program. The TPF/APPC conversation parameters between TPFAR and the DB2 system are saved in an entry in the HCT. When another ECB requests a conversation with the same remote application server, TPFAR reuses the active conversation.
- A hotcon for TCP/IP is a TCP/IP connection that remains active past the completion of the transaction. The socket descriptors are saved in an entry in the HCT. When another ECB requests a connection with the same remote application server, TPFAR reuses the active connection.

The value specified for the MAXHCT parameter must be less than or equal to the value specified for the MAXCCB parameter −2 plus the value specified for the MAXSOCK parameter. In addition, the value specified for MAXHCT should be equal to or greater than the sum of the values specified for the MAXHC parameter of the ZSQLD command. For more information about the maximum number of hotcons, see *TPF Operations* and the *TPF Application Requester User's Guide*.

**MAXSDD**
Specifies the maximum number of structured query language (SQL) database management systems (DBMS) that are defined to the TPF system by the ZSQLD command. For more information about the ZSQLD command, see *TPF Operations*.

## SNA Main Storage Allocation Table Parameters

**HPRMTSIZ**
Specifies the number of 4-KB frames in the high-performance routing message table (HPRMT). You can specify a nonzero value for this parameter only if you also specify a nonzero value for the MAXRTPCB and MAXHPRSA parameters.

See *TPF ACF/SNA Data Communications Reference* for more information about the HPRMT.

**MAXALS**
Specifies the maximum number of channel-attached 37*x*5 and 3174 controllers that can be physically attached to the TPF system and running at the same time. This number is used to allocate buffers (CCW areas) and the session index table (SIT) for each NCP or 3174 Advanced Peer-to-Peer Networking (APPN) controller. See the description of the MAXSID parameter for information about determining the size of the SIT.

**MAXCCB**
Specifies the number of conversation control block (CCB) entries that can be active in a TPF system. The default value of zero indicates that TPF/APPC is not being used. Values 1 and 2 are not valid.

**MAXCDRSC**
Specifies the maximum number of cross-domain resources by processor. This parameter is required when the TPF system can connect to the network as a PU 5 node. The number specified must be equal to or greater than the sum of the following:

- Number of TPF local applications
- Maximum number of cross-domain sessions you plan to have active at one time
- Number of cross-domain resource managers that can communicate with the TPF system as a PU 5 node.

This is the value used to allocate main storage for the network address table (NAT).

**MAXCTC**
Specifies the maximum number of channel-attached channel-to-channel (CTC) link stations that can simultaneously be physically attached to the SNA CTC connection and running.

**MAXHPRSA**
Specifies the number of entries in the high-performance routing session address table (HPRSAT). This is the maximum number of high-performance routing (HPR) LU-LU sessions per TPF processor. You can specify a nonzero value for this parameter only if you also specify a nonzero value for the MAXRTPCB parameter.

**Note:** Entries in the HPRSAT are located using a hashing algorithm. To optimize the distribution of entries in the HPRSAT, specify a prime number for this parameter.

See *TPF ACF/SNA Data Communications Reference* for more information about the HPRSAT.

**MAXPCID**

Specifies the number of procedure correlation identification (PCID) table entries. Entries in the PCID table are used for APPN LU-LU sessions that are being activated. Once the LU-LU session becomes active, the PCID table entry can be reused. This table is maintained only on the TPF processor that has the CP-CP sessions; however, the table contains an entry for each APPN LU-LU session that is being activated with any TPF processor in the loosely coupled complex.

**MAXPRIM**

Specifies the number of resource name hash prime table (RNHPT) entries. For more information about RNHPT entries, see *TPF ACF/SNA Data Communications Reference*.

See note 8 in “Notes on Minimum, Maximum, and Default SNAKEY Values” on page 9 for more information about the MAXPRIM parameter.

**MAXRTPCB**

Specifies the number of entries in the rapid transport protocol control block (RTPCB) table. This is the maximum number of rapid transport protocol (RTP) connections per TPF processor. You can specify a nonzero value for this parameter only if you also specify a nonzero value for the MAXHPRSA parameter.

See *TPF ACF/SNA Data Communications Reference* for more information about the RTPCB table.

**MAXRVT**

Specifies the number of entries to create in the RVT. This value corresponds to the maximum number of resources that can be defined to the TPF system at any one time.

An RVT entry is defined as a *spare RVT entry* until a resource definition is created for that RVT entry. A resource definition is created by using the OSTG program, the ZNDYN ADD command, or dynamic LU support.

See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about spare RVT entries and dynamic LU support.

Resource definitions are created for the following resources:

- Each TPF system services control point (SSCP)
- TPF control point (CP)
- Each TPF application
- Each TPF SLU appearance of a TPF application
- Each TPF functional management message router (FMMR)
- Each TPF processor NEF LU
- Each 37*x*5/3174 channel attached to the TPF system as a PU 2.1 link station
- Each control LU (CLU) associated with a PU 2.1 link station
- Each Logon Manager that can communicate with the TPF processor
- Each 37*x*5 channel attached to the TPF system as a PU 4 node
- Each remote LU
- Each remote SSCP.

**Notes:**

1. If MAXALS and MAXRVT are equal to zero, main storage is not allocated for 37*x*5 I/O areas and a single *end of table* RVT entry is created. These I/O areas (CCW areas) are also allocated for 3174 APPN connections. Any further references to 37*x*5 I/O areas or CCW areas should be considered to include 3174 APPN controllers.
2. If you decrease the value of this parameter, enter the ZNOPL BUILD command ***before*** you perform an initial program load (IPL) to load the new version of the SNA keypoint (CTK2). This will force the TPF system to perform a fresh load during the IPL. Otherwise, an error will occur.

   If you increase the value of this parameter, you do not need to perform a fresh load.

**MAXSCB**

Specifies the number of session control block (SCB) entries that can be active in the TPF system. The default value of zero indicates that TPF/APPC is not being used.

**Note:** There is an interdependency between CCBs and SCBs. If you want to use TPF/APPC support, you must code both MAXCCB and MAXSCB greater than 0.

**MAXSID**

The session index table (SIT) is used by the TPF system to correlate session identifiers (SIDs) to session control blocks (or RVTs). SIDs are assigned by the BIND sender, the TPF system, or the NCP. The value specified must be equal to or greater than the maximum number of PU 2.1 sessions that will flow through any channel-attached 37*x*5 or 3174 controller. This value is used to allocate an SIT as a CCW extension for every 37*x*5 or 3174 controller.

A value of 0 indicates that the TPF system cannot attach to the network as a PU 2.1 node. Because CTC links between PU 2.1 nodes are not supported, the SIT is not used for CTC links. Storage is allocated for MAXALS SITs.

**Note:** For 3174 connections, SIDs X'0101'–X'01FF' are reserved by the 3174. To compensate for this, you must add an additional 255 SIDs when specifying the MAXSID parameter.

**MAXSNF**

Specifies the number of slots in the sequential number field table. This table is used to correlate responses to TPF requests sent over a CDRM-CDRM session from one LU to another LU. The table contains the RID of the LU, which is found by using the session sequence number sent in the original request and returned in the corresponding response PIU.

**MAXSRT**

Specifies the number of system recovery table (SRT) entries generated in a processor. The default value of zero indicates that the TPF message recovery facility will not be invoked.

**MAXTPI**

Specifies the number of transaction program instance control block (TPICB) entries that can be active in the TPF system. For the TPF Application Requester (TPFAR) feature, one TPICB block is needed for each conversation that is active with the remote application server (AS). These active conversations also include those already saved as *hotcons* in the hotcon table (HCT). See the *TPF Application Requester User’s Guide* for more information.

**MQITRC**
Specifies the size of the MQI trace table in increments of 4-KB page units.

**NUMALS**
Specifies the number of entries to allocate in the non-LU section of the RVT. The non-LU section of the RVT contains the resource definitions for ALS, CDRM, CTC, and NCP resources. If an entry in the non-LU section of the RVT is not being used, it is referred to as a *spare RVT entry*.

**Note:** If you change this parameter to a value that requires the TPF system to reassign the RIDs, a fresh load is automatically performed when you load the new SNA keypoint (CTK2). If it is not necessary to reassign the RIDs, a nonfresh load is performed.

See note 9 in “Notes on Minimum, Maximum, and Default SNAKEY Values” on page 9 for more information about the NUMALS parameter.

**SNSESZ**
Specifies the number of sense table (SNS) entries that will be defined in the TPF system. The default value of zero indicates that the TPF sense table facility will not be called.

**TRACSZ**
Specifies the size of the main storage PIU trace table in increments of 4-KB page units.

## TPF Network Parameters

**CTCANS**
Specifies the number of seconds that the TPF system waits before declaring the other side inoperative. This is equivalent to the NCP Automatic Network Shutdown (ANS) or VTAM Missing Interrupt Handler (MIH) functions.

**CTCRBFR**
Specifies the number of 4-KB input buffers per CTC link that the TPF system allocates for receiving from a CTC link station.

**CTCTGANY**
Specifies that the TPF system supports any transmission group (TG) for CTC links. Always specify CTCTGANY=YES. (If you specify CTCTGANY=NO, the value is changed to CTCTGANY=YES.)

**CTCWBFRS**
Species the number of 4-KB output buffers that the TPF system allocates as a pool for sending data to SNA CTC link stations. This value is the total number of output buffers for all CTC links. This buffer pool is shared by all active CTC link stations.

**DYNTO**
Specifies the minimum time, in seconds, that an RVT entry must remain on the RVT termination list before it can be reused by a new logical unit that is logging on to the TPF system.

**FMHDR**
Specifies the FM header data to prefix system-generated messages to 3600 LUs. This header data allows 3600 programs to distinguish between TPF system-generated messages and application replies. The FM header description is placed in the SNA keypoint and SNA parameter list.

All entries must be 2 hexadecimal characters. The first entry is the total length of the header, including this length byte. It must be from 2–16. After the first length byte, the remainder contains as many as 15 additional bytes of FM header data.

**HARDREC**
Allows the TPF system to resynchronize the virtual route (VR) sequence number for each NCP using FID4 following a soft or a hard initial program load (IPL). When HARDREC=NO is specified, a DISCONTACT request is issued to each NCP using FID4 if the RVT entry indicates that the NCP is active and bound.

**HPFMMR**
Specifies if the high-performance functional management message router (HPFMMR) transmit process is used. HPFMMR uses an ENQC|DEQC method of serializing activity on the FMMR-FMMR session instead of using the NCB. HPFMMR allows multiple processors in a loosely coupled complex to transmit to the same destination simultaneously. See note 6 in “Notes on Minimum, Maximum, and Default SNAKEY Values” on page 9.

**HPRALIVE**
Specifies the value, in seconds, of the *alive timer*. The alive timer is used by the TPF system to detect failures for idle rapid transport protocol (RTP) connections in a high-performance routing (HPR) network. See *TPF ACF/SNA Data Communications Reference* for more information about the alive timer.

**HPRPST**
Specifies the number of seconds the TPF system will suggest the remote rapid transport protocol (RTP) endpoint use for its *path switch timer*. The path switch timer is used to detect path switch failures in a high-performance routing (HPR) network. See *TPF ACF/SNA Data Communications Reference* for more information about the path switch timer.

**ILWPC**
Specifies the percentage of common blocks that must be available in the TPF system. If the percentage falls below this value, the TPF system will not perform certain functions that require common blocks, such as sending virtual route (VR) pacing responses, until the minimum percentage is available again. This helps the TPF system avoid running out of critical resources.

This parameter is also used by adaptive rate-based (ARB) pacing to regulate the rate at which traffic is sent across high-performance routing (HPR) rapid transport protocol (RTP) connections. See *TPF ACF/SNA Data Communications Reference* for more information about ARB pacing and RTP connections.

If you specify a value of 0 for this parameter, the TPF system will not check the percentage of common blocks available and will use a default value of 20 to calculate ARB pacing.

**ILWPE**
Specifies the percentage of entry control blocks (ECBs) that must be available in the TPF system. If the percentage falls below this value, the TPF system will not perform certain functions that require ECBs, such as sending virtual route (VR) pacing responses, until the minimum percentage is available again. This helps the TPF system avoid running out of critical resources.

This parameter is also used by adaptive rate-based (ARB) pacing to regulate the rate at which traffic is sent across high-performance routing (HPR) rapid transport protocol (RTP) connections. See *TPF ACF/SNA Data Communications Reference* for more information about ARB pacing and RTP connections.

If you specify a value of 0 for this parameter, the TPF system will not check the percentage of ECBs available and will use a default value to calculate ARB pacing.

**ILWPF**
Specifies the percentage of frame blocks that must be available in the TPF system. If the percentage falls below this value, the TPF system will not perform certain functions that require frame blocks, such as sending virtual route (VR) pacing responses, until the minimum percentage is available again. This helps the TPF system avoid running out of critical resources.

This parameter is also used by adaptive rate-based (ARB) pacing to regulate the rate at which traffic is sent across high-performance routing (HPR) rapid transport protocol (RTP) connections. See *TPF ACF/SNA Data Communications Reference* for more information about ARB pacing and RTP connections.

If you specify a value of 0 for this parameter, the TPF system will not check the percentage of frames available and will use a default value to calculate ARB pacing.

**ILWPI**
Specifies the percentage of input/output blocks (IOBs) that must be available in the TPF system. If the percentage falls below this value, the TPF system will not perform certain functions that require IOBs, such as sending virtual route (VR) pacing responses, until the minimum percentage is available again. This helps the TPF system avoid running out of critical resources.

This parameter is also used by adaptive rate-based (ARB) pacing to regulate the rate at which traffic is sent across high-performance routing (HPR) rapid transport protocol (RTP) connections. See *TPF ACF/SNA Data Communications Reference* for more information about ARB pacing and RTP connections.

If you specify a value of 0 for this parameter, the TPF system will not check the percentage of IOBs available and will use a default value to calculate ARB pacing.

**ILWPS**
Specifies the percentage of system work blocks (SWBs) that must be available in the TPF system. If the percentage falls below this value, the TPF system will not perform certain functions that require SWBs, such as sending virtual route (VR) pacing responses, until the minimum percentage is available again. This helps the TPF system avoid running out of critical resources.

This parameter is also used by adaptive rate-based (ARB) pacing to regulate the rate at which traffic is sent across high-performance routing (HPR) rapid transport protocol (RTP) connections. See *TPF ACF/SNA Data Communications Reference* for more information about ARB pacing and RTP connections.

If you specify a value of 0 for this parameter, the TPF system will not check the percentage of SWBs available and will use a default value to calculate ARB pacing.

**INREC**
Specifies whether to create a safe store file copy of recoverable input messages. This parameter is ignored if MAXSRT=0 is coded.

**LENNETID**
Specifies the network identifier (NETID) of the TPF system when it is connected to the network as a PU 2.1 node. This network ID is used for both 37*x*5/VTAM and non-37*x*5 connections. If you do not specify a value for the LENNETID

parameter, the value stored in the SNA keypoint defaults to 8 blanks. The LENNETID parameter can be as many as 8 characters long.

**LMSCTI**
Specifies the time interval, in seconds, used to notify the Logon Manager (LM) of the number of active sessions on an ALS.

**LUBLKT**
Specifies the time interval before the LU blocking package transmits queued output data. For compatibility with NPSI, the time interval is specified in 100-ms increments. The default value is 1.

**MAXBFRU**
Specifies the number of input buffers the TPF system allocates for receiving data from channel attached 37*x*5s. This value is passed to the NCP at channel contact time as the number of read channel commands the TPF system will execute per channel program. (See note 5 in "Notes on Minimum, Maximum, and Default SNAKEY Values" on page 9 for more information about the MAXBFRU parameter.)

**MAXSMTB**
Specifies the number of 4-KB storage areas used per I-stream to hold the structured query language (SQL) trace information. If 0 is specified, no trace information is collected.

**NBLKLU**
Specifies the number of control blocks for the blocking of output messages destined for General Access to X.25 Transport Extensions (GATE)/ Fast Transaction Processing Interface (FTPI) resources. For detailed information about GATE and FTPI, see *TPF ACF/SNA Data Communications Reference*.

**NETID**
Specifies the network ID of the TPF system when it is attached as a PU 5 node. For example, if VTAM defines a network ID of TPFNET to the TPF system, specify TPFNET for this parameter. If you do not specify a value for this parameter, a default value of 8 blanks is stored in the SNA keypoint. There is an 8-character limitation for the NETID parameter.

**OLDAPPL**
Allows the TPF system to support ***old*** applications and SNA 3270 terminals, and to bypass the restriction for the multiple chained output message segments. When OLDAPPL=NO is specified, current 3270 message flow is not affected.

**PARACOS**
Specifies the class of service name that the TPF system uses when building a CDINIT response for a TPF/APPC parallel session.

**PIUTAPEQ**
Specifies the RTL tape queue threshold for writing to the PIU trace tape. If the RTL tape queue threshold is exceeded, an attention message is issued to the operator and PIU trace blocks are not written to tape until the queue count falls below the threshold.

**RECIT**
Specifies the activation frequency of the SNA input recovery timeout program. The default timeout value is the product of the following:

- Multiplier used to calculate the recovery timeout.
- Resolution factor, in seconds, use to calculate the recovery timeout.

This parameter is ignored if MAXSRT=0 (no message recovery) is specified.

**RECOP**
Specifies the number of seconds that the TPF system waits for a response to an output data message to a 3270 printer before taking recovery action.

**RECOT**
Specifies the number of seconds that the TPF system waits for a response to an output data message before taking recovery action.

**RSPTO**
Specifies the number of seconds that the TPF system waits for a response to a SNA command before the response is considered lost. This value should be greater than the maximum time that the NCP can attempt to send a command. This maximum NCP time is a function of the NCP system generation parameters REPLYTO=(a) and RETRIES=(m,t,n). For more information, see *NCP/SSP/EP Resource Definition Reference*.

A close approximation of this NCP time can be calculated using the following formula:

$$\text{Time} = (n \times t) + (m \times n \times a)$$

This, of course, does not include text transmission or internal NCP processing time. Failure to make this value large enough can cause retransmission of a command and result in incorrect processing in the NCP or cluster controller. Specifying too large a value will result in error conditions not being reported to the operator within an appropriate amount of time.

