IRIS SNA SERVER Programming Guide

Document Number 007-0871-030

CONTRIBUTORS

Engineering contributions by Jay Lan.
Cover design and illustration by Rob Aguilar, Rikk Carey, Dean Hodgkinson,
Erik Lindholm, and Kay Maitz

IRIS SNA SERVER Programming Guide
Document Number 007-0871-030

Contents

List of Tables

Introduction

This guide is designed for application programmers who write programs that utilize IRIS® SNA LU 0-3 from Silicon Graphics®. IRIS SNA LU 0-3, which is included and shipped with the IRIS SNA SERVER, provides an interface to the services available in Systems Network Architecture (SNA) Logical Unit (LU) Types 0, 1, 2, and 3.

This guide focuses only on those aspects of the SNA product that pertain to LU 0-3 and should be used in conjunction with the *IRIS SNA SERVER Administration Guide,* which covers such topics as the graphical user interface network management tool (IRIS SNAView) and configuring and operating the IRIS SNA SERVER.

Using This Guide

The *IRIS SNA SERVER Programming Guide* contains the following chapters and appendices:

Chapter 1 “Before You Begin” describes how to set the run-time variables and provides an overview of the verbs used to implement IRIS SNA LU 0-3. Detailed verb documentation is also provided in Appendix H, “Man Pages,” of this guide.

Chapter 2 “Programming with the SNA SERVER Verbs” describes the Application Programmers Interface (API), or verb functions, which are provided separately as manual pages in Appendix H. A diagnostic guide for applications is also included.

Chapter 3 “Programming with the LU 0-3 Verbs” presents an overview to the Programmable Interface (PI), or verb functions for the Logical Unit Types 0, 1, 2, and 3.

Appendix A	“Major and Minor Return Codes” lists the major and minor error codes returned by the API verbs (that is, the SNA SERVER, LU 6.2, and LU 0-3 verbs).
Appendix B	“Supported SNA Request Units” lists RUs that can be sent or received by a transaction program using IRIS SNA LU 0-3.
Appendix C	“Notification Events” presents the valid notifications reported in the *ru_cmd* parameter of the **rcvru** verb and the layout of the **rcvru** verb for each valid event.
Appendix D	“LU Finite-State Machines” describes the members contained in the LU finite-state machine structure, *fsm_lu*.
Appendix E	“PU Finite-State Machines” describes the members contained in the PU finite-state machine structure, *fsm_pu*.
Appendix F	“State Definition” describes valid states for LU and PU services and provides a recommended action for each state.
Appendix G	“API Verb Catalog” provides an index to all API verbs.
Appendix H	“Man Pages” contains the category (3) IRIS SNA SERVER and LU 0-3-related man pages.

Conventions

Within text, file names, parameters, commands, and command arguments are shown in *italics.* Machine names are shown in ***bold italics.***

Command syntax descriptions and examples appear in `typewriter font`.

User input and keyboard commands appear in **bold typewriter font**.

API verb names are shown in **bold** face.

Related Documentation

This guide is part of a three-volume set that includes:

- IRIS SNA SERVER Administration Guide

- IRIS SNA SERVER VT100 Interface Guide

In addition, the following reference materials from Silicon Graphics and IBM® provide supplementary information on topics covered in this guide.

Token Ring Administration Guide

IRIS SNA LU 6.2 Programming Guide

IRIS SNA 3270 Administrator's Guide

International Business Machines (IBM order numbers follow title)

Systems Network Architecture Concepts and Products (GC30-3072)

Systems Network Architecture Technical Overview (GC30-3073)

An Introduction to Advanced Program-to-Program Communication (GG24-1584)

Systems Network Architecture Transaction Programmer's Reference Manual for LU Type 6.2 (GC30-3084)

Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2 (SC30-3269)

Systems Network Architecture Network Product Formats (LY43-0081)

Synchronous Data Link Control Concepts (GA27-3093)

Systems Network Architecture Reference Summary (GA27-3136)

Token Ring Network Architecture Reference (SC30-3374)

Product Support

Silicon Graphics provides a comprehensive product support and maintenance program for IRIS products. For further information, contact the Technical Assistance Center.

Chapter 1

Before You Begin

This chapter contains preliminary information for the application programmer who writes transaction programs that use the IRIS SNA LU 0-3 Programmable Interface (PI) verbs.

After describing how to set the run-time variables, the chapter identifies the IRIS SNA SERVER verbs used with the LU 0-3 Programmable Interface verbs. When used together, the IRIS SNA SERVER verbs and the LU 0-3 PI verbs make up the Application Program Interface, or API. Information about these verbs is presented in Appendix H, "Man Pages," of this guide. The programmer needs access to the verb man pages in order to write transaction programs. Appendix G, "API Verb Catalog,"contains a catalog of all verbs and the location of their man pages in the IRIS SNA SERVER manual set.

This chapter introduces the PI verbs, outlines the programmer's responsibilities and describes verb execution and completion, the *naustat* variable, session management, PU and LU structures, verb-boundary integration, and data types.

Setting Run-Time Variables

If your programs are to be executed on a system that does not have the IRIS SNA SERVER installed, you must set the run-time variable *SNAHOST* to the system ID of the IRIS SNA SERVER system before your start your SNA API programs.

IRIS SNA SERVER Verbs

This section describes functions provided by the IRIS SNA SERVER

that are of interest to the programmer using LU 0-3 Programmable Interface (PI) verbs or programmers using the LU 6.2 API verbs. Outlined below are three categories of IRIS SNA SERVER verbs.

- Implementation-specific verbs
- Configuration verbs
- Node Operator verbs

Implementation-specific Verbs

Implementation-specific verbs, while not a part of the IBM SNA, are required for the IRIS SNA SERVER. These verbs perform specific functions, such as opening and closing the connection between the transaction program and the IRIS SNA Scheduler.

Transaction Program Connection Verbs

Before a program can issue any conversation, control operator, or node operator function, it must establish a connection with the IRIS SNA Scheduler. Establishing this connection is called “attaching.” For the conversation verbs and the control operator verbs, this is done by specifying the name of the configuration and the local LU that the transaction program wants to use. The names specified in the attach request are called the “context” of the attach. Since a program can issue verbs to more than one configuration or LU, the program can issue multiple attach requests. Each of these “attaches” creates a new logical instance of the program. The **setctx** verb is used to switch from one instance to the other before issuing verbs to the different LUs.

Transaction Program Connection verbs establish or break the connection between the transaction program and the IRIS SNA Scheduler. Consequently, they are the first and last verbs issued by the program. All applications must use the **attach** and **detach** verbs. The **setctx** verb is optional.

The man pages for these Transaction Program Connection verbs are included in Appendix H. Table 1-1 summarizes these verbs.

Table 1-1 Transaction Program Connection Verbs

Verb	Full Name	Function
attach	Attach	Initiates communication between the local program and the IRIS SNA Scheduler.
detach	Detach	Detaches the current context from the IRIS SNA Scheduler.
setctx	Set Context	Establishes the current context under which subsequent verbs are issued.

Configuration Verbs

IRIS SNA SERVER Configuration verbs, used to define and display the resources of the node and logical unit, are divided into two groups: define verbs and display verbs. The Define and Display verbs listed in the Table 1-2 are of interest to the programmer using the LU 0-3 Programmable Interface (PI) verbs.

Table 1-2 Configuration Verbs

Verb	Full Name
dfncp	Define Control Point
dfnline	Define Line
dfnllu	Define Local LU
dfnnode	Define Node
dfnrlu	Define Remote LU
dfnslu	Define Secondary Logical Unit
dfnsta	Define Station
dfntp	Define Transaction Program
dspcp	Display Control Point
dspcph	Display Host Status
dspline	Display Line
dspllu	Display Local LU
dspnode	Display Node
dsprlu	Display Remote LU
dspslu	Display Secondary Logical Unit
dspsta	Display Station
dsptp	Display Transaction Program

Node Operator Verbs

The Node Operator verbs in Table 1-3 activate, deactivate, and supervise the configured local resources of the IRIS SNA SERVER. These verbs control the links.

Table 1-3 Node Operator Verbs

Verb	Full Name
actline	Activate Line
actlu	Activate Logical Unit
actpu	Activate Physical Unit
actsta	Activate Station
chgmsgq	Change Message Queue
dctline	Deactivate Line
dctlu	Deactivate Logical Unit
dctpu	Deactivate Physical Unit
dctsta	Deactivate Station
dspmsgq	Display Message Queue
rtvnmsg	Retrieve Node Message

PI Verb Catalog

The PI verbs provide a function-call interface to the services available in a Logical Unit Type 0, 1, 2, or 3, or the Configuration Management Services component of the Physical Unit (PU). This interface is not used for Logical Unit Type 6.2.

Table 1-4 identifies the LU 0-3 PI verbs, provides the full name of the verb, and briefly describes the verbs function. See the man pages in Appendix H for detailed information on the verbs.

Table 1-4 LU 0-3 PI Verbs

Verb	Full Name	Function
initpi	Initialize Verb Data Structures	Initializes the data structures used by the other PI verbs.
rcvru	Receive Request Unit	Receives data from the IRIS SNA SERVER Scheduler. Data can be SNA RUs from the session or notifications from the scheduler.
rejru	Reject Request Unit	Rejects an RU chain or part of an RU chain received by the program.
sndru	Send Request Unit	Sends an RU chain according to the specifications supplied by the program.
ssync	Set Sync Point	Sets the sync-point sequence numbers in the

Table 1-4 (continued) LU 0-3 PI Verbs

Verb	Full Name	Function
		IRIS SNA Scheduler.
LU		

These verb functions must be loaded with the LU 0-3 verb library *liblu03.a.*

Chapter 2

Programming with the SNA SERVER Verbs

This chapter contains information to help you write transaction programs using the verb library functions and to help diagnose application errors should they arise. The material in this chapter provides an introduction and overview of the verbs. Appendix A lists the major and minor codes returned by the verb functions. Appendix C describes messages displayed during normal system activity or when errors are detected. Appendix G, "API Verb Catalog," lists all of the API verbs in alphabetical order, giving their full names and verb types.

Some of the verbs presented in this chapter are relevant to users of IRIS SNA LU 6.2. See the *IRIS SNA LU 6.2 Programming Guide* for information about these verbs.

API Overview

The Application Program Interface (API) is a library of verb functions and header files that define the structures and variables needed by applications that call the verb functions. In the Silicon Graphics implementation, these verbs take the form of C-language function calls. The API supplied by the IRIS SNA SERVER also contains the verbs needed by other related Silicon Graphics SNA products such as IRIS SNA LU 6.2, IRIS SNA 3770, and IRIS SNA 3270. The API library for the IRIS SNA SERVER is installed as */usr/lib/libsna.a* and */usr/lib/liblu03.a.*

Verb Categories

The IRIS SNA SERVER verbs are divided into three categories according to their function:

- Implementation-specific verbs
- Configuration verbs

- Node Operator verbs

Implementation-specific Verbs

Implementation-specific verbs are not specified in the IBM *SNA Transaction Programmer's Reference Manual for LU Type 6.2* (referred to subsequently in this guide as *TPRM*) but are defined and implemented in Silicon Graphics SNA products. These verbs are divided into two groups:

- Transaction program connection verbs
- Transaction program utility functions

The connection verbs establish or break the connection between a transaction program and the IRIS SNA Scheduler. The utility functions, while not true verbs, provide routines that APPC transaction programs commonly need, such as ASCII-to-EBCDIC translation.

Configuration Verbs

Configuration verbs define and display the resources of the node and logical unit. They are divided into two groups:

- Define verbs
- Display verbs

The first group contains both LU define and node define verbs. The LU define verbs configure information about LUs, modes, and transaction programs. The node define verbs configure PU 2.1 components. The second group, the display verbs, provide information about the configured LUs and PU 2.1 components.

Node Operator Verbs

Node operator verbs activate and deactivate PU 2.1 components. These verbs are not specified in IBM's *TPRM*, but are Silicon Graphics-defined.

Verb Library

Verb functions, and functions called by the verb, are archived in */usr/lib/libsna.a.* Programs that use verb functions must be linked against this library. Under most compilers, this is accomplished by using the *-l* option; for example:

```
cc sample.c -L/usr/lib/sna -lsna.a
```

Note the case-specific use of the letter *l* in the example. See your compiler documentation for instructions on linking against product libraries.

Header Files

The structures and variables used in the SNA verbs are defined in header files located in */usr/include/sna*. Here are some of the header files you can include in your application:

global.h Contains the definitions of the global variables used by all of the verb types.

imp.h Contains the definitions of the verb parameter structures for the implementation-specific verbs.

ludef.h Contains the definitions of the verb parameter structures for the LU define verbs.

msgdef.h Contains the definitions of the structures used to retrieve and parse node messages.

noop.h Contains the definitions of the verb parameter structures for the node operator verbs.

uadef.h Contains the definitions of the verb parameter structures for the node define verbs.

In addition to these header files, */usr/include/sna* contains several files that contain the type definitions and constant values used by the SNA verbs. They do not need to be explicitly included; examine them for their contents.

ddhvtyp.h Contains the type definitions for the SNA verbs.

ddhverr.h Contains constants for the major and minor return codes.

ddhvicn.h Contains the constants used for the verb parameter values.

ddhviex.h Contains external variable declarations.

Data Type Definitions

The verb parameter structures use data types defined in the header file *ddhvtyp.h*. Several of these data types may not be familiar to you:

- typedef unsigned char hex

  Used for strings or bytes where all bits are significant
- typedef unsigned short shex

  Used for values that must be two bytes long
- typedef unsigned long lhex

  Used for values that must be four bytes long

Data Structures

Each member (referred to as both parameter and field) of the data structure is described as being Supplied, Returned, or Supplied/Returned.

- Supplied parameters are set by the application program.
- Returned parameters are set automatically by the successful operation of the verb.
- Supplied/Returned parameters are set by the application program when the verb is issued, but their value can change after the successful operation of the verb.

Initialization of every member of the structure is the responsibility of the application. This is especially true of pointers, which must be set to null if not set to a specific address.

