APPC Subsystem on MPE XL
Node Manager's Guide

HP 3000 MPE/iX Computer Systems

Edition 2

Manufacturing Part Number: 30294-90007
E0692

U.S.A. June 1992

Notice

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

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

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

Restricted Rights Legend

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

Acknowledgments

UNIX is a registered trademark of The Open Group.

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

Contents

Contents

4. APPC Subsystem Configuration

5. Managing the APPC Subsystem

Contents

Contents

Figures

Figures

Tables

Tables

Preface

This manual introduces the fundamental concepts and vocabulary of Advanced Program-To-Program Communication. It also provides the information necessary for configuration and day-to-day operation of the APPC subsystem on the MPE XL operating system.

NOTE

MPE/iX, Multiprogramming Executive with Integrated POSIX, is the latest in a series of forward-compatible operating systems for the HP 3000 line of computers.

In HP documentation and in talking with HP 3000 users, you will encounter references to MPE XL, the direct predecessor of MPE/iX. MPE/iX is a superset of MPE XL. All programs written for MPE XL will run without change under MPE/iX. You can continue to use MPE XL system documentation, although it may not refer to features added to the operating system to support POSIX (for example, hierarchical directories).

Finally, you may encounter references to MPE V, which is the operating system for HP 3000s, not based on the PA-RISC architecture. MPE V software can be run on the PA-RISC HP 3000s (Series 900) in what is known as *compatibility mode*.

Audience

This manual is intended for the HP 3000 node manager or HP 3000 system manager who is responsible for data communications between the HP 3000 and the remote system. The node manager should have knowledge of the MPE XL operating system, should be familiar with the remote environment, and should have HP 3000 Node Manager (NM) capabilities.

Organization

This manual is divided into the following sections and appendices:

Chapter 1 , “Introduction,” briefly describes Advanced Program-to-Program Communication and gives a general description of the APPC subsystem.

Chapter 2 , “Interactive Control Operator Commands,” describes the commands that can be used interactively to control and monitor the APPC subsystem.

Chapter 3 , “Control Operator Intrinsics,” describes the intrinsics that can be used programmatically to control and monitor the APPC subsystem.

Chapter 4 , “APPC Subsystem Configuration,” gives instructions for configuring the APPC subsystem through the NMMGR configuration utility.

Chapter 5 , “Managing the APPC Subsystem,” discusses how the APPC subsystem is activated and deactivated, how conversations are established between local and remote transaction programs, and how the number of active APPC sessions can be changed.

Chapter 6 , “Troubleshooting the APPC Subsystem,” describes how to identify an APPC subsystem problem and the steps to take to resolve the problem.

Appendix A , “Messages,” lists the error and warning messages that the APPC subsystem can generate, the return codes that APPC subsystem intrinsics can return, and the configuration validation messages that NMMGR can generate during APPC subsystem configuration.

Appendix B , “Sample Configuration,” shows the NMMGR configuration screens for an example SNA node and APPC subsystem configuration. It also shows a critical summary of the HP 3000 configuration generated by NMMGR, and it lists portions of a remote system APPC configuration.

Appendix C , “Configuration Worksheets,” provides forms you can fill out to help you plan your APPC subsystem configuration. It also includes instructions for filling out the forms.

Appendix D , “LU 6.2 API/XL Installation Guidelines,” describes the tasks involved in installing LU 6.2 API/XL on your system. It lists the hardware and software required on the IBM system and the HP 3000 in order to run LU 6.2 API/XL.

Related HP Publications

The MPE XL node manager can find related information in these manuals:

- *SNA Link/XL Node Managers Guide*
- *Using the Node Management Services Utilities*
- *LU 6.2 API Application Programmer's Reference Manual*
- *MPE XL Commands Reference Manual*
- *MPE XL Intrinsics Reference Manual*
- *HP SNA Server/Access User's Guide*
- *IBM Host System Programmer References:*
  - *HP SNA Products: Manager's Guide*
  - *HP SNA Products: ACF/VTAM and ACF/NCP Guide*
  - *HP SNA Products: IMS Guide*
  - *HP SNA Products: CICS Guide*
  - *HP SNA Products: AS/400 Guide*

1 Introduction

This chapter introduces the basic concepts of Advanced Program-to-Program Communication (APPC) and gives a general description of the APPC subsystem.

APPC Overview

IBM's Logical Unit 6.2 (LU 6.2) architecture specifies a common set of functions and protocols that application programs running on separate processors can use to communicate with each other. LU 6.2 allows for standardized program-to-program communication, which IBM calls **Advanced Program-to-Program Communication**, or **APPC**.

The functions and protocols that LU 6.2 architecture specifies are called verbs. Application programs used for communicating in an LU 6.2 environment are called **transaction programs**, or TPs. When two **TPs** communicate with each other, they are said to communicate with each other over a **conversation**.

LU 6.2 Verbs

The functions and protocols for communication are specified in the form of programmatic conversation verbs. These **conversation verbs** allow programs to perform such functions as initiating a conversation with another program, sending and receiving data, requesting confirmation from the partner program, requesting and granting permission to send data, and terminating a conversation.

LU 6.2 architecture specifies two types of conversation verbs that TPs can use to communicate: basic conversation verbs and mapped conversation verbs.

- **Basic conversation verbs**. Basic conversation verbs are used to write TPs that provide a service or system for other TPs. They are also used to write TPs that require certain privileged functions that only basic conversation verbs allow. The LU 6.2 data stream (called the **general data stream**) requires a unique data format. A programmer using basic conversation verbs must provide data formatting and certain error recovery functions in the TP itself.
- **Mapped conversation verbs**. Mapped conversation verbs are used to simplify the programmatic interface. A programmer using mapped conversation verbs is freed from providing formatting functions for the data in the general data stream. However, data formatting must be provided in an underlying TP or system.

Both groups of conversation verbs (basic and mapped) are further divided into **base sets** and **option sets**. The base set of a group of verbs must be implemented in all products that use that set, while product developers may or may not choose to implement additional option sets. The APPC subsystem implements the **base set of basic conversation verbs** and certain option sets. The LU 6.2 API product implements the **base set of mapped conversation verbs** and certain option sets.

Transaction Programs

There are two kinds of TPs: **application TPs** and **service TPs**.

- Application TPs are programs that perform services for end users. They typically use mapped conversation verbs. An application TP written with mapped conversation verbs must rely on a service TP or underlying system to perform the data formatting and error recovery functions.
- Service TPs are programs that provide a service or system to other TPs. For example, a service TP might provide data formatting and error recovery for application TPs. Service TPs are generally written in assembly language, and they usually use basic conversation verbs.

Conversations

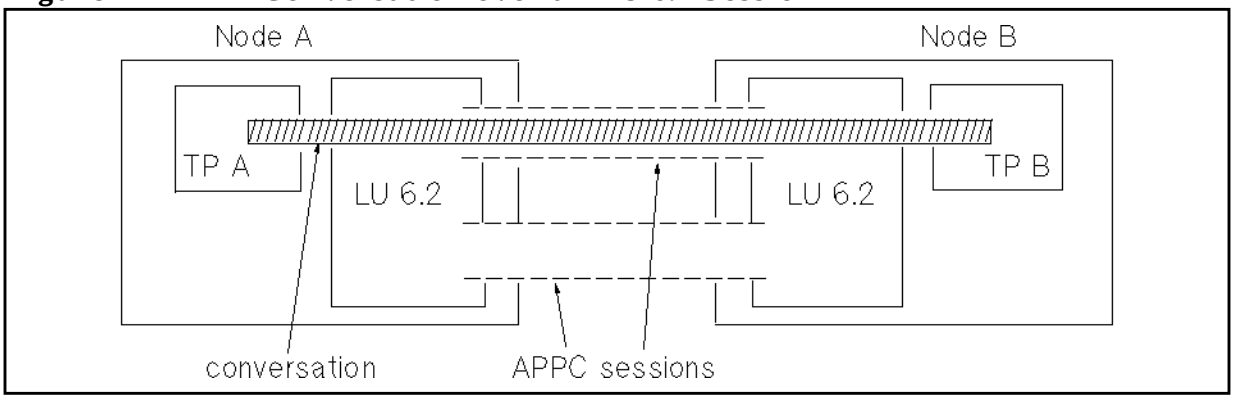
TPs are said to communicate with each other over conversations. Just as in human conversations, one TP must initiate the conversation. Once the conversation is established, a TP may perform such functions as sending or receiving data, asking for or giving confirmation of data received or sent, asking for or giving permission to send data, and informing the partner TP of an error condition. When the transaction is complete, one of the TPs must terminate the conversation.

The APPC Session

The APPC session is the logical connection between LUs that allows a conversation to take place. It is independent of any conversation conducted over it. Just as two people cannot have a telephone conversation until a phone connection is established, TPs cannot conduct a conversation until a session is established. After a conversation using a session has stopped, the session may be terminated or allowed to remain active.

In Figure 1-1, two sessions are active between node A and node B. One of these sessions is being used for a conversation between transaction program A (TP A) and transaction program B (TP B). The other session is active, but it is not being used.

Figure 1-1 Conversation over an LU 6.2 Session



This section has introduced terms and concepts you need to be familiar with as you work in the APPC environment. The remainder of this chapter describes the APPC subsystem.

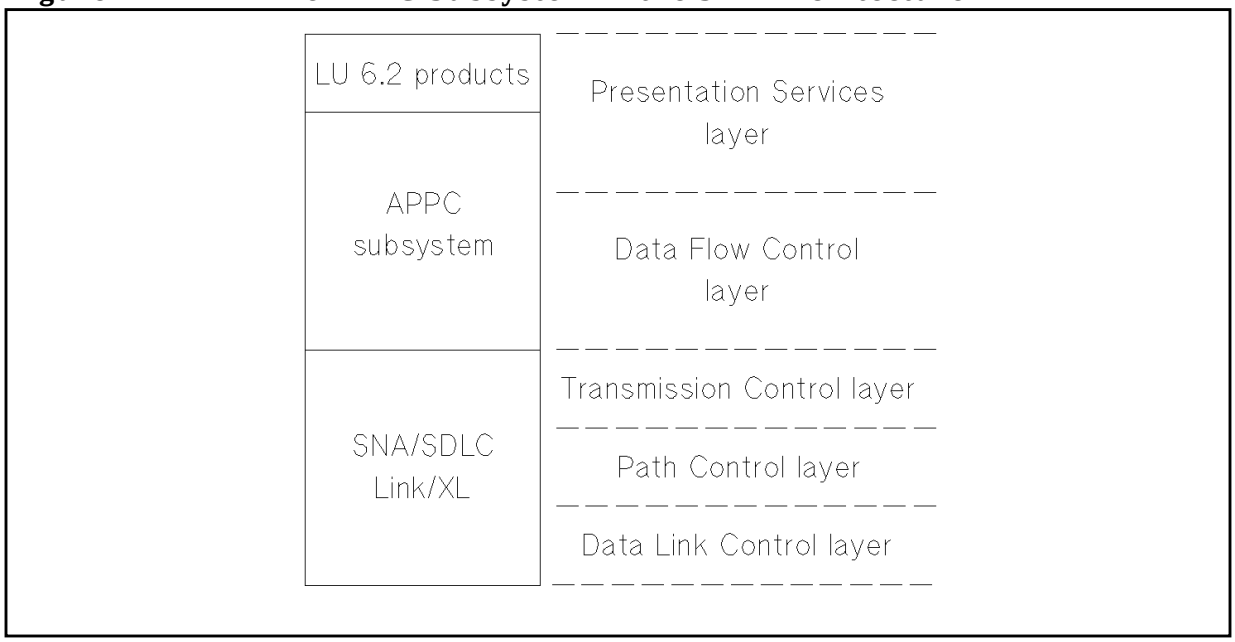
The APPC Subsystem

The APPC subsystem is a component of several of Hewlett-Packard's LU 6.2 products; multiple LU 6.2 products on the same node use one APPC subsystem. The APPC subsystem implements the base set of basic conversation verbs and certain option sets. These verbs are implemented as **intrinsics** that are accessed by the transaction programs associated with Hewlett-Packard's LU 6.2 products.

Structure of the HP 3000 Node

The APPC subsystem implements the Data Flow Control and Presentation Services layers of the SNA architecture. SNA/SDLC Link/XL must be installed on the same processor with the APPC subsystem. SNA/SDLC Link/XL provides the Path Control and Transmission Control layers of SNA. Figure 1-2 shows the APPC subsystem and the SNA link product as they relate to the SNA architecture.

Figure 1-2 The APPC Subsystem in the SNA Architecture



NOTE SNA/X.25 Link/XL can be used in place of SNA/SDLC Link/XL, but SNA/X.25 Link/XL emulates a Type 2.0 node, not a Type 2.1 node. Therefore, if you run the APPC subsystem on top of SNA/X.25 Link/XL, all your HP 3000 LUs must be dependent LUs, and you cannot communicate peer-to-peer with a Type 2.1 node like an AS/400.

The SNA Link Product

SNA/SDLC Link/XL is made up of a Programmable Serial Interface (PSI) card and a software module called SNA Transport. In order to connect the HP 3000 directly to multiple remote systems, or to connect multiple communication lines from the HP 3000 to a remote system, you need to install multiple copies of SNA/SDLC Link/XL.

HP 3000 Node Types

The APPC subsystem, together with SNA/SDLC Link/XL, allows the HP 3000 to function as either a **Node Type 2.0** or a **Node Type 2.1**. As a Node Type 2.0, the HP 3000 can communicate with a Node Type 5 (like an IBM mainframe). As a Node Type 2.1, the HP 3000 can communicate peer-to-peer with another Node Type 2.1 (like an IBM AS/400).

As a Node Type 2.1, the HP 3000 can participate in Advanced Peer-to-Peer Networking (APPN). APPN is an extension to SNA that allows Type 2.1 (peer) nodes to communicate without the intervention of the SSCP on a host node. In an APPN network, certain nodes are configured as Network Nodes, which perform intermediate session routing between nodes.

The HP 3000 functions as a Low Entry Networking (LEN) node in an APPN network. It cannot perform intermediate session routing, but if it is connected to a Network Node, it can take advantage of the Network Node's intermediate routing capabilities to communicate with non-adjacent nodes (nodes not directly connected to the HP 3000).

NOTE When you configure an IBM AS/400 to communicate with LU 6.2 API on the HP 3000, you must configure the HP 3000 controller as a LEN node in an APPN network.

APPC Sessions and LUs

The APPC subsystem supports 256 active APPC sessions. The combined total of active sessions, for all LU 6.2 products running on the APPC subsystem, may not exceed 256. Each TP conversation requires one session.

Though conversations can be initiated locally or remotely, all APPC sessions are controlled locally by the node manager. The node manager can activate or deactivate sessions interactively with control operator commands or programmatically with control operator intrinsics. The node manager can also control the number of active sessions automatically through configuration.

APPC session parameters must be defined in APPC subsystem configuration. For more information on APPC subsystem configuration, see Chapter 4 , “APPC Subsystem Configuration,” in this manual.

The LUs on an HP 3000 node can be configured as **dependent LUs** or **independent LUs**.

- A **dependent LU** can communicate only with dependent LUs on a Type 5 (host) node. It functions as a secondary LU; that is, it cannot issue a BIND to initiate an APPC session. When a dependent LU wants an APPC session with the host, it must wait for the host to send the BIND. It can send an INIT_SELF request to the host, requesting that the host send a BIND, or it can be configured to accept unsolicited BINDs. A dependent LU can carry on only one session at a time.
- An **independent LU** can communicate directly with an independent LU on a Type 2.1 (peer) node, like an IBM AS/400. The APPC subsystem allows the HP 3000 to operate as a Type 2.1 node in a peer-to-peer environment, where two Type 2.1 nodes can establish APPC sessions between themselves without the supervision of the SSCP on a host node.

  An independent LU can function as either a primary or a secondary LU; that is, it can initiate a session by sending a BIND to the remote LU, or it can receive a BIND from the remote LU.

  An independent LU on the HP 3000 can carry on multiple (parallel) sessions with a remote independent LU. It can also carry on sessions with several remote independent LUs at once; however, all the remote LUs with which it communicates must be connected to the HP 3000 through the same local SNA node (the same copy of SNA/SDLC Link/XL).

  Some remote systems, like the IBM AS/400, can perform intermediate routing between nodes in an SNA network. An independent LU connected to one of these systems can take advantage of its routing capabilities to communicate with nodes that are not directly connected to the HP 3000.

Figure 1-3 shows an HP 3000 directly connected to two IBM AS/400s and to an IBM host. A separate copy of SNA/SDLC Link/XL is required for each HP-to-IBM connection.

The dotted lines in Figure 1-3 represent APPC sessions.

Independent LU `INDLUA` on the HP 3000 is conducting a session with independent LU `AS400LU1` on IBM AS/400 #1. `INDLUA` cannot communicate with either of the remote independent LUs `AS400LU2` or `AS400LU3`, because they are connected to the HP 3000 through a different local SNA node (a different copy of SNA/SDLC Link/XL).

Independent LU `INDLUB` on the HP 3000 is conducting three parallel sessions with independent LU `AS400LU2` on IBM AS/400 #2. `INDLUB` is also conducting a session with `AS400LU3` on IBM AS/400 #3. `INDLUB` can

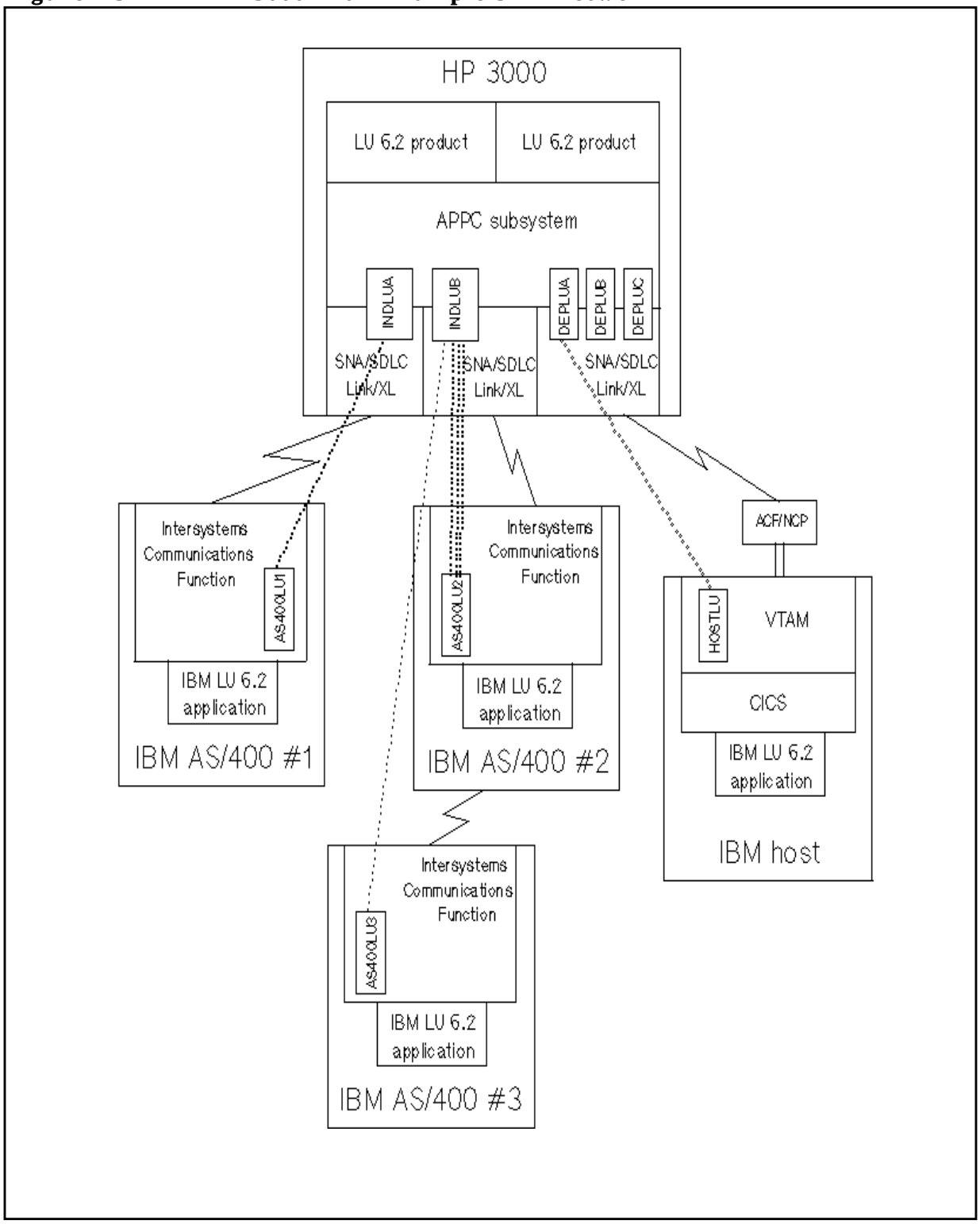
conduct sessions with LUs on two different nodes, because both nodes are connected to the HP 3000 through the same copy of SNA/SDLC Link/XL.

AS/400 #2 is directly connected to the HP 3000. It is configured as a Network Node, so it can route sessions from the HP 3000 to AS/400 #3, which is not directly connected to the HP 3000.

Dependent LU DEPLUA on the HP 3000 is conducting one session with dependent LU HOSTLU on the IBM host. Because it is a dependent LU, it can conduct only one session at a time. Every APPC session with a remote dependent LU requires one local dependent LU on the HP 3000.

NOTE The HP 3000 does not have primary SDLC link capability, so it cannot activate the link to the remote system.

Figure 1-3 HP 3000 in an Example SNA Network



The Node Manager's Interface to the APPC Subsystem

Through the APPC subsystem, the node manager can control the network activities for LU 6.2 products. The APPC subsystem interface provides interactive control operator commands that allow the node manager to perform the following functions:

- Activate and deactivate the APPC subsystem.
- Control the number of active APPC sessions.
- Determine the status of APPC sessions.
- Display performance statistics for active APPC sessions.
- Enable and disable the gathering of performance statistics for the APPC subsystem.
- Enable and disable tracing of APPC subsystem internal events.
- Dump the APPC subsystem internal data structures to a dump file.
- Perform LU 6.2 software module version checking.

The APPC subsystem also provides programmatic control operator intrinsics that allow transaction programs to perform the following functions:

- Activate and deactivate the APPC subsystem.
- Control the number of active APPC sessions.
- Determine whether or not the APPC subsystem is active.

Chapter 2 , "Interactive Control Operator Commands," describes the control operator commands, and Chapter 3 , "Control Operator Intrinsics," describes the control operator intrinsics.

The APPC subsystem generates error and warning messages and provides an internal tracing facility. Internal tracing records APPC subsystem internal activity. It can be activated or deactivated as needed. There is also a logging facility that you can configure to record major APPC subsystem events such as startup and shutdown. Logging configuration is described in the *SNA Link/XL Node Manager's Guide*.

2 Interactive Control Operator Commands

This chapter describes the APPCCONTROL commands used to control APPC subsystem functions.

NOTE

For all releases of the APPC subsystem that do not support Node Type 2.1, the APPCCONTROL commands are MPE commands interpreted by the MPE command interpreter. However, for the Node Type 2.1 version of the APPC subsystem, the APPC subsystem has its own command interpreter for APPCCONTROL commands.

For the Node Type 2.1 version, a system UDC file, which is installed with the APPC subsystem, will translate APPCCONTROL commands into MPE RUN commands that invoke the APPCCONTROL command interpreter. Before you can issue any APPCCONTROL commands, you must issue the following command to set the system UDC file:

```
SETCATALOG APPCUDC.APPC.SYS;SYSTEM;APPEND
```

You must have NM (Node Manager) or NA (Network Administrator) capability to issue APPCCONTROL commands.

The commands in this chapter are listed alphabetically.

Table 2-1 lists the APPC subsystem control operator commands.

Table 2-1 Control Operator Commands

APPCCONTROL DUMP	Dumps APPC subsystem internal data structures to dump file APPCDPxx.APPC.SYS.
APPCCONTROL HELP	Displays help text for APPCCONTROL commands.
APPCCONTROL PERFORMANCE	Displays session statistics and performance-related data, if statistics gathering is enabled.
APPCCONTROL PERFORMANCEOFF	Disables the gathering of APPC subsystem performance statistics.
APPCCONTROL PERFORMANCEON	Enables the gathering of APPC subsystem performance statistics during run time.
APPCCONTROL SESSIONS	Controls the number of active APPC sessions of a specified APPC session type.
APPCCONTROL START	Activates the APPC subsystem and all APPC sessions configured for automatic activation.
APPCCONTROL STATUS	Displays the status of the APPC subsystem.
APPCCONTROL STOP	Deactivates all APPC sessions, writes out performance data to the log file (if statistics gathering is enabled and logging is configured), and brings down the APPC subsystem.
APPCCONTROL STOPSESSION	Deactivates an APPC session for a specified APPC session type.
APPCCONTROL TRACEOFF	Disables APPC subsystem internal state tracing.
APPCCONTROL TRACEON	Enables APPC subsystem internal state tracing.
APPCCONTROL VERSION	Displays software module version information for the APPC subsystem and LU 6.2 products.

APPCCONTROL DUMP

Dumps APPC subsystem internal data structures to dump file APPCDPxx.APPC.SYS.

Syntax

```
APPCCONTROL DUMP
```

Description

APPCCONTROL DUMP creates a dump of the APPC subsystem internal data structures. The name of the dump file it creates is APPCDP*xx*.APPC.SYS, where *xx* is a number from 00 through 49.

Only 25 dump files can exist on disk at one time. Files APPCDP00 through APPCDP24 are created, and then when APPCDP25 is created, APPCDP00 is purged. When APPCDP26 is created, APPCDP01 is purged, and so on. After APPCDP49 is created, the numbering wraps, and the next file created is APPCDP00.

When you store off dump files for analysis by your HP representative, store off all files named APPCDP*xx*.APPC.SYS. That way, you can be sure you have the most recent dump file.

APPCCONTROL HELP

Displays help text for APPCCONTROL commands.

Syntax

```
APPCCONTROL HELP
```

Description

Type APPCCONTROL HELP to get online help for APPCCONTROL commands.

NOTE The MPE command HELP APPCCONTROL will not work, because the APPCCONTROL commands are not interpreted by the MPE command interpreter.

If you want to print the output from the APPCCONTROL HELP command, you can redirect the output to a disk file and then print the file. Redirecting the output creates a temporary file, and you must save the file before you can print it. The following commands send the output from the APPCCONTROL HELP command to the temporary file outfile and then save outfile to disk:

```
:APPCCONTROL HELP > outfile
:SAVE outfile
```

APPCCONTROL PERFORMANCE

Displays session statistics and performance-related data, if the gathering of performance statistics has been enabled.

Syntax

```
APPCCONTROL PERF[ORMANCE]
             [;STYPE=Session Type Name]
[;SID=Session ID]
```

Parameters

Session Type Name

An active session type name. Performance statistics are displayed for all active sessions of the session type specified. The name you specify must be configured in the APPC subsystem configuration file. See Chapter 4 , "APPC Subsystem Configuration," of this manual for more information on configuring session type names.

Default: All active session type names.

Session ID

The identification number of the session for which you are requesting performance data. You can obtain session ID numbers by issuing the APPCCONTROL STATUS command, described later in this chapter.

Description

This command displays session performance statistics for the session type or session specified. If no session type or session ID is specified, statistics will be displayed for all active sessions.

If APPC subsystem logging is configured (through the LOGGING branch of NMMGR), the same types of statistics that APPCCONTROL PERFORMANCE displays will be logged whenever an APPC session is brought down, locally or remotely.

NOTE The APPCCONTROL PERFORMANCE command does not write performance data to the log file. If APPC subsystem logging is configured and performance statistics gathering is enabled, performance data is written to the log file whenever a session is brought down.

Executing the APPCCONTROL PERFORMANCE command will not interrupt the sessions for which statistics are being gathered.

The APPCCONTROL PERFORMANCE display pauses at the end of every screen of data. To see the next screen of data, press **[RETURN]**.

NOTE

You must enable the gathering of performance statistics before issuing the APPCCONTROL PERFORMANCE command, or no statistics will be available for display.

To enable statistics gathering at startup, specify PERFON = YES in the APPCCONTROL START command, or set the *PerformanceOn* parameter of the APPCStart intrinsic to 1. To enable statistics gathering once the APPC subsystem is active, issue the APPCCONTROL PERFORMANCEON command.

The APPCCONTROL START and APPCCONTROL PERFORMANCEON commands are described later in this chapter. The APPCStart intrinsic is described in Chapter 3 , “Control Operator Intrinsics.”

If you want to print the output from the APPCCONTROL PERFORMANCE command, you can redirect the output to a disk file and then print the file. Redirecting the output creates a temporary file, and you must save the file before you can print it. The following commands send the output from the APPCCONTROL PERFORMANCE command to the temporary file outfile and then save outfile to disk:

```
:APPCCONTROL PERFORMANCE > outfile
:SAVE outfile
```

Examples

In this example, the APPCCONTROL PERFORMANCE command was issued for the independent session type SESS1. Two sessions of session type SESS1 are active on the independent LU INDLU1.

```
:APPCCONTROL PERFORMANCE;STYPE=SESS1

   Current Time      : FRI, JUN 20, 1990,  4:36 AM
   Subsystem Started : FRI, JUN 20, 1990,  1:50 AM
   Duration          : 2 hours 46 minutes 39 seconds
   TPs Started per Subsystem          = 26

SESS1   , LU = INDLU1  , NODE = SNANODE , SID = 6612

   Session Started   : FRI, JUN 20, 1990,  3:37 AM
   Duration          : 59 minutes 2 seconds
   Bytes Sent                         = 0
   Bytes Received                     = 5
   RUs Sent                           = 0
   RUs Received                       = 1
   Negative Responses Sent            = 0
   Negative Responses Received        = 0
   Avg. Remote Response Time (msec)   = 0
   Avg. CONFIRM Time (msec)           = 0
   Conversations Started              = 0
   Avg. bytes per Conversation        = 0
   Avg. Conversation Time (sec)       = 0

SESS1   , LU = INDLU1  , NODE = SNANODE , SID = 7534

   Session Started   : FRI, JUN 20, 1990,  3:40 AM
   Duration          : 43 minutes 3 seconds
   Bytes Sent                         = 0
   Bytes Received                     = 34
   RUs Sent                           = 0
   RUs Received                       = 1
   Negative Responses Sent            = 0
   Negative Responses Received        = 0
   Avg. Remote Response Time (msec)   = 0
   Avg. CONFIRM Time (msec)           = 0
   Conversations Started              = 0
   Avg. bytes per Conversation        = 0
   Avg. Conversation Time (sec)       = 0
```

In this example, the APPCCONTROL PERFORMANCE command was issued for an individual active session (session ID 2298).

```
:APPCCONTROL PERFORMANCE;SID=2298

   Current Time      : FRI, JUN 20, 1990,  5:36 AM
   Subsystem Started : FRI, JUN 20, 1990,  1:50 AM
   Duration          : 2 hours 46 minutes 39 seconds
   TPs Started per Subsystem          = 26
```

```
SESS3   , LU = DEPLU8  , NODE = IBMNODE , SID = 2298

   Session Started   : FRI, JUN 20, 1990,  2:37 AM
   Duration          : 30 minutes 2 seconds
   Bytes Sent                          = 1011
   Bytes Received                      = 5
   RUs Sent                            = 5
   RUs Received                        = 1
   Negative Responses Sent             = 0
   Negative Responses Received         = 0
   Avg. Remote Response Time (msec)    = 0
   Avg. CONFIRM Time (msec)            = 0
   Conversations Started               = 0
   Avg. bytes per Conversation         = 0
   Avg. Conversation Time (sec)        = 0
```

Reading the Display

Each line of the APPCCONTROL PERFORMANCE display is described below:

Current Time

The time at which APPCCONTROL PERFORMANCE was issued.

Subsystem Started

The time at which APPCCONTROL START was issued.

Duration

The length of time the APPC subsystem has been active.

TPs Started per Subsystem

The number of transaction program processes started since APPCCONTROL START was issued.

Session Type Name, LU Name, Node Name, and Session ID

The session type name, local LU name, and SNA node name configured for this session type, and the session identification number being used for the session. A separate set of statistics will be seen for each active APPC session for which performance data is requested.

Session Started

The time at which the APPC session was activated.

Duration

The length of time the APPC session has been active.

Bytes Received

The total number of bytes received since session initiation.

Bytes Sent

The total number of bytes sent since session initiation.

RUs Sent

The total number of RUs sent since session initiation.

RUs Received

The total number RUs received since session initiation.

Negative Responses Sent

The total number of negative responses sent over this session since session initiation.

Negative Responses Received

The total number of negative responses received over this session since session initiation.

Avg. Remote Response Time (msec)

The average time, in milliseconds, that the remote system has taken to respond after the last message has been sent and the local system has prepared to receive messages from the remote system.

Avg. CONFIRM Time (msec)

The average time, in milliseconds, that the remote system has taken to send a CONFIRMED reply after a confirmation request has been issued.

Conversations Started

The total number of conversations initiated since session initiation.

Avg. Bytes per Conversation

The average number of bytes sent and received per conversation over this session.

Avg. Conversation Time (sec)

The average time, in seconds, that each conversation being conducted over this session has taken.

APPCCONTROL PERFORMANCEOFF

Disables the gathering of session statistics and performance-related data.

Syntax

```
APPCCONTROL PERF[ORMANCE]OFF
```

Description

The APPCCONTROL PERFORMANCEOFF command disables the gathering of APPC subsystem performance statistics.

When the APPC subsystem starts up, statistics gathering is disabled by default. To enable statistics gathering at subsystem startup, you specify PERFON = YES in the APPCCONTROL START command, described later in this chapter. If you start the APPC subsystem programmatically, you can enable statistics gathering by setting the *PerformanceOn* parameter of the APPCStart intrinsic to 1. See Chapter 3 , “Control Operator Intrinsics,” for more information on control operator intrinsics.

Once the APPC subsystem is active, you can enable statistics gathering by issuing the APPCCONTROL PERFORMANCEON command, described later in this chapter.

The APPCCONTROL PERFORMANCEOFF command is used to disable statistics gathering once the APPC subsystem is active. Executing the APPCCONTROL PERFORMANCEOFF command will not interrupt the sessions for which statistics are being gathered.

APPCCONTROL PERFORMANCEON

Enables the gathering of session statistics and performance-related data.

Syntax

```
APPCCONTROL PERF[ORMANCE]ON
```

Description

The APPCCONTROL PERFORMANCEON command enables the gathering of APPC subsystem performance statistics.

NOTE The APPCCONTROL PERFORMANCEON command does not display performance statistics; it only enables the gathering of performance statistics. To display performance statistics, issue the APPCCONTROL PERFORMANCE command.

When the APPC subsystem starts up, statistics gathering is disabled by default. Once the APPC subsystem is active, you can enable statistics gathering by issuing the APPCCONTROL PERFORMANCEON command. Executing the APPCCONTROL PERFORMANCEON command will not interrupt the sessions for which statistics are being gathered.

To enable statistics gathering at subsystem startup, you specify PERFON = YES in the APPCCONTROL START command, described later in this chapter. If you start the APPC subsystem programmatically, you can enable statistics gathering by setting the *PerformanceOn* parameter of the APPCStart intrinsic to 1. See Chapter 3 , “Control Operator Intrinsics,” for more information on control operator intrinsics.

NOTE Statistics gathering must be enabled before you issue the APPCCONTROL PERFORMANCE command. If you issue the APPCCONTROL PERFORMANCE command with statistics gathering disabled, no statistics will be available for display, and you will receive an error message. See the description of APPCCONTROL PERFORMANCE, earlier in this chapter.

APPCCONTROL SESSIONS

Controls the number of active APPC sessions of a specified APPC session type.

Syntax

```
APPCCONTROL SESS[IONS]
              ;STYPE=Session Type Name
             ;LIMIT=New Session Limit
```

Parameters

Session Type Name

Required. A session type that is configured in the APPC subsystem configuration file.

New Session Limit

Required. The new number of active sessions for the session type. The following restrictions apply to the new session limit:

1. The new session limit for the specified session type may not exceed the `Maximum Number of Sessions` value configured for the session type. To find out the `Maximum Number of Sessions` value for a session type, issue the `APPCCONTROL STATUS` command, described later in this chapter. See Chapter 4 , “APPC Subsystem Configuration,” for information on session type configuration.
2. The total number of simultaneously active sessions, for all LU 6.2 products using the APPC subsystem, may not exceed 256.

Description

The `APPCCONTROL SESSIONS` command changes the number of active sessions for a specified session type. The new session limit for the session type is in effect only while the APPC subsystem is currently active; the `APPCCONTROL SESSIONS` command does not update the APPC subsystem configuration file.

When you issue the `APPCCONTROL SESSIONS` command to raise the session limit for a session type, sessions will be activated until the new session limit is reached.

Using the APPCCONTROL SESSIONS command to reduce the session limit for a session type causes an orderly termination of the sessions affected; termination will not occur until conversations on the affected sessions have ended. All current conversations for the affected sessions are allowed to complete, but no new conversations are allocated for those sessions.

NOTE

If you are changing the session limit for an independent LU running parallel sessions, the new limit will be negotiated with the partner LU on the remote system. Therefore, the actual new limit might not agree with the limit you requested in your APPCCONTROL SESSIONS command.

To find out the actual, negotiated session limit, issue the APPCCONTROL STATUS command, described later in this chapter.

For dependent LUs and independent LUs running single sessions, the new value is not negotiated.

You can modify session limits programmatically by calling the APPCSessions intrinsic, described in Chapter 3 , “Control Operator Intrinsics.”

APPCCONTROL START

Activates the APPC subsystem. Any APPC sessions configured for automatic activation will be established when the subsystem is activated

Syntax

```
APPCCONTROL START
           [;TRACEON=tracing option]
           [;TFILENAME=trace file name]
           [;TFILESIZE=record count]
           [;PERFON=performance option]
           [;ERROPT=error option]
```

Parameters

tracing option

Enables or disables APPC subsystem internal state tracing. Internal tracing records significant state changes, events, and data to a disk file.

YES	APPC subsystem internal state tracing is enabled.
NO	APPC subsystem internal state tracing is disabled.

If tracing is enabled at subsystem startup, it can be turned off during run time by issuing the APPCCONTROL TRACEOFF command. If tracing is disabled at subsystem startup, it can be turned on during run time by issuing the APPCCONTROL TRACEON command.

Default: NO

trace file name

An 8-character name consisting of alphanumeric characters and beginning with a letter. A trace file with this name will be created in the group and account from which the APPCCONTROL START command was issued.

The default trace file name is NMTC*xxxx*.PUB.SYS, where *xxxx* is a number from 0000 through 9999. This number is incremented whenever a new trace file is created.

A new trace file is created whenever tracing is enabled, at subsystem startup with the *TRACEON* parameter, or during run time with the APPCCONTROL TRACEON

command. To start a new trace file without bringing down the APPC subsystem, turn tracing off and back on again with the APPCCONTROL TRACEOFF and APPCCONTROL TRACEON commands.

record count

A decimal number, from 0 through 32767, specifying the trace file size, in 128-word records. Once the record count has been reached, trace records are overwritten starting at the beginning of the file.

Default: 4096.

performance option

Enables or disables the gathering of APPC subsystem performance statistics. These statistics are displayed when you issue the APPCCONTROL PERFORMANCE command.

YES	APPC subsystem performance statistics are gathered during run time.
NO	APPC subsystem performance statistics are not gathered.

Once the APPC subsystem is active, statistics gathering can be enabled and disabled during run time with the APPCCONTROL PERFORMANCEON and APPCCONTROL PERFORMANCEOFF commands, described earlier in this chapter.

Default: NO

error option

This parameter specifies the action taken when an irrecoverable subsystem internal error occurs. Possible values are as follows:

SYS	If ERROPT=SYS is specified, the system will be brought down when an irrecoverable internal error within the APPC subsystem occurs, and a full system dump will be taken.
SUB	If ERROPT=SUB is specified, only the APPC subsystem will be brought down when an irrecoverable internal error occurs within the APPC subsystem.

Hewlett-Packard recommends that you specify ERROPT=SUB so that the rest of the system can remain active when there is a problem with the APPC subsystem. However, if an APPC subsystem problem cannot be found, your HP representative may ask you to specify ERROPT=SYS so that a system dump is taken when the internal error occurs.

Default: ERROPT=SUB

Description

The APPCCONTROL START command activates the APPC subsystem and all APPC sessions configured for automatic activation. (See Chapter 4 , "APPC Subsystem Configuration," of this manual for more information on configuring the APPC subsystem.)

You can activate the APPC subsystem programmatically by calling the APPCStart intrinsic, described in Chapter 3 , "Control Operator Intrinsics."

NOTE The APPCCONTROL START command will not cause the activation of the SNA link product. The SNA link product must be explicitly activated using the SNACONTROL START command.

APPCCONTROL STATUS

Displays the status of the APPC subsystem.

Syntax

```
APPCCONTROL STAT[US] [;STYPE=Session Type Name]
```

Parameters

Session Type Name

The name of a session type configured in the APPC subsystem configuration file. Status information will be displayed for all APPC sessions of the specified type.

If a CNOS (Change-Number-Of-Sessions) request fails (at subsystem startup or during run time when you issue the APPCCONTROL SESSIONS command), specify STYPE=SNASVCMG to check whether any CNOS sessions are active. CNOS session information will be displayed only if you have independent LUs configured to conduct parallel sessions. The session type names displayed for CNOS sessions are internally defined and meaningless to the user. The mode name for CNOS sessions is always SNASVCMG.

Description

Use the APPCCONTROL STATUS command to check the status of the APPC subsystem. If no parameters are specified, status information will be displayed for all configured APPC session types and all active transaction programs.

For troubleshooting purposes, you can specify STYPE=SNASVCMG to determine whether CNOS sessions are active. Information will be displayed only for independent LU session types with parallel sessions.

The APPCCONTROL STATUS display pauses at the end of every screen of data. To see the next screen of data, press **[RETURN]**.

NOTE The APPCCONTROL STATUS command displays the current number of active sessions for each session type. This value is sometimes negotiated with the remote LU and may be different from the configured Automatically Activated Sessions value or the session limit you specified in an APPCCONTROL SESSIONS command.

If you want to print the output from the APPCCONTROL STATUS command, you can redirect the output to a disk file and then print the file. Redirecting the output creates a temporary file, so you must save the file to disk before you can print it. The following commands redirect the output from the APPCCONTROL STATUS command to the file outfile and then save outfile to disk:

```
:APPCCONTROL STATUS > outfile
:SAVE outfile
```

Examples

In the following example, the APPCCONTROL STATUS command was issued with no parameters. The status display includes all configured session types and all active TPs.

```
:APPCCONTROL STATUS

Internal Trace File : NMTC0020.PUB.SYS
Performance Tracing : ON

-------------------------------------------------
Number of Active Sessions for APPC Subsystem :   5
-------------------------------------------------

Session Type Information :

   ---------------------------------------
   Independent LU Session Type : STYPE1
   ---------------------------------------

   Number of Active Sessions         :   4
   Number of Queued Session Requests :   3
                                                    Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU1     SNANODE    NET1.IBMLUA         MODE1       50       50

   Session ID    LFSID      State      LU Type    TP ID
   ----------    -----      ------     -------    -----
     62          0103 1     ACTIVE      PRI       1011
     75          0104 1     ACTIVE      PRI       6712
     11          0106 1     ACTIVE      SEC       1788
     98          0105 1     ACTIVE      SEC       1219
```

```
   ---------------------------------------
   Independent LU Session Type : STYPE2
   ---------------------------------------

   Number of Active Sessions         :   0
   Number of Queued Session Requests :   0
                                                      Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU1     SNANODE    NET1.IBMLUA         MODE2        8        8

   Session ID    LFSID      State      LU Type    TP ID
   ----------    -----      ------     -------    -----
   * No active sessions

   ------------------------------------
   Dependent LU Session Type : STYPE3
   ------------------------------------

   Number of Active Sessions         :   1
   Number of Queued Session Requests :   1

                                          Session Limits
   SNA Node   Remote LU           Mode     Maximum  Current
   --------   -----------------   ------   -------  -------
   IBMNODE    DISOSSC             MODE3        8        8

   Session ID   LFSID   Local LU   State      LU Type   TP ID
   ----------   -----   --------   --------   -------   -----
      28        0101 1  DEPLU8     ACTIVE       SEC     2399

Transaction Program Information:

   TP ID    Program File Name       TP NAME      # Conv.    User Name
   -----    --------------------    ---------    -------    ------------
   1011     TP1.PUB.SYS             TP1             1       JOE.USER
   6712  R  TP3.PUB.SYS             TP3             0       JOE.USER
   1788     COPRTP.APPC.SYS         COPRTP          1       MGR.APPC
   1219  R  CNOSTP.APPC.SYS         CNOSTP          1       MGR.APPC
   2399     TP2.PUB.SYS             TP2             1       JOE.USER
```

In the following example, the APPCCONTROL STATUS command was issued for one independent session type: STYPE1. The display includes information about the active sessions of that session type and the transaction programs running on those sessions.

```
:APPCCONTROL STATUS;STYPE=SESS1

Internal Trace File : NMTC0020.PUB.SYS
Performance Tracing : ON

-------------------------------------------------
Number of Active Sessions for APPC Subsystem :    5
-------------------------------------------------

Session Type Information :

   ---------------------------------------
   Independent LU Session Type : STYPE1
   ---------------------------------------

   Number of Active Sessions         :   2
   Number of Queued Session Requests :   3
                                                    Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU1     SNANODE    NET1.IBMLUA         MODE1       50       50

   Session ID    LFSID    State       LU Type    TP ID
   ----------    -----    --------    -------    -----
     62          0000 1   ACTIVE        PRI      1011
     74          0000 2   ACTIVE        PRI      6712

Transaction Program Information:

   TP ID    Program File Name       TP Name      # Conv.    User Name
   -----    --------------------    ---------    -------    ------------
   1011     TP1.PUB.SYS             TP1             1       JOE.USER
   6712     TP2.PUB.SYS             TP2             1       JOE.USER
```

If APPC subsystem internal tracing has been turned off, “TRACEOFF” is displayed instead of the trace file name (NMTC0020.PUB.SYS, in the example).

Reading the Display

Each of the columns in the example status display is described below:

Number of Active Sessions

The current number of active or pending sessions for the session type. This number should be equal to the configured Automatically Activated Sessions value at subsystem startup, or, if you have issued the APPCCONTROL SESSIONS command to raise the session limit, it should be equal to the number you specified in the command. However, for independent LU session types running parallel sessions, the number of active sessions might be negotiated down from the number you specified.

Number of Queued Session Requests

The number of outstanding session requests for the session type. Session requests are queued when TPs call the MCAllocate or MCGetAllocate intrinsic and no sessions are immediately available.

Local LU

The name of a local LU associated with the session type.

SNA Node

The name of the SNA node configured for the session type.

Remote LU

The name of the destination LU on the remote system.

Mode

The mode configured for the session type. See Chapter 4 , “APPC Subsystem Configuration,” of this manual for more information on mode configuration.

Session Limits

The maximum number of sessions for the session type.

Maximum	The *configured* maximum number of sessions for the session type.
Current	The current maximum number of sessions. This number will be equal to the number of active sessions, unless the remote system has some sessions configured as contention winners (controlled by the remote LU).

Session ID

The unique identification number given to each session activated for the session type.

LFSID

Local Form Session Identifier. This number identifies the session to the SNA link product. The SNACONTROL STATUS command, which displays the status of the SNA link product, identifies each LU-LU session by LFSID. For more information on the SNACONTROL STATUS command, see the *SNA Link/XL Node Manager's Guide*.

State

The state of a session. Each of the possible states is described below:

NO SESS.	No APPC sessions are active or being activated for this session type. The session type's session limit has been set to 0.
WAITING	The APPC session is waiting for activation. No response has been received from the remote system or from SNA Transport.
PENDING	Activation of the APPC session is pending. Activation is proceeding but is not complete.
ACTIVE	The APPC session is ready for conversation.
TERM	The APPC session is in termination state.
ERROR	The APPC session is in an error state. You should bring down the APPC subsystem.

LU Type

Whether the local LU is the primary or secondary LU for the session. LU Types are as follows:

PRI	Primary LU. The local LU sends the BIND to initiate the session.
SEC	Secondary LU. The local LU receives the BIND from the remote LU.

TP ID

The unique identification number of an active transaction program process.

R	When the TPID is followed by R, the transaction program was started up by the remote system.

Program File Name

The fully qualified file name for an active transaction program.

TP Name

The configured transaction program name.

 # Conv.

The number of conversations being conducted by an active TP process. If this number is 0, the TP is waiting for an APPC session to be activated or freed, or it is waiting for an allocate request to arrive from the remote TP.

User Name

The user name and group name associated with an active TP process.

APPCCONTROL STOP

Deactivates all APPC sessions, writes out performance data to the log file (if performance statistics gathering is enabled and APPC subsystem logging is configured), and brings down the APPC subsystem.

Syntax

```
APPCCONTROL STOP [;TYPE=StopType]
```

Parameters

StopType Indicates the manner in which you wish the APPC subsystem to be shut down. Valid choices are as follows:

`K[ILL]` Immediate shutdown of the APPC subsystem. Any conversations will be terminated immediately. All APPC sessions will also be terminated immediately. User TPs will not be terminated.

`P[ROTOCOL]` Immediate shutdown of the APPC subsystem, pending notification of any transaction program currently in conversation. Once a TP has been notified, its session is terminated.

`Q[UIESCE]` Orderly shutdown of the APPC subsystem, allowing all current conversations to be completed, but preventing any new conversations from being allocated. No APPC session will be brought down until the conversation using it has been deallocated. When the conversation has completed, the session will be terminated.

Default: `QUIESCE`

Description

Try to use APPCCONTROL STOP with the *TYPE=Q* parameter; this will ensure orderly shutdown of the APPC subsystem. If it is not possible to wait for current conversations to complete, escalate the shutdown type to TYPE=P (protocol shutdown). A TYPE=P shutdown ensures that resources used by the APPC subsystem for conversations are successfully returned to the system before shutdown. If TYPE=P does not work, use TYPE=K.

This command shuts down APPC sessions only; any LU-LU sessions being used with other SNA Services remain active.

NOTE This command will not terminate any transaction programs. Users must terminate any transaction programs that are still active after their sessions have been terminated.

You can shut down the APPC subsystem programmatically by calling the APPCStop intrinsic, described in Chapter 3 , “Control Operator Intrinsics.”

APPCCONTROL STOPSESSION

Terminates the session with the specified *Session ID*.

Syntax

```
APPCCONTROL STOPSESS[ION]
            ;SID=Session ID
```

Parameters

Session ID **Required.** The unique session identification number of the session you want to terminate. You can obtain the *Session ID* by issuing the APPCCONTROL STATUS command.

Description

The APPCCONTROL STOPSESSION command is used to terminate a single APPC session. It does not alter the current session limit for the session type.

Use the APPCCONTROL STOPSESSION command to kill APPC sessions that could not be terminated with the APPCCONTROL SESSIONS command.

The APPCCONTROL STOPSESSION command does not provide a *StopType* parameter; the StopType is always KILL, meaning that the session will be terminated immediately, along with any conversation being conducted over the session. If you want to terminate a session with a StopType of PROTOCOL or QUIESCE, try using the APPCCONTROL SESSIONS command to reduce the current session limit for the session type. Use the APPCCONTROL STOPSESSION command only if the APPCCONTROL SESSIONS command fails to terminate an APPC session.

This command terminates APPC sessions only; it cannot be used to terminate LU-LU sessions being used with other SNA Services.

NOTE This command will not terminate any transaction programs. Users must terminate any transaction programs that are still active after their sessions have been terminated.

APPCCONTROL TRACEOFF

Disables internal state tracing for the APPC subsystem.

Syntax

```
APPCCONTROL  TRACEOFF
```

Description

The APPCCONTROL TRACEOFF command disables internal state tracing. Internal state tracing records significant state changes, events and data to a disk file. The disk file is closed when tracing is disabled.

When the APPC subsystem is activated, internal tracing is disabled by default. It can be enabled at startup by specifying TRACEON = YES in the APPCCONTROL START command. During run time, you can enable internal tracing with the APPCCONTROL TRACEON command.

If you start up the APPC subsystem programmatically, you can enable internal tracing by setting the *TraceOn* parameter of the APPCStart intrinsic to 1. See Chapter 3 , “Control Operator Intrinsics,” for more information on control operator intrinsics.

The APPCCONTROL TRACEOFF command allows you to disable internal state tracing without bringing down the APPC subsystem.

APPCCONTROL TRACEON

Enables internal state tracing for the APPC subsystem.

Syntax

```
APPCCONTROL TRACEON
            [;TFILENAME=trace file name]
            [;TFILESIZE=record count]
```

Parameters

trace file name

An 8-character name consisting of alphanumeric characters and beginning with a letter. A trace file with this name will be created in the group and account from which the APPCCONTROL TRACEON command was issued.

The default trace file name is NMTC*xxxx*.PUB.SYS, where *xxxx* is a number from 0000 through 9999. This number is incremented whenever a new trace file is created.

record count

A decimal number from 0 through 32767 specifying the trace file size, in 128-word records. Once the record count has been reached, trace records are overwritten starting at the beginning of the file.

Default: 4096.

Description

The APPCCONTROL TRACEON command enables internal state tracing. It can be issued during run time, after the APPC subsystem has been activated. Internal state tracing records significant state changes, events and data to a disk file. The default name of the disk file is NMTC*xxxx*.PUB.SYS, where *xxxx* is a number from 0000 through 9999. This number is incremented whenever a new trace file is opened.

A new trace file is created whenever tracing is enabled, at subsystem startup with the *TRACEON* parameter, or during run time with the APPCCONTROL TRACEON command. To start a new trace file without bringing down the APPC subsystem, turn tracing off and back on again with the APPCCONTROL TRACEOFF and APPCCONTROL TRACEON commands.

When the APPC subsystem is activated, internal tracing is disabled by default. It can be enabled at startup by specifying TRACEON = YES in the APPCCONTROL START command. See the description of APPCCONTROL START, earlier in this chapter.

If you start up the APPC subsystem programmatically, you can enable internal tracing by setting the *TraceOn* parameter of the APPCStart intrinsic to 1. See Chapter 3 , “Control Operator Intrinsics,” for more information on control operator intrinsics.

APPCCONTROL VERSION

Displays software module version information for the APPC subsystem and LU 6.2 products.

Syntax

```
APPCCONTROL VERSION
```

Description

The APPCCONTROL VERSION command is used to obtain the versions of LU 6.2 product modules that are currently installed on the system. The user will be informed if module versions do not match or if modules are missing, which indicates that the subsystem has been installed incorrectly.

Issue this command after installing the LU 6.2 software to determine whether you have a valid software installation and whether the software module versions match.

Example

In this example, the APPC subsystem is serving LU 6.2 API/XL.

```
:APPCCONTROL VERSION

LU 6.2 API   HP30294 module versions:

XL procedure:  APIVERS             Version:  B0000200
Catalog file:  CATAPI.PUB.SYS      Version:  B0000200

LU 6.2 API   HP30294 overall version = B.00.00

APPC Subsystem 32098-20057 module versions:

NL procedure:     HSVERS              Version:  B0000200
XL procedure:     COPRVERS            Version:  B0000200
XL procedure:     PSVERS              Version:  B0000200
XL procedure:     NMUTILVERS          Version:  B0000200
XL procedure:     SUBSYS16TRCFMTVERS  Version:  B0000200
SL procedure:     SUBSYS16LOGFMTVERS  Version:  B0000200
SL procedure:     CMUTILVERS          Version:  B0000200
NM program file:  RM.APPC.SYS         Version:  B0000206
NM program file:  SM.APPC.SYS         Version:  B0000200
NM program file:  COPRTP.APPC.SYS     Version:  B0000200
NM program file:  CNOSTP.APPC.SYS     Version:  B0000200
NM program file:  APPCCI.APPC.SYS     Version:  B0000200
NM program file:  APPCSYM.APPC.SYS    Version:  B0000200
DAT macros file:  APPCMAC.APPC.SYS    Version:  B0000200
UDC file:         APPCUDC.APPC.SYS    Version:  B0000200
Help file:        APPCHELP.APPC.SYS   Version:  B0000200
Catalog file:     CATAPPC.APPC.SYS    Version:  B0000202

APPC Subsystem 32098-20057    overall version = B.00.00
```

Each version ID number includes a version, update, and fix level (*V.uu.ff*) as well as an internal fix level. For example, where RM.APPC.SYS is listed above, its version ID number is B0000206. The "B" is the version level, the next two digits (00) represent the update level, and the following digits (00) represent the fix level. The remaining digits (206) show the internal fix level, which is used only within Hewlett-Packard.

Each software module within a subsystem, such as a program file or SL procedure, has its own version ID number. If the version, update, and fix levels of these modules do not match, the subsystem will not work correctly.

Notice that the first five characters of the version for each module listed in this group are B0000. This means that all the software modules in the subsystem match. The internal fix level, represented by the last three digits, does not need to match between modules for the software to be compatible. Because the module version ID numbers match, NMMAINT considers the overall subsystem version number to be B.00.00.

3 Control Operator Intrinsics

Control operator intrinsics are subroutines that you can call from inside programs written in COBOL II, Pascal, C, or Transact. They have the same functions as the interactive control operator commands described in Chapter 2 , “Interactive Control Operator Commands.”