**RTPRSYNC**
Enables or disables the rapid transport protocol (RTP) resynchronization process. When enabled, the RTP resynchronization process attempts to keep RTP connections active after a hard initial program load (IPL) is performed.

See *TPF ACF/SNA Data Communications Reference* for more information about the RTP resynchronization process.

**RVTCTRL**
Specifies the number of entries that are processed when keypointing or validating the RVT or when searching the RVT for timed-out commands before control of the processor is given up.

**SINGMODE**
Specifies the mode name used for TPF/APPC single sessions started by the TPF system.

**SLOWTIME**
Specifies the number of SNA polling intervals that an ALS can be in slowdown mode before the TPF system deactivates the link. The SNA polling interval value is defined by the SNAPOLL parameter. The SLOWTIME parameter also defines the amount of time, in 10-ms intervals, that an Internet Protocol (IP) router is allowed to be in slowdown mode before the TPF system deactivates the link.

**SNAPOLL**
Specifies the SNA polling interval. The value can range from 1–5, where 1 represents 10 milliseconds (ms), 2 represents 20 ms, and so on, up to a maximum of 50 ms.

**SNDWN**
Specifies the SNA shutdown level. If the number of available system work blocks (SWBs) falls below the number specified for this parameter, RVT scan will not issue any command. This value must be less than the number specified for the SNRST parameter.

**SNKEY**
Specifies the time interval, in seconds, at which the SNA keypointable control blocks (for example, RVTs) will be written to file.

**SNQDPT**
Specifies the SOUTC queue depth for SNA output command handler, RVTSCAN. Specifying a higher value for this parameter can increase SNA outbound traffic flow under certain conditions.

**SNRST**
Specifies the SNA restart level. After the shutdown level is reached, this number specifies the number of system work blocks (SWBs) that must be available before RVT scan can issue any commands. This value must be greater than the number specified for the SNDWN parameter.

**SNSETO**
Specifies the number of seconds that the TPF system will collect the negative responses before issuing a message to indicate the total number received during this time period.

**TPALLOC**
Specifies the number of seconds that the TPF system waits for a remote LU to respond to a TPF/APPC ALLOCATE verb before timing out.

**TPRECV**
Specifies the number of seconds that the TPF system waits for data after a TPF/APPC RECEIVE verb or any verb that implies CONFIRM before timing out.

**TPWAIT**
Specifies the number of seconds that the TPF system waits for a remote LU to respond to a TPF/APPC WAIT verb before timing out.

**TRANA**
Specifies if the user-written transaction analysis exit must be called for each input message. This parameter is ignored if the message recovery facility was excluded from the TPF system by coding MAXSRT=0.

**UNITSZ**
Specifies the TPF input buffer size used for data transfer from a 37*x*5. This value is passed to the NCP at channel contact to specify the number of bytes allocated by the TPF system per NCP read command. See note 5 in “Notes on Minimum, Maximum, and Default SNAKEY Values” on page 9 for special coding considerations for the UNITSZ parameter. The recommended value for this parameter is a multiple of 8.

**VRRTO**
Specifies the number of seconds that virtual route (VR) resync waits before declaring a timeout. The default value is 10 seconds.

## TCP/IP Support Parameters

**MAXASCU**
Specifies the maximum number of Mapping of Airline Traffic over Internet Protocol (MATIP) agent set control units (ASCUs) that can be connected to the TPF system at one time.

**MAXMATIP**
Specifies the maximum number of MATIP sessions that can exist on the TPF system at one time.

**SOCKSWP**
Specifies, in minutes, the CRET timeout interval for the TCP/IP socket sweeper and the Secure Sockets Layer (SSL) sweeper.

See *TPF Transmission Control Protocol/Internet Protocol* for information about the values to specify for these parameters.

## TCP/IP Offload Support Parameters

**CLAWADP**
Specifies the number of Common Link Access to Workstation (CLAW) adapters.

**CLAWFD**
Specifies the number of file descriptors.

**CLAWIP**
Specifies the number of local Internet Protocol (IP) addresses.

See *TPF Transmission Control Protocol/Internet Protocol* for information about the values to specify for these parameters.

## TCP/IP Native Stack Support Parameters

**IPMTSIZE**
Specifies the size of the IP message table (IPMT) in 4-KB blocks.

**IPRBUFFS**
Specifies the number of read buffers assigned per read channel program to each IP router using channel data link control (CDLC) support.

**IPRBUFSZ**
Specifies the size of each read buffer for IP routers using CDLC support.

**IPTOS**
Specifies the type of service (TOS) value to use for the network priority of outbound TPF IP packets.

**IPTRCNUM**
Specifies the maximum number of individual IP traces that can be defined.

**IPTRCSIZ**
Specifies the size of each individual IP trace table, defined in 4-KB blocks.

**MAXIPCCW**
Specifies the maximum number of IP routers that can be active on the TPF system.

**MAXOSA**
Specifies the maximum number of Open Systems Adapter (OSA)-Express connection that can be active on the TPF system.

**MAXRTE**
Specifies the maximum number of IP routing table entries.

**MAXSOCK**
Specifies the maximum number of sockets using TCP/IP native stack support that can be active on the TPF system.

**OSABUFF**
Specifies the number of 64-KB read buffers assigned to each OSA-Express connection.

See *TPF Transmission Control Protocol/Internet Protocol* for information about the values to specify for these parameters.

### Shared SSL Session Support Parameters

**SSLPROC**
Specifies the number of Secure Sockets Layer (SSL) daemon processes. The SSLPROC and SSLTHRD parameters must both be 0 or a value other than 0. If you want to use shared SSL session support, you must set the SSLPROC and SSLTHRD parameters to values other than 0.

**SSLTHRD**
Specifies the number of thread ECBs per SSL daemon process. The SSLTHRD and SSLPROC parameters must both be 0 or a value other than 0. If you want to use shared SSL session support, you must set the SSLTHRD and SSLPROC parameters to values other than 0.

## Considerations for the MAXCTC, CTCRBFR, and CTCWBFRS Parameters

You should understand the relationship among the MAXCTC, CTCRBFR, and CTCWBFRS parameters:

- The value specified for the CTCRBFR (read buffers) and CTCWBFRS (write buffers) parameters must be proportional to the value specified for the MAXCTC parameter. This proportion represents the total number of frames available in the read and write buffers.
- The required number of write buffers (CTCWBFRS) depends on the number of read buffers (CTCRBFR) in the other side of the network.

The following examples provide some guidelines for you to use when determining the values for the CTCRBFR and CTCWBFRS parameters:

- You have 3 CTC links in your network (MAXCTC=3). The value of the CTCRBFR parameter is 10. There are the 2 standard read buffers per CTC link. The value of the CTCWBFRS parameter is 20.

  ```
  3 (MAXCTC) × 2 read buffers per link × 10 (CTCRBFR) = 60

  20 (CTCWBFRS) + 60 = 80 is total number of frames.
  ```
- Two TPF systems, TPFA and TPFB, are connected with 2 CTC links, CTC1 and CTC2. Both systems have the MAXCTC parameter set to 2. The TPFA system has the CTCRBFR parameter set to 10; the TPFB system has the CTCRBFR parameter set to 16. The number of frames automatically carved out on TPFA is:

  ```
  2 (MAXCTC) × 2 read buffers per link × 10 (CTCRBFR) =  40 frames
  ```

  The number of frames automatically carved out on TPFB is:

  ```
  2 (MAXCTC) × 2 read buffers per link × 16 (CTCRBFR) =  64 frames
  ```

  When the first link is activated (CTC1), the number of frames needed on the TPFA system is:

  ```
  2 read buffers × 10 (from CTCRBFR) frames per buffer       =  20

  1 write buffer with 16 frames                               =  16
      (found out at link activation time from the
       value of CTCRBFR on TPFB
  ```

```
                                          -------
              Total frames needed on TPFA      36
```

Because the second CTC link, CTC2, is between the same 2 systems, the same number of frames (36) is used. Therefore, in order to activate both links, there must be a minimum of 72 frames allocated.

Because 40 frames were initially allocated, the value of the CTCWBFRS parameter must be set to a minimum of 32 frames (72 − 40 = 32). However, if you knew that you would never have both CTC links activated at the same time, you could set CTCWBRFS to 0, because enough frames are already allocated (40) to bring up one link (36).

## SNAKEY Example

Figure 2 shows an example of how you can code the SNAKEY macro.

```
SNAKEY
       CLAWADP=3,        Maximum of 3 CLAW adapters
       CLAWFD=6000,      Maximum of 6000 file descriptors
       CLAWIP=12,        Maximum of 12 local IP addresses
       CTCRBFR=4,        Allocate 16K for receiving from a CTC link
       CTCTGANY=YES,     Use TG=ANY definition for CTC links
       CTCWBRFS=16,      Allocate 64KB buffer pool for all CTC output
       DYNTO=60,         Allow RVTs on the RVT termination list to be recycled after 60 seconds
       FMHDR=(03,81,02), FM header with 2 bytes of data
       HARDREC=YES,      Allows TPF to resynchronize the virtual route (VR) sequence
                         following a soft or a hard IPL.
       HPFMMR=YES,       Use the high performance FMMR transmit process
       HPRALIVE=30,      HPR alive timer is 30 seconds
       HPRMTSIZ=100,     Allocate 100 4-KB frames for the HPRMT
       HPRPST=60,        Path switch timer is 60 seconds
       ILWPC=20,         20% of the common blocks must be free for a VR pacing response
       ILWPE=42,         42% of the ECBs must be free for a VR pacing response
       ILWPF=25,         25% of the frame blocks must be free for a VR pacing response
       ILWPI=25,         25% of the IOB blocks must be free for a VR pacing response
       ILWPS=30,         30% of the SWBs must be free for a VR pacing response
       INREC=YES,        Include input recovery option
       IPMTSIZE=1000,    1000 4-KB blocks allocated to the IP message table for this processor
       IPRBUFFS=16,      16 read buffers allocated for each of the two read channel programs for an active
                         IP router that is using CDLC support
       IPRBUFSZ=1024,    Each read buffer allocated to an IP router using CDLC support is 1024 bytes
       IPTOS=2,          Use a value of 2 for the network priority of outbound TPF IP packets
       IPTRCNUM=5,       Maximum number of individual IP trace table definitions
       IPTRCSIZ=20,      20 4-KB blocks allocated for each individual IP trace table
       LENNETID=VTAMNET, Network ID for the TPF system as a PU 2.1 node
       LUBLKT=20,        Maximum of 2000 ms
       MAXALS=3,         Maximum of 3 37x5 resources for this processor
       MAXASCU=5000,     Maximum of 5000 MATIP ASCUs for this processor
       MAXBFRU=10,       Maximum of 10 37x5 input buffers
       MAXCCB=2000,      Maximum of 2000 CCB entries
       MAXCTC=2,         Maximum of 2 concurrently active CTC links
       MAXHCT=40,        Maximum of 40 hotcon table entries
       MAXHPRSA=500,     Number of entries in the HPRSAT
       MAXIPCCW=10,      Maximum of 10 concurrently active IP routers for this processor
       MAXMATIP=1000,    Maximum of 1000 MATIP sessions
       MAXOSA=30,        Maximum of 30 OSA-Express active connections
       MAXPCID=500,      Maximum of 500 PCID table entries
       MAXPRIM=293,      Number of RNHPT entries
```

*Figure 2. Sample SNA Keypoint Definition (Part 1 of 2)*

```
MAXRTE=500,        Maximum of 500 Internet Protocol (IP) routing table entries
MAXRTPCB=500,      Number of entries in the RTPCB table
MAXRVT=7500,       Maximum of 7500 RVT entries
MAXSCB=2000,       Maximum of 2000 SCB entries
MAXSDD=2,          Maximum of 2 SQL DBMS entries
MAXSID=2500,       Maximum of 2500 sessions per ALS
MAXSMTB=5,         Five 4KB storage areas for SQL trace areas
MAXSOCK=100,       Maximum of 100 active sockets using TCP/IP native stack support for
                   this processor
MAXSRT=200,        Maximum of 200 SRT entries
MAXTPI=255,        Maximum of 255 TPICB entries
MQITRC=4,          Number of 4KB pages in the MQI trace table
NBLKLU=5,          Maximum of 5 GATE/FTPI LUs
NETID=TPFNET,      TPF network ID as known by VTAM
NUMALS=200,        Number of entries in the non-LU section of the RVT
OLDAPPL=YES,       Allows TPF to support old applications and SNA 3270 terminals
OSABUFF=16,        Sixteen 64-KB read buffers allocated for each OSA-Express connection
PARACOS=MODE1,     Class of service name is MODE1
PIUTAPEQ=25,       PIU trace RTL tape queue threshold
RECIT=(30,3),      Input recovery timeout value
RECOP=900,         3270 printer timeout value
RSPTO=30,          Output recovery timeout value
RTPRSYNC=YES,      RTP resynchronization process is enabled
RVTCTRL=7000,      Give up processor control every 7000 RVT entries
SINGMODE=SINGLE1,  Single session mode name is SINGLE1
SLOWTIME=100,      ALS slowdown timeout is 100
SNAPOLL=4,         Set the SNA polling interval to 40 ms
SNDWN=250,         Shutdown SNA output command processing when 250 or
                   fewer SWBs are available
SNKEY=60,          Keypoint SNA tables at 60 second intervals
SNQDPT=15,         SOUTC queue depth for SNA output command handler, RVTSCAN
SNRST=300,         Restart SNA output command processing when 300 or
                   more 381-byte core blocks are available
SNSESZ=10,         Number of sense table (SNS) entries
SNSETO=5,          Time in seconds before the summary message
SOCKSWP=30,        CRET interval, in minutes, for TCP/IP socket sweeper
SSLPROC=1,         Number of SSL daemon processes
SSLTHRD=6,         Number of thread ECBs per SSL daemon process
TPALLOC=180,       TPF/APPC ALLOCATE timeout value
TPRECV=180,        TPF/APPC RECEIVE and CONFIRM timeout value
TPWAIT=180,        TPF/APPC WAIT timeout value
TRACSZ=3,          Number of 4KB pages in the PIU trace table
TRANA=NO,          Exclude transaction analysis option
UNITSZ=400,        400-Byte 37x5 input buffer
VRRTO=10           VR resync timeout value
```

*Figure 2. Sample SNA Keypoint Definition (Part 2 of 2)*

## Considerations for Defining Records for SNA Structures

The following information describes some SNAKEY-related considerations for defining the records for the SNA structures.

### #NCBRI and #SPARI Records

#NCBRI and #SPARI records are used only by resources that are defined using the OSTG program. Therefore, the number of #NCBRI and #SPARI records allocated during SIP must be equal to or greater than the number of resources that you plan to define using the OSTG program.

You can also use the ZNDYN ADD command and dynamic LU support to define resources to the TPF system. Resources defined using the ZNDYN ADD command or dynamic LU support do ***not*** use #NCBRI and #SPARI records.

See the following for more information:
- “Offline ACF/SNA Table Generation (OSTG) Program” on page 27 for more information about the OSTG program.
- *TPF Operations* for more information about the ZNDYN ADD command.
- *TPF ACF/SNA Data Communications Reference* for more information about dynamic LU support and defining resources to the TPF system.

## #CCBRU Records

Enough #CCBRU records must be allocated during system initialization program (SIP) processing to hold all the conversational control block (CCB) entries defined by the MAXCCB parameter. The minimum number of records can be determined by the following formula:

```
Number of CCB entries × Length of CCB entry
------------------------------------------------
                    4079
```

Where:

| | |
|---|---|
| `Number of CCB entries` | is the value specified for the MAXCCB parameter. |
| `Length of CCB entry` | is defined in data macro ICCB. |
| `4079` | is calculated as follows: 4 KB − header size. |

The #CCBRU fixed file records are used to keypoint (update) the CCB main storage control blocks.

## #RV1RU and #RV2RU Records

Enough #RV1RU and #RV2RU records must be allocated during SIP processing to hold all the resource vector table (RVT) entries defined by the MAXRVT parameter. The minimum number of records can be determined by the following formula:

```
Number of RVT entries × Length of RVT entry
------------------------------------------------
                 4079
```

Where:

| | |
|---|---|
| `Number of RVT entries` | is the value specified for the MAXRVT parameter. |
| `Length of RVT entry` | is defined in data macros RV1VT and RV2VT. |
| `4079` | is calculated as follows: 4 KB − header size. |

## #SRTRU Records

Enough #SRTRU records must be allocated during SIP processing to hold all the system recovery table (SRT) entries defined by the MAXSRT parameter. The minimum number of records can be determined by the following formula:

```
Number of SRT entries × Length of SRT entry
------------------------------------------------
                    4079
```

Where:

| | |
|---|---|
| `Number of SRT entries` | is the value specified for the MAXSRT parameter. |

| | |
|---|---|
| `Length of RVT entry` | is defined in data macro SR0RT. |
| `4079` | is calculated as follows: 4 KB − header size. |

## #SC1RU and #SC2RU Records

Enough #SC1RU and #SC2RU records must be allocated during SIP processing to hold all the session control block (SCB) entries defined by the MAXSCB parameter. The minimum number of #SC1RU records can be determined by the following formula:

```
Number of SCB entries × Length of SCB1 entry
--------------------------------------------
                 4079
```

Where:

| | |
|---|---|
| `Number of SCB entries` | is the value specified for MAXSCB + 1 for the SCB anchor. |
| `Length of SCB1 entry` | is defined in data macro ISCB. |
| `4079` | is calculated as follows: 4 KB − header size. |

The minimum number of #SC2RU records can be determined by the following formula:

```
Number of SCB entries × Length of SCB2 entry
--------------------------------------------
                 4079
```

Where:

| | |
|---|---|
| `Number of SCB entries` | is the value specified for the MAXSCB parameter. |
| `Length of SCB2 entry` | is defined in data macro ISCB. |
| `4079` | is calculated as follows: 4 KB − header size. |

The #SC1RU and #SC2RU fixed file records are used to keypoint the SCB main storage control blocks.

## #RT1RI and #RT2RI Records

Enough #RT1RI and #RT2RI records must be allocated during SIP to hold all the rapid transport protocol control block (RTPCB) entries defined by the MAXRTPCB parameter. See *TPF ACF/SNA Data Communications Reference* for information about how many of these records to define.