Note: For the character-string parameters in the display and get attribute verbs, Returned and Supplied have a slightly different meaning. These verbs require the application program to allocate space for names and other strings that are to be returned. If the pointer is nulled, the name is not returned.

Even though the value in a name field is returned by the verb, the pointer must still be supplied by the application program. A display verb with a name parameter listed as Supplied/Returned can return a different name in that space after the successful completion of the verb.

Supplied parameters are further specified as Required, Conditional, or Optional.

- Required parameters must be set by the application program.
- Conditional parameters can have a value required, depending on the setting of another parameter.
- Optional parameters need not be set.

Global Variables

Information on the state of the conversation and feedback on the execution of verb calls are returned in three global variables. These variables are defined in the header file *global.h.* This file must be included in each program that uses functions described in the IRIS SNA SERVER API. Routines that query the values of these variables can refer to them as external variables.

After a verb function is executed, return information is placed in global variables *snamaj* and *snamin*. The values carried here correspond to the verb return-code parameters defined in the *TPRM*. If the function completed normally (return code of "OK"), the function returns 0 and the major and minor codes are also 0. If a return code was set in the major or minor fields, the verb function returns -1. In general, the major code is sufficient for controlling program logic.

Table 2-1 describes the major codes used. (For a complete list of the major and minor return codes, see Appendix A.)

Table 2-1 Major Return Codes

Major Code	Description
S2_OK (0)	Function completed successfully.
S2_USAGE (1)	Function aborted, usage error.

Table 2-1 (continued) Major Return Codes

Major Code	Description
	The function was not performed because a parameter was in error or requested a function that was not supported. Most of the minor codes provide specific information on usage errors.
S2_UNSUC (2)	Function completed unsuccessfully. Set when a function, such as "Receive Immediate" or "Test," completes normally but does not return data.
S2_STATE (3)	Function aborted, state error. The function was not performed because it is not allowed in the current conversation state.
S2_ALCER (5)	Allocation error. The program could not allocate a conversation for the reason specified in the minor code. The conversation is in deallocated state when an allocation error occurs.
S2_PGMER (7)	Program error. The partner program has issued an error indication. If the conversation was in send state, it is now in receive state.
S2_DEALC (9)	Deallocation indication. The conversation has been deallocated, normally or abnormally, for the reason specified in the minor code. The conversation is in deallocated state.
S2_COERR (10)	Control operator function error.

Table 2-1 (continued) Major Return Codes

Major Code	Description
	A control operator function has ended abnormally. Since the control operator verb may not have been using a conversation, the conversation state does not apply.
S2_NPERR (11)	Node-operator function error. A node-operator function was not accepted. Since node operator verbs do not use conversations, the conversation state does not apply.

Implementation-specific Verbs

Although the implementation-specific verbs are not part of the IBM SNA, they are required for the IRIS SNA SERVER. These verbs perform specific functions, such as opening and closing the connection between the transaction program and the IRIS SNA Scheduler.

The implementation-specific verbs are divided into two groups: transaction program connection verbs and transaction program utility functions. Each of these groups is described on the following pages. The verb structures and variables are in */usr/include/sna/imp.h.*

Transaction Program Connection Verbs

Before a program can issue any conversation, control-operator, or node-operator functions, it establishes a connection with the IRIS SNA Scheduler. Establishing this connection is called *attaching*. Conversation verbs and control operator verbs attach by specifying the name of the configuration and the local LU that the transaction program wants to use. Node operator verbs require only the configuration. The names specified in the attach request are called the *context* of the attach. Since a program can issue verbs to more than one configuration or LU, the program can issue multiple attach requests. Each of these attaches creates a new logical instance of the program. The **setctx** verb switches from one instance to the other before issuing verbs to the different LUs.

Because transaction program connection verbs establish or break the connection between the transaction program and the IRIS SNA Scheduler, they are the first and last verbs the program issues.

Table 2-2 shows the supported transaction program connection verbs.

Table 2-2 Transaction Program Connection Verbs

Verb	Function
attach	Initiates communication between the local program and the IRIS SNA Scheduler.
detach	Detaches the current context from the IRIS SNA Scheduler.
rattach	Initiates communication between a remotely invoked transaction program and the IRIS SNA Scheduler.
setctx	Sets the current context under which subsequent verbs are issued.

Transaction Program Utility Functions

The transaction program utility functions shown in Table 2-3 are not considered verbs because they do not set *snastat, snamaj*, or *snamin*. Rather, they provide useful routines for writing APPC applications.

Table 2-3 Transaction Program Utility Functions

Utility	Function
atoe	Translates an ASCII string into EBCDIC for partner programs running on IBM computers so they may receive application-specific information.
etoa	Translates an EBCDIC string into ASCII so partner programs running on IBM machines can send application-specific information.
dspmaj	Formats major return code messages.
dspmin	Formats minor return code messages.
prtnmsg	Prints a formatted message from the message structure returned by the rtvnmsg verb.

Configuration Verbs

Configuration verbs are divided into two categories: define verbs and display verbs. The define verbs include LU define verbs, which configure local and remote LUs, modes, and transaction programs; and node define verbs, which define the components of the PU 2.1 node. (Display verbs display information about LUs, modes, transaction programs, and PU 2.1 node components.)

The verb functions, and functions called by the verb, are archived in */usr/lib/libsna.a.* Programs that use verb functions are linked against this library. LU define verb data structures are in */usr/include/sna/ludef.h.* Data structures for the node define verbs are in */usr/include/sna/noop.h.* The global variables are defined in header file */usr/include/sna/global.h.*

Each verb has an associated data structure that contains the verb's parameters. These data structures are more complex than those for the conversation verbs. To simplify their use, two special verbs are provided—**initcbu** for the node and **initcbl** for the LU—to initialize the values in the structure members. Each pointer within the structure is nulled. All other fields are set to the constant value, indicating that the parameter is not being specified.

As stated earlier, each member of the data structure is described as being Supplied, Returned, or Supplied/Returned. For more information on these parameter qualifiers, see Section , "Data Structures." For the character-string parameters in the display verbs, Returned and Supplied have a slightly different meaning. These verbs require the application program to allocate space for names and other strings that are to be returned. If the pointer is nulled, the name is not returned. Even though the value in a name field is returned by the verb, the pointer must still be supplied by the application program. A name parameter listed as Supplied/Returned in a display verb means a different name can be returned in the same space after the successful completion of the verb.

Define Verbs

Define verbs define the configuration elements of the LU (the local LU, remote LU, modes, and transaction programs) and the PU 2.1 node (the lines, stations, control points, and the node itself). Table 2-4 lists the define verbs that are provided.

Table 2-4 Define Verbs

Verb	Function
dfncp	Defines the addressing information for the control point at the remote node.
dfnline	Defines line name, line type, and line characteristics.
dfnllu	Initializes or modifies parameter values that control the operation of the local LU.

Table 2-4 (continued) Define Verbs

Verb	Function
dfnmode	Initializes or modifies parameter values that control the operation of the local LU in conjunction with a group of sessions (identified by a mode name) with a remote LU.
dfnnode	Defines the system-wide operating parameters for an instance of the SNA node.
dfnrlu	Initializes or modifies parameter values that control the operation of the local LU in conjunction with a remote LU.
dfnslu	Defines a secondary LU.
dfnsta	Defines the station name and other parameters that pertain to the adjacent link station.
dfntp	Initializes or modifies parameter values that control the operation of the local LU in conjunction with a local transaction program.
dltcbl	Deletes the operating parameters for the local LU that were set by the define verbs.
dltcbu	Deletes control blocks defined by the node define verbs.
initcbl	Initializes a define verb structure.
initcbu	Initializes a define verb structure of the type specified in the ***stype*** parameter.

The node define verbs use the parameter constants listed in Table 2-5.

Table 2-5 Parameter Constants for Node Define Verbs

Constant	Description
IGN_VAR	Ignore this parameter. Specifies that an optional parameter is to be ignored. Use the current or default value for this parameter.
SUPPORT	Function is supported. Indicates the specific function is supported.
NOT_SUP	Function is not supported. Indicates the specific function is not supported.

These parameter constants and the data structures are defined in the header file *uadef.h.*

Note: Some of the node define verb parameter constants listed in Table 2-5 and Table 2-6 have the same underlying values. Specifically, NOT_SUP, ADD_VAR, and RPL_VAR have a value of 0. DEL_VAR and SUPPORT have a value of 1. IGN_VAR has a value of -1. Use only the documented constants for any parameter value since these equivalencies can change in later releases.

The LU define verbs use the parameter constants listed in Table 2-6.

Table 2-6 Parameter Constants for LU Define Verbs

Constant	Description
IGN_VAR	Ignore this parameter. Specifies that an optional parameter is to be ignored. The current or default value for this parameter is to be used.
SUPPORT	Function is supported. Indicates the specific function is supported.
NOT_SUP	Function is not supported. Indicates the specific function is not supported.
RPL_VAR	Replace parameter. Specifies that a related field (such as a network name) is to replace the existing value on the control block.
DEL_VAR	Delete parameter. Specifies that a related field (such as a network name) is to be deleted from a control block.
ADD_VAR	Add parameter. Specifies that a related field (such as an user ID) is to be added to a control block.

These parameter constants and the data structures are defined in the header file *ludef.h.*

Note: Before issuing a define or display verb, the transaction program uses the **attach** verb to establish a connection to *s2_schd,* the IRIS SNA SERVER.

The verbs return 0 if the control block was successfully defined or displayed and -1 if an error occurred. For errors, the global variables *snamaj* and *snamin* are set to the major code and minor code of the error. Use the verbs **dspmaj** and **dspmin** to translate these codes into printed error messages. (For a list of the major and minor codes, see Appendix A.)

Display Verbs

Display verbs return information about configuration elements for 6.2 LUs (the local LU, remote LU, modes, and transaction programs), secondary LUs, and the PU 2.0/PU 2.1 node (the lines, stations, control points, and the node itself). Table 2-7 lists the display verbs.

Table 2-7 Display Verbs

Verb	Function
dspcp	Returns addressing information for a control point defined with the **dfncp** verb.
dspcph	Returns data describing the current status of sessions with a host-type control point.
dspline	Returns information about a line defined with the **dfnline** verb.
dspllu	Returns parameter values that control Local LU operation.
dspmode	Returns parameter values that control the operation of the local LU in conjunction with a group of sessions (identified by a mode name) with a remote LU.
dspnode	Returns information stored in the Node Control Block.
dsprlu	Returns parameter values that control the operation of the local LU in conjunction with the remote LU.

Table 2-7 (continued) Display Verbs

Verb	Function
dspses	Returns information about sessions between local and remote LUs.
dspslu	Returns information about a secondary LU.
dspsta	Returns information about a station defined with the **dfnsta** verb.
dsptp	Returns parameter values that control the operation of the local LU in conjunction with a local transaction program.

Display verbs for the node use the parameter constants listed in Table 2-8.

Table 2-8 Display Verb Parameter Constants for the Node

Constant	Description
IGN_VAR	Ignore this parameter. Specifies that an optional parameter is to be ignored. The current or default value for this parameter is to be used.
SUPPORT	Function is supported. Indicates the specific function is supported.
NOT_SUP	Function is not supported. Indicates the specific function is not supported.

Table 2-8 (continued) Display Verb Parameter Constants for the Node

Constant	Description
NEXT	Display information on the next control block. The Display Verbs can return information on a specific block or the next block so that you can step through all of the blocks of a specific type. If a block is named and the next parameter is set to IGN_VAR, the information for that block is returned. If next is set to NEXT, information for the next block is returned.

These parameter constants and the data structures are defined in the header file *uadef.h.*

Note: Some of these widely used constants have the same underlying values. Specifically, NOT_SUP and NEXT have a value of 0. SUPPORT has a value of 1. IGN_VAR has a value of -1. Use only the documented constants for any parameter value since these equivalencies can change in later releases.

Display verbs for the LU use the parameter constants listed in Table 2-9.

Table 2-9 Display Verb Parameter Constants for the LU

Constant	Description
IGN_VAR	Ignore this parameter. Specifies that an optional parameter is to be ignored. The current or default value for this parameter is to be used.
SUPPORT	Function is supported. Indicates the specific function is supported.
NOT_SUP	Function is not supported.

Table 2-9 (continued) Display Verb Parameter Constants for the LU

Constant	Description
	Indicates the specific function is not supported.
NEXT	Display information on the next control block.

These parameter constants and data structures are defined in the header file *ludef.h.*

Note: Some of these widely used constants have the same underlying values. Specifically, NOT_SUP, ADD_VAR, RPL_VAR, and NEXT all have a value of 0. DEL_VAR and SUPPORT have a value of 1. IGN_VAR has a value of -1. Use only the documented constants for any parameter value since these equivalencies can change in later releases.

The verbs return 0 if the control block was successfully defined or displayed and -1 if an error occurred. If there was an error, the global variables *snamaj* and *snamin* are set to the major code and minor code of the error. Use the verbs **dspmaj** and **dspmin**, described in “Transaction Program Utility Functions,” to translate these codes into printed error messages. (For a list of the major and minor codes, see Appendix A.)

Note: Before issuing a define or display verb, the transaction program uses the **attach** verb to establish a connection to the IRIS SNA SERVER, *s2_schd.*

Node Operator Verbs

Node operator verbs activate lines, stations, LUs, and the local PU and control the node logging and message facilities. While their format is not determined by IBM, their function is implied.

The Node Operator verb function is archived in */usr/lib/libsna.a.* Programs that use verb functions must be linked against this library. The data structures used in these verbs are in */usr/include/sna/noop.h.* The global variables are defined in the header file */usr/include/sna/global.h.*

Table 2-10 lists the node operator verbs provided.

Table 2-10 Node Operator Verbs

Verb	Function
actline	Activates the line.
actlu	Activates the LU locally.
actpu	Activates the node's physical unit locally.
actsta	Instructs the node to enable the link connection to a particular remote node.
chgmsgq	Instructs the node to change the status parameters of a message queue.
dctline	Deactivates the named line from use.
dctlu	Terminates the use of the named LU.
dctpu	Terminates the use of the PU.
dctsta	Instructs the node to issue the proper commands to disable the link connection to a particular remote node.
dspmsgq	Instructs the node to display the status parameters of a message queue.
rtvnmsg	Retrieves a message from the named message queue.