You must have Node Manager (NM) capability to call control operator intrinsics. You must declare intrinsics with the keyword `INTRINSIC`. (See the appropriate HP language reference manual for more information on declaring and calling HP intrinsics.)

Table 3-1 lists the control operator intrinsics and their functions.

Table 3-1 Control Operator Intrinsics

`APPCSessions`	Controls the number of active sessions of a specified APPC session type.
`APPCStart`	Starts up the APPC subsystem and all APPC sessions configured for automatic activation.
`APPCStatus`	Indicates whether the APPC subsystem is active.
`APPCStop`	Shuts down the APPC subsystem and all active APPC sessions.

This chapter describes the control operator intrinsics, in alphabetical order. Data types are listed above the parameters, abbreviated as follows:

`I32`	32-bit signed integer, passed by reference
`I16V`	16-bit signed integer, passed by value
`CA`	character array

Output parameters are underlined. Optional parameters are enclosed in square brackets.

Return Codes

The *`ReturnCode`* parameter, a 32-bit integer passed by reference, is required in all intrinsic calls. It consists of two 16-bit fields: an information field and a subsystem field.

The information field (bits 0–15) may contain the following values:

0	The intrinsic executed successfully and there are no messages.
<0	Execution was unsuccessful. The negative value corresponds to an error message in set 10 of the APPC message catalog, `CATAPPC.APPC.SYS`. APPC subsystem error messages are followed by (APPCERR *xxxx*) is the message number.

>0	Execution was successful, but a warning condition was encountered. The positive value corresponds to a warning message in set 11 of the APPC message catalog, CATAPPC.APPC.SYS. APPC subsystem warning messages are followed by (APPCWARN *xxxx*), where *xxxx* is the message number.

The subsystem field (bits 16–31) may contain the following values:

0	The intrinsic executed successfully and there are no messages.
732	The message in the information field was generated by the APPC subsystem.

Appendix A , “Messages,” contains a list of all return codes returned by the APPC subsystem, their causes, and the actions you should take to resolve any problems.

APPCSessions

Controls the number of active sessions of a specified APPC session type.

Syntax

```
                CA          I16V        I32
APPCSessions (SessionType, NewLimit, ReturnCode)
```

Parameters

SessionType **Required.** Character array; input. The name of the session type whose session limit you want to change. This must be a session type name configured in the APPC subsystem configuration file. *SessionType* is an 8-character ASCII array, left justified and padded with blanks.

NewLimit **Required.** 16-bit signed integer; input. The new session limit for the active session type you specified in the *SessionType* parameter.

The following restrictions apply to the new session limit:

1. The new session limit for the specified session type may not exceed the Maximum Number of Sessions value configured for the session type. To find out the Maximum Number of Sessions value for a session type, issue the APPCCONTROL STATUS command, described in Chapter 2 , “Interactive Control Operator Commands.” See Chapter 4 , “APPC Subsystem Configuration,” for information on session type configuration.
2. The total number of simultaneously active sessions, for all LU 6.2 products using the APPC subsystem, may not exceed 256.

ReturnCode **Required.** 32-bit signed integer; output. Indicates the result of the intrinsic execution. The *ReturnCode* consists of two 16-bit fields: an information field, containing the number of an APPC warning or error message, and a subsystem field, containing the number of the subsystem that generated the message. If both fields contain zeros, the intrinsic executed successfully, and there are no messages. For more information, see “Return Codes” at the beginning of this chapter. Appendix A , “Messages,” contains a list of all return codes returned by the APPC subsystem, their causes,

and the actions you should take to resolve any problems.

Description

The APPCSessions intrinsic changes the number of active sessions for a specified session type. The new session limit for the session type is in effect only while the APPC subsystem is currently active; the APPCSessions intrinsic does not update the APPC configuration file.

When you call the APPCSessions intrinsic to raise the session limits for a session type, sessions will be activated until the new session limit is reached.

Using the APPCSessions intrinsic to reduce the session limit for a session type causes an orderly termination of the sessions affected; termination will not occur until conversations on the affected sessions have ended. All current conversations for the affected sessions are allowed to complete, but no new conversations are allocated for those sessions.

NOTE

If you are changing the session limit for an independent LU running parallel sessions, the new limit will be negotiated with the partner LU on the remote system. Therefore, the actual new limit might not agree with the limit you requested in your APPCSessions intrinsic call.

To find out the actual, negotiated session limit, issue the APPCCONTROL STATUS command, described in Chapter 2 , “Interactive Control Operator Commands.”

For dependent LUs and independent LUs running single sessions, the new value is not negotiated.

The APPCSessions intrinsic performs the same function as the control operator command APPCCONTROL SESSIONS.

Return Codes

```
    0 = Successful completion
-1001 = Missing required parameter
-1002 = Parameter error
-1003 = Parameter string length too long
-1004 = Too many parameters specified
-1005 = Syntax error
-1006 = Redundant parameter
-1007 = Missing parameter after equal sign
-1008 = Missing or invalid delimiter
-1009 = Parameter out of bounds
-1010 = User missing NM capability
-1015 = Invalid SessionTypeName parm value
-1016 = Invalid SessionLimit parm value
-1063 = Resource Manager error - CHANGE_SESSIONS_FAILED
-1065 = Control Operator internal error
-1072 = APPC subsystem not active
-1075 = APPC subsystem shutting down
-1201 = Session limit exceeded
```

APPCStart

Activates the APPC subsystem and all APPC sessions configured for automatic activation.

Syntax

```
                 I32         I16V         CA           I16V
APPCStart (ReturnCode [,TraceOn] [,TFileName] [,TFileSize]

                   I16V           I16V
             [,PerformanceOn] [,Erropt])
```

Parameters

ReturnCode

Required. 32-bit signed integer; output. Indicates the result of the intrinsic execution. The *ReturnCode* consists of two 16-bit fields: an information field, containing the number of an APPC warning or error message, and a subsystem field, containing the number of the subsystem that generated the message. If both fields contain zeros, the intrinsic executed successfully, and there are no messages. For more information, see "Return Codes" at the beginning of this chapter. Appendix A , "Messages," contains a list of all return codes returned by the APPC subsystem, their causes, and the actions you should take to resolve any problems.

TraceOn

Optional. 16-bit signed integer; input. Specifies whether or not internal state tracing will be performed. Internal tracing records significant state changes, events and data to a disk file. Possible values are as follows:

1	Internal state tracing is enabled. Tracing will remain on until the APPCCONTROL TRACEOFF command is issued or until the APPC subsystem is stopped.
0	Internal state tracing is disabled. Tracing will remain off until the APPCCONTROL TRACEON command is issued.

Default: 0 (tracing disabled)

TFileName

Optional. 8-character ASCII array, left justified and padded with blanks. This parameter specifies the name of the trace file, which will be created in the logon group and account of the user who ran the calling program. A valid name is up to 8 alphanumeric characters long and begins with a letter.

The default trace file name is NMTC*xxxx*.PUB.SYS, where *xxxx* is a number from 0000 through 9999. This number is incremented whenever a new trace file is created.

TFileSize

Optional. 16-bit signed integer; input. A decimal number, from 0 through 32767, specifying the size of the trace file, in 128-word records. When *TFileSize* has been reached, trace records are overwritten, starting at the beginning of the file.

Default: 4096

PerformanceOn

Optional. 16-bit signed integer; input. This parameter causes the APPC subsystem to gather performance statistics during run time. These statistics are displayed when you issue the APPCCONTROL PERFORMANCE command.

Once the APPC subsystem is active, statistics gathering can be enabled and disabled with the APPCCONTROL PERFORMANCEON and APPCCONTROL PERFORMANCEOFF commands, described in Chapter 2 , "Interactive Control Operator Commands."

Possible values for the *PerformanceOn* parameter are as follows:

1	Performance statistics gathering is enabled.
0	Performance statistics gathering is disabled.

Default: 0 (Performance statistics gathering disabled)

Erropt

Optional. 16-bit signed integer; input. This parameter specifies the action taken when an irrecoverable internal error occurs. Possible values are as follows:

1	System shutdown. The system is brought down when an irrecoverable error occurs.
2	Subsystem shutdown. Only the APPC subsystem is brought down when an irrecoverable error occurs.

Default: 2 (Subsystem shutdown)

Description

The APPCStart intrinsic activates the APPC subsystem and all APPC sessions configured for automatic activation. (See Chapter 4 , “APPC Subsystem Configuration,” for more information on configuring the APPC subsystem.)

NOTE The APPCStart intrinsic will not cause the activation of the SNA link product if it is not already active. The SNA link product must be explicitly activated using the SNACONTROL START command.

The APPCStart intrinsic performs the same function as the control operator command APPCCONTROL START.

Return Codes

```
    0 = Successful completion
-1002 = Parameter error
-1004 = Too many parameters specified
-1005 = Syntax error
-1006 = Redundant parameter
-1007 = Missing parameter after equal sign
-1008 = Missing or invalid delimiter
-1009 = Parameter out of bounds
-1010 = User missing NM capability
-1011 = Invalid value for Erropt parameter
-1012 = Invalid value for TraceOn parameter
-1013 = Invalid value for TFileSize parameter
-1014 = Invalid value for PerformanceOn parameter
-1030 = APPC module versions mismatch
-1031 = Missing required module
-1032 = Invalid module ID in version check
-1033 = Invalid remotely initiated transaction program name
-1034 = Invalid APPC configuration
-1035 = Unable to open NMCONFIG file
-1036 = Unable to close NMCONFIG file
-1037 = Configuration error
-1038 = Unable to open CATAPPC.APPC.SYS
-1039 = Unable to close CATAPPC.APPC.SYS
-1040 = Unable to open NM log file
-1041 = Unable to close NM log file
-1042 = Unable to open internal trace file
-1043 = Unable to close internal trace file
-1046 = Unable to create Resource Manager at subsystem
start-up
-1047 = Unable to create control operator TP at subsystem
start-up
-1048 = Unable to create CNOS TP at subsystem start-up
-1049 = Resource Manager detected error during subsystem
start-up
-1050 = COPR TP detected error during subsystem start-up
-1051 = CNOS TP detected error during subsystem start-up
-1063 = Resource Manager error - CHANGE_SESSIONS_FAILED
-1065 = Control Operator internal error
-1072 = APPC subsystem not active
-1073 = APPC subsystem pending active
-1074 = APPC subsystem already active
-1075 = APPC subsystem shutting down
-1201 = Session limit exceeded
-1202 = Catread failed
```

APPCStatus

Indicates whether the APPC subsystem is active.

Syntax

```
                    I16             I32
APPCStatus  (SubsystemState, ReturnCode)
```

Parameters

SubsystemState

Required. 16-bit integer; output. Indicates the current state of the APPC subsystem. Possible values are as follows:

1	The APPC subsystem is inactive.
2	APPC subsystem startup is pending.
3	The APPC subsystem is active.
4	The APPC subsystem is shutting down.

ReturnCode **Required.** 32-bit signed integer; output. Indicates the result of the intrinsic execution. The *ReturnCode* consists of two 16-bit fields: an information field, containing the number of an APPC warning or error message, and a subsystem field, containing the number of the subsystem that generated the message. If both fields contain zeros, the intrinsic executed successfully, and there are no messages. For more information, see "Return Codes" at the beginning of this chapter. Appendix A , "Messages," contains a list of all return codes returned by the APPC subsystem, their causes, and the actions you should take to resolve any problems.

Description

The APPCStatus intrinsic performs a subset of the functions that the APPCCONTROL STATUS command performs. It returns a *SubsystemState* parameter, which indicates whether the APPC subsystem is inactive, is in the process of starting up, is active, or is in the process of shutting down.

Return Codes

```
    0 = Successful completion
-1002 = Parameter error
-1004 = Too many parameters specified
-1005 = Syntax error
-1006 = Redundant parameter
-1007 = Missing parameter after equal sign
-1008 = Missing or invalid delimiter
-1009 = Parameter out of bounds
-1010 = User missing NM capability
-1065 = Control Operator internal error
-1072 = APPC subsystem not active
```

APPCStop

Deactivates the APPC subsystem and all active APPC sessions.

Syntax

```
                I32          I16V
APPCStop (ReturnCode [,StopType])
```

Parameters

ReturnCode **Required.** 32-bit signed integer; output. Indicates the result of the intrinsic execution. The *ReturnCode* consists of two 16-bit fields: an information field, containing the number of an APPC warning or error message, and a subsystem field, containing the number of the subsystem that generated the message. If both fields contain zeros, the intrinsic executed successfully, and there are no messages. For more information, see "Return Codes" at the beginning of this chapter. Appendix A , "Messages," contains a list of all return codes returned by the APPC subsystem, their causes, and the actions you should take to resolve any problems.

StopType **Optional.** 16-bit signed integer; input. Specifies the type of subsystem shutdown (Kill, Protocol, or Quiesce). Allowable values are as follows:

1	**Kill.** All active sessions will be terminated immediately. Conversations using the sessions will not be allowed to complete. TPs will not be terminated, so users must terminate their TP processes when the subsystem goes down.
2	**Protocol.** All active sessions will be terminated, pending notification of any transaction programs currently in conversation. Once the active TPs have been notified, their sessions will be terminated. TPs will not be terminated.
3	**Quiesce.** Orderly shutdown of the APPC subsystem, allowing all current conversations to be completed but preventing any new conversations from being allocated. No APPC session will

be brought down until the conversation using it has been deallocated. When the conversation has completed, the session will be terminated. TPs will not be terminated.

Default: 3 (Quiesce)

Description

The APPCStop intrinsic is used to shut down the APPC subsystem by deactivating all active APPC sessions. If possible, you should specify a *StopType* of 3 (Quiesce). If you cannot wait for all conversations to end before shutting down the subsystem, specify a *StopType* of 2 (Protocol), to ensure that resources are returned to the system before shutdown. If a protocol shutdown does not work, specify a *StopType* of 1 (Kill).

NOTE APPCStop does not terminate any active transaction programs. Users must terminate their TP processes when the subsystem goes down.

The APPCStop intrinsic performs the same function as the control operator command APPCCONTROL STOP.

Return Codes

```
    0 = Successful completion
-1002 = Parameter error
-1004 = Too many parameters specified
-1005 = Syntax error
-1006 = Redundant parameter
-1007 = Missing parameter after equal sign
-1008 = Missing or invalid delimiter
-1009 = Parameter out of bounds
-1010 = User missing NM capability
-1017 = Invalid StopType parm value
-1064 = Resource Manager error - subsystem shutdown failed
-1065 = Control Operator internal error
-1072 = APPC subsystem not active
```

4 APPC Subsystem Configuration

This chapter describes how to use the NM configuration manager (NMMGR) to create or modify a configuration for the APPC subsystem.

Appendix C , “Configuration Worksheets,” provides configuration worksheets to help you plan APPC subsystem configuration. Step-by-step instructions for completing the worksheets are included in Appendix C , “Configuration Worksheets.”

NOTE

Before you can configure the APPC subsystem, you must have successfully configured SNA/SDLC Link/XL or SNA/X.25 Link/XL (including logging). Configuration of the SNA link products is described in the *SNA Link/XL Node Manager's Guide*.

You must also coordinate the APPC subsystem configuration with the remote system configuration. The *HP SNA Products: ACF/VTAM and ACF/NCP Guide*, the *HP SNA Products: CICS Guide*, and the *HP SNA Products: AS/400 Guide* contain Hewlett-Packard's recommendations for configuring the remote system.

This chapter contains the following sections:

- Data Required from the Remote Configuration
- NMMGR Configuration Overview
- Configuring the APPC Subsystem
- Planning APPC Configuration
- Configuration Illustrations
- Migrating an APPC Configuration

NOTE

This chapter does not cover logging configuration for the APPC subsystem. Logging configuration is covered in the *SNA Link/XL Node Manager's Guide*.

Data Required From the Remote Configuration

Data in the APPC subsystem configuration must correspond to information from the remote configuration. Before using NMMGR, you should consult with the remote system programmer to ensure that the information in the following tables is consistent between the HP 3000 configuration and the remote system configuration.

If you are configuring the HP 3000 as a Node Type 2.0 that communicates with a Type 5 (host) node, Table 4-1 lists the local and remote configuration items that must agree.

Table 4-1 NCP, VTAN, and CICS Corresponding Values

Remote Configuration Item	Field in NMMGR Screen	NMMGR Screen
puname operand of PU macro	SNA Node Name	APPC: Dependent LU Session Type Data
luname operand of LU macro	Local LU Names	APPC: Dependent LU List Data
APPLID of Label field of the VTAM APPL definition statement	Remote LU Name	APPC: Dependent LU Session Type Data
CICS MODENAME operand of DFHTCT TYPE=SYSTEM macro	Mode Name	APPC: Dependent LU Session Type Data
RUSIZES parameter of MODEENT Logmode Table macro	Maximum RU Size (Send) and Receive)	APPC: Mode Type Data
SRCVPAC parameter of MODEENT Logmode Table macro	Maximum Pacing Value (Send)	APPC: Mode Type Data
SSNDPAC parameter of MODEENT Logmode Table macro	Maximum Pacing Value (Receive)	APPC: Mode Type Data

NOTE Some sites use separate copies of the same application subsystem to accommodate different types of users. For example, you may be given one VTAM Appl ID for CICS to use for development sessions, and another to user for run-time sessions.

You should also obtain the following items from the IBM host programmer:

- The LINEID for each link you have to a remote system.
- The STARTED_TASK_NAME for each application subsystem you will use.

Though these two items have no corollaries in the HP 3000 configuration, you will need them when communicating with people at the remote site.

If you are configuring the HP 3000 as a Node Type 2.1 that communicates with an AS/400 (Type 2.1 node), Table 4-2 lists the local and remote configuration items that must agree.

NOTE When you configure an IBM AS/400 to communicate with the HP 3000, you must configure the HP 3000 controller as an APPN-capable LEN node.

Table 4-2 AS/400 Corresponding Values

Remote Configuration Item	Field in NMMGR Screen	NMMGR Screen
RMTLOCNAME of Device Description	Local LU Name	APPC: Independent LU Session Type Data
RMTNETID of Controller Description (If RMTNETID = *NETATTR, use Local network ID of Network Attributes Table.)	Local Network ID	APPC: Network ID Data
Local network ID of Network Attribute Table, and LCLLOCNAME of Device Description	Fully Qualified Remote LU Name	APPC: Independent LU Session Type Data
MODE of Device Description	Mode Name	APPC: Independent LU Session Type Data
MAXLENRU of Mode Description	Maximum RU Size (Send and Receive)	APPC: Mode Type Data
INPACING of Mode Description	Maximum Pacing Value (Send)	APPC: Mode Type Data
OUTPACING of Mode Description	Maximum Pacing Value (Receive)	APPC: Mode Type Data

Because the APPC subsystem does not support remotely controlled sessions, you must configure the following values on the AS/400:

- For parallel sessions, the Locally controlled sessions value in the Mode Description must be 0.
- For single independent sessions, the Locally controlled session value in the Device Description must be NO.

NMMGR Configuration Overview

Use NMMGR to create or modify an APPC subsystem configuration. The NMMGR utility is described in *Using the Node Management Services Utilities*; if you are not familiar with NMMGR, you should read that manual before continuing.

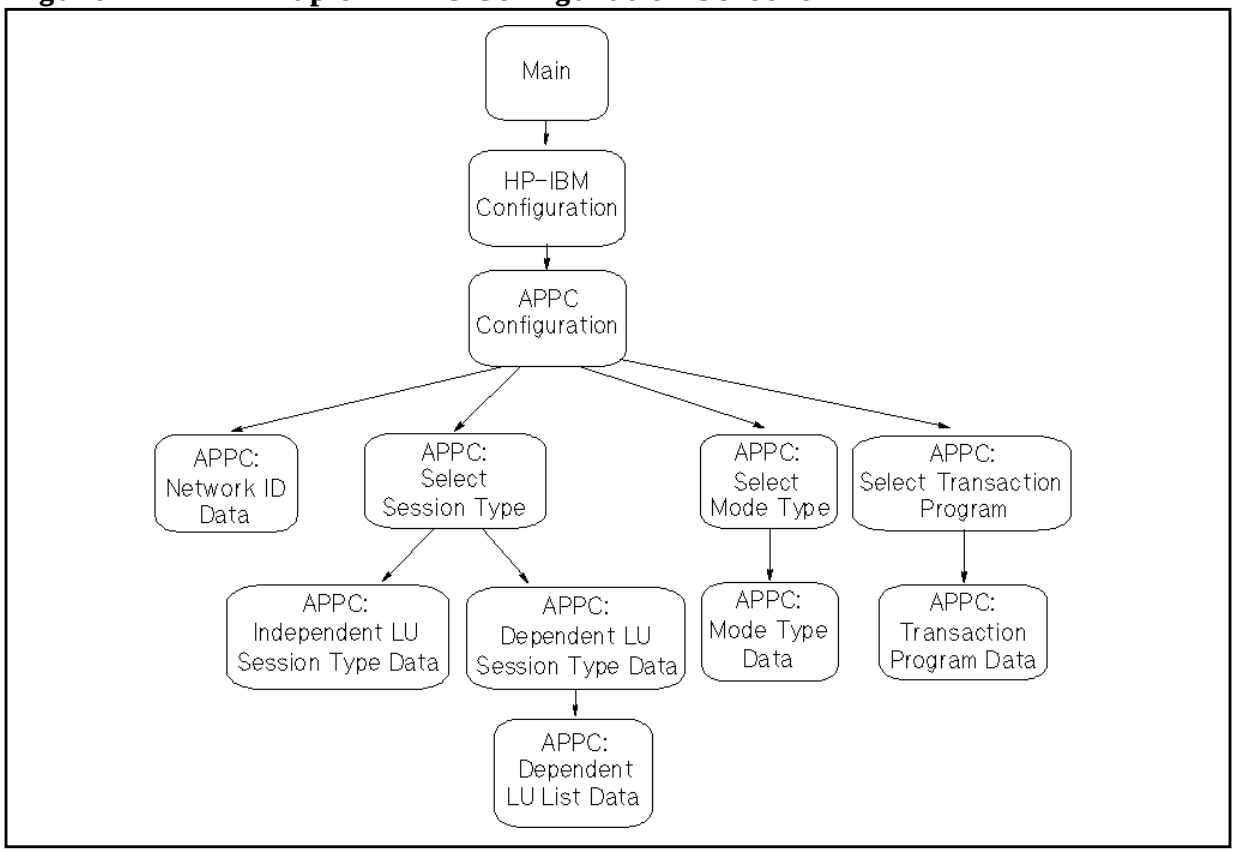
You configure the APPC subsystem by defining the network in which it will operate, the nodes and LUs that will communicate over APPC sessions and the properties of the APPC sessions. Table 4-3 lists the NMMGR screens used to configure the APPC subsystem, and it gives the page number in this chapter where each screen is described.

Table 4-3 NMMGR Screen Index

Main	Figure 4-2 on page 77
HP-IBM Configuration	Figure 4-3 on page 78
APPC Configuration	Figure 4-4 on page 79
APPC: Network ID Data	Figure 4-5 on page 80
APPC: Select Session Type	Figure 4-6 on page 81
APPC: Independent LU Session Type Data	Figure 4-7 on page 83
APPC: Dependent LU Session Type Data	Figure 4-9 on page 88
APPC: Dependent LU List Data	Figure 4-10 on page 91
APPC: Select Mode Type	Figure 4-11 on page 92
APPC: Mode Type Data	Figure 4-12 on page 94
APPC: Select Transaction Program	Figure 4-13 on page 95
APPC: Transaction Program Data	Figure 4-15 on page 98

Figure 4-1 is a map of the NMMGR screens you use to configure the APPC subsystem.

Figure 4-1 Map of APPC Configuration Screens



Configuration File Validation

NMMGR has a utility for validating the configuration file to ensure that the APPC subsystem configuration data is logically and syntactically correct. It also checks the APPC subsystem configuration against the SNA node (SNA/SDLC Link/XL) configuration for correctness and consistency. For example, it checks to make sure an LU name you configure for the APPC subsystem is associated with the same SNA node in both the APPC subsystem configuration and the SNA node configuration.

You should validate the configuration file after completing any of these tasks:

- An initial configuration of one or more SNA nodes.
- An initial configuration of one or more SNA Services.
- A modification of an existing configuration.

The NMMGR validation utility validates the entire HP-IBM portion of the configuration file; see *Using the Node Management Services Utilities* for details. Appendix A , “Messages,” contains a list of the NMMGR validation messages that pertain to APPC subsystem configuration.

Creating a Critical Summary

NMMGR allows you to create a summary of critical APPC subsystem configuration data. The critical summary is a printout of the APPC subsystem data in the configuration file. It can be used to check current contents of the configuration file to determine whether the configuration is correct. The critical summary is generated through NMMGR; see *Using the Node Management Services Utilities*.

Appendix B , “Sample Configuration,” gives an example of critical summary output.

Configuring the APPC Subsystem

This section describes the NMMGR screens used for APPC subsystem configuration.

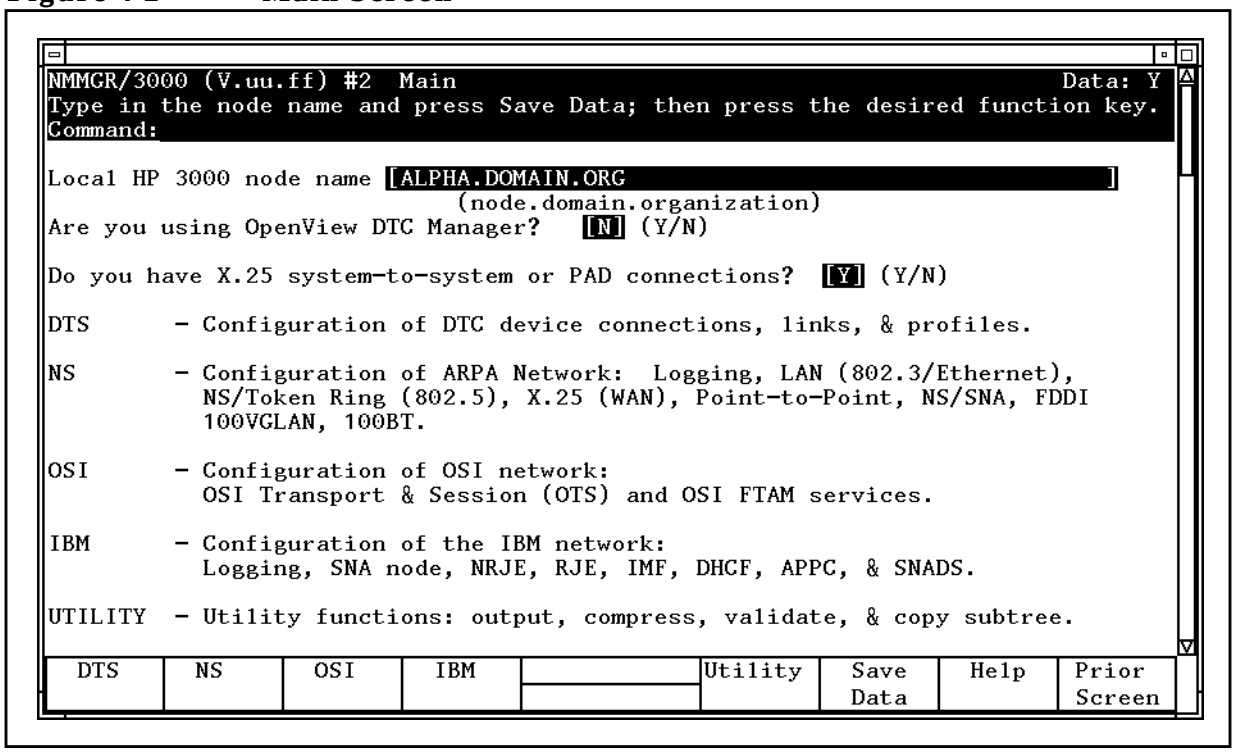
NOTE NMMGR will operate on a file of any legal name. However, the SNA node will not activate unless the configuration file is named NMCONFIG.PUB.SYS.

Main Screen

The NMMGR "Main" screen lets you select the category of network subsystems you want to configure. The "Main" screen is shown in Figure 4-2.

To go to the "HP-IBM Configuration" screen, press **[f4]** (IBM).

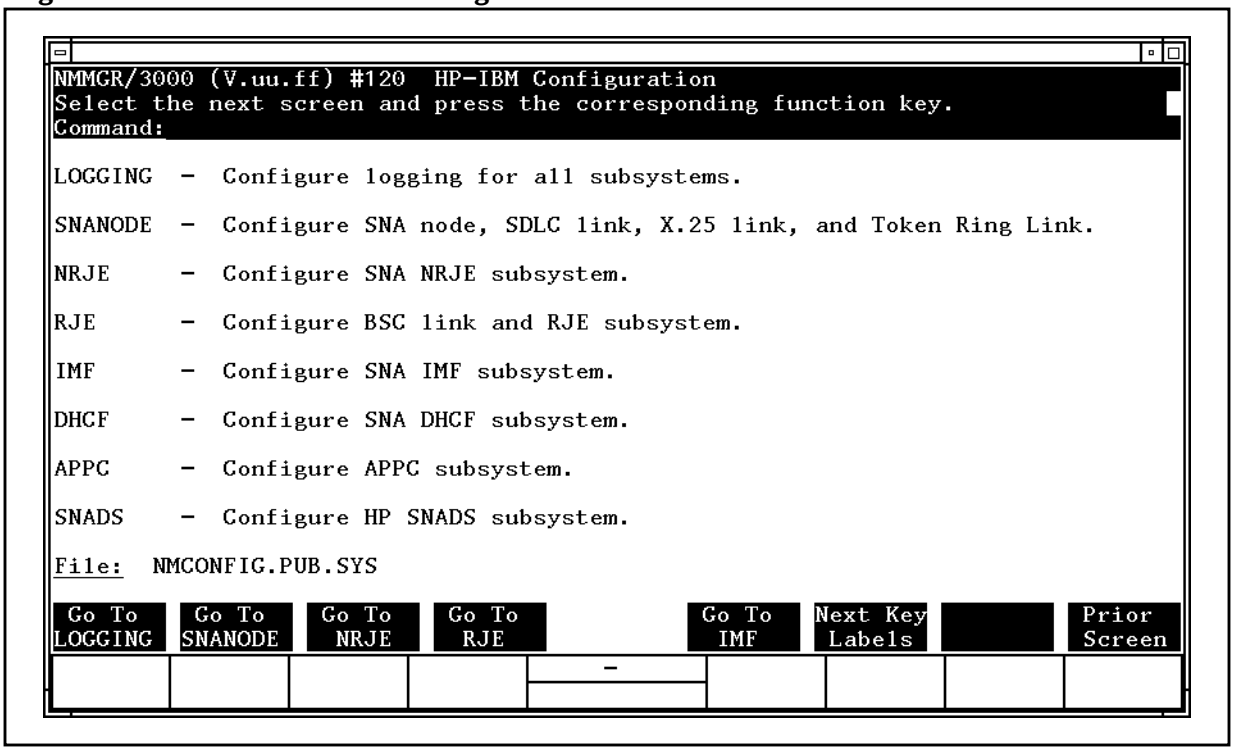
Figure 4-2 Main Screen



HP-IBM Configuration Screen

From the “HP-IBM Configuration” screen, shown in Figure 4-3, you can select the SNA Service or subsystem you want to configure.

Figure 4-3 HP-IBM Configuration Screen

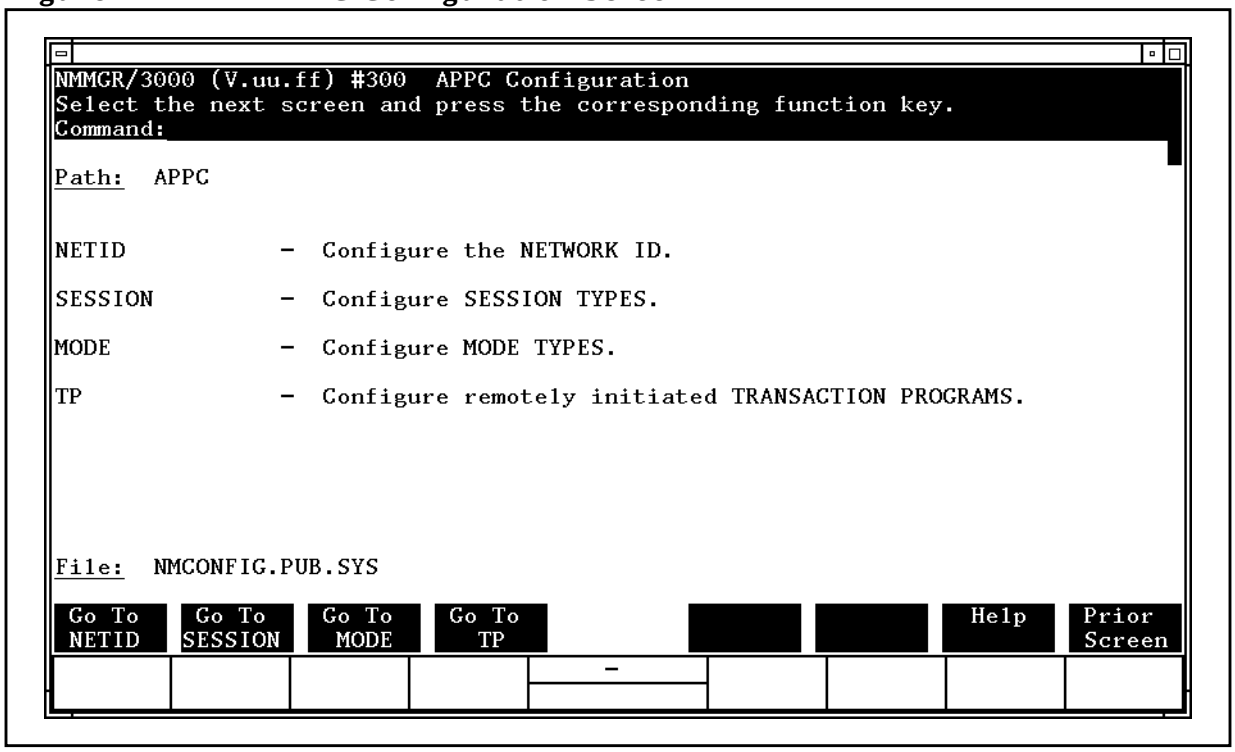


To get to the APPC subsystem configuration screens, first press **[f6]** (`Next Key Labels`), and then press **[f2]** (`Go To APPC`).

APPC Configuration Screen

The “APPC Configuration” screen, shown in Figure 4-4, allows you to choose among the four branches of the APPC subsystem configuration: Network ID, Session Types, Mode Types, and TPs.

Figure 4-4 APPC Configuration Screen



The network ID must be configured only if the HP 3000 will operate as a Node Type 2.1 in a peer-to-peer environment. Press **[f1]** (`Go To NET ID`) to get to the “APPC: Network ID Data” screen.

To configure session types, press **[f2]** (`Go To SESSION`) to get to the “APPC: Select Session Type” screen.

To configure mode types, press **[f3]** (`Go To MODE`) to get to the “APPC: Select Mode Type” screen.

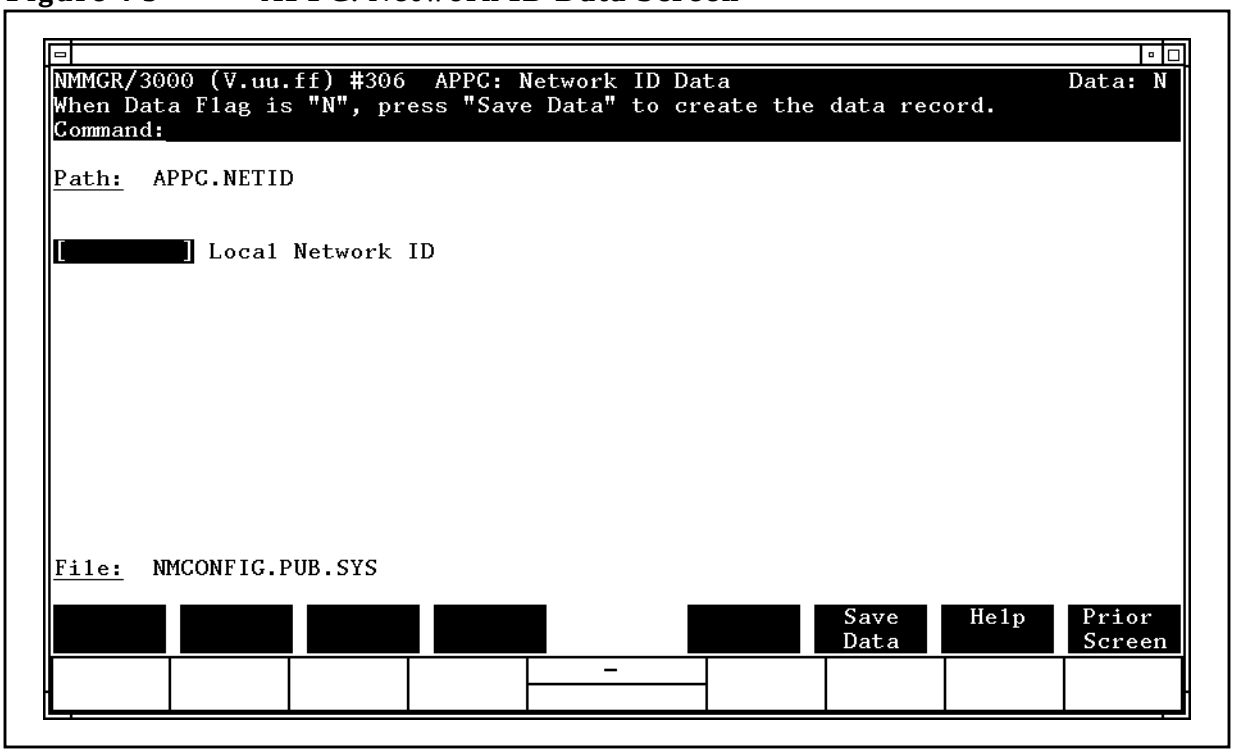
To configure remotely initiated local TPs, press **[f4]** (`Go To TP`) to get to the “APPC: Select Transaction Program” screen.

APPC: Network ID Data Screen

The "APPC: Network ID Data" screen, shown in Figure 4-5, is where you configure the name of the local SNA network. Configure the network ID only if your HP 3000 will operate as a Node Type 2.1 in a peer-to-peer environment.

When you have finished entering the network ID, press **[f6]** (Save Data), and then press **[f8]** (Prior Screen) to return to the "APPC Configuration" screen.

Figure 4-5 APPC: Network ID Data Screen



Fields *Local Network ID*

Required. Enter the network ID of the local SNA network. For AS/400 communication, the Local Network ID should match the RMTNETID in the Controller Description of the AS/400 configuration. If the RMTNETID on the AS/400 is specified as *NETATTR, the Local Network ID in this screen must match the Local network ID in the AS/400 Network Attributes Table.

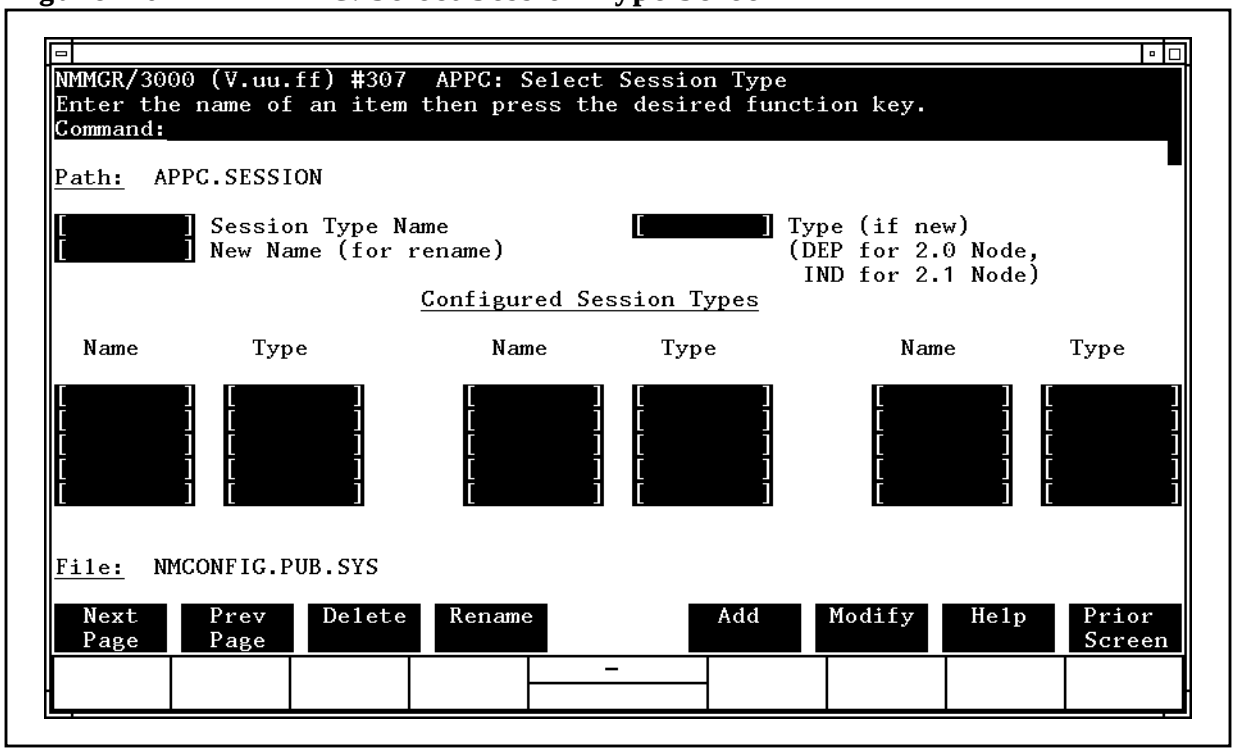
APPC: Select Session Type Screen

In the "APPC: Select Session Type" screen, shown in Figure 4-6, you enter the name of a new session type you want to configure or a previously configured session type you want to modify. A session type is a grouping of a remote LU, one or more local LUs, and a mode name. Every transaction program specifies a session type name to indicate the remote LU with which it wants to communicate and the local LU and mode it wants to use.

Enter the Session Type Name and the Type (IND for independent or DEP for dependent) for the session type you wish to configure. Then, when you press **[f5]** (Add) or **[f6]** (Modify), you will go to the next screen automatically. If the Type is IND, you will go to the "APPC: Independent LU Session Type Data" screen, and if the Type is DEP, you will go to the "APPC: Dependent LU Session Type Data" screen.

The "APPC: Select Session Type" screen lists all the configured session types. Up to 60 session types may be configured for the APPC subsystem. These session types are shared by all LU 6.2 products running on the APPC subsystem. Press **[f1]** (Next Page) to see the session types listed on subsequent pages.

Figure 4-6 APPC: Select Session Type Screen



Fields

Session Type Name

Required. The name of a new session type you want to add or a previously configured session type you want to modify. You can configure 60 session types for the APPC subsystem. A session type name can be up to 8 alphanumeric characters; the first character must be alphabetic. The session type name does not have to match anything in the local or remote configuration.

Type

This field is required if you are adding a new session type. It is not required if you are updating the configuration for an existing session type. Enter `IND` for an independent LU session type or `DEP` for a dependent LU session type. An independent LU is used for peer-to-peer communication with a Type 2.1 node, like an AS/400, and dependent LUs are used for communication with a host node (Type 5).

For information on independent LUs, see the description of the "APPC: Independent LU Session Type Data" screen, later in this chapter. For information on dependent LUs, see the description of the "APPC: Dependent LU Session Type Data" screen, later in this chapter.

If you want to change a configured session type from independent to dependent (or from dependent to independent) you must delete the session type and reenter it.

New Name

The new name of a session type you want to rename. To rename a configured session type, enter its current name in the `Session Type Name` field, enter its new name in the `New Name` field, and press **[f4]** (`Rename`).

APPC: Independent LU Session Type Data Screen

The "APPC: Independent LU Session Type Data" screen, shown in Figure 4-7, is used to configure session types for independent LUs. Enter the data into the "APPC: Independent LU Session Type Data" screen, press **[f6]** (`Save Data`), and then press **[f8]** (`Prior Screen`) to get back to the "APPC: Select Session Type" screen. From there, you can configure another session type, or you can press **[f8]** (`Prior Screen`) to get back to the "APPC Configuration" screen.

An independent LU can communicate with another independent LU on a Type 2.1 node, like an IBM AS/400. The APPC subsystem allows the HP 3000 to act as a Type 2.1 node in a peer-to-peer environment, where two independent LUs can establish an APPC session between themselves without the supervision of the SSCP on a host node.

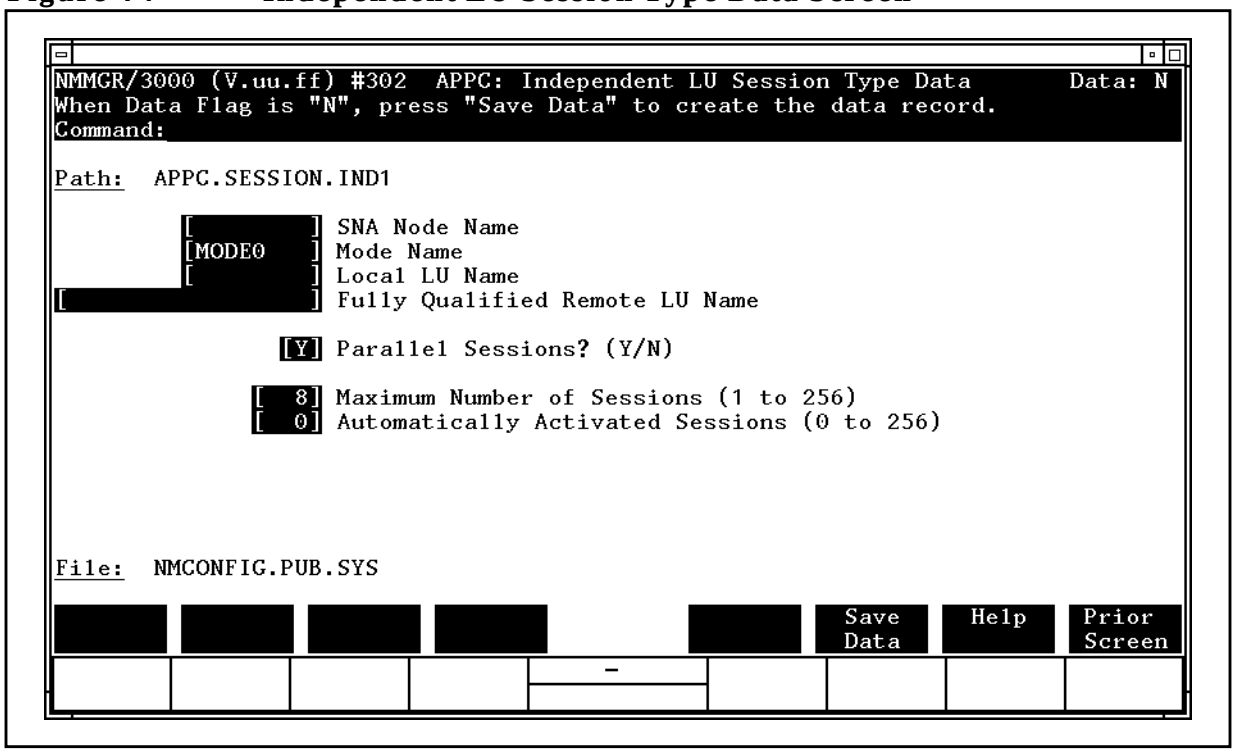
An independent LU can function as either a primary or a secondary LU; that is, it can initiate an APPC session by sending a BIND to the remote LU, or it can receive a BIND from the remote LU.

An independent LU on the HP 3000 can carry on multiple (parallel) sessions with a remote independent LU. To allow parallel sessions with the remote LU, enter Y in the `Parallel Sessions` field of the “APPC: Independent LU Session Type Data” screen.

An independent LU can communicate with several remote LUs at once; however, all the remote LUs with which it communicates must be connected to the HP 3000 through the same local SNA node (the same copy of SNA/SDLC Link/XL). (To configure an independent LU to communicate with multiple remote LUs, configure one session type for each remote LU, and specify the same `Local LU Name` for each session type.

If you want independent LU sessions with remote LUs that are connected to the HP 3000 through different local SNA nodes, you must configure a different local independent LU for each node.

Figure 4-7 Independent LU Session Type Data Screen



Fields

SNA Node Name

Required. The name of the SNA node on which the LU for this session type resides. The SNA node name must also be configured in the SNANODE branch of NMMGR. See the *SNA Link/XL Node Manager's Guide* for information on SNA node configuration. The SNA node name can be up to 8 alphanumeric characters; the first character must be alphabetic.

Every SNA node is a link to a remote system, so by configuring multiple session types with different SNA node names, you can connect the APPC subsystem to multiple remote systems.

Mode Name

Required. The name of a configured mode type. If the HP 3000 will communicate with an AS/400, the mode name must match the MODE in the Device Description for the partner LU on the AS/400. A mode name can be up to 8 alphanumeric characters; the first character must be alphabetic. The Mode Name must also be configured in the "APPC: Select Mode Type" screen. Data for each mode type is configured in the "APPC: Mode Type Data" screen, described later in this chapter. The default mode name is MODE0.

NOTE If you use the default mode name, MODE0, you still must configure a mode named MODE0 on the AS/400.

Local LU Name

Required. The name of the local independent LU that will use the session type. If the HP 3000 will communicate with an AS/400, the Local LU Name must match the RMTLOCNAME in the Device Description for the partner LU on the AS/400.

The local LU must be configured on the SNA node you specified in the SNA Node Name field. SNA nodes and LUs are configured in the SNANODE branch of NMMGR. See the *SNA Link/XL Node Manager's Guide* for more information. A local LU name can be up to 8 alphanumeric characters; the first character must be alphabetic.

Only one local independent LU is needed for each independent LU session type, because an independent LU can conduct multiple, simultaneous sessions with the remote LU.

Fully Qualified Remote LU Name

Required. The network ID and LU name of the remote LU with which the local LU will communicate. A fully qualified LU name is of the form *NetworkID.LUName*, where *NetworkID* and *LUName* are strings of up to 8 alphanumeric characters, each beginning with a letter.

If the HP 3000 will communicate with an AS/400, the *NetworkID* part must match the Local network ID configured in the Network Attributes Table on the AS/400, and the *LUName* part must match the LCLLOCNAME in the Device Description for the partner LU on the AS/400.

Parallel Sessions

Required. Enables or disables parallel sessions between the specified local and remote LUs. Enter Y or N. The partner LU on the remote system must be similarly configured for parallel or single sessions.

Y	The local LU can participate in multiple, simultaneous sessions with the remote LU. If you specify Y, the Maximum Number of Sessions field will be set automatically to 8. You can change the default simply by typing a different value into the field. If the HP 3000 will communicate with an AS/400, the SNGSSN value configured in the Device Description on the AS/400 must be *NO.
N	The local LU may participate in only one session at a time with the remote LU. If you specify N, the Maximum Number of Sessions field will be set automatically to 1. You can change the default simply by typing a different value into the field. If the HP 3000 will communicate with an AS/400, the SNGSSN value configured in the Device Description on the AS/400 must be *YES.

Default: Y (parallel sessions enabled)

Maximum Number of Sessions

Required. The maximum number of APPC sessions that may be simultaneously active between the specified local and remote LUs.

If parallel sessions are disabled (Parallel Sessions = N), the Maximum Number of Sessions value must be 1.

If parallel sessions are enabled (Parallel Sessions = Y), the following restrictions apply to the Maximum Number of Sessions value:

1. The maximum number of simultaneously active sessions for an independent session type is 256.
2. The maximum number of simultaneously active sessions for the whole APPC subsystem is 256.
3. Maximum Number of Sessions must be greater than or equal to the Automatically Activated Sessions value

If parallel sessions are enabled, the Maximum Number of Sessions value might be negotiated at session activation time.

Default: 8

Automatically Activated Sessions

Required. The number of sessions that will be activated automatically when the APPCCONTROL START command is issued or the APPCStart intrinsic is called. The Automatically Activated Sessions value must be less than or equal to the value in the Maximum Number of Sessions field. The sum of the Automatically Activated Sessions values for all the session types on the APPC subsystem must not exceed 256.

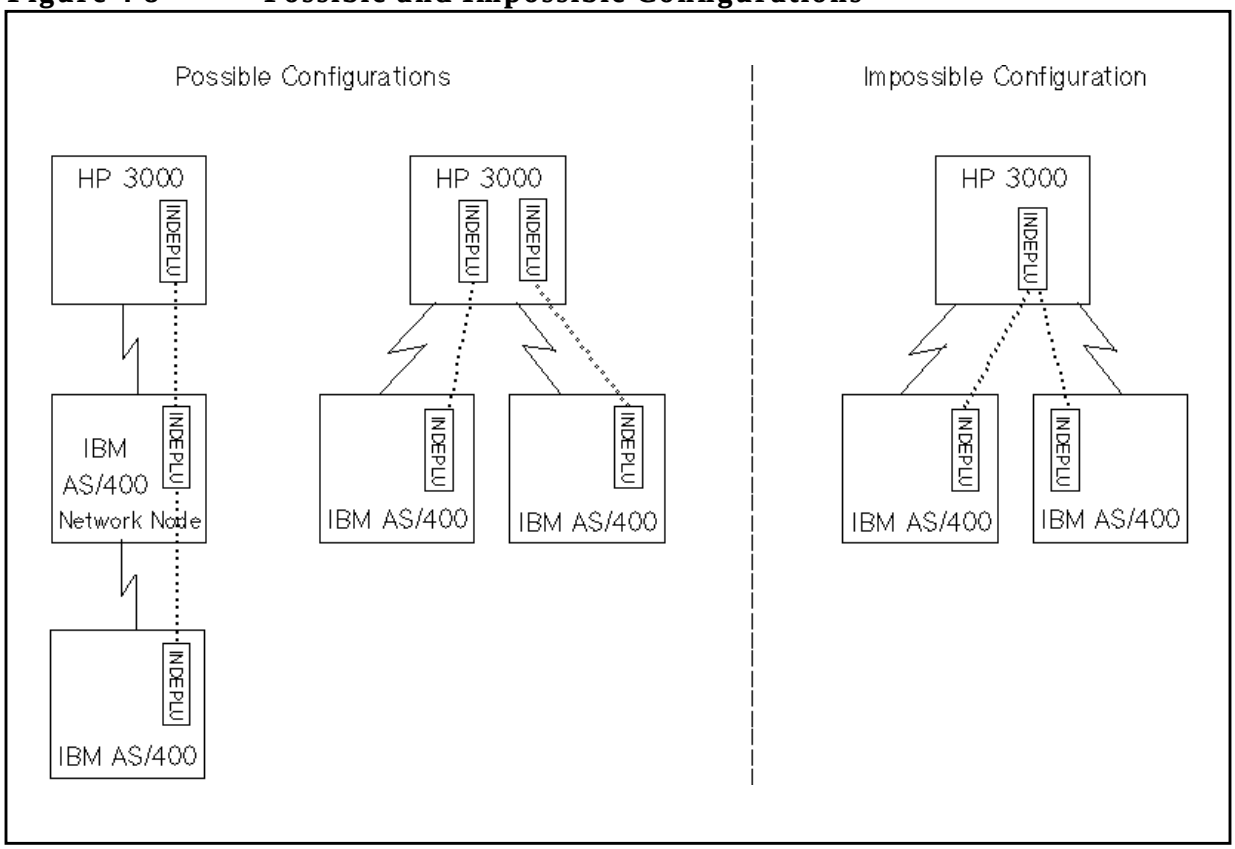
Default: 0

Figure 4-8 shows possible and impossible configurations for independent LUs on the HP 3000. The dotted lines in Figure 4-8 represent APPC sessions.

Possible configurations: An independent LU on the HP 3000 can communicate with multiple remote independent LUs, as long as all the routes to the remote LUs pass through the same PSI card (the same copy of SNA/SDLC Link/XL) on the HP 3000. The remote LUs can be on the same node, or they can be on nodes connected to a Network Node that is directly connected to the HP 3000.

Impossible configurations: An independent LU on the HP 3000 cannot communicate with multiple remote independent LUs if they are connected to the HP 3000 through different SNA nodes (different PSI cards). To communicate with multiple remote LUs that are connected to the HP 3000 through different PSI cards, you must configure different independent LUs on the HP 3000.