# Considerations for Defining Records for Internet Protocol (IP) Structures

The following information describes some SNAKEY-related considerations for defining the records for the IP structures.

## #IPRTE Records

Enough IP routing table fixed file records (#IPRTE) must be allocated during SIP to hold all the IP routing table entries defined by the MAXRTE parameter. The minimum number of records can be determined by the following formula, where `IPRT` means IP routing table:

```
   Number of IPRT entries × Length of IPRT entry defined in IPRTE
 ----------------------------------------------------------------------------
          4095 - Size of IPRT header area defined in IPRTE
```

Where:

| | |
|---|---|
| `Number of IPRT entries` | is the value specified for the MAXRTE parameter. |
| `Length of IPRT entry` | is defined in data macro IPRTE. |
| `Size of IPRT header area` | is defined in data macro IPRT. |

# Offline ACF/SNA Table Generation (OSTG) Program

The offline ACF/SNA table generation (OSTG) program is used to create an SNA communication configuration pilot tape or general data set (GDS) for the TPF system. The program runs under MVS.

**Note:** *MVS* is used except where it is necessary to distinguish between *MVS/XA* and *MVS/370.*

The OSTG program reads the input definition statements produced by SIP Stage I and the input definition statements that you code. It then produces a pilot tape or GDS, which contains the resource resolution table (RRT) records that are used to load resources to the online TPF system. In addition, several reports are produced to provide information about the generated SNA communications network.

## Using the OSTG Program

Figure 3 on page 28 shows how the required information is defined and passed from SIP to the TPF online routines through the various stages.

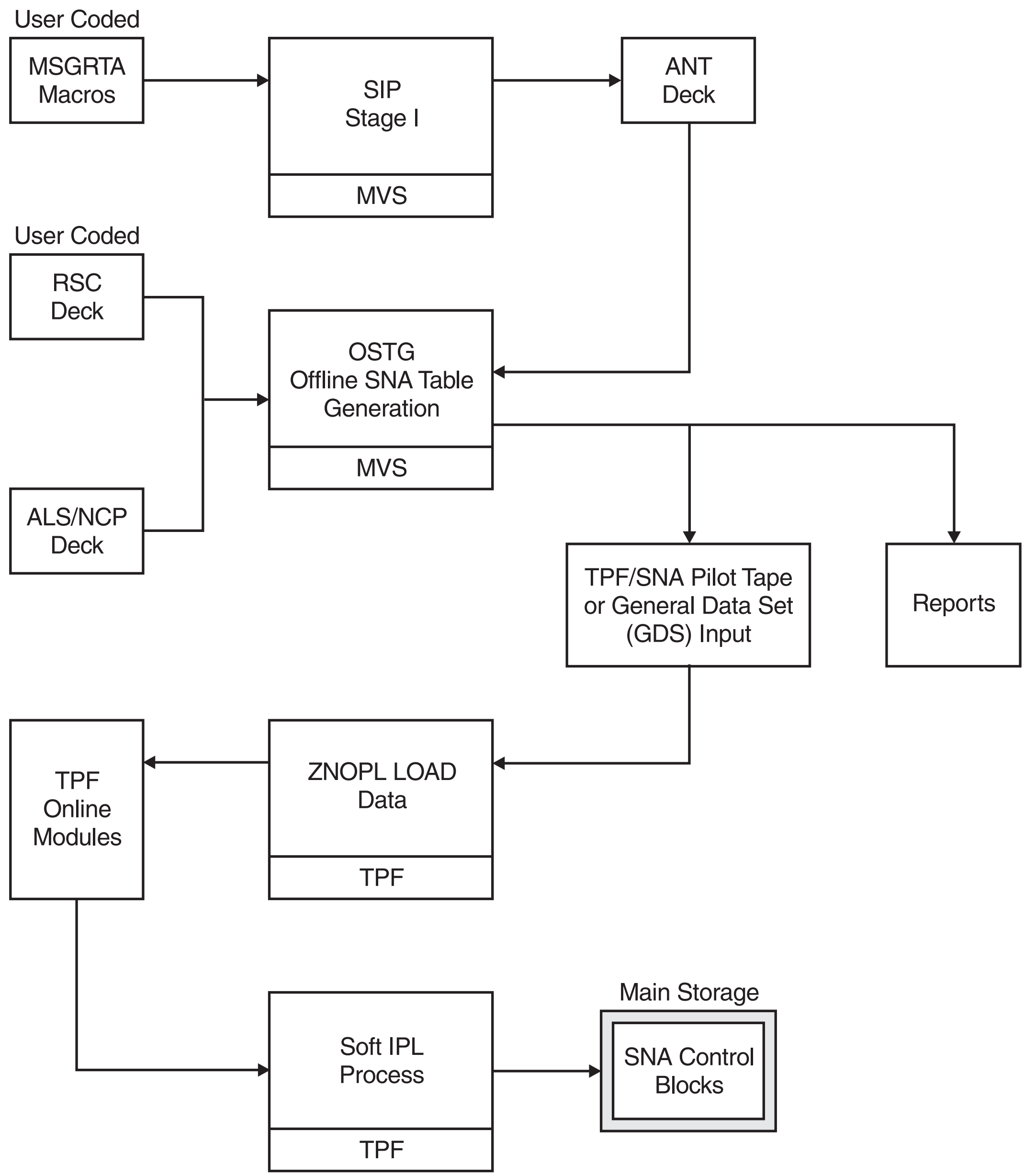



*Figure 3. OSTG Overview*

The SNA network definition process consists of the following steps:

1. Code the MSGRTA macros in the SIP process for the TPF applications that you want to define to the TPF system being generated. See *TPF ACF/SNA Data Communications Reference* for information about the applications to define to the TPF system using the OSTG program.

   The MSGRTA macros will cause the SIP process to generate the application name table (ANT) deck that is needed as input to the OSTG program. See “ANT Deck” on page 34 for information about the ANT deck.

2. Code the RSC deck for the CDRM resources and remote LU resources that you want to define to the TPF system being generated.

   See *TPF ACF/SNA Data Communications Reference* for information about the CDRM resources and remote LU resources to define to the TPF system using the OSTG program. See “RSC Deck” on page 37 for information about the RSC deck.

3. Code the ALS deck for the channel-attached NCP, CTC, or ALS resources that you want to define to the TPF system being generated. This includes the resources that will be connected as PU 4 nodes and the resources that allow the TPF system to connect as a PU 2.1 node.

   See *TPF ACF/SNA Data Communications Reference* for information about the NCP, CTC, and ALS resources to define to the TPF system using the OSTG program. See “ALS Deck” on page 45 for more information about the ALS deck.
4. Prepare the JCL to run the OSTG program. See Figure 15 on page 57 and Figure 16 on page 58 for examples of the JCL needed to run the OSTG program.
5. Run the OSTG job step and review the output reports. You may need to run the OSTG program several times to eliminate all of the errors that can occur when you define a large network.
6. After the OSTG program is run successfully with no errors, load the resulting pilot tape or GDS to the online TPF system.
7. Perform a soft initial program load (IPL) on the TPF system to include the new SNA network definitions.

## OSTG Data Sets

OSTG data sets consist of the input data set, several output data sets, and several work data sets, each of which is described in the following sections.

As a general rule, the logical record length (LRECL) for all of the data control blocks (DCBs) are defined in the OSTG program. The block size (BLKSIZE) and record format (RECFM), however, are usually not defined on the program DCB, which allows you to specify this information on the JCL DD statements for the data sets.

## OSTG EXEC PARM Field

The OSTG program uses the PARM field of the JCL EXEC statement to provide important information that is used to control the OSTG processing. The PARM field contains the following parameters:

**CPUID**
: Required. Specify the processor for which the table generation is being performed. When generating a loosely coupled complex, this must be the ID of the first processor specified in the loosely coupled complex. The processor ID that is specified must be the same as the processor ID specified in the ANTDEF statement. The processor ID is a 1-character alphanumeric value from A–Z or 0–9.

**DESC**
: Optional. Specify a description (as many as 25 bytes) for the OSTG input data sets. This description is included in the information that is displayed when you enter the ZNOPL STATUS command. See *TPF Operations* for more information about the ZNOPL STATUS command and an example of the information that is displayed.

**SDPSID**
: Required for a loosely coupled system. This parameter specifies the IDs of all the TPF processors in a loosely coupled complex. The processor ID is a 1-character alphanumeric value from A–Z or 0–9. The value is specified as a list of processor IDs separated by commas, as a range of processor IDs with the first and last processor ID separated by a dash, or as a combination of the two. The following code examples are equivalent:

```
SDPSID=(A,C,D,E,F,H,J)
SDPSID=(A,C-F,H,J)
```

**Notes:**

1. The value coded for the SDPSID parameter in the OSTG EXEC statement PARM field must contain the same processor IDs in the same order as coded in the SDPSID parameter in the ANTDEF statement.
2. OSTG will consider `SDPSID=(A,C-F)`coded in the OSTG EXEC PARM field to be the same as `SDPSID=(A,C,D,E,F)` coded in the ANTDEF statement. Because of a JCL restriction of 100 characters for the PARM field, OSTG allows the use of the dash (–) to indicate a range.

**PILOTID**
Required. Specify the 1-character pilot tape or GDS identification code.

**GENTYPE**
Optional. Specify VAL to perform a validation-only run, which reads and processes all of the input statements. Only an output report is created during a validation-only run. Specify GEN to request a full OSTG run, which creates the output files only if no S-level or E-level errors are found. The default value is GEN.

**SUBAREA**
Required for PU 5 support; otherwise optional. Specify the subarea of the generating processor; that is, the processor specified for the CPUID parameter. This value can be from 1–255.

**Notes:**

1. The optional SUBAREA parameter is used to determine if the TPF system will support PU 5. If this parameter is omitted, the TPF system will only support connections to the SNA network as a PU 2.1 link station.
2. For PU 5 support, the correlation between subarea and processor ID for the generating processor is determined from this information in the EXEC PARM field. The correlation between subarea and processor ID for other TPF processors is determined from the CDRM statements that must be provided to the OSTG program in the RSC deck. See “CDRM Statement” on page 38 for more information.

See Figure 16 on page 58 for an example of the PARM field.

## Input Data Sets

The OSTG input data set (DDNAME=STGINP) contains all of the input definition statements. This data set is read sequentially by the OSTG program and broken logically into 3 decks, which must be processed in the following order:
1. ANT deck
2. RSC deck
3. ALS deck.

The input definition statements can reside in separate data sets, either as members of partitioned data sets (PDS members) or as one or more sequential data sets. In any case, these input definition statements must be provided to the OSTG program as a sequential file. See Figure 16 on page 58 for an example of how to concatenate the various data sets together as one sequential data set.

See the following for more information:
- “ANT Deck” on page 34 for more information about the ANT deck
- “RSC Deck” on page 37 for more information about the RSC deck

- “ALS Deck” on page 45 for more information about the ALS deck
- Figure 21 on page 67,Table 5 on page 68,Table 4 on page 67, and Figure 22 on page 70 for examples of how to code the ANT deck, RSC deck, and ALS deck.

## Output Data Sets

The output of the OSTG program consists of the SNA configuration data on either a pilot tape or a GDS, and various reports.

### SNA Configuration Data

The data records created by the OSTG program are written to a system pilot tape or GDS. The 1-character name (such as PILOT S) associated with these records is assigned by you using the PILOTID parameter in the PARM field of the JCL EXEC statement. This character is used to load the SNA data records to the online TPF system using the ZNOPL LOAD command. For more information about the ZNOPL LOAD command, see *TPF Operations*. The offline program CNVNNR1 is a service routine that uses these files for the conversion process.

### OSTG Reports

The OSTG program produces the following reports, which are all written to the same data set (DDNAME=SYSOUT):
- Input list report
- Update list report
- LEID list report
- Duplicate node name list report
- Subarea report
- PSV report.

See “Sample OSTG Reports” on page 63 for examples of the OSTG reports.

***Input List Report:*** This report lists all the input definition statements and any information or error messages that occurred.

Error messages are listed following the input definition statement where they occurred. Each error message contains a severity code, which can be one of the following:
- S – Severe error.

  An S-level error causes the OSTG program to be canceled. Any additional processing normally performed by the OSTG program will not occur. You must correct all of the S-level errors and then rerun the OSTG program.
- E – Error on statement.

  An E-level error indicates that an error was detected that invalidates any attempt to produce output. The OSTG program attempts to continue processing; however, no output is produced. You must correct all of the E-level errors and then rerun the OSTG program.
- W – Attention message.

  A W-level error indicates that the OSTG program detected a condition that may not give the results you intended. The OSTG program continues to process and attempts to produce output. Review all W-level errors to ensure proper results.
- I – Informational message.

  I-level messages are strictly informational. They advise you of actions or processes that occurred. Review I-level messages to ensure that the OSTG program produced the correct results.

Severity S and E errors are offset from severity W and I messages to help you isolate these messages. See *Messages (System Error and Offline)* and *Messages (Online)* for a complete list of all OSTG error messages.

***Update List Report:*** This report shows all of the resources that were defined using the OSTG program and the value of the relevant parameters for each resource. The resources are ordered on the update list report in the sequence that they will be loaded online. This report may also list additional errors found in the input definition.

***LEID List Report:*** This report is produced if the LEID parameter was specified on any RSC statement. All resources defined with an LEID are listed in LEID sequence and any duplicate LEIDs are flagged.

***Duplicate Node Name List Report:*** This report is produced if any duplicate node names were found. It lists all of the duplicate node names that were found by the OSTG program. A node name is considered a *duplicate node name* if one or more of the following conditions exist:

- The node name is defined more than once for the same network ID.
- The node name is defined for more than one network ID.
- The node name is defined for a specific network ID, and it is also defined as NETID=ANY.

***Subarea Report:*** This report is produced only if PU 5 support was included in the TPF system. It lists the SAT tables in order to provide you with enough data to get a full perspective of the TPF network.

***PSV Report:*** This report lists all of the PSV routines that are defined in the TPF system.

## Intermediate Work Data Sets

The intermediate work data sets consist of the intermediate RRT work files and the SORT work files.

### Intermediate RRT Work Files

SORTIN and SORTOUT are the data sets used to hold the intermediate resource resolution table (RRT) records. These files are used by each of the various OSTG CSECTs to pass information, as well as by the SORT package.

These files must be large enough to handle one RRT entry for each resource defined using the OSTG program. The space allocation depends on the blocking factor used, the number of resources defined, and the type of device used. The logical record length is specified in the DCB macro that defines the data set. (The size of an RRT record is 4 KB.)

See Figure 16 on page 58 for sample definitions of the SORTIN and SORTOUT data sets.

### SORT Work Files

The SYSUT1, SYSUT2, and SYSUT3 data sets are work files required by the SORT program. See the appropriate SORT manual for estimating the required size of these data sets.

See Figure 15 on page 57 for sample definitions of the SYSUT1, SYSUT2, and SYSUT3 data sets.

## Coding the OSTG Input Definition Statements

All of the input definition statements are patterned after the assembler language macro syntax.

Any record beginning with an asterisk (*) in column 1 is considered by the OSTG program to be a comment record. The record will be printed as it is found on the input list report and the text will not be examined by the OSTG program.

The general format of input definition statements is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   OPCODE PARAMETER=value1,      commentary                       X
              <PARAMETER=value2>
```

The symbol parameter is used on some statements and ignored if specified on others. Symbols must start in column 1 of the input definition statement and can be a maximum of 8 alphanumeric characters. The first character must be A–Z. Alphabetic characters must be in uppercase. There must be at least 1 blank character after a symbol.

The OPCODE is the name of the input definitions statement and must be coded as shown in this publication. The OPCODE must be preceded and followed by 1 or more blank characters

The parameter must be coded as shown in this publication. Optional parameters are denoted by brackets (< >). All allowable values are also defined in this publication for each permissible parameter. The last parameter value must be followed by 1 or more blanks.

Allowable parameter values are separated by the vertical bar (|) in this publication. This notation is used to show that you must select one of the specified values.

Each statement can be continued to subsequent records by ending the parameter value with a comma, placing one or more blanks after the comma, and placing a nonblank character in the continuation column (column 72). Any text following the comma and blanks is treated as commentary. Commentary is printed in the input list report but is not examined. The continued record must begin with the parameter in column 16. Columns 1–15 inclusive must contain blanks. If an input definition statement has no parameters, but commentary is desired, code a comma in place of the parameter and one or more blanks before any commentary text.

Comment records (* in column 1) are not allowed in a continued statement. They can only precede or follow completed input definition statements.

## SNA Network IDs and Resource Names

SNA network IDs and resource names can contain from 1–8 characters. They must begin with an uppercase letter (A–Z), @, #, or $. The remaining characters can be uppercase letters (A–Z), numbers (0–9), @, #, or $.

Be aware that some of the older terminals may not support special characters. Therefore, if you include the @, #, or $ in an SNA network ID or resource name, you will not be able to type that network ID or resource name on those terminals. Also, these characters may not display in output messages on some of the older terminals. However, you can use the UCCWTOP user exit to change these special characters to printable characters. For more information about the UCCWTOP user exit, see *TPF System Installation Support Reference*. For more information about output message restrictions, see *TPF Programming Standards*.

## Reserved Names

It is necessary to restrict the use of certain names in the OSTG input definition statements. These reserved names are:
- ACPAFMMR–ACPZFMMR
- ACP0FMMR–ACP9FMMR
- ALSNODES
- CTCNODES
- NEFA–NEFZ
- NEF0–NEF9
- SMPA–SMPZ
- SMP0–SMP9
- TPFA–TPFZ
- TPF0–TPF9
- TPFAFMMR–TPFZFMMR
- TPF0FMMR–TPF9FMMR.

Each definition statement explains which, if any, of the reserved names are allowed.

ALSNODES is a reserved name that is used to activate or deactivate all of the PU 2.1 links for a given processor.

CTCNODES is a reserved name that indicates priming is performed for all inactive CTC links. The priming process starts XID processing over CTC links with an active partner.

## ANT Deck

The ANT deck consists of the following input definition statements:
- ANTDEF
- ANTNME
- ANTEND.

This deck is produced as a PDS member by the SIP process from information provided by the MSGRTA macros.

The ANT deck consists of a series of input definition statements used to describe the TPF applications that are available to the TPF system being generated. Input this deck to the OSTG program exactly as it is produced by SIP.

Changes to this deck after it is produced by SIP can cause results that cannot be predicted in the TPF system. The OSTG program assumes that the sequence in which the applications are defined in this deck is the same as the sequence in which they were defined in SIP, and in the application name table (ANT). Therefore, additions, deletions, and other changes can cause the application index field to be calculated incorrectly.

The name of the ANT deck is automatically generated by SIP. SIP uses the first processor ID coded in the SYSID parameter of the CONFIG macro and appends it to the characters ANT. For example, if the first processor ID coded is A, the name of the ANT deck is ANTA. If the first processor ID coded is B, the ANT deck is named ANTB.

See “Sample ANT Deck” on page 59 for an example of the ANT deck.

Resource resolution table (RRT) entries are created only for local processor IDs in the loosely coupled complex. RRT entries are not generated for remote applications. Code RSC statements for these remote applications in order to generate RRT entries.

The following RRTs are created for each processor in the TPF system:

TPF*c*
: The system services control point (SSCP), where *c* is the processor ID.

NEF*c*
: The network extension facility (NEF) application, where *c* is the processor ID.

TPF*c*FMMR
: The SNA functional management message routing (FMMR), where *c* is the processor ID.

These RRTs are required for every TPF processor that is being generated, and for the remaining processors in the loosely coupled complex. These RRTs are always generated automatically when the processor is first found.

For remote TPF processors, the SSCP, SNA FMMR, and SMP application must be explicitly coded in RSC statements to generate RRT entries.

## ANTDEF Statement

The ANTDEF input definition statement is always the first statement in the ANT deck and begins the definition of the applications in the TPF system. Only one ANTDEF input definition statement is allowed in the ANT deck.

See “Sample ANT Deck” on page 59 for an example of the ANTDEF statement.

The format of the ANTDEF statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   ANTDEF CPUID=c,                                                X
               <SDPSID=(n,...,n)>
```

*Figure 4. ANTDEF Statement*

*symbol*
: The symbol parameter is not required and is ignored if coded.