The activation and deactivation requests issue a message to the Node Operator message queue indicating whether the activation or deactivation was successful. Each of these requests returns a correlation number used to retrieve the completion message from the queue.

Operation Diagnostics

Table 2-11 provides information to help you locate and solve a problem if you encounter trouble using the IRIS SNA SERVER**.**

Table 2-11 Operation Diagnostics

Problem	**Solution**
The line is active but the station is pending-active, even after the partner activates its side.	Check that the SDLC address in the station definition block matches your partner′s. If it does not and the problem persists, check for a defective cable between the system and the modem.
“Link Inoperative” message received.	Deactivate the line, then reactivate it. Consult Appendix C in the *IRIS SNA Administration Guide* for specific information about the condition that caused the line to drop.
“Link Inoperative” message received (“too many aborts”)	The duplex, NRZI, and line speed settings received must match. If they do not, reconfigure and try again. Also, check the modem’s duplex setting.
Primary is polling; secondary is responding.	The duplex, NRZI and line speed settings must match. If they do not, reconfigure and try again. Also, check your modem’s duplex setting.
Line drops immediately after activation.	Check the log. It is likely that the exchange IDs were not recognized. Contact your partner and exchange IDs. Update the local node or remote node block, regenerate the configuration, and retry.
The message ends with “CNOS allocation error.”	Partner needs to initialize session limits.

Table 2-11 (continued) Operation Diagnostics

Problem	Solution
The message states that the LU was not recognized.	Your configuration does not match your partner's mode. Modify the resource definition blocks, regenerate the configuration, and try again.
No sessions are activated.	If you specified 0 auto-initiated sessions in the mode definitiion block, sessions are not activated until needed by an application. If you did specify auto-initiated sessions, check the log for session-failure messages. Usually this problem is a mismatch of the mode parameters for CNOS support and re-initialization.

Chapter 3

Programming with the LU 0-3 Verbs

The LU 0-3 PI interface is designed to support the protocols of Logical Unit Types 0, 1, 2, and 3. Since the protocols vary greatly among these LU types, the interface provides only basic session-level support, and leaves the specifics of the Presentation Services as a program responsibility. The interface assumes the programmer is responsible for:

- Understanding SNA formats and protocols
- Managing the session
- Understanding the relationship between the program and the IRIS SNA Scheduler

SNA Formats and Protocols

The programmer must be able to:

- Format the RUs correctly
- Understand such matters as the bracket, contention, and chaining protocols so that the RH can be correctly set
- Know how to bring up and take down sessions

Note: This guide makes no attempt to serve as an introductory tutorial in SNA programming. Programmers needing such information can refer to the IBM manuals listed in the Related Documentation section of the "Introduction" to this guide.

Managing the Session

The interface leaves the initiation, takedown, and cleanup of LU-LU sessions as a responsibility of the program. Session management is discussed in more detail later in this section.

Program vs. IRIS SNA Scheduler

The program runs as a separate process from the IRIS SNA Scheduler. The interface has been designed to address the problems of timing and performance that arise because of this separation. In order to write effective programs, the programmer needs to know how this distribution of function is accomplished. For a more detailed description, see the “Verb Execution” and “Verb Completion” sections of this chapter.

Verb Execution

The verbs provide the only interface into the IRIS SNA Scheduler. Each verb accepts valid parameters, executes to completion, and returns control to the program. There is no concept of interrupting and resuming a verb's execution. Each verb has been designed, however, so that it can execute in a non-blocking fashion. Blocking is optional on the **accru, alnau, sndru, rcvru, rejru, gsync**, and **ssync** verbs; the other verbs do not block.

Each verb returns a return code and the global variables *snamaj, snamin, snamsg,* and *snastat*. The return code is set to OK if the verb executed successfully, and to NOT OK if it did not. If the return code is NOT OK, *snamaj* and *snamin* contain codes describing the error.

The *snamsg* variable indicates if a message from the IRIS SNA Scheduler is queued within the PI verbs. Any time a PI verb is issued that reads messages from the scheduler, it is possible that other messages are received that are not returned on the current verb call.

For instance, a **rcvru** verb that returns a message for the requested NAU can have read messages for other allocated NAUs before reading the desired message. If this has occurred, the messages are queued within the PI layer, and *snamsg* is set. In general, the programmer should not let messages remain in the PI layer, as there is a finite amount of data space that PI will use before it no longer can receive messages from the scheduler.

The *snastat* variable is set to show which verbs can be issued at which time. It is a simple state, showing whether the NAU is not allocated, pending allocation, or allocated. When the NAU is pending allocation, only the **rcvru** and **dalnau** verbs can be issued. When *snastat* is set, any verb for the allocated NAU can be issued, except another **alnau**.

Verb Library

Verb functions, and any functions called by the verb, are archived in */usr/sna/lib/liblu03.a.* Programs that use verb functions must be linked against this library. In addition, link programs that use SNA LU 0-3 verb functions must be linked against two other archival libraries: */usr/sna/lib/libsna.a,* which includes the IRIS SNA verb functions provided by the IRIS SNA SERVER, and */usr/lib/libbsd.a,* which provides required TCP/IP support. Under most compilers, this is done with the *-l* option, for example:

```
cc sample.c -L/usr/sna/lib -llu03 -lsna -lbsd
```

Consult your compiler documentation for instructions on linking against the product's library.

Verb Completion

The program operates as a process separate from the IRIS SNA Scheduler. The verbs communicate with the scheduler by exchanging messages over an interprocess communication channel, the nature of which varies from implementation to implementation. Verbs either execute locally, send messages to the IRIS SNA Scheduler, or send messages and expect a reply from the scheduler. Because some applications have a need to service events from sources other than the IRIS SNA Scheduler, verbs that send messages to the scheduler can execute in either a blocking or non-blocking fashion.

The following describes how verbs complete, in either the blocking or non-blocking manner. The blocking execution is easier to use, and is recommended for most applications. The non-blocking style is more complex, but is necessary for applications that service multiple sources of events. The programmer should determine which style of verb execution is appropriate to the application.

Execute Locally

The **gfsm, initpi**, and **rcvru** verbs execute locally. They do not send any message to the IRIS SNA Scheduler and return control to the application in the following manner:

- **Local**

  All processing occurs within the application. Control returns to the application immediately.

- **Receive Nowait**

  A message, if previously sent by the IRIS SNA Scheduler, is retrieved. Control returns to the application immediately, whether data is present or not.

- **Receive Wait**

  A message, if previously sent by the IRIS SNA Scheduler, is received. If not, the program will wait until the message is sent by the scheduler, or until the timer expires. Control returns to the application when a message is received or the timer expires.

Table 3-1 Illustrates the parameter settings that control verb execution.

Table 3-1 Locally Executed Verbs

Verb	Parameter	Verb	Completion	Type
			Receive	Receive
	wait	Local	Nowait	Wait
gfsm	- - -	X		
initpi	- - -	X		
rcvru	timeout value		X	
	P_NOWAIT			X
	P_WAIT			X

Note: The *wait* parameter, present only on the **rcvru** verb, can be set to a timeout value in seconds, to constant P_NOWAIT, or to P_WAIT.

Send, No Reply

The **accru, rejru**, and **sndru** verbs can execute by sending a message to the IRIS SNA Scheduler without expecting a reply from the scheduler. The **accru** and **rejru** verbs always operate in this way. The **sndru** verb operates this way if certain parameter values are set. Cases when **sndru** generates a reply from the scheduler are covered in the next section, “Send, Reply.”

These verbs can execute in a blocking or non-blocking fashion. If blocking, the verb returns control to the program when the message is accepted by the transport channel. If the transport channel is not immediately available due to congestion, the verb will wait until the congestion is cleared. The program need not concern itself with retry logic. If non-blocking, the verb completes successfully, providing the transport channel can accept the message immediately. If the transport channel cannot do this, the verb returns immediately, with an indication that the transmission should be retried.

There are two types of retry conditions requiring different recovery logic. The first is that the flow-control window with the IRIS SNA Scheduler has been exhausted. In this case, the program must issue a **rcvru** specifying P_PACE on the *loc_pac* parameter to receive the pacing response from the scheduler. No data can be sent to the scheduler until this response is received. In the second case, the flow control window is large enough for the request, but the transport channel itself is temporarily full. In this case, reissue the verb.

An application using the blocking options is simpler in operation, but must wait occasionally for verbs to complete. An application using the non-blocking options is exposed to retry conditions, but can always remain free to service other events.

These verbs return control to the application in the following manner:

- **Send No Block**

A message is sent to the IRIS SNA Scheduler. Control returns to the application immediately. The verb completes successfully if the transport channel accepted the message. The verb completes unsuccessfully if the transport channel was not able to accept the message; the minor code gives the reason for the failure.

- **Send Block**

  A message is sent to the IRIS SNA Scheduler. The verb completes successfully when the transport channel accepts the message. The verb handles any retry logic required.

Table 3-2 illustrates the parameter settings that control verb execution.

Table 3-2 Send, No Reply Verbs and Parameters

Verb		Parameters		Verb	Completion	Type
			Send	Send		
	block	*when*	*rh_ind*	No Block	Block	
accru	P_BLOCK	- - -	- - -	X		
	P_NBLOCK	- - -	- - -	X		
rejru	P_BLOCK	- - -	- - -	X		
	P_NBLOCK	- - -	- - -	X		
sndru		P_BLOCK	- - -	!LIC		X
	P_BLOCK	P_NONE	LIC,RQE		X	
	P_BLOCK	P_NONE	LIC,RQD		X	
	P_NBLOCK	- - -		LIC	X	
	P_NBLOCK	P_NONE	LIC,RQE	X		
	P_NBLOCK	P_NONE	LIC,RQD	X		

The *block* parameter is present on the **accru, rejru**, and **sndru** verbs.

The *when* parameter is present only on the **sndru** verb. It indicates to the scheduler when to construct a reply to this verb. The value P_NONE indicates that no reply should be constructed. When the message is not *last in chain* (LIC), the parameter is not applicable.

The *rh_ind* parameter is present only on the **sndru** verb. It contains the RH indicators for the RU. LIC means last in chain, RQD means request definite response, and ! represents not. When the *when* parameter is set to P_NONE, and the chain is marked RQD, the scheduler does not create a reply of its own. The application must issue a **rcvru** verb to read the response sent to a chain marked RQD.

Send, Reply

The **alnau**, **gsync**, **sndru**, and **ssync** verbs generate replies from the IRIS SNA Scheduler. The **alnau**, **gsync**, and **ssync** verbs always operate in this way. The **sndru** verb operates in this way if certain parameter values are set. (The cases when the **sndru** verb does not generate a reply from the scheduler were covered in the previous section, “Send, No Reply.”)

The scheduler generates the reply to the **gsync** and **ssync** verbs as soon as it receives the request. The reply to the **alnau** verb can be constructed as soon as the scheduler receives the request, or the reply can be generated when the requested NAU is free. The *when* parameter controls this. The reply to the **sndru** verb is constructed according to the *when* parameter and the RH indicator settings. If the *when* parameter is P_NONE, or the RU is not last in chain (LIC), no reply is generated. If the *when* parameter is P_WHEN, and the chain is marked RQE, the reply is generated as soon as the scheduler has assigned the TH sequence number to the RU. If the chain is marked RQD, the scheduler waits until the response arrives from the partner half-session before constructing the reply. In this case, the reply contains not only the TH sequence number assigned, but the value of the response from the partner half-session.

Blocking has a meaning for these verbs in addition to the considerations described in the previous section. When blocking these verbs, wait for the reply from the scheduler before returning control to the application. The values on the reply are returned to the application as returned parameters on the verb. If the verb executes non-blocking, the verb does not wait for the reply before returning control. The application must issue a **rcvru** verb to read the reply.

Note: The blocking parameter on the **alnau** and **sndru** verbs is independent of the parameters that indicate when the reply is constructed.

These verbs return control to the application in the following manner:

- **Send No Block**

  A message is sent to the IRIS SNA Scheduler. Control returns to the application immediately. The verb completes successfully if the transport channel accepted the message. The verb completes unsuccessfully if the transport channel was not able to accept the message; the minor code gives the reason for the failure. An **rcvru** verb must be issued to read the reply.

- **Send Block Short**

  A message is sent to the IRIS SNA Scheduler. The verb handles any retry logic required to send the request. The IRIS SNA Scheduler constructs the reply as soon as the request is received in a relatively short period. The verb completes when the reply is received from the scheduler. The values on the reply are presented to the program as return parameters on the verb.

- **Send Block Long**

  A message is sent to the IRIS SNA Scheduler. The verb handles any retry logic required to send the request. The IRIS SNA Scheduler constructs the reply as directed, either when the requested NAU is free (**alnau**) or when the response is received from the partner half-session (**sndru**). The time required to complete is indefinite. The verb completes when the reply is received from the scheduler. The values on the reply are presented to the program as return parameters on the verb.

Table 3-3 illustrates the parameter settings that control verb execution.

Table 3-3 Send, Reply Verbs and Parameters

Verb	**Parameters**			**Verb**	**Completion**	**Type**
	block	when	rh_ind	Send	Send	Send
				No	Block	Block
				Block	Short	Long
alnau	P_BLOCK	P_IMMED	- - -		X	
	P_BLOCK	P_WHEN	- - -			X
	P_NBLOCK	P_IMMED	- - -	X		
	P_NBLOCK	P_WHEN	- - -	X		
gsync	P_BLOCK	- - -	- - -		X	
	P_NBLOCK	- - -	- - -	X		
ssync	P_BLOCK	- - -	- - -		X	
	P_NBLOCK	- - -	- - -	X		
sndru	P_BLOCK	P_WHEN	LIC,RQE		X	
	P_BLOCK	P_WHEN	LIC,RQD			X
	P_NBLOCK	P_WHEN	LIC,RQE	X		
	P_NBLOCK	P_WHEN	LIC,RQD	X		

The *block* parameter is present on the **alnau**, **gsync**, **ssync**, and **sndru** verbs.