Figure 4-8 Possible and Impossible Configurations

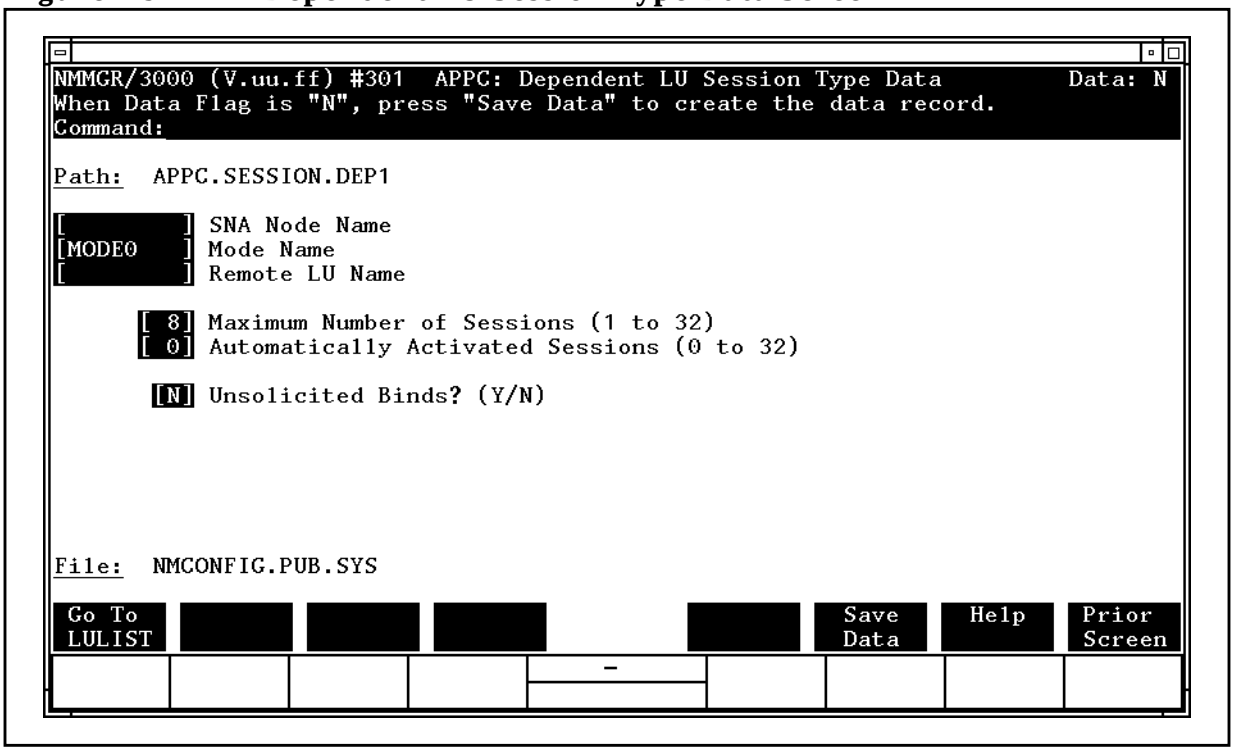


APPC: Dependent LU Session Type Data Screen

The "APPC: Dependent LU Session Type Data" screen, shown in Figure 4-9, is used to configure session types for dependent LUs. Dependent LUs can communicate only with other dependent LUs on a host (Type 5) node. They always function as secondary LUs; that is, they cannot initiate a session by sending a BIND. When a dependent LU wants a session, it must send an INIT_SELF request to solicit a BIND from the remote LU, or it must be configured to accept unsolicited BINDs.

A dependent LU can participate in only one APPC session at a time, so you must configure at least as many dependent LUs as you will have active dependent LU sessions. After you enter the data into the "APPC: Dependent LU Session Type Data" screen, press **[f6]** (`Save Data`), and then press **[f1]** (`Go To LULIST`) to get to the "APPC: Dependent LU List Data" screen.

Figure 4-9 Dependent LU Session Type Data Screen



Fields

SNA Node Name

Required. The name of the SNA node on which the LUs for this session type reside. The SNA node name must also be configured in the SNANODE branch of NMMGR. See the *SNA Link/XL Node Manager's Guide* for information on SNA node configuration. The SNA node name can be up to 8 alphanumeric characters; the first character must be alphabetic. For troubleshooting purposes, Hewlett-Packard recommends that the SNA Node Name match the *puname* operand of the PU macro on the IBM host.

Every SNA node is a link to a remote system, so by configuring multiple session types with different SNA node names, you can connect the APPC subsystem to multiple remote systems.

Mode Name

Required. The name of a configured mode type. The mode name must match the CICS MODENAME operand of the DFHTCT TYPE=SYSTEM macro. A mode name can be up to 8 alphanumeric characters; the first character must be alphabetic. The Mode Name must also be configured in the "APPC: Select Mode Type" screen. Data for each mode type is configured in the

"APPC: Mode Type Data" screen, described later in this chapter. The default mode name is MODE0.

If you use the default mode name, MODE0, dependent LU sessions of this type will use the default mode name configured for the partner LU on the IBM mainframe. The mode name configured on the IBM mainframe does not have to be MODE0. (For independent LU sessions, the mode name configured on the remote system must match the mode name configured on the HP 3000, even if MODE0 is used.)

Remote LU Name

Required. The name of the remote LU with which the dependent LUs of this session type will communicate. The Remote LU Name must match the APPLID in the Label field of the VTAM APPL definition statement. A remote LU name can be up to 8 alphanumeric characters; the first character must be alphabetic.

Only one remote LU name is configured for each session type. The local LUs for the session type are configured in the "APPC: Dependent LU List Data" screen.

Maximum Number of Sessions

Required. The maximum number of APPC sessions that may be simultaneously active between the dependent LUs of this session type and the specified remote LU. This value is not negotiated with the remote LU.

The following restrictions apply to the Maximum Number of Sessions value:

1. The maximum number of simultaneously active sessions for a dependent LU session type is 32. You must have a dependent LU configured for every active session; therefore, the Maximum Number of Sessions value may not exceed the number of dependent LUs listed in the "APPC: Dependent LU List Data" screen.
2. The maximum number of simultaneously active sessions for the whole APPC subsystem is 256.
3. Maximum Number of Sessions must be greater than or equal to the Automatically Activated Sessions value.

Default: 8

Automatically Activated Sessions

Required. The number of sessions that will be activated automatically when the APPCCONTROL START command is issued or the APPCStart intrinsic is called. The Automatically Activated Sessions value must be less than or equal to the value in the Maximum Number of Sessions field. The sum of the Automatically Activated Sessions for all the session types on the APPC subsystem must not exceed 256.

Default: 0

Unsolicited Binds

Required. Enables or disables unsolicited BINDs from the remote LU. Enter Y or N. Hewlett-Packard recommends that you disable unsolicited BINDs.

Y	The remote LU will initiate all APPC sessions by sending unsolicited BINDs to the local LU. The HP 3000 will wait for a BIND rather than sending an INIT_SELF request to solicit a BIND. The IBM host must be specifically configured to send unsolicited BINDs.
N	The local LU will initiate all APPC sessions by sending INIT_SELF requests to the host. The host will send a BIND only in response to an INIT_SELF request.

Default: N (unsolicited BINDs disabled)

APPC: Dependent LU List Data Screen

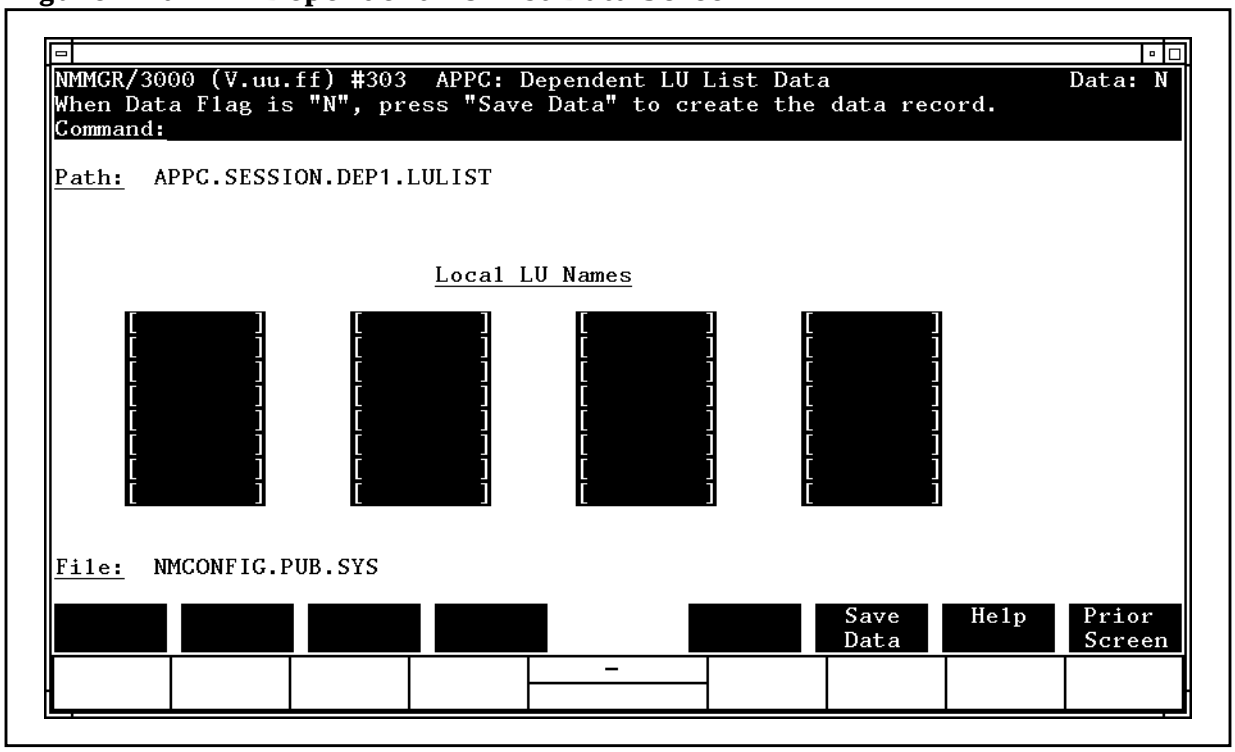
The "APPC: Dependent LU List Data" screen, shown in Figure 4-10, is where you list the dependent LUs that are associated with a dependent LU session type. You get to it from the "APPC: Dependent LU Session Type Data" screen by pressing **[f1]** (Go To LULIST).

The Path field of the "APPC: Dependent LU List Data" screen gives the name of the dependent LU session type to which the LUs in the list belong.

You can configure a maximum of 32 LUs for a dependent LU session type. The number of LUs you configure must be greater than or equal to the Maximum Number of Sessions value in the "APPC: Dependent LU Session Type Data" screen.

After you enter the LU names into the "APPC: Dependent LU List Data" screen, press **[f6]** (Save Data), and then press **[f8]** (Prior Screen) twice to get back to the "APPC: Select Session Type" screen. From there, you can configure another session type, or you can press **[f8]** (Prior Screen) again to get back to the "APPC Configuration" screen.

Figure 4-10 Dependent LU List Data Screen



Fields *Local LU Names*

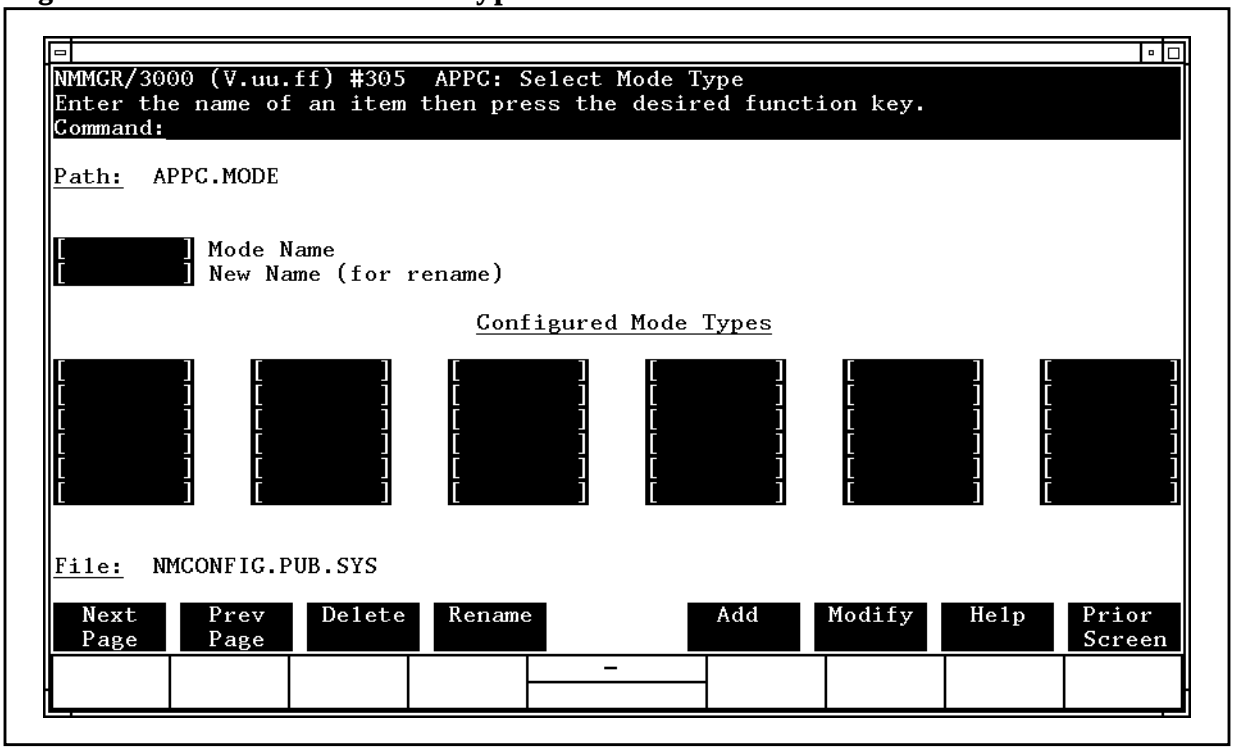
The LUs you enter must be configured in the SNANODE branch of NMMGR. The SNA node with which they are associated in the SNANODE configuration must be the same one specified in the SNA Node Name field of the "APPC: Dependent LU Session Type Data" screen. See the *SNA Link/XL Node Manager's Guide* for information on SNA node and LU configuration. A local LU name can be up to 8 alphanumeric characters; the first character must be alphabetic. For troubleshooting purposes, Hewlett-Packard recommends that the Local LU Names match the values in the *luname* operand of the LU macro on the IBM host.

APPC: Select Mode Type Screen

You configure mode types through the "APPC: Select Mode Type" screen, shown in Figure 4-11, and the "APPC: Mode Type Data" screen, shown in Figure 4-12. A mode type determines the following four characteristics of a session type:

1. The maximum size of RUs sent by the local LU.
2. The maximum size of RUs received by the local LU.
3. The maximum number of RUs that the local LU can send before it receives a response from the remote LU.
4. The maximum number of RUs that the local LU can receive before it must send a response to the remote LU.

Figure 4-11 Select Mode Type Screen



To configure a mode type, enter the name of the mode type you wish to add or modify in the Mode Name field, and press **[f5]** (Add) or **[f6]** (Modify). Pressing **[f5]** or **[f6]** takes you automatically to the "APPC: Mode Type Data" screen.

Fields

Mode Name **Required.** The name of a new mode you want to configure or an existing mode whose configuration you want to modify. A mode name can be up to 8 alphanumeric characters; the first character must be alphabetic. Up to 36 mode types may be configured.

If the HP 3000 will communicate with CICS on an IBM host, the Mode Name must match the CICS MODENAME operand of the DFHTCT TYPE=SYSTEM macro. If the HP 3000 will communicate with an IBM AS/400, the Mode Name must match the MODE in the Device Description for the partner LU on the AS/400.

The default mode name MODE0 is defined internally in the APPC subsystem. Its defined RU size, for sending and receiving, is 256. The size of its pacing window, for sending and receiving, is 7. See "APPC: Mode Type Data Screen," later in this chapter, for more information.

NOTE If you use MODE0 for an independent LU session type that communicates with an IBM AS/400, you must configure a mode called MODE0 on the AS/400.

If you use MODE0 for a dependent LU session type that communicates with an IBM mainframe, the sessions of that type actually use the default mode name configured for the partner LU on the IBM mainframe. Therefore, the mode name on the IBM mainframe does not have to be MODE0.

NOTE The names MODE0 and SNASVCMG are reserved names. You cannot configure a mode with either of these names. (SNASVCMG is a mode used internally by APPC to control parallel sessions.)

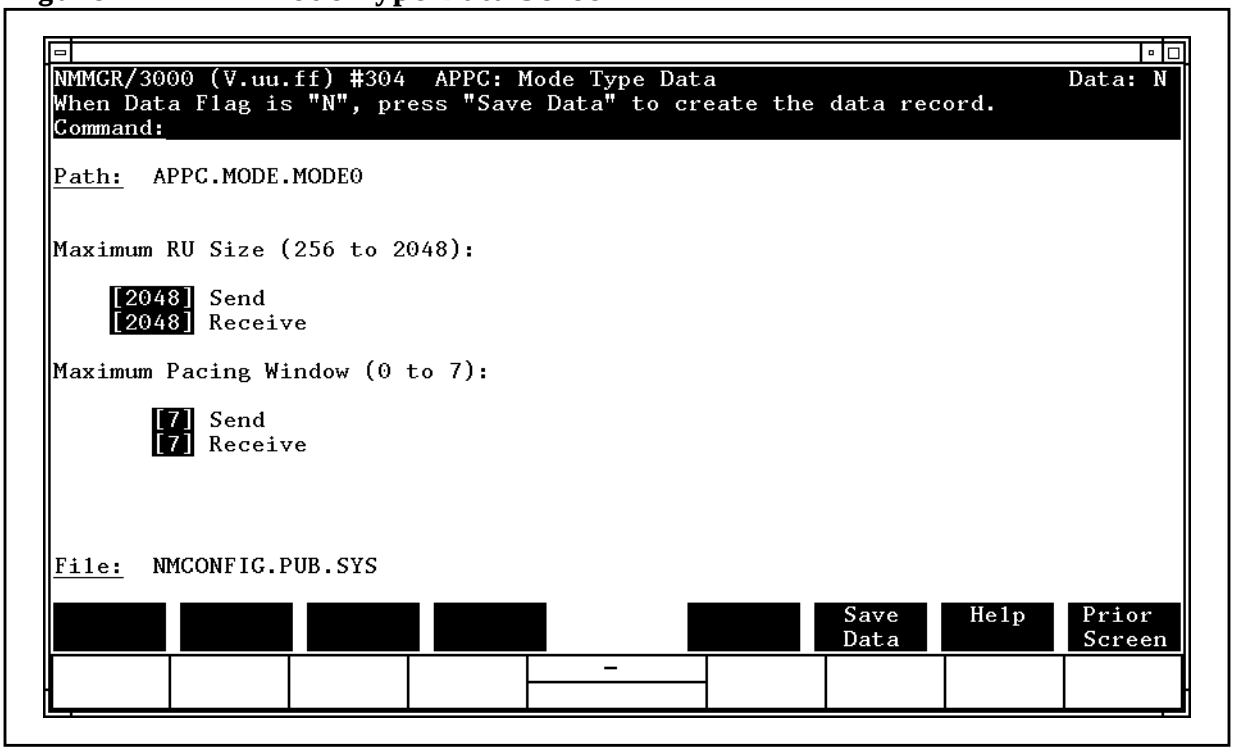
New Name The new name you want to assign to an existing mode type. To rename a mode type, enter its old name in the Mode Name field, enter its new name in the New Name field, and press **[f4]** (Rename).

APPC: Mode Type Data Screen

In the "APPC: Mode Type Data" screen, shown in figure Figure 4-12, you configure the maximum send and receive RU sizes and the sizes of the send and receive pacing windows. These values must be configured for every mode type named in the "APPC: Select Mode Type" screen.

Enter the data into the "APPC: Mode Type Data" screen, and then press **[f6]** (Save Data). To get back to the "APPC Configuration" screen, press **[f8]** (Prior Screen) twice.

Figure 4-12 Mode Type Data Screen



Fields

Maximum RU Size (Send)

Required. The size, in bytes, of the largest RU that can be sent by the local LU over a session with this mode type. For IBM mainframe communication, this must match the receive part of the *RUSIZES* parameter of the MODEENT macro in the Logmode Table. For AS/400 communication, this must match the MAXLENRU value in the Mode Description on the AS/400.

Default: 2048

Maximum RU Size (Receive)

Required. The size, in bytes, of the largest RU that can be received by the local LU over a session with this mode type. For IBM mainframe communication, this must match the send part of the *RUSIZES* parameter of the MODEENT macro in the Logmode Table. For AS/400 communication, this must match the MAXLENRU value in the Mode Description on the AS/400.

Default: 2048

Maximum Pacing Window (Send)

Required. The maximum number of RUs the local LU can send before it must receive a response from the remote LU. For IBM mainframe communication, this

must match the *SRCVPAC* parameter of the MODEENT macro in the Logmode Table. For AS/400 communication, this must match the INPACING value in the Mode Description on the AS/400.

Default: 7

Maximum Pacing Window (Receive)

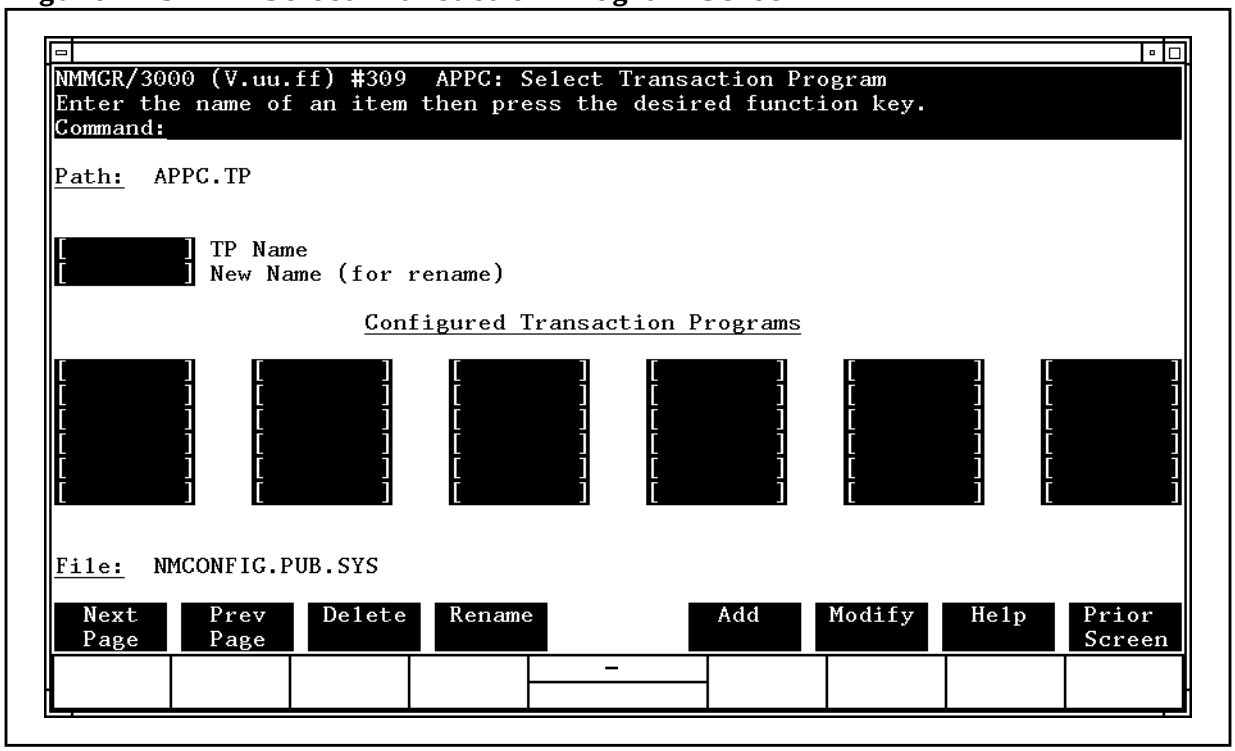
Required. The maximum number of RUs the remote LU can send before it must receive a response from the local LU. For IBM mainframe communication, this must match the *SSNDPAC* parameter of the MODEENT macro in the Logmode Table. For AS/400 communication, this must match the OUTPACING value in the Mode Description on the AS/400.

Default: 7

APPC: Select Transaction Program Screen

Any transaction programs that will call the MCGetAllocate intrinsic to accept allocate requests from remote TPs must be configured in the "APPC: Select Transaction Program" screen, shown in Figure 4-13. The information needed to run each remotely initiated local TP is configured in the "APPC: Transaction Program Data" screen, which you reach by pressing **[f5]** (Add) or **[f6]** (Modify).

Figure 4-13 Select Transaction Program Screen



Fields

TP Name

Required. The name of a new remotely initiated TP you want to configure or an existing remotely initiated TP whose configuration you want to modify. A TP name can be up to 8 alphanumeric characters; the first character must be alphabetic. Up to 256 TP names may be configured.

The remote TP sends a TP name to indicate the local TP with which it wants a conversation. When the APPC subsystem receives the local TP name from the remote TP, it streams a job that runs the local TP. The job name for each remotely initiated local TP is configured in the "APPC: Transaction Program Data" screen.

The TP Name field must match the TP name sent by the remote TP. It must also match the *LocalTPName* parameter of the local TP's TPStarted intrinsic call, and it must match the *LocalTPName* parameter of the local TP's MCGetAllocate intrinsic call.

NOTE

On older versions of the APPC subsystem, the remote TP sends the job name in its allocate request. To avoid changing remote Taps that work with older versions of the APPC subsystem, make the TP Name field of the "APPC: Select Transaction Program" screen (and the *LocalTPName* parameter in the local TP) match the name of the job file and the job name in the JOB card statement of the file.

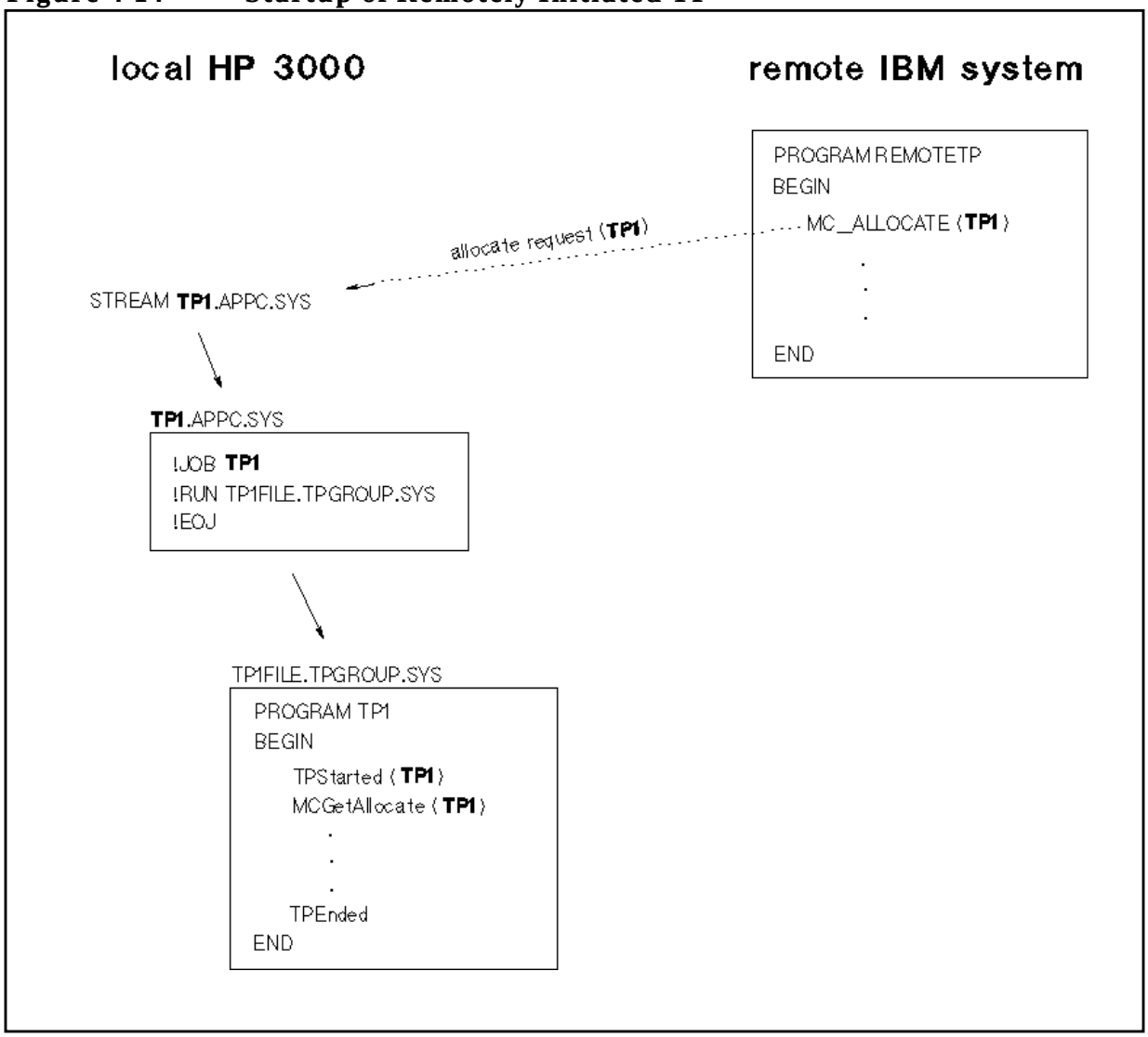
New Name

The new name you want to assign to a configured TP. To rename a TP, enter its old name in the TP Name field, enter its new name in the New Name field, and press **[f4]** (Rename).

Figure 4-14 shows how remotely initiated TPs are started. All of the following values must match:

- The TP Name configured in the "APPC: Select Transaction Program" screen.
- The local TP name sent by the remote TP in the allocate request.
- The *LocalTPName* parameter of the TPStarted intrinsic.
- The *LocalTPName* parameter of the MCGetAllocate intrinsic.

Figure 4-14 Startup of Remotely Initiated TP

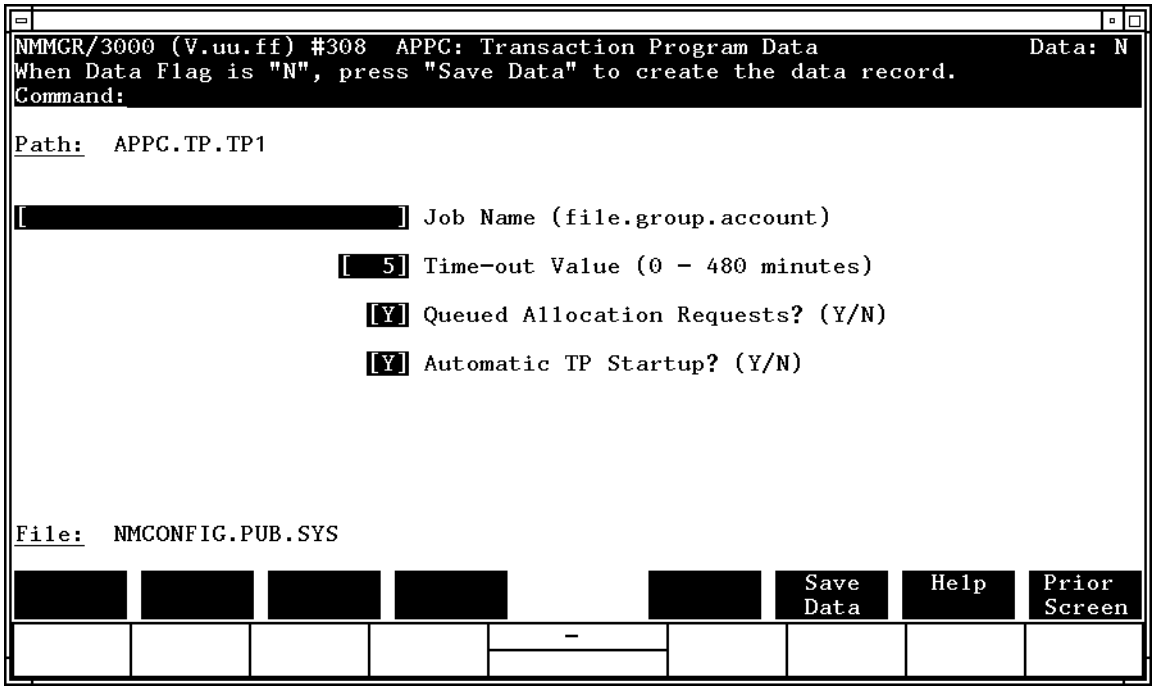


APPC: Transaction Program Data Screen

The "APPC: Transaction Program Data" screen, shown in Figure 4-15, indicates the following things:

1. The location of the job file used to run the TP.
2. How long the local TP's `MCGetAllocate` intrinsic call will wait for an allocate request from the remote TP before timing out.
3. Whether allocate requests from the remote TP should be queued to wait for the current local TP process to call the `MCGetAllocate` intrinsic, or whether a new local TP process should be started for every allocate request.
4. How the TP will be started—manually by a user or automatically by the APPC subsystem.

Figure 4-15 Transaction Program Data Screen



Fields

Job Name

Required. This is the 26-character location (*filename.groupname.accountname*) of the job file that runs the local TP. This file must not contain any lockwords. When the APPC subsystem receives an allocate request from the remote TP, it streams this job, which runs the local TP.

If you want to start the local TP manually, leave this field blank and configure `N` in the `Automatic TP Startup` field.

Time-out Value

This parameter tells the APPC subsystem how long to wait for an allocate request from the remote TP after the local TP calls the `MCGetAllocate` intrinsic. If the timer expires before an allocate request arrives, the `MCGetAllocate` intrinsic returns with a status info value of +29. For more information on the `MCGetAllocate` intrinsic, see the *LU 6.2 API Application Programmer's Reference Manual*.

Allowable `Time-out Values` are as follows:

1–480 minutes If `Queued Allocate Requests = Y`. (A `Time-out Value` of `0` is not allowed if allocate requests are queued.)

0–480 minutes If `Queued Allocate Requests = N`

Default: 5 minutes, if `Queued Allocate Requests = Y`

0 minutes, if `Queued Allocate Requests = N`

`Queued Allocate Requests?`

Required. This parameter tells the APPC subsystem whether to queue allocate requests from the remote TP until the current local TP process calls the `MCGetAllocate` intrinsic, or whether to initiate a new TP process for every allocate request that arrives from the remote TP.

`Y` The local TP accepts queued allocate requests, so only one instance of it may be running at once. It must be written to conduct multiple conversations initiated by the remote TP. (It must make multiple calls to the `MCGetAllocate` intrinsic.) If the local TP is not running when an allocate request arrives from the remote TP, the APPC subsystem runs the local TP. If the local TP is running and in conversation when an allocate request arrives, the allocate request is queued until the local TP deallocates the current conversation and calls `MCGetAllocate` again. (See Figure 4-17.)

`N` The local TP does not accept queued allocate requests, so multiple instances of it may be running at once. It must be written to conduct only one remotely initiated conversation. (It must make only one call to the `MCGetAllocate` intrinsic.) Every time an allocate request arrives from the remote TP, the APPC subsystem creates a new instance of the local TP.

Default: `Y`

Automatic TP Startup?

This parameter tells the APPC subsystem whether or not to stream a job to run the local TP when an allocate request arrives from the remote TP.

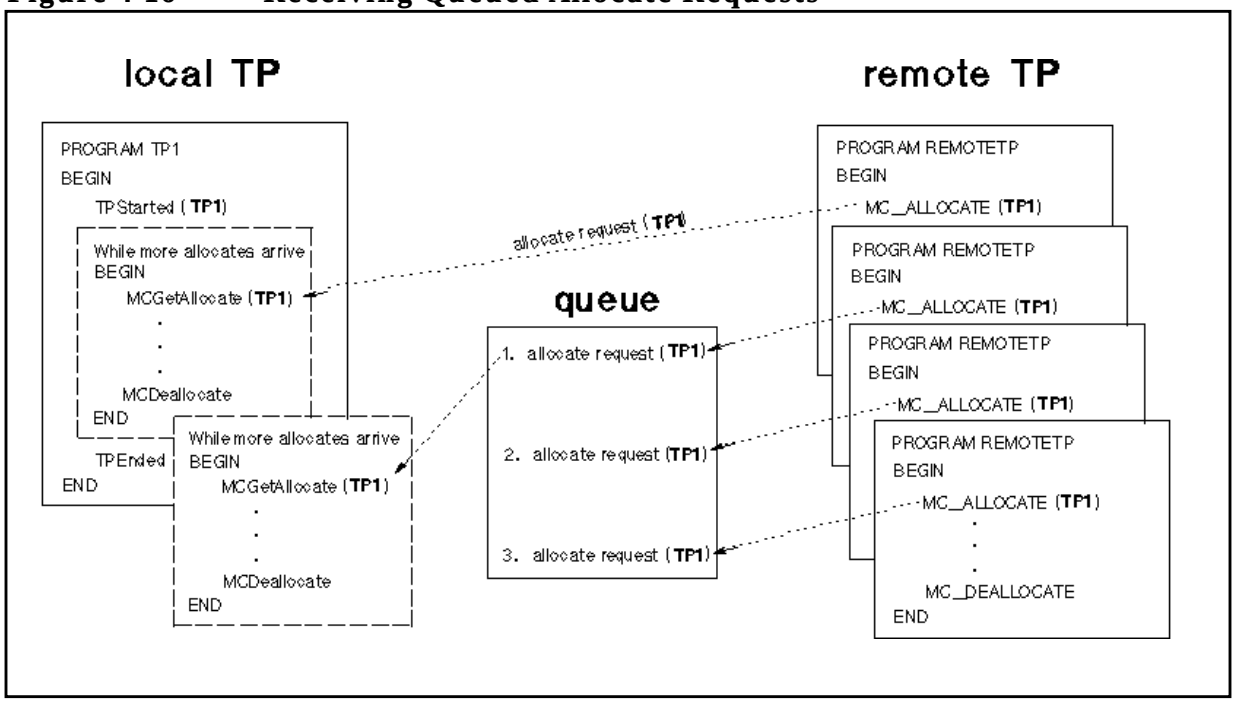
Y	The APPC subsystem will start the local TP automatically by streaming a job when it receives an allocate request from the remote TP. If the local TP is configured to accept queued allocate requests, the APPC subsystem will stream the job only once. If the local TP is not configured to accept queued allocate requests, the APPC subsystem will stream the job to start a new TP process every time it receives an allocate request from the remote TP.
N	The APPC subsystem will not start the local TP automatically. The user must start the local TP by issuing the MPE RUN command or by streaming a job that issues the RUN command.

NOTE While you are developing and debugging a remotely initiated TP, you might want to start it manually. However, after the TP has been debugged, Hewlett-Packard recommends that you allow the APPC subsystem to start it up.

Figure 4-16 shows how allocate requests from the remote TP can be queued to wait until the local TP calls the MCGetAllocate intrinsic. The TP in Figure 4-16 loops through the same conversation, beginning with MCGetAllocate and ending with MCDeallocate, until it receives the last allocate request.

The TP could be written to handle a predetermined number of allocate requests, or it could loop through the conversation until the last MCGetAllocate call timed out because the queue was empty. When MCGetAllocate times out, it returns with a status value of +29. The MCGetAllocate time-out value is configured in the “APPC: Transaction Program Data” screen.

Figure 4-16 Receiving Queued Allocate Requests



Planning APPC Configuration

Before using NMMGR, you should plan the following things:

1. The number of session types you will configure.
2. The number of sessions you will need for each session type.
3. The number of sessions that will be activated automatically at subsystem startup.

Appendix C , “Configuration Worksheets,” provides forms you can fill out to help you plan APPC subsystem configuration. Step-by-step instructions for completing the forms are included in Appendix C , “Configuration Worksheets.”

Planning the Number of Session Types

You may configure up to 60 session types for the APPC subsystem. This total includes both independent and dependent LU session types. The number of session types you will need depends on three things:

1. The number of SNA nodes (copies of SNA/SDLC Link/XL) the APPC subsystem will use. Each session type is configured to use only one link to a remote system.
2. The number of remote LUs with which APPC LUs will interact. Each session type is configured to communicate with only one remote LU.
3. The number of sessions you want to reserve for particular groups of users. A reserved session type should use LUs that are not configured for any other session type.

Session Types for Different SNA Nodes

Each SNA node (each copy of SNA/SDLC Link/XL) on the HP 3000 is a link with one remote system. If you plan to use several SNA nodes, you will need to configure at least one session type for each node. The SNA Node Name field of the “APPC: Independent LU Session Type Data” or “APPC: Dependent LU Session Type Data” screen specifies the SNA node you want to use.

Some remote systems, like the IBM AS/400, can perform intermediate routing between nodes in an SNA network. If your HP 3000 is connected to one of these systems, you can configure independent LU session types to communicate with nodes that are not directly connected to your local system. The Fully Qualified Remote LU Name field of the “APPC: Independent LU Session Type Data” screen specifies the network location of the LU with which you want to communicate.

Session Types for Different Remote LUs

You must configure at least one session type for each remote LU with which the APPC subsystem will communicate.

A session type that will communicate with an independent LU on the remote system must be an independent LU session type. An independent LU on the HP 3000 can conduct multiple, simultaneous APPC sessions with an independent remote LU. One local independent LU can be configured to communicate with several remote LUs simultaneously, as long as the remote LUs are all connected to the HP 3000 through the same copy of SNA/SDLC Link/XL.

Session types that will communicate with dependent LUs on the host must be dependent LU session types.

Session Types for Reserved Sessions

A group of users or an LU 6.2 product requiring guaranteed session access should have one or more session types reserved for its use. When you configure a reserved session type, you should specify LUs that no other session type will use.

Planning the Number of Active Sessions

Every conversation between transaction programs requires an APPC session. When planning the number of active sessions, you must take into account the following restrictions:

1. The total number of simultaneously active sessions, for all session types combined, may not exceed 256. You can configure more than 256 sessions, as long as you never have more than 256 sessions active at once.
2. An independent LU session type has only one independent LU associated with it. This independent LU can support a maximum of 256 active sessions.
3. A dependent LU session type can support up to 32 active sessions. You must configure at least one dependent LU for every active dependent LU session.

The number of sessions you will need for each session type depends on 3 things:

1. How many transaction programs will use the session type at once.
2. How many instances of each transaction program will be running at once.
3. How many conversations each instance of a transaction program will be conducting at once.

The number of sessions you will need for each session type is

```
[(instances of TP1) ö (simultaneous conversations with TP1)]
+ [(instances of TP2) ö (simultaneous conversations with TP2)]
                              .
                              .
                              .
+ [(instances of TPn) x (simultaneous conversations with TPn)]
```

where TP1, TP2, . . . TP*n* are the transaction programs that will be using the session type at the same time.

Planning Automatic Session Activation

When you configure a session type, you specify how many sessions of that type will activate automatically at subsystem startup. No more than 256 sessions may be active at once, so the sum of all the automatically activated sessions, for all APPC session types, may not exceed 256.

Once the APPC subsystem has started up, you can use the APPCCONTROL SESSIONS command or the APPCSessions intrinsic to change the number of active sessions or reapportion sessions among different session types during run time. See Chapter 5 , “Managing the APPC Subsystem,” for more information.

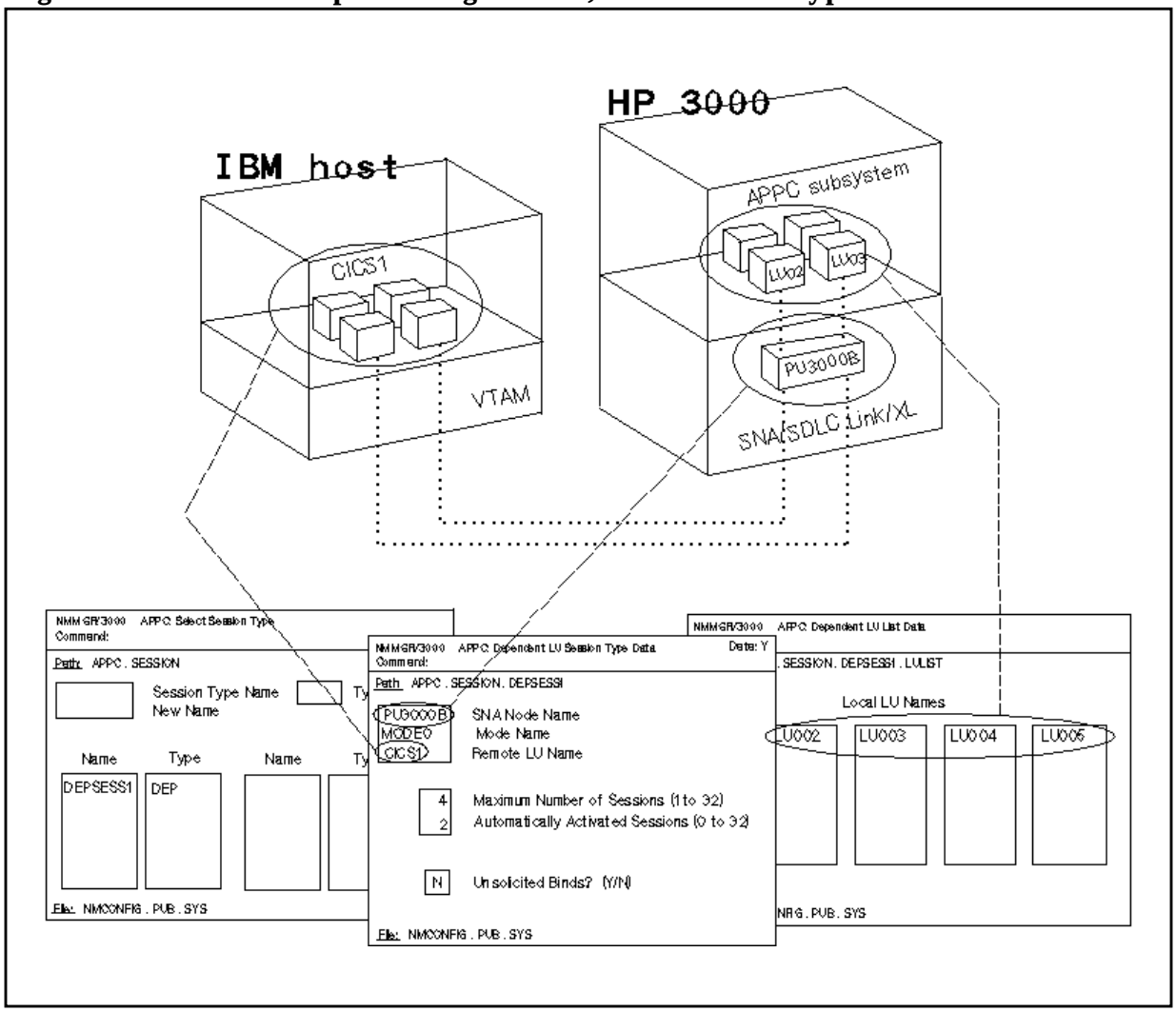
Configuration Illustrations

This section contains network diagrams for three example configurations.

The first illustration, Figure 4-17, shows one dependent LU session type. The three screens pictured at the bottom of the illustration are the APPC configuration screens used to configure a dependent LU session type.

The dashed lines show the correspondence between configuration items and software entities in the SNA network. The dotted lines show the APPC sessions that are activated at subsystem startup.

Figure 4-17 Example Configuration, One Session Type

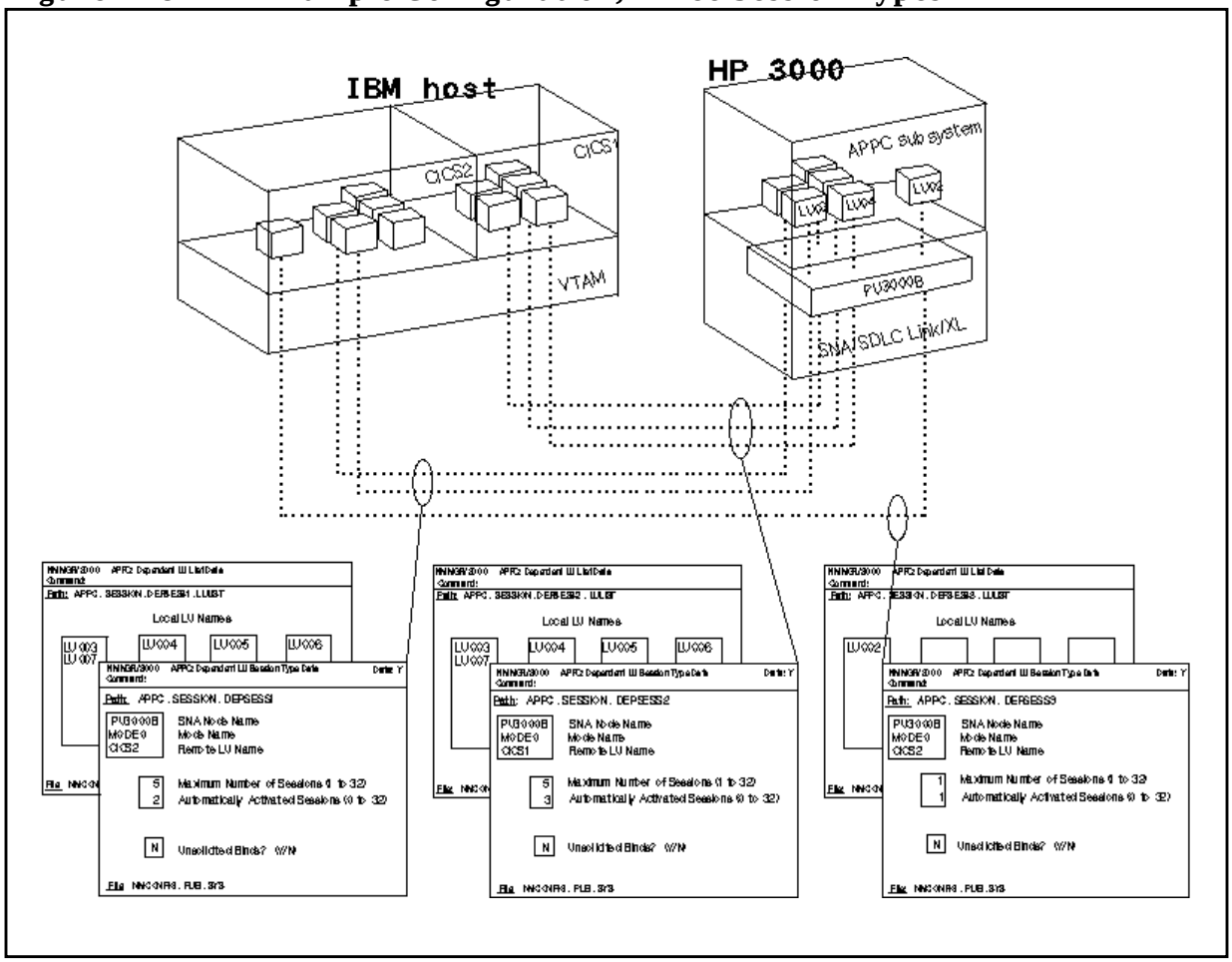


The second illustration, Figure 4-18, shows three dependent LU session types using the same SNA node.

Session types DEPSESS1 and DEPSESS2 share the same five LUs, each of which can communicate with either copy of CICS on the remote system. At subsystem startup, sessions are activated on all five LUs: two of type DEPSESS1 and three of type DEPSESS2. To activate another session of either type, you would have to deactivate a session of the other type in order to free an LU.

Session type DEPSESS3 has guaranteed access to one session, because its LU (LU002) is not configured under any other session type.

Figure 4-18 Example Configuration, Three Session Types



The third illustration, Figure 4-19, shows four session types using three SNA nodes. The dotted lines represent APPC sessions that are activated at subsystem startup.

Figure 4-19 Example Configuration, Four Session Types

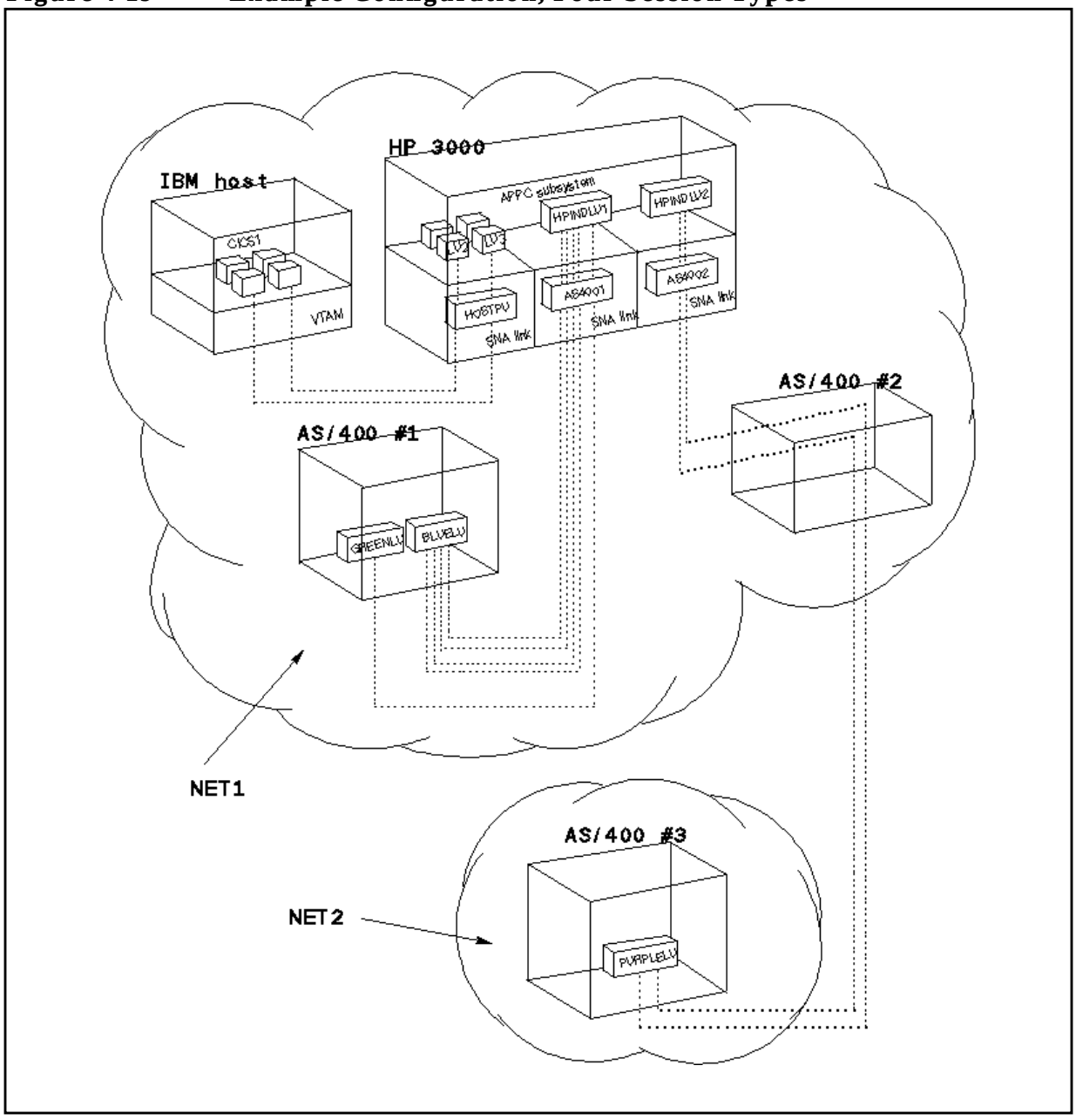


Figure 4-19 shows two SNA networks. AS/400 #2 has intermediate routing capabilities that allow it to connect the HP 3000 in network 1 with AS/400 #3 in network 2.

The APPC subsystem is using three SNA nodes on the HP 3000: HOSTPU, AS4001, and AS4002. Each connection to a remote system requires a separate configured SNA node and a separate copy of SNA/SDLC Link/XL.

Independent LU HPINDLU1 on the HP 3000 is configured to communicate with two independent LUs on AS/400 #1: GREENLU and BLUELU. One session type is configured for each remote independent LU.

The other independent LU on the HP 3000, HPINDLU2, uses the intermediate routing capabilities of AS/400 #2 to communicate with independent LU PURPLELU on AS/400 #3. AS/400 #3 is not directly connected to the HP 3000, and it resides in a different network. The Fully Qualified Remote LU Name configured for HPINDLU2 tells AS/400 #2 the network location of the destination LU, PURPLELU.

The dependent LU session type using HOSTPU has a limit of 4 active sessions. Two of them are configured for automatic activation, so the illustration shows two sessions active between the LUs on the HP 3000 and the LUs on the IBM host.

See Appendix B , “Sample Configuration,” for sample SNA node configuration screens and APPC configuration screens for the example in Figure 4-19.

Coordinating with the TP Programmer

To use sessions, each local TP must pass the name of a configured session type to the APPC subsystem. After you have completed APPC configuration, you must tell the TP programmers the session type names to use for development and run time. For information on TP development, see the *LU 6.2 API Application Programmer's Reference Manual*.

Migrating an APPC Configuration

If you are migrating your APPC subsystem configuration from MPE V to MPE XL, or from an earlier version of MPE XL to a version that supports Node Type 2.1, run the NMS utility `NMMGRVER.PUB.SYS`, described in *Using the Node Management Services Utilities*. NMMGRVER will convert your old configuration to a configuration that is compatible with your new software.

5 Managing the APPC Subsystem

This chapter contains the following sections:

- **Activating the APPC Subsystem** tells you how to start up the APPC subsystem and all APPC sessions configured for automatic activation.
- **Managing APPC Sessions** tells you how to check the status of the APPC subsystem and how to activate and deactivate APPC sessions during run time.
- **Establishing Conversations** tells you how to establish remotely and locally initiated conversations between transaction programs. It describes the tasks that must be completed by the node manager and those that must be completed by the TP programmer.
- **Shutting Down the APPC Subsystem** tells you how to deactivate the APPC subsystem and all APPC sessions.