**CPUID**
: Specifies the 1-character identifier of the TPF processor for which the ANT is being defined, where *c* is a character from A–Z or 0–9. For a loosely coupled complex, this is the first processor in the complex. The value specified for this parameter must match the value specified in the EXEC PARM data.

**SDPSID**
: Specifies the 1-character identifiers of all the TPF processors in the loosely coupled system, where *n* is a character from A–Z or 0–9. The maximum number of TPF processors in a loosely coupled system is eight. The value specified for this parameter must match the value specified in the EXEC PARM data.

  **Note:** If loosely coupled support is not included in the TPF system, you must omit this parameter.

## ANTNME Statement

The ANTNME statements follow the ANTDEF statement. There must be one ANTNME statement for each application defined using the MSGRTA macro in SIP, and an ANTNME statement for any special applications required by the TPF system, such as SMP. MSGRTA macros must be specified for all applications residing in the TPF processor being generated. There can be a maximum of 256 ANTNME statements.

See “Sample ANT Deck” on page 59 for an example of the ANTNME statement.

The format of the ANTNME statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   ANTNME NAME=application_name,                                  X
               SNA=YES|NO|LU62|APPC|LOCP,                               X
               CPUID=c,                                                 X
              <RECVRY=YES|NO,>                                          X
              <APPL=P|(S,nnn),>                                         X
              <SAWARE=YES|NO>
```

*Figure 5. ANTNME Statement*

*symbol*
: The symbol parameter is not required and will be ignored if present.

**NAME**
: Specifies the 4-character alphanumeric name of the TPF application. None of the reserved names can be specified. See “Reserved Names” on page 34 for more information about the reserved names.

**SNA**
: Specifies the form of terminal addressing accepted by the application. In the TPF system, a terminal can be addressed by line, interchange, and terminal (LIT) address or by the resource identifier (RID) of the terminal (LU). The valid options are:

  **NO**
  : The application can accept only LIT-type terminal addresses.

  **YES**
  : The application can accept both LIT- and RID-type terminal addresses.

  **LU62**
  : The application can communicate as a logical unit (LU) type 6.2 and is precluded from communicating as any other LU type.

  **APPC**
  : The application is defined as the default TPF/APPC LU.

  **LOCP**
  : The application is defined as the local TPF/APPN control point (CP) LU.

**CPUID**
: Specifies the 1-character ID of the TPF processor where the application resides. This parameter is required for all applications so that the OSTG program can determine where a TPF application resides.

  If the application resides on only one processor, *c* is a character from A–Z or 0–9. If the application resides on all processors in the loosely coupled complex, *c* is an asterisk (*).

**Note:** Define all applications in the loosely coupled complex as CPUID=*. Defining an application to run only on one specific processor implies that if that processor is not online, the application is not available to the entire network. Processor-unique applications also create difficulties in defining load-balancing algorithms.

**RECVRY**
Specifies if the application is recoverable. YES indicates that it is, and NO indicates that it is not. The default is NO.

**Note:** Message recovery is an optional feature, which is defined in the SNA keypoint (CTK2) using the SNAKEY macro. If the TPF system does not support message recovery, coding YES for this parameter has no effect.

**APPL**
Specifies if the application can act as both a primary LU (PLU) and a secondary LU (SLU), or only as a PLU. The valid options are:

**P** The application can ***only*** be a PLU, which is the default.

**(S,***nnn***)**
The application can act as both a PLU and an SLU, where *nnn* indicates the number of concurrent sessions (threads) that this SLU can have with an LU in a remote processor. The maximum number of sessions allowed is 255.

**SAWARE**
Specifies if session awareness user exit support is available for SLU threads associated with the application. YES indicates that session awareness user exit support is available for SLU threads. NO indicates that session awareness user exit support is not available for SLU threads. The default is NO.

### ANTEND Statement

The ANTEND input definition statement is used to delimit the ANT deck. It is the last statement in the ANT deck and it has no parameters.

See “Sample ANT Deck” on page 59 for an example of the ANTEND statement.

The format of the ANTEND statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   ANTEND
```

*Figure 6. ANTEND Statement*

*symbol*
The symbol parameter is not required and is ignored if coded.

## RSC Deck

The RSC deck must follow the ANT deck. It contains a series of input definition statements that describe the cross-domain resource manager (CDRM) resources and remote logical units (LUs) that can communicate with the TPF system being generated.

You can also use the ZNDYN ADD command to define CDRM resources and dynamic LU support to define remote LU resources. See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about dynamic LU support and defining resources to the TPF system.

The RSC deck contains the following input definition statements:

**CDRM** Defines a CDRM resource.

**RSCDEF** Begins the definition of resources that have a particular network ID.

**RSC** Defines a remote LU resource to the TPF system.

**RSCSET** Temporarily overrides the RSC system defaults.

**RSCEND** Ends the definition of resources that have a particular network ID.

The CDRM statements can be specified only if the TPF system supports a network control program (NCP) that is running in a PU 5 environment.

The network definitions follow the optional CDRM statements. Each network ID is defined by an RSCDEF statement, which is followed by all of the RSC statements for that network ID and ended with an RSCEND statement. One or more RSCSET statements can exist anywhere between the RSCDEF and RSCEND statements. You can define any number of network IDs by coding one or more sets of RSCDEF, RSC, and RSCEND statements.

See “Sample RSC Deck” on page 60 for an example of the RSC deck.

## CDRM Statement

The CDRM definition statement describes a cross-domain resource manager (CDRM) that is allowed to communicate with the TPF CDRM as a PU 5 node. Therefore, this statement is valid only if PU 5 support was included in the TPF system. (That is, the SUBAREA parameter was specified in the EXEC PARM field of the JCL deck.) A CDRM is the portion of an SSCP that controls the cross-domain resource connections.

A CDRM definition statement is also required for each processor in a TPF loosely coupled complex when PU 5 support is included in the TPF system. The CDRM statements for the loosely coupled processors are used to define the unique SNA subarea values for each processor.

You can also use the ZNDYN ADD command to define CDRM resources. See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about defining resources to the TPF system.

See “Sample RSC Deck” on page 60 for an example of the CDRM statement.

The format of the CDRM statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

cdrmname CDRM  SUBAREA=n,                                              X
              <ELEMENT=n,>                                             X
              <CPUID=c,>                                               X
              <REALNAME=name2>
```

*Figure 7. CDRM Statement*

*cdrmname*
Specifies the name of the CDRM. Specify the CDRM name as follows:
- For a TPF CDRM, the name must be TPF*c*, where *c* is the processor ID of the TPF system as specified for the CPUID parameter.
- If you specify a reserved name, only TPFA–TPFZ and TPF0–TPF9 are allowed.
- Do not specify a reserved CDRM name for a non-TPF CDRM. See “Reserved Names” on page 34 for more information about the reserved names.

See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in a CDRM name.

If the CDRM is accessed through an SNI NCP to a VTAM system that also has APPN connections with the TPF system, you must use an alias CDRM name because the VTAM SSCP name and CP name are identical. In this case, *cdrmname* must be an alias name for the CDRM, and the real name of the CDRM must be specified by the REALNAME parameter. See *TPF ACF/SNA Data Communications Reference* for more information about TPF/APPN and SNI considerations.

**SUBAREA**
Specifies the pseudo subarea to be used for PU 5 SNI gateway support. A SAT entry will be created for this subarea. This value is used to group resources in the CDRM domain. It is referred to as the *owning* subarea (RV1OWNSA). The actual network address of the CDRM is determined dynamically online at ACTCDRM time. This parameter is required.

**ELEMENT**
Specifies the element address portion of the CDRM network address. The maximum value of *n* is 32 767. This parameter is ignored for a CDRM accessed through an SNI NCP. (The network address is determined dynamically at ACTCDRM time.)

**CPUID**
Specifies the processor ID of this CDRM if the CDRM is a TPF processor. This parameter must be omitted if the CDRM is not a TPF processor, but is required for all TPF processors defined in the network.

**REALNAME**
Specifies the real name of the CDRM. Code this parameter only when the CDRM is accessed through an SNI NCP to a VTAM system that also has APPN connections with the TPF system. In this case, the real name of the CDRM is specified by the REALNAME parameter while an alias name must be specified for *cdrmname*.

## RSCDEF Statement

The RSCDEF statement identifies the beginning of the remote LU descriptions. It is used to delimit a network ID. A network ID can be thought of in the same way that a domain was thought of in previous TPF releases. The network ID is used to qualify network names in order to resolve any duplication of names among different networks.

See “Sample RSC Deck” on page 60 for an example of the RSCDEF statement.

The format of the RSCDEF statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   RSCDEF NETID=netid|ANY
```

Figure 8. RSCDEF Statement

*symbol*
The symbol parameter is not required and will be ignored if specified

**NETID**
Specifies the network identifier for the LU resources defined between this statement and the associated RSCEND statement. The valid options are:

*netid*
Specifies the network ID.

See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in a network ID.

**ANY**
Creates an RRT entry with a blank network ID. The actual network ID is determined at session initiation time. For PU 5, the network ID should always be coded NETID=ANY.

## RSC Statement

The RSC definition statement is used to describe each remote LU resource. You can also use dynamic LU support to define remote LU resources. See *TPF ACF/SNA Data Communications Reference* for more information about dynamic LU support and defining resources to the TPF system.

See “Sample RSC Deck” on page 60 for an example of the RSC statement. Also see “Valid Combinations for the CCTYPE and LUTYPE Parameters” on page 54 for more information about the valid combinations for the following parameters.

The format of the RSC statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   RSC  <LUTYPE=ANY|3277|3278|3284|                              X
               |3286|3287|3288|3289|360X|3614|APPC|REMCP|             X
               APPLU|APSLU|AX001|AX002|BATCH|FMMR|                    X
               L6PLU|L6SLU|MCHLU|NEF|VCLU|XALCI|FTPI,>                X
              <CCTYPE = 3601|3602|3274|3276|3271,>                    X
              <LUMOD=2|3|4|,>                                         X
              <SCSBUF=2K|4K,>                                         X
              <RECVRY=YES|NO,>                                        X
              <IATA=xxxx,>                                            X
              <THREAD=SINGLE|MULTI,>                                  X
              <UMODE=xx,>                                             X
              <MODE=NETVIEW,>                                         X
              <PSV=name,>                                             X
              <LEID=xx|yyyy|zzzzzz,>                                  X
              <AWARE=YES|NO,>                                         X
              <PRSHR=YES|NO,>                                         X
               CHAIN=YES|NO>
```

Figure 9. RSC Statement

*symbol*
Specifies the name of the LU being defined. This parameter is required on the RSC statement. The LU name must be unique for each network ID.

See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in an LU name.

The following are valid reserved names:

- For LUTYPE=APPLU:
  - NEFA through NEFZ
  - NEF0 through NEF9.
- For LUTYPE=FMMR:
  - TPFAFMMR through TPFZFMMR
  - TPF0FMMR through TPF9FMMR.

See “Reserved Names” on page 34 for more information about the reserved names.

**LUTYPE**
Specifies the device or LU type of the resource being defined. The valid options are:

**ANY** Unspecified at OSTG definition time. This is the default option.

LUTYPE=ANY creates an RRT entry that simply predefines the LU node name. If LUTYPE=ANY is specified, omit all of the other parameters with the exception of the IATA and UMODE parameters. The information for the other parameters is determined and the RVT is updated at session initiation time. See *TPF ACF/SNA Data Communications Reference* for more information.

**360X** IBM 360x devices

**3614** IBM 3614 Consumer Transaction Facility

**3277** IBM 3277 display

**3278** IBM 3278 display

**3284** IBM 3284 printer

**3286** IBM 3286 printer

**3287** IBM 3287 printer

**3288** IBM 3288 printer

**3289** IBM 3289 printer

**APPC** LU 6.2 primary LU (PLU)

**Note:** This option is treated the same as the L6PLU option. It is still available for migration purposes.

**REMCP**
Remote APPN control point (CP) LU

**APPLU**
Primary LU application

**APSLU**
Secondary LU application

**BATCH**
IBM 3600 Financial Controller

**FMMR** Functional management message router

**L6PLU**
LU 6.2 primary LU (PLU)

**L6SLU**
LU 6.2 secondary LU (SLU)

**MCHLU**
NPSI multichannel link

**NEF** The NEF application resident in the 37*x*5

**VCLU** NPSI virtual circuit

**AX001**
AX25 virtual circuit for single terminal controllers

**AX002**
AX25 virtual circuit for multiple terminal controllers

**XALCI** ALC data formats across GATE/FTPI connections

**FTPI** Generic GATE/FTPI resource.

**CCTYPE**
Specifies the cluster (PU) type to which the resource being defined is connected.

The valid options are:

**3601** Finance Communications Controller

**3602** Finance Communications Controller

**3271** IBM 3271 Controller

**3274** IBM 3274 Controller

**3276** IBM 3276 Controller.

**Note:** The LUTYPE and CCTYPE parameters depend on each other. See Table 2 on page 55 and Table 3 on page 56 for information about the valid combinations.

See the *TPF Migration Guide: Program Update Tapes* for a list of all supported controllers.

**LUMOD**
Specifies the model of the 3270-type display device being defined. The valid options are:

**2** Model 2

**3** Model 3

**4** Model 4.

**Note:** See the *TPF Migration Guide: Program Update Tapes* for a list of the display devices supported by the TPF system.

**SCSBUF**
Specifies the buffer size for an IBM 3287 or 3289 printer. The valid options are:

**2K**
Valid only for 3287 printers.

**4K**
Valid only for 3287 or 3289 printers.

**Note:** This parameter is ignored unless LUTYPE=3287 or LUTYPE=3289 is specified.

**RECVRY**
Specifies if the resource is eligible for output message recovery. YES specifies that this resource is eligible for output message recovery. NO specifies that this resource is ***not*** eligible for output message recovery support. The default is NO.

**Notes:**

1. A resource can be eligible for output message recovery only when message recovery is defined for the TPF system. If the TPF system does not support message recovery, coding YES for this parameter has no effect.
2. The RECVRY parameter and the PSV parameter are mutually exclusive.
3. The RECVRY parameter and the THREAD parameter are mutually exclusive.

**IATA**
Specifies the pseudo interchange address/terminal address (IATA) to be used by this resource to communicate with the old type applications that require a line number/interchange address/terminal address (LNIATA). Each IATA specified must be unique. Duplicate IATAs are not allowed by the OSTG program.

**THREAD**
Specifies if the resource being defined supports single (SINGLE) or multiple (MULTI) thread operations.

**Note:** The THREAD parameter and the RECVRY parameter are mutually exclusive.

**UMODE**
Specifies a 1-byte hexadecimal indicator that is stored in RVT mode byte 3. The contents and use of this user mode byte are left solely to your discretion. This information is accessible to user applications using the INQRC macro. See *TPF General Macros* for more information about the INQRC macro.

**MODE**
NETVIEW specifies that this resource is a NetView Terminal Access Facility (TAF) LU type 1 printer. It is valid only when CCTYPE=3274 or CCTYPE=3276 and LUTYPE=3287 or LUTYPE=3289 are specified. As a result, this causes the SNANETV bit in RV1MOD2 to be set. This parameter is ignored for all other values of the CCTYPE and LUTYPE parameters.

**PSV**
Specifies the 1- to 6-character alphanumeric process selection vector (PSV) name. Reserved PSV names for IBM use start with the letter I.

**Notes:**

1. A maximum of 96 user-specified PSV names is allowed in the RSC statement.
2. A maximum of 32 IBM-reserved PSV names is allowed in the RSC statement.
3. If this parameter is omitted, an exit routine will not be called for this LU.
4. The PSV parameter and the RECVRY parameter are mutually exclusive.

**LEID**
Assigns the resource a pseudo line, interchange, and terminal address.

The logical end-point identifier (LEID) is a hexadecimal value, which can be specified in the following formats:

*xx* The first byte (LN) of the LEID associated with this LU. All inbound data messages contain the remaining 2 bytes of LEID information (IATA).

*yyyy*
The first 2 bytes (LN and IA) of the LEID associated with this LU. All inbound data messages contain the remaining byte of LEID information (TA).

*zzzzzz*
The 3-byte LEID associated with this LU. This full 3-byte LEID value can be used for 3270 devices, thereby replacing the IATA parameter.

**Note:** All 3270 devices must have a corresponding WGTA entry. All other RSC statements with the LEID parameter must have a WGTA entry unless they have a PSV routine.

**AWARE**
Specifies if the session awareness user exits are to be driven for this LU. YES specifies that the session awareness user exists are to be driven for this LU. NO specifies that the session awareness user exits are ***not*** to be driven for this LU. NO is the default value.

**Note:** This parameter is valid for all resource types except LU 6.2.

**PRSHR**
Specifies if this is a printer that can be shared. YES specifies that this is a printer that can be shared. NO specifies that this is a printer that cannot be shared. This parameter is valid only for resources that are printers. This parameter is not valid for resources that are defined as LUTYPE=ANY.

**CHAIN**
Specifies if large messages are transmitted in multiple segments for NPSI devices. YES specifies that large messages are transmitted in multiple segments for NPSI devices. NO specifies that large messages are ***not*** transmitted in multiple segments for NPSI devices. NO is the default value.

**Notes:**

1. YES is supported only on X.25 virtual circuits and X.25 multichannel links.
2. NO is supported only for AX.25, FTPI, and XALCI NPSI devices.

## RSCSET Statement

The RSCSET input definition statement allows you to temporarily override the system defaults of the corresponding parameters in the RSC statement. This statement can exist in the RSC deck anywhere between the RSCDEF statement and the RSCEND statement.

The values specified for the RSCSET parameters replace the system default values for all of the following RSC statements until the values are changed by another RSCSET statement or the RSCEND statement is reached.

If a value you specify for one of the RSCSET parameters is not valid, the value is flagged with an error message and the default for that parameter remains unchanged.

No RRT entry is created by this statement.

See “Sample RSC Deck” on page 60 for an example of the RSCSET statement.

The format of the RSCSET statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   RSCSET  <LUTYPE=ANY|3277|3278|3284|                           X
               |3286|3287|3288|3289|360X|3614|APPC|REMCP|              X
               APPLU|APSLU|AX001|AX002|BATCH|FMMR|                     X
               L6PLU|L6SLU|MCHLU|NEF|VCLU|XALCI|FTPI,>                 X
              <CCTYPE = 3601|3602|3274|3276|3271,>                     X
              <LUMOD=2|3|4|,>                                          X
              <SCSBUF=2K|4K,>                                          X
              <RECVRY=YES|NO,>                                         X
              <THREAD=SINGLE|MULTI,>                                   X
              <UMODE=xx,>                                              X
              <PSV=name,>                                              X
              <AWARE=YES|NO,>                                          X
              <PRSHR=YES|NO,>                                          X
               CHAIN=YES|NO>
```

*Figure 10. RSCSET Statement*

*symbol*
The symbol parameter is not required and is ignored if coded.