The *when* parameter is present only on the **alnau** and **sndru** verbs. On the **alnau** verb, P_IMMED means that the IRIS SNA Scheduler should reply as soon as it receives the request. Either the NAU is allocated or it is not available. P_WHEN means that the scheduler should queue the request until the NAU is available. On the **sndru** verb, P_WHEN indicates that the reply should be generated according to the settings on the *rh_ind* parameter.

The *rh_ind* parameter, present only on the **sndru** verb, contains the RH indicators for the RU. LIC means last in chain, RQE means request exception response, and RQD means request definite response. When the chain is marked RQE, the reply is generated when the TH sequence number is assigned. When the chain is marked RQD, the scheduler waits for the response from the partner half-session before constructing a reply.

Receiving Data and the *naustat* Variable

The IRIS SNA Scheduler sends unsolicited messages to the program. These messages are of two types: either SNA RUs from the session, or notification of an event within the IRIS SNA Scheduler. There are three varieties of notification:

- The IRIS SNA Scheduler has already responded to an SNA request and is telling the program that it has done so.
- The program is told of a non-session flow event, such as an allocation succeeding or a send check error.
- The IRIS SNA Scheduler is responding to a verb issued with the *block* parameter specified as P_NBLOCK.

Appendix B, “Supported SNA Request Units,” lists the SNA RUs and notes whether the IRIS SNA Scheduler passes them up to the program unaffected or whether the Scheduler responds to them itself and sends a notification. Appendix C, “Notification Events,” lists all the potential notifications that can be generated by the IRIS SNA Scheduler. Throughout the documentation, the term “receive data” is used when either an SNA RU or a notification can be received.

Whenever the scheduler sends these RUs or notifications to the program, it appends to the message an image of the states of the FSMs (*finite-state machines*) of the NAU. This image is the state of the FSMs *immediately after* the notification or RU in question has been processed by the scheduler. When the **rcvru** verb is executed, the notification or RU is returned to the program, and the FSM states that accompanied that message are displayed in a composite form in the *naustat* variable. The intention is to provide a simple method for a programmer to control the application in most cases.

Appendix F, “State Definition,” gives a complete listing of the various states of the *naustat* variable with suggested actions. If the programmer needs to see the complete set of FSM values, the **gfsm** verb displays those values. Note that the **gfsm** verb displays the complete image of the FSM states of the *last* notification or RU received via the **rcvru** verb. The *naustat* value is initially returned by the **alnau** verb, and is updated whenever a **rcvru** verb returns with data.

If the program issues verbs without intervening **rcvru** verbs, it is possible that the state of the session can change without the program being aware of it. For example, the program can be issuing **sndru** verbs with the RUs marked not LIC, so that no reply is coming from the scheduler. The link, meanwhile, may have become inoperative, and session-outage procedures would have brought the session down. If the program then issues a **sndru** verb marked LIC and P_NOWAIT, it sees the session-outage notification (SON) on the subsequent **rcvru**. If the program issues the **sndru** marked LIC and WAIT, however, the verb returns unsuccessfully, with a sense code indicating the error. The program, as part of its standard error recovery, should then issue **rcvru** verbs to read all the notifications and RUs that may have been sent by the scheduler. After all the messages from the scheduler have been read, the state values are re-established to reflect the correct state of the session.

Session Management

Although the program is notified of the activation and deactivation of the SSCP-PU and SSCP-LU sessions, there is no management action for these sessions that the program can initiate via the PI verbs. The program can, however, use the node operator verbs to activate the link. After the link is active, it is the host's responsibility to issue the **actpu** and **actlu** verbs. A program can terminate all sessions on the link by issuing a Node Operator verb to deactivate the link. On a host connection, the Node Operator verb to deactivate a link issues a REQDISCONT, which causes the host to bring down the sessions in an orderly fashion. If the link should fail, the IRIS SNA Scheduler initiates session-outage procedures, bringing down the sessions. Programs are notified when this occurs.

The program is responsible for managing the LU-LU session when allocating an LU. If a BIND request arrives, it is a program's responsibility to reply to it. A program has the capability to initiate session activation by sending a formatted or character-coded login to the SSCP, and responds to the subsequent flows. A program can deactivate the session by sending an RSHUTD, TERMSELF, or an UNBIND. In addition, session management can be shared among several cooperating programs. This is because deallocating an LU does not cause the session to deactivate; the session, in fact, is not affected in any way if the program deallocates the LU. This enables several programs to share the same session serially, without having to activate the session each time. This feature provides a performance boost, but the programs using this feature need to account for the state of the session when allocating the LU.

PU and LU Structure

Because the IRIS SNA Scheduler provides integrated support for PU 2.1 peer connections as well as PU 2.0 host connections, there are some differences in the way PUs are structured.

The IRIS SNA Scheduler is organized as a PU 2.1 node, which appears as a PU 2.0 node when seen from the host. As a PU 2.1 node, it can support multiple links, and, therefore, multiple SSCP-PU sessions. In this documentation, the term “PU” refers to one of these SSCP-PU sessions. When a PU is allocated, it means that the use of one of these sessions is allocated, and nothing more. Each of these PUs operates independently of the other. For instance, a **dctpu** verb received on one of these sessions resets all the LUs active on that link, but does not affect any other LU running in the IRIS SNA Scheduler.

When an LU is configured, the SSCP that controls it must be named. The DAF address for the LU must be unique for that SSCP’s LUs, but need not be unique node-wide. The local names of each LU and LU pool, however, must be unique node-wide.

An LU can also be designated as a member of a pool, a group of LUs that has been combined for use on the **alnau** verb. When a pool name is requested on the **alnau** verb, an IRIS SNA Scheduler selects one of the free LUs in the pool. The system imposes no restriction on how LUs can be grouped into pools.

The consideration should be that a user would not care which LU out of the pool was allocated when a pool name was requested on the **alnau** verb.

Integration with Other Verb Boundaries

A single program can issue verbs from all the verb types, if necessary. That is, a single program can allocate an LU to issue PI verbs, and it can also allocate and use LU 6.2 conversations. There are no particular restrictions involved in doing this. The program needs only to be aware of which resource type it is working on, and to issue the correct verbs for it.

Data Types

In the man pages, most of the parameters are defined as hex, shex, or lhex. These data types are used so that the valid values for each parameter are always the same, regardless of the actual integer size of the particular machine. Table 3-4 defines these three data types.

Table 3-4 Data Types

Data Type	Definition
hex	An unsigned integer, ranging from 0x00 to 0xFF
shex	An unsigned integer, ranging from 0x0000 to 0xFFFF
lhex	An unsigned integer, from 0x00000000 to 0xFFFFFFFF
num	A signed integer in the range of -32,768 to 32,767

The file *ddityp.h* contains the actual typed statement for each of these types. This file is automatically included in the application when *pgmin.h* is included.

Data Structures

Each member (referred to in the following documentation as both parameter and field) of the data structure can be Supplied, Returned, or Supplied/Returned.

- Supplied parameters are set by the application program.
- Returned parameters are set automatically by the successful operation of the verb.
- Supplied/Returned parameters are set by the application program when the verb is issued, but their value can change after the successful operation of the verb.

Initialization of every member of the structure is the responsibility of the application. This is particularly true of pointers, which must be set to null if they are not set to a specific address.

Supplied parameters are further specified as Required, Conditional, or Optional.

- Required parameters must be set by the application program.
- Conditional parameters can have a value required, depending on the setting of another parameter.
- Optional parameters need not be set.

Appendix A

Major and Minor Return Codes

This appendix lists and defines the return codes that may be displayed in the message line area of the Information Panel in the IRIS SNAView main window. Table A-1 lists return codes for functions that complete normally. Table A-2 lists return codes for functions that are aborted. Table A-3 lists return codes for functions that do not complete normally. Table A-4 lists return codes for functions that terminate abnormally with state errors. Table A-5 lists return codes for allocation errors. Table A-6 lists return codes for program errors. Table A-7 lists return codes for deallocation errors. Table A-8 lists return codes for node operator errors

Table A-1 Major Code 00 (S2_OK): Function Completed Normally

Major Code	Minor Code	Code Meaning
00	0000	Completed normally. Function completed normally.
00	0001	Completed as negotiated. Function completed as negotiated.
00	0003	Data available. Returned by test conversation if data has arrived on the posted conversation.
00	0004	Control information available. Returned by test conversation if control information has arrived on the posted conversation.
00	0086	Logic error.

Table A-2 Major Code 01 (S2_USAGE): Function Aborted, Usage Error

Major Code	Minor Code	Code Meaning
01	0001	Program not attached. The transaction issued another verb before issuing an **attach** verb. The program must attach before calling other verbs.
01	0002	Duplicate attach attempted. After a previous **attach** verb and before a **detach** verb, the transaction program attempted to reissue the **attach** verb.
01	0003	Invalid conversation ID (rsrc unknown). The conversation specified in the ***cnvid*** parameter is not a valid conversation.
01	0004	Context not set.
01	0009	Null structure pointer parameter passed. A verb was called with a null structure pointer as a parameter.
01	0010	Attach: path is a required parameter. The configuration file path is required for **attach** and **rattach**.
01	0011	Attach: LU name is a required parameter. The LU name is required for attach type AT_LU(0).
01	0012	Attach: Invalid attach type parameter. Valid attach types for a transaction program are AT_LU(0) AT_NODE(1).
01	0013	Attach: LU unkown. The specified LU is not configured.
01	0014	Attach: LU not available, limits reached. The LU specified in the attach is not activated, or has already
01	0015	Attached: TP name is not configured. The transaction program name specified in the request is not configured.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0016	Attach, rattach: Configuration file not authorized or invalid. Cannot authorize attach. The attach failed because the transaction program process does not have access permission to the node, the node is not active, or the **attach** verb configuration file parameter was incorrect.
01	0017	Rattach: Conversation ID required. Conversation ID required for remote attach.
01	0018	Rattach: Invalid conversation ID for rattach. The conversation specified in the remote attach is invalid or cannot be remotely attached.
01	0019	Attach, rattach: Invalid wait-time parameter. Invalid wait-time. Valid wait parameters are 0, 1, 2, ..., or -1 to specify no maximum wait time.
01	0020	Attach: not enough space. Attach rejected. Not enough space to support the transaction program.
01	0021	Attach: SNA Scheduler TP limit reached. Attach rejected. The SNA Scheduler process limit has been reached.
01	0022	Attach: Not enough space in TP.
01	0023	Attach: Pathname is too long.
01	0030	Alcnv: Partner (rlu) is a required parameter. Remote LU name is required.
01	0031	Alcnv: Mode name is a required parameter. Mode name is required.
01	0032	Alcnv: Program name (tpn) is a required parameter. Remote transaction program name is required.
01	0033	Alcnv: Partner LU (rlu) not found. Remote LU unknown. The specified remote LU is not configured for the attached LU.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0034	Alcnv: Mode not found. Mode unknown. The specified mode is not configured between the remote LU ands the attached LU.
01	0035	Alcnv: Invalid parameter (type). Invalid **alcnv** type. Valid conversation types for **alcnv** are 0 for basic conversation and 1 for mapped conversation.
01	0036	Alcnv: Mapped verb interface not allowed. Mapped conversations not available. Either the attached LU or the remote LU is not configured to support mapped conversations, or the transaction program name specified in the previous attach is not configured to support mapped conversations.
01	0037	Alcnv: Basic verb interface not allowed. Basic conversations not available. The transaction program name specified in the previous attach is not configured to support basic conversations.
01	0038	Alcnv: Invalid parameter (when). Invalid session allocation parameter. Valid session allocation parameters for **alcnv** are AC_WHEN(0), AC_DELAY(1), and AC_IMMED(2).
01	0039	Alcnv: Delayed allocation not allowed. No delayed session allocation. The attached LU is not configured to support delayed session allocation.
01	0040	Alcnv: Immediate allocation not allowed. No immediate session allocation. The attached LU is not configured to support delayed session allocation.
01	0041	Alcnv: Invalid parameter (sync). Invalid sync-level parameter. Valid sync level parameters for **alcnv** are 0 for NONE and 1 for CONFIRM processing.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0042	Alcnv: Requested sync level not allowed. Requested sync level not available. The requested sync-level support must be configured for the requested mode and for the transaction program name specified in the previous attach request.
01	0043	Alcnv: PIP not allowed. No PIP data. PIP data-support must be configured for the transaction program name specified in the previous attach request
01	0044	Alcnv: Invalid PIP length. PIP data too large. Too many PIP parameters were passed in the PIP data, or the PIP data exceeds the maximum length.
01	0045	Alcnv: Invalid parameter (sec). Invalid security-level parameter. Valid security-level parameters are SEC_NONE(0), SEC_SAME(1), and SEC_PGM(2).
01	0046	Alcnv: Requested security not allowed. Requested security level not available. The requested security-level support is not configured. SEC_SAME(1) must be configured for the attached LU and the remote LU. SEC_PGM(2) must be configured for the attached LU, the remote LU, and for the transaction program name specified in the previous attach request.
01	0047	Alcnv: Invalid format for user parameter.
01	0048	Alcnv: Invalid format for password parameter.
01	0049	Invalid mode name. Mode name SNASVCMG cannot be specified when using a mapped conversation.
01	0060	Cnfrm, cnfrmed: Conflict with conversation sync level. Sync level conflict. A **cnfrm** or **cnfrmed** verb was attempted on a conversation that was allocated sync level NONE.
01	0070	Dalcnv: Invalid parameter (type). Invalid **dalcnv** type. Valid **dalcnv** types are: DC_SYNC(1)

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning

		DC_FLUSH(2)
		DC_AB_PGM(3)
		DC_AB_SVC(4)
		DC_AB_TMR(5)
		DC_LOCAL(6)
		DC_CNFRM(7)
		Invalid **mdalcnv** types are:
		DC_AB_PGM(3)
		DC_AB_SVC(4)
		DC_AB_TMR(5)
		Valid **mdalcnv** types are:
		DC_SYNC(1)
		DC_FLUSH(2)
		DC_LOCAL(6)
		DC_CNFRM(7)
		DC_ABEND(8)
		Prprcv: Invalid parameter (type).
		Invalid **prprcv** type. Valid **prprcv** types are 0 for FLUSH and 1 for SYNC.