NOTE The control operator commands `APPCCONTROL START`, `APPCCONTROL STOP`, and `APPCCONTROL SESSIONS` can be issued programmatically by calling the control operator intrinsics `APPCStart`, `APPCStop`, and `APPCSessions`. Control operator intrinsics are described in Chapter 3 , “Control Operator Intrinsics.”

Activating the APPC Subsystem

The subsystem is activated by issuing the APPCCONTROL START command. The APPCCONTROL START command can be issued from any user terminal by anyone with NM capability. Once the command is issued, the APPC subsystem and all APPC sessions configured for automatic activation will come up. (Automatic activation is discussed in Chapter 4 , “APPC Subsystem Configuration,” “APPC Subsystem Configuration.”)

NOTE The APPCCONTROL START command will not cause the activation of the SNA link product if it is not already activated. The SNA link product must be explicitly activated using the SNACONTROL START command.

When the APPC subsystem starts up, it generates startup and logging messages. These messages tell you how activation is progressing. Logging messages can be directed to the system console, a disk file, or a user terminal (depending upon configuration). The startup messages returned by the APPC subsystem are directed back to the terminal from which the APPCCONTROL START command was issued. The startup messages you should see after issuing the APPCCONTROL START command are shown below:

```
:APPCCONTROL START
    APPC:  Subsystem start up begun.
    APPC:  Opening internal trace file - NMTCnnnn.PUB.SYS or user trace file.
    APPC:  Control Operator start-up completed.
```

At this point, the APPC subsystem is active, but no sessions have been started yet.

APPC subsystem logging messages record the establishment of APPC sessions. Typical messages are shown below. They could be directed to a user terminal, the console, a disk file, or all three.

```
APPC:  Subsystem Started (APPCLOG 0).
APPC:  Session Started, Session ID=41 (APPCLOG 6).
APPC:  Session Started, Session ID=42 (APPCLOG 6).
```

If, for any reason, the APPC subsystem is unable to execute the APPCCONTROL START command, an error message will be displayed on the terminal from which the command was issued. Appendix A , "Messages," lists APPC subsystem error messages, their meanings, and the actions you should take to resolve errors.

If you configure more than 256 sessions for automatic activation, the following messages will be generated:

```
SESSION LIMIT EXCEEDED, REMAINING SESSION IGNORED.  (APPCWARN 1201)

   APPC:  SNA/LINK Error, Session ID=43, Result=!  (APPCLOG 11)
   APPC:  Session Stopped, Session ID=43, Reason=SNA_Error (APPCLOG 7)
```

The first message will appear on the terminal from which the APPCCONTROL START command was issued. The remaining two are logging messages; they will be logged to the destinations configured.

Managing APPC Sessions

This section discusses the APPCCONTROL STATUS and APPCCONTROL SESSIONS commands.

You issue the APPCCONTROL STATUS command to check the status of the APPC subsystem during run time.

You use the APPCCONTROL SESSIONS command to control the number of active APPC sessions.

Checking APPC Subsystem Status

At subsystem startup, the number and characteristics of active sessions are determined by the parameters in the APPC configuration file. However, once the APPC subsystem is active, you can use the APPCCONTROL SESSIONS command to change session limits and session characteristics so that they no longer match the configuration parameters. The APPCCONTROL STATUS command is used to get the current status of the APPC subsystem.

The description of the APPCCONTROL STATUS command in Chapter 2 , “Interactive Control Operator Commands,” contains an item-by-item description of the display. This section uses several examples to illustrate the use of the APPCCONTROL STATUS command.

In the following example, the APPCCONTROL STATUS command is issued with no parameters, so information is displayed for all configured session types and all active TPs. The display shows two configured session types: one independent and one dependent. The independent session type, INDSESS1, has no active sessions. The dependent session type, DEPSESS1, has two active sessions.

Each active session is being used by a separate instance of the same TP. The TPIDs are different, indicating two different TP processes, but the Program File Name and TP Name are the same for both processes.

```
:APPCCONTROL STATUS

Internal Trace File : NMTC0020.PUB.SYS
Performance Tracing : ON

-------------------------------------------------
Number of Active Sessions for APPC Subsystem :   2
-------------------------------------------------

Session Type Information:

   --------------------------------------
   Independent LU Session Type : INDSESS1
   --------------------------------------

   Number of Active Sessions         :   0
   Number of Queued Session Requests :   0
                                                        Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU1     SNANODE    NET1.IBMLUA         MODE1       50       0

   Session ID    LFSID    State       LU Type    TP ID
   ----------    -----    --------    -------    -----
   * No active sessions

   -------------------------------------
   Dependent LU Session Type : DEPSESS1
   -------------------------------------

   Number of Active Sessions         :   2
   Number of Queued Session Requests :   1

                                 Maximum
   SNA Node   Remote LU   Mode    Sessions
   --------   ---------   ------  --------
   IBMNODE    DISOSSC     MODE3        8

   Session ID   LFSID   Local LU (NAU)  State     LU Type  TP ID
   ----------   -----   --------------  --------  -------  -----
      15        0101 1  DEPLU8 (  8)    ACTIVE      SEC    2398
      52        0102 1  DEPLU9 (  9)    ACTIVE      SEC    2402
Transaction Program Information:

   TP ID    Program File Name       TP NAME      # Conv.    User Name
   -----    --------------------    ---------    -------    ------------
   2398     TP1.PUB.SYS             TP1             1       JOE.USER
   2402     TP1.PUB.SYS             TP1             1       FRED.USER
```

In the following example, the APPCCONTROL STATUS command is issued with the parameter STYPE=INDSESS2, so information is displayed for only one session type. Session type INDSESS2 is an independent session type with two active sessions. The identical TPIDs show that both sessions are being used by one instance of the same TP; that is, one instance of the TP is conducting two conversations simultaneously, over two APPC sessions.

```
:APPCCONTROL STATUS;STYPE=INDSESS2

Internal Trace File : NMTC0020.PUB.SYS
Performance Tracing : ON

-------------------------------------------------
Number of Active Sessions for APPC Subsystem :   3
-------------------------------------------------

Session Type Information:

   --------------------------------------
   Independent LU Session Type : INDSESS2
   --------------------------------------

   Number of Active Sessions         :   2
   Number of Queued Session Requests :   0
                                                    Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU2     SNANODE    NET1.IBMLUA         MODE1        8        2

   Session ID    LFSID    State       LU Type    TP ID
   ----------    -----    --------    -------    -----
     12          0103 1   ACTIVE        PRI      1256
     74          0104 1   ACTIVE        PRI      1256
```

Controlling the Number of Sessions

The APPC subsystem supports up to 256 active APPC sessions. By issuing the APPCCONTROL SESSIONS command, you can vary the number of active sessions and reapportion active sessions among different session types.

The following APPCCONTROL STATUS display shows the status of an example APPC session type at subsystem startup. This session type will be used in the rest of the examples in this section to illustrate the effects of the APPCCONTROL SESSIONS command.

```
-------------------------------------
   Independent LU Session Type : INDSESS1
   -------------------------------------

   Number of Active Sessions         :   0
   Number of Queued Session Requests :   4
                                                        Session Limits
   Local LU   SNA Node   Remote LU           Mode     Maximum  Current
   --------   --------   -----------------   ------   -------  -------
   INDLU2     SNANODE    NET1.IBMLUA         MODE1       8        0

   Session ID    LFSID    State       LU Type    TP ID
   ----------    -----    --------    -------    -----
   * No active sessions
```

In the example above, independent session type INDSESS1 has four queued session requests. That means four TP processes have called the MCAllocate or MCGetAllocate intrinsic, specifying INDSESS1 in the *SessionType* parameter. The session requests were queued to wait for available active sessions.

No sessions of type INDSESS1 are active, because the configured number of automatically activated sessions is 0, and no one has issued the APPCCONTROL SESSIONS command to activate more sessions. After the APPCCONTROL SESSIONS command is issued to activate more sessions, the Current session limit in the APPCCONTROL STATUS display will be equal to the Number of Active Sessions.

Activating Sessions

To activate sessions of type INDSESS1, you issue the APPCCONTROL SESSIONS command to raise the session limit for that session type:

```
APPCCONTROL SESSIONS; STYPE=INDSESS1; LIMIT=5
```

When you issue the APPCCONTROL SESSIONS command to raise the session limit to 5, five sessions of the specified session type are activated. The maximum session limit configured through NMMGR is still 8, so you cannot raise the session limit beyond 8 with the APPCCONTROL SESSIONS command. Under Session Limits in the APPCCONTROL STATUS display, the Maximum value is the configured

value. The Current value is the new session limit after you issue the APPCCONTROL SESSIONS command.

NOTE If you issue the APPCCONTROL SESSIONS command to raise the session limit for an independent LU session type with parallel sessions, the limit you specify in the command might be negotiated with the remote LU. Therefore, after the command has executed, the new session limit might not be the same limit you specified in the command. The Session Limits in the APPCCONTROL STATUS display are the configured value (Maximum) and the negotiated value (Current).

After you raise the session limit, the following messages are logged as sessions are activated:

```
APPC: Session Started, Session ID=26 (APPCLOG 6)
APPC: Session Started, Session ID=27 (APPCLOG 6)
APPC: Session Started, Session ID=28 (APPCLOG 6)
APPC: Session Started, Session ID=29 (APPCLOG 6)
APPC: Session Started, Session ID=30 (APPCLOG 6)
```

The LU acting as primary LU activates the session. Sessions can be activated by either the local or the remote LU.

The following APPCCONTROL STATUS display reflects the status of session type INDSESS1 after the APPCCONTROL SESSIONS command has been issued to raise the session limit to 5.

```
-------------------------------------
Independent LU Session Type : INDSESS1
-------------------------------------

Number of Active Sessions         : 5
Number of Queued Session Requests : 0
                                                    Session Limits
Local LU   SNA Node   Remote LU           Mode     Maximum  Current
--------   --------   -----------------   ------   -------  -------
INDLU1     AS400PU    NET1.AS400LU1       MODE1        8        5

Session ID    LFSID    State       LU Type    TP ID
----------    -----    --------    -------    -----
26            0101 1   ACTIVE      PRI        4425
27            0102 1   ACTIVE      PRI        3279
28            0103 1   ACTIVE      PRI        5413
29            0104 1   ACTIVE      PRI        6924
30            0105 1   ACTIVE      PRI
```

The number of active sessions for INDSESS1 is now 5. This is the *current* session limit, which differs from the *configured* session limit. The configured session limit for INDSESS1 will always be 8, unless you modify the configuration file and restart the APPC subsystem.

The four session requests that were queued have been given sessions, so the Number of Queued Session Requests is now 0.

One session is active but idle; it is not being used for a conversation, so its TP ID field is blank.

If two more TP processes request sessions, the first TP process will be given the idle session (the one with no TP ID listed), and the second session request will be queued to wait for an available session.

The following APPCCONTROL STATUS display reflects the status of session type INDSESS1 after two more TP processes call MCAllocate or MCGetAllocate to request sessions. Five sessions are still active, and one session request is now queued. The session that was idle is now being used by the TP process with TP ID 1258.

```
-------------------------------------
Independent LU Session Type : INDSESS1
-------------------------------------

Number of Active Sessions         :   5
Number of Queued Session Requests :  <$&shade 10-1> 1
                                                    Session Limits
Local LU   SNA Node   Remote LU           Mode     Maximum  Current
--------   --------   -----------------   ------   -------  -------
INDLU1     AS400PU    NET1.AS400LU1       MODE1        8        5

Session ID    LFSID    State       LU Type    TP ID
----------    -----    --------    -------    -----
26            0101 1   ACTIVE      PRI        4425
27            0102 1   ACTIVE      PRI        3279
28            0103 1   ACTIVE      PRI        5413
29            0104 1   ACTIVE      PRI        6924
30            0105 1   ACTIVE      PRI 1258
```

If three conversations are deallocated, three sessions will become available for conversations. One session will be given to the TP with the queued session request, and the other two available sessions will remain active but idle.

The following APPCCONTROL STATUS display reflects the status of session type INDSESS1 after three conversations are deallocated. The TP ID associated with Session ID 28 changes from 5413 to 2215; the session remains active, but a different TP is using it. The TP ID field for each of the idle sessions is blank.

```
--------------------------------------
Independent LU Session Type : INDSESS1
--------------------------------------

Number of Active Sessions         :   5
Number of Queued Session Requests :  <$&shade 11-1> 0
                                                    Session Limits
Local LU   SNA Node   Remote LU           Mode     Maximum  Current
--------   --------   -----------------   ------   -------  -------
INDLU1     AS400PU    NET1.AS400LU1       MODE1        8        5

Session ID    LFSID    State       LU Type    TP ID
----------    -----    --------    -------    -----
26            0101 1   ACTIVE      PRI        4425
27            0102 1   ACTIVE      PRI
28            0103 1   ACTIVE      PRI       <$&shade 11-2> 2215
29            0104 1   ACTIVE      PRI       <$&shade 11-3> 6924
30            0105 1   ACTIVE      PRI
```

Terminating Sessions

To terminate sessions, you issue the APPCCONTROL SESSIONS command to reduce the session limit for a session type. The following command reduces the number of active sessions of type INDSESS1 from 5 to 2.

```
APPCCONTROL SESSIONS; STYPE=INDSESS1; LIMIT=2
```

Idle sessions are terminated first. Since two sessions of type INDSESS1 are idle, these sessions are terminated immediately. Another session will be terminated as soon as the conversation using it is deallocated. The following messages are logged as sessions are terminated:

```
APPC: Session Stopped, Session ID=27, Reason=APPC_Quiesce
APPC: Session Stopped, Session ID=30, Reason=APPC_Quiesce
```

The following APPCCONTROL STATUS display reflects the status of session type INDSESS1 after the APPCCONTROL SESSIONS command is issued to reduce the session limit from 5 to 2. The Number of Active Sessions is 3, not 2, because the APPC subsystem is waiting for a conversation to be deallocated before terminating another session.

```
-------------------------------------
Independent LU Session Type : INDSESS1
-------------------------------------

Number of Active Sessions         :   3
Number of Queued Session Requests :   0
                                                        Session Limits
Local LU   SNA Node   Remote LU           Mode     Maximum  Current
--------   --------   -----------------   ------   -------  -------
INDLU1     AS400PU    NET1.AS400LU1       MODE1        8        2

Session ID    LFSID    State       LU Type    TP ID
----------    -----    --------    -------    -----
26            0101 1   ACTIVE      PRI        4425
28            0103 1   ACTIVE      PRI        2215
29            0104 1   ACTIVE      PRI        6924
```

To terminate all sessions of type INDSESS1, you issue the APPCCONTROL SESSIONS command to reduce the session limit to 0:

```
:APPCCONTROL SESSIONS; STYPE=INDSESS1; LIMIT=0
```

This command terminates all active sessions of type INDSESS1. Sessions are terminated with a shutdown type of QUIESCE, meaning that all conversations on the affected sessions are allowed to end before the sessions are terminated. A message is logged for each session that is terminated.

NOTE

If, for some reason, the APPCCONTROL SESSIONS command fails to bring down all the sessions you wish to terminate, you can use the APPCCONTROL STOPSESSION command. The APPCCONTROL STOPSESSION command is used to kill a session that is still active after you have tried to terminate it with the APPCCONTROL SESSIONS command. The APPCCONTROL STOPSESSION command terminates a session with a StopType of KILL; that is, it terminates the session immediately, without waiting for any conversation using the session to complete.

An APPC session can be terminated from the remote side. When the remote system terminates a session, the APPC subsystem attempts to reestablish it. The following is an example of the logging messages you would see if this occurred:

```
APPC: Session stopped, Session ID=26, Reason=Host_Unbind  (APPCLOG 7)
APPC: Session started, Session ID=26  (APPCLOG 6)
```

Establishing Conversations

Conversations between transaction programs can be locally initiated (initiated by the TP on the HP 3000), or remotely initiated (initiated by the TP on the remote system). This section describes the tasks that the node manager, the local application programmer, and the remote application programmer must perform in order to establish a locally or remotely initiated conversation.

Locally Initiated Conversations

The following things must occur for a local TP to initiate a conversation:

1. An APPC session of the appropriate session type must be established.
2. A local end user must run the local TP.
3. The local TP must send an allocate request over the session assigned to it, to request a conversation with the remote TP.
4. The remote TP must be coded to receive the allocate request from the local TP.

Note that the session must be established before the local TP can use it to send the allocate request. This section describes the tasks that the node manager and the local application programmer must perform in order to establish a locally initiated conversation.

Node Manager Tasks

To prepare for a locally initiated conversation, you must do the following:

1. Configure an appropriate session type. The session type must direct data to the remote LU that serves the remote TP. For information on configuring session types, see Chapter 4 , “APPC Subsystem Configuration.”
2. Tell the application programmer the name of the session type. The programmer must code the name of the session type into the local TP.
3. Activate a session of the appropriate session type by issuing the `APPCCONTROL SESSIONS` command, or configure a session for automatic activation at subsystem startup.

Local Programmer Tasks

To prepare for a locally initiated conversation, the local TP programmer must do the following:

1. Work with the programmer on the remote system to design and code the TP.
2. Ask the node manager for the name of an appropriately configured session type, or ask the node manager to configure a session type for the conversation.
3. Code the name of the session type into the local TP.

Remotely Initiated Conversations

The following things must occur for a remote TP to initiate a conversation with a local TP:

1. An APPC session of the appropriate session type must be established.
2. The remote TP must issue an allocate request over that session, specifying the name of the local TP with which it wants a conversation.
3. The APPC subsystem must receive the allocate request, look up the local TP name in the APPC subsystem configuration file, and determine from the configuration file what to do with the allocate request.
   a. If the local TP is configured to conduct multiple remotely initiated conversations, and if it is already active and in conversation, the APPC subsystem must queue the allocate request until the local TP finishes the current conversation and calls the `MCGetAllocate` intrinsic again. If the local TP is not currently running, the APPC subsystem must stream the job that runs the local TP.
   b. If the local TP is configured to conduct only one remotely initiated conversation, the APPC subsystem must stream the job that runs the local TP.
4. The local TP must be coded to receive the allocate request from the remote TP.

NOTE Any local TPs that will conduct remotely initiated conversations must be configured through NMMGR. See Chapter 4 , “APPC Subsystem Configuration,” for information on TP configuration.

Node Manager Tasks

To prepare for a remotely initiated conversation, you must do the following:

1. Configure an appropriate session type. The session type must direct data to the remote LU that serves the remote TP. See Chapter 4 , "APPC Subsystem Configuration," for information on configuring session types.
2. Tell the application programmer the name of the session type. The programmer must code the name of the session type into the local TP.
3. Activate a session of the appropriate session type by issuing the `APPCCONTROL SESSIONS` command, or configure a session for automatic activation at subsystem startup.
4. Ask the local TP programmer for the TP name, and configure it in the "APPC: Select Transaction Program" screen. Make sure the configuration file, the local TP programmer, and the remote TP programmer all agree on the TP name. The same TP name must be configured through NMMGR, coded into the local TP, and sent by the remote TP in the allocate request.
5. Create a job that runs the executable TP file. Job files and executable TP files may reside in any group and account. A job file name must be eight characters long and begin with an alphabetic character. Job files must not contain any passwords or lockwords.

   The APPC subsystem configuration file associates each configured TP name with a job file name. When the APPC subsystem receives the local TP name from the remote TP, it gets the job file name from the configuration file and streams the job, which runs the local TP. Each remotely initiated local TP requires a separate job file.

   Here is an example job file that would run the local TP `HPTP.BOB.MAPPED` at the request of a remote TP. The name of the job file is `JOBFILE1.APPC.SYS`.

   ```
   !JOB JOBFILE1.APPC.SYS
   !RUN HPTP.BOB.MAPPED
   !EOJ<F255P255>
   ```

6. Configure the name of the job file in the "APPC: Transaction Program Data" screen in NMMGR.
7. Configure the time-out value for the `MCGetAllocate` intrinsic in the "APPC: Transaction Program Data" screen in NMMGR. After the local TP calls the `MCGetAllocate` intrinsic to receive an allocate request, it will be suspended until an allocate request arrives from the remote TP or until the configured time-out value expires. Ask the local TP programmer how long the TP should wait for an allocate request from the remote TP.

8. Configure the local TP to receive queued or non-queued allocate requests. Ask the local TP programmer whether the local TP will call the `MCGetAllocate` intrinsic multiple times or only once. If the local TP will call `MCGetAllocate` multiple times, configure it to accept queued allocate requests; if it will call `MCGetAllocate` only once, configure it to accept non-queued allocate requests. Queued or non-queued allocate requests are configured in the "APPC: Transaction Program Data" screen in NMMGR.
9. Configure the local TP for manual or automatic startup. Ask the local TP programmer whether the APPC subsystem should automatically stream a job to run the local TP when it receives an allocate request from the remote TP or whether the TP programmer wants to start up the local TP manually by issuing the MPE `RUN` command. Automatic or manual TP startup is configured in the "APPC: Transaction Program Data" screen in NMMGR.

Local Programmer Tasks

To prepare for remotely initiated conversations, the local TP programmer must do the following:

1. Work with the programmer on the remote system to design and code the TP. The local TP can be designed to receive multiple allocate requests from the remote TP, or it can be designed to receive only one allocate request.
2. Ask the node manager for the name of an appropriately configured session type, or ask the node manager to configure a session type for the conversation.
3. Code the name of the session type into the local TP.
4. Together with the programmer on the remote system, decide on a name for the TP. It must be coded into the local TP, and it must be sent by the remote system in the allocate request. Tell the node manager the TP name. The node manager must configure the TP name in the "APPC: Transaction Program Data" screen in NMMGR.
5. Code the TP name into the local TP.
6. Tell the node manager the executable file name of the TP. The node manager will create a job to run the TP.
7. Tell the node manager how long the local TP should wait for an allocate request from the remote TP before the `MCGetAllocate` intrinsic times out. The node manager will configure the time-out value.
8. Tell the node manager whether the TP calls the `MCGetAllocate` intrinsic multiple times or only once. The node manager must configure the TP to accept either single or queued allocate requests.

9. Tell the node manager whether to configure the TP for manual or automatic startup.

Remote Programmer Tasks

To prepare a remote program to initiate a conversation with a TP on the HP 3000, the remote programmer must do the following:

1. Design and code the program to initiate a conversation with the corresponding TP on the HP 3000.
2. Make sure that the remote TP passes the proper TP name in the allocate request. The HP application programmer must tell the remote TP programmer which TP name to use.

Shutting Down the APPC Subsystem

You shut down the APPC subsystem by issuing the APPCCONTROL STOP command. Whenever the APPC subsystem shuts down, performance statistics are logged for all sessions that were active at the time of the shutdown (if logging is configured and statistics gathering is enabled).

The APPCCONTROL STOP command shuts down the APPC subsystem by deactivating all APPC sessions. There are three shutdown types you can use: KILL, PROTOCOL, and QUIESCE. See the description of APPCCONTROL STOP in Chapter 2 , “Interactive Control Operator Commands,” for a description of shutdown types.

The following is an example of logging messages you should see for normal APPC subsystem shutdown:

```
APPC: Session Stopped, Session ID=4101, Reason=APPC_Quiesce  (APPCLOG 7)
APPC: Session Stopped, Session ID=4102, Reason=APPC_Quiesce  (APPCLOG 7)
APPC: Subsystem Stopped.  (APPCLOG 1)
```

A session termination message will be logged for each session that is deactivated. The Reason field in the termination message indicates the type of shutdown that occurred. (The description of the APPCCONTROL STOP command in Chapter 2 , “Interactive Control Operator Commands,”explains all the shutdown types.) QUIESCE, the type that is seen in the example above, is the default shutdown type.

Shutting down the APPC subsystem does not affect the operation of SNA/SDLC Link/XL, so any other SNA Services that may be using the SNA link are not affected.

The SNACONTROL STOP command shuts down SNA/SDLC Link/XL, which causes *all* LU-LU sessions on the SNA node to shut down, including those used for the APPC subsystem. The shutdown type (QUIESCE, PROTOCOL, or KILL) that you specify in the SNACONTROL STOP command is the shutdown type for every LU-LU session on the node you are shutting down. For example, an SNACONTROL STOP command issued with a shutdown type of KILL (immediate shutdown of the node) causes a similar termination of all LU-LU sessions on the node specified. For information on the SNACONTROL STOP command, see the *SNA Link/XL Node Manager's Guide*.

6 Troubleshooting the APPC Subsystem

This chapter gives you troubleshooting procedures to follow when you suspect a problem with the APPC subsystem. It should be used with the *SNA Link/XL Node Manager's Guide*.

This chapter contains the following sections:

- **Troubleshooting tools.** This section describes the tools available to the APPC subsystem node manager.
- **Startup problems.** These problems commonly occur during installation and configuration.
- **Run-time problems.** These problems occur after the APPC subsystem and the LU 6.2 products have been up and running.
- **Submitting an SR.** This section tells you what information to include when you submit an SR (Service Request).

Troubleshooting Tools

This section describes the available troubleshooting tools and tells you how to use them to troubleshoot the APPC subsystem. The tools provided on the HP 3000 for troubleshooting the APPC subsystem are as follows:

- the APPC subsystem internal trace
- APPC subsystem logging
- APPC subsystem messages and return codes
- the APPCCONTROL STATUS command
- the SNACONTROL STATUS command

NOTE User tracing is available for debugging transaction programs that call LU 6.2 API intrinsics. User tracing is documented in the *LU 6.2 API Application Programmer's Reference Manual*.

APPC Internal Tracing

Internal tracing can be turned on at subsystem startup or during run time. It traces the internal events of the APPC subsystem. Once the trace has been turned on, it remains on until the APPC subsystem shuts down or until you issue the APPCCONTROL TRACEOFF command.

At subsystem startup, internal tracing is turned off by default, but you can turn tracing on when you issue the APPCCONTROL START command by specifying TRACEON=YES. If you activate the APPC subsystem programmatically, you can turn tracing on by setting the *TraceOn* parameter of the APPCStart intrinsic to 1.

During run time, you can turn internal tracing on by issuing the APPCCONTROL TRACEON command.

NOTE Hewlett-Packard recommends that you turn on internal tracing when you install an LU 6.2 product on your system for the first time.

It is not necessary for you to read and interpret the internal trace, but you may be asked to collect it for analysis by your HP representative.

When you turn on APPC subsystem internal tracing, you can specify a name for the trace file, or you can allow the trace file name to default. The default trace file name is NMTC*xxxx*.PUB.SYS, where *xxxx* is a number from 0000 through 9999. This number is incremented whenever a new trace file is created.

A new trace file is created every time APPC subsystem internal tracing is turned on, at subsystem startup or during run time. To start a new trace file during run time, turn tracing off with the APPCCONTROL TRACEOFF command, and then turn it back on again with the APPCCONTROL TRACEON command.

NOTE When storing trace files to tape for analysis, be sure to include all the files on disk at the time the problem occurred. An incomplete set of trace files can hinder problem diagnosis.

When you turn on APPC subsystem internal tracing, you can specify a size for the trace file, in 128-word records. The trace file size can range from 0 through 32767 records. When you are trying to diagnose a problem, you should specify a trace file size of 32000 records or larger. Once a trace file has reached its specified size, trace records are overwritten starting from the beginning of the file.

APPC Logging

Node Management Services (NMS) provides the logging services for the APPC subsystem. APPC subsystem logging records the major events of the APPC subsystem.

Logging is turned on through NMMGR configuration. The APPC subsystem number is SUB0016. Logging events are divided into five classes:

CLAS0010 records internal errors.

CLAS0011 records APPC subsystem warnings.

CLAS0012 records APPC subsystem information messages.

CLAS0013 records conversation information messages.

CLAS0014 records session performance statistics.

You can configure the logging messages to go to the system console, to the log file, and to three separate *user.account*s. Critical messages should be logged to both the system console and the log file. That way, messages erased from the terminal buffer can still be found in the log file. Hewlett-Packard recommends that all classes of logging messages be recorded in the log file and that all classes except CLAS0014 (performance statistics) be logged to the system console.

The logging messages and their meanings are documented under "Logging Messages," in Appendix A , "Messages," of this manual. For more information on logging and logging configuration, see the *SNA Link/XL Node Manager's Guide*.

APPC Subsystem Messages and Return Codes

Appendix A , “Messages,” contains a complete list of all APPC subsystem messages and return codes. It describes the cause of each message and any actions you should take to resolve a problem. The APPC subsystem generates the following kinds of messages:

- **Subsystem startup messages** are returned to the terminal from which the APPCCONTROL START command is issued. They tell you how subsystem startup is progressing. They are informational only and require no action.
- **Error messages** may be returned when you issue an APPCCONTROL command or when you use the NMDUMP utility to format the user trace or internal trace. They are returned to the terminal from which the APPCCONTROL command was issued. Error messages indicate that an APPCCONTROL command could not be executed or that a trace file could not be formatted.
- **Warning messages** may be returned when you issue an APPCCONTROL command. Some of them indicate problems that you should resolve, and some of them are informational only and require no action.
- **Return codes** are returned in the *ReturnCode* parameter of control operator intrinsics. A return code is the number of a message in the message catalog that indicates whether an intrinsic call was successful and the result of the intrinsic call.
- **APPC subsystem configuration validation messages** are generated through NMMGR when you validate your configuration file.

The APPCCONTROL STATUS Command

The APPCCONTROL STATUS command provides a summary of APPC subsystem status. Chapter 2 , “Interactive Control Operator Commands,” in this manual describes the APPCCONTROL STATUS command and tells you how to read the display.

The APPCCONTROL STATUS command allows you to specify a session type for which you want status information. You can specify a configured session type, or you can specify STYPE=SNASVCMG to get the status of the sessions used by the CNOSTP (Change-Number-of-Sessions TP) and the COPRTP (Control Operator TP). These two transaction programs, which are part of the APPC subsystem, control the number of active sessions and process your requests to change the number of active sessions. The CNOSTP and COPRTP are active only for independent LU session types with parallel sessions.

The SNACONTROL STATUS Command

To get the status of the local SNA node and LU, issue the SNACONTROL STATUS command. This command will tell you whether SNA Transport is active and will display the status of the PU-SSCP sessions and LU-SSCP sessions.

NOTE

For Type 2.1 nodes, the SNACONTROL STATUS command always lists the SESSION STATE as RESET (unless Node 2.1 Dependent LU Support = Y in the "SNA Node Configuration: PU Data" screen).

To find out whether a Type 2.1 node is active, issue the command

```
SNACONTROL STATUS;LINK=LINKSTATE
```

The SESSION STATE will still be listed as RESET, but in the upper right corner of the display, the Linkstate will tell you whether the node has been activated. The Linkstate should be CONNECTED LEVEL 2.

See the *SNA Link/XL Node Manager's Guide* for a complete description of the SNACONTROL STATUS command.

Startup Problems

This section gives some suggestions for troubleshooting startup problems. It is divided into two sections:

- Link and Node Level Problems
- APPC Subsystem Problems

Check that the recommended logging classes have been enabled at the system console through NMMGR. See the *SNA Link/XL Node Manager's Guide* for Hewlett-Packard's recommended logging configuration.

Link and Node Level Problems

To determine whether the link and node have activated, issue the SNACONTROL STATUS command, described in the *SNA Link/XL Node Manager's Guide*. To check the status of a Type 2.1 node, use the *LINK=LINKSTATE* parameter to the SNACONTROL STATUS command. If the display does not say CONNECTED LEVEL 2, the node did not activate.

This section describes some troubleshooting procedures to be used when the SNA link does not activate properly. The *SNA Link/XL Node Manager's Guide* contains more detailed instructions for troubleshooting the SNA link product.

NOTE The APPCCONTROL START command will not activate the SNA node. To activate the SNA node, issue the SNACONTROL START command, described in the *SNA Link/XL Node Manager's Guide*.

1. Validate the configuration file. The NMMGR validation utility cross checks references to SNA nodes and LUs to ensure that they are configured consistently between the SNANODE and APPC branches of NMMGR. The NMMGR validation utility is documented in *Using the Node Management Services Utilities*.
2. Verify that the HP 3000 has been configured properly. Check the following items:
   - **One PU per PSI**. When you run multiple PUs concurrently, each PSI requires a unique PU and each PU requires a unique PSI.
   - **Configuration file name**. You must name the configuration file NMCONFIG.PUB.SYS.
   - **Consistent LU Names, SNA Node Name**. The LUs you configure for use by the APPC subsystem must also be configured in the SNANODE branch of NMMGR. Make sure the LUs you configure for the APPC subsystem are configured on the same

SNA node in the APPC subsystem configuration and the SNA node configuration.

- **Node Type 2.1**. If the HP 3000 is configured to communicate with a peer node, like an IBM AS/400, make sure the `Node Type` in the "SNA Node Configuration: PU Data" screen is configured as `2.1`.
- **ID BLK and ID NUM.** If the HP 3000 is configured to communicate with a peer node, like an IBM AS/400, make sure the `ID BLK` and ID NUM, together, match the Exchange ID configured on the remote system. *This is for switched and non-switched lines.*
- **LU numbers.** For independent LUs, the `LU#` field of the "SNA Node Configuration: LU Data" screen must be left blank.

3. For problems that may involve NMMGR or Node Management Services, give your HP representative a copy of the forms file, `NMMGRF.PUB.SYS`, and the configuration file, `NMCONFIG.PUB.SYS`. With this information, your HP representative may be able to determine whether the problem is related to the configuration of the APPC subsystem or SNA Transport.

NOTE Certain values in the remote system configuration must match values in the HP 3000 configuration file. See Chapter 4 , "APPC Subsystem Configuration," in this manual and the *SNA Link/XL Node Manager's Guide* for more information.

4. Verify the following items in the remote system configuration:
   - **Device configuration.** Perhaps the remote system programmer configured too many or the wrong type of device for a particular line. If two LUs are configured, they should belong to a single PU.
   - **Incorrect macro definition.** A macro could be missing a required operand, or the host system programmer may have specified an invalid operand for a particular macro.

     For switched lines, the `VBUILD` macro should be in a separate member in the VTAM parameter library (SYS1.VTAMLST).
   - **APPC LUs configured in the application subsystem.** APPC LUs may need to be configured in an application subsystem (such as CICS) as well as in VTAM.
5. Check to see whether the remote LU is active. If the node is active, but the `APPCCONTROL STATUS` command shows that the APPC subsystem is not active, make sure that the host has activated the line, the PU, and the LU. If you are communicating with an AS/400, make sure the Line description, Controller description, and Device description have been activated.

6. Check for hardware problems. Check that all hardware is installed properly and working correctly. Make sure that the proper cabling has been installed and connected and that the modems are turned on and working properly. Many problems are related to such things as incompatible modems, bad phone lines and cables and so forth.
7. Check for activity on the line. Make sure the line is active (being polled) and the remote system's teleprocessing subsystem (for example, VTAM) is active. You might need to use a line monitor to determine whether the line is being polled and to determine the addresses being polled. The standard HP line monitor (HP 4953 or 4951) with the SDLC package can be used.

   If a PSI failure has occurred, give the file `NETDMP`*xx*`.PUB.SYS` (*xx* can equal 00 through 99) to your HP representative for analysis.
8. If you have thoroughly checked out all installation and configuration items, and the HP 3000 link still does not activate, perform a link level trace. Link level tracing is described in the *SNA Link/XL Node Manager's Guide*.

   When the problem has been reproduced, turn off link level tracing, and give the trace file to your HP representative for analysis.
9. Print a copy of the output from the NMMAINT program to record the software versions, and give this to your HP representative.
10. If data is being received on the link, but the node is not starting up, examine the SNA Transport node trace. The node trace can be enabled at startup with the `SNACONTROL START` command. For more information, see the *SNA Link/XL Node Manager's Guide*.

APPC Subsystem Problems

To determine whether the APPC subsystem has activated properly, issue the `APPCCONTROL STATUS` command, described in Chapter 2 , "Interactive Control Operator Commands," of this manual. The `Number of Active Sessions` listed in the `APPCCONTROL STATUS` display should be equal to the `Automatically Activated Sessions` value for each configured session type. This section describes some troubleshooting procedures to be used when the APPC subsystem does not activate properly.

1. Validate the configuration file. The NMMGR validation utility cross checks references to SNA nodes and LUs to ensure that they are configured consistently between the SNANODE and APPC branches of NMMGR. The NMMGR validation utility is documented in *Using the Node Management Services Utilities*.

2. Verify that the HP 3000 has been configured properly. Check the following items:

   - **LU session types.** Make sure the independent LU session types are configured to use independent LUs on Type 2.1 nodes. Make sure the dependent LU session types are configured to use dependent LUs on Type 2.0 nodes or on Type 2.1 nodes with dependent LU support.
   - **Local LU Name of independent LU.** The `Local LU Name` field of the "APPC: Independent LU Session Type Data" screen must match the `RMTLOCNAME` in the Device Description on the AS/400.
   - **Fully Qualified Remote LU Name.** The `Fully Qualified Remote LU Name` field of the "APPC: Independent LU Session Type Data" screen must be specified as *`RemoteNetID.RemoteLUName`*, where *`RemoteNetID`* matches the `Local network ID` in the Network Attributes Table on the AS/400, and *`RemoteLUName`* matches the `LCLLOCNAME` in the Device Description on the AS/400.
   - **Mode name**. Make sure the mode name configured for the local LUs matches the mode name for the remote LUs with which they will communicate. On an IBM mainframe, a mode name is associated with an LU in the CICS `MODENAME` operand of the `DFHTCT TYPE=SYSTEM` macro. On an AS/400, a mode name is associated with a device in the `MODE` field of the Device Description.

     Make sure any mode name you specify in the "APPC: Independent LU Session Type Data" screen or the "APPC: Dependent LU Session Type Data" screen is also configured in the "APPC: Select Mode Type" and "APPC: Mode Type Data" screens (unless you are using `MODE0`, which is internally defined).
   - **Parallel Sessions.** If a device on the AS/400 is configured to conduct a single session (`SNGSSN` in the Device Description is `*YES`), make sure the `Parallel Sessions` field of the "APPC: Independent LU Session Type Data" screen is set to `N`.

     If a device on the AS/400 is configured to conduct parallel sessions (`SNGSSN` in the Device Description is `*NO`), make sure the `Parallel Sessions` field of the "APPC: Independent LU Session Type Data" screen is set to `Y`.
   - **Unsolicited BINDs.** Hewlett-Packard recommends that you configure `Unsolicited Binds = N` in the "APPC: Dependent LU Session Type Data" screen. If you configure `Unsolicited Binds = Y`, the HP 3000 will not send an INIT_SELF request to initiate a session. It will simply wait for a BIND from the host. If you specify `Unsolicited Binds = Y`, the CICS configuration on the host must specify `AUTOCONNECT = Y` for the remote LU.

- **Maximum RU size.** The Maximum RU Size (Send and Receive) configured in the "APPC: Mode Type Data" screen, must match the maximum RU size configured for the remote LU. The maximum RU size for LUs on an IBM mainframe is configured in the *RUSIZES* parameter of the MODEENT in the Logmode Table. The maximum RU size for LUs on an AS/400 is configured in the MAXLENRU field of the Mode Description.

NOTE Certain values in the remote system configuration must match values in the HP 3000 configuration file. See Chapter 4 , "APPC Subsystem Configuration," in this manual and the *SNA Link/XL Node Manager's Guide* for more information.

3. If the HP 3000 is configured to communicate with an AS/400, make sure the LCLCTLSSN value is 0 in the Mode Description on the AS/400. Make sure the LCLCTLSSN value is *NO in the Device Descriptions for all AS/400 LUs that will communicate with the HP 3000.

If you have checked your configuration, and the APPC sessions still do not activate, collect the SNA Transport user interface trace and the APPC subsystem internal trace, and give them to your HP representative for analysis. For instruction on turning on APPC subsystem internal tracing, see the description of the APPCCONTROL START command, in Chapter 2 , "Interactive Control Operator Commands," of this manual. For instructions on turning on SNA Transport tracing, see the *SNA Link/XL Node Manager's Guide*.

Run-Time Problems

This section gives some suggestions for troubleshooting run-time problems. It is divided into the following sections:

- Lost Data
- APPC Subsystem Hang
- APPC Subsystem Failure
- Transaction Program Hang
- Remotely Initiated TP Failure
- System Hang or Failure

Lost Data

If data loss occurs at either the HP 3000 side or the remote side, follow these steps:

1. Get a description of what is missing. Get copies of the original document or file sent and the document or file that was received with data missing.
2. Use a protocol analyzer to determine whether data is being transferred.
3. If you suspect that the HP 3000 is causing the data loss, collect the following things and send them to your HP representative for analysis:
   - the SNA Transport user trace (see the *SNA Link/XL Node Manager's Guide*)
   - copies of the document or file sent and the document or file received with data missing
   - the APPC subsystem internal trace
   - the `NMMAINT.PUB.SYS` version stamp

APPC Subsystem Hang

After you issue the `APPCCONTROL STOP` command to stop the APPC subsystem (or after a TP calls the `APPCStop` intrinsic), the message `APPC: Subsystem Stopped (APPCLOG 1)` should be logged to the console screen. Issue the `APPCCONTROL STATUS` command to make sure the subsystem has shut down completely. The `APPCCONTROL STATUS` display will list any APPC sessions or transaction programs that are still active after the `APPCCONTROL STOP` command has executed.

If the APPC: Subsystem Stopped message does not appear, or if the APPCCONTROL STATUS display indicates that APPC sessions are still active, then the APPC subsystem is probably hung. Follow the procedure below to collect all needed diagnostic information and bring the subsystem down. Note that if user and internal tracing were not enabled at the time the problem occurred, you will have to recreate the problem with both traces enabled in order to get tracing information.

1. Collect the following information for analysis by your HP representative:
   - The APPC Object dump. Use the APPCCONTROL DUMP command, described in Chapter 2 , "Interactive Control Operator Commands."
   - The version stamps from running NMMAINT.PUB.SYS.
   - The NMS log file (NMLG*xxxx*). The current log file will be locked, so issue the SWITCHNMLOG command to close the current log file and start a new one.
   - Any console messages you receive.
2. Try to force the APPC subsystem to go down. The APPCCONTROL STOP command allows you to choose among three StopTypes. The highest priority StopType is K[ILL], followed by P[ROTOCOL], and Q[UIESCE] (the default). Try issuing APPCCONTROL STOP;TYPE=P[ROTOCOL]. If the APPC subsystem still remains active, issue APPCCONTROL STOP;TYPE=K[ILL].
3. If you do not see the APPC: Subsystem Stopped message, issue the APPCCONTROL STATUS command to see which sessions are still active. Issue the APPCCONTROL STOPSESSION command for each active session, then issue the APPCCONTROL STOP command again.
4. If the APPC subsystem still does not come down, try to bring down the line from the host side. (Be sure to check with other users of the line.)
5. Collect the user and internal trace files.
6. Submit all diagnostic information to your HP representative.

APPC Subsystem Failure

If the APPC subsystem comes down unexpectedly, follow the procedure below to collect the data your HP representative will need to analyze the problem. Note that if user and internal tracing were not enabled at the time the problem occurred, you will have to recreate the problem with both traces enabled in order to collect the tracing information.

1. Issue the SNACONTROL STATUS command to check the activity of SNA Transport. Create a copy of the display or any error messages that the command generates.

2. Find the APPC dump if there is one. APPC dumps are created automatically in certain cases. The following message should appear on the console, containing the name of the dump file:

   ```
   APPC:  Object dumped to file APPCDP03  (APPCLOG 29).
   ```

   The file will be called APPCDP*xx*.APPC.SYS where *xx* is a number from 00 through 49. To be sure you collect the most recent file, store off all files with the name APPCDP*xx*.

3. Collect the following information for analysis by your HP representative:

   - The user trace files and the APPC internal trace files.
   - The version stamps from running NMMAINT.PUB.SYS.
   - The NMS log file. The name will be NMLG*xxxx*.PUB.SYS. The current log file will be locked, so issue the command SWITCHNMLOG to close the current log file and start a new one.

4. Submit the data you collected to your HP representative for analysis.

Transaction Program Hang

When a user reports a deadlocked or hung transaction program (a program that does not return control to the user), the problem could be caused by the program or by the APPC subsystem. Follow the procedure below:

1. Record any messages the user received.
2. Check the error and logging messages.
3. Issue the APPCCONTROL STATUS command. If the subsystem is not active, see "APPC Subsystem Failure," earlier in this chapter. If the subsystem is still active, find out which session the deadlocked TP is using.
4. Try to bring down the session with the APPCCONTROL STOPSESSION command. If the session does not come down, see "APPC Subsystem Hang," earlier in this chapter.

Remotely Initiated TP Failure

If a remotely initiated TP that worked on a previous version of APPC fails after you install a new version, get a user trace of the TP and check the status info value returned on the `MCGetAllocate` intrinsic.

If the status info value is -1008, do the following things:

1. Modify the local TP so that it passes the *`LocalTPName`* as an input parameter to the `MCGetAllocate` intrinsic.

   On older versions of the APPC subsystem, the *`LocalTPName`* was an output parameter. However, on the Node Type 2.1 version of the APPC subsystem, the *`LocalTPName`* is an input parameter.

2. Make sure the *`LocalTPName`* parameter of the `MCGetAllocate` intrinsic matches the *`LocalTPName`* parameter of the `TPStarted` intrinsic.

3. Use NMMGR to configure the *`LocalTPName`* as a remotely initiated TP. See Chapter 4 , “APPC Subsystem Configuration,” in this manual for more information.

System Hangs or Failures

In the event of a system hang or failure, perform the following steps:

1. Take a memory dump, format the dump and store off the spool file to tape.

2. Store the following files off to tape and submit them, along with the memory dump, to your HP representative for analysis.

   - The APPC subsystem internal trace and the LU 6.2 product user trace.
   - The version stamps from running `NMMAINT.PUB.SYS`.
   - The NMS log file (`NMLG`*`xxxx`*`PUB.SYS`). The current log file will be locked, so issue the command `SWITCHNMLOG` to close the current log file and start a new one.

Submitting an SR

Listed below are some guidelines for submitting a service request (SR). Some of the guidelines refer to Node Management Services programs (NMDUMP, NMMAINT, NMMGR, etc.). For more information about these programs, see *Using the Node Management Services Utilities*.

Common Information

For any SR, include this common information, where applicable:

- A characterization of the problem. Describe the events leading up to and including the problem. Attempt to describe the source of the problem. Describe the symptoms of the problem.

  Your characterization should include MPE commands, communication subsystem commands, job streams, result codes and messages, and data that can reproduce the problem.

  List the context and environment in which the message occurred. Prepare copies of the HP 3000 system console and workstation information.

  Give a brief history of your site, including when your LU 6.2 product was installed, the circumstances that usually produce problems, other data communications products installed, and any other products you were using when the problem arose.

- Obtain the version, update, and fix information for all software by running NMMAINT. This allows Hewlett-Packard to determine whether the problem is already known and if the correct software is installed at your site.

- Run NMDUMP to format the NM log file (`NMLG`*`nnnn`*`.PUB.SYS`) that was active when the problem occurred. You may need to issue the MPE command `SWITCHNMLOG` to free the NM log file. For more information, see the *SNA Link/XL Node Manager's Guide*. Inspect the formatted output and try to locate errors. Prepare the formatted output and a copy of the log file for your HP representative to analyze.

- Prepare a listing of the configuration file you are using for your HP representative to analyze. Inspect the output and try to locate errors.

- Try to determine the general area, within the software, where you think the problem exists. If you think the problem is caused by an SNA service, refer to the *Node Manager's Guide* for that service, and follow the guidelines for gathering information for problems.

- Document your interim, or “workaround,” solution. The cause of the problem can sometimes be found by comparing the circumstances in which it occurs with the circumstances in which it does not occur.
- Save copies of any SNA link or SNA service trace files that were active when the problem occurred.
- If the problem involves NMMGR, give a copy of `NMMGRF.PUB.SYS` to your HP representative.
- If a system failure has occurred, take a full memory dump.
- If a fatal link error has occurred, an error message will appear on the operator’s console. Be sure to include this error message in the SR exactly as it appears on the operator's console.
- Save copies of all active or in-use trace files for analysis by your HP representative.
- Save copies of all active or in-use log files for analysis by your HP representative.
- Record all intrinsic result codes and messages that appear at the HP 3000 console or at the terminal from which the `APPCCONTROL START` command was issued.
- Always obtain a copy of the host generation for ACF/VTAM and ACF/NCP and for the application subsystems that the APPC subsystem uses. If the HP 3000 is communicating with an AS/400, obtain a copy of the configuration on the AS/400.

  Note any differences between the actual system generation and your expectations. Keep this information for your HP representative.
- Give a copy of `NMCONFIG.PUB.SYS` to your HP representative.

WARNING **The normal checks and limitations that apply to the standard users are bypassed in privileged mode. It is possible for a privileged mode program to destroy file integrity, including the MPE operating system software itself. Hewlett-Packard will investigate and attempt to resolve problems resulting from the use of privileged mode code. This service, which is not provided under the standard Service Contract, is available on a time and materials billing basis. Hewlett-Packard will not support, correct, or attend to any modification of the MPE operating system software.**

A Messages

This appendix lists the following types of messages that can be returned by the APPC subsystem:

- Subsystem startup messages, returned to the terminal from which an APPCCONTROL START command is issued.
- Logging messages, which are returned to the system console, to a disk file, or to a user terminal, depending upon configuration.
- Error messages, which are returned to the terminal from which an APPCCONTROL command is issued. (The types of messages logged depend upon the logging classes that have been configured.)
- Warning messages, which are returned to the terminal from which an APPCCONTROL command is issued.
- Return codes, which are returned by control operator intrinsics to the TPs that call them.
- Configuration validation messages, which can be returned by NMMGR when you validate the configuration file.

After each message in this appendix is a short description of the possible causes of the message and the actions that each message indicates you should take.

Subsystem Startup Messages

Startup messages are returned by the APPC subsystem when the node manager issues an APPCCONTROL START command. These messages are returned to the terminal from which the APPCCONTROL START command is issued.

Note that these messages are informational only, and no action is required.

1 **MESSAGE: Successful completion.**

CAUSE: APPC subsystem activation has begun.

ACTION: None.

2 **MESSAGE: APPC: Opening internal trace file — NMTC#### or user trace file.**

CAUSE: The APPC subsystem internal trace file is being opened.

ACTION: None.

3 **MESSAGE: APPC: Starting SNA/LINK PU Node**

CAUSE: The SNA link is being activated.

ACTION: None.

4 **MESSAGE: APPC: Control Operator start-up completed.**

CAUSE: APPC subsystem startup by the APPCCONTROL START command or the APPCStart intrinsic is now completed.

ACTION: None.

6 **MESSAGE: APPC: Closing internal trace file — NMTC#### or user trace file.**

CAUSE: APPC subsystem is closing the internal trace file.

ACTION: None.

7 **MESSAGE: APPC: Performance tracing is enabled.**

CAUSE: APPC subsystem will gather performance data during run time.

ACTION: None.

8 **MESSAGE: APPC: Performance tracing is disabled.**

CAUSE: APPC subsystem will not gather performance data during run time.

ACTION: None.

Logging Messages

APPC logging messages are returned during subsystem startup and while the subsystem is active.

There are five classes of events that can be logged for the APPC subsystem (SUB0016). They are as follows:

- CLAS0010 records internal errors.
- CLAS0011 records subsystem warnings.
- CLAS0012 records subsystem information messages.
- CLAS0013 records conversation information messages.
- CLAS0014 records session performance statistics.

These messages are returned to the system console, a user terminal, or a disk file, depending upon what you have configured in the NMCONFIG file for logging.

0 **MESSAGE: APPC: Subsystem Started (APPCLOG 0).**

CAUSE: The control operator has completed subsystem startup.

ACTION: No action required.

1 **MESSAGE: APPC: Subsystem Stopped (APPCLOG 1).**

CAUSE: The subsystem has completely shut down.

ACTION: No action required.

4 **MESSAGE: APPC: !**

CAUSE: APPC performance data for APPC subsystem. These messages appear when the APPCCONTROL PERFORMANCE command is issued. See the description of APPCCONTROL PERFORMANCE for examples.

ACTION: No action required.

5 **MESSAGE: APPC: !**

CAUSE: APPC performance data for each active session. These messages appear when the APPCCONTROL PERFORMANCE command is issued. See the description of APPCCONTROL PERFORMANCE for examples.

ACTION: No action required.

6 **MESSAGE: APPC: Session Started, Session ID=! (APPCLOG 6).**

CAUSE: An APPC session with the specified Session ID number has started. This session is now available for use by transaction programs.

ACTION: No action required.

7 **MESSAGE: APPC: Session Stopped, Session ID=!, Reason=! (APPCLOG 7).**

CAUSE: An APPC session with the specified Session ID number has been stopped (unbound) due to the Reason given. (Reasons can include APPC_Kill, APPC_Protocol, APPC_Quiesce, Host_Unbind, Link_Failure, SNA_Error, Convers_Abend, BIS_Error, CTLA_Shutdown, SNA_Kill, SNA_Protocol, SNA_Quiesce, HS_Error, SM_Error, and RM_Error.)

ACTION: If you suspect a problem, note the Reason and contact your HP representative.

8 **MESSAGE: APPC: Bind Rejected, Session ID=!, Sense Data=! (APPCLOG 8)**

CAUSE: The session with the specified Session ID could not be established, because the local LU rejected the BIND from the remote system.

ACTION: Make sure the LUs are correctly configured on the host. If the host configuration is correct, recreate the problem: enable the SNA Transport link trace, and activate the APPC subsystem again. Give the trace to your HP representative.

9 **MESSAGE: APPC: Init-Self Negative Rsp, Session ID=!, Sense Data=! (APPCLOG 9).**

CAUSE: The session with the specified Session ID could not be established, because the host sent a negative response to the INIT_SELF sent by the local LU.

ACTION: There could be a problem with the remote system, or the remote system might be down. The Remote LU Name specified in the APPC subsystem configuration might be incorrect. If the remote system is up, and the Remote LU Name is configured correctly, give the SNA Transport intrinsic trace to your HP representative.

10 **MESSAGE: APPC: SNAOpenUser Failed, Session ID=!, Result=! (APPCLOG 10).**

CAUSE: SNA Transport was unable to establish a session with the specified LU.

ACTION: If SNA Transport is not active, issue the SNACONTROL START command to activate it. Check to make sure the LU or LUs configured for the session type are not being used by another SNA Service. The Result listed in the message is the number of an SNA Transport error message; see the *SNA Link/XL Node Manager's Guid*e for more information.

11 **MESSAGE: APPC: SNA/LINK Error, Session ID=!, Result=! (APPCLOG 11).**

CAUSE: The APPC subsystem has encountered an error in SNA/SDLC Link/XL.

ACTION: Give the APPC subsystem internal trace to your HP representative.

12 **MESSAGE: APPC: Received Notify, Session ID=! (APPCLOG 12).**

CAUSE: The remote system has sent an SNA NOTIFY command over the session with the specified Session ID.

ACTION: This could occur (1) because of a problem with the remote system, (2) because the remote application is not up, (3) because the LU is not configured for use by the remote, or (4) because the Remote LU Name specified in the configuration for the session type is incorrect. If none of these is the case, give the APPC subsystem internal trace to your HP representative.

13 **MESSAGE: APPC: SM Detected Error, Session ID=!, Error=!, ! (APPCLOG 13).**

CAUSE: The APPC subsystem has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

14 **MESSAGE: APPC: Stray Negative Rsp, Session ID=! (APPCLOG 14).**

CAUSE: The APPC subsystem has detected a stray negative response coming from the remote system.

ACTION: No action required; however, if this occurs frequently, it may affect the remote system. If other problems are occurring simultaneously, notify your HP representative.

15 **MESSAGE: APPC: Stray Signal, Session ID=! (APPCLOG 15).**

CAUSE: The APPC subsystem has detected a stray SNA SIGNAL command coming from the remote system.

ACTION: No action required; however, if this occurs frequently, it may affect the remote system. If other problems are occurring simultaneously, notify your HP representative.

16 **MESSAGE: APPC: HS Detected Error, Session ID=!, Error=! (APPCLOG 16).**

CAUSE: The APPC subsystem has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

17 **MESSAGE: APPC: TP started, TPName=!, TCB ID=! (APPCLOG 17).**

CAUSE: A transaction program with the specified transaction program name and Transaction Control Block ID (TCB ID) has been started.

ACTION: No action required.

18 **MESSAGE: APPC: TP stopped, TPName=!, TCB ID=! (APPCLOG 18).**

CAUSE: A transaction program with the specified transaction program name and Transaction Control Block ID (TCB ID) has been stopped.

ACTION: No action required.

19 **MESSAGE: APPC: TP Abend, TPName=!, TCB ID=! (APPCLOG 19).**

CAUSE: A transaction program with the specified transaction program name and Transaction Control Block ID (TCB ID) has been aborted.

ACTION: The transaction program has terminated abnormally. Notify the TP programmer.

20 **MESSAGE: APPC: TP Start Failed: APPC Subsystem not started (APPCLOG 20).**

CAUSE: A transaction program could not be started because the local APPC subsystem is not active.

ACTION: Issue the command APPCCONTROL START to activate the APPC subsystem.

21 **MESSAGE: APPC: TP Local Attach Failed, Reason=! (APPCLOG 21).**

CAUSE: A transaction program could not be started because the local attach failed.

ACTION: Notify your HP representative.

22 **MESSAGE: APPC: TP Not Found in SL, TPName=! (APPCLOG 22).**

CAUSE: The transaction program name given could not be found in the appropriate segmented library file.