See “RSC Statement” on page 40 for more a description of the parameters.

### RSCEND Statement

Code the RSCEND input definition statement to mark the end of the RSC deck. This statement is used by the OSTG logic as a delimiter to complete the remote LU definitions for a set of resources residing in a network ID.

This statement also resets all of the default values that may have been overridden with RSCSET statements back to the system default values.

See “Sample RSC Deck” on page 60 for an example of the RSCEND statement.

The format of the RSCEND statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

symbol   RSCEND
```

*Figure 11. RSCEND Statement*

*symbol*
The symbol parameter is not required and is ignored if coded.

## ALS Deck

The ALS deck must follow the RSC deck. It defines the NCP, CTC, and ALS resources that are channel-attached to the TPF system.

You can also use the ZNDYN ADD command to define NCP and CTC resources. If the TPF system is running in Advanced Peer-to-Peer Networking (TPF/APPN) mode, you can use the ZNDYN ADD command and dynamic LU support to define ALS resources as well. See *TPF Operations* for more information about the ZNDYN

ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about TPF/APPN support, dynamic LU support, and defining resources to the TPF system.

The following input definition statements are included in the ALS deck:

**ALS** Defines the 37*x*5 and non-37*x*5 ALS resources that can attach to the TPF system as PU 2.1 link stations.

**NCP** Defines the 37*x*5 NCP resources that can attach to the TPF system as PU 4 nodes.

**CTC** Defines the channel-to-channel stations.

The ALS, NCP, and CTC input definition statements can exist in the ALS deck in any order.

See “Sample ALS Deck” on page 62 for an example of the ALS deck.

## ALS Statement

The ALS statement defines both the 37*x*5 and non-37*x*5 devices that can connect to the TPF system as a PU 2.1 link station.

If the TPF system is running in Advanced Peer-to-Peer Networking (TPF/APPN) mode, you can also use the ZNDYN ADD command and dynamic LU support to define ALS resources. See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about TPF/APPN support, dynamic LU support, and defining resources to the TPF system.

See “Sample ALS Deck” on page 62 for an example of the ALS statement.

The format of the ALS statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

alsname  ALS   <CLU=cluname,>                                         X
               <QN|ALSQN=xx>
```

*Figure 12. ALS Statement*

*alsname*
This parameter is required and specifies the link station name of the ALS. See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in an ALS name.

For 37*x*5 connections, this name is used by the TPF system and VTAM to refer to the 37*x*5 channel adapter. The name must match the name coded on the PU statement in the NCP generation deck.

For non-37*x*5 devices such as a 3174 APPN controller or RISC System/6000 system, code the ALS name using the following format:

**TPF***xnnnn*

Where:

*x* The 1-character ID of the TPF processor.

*nnnn*
The 4-character SDA number of the ALS channel adapter.

**CLU**

This parameter is optional and should be coded only when the resource being defined by this ALS statement is a 37*x*5 NCP. It specifies the 1- to 8-character control LU (CLU) name coded in the NCP generation deck. See "SNA Network IDs and Resource Names" on page 33 for information about the characters that you can specify in a CLU name.

If the TPF system is running in APPN mode, do not code this parameter. If the TPF system is running in LEN mode, the CLU parameter on the ALS statement is required when defining the primary ALS. The CLU parameter is not required when the ALS statement is used for the backup section of an NCP used in a 37*x*5 multiple central control unit (CCU).

When specifying a CLU name, follow the rules for defining a symbol, which are located in "Coding the OSTG Input Definition Statements" on page 33.

None of the reserved names can be specified. See "Reserved Names" on page 34 for more information about the reserved names.

This name is used by the TPF system and VTAM when establishing the CLU-CLU session with the Logon Manager (LM).

It is recommended that the name you use matches the corresponding LU name in the NCP generation deck for this ALS. The TPF system uses this method to give you a way to correlate the CLU name with the ALS and to ensure that the correct number of CLUs are defined. When defining the ALS for a backup CCU of a multi-engine 37*x*5, the CLU name is not required because the LU name from the primary ALS is used. See Figure 23 on page 71 and Figure 24 on page 72 for an example of the NCP channel adapter definition for the PU 2.1 environment for the 3745-410 fallback mode support.

**QN|ALSQN**

This parameter is required only when defining a 37*x*5 NCP and the TPF system is running in LEN mode. If the TPF system is running in APPN mode or you are defining a non-37*x*5 device, do not code this parameter.

This parameter represents the unique qualifier number (QN) for each ALS defined. This value is specified as 2 hexadecimal digits. This qualifier is appended to the application name and processor ID defining TPF applications as independent LUs in the NCP. These TPF alias names are used by the VTAM Logon Manager for load balancing sessions across multiple links to the TPF system. See "ALS and CTC Qualifiers" on page 53 for more information on this parameter.

**Note:** The ALS name is not network qualified. Therefore, all ALS and CLU names must be unique. (You cannot define the same ALS or CLU name for more than one network ID.)

## NCP Statement

The NCP statement defines each 37*x*5 that can be channel attached to a local TPF processor as a PU 4 node.

You can also use the ZNDYN ADD command to define NCP resources. See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about defining resources to the TPF system.

The following information is specified on the NCP input definition statement:
- The name assigned to the network control program (NCP)
- The subarea address assigned to the NCP

- The virtual route pacing window size
- The virtual route pacing request timeout value
- The virtual route pacing request and input list shutdown timeout value.

The NCP statement is valid only if PU 5 support has been included in the TPF system.

See “Sample ALS Deck” on page 62 for an example of the NCP statement.

The format of the NCP statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

ncpname  NCP   SUBAREA=n,WINSIZE=n,<VRTO=n,><VRILTO=n>
```

*Figure 13. NCP Statement*

*ncpname*
: This parameter is required and specifies the name of the network control program (NCP). This name is used by the TPF system to verify that the 37*x*5 contains the correct NCP load image.

See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in an NCP name.

None of the reserved names can be specified. See “Reserved Names” on page 34 for more information about the reserved names.

**SUBAREA**
: This parameter is required and specifies the subarea address of the NCP being defined.

**WINSIZE**
: This parameter is required and specifies the virtual route pacing window size to be set by the TPF system when activating the virtual route between the TPF system and the NCP. The TPF system uses XID format 2 to connect the NCP and uses FID4 PIUs to communicate with the NCP.

You can specify a value in the range 1–255. The recommended value is 42. This value will be the minimum and maximum window size used on the virtual route.

Setting the number too low can result in unnecessary overhead (a virtual route pacing response [VRPRS] is required once every window) and delay in response time (the TPF system will queue messages if a VRPRS is outstanding). Setting the number too high can cause NCP buffer depletion because an NCP reserves three times the minimum number of buffers for the virtual route.

**VRTO**
: This optional parameter specifies the virtual route timeout value when the virtual route of the link is blocked. Specify the number of SNA polling intervals for the virtual route timeout value. The SNA polling interval is defined by the SNAPOLL parameter.

When the virtual route (VR) is blocked for the specified time, the link attempts to issue 1 more PIU with the VR pacing request indicator on. The TPF system may have lost the VR pacing response, or the NCP may have lost the VR pacing request. If the link times out again, the link is discontacted.

The default for this parameter is 0, which indicates that no timeout processing occurs on this link. The maximum value is 65 535 (about 54 minutes).

There are two considerations for setting this value:

- Note how long the NCP is normally blocked. You must set the VRTO pacing timeout value longer than this blockage time.
- Determine how many messages per second are queued if the VR is blocked and relate this to how long the TPF system can stand having these messages queued in core. For example, if you were processing 100 messages per second to this NCP and the VR became blocked, the 100 messages continue to be queued in core. If you decide that the TPF system can handle only 800 messages queued to this NCP and you want the discontact of the NCP to occur in 8 seconds, set the value of VRTO to 80, which represents 4 seconds. Remember that the first time the timer goes to 0, another VR pacing request is sent.

**VRILTO**

This optional parameter specifies the virtual route input list timeout value when the virtual route of this link is blocked and the TPF system is in input list shutdown. Specify the number of SNA polling intervals for the virtual route input list timeout value. The SNA polling interval is defined by the SNAPOLL parameter.

When the virtual route (VR) is blocked for the specified time and the input list is shut down, the link attempts to issue the equivalent of 1 VR pacing window of PIUs (WINSIZE value) each time the timer goes to 0. Because the TPF system is in input list shutdown, the pacing response is not yet received because both of the read buffers are on the input list that is shut down.

An attempt is made to send out the PIUs in these blocks to the NCP in order to relieve the input list shutdown condition. To prevent the NCP from being flooded, set this value equal to the value it normally takes for the NCP to process 1 window of PIUs.

The default value for this parameter is 0, which indicates that no timeout processing occurs on this link. The maximum value is 255 (about 12 seconds).

## CTC Statement

The CTC statement defines each CTC link station that can be attached to the local TPF processor as a PU 5 node.

You can also use the ZNDYN ADD command to define CTC resources. See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about defining resources to the TPF system.

You can specify the following information on the CTC definition statement:

- The name assigned to the CTC link station
- The subarea address accessed by the CTC
- The virtual route (VR) pacing window size
- The name of the control LU (CLU) for each CTC connection to VTAM
- The qualifier number (QN) for each CTC connection to VTAM, which is used when creating TPF alias names for the VTAM Logon Manager.

See “Sample ALS Deck” on page 62 for an example of the CTC statement.

The format of the CTC statement is:

```
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7...

ctcname  CTC   SUBAREA=n,WINSIZE=n,                                     X
               <CLU=cluname|CLU=(cluname1, cluname2,...,cluname8),>     X
               <QN=xx,>                                                 X
               <TG2SDA>,<VRTO=n,><VRILTO=n>,<REMOTE=yes|no>
```

*Figure 14. CTC Statement*

*ctcname*
: This parameter is required and specifies the name of channel-to-channel (CTC) link station. The CTC name is not network qualified. Therefore, all CTC and ALS names must be unique. Duplicate CTC names are not allowed. See the description of duplicate node names in “Duplicate Node Name List Report” on page 32 for more information.

See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in a CTC name.

**SUBAREA**
: This parameter is required and specifies the remote subarea address accessed by the CTC link station that is being defined. Values range from 1–255. You must define a CDRM with the same subarea. You can define no more than 2 CTC link stations with the same subarea.

**WINSIZE**
: This parameter is optional and specifies the virtual route pacing window size to be set by the TPF system when activating the virtual route through the CTC. The value can range from 1–255. The default (and recommended) value is 100. The TPF system requires that the virtual route use pacing windows with equal minimum and maximum values.

**TG2SDA**
: This parameter is no longer supported. If you code this parameter it is ignored.

**CLU**
: This parameter specifies the 1- to 8-character name of the control LU (CLU) for this CTC. See “SNA Network IDs and Resource Names” on page 33 for information about the characters that you can specify in a CLU name.

This parameter is required only for CTC connections to VTAM when APPC sessions are active and the TPF system is running in LEN mode. Do not code this parameter when the TPF system is running in APPN mode.

If this is a loosely coupled complex, multiple CLU names must be specified. There must be a CLU name for each TPF processor in the complex in order for each TPF processor to have its own CLU-Logon Manager session. The CLU names must be defined to VTAM as a CDRSC.

**Note:** If 2 CTC connections are defined between a TPF processor and VTAM, at least 1 CTC connection must have a CLU defined. Both connections should have CLUs defined for load balancing and backup.

**QN**
: This parameter is required for CTC connections to VTAM when the TPF system is running in LEN mode. When the TPF system is running in APPN mode, do not code this parameter. The qualifier number (QN) is 2 hexadecimal characters that uniquely represent each CTC link to a VTAM host.

To create an alias name for a TPF application, the qualifier number is appended to the application name and processor ID. The TPF CLU uses this alias name to inform the Logon Manager of the availability of a TPF application across the link. Therefore, define TPF application alias names as a CDRSC to VTAM. See "ALS and CTC Qualifiers" on page 53 for more information about this parameter.

**VRTO**

This optional parameter specifies the virtual route timeout value when the virtual route of this link is blocked. Specify the number of SNA polling intervals for the virtual route timeout value. The SNA polling interval is defined by the SNAPOLL parameter.

When the virtual route (VR) is blocked for the specified time, the link attempts to issue 1 more PIU with the VR pacing request indicator on. The TPF system may have lost the VR pacing response, or the CTC may have lost the VR pacing request. If the link times out again, the link is discontacted.

The default value for this parameter is 0, which indicates that no timeout processing occurs on this link. The maximum value is 65 535 (about 54 minutes).

There are two considerations for setting this value:

- Note how long the CTC is normally blocked. You must set the VRTO pacing timeout value longer than this blockage time.
- Determine how many messages per second are queued if the VR is blocked and relate this to how long the TPF system can stand having these messages queued in core. For example, if you were processing 100 messages per second to this CTC and the VR became blocked, the 100 messages continue to be queued in core. If you decide that the TPF system can handle only 800 messages queued to this CTC and you want the discontact of the CTC to occur in 8 seconds, set the value of VRTO to 80, which represents 4 seconds. Remember that the first time the timer goes to 0, another VR pacing request is sent.

**VRILTO**

This optional parameter specifies the virtual route input list timeout value when the virtual route of this link is blocked and the TPF system is in input list shutdown. Specify the number of SNA polling intervals for the virtual route input list timeout value. The SNA polling interval is defined by the SNAPOLL parameter.

When the virtual route (VR) is blocked for the specified time and the input list is shut down, the link attempts to issue the equivalent of 1 VR pacing window of PIUs (WINSIZE value) each time the timer goes to 0. Because the TPF system is in input list shutdown, the pacing response is not yet received because both of the read buffers are on the input list that is shut down.

An attempt is made to send out the PIUs in these blocks to the CTC in order to relieve the input list shutdown condition. To prevent the CTC from being flooded, set this value equal to the value it normally takes for the CTC to process a window of PIUs.

The default value for this parameter is 0, which indicates that no timeout processing occurs on this link. The maximum value is 255 (about 12 seconds).

**REMOTE**

This optional parameter specifies the type of hardware used to provide the CTC link. A remote CTC controller connects 2 systems through a telecommunications link. The IBM 3737 is an example of a remote CTC device. If REMOTE=YES is specified, the TPF system will send a DISCONTACT to the remote system following a hard initial program load (IPL). Otherwise, a clear subchannel (CSCH) instruction will be executed to end a previously active link.

## Coding Hints and Considerations

Remember the following when using the OSTG program to define resources to the TPF system.

### Scratch Pad Areas

Only resources defined using the OSTG program can have a scratch pad area (SPA). Resources defined using the ZNDYN ADD command and dynamic LU support do not have a scratch pad area.

You can, however, assign spare SPA ordinals to the dynamic LUs using the dynamic LU user exit. For a dynamic LU, the ordinal saved at RV1ORDN only means the SPA ordinal and cannot be used as an NCB ordinal. You can use the CSNB segment to access the SPA record. However, the SPA fixed file record must be initialized before calling the CSNB segment. Without doing so, the CSNB segment sets an error return code and returns to the calling segment.

**Note:** The TPF system uses the NCB ordinal number to retrieve the scratch pad area (SPA) for a resource.

See *TPF Operations* for more information about the ZNDYN ADD command. See *TPF ACF/SNA Data Communications Reference* for more information about dynamic LU support and defining resources to the TPF system.

### Fixed File Requirements

Enough #RRTRI records must be allocated during SIP to hold the following records and allow for future expansion:

- All of the RRT entries defined using the OSTG program
- One additional record for the PSV name table.

The minimum number of #RRTRI records required can be determined by using the following formula:

```
                 2(# of resources (highest RID from OSTG) + 2(first entry and last entry))
1 (extra entry + ------------------------------------------------------------------------
   for PSVNT)    RR0ENT(size of each # of RRT entries per record from RR0RT DSECT)
```

**Note:** Rounded up, this value provides the minimum number of #RRTRI records required with no real expansion capabilities. You must evaluate the need for future expansion and allocate accordingly.

### System Definition Consistency

When preparing input for the OSTG program, consider the resource definitions that exist in other components of the network, specifically, in the NCP and VTAM system generations. Resource definitions must be compatible across all network components.

### Resource Names

The ZNDLU command can be used to display information about resources that have a common string of characters in their resource name. Therefore, when using the RSC statement to define resources, consider assigning resource names in such a way that resources with similar characteristics share names with a common string of characters.

For example, if you prefix all of the names for New York City Ticket Counter resources with NYC, you can enter **ZNDLU ID-NYC*** to display information about all of these resources.

See *TPF Operations* for more information about the ZNDLU command.

## ALS and CTC Qualifiers

This section is significant only when the TPF system is running in LEN mode. If the TPF system is running in APPN mode, ALS and CTC qualifiers are not used and do not need to be defined. ALS qualifiers are not used for non-37*x*5 connections.

The QN and ALSQN parameters on the ALS input definition statement of the OSTG program define alias names to the VTAM Logon Manager for TPF applications. If a TPF system will connect to multiple 37*x*5 devices as an ALS, the TPF applications must appear unique to each ALS. This is done by defining the TPF applications in the NCP generation in the following format:

```
aaaacqq
```

where:

*aaaa*
: is the TPF application name.

*c*
: is the TPF processor ID.

*qq*
: is the unique qualifier specified on the ALS statement.

The QN parameter on the CTC statement serves the same purpose. TPF applications must appear unique across each CTC connection. This is done by defining the TPF applications to VTAM as CDRSCs under the TPF CTC CDRM in the following format:

```
aaaacqq
```

where:

*aaaa*
: is the TPF application name.

*c*
: is the TPF processor ID.

*qq*
: is the unique qualifier specified on the CTC statement.

## LUTYPE=ANY

The LUTYPE=ANY option on the RSC statement allows you to define the cross-domain resources of the SNA network with little or no knowledge of the real attributes or topology of the network configuration. It also allows you to easily predefine resource names that do not yet exist in the network configuration. This can significantly reduce the number of times that you must rerun the OSTG program and reload the SNA configuration data to the TPF system to reflect changes in the network.

Specifying LUTYPE=ANY for a resource simply defines a resource name to the TPF system. The characteristics or attributes of the resource are determined when the resource requests a session with a TPF application. The RVT entry of the resource that is defined as LUTYPE=ANY is updated using the information in the suggested BIND image, which is part of the CDCINIT (PU 5 support) or SESINIT (PU 2.1 support) request.

When the session of a resource defined as LUTYPE=ANY ends, the updated information in the RVT entry is removed. This allows subsequent sessions for this resource to be started with a different suggested BIND image, which allows the resource to appear as different device types at different times.

**Note:** To use the LUTYPE=ANY option, the TPF system depends on the VTAM mode table entries being defined correctly.

## RSCSET Usage

The system defaults for the various RSC parameter values depend on the values specified for the LUTYPE and CCTYPE parameters. You can use the RSCSET statement to override these system defaults.

Any valid parameter value specified on the RSCSET statement is applied to all subsequent RSC statements until either another RSCSET statement with the same parameter is found in the input data set or until the RSCEND statement is found. Coding the RSCSET statement with a valid parameter value is treated as if the parameter value was explicitly coded on each of the subsequent RSC statements.

In certain cases, the OSTG program imposes values for certain parameters. This occurs whether the parameter value is explicitly coded on the RSC statement or implicitly included by the effects of an RSCSET statement. When a parameter value is imposed, an informational message is included in the input list report.

## Valid Combinations for the CCTYPE and LUTYPE Parameters

Table 2 on page 55 and Table 3 on page 56 shows the valid combinations for the LUTYPE and CCTYPE parameters. The following list is of incompatible combinations, which are flagged as an error.

*N/A* Not applicable. The parameter is ignored if it is coded.

*Yes* Valid parameter for this RSC type.

*N1* THREAD=SINGLE is imposed.

*N2* THREAD=MULTI is imposed.

*N3* RECVRY=NO is imposed.

*N4* SCSBUF=2K is valid only if LUTYPE=3287 is specified.

SCSBUF=4K is valid only if LUTYPE=3287 or LUTYPE=3289 is specified. SCSBUF is ignored unless LUTYPE=3287 or LUTYPE=3289 is specified.

*N5* MODE=NETVIEW is valid only if CCTYPE=3274 or CCTYPE=3276 and LUTYPE=3287 or LUTYPE=3289 are specified.

The MODE parameter is ignored if the value specified for the CCTYPE or LUTYPE parameter is not valid.

*Y1* THREAD=MULTI and RECVRY=YES cannot both be specified. These parameters are mutually exclusive.

*Table 2. SNAKEY Macro – Minimum, Maximum, and Default Values*

| **PARAMETERS** | **CCTYPE** | | | | | | |
|---|---|---|---|---|---|---|---|
| | **3274 3276** | **3274 3276** | **3271** | **3271** | **3601 3602** | **3601 3602** | **3601 3602** |
| | **LUTYPE** | | | | | | |
| | **3277 3278** | **3284 3286 3287 3288 3289** | **3277 3278** | **3284 3286 3287 3288 3289** | **3614** | **360*x*** | **BATCH** |
| RECVRY | YES | N/A | YES | N/A | N3 | Y1 | N3 |
| LUMOD | YES | YES | YES | YES | N/A | N/A | N/A |
| SCSBUF | N/A | N4 | N/A | N/A | N/A | N/A | N/A |
| THREAD | N1 | N1 | N1 | N1 | N2 | Y1 | N1 |
| IATA | YES | YES | YES | YES | N/A | N/A | N/A |
| UMODE | YES | YES | YES | YES | YES | YES | YES |
| MODE | N/A | N5 | N/A | N/A | N/A | N/A | N/A |
| PSV | YES | YES | N/A | N/A | YES | YES | N/A |
| AWARE | YES | YES | YES | YES | YES | YES | YES |
| CHAIN | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| LEID | YES | YES | YES | YES | YES | YES | YES |
| PRSHR | N/A | YES | N/A | YES | N/A | N/A | N/A |

*Table 3. SNAKEY Macro – Minimum, Maximum, and Default Values (Continued)*

| **PARAMETERS** | **CCTYPE** | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** | **N/A** |
| | **LUTYPE** | | | | | | | | | | | | | |
| | **APPL PLU** | **APPL SLU** | **NEF** | **ANY** | **L6PLU** | **L6SLU** | **APPC** | **FMMR** | **MCHLU** | **VCLU** | **AX001** | **AX002** | **XALCI** | **FTPI** |
| RECVRY | YES | YES | N/A | N/A | N3 | N3 | N3 | N/A | N3 | N3 | N3 | N3 | N/A | N/A |
| LUMOD | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| SCSBUF | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| THREAD | N1 | N1 | N/A | N/A | N1 | N1 | N1 | N/A | N2 | N2 | N2 | N2 | N/A | N/A |
| IATA | N/A | N/A | N/A | YES | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| UMODE | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | N/A | N/A |
| MODE | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |
| PSV | N/A | N/A | YES | YES | N/A | N/A | N/A | N/A | YES | YES | YES | YES | YES | YES |
| AWARE | YES | YES | YES | YES | N/A | N/A | N/A | YES | YES | YES | YES | YES | YES | YES |
| CHAIN | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | YES | YES | N/A | N/A | N/A | N/A |
| LEID | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | YES | N/A | N/A |
| PRSHR | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A |

## Sample JCL

The following JCL is provided as an example for link-editing and running the OSTG program. Obviously, you must change this example to reflect your environment, standards, and naming conventions. Figure 15 provides an example of the JCL needed to create the offline OSTG program.

Figure 16 on page 58 provides an example of the JCL needed to generate the SNA

```
//OSTGLKGO JOB
//***********************************************
//*  OSTG LINKAGE                               *
//***********************************************
//OSTGLK   EXEC PGM=IEWL,REGION=512K,
//              PARM='XREF,DCBS,LIST,LET,SIZE=(300K,30K)'
//OBJLIB   DD   DSN=TPF.BASE.RLSE.OB,DISP=SHR
//SYSLMOD  DD   DSN=TPF.BASE.RLSE.LK,DISP=OLD
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,(9,2))
//SYSPRINT DD   SYSOUT=*
//SYSLIN   DD   *
       INCLUDE OBJLIB(OSTGRT40)  The Root   Phase
       INCLUDE OBJLIB(OSTGIP40)  The Input  Phase
       INCLUDE OBJLIB(OSTGUP40)  The Update Phase
       INCLUDE OBJLIB(OSTGOT40)  The Output Phase
       INCLUDE OBJLIB(OSTGP240)  The Error  Phase
       INCLUDE OBJLIB(JULTOACT)  Convert date subroutine
       NAME    OSTG40(R)
/*
```

*Figure 15. Sample JCL to Create the Offline OSTG Program*

pilot tape. To generate a general data set (GDS) instead of a tape, you only need to change the STGPILOT statement. For example:

```
//STGPILOT DD   DSN=OSTGGDS,DISP=(NEW,KEEP),UNIT=SYSDA,
//              SPACE=(TRK,(10,1)),VOL=SER=PILOTB
```

The EXEC PARM field shown specifies all of the valid parameters. Each parameter is shown inside quotes to demonstrate one way of specifying them on a continuation card. If a continuation card is not needed, the parameters and their values do not need to be enclosed separately in quotes.

This example assumes that the ANT deck is read from the SIP-created PDS data set, and that you already created the RSC and ALS decks as sequential data sets.

```
//***********************************************
//*  OSTG EXECUTION                             *
//***********************************************
//OSTGRUN  EXEC PGM=OSTG40,
//* See note 1 on page 65 regarding the PARM input.
//   PARM=('CPUID=B','PILOTID=S','GENTYPE=GEN','SDPSID=(B,C,D)',
//            'SUBAREA=02','DESC=BASE OSTG RUN')
//STEPLIB  DD   DSN=TPF.BASE.RLSE.LK,DISP=SHR
//SYSUDUMP DD   SYSOUT=*
//STGINPT  DD   DSN=TPF40.ANTS.BSS(ANTB),DISP=SHR
//         DD   DSN=TPF40.RSCS.BSS,DISP=SHR
//         DD   DSN=TPF40.ALSS.BSS,DISP=SHR
//SYSOUT   DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STGPRNT  DD   SYSOUT=*
//STGPRN2  DD   SYSOUT=*
//SORTIN   DD   UNIT=SYSDA,SPACE=(TRK,(120),,CONTIG),DSN=&&WKFO,
//              DCB=(RECFM=FB,BLKSIZE=5200)   LRECL=52
//SORTOUT  DD   UNIT=SYSDA,SPACE=(TRK,(120),,CONTIG),DSN=&&WKFI,
//              DCB=(RECFM=FB,BLKSIZE=5200)   LRECL=52
//SORTLIB  DD   DISP=SHR,DSN=SYS1.SORTLIB
//SORTWK01 DD   UNIT=SYSDA,SPACE=(TRK,(120),,CONTIG)
//SORTWK02 DD   UNIT=SYSDA,SPACE=(TRK,(120),,CONTIG)
//SORTWK03 DD   UNIT=SYSDA,SPACE=(TRK,(120),,CONTIG)
//STGPILOT DD   UNIT=(TAPE,,DEFER),DISP=(NEW,PASS),LABEL=(,NL),
//              DSN=OSTGTAPE,VOL=SER=PILOTB
```

*Figure 16. Sample JCL to Generate the SNA Pilot Tape*

## Sample Configuration Definition

The following sample input definition statements define a loosely coupled TPF system consisting of processor B, C, and D. See “Sample OSTG Reports” on page 63 for the OSTG output report based on this configuration.

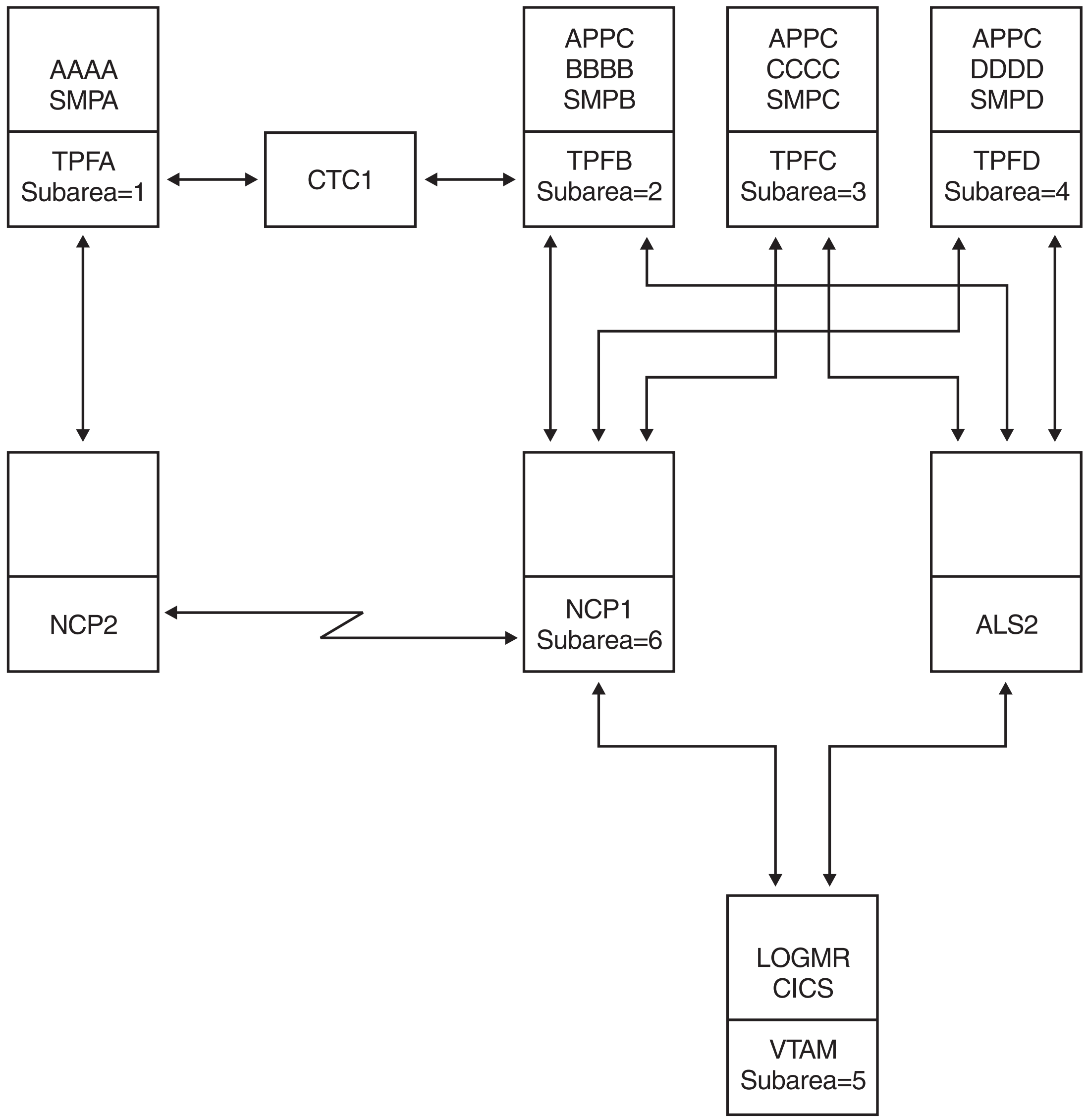



*Figure 17. Sample Configuration Diagram*

## Sample ANT Deck

The following is an example of the ANT deck that would be produced by SIP for the configuration shown in Figure 17.

See “ANT Deck” on page 34 for more information about the ANT deck and the ANTDEF, ANTNME, and ANTEND statements.

```
* See note 2 on page 66 regarding the ANTDEF definition.
          ANTDEF   CPUID=B,SDPSID=(B,C,D)
          ANTNME   NAME=SMPB,CPUID=B,RECVRY=NO,SNA=NO,APPL=P
          ANTNME   NAME=BBBB,CPUID=B,RECVRY=YES,SNA=YES,APPL=(S,002)
          ANTNME   NAME=SMPC,CPUID=C,RECVRY=NO,SNA=NO,APPL=P
          ANTNME   NAME=CCCC,CPUID=C,RECVRY=YES,SNA=YES,APPL=P
          ANTNME   NAME=SMPD,CPUID=D,RECVRY=NO,SNA=NO,APPL=P
          ANTNME   NAME=DDDD,CPUID=D,RECVRY=YES,SNA=YES,APPL=P
          ANTNME   NAME=APPC,CPUID=*,RECVRY=NO,SNA=APPC,APPL=P
          ANTNME   NAME=SMPA,CPUID=A,RECVRY=NO,SNA=NO,APPL=P
          ANTNME   NAME=AAAA,CPUID=A,RECVRY=YES,SNA=YES,APPL=P
          ANTEND
* See note 3 on page 66 regarding CPUID=A.
```

*Figure 18. Sample ANT Deck*

## Sample RSC Deck

The following is an example of the RSC deck for the configuration shown in Figure 17 on page 59.

See “RSC Deck” on page 37 for more information about the RSC deck and the CDRM, RSCDEF, RSC, RSCSET, and RSCEND statements.