01	0080	Prprcv: Invalid parameter (type). Invalid **prprcv** type. Valid **prprcv** types are 0 for FLUSH and 1 for SYNC.
01	0081	Prprcv: Invalid parameter (lock).

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0090	Rcvwt, pstrct: Invalid parameter (fill). Invalid **prprcv** lock parameter. Valid **prprcv** lock parameters are 0 for Short lock, and 1 for Long lock. This parameter is significant for **prprcv** type SYNC. Invalid fill parameters for **rcvwt** and **pstrct** are RW_BUFF(0) and RW_LL(1).
01	0091	Invalid parameter (length) cannot be negative. Negative length is invalid. The length parameter is negative.
01	0100	Snddta: Invalid LL. Invalid LL field. The data passed to **snddta** contains an invalid LL field of 0x0000, 0x8000, or 0x8001.
01	0101	Snddta: Invalid LL within the data stream. PS headers not supported. indicates a PS header. This is not supported. The data passed to **snddta** contains an LL field of 0x0001, which indicates a PS header. This is not supported.
01	0102	Mapping Error: Map name not found.
01	0103	Mapping Error: Map execution failure.
01	0104	Mapping Error: Duplicate map name.
01	0110	Snderr: Invalid parameter (type). Invalid **snderr** type. Valid **snderr** types for transaction programs are SE_PGM(1) and SE_SVC(0).
01	0111	Snderr: SE_ALC(2) is reserved.
01	0120	Waitcv: Negative count value is invalid. Negative **waitcv** count is invalid.
01	0121	Waitcv: Invalid conversation ID on list. Invalid conversation for **waitcv**. The list parameter passed to **waitcv** contained an invalid conversation ID.
01	0122	Waitcv: Listed conversation ID does not have posting active.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0123	Waitcv: No conversation with posting is active. If a list of conversations was passed to **waitcv**, none of the conversations had posting active. If no list was passed, then no conversations for the TP had posting active.
01	0124	Waitcv: List size (count) must be positive. Waitcv requires list when non-zero count specified. When the **waitcv** request specifies one or more conversations in the count parameter, those conversations must be passed in the list parameter. No list parameter was passed.
01	0125	Waitcv: List size (count) too large. Waitcv list too large. The number of conversations specified in the **waitcv** request exceeds the ability of the verb interface layer to process the request.
01	0130	Testcv: Invalid parameter (type).
01	0131	Testcv: Posting not active for conversation.
01	0140	Setctx: Path required if *tcbid* is supplied.
01	0141	Setctx: Requested context not found.
01	0300	Not authorized to CO verbs. TP not authorized to the type of Control Operator verb issued.
01	0301	Requested limits are invalid. Limits requested in CNOS verb are invalid.
01	0302	Service Manager Mode not initialized. CNOS verb issued before service manager mode is initialized.
01	0310	Session limit of 0 is invalid. Session limit must be greater than 0.
01	0311	Responsible value invalid. Invalid responsible parameter.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0312	C_maxs must be greater than c_minf + c_minb. Sum of the minimum first-speaker sessions and the minimum bidder sessions cannot exceed session limit.
01	0313	Dctses: Invalid parameter (immediate).
01	0314	Rstsl: Invalid parameter (dtrg).
01	0315	Rstsl: Invalid parameter (dsrc).
01	0316	Rstsl: Invalid parameter (force).
01	0320	Next session identifier not found.
01	0321	Invalid session identifier.
01	0400	Name must be less than 20 characters. Node Operator: String parameter too long. Maximum name length is 20 characters.
01	0401	Line is a required parameter. Node Operator: Line name is required.
01	0402	LU is a required parameter. Node Operator: LU name is required.
01	0403	PU is a required parameter. Node Operator: PU name is required.
01	0404	Actsta: Invalid parameter (dial). Node Operator: Invalid dial parameter. Valid dial parameters are 0, 1, 2.
01	0410	Rspque must be NULL. Node Operator: Response queue name unsupported this release.
01	0411	Chgmsgq: Severity must be between 0 and 99.
01	0501	Invalid local LU. The fully qualified LU name of the local LU specified on the **dfnllu** is invalid.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0502	Invalid remote LU. The value specified in the local LU session-limit parameter of the **dfnllu** verb is less than the sum of the currently defined LU mode session limits.
01	0503	Invalid mode.
01	0504	Invalid ALS.
01	0505	Invalid Send Pacing Window.
01	0506	Invalid Receive Pacing Window.
01	0507	Invalid Max RU Upper Bound.
01	0508	Invalid Max RU Lower Bound.
01	0509	Invalid Sync Level option.
01	0510	Invalid Single Session Reinitialization option.
01	0511	Invalid Cryptography option.
01	0512	Invalid Maximum Number Sessions. Parallel-session support (YES) is specified on the **dfnrlu** verb but the local LU session limit is 1.
01	0513	Invalid Minimum Number First Speaker. Parallel-session support (YES) and CNOS support (NO) are specified on the **dfnrlu** verb.
01	0514	Invalid Minimum Number First Prebound. CNOS support (YES) and parallel-session support (NO) is not specified on the **dfnrlu** verb.
01	0516	Invalid Blank Mode option.
01	0517	Invalid Network Name operator.
01	0518	Invalid Network Qualifier operator.
01	0519	Invalid Network Name.
01	0520	Invalid Network Qualifier.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0521	Invalid Init Type.
01	0522	Invalid Parallel Session option. The fully qualified LU name of the remote LU specified on the **dfnmode** verb is not currently defined for the local LU.
01	0523	Invalid CNOS ALS.
01	0524	Invalid LU Password.
01	0525	Invalid Security Acceptance option.
01	0526	Invalid Password operation.
01	0527	Invalid LU Session Limits.
01	0528	Invalid Conversation Security.
01	0529	Invalid Security operation.
01	0530	Invalid User ID.
01	0531	Invalid Password.
01	0532	Invalid Profile.
01	0533	Invalid Wait.
01	0534	Invalid Max Number of TPs.
01	0535	Invalid LU ID Number.
01	0536	Invalid TP Name.
01	0537	Invalid Status.
01	0538	Invalid Conversation Type.
01	0539	Invalid Security Required.
01	0540	Invalid PIP option.
01	0541	Invalid PIP Number.
01	0542	Invalid PIP check.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0543	Invalid Data Mapping.
01	0544	Invalid FMH.
01	0545	Invalid Privilege.
01	0546	Invalid LUW Indicator.
01	0550	Invalid LU.
01	0551	Invalid Partner.
01	0552	Invalid ALS.
01	0553	Not enough space to complete operation.
01	0554	Invalid Remote PU.
01	0555	Invalid Line.
01	0556	Logic error: LSCB not found.
01	0557	Dfnsta: Associated Line not inactive.
01	0558	Logic error: ALCB not found.
01	0560	Dfnmode: Invalid Parameter.
01	0561	Dfnmode: Lower Bound exceeds Upper Bound.
01	0562	Dfnmode: Single Session Reinit not compatible with Partner.
01	0564	Defend: Minf cannot be greater than maxs.
01	0565	Dfnmode: Minpf cannot be greater than minf.
01	0566	Dfnmode: Maxs not compatible with partner.
01	0567	Dfnmode: Blank mode already exists.
01	0570	Dfnrlu: Cnosals and parallel session support not compatible.
01	0571	Dfnrlu: Parallel session support not compatible with maxs.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0574	Dfnrlu: Invalid reinit option for parallel-session support.
01	0575	Dfnrlu: CNOS also unknown.
01	0576	Dfnrlu: Null net name already exists.
01	0577	Dfnrlu: Parallel support not compatible with PU type.
01	0580	Logic error: drcb for local PU does not exist.
01	0581	Dfnllu: Session limits exceed LU session limits specified.
01	0582	Dfnllu: Network name specified after initsl.
01	0583	Dfnllu: Security parameters to be deleted not found.
01	0584	Dfnllu: LU ID already specified.
01	0585	Dfnllu: LU ID must be specified when lucb is initialized.
01	0586	Dfnllu: LU ID cannot be updated when session limit initialized.
01	0590	Dfnllu: Security access parameters conflict with security required.
01	0591	Dfnllu: Security parameters to be deleted not found.
01	0600	Too many resources defined—no internal address available.
01	0601	Invalid Exchange ID.
01	0602	Invalid Master Device.
01	0603	Invalid Monitor Timer.
01	0604	Invalid NOOP Messages.
01	0605	Invalid LOG Messages.
01	0606	Invalid Debug Messages.
01	0607	Invalid PU Name.
01	0608	Invalid CPU ID.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0609	Invalid Line Name.
01	0610	Invalid Line Type.
01	0611	Invalid Device Name.
01	0612	Invalid SDLC Role.
01	0613	Invalid Connection Type.
01	0614	Invalid NRZI.
01	0615	Invalid Half-duplex.
01	0616	Invalid Max BTU.
01	0617	Invalid Max Retries.
01	0618	Invalid Idle.
01	0619	Invalid Nonprod Rcv Time.
01	0620	Invalid Max I-Frames.
01	0621	Invalid Rate.
01	0622	Invalid SDLC Address.
01	0623	Invalid Phone Name.
01	0624	Dfnsta: Exceeded maximum number of stations (8).
01	0626	Dfnline: Multidrop specified and role not primary.
01	0627	Dfnsta: One station already defined for leased line.
01	0628	Dfnline: Line already exists.
01	0630	Dfnsta: Line not specified.
01	0631	Dfnsta: PU not specified.
01	0632	Dfnsta: Station already exists.
01	0634	Dspline: Invalid line name.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0635	Dspline: End of line list.
01	0636	Dspline: No lines defined.
01	0637	Dspcp: Invalid control point name.
01	0638	Dspcp: End of PU list.
01	0639	Dspcp: No PUs defined.
01	0640	Dltcbl: Mode unknown.
01	0641	Dltcbl: TP Name unknown.
01	0642	Dltcbl: Object is in use.
01	0646	Dltcbl: Local LU name not specified.
01	0647	Dltcbl: Remote LU name not specified.
01	0648	Dltcbl: No parameters specified.
01	0650	Dfnrpu: Neither XID or CPID specified.
01	0651	Dfnrpu: Both XID and CPID specified.
01	0659	Control point is not a host.
01	0660	Dspllu: Userid invalid.
01	0661	Dspllu: End of security list.
01	0662	No userid found.
01	0663	Dspllu: Invalid profile.
01	0664	Dspllu: Invalid LU name.
01	0665	Dspllu: End of LU list.
01	0666	Dspllu: No LU defined.
01	0667	Dsprlu: Invalid Remote LU name.
01	0668	Dsprlu: No more Remote LUs.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	0669	Dsprlu: No Remote LUs defined.
01	0670	Dspmode: Invalid mode name.
01	0671	Dspmode: End of mode list.
01	0672	Dspmode: No modes defined.
01	0673	Dsptp: Invalid TP name.
01	0674	Dsptp: End of TP list.
01	0675	Dsptp: No TP defined.
01	0676	Dsptp: No Network Name.
01	0677	Invalid NEXT parameter value.
01	0691	Control point name is required.
01	0692	No more remote LUs or secondary LUs.
01	0693	LU_ID value does not match any LU.
01	0830	User crash code not described.
01	0850	*s2_schd* is not active. The SNA Scheduler is not active.
01	0860	*s2_schd* has terminated abnormally. The SNA Scheduler is no longer active. The transaction program has been detached.
01	0870	Time out—request not recovered.
01	0880	Time out—request recovered.
01	0890	Time out—message queue full.
01	0900	Verb logic error. An internal logic error occurred while processing the verb.
01	0901	Duplicate *s2_tpi*. Duplicate TPI attach attempted.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	1000	Invalid NAU name. The NAU must be no greater than eight characters and all characters must be of symbol-string Type A.
01	1001	Invalid NAU type. Valid values are P_PU and P_LU.
01	1002	Invalid when parameter. Valid values for **sndru** are P_WHEN and P_NONE. Valid values for **alnau** are P_WHEN and P_IMMED.
01	1003	Invalid block parameter. Valid values are P_BLOCK and P_NBLOCK.
01	1004	Invalid bind parameter. Buffer length is greater than zero and no bind pointer supplied.
01	1005	Resources temporarily unavailable. Retry.
01	1006	Not enough TP space to allocate another NAU.
01	1008	Not enough resources. An attempt was made to send an RU that exceeds the maximum send size when **sndru** is issued in non-blocking mode.
01	1009	Null structure pointer supplied to a PI verb.
01	1010	Invalid structure type supplied for **initpi**.
01	1011	NAU identifier invalid. The NAU identifier supplied on the verb call is not the ***nauid*** of a currently allocated NAU.
01	1012	Specified NAU not allocated. An attempt was made to use the ***nauid*** of an NAU that is pending allocation.
01	1013	Invalid svcid. Service identifier is not one of the valid values or is inconsistent with the NAU type specified on the **alnau** verb.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	1014	The pointer to the pru structure is NULL. This is a required field.
01	1015	The RU command is zero.
01	1016	The pointer to the RU buffer is NULL. In this verb call, the RU buffer is required.
01	1017	The pointer to the ufsm structure is NULL. This is a required field for the **gfsm** verb.
01	1018	RU sense data is required. Zero is an invalid value.
01	1019	The flow parameter of the **ssync** verb is invalid. Valid values are P_SND, P_RCV, and P_FLOW.
01	1020	The wait parameter is invalid. Valid values are P_WAIT, P_NOWAIT, or a positive numeric value representing the time, in seconds, that the verb should wait.
01	1021	The loc_pac parameter is invalid. This is an optional field, but if specified, valid values are P_PACE and P_NPACE.
01	1022	Required UFSM pointer is NULL. The pointer to the ***ufsm*** structure is NULL. This field is required on the **rcvru** verb and on the **alnau** verb issued with block = P_BLOCK.
01	1023	svcid inconsistent with NAU type. The ***svcid*** parameter is compatible with the nau type of the specified NAU.
01	1024	The **dfnslu** verb was issued while LU 0-3 was not configured by the configuration manager.
01	1025	LU not defined. A secondary LU with the name specified in ***naunm*** is no currently defined.