ACTION: Notify your HP representative.

23 **MESSAGE: APPC: Remote Attach Failed: TPName=! Not Found (APPCLOG 23).**

CAUSE: The APPC subsystem could not find the requested TP Name in the configuration file.

ACTION: Make sure the TP Name is configured. Make sure the TP Name passed by the remote TP in the allocate request matches the configured TP Name.

24 **MESSAGE: APPC: Remote Attach Failed: TPName=! Creation Failed (APPCLOG 24).**

CAUSE: The APPC subsystem was unable to stream the job to start up the local TP.

ACTION: Make sure the job name is configured correctly through NMMGR and that the configured job file exists. Make sure there are no lockwords on the job file.

25 **MESSAGE: APPC: Conversation Started, TCB ID=!, RCB ID=! (APPCLOG 25).**

CAUSE: A conversation has been started on the transaction program specified by the Transaction Control Block ID (TCB ID), with the appropriate conversation ID (Resource Control Block ID [RCB ID]).

ACTION: No action required.

26 **MESSAGE: APPC: Conversation Stopped, TCB ID=!, RCB ID=! (APPCLOG 26).**

CAUSE: A conversation has been stopped on the transaction program specified by the Transaction Control Block ID (TCB ID), with the appropriate conversation ID (Resource Control Block ID [RCB ID]).

ACTION: No action required.

27 **MESSAGE: APPC: Conversation Abend, TCB ID=!, RCB ID=! (APPCLOG 27).**

CAUSE: A conversation has been aborted on the transaction program specified by the Transaction Control Block ID (TCB ID), with the appropriate conversation ID (Resource Control Block ID [RCB ID]).

ACTION: A conversation has terminated abnormally. Notify the TP programmer.

28 **MESSAGE: APPC: Presentation Services Detected Error, Error=! (APPCLOG 28).**

CAUSE: The APPC module PS (Presentation Services) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

29 **MESSAGE: APPC: Object dumped to file ! (APPCLOG 29).**

CAUSE: The module COPR (Control Operator) has dumped the APPC Object to the file specified.

ACTION: This file and the APPC subsystem internal trace file should be given to your HP representative.

30 **MESSAGE: APPC: Half Session Creation Error, SUBSYS=!, INFO=! (APPCLOG 30).**

CAUSE: The APPC subsystem was unable to create the Half Session process

ACTION: Contact your HP representative.

31 **MESSAGE: Resource Manager Creation Error, NMError=! (APPCLOG 31).**

CAUSE: The APPC subsystem was unable to create the Resource Manager process.

ACTION: Verify that the program file RM.APPC.SYS exists

32 **MESSAGE: APPC: Session Manager Creation Error, NMError=! (APPCLOG 32).**

CAUSE: The APPC subsystem was unable to create the Session Manager process.

ACTION: Verify that the program file SM.APPC.SYS exists.

33 **MESSAGE: APPC: COPRTP Creation Error, NMError=! (APPCLOG 33).**

CAUSE: The APPC subsystem was unable to create the COPRTP process.

ACTION: Verify that the program file COPRTP.APPC.SYS exists.

34 **MESSAGE: APPC: CNOSTP Creation Error, NMError=! (APPCLOG 34).**

CAUSE: The APPC subsystem was unable to create the CNOSTP process.

ACTION: Verify that the program file CNOSTP.APPC.SYS exists.

35 **MESSAGE: APPC: Control Operator Internal Error, Error=! (APPCLOG 35).**

CAUSE: The APPC module COPR (Control Operator) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

36 **MESSAGE: APPC: Resource Manager Internal Error, Error=! (APPCLOG 36).**

CAUSE: The APPC module RM (Resource Manager) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

37 **MESSAGE: APPC: Session Manager Internal Error, Error=! (APPCLOG 37).**

CAUSE: The APPC module SM (Session Manager) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

38 **MESSAGE: APPC: Half Session Internal Error, Error=! (APPCLOG 38).**

CAUSE: The APPC module HS (Half Session) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

39 **MESSAGE: APPC: COPRTP Internal Error, Error=! (APPCLOG 39).**

CAUSE: The APPC module COPRTP (Control Operator TP) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

40 **MESSAGE: APPC: CNOSTP Internal Error, Error=! (APPCLOG 40).**

CAUSE: The APPC module CNOSTP (Change-Number-Of-Sessions TP) has detected an internal error.

ACTION: Give the APPC subsystem internal trace to your HP representative.

41 **MESSAGE: APPC: Control Operator Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 41).**

CAUSE: The Control Operator encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS values, and notify your HP representative.

42 **MESSAGE: APPC: Resource Manager Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 42).**

CAUSE: The Resource Manager encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS values, and notify your HP representative.

43 **MESSAGE: APPC: Session Manager Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 43).**

CAUSE: The Session Manager encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS< values, and notify your HP representative.

44 **MESSAGE: APPC: Half Session Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 44).**

CAUSE: Half Session encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS values, and notify your HP representative.

45 **MESSAGE: APPC: COPRTP Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 45).**

CAUSE: The COPRTP encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS values, and notify your HP representative.

46 **MESSAGE: APPC: CNOSTP Non-Fatal Error: INFO=!, SUBSYS=! (APPCLOG 46).**

CAUSE: The CNOSTP encountered a non-fatal error. The INFO value in the message is the number of a warning or informational message. The SUBSYS value indicates which subsystem generated the message. If SUBSYS=732, the message was generated by the APPC subsystem.

ACTION: Note the INFO and SUBSYS values, and notify your HP representative.

47 **MESSAGE: APPC: PRIMARY STATUS: INFO=!, SUBSYS=! (APPCLOG 47).**

CAUSE: This message follows another message and supplies the subsystem number and the return code from the internal intrinsic that failed.

ACTION: See the message that was logged just before this one for more information.

48 **MESSAGE: APPC: SECONDARY STATUS: INFO=!, SUBSYS=! (APPCLOG 48).**

CAUSE: This message follows another message and supplies the subsystem number and the return code from the MPE XL intrinsic that failed.

ACTION: See the message that was logged just before this one for more information

49 **MESSAGE: APPC: ! ! Limits Succeeded: OK AS SPECIFIED (APPCLOG 49).**

CAUSE: The requested session limits have been accepted as specified; they were not negotiated.

ACTION: No action required.

50 **MESSAGE: APPC: ! ! Limits Succeeded: OK AS NEGOTIATED (APPCLOG 50).**

CAUSE: The requested session limits have been negotiated successfully.

ACTION: No action required.

51 **MESSAGE: APPC: ! ! Limits Failed: UNRECOGNIZED MODE NAME (APPCLOG 51).**

CAUSE: The mode name for the remote LU does not match the mode name for the local LU.

ACTION: Check the APPC subsystem configuration to verify that the mode name configured for the local LU is the same mode name configured for the remote LU.

52 **MESSAGE: APPC: ! ! Limits Failed: MODE SESSION LIMIT CLOSED (APPCLOG 52).**

CAUSE: The session limit on the IBM AS/400 was set to 0, and the APPC subsystem attempted to raise the session limit.

ACTION: Start the mode on the AS/400.

53 **MESSAGE: APPC: ! ! Limits Failed: COMMAND RACE REJECT (APPCLOG 53).**

CAUSE: CNOS requests from both the local and the remote systems attempted to change the session limits at the same time. One or both requests failed.

ACTION: Reissue the APPCCONTROL SESSIONS command. If this error occurred at subsystem startup, restart the APPC subsystem.

54 **MESSAGE: APPC: ! ! Limits Failed: TRANSACTION FAILURE (APPCLOG 54).**

CAUSE: Internal error in CNOSTP or COPRTP.

ACTION: Give the internal trace to your HP representative.

55 **MESSAGE: APPC: ! ! Limits Failed: SESSION OUTAGE OCCURRED (APPCLOG 55).**

CAUSE: The APPC session ended unexpectedly during a CNOSTP or COPRTP conversation.

ACTION: Use the APPCCONTROL STATUS command, specifying STYPE=SNASVCMG, to verify that a CNOS session is active.

56 **MESSAGE: APPC: ! ! Limits Failed: NO CNOS SESSION AVAILABLE (APPCLOG 56).**

CAUSE: No sessions are available for the CNOSTP.

ACTION: Restart the APPC subsystem. Issue the APPCCONTROL STATUS command, specifying STYPE=SNASVCMG, to verify that CNOS sessions are active.

57 **MESSAGE: APPC: ! ! Limits Failed: CHANGE SESSIONS FAILURE (APPCLOG 57).**

CAUSE: Internal APPC subsystem error.

ACTION: Give the internal trace to your HP representative.

58 **MESSAGE: APPC: ! ! Limits Failed: RESOURCE FAILURE NO RETRY (APPCLOG 58).**

CAUSE: A CNOSTP or COPRTP conversation failed. The session might have failed, or the TP on the remote system might have encountered an error and deallocated the conversation.

ACTION: Reissue the APPCCONTROL SESSIONS command, or restart the APPC subsystem.

59 **MESSAGE: APPC: ! ! Limits Failed: UNKNOWN ERROR (APPCLOG 59).**

CAUSE: An error has occurred in the CNOSTP or COPRTP.

ACTION: Give the APPC subsystem internal trace to your HP representative.

60 **MESSAGE: APPC: Stype=!, Mode=!, Limit=!, Source=!, Target=! (APPCLOG 60).**

CAUSE: A CNOS (Change-Number-Of-Sessions) transaction completed successfully. This message follows message APPCLOG 49 or APPCLOG 50 and supplies more information.

ACTION: No action required.

61 **MESSAGE: APPC: Stype=!, Mode=! (APPCLOG 61).**

CAUSE: A CNOS (Change-Number-Of-Sessions) transaction failed. This message follows a message from APPCLOG 51 through APPCLOG 59 and supplies more information.

ACTION: See the message that was logged just before this one for more information.

Error Messages

Error messages may be returned by the APPC subsystem when the node manager issues an APPCCONTROL command. These messages are returned to the terminal from which the command is issued. Some error messages are generated by the NMDUMP utility, which is used to format the APPC subsystem internal trace and the LU 6.2 API/XL user trace.

NOTE Some syntax checking is done by the Native Mode Command Interpreter, so some syntax errors might generate MPE XL Command Interpreter error messages as well.

0 **MESSAGE: SUCCESSFUL COMPLETION.**

CAUSE:

ACTION:

901 **MESSAGE: Invalid character selection. (APPCERR 901)**

CAUSE: An invalid character has been selected when choosing a trace type or a control block index. The APPC subsystem trace formatter reprompts for a valid selection.

ACTION: When selecting a trace type, enter an integer of 1 (internal trace) or 2 (user trace). When selecting a control block index, enter an integer from 0 through 99999.

902 **MESSAGE: Invalid trace selection. (APPCERR 902)**

CAUSE: An invalid trace formatting type has been selected. The APPC subsystem trace formatter reprompts for a valid selection.

ACTION: Select either 1 (internal trace) or 2 (user trace).

903 **MESSAGE: APPC is unable to write to $STDLIST file. FS Error: errornum. (APPCERR 903)**

CAUSE: The APPC trace formatter failed to write to the $STDLIST file passed to it from NMDUMP.

ACTION: Check that $STDLIST, if redirected when running NMDUMP, is equated properly to an existing MPE file or that the NEW option is specified in the RUN command. See the *MPE XL Intrinsics Reference Manual* for the exact meaning of the FWRITE error indicated by *errornum*.

904 **MESSAGE: APPC is unable to read from $STDIN file. FS Error: errornum. (APPCERR 904)**

CAUSE: The APPC trace formatter failed to read from the $STDIN file passed to it from NMDUMP.

ACTION: Check that $STDIN, if redirected when running NMDUMP, is equated properly to an existing MPE text file. See the *MPE XL Intrinsics Reference Manual* for the exact meaning of the FREAD error indicated by *errornum*.

905 **MESSAGE: APPC failed to read message #messagenum in set #setnum of NMCAT.PUB.SYS. (APPCERR 905) Intrinsic Error Number: errornum.**

CAUSE: The APPC subsystem failed to read the specified message in the specified set of NMCAT.PUB.SYS.

ACTION: The intrinsic error number indicates the CATREAD intrinsic return code for failure to read the catalog. See the *MPE XL Intrinsics Reference Manual* to determine the meaning of the CATREAD failure.

906 **MESSAGE: Invalid TCB index selection. (APPCERR 906)**

CAUSE: A TCB index bigger than the allowable maximum value has been selected. A TCB index cannot be greater than a five-digit integer. The APPC subsystem trace formatter reprompts for a new value.

ACTION: Select a TCB index from 0 through 99999.

907 **MESSAGE: Invalid RCB index selection. (APPCERR 907)**

CAUSE: An RCB index bigger than the allowable maximum value has been selected. An RCB index cannot be greater than a five-digit integer. The APPC subsystem trace formatter reprompts for a new value.

ACTION: Select an RCB index from 0 through 99999.

908 **MESSAGE: Invalid SCB index selection. (APPCERR 908)**

CAUSE: An SCB index bigger than the allowable maximum value has been selected. An SCB index cannot be greater than a five-digit integer. The APPC subsystem trace formatter reprompts for a new value.

ACTION: Select an SCB index from 0 through 99999.

1001 **MESSAGE: MISSING REQUIRED PARAMETER. (APPCERR 1001)**

CAUSE: The APPCCONTROL command you typed is missing a necessary parameter.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1002 **MESSAGE: INVALID APPCCONTROL COMMAND OR PARAMETER. (APPCERR 1002)**

CAUSE: Incorrect command or parameter specified.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1003 **MESSAGE: PARAMETER STRING LENGTH TOO LONG. (APPCERR 1003)**

CAUSE: Parameter string exceeded specified length.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1004 **MESSAGE: TOO MANY PARAMETERS SPECIFIED. (APPCERR 1004)**

CAUSE: An APPCCONTROL command was issued with too many parameters.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1005 **MESSAGE: SYNTAX ERROR — TYPE 'APPCCONTROL HELP' FOR SYNTAX. (APPCERR 1005)**

CAUSE: Incorrect syntax of command.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1006 **MESSAGE: REDUNDANT PARAMETER. (APPCERR 1006)**

CAUSE: A parameter has been used more than once.

ACTION: Retype the command, leaving out the duplicated parameter.

1007 **MESSAGE: MISSING PARAMETER AFTER EQUAL SIGN. (APPCERR 1007)**

CAUSE: A parameter was specified, but no value was assigned to it, or there was a syntax error in the command line.

ACTION: Retype the command with the assigned value, or check the syntax by typing APPCCONTROL HELP.

1008 **MESSAGE: MISSING OR INVALID DELIMITER — USE “;” AS DELIMITER. (APPCERR 1008)**

CAUSE: Incorrect syntax of command.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP> or by looking in Chapter 2 , “Interactive Control Operator Commands,” of this manual.

1009 **MESSAGE: PARAMETER OUT OF BOUNDS IN PROGRAMMATIC INTERFACE. (APPCERR 1009)**

CAUSE: Programmatic interface detected a parameter out of bounds.

ACTION: Call your HP representative.

1010 **MESSAGE: NM CAPABILITY REQUIRED. (APPCERR 1010)**

CAUSE: A user without NM capability issued a command that requires NM capability.

ACTION: Assign NM capability to the user, or log on to another user that has NM capability

1011 **MESSAGE: INVALID ERROPT — [SYS OR SUB]. (APPCERR 1011)**

CAUSE: Incorrect value assigned to the *ERROPT* parameter of the APPCCONTROL START command.

ACTION: Retype the command with the correct ERROPT value. If you need help, check the syntax of this command by typing APPCCONTROL HELP, or refer to Chapter 2 , "Interactive Control Operator Commands," of this manual.

1012 **MESSAGE: INVALID TRACEON — [YES OR NO]. (APPCERR 1012)**

CAUSE: An invalid value was specified in the *TRACEON* parameter of the APPCCONTROL START command.

ACTION: Type APPCCONTROL HELP for information about APPCCONTROL commands, or see Chapter 2 , "Interactive Control Operator Commands," of this manual.

1013 **MESSAGE: INVALID TFILESIZE — [0 TO 32767]. (APPCERR 1013)**

CAUSE: Incorrect value assigned to the *TFILESIZE* parameter of the APPCCONTROL START command.

ACTION: Retype the command with a valid TFILESIZE value. If you need help, check the syntax of this command by typing APPCCONTROL HELP, or refer to Chapter 2 , "Interactive Control Operator Commands," of this manual.

1014 **MESSAGE: INVALID PERFORMANCEON — [YES or NO]. (APPCERR 1014)**

CAUSE: An incorrect value was specified in the *PERFORMANCEON* parameter of the APPCCONTROL START command.

ACTION: Specify YES or NO. Check the syntax of this command by typing APPCCONTROL HELP, or see Chapter 2 , "Interactive Control Operator Commands," of this manual.

1015 **MESSAGE: INVALID OR UNCONFIGURED SESSION TYPE NAME (APPCERR 1015)**

CAUSE: Session type name specified in the *STYPE* parameter of the APPCCONTROL SESSIONS, APPCCONTROL PERFORMANCE, or APPCCONTROL STATUS command is not configured or is not recognized.

ACTION: Determine which session types are configured by issuing the APPCCONTROL STATUS command. If necessary, shut down the APPC subsystem to update the configuration file.

1016 **MESSAGE: INVALID SESSION LIMIT OR SUBSYSTEM'S LIMIT EXCEEDED (APPCERR 1016)**

CAUSE: Session limit specified in the *LIMIT* parameter of the APPCCONTROL SESSIONS command is greater than the number currently allowed.

ACTION: You could receive this message because you tried to exceed the maximum number of sessions that the APPC subsystem can support, or because you exceeded the number of sessions configured for this session type. Check the number of active sessions and the configured session limit for the session type by issuing the APPCCONTROL STATUS command.

1017 **MESSAGE: INVALID STOP_TYPE — [K, P, OR Q]. (APPCERR 1017)**

CAUSE: Incorrect value assigned to the *TYPE* parameter of the APPCCONTROL STOP command.

ACTION: Check the syntax of this command by typing APPCCONTROL HELP or by looking in Chapter 2 , "Interactive Control Operator Commands," of this manual.

1030 **MESSAGE: VERSION NUMBER MISMATCH DETECTED. (APPCERR 1030)**

CAUSE: Type APPCCONTROL VERSION to verify the version mismatch. Call your HP representative.

ACTION: Version mismatch detected between the modules.

1031 **MESSAGE: MISSING REQUIRED APPC SUBSYSTEM MODULE (APPCERR 1031)**

CAUSE: A required file has not been installed on the subsystem.

ACTION: Issue the APPCCONTROL VERSION command to determine which software modules are installed on the system, and save the output to a file. Call your HP representative.

1032 **MESSAGE: INVALID APPC SUBSYSTEM MODULES INSTALLED (APPCERR 1032)**

CAUSE: APPC subsystem detected NM return code 105 during version checking.

ACTION: Issue the APPCCONTROL VERSION command, and save the output to a file. Call your HP representative.

1034 **MESSAGE: INVALID APPC SUBSYSTEM CONFIGURATION. (APPCERR 1034)**

CAUSE: The APPC subsystem detected an error trying to validate the configuration file during subsystem startup.

ACTION: Run NMMGR and validate the APPC subsystem. Note and correct any errors. If the problem persists, contact your HP representative.

1035 **MESSAGE: UNABLE TO OPEN NMCONFIG FILE. (APPCERR 1035)**

CAUSE: The APPC subsystem is unable to open the NMCONFIG.PUB.SYS file.

ACTION: Restart the APPC subsystem. If the same message is returned, call your HP representative.

1036 **MESSAGE: UNABLE TO CLOSE NMCONFIG FILE. (APPCERR 1036)**

CAUSE: The APPC subsystem is unable to close the NMCONFIG.PUB.SYS file.

ACTION: Restart the APPC subsystem. If the problem still exists, call your HP representative.

1037 **MESSAGE: CONFIGURATION ERROR. (APPCERR 1037)**

CAUSE: The APPC subsystem detected an error return code from the NMCONF call of Node Management Services.

ACTION: Check the configuration of the APPC subsystem.

1038 **MESSAGE: UNABLE TO OPEN SUBSYSTEM CATALOG FILE. (APPCERR 1038)**

CAUSE: The APPC subsystem is unable to open the catalog file CATAPPC.APPC.SYS.

ACTION: Type LISTF CATAPPC.APPC.SYS to see if the file exists. Call your HP representative.

1039 **MESSAGE: UNABLE TO CLOSE SUBSYSTEM CATALOG FILE. (APPCERR 1039)**

CAUSE: The APPC subsystem is unable to close the catalog file CATAPPC.APPC.SYS.

ACTION: Type LISTF CATAPPC.APPC.SYS to see if the file exists. Call your HP representative.

1040 **MESSAGE: NMLOG FILE OPEN FAILED. (APPCERR 1040)**

CAUSE: The APPC subsystem is unable to open the Node Management Log File.

ACTION: Check the logging class configuration in NMMGR.

1041 **MESSAGE: NMLOG FILE CLOSE FAILED. (APPCERR 1041)**

CAUSE: The APPC subsystem is unable to close the Node Management Log File.

ACTION: Check the logging class configuration in NMMGR.

1042 **MESSAGE: TRACE FILE OPEN FAILED. (APPCERR 1042)**

CAUSE: The APPC subsystem is unable to create the APPC internal trace file.

ACTION: Restart the APPC subsystem. If the problem still exists, call your HP representative.

1043 **MESSAGE: TRACE FILE CLOSE FAILED. (APPCERR 1043)**

CAUSE: The APPC subsystem is unable to close the APPC internal trace file.

ACTION: Restart the APPC subsystem. If the problem still exists, call your HP representative.

1044 **MESSAGE: UNABLE TO CREATE APPC DUMP FILE. (APPCERR 1044)**

CAUSE: A file could not be created to dump the APPC data structures.

ACTION: Verify that there is enough disk space to build the dump file, and then retry the command. If the problem still exists, call your HP representative.

1045 **MESSAGE: UNABLE TO OPEN APPC DUMP FILE. (APPCERR 1045)**

CAUSE: The APPC dump file could not be opened.

ACTION: Verify that the capabilities for the APPC.SYS group and account are correct and that there is enough disk space on the system. Then retry the command. If the problem still exists, call your HP representative.

1046 **MESSAGE: UNABLE TO CREATE RESOURCE MANAGER AT SUBSYSTEM STARTUP. (APPCERR 1046)**

CAUSE: The APPC subsystem detected an error trying to create the Resource Manager process during subsystem startup.

ACTION: Verify that the program file RM.APPC.SYS exists and that the capabilities and access privileges for the APPC.SYS group and account are correct. If the problem still exists, contact your HP representative.

1047 **MESSAGE: UNABLE TO CREATE COPR TP AT SUBSYSTEM STARTUP. (APPCERR 1047)**

CAUSE: The APPC subsystem detected an error trying to create the Control Operator TP process during subsystem startup.

ACTION: Verify that the program file COPRTP.APPC.SYS exists and that the capabilities and access privileges for the APPC.SYS group and account are correct. If the problem still exists, contact your HP representative.

1048 **MESSAGE: UNABLE TO CREATE CNOS TP AT SUBSYSTEM STARTUP. (APPCERR 1048)**

CAUSE: The APPC subsystem detected an error trying to create the Change Number of Sessions (CNOS) TP process during subsystem startup.

ACTION: Verify that the program file CNOSTP.APPC.SYS exists and that the capabilities and access privileges for the APPC.SYS group and account are correct. If the problem still exists, contact your HP representative.

1049 **MESSAGE: RESOURCE MANAGER DETECTED ERROR DURING SUBSYSTEM STARTUP. (APPCERR 1049)**

CAUSE: The Resource Manager detected an error during subsystem startup.

ACTION: Restart the subsystem. If the error still exists, call your HP representative.

1050 **MESSAGE: COPRTP DETECTED ERROR DURING SUBSYSTEM STARTUP. (APPCERR 1050)**

CAUSE: The Control Operator TP detected an error during subsystem startup.

ACTION: Restart the subsystem. If the error still exists, call your HP representative.

1051 **MESSAGE: CNOSTP DETECTED ERROR DURING SUBSYSTEM STARTUP. (APPCERR 1051)**

CAUSE: The Change Number of Sessions (CNOS) TP detected an error during subsystem startup.

ACTION: Restart the subsystem. If the error still exists, call your HP representative.

1060 **MESSAGE: RESOURCE MANAGER ERROR — SESSION DEACTIVATION FAILED. (APPCERR 1060)**

CAUSE: The APPC subsystem detected an error while trying to deactivate a session after the APPCCONTROL SESSIONS command was issued or the APPCSessions intrinsic was called.

ACTION: Call your HP representative.

1062 **MESSAGE: UNABLE TO DUMP APPC TO DUMP FILE. (APPCERR 1062)**

CAUSE: The APPCCONTROL DUMP command did not execute.

ACTION: Retry the command. If the problem still exists, call your HP representative.

1063 **MESSAGE: RESOURCE MANAGER ERROR — CHANGE SESSIONS FAILED. (APPCERR 1063)**

CAUSE: The APPC subsystem detected an internal error.

ACTION: Call your HP representative.

1064 **MESSAGE: RESOURCE MANAGER ERROR — SUBSYSTEM SHUTDOWN FAILED. (APPCERR 1064)**

CAUSE: The APPC subsystem detected an internal error while trying to shut down the APPC subsystem after the APPCCONTROL STOP command was issued.

ACTION: Call your HP representative.

1065 **MESSAGE: CONTROL OPERATOR (COPR) INTERNAL ERROR. (APPCERR 1065)**

CAUSE: The COPR TP detected a run-time error.

ACTION: Call your HP representative.

1066 **MESSAGE: APPC OBJECT ACCESS ERROR. (APPCERR 1066)**

CAUSE: The Control Operator was unable to access the APPC KSO.

ACTION: Call your HP representative.

1067 **MESSAGE: UNABLE TO OPEN $STDIN. (APPCERR 1067)**

CAUSE: The Control Operator was unable to open $STDIN to read from the terminal.

ACTION: Call your HP representative.

1068 **MESSAGE: UNABLE TO OPEN $STDLIST. (APPCERR 1068)**

CAUSE: The Control Operator was unable to open $STDLIST to write output to the terminal.

ACTION: Call your HP representative.

1071 **MESSAGE: APPC PERFORMANCE TRACING NOT ENABLED. (APPCERR 1071)**

CAUSE: The APPCCONTROL PERFORMANCE command was issued when APPC performance tracing was not active.

ACTION: Issue the APPCCONTROL PERFORMANCEON command to enable the gathering of performance statistics before issuing the APPCCONTROL PERFORMANCE command.

1072 **MESSAGE: APPC SUBSYSTEM NOT ACTIVE. (APPCERR 1072)**

CAUSE: The command issued cannot be executed because the APPC subsystem was not started.

ACTION: If you wish to execute the command, do so after starting up the APPC subsystem with the APPCCONTROL START command.

1073 **MESSAGE: APPC SUBSYSTEM PENDING ACTIVE. (APPCERR 1073)**

CAUSE: The command issued cannot be executed because the APPC subsystem is in the progress of starting up.

ACTION: Check the status of the APPC subsystem and reenter your command when the subsystem is active. If problems still exist, call your HP representative.

1074 **MESSAGE: APPC SUBSYSTEM ALREADY ACTIVE. (APPCERR 1074)**

CAUSE: The APPCCONTROL START command was issued (or the APPCStart intrinsic was called) after the APPC subsystem had already been activated.

ACTION: No action necessary.

1075 **MESSAGE: APPC SUBSYSTEM SHUTTING DOWN. (APPCERR 1075)**

CAUSE: An APPCCONTROL START or an APPCCONTROL SESSIONS command was issued after initiation of subsystemn shutdown.

ACTION: Check the status of the subsystem and wait until the subsystem has shut down completely. If the command entered was APPCCONTROL START, re-enter the command. If the command entered was APPCCONTROL SESSIONS, restart the subsystem with APPCCONTROL START, and then reissue the APPCCONTROL SESSIONS command.

1076 **MESSAGE: INVALID OR INACTIVE SESSION TYPE REQUESTED. (APPCERR 1076)**

CAUSE: The command issued cannot be executed because the requested session type is not in ACTIVE state.

ACTION: Use the APPCCONTROL STATUS command to verify the state of the session type. Use APPCCONTROL SESSIONS to activate the session type if it is in a NO SESS state.

Warning Messages

Warning messages may be returned by the APPC subsystem when the node manager issues an APPCCONTROL command. These messages are returned to the terminal from which the command is issued.

1201 **MESSAGE: SESSION LIMIT EXCEEDED, REMAINING SESSION IGNORED. (APPCWARN 1201)**

CAUSE: The total session limit of the subsystem has been reached before the end of the configuration file. Remaining session types are initialized in the APPC Object with SESSION_LIMIT = 0.

ACTION: No action is necessary if you don't use those session types that are not initialized in the APPC Object. Use the APPCONTROL SESSIONS command to rearrange the session limit. Shut down the subsystem and update the configuration file when necessary.

1202 **MESSAGE: CATREAD FAILED. (APPCWARN 1202)**

CAUSE: Detected non-zero result code during CATREAD call. Unable to continue printing the startup messages.

ACTION: Make sure the subsystem is started up correctly. If this problem occurs the next time the subsystem is started, call your HP representative.

1203 **MESSAGE: NO SESSION TYPES CONFIGURED FOR PARALLEL SESSIONS. (APPCWARN 1203)**

CAUSE: The APPCCONTROL STATUS command was issued with STYPE=SNASVCMG to display the CNOS sessions. However, no session types are configured for parallel sessions, so no CNOS sessions can be displayed.

ACTION: No action required.

1204 **MESSAGE: APPC INTERNAL STATE TRACING ALREADY ENABLED. (APPCWARN 1204)**

CAUSE: APPCCONTROL TRACEON command was issued when APPC subsystem internal state tracing was already enabled.

ACTION: No action required.

1205 **MESSAGE: APPC INTERNAL STATE TRACING ALREADY DISABLED. (APPCWARN 1205)**

CAUSE: An APPCCONTROL TRACEOFF command was issued when APPC subsystem internal state tracing was already disabled

ACTION: No action required.

1206 **MESSAGE: APPC PERFORMANCE TRACING ALREADY ENABLED. (APPCWARN 1206)**

CAUSE: An APPCCONTROL PERFORMANCEON command was issued when APPC subsystem performance tracing was already enabled.

ACTION: No action required.

1207 **MESSAGE: APPC PERFORMANCE TRACING ALREADY DISABLED. (APPCWARN 1207)**

CAUSE: An APPCCONTROL PERFORMANCEOFF command was issued when APPC subsystem performance tracing was already disabled.

ACTION: No action required.

Return Codes

This section lists the intrinsic return codes that may be returned to a TP by control operator intrinsics. The return codes are returned in the *ReturnCode* parameter of the intrinsics.

0 **MESSAGE: Successful completion**

CAUSE: The intrinsic completed successfully, and there are no error or warning messages.

ACTION: No action necessary.

-1001 **MESSAGE: Missing required parameter**

CAUSE: A required parameter was omitted from the intrinsic call.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1002 **MESSAGE: Parameter error**

CAUSE: Incorrect command or parameter specified.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1003 **MESSAGE: Parameter string length too long**

CAUSE: Parameter string exceeded specified length.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1004 **MESSAGE: Too many parameters specified**

CAUSE: Too many parameters were specified in a control operator intrinsic call

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1005 **MESSAGE: Syntax error**

CAUSE: A syntax error was encountered in a control operator intrinsic call.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1006 **MESSAGE: Redundant parameter**

CAUSE: A parameter was used more than once in a control operator intrinsic call.

ACTION: Change the intrinsic call, leaving out the duplicated parameter.

-1007 **MESSAGE: Missing parameter after equal sign**

CAUSE: A parameter was specified with no value assigned to it, or there was a syntax error in the intrinsic call.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1008 **MESSAGE: Missing or invalid delimiter**

CAUSE: Incorrect syntax for intrinsic call.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1009 **MESSAGE: Parameter out of bounds**

CAUSE: A parameter value outside the allowable range was specified.

ACTION: See Chapter 3 , “Control Operator Intrinsics,” of this manual for the syntax of control operator intrinsics. Check the syntax of the intrinsic call in your TP.

-1010 **MESSAGE: User missing NM capability**

CAUSE: A user without NM capability called a control operator intrinsic.

ACTION: Control operator intrinsics require NM (Node Manager) capability.

-1011 **MESSAGE: Invalid value for Erropt parameter**

CAUSE: The *Erropt* parameter of the APPCStart intrinsic is not a valid value.

ACTION: Check the APPCStart intrinsic call and make sure the *Erropt* parameter is either 1 or 2. See Chapter 3 , “Control Operator Intrinsics.”

-1012 **MESSAGE: Invalid value for TraceOn parameter**

CAUSE: The *TraceOn* parameter of the APPCStart intrinsic is not a valid value.

ACTION: Check the APPCStart intrinsic call and make sure the *TraceOn* parameter is either 0 or 1. See Chapter 3 , “Control Operator Intrinsics.”

-1013 **MESSAGE: Invalid value for TFileSize parameter**

CAUSE: The *TFileSize* APPCStart intrinsic is not a valid value.

ACTION: Check the APPCStart intrinsic call and make sure the *TFileSize* parameter is from 0 through 32767. See Chapter 3 , "Control Operator Intrinsics."

-1014 **MESSAGE: Invalid value for PerformanceOn parameter**

CAUSE: The parameter of the APPCStart intrinsic is not a valid value.

ACTION: Check the APPCStart intrinsic call and make sure the *PerformanceOn* parameter is either 0 or 1. See Chapter 3 , "Control Operator Intrinsics."

-1015 **MESSAGE: Invalid SessionTypeName parm value**

CAUSE: Session type name specified in the *SessionType* parameter of the APPCSessions intrinsic is not configured or is not recognized.

ACTION: Determine which session types are configured by issuing the APPCCONTROL STATUS command. If necessary, shut down the APPC subsystem to update the configuration file.

-1016 **MESSAGE: Invalid SessionLimit parm value**

CAUSE: Session limit specified in the *NewLimit* parameter of the APPCSessions intrinsic is greater than the number currently allowed.

ACTION: You could receive this message because you tried to exceed the maximum number of sessions that the APPC subsystem can support, or because you exceeded the maximum number of sessions configured for this session type. Check the number of active sessions and the configured session limit for the session type by issuing the APPCCONTROL STATUS command.

-1017 **MESSAGE: Invalid StopType parm value**

CAUSE: The *StopType* parameter of the APPCStop intrinsic is not a valid value.

ACTION: Check the APPCStop intrinsic call and make sure the *StopType* parameter is 1, 2, or 3. See Chapter 3 , "Control Operator Intrinsics."

-1030 **MESSAGE: APPC module versions mismatch**

CAUSE: Version mismatch detected between the modules.

ACTION: Type APPCCONTROL VERSION to verify the mismatch. Call your HP representative.

-1031 **MESSAGE: Missing required module**

CAUSE: A required file has not been installed on the subsystem.

ACTION: Issue the APPCCONTROL VERSION command to determine which software modules are installed on the system, and save the output to a file. Call your HP representative.

-1032 **MESSAGE: Invalid module ID in version check**

CAUSE: The APPC subsystem detected NM return code 105 during version checking.

ACTION: Issue the APPCCONTROL VERSION command and save the output to a file. Contact your HP representative.

-1034 **MESSAGE: Invalid APPC configuration**

CAUSE: The APPC subsystem detected an error trying to validate the configuration file during subsystem startup.

ACTION: Check the configuration of the APPC subsystem. If the problem persists, contact your HP representative.

-1035 **MESSAGE: Unable to open NMCONFIG file**

CAUSE: The APPC subsystem is unable to open the NMCONFIG.PUB.SYS file.

ACTION: Restart the APPC subsystem. If the same message is returned, call your HP representative.

-1036 **MESSAGE: Unable to close NMCONFIG file**

CAUSE: The APPC subsystem is unable to close the NMCONFIG.PUB.SYS file.

ACTION: Restart the APPC subsystem. If the same message is returned, call your HP representative.

-1037 **MESSAGE: Configuration error**

CAUSE: The APPC subsystem detected an error return code from the NMCONF call of Node Management Services.

ACTION: Check the configuration of the APPC subsystem.

-1038 **MESSAGE: Unable to open CATAPPC.APPC.SYS**

CAUSE: The APPC subsystem is unable to open the catalog file CATAPPC.APPC.SYS.

ACTION: Type LISTF CATAPPC.APPC.SYS to see if the file exists. Call your HP representative.

-1039 **MESSAGE: Unable to close CATAPPC.APPC.SYS**

CAUSE: The APPC subsystem is unable to close the catalog file CATAPPC.APPC.SYS.

ACTION: Type LISTF CATAPPC.APPC.SYS to see if the file exists. Call your HP representative.

-1040 **MESSAGE: Unable to open NM log file**

CAUSE: The APPC subsystem is unable to open the Node Management Log File.

ACTION: Check the logging class configuration in NMMGR.

-1041 **MESSAGE: Unable to close NM log file**

CAUSE: The APPC subsystem is unable to close the Node Management Log File.

ACTION: Check the logging class configuration in NMMGR.

-1042 **MESSAGE: Unable to open internal trace file**

CAUSE: The APPC subsystem is unable to create the internal trace file.

ACTION: Restart the APPC subsystem. If the problem still exists, call your HP representative.

-1043 **MESSAGE: Unable to close internal trace file**

CAUSE: The APPC subsystem is unable to close the internal trace file.

ACTION: Restart the APPC subsystem. If the problem still exists, call your HP representative.

-1046 **MESSAGE: Unable to create Resource Manager at subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1047 **MESSAGE: Unable to create control operator TP at subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1048 **MESSAGE: Unable to create CNOS TP at subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1049 **MESSAGE: Resource Manager detected error during subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1050 **MESSAGE: COPR TP detected error during subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1051 **MESSAGE: CNOS TP detected error during subsystem start-up**

CAUSE: Internal error in APPC subsystem.

ACTION: Contact your HP representative.

-1063 **MESSAGE: Resource Manager error — CHANGE_SESSIONS_FAILED**

CAUSE: The APPC subsystem detected an internal error.

ACTION: Call your HP representative.

-1064 **MESSAGE: Resource Manager error — subsystem shutdown failed**

CAUSE: Internal error in APPC subsystem.

ACTION: Call your HP representative.

-1065 **MESSAGE: Control operator internal error**

CAUSE: Internal error in APPC subsystem.

ACTION: No action required.

-1066 **MESSAGE: APPC object access error**

CAUSE: The Control Operator TP was unable to access the APPC KSO.

ACTION: Call your HP representative.

-1072 **MESSAGE: APPC subsystem not active**

CAUSE: The intrinsic called cannot be executed because the APPC subsystem was not started.

ACTION: Start the APPC subsystem with the APPCCONTROL START command before running your program, or code your program to call the APPCStart intrinsic before calling any other APPC intrinsics.

-1073 **MESSAGE: APPC subsystem pending active**

CAUSE: The intrinsic called cannot be executed because the APPC subsystem startup is in progress.

ACTION: Check the status of the APPC subsystem and restart your program when the subsystem is active. If problems still exist, call your HP representative.

-1074 **MESSAGE: APPC subsystem already active**

CAUSE: The APPCStart intrinsic was called after the APPC subsystem had already been activated.

ACTION: No action necessary.

-1075 **MESSAGE: APPC subsystem shutting down**

CAUSE: A control operator intrinsic was called after initiation of subsystem shutdown.

ACTION: Check the status of the subsystem and wait until the subsystem has completed shutdown. Then, restart your program.

-1201 **MESSAGE: Session limit exceeded**

CAUSE: The session limit for the APPC subsystem was reached before the end of the configuration file. Remaining session types are initialized in the APPC object with a session limit of 0.

ACTION: No action is necessary if you don't use the session types that are not initialized. Use the APPCCONTROL SESSIONS command or the APPCSessions intrinsic to reapportion sessions among the session types. If necessary, shut down the APPC subsystem and update the configuration file.

-1202 **MESSAGE: CATREAD failed**

CAUSE: A non-zero result code was returned from a call to CATREAD.

ACTION: Restart the APPC subsystem. If the problem still exists, contact your HP representative.

Configuration Messages

This section lists the messages that NMMGR may generate when validating the APPC portion of HP-IBM configuration.

NOTE During HP-IBM validation, NMMGR attempts to validate all SNA subsystems. Whenever it finds no configuration data for a particular subsystem, it generates a warning message. Disregard any warning messages about subsystems you have not installed on your system.

In the following messages, an exclamation point (!) indicates a value that the system will provide when the message is generated.

1 **MESSAGE: ---> VALIDATION OF APPC SUBSYSTEM STARTED <---**

CAUSE: NMMGR has started to validate the APPC subsystem portion of the configuration file.

ACTION: No action required.

2 **MESSAGE: ---> VALIDATION OF APPC SUBSYSTEM FINISHED <---**

CAUSE: NMMGR has finished validating the APPC subsystem portion of the configuration file.

ACTION: No action required.

3 **MESSAGE: INTERNAL ERROR. Get data failed with result code !. (APPCVALERR 3)**

CAUSE: An internal error in NMMGR has occurred.

ACTION: Notify your HP Representative

4 **MESSAGE: INTERNAL ERROR. Get path failed with result code !. (APPCVALERR 4)**

CAUSE: An internal error in NMMGR has occurred.

ACTION: Notify your HP Representative.

5 **MESSAGE: Configuration file corrupt. Missing APPC Path. (APPCVALERR 5)**

CAUSE: The configuration file you are validating does not contain APPC path information and cannot be used.

ACTION: Notify your HP Representative.

6 **MESSAGE: Configuration file corrupt. Missing APPC Version. (APPCVALERR 6)**

CAUSE: The configuration file you are validating does not contain APPC version information and cannot be used.

ACTION: Notify your HP Representative.

7 **MESSAGE: No configuration data exists for the APPC subsystem. (APPCVALWARN 7)**

CAUSE: You have attempted to validate the APPC subsystem portion of the configuration file, but no data has been configured for the APPC subsystem.

ACTION: Configure the APPC subsystem before validating the configuration file.

8 **MESSAGE: Path : APPC.NETID**
Required NETID is not configured. (APPCVALERR 8)

CAUSE: You have attempted to validate the APPC subsystem configuration without having entered a Local Network ID in the APPC: Network ID Data screen.

ACTION: Configure your Local Network ID in the APPC: Network ID Data screen.

9 **MESSAGE: Path : APPC.MODE.!**
! is an invalid value for Maximum Send RU Size. (APPCVALERR 9)

CAUSE: You have entered a `Maximum Send RU Size` that is outside the allowable range.

ACTION: Configure an RU size from 256 through 2048.

10 **MESSAGE: Path : APPC.MODE.!**
! is an invalid value for Maximum Receive RU Size. (APPCVALERR 10)

CAUSE: You have entered a `Maximum Receive RU Size` that is outside the allowable range.

ACTION: Configure an RU size from 256 through 2048.

11 **MESSAGE: Path : APPC.MODE**
“!” is a reserved name and can not be reconfigured. (APPCVALERR 11)

CAUSE: You have attempted to configure a mode name that is internally defined by the APPC subsystem and cannot be reconfigured.

ACTION: Rename the mode type.

12 **MESSAGE: Path : APPC.MODE.!**
No data is configured for this Mode Type. (APPCVALERR 12)

CAUSE: A mode was configured in the APPC: Select Mode Type screen, but no data was configured for it in the APPC: Mode Type Data screen.

ACTION: Configure the data for the specified mode type in the APPC: Mode Type Data screen.

13 **MESSAGE: Path : APPC.SESSION**
At least one Session Type must be configured. (APPCVALERR 13)

CAUSE: You have attempted to validate the APPC subsystem configuration file without configuring any session types.

ACTION: Configure at least one session type before validating the configuration file.

14 **MESSAGE: Path : APPC.SESSION.!**
No data is configured for this Session Type. (APPCVALERR 14)

CAUSE: A session type was configured in the APPC: Select Session Type screen, but no data was configured for it in the APPC: Independent or Dependent LU Session Type Data screen.

ACTION: Configure the data for the specified session type.

15 **MESSAGE: Paths : APPC.SESSION.!, APPC.SESSION.!**
Local LU Name "!" is not unique. It can not be configured for both SNA Node "!" and SNA Node "!". (APPCVALERR 15)

CAUSE: The same local LU is configured on two different SNA Nodes (PUs).

ACTION: Correct the SNA Node configuration so that each local LU is configured on exactly one SNA Node.

16 **MESSAGE: Paths : APPC.SESSION.!, APPC.SESSION.!**
These Independent Session Types have the same triple (Local LU, Mode, Remote LU). (APPCVALERR 16)

CAUSE: Duplicate independent LU session types have been configured.

ACTION: Delete one of the duplicate session types, or modify one to use a different Local LU, Mode Type, or Remote LU.

17 **MESSAGE: Paths : APPC.SESSION.!, APPC.SESSION.!**
A single Local LU — Remote LU pair can not support both single and parallel sessions. (APPCVALERR 17)

CAUSE: Two independent LU session types have been configured using the same Local LU and Remote LU; one uses a single session and one uses parallel sessions.

ACTION: Configure a different Local LU for one of the session types, or modify one session type so that both use single sessions or both use parallel sessions.

18 **MESSAGE: Path : APPC.SESSION.!**
Mode Name "!" is not configured under APPC.MODE. (APPCVALERR 18)

CAUSE: You have configured a session type that uses an unconfigured Mode Name.

ACTION: Configure the Mode Name in the APPC: Select Mode Type screen and the APPC: Mode Type Data screen.

19 **MESSAGE: Path : APPC.SESSION.!**
SNA Node "!" does not support dependent LUs. (APPCVALERR 19)

CAUSE: You have configured a dependent LU session type to use a Type 2.1 SNA node without dependent LU support.

ACTION: Specify `Node 2.1 Dependent LU Support = Y` in the "SNA Node Configuration: PU Data" screen for that node, or specify a different SNA node for the session type.

20 **MESSAGE: Path : APPC.SESSION.!**
SNA Node "!" is not a PU 2.1 type node. (APPCVALERR 19)

CAUSE: You have configured an independent LU session type to use a Type 2.0 SNA node. Independent LU session types must use Type 2.1 SNA nodes.

ACTION: Change the node type of the SNA node to 2.1, or configure a different SNA node for the session type.

21 **MESSAGE: Path : APPC.SESSION.!.LULIST**
There are no LU names configured for this path. (APPCVALERR 21)

CAUSE: You have configured a dependent LU session type with no local dependent LUs.

ACTION: Configure as many dependent LUs as you will have sessions of the session type configured. Dependent LUs are configured in the APPC: Dependent LU List Data screen.

22 **MESSAGE: Path : APPC.SESSION.!**
Node "!" is not configured under the SNANODE screen. (APPCVALERR 22)

CAUSE: You have configured a session type to use an unconfigured SNA node.

ACTION: Configure the SNA node in the SNA Node Configuration screen.

23 **MESSAGE: Path : !**
LU "!" is not configured under the SNANODE.! screen. (APPCVALERR 23)

CAUSE: You have configured a session type to use a local LU that is not configured on the SNA node you specified for the session type.

ACTION: Configure the local LU in the SNA Node Configuration: LU Data screen for the SNA node used by the session type, or change the local LU to one that is already configured.

24 **MESSAGE: Path : APPC.SESSION.!**
LU "!" is not an independent LU under SNANODE.! screen. (APPCVALERR 24)

CAUSE: You have configured an independent LU session type to use a local LU that is not configured as an independent LU on the SNA node you specified for the session type.

ACTION: Specify an LU that is configured as an independent LU on the correct SNA Node. (Independent LUs must have no LU# configured in the "SNA Node Configuration: LU Data" screen.)

25 **MESSAGE: Path : APPC.SESSION.!**
Number of LUs in LULIST (!) does not equal the Maximum Number of Sessions (!). (APPCVALERR 25)

CAUSE: You have configured more or fewer dependent LUs than you can have sessions of the specified session type.

ACTION: Delete some dependent LUs from the LU list, or add some dependent LUs to the LU list, until the number of LUs equals the `Maximum Number of Sessions` configured for that session type. Or change the `Maximum Number of Sessions` value to match the number of LUs configured.

26 **MESSAGE: Path : APPC.SESSION.!**
"!" is not a correctly formatted Remote LU Name. (APPCVALERR 26)

CAUSE: You have specified a `Remote LU Name` that does not have the correct format.

ACTION: Specify the `Remote LU Name` as *`netid.luname`*, where *`netid`* is the network identifier for the remote system, and *`luname`* is the name of the remote LU.

27 **MESSAGE: Path : APPC.TP.!**
There is no Transaction Program Data configured. (APPCVALERR 27)

CAUSE: A transaction program name was configured in the APPC: Select Transaction Program screen, but no data was configured for it in the APPC: Transaction Program Data screen.

ACTION: Configure the data for the specified transaction program.

28 **MESSAGE: Path : APPC.TP.!**
"!" is not a correctly formatted job name. (APPCVALERR 28)

CAUSE: You have specified a `Job Name` that does not have the correct format.

ACTION: Specify the `Job Name` as *`file.group.account`*. The file must not contain any lockwords.

29 **MESSAGE: Path : APPC.TP.!**
TP Job Name "!" does not exist on the system. (APPCVALWARN 29)

CAUSE: You have specified a Job Name that cannot be found.

ACTION: Create a job to stream the configured local TP. Make sure it has the same file name and is located in the same group and account you specified in the Job Name field of the APPC: Transaction Program Data screen.

30 **MESSAGE: Path : APPC.TP**
"!" is a reserved name and can not be reconfigured. (APPCVALERR 30)

CAUSE: You have specified a TP Name that is already being used internally by the APPC subsystem.

ACTION: Rename the configured TP.

31 **MESSAGE: Path : LOGGING**
Logging subsystem "SUB0016" is not configured. (APPCVALERR 31)

CAUSE: No logging has been configured for subsystem SUB0016, the logging subsystem for the APPC subsystem.

ACTION: Configure the logging options for that subsystem, and make sure the data in the logging configuration screen has been saved. See the *SNA Link/XL Node Manager's Guide* for information about logging configuration.

32 **MESSAGE: Path : LOGGING.SUB0016**
No classes are configured for SUB0016 logging. (APPCVALERR 32)

CAUSE: SUB0016 has been named, but no logging classes have been configured for it.

ACTION: Go to the Logging Configuration: Logging Classes screen for SUB0016 and configure the logging classes. See the *SNA Link/XL Node Manager's Guide* for information about logging configuration.

33 **MESSAGE: Path : LOGGING.SUB0016**
Logging class "!" is not configured. (APPCVALERR 33)

CAUSE: The specified logging classes has not been configured for subsystem SUB0016.

ACTION: Go to the Logging Configuration: Logging Classes screen for SUB0016 and configure the indicated logging class. See the *SNA Link/XL Node Manager's Guide* for information about logging configuration.

34 **MESSAGE: Path : LOGGING.SUB0016**
"!" is not a valid class for SUB0016 logging. (APPCVALERR 34)

CAUSE: The specified logging class cannot be configured for subsystem SUB0016.

ACTION: Go to the Logging Configuration: Logging Classes screen for SUB0016 and correct the logging class specifications. Valid SUB0016 logging class names are CLAS0010, CLAS0011, CLAS0012, CLAS0013, and CLAS0014. See the *SNA Link/XL Node Manager's Guide* for information about logging configuration.

35 **MESSAGE: Path : LOGGING.SUB0016**
"!" is not a correctly formatted User Name. (APPCVALERR 35)

CAUSE: The User Name specified was not in the proper format.

ACTION: See the *SNA Link/XL Node Manager's Guide* for information about logging configuration.

36 **MESSAGE: Path : APPC.SESSION.!.LULIST**
LU "!" is not a dependent LU under SNANODE.! screen. (APPCVALERR 36)

CAUSE: The specified dependent LU, which you configured in the "APPC: Dependent LU List Data" screen, is not configured as a dependent LU on the SNA node you specified in the "APPC: Dependent LU Session Type Data" screen.

ACTION: In the "APPC: Dependent LU List Data" screen, list only the dependent LUs that are configured on the SNA node you will use for the session type.

B Sample Configuration

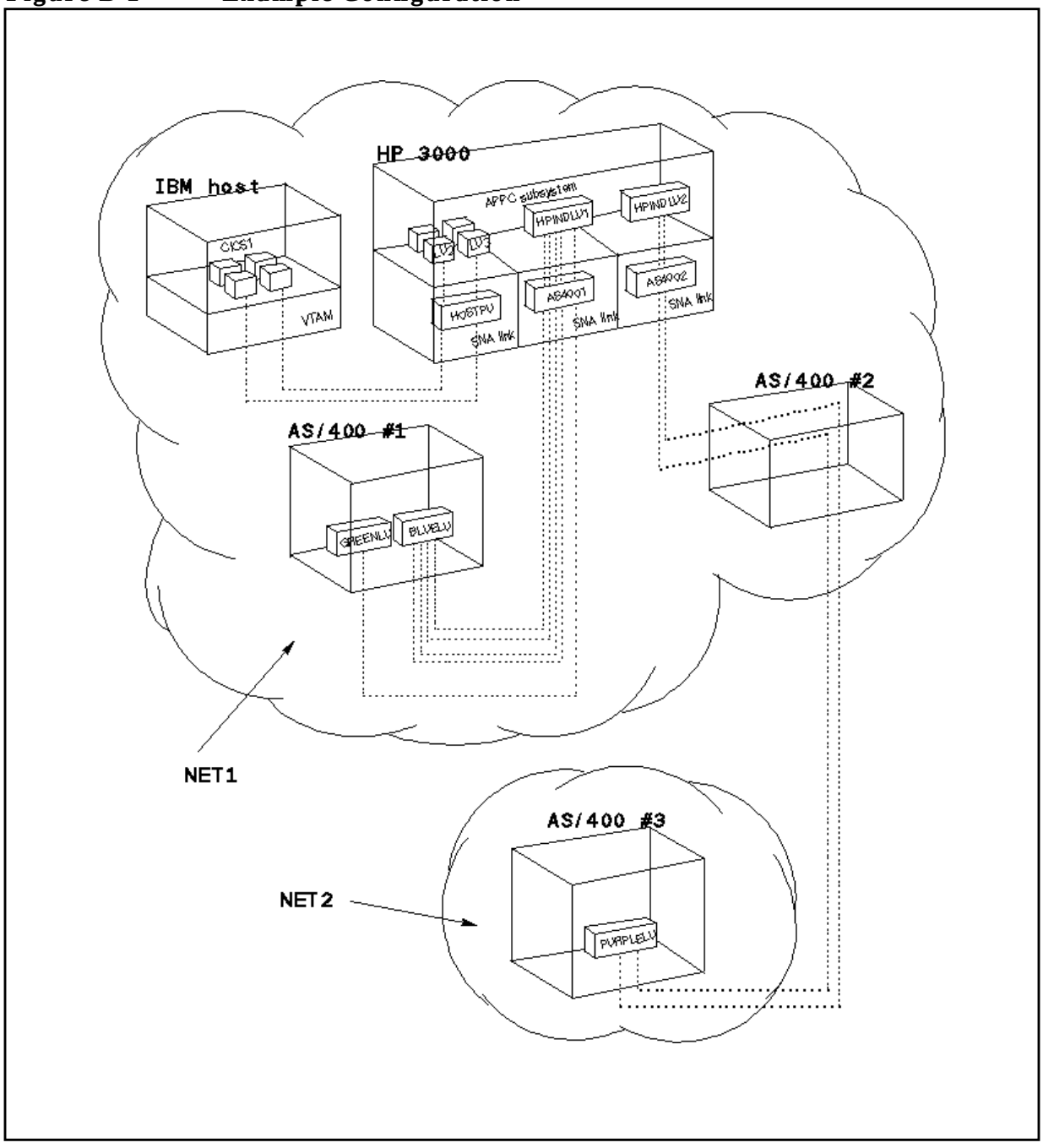
The sample configuration in this appendix is based on Figure 4-19. The figure is reproduced in this section as Figure B-1.

This appendix contains samples of the following:

- SNA node configuration screens based on the LU 6.2 configuration shown in Figure B-1.
- APPC subsystem configuration screens based on the LU 6.2 configuration shown in Figure B-1.
- A sample of the NMMGR critical summary output for the APPC subsystem based on the LU 6.2 configuration in shown in this appendix.
- Selected lines from an NCP gen showing host configuration values that correspond to HP 3000 configuration items.
- Selected lines from an AS/400 gen showing configuration values that correspond to HP 3000 configuration items.

The example shown in Figure B-1 is reproduced from Chapter 4 , “APPC Subsystem Configuration.” Configuration screens, critical summary, and remote system gens in this appendix are based on the configuration in Figure B-1.