```
*********************************************************************
*                                                                   *
*          Define all CDRM resources                                *
*                                                                   *
*********************************************************************
VTAM      CDRM     SUBAREA=5,ELEMENT=1
TPFA      CDRM     SUBAREA=1,ELEMENT=0,CPUID=A
TPFB      CDRM     SUBAREA=2,ELEMENT=0,CPUID=B
TPFC      CDRM     SUBAREA=3,ELEMENT=0,CPUID=C
TPFD      CDRM     SUBAREA=4,ELEMENT=0,CPUID=D
*********************************************************************
*                                                                   *
*          Define all cross-domain resources that may reside        *
*          in any NETID.  The actual NETID will be discovered       *
*          at session-initiation time.                              *
*                                                                   *
* See note 4 on page 66 regarding the RSCDEF definition.            *
* See note 5 on page 66 regarding the VTAM LOGON MANAGER.           *
* See note 11 on page 66 regarding CICS RSC LUTYPE=L6PLU.           *
*********************************************************************
          RSCDEF   NETID=ANY
CICS      RSC      LUTYPE=L6PLU
LOGMR     RSC      LUTYPE=APPLU        VTAM LOGON MANAGER
NCPNEF2   RSC      LUTYPE=NEF
          RSCSET   LUTYPE=3614,CCTYPE=3602,UMODE=F9  CHANGE RSC DEFAULTS
T36A1     RSC
T36A2     RSC      MAXRUSZ=0,RECVRY=YES,UMODE=F8
T36A3     RSC
NEFA      RSC      LUTYPE=APPLU
TPFAFMMR  RSC      LUTYPE=FMMR
SMPA      RSC      LUTYPE=APPLU
AAAA      RSC      LUTYPE=APPLU
          RSCEND
*********************************************************************
*                                                                   *
*          Define all cross-domain resources that reside            *
*          only in specific NETIDs.                                 *
*                                                                   *
*********************************************************************
```

```
*                                                                   *
*                   NETID = NET1                                    *
*                                                                   *
* See notes 7 on page 66 and 8 on page 66 regarding                 *
* the RSCSET definition and NETID.                                  *
*                                                                   *
*********************************************************************
          RSCDEF   NETID=NET1
          RSCSET   LUTYPE=3277,CCTYPE=3274
DEV1      RSC
DEV2      RSC      RECVRY=YES
DEV3      RSC      LUMOD=2,IATA=0002
DEV4      RSC
DEV5      RSC      CCTYPE=3271
DEV6      RSC      LUTYPE=3287
DEV7      RSC      LUTYPE=3289,SCSBUF=4K
          RSCEND
*********************************************************************
*                                                                   *
*                   NETID = NET2                                    *
*                                                                   *
* See note 9 on page 66 regarding duplicate resource names.         *
*                                                                   *
*********************************************************************
          RSCDEF   NETID=NET2
TOM       RSC      LUTYPE=ANY,UMODE=F5
DICK      RSC      LUTYPE=ANY
HARRY     RSC      LUTYPE=ANY
MARY      RSC      LUTYPE=ANY
          RSCEND
*********************************************************************
*                                                                   *
*                   NETID = NET3                                    *
*                                                                   *
*********************************************************************
          RSCDEF   NETID=NET3
TOM       RSC      LUTYPE=ANY
DICK      RSC      LUTYPE=ANY
HARRY     RSC      LUTYPE=ANY
MERRY     RSC      LUTYPE=ANY
          RSCEND
*********************************************************************
*                                                                   *
*                   NETID = NET4                                    *
*                                                                   *
*  Sample of AX.25 and NPSI Devices                                 *
*********************************************************************
          RSCDEF   NETID=NET4
XAX1DEV1  RSC      LUTYPE=AX001,LEID=E1A0
XAX2DEV2  RSC      LUTYPE=AX002,LEID=E1
XMCHLU01  RSC      LUTYPE=MCHLU,LEID=E1C0C0,PSV=MHIN01
XVCLU01   RSC      LUTYPE=VCLU,LEID=E1C0C1,PSV=MHIN02
          RSCEND
*********************************************************************
*                                                                   *
*                   NETID = NET5                                    *
*                                                                   *
*  Sample of LU 6.2 Devices                                         *
*********************************************************************
          RSCDEF   NETID=NET5
T62SLU01  RSC      LUTYPE=L6SLU
T62SLU02  RSC      LUTYPE=L6SLU
          RSCEND
***********************************************************************
*  FTPI Pseudo links, also known as CTCP pseudo LUs, that use the    *
*  XALCI Protocol.  COBU must have an entry for PSV name XALCI that  *
*  refers to segment CNEA.                                           *
```

```
********************************************************************
         RSCDEF   NETID=ANY
M30CPU0  RSC   LUTYPE=XALCI,AWARE=NO,PSV=XALCI
M30CPU1  RSC   LUTYPE=XALCI,AWARE=NO,PSV=XALCI
********************************************************************
*  FTPI Pseudo links are also known as CTCP pseudo LUs. A user CTCP  *
*  is required to handle call setup and takedowns, as well as       *
*  managing relationships of DTE addresses to  RIDs/LEIDs. COBU must *
*  have an entry for PSV name FTPIUSR that refers to this user CTCP. *
********************************************************************
M30CPU2  RSC   LUTYPE=FTPI,AWARE=NO,PSV=FTPUSR
M34CPU0  RSC   LUTYPE=FTPI,AWARE=NO,PSV=FTPUSR
********************************************************************
*  MCH_LU     Type Devices -  3705 Box 94 (01A) to 3745 Box 93  CCUB *
*             Virtual Circuits subordinate to T3452MCH support RH    *
*             Chaining.                                              *
********************************************************************
T3451MCH RSC   LUTYPE=MCHLU,AWARE=YES,LEID=FF06
T3452MCH RSC   LUTYPE=MCHLU,AWARE=YES,LEID=FF07,CHAIN=YES
********************************************************************
*  VCLU       Type Devices -  3705 Box 94 (01A) to 3745 Box 93  CCUB *
*             Virtual Circuits subordinate to T3452MCH support RH    *
*             Chaining.                                              *
********************************************************************
T3452X01 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=80,CHAIN=YES
T3452X02 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=81,CHAIN=YES
T3452X03 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=82,CHAIN=YES
T3452X04 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=83,CHAIN=YES
T3452X05 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=84,CHAIN=YES
T3452X06 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=85,CHAIN=YES
T3452X07 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=86,CHAIN=YES
T3452X08 RSC   LUTYPE=VCLU,AWARE=NO,                                   X
               LEID=87,CHAIN=YES
          RSCEND
```

## Sample ALS Deck

The following is an example of the ALS deck for the configuration shown in Figure 17 on page 59.

See “ALS Deck” on page 45 for more information about the ALS deck and the ALS, NCP, and CTC statements.

```
*
*****************************************************************
*                                                               *
*          Define all channel-attached 37x5s/3174s.             *
*                                                               *
* See note 10 on page 66 regarding 37x5s/3174s.                 *
* See note 6 on page 66 regarding NEF support in the NCP.       *
*                                                               *
*****************************************************************
ALS2B     ALS      CLU=CLU2B,ALSQN=2B
ALS2C     ALS      CLU=CLU2C,ALSQN=2C
ALS2D     ALS      CLU=CLU2D,ALSQN=2D
NCP1      NCP      SUBAREA=6,WINSIZE=42
CTC1      CTC      SUBAREA=1,WINSIZE=100,VRTO=200,VRILTO=20
```

*Figure 19. Sample ALS Deck*

## Sample OSTG Reports

The following sample OSTG output reports are produced by using the sample ANT deck described in “Sample ANT Deck” on page 59, the sample RSC deck described in “Sample RSC Deck” on page 60, and the sample ALS deck described in “Sample ALS Deck” on page 62. It contains examples of the following output reports:

- Update list report
- LEID list report
- Subarea report
- PSV report.

For more information about the output reports, see “OSTG Reports” on page 31.

```
         THIS REPORT SHOWS THE RRTS IN THE ORDER WHICH THEY ARE WRITTEN TO THE OUTPUT DATA SET.
  NETID    NAME     ELEM  MODE    TYPE WNSZ NASA OWSA CPU   SDA  APIX BIND QN LEID    PSV    VRTO VRILTO REAL     SAW  PG   10

                    0000  00000000 0000      00   00        0000 00   00
           TPFA     0000  00000000 8100      01   01   A    0000 00   00
           TPFB     0000  00000000 8100      02   02   B    0000 00   00
           TPFC     0000  00000000 8100      03   03   C    0000 00   00
           TPFD     0000  00000000 8100      04   04   D    0000 00   00
           VTAM     0001  00000000 8100      05   05        0000 00   00
           ALS2B    0000  00000000 8200      00   00        0000 00   00      2B
           ALS2C    0000  00000000 8200      00   00        0000 00   00      2C
           ALS2D    0000  00000000 8200      00   00        0000 00   00      2D
           CTC1     0000  00080000 8400 64   01   01        0000 00   00                     00C8   14
           NCP1     0000  00080000 8000 2A   06   06        0000 00   00                     0000   00
           AAAA     0000  1800F900 1600      01   01   A    0000 00   06
           APPC     0000  10700080 1600      00   00   *    0000 07   14
           BBBB     0000  90000000 1602      02   02   B    0000 02   07
           BBBBB001 0000  88000000 1601      02   02   B    0000 02   07                                           Y
           BBBBB002 0000  88000000 1601      02   02   B    0000 02   07                                           Y
           CCCC     0000  90000000 1600      03   03   C    0000 04   07
           CICS     0000  18700000 1600      00   00        0000 00   14
           CLU2B    0000  04000000 1800      00   00        0000 00   01
           CLU2C    0000  04000000 1800      00   00        0000 00   01
           CLU2D    0000  04000000 1800      00   00        0000 00   01
           DDDD     0000  90000000 1600      04   04   D    0000 06   07
NET1       DEV1     0000  00000000 1511      00   00        0000 00   05
NET1       DEV2     0000  80000000 1511      00   00        0000 00   05
NET1       DEV3     0000  00000000 1511      00   00        0000 00   05              FE0002
NET1       DEV4     0000  00000000 1511      00   00        0000 00   05
NET1       DEV5     0000  00000000 1411      00   00        0000 00   04
NET1       DEV6     0000  00000000 15C1      00   00        0000 00   05
NET1       DEV7     0000  00000000 15C9      00   00        0000 00   05
NET2       DICK     0000  40000000 1000      00   00        0000 00   00
NET3       DICK     0000  40000000 1000      00   00        0000 00   00
NET2       HARRY    0000  40000000 1000      00   00        0000 00   00
NET3       HARRY    0000  40000000 1000      00   00        0000 00   00
           LOGMR    0000  18000000 1600      00   00        0000 00   06
NET2       MARY     0000  40000000 1000      00   00        0000 00   00
NET3       MERRY    0000  40000000 1000      00   00        0000 00   00
           M30CPU0  0000  28000000 1A18      00   00        0000 00   12              XALCI
           M30CPU1  0000  28000000 1A18      00   00        0000 00   12              XALCI
           M30CPU2  0000  28000000 1A10      00   00        0000 00   12              FTPUSR
           M34CPU0  0000  28000000 1A10      00   00        0000 00   12              FTPUSR
           NCPNEF2  0000  28000000 1900      00   00        0000 00   09
           NEFA     0000  1800F900 1600      00   00        0000 00   06
           NEFB     0000  10000000 1600      02   02   B    0000 00   09
           NEFC     0000  10000000 1600      03   03   C    0000 00   09
           NEFD     0000  10000000 1600      04   04   D    0000 00   09
           SMPA     0000  1800F900 1600      01   01   A    0000 00   06
           SMPB     0000  10000000 1600      02   02   B    0000 01   06
           SMPC     0000  10000000 1600      03   03   C    0000 03   06
           SMPD     0000  10000000 1600      04   04   D    0000 05   06
NET2       TOM      0000  4000F500 1000      00   00        0000 00   00
NET3       TOM      0000  40000000 1000      00   00        0000 00   00
           TPFAFMMR 0000  0800F900 1700      00   00        0000 00   13
           TPFBFMMR 0000  00000000 1700      02   02   B    0000 00   13
           TPFCFMMR 0000  00000000 1700      03   03   C    0000 00   13
           TPFDFMMR 0000  00000000 1700      04   04   D    0000 00   13
           T3451MCH 0000  2A000000 1A80      00   00        0000 00   15      FF0600
           T3452MCH 0000  2A010000 1A80      00   00        0000 00   15      FF0700
           T3452X01 0000  28010000 1A40      00   00        0000 00   15      800000
           T3452X02 0000  28010000 1A40      00   00        0000 00   15      810000
           T3452X03 0000  28010000 1A40      00   00        0000 00   15      820000
```

*Figure 20. Sample OSTG Report (Part 1 of 2)*

```
   NETID    NAME     ELEM  MODE    TYPE WNSZ NASA OWSA CPU  SDA  APIX BIND QN LEID    PSV    VRTO VRILTO REAL     SAW  PG   10
            T3452X04 0000  28010000 1A40      00   00       0000  00   15     830000
            T3452X05 0000  28010000 1A40      00   00       0000  00   15     840000
            T3452X06 0000  28010000 1A40      00   00       0000  00   15     850000
            T3452X07 0000  28010000 1A40      00   00       0000  00   15     860000
            T3452X08 0000  28010000 1A40      00   00       0000  00   15     870000
            T36A1    0000  2800F900 1200      00   00       0000  00   02
            T36A2    0000  2800F800 1200      00   00       0000  00   02
            T36A3    0000  2800F900 1200      00   00       0000  00   02
 NET5       T62SLU01 0000  00600000 1600      00   00       0000  00   14
 NET5       T62SLU02 0000  00600000 1600      00   00       0000  00   14
 NET4       XAX1DEV1 0000  20000000 1A28      00   00       0000  00   15     E1A000
 NET4       XAX2DEV2 0000  20000000 1A24      00   00       0000  00   15     E10000
 NET4       XMCHLU01 0000  20000000 1A80      00   00       0000  00   15     E1C0C0  MHIN01
 NET4       XVCLU01  0000  20000000 1A40      00   00       0000  00   15     E1C0C1  MHIN02

          THE NUMBER OF RRT ENTRIES GENERATED: 000000074


         THIS REPORT SHOWS THE RRT ENTRIES IN  LEID SEQUENCE - DUPLICATE ENTRIES ARE NOTED                              PAGE   1

          NODE NAME      LEID      LENGTH

           T3452X01      800000      1
           T3452X02      810000      1
           T3452X03      820000      1
           T3452X04      830000      1
           T3452X05      840000      1
           T3452X06      850000      1
           T3452X07      860000      1
           T3452X08      870000      1
           XAX2DEV2      E10000      1
           XAX1DEV1      E1A000      2
           XMCHLU01      E1C0C0      3
           XVCLU01       E1C0C1      3
           DEV3          FE0002      3
           T3451MCH      FF0600      2
           T3452MCH      FF0700      2


         SAT DATA FOR GENERATING HOST SUB AREA                   PAGE 0001
               SUB AREAS NOT SHOWN IN THIS REPORT ARE UNUSED IN THIS NETWORK GENERATION

          SUBAREA     NAME        CPUID    RCPU

             01       TPFA        A
             02       TPFB        B       B
             03       TPFC        C       B
             04       TPFD        D       B
             05       VTAM
             06       NCP1


         THIS REPORT SHOWS THE PSV    ROUTINE NAME IN SEQUENCE                                PAGE   1

           FTPUSR
           MHIN01
           MHIN02
           XALCI

          *I901* NO STATEMENTS FLAGGED
          *I903* OSTG RUN COMPLETED
```

*Figure 20. Sample OSTG Report (Part 2 of 2)*

## Notes for the Sample Configuration Definition

1. The OSTG input is for a loosely coupled complex that consists of processors B, C, and D. This TPF system is connected to a TPF system that consists of processor A, but processor A is ***not*** part of the loosely coupled complex.

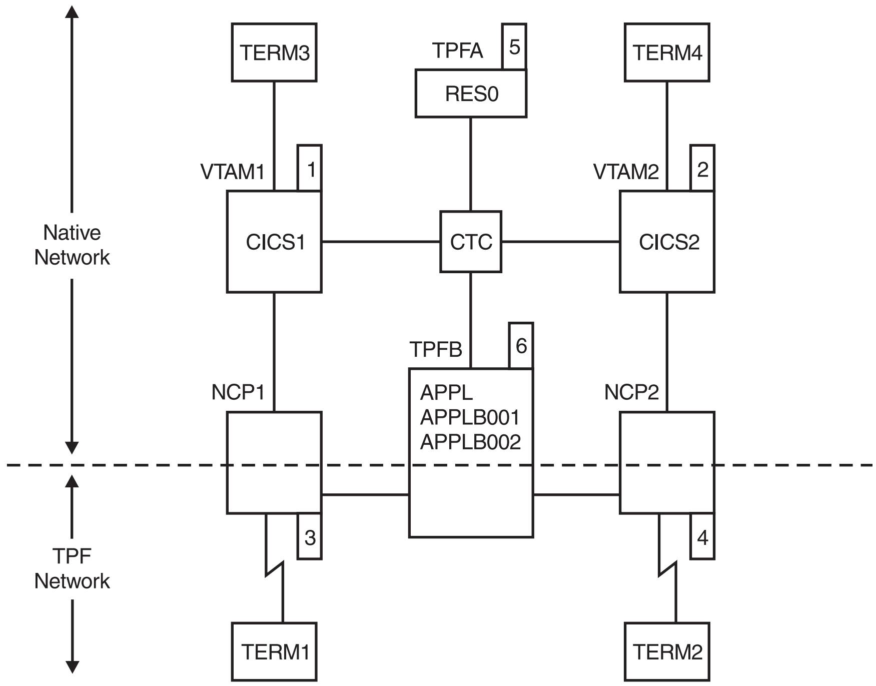
2. The parameters on the ANTDEF statement must be the same as the information provided on the JCL EXEC PARM statements. This is used to cross check that the ANT deck input to the OSTG program is correct.
3. The applications residing in processor A (SMPA and AAAA) are defined to this TPF system
4. The first RSCDEF statement defines resources that can reside in any network.
5. The VTAM Logon Manager is defined as a remote primary application LU.
6. The NEF application LU residing in the NCP is defined to the TPF system.
7. The RSCSET statement is used to override the system defaults for the RSC statements that follow it. Parameters specified on the individual RSC statements override any defaults set with the RSCSET statement (see T36A2).
8. Specific network IDs are defined with the resources that reside in that network.
9. Duplicate resource names can exist (see TOM, DICK, and HARRY) as long as they are uniquely qualified with a different network ID. The TOM resource in NET2 is ***not*** the same resource as TOM in NET3. If the TOM resource had been defined in as NETID=ANY, this definition in a named NETID would cause a duplicate name error. See the description of duplicate node names in “Duplicate Node Name List Report” on page 32 for more information.
10. The 37*x*5 and 3174 resources that are channel attached to the TPF system are defined in the ALS deck. Notice that the 37*x*5 and 3174 resources that are ***not*** channel attached to the TPF system are not defined and, therefore, are not known to the TPF system.
11. Defining a remote CICS system as LUTYPE=L6PLU allows both LU 0 and LU 62 sessions to be established between the TPF system and CICS.

## OSTG Sample Configurations

The following information contains additional sample configurations and input definition statements.

## Sample Configuration for SNA CTC

The CTC box in Figure 21 represents channel-to-channel support; for example, a 3088 communication unit or an Enterprise Systems Connection (ESCON) channel operating in CTC mode.