Table A-2 (continued) Major Code 01 (S2_USAGE): Function Aborted, Usage

Major Code	Minor Code	Code Meaning
01	1026	LU not available. The LU specified on the **alnau** verb is already allocated.
01	1027	PU not defined. A PU with the name specified in ***naunm*** is not currently defined.
01	1028	PU not available. The PU specified on the **alnau** verb is already allocated.
01	1030	Attach type is not specified. The Transaction Program that issued the verb did not attach with attach type AT_ANODE or AT_ALU.
01	1031	Send check. The reply to the **sndru** verb received from the SNA Scheduler indicated a send check.
01	1032	Negative response. The reply to the **sndru** verb received from the SNA Scheduler indicated a negative response.
01	1033	Receive check. The reply to the **sndru** verb received from the SNA Scheduler
01	1034	Session lost.

Table A-3 Major Code 02 (S2_UNSUC): Completed Unsuccessfully

Major Code	Minor Code	Code Meaning
02	0001	No information for conversation.
02	0002	Conversation not posted.
02	0003	Requested-to-send not received.

Table A-4 Major Code 03 (S2_STATE): Function Aborted, State Error

Major Code	Minor Code	Code Meaning
03	0001	Conversation State error. Request is illegal in the current conversation state.
03	0002	Logical Record State error. Request is illegal because the current logical record has not been completed.
03	0003	Conv for waitcv not S2_RECV. A conversation specified in the **waitcv** request is not in receive state.

Table A-5 Major Code 05 (S2_ALCER): Allocation Error

Major Code	Minor Code	Code Meaning
05	0001	Remote TP not available—no retry. The transaction program could not be started on the remote system because of a lack of resources, which is not temporary. Retry is not suggested.
05	0002	Remote TP not available—retry. The transaction program could not be started on the remote system because of a temporary resource shortage. Retry is suggested.
05	0003	Remote LU does not support conversation type. The remote LU does not support the requested conversation type.
05	0004	PIP data not supported by remote. Program initialization data is not supported by the remote LU.
05	0005	PIP specification error. Program initialization data was specified incorrectly.
05	0006	Invalid access security information. Security information was not specified correctly.

Table A-5 (continued) Major Code 05 (S2_ALCER): Allocation Error

Major Code	Minor Code	Code Meaning
05	0007	Remote program does not support requested sync level.
05	0008	Remote TP not recognized by remote LU. The transaction program requested was not recognized at the remote LU.
05	0009	Allocation failure—no retry.
05	0010	Allocation failure—retry.
05	0011	First Speaker session not available. The session is not immediately available.
05	0012	Local resource allocation failure.
05	0013	Cannot authorize security access.
05	0022	Allocation rejected by Resource Manager.
05	0099	Unknown sense code data received.

Table A-6 Major Code 07 (S2_PGMER): Program Error

Major Code	Minor Code	Code Meaning
07	0001	Program Error: Current logical record not truncated. Partner has reported a program error; the current logical record was not truncated.
07	0002	Program Error: Current logical record truncated. Partner has reported a program error; current logical record was truncated.
07	0003	Program Error: Data may have been purged. Partner has reported a program error; current logical record may have been purged.
07	0011	Service Error: Current logical record not truncated. Service Transaction Program has reported an error; current logical record was not truncated.

Table A-6 Major Code 07 (S2_PGMER): Program Error

Major Code	Minor Code	Code Meaning
07	0012	Service Error: Current logical record was truncated. Service Transaction Program has reported an error; current logical record was truncated.
07	0013	Service Error: Data may have been purged. Service transaction program has reported a program error; current logical record may have been purged.
07	0020	Mapping Error: FMH not supported.
07	0021	Mapping Error: Mapping not supported.

Table A-7 Major Code 09 (S2_DEALC): Deallocated

Major Codew	Minor Code	Code Meaning
09	0000	Normal deallocation by partner. Partner has deallocated the conversation
09	0001	Abnormal deallocation by partner. Partner has terminated the conversation abnormally.
09	0002	Abnormal deallocation by service program. A service transaction program has terminated the conversation abnormally.
09	0003	Abnormal deallocation verb time-out. Time-out has occurred.
09	0004	Abnormal deallocation, Session Failure. The session has failed.
09	0005	Abnormal deallocation by partner.
10	0063	Limits not zero.
10	0064	Requested limits exceed configuration. This error should be interpreted as ACTIVATION_FAILURE_RETRY.

Table A-7 (continued) Major Code 09 (S2_DEALC): Deallocated

Major Codew	Minor Code	Code Meaning
10	0065	Sum of minimums exceed maximum session. The sum of the minimum first-speaker and bidder sessions exceeds the requested maximum.
10	0066	Invalid SNASVCMG limits. SNASVCMG limits must be 2: 1 bidder and 1 first speaker.
10	0067	SNASVCMG mode not initialized.
10	0068	Mode limits are closed.
10	0069	Chgsl not valid for this mode.
10	0070	SNASVCMG mode not reset, other modes still not reset. SNASVCMG mode cannot be reset because user modes are still open.
10	0071	CNOS race with remote—remote won.
10	0072	Partner does not recognize mode.
10	0073	CNOS is in process locally.
10	0074	CNOS allocation error.
10	0075	CNOS resource failure. The SNASVCMG session with the partner LU either could not be started or failed.
10	0076	Insufficient space for a new session.
10	0077	Partner LU is not active.
10	0078	Modes incompatible: session not started.

Table A-8 Major Code 11 (S2_NPERR): Node Operator Error

Major Code	Minor Code	Code Meaning
11	0001	Node operator verb failure.
11	0002	Node operator verb not recognized.

Table A-8 (continued) Major Code 11 (S2_NPERR): Node Operator Error

Major Code	Minor Code	Code Meaning
11	0003	Attach of type AT_NODE. Transaction Program must attach with type AT_NODE to use node operator verbs.
11	0010	Link name not recognized.
11	0011	Adjacent link station not recognized.
11	0012	Physical unit not recognized.
11	0013	Earlier request still active.
11	0014	Dial in or dial out required.
11	0015	Logical unit name not recognized.
11	0016	Message queue name not recognized.
11	0017	Message queue not enabled.
11	0018	No message in message queue.
11	0100	Actpu failure.
11	0110	Actlu failure.
11	0120	Dctpu failure.
11	0130	Dctlu failure.
11	0140	Chgmsgq failure.
11	0150	Dspmsgq failure.
11	0190	Rtvnmsg failure.

Appendix B

Supported SNA Request Units

This appendix lists all request units (RUs) that can be sent or received by a transaction program that uses IRIS SNA LU 0-3. The RUs can be sent as requests, positive responses, or negative responses.

In most cases, the value assigned to the constants represent the corresponding SNA-defined request code. The following constants were assigned different values so that each constant has a unique value: P_NSPE, P_INTSLF, P_TRMSLF, P_RECFMS, P_REQMS.

Sending RUs

When sending request RUs, use the **sndru** verb and if a positive or negative response is sent, use the **accru** or **rejru** verb, respectively. Table B-1 shows the *ru_cmd* parameter and whether RU is a request or response. The *ru* parameter is specified as being Required, Not Required, or Optional.

Table B-1 ru_cmd and ru Parameters for Sending RUs

ru_cmd		**ru**
P_BID	Response	Not Required
P_BIND	Response	Optional
P_CANCEL	Request	Not Required
P_CHASE	Request	Not Required
P_CHASE	Response	Not Required
P_DAT	Response	Not Required
P_DAT	Request	Optional (***ru_len***=0)
P_ECHTST	Response	Not Required

Table B-1 (continued) ru_cmd and ru Parameters for Sending RUs

ru_cmd		ru
P_INTSLF	Request	Required
P_LUSTAT	Request	Required
P_NMVT	Request	Required
P_NMVT	Response	Not Required
P_NOTIFY	Request	Required
P_QC	Request	Not Required
P_QEC	Request	Not Required
P_RECFMS	Request	Required
P_RELQ	Request	Not Required
P_REQECH	Request	Required
P_REQMS	Response	Not Required
P_RQR	Request	Not Required
P_RSHUTD	Request	Not Required
P_RTR	Request	Not Required
P_SHUTC	Request	Not Required
P_SIGNAL	Request	Required
P_SIGNAL	Response	Not Required
P_TRMSLF	Request	Required
P_UNBIND	Request	Not Required

Receiving RUs

The receiving RUs can be requests or responses. The type is indicated in the *ru_cmd* and *wht_rcv p*arameters. Respond to the *wht_rcv* types of P_FMH, FMH, P_DATA, and P_SNA by using the **accru** or **rejru** verb. The layout of the *wht_rcv t*ype P_NOTIF is found in Appendix C, "Notification Events." Table B-2 shows the receiving RU parameters.

Table B-2 The ru_comd and wht_rcv Parameters for Sending RUs

ru_cmd		**wht_rcv**
P_ACTLU	Request	P_NOTIF
P_ACTPU	Request	P_NOTIF
P_BID	Request	P_SNA
P_BIND	Request	P_SNA
P_CANCEL	Request	P_NOTIF
P_CANCEL	Response	P_NRSP/P_PRSP
P_CHASE	Request	P_SNA
P_CHASE	Response	P_NRSP/P_PRSP
P_CLEAR	Request	P_NOTIF
P_DACTLU	Request	P_NOTIF
P_DACTPU	Request	P_NOTIF
P_DAT	Request	P_DATA/P_FMH
P_DAT	Response	P_NRSP/P_PRSP
P_ECHTST	Request	P_SNA
P_INTSLF	Response	P_NRSP/P_PRSP
P_LUSTAT	Request	P_NOTIF
P_LUSTAT	Response	P_NRSP/P_PRSP
P_NMVT	Request	P_SNA
P_NMVT	Response	P_NRSP/P_PRSP
P_NOTIFY	Request	P_NOTIF
P_NOTIFY	Response	P_NRSP/P_PRSP
P_NSPE	Request	P_NOTIF

Table B-2 (continued) The ru_comd and wht_rcv Parameters for Sending RUs

ru_cmd		wht_rcv
P_QC	Request	P_NOTIF
P_QC	Response	P_NRSP/P_PRSP
P_QEC	Request	P_NOTIF
P_QEC	Response	P_NRSP/P_PRSP
P_RECFMS	Response	P_NRSP/P_PRSP
P_RELQ	Request	P_NOTIF
P_RELQ	Response	P_NRSP/P_PRSP
P_REQECH	Response	P_NRSP/P_PRSP
P_REQMS	Request	P_SNA
P_RQR	Response	P_NRSP/P_PRSP
P_RSHUTD	Response	P_NRSP/P_PRSP
P_RTR	Response	P_NRSP/P_PRSP
P_SDT	Request	P_NOTIF
P_SHUTC	Response	P_NRSP/P_PRSP
P_SHUTD	Request	P_NOTIF
P_SIGNAL	Request	P_SNA
P_SIGNAL	Response	P_NRSP/P_PRSP
P_STSN	Request	P_NOTIF
P_TRMSLF	Response	P_NRSP/P_PRSP
P_UNBIND	Request	P_NOTIF
P_UNBIND	Response	P_NRSP/P_PRSP

Appendix C

Notification Events

An **rcvru** verb can be completed because of events reported as notifications. When a notification occurs, the *wht_rcv p*arameter of the **rcvru** verb is set to P_NOTIF and the type of notification is reported in the *ru_cmd* parameter. The notification can be of three types:

- An asynchronous response message to a verb issued by the transaction program
- A special event, such as a P_EN_PRG indication
- An SNA request that has been responded to by the IRIS SNA Scheduler, where only the event is reported

Valid Notifications

Table C-1 Valid Notifications

Valid ru_cmd	Description
P_ACTLU	An ACTLU RU has been received and responded to.
P_ACTPU	An ACTPU RU has been received and responded to.
P_ALNAU	An **alnau** reply message.
P_BETB	A between-bracket notification.
P_CANCEL	A CANCEL RU has been received and responded to.
P_CLEAR	A CLEAR RU has been received and responded to.

Table C-1 (continued) Valid Notifications

Valid ru_cmd	Description
P_DACTLU	A DACTLU RU has been received and responded to.
P_DACTPU	A DACTPU RU has been received and responded to.
P_EN_PRG	A complete chain has been purged.
P_GSYNC	A **gsync** reply message.
P_LUSTAT	An LUSTAT RU has been received and responded to.
P_NOTIFY	A NOTIFY RU has been received and responded to.
P_NSPE	An NSPE RU has been received and responded to.
P_QC	A QC RU has been received and responded to.
P_QEC	A QEC RU has been received and responded to.
P_RCVCHK	The IRIS SNA Scheduler detected a receive check error.
P_RELQ	A RELQ RU has been received and responded to.
P_SDT	An SDT RU has been received and responded to.
P_SNDCHK	A send check error was detected by the IRIS SNA
	Scheduler.
P_SHUTD	A SHUTD RU has been received and responded to.
P_SNDRU	A **sndru** reply message.
P_SSYNC	An **ssync** reply message.

Table C-1 (continued) Valid Notifications

Valid ru_cmd	Description
P_STSN	An STSN RU has been received and responded to.
P_UNBIND	An UNBIND RU has been received and responded to.

Layout of the rcvru Verb

The layout of the **rcvru** verb for each valid event follows. Only the *pru* structure parameters are described.

P_ACTLU

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_ACTLU
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the ACTLU response
ru	The ACTLU response RU

Meaning:The ACTLU request has been received from the host. If a "cold" activation is executed, the LU-LU half-session is terminated and the LU enters the P02_NAU state. If an "ERP" activation is executed, the LU-LU half-session is not affected and the LU remains in the current state. Applicable only if LU services are allocated.

P_ACTPU

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size

ru_cmd	P_ACTPU
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the ACTPU response
ru	The ACTPU response RU

Meaning: The ACTPU request has been received and responded to and the SSCP-PU half-session is active. Applicable only if PU services are allocated.