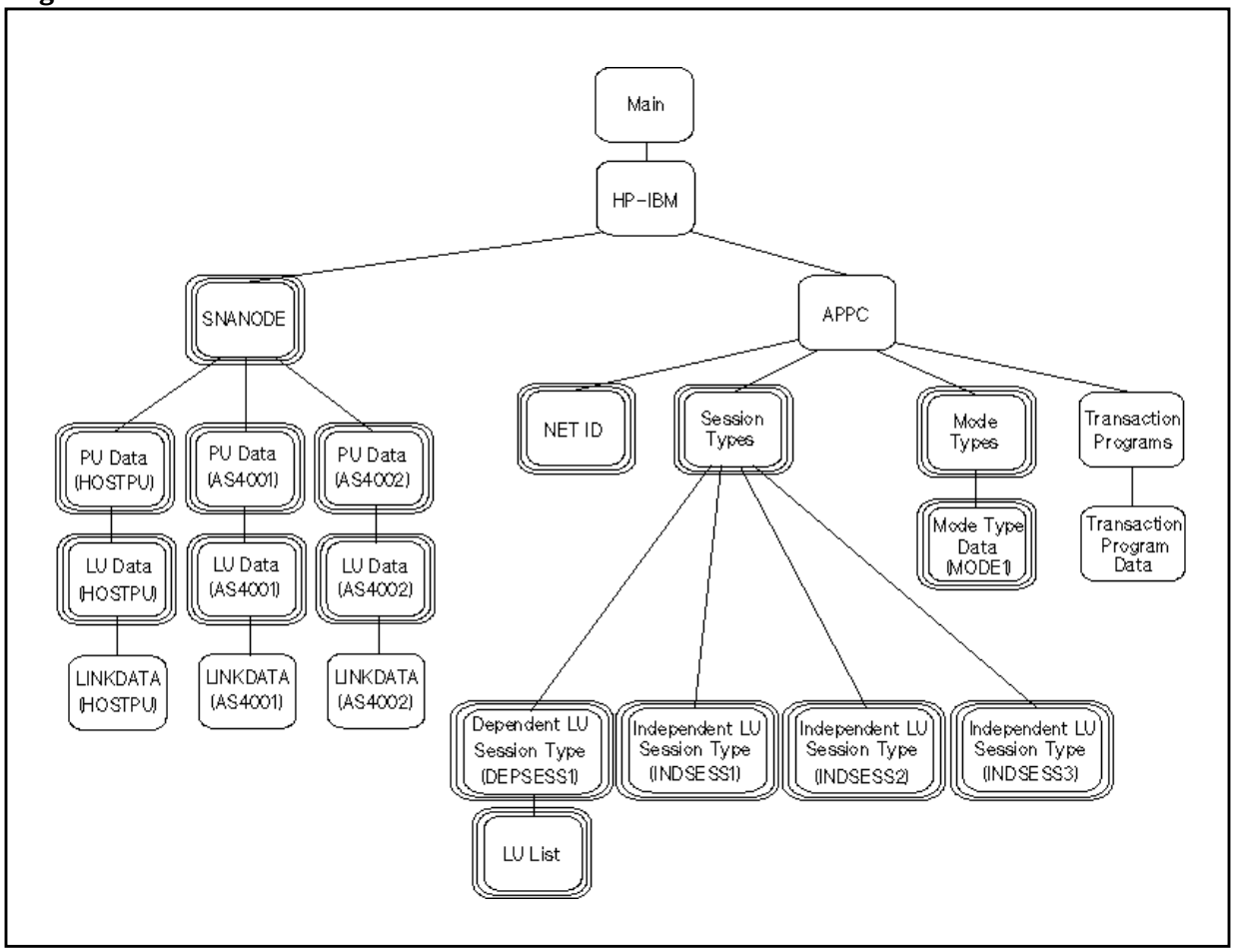
Figure B-1 Example Configuration



HP 3000 Configuration

Figure B-2 shows the NMMGR screen structure for the LU 6.2 configuration shown in Figure B-1. The configuration must be performed across two separate NMMGR subtrees: “SNANODE” and “APPC”. You select the subtree you want to configure from the list on the “HP-IBM Configuration” screen. The screens discussed in this appendix have been highlighted in Figure B-2.

Figure B-2 NMMGR Screen Structure



Configuration Screens

The following NMMGR screens are used to illustrate this configuration example:

- "SNA Node Configuration" screen
- "SNA Node Configuration: PU Data" screens
- "SNA Node Configuration: LU Data" screens
- "APPC: Network ID Data" screen
- "APPC: Select Session Type" screen
- "APPC: Dependent LU Session Type Data" screen
- "APPC: Dependent LU List Data" screen
- "APPC: Independent LU Session Type Data" screens
- "APPC: Select Mode Type" screen
- "APPC: Mode Type Data" screen

The discussion of each screen includes an illustration of the screen, a diagram showing where it fits into the NMMGR screen structure, and a description of the items configured in it.

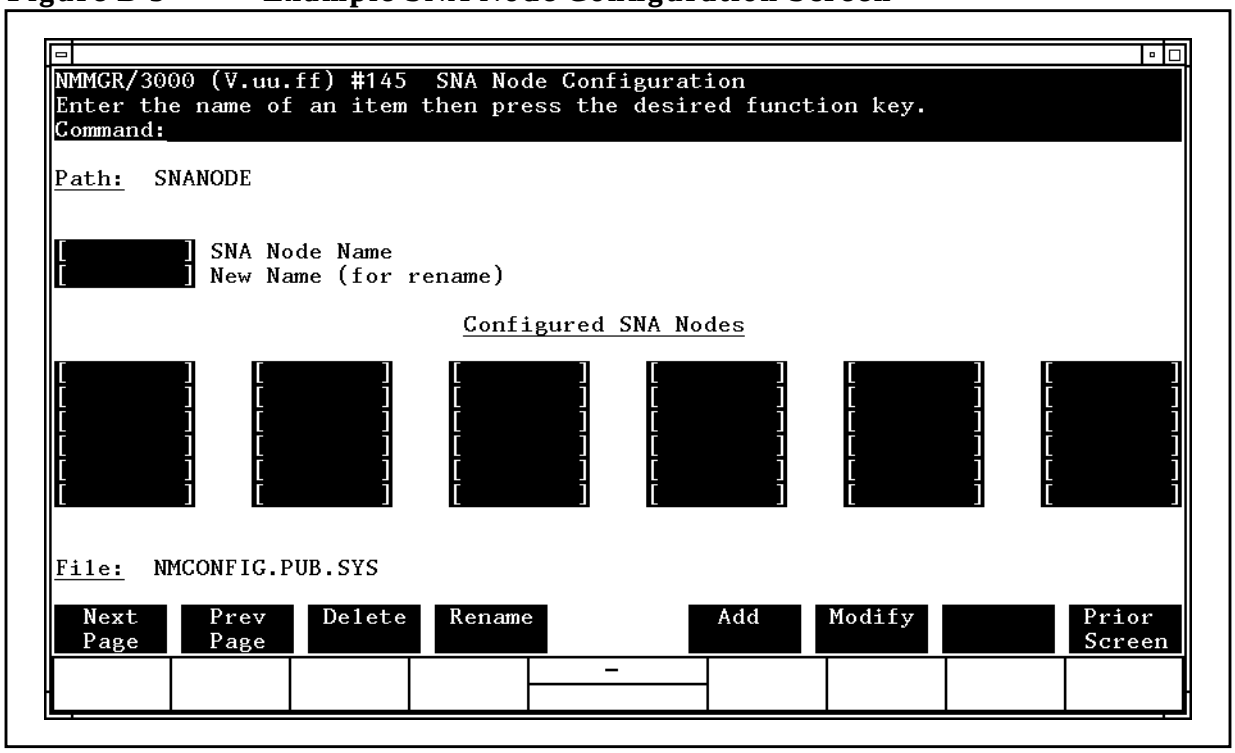
NOTE The APPC subsystem configuration screens are also described in Chapter 4 , "APPC Subsystem Configuration," of this manual. The SNA node configuration screens are described in the *SNA Link/XL Node Manager's Guide*.

SNA Node Configuration Screen

Figure B-3 shows the “SNA Node Configuration” screen. Three nodes are configured in it: HOSTPU, AS4001 and AS4002. Each node requires a separate copy of SNA/SDLC Link/XL, which links the HP 3000 to a remote system.

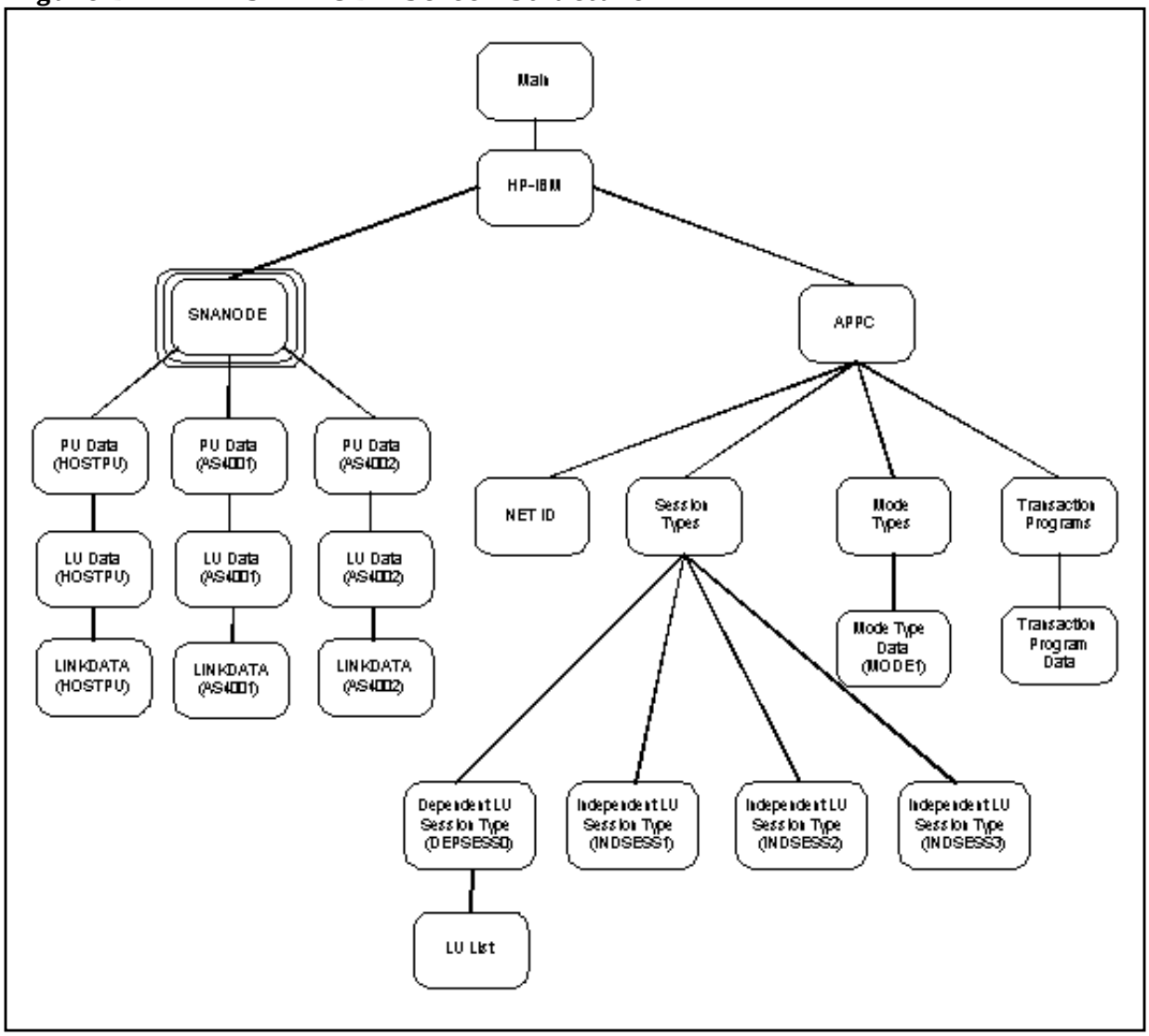
The SNA node name HOSTPU matches the PU name configured on the IBM host. The SNA node names AS4001 and AS4002 match the RMTCPNAME (remote control point name) values configured on the two AS/400s.

Figure B-3 Example SNA Node Configuration Screen



The following diagram as shown in Figure B-4 shows how the “SNA Node Configuration” screen fits into the NMMGR screen structure. The SNA nodes configured here will be entered in the “APPC: Dependent LU Session Type Data” and “APPC: Independent LU Session Type Data” screens for the session types that will use these nodes.

Figure B-4 SNANODE Screen Structure

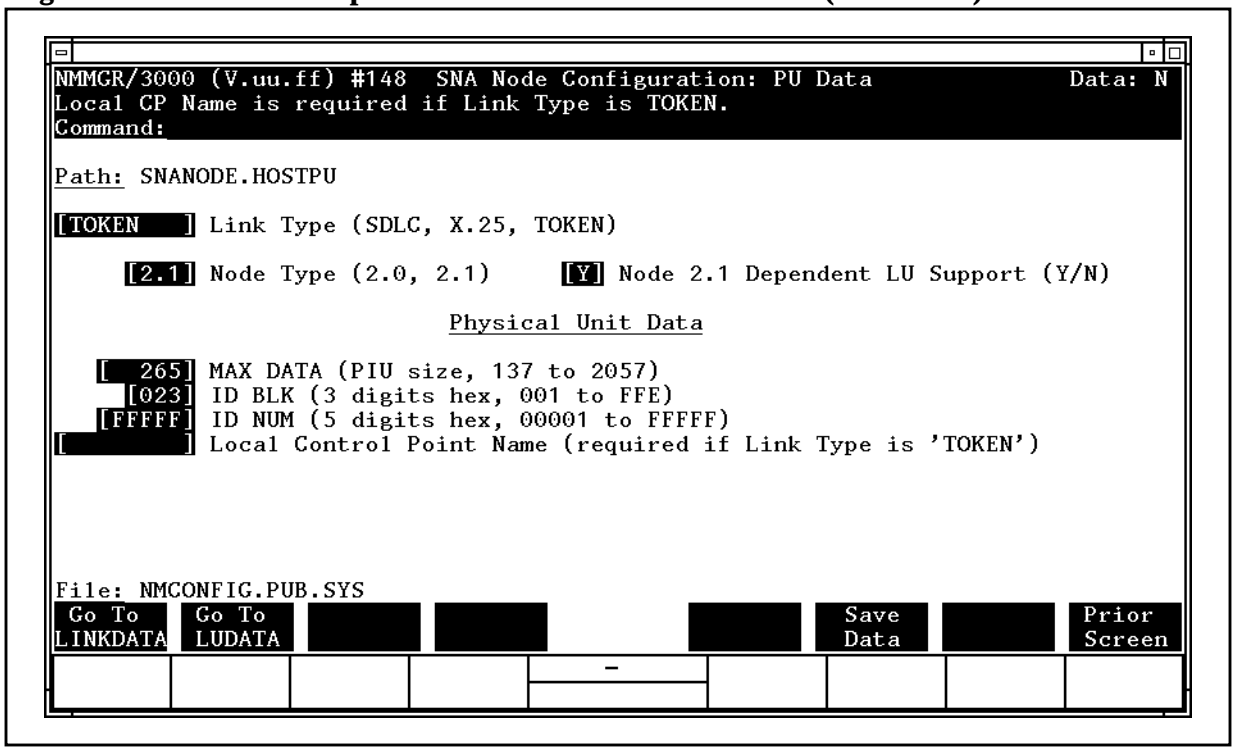


SNA Node Configuration: PU Data and LU Data Screens

Figure B-5 shows the "SNA Node Configuration: PU Data" screen for node HOSTPU. The Node Type for HOSTPU is configured as 2.0, because all the LUs using this node are dependent LUs. Independent LUs must be configured on Type 2.1 nodes.

If you wanted to configure both independent and dependent LUs on the same node, you would configure it as a Node Type 2.1, and you would specify Y in the Node 2.1 Dependent LU Support field.

Figure B-5 Example SNANODE PU Data Screen (HOSTPU)



The following diagram as shown in Figure B-6 shows how the “SNA Node Configuration: PU Data” screen for HOSTPU fits into the NMMGR screen structure.

Figure B-6 PU Data (HOSTPU) Screen Structure

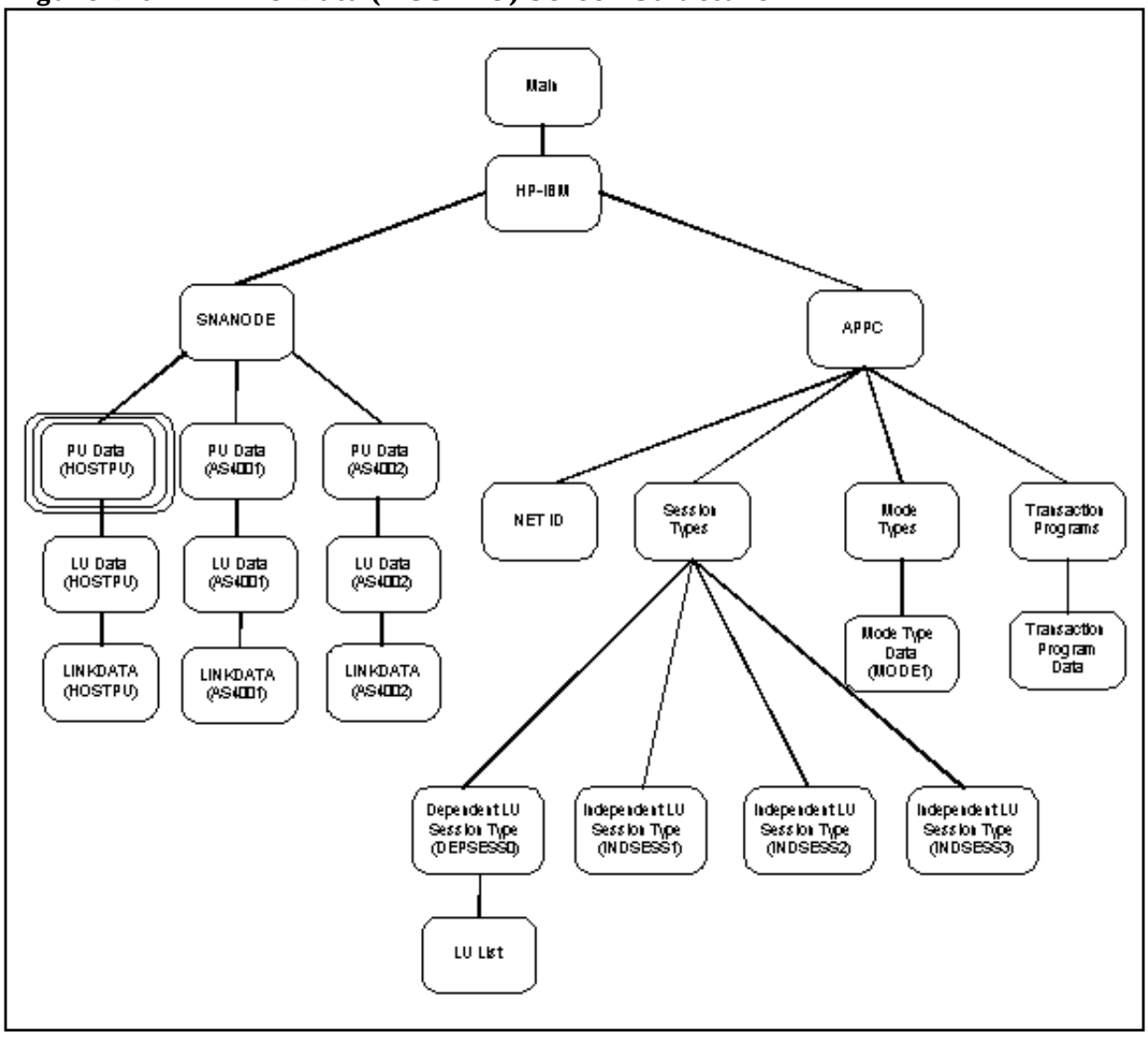
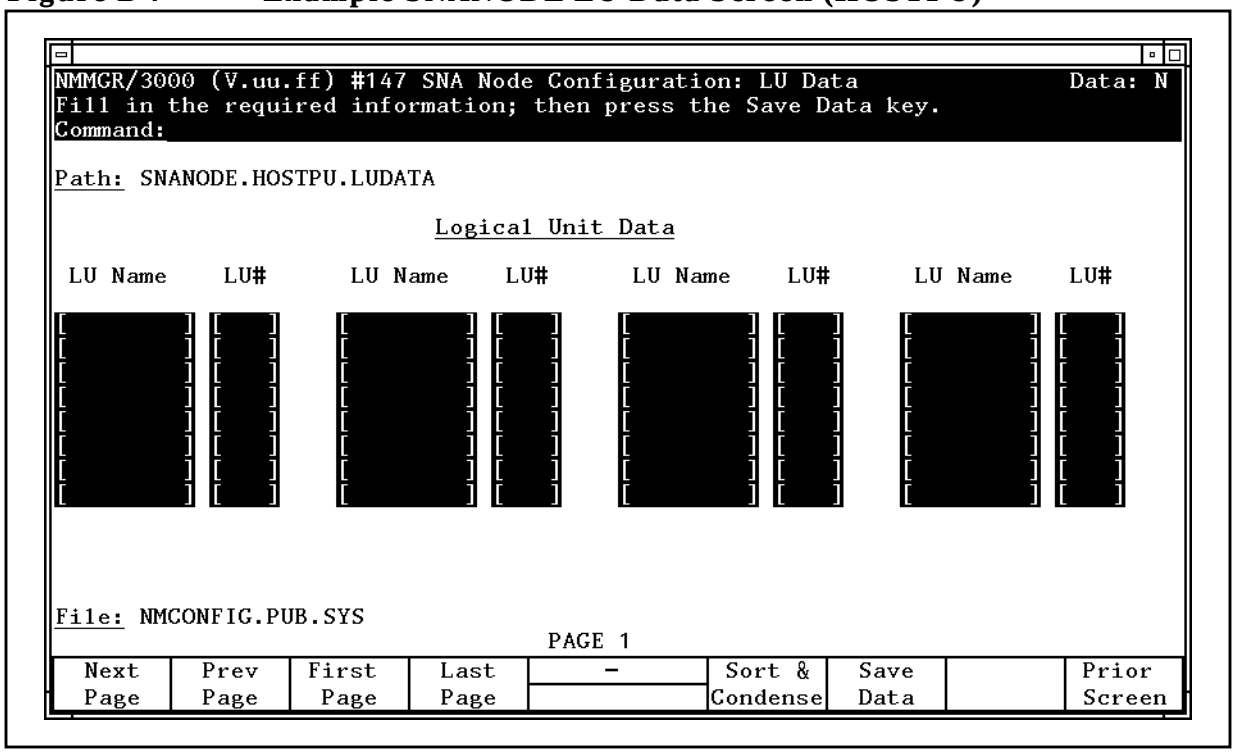


Figure B-7 shows the “SNA Node Configuration: LU Data” screen for node HOSTPU. The four LUs configured here are used by the session type DEPSESS1, the only session type configured to use this node.

The LU names configured here match the LU names configured on the IBM host.

Figure B-7 Example SNANODE LU Data Screen (HOSTPU)



The following diagram shown in Figure B-8 shows how the “SNA Node Configuration: LU Data” screen for HOSTPU fits into the NMMGR screen structure. The LUs configured here will be entered in the “APPC: Dependent LU List Data” screen for session type DEPSESS1.

Figure B-8 LU Data (HOSTPU) Screen Structure

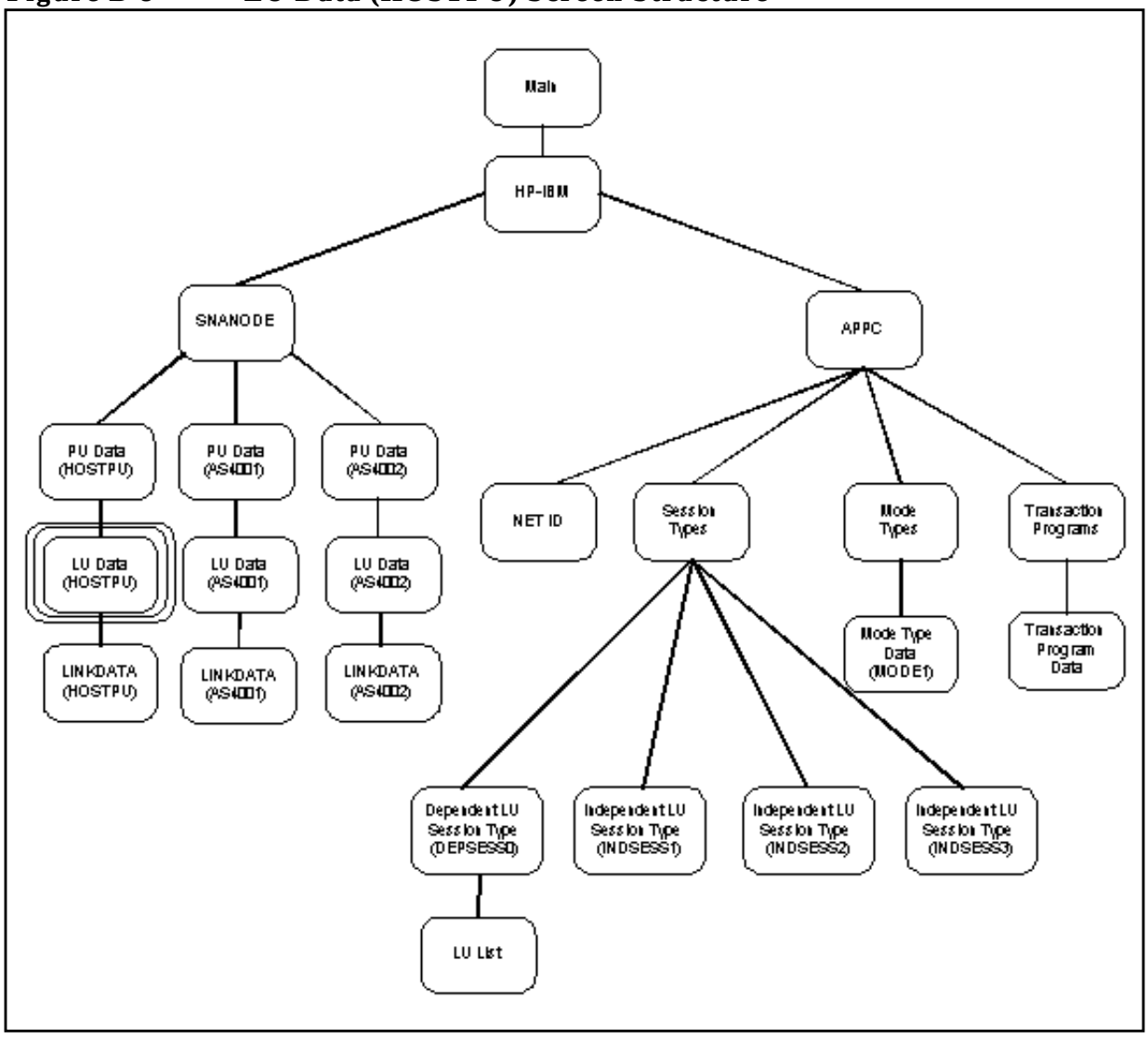
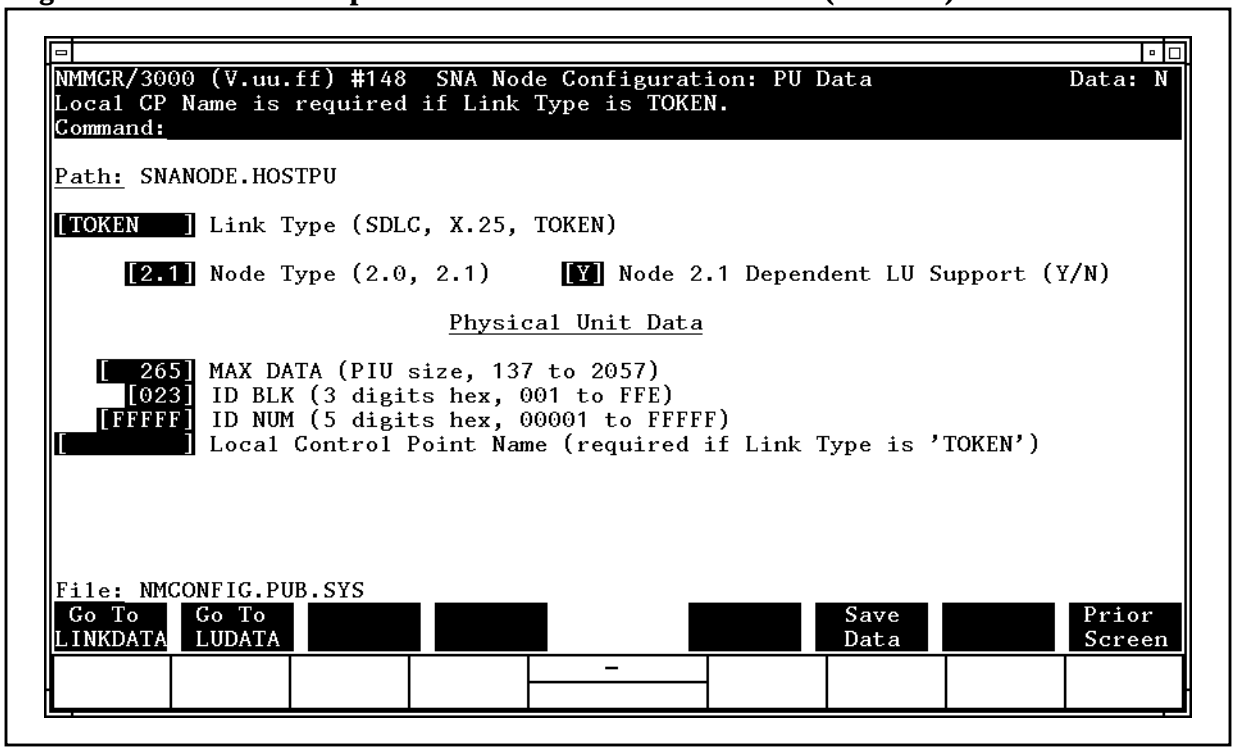


Figure B-9 shows the "SNA Node Configuration: PU Data" screen for node AS4001. The Node Type for AS4001 is configured as 2.1, because the HP 3000 functions as a peer node when it communicates with an AS/400. Since no dependent LUs will use AS4001, the Node 2.1 Dependent LU Support field is set to N.

The ID BLK and ID NUM fields, together, make up the EXCHID configured in the Controller Description on the AS/400. (The EXCHID in the Controller Description, for this example configuration, is 05600001.)

Figure B-9 Example SNANODE PU Data Screen (AS4001)



The following diagram shown in Figure B-10 shows how the “SNA Node Configuration: PU Data” screen for AS4001 fits into the NMMGR screen structure.

Figure B-10 PU Data (AS4001) Screen Structure

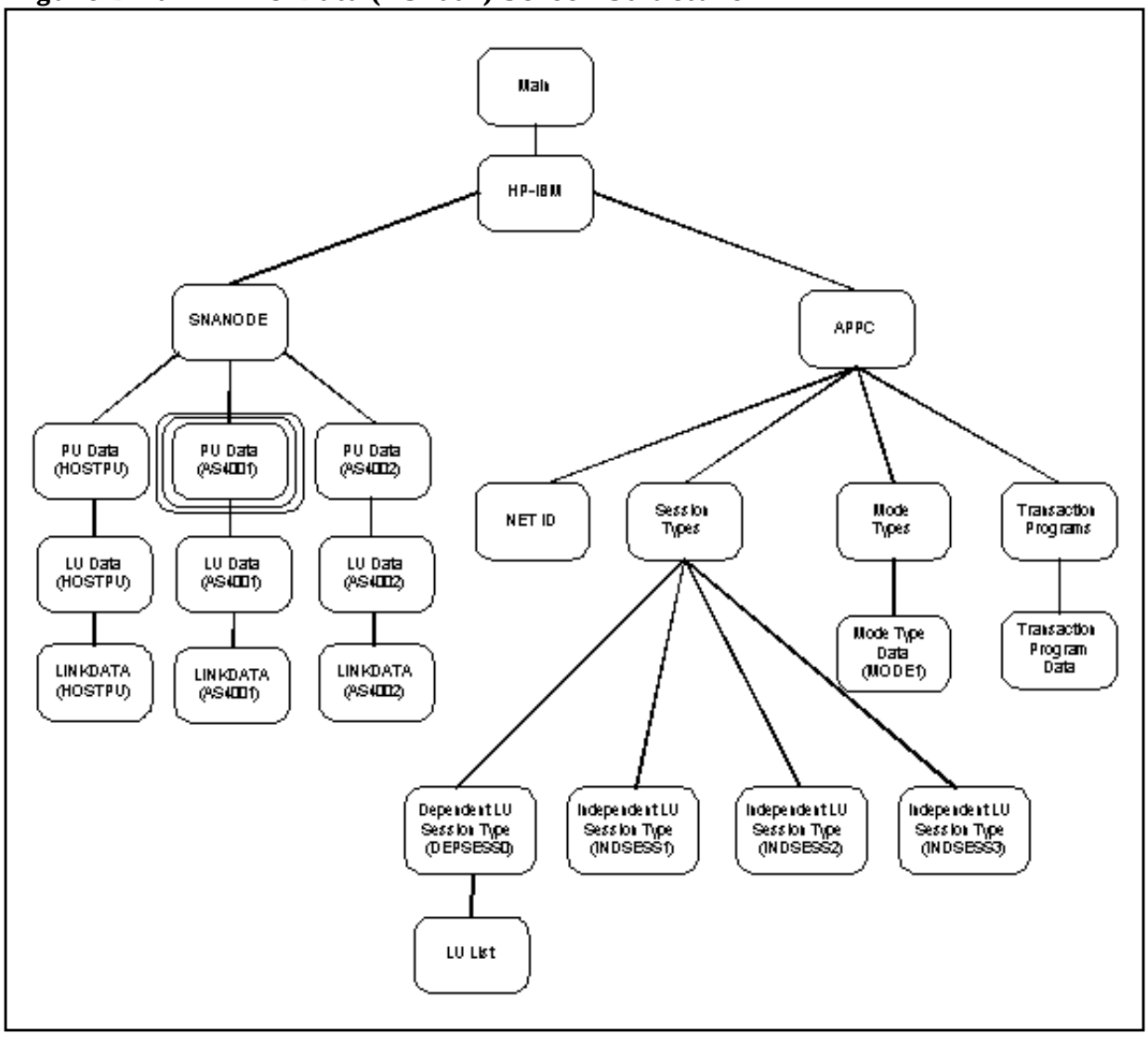
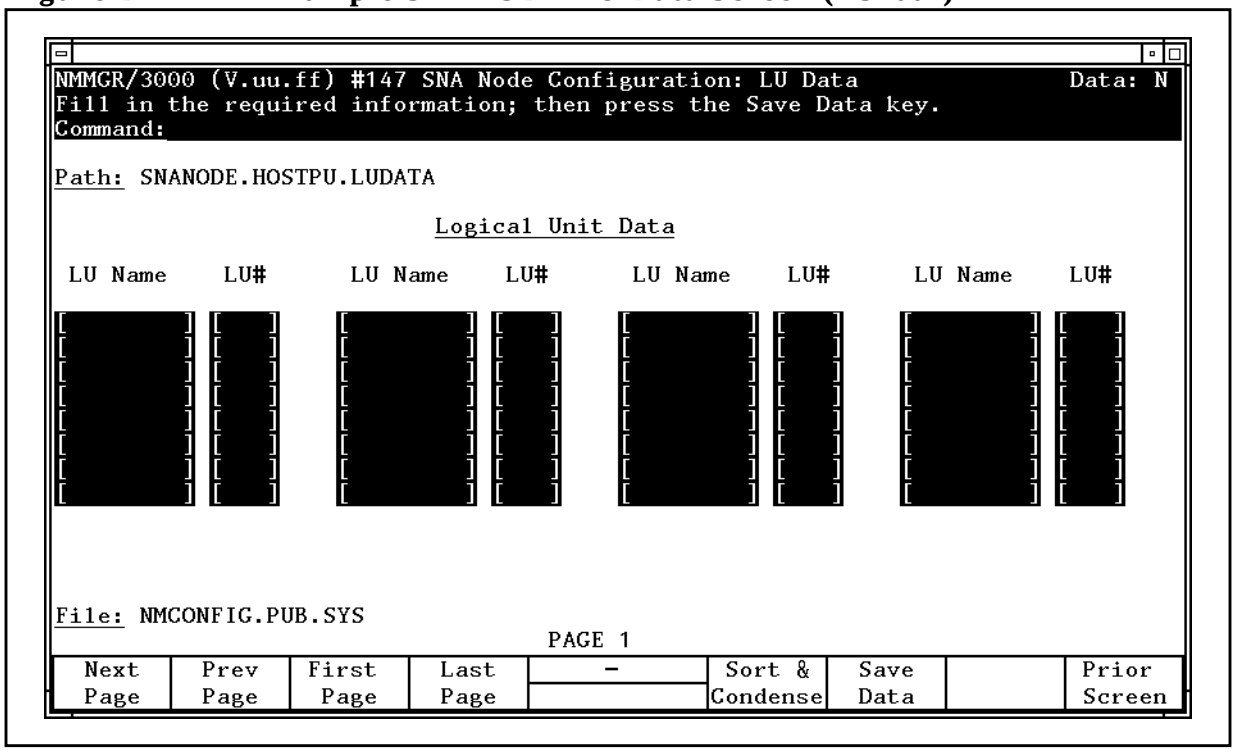


Figure B-11 shows the “SNA Node Configuration: LU Data” screen for node AS4001. The independent LU configured here is used by two session types: INDSESS1 and INDSESS2. The LU# field must be left blank for an independent LU.

The LU name HPINDLU1 must match the RMTLOCNAME (remote location name) configured in the Device Description on the AS/400.

Figure B-11 Example SNANODE LU Data Screen (AS4001)



The following diagram shown in Figure B-12 shows how the “SNA Node Configuration: LU Data” screen for AS4001 fits into the NMMGR screen structure. The LU configured here will be entered in the “APPC: Independent LU Session Type Data” screens for session types INDSESS1 and INDSESS2.

Figure B-12 LU Data (AS4001) Screen Structure

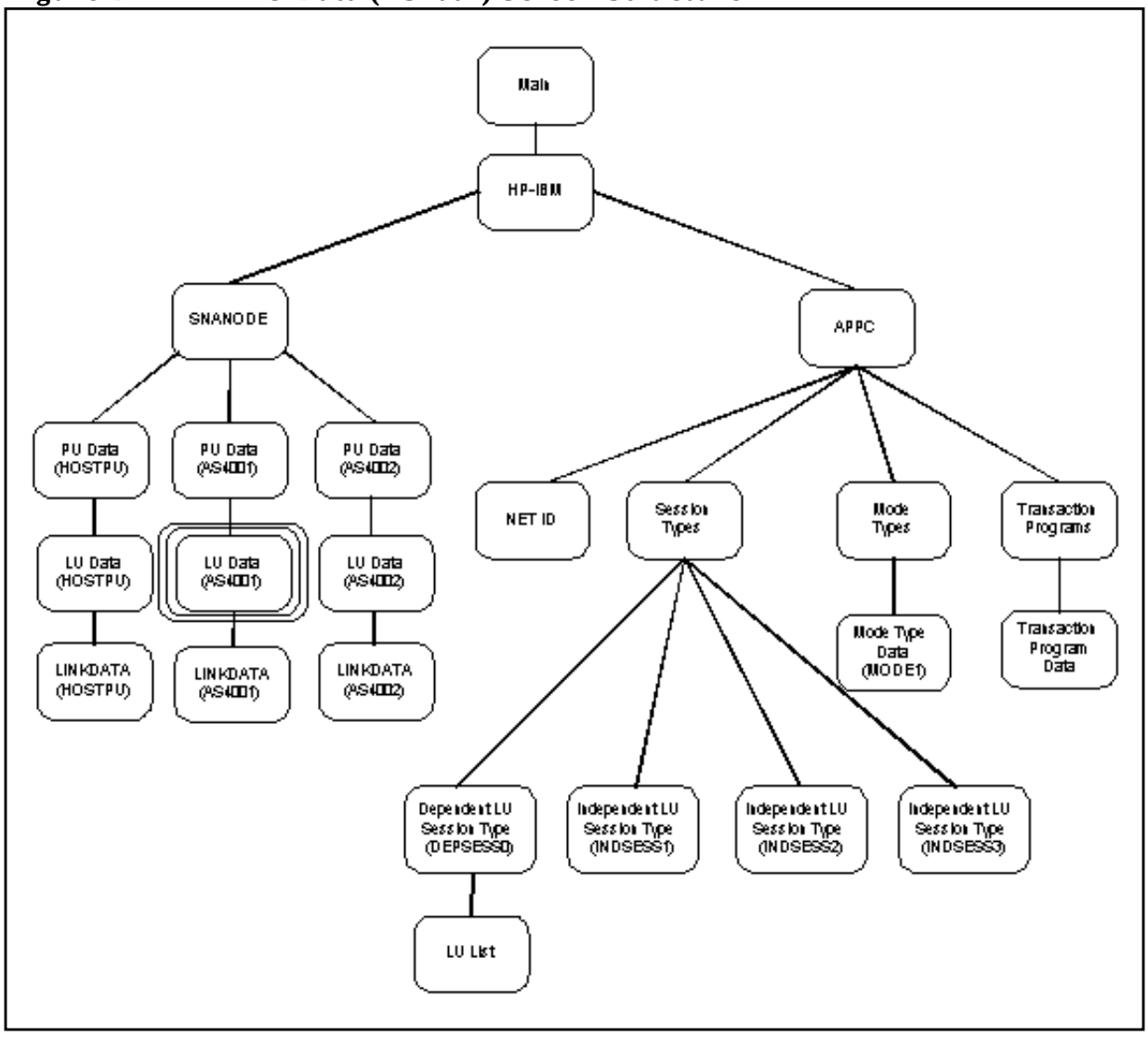
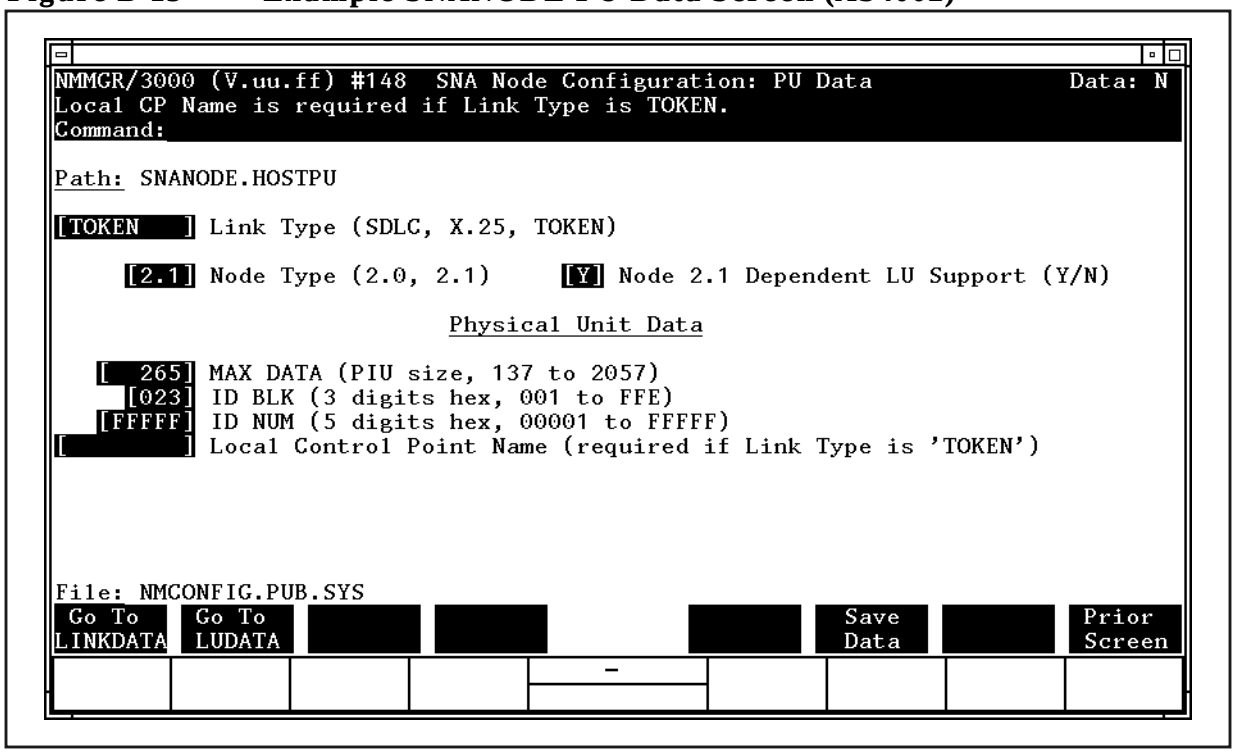


Figure B-13 shows the “SNA Node Configuration: PU Data” screen for node AS4002.

The ID BLK and ID NUM fields, together, make up the EXCHID configured in the Controller Description on the AS/400. (The EXCHID in the Controller Description, for this example configuration, is 05600002.)

Figure B-13 Example SNANODE PU Data Screen (AS4002)



The following diagram shown in Figure B-14 shows how the “SNA Node Configuration: PU Data” screen for AS4002 fits into the NMMGR screen structure.

Figure B-14 PU Data (AS4002) Screen Structure

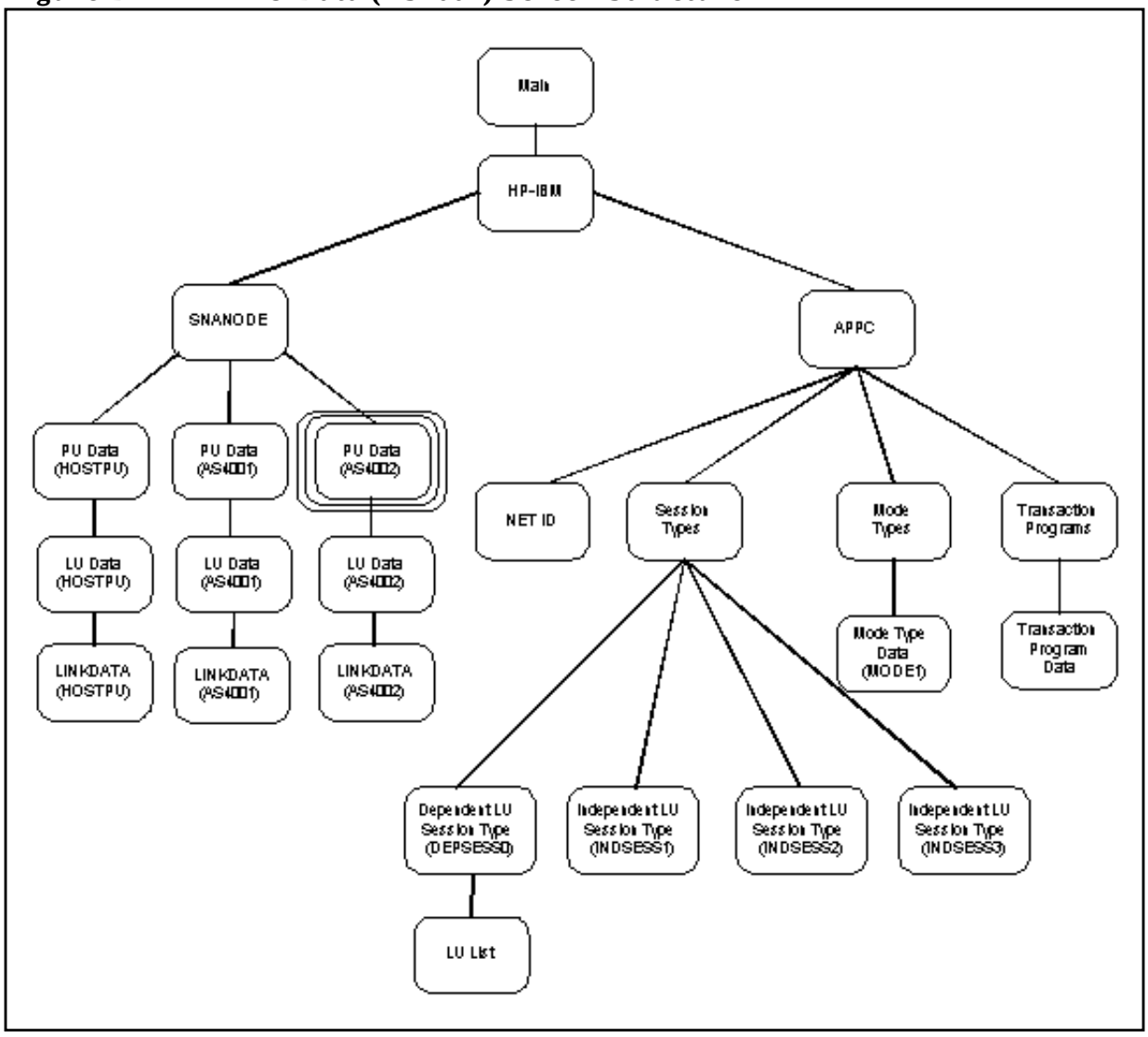
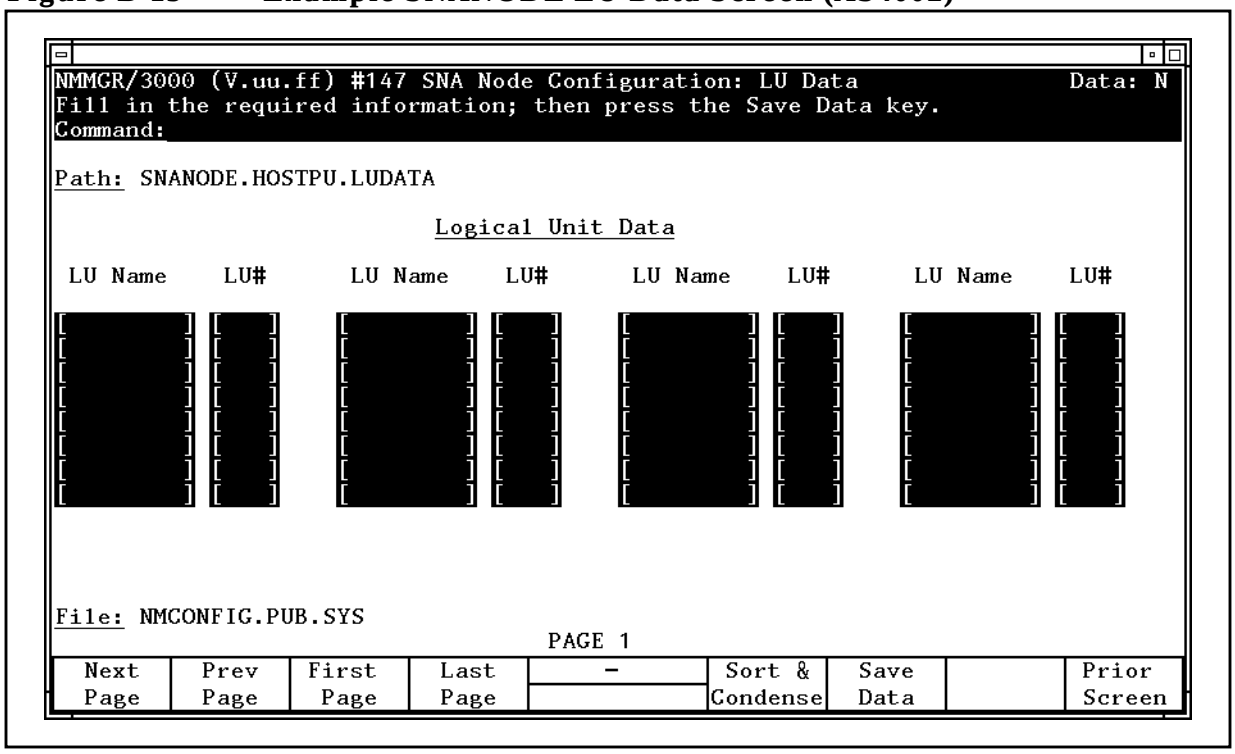


Figure B-15 shows the “SNA Node Configuration: LU Data” screen for node AS4002. The independent LU configured here is used by session type INDSESS3. The LU# field must be left blank for an independent LU.

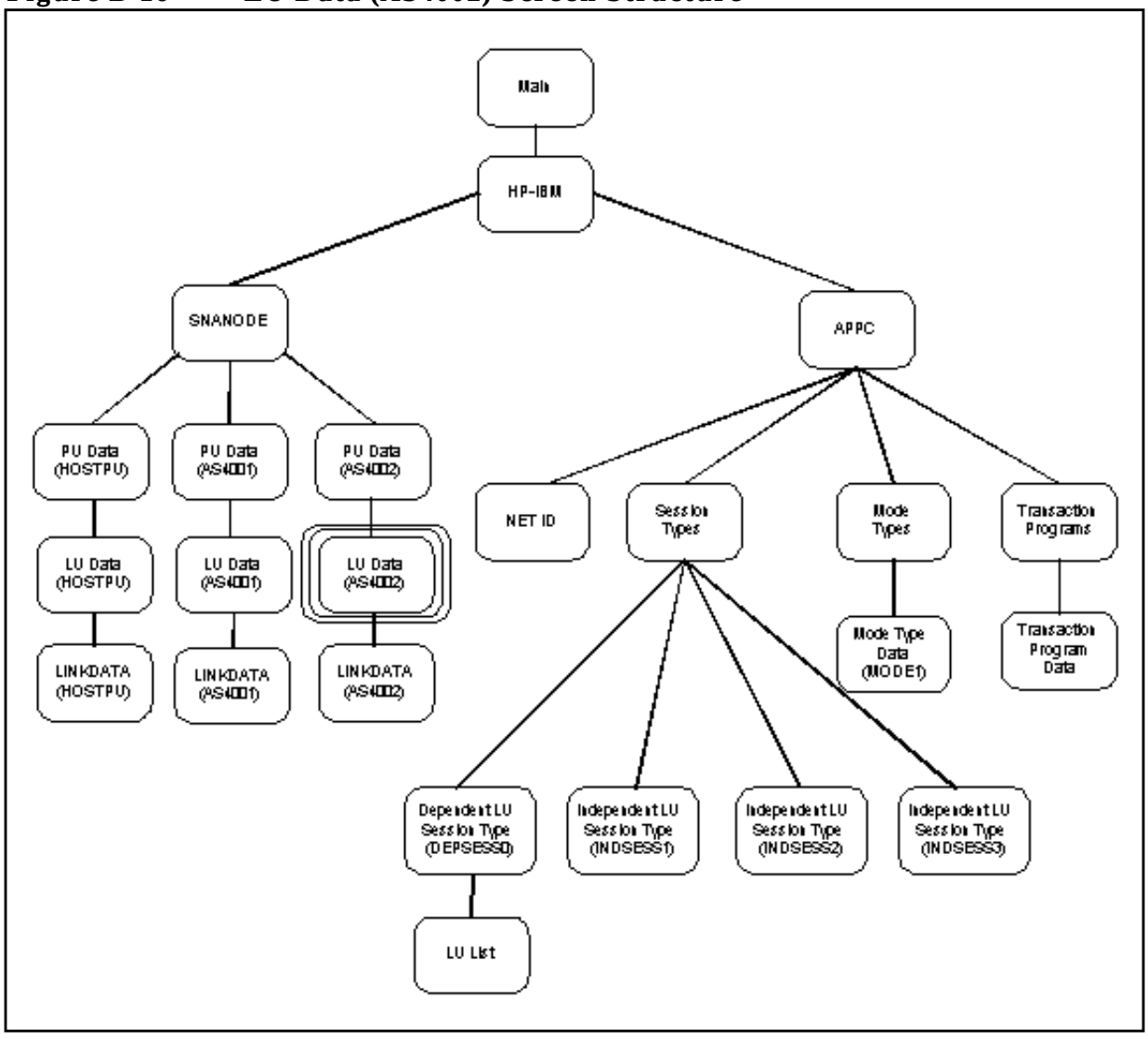
The LU name HPINDLU2 must match the RMTLOCNAME (remote location name) configured in the Device Description on the AS/400.

Figure B-15 Example SNANODE LU Data Screen (AS4002)



The following diagram shown in Figure B-16 shows how the “SNA Node Configuration: LU Data” screen for AS4002 fits into the NMMGR screen structure. The LU configured here will be entered in the “APPC: Independent LU Session Type Data” screen for session type INDSESS3.