```
         ANTDEF  CPUID=B
         ANTNME  NAME=APPL,CPUID=B,RECVRY=YES,SNA=YES,APPL=(S,2)
         ANTNME  NAME=RES0,CPUID=A
         ANTEND
**
****   Local and remote CDRM resources
**
TPFA     CDRM     SUBAREA=5,ELEMENT=0,CPUID=A
TPFB     CDRM     SUBAREA=6,ELEMENT=0,CPUID=B[1]
VTAM1    CDRM     SUBAREA=1,ELEMENT=1[2]
VTAM1SNI CDRM     SUBAREA=91[3]
VTAM2    CDRM     SUBAREA=2,ELEMENT=1[2]
VTAM2SNI CDRM     SUBAREA=92[3]
         RSCDEF   NETID=ANY
CICS1    RSC      LUTYPE=APPLU
CICS2    RSC      LUTYPE=APPLU
TERM1    RSC      LUTYPE=ANY
TERM2    RSC      LUTYPE=ANY
TERM3    RSC      LUTYPE=3278[4]
TERM4    RSC      LUTYPE=3278[4]
         RSCEND
TPFACTC  CTC      WINSIZE=96,SUBAREA=5[5]
VTAM1CTC CTC      WINSIZE=64,SUBAREA=1[5]
VTAM2CTC CTC      WINSIZE=64,SUBAREA=2[5]
NCP03    NCP      WINSIZE=42,SUBAREA=3,VRTO=140,VRILTO=10
NCP04    NCP      WINSIZE=42,SUBAREA=4,VRTO=140,VRILTO=10
```

*Figure 21. OSTG Example 1 with SNA CTC*

*Table 4. SUBAREA ADDRESS TABLE*

| Subarea | Name | RID |
|---|---|---|
| 1 | VTAM1 | 4 |
| 2 | VTAM2 | 6 |

1. This CDRM statement assigns the subarea of the local TPF processor. This subarea is the same for both the native VTAM network and for the TPF network.

*Table 4. SUBAREA ADDRESS TABLE (continued)*

| **Subarea** | **Name** | **RID** |
|---|---|---|
| 3 | NCP03 | 7 |
| 4 | NCP04 | 8 |
| 5 | TPFA | 1 |
| 6 | TPFB | 2 |
| 91 | VTAM1SNI | 3 |
| 92 | VTAM2SNI | 5 |

*Table 5. RESOURCE VECTOR TABLE*

| **RID** | **Name** | **Network Address Subarea** | **Element** | **Owning Subarea** |
|---|---|---|---|---|
| 01 | TPFA | (5) | (0) | (5) |
| 02 | TPFB | (6) | 0 | (6) |
| 03 | VTAM1SNI | 3 | 1 | (91) |
| 04 | VTAM1 | (1) | (1) | (1) |
| 05 | VTAM2SNI | 4 | 1 | (92) |
| 06 | VTAM2 | (2) | (1) | (2) |
| 07 | NCP03 | (3) | 0 | 0 |
| 08 | NCP04 | (4) | 0 | 0 |
| 09 | TPFACTC | (5) | 0 | (5) |
| 0A | VTAM1CTC | (1) | 0 | (1) |
| 0B | VTAM2CTC | (2) | 0 | (2) |
| 0C | APPL | (6) | 4 | (6) |
| 0D | APPLB001 | (6) | 5 | (6) |
| 0E | APPLB002 | (6) | 5 | (6) |
| 0F | CICS1 | 1 | 2 | 1 |
| 10 | CICS2 | 2 | 2 | 2 |
| 11 | RES0 | 5 | 2 | 5 |
| 12 | TERM1 | 3 | 2 | 91 |
| 13 | TERM2 | 4 | 2 | 92 |
| 14 | TERM3 | 1 | 3 | 1 |

2. This OSTG statement represents the CTC path to VTAM. From the point of view of the TPF system, each VTAM host appears as two nodes: one node attached by a CTC and another node attached by an SNI NCP. The subarea specified on the CTC statement is for the subarea of VTAM in its network.
3. The subarea of a VTAM host reached through an SNI NCP is used to group the resources in that domain. The subarea value is not used as the VTAM domain's network address.
4. Local SNA 3270s channel-attached to a VTAM system adjacent to the TPF system are supported across a SNA CTC with the constraint that the 3174 and VTAM buffer sizes are defined to ensure that only RH chaining is used. This is because the TPF system does not support both TH and RH chaining on an LU session.
5. The subarea specified on the CTC statement allows the TPF system to dynamically discover the name of the CTC link station when the remote processor initiates the XID procedure. That is, the XID data contains the subarea of the sender and this value is used to locate the corresponding CTC link station definition.

*Table 5. RESOURCE VECTOR TABLE (continued)*

| RID | Name | Network Address Subarea | Element | Owning Subarea |
|---|---|---|---|---|
| 15 | TERM4 | 2 | 3 | 2 |
| **Note:** Numbers in ( ) are filled in offline. | | | | |

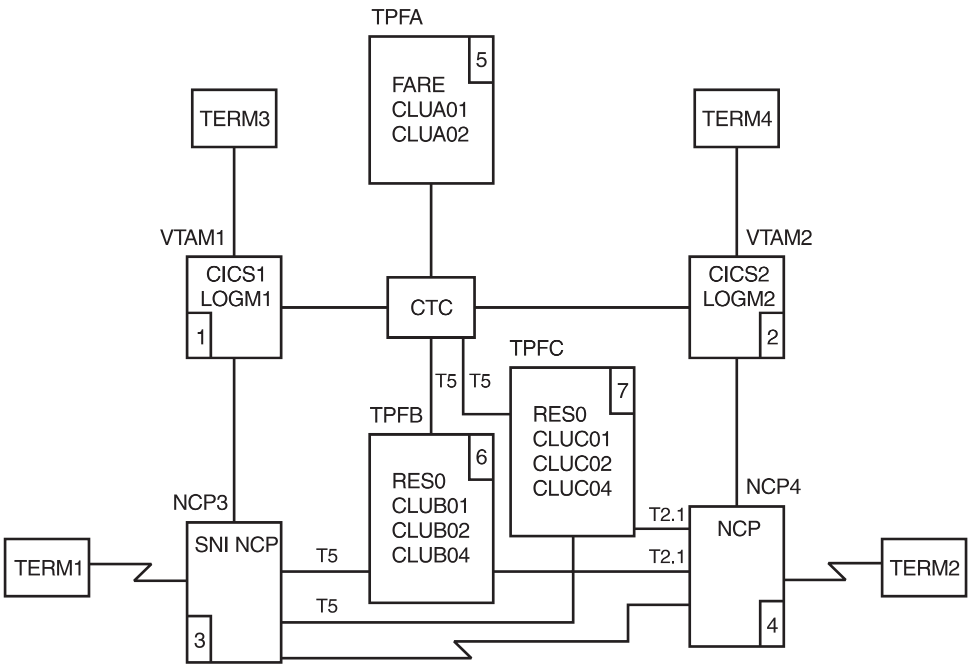
## Sample Configuration for SNA CTC and PU 2.1 for a Loosely Coupled Complex

Figure 22 shows a TPF loosely coupled complex attached to the subarea network in three ways:

- As a PU 5 node in the VTAM network (the CTC connection)
- As a cross-network PU 5 node in another network (the SNI NCP connection)
- As a PU 2.1 node in the VTAM network (the ALS connection).

Consequently, the VTAM definition statements must define three TPF images. Both the same network and cross-network definitions require CDRM and CDRSC statements to define the TPF processor and its resources. The PU 2.1 image is derived by VTAM from the NCP generation deck.

The CTC box in Figure 22 on page 70 represents channel-to-channel support; for example, a 3088 communication unit or an Enterprise Systems Connection (ESCON) channel operating in CTC mode.




```
         ANTDEF   CPUID=A,
         ANTNME   NAME=FARE,CPUID=*,SNA=APPC
         ANTNME   NAME=RES0,CPUID=B,SNA=LU62
         ANTEND
*
*      Local and remote CDRM resources
*
TPFA     CDRM     SUBAREA=5,ELEMENT=1,CPUID=A[6]
TPFB     CDRM     SUBAREA=6,ELEMENT=1,CPUID=B
TPFC     CDRM     SUBAREA=7,ELEMENT=1,CPUID=C
VTAM1C   CDRM     SUBAREA=1,ELEMENT=1
VTAM2C   CDRM     SUBAREA=2,ELEMENT=1
         RSCDEF   NETID=ANY
LOGM1    RSC      LUTYPE=APPLU
LOGM2    RSC      LUTYPE=APPLU
CICS1    RSC      LUTYPE=APPLU
CICS2    RSC      LUTYPE=APPLU
RES0     RSC      LUTYPE=L6PLU
TERM3    RSC      LUTYPE=3278[4]
TERM4    RSC      LUTYPE=3278[4]
         RSCEND
TPFBCTC  CTC      WINSIZE=96,SUBAREA=6
TPFCCTC  CTC      WINSIZE=96,SUBAREA=7
VTAM1CTC CTC      WINSIZE=64,SUBAREA=1,
                  CLU=CLUA01,QN=01
VTAM2CTC CTC      WINSIZE=64,SUBAREA=2,
                  CLU=CLUA02,QN=02
```

```
         ANTDEF   CPUID=B,SDPSID=(B,C)
         ANTNME   NAME=FARE,CPUID=A,SNA=LU62
         ANTNME   NAME=RES0,CPUID=*,SNA=APPC
         ANTEND
*
*      Local and remote CDRM resources
*
TPFA     CDRM     SUBAREA=5,ELEMENT=1,CPUID=A
TPFB     CDRM     SUBAREA=6,ELEMENT=1,CPUID=B[6]
TPFC     CDRM     SUBAREA=7,ELEMENT=1,CPUID=C[6]
VTAM1C   CDRM     SUBAREA=1,ELEMENT=1
VTAM1    CDRM     SUBAREA=91
VTAM2C   CDRM     SUBAREA=2,ELEMENT=1
VTAM2    CDRM     SUBAREA=92
         RSCDEF   NETID=ANY
LOGM1    RSC      LUTYPE=APPLU
LOGM2    RSC      LUTYPE=APPLU
CICS1    RSC      LUTYPE=APPLU
CICS2    RSC      LUTYPE=APPLU
FARE     RSC      LUTYPE=L6PLU
TERM1    RSC      LUTYPE=ANY
TERM2    RSC      LUTYPE=ANY
TERM3    RSC      LUTYPE=3278[4]
TERM4    RSC      LUTYPE=3278[4]
         RSCEND
TPFACTC  CTC      WINSIZE=96,SUBAREA=5
VTAM1CTC CTC      WINSIZE=64,SUBAREA=1,
                  CLU=(CLUB01,CLUC01),QN=01[7]
VTAM2CTC CTC      WINSIZE=64,SUBAREA=2,
                  CLU=(CLUB02,CLUC02),QN=02[7]
ALSB04   ALS      CLU=CLUB04,QN=04[7]
ALSC04   ALS      CLU=CLUC04,QN=04[7]
NCP3     NCP      SUBAREA=3
```

*Figure 22. OSTG Example 2 with SNA CTC and PU 2.1 for a TPF Loosely Coupled Complex*

# Sample Configuration for a 3745 Resource, Model 410

Figure 23 shows 2 central control units (CCUA and CCUB).

---

6. These CDRM statements assign the subarea of the local TPF processors. These subareas are the same for both the native VTAM network and for the TPF network.
7. These statements define the TPF CLUs for the Logon Manager. The LU names are arbitrary. In this example CLUB01 and CLUC01 are for the CTC connections to Logon Manager LOGM1, CLUB02 and CLUC02 are for the CTC connections to LOGM2, and CLUB04 and CLUC04 are for the ALS connections to LOGM2.

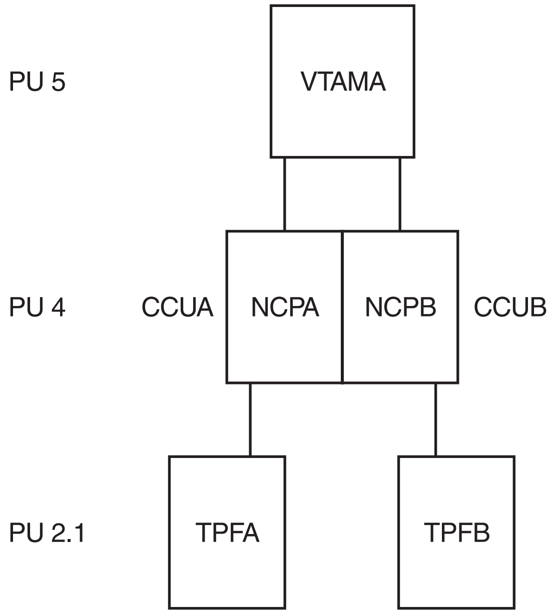



*Figure 23. Sample Configuration for a 3745 Resource, Model 410*

## Sample ALS Deck

The following is a sample ALS deck for the configuration in Figure 23.

```
ALSA      ALS      CLU=CLUAA001,ALSQN=20
XALSA     ALS      ALSQN=20
```

**Notes:**

1. ALSA and XALSA are the PU names coded in the NCP generation for the NCPA and NCPB central control units (CCUs) respectively.
2. CLUAA001 is the LU name coded in the NCP generation under both ALSA and XALSA.
3. CLUAA001 is not required on XALSA for CCUB, which is the backup CCU. With 3745 running in fallback mode, CLUAA001 can be switched to XALSA without changing the LU name.

## Sample NCP Generations for NCPA and NCPB

Figure 24 on page 72 shows an NCP generation for 3745-410 channel adapter definitions that are required for fallback mode.

```
*    NCPA                             |  *    NCPB
                                      |
PCCUA    PCCU  OWNER=VTAMA,           |  PCCUA    PCCU  OWNER=VTAMA,
               BACKUP=YES             |                 BACKUP=YES
NCPA     BUILD SUBAREA=1              |  NCPB     BUILD SUBAREA=2
         .                            |           .
         .                            |           .
*                                     |  *
* CHANNEL ADAPTERS                    |  * CHANNEL ADAPTERS
* ON CCUA OWNED BY                    |  * ON CCUB OWNED BY
* VTAMA                               |  * VTAMA
*                                     |  *
GROUPA   GROUP LNCTL=CA,              |  GROUPB   GROUP LNCTL=CA,
               CA=TYPE6               |                 CA=TYPE6
*                                     |  *
* CHANNEL ADAPTER                     |  * CHANNEL ADAPTER
* DEFINITIONS FOR                     |  * DEFINITIONS FOR
* PU2.1                               |  * PU2.1
* NORMAL MODE                         |  * BACKUP MODE
*                                     |  *
CALINE8  LINE  ADDRESS=(8),           |  CALINE8  LINE  ADDRESS=(8),
               OWNER=VTAMA            |                 OWNER=DUMMY
ALSA     PU    PUTYPE=2,              |  XALSA    PU    PUTYPE=2,
               ANS=CONT               |                 ANS=CONT
CLUAA001 LU    LOCADDR=0              |  CLUAA001 LU    LOCADDR=0
```

*Figure 24. Sample NCP Generation for NCPA and NCPB*

**Note:** When fallback is performed from CCUA to CCUB, VTAMA must acquire and activate CALINE8, and then activate XALSA and CLUAA001.

## Sample NCP Generations for a 3746 Channel Adapter

Figure 25 on page 73 shows an NCP generation for 3746 channel adapter definitions using logical addressing, which allows multiple logical addresses for the same physical link.

```
*    NCPA

PCCUA    PCCU  OWNER=VTAMA,
               BACKUP=YES
NCPA     BUILD SUBAREA=1
         .
         .
*
* CHANNEL ADAPTERS
* DEFINITIONS FOR ESCON
* CONNECTIONS
*
* FIRST DEFINE PHYSICAL
* CONNECTION
*
GROUPA   GROUP LNCTL=CA,
CALINE8  LINE  ADDRESS=(8),
               OWNER=VTAMA
PHYSPUA  PU    PUTYPE=1,
*
* DEFINE LOGICAL CONNECTIONS
* FOR PU2.1 NORMAL MODE
*
* PHYSRC SHOWS ACTUAL PHYSICAL
* CONNECTION
LOGGRPA  GROUP LNCTL=CA,
               LOGICAL=YES,
               PHYSRC=PHYSPUA
LOGLINE8 LINE  ADDRESS=(8),
               OWNER=VTAMA
ALSA     PU    PUTYPE=2,
               ANS=CONT
```

```
*    NCPB

PCCUA    PCCU  OWNER=VTAMA,
               BACKUP=YES
NCPB     BUILD SUBAREA=2
         .
         .
*
* CHANNEL ADAPTERS
* DEFINITIONS FOR ESCON
* CONNECTIONS
*
* FIRST DEFINE PHYSICAL
* CONNECTION
*
GROUPB   GROUP LNCTL=CA,
CALINE8  LINE  ADDRESS=(8),
               OWNER=DUMMY
PHYSPUA  PU    PUTYPE=1,
*
* DEFINE LOGICAL CONNECTIONS
* FOR PU2.1 NORMAL MODE
*
* PHYSRC SHOWS ACTUAL PHYSICAL
* CONNECTION
LOGGRPA  GROUP LNCTL=CA,
               LOGICAL=YES,
               PHYSRC=PHYSPUA
LINELOG8 LINE  ADDRESS=(8),
               OWNER=DUMMY
ALSA     PU    PUTYPE=2,
               ANS=CONT
```

*Figure 25. Sample NCP Generation for a 3746 Channel Adapter*

# Index

## Special Characters



## A



## C

# J



# L



# M

## P



## Q



## R

File Number: S370/30XX-34
Program Number: 5748-T14

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SH31-0131-14