P_ALNAU

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	Length of information in the *ru* parameter
ru_cmd	P_ALNAU
sense	Not applicable
ru_seq	Not applicable
rh_ind	Not applicable
ru	contains:
	byte 0 snamaj
	byte 2 snamin
	byte 4 *bind_len* (meaningful only if LU services)
	byte 6 *bind* (meaningful only if LU services)

Meaning:The asynchronous reply to the allocation request is returned in this notification.

P_BETB

loc_cor	Not applicable
buf_len	Not applicable
ru_len	Not applicable

ru_cmd	P_BETB
sense	Not applicable
ru_seq	Not applicable
rh_ind	Not applicable
ru	Not applicable

Meaning: Signals that the session is between brackets.

P_CANCEL

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_CANCEL
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the CANCEL response
ru	The CANCEL response RU

Meaning: A CANCEL request has been received and responded to by the IRIS SNA Scheduler. No more RUs of the cancelled RU chain are received after this notification. Applicable only if LU services are allocated.

P_CLEAR

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_CLEAR
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the CLEAR response returned

ru	The CLEAR response RU

Meaning: A CLEAR request has been received and responded to. The LU enters the P05_NAU state. Applicable only if LU services are allocated.

P_DACTLU

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_DACTLU
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the DACTLU request
ru	The DACTLU request RU

Meaning: The DACTLU request has been received from the host and responded to by the IRIS SNA Scheduler. The LU enters the P01_NAU state. Applicable only if LU services are allocated.

P_DACTPU

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_DACTPU
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the DACTPU response
ru	The DACTPU response RU

Meaning: The DACTPU request has been received and responded to and the SSCP-PU half-session is reset. Applicable only if PU services are allocated.

P_EN_PRG

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	0
ru_cmd	P_EN_PRG
sense	Not applicable
ru_seq	Not applicable
rh_ind	Not applicable
ru	Not applicable

Meaning: An RU chain that has been rejected by the user has been purged up to the end of the chain by the IRIS SNA Scheduler. Applicable only if LU services are allocated.

P_GSYNC

loc_cor	The correlation ID returned on the **gsync** verb
buf_len	Supplied parameter
ru_len	Length of information in *ru* parameter
ru_cmd	P_GSYNC
sense	Not applicable
ru_seq	Not applicable
rh_ind	Not applicable
ru	contains:
	byte 0 snamaj
	byte 2 snamin
	byte 4 *snd_seq*
	byte 6 *snd_res*
	byte 8 *snd_upd*
	byte 10 *rcv_seq*

byte 12	*rcv_res*
byte 14	*rcv_upd*

Meaning:This is the asynchronous reply message to a **gsync** verb issued with a P_NBLOCK parameter. The sync-point information is returned in the *gsync_ds* of the *ru*. Applicable only if LU services are allocated.

P_LUSTAT

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_LUSTAT
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the LUSTAT request
ru	The LUSTAT request RU

Meaning:The LUSTAT request has been received and responded to by the IRIS SNA Scheduler. Applicable only if LU services are allocated.

P_NOTIFY

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_NOTIFY
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the NOTIFY request
ru	The NOTIFY request RU

Meaning:The NOTIFY request has been received from the host and responded to by the IRIS SNA Scheduler. Applicable only if LU services are allocated.

P_NSPE

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_NSPE
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the NSPE request
ru	The NSPE request RU

Meaning:The NSPE request has been received from the host after receipt of a positive INITSELF or TERMSELF response. The RU has been responded to by the IRIS SNA Scheduler. Applicable only if LU services are allocated.

P_QC

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_QC
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the QC response
ru	The QC response RU

Meaning:The QC request has been received and responded to. The LU enters a state where no more normal flow data is received. Applicable only if LU services are allocated.

P_QEC

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_QEC
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the QEC response
ru	The QEC response RU

Meaning:The QEC request has been received and responded to by the IRIS SNA Scheduler. The transaction program is asked to stop sending normal flow data at the end of the current RU chain. Upon completion of the current RU chain, the transaction program should send a QC request. Applicable only if LU services are allocated.

P_RCVCHK

loc_cor	Correlation ID returned on the **accru** or **rejru** verb
buf_len	Supplied parameter
ru_len	Length of information in the *ru* parameter
ru_cmd	P_RCVCHK
sense	Receive check
ru_seq	Not applicable
rh_ind	Not applicable
ru	snamaj snamin

Meaning:A receive check was detected by the IRIS SNA Scheduler.

P_RELQ

loc_cor	Not applicable

buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_RELQ
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the RELQ response
ru	The RELQ response RU

Meaning:The RELQ request has been received and responded to by the IRIS SNA Scheduler. The transaction program may start sending normal flow data. Applicable only if LU services are allocated.

P_SDT

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_SDT
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the SDT response
ru	The SDT response RU

Meaning:An SDT request has been received and responded to by the IRIS SNA Scheduler. The LU enters the P16_NAU state. Applicable only if LU services are allocated.

P_SNDCHK

loc_cor	Correlation ID returned on the **accru**, **rejru**, or **sndru** verb
buf_len	Supplied parameter
ru_len	Length of information in *ru* parameter
ru_cmd	P_SNDCHK

sense	Send check
ru_seq	Not applicable
grh_ind	Not applicable
ru	snamaj snamin

Meaning:A send check was detected by the IRIS SNA Scheduler.

P_SHUTD

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_SHUTD
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the SHUTD response
ru	The SHUTD response RU

Meaning:A SHUTD request has been received and responded to as the first step in an orderly takedown of the LU-LU half-session. The transaction program is asked to stop sending normal flow data when convenient and to return a SHUTC request when finished. Applicable only if LU services are allocated.

P_SNDRU

loc_cor	Correlation ID returned on the **sndru** verb
buf_len	Supplied parameter
ru_len	Length of information in the *ru* parameter
ru_cmd	P_SNDRU
sense	Sense code if a negative response is received
ru_seq	Sequence number assigned to the RU sent

rh_ind	Not applicable
ru	contains:
	byte 0 snamaj
	byte 2 snamin

Meaning:This is the asynchronous reply message to an **sndru** verb issued with *block* set to P_NBLOCK and *when* set to P_WHEN.

P_SSYNC

loc_cor	Correlation ID returned on the **ssync** verb
buf_len	Supplied parameter
ru_len	Length of information in the *ru* parameter
ru_cmd	P_SSYNC
sense	Not applicable
ru_seq	Not applicable
rh_ind	Not applicable
ru	contains:
	byte 0 snamaj
	byte 2 snamin

Meaning:This is the asynchronous reply message to an **ssync** verb issued with *block* set to P_NBLOCK. Applicable only if LU services are allocated.

P_STSN

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_STSN
sense	Not applicable
ru_seq	Sequence number of the RU

rh_ind	Indicators of the STSN response
ru	The STSN response RU

Meaning:Host-initiated resynchronization of send and receive sync-point sequence numbers has been executed via the exchange of STSN request and response. The result of the resynchronization may be retrieved via the **gsync** verb. Applicable only if LU services are allocated.

P_UNBIND

loc_cor	Not applicable
buf_len	Supplied parameter
ru_len	RU size
ru_cmd	P_UNBIND
sense	Not applicable
ru_seq	Sequence number of the RU
rh_ind	Indicators of the UNBIND request
ru	The UNBIND request RU

Meaning:An UNBIND request has been received and responded to by the IRIS SNA Scheduler. The LU-LU half-session is terminated. The LU enters the P02_NAU state. Applicable only if LU services are allocated.

Appendix D

LU Finite-State Machines

To be able to fully control the SSCP-LU and LU-LU half-sessions of an LU in all details, the LU finite-state machine ***fsm_lu*** may be retrieved by the application via the **gfsm** verb. The ***fsm_lu*** contains most of IBM's FAPL-defined finite-state machines of the Data Flow Control and Transmission Control layers.

The ***fsm_lu*** is a structure with the following members:

hex	***cp_sess***	DFC SSCP-LU session
		status
hex	***cp_im_s***	DFC SSCP-LU **immediate request** mode send
hex	***lu_sess***	DFC LU-LU session **status status**
hex	***chain_r***	DFC LU-LU chain receive
hex	***chain_s***	DFC LU-LU chain send
hex	***qec_r***	DFC LU-LU quiesce **receive**
hex	***qec_s***	DFC LU-LU quiesce **send**
hex	***rtr_fsp***	DFC LU-LU ready to

hex	***cp_sess***	DFC SSCP-LU session receive first speaker
hex	***shutd_r***	DFC LU-LU shutdown receive
hex	***lu_im_s***	DFC LU-LU immediate request mode send
hex	***dt_r***	TC LU-LU data traffic receive
hex	***rqr_s***	TC LU-LU request recovery send
hex	***stsn_r***	TC LU-LU set and test sequence numbers receive
hex	***pac_s***	TC LU-LU pacing send

cp_sess is the finite-state machine used to control the state of the secondary SSCP-LU half-session. The following states are maintained by this FSM:

P01_CPLU Reset.

P02_CPLU An ACTLU has been received but not responded to.

P03_CPLU An ACTLU has been positively responded to.

P04_CPLU A DACTLU (not SON) has been received but not responded to.

P05_CPLU A DACTLU (SON) has been received but not responded to.

cp_im_s is the finite-state machine used to enforce the immediate request mode protocol for the normal flow of the SSCP-LU session. The following states are maintained by this FSM:

P01_CPIM	Not in chain, no definite response required.
P02_CPIM	Definite response required, CANCEL not sent.
P03_CPIM	In chain.
P04_CPIM	Response received while in chain.
P05_CPIM	Definite response to CANCEL required.

lu_sess is the finite-state machine used to control the state of the secondary LU-LU half-session. This state and the remainder of the states in this structure refer to the LU_LU session. The following states are maintained by this FSM:

P01_LULU	Reset.
P02_LULU	A BIND has been received but not responded to.
P03_LULU	A BIND has been positively responded to.
P04_LULU	An UNBIND (not SON) has been received but not responded to.
P05_LULU	An UNBIND has been sent but not responded to.
P06_LULU	An UNBIND has been both sent and received but not responded to.
P07_LULU	An UNBIND (SON) has been received but not responded to.

chain_r is used to enforce the chaining protocol for received chains. The following states are maintained by this FSM:

P01_CHNR	Between chain.
P02_CHNR	In chain.
P03_CHNR	Purge up to end of chain.

chain_s is used to enforce the chaining protocol for sending chains. The following states are maintained by this FSM:

P01_CHNS	Between chain.

P02_CHNS	In chain.

qec_r is used to enforce the quiesce protocol for the half-session that received the Quiesce-at-End-of-Chain (QEC) request. The following states are maintained by this FSM:

P01_QECR	QEC not received or not yet responded to.
P02_QECR	Positive response sent to QEC.
P03_QECR	QC sent.

qec_s is used to enforce the quiesce protocol for the half-session that sent the QEC request. The following states are maintained by this FSM:

P01_QECS	QEC not sent or no QEC-response received.
P02_QECS	Positive response received to QEC
P03_QECS	QC received.

rtr_fsp is used to enforce the Ready-to-Receive (RTR) portion of the bracket protocol for the first speaker. The following states are maintained by this FSM:

P01_RTRF	Transmission of RTR not required.
P02_RTRF	RTR should be sent.

shutd_r is used to enforce the shutdown protocol for the half-session that receives the Shutdown (SHUTD). The following states are maintained by this FSM:

P01_SHDR	No SHUTD received or no SHUTD-response sent.
P02_SHDR	Positive response sent to SHUTD.
P03_SHDR	Positive SHUTC-response received.

lu_im_s is used to enforce the immediate request mode protocol for normal flow. The following states are maintained by this FSM:

P01_LUIM	Reset.
P02_LUIM	Sent RQD, no cancel.
P03_LUIM	In chain.

P04_LUIM	In chain, response received.
P05_LUIM	Sent RQD, cancel.

dt_r is used to record the ability for data to flow on the LU-LU session. The following states are maintained by this FSM:

P01_DTR	Reset.
P02_DTR	An SDT is being processed.
P03_DTR	All traffic may flow.
P04_DTR	A CLEAR is being processed.

rqr_s records if a Request Recovery (RQR) request has been sent. The following states are maintained by this FSM:

P01_RQRS	No outstanding RQR.
P02_RQRS	RQR outstanding.

stsn_r records the receipt of the Set and Test Sequence Numbers (STSN) request. The following states are maintained by this FSM:

P01_STSN	No outstanding STSN.
P02_STSN	STSN outstanding.

pac_s records the ability to send a session-level pacing request for send pacing. The following states are maintained by this FSM:

P01_PACS	A pacing request may be sent.
P02_PACS	Pacing request sent but no response received.

Appendix E

PU Finite-State Machines

The PU finite-state machines returned by the **gfsm** verb structure ***fsm_pu*** holds the current state of some of the finite-state machines used in the PU.

The ***fsm_lu*** is a structure with the following members:

hex	***sess***	SSCP-PU session status
hex	***im_rq_s***	Immediate request mode send

sess is used to record the secondary SSCP-PU half-session state. The following values are defined:

P01_CPPU	Reset.
P02_CPPU	An ACTPU has been received but not responded to.
P03_CPPU	An ACTPU has been positively responded to.
P04_CPPU	A DACTPU (not SON) has been received but not responded to.
P05_CPPU	A DACTPU (SON) has been received but not responded to.

im_rq_s is used to enforce the immediate request mode protocol. The following states are maintained:

P01_IMRQ	Not in chain, no definite response required.
P02_IMRQ	Definite response required, CANCEL not sent.
P03_IMRQ	In chain.
P04_IMRQ	Response received while in chain.
P05_IMRQ	Definite response to CANCEL required.

Appendix F

State Definition

The status of an allocated NAU is maintained in the *naustat* variable, which contains valid information if the *snastat* variable indicates that the NAU is allocated. The *naustat* variable is returned on the **alnau**, **gfsm**, and **rcvru** verbs in the ufsm structure.