Figure B-16 LU Data (AS4002) Screen Structure

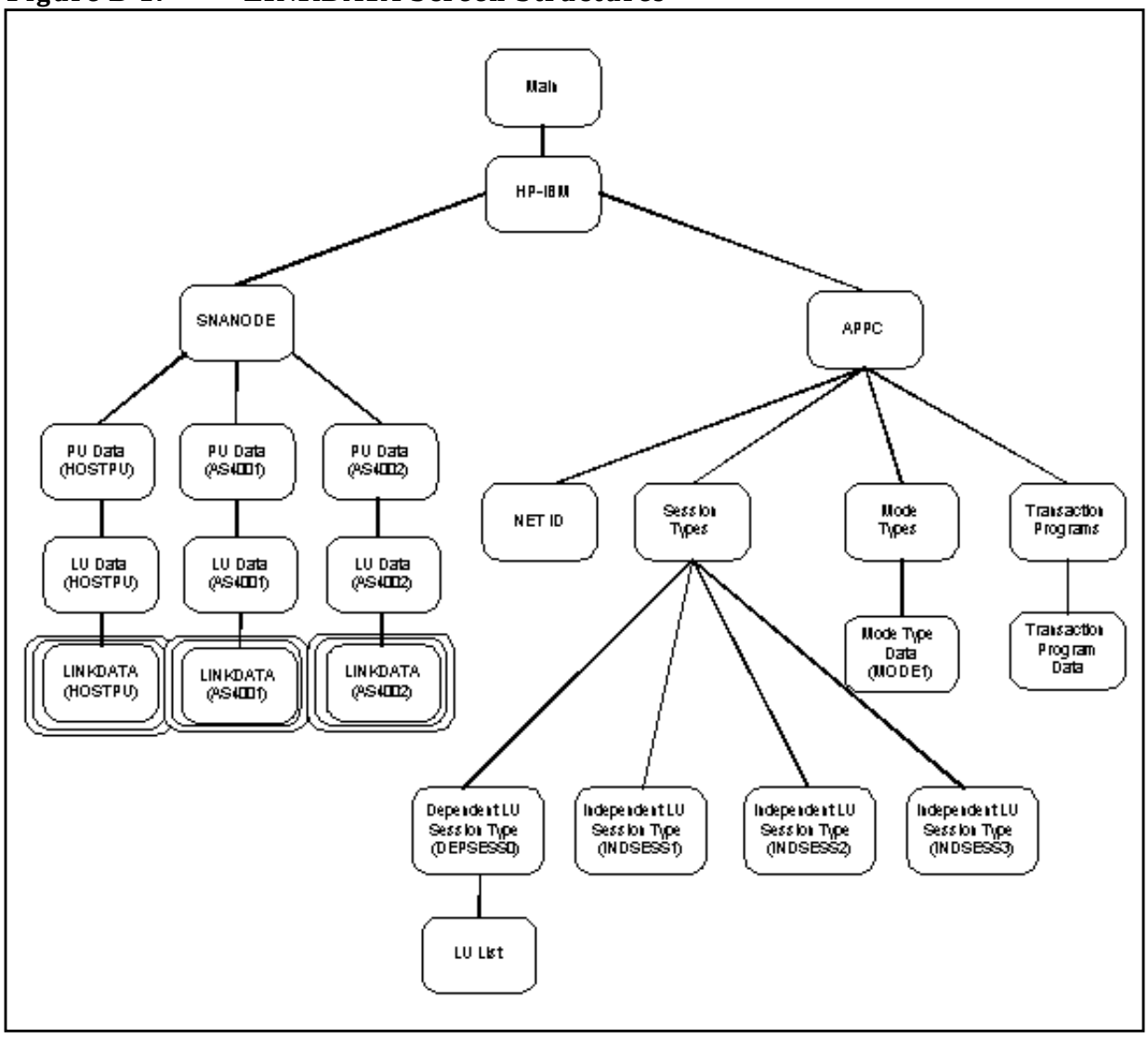


SNA Node Configuration: SDLC Link Data screen

Every node configured in the "SNA Node Configuration" screen must have its communications link parameters and phone data configured in the "SNA Node Configuration: SDLC Link Data" screen. The "SNA Node Configuration: SDLC Link Data" screen is not shown here, but it is fully documented in the *SNA Link/XL Node Manager's Guide*.

The following diagram shown in Figure B-17 shows how the "SNA Node Configuration: SDLC Link Data" screens for `HOSTPU`, `AS4001` and `AS4002` fit into the NMMGR screen structure.

Figure B-17 LINKDATA Screen Structures

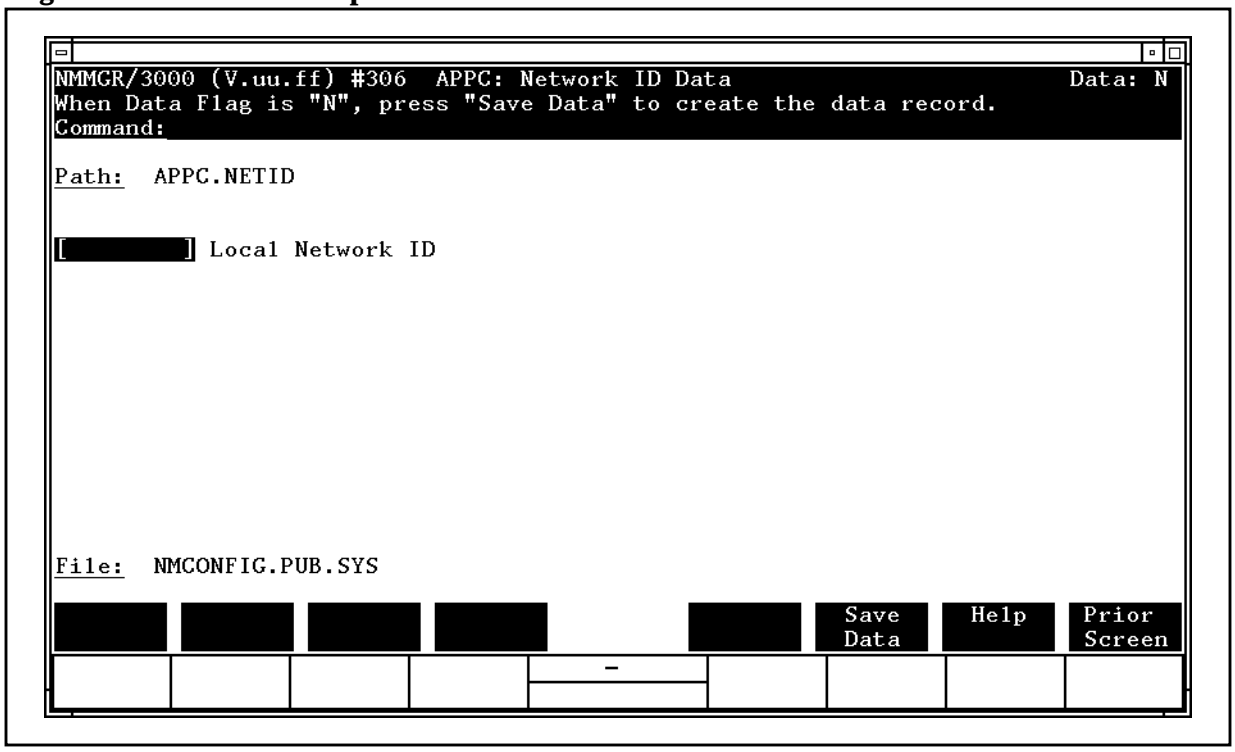


APPC: Network ID Data Screen

If your HP 3000 will operate as a Type 2.1 node, you must configure the identifier for your local network in the “APPC: Network ID Data” screen. Chapter Figure B-18, “Example Network ID Data Screen.” shows the “APPC: Network ID Data” screen for the example configuration illustrated in this appendix.

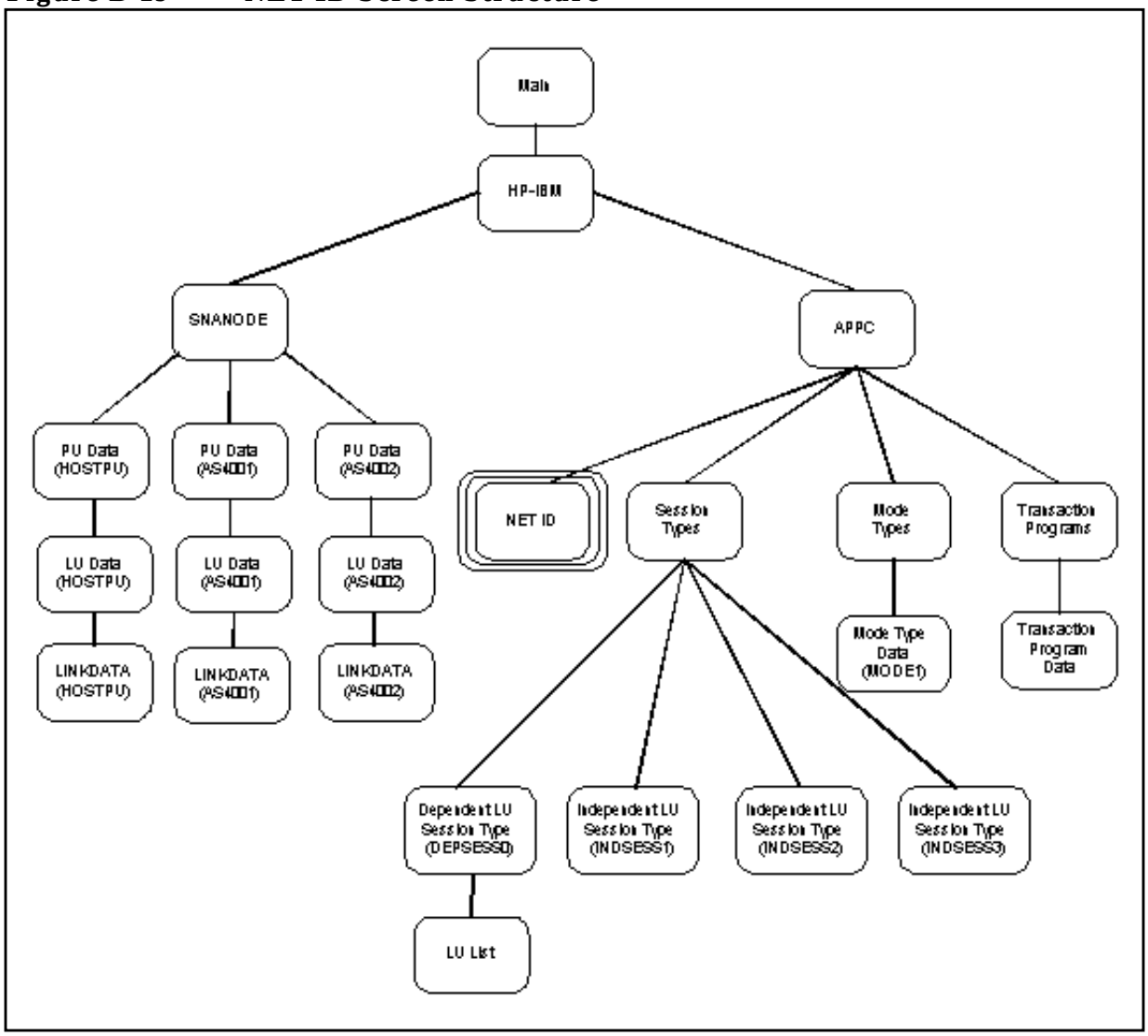
The `Local Network ID` matches the `RMTNETID` configured in the Device Description on the AS/400.

Figure B-18 Example Network ID Data Screen



The following diagram shown in Figure B-19 shows how the “APPC: Network ID Data” screen fits into the NMMGR screen structure.

Figure B-19 NET ID Screen Structure



APPC: Select Session Type Screen

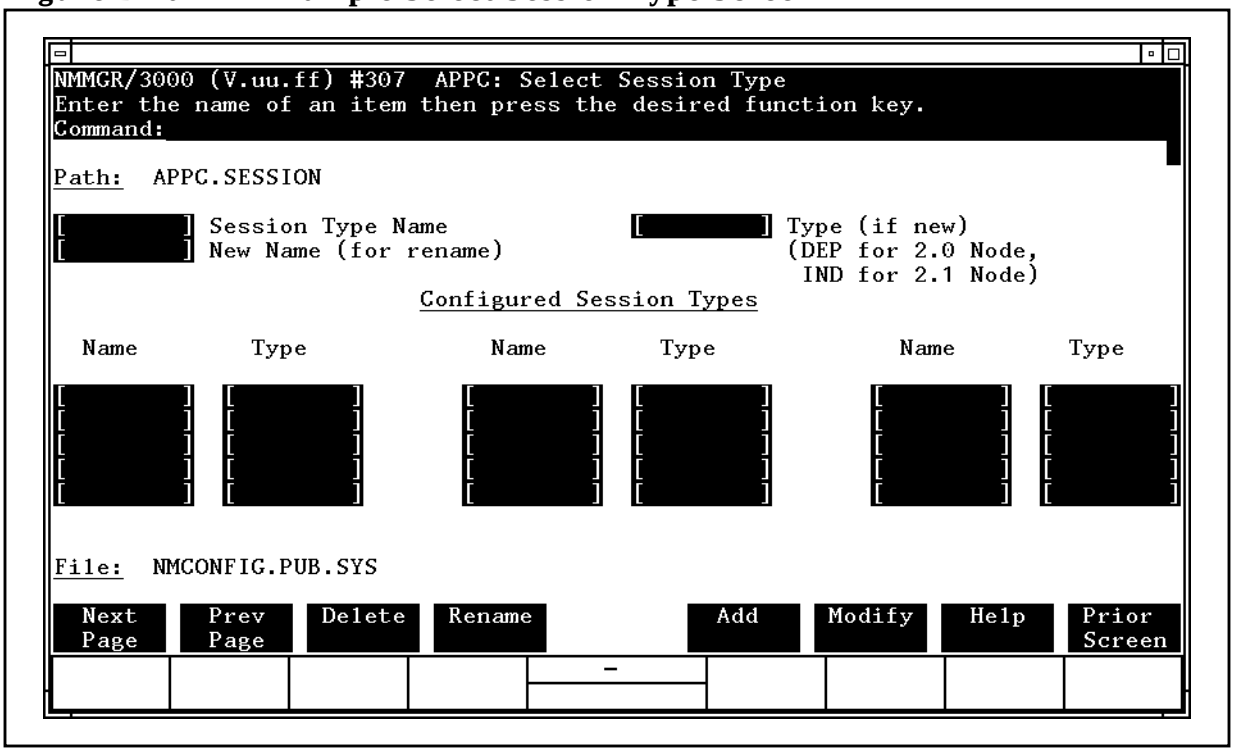
Figure B-20 shows the "APPC: Select Session Type" screen. Four session types are configured for the APPC subsystem: DEPSESS1, INDSESS1, INDSESS2, and INDSESS3.

DEPSESS1 communicates with an IBM mainframe through node HOSTPU.

INDSESS1 and INDSESS2 communicate with IBM AS/400 #1 through node AS4001.

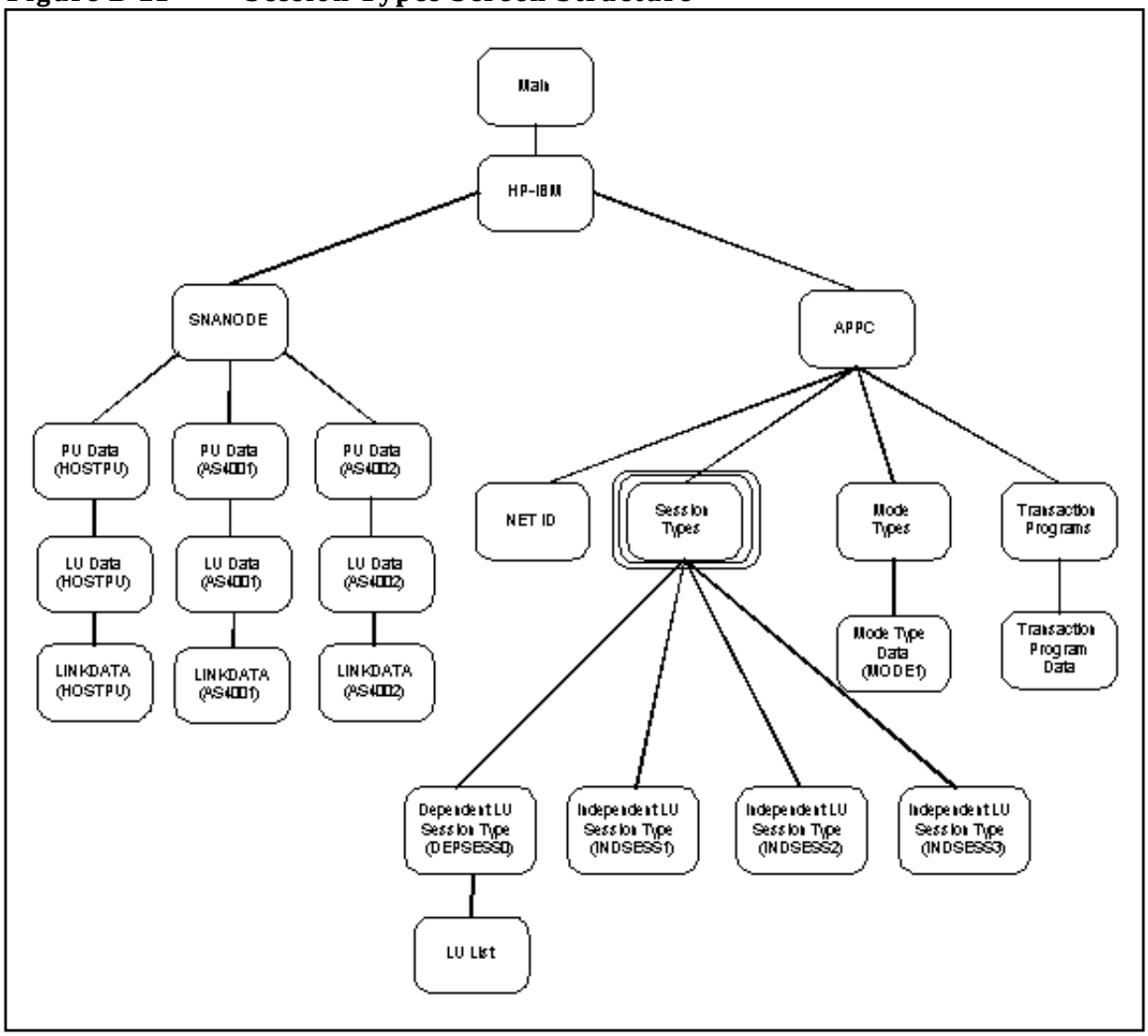
INDSESS3 communicates with IBM AS/400 #3 through node AS4002. AS/400 #3 is not directly connected to the HP 3000; sessions of type INDSESS3are routed to AS/400 #3 through AS/400 #2, which is directly connected to the HP 3000.

Figure B-20 Example Select Session Type Screen



The following diagram shown in Figure B-21 shows how the “APPC: Select Session Type” screen fits into the NMMGR screen structure.

Figure B-21 Session Types Screen Structure



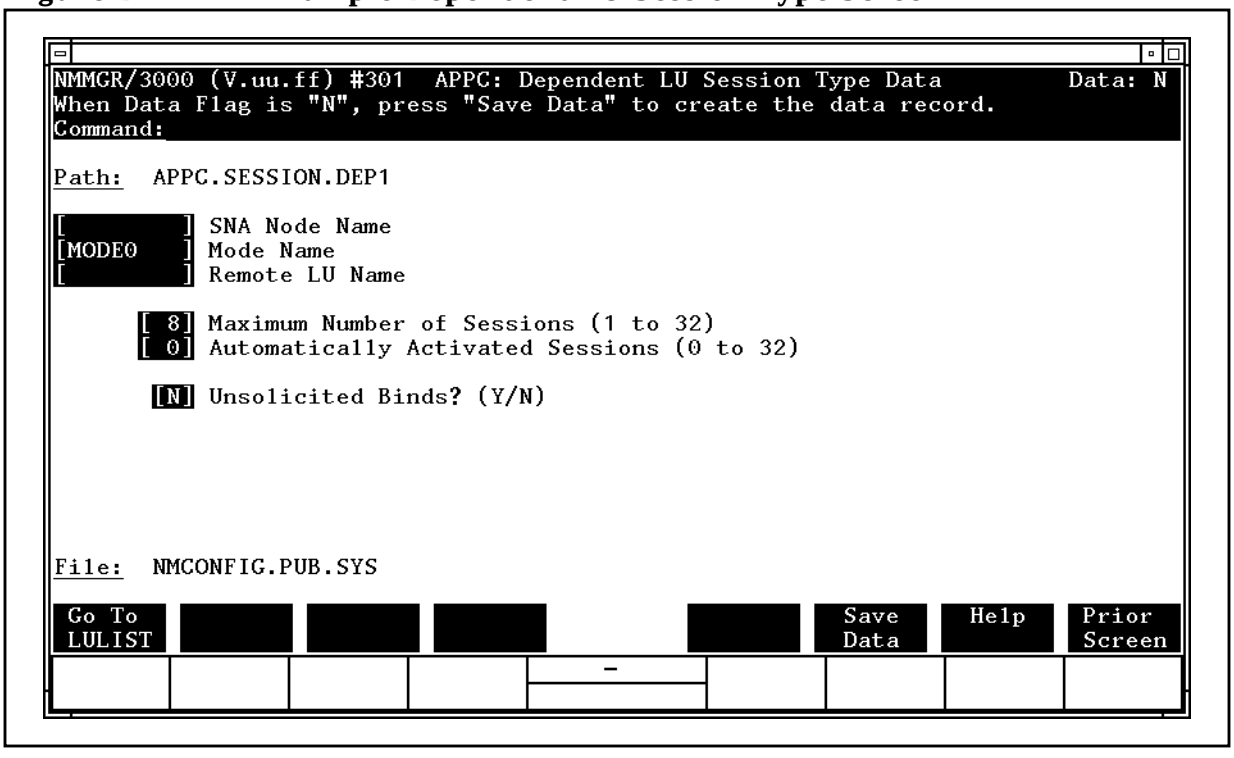
APPC: Dependent LU Session Type Data Screen

Figure B-22 shows the “APPC: Dependent LU Session Type Data” screen for session type DEPSESS1. Two APPC sessions are configured to activate automatically at subsystem startup. Beyond the automatically activated sessions, two more sessions can be activated before the maximum session limit for DEPSESS1 is reached. See Chapter 5 , “Managing the APPC Subsystem,” for more information on session activation.

The SNA node name (HOSTPU) configured in this screen must first be configured in the “SNA Node Configuration” screen. The LUs associated with DEPSESS1 are configured in the “APPC: Dependent LU List Data” screen. They must first be configured in the “SNA Node Configuration: LU Data” screen for HOSTPU.

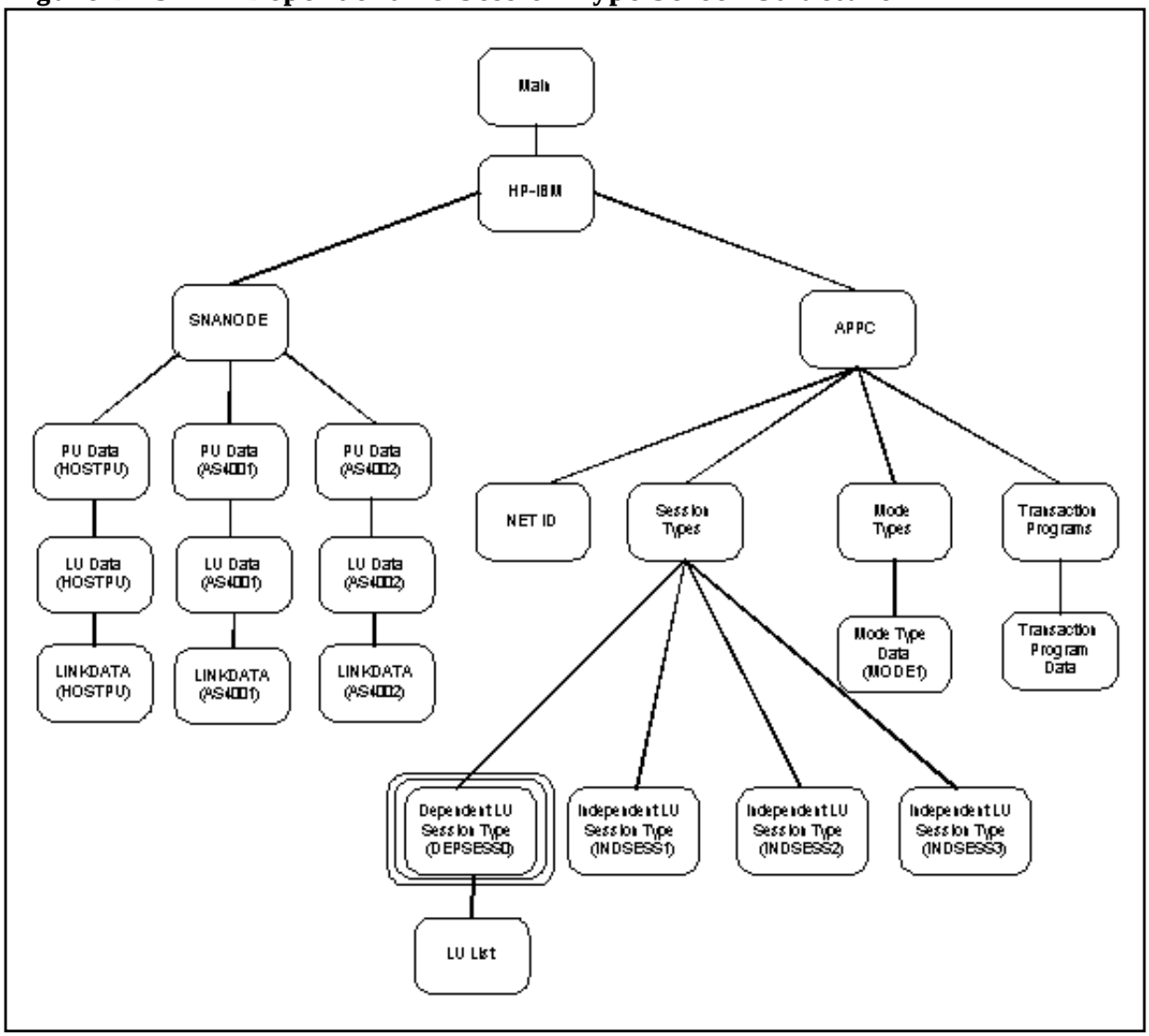
The sessions of type DEPSESS1 use mode type MODE0. Mode type MODE0 is internally defined, so it is not explicitly configured in the “APPC: Select Mode Type” and “APPC: Mode Type Data” screens, shown later in this appendix. For more information on mode types and mode type configuration, see Chapter 4 , “APPC Subsystem Configuration,” of this manual.

Figure B-22 Example Dependent LU Session Type Screen



The following diagram shown in Figure B-23 shows how the “APPC: Dependent LU Session Type Data” screen fits into the NMMGR screen structure.

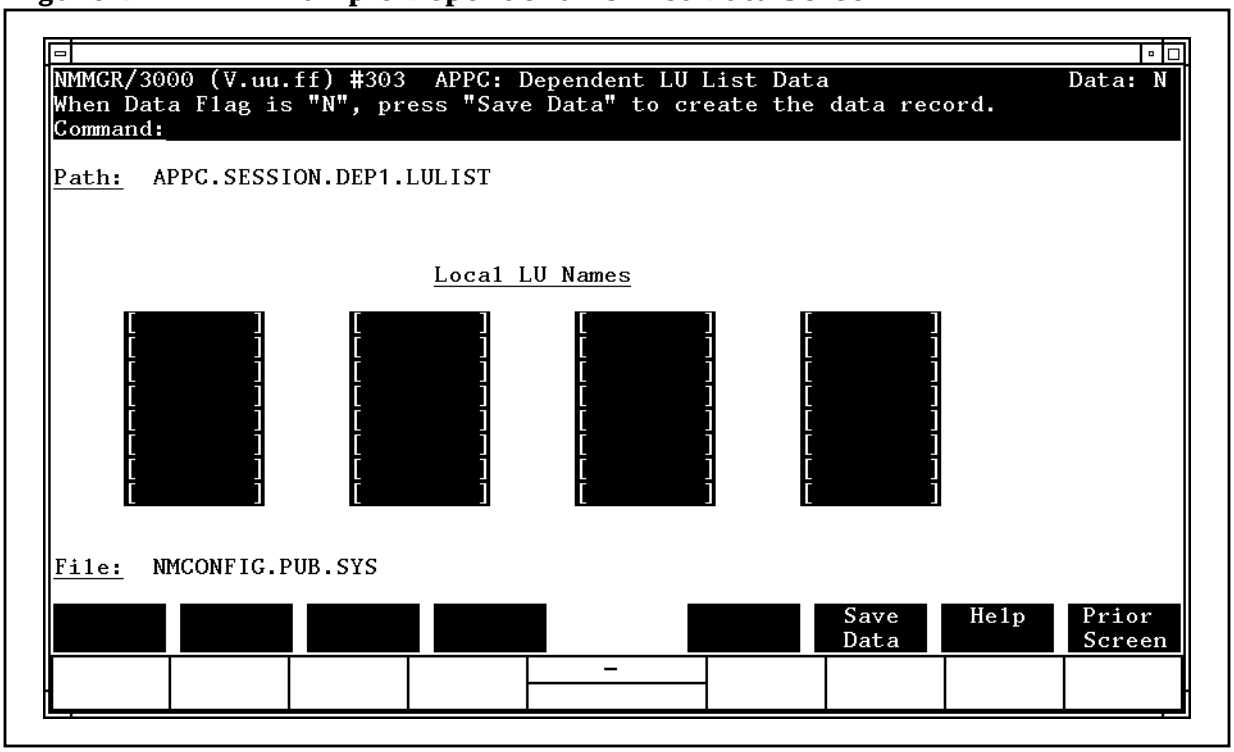
Figure B-23 Dependent LU Session Type Screen Structure



APPC: Dependent LU List Data Screen

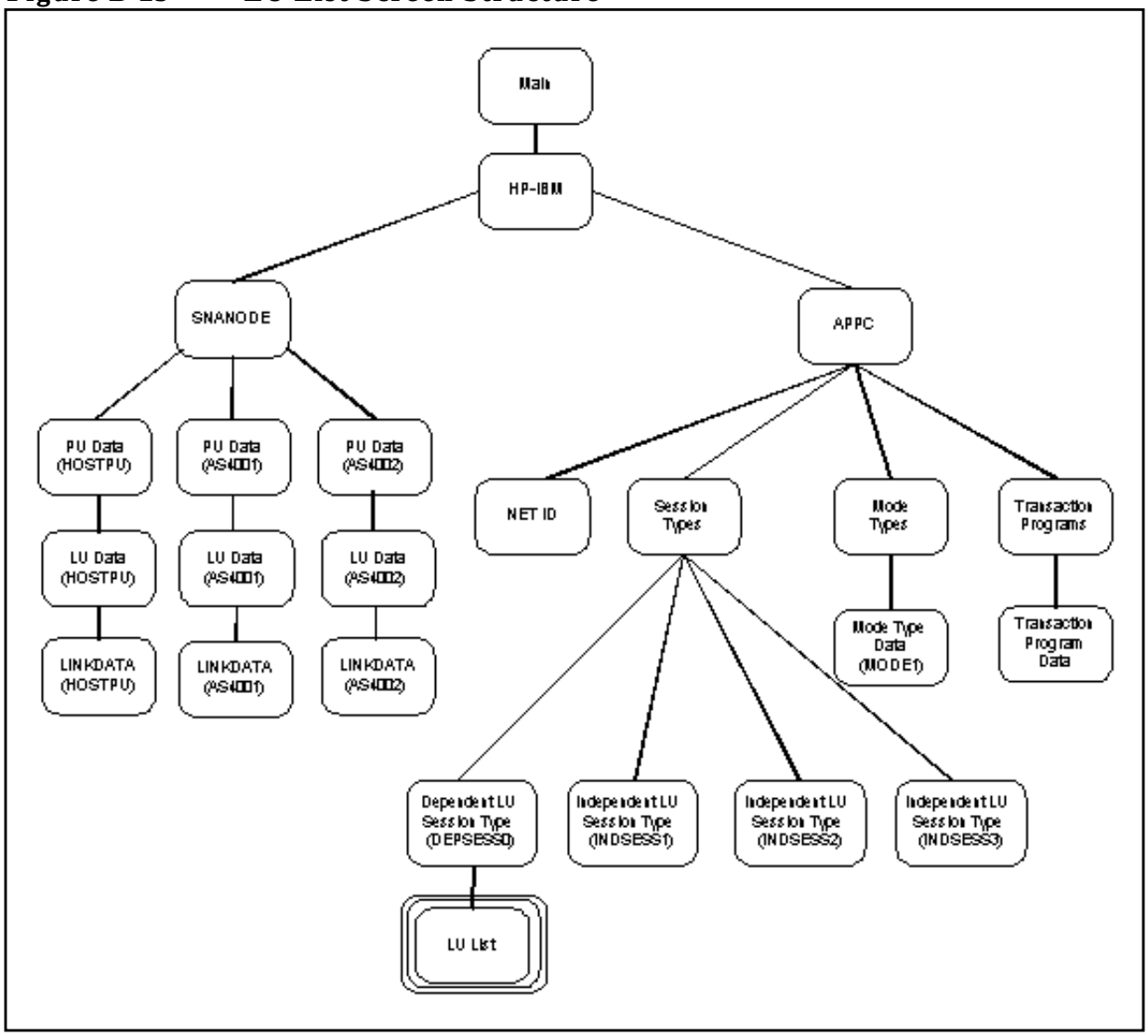
Figure B-24 shows the “APPC: Dependent LU List Data” screen for DEPSESS1. The LUs listed in this screen must first be configured in the “SNA Node Configuration: LU Data” screen for HOSTPU. Since the maximum session limit for DEPSESS1 is 4, there must be four LUs configured in the LU list.

Figure B-24 Example Dependent LU List Data Screen



The following diagram shown in Figure B-25 shows how the “APPC: Dependent LU List Data” screen fits into the NMMGR screen structure.

Figure B-25 LU List Screen Structure



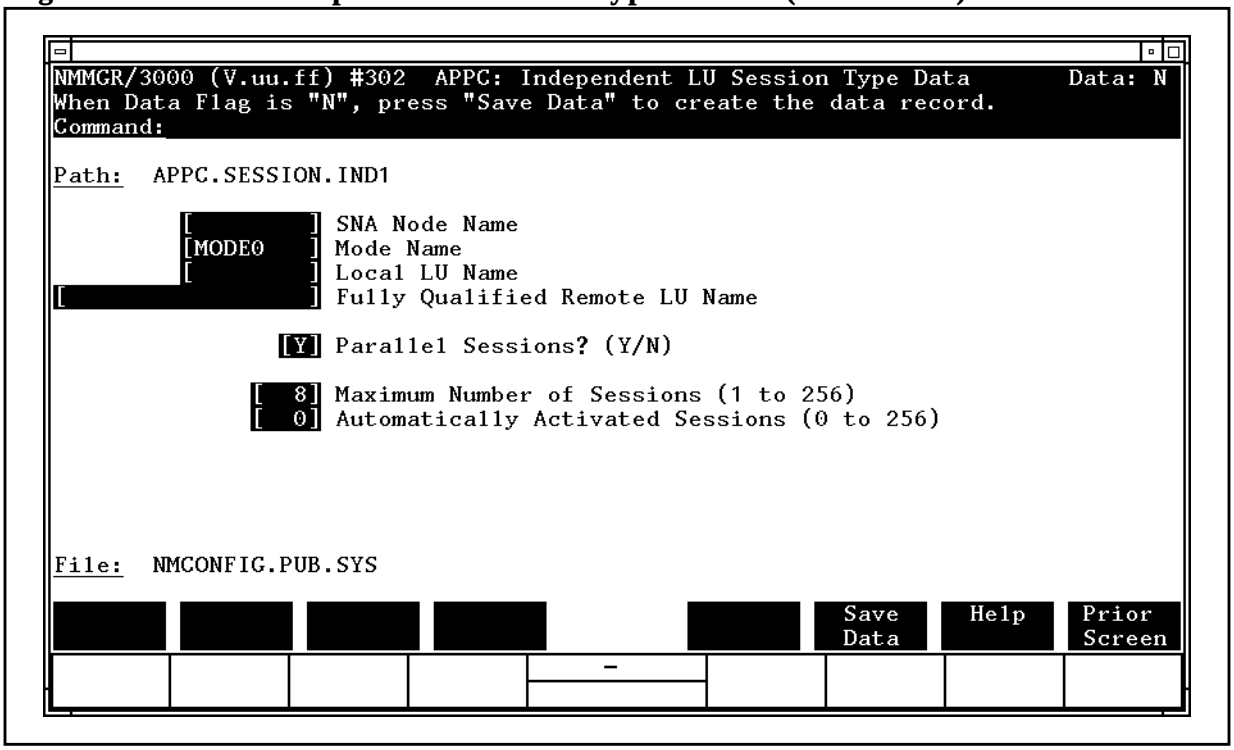
APPC: Independent LU Session Type Data Screens

Figure B-26 shows the "APPC: Independent LU Session Type Data" screen for INDSESS1. In this screen, INDSESS1 is configured to use node AS4001. Only one independent LU is configured for that node: HPINDLU1. HPINDLU1 is capable of participating in multiple, simultaneous sessions with both of the independent LUs on IBM AS/400 #1.

Sessions of type INDSESS1 are configured to communicate with remote LU BLUELU. Four sessions will be activated automatically at subsystem startup.

Once sessions have been automatically activated, you can change the number of active sessions and reapportion sessions among the configured session types by issuing the APPCCONTROL SESSIONS command. See Chapter 2 , "Interactive Control Operator Commands," for more information on APPCCONTROL commands.

Figure B-26 Independent Session Type Screen (INDSESS1)



The following diagram shown in Figure B-27 shows how the “APPC: Independent LU Session Type Data” screen for INDSESS1 fits into the NMMGR screen structure.

Figure B-27 Independent LU Session Type Screen Structure

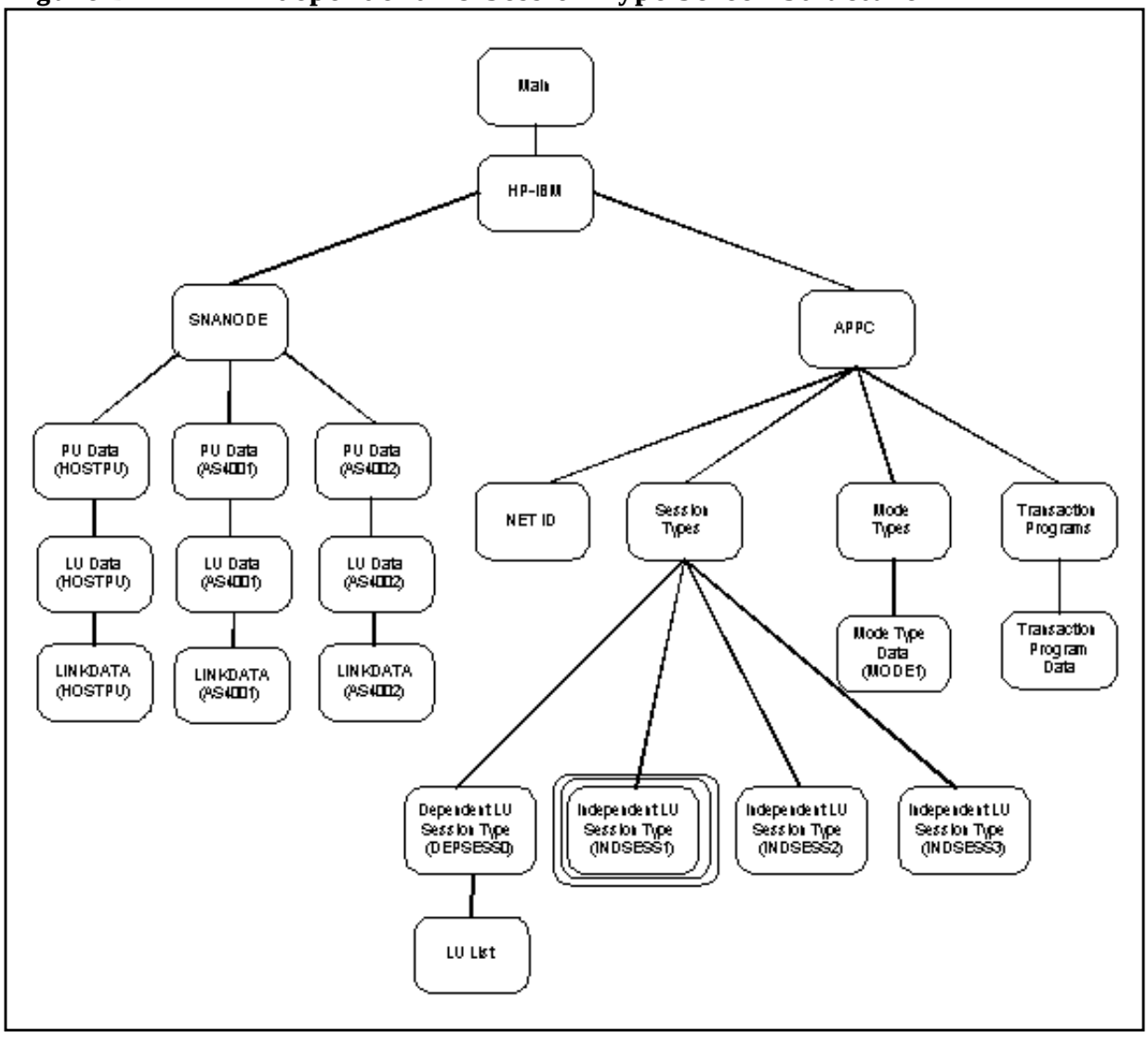
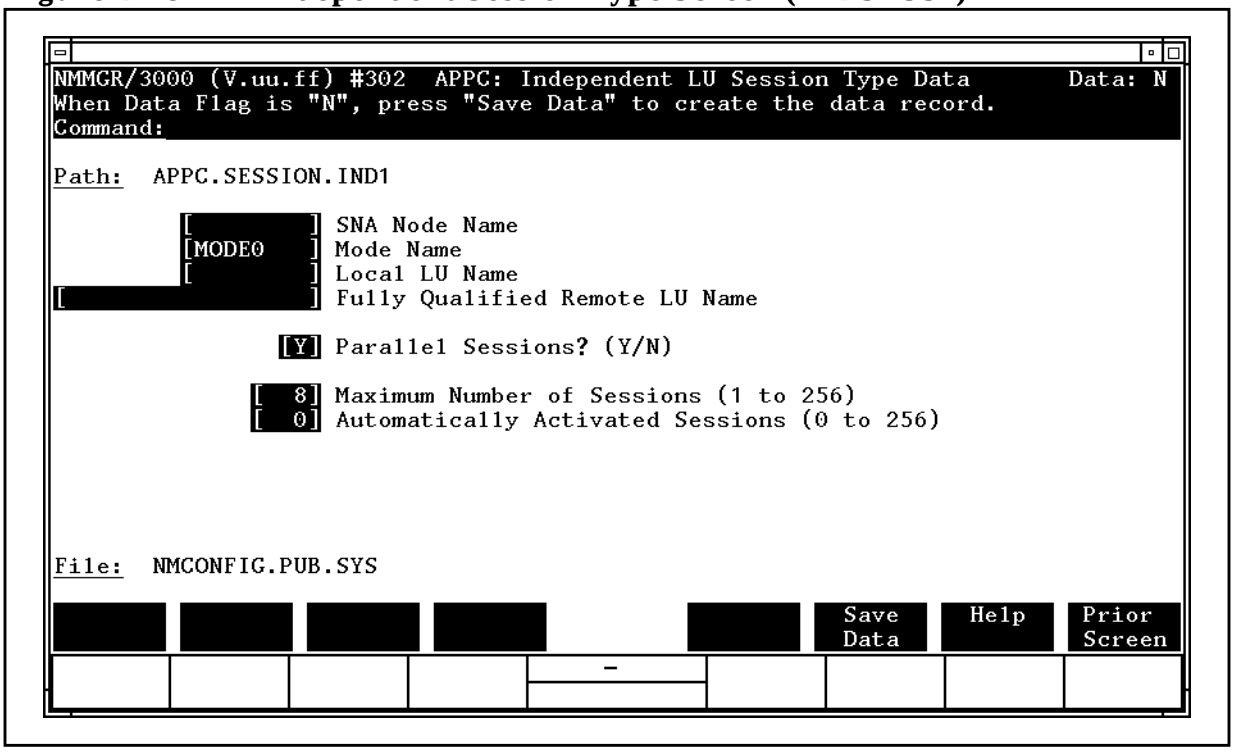


Figure B-28 shows the "APPC: Independent LU Session Type Data" screen for INDSESS2. INDSESS1 and INDSESS2 both use node AS4001. They also use the same independent LU, HPINDLU1. Sessions of type INDSESS2 are configured to communicate with remote LU GREENLU on AS/400 #1.

Session type INDSESS2 has parallel sessions disabled (Parallel Sessions = N), so only one session of this type can be active at a time. The value for Maximum Number of Sessions is the default that appears when you specify N for Parallel Sessions. INDSESS2 has an Automatically Activated Sessions value of 1. (The default is 0.)

Figure B-28 Independent Session Type Screen (INDSESS2)



The following diagram shown in Figure B-29 shows how the “APPC: Independent LU Session Type Data” screen for INDSESS2 fits into the NMMGR screen structure.

Figure B-29 Independent LU Session Type Screen Structure

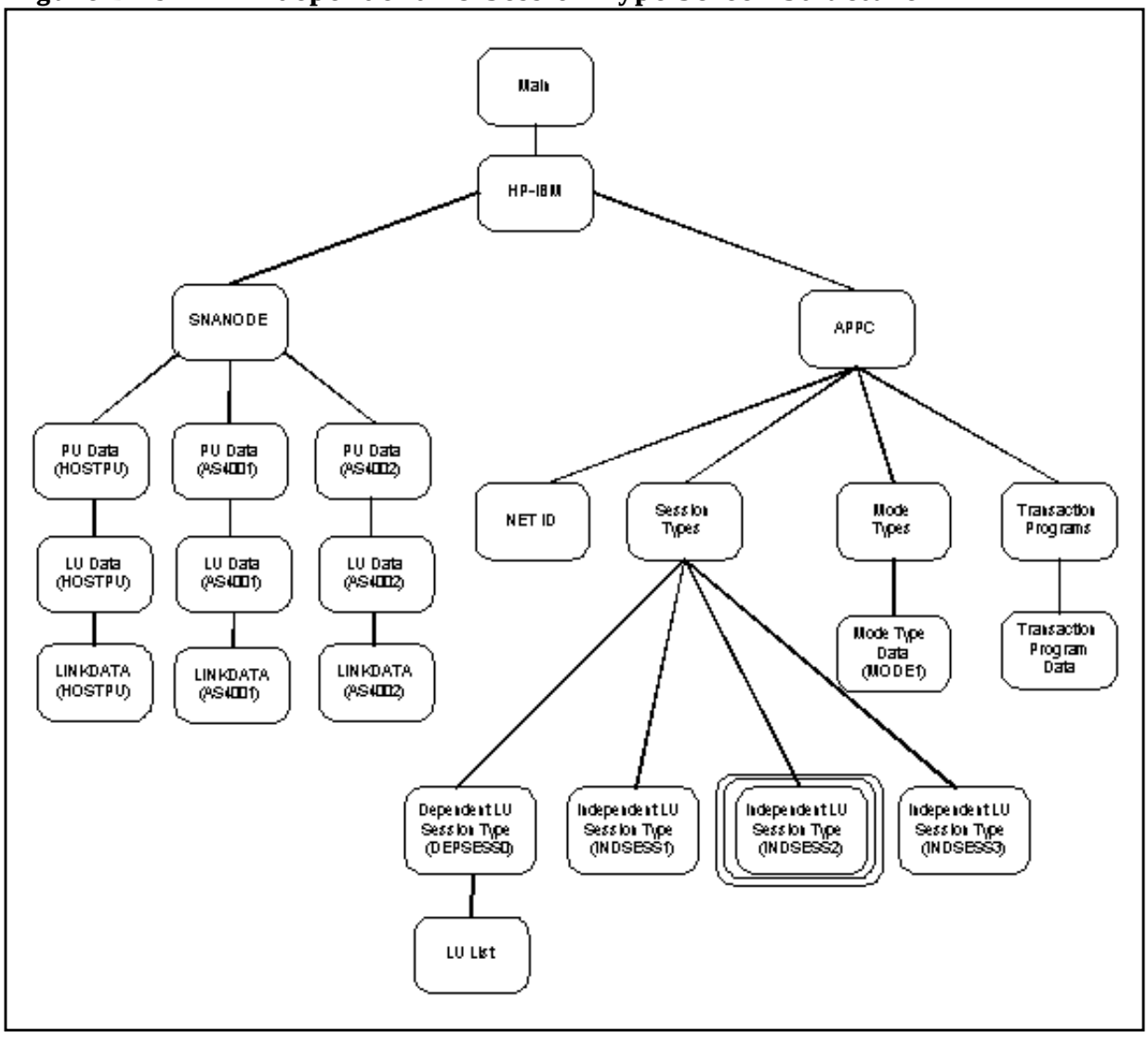
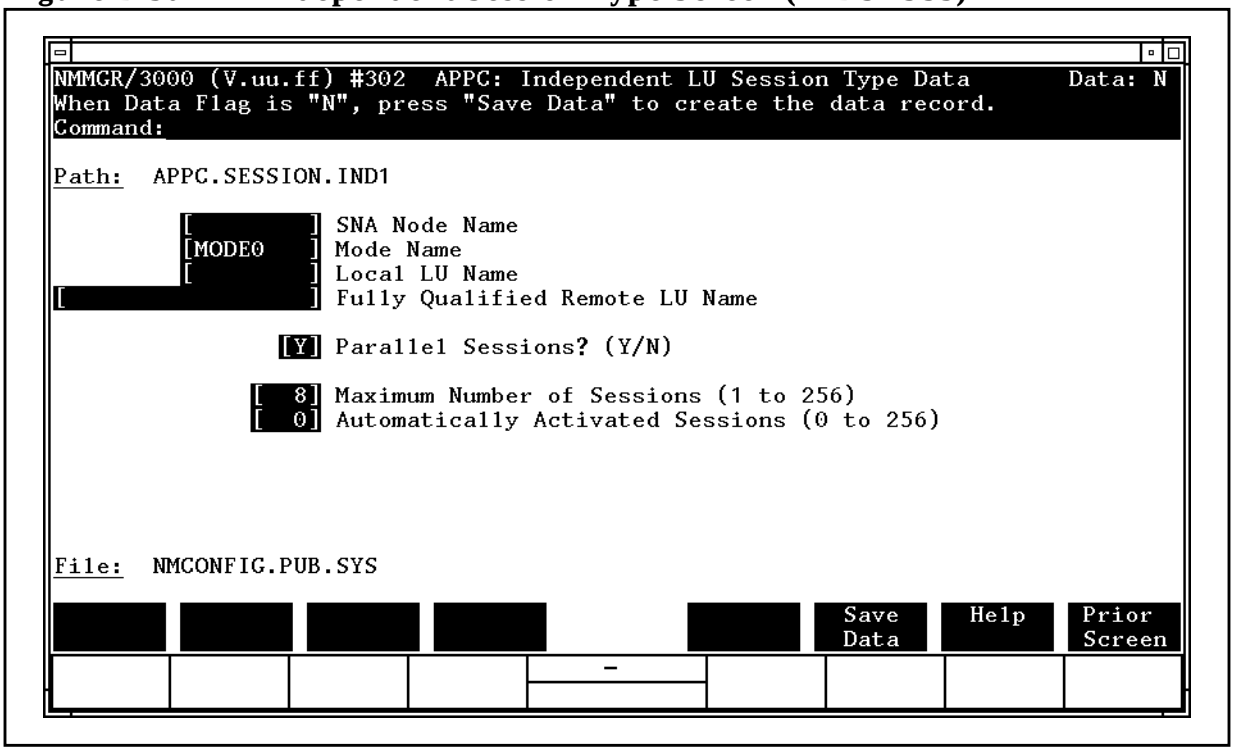


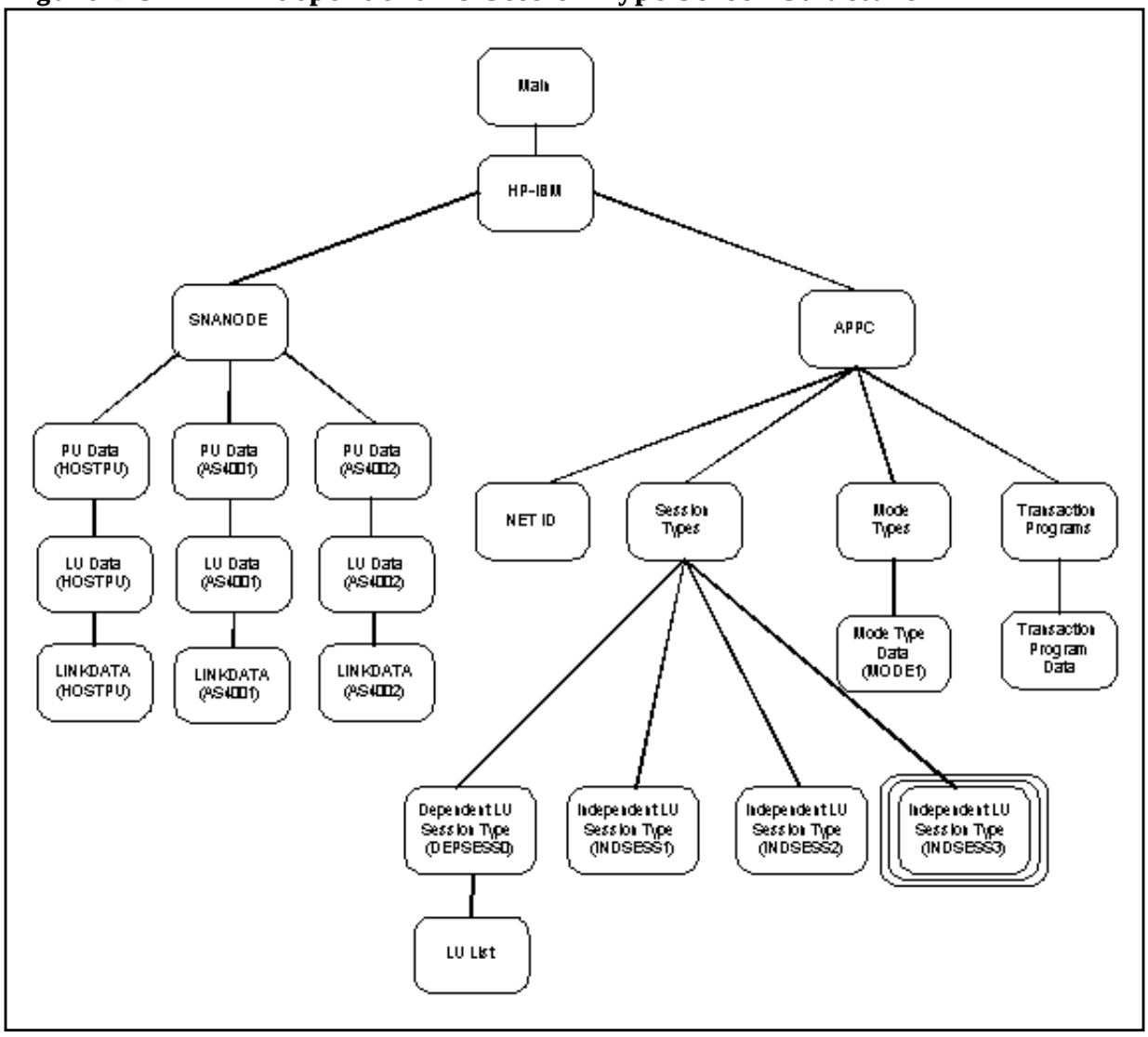
Figure B-30 shows the "APPC: Independent LU Session Type Data" screen for INDSESS3. Sessions of type INDSESS3 are configured to communicate with remote LU PURPLELU on AS/400 #3, which is indirectly connected to the HP 3000 through AS/400 #2. AS/400 #3 is located in NET2, a different network from the HP 3000 and AS/400 #2. AS/400 #2 identifies the location of the destination LU (NET2.PURPLELU) from the Fully Qualified Remote LU Name field of this screen.

Figure B-30 Independent Session Type Screen (INDSESS3)



The following diagram shown in Figure B-31 shows how the “APPC: Independent LU Session Type Data” screen for INDSESS3 fits into the NMMGR screen structure.

Figure B-31 Independent LU Session Type Screen Structure

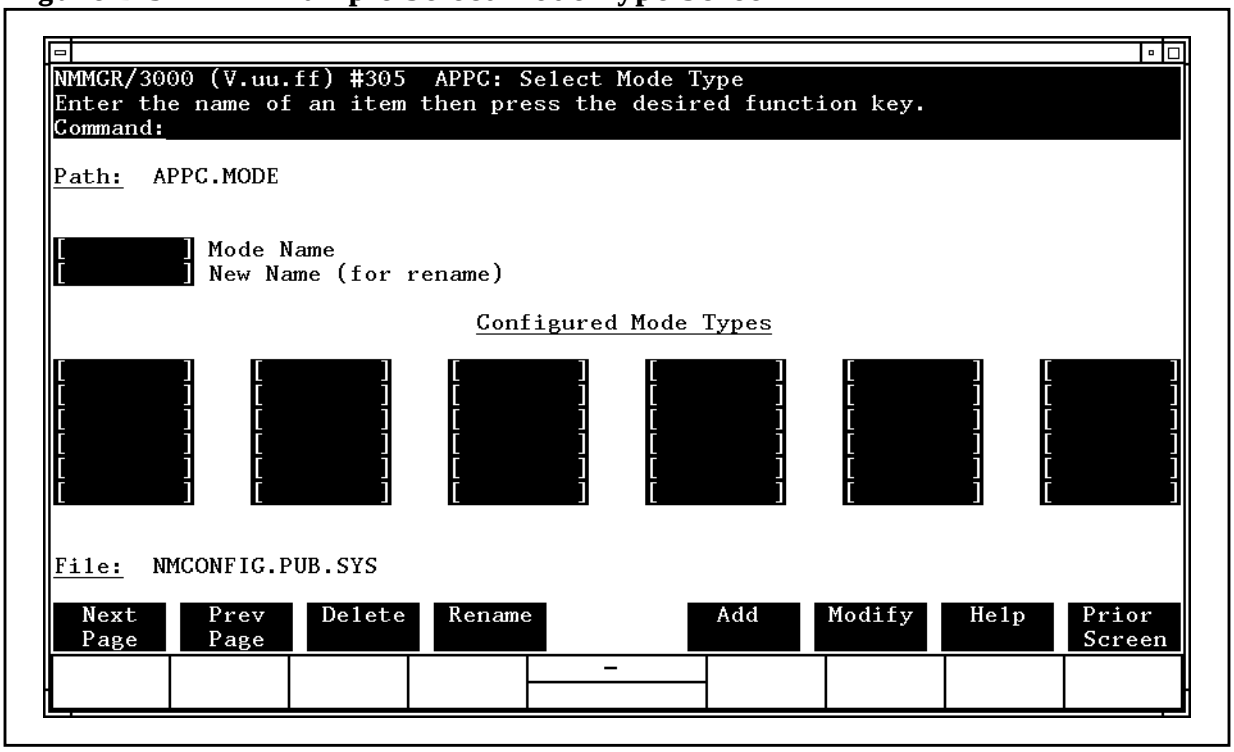


APPC: Select Mode Type Screen

Figure B-32 shows the “APPC: Select Mode Type” screen. The mode type MODE1, used by all the independent LU session types in this example configuration, is the only mode type configured here. The data for mode type MODE1 is configured in the “APPC: Mode Type Data” screen, shown in figure B-17. The mode type MODE0, used by the dependent LU session type in this example configuration, is a predefined default mode on the HP 3000 and does not appear in the “APPC: Select Mode Type” screen.

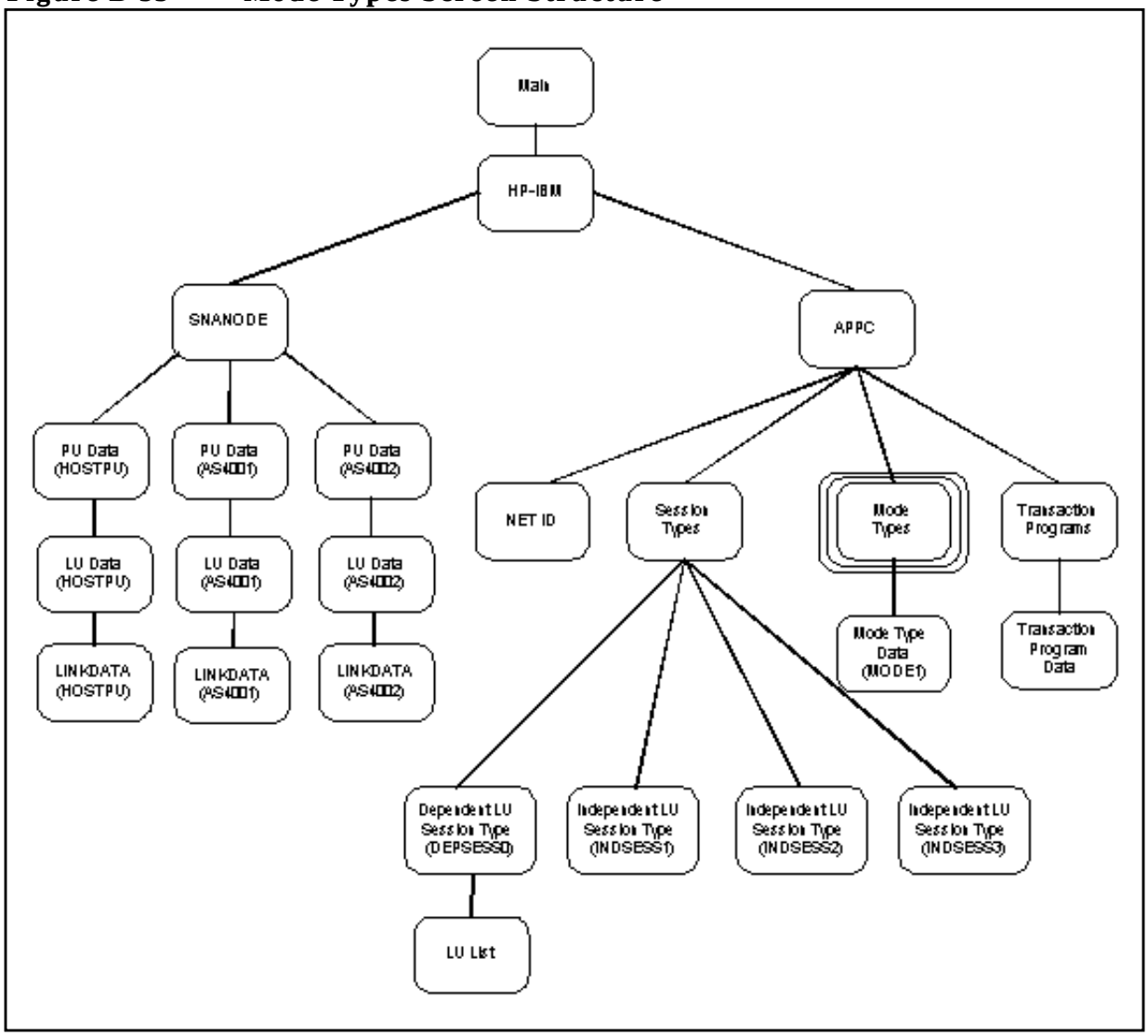
The Mode Name for a session type on the HP 3000 must match the mode name configured for the remote LU on the IBM system. The Mode Name configured here matches the MODE in the Device Description on the AS/400.

Figure B-32 Example Select Mode Type Screen



The following diagram shown in Figure B-33 shows how the “APPC: Select Mode Type” screen fits into the NMMGR screen structure.

Figure B-33 Mode Types Screen Structure



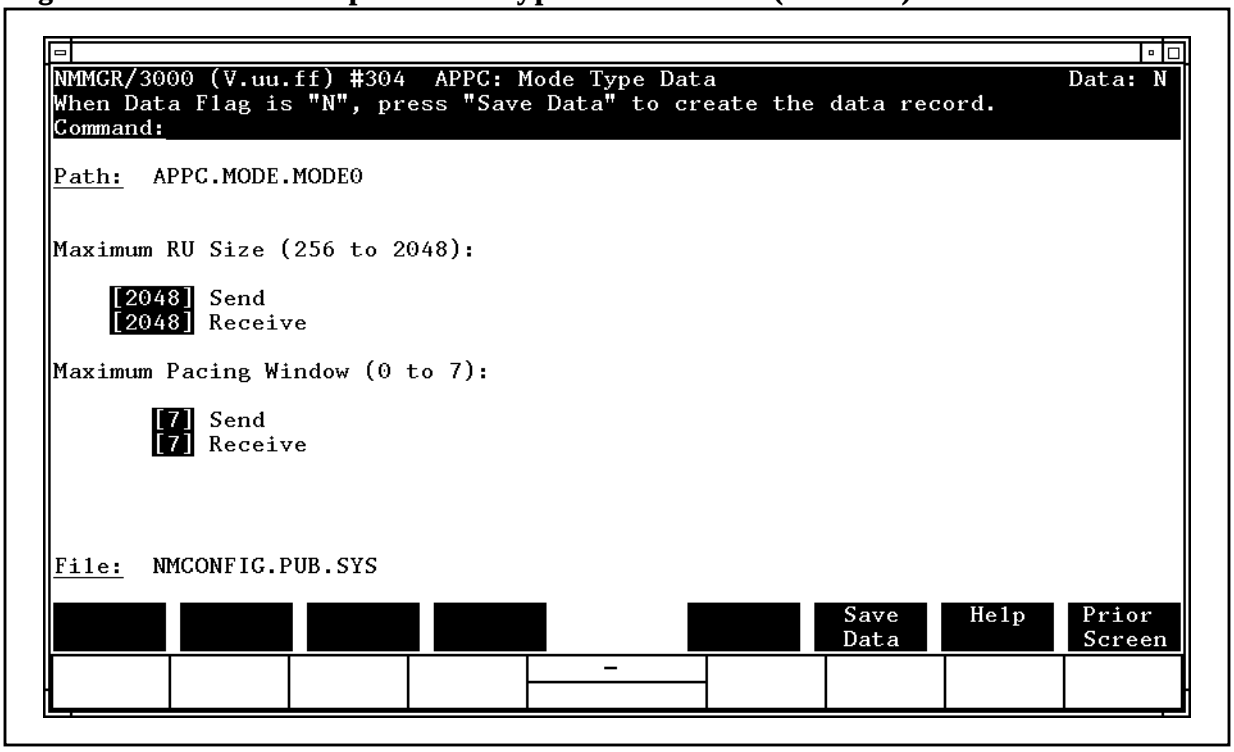
APPC: Mode Type Data Screen

Figure B-34 shows the “APPC: Mode Type Data” screen for mode type MODE1. All the independent LU session types in this example configuration use mode type MODE1. independent LUs on the HP 3000 can send and receive RUs no larger than 2048 bytes. This is the default maximum RU size for both sending and receiving.

The local LU can send 7 RUs before it must wait for a response from the remote LU. The local LU can receive 7 RUs from the remote LU before it must send a response. The default size of both the send and receive pacing windows is 7. See Chapter 4 , “APPC Subsystem Configuration,” for more information on mode types.

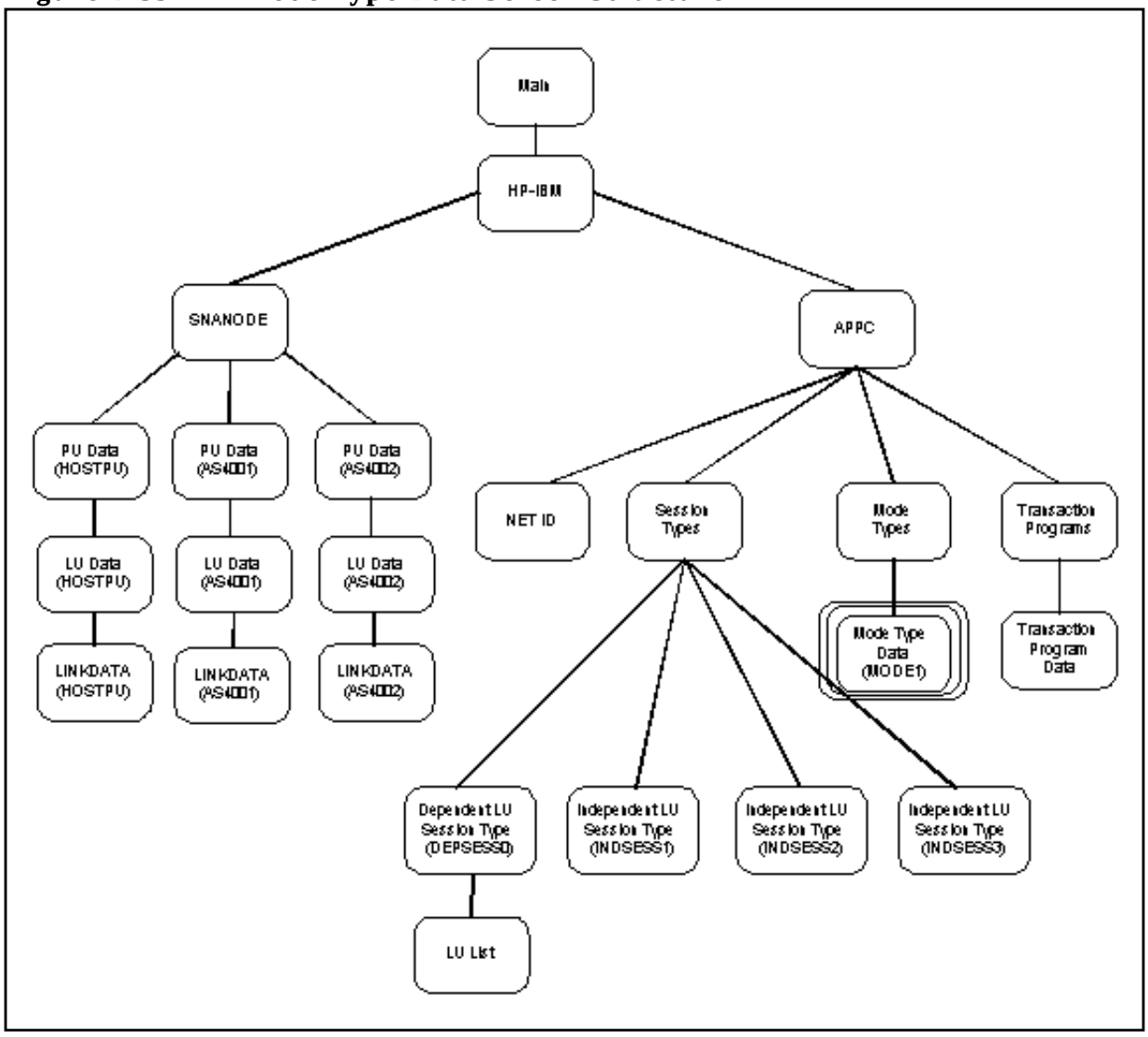
The values configured in this screen match the values configured in the Mode Description on the AS/400.

Figure B-34 Example Mode Type Data Screen (MODE1)



The following diagram shown in Figure B-35 shows how the “APPC: Mode Type Data” screen for mode type MODE1 fits into the NMMGR screen structure.

Figure B-35 Mode Type Data Screen Structure

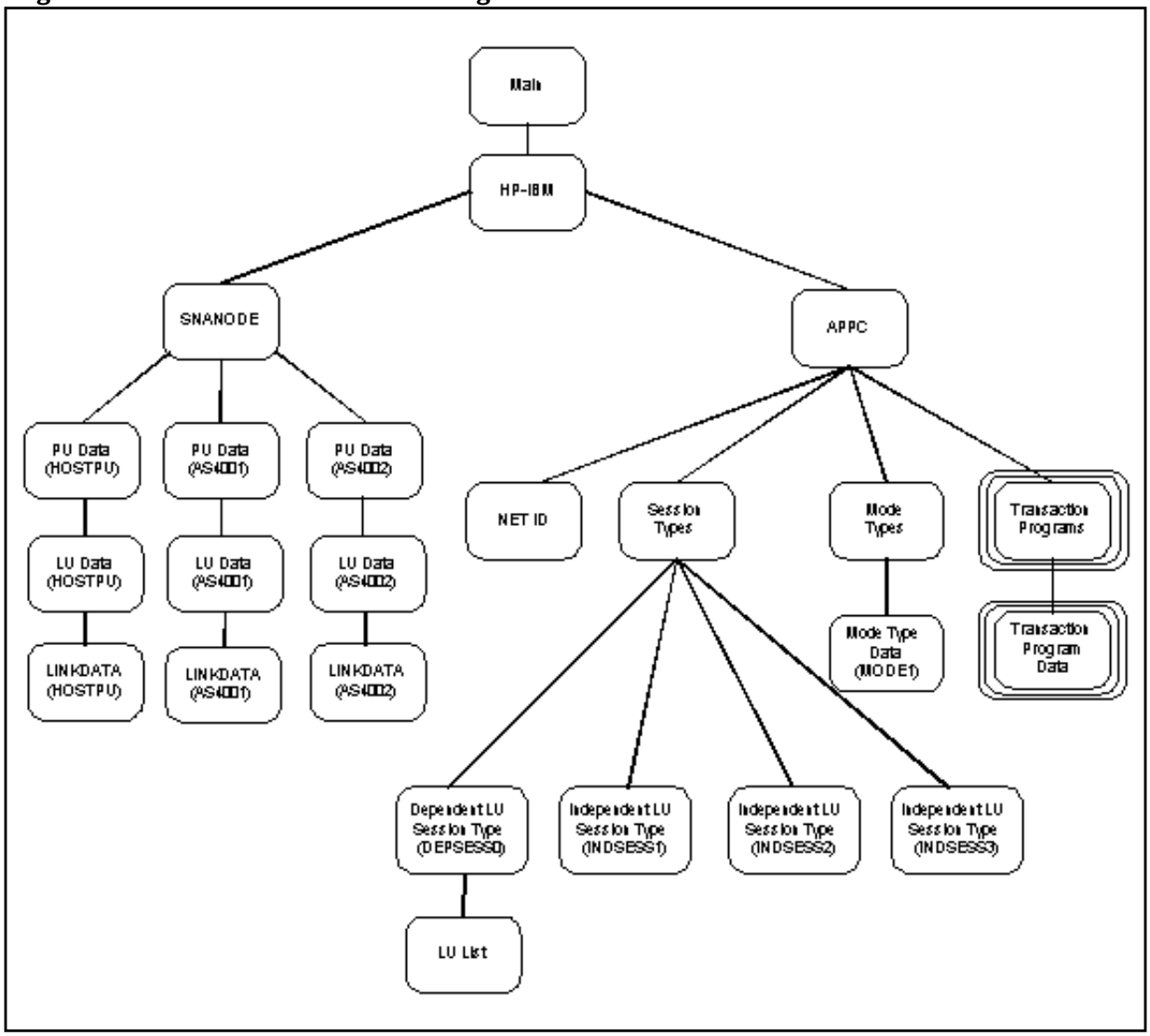


Transaction Program Configuration Screens

The “APPC: Select Transaction Program” and “APPC: Transaction Program Data” screens are used to configure transaction programs that call the `MCGetAllocate` intrinsic to accept allocate requests from remote LUs. The TP screens are not pictured here, but they are fully documented in Chapter 4 , “APPC Subsystem Configuration,” of this manual. For more information on remotely initiated TPs, see “Establishing Conversations,” in Chapter 5 , “Managing the APPC Subsystem.”

The following diagram shown in Figure B-36 shows how the “APPC: Select Transaction Program” and “APPC: Transaction Program Data” screens fit into the NMMGR screen structure.

Figure B-36 Transaction Programs Screen Structure



NMMGR Critical Summary

This is a sample critical summary reflecting the APPC configuration shown in this appendix.

```
                CRITICAL SUMMARY - APPC CONFIGURATION

THU, SEP 6, 1991,  2:32 PM
CONFIGURATION FILE NAME: NMCONFIG.PUB.SYS
Validated subsystem.

NETWORK ID:  NET1

SESSION NAME:  INDSESS1           TYPE:  IND
   SNA Node Name:  AS4001               MODE Name:         MODE1
   Local LU Name:  HPINDLU1             Remote LU Name:  NET1.BLUELU

   Parallel Sessions:  Y

   Max Sessions:                        8
   Auto Active Count:                   4

SESSION NAME:  INDSESS2           TYPE:  IND
   SNA Node Name:  AS4001               MODE Name:         MODE1
   Local LU Name:  HPINDLU1             Remote LU Name:  NET1.GREENLU

   Parallel Sessions:  N

   Max Sessions:                        1
   Auto Active Count:                   1

SESSION NAME:  INDSESS3           TYPE:  IND
   SNA Node Name:  AS4002:              MODE Name:         MODE1
   Local LU Name:  HPINDLU2             Remote LU Name:  NET2.PURPLELU

   Parallel Sessions:  Y

   Max Sessions:                         8
   Auto Active Count:                    2
SESSION NAME:  DEPSESS1           TYPE:  DEP
   SNA Node Name:  HOSTPU               MODE Name:         MODE0
   Remote LU Name:  CICS1

   Unsolicited Binds:  N

   Max Sessions:       4
   Auto Active Count:  2

   LOCAL LU NAMES:
      LU002    LU003    LU004    LU005

MODE NAME:  MODE1
   Maximum RU Size       --  SEND:  2048  RECEIVE:  2048
   Maximum Pacing Window --  SEND:  7     RECEIVE:  7
```

Remote Configuration

Figure B-37 shows sample lines from an NCP gen that correspond to values configured on the HP 3000.

Figure B-37 Sample Lines from an NCP Gen

```
************************************************
*
LN044    LINE       ADDRESS=044
*
*  This is the LINEID.  It isn't configured on the HP 3000.
*
HOSTPU   PU         ADDR=C1
*
*  PU name on the host corresponds to SNA Node Name on the HP 3000.
*  (SNA Node Name is configured in the "SNA Node Configuration" screen,
*  and the "APPC: Dependent LU Session Type Data" screen.)
*
LU00002  LU         LOCADDR=02
LU00003  LU         LOCADDR=03
LU00004  LU         LOCADDR=04
LU00005  LU         LOCADDR=05
*
*  LU name on the host (e.g. LU00002) corresponds to LU Name on the
HP 3000.
*  (LU names are listed in the "SNA Node Configuration: LU Data" screen
*  and in the "APPC: Dependent LU List Data" screen.)
*  The LU# field of the "SNA Node Configuration: LU Data" screen
*  is a decimal representation of the LOCADDR.
*
```

Figure B-38 shows sample lines from an AS/400 configuration that correspond to values configured on the HP 3000. The lines in Figure B-38 are from the configuration file on AS/400 #1, in this example configuration.

Figure B-38 Sample Lines from an AS/400 Configuration

```
Display Mode Description
Mode description name. . . . . .:   MODD        MODE1
Maximum number of sessions . . . . . .:   MAXSSN      8
Locally controlled sessions. . . . . .:   LCLCTLSSN   0
Inbound pacing value . . . . . .:   INPACING    7
Outbound pacing value. . . . . .:   OUTPACING   7

@COMPUTERTEXTW = Display Line Description
Line description . . . . . .:   LIND        SDLCL1
Data link role . . . . . .:   ROLE        *PRI
Exchange identifier. . . . . .:   EXCHID      05600000

@COMPUTERTEXTW = Display Controller Description
Controller description . . . . . .:   CTLD        HP3000
Link type. . . . . .:   LINKTYPE    *SDLC
APPN-capable . . . . . .:   APPN        *YES
Attached nonswitched line. . . . . .:   LINE        SDLCL1
Attached devices . . . . . .:   DEV         HPINDLU1
Remote control point name. . . . . .:   RMTCPNAME   AS4001
Data link role . . . . . .:   ROLE        *SEC
Maximum frame size . . . . . .:   MAXFRAME    265
Exchange identifier. . . . . .:   EXCHID      05600001
APPN node type . . . . . .:   NODETYPE    *LENNODE

@COMPUTERTEXTW = Display Device Description
Device description . . . . . .:   DEVD        HPINDLU1
Remote location name . . . . . .:   RMTLOCNAME  HPINDLU1
Attached controller. . . . . .:   CTL         HP3000
Local location name. . . . . .:   LCLLOCNAME  BLUELU
Remote network identifier. . . . . .:   RMTNETID    NET1
Mode . . . . . .:   MODE        MODE1
APPN-capable . . . . . .:   APPN        *YES
Single session . . . . . .:   SNGSSN      *NO
```

C Configuration Worksheets

The worksheets in this appendix are provided to help you plan your APPC subsystem configuration. Before you fill them out, you should read Chapter 4 , “APPC Subsystem Configuration.”

At the beginning of Chapter 4 , “APPC Subsystem Configuration,” is a section called “Data Required from the Remote Configuration.” You will need the information listed in that section to complete the configuration worksheets.

NOTE

You must configure all the SNA nodes and LUs you will use before you configure the APPC subsystem. SNA node configuration is described in the *SNA Link/XL Node Manager's Guide.*

This appendix contains the following:

- The “Completing the Worksheets” section gives step-by-step instructions for filling out the **Independent LU Session Types** worksheet, the **Dependent LU Session Types** worksheet, and the **Mode Types** worksheet.
- The “Example Worksheets” are completed worksheets for the example configuration described in Appendix B , “Sample Configuration.”
- One copy of the **Independent LU Session Types** worksheet, one copy of the **Dependent LU Session Types** worksheet, and one copy of the **Mode Types** worksheet are included at the end of this appendix.

Make some photocopies of the worksheets before you start filling them out so you will have enough room to plan your whole configuration.

Completing the Worksheets

This section gives you step-by-step instructions for filling out the Independent LU Session Types worksheet, the Dependent LU Session Types worksheet, and the Mode Types worksheet. One copy of each worksheet is included at the end of this appendix. Make some photocopies of the worksheets before you start filling them out so that you will have enough room to plan your whole configuration.

Completing the Independent LU Session Types Worksheet

An independent LU session type is used to communicate with an independent LU on a peer (Type 2.1) node. Follow these steps to complete the Independent LU Session Types worksheet:

1. Write the name of each independent remote LU with which you will communicate in a separate `Remote LU Name` field on the worksheet. For AS/400 communication, the `Remote LU Name` must match the `LCLLOCNAME` in the Device Description on the AS/400.

   The number of independent remote LUs you write down is the minimum number of independent LU session types you will need. (You might want to configure additional session types for reserved sessions.)

2. Write a name for each independent LU session type in the `Session Type Name` field next to each `Remote LU Name`. A `Session Type Name` must be 8 or fewer characters long and begin with a letter. It does not have to match anything in the remote configuration. Every `Session Type Name` must be unique.

3. In the `Remote Network ID` field under each `Remote LU Name`, write the network ID of the network where the remote LU is located. For AS/400 communication, the `Remote Network ID` must match the `Local network ID` in the Network Attributes Table on the AS/400.

   The `Remote Network ID` and the `Remote LU Name` make up the `Fully Qualified Remote LU Name` in the “APPC: Independent LU Session Type Data” screen in NMMGR.

4. In the `SNA Node Name` field under each `Session Type Name`, write the name of the configured local SNA node that connects the HP 3000 with the system where the remote LU is located. The `SNA Node Name`, and the names of all the LUs associated with it, must be already configured in the SNANODE branch of NMMGR.

5. In the Local Independent LU Name field under each SNA Node Name field, write the name of the local independent LU that will communicate with the remote independent LU. The Local Independent LU Name you write must be configured on the node you wrote in the SNA Node Name field.

   You can use the same independent LU for all the session types that use the same local SNA node (the same copy of SNA/SDLC Link/XL). Ordinarily, you need to configure only one independent LU per local SNA node; however, if you want to reserve sessions for a specific application or group of users, you can configure an independent LU for that purpose alone.

6. In the Mode Name field for each session type, write the mode name configured for the remote LU. The mode name configured on the HP 3000 must match the mode name configured for the remote LU. For AS/400 configuration, the Mode Name must match the MODE in the Device Description on the AS/400.

7. For each session type, determine the maximum number of active sessions you will need and write that number in the Maximum Number of Sessions field. You will need as many active sessions as you will have conversations using the session type at once. The number of sessions you will need for each session type is

   ```
   [(instances of TP1 x (simultaneous conversations with TP1)]

   + [(instances of TP2 x (simultaneous conversations with TP2)]
                                .
                                .
                                .
   + [(instances of TPn)x (simultaneous conversations with TPn)]
   ```

   where TP1, TP2, . . . TP*n* are the transaction programs that will be using the session type at the same time.

8. For each session type, determine the number of sessions that will be activated when the APPC subsystem is started, and write the number in the Automatically Activated Sessions field. The Automatically Activated Sessions value must be less than or equal to the Maximum Number of Sessions value.

   When you have completed the Independent LU Session Types worksheet, add up all the Automatically Activated Sessions for all the session types. If the sum is greater than 256, adjust the Automatically Activated Sessions values until the sum is less than or equal to 256. The sum of all the Automatically Activated Sessions, for all independent and dependent LU session types, must be less than or equal to 256.

Completing the Dependent LU Session Types Worksheet

A dependent LU session type is used to communicate with dependent LUs on a host (Type 5) node. Follow these steps to complete the Dependent LU Session Types worksheet:

1. Write the name of each remote LU with which you will communicate in a separate `Remote LU Name` field on the worksheet. The `Remote LU Name` must match the `APPLID` in the `Label` field of the VTAM `APPL` definition statement.

   The number of remote LUs you write down is the minimum number of dependent LU session types you will need. (You might want to configure additional session types for reserved sessions.)

2. Write a name for each dependent LU session type in the `Session Type Name` field next to each `Remote LU Name`. A `Session Type Name` must be 8 or fewer characters long and begin with a letter. It does not have to match anything in the remote configuration. Every `Session Type Name` must be unique.

3. In the `SNA Node Name` field under each `Session Type Name`, write the name of the configured local SNA node that connects the HP 3000 with the system where the remote LU is located. The `SNA Node Name`, and the names of all the LUs associated with it, must be already configured in the SNANODE branch of NMMGR.

4. In the `Mode Name` field for each session type, write the mode name configured for the remote LU. The mode name configured on the HP 3000 must match the CICS `MODENAME` operand of the `DFHTCT TYPE=SYSTEM` macro.

5. For each session type, determine the maximum number of active sessions you will need and write that number in the `Maximum Number of Sessions` field. You will need as many active sessions as you will have conversations using the session type at once. The number of sessions you will need for each session type is

   ```
   [(instances of TP1) x (simultaneous conversations with TP1)]

   + [(instances of TP2) x (simultaneous conversations with TP2)]
                                .
                                .
                                .
   + [(instances of TPn) x (simultaneous conversations with TPn)]
   ```

   where `TP1`, `TP2`, . . . `TP`*n* are the transaction programs that will be using the session type at the same time.

6. Under `Local Dependent LU Names`, list the local dependent LUs that will communicate with the remote LU. The `Local Dependent LU Names` you list must be configured on the node you wrote in the

SNA Node Name field. The number of dependent LUs must match the number in the Maximum Number of Sessions field.

You can list the same LUs for more than one session type, but you must make sure you have enough LUs to accommodate all the sessions, of all types, that will be active at once. A dependent LU can carry on only one session at a time.

7. For each session type, determine the number of sessions that will be activated when the APPC subsystem is started, and write the number in the Automatically Activated Sessions field. The Automatically Activated Sessions value must be less than or equal to the Maximum Number of Sessions value.

   When you have completed the Dependent LU Session Types worksheet, add up all the Automatically Activated Sessions for all the dependent and independent LU session types. If the sum is greater than 256, adjust the Automatically Activated Sessions values until the sum is less than or equal to 256. The sum of all the Automatically Activated Sessions, for all independent and dependent LU session types, must be less than or equal to 256.

Completing the Mode Types Worksheet

Follow these steps to complete the Mode Types worksheet:

1. Write each different Mode Name from your Independent LU Session Types worksheet and Dependent LU Session Types worksheet in a separate Mode Name field on the Mode Types worksheet.
2. Under each Mode Name, write the Maximum Size of RU to Send and the Maximum Size of RU to Receive. These values must match the values configured for the remote LU.

   For IBM mainframe communication, these values must match the values configured in the RUSIZES parameter of the MODEENT Logmode Table macro. For AS/400 communication, these values must match the MAXLENRU value in the Mode Description on the AS/400.

3. For each mode type, write the Maximum Size of Send Pacing Window and the Maximum Size of Receive Pacing Window. These values must match the values configured for the remote LU.

   For IBM mainframe communication, these values must match the *SRCVPAC* and *SSNDPAC* parameters of the MODEENT Logmode Table macro. For AS/400 configuration, these values must match the INPACING and OUTPACING values configured in the Mode Description on the AS/400.

Example Worksheets

The completed worksheets in this section are based on the configuration illustrated in Figure C-1. Appendix B , “Sample Configuration,” shows the NMMGR screens used for the configuration in this figure.

Figure C-1 Example Configuration

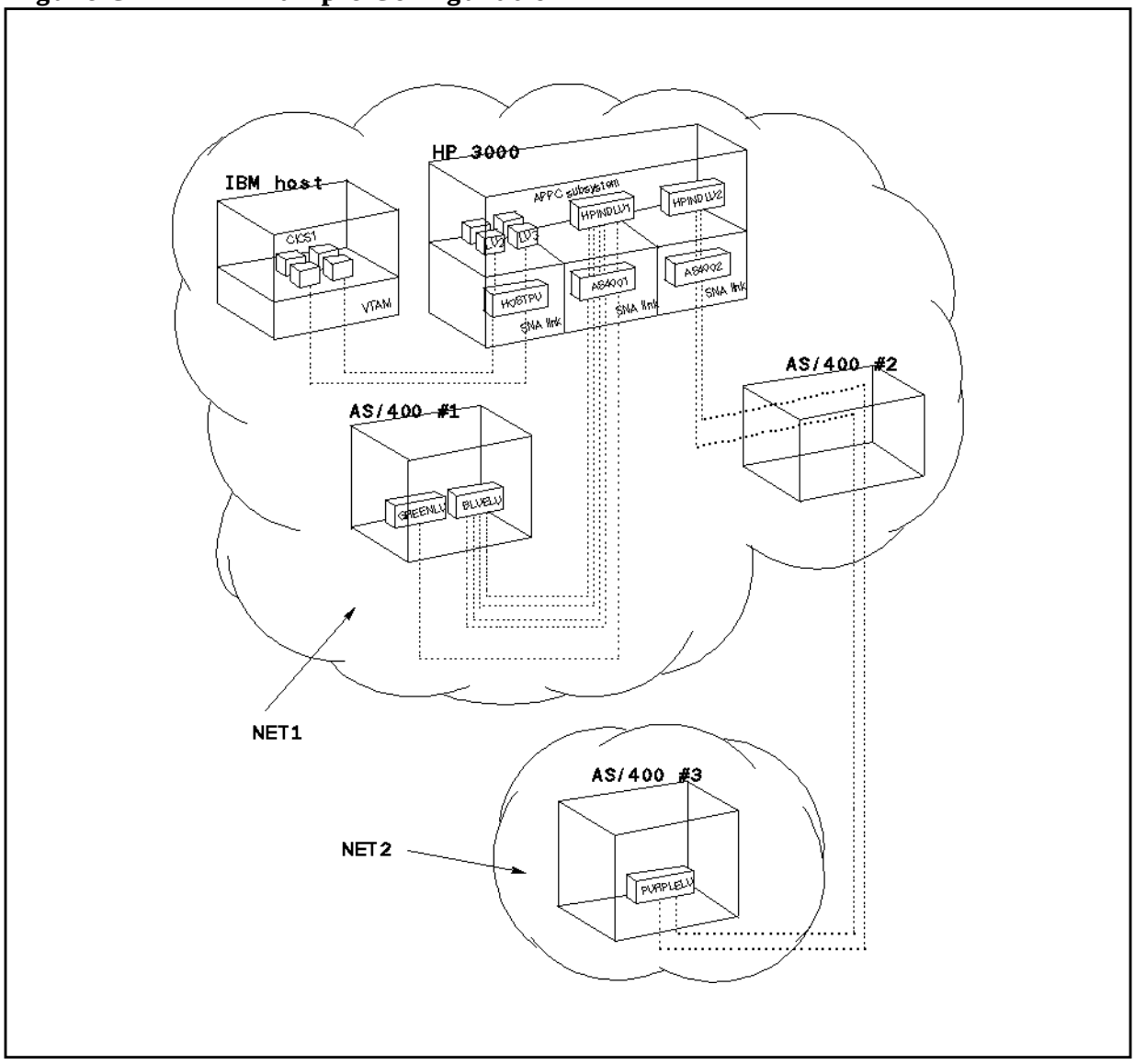


Figure C-2 is a Independent LU Session Types worksheet for the example configuration in Figure C-1.

Figure C-2 Independent LU Session Types Worksheet

Independent LU Session Types

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Session Type Name ________	Remote LU Name ________	Maximum Number of Sessions ____
SNA Node Name ________	Remote Network ID ________	Automatically Activated Sessions ____
Local Independent LU Name ________	Mode Name ________	

Figure C-3 is a Dependent LU Session Types worksheet for the example configuration in Figure C-1.

Figure C-3 Dependent LU Session Types Worksheet

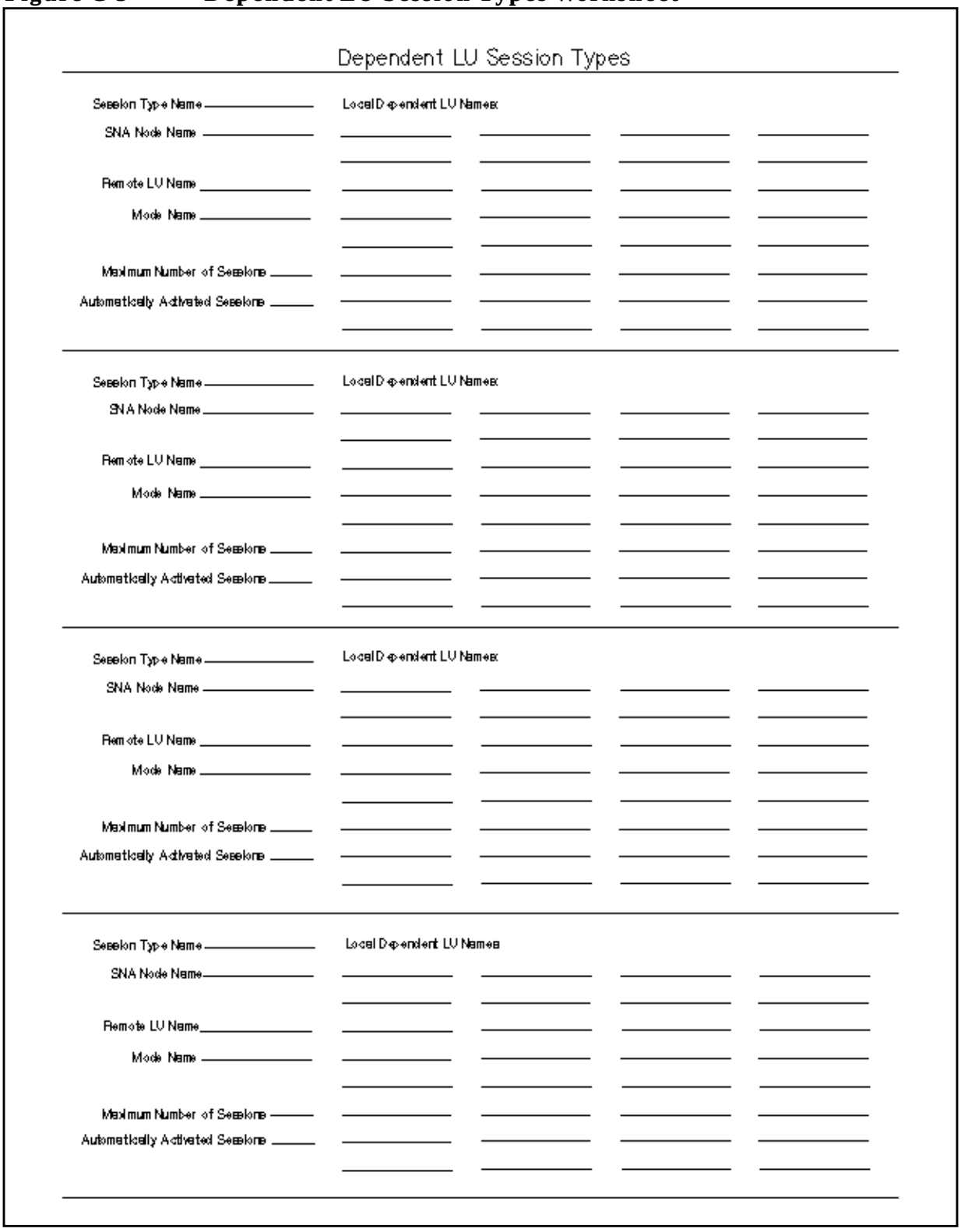
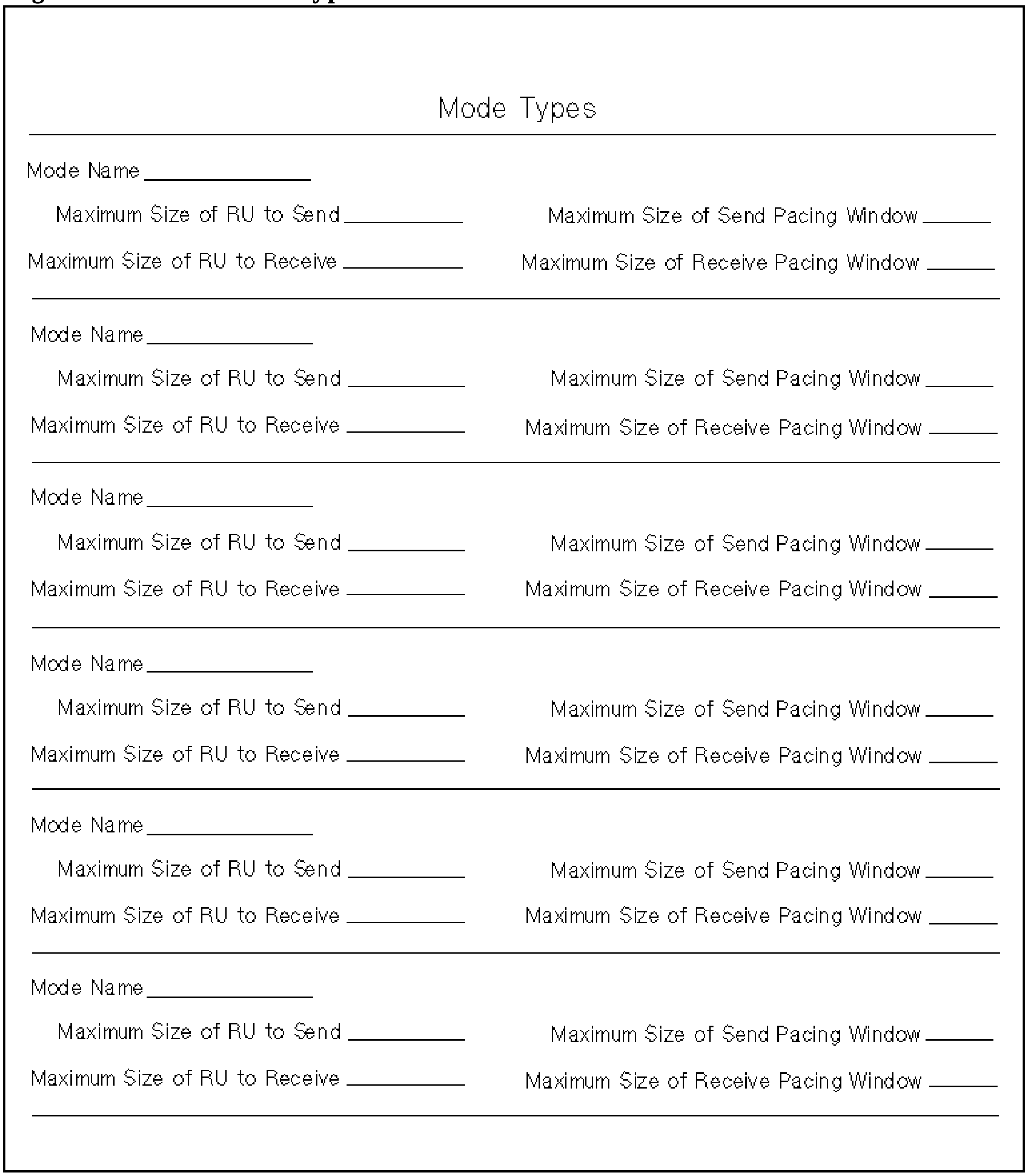


Figure C-4 is a Mode Types worksheet for the example configuration in Figure C-1.

Figure C-4 Mode Types Worksheet



Mode Types

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

Mode Name ____________
Maximum Size of RU to Send ________ Maximum Size of Send Pacing Window ______
Maximum Size of RU to Receive ________ Maximum Size of Receive Pacing Window ______

D LU 6.2 API/XL Installation Guidelines

To run LU 6.2 API/XL on your HP 3000, you must install the product and configure both your HP 3000 environment and the remote system to support LU 6.2 API/XL. HP 3000 configuration is described in Chapter 4 , “APPC Subsystem Configuration,” of this manual. Remote system configuration is described in the *IBM Host System Programmer References*, listed in the preface of this manual.

This appendix describes the tasks you need to perform to install LU 6.2 API/XL. It also lists the hardware and software required on the HP 3000 and the remote system in order to run LU 6.2 API/XL.

Pre-Installation Tasks

Before you install LU 6.2 API/XL on the HP 3000 you must do the following:

1. Ensure that the remote system has all the required hardware and software listed under “Remote System Requirements,” later in this appendix.
2. Ensure that the modem link operates between the HP 3000 and the remote system. If you use HP modems, Hewlett-Packard will help you verify that these modems work.
3. Configure a remote PU and terminal to the remote system using Hewlett-Packard's recommended parameters. You may have to make system generation changes to ACF/NCP, ACF/VTAM, and application subsystems running under ACF/VTAM. To connect to an AS/400, you may have to modify the AS/400 configuration. For more information on host configuration, consult the following manuals:

   HP SNA Products: Manager's Guide
   HP SNA Products: ACF/VTAM and ACF/NCP Guide
   HP SNA Products: CICS Guide
   HP SNA Products: AS/400 Guide

   Obtain a copy of the remote system configuration. You will need this information later during LU 6.2 API configuration.
4. Ensure that SNA/SDLC Link/XL or SNA/X.25 Link/XL has been installed correctly.
5. Ensure that your HP 3000 is at the proper release level to run LU 6.2 API/XL.

Installation Tasks

After you perform the pre-installation tasks, your HP 3000 will be prepared for LU 6.2 API/XL installation.

1. Log on as MANAGER.SYS. Ask all users to log off.
2. Check whether you have enough disk space to install LU 6.2 API/XL. To do this, issue the following command to build a dummy file:

   ```
   :BUILD INSTFILE;DISC=300000,32,32
   ```

   If the command fails, you do not have enough disk space, and you need to purge some files or store them on tape.

   If the command is successful, issue the following command to purge the dummy file:

   ```
   :PURGE INSTFILE
   ```

3. Write down the passwords for the SYS account, the PUB.SYS group, and the MANAGER.SYS user. Then, issue the following commands to remove the passwords:

   ```
   :ALTACCT SYS;PASS=
   :ALTGROUP PUB.SYS;PASS=
   :ALTUSER MANAGER.SYS;PASS=
   ```

4. Write down the names of any UDC files for MANAGER.SYS,PUB. Then, issue the following commands to disable the UDCs:

   ```
   :SETCATALOG;SYSTEM
   :SETCATALOG;ACCOUNT
   :SETCATALOG
   ```

5. Follow the instructions in the *HP 3000 MPE XL Installation and Update Manual* or the *HP 3000 MPE XL Add-On Manual* to install LU 6.2 API/XL with the MPE AUTOINST utility.
6. After the installation job completes, shut down the system and install the update type. Then, restart the system, and log on again as MANAGER.SYS.
7. Issue the following commands to restore the passwords to the SYS account, the PUB.SYS group, and the MANAGER.SYS user:

   ```
   :ALTACCT SYS;PASS=password
   :ALTGROUP PUB.SYS;PASS=password
   :ALTUSER MANAGER.SYS;PASS=password
   ```

8. Issue the following commands to reset all UDCs:

   ```
   :SETCATALOG udcfile[,udcfile ...];SYSTEM
   :SETCATALOG udcfile[,udcfile ...];ACCOUNT
   :SETCATALOG udcfile[,udcfile ...]
   ```

9. Issue the following command to install the UDCs that invoke APPCCONTROL commands:

   ```
   SETCATALOG APPCUDC.APPC.SYS;SYSTEM;APPEND
   ```

10. Make sure all users who will issue APPCCONTROL commands have at least the following capabilities:

    Account Manager (AM)
    Node Manager (NM)
    Save Files (SF)
    Network Administrator (NA)
    Group Librarian (GL)
    Access to nonsharable I/O devices (ND)
    Batch Access (BA)
    Interactive Access (IA)

    To change the capabilities of a user, issue the following command:

    ```
    :ALTUSER {user};CAP=AM,NM,SF,NA,GL,ND,BA,IA
    ```

11. After you have installed LU 6.2 API/XL, run NMMAINT.PUB.SYS to ensure that the AUTOINST utility successfully installed the right versions of all the necessary software modules. The overall version of LU 6.2 API and the APPC subsystem must be greater than or equal to B.00.00. The overall version of the Node Management Services (NMS) utilities must be B.03.00 or greater.

Your HP representative will perform a minimum configuration of LU 6.2 API/XL to verify that the product is working correctly.

Remote System Requirements

This section describes the remote hardware and software required to communicate with Hewlett-Packard's LU 6.2 API.

Hardware

The following hardware is required to communicate with an HP 3000 that is acting as a Type 2.0 node:

- An IBM host, which can be any IBM plug-compatible computer. The IBM mainframes include the 433x, 434x, 438x, 303x, 308x or 309x.
- An IBM 3705, 3710, 3720 ICA, 3725, or 3745 compatible Communications Controller that supports ACF/NCP.

The following hardware is required to communicate in a peer-to-peer environment with an HP 3000 that is acting as a Type 2.1 node:

- An IBM AS/400 computer.

Software

The following software is required to communicate with an HP 3000 that is acting as a Type 2.0 node:

- Multiple Virtual Storage (MVS) or Disk Operating System/Virtual Storage Extended (DOS/VSE) operating system.
- Advanced Communication Function for the Virtual Telecommunications Access Method (ACF/VTAM) version 2.1 or later.
- Advanced Communication Function for the Network Control Program (ACF/NCP) version 3.1 or later.
- The Customer Information Control System (CICS) application subsystem version 1.7 or later.

The following software is required to communicate in a peer-to-peer environment with an HP 3000 that is acting as a Type 2.1 node:<

- OS/400, the operating system on the IBM AS/400, version 1 release 2 or later.

HP 3000 Requirements

This section describes the HP 3000 operating environment that supports LU 6.2 API/XL.

Hardware

The following hardware is required to run LU 6.2 API/XL:

- A 900 Series HP 3000.
- An HP 30263A Programmable Serial Interface (PSI) card and the appropriate PSI cables. The PSI is bundled with SNA/SDLC Link/XL or SNA/X.25 Link/XL. One PSI is required for each link to a remote system.
- A full or half duplex data communication line (switched or leased) between the HP 3000 and the remote system.
- A pair of modems for the communication line. Only full-duplex, synchronous modems are supported.
- A block-mode terminal which is supported by VPLUS/3000 for Node Management Services configuration.

Software

The following software is required to run LU 6.2 API/XL on the HP 3000:

- LU 6.2 API/XL, version B.00.00 or later.
- The HP 3000 Multi-Programming Executive (MPE XL) operating system (release 4.0 or later).
- SNA/SDLC Link/XL or SNA/X.25 Link/XL bundled hardware and software product, version B.00.10 or later.
- Node Management Services (NMS) version B.03.00 or later.

Glossary

A

Advanced Program-to-Program Communication (APPC):

Programmatic communication based on IBM's LU 6.2 architecture. APPC provides partner programs with a common set of rules for communication.

Application Program Interface (API): A set of subprograms, callable from inside applications, that carry out data communications tasks.

B

basic conversation: A programmatic conversation in which the applications must be able to create and interpret GDS headers (see GDS header) for transmitting and receiving data.

basic conversation verbs: The programmatic implementation of functions and protocols in a basic conversation between transaction programs. (See mapped conversation verbs.)

bidder session: A remotely controlled session. See remotely controlled session.

BIND: An SNA command sent by the host to a remote LU. This command specifies the detailed protocol that the Network Service accepts before initiating an LU session.

C

cluster controller: A machine that allows multiple devices to send and receive data over the same communications link.

communications controller: A front-end processor that provides an interface between the communications facilities and a computer. IBM provides programmable and non-programmable communications controllers, such as the IBM 3705, 3725, and 3745 Communication Controllers. The Intelligent Network Processor (INP) on MPE V and the Programmatic Serial Interface (PSI) on MPE XL are HP communications controllers.

contention loser session: A remotely controlled session. See remotely controlled session.

contention winner session: A locally controlled session. See locally controlled session.

control operator commands:

Interactive commands, issued by the node manager, used to activate and deactivate APPC sessions and to monitor the activity of the APPC subsystem.

control operator intrinsics:

Programmatic subroutines, callable from inside applications, used to control and monitor the activity of the APPC subsystem. Control operator intrinsics

implement the same functions as control operator commands. (See control operator commands.)

conversation: The logical communication between two transaction programs.

Customer Information Control System (CICS): An IBM application subsystem that provides file handling and data communications services for application programs.

D

dependent LU: An LU capable of conducting only one APPC session at a time. A dependent LU always functions as a secondary LU; that is, it cannot send a BIND to initiate a session but must wait for the host to send the BIND. It can send an INIT_SELF to the host to solicit a BIND, or it can be configured to accept unsolicited BINDS. See independent LU.

domain: The part of a network managed by a Systems Services Control Point. This includes the Physical Units (PUs), Logical Units (LUs), and links that the SSCP activates, deactivates, and otherwise manages.

E

end user: The ultimate destination of data in a communications network. An end user can be a human user, a peripheral device (like a printer or a terminal), or an application program.

F

first speaker session: A locally controlled session. See locally controlled session.

FM header: See Function Management header.

Function Management header (FM header): A header that selects the destination or source for subsequent data transmission and supplies information for other data management tasks.

G

Generalized Data Stream (GDS): The name of the LU 6.2 data stream. LU 6.2 data packets must include GDS headers (see GDS header).

GDS header: A portion of an LU 6.2 data packet that contains information about the kind of data being sent or received.

H

half-session: An SNA session must be supported by two sides. A half session is the resource on a Network Addressable Unit that provides FMD services, data flow control, and transmission control for its half of a complete session.

host: A central computer that provides services for other computers and terminals attached to it.

I

independent LU: An LU capable of conducting multiple, simultaneous (parallel) APPC sessions with another independent LU on a remote system. An independent LU can function as either a primary or secondary LU. See dependent LU.

INIT-SELF: An SNA request to the SSCP to initiate an LU-LU session.

internal trace: A trace file, created by the APPC subsystem, that records subsystem internal events and state changes.

intrinsic: A subprogram provided by Hewlett-Packard to perform common functions such as opening files, opening communications lines, performing subsystem-defined functions, or transmitting data over a communications line.

L

local TP: The TP running on the local processor.

locally controlled session: An APPC session controlled by an LU on the local system. Also called a first speaker session. If a transaction program on the local system attempts to allocate a conversation, a locally controlled session will always be granted if one is available. If a transaction program on the remote system wants to allocate a conversation over a locally controlled session, it must ask permission of the local LU. The local LU may grant or deny the request. The APPC subsystem supports only locally controlled sessions.

logical device (ldev): A disk file, a set of disk files, or a physical device. An HP 3000 ldev is identified by a number or by a device class name.

Logical Unit (LU): The SNA entity through which application data is transmitted within an SNA network. Logical Units are the ports through which end users have access to the network (see end user). They format message units, display information, and handle error recovery. There can be multiple LUs in an SNA node.

LU type: A Logical Unit type, defined by SNA to perform a particular type of communication.

LU-LU session: The logical connection between two logical devices in an SNA network.

LU 6.2: An SNA LU type that defines the communication that can take place between two application programs on separate processors. LU 6.2 includes

specifications for programmatic interfaces, document interchange, and data distribution.

M

mapped conversation: A programmatic conversation in which the application is freed from handling the GDS headers required by the LU 6.2 architecture.

mapped conversation verbs:

The programmatic implementation of functions and protocols in a mapped conversation between transaction programs. (See basic conversation verbs.)

N

Network Addressable Unit (NAU): An entity in a network that can send or receive data. The three types of NAUs are System Service Control Points (SSCPs), Physical Units (PUs), and Logical Units (LUs).

Network Control Program (NCP): The program in the communications controller that controls data traffic for the host.

Node Type 2.0: The node type for a peripheral node or cluster controller. Node Type 2.0 is supported by the APPC subsystem on MPE V and MPE XL.

Node Type 2.1: The node type for a peripheral node or cluster controller capable of peer-to-peer communication. Node Type 2.1 is supported by the APPC subsystem on MPE XL.

Node Type 4: The node type for a subarea node with a communications controller.

Node Type 5: The node type for a host node with an SSCP (System Services Control Point).

P

pacing window: The number of RUs that can be sent before a response is required from the receiving RU.

peripheral: A device on the network, like a printer or disk drive, that functions as an end user for data.

Physical Unit (PU): The software and hardware of an SNA node that controls the resources of the node and reports errors and physical failures to the SSCP. There is one PU per node.

PU: See Physical Unit.

PU 2.0: See Node Type 2.0.

PU 2.1: See Node Type 2.1.

R

remote TP: The TP running on the remote processor.

remotely controlled session:

An APPC session controlled by an LU on the remote system. Also called a bidder session. The APPC subsystem does not support remotely controlled sessions. See locally controlled session.

RH: Request or response header. These are attached to SNA RUs to control data flow and support other network services.

RU: Request or response unit.

S

SDLC: See Synchronous Data Link Control.

SNA: See Systems Network Architecture.

SNA/SDLC Link/XL: A bundled HP hardware and software product that provides the logical and physical connection between the HP 3000 and an IBM host. SNA/SDLC Link/XL is used in combination with an SNA Service, such as SNA IMF, SNA NRJE, or LU 6.2 API.

SSCP: System Services Control Point. An SSCP manages nodes within its domain and stores status information for personnel to maintain a network. It exists only in Type 5 nodes, implemented by the communications access method.

Synchronous Data Link Control: The data link level communications protocol used in SNA.

Systems Network Architecture (SNA): A comprehensive specification for distributed data processing developed by IBM. SNA defines a layered protocol for communicating and controlling a communications network within the IBM environment.

T

transaction program (TP): An application program that processes distributed transactions.

transmission header (TH): The portion of an SNA data packet that contains routing and sequencing information.

V

verbs: The programmatic implementation of functions and protocols in a conversation between transaction programs.

Virtual Telecommunications Access Method (VTAM): The SNA program on the host that controls the resources of a domain in an SNA network.

Index

Q

R

S

W