The values assigned to the *naustat* constants are organized in an increasing order that can be used by the application in testing for a specific condition. For example, data traffic on the LU-LU session is not allowed if *naustat* is less than P06_NAU. Even though more than one state can be valid at the same time, the state contained in the *naustat* variable is considered to be the most important of the valid states. The *naustat* is sufficient for most transaction programs to perform the actions required by SNA protocol.

The information presented in this appendix describes valid states for LU and PU services, depending on which was selected at allocation time.
A recommended action is provided for each state.

Table F-1 lists the valid states for LU services.

Table F-1 Valid States for LU Services

State	Description
P01_NAU	Pending ACTLU
P02_NAU	SSCP-LU half-session active
P03_NAU	Pending BIND
P04_NAU	Pending BIND response
P05_NAU	Data traffic reset
P06_NAU	Pending UNBIND
P07_NAU	Pending QC and receive flow quiesced
P08_NAU	Pending QC
P09_NAU	Pending SHUTC and send flow quiesced
P10_NAU	Pending SHUTC and receive flow quiesced
P11_NAU	Pending SHUTC
P12_NAU	Send and receive flow quiesced
P13_NAU	Send flow quiesced
P14_NAU	Receive flow quiesced
P15_NAU	Pending RTR
P16_NAU	Data traffic allowed

Table F-2 lists the valid states for PU services.

Table F-2 Valid States for PU Services

State	Description
P01_NAU	Pending ACTPU
P02_NAU	SSCP-PU half-session active

Naustat: LU Services

P01_NAU ACTLU has not been received. Neither the SSCP-LU nor the LU-LU session is active.

Recommended Action: Wait for the notification of the ACTLU via an **rcvru** verb.

P02_NAU The SSCP-LU half-session has been activated by an ACTLU request from the host. The transaction program may send data on the SSCP-LU half-session, and LU-LU session initiation may start. The transaction program may send an INITSELF request or a character-coded login to activate the LU-LU half-session, or it may wait for the host to start the session via a BIND request.

Recommended Action: Send an INITSELF RU or character-coded login to the SSCP via the **sndru** verb or wait for the BIND RU via a **rcvru** verb if automatic login by the host.

Note: If a previous login has failed, delay the next try to avoid streaming logins to the host.

P03_NAU The user has initiated the establishment of an LU-LU half-session by sending an INITSELF request and a positive response has been received from the host. A BIND request is expected from the host that would change the state to P04_NAU.

Recommended Action: Wait for the BIND RU via an **rcvru** verb.

P04_NAU A BIND request has been received from the host. If the transaction program sends a positive BIND response, the LU-LU half-session enters the P05_NAU state. If a negative BIND response is sent, the P02_NAU state is entered.

Recommended Action: Accept or reject the received BIND RU via the **accru** or **rejru** verb. If TS Profile 4 and sync points are used, the sync-point numbers should be set before responding to the BIND. This is done via the **ssync** verb. When finished, wait for the state to change into data traffic allowed via the **rcvru** verb.

P05_NAU The LU-LU half-session has been established. This is done when a transaction program sends a positive response to the BIND request with the **accru** verb. This state is also entered when a CLEAR request is received from the host.

Recommended Action: Wait for the STSN RU (if TS Profile 4 is used) and the SDT RU notification via the **rcvru** verb.

P06_NAU The user has initiated the termination of an LU-LU half-session by sending a TRMSLF or RSHUTD request. An UNBIND request is expected from the host that would transfer the LU-LU half-session into the P02_NAU state.

Recommended Action: Wait for the UNBIND RU via the **rcvru** verb.

P07_NAU The receive flow is quiesced and a QEC request has been received from the host. The transaction program is asked to stop sending normal-flow requests at end of chain.

Recommended Action: If a chain is currently being sent by the transaction program, it should be completed. When the between-chain state is reached, a QC RU should be sent to the host via a **sndru** verb.

P08_NAU A QEC request has been received from the host. The transaction program is asked to stop sending normal-flow requests at end of chain.

Recommended Action: Complete chains currently being sent by the transaction program. When the between-chain state is reached, a QC RU should be sent to the host via a **sndru** verb.

P09_NAU The send flow is quiesced and a SHUTD request has been received from the host. A SHUTC request should be sent to the host as soon as possible.

Recommended Action: A SHUTC RU should be sent to the host via a **sndru** verb.

P10_NAU The receive flow is quiesced and a SHUTD request has been received from the host as a first step in an orderly termination of the LU-LU half-session. The transaction program is asked to stop sending normal-flow requests when convenient. The send flow becomes quiesced when the transaction program sends the SHUTC request to the host.

Recommended Action: The transaction program should send a SHUTC RU when convenient via a **sndru** verb.

P11_NAU A SHUTD request has been received from the host as a first step in an orderly termination of an LU-LU half-session. The transaction program is asked to stop sending normal-flow requests when convenient. The send flow is quiesced when the transaction program sends the SHUTC request to the host.

Recommended Action: The transaction program should send a SHUTC RU when convenient via a **sndru** verb.

P12_NAU Both the send and receive flows are quiesced. No normal-flow data may be sent or received on the LU-LU half-session in this state. This state is changed, for example, when the RELQ request is either sent or received.

Recommended Action: The transaction program can send a RELQ RU to the host via a **sndru** verb if the receive quiesce state can be released. Otherwise, wait for a RELQ notification from the host via a **rcvru** verb.

P13_NAU The send flow is quiesced. Either a QEC request has been received from the host and responded to by QC or a SHUTD request has been received from the host and responded to by SHUTC. No normal-flow data may be sent by the transaction program on the LU-LU half-session in this state.

Recommended Action: Wait for a RELQ notification from the host via a **rcvru** verb.

P14_NAU The receive flow is quiesced. A QEC request has been sent by the transaction program and responded to by a QC request from the host. No normal-flow requests may be sent by the host in this state.

Recommended Action: The sending of data to the host can continue but the receive quiesce state should be released when possible in order to get normal-flow requests from the host. The quiesce is released when a RELQ RU is sent to the host via a **sndru** verb.

P15_NAU A BID request has been rejected by the transaction program with the sense code 0814 (RTR forthcoming). The transaction program should send an RTR request when ready to receive data from the host.

Recommended Action: When the transaction program is ready to receive data from the host, an RTR RU should be sent via the **sndru** verb.

P16_NAU This state indicates that data traffic is allowed. The transaction program may now send data on the LU-LU half-session.

Recommended Action: In this state the transaction program can exchange data with the host via the **sndru** and **rcvru** verbs.

Naustat: PU Services

P01_NAU The SSCP-PU session is inactive and no FMD NS (MA) RUs may be exchanged with the host. This is the normal state after IPL.

Recommended Action: Wait for the notification of the ACTPU via a **rcvru** verb.

P02_NAU The SSCP-PU session has been activated by an ACTPU request from the host. FMD NS (MA) RUs may be exchanged with the host.

Recommended Action: The transaction program can exchange data with the CNMA in the host via **sndru** and **rcvru** verbs.

Appendix G

API Verb Catalog

This appendix lists all of the API verbs in alphabetical order. These verbs and their supporting man pages are used with this guide (referred to as *SNA* in the table) and the *IRIS SNA LU 6.2 Programming Guide* (referred to as *LU 6.2* in the table).

Verb	Verb's Full Name	Verb Type	Guide
accru	Accept Request Unit	PI	SNA
actline	Activate Line	Node Operator	SNA
actlu	Activate Logical	Node Operator Unit	SNA
actpu	Activate Physical Unit	Node Operator	SNA
actses	Activate Session	Session Control	LU 6.2
actsta	Activate Station	Node Operator	SNA
alcnv	Allocate	Basic Conversation	LU 6.2
alnau	Allocate NAU	PI	SNA
atoe	ASCII to EBCDIC Translation	Implementation-specific	SNA
attach	Local Attach	Implementation-specific	SNA
chgmsgq	Change Message Queue	Node Operator	SNA
chgsl	Change Session Limit	CNOS	LU 6.2
cnfrm	Confirm	Basic Conversation	LU 6.2
cnfrmed	Confirmed	Basic Conversation	LU 6.2
dalcnv	Deallocate	Basic Conversation	LU 6.2

Verb	Verb's Full Name	Verb Type	Guide
dalnau	Deallocate NAU	PI	SNA
dctline	Deactivate Line	Node Operator	SNA
dctlu	Deactivate Logical Unit	Node Operator	SNA
dctpu	Deactivate Physical Unit	Node Operator	SNA
dctses	Deactivate Session	Session Control	LU 6.2
dctsta	Deactivate Station	Node Operator	SNA
detach	Detach	Implementation-specific	SNA
dfncp	Define Control Point	Define	SNA
dfnline	Define Line	Define	SNA
dfnllu	Define Local LU	Define	SNA
dfnmode	Define Mode	Define	SNA
dfnnode	Define Node	Define	SNA
dfnrlu	Define Remote LU	Define	SNA
dfnslu	Define Secondary LU	Define	SNA
dfnsta	Define Station	Define	SNA
dfntp	Define Transaction Program	Define	SNA
dltcbl	Delete LU Control Block	Define	SNA
dltcbu	Delete Control Block	Define	SNA
dspcp	Display Control Point	Display	SNA
dspcph	Display Host Control Point	Display	SNA
dspline	Display Line	Display	SNA
dspllu	Display Local LU	Display	SNA
dspmaj	Display Major Code	Implementation-specific	SNA
dspmin	Display Minor Code	Implementation-specific	SNA

Verb	Verb’s Full Name	Verb Type	Guide
dspmode	Display Mode	Display	SNA
dspmsgq	Display Message Queue	Node Operator	SNA
dspnode	Display Node	Display	SNA
dsprlu	Display Remote LU	Display	SNA
dspses	Display Session	Display	SNA
dspslu	Display Secondary LU	Display	SNA
dspsta	Display Station	Display	SNA
dsptp	Display Transaction Program	Display	SNA
etoa	EBCDIC to ASCII Translation	Implementation-specific	SNA
flush	Flush	Basic Conversation	LU 6.2
getatr	Get Attributes	Basic Conversation	LU 6.2
gfsm	Get Finite State Machine	PI	SNA
gsync	Get Sync Point	PI	SNA
gtype	Get Type	Type-independent Conversation	LU 6.2
initcbl	Initialize LU Definition Structure	Define	SNA
initcbu	Initialize Node Definition Structure	Define	SNA
initpi	Initialize Verb Data Structure	PI	SNA
initsl	Initialize Session Limit	CNOS	LU 6.2
malcnv	MC Allocate	Mapped Conversation	LU 6.2

Verb	Verb’s Full Name	Verb Type	Guide
mcnfrm	MC Confirm	Mapped Conversation	LU 6.2
mdalcnv	MC Deallocate	Mapped Conversation	LU 6.2
mflush	MC Flush	Mapped Conversation	LU 6.2
mgetatr	MC Get Attributes	Mapped Conversation	LU 6.2
mprprcv	MC Prepare to Receive	Mapped Conversation	LU 6.2
mpstrct	MC Post on Receipt	Mapped Conversation	LU 6.2
mrcvim	MC Receive Immediate	Mapped Conversation	LU 6.2
mrcvwt	MC Receive and Wait	Mapped Conversation	LU 6.2
mrqssnd	MC Request to Send	Mapped Conversation	LU 6.2
msnddta	MC Send Data	Mapped Conversation	LU 6.2
msnderr	MC Send Error	Mapped Conversation	LU 6.2
mtestcv	MC Test	Mapped Conversation	LU 6.2
procsl	Process Session Ldimit	CNOS	LU 6.2
prprcv	Prepare to Receive	Basic Conversation	LU 6.2
prtnmsg	Print Node Message	Implementation-specific	SNA
pstrct	Post on Receipt	Basic Conversation	LU 6.2
rattach	Remote Attach	Implementation-specific	SNA
rcvim	Receive Immediate	Basic Conversation	LU 6.2
rcvru	Receive Request Unit	PI	SNA
rcvwt	Receive and Wait	Basic Conversation	LU 6.2
rejru	Reject Request Unit	PI	SNA
rqssnd	Request to Send	Basic Conversation	LU 6.2
rstsl	Reset Session Limit	CNOS	LU 6.2
rtvnmsg	Retrieve Node Message	Node Operator	SNA
setctx	Set Context	Implementation-specific	SNA

Verb	Verb’s Full Name	Verb Type	Guide
snddta	Send Data	Basic Conversation	LU 6.2
snderr	Send Error	Basic Conversation	LU 6.2
sndru	Send Request Unit	PI	SNA
ssync	Set Sync Point	PI	SNA
testcv	Test	Basic Conversation	LU 6.2
waitcv	Wait	Type-independent Conversation	LU 6.2

Appendix H

Man Pages

This appendix contains the following category (3X) man pages related to the IRIS SNA SERVER. The man pages are organized in alphabetical order.

Programmable Interface

- accru (3X)
- alnau (3X)
- dalnau (3X)
- gfsm (3X)
- gsync (3X)
- initpi (3X)
- rcvrn (3X)
- rejru (3X)
- sndru (3X)
- ssync (3X)

SNA SERVER Implementation Specific

- attach (3X)
- detach (3X)
- rattach (3X)
- setctx (3X)

TP Utility Function

- atoe (eX)

- dspmaj (3X)
- dspmin (3X)
- etoa (3X)
- prtnmsg (3X)

Configuration/Define

- dfncp (3X)
- dfnline (3X)
- dfnllu (3X)
- dfnmode (3X)
- dfnnode (3X)
- drnrlu (3X)
- dfnslu (3X)
- dfnsta (3X)
- dfntp (3X)
- dltcbl (3X)
- dltcbu (3X)
- initcbl (3X)
- initcbu (3X)

Configuration/Display

- dspcp (3X)
- dspcph (3X)
- dspline (3X)
- dspllu (3X)
- dspmode (3X)
- dspnode (3X)

- dsprlu (3X)
- dspses (3X)
- dspslu (3X)
- dspsta (3X)
- dsptp (3X)

Node Operation

- actline (3X)
- actlu (3X)
- actpu (3X)
- actsta (3X)
- chgmsgq (3X)
- dctline (3X)
- dctlu (3X)
- dctpu (3X)
- dctsta (3X)
- dspmsgq (3X)
- rtvnmsg (3X)

